FUJITSU SEMICONDUCTOR
CONTROLLER MANUAL

CM71-10104-6EA

FR50

32-BIT MICROCONTROLLER

MB91360 Series

HARDWARE MANUAL

FUJITSU

FR50

32-BIT MICROCONTROLLER

MB91360 Series

HARDWARE MANUAL

FUJITSU LIMITED

PREFACE

■ Objectives and Intended Readership

The MB91360 was developed as a product in the FR50 series of 32-bit, single-chip microcontrollers. The MB91360 contains a RISC architecture CPU core and is suitable for embedded applications requiring a high level of CPU processing power.

This manual describes the functions and operation of the MB91360 and is written for engineers who are developing actual products using MB91360 family products. Refer to the "*FR Family Instruction Manual*"for further information on the MB91360 instruction set.

Note: FR: Fujitsu RISC

■ Configuration of This Manual

This manual consists of the following chapters:

CHAPTER 1 MB91360 DESCRIPTION

This chapter provides an overview of the MB91360 and describes key information such as the features, block diagram, and function description.

CHAPTER 2 CPU

This chapter describes key information about the functions of the FR50 series CPU core including the architecture, specifications, and instructions.

CHAPTER 3 INSTRUCTION CACHE

This chapter describes the instruction cache memory included in MB91360 family members and it operation. This only applies to MB91FV360GA.

CHAPTER 4 BOOT ROM / CONFIGURATION REGISTER

This chapter describes the functionality of the embedded boot ROM which is used in single chip mode.

CHAPTER 5 CLOCK GENERATION AND DEVICE STATES

This chapter describes details about generation and control of the clock used to control the MB91360. In addition device states and low power modes are explained. This chapter assumes operation without subclock. For a description of subclock operation see the corresponding chapter.

CHAPTER 6 CLOCK MODULATOR

This chapter provides an overview of the Clock Modulator and its features. It describes the register structure and operation of the Clock Modulator.

CHAPTER 7 I/O PORTS

This chapter provides an overview of I/O ports, lists the registers, and describes conditions for using external pins as I/O ports.

CHAPTER 8 EXTERNAL BUS INTERFACE

This Chapter describes in detail basic information about the external bus interface, the register structure and functions, bus operation basics and bus timing.

CHAPTER 9 INTERRUPT CONTROLLER

This Chapter provides an overview of the interrupt controller, describes the register structure and functions, and describes the interrupt controller operation

CHAPTER 10 EXTERNAL INTERRUPT/NMI CONTROLLER

This chapter provides an overview of the external interrupt/NMI controller, describes the register structure and functions, and describes the operation of the external interrupt/NMI controller.

CHAPTER 11 DMA CONTROLLER (DMAC)

This Chapter provides an overview of the DMA controller (DMAC), describes the register structure and functions, and describes the operation of the DMA controller.

CHAPTER 12 MODULES FOR OS SUPPORT

This Chapter provides an overview of the delayed interrupt and the bit search module and describes their register structure and functions, operation, and save and restore processing for the search module.

CHAPTER 13 PWM TIMER

This chapter provides an overview of the PWM timer, describes the register structure and functions, and describes the operation of the PWM timer.

CHAPTER 14 A/D CONVERTER

This chapter provides an overview of the A/D converter, describes the register structure and functions, and describes the operation of the A/D converter.

CHAPTER 15 16-BIT RELOAD TIMER

This chapter provides an overview of the 16-bit reload timer, describes the register structure/ functions, and describes the operation of the 16-bit reload timer.

CHAPTER 16 CAN CONTROLLER

This Chapter provides an overview of the CAN Interface, describes the register structure and functions, and describes the operation of the CAN Interface.

CHAPTER 17 D/A CONVERTER

This Chapter provides an overview of the D/A converter, describes the register structure and functions, and describes the operarton of the D/A converter.

CHAPTER 18 100 kHz I2C INTERFACE

This section describes the functions and operation of the MB91360 series basic I2C interface. This interface allows operation up to 100 kHz and 8-bit-addressing.

CHAPTER 19 400 kHz I2C INTERFACE

This section describes the functions and operation of the fast I2C interface.

CHAPTER 20 16-BIT I/O TIMER

The MB91360 Series contains two 16-bit free-running timer modules, two output compare modules, and two input capture modules and supports four input channels and four output channels.

CHAPTER 21 ALARM COMPARATOR

This chapter provides an overview of the Alarm Comparator (also called Under/Overvoltage Detection), describes the register structure and functions, and describes the operation of the Alarm Comparator.

CHAPTER 22 POWER DOWN RESET

This Chapter provides an overview of the Power Down Reset, describes the register structure and functions, and describes the operation of the Power Down Reset Module

CHAPTER 23 SERIAL I/O INTERFACE (SIO)

This Chapter provides an overview of the Serial I/O Interface (SIO), describes the register structure and functions, and describes the operation of the SIO.

CHAPTER 24 SOUND GENERATOR

This Chapter provides an overview of the Sound Generator, describes the register structure and functions, and describe the operation of the Sound Generator.

CHAPTER 25 STEPPER MOTOR CONTROLLER

This Chapter provides an overview of the Stepper Motor Control Module, describe the register structure and functions, and described the operation of the Stepper Motor Control Module.

CHAPTER 26 U-TIMER

The U-timer (U-TIMER) is a 16-bit timer used to generate the baud rate for the UART. This chapter provides an overview of the U-timer, describes the register structure and functions, and describes the operation of the U-timer.

CHAPTER 27 UART

The UART is a serial I/O port for performing asynchronous (start bit synchronization) communications. This chapter provides an overview of the UART, describes the register structure and functions, and describes the operation of the UART.

CHAPTER 28 USART WITH LIN-FUNCTIONALITY

This chapter explains the functions and operation of USART. The USART with LIN (Local Interconnect Network) - Function is a general-purpose serial data communication interface for performing synchronous or asynchronous communication with external devices.

CHAPTER 29 REAL TIME CLOCK

This Chaper provides an overview of the Real Time Clock (also called Watchtimer), describes the register structure and functions , and describes the operation of RTC module.

CHAPTER 30 SUBCLOCK

This section describes the functions and operation of the subclock

CHAPTER 31 32kHz CLOCK CALIBRATION UNIT

The 32kHz Clock Calibration Module provides possibilities to calibrate the 32kHz oscillation clock with respect to the 4MHz oscillation clock. This chapter gives an overview of the calibration unit, describes the registers and provides some application notes.

CHAPTER 32 FLASH MEMORY

MB91360 devices feature 256 KB, 512 KB or 768 KB of embedded flash memory. It is connected to the F-bus.

CHAPTER 33 EDSU

The Embedded Debug Support Unit (EDSU) is a module which enables basic in circuit debugging support functions on single chip FLASH MCU devices.

CHAPTER 34 ELECTRICAL SPECIFICATION

This Chapter provides information on maximum ratings, operating conditions and AC

specifications.

Appendices

The appendices contain details that could not be included in the main text and reference information for programming. These include the “I/O Map, ” “Interrupt Vectors, ” “Pin States in Each CPU State, and “Instructions”.

■ Trademarks

FR is an abbreviation for FUJITSU RISC controller, and is a product of Fujitsu, Ltd.

- The contents of this document are subject to change without notice. Customers are advised to consult with FUJITSU sales representatives before ordering.
- The information and circuit diagrams in this document are presented as examples of semiconductor device applications, and are not intended to be incorporated in devices for actual use. Also, FUJITSU is unable to assume responsibility for infringement of any patent rights or other rights of third parties arising from the use of this information or circuit diagrams.
- The products described in this document are designed, and manufactured as contemplated for general use, including without limitation, ordinary industrial use, general office use, personal use, and household use, but are not designed, developed and manufactured as contemplated (1) for use accompanying fatal risks or dangers that, unless extremely high safety is secured, could have a serious effect to the public, and could lead directly to death, personal injury, severe physical damage or other loss (i.e., nuclear reaction control in nuclear facility, aircraft flight control, air traffic control, mass transport control, medical life support system, missile launch control in weapon system), or (2) for use requiring extremely high reliability (i.e., submersible repeater and artificial satellite).
  Please note that Fujitsu will not be liable against you and/or any third party for any claims or damages arising in connection with above-mentioned uses of the products.
- Any semiconductor devices have an inherent chance of failure. You must protect against injury, damage or loss from such failures by incorporating safety design measures into your facility and equipment such as redundancy, fire protection, and prevention of over-current levels and other abnormal operating conditions.
- If any products described in this document represent goods or technologies subject to certain restrictions on export under the Foreign Exchange and Foreign Trade Law of Japan, the prior authorization by Japanese government will be required for export of those products from Japan.

© 2002 FUJITSU LIMITED Printed in Japan

CONTENTS

CHAPTER 1 MB91360 DESCRIPTION

This chapter provides an overview of the MB91360 and describes key information such as the features, block diagram, and function description.

1.1 MB91360 FEATURES

The Fujitsu MB91360 is a standard microcontroller containing a range of I/O peripherals and bus control functions. The MB91360 features a 32-bit RISC CPU (FR50 series) core and is suitable for embedded control applications requiring high-performance and high-speed CPU processing.
The MB91360 also contains up to 4 KByte instruction cache memory and other internal memories to improve the execution speed of the CPU.

■ Features

- Execution time: up to 15.6 ns (64 MHz) - device dependent
- FR50 series CPU: RISC architecture

  The CPU has a general-purpose register architecture with improved numeric implementation whereby a wide range of delayed branch instructions reduces losses in execution time due to pipeline breaks.

  Bit manipulation instructions and memory access instructions have been enhanced resulting in improved code efficiency and execution speed for control implementation.

  - A five-stage pipeline structure provides high-speed processing (one instruction per cycle)
  - 32-bit linear address space: 4 Gbytes
  - Fixed 16-bit instruction size (basic instructions)
  - High-speed multiplication/step division
  - High-speed interrupt processing (6 cycles)
  - General-purpose registers: 16 × 32 bits
- External bus interface unit with a wide range of functions

  Divides the external memory space into a maximum of eight areas. Chip select signal setting, data bus width selection (8, 16, 32-bit), and area size can be specified for each area.

  - Address bus up to 32 bit wide
  - Programmable auto-wait function
- Internal instruction cache

  The MB91360 contains up to 4-KByte instruction cache to improve the execution speed of external programs.

  - Two-way set associative caching
- DMAC

  Direct memory access (DMA) can be used to perform various types of data transfer without going via the CPU. This improves system performance.

  - Eight channels (including up to 3 external channels)
  - Four transfer modes supported: single/block, burst, continuous transfer, and fly-by
- Power consumption control mechanisms

The MB91360 contains a number of functions for controlling the operating clock to reduce power consumption.

- Software control: Sleep and stop/real time clock functions
- Hardware control: Hardware standby function
- Gear (divider) function: The CPU and peripheral clock frequencies can be set independently.

● Contains a range of peripheral functions

- UART, U-timer, USART with LIN functionality
- Real Time Clock (with optional subclock operation and subclock calibration module)
- Stepper Motor Control
- Sound Generator
- Serial IO (SIO), SIO-Prescaler
- Power Down Reset
- Alarm Comparator
- IO-Timer
- I^2C Interface
- 10 Bit D/A Converter
- CAN Interface
- 10-bit A/D converter
- 16-bit reload timer
- 16-bit PWM timer
- Watchdog timer
- Bit search module
- Interrupt controller
- External interrupt inputs
- I/O port function

● Interrupt levels

- 16 maskable interrupt levels

● Power Supply

5 V power supply is used, the internal regulator creates internal supply of 3.3V

● Packages

MB91FV360GA uses a PGA401 package, MB91F362GB will be delivered in a QFP208 package, while MB91F369GA will be delivered in a QFP160 package.

MB91F364G, MB91F365GB, MB91F366GB, MB91F367GB, MB91F368GB, MB91366GA and MB91F376G will be delivered in a QFP120 package.

See also section “PACKAGE DIMENSIONS” on page 8.

1.2 MB91360 PRODUCT LINEUP

This section shows the products of the MB91360 family and lists their functions and memory sizes.

■ MB91360 Product Lineup

Table 1.2a Product Lineup MB91FV360, MB91F362GB, MB91F364G, MB91F369GA

Ressource Channels Memory Size	MB91FV360GA	MB91F362GB	MB91F364G	MB91F369GA
Cache/Instruction RAM	4 KB / 4 KB	- / 4 KB	- / -	-/4 KB
D-bus RAM	16 KB	12 KB	12 KB	16 KB
F-bus RAM	16 KB	4 KB	4 KB	16 KB
Flash/ROM on F-Bus	512 KB	512 KB	256 KB	512 KB
Boot ROM	2 KB	2 KB	2 KB	2 KB
EDSU	-	-	1	-
CAN	4 ch	3 ch	1 ch	2 ch
Stepper Motor Control	4 ch	4 ch	-	-
Sound Generator	1 ch	1 ch	-	1 ch
PPG	8 ch	8 ch	4 ch	4 ch
Input Capture	4 ch	4 ch	4 ch	-
Output Compare	4 ch	4 ch	4 ch	-
Free Running Timer	2 ch	2 ch	2 ch	-
D/A Converter	2 ch	2 ch	2 ch	-
A/D Converter	16 ch	16 ch	12 ch	10 ch
I^2C 100kHz I^2C 400kHz	1 ch	1 ch 1 ch (1)	- 1 ch	1 ch 1 ch (1)
Alarm Comparator	1 ch	1 ch	-	1 ch
SIO/SIO prescaler	2 ch	2 ch	1 ch	2 ch
UART/U-Timer	3 ch	3 ch	1 ch	1 ch
USART with LIN function	-	-	2 ch	-

Table 1.2a Product Lineup MB91FV360, MB91F362GB, MB91F364G, MB91F369GA

Ressource Channels Memory Size	MB91FV360GA	MB91F362GB	MB91F364G	MB91F369GA
16-bit Reload Timer	6 ch	6 ch	3 ch	6 ch
Ext. Interrupt	8 ch	8 ch	8 ch	8 ch
Non maskable Interrupt	1	-	1	-
Real Time Clock	1	1	1	1
32 kHz subclock option for RTC	yes	no	yes	no
subclock calibration	yes	no	yes	no
LED port	8 bit	8 bit	8 bit	-
Power down Reset	1	1	-	1
Bit search Module	1	1	1	1
Watchdog timer	1	1	1	1
Ext. Address Bus	32 bit	21 bit	-	up to 24 bit
Ext. Data Bus	32 bit	32 bit	-	32 bit
Ext. DMA	3 ch	1 ch	-	1 ch
Max. operating frequency (for device specific limitations see datasheets)	64 MHz	64 MHz	64 MHz	64 MHz

Table 1.2b Product Lineup MB91F365GB, MB91F366GB, MB91F367GB, MB91F368GB

Ressource Channels Memory Size	MB91F365GB	MB91F366GB MB91366GA	MB91F367GB	MB91F368GB	MB91F376G
Cache/Instruction RAM	-/4 KB	-/4 KB	-/4 KB	-/4 KB	-/4 KB
D-bus RAM	16 KB	16 KB	16 KB	16 KB	16 KB
F-bus RAM	16 KB	16 KB	16 KB	16 KB	16 KB
Flash/ROM on F-Bus	512 KB	512 KB	512 KB	512 KB	768 KB
Boot ROM	2 KB	2 KB	2 KB	2 KB	2 KB
EDSU	-	-	-	-	-
CAN	2 ch	2 ch	2 ch	2 ch	2 ch

Table 1.2b Product Lineup MB91F365GB, MB91F366GB, MB91F367GB, MB91F368GB

Ressource Channels Memory Size	MB91F365GB	MB91F366GB MB91366GA	MB91F367GB	MB91F368GB	MB91F376G
Stepper Motor Control	4 ch	4 ch	-	-	4 ch
Sound Generator	1 ch	1 ch	-	-	1 ch
PPG	8 ch	8 ch	4 ch	4 ch	8 ch
Input Capture	4 ch	4 ch	4 ch	4 ch	4 ch
Output Compare	2 ch	2 ch	2 ch	2 ch	2 ch
Free Running Timer	2 ch	2 ch	2 ch	2 ch	2 ch
D/A Converter	2 ch	-	-	-	-
A/D Converter	8 ch	8 ch	8 ch	8 ch	8 ch
I^2C 100kHz I^2C 400kHz	1 ch	1 ch	1 ch	1 ch	1 ch
Alarm Comparator	1 ch	1 ch	1 ch	1 ch	1 ch
SIO/SIO prescaler	2 ch	2 ch	2 ch	2 ch	2 ch
UART/U-Timer	2 ch	2 ch	1 ch	1 ch	2 ch
USART with LIN function	-	-	-	-	-
16-bit Reload Timer	6 ch	6 ch	3 ch	3 ch	6 ch
Ext. Interrupt	8 ch	8 ch	8 ch	8 ch	8 ch
Non maskable Interrupt	-	-	-	-	-
Real Time Clock	1	1	1	1	1
32 kHz subclock option for RTC	no	yes	no	yes	yes
subclock calibra-tion	no	yes	no	yes	yes
LED port	-	-	-	-	-
Power down Reset	1	1	1	1	1
Bit search Module	1	1	1	1	1
Watchdog timer	1	1	1	1	1
Ext. Address Bus	-	-	-	-	-
Ext. Data Bus	-	-	-	-	-
Ext. DMA	-	-	-	-	-

Table 1.2b Product Lineup MB91F365GB, MB91F366GB, MB91F367GB, MB91F368GB

Ressource Channels Memory Size	MB91F365GB	MB91F366GB MB91366GA	MB91F367GB	MB91F368GB	MB91F376G
Max. operating frequency (for device specific limitations see datasheets)	64 MHz	64 MHz	64 MHz	64 MHz	64 MHz

1.3 PACKAGE DIMENSIONS

This section lists the packages for the devices of the MB91360 family.

For package dimensions please consult the Fujitsu Semiconductor "Package∀ Databook.

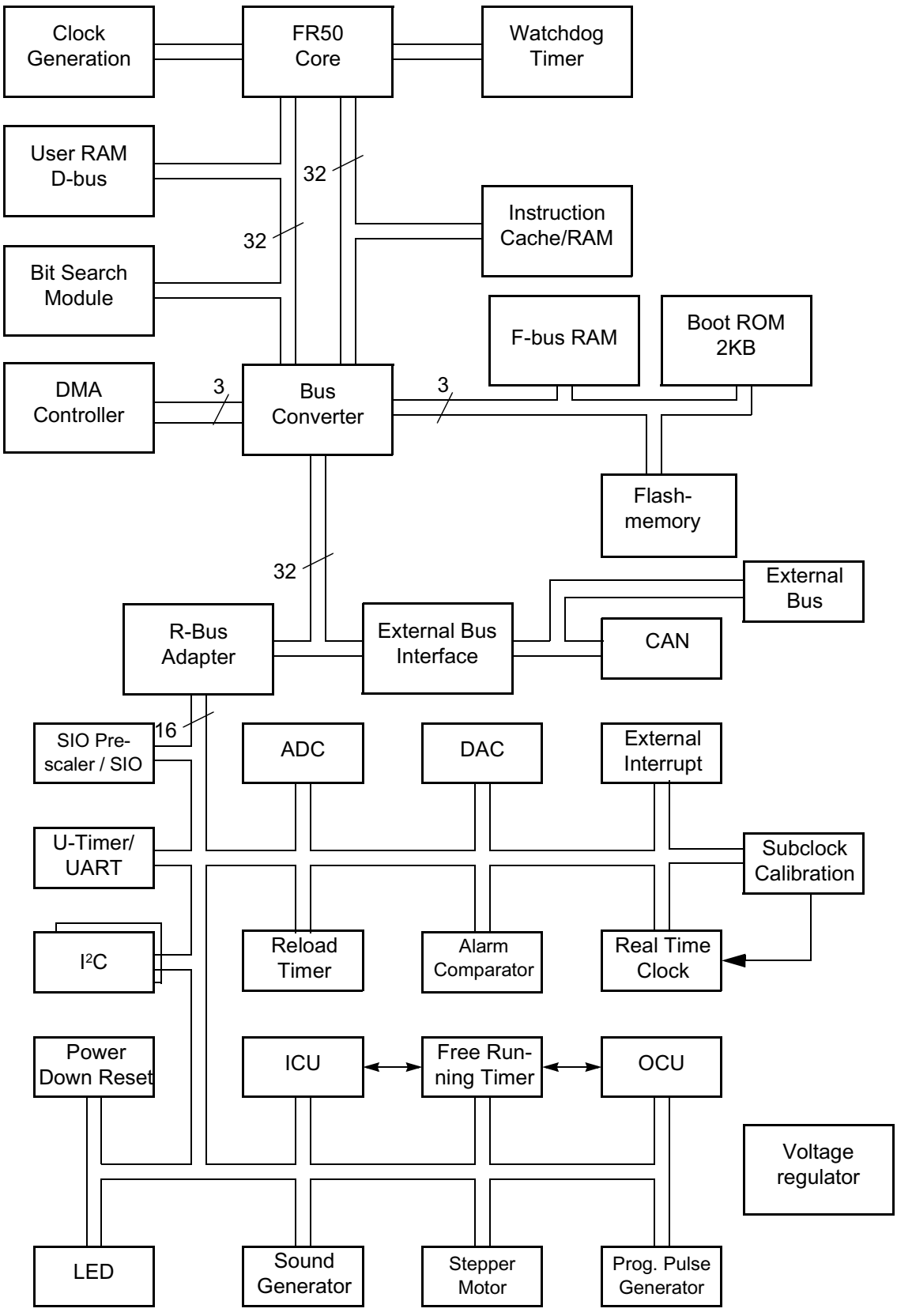
MB91FV360GA will be delivered in a 401 pin ceramic pin grid array package called PGA-401C-A02.

MB91F362GB will be delivered in a 208 pin plastic quad flat l-leaded package called FPT-208P-M04, while MB91F369GA will be delivered in a 160 pin package called FPT-160P-M15.

MB91F364G, MB91F365GB, MB91F366GB, MB91F367GB, MB91F368GB, MB91366GA and MB91F376G will be delivered in a 120 pin plastic quad flat l-leaded package called FPT-120P-M21.

1.4 MB91360 BLOCK STRUCTURE

Figure 1.4a Overall Block Diagram of MB91360 Devices



■ MB91360 Block Functions

Table 1.4a MB91360 Block Functions

Function	Feature	Remarks
FR50 Core	32-bit Fujitsu RISC Core FR30 software compatible	
Clock module (clock control, clock divider, PLLs)	Setting of frequencies for CPU and peripherals Low power consumption modes: RTC mode: only the **R**eal **T**ime **C**lock and the oscillator are active STOP mode: all internal circuits and the oscillation circuits are halted	initial value for osciallation stabilisation time in mode MD=”000”: 32 ms at 4 MHz oscillation clock. Time starts after release of INITX
Sub-clock/Calibration 32 KHz	RTC module can be clocked either from 32kHz quartz or from 4 MHz quartz. Dynamic switching is not allowed. Additionally, a calibration of the RTC timer in 32 kHz operation, based on the more accurate 4 MHz clock timing, is possible.	For availibility, see “MB91360 PRODUCT LINEUP” on page 4
Watchdog	adjustable watchdog timer interval (between 2^{20} and 2^{26} system clock cycles)	
Cache or Instruction RAM up to 4 KB	cache mode or RAM mode, lock function to make a specific program section cache resident	see remark below table
User RAM up to 16 KB	RAM for user data	see remark below table
F-bus RAM up to 16 KB	RAM for data and code	see remark below table
Flash Memory 512 KB	sector architecture: sector 0: 64 KB | sector 7: 64 KB sector 1: 64 KB | sector 8: 64 KB sector 2: 64 KB | sector 9: 64 KB sector 3: 32 KB | sector 10: 32 KB sector 4: 8 KB | sector 11: 8 KB sector 5: 8 KB | sector 12: 8 KB sector 6: 16 KB | sector 13: 16 KB | | V V 16 bit 16 bit write access is 16 bit wide, read access can be 16 or 32 bit wide	Minimum 10000 program/erase cycles Minimum 10 years data retention located on F-Bus For MB91F376G, please refer to the data sheet.
Boot ROM 2 KB		

Table 1.4a MB91360 Block Functions (Continued)

DMA	5 channels (up to 3 channels external) up to 16 DMA sources can be used transfer modes: single/block, burst, continuous	On MB91F362GB the DMA0 pin locations can be mapped into the power-separated area of the External Bus, controlled by the mode register (F362MD): DREQ0 <==> BRQ DACK0 <==> BGRNTX DEOP0 <==> AS
DSU	Debug Support Unit with trace functionallity (128 steps with internal FIFO; more steps with optional external trace memory)	Only on MB91FV360GA
EDSU	Embedded debug support unit: 4 instructions address breakpoints, 2 operand address breakpoints, 2 operand data breakpoints; breakpoints with mask- and range function	Only on MB91F364G
32 bit demultiplexed External Bus Interface	32 bit data, up to 27 bit address, 7 CS, CLK	Can be set to HiZ by software: (refer to feature “Mode Register”) Can be operated at 3.3V on MB91F362GB and MB91F369GA.
Interrupt Controller	8 external interrupt channels, 38 internal interrupts, 16 programmable priority levels	
Bit Search Module	Searches a word for the position of the first “1” and “0” change bit, starting from the MSB. Performs the search in 1 cycle.	
Fixed Reset Vector	Hardwired reset and mode vector	Code starts at 0F:4000H
Voltage Regulator	Generates internal voltage of 3.3 V	

Remark 1:

The MB91360 clock module allows the generation of a lot of clock frequencies. It cannot be guaranteed that at all those frequencies the resources will be able to generate all their specified output baud rates and frequencies.

Remark 2:

Set bit 9 (SYNCR) of TBCR to 1 to enable the synchronisation of the reset signal; a reset will be generated only after all bus accesses have been done. This avoids that erronous data are written to the RAMs during reset.

■ MB91360 Block Functions continued

Table 1.4b MB91360 Block Functions (Continued)

Function	Feature	Remarks
PPG for dimmer	16-bit PWM Timer 16 bit down counter, cycle and duty setting registers interrupt at triggering, cycle or duty match can be triggered by software or reload timer PWM operation and one-shot operation Clock disable internal prescaler allows f_{RES}/1, f_{RES}/4, f_{RES}/16, f_{RES}/64 as counter clock	Required frequencies are 90-300 Hz
ADC	successive approximation, internal sample and hold circuit 10-bit resolution, 5 V operation, (conversion time: 178cycles of CLKP) program selectable analogue input channels: single conversion mode continuous conversion mode stop conversion mode interrupt at the end of a conversion can be used to activate DMA transfer activation by software, external trigger or reload timer can be selected Prescaling is done internally Clock disable	
Basic Interval Timer	16-bit reload timer, includes clock prescaler ($f_{RES}/2^1$, $f_{RES}/2^3$, $f_{RES}/2^5$)	
CAN	conforms to CAN specification version 2.0 A and B, including 0-TQ bit timing automatic re-transmission in case of error automatic transmission responding to remote frame prioritized 16 message buffers for data and IDs supports multiple messages flexible configuration of acceptance filtering: full bit compare / full bit mask / two partial bit masks supports up to 1 Mb/s Clock Disable	

Table 1.4b MB91360 Block Functions (Continued)

DAC	R-2R D/A converter 10-bit resolution, 5 V operation Clock disable	Available on FV360GA and F362GA
External Interrupt	can be programmed to be edge sensitive or level sensitive interrupt mask and request pending bits per channel	
Mode Register (F362MD)	Special Mode Register, controls - the I²C module selection, - the DMA pin mapping, - the external bus disable mode - create asymmetric CLKT - address / CSX swap	Address: 0001FEH Bit 8: New I²C select Bit 9: DMA mapping enable Bit 10-13: Ext.Bus high-Z control Bit 14: ASYMCLKT Bit 15: ADRSWAP See section 4.2 for detalis
I²C-1 for standard mode	master or slave transmission arbitration function clock synchronization function slave address and general call address detect function transfer direction detect function start condition repeat generation and detection function bus error detect function compatible to I²C standard mode specification (operation up to 100 kHz, 7 bit addressing) includes clock divider functionality Clock disable	Only I²C-1 or I²C-2 can be used, not both in parallel. Bit 0 of F362MD will be used to decide which module is connected to the SCL and SDA pads. By default it is I²C-1.
I²C-2 for standard and fast mode	master or slave transmission arbitration function clock synchronization function slave address and general call address detect function transfer direction detect function start condition repeat generation and detection function bus error detect function compatible to I²C standard and fast mode specification (operation up to 400 kHz, 10 bit addressing) includes clock divider functionality Clock disable	Only I²C-1 or I²C-2 can be used, not both in parallel. Bit 0 of F362MD will be used to decide which module is connected to the SCL and SDA pads. By default it is I²C-1. SCL and SDA lines include optional noise filter. The noise filter allows the suppression of spikes in the range of 1 to 1.5 cycles of CLKP. Communication on the I²C bus between other connected devices is not possible if MB91360 devices are switched off.

Table 1.4b MB91360 Block Functions (Continued)

16-bit Input Capture (ICU)	rising edge, falling edge or rising & falling edge sensitive two 16-bit capture registers signals an interrupt at external event Clock disable	Available for FV360GA and F362GA
16-bit Output Compare OCU	signals an interrupt when a match with the timer value of the corresponding free running timer occurs an output signal can be generated Clock disable	Available for FV360GA and F362GA
Free running Timer	16-bit free running timer, signals an interrupt when overflow or match with compare register_0 includes prescaler ($f_{RES}/2^2$, $f_{RES}/2^4$, $f_{RES}/2^6$) timer data register has R/W access Clock disable	Available for FV360GA and F362GA
LED Port	allows to source 14 mA at Vdd-0.8 V and sink 24 mA at Vss+0.8 V respectively	The sum of all IOL currents of these 8 channels may not exceed 120 mA. The sum of all IOH currents of these channels may not exceed 120 mA. Available for FV360GA and F362GA
Alarm Comparator (OV/UV detection)	monitors an external voltage and generates an interrupt in case of a voltage lower or higher than the defined thresholds status is readable, interrupts can be masked separately Clock disable	uses external 4:1 voltage divider
Power down reset	monitors Vdd and generates a reset if Vdd is less than a defined threshold voltage	is disabled in RTC and STOP mode
Serial IO SIO Synchronous Serial Interface + SIO-Prescaler	Serial IO transfer can be started from MSB or LSB supports internal clock synchronized transfer and external clock synchronized transfer prescaler for shift clock allows: $f_{RES}/3$, $f_{RES}/4$, $f_{RES}/5$, $f_{RES}/6$, $f_{RES}/7$, $f_{RES}/8$ Clock disable	supports positive and negative clock edge synchronization

Table 1.4b MB91360 Block Functions (Continued)

Sound Generator (Buzzer)	8-bit PWM signal is mixed with tone frequency from 8-bit reload counter PWM clock by internal prescaler: fRES/1, fRES/2, fRES/4, fRES/5, fRES/8 tone frequency: PWM frequency / 2 / (reload value + 1) Clock disable	Target frequency to be programmable in the range of 300 Hz to 5 kHz fRES/1 and a reload value of 5 result in 5.2 kHz at fRES = 16MHz, fRES/4 and a reload value of 25 result in 300.48 Hz @ fRES = 16MHz
Stepper Motor Control	four high current outputs for each channel two synchronized 8-bit PWMs per channel internal prescaling for PMW clock: fRES, fRES/2, fRES/4, fRES/5, fRES/6, fres/8 includes zero detection circuit Clock disable	Available for FV360GA and F362GA target frequency: 16 kHz MB91360 devices include output drivers with a slew rate of typically 40 ns.
UART U-Timer (1 per UART)	serial I/O port for performing asynchronous (start-stop synchronization) communication full duplex, double buffering supports multi-processor mode variable data length (7/8 bit) 1 or 2 stop bits error detection function (parity, framing, overrun) interrupt function NRZ type transfer format baud rate generated by U-Timer 16-bit timer to generate the required UART clock: $f_{RES}/2^5,\ldots,\sim f_{RES}/2^{21}$ (asynchr. mode) Clock disable	polarity of the port signals for receive and transmit is programmable

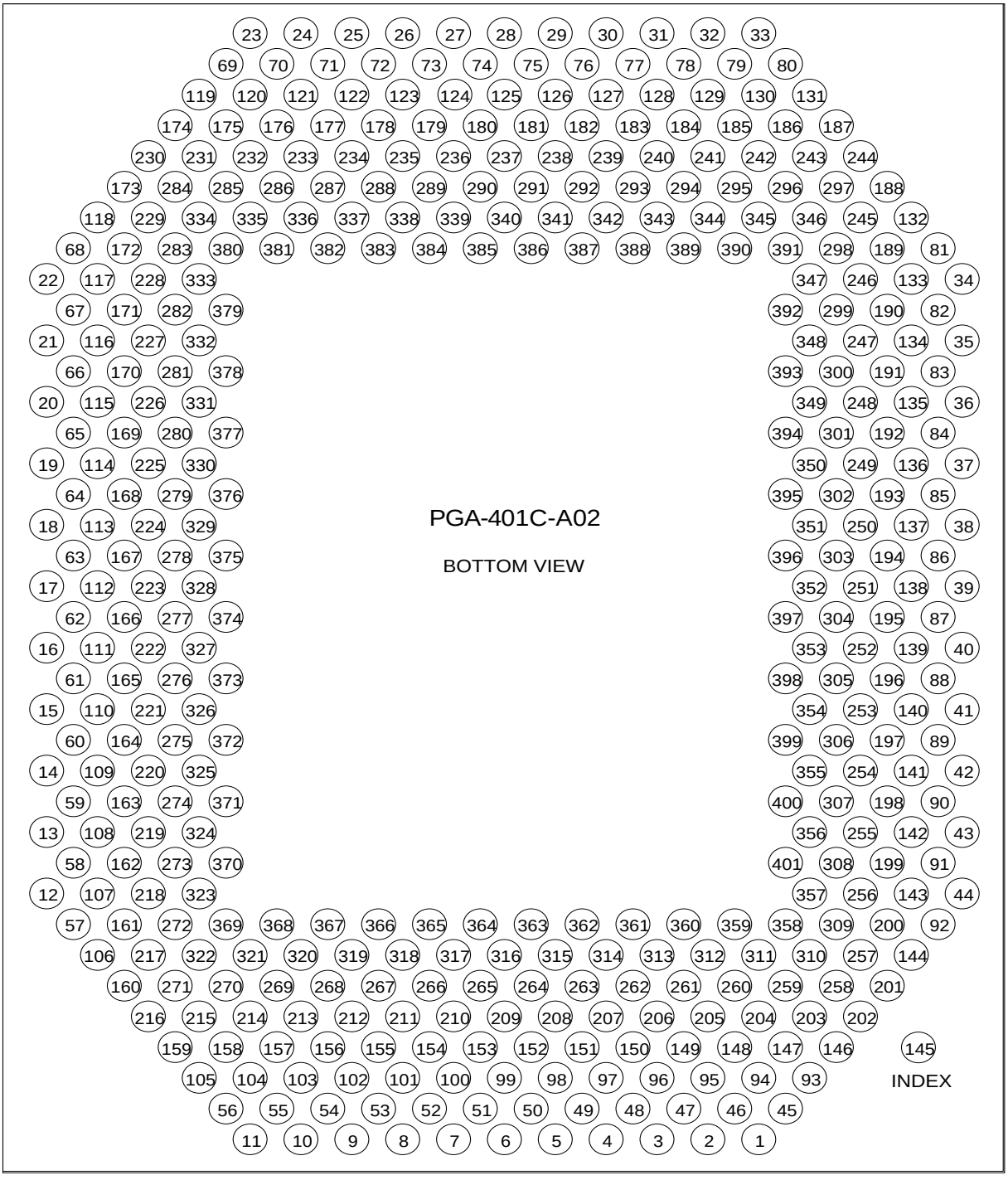
Table 1.4b MB91360 Block Functions (Continued)

USART with LIN option	serial I/O port for performing synchronous and asynchronous (start-stop synchronization) communication full duplex, double buffering supports multi-processor mode special features for LIN-bus systems variable data length (7/8 bit) 1 or 2 stop bits error detection function (parity, framing, overrun) interrupt function NRZ type transfer format baud rate generated by Baudrate Generator 15-bit timer to generate the required UART clock: $f_{CLKP}/2^1,\ldots,\sim f_{CLKP}/2^{15}$ Clock disable	This is a new module which cannot be emulated by MB91FV360GA. polarity of the port signals for receive and transmit is programmable
Real Time Clock (RTC) (Watch Timer)	facility to correct oscillation deviation read/write accessible second/minute/hour registers can signal interrupts every second/minute/hour/day internal clock divider and prescaler provide exact 1s clock based on the 4 MHz input Clock disable	prescaler value for 4 MHz is 1E847F$_H$ prescaler value for 32 kHz is 04000F$_H$
Clock modulator	16 tap modulation circuit max. input frequency 48 MHz, max output frequency 64 MHz	

1.5 PIN ASSIGNMENT

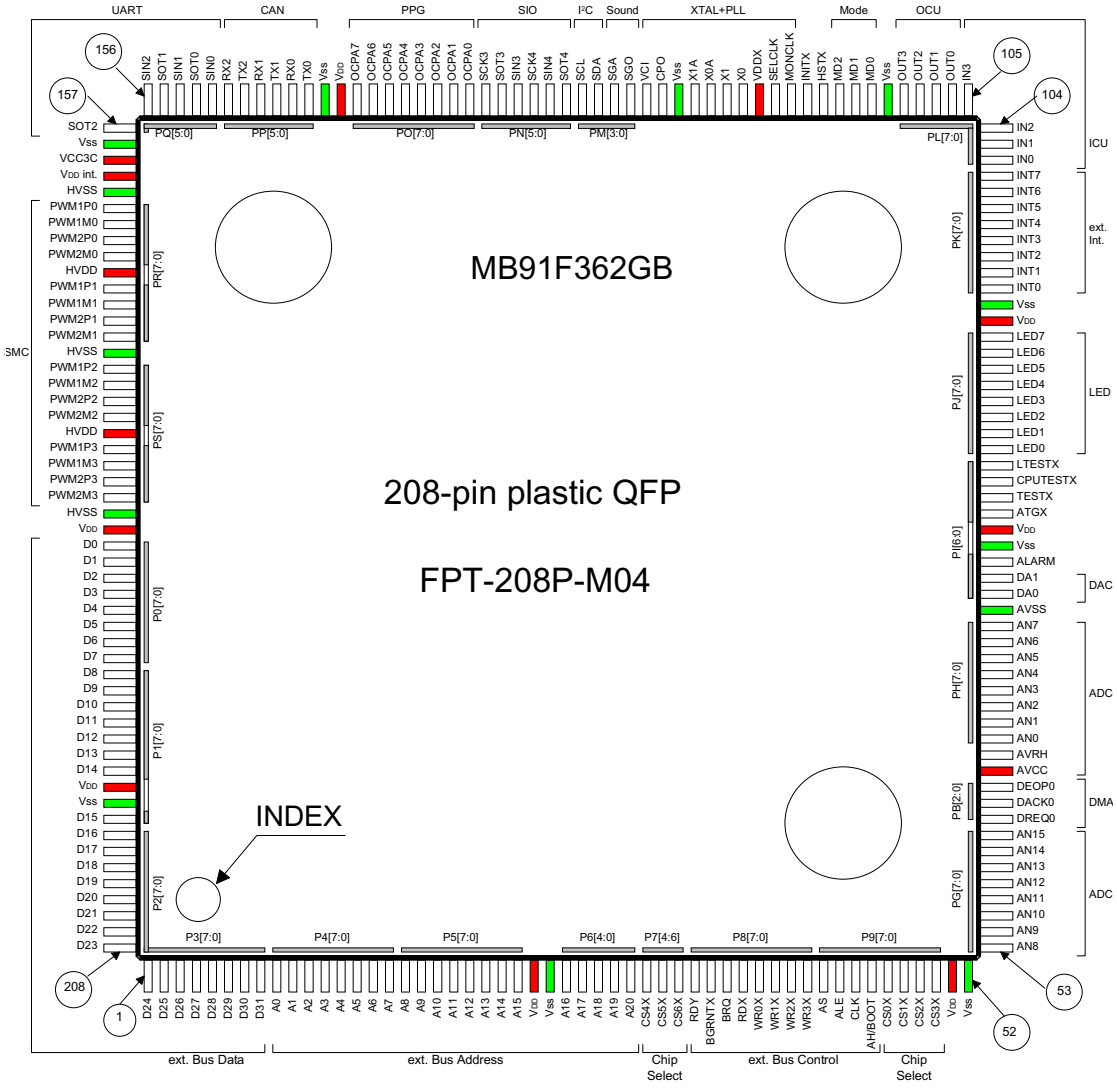
1.5.1 Pin Diagram MB91FV360GA

Figure 1.5.1 Pin diagram MB91FV360GA



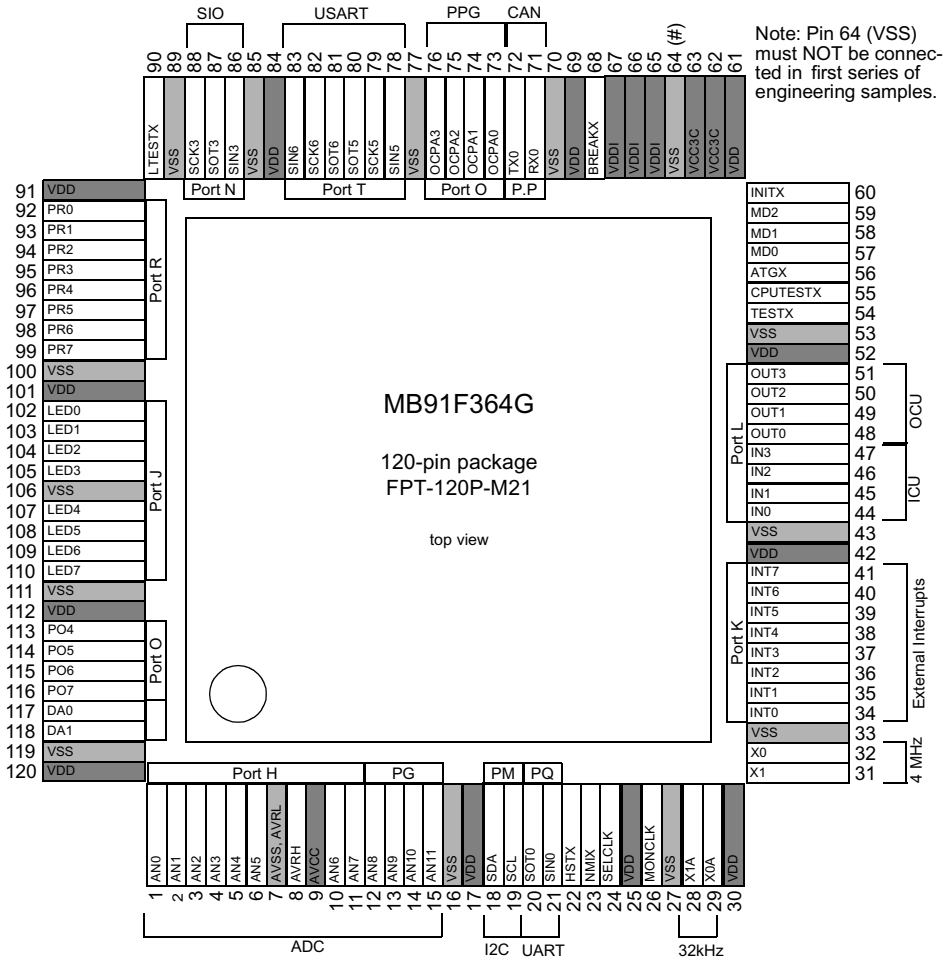
1.5.2 Pin Diagram MB91F362GB

Figure 1.5.2 Pin diagram MB91F362GB



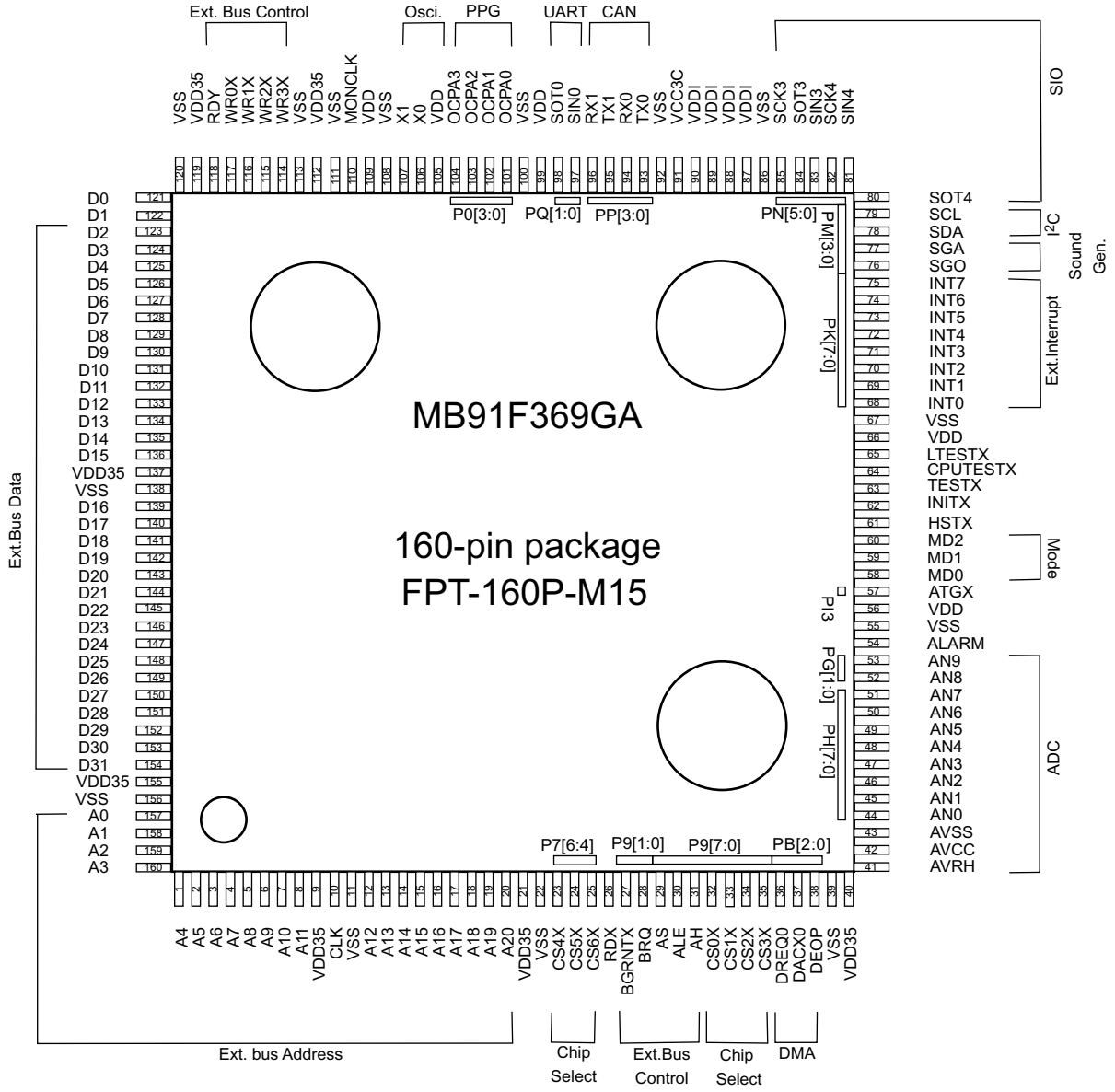
1.5.3 Pin Diagram MB91F364G

Figure 1.5.3 Pin diagram MB91G364G



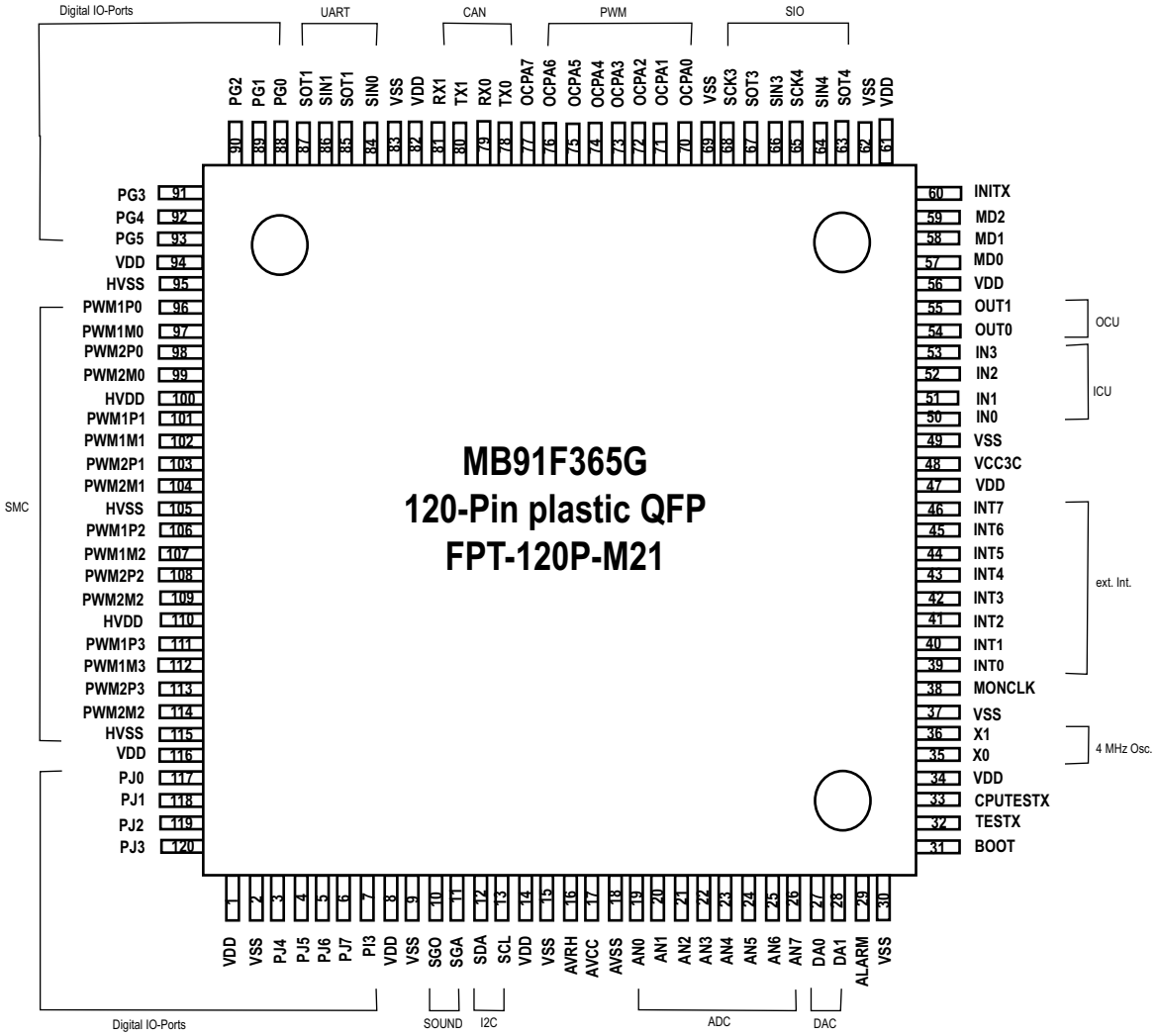
1.5.4 Pin Diagram MB91F369GA

Figure 1.5.4 Pin diagram MB91F369GA



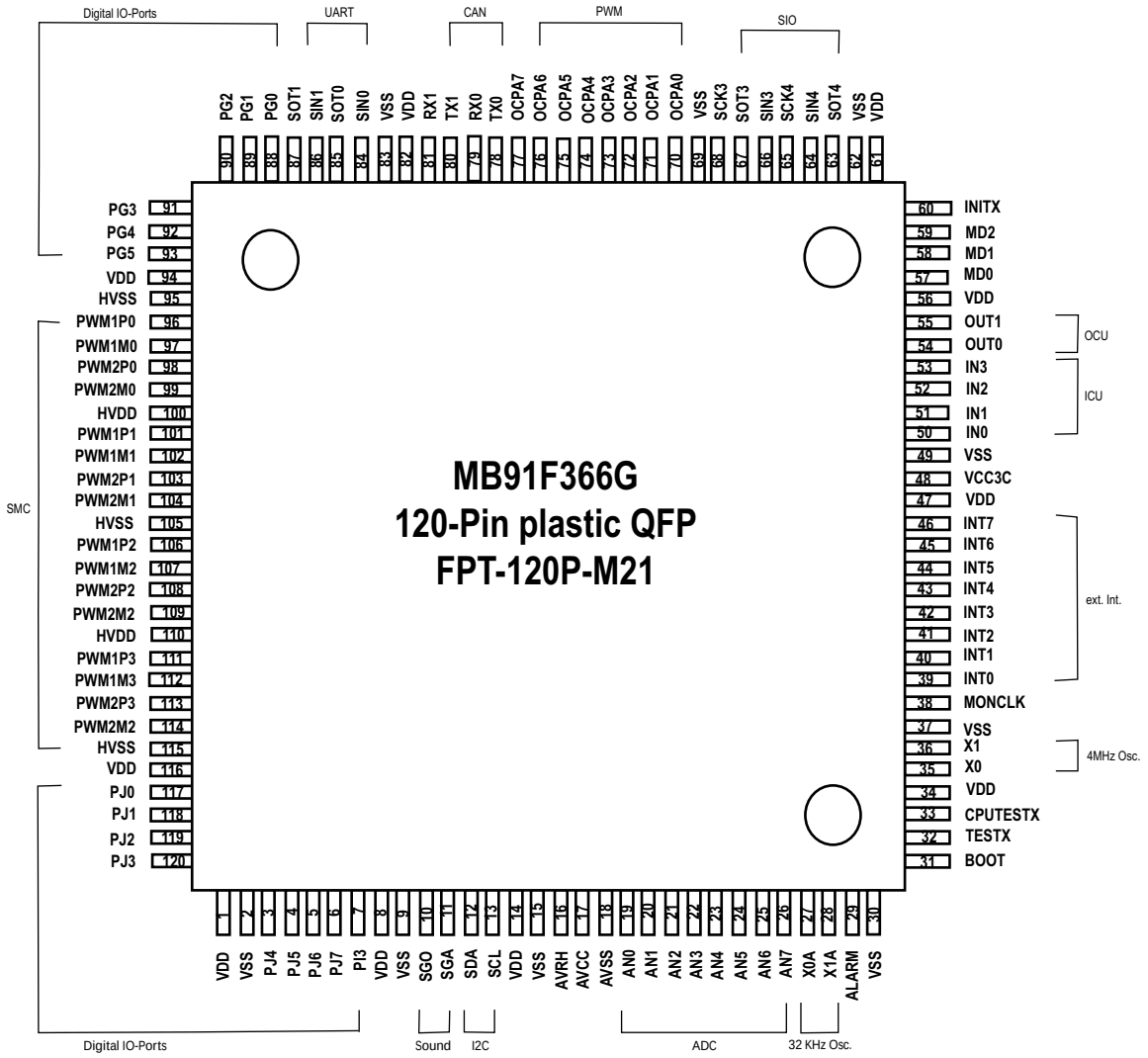
1.5.5 Pin Diagram MB91F365GB

Figure 1.5.5 Pin diagram MB91F365GB



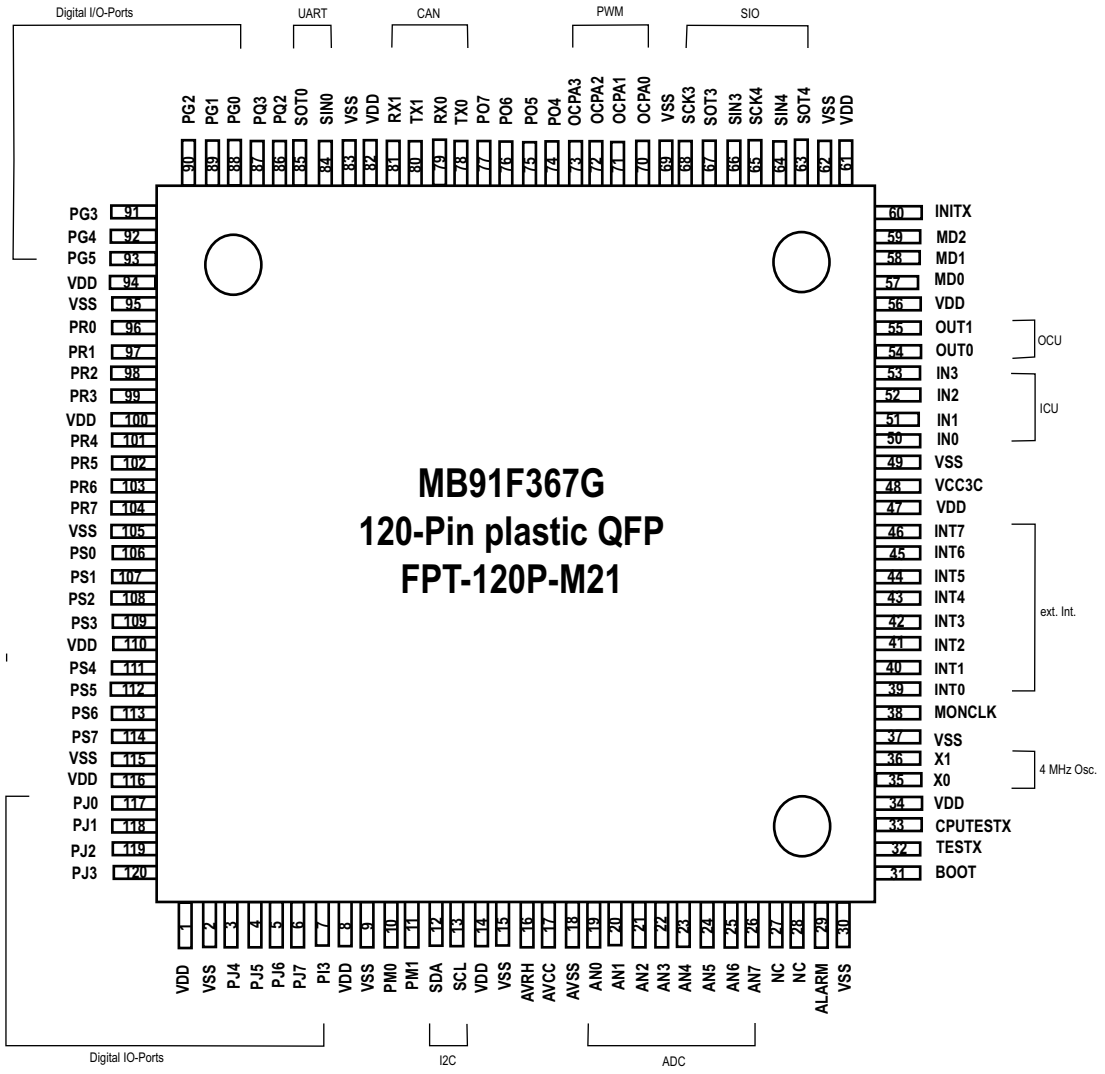
1.5.6 Pin Diagram MB91F366GB/MB91F376G

Figure 1.5.6 Pin diagram MB91F366GB



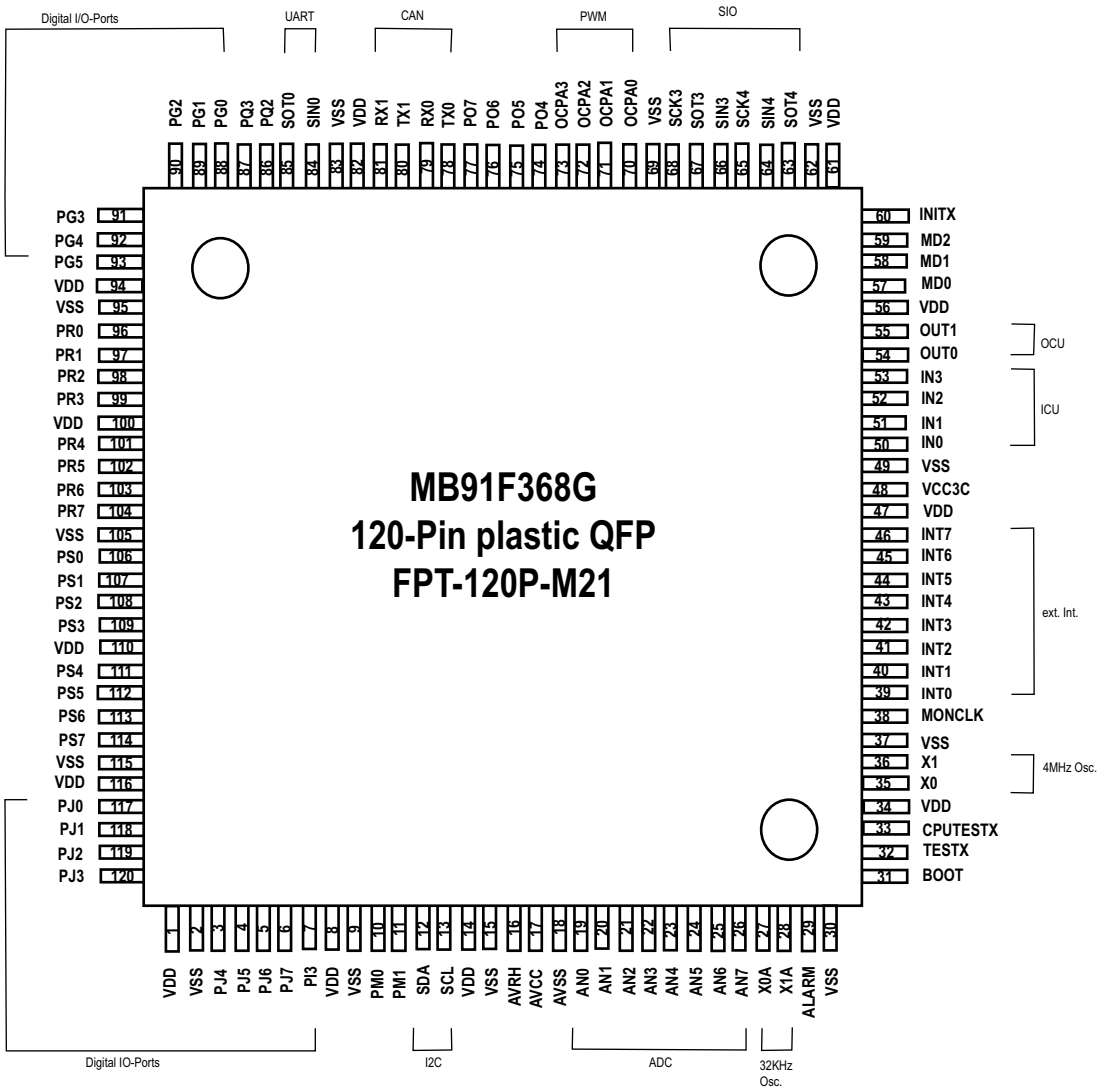
1.5.7 Pin Diagram MB91F367GB

Figure 1.5.7 Pin diagram MB91F367GB



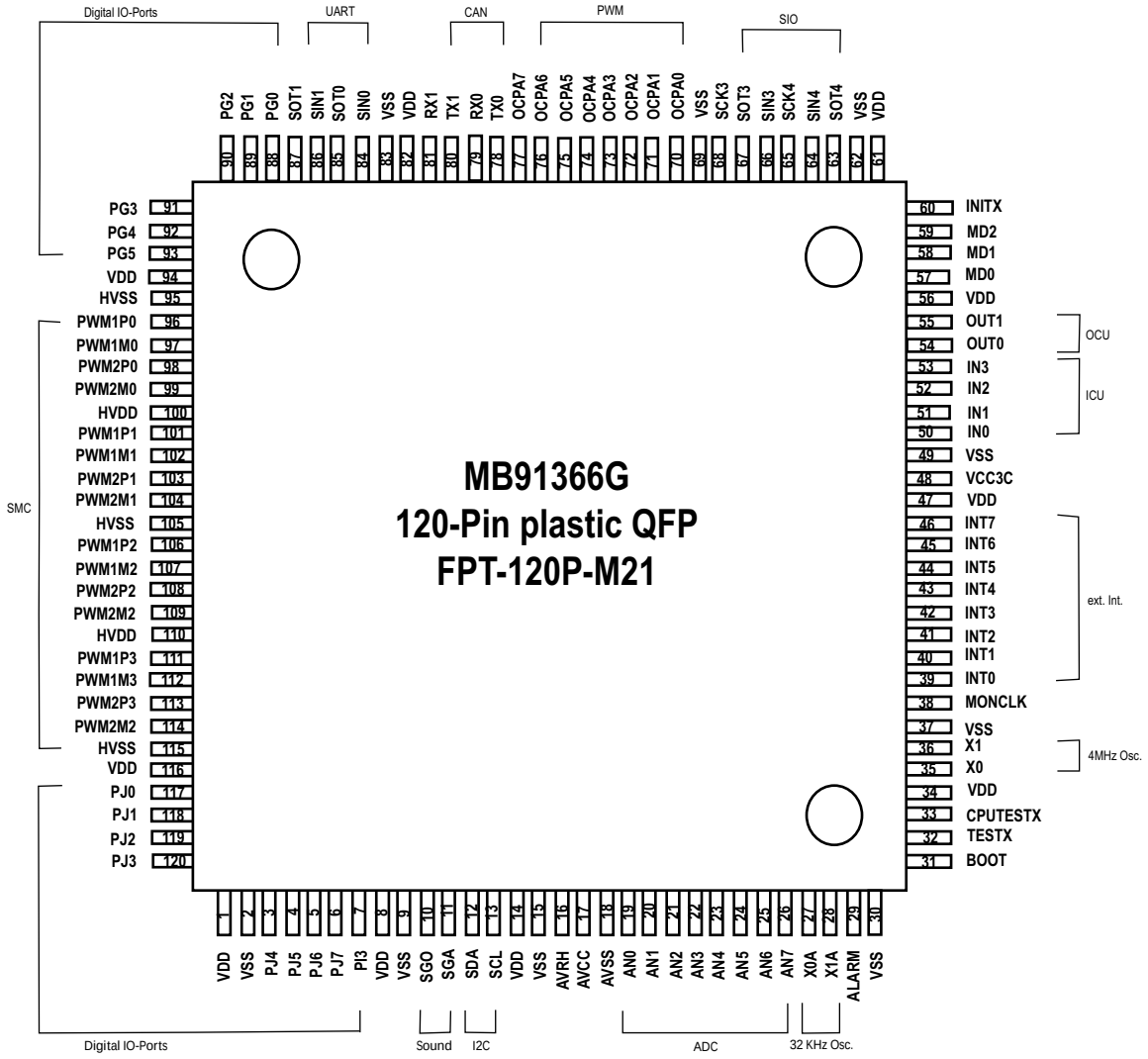
1.5.8 Pin Diagram MB91F368GB

Figure 1.5.8 Pin diagram MB91F368GB



1.5.9 Pin Diagram MB91366GA

Figure 1.5.9 Pin diagram MB91366GA



1.6 I/O PINS AND THEIR FUNCTION

1.6.1 MB91FV360GA I/O Pins and Their Function

Table 1.6.1 MB91FV360 I/O pins and their functions

Pin No.	Pin Name	I/O	General Purpose IO Port	Circuit Type	Function
1	D18	I/O		Q	Ext. Bus Data Bit 18
2	D11	I/O		Q	Ext. Bus Data Bit 11
3	D2	I/O		Q	Ext. Bus Data Bit 2
4	not connected				
5	HVSS				
6	HVDD5B				
7	PWM2M1	I/O	PR7	M	SMC 1
8	PWM1M1	I/O	PR5	K	SMC 1
9	PWM1P0	I/O	PR0	K	SMC 0
10	VDD5R				
11	VDD5P				
12	SCK4	I/O	PN2	A	SIO Clock
13	VDD5J				
14	EXRAM	I		P	Trace Control
15	TWRX	O		X	Trace Control
16	TAD9	O		X	Trace Address
17	TAD5	O		X	Trace Address
18	TAD3	O		X	Trace Address
19	TDT68	I/O		W	Trace Data
20	TDT63	I/O		W	Trace Data
21	TDT57	I/O		W	Trace Data
22	TDT49	I/O		W	Trace Data
23	TDT23	I/O		W	Trace Data
24	TDT16	I/O		W	Trace Data
25	TDT7	I/O		W	Trace Data
26	TDT2	I/O		W	Trace Data
27	ICD0	I/O		N	ICE Data
28	ICLK	I/O		L	ICE Clock
29	X0			H	4 MHz Oscillator Pin
30	INITX	I		U	Initial Pin
31	MD1	I		T	Mode Pin 1

Table 1.6.1 MB91FV360 I/O pins and their functions (Continued)

32	IN3	I/O	PL3	A	ICU Input 3
33	INT3	I/O	PK3	A	Ext. Interrupt 3
34	AN3	I/O	PH3	B	ADC Input 3
35	DACK2	I/O	PB6	A	DMA Acknowledge 2
36	AN13	I/O	PG5	B	ADC Input 13
37	AN8	I/O	PG0	B	ADC Input 8
38	ALE	I/O	P91	A	Ext. Bus Control
39	WR1X	I/O	P85	S	Ext. Bus Control
40	RDX	I/O	P83	S	Ext. Bus Control
41	CS7X	I/O		A	Chip Select 7 (CANs)
42	A26	I/O		Q	Ext. Bus Address Bit 26
43	A20	I/O		Q	Ext. Bus Address Bit 20
44	A12	I/O		Q	Ext. Bus Address Bit 12
45	D21	I/O		Q	Ext. Bus Data Bit 21
46	D16	I/O		Q	Ext. Bus Data Bit 16
47	D13	I/O		Q	Ext. Bus Data Bit 13
48	D7	I/O		Q	Ext. Bus Data Bit 7
49	D3	I/O		Q	Ext. Bus Data Bit 3
50	VSS				
51	PWM2P2	I/O	PS2	K	SMC 2
52	PWM2P1	I/O	PR6	K	SMC 1
53	PWM1P1	I/O	PR4	K	SMC 1
54	not connected				
55	SIN1	I/O	PQ2	A	UART 1 Input
56	TX3	I/O	PP6	Q	CAN 3 TX
57	SOT3	I/O	PN4	A	SIO Output
58	SOT4	I/O	PN0	A	SIO Output
59	not connected				
60	not connected				
61	SGO	I/O	PM0	A	Sound Generator SGO
62	TOEX	O		X	Trace Control
63	TAD8	O		X	Trace Address
64	TAD2	O		X	Trace Address
65	TDT67	I/O		W	Trace Data
66	TDT60	I/O		W	Trace Data
67	TDT54	I/O		W	Trace Data

Table 1.6.1 MB91FV360 I/O pins and their functions (Continued)

68	TDT48	I/O		W	Trace Data
69	TDT26	I/O		W	Trace Data
70	TDT21	I/O		W	Trace Data
71	TDT18	I/O		W	Trace Data
72	TDT12	I/O		W	Trace Data
73	TDT8	I/O		W	Trace Data
74	TDT3	I/O		W	Trace Data
75	ICS2	O		G	ICE Status
76	VDD5F				
77	RSTX	I		E	Reset Pin
78	OUT2	I/O	PL6	A	OCU Output 2
79	IN0	I/O	PL0	A	ICU Input 0
80	INT2	I/O	PK2	A	Ext. Interrupt 2
81	AN6	I/O	PH6	B	ADC Input 6
82	AN1	I/O	PH1	B	ADC Input 1
83	AVCC				Analog VCC
84	DEOP0	I/O	PB2	A	DMA EOP 0
85	AN14	I/O	PG6	B	ADC Input 14
86	AN9	I/O	PG1	B	ADC Input 9
87	AS	I/O	P90	A	Ext. Bus Control
88	BRQ	I/O	P82	A	Ext. Bus Control
89	CS6X	I/O	P76	A	Chip Select 6
90	A23	I/O		Q	Ext. Bus Address Bit 23
91	A17	I/O		Q	Ext. Bus Address Bit 17
92	A11	I/O		Q	Ext. Bus Address Bit 11
93	D27	I/O		Q	Ext. Bus Data Bit 27
94	D22	I/O		Q	Ext. Bus Data Bit 22
95	D17	I/O		Q	Ext. Bus Data Bit 17
96	D6	I/O		Q	Ext. Bus Data Bit 6
97	VDD5S				
98	PWM1M3	I/O	PS5	K	SMC 3
99	PWM2M3	I/O	PS7	M	SMC 3
100	HVDD5A				
101	PWM2P0	I/O	PR2	K	SMC 0
102	VCC3/C			C	Bypass Capacitor Pin
103	SOT1	I/O	PQ3	A	UART 1 Output

Table 1.6.1 MB91FV360 I/O pins and their functions (Continued)

104	SIN0	I/O	PQ0	A	UART 0 Input
105	TX1	I/O	PP2	Q	CAN 1 TX
106	OCPA2	I/O	PO2	A	PPG Output
107	SCK3	I/O	PN5	A	SIO Clock
108	SIN4	I/O	PN1	A	SIO Input
109	SCL	I/O	PM3	Y	I^2C SCL
110	TCLK	I/O		W	Trace Control
111	TAD12	O		X	Trace Address
112	TAD15	O		X	Trace Address
113	TAD1	O		X	Trace Address
114	TDT65	I/O		W	Trace Data
115	TDT59	I/O		W	Trace Data
116	TDT55	I/O		W	Trace Data
117	TDT51	I/O		W	Trace Data
118	TDT42	I/O		W	Trace Data
119	TDT32	I/O		W	Trace Data
120	TDT27	I/O		W	Trace Data
121	TDT22	I/O		W	Trace Data
122	TDT11	I/O		W	Trace Data
123	TDT4	I/O		W	Trace Data
124	ICD3	I/O		N	ICE Data
125	TDT1	I/O		W	Trace Data
126	SELCLK	I		F	Clock Selection
127	NMIX	I		E	Non maskable Interrupt
128	OUT1	I/O	PL5	A	OCU Output 1
129	IN1	I/O	PL1	A	ICU Input 1
130	INT5	I/O	PK5	A	Ext. Interrupt 5
131	LED4	I/O	PJ4	J	LED Port 4
132	ALARM	I		D	Alarm Comparator Input
133	AN7	I/O	PH7	B	ADC Input 7
134	AN2	I/O	PH2	B	ADC Input 2
135	DACK0	I/O	PB1	A	DMA Acknowledge 0
136	AN10	I/O	PG2	B	ADC Input 10
137	CS0X	I/O	P94	A	Chip select 0
138	CS3X	I/O	P97	A	Chip select 3
139	BGRNTX	I/O	P81	A	Ext. Bus Control

Table 1.6.1 MB91FV360 I/O pins and their functions (Continued)

140	CS4X	I/O	P74	A	Chip Select 4
141	A22	I/O		Q	Ext. Bus Address Bit 22
142	A18	I/O		Q	Ext. Bus Address Bit 18
143	A14	I/O		Q	Ext. Bus Address Bit 14
144	A5	I/O		Q	Ext. Bus Address Bit 5
145	Index Pin	-	-	-	Index Pin
146	D30	I/O		Q	Ext. Bus Data Bit 30
147	D26	I/O		Q	Ext. Bus Data Bit 26
148	D19	I/O		Q	Ext. Bus Data Bit 19
149	D10	I/O		Q	Ext. Bus Data Bit 10
150	D9	I/O		Q	Ext. Bus Data Bit 9
151	D5	I/O		Q	Ext. Bus Data Bit 5
152	PWM2M2	I/O	PS3	M	SMC 2
153	PWM1P3	I/O	PS4	K	SMC 3
154	PWM2M0	I/O	PR3	M	SMC 0
155	VSS				
156	SOT2	I/O	PQ5	A	UART 2 Output
157	SOT0	I/O	PQ1	A	UART 0 Output
158	VDD5O				
159	OCPA7	I/O	PO7	A	PPG Output
160	OCPA5	I/O	PO5	A	PPG Output
161	OCPA1	I/O	PO1	A	PPG Output
162	VDD5K				
163	X1A	O		I	32 kHz Oscillator Pin
164	X0A	I		I	32 kHz Oscillator Pin
165	SDA	I/O	PM2	Y	I^2C SDA
166	TAD10	O		X	Trace Address
167	TAD11	O		X	Trace Address
168	TDT66	I/O		W	Trace Data
169	TDT61	I/O		W	Trace Data
170	TDT58	I/O		W	Trace Data
171	TDT52	I/O		W	Trace Data
172	TDT45	I/O		W	Trace Data
173	TDT39	I/O		W	Trace Data
174	TDT35	I/O		W	Trace Data
175	TDT31	I/O		W	Trace Data

Table 1.6.1 MB91FV360 I/O pins and their functions (Continued)

176	TDT24	I/O		W	Trace Data
177	TDT15	I/O		W	Trace Data
178	TDT14	I/O		W	Trace Data
179	TDT10	I/O		W	Trace Data
180	ICD1	I/O		N	ICE Data
181	ICD2	I/O		N	ICE Data
182	HSTX	I		E	Hardware Standby
183	OUT3	I/O	PL7	A	OCU Output 3
184	OUT0	I/O	PL4	A	OCU Output 0
185	INT6	I/O	PK6	A	Ext. Interrupt 6
186	LED7	I/O	PJ7	J	LED Port 7
187	LED1	I/O	PJ1	J	LED Port 1
188	CPUTESTX	I		E	Test Input
189	DA1	O		C	DAC Output
190	AN4	I/O	PH4	B	ADC Input 4
191	DEOP1	I/O	PB5	A	DMA EOP 1
192	DACK1	I/O	PB4	A	DMA Acknowledge 1
193	DREQ0	I/O	PB0	A	DMA Request 0
194	CLK	I/O	P92	A	Ext. Bus Clk
195	AH/BOOT	I/O	P93	A	Ext. Bus Control/Boot Signal
196	CS5X	I/O	P75	A	Chip Select 5
197	A24	I/O		Q	Ext. Bus Address Bit 24
198	A21	I/O		Q	Ext. Bus Address Bit 21
199	A15	I/O		Q	Ext. Bus Address Bit 15
200	A8	I/O		Q	Ext. Bus Address Bit 8
201	A2	I/O		Q	Ext. Bus Address Bit 2
202	A0	I/O		Q	Ext. Bus Address Bit 0
203	D29	I/O		Q	Ext. Bus Data Bit 29
204	D25	I/O		Q	Ext. Bus Data Bit 25
205	D20	I/O		Q	Ext. Bus Data Bit 20
206	D15	I/O		Q	Ext. Bus Data Bit 15
207	D4	I/O		Q	Ext. Bus Data Bit 4
208	HVDD5C				
209	PWM1M2	I/O	PS1	K	SMC 2
210	PWM1P2	I/O	PS0	K	SMC 2
211	PWM1M0	I/O	PR1	K	SMC 0

Table 1.6.1 MB91FV360 I/O pins and their functions (Continued)

212	SIN2	I/O	PQ4	A	UART 2 Input
213	RX3	I/O	PP7	Q	CAN 3 RX
214	VSS				
215	RX0	I/O	PP1	Q	CAN 0 RX
216	VDD5N				
217	OCPA4	I/O	PO4	A	PPG Output
218	OCPA0	I/O	PO0	A	PPG Output
219	SIN3	I/O	PN3	A	SIO Input
220	VSS				
221	SGA	I/O	PM1	A	Sound Generator SGA
222	TAD13	O		X	Trace Address
223	TAD7	O		X	Trace Address
224	TAD6	O		X	Trace Address
225	TDT64	I/O		W	Trace Data
226	TDT56	I/O		W	Trace Data
227	TDT50	I/O		W	Trace Data
228	TDT44	I/O		W	Trace Data
229	TDT41	I/O		W	Trace Data
230	TDT37	I/O		W	Trace Data
231	TDT34	I/O		W	Trace Data
232	TDT30	I/O		W	Trace Data
233	TDT25	I/O		W	Trace Data
234	TDT20	I/O		W	Trace Data
235	TDT9	I/O		W	Trace Data
236	BREAK	I		O	ICE Break
237	ICS1	O		G	ICE Status
238	ICS0	O		G	ICE Status
239	MD2	I		T	Mode Pin 2
240	IN2	I/O	PL2	A	ICU Input 2
241	INT4	I/O	PK4	A	Ext. Interrupt 4
242	LED6	I/O	PJ6	J	LED Port 6
243	LED3	I/O	PJ3	J	LED Port 3
244	not connected				
245	TESTX	I		E	Test Input
246	DA0	O		C	DAC Output
247	AN5	I/O	PH5	B	ADC Input 5

Table 1.6.1 MB91FV360 I/O pins and their functions (Continued)

248	AN0	I/O	PH0	B	ADC Input 0
249	AN15	I/O	PG7	B	ADC Input 15
250	CS1X	I/O	P95	A	Chip select 1
251	WR3X	I/O	P87	S	Ext. Bus Control
252	WR2X	I/O	P86	S	Ext. Bus Control
253	DREQ2	I/O	P73	A	DMA Request 2
254	A19	I/O		Q	Ext. Bus Address Bit 19
255	A13	I/O		Q	Ext. Bus Address Bit 13
256	A7	I/O		Q	Ext. Bus Address Bit 7
257	A4	I/O		Q	Ext. Bus Address Bit 4
258	D31	I/O		Q	Ext. Bus Data Bit 31
259	D28	I/O		Q	Ext. Bus Data Bit 28
260	D23	I/O		Q	Ext. Bus Data Bit 23
261	D14	I/O		Q	Ext. Bus Data Bit 14
262	D8	I/O		Q	Ext. Bus Data Bit 8
263	D1	I/O		Q	Ext. Bus Data Bit 1
264	D0	I/O		Q	Ext. Bus Data Bit 0
265	not connected				
266	HVSS				
267	not connected				
268	VSS				
269	RX2	I/O	PP5	Q	CAN 2 RX
270	RX1	I/O	PP3	Q	CAN 1 RX
271	VSS				
272	OCPA3	I/O	PO3	A	PPG Output
273	VSS				
274	not connected				
275	VDD5I				
276	TADSCX	O		X	Trace Control
277	TCE1X	O		X	Trace Control
278	TAD4	O		X	Trace Address
279	TAD0	O		X	Trace Address
280	TDT62	I/O		W	Trace Data
281	TDT53	I/O		W	Trace Data
282	TDT47	I/O		W	Trace Data
283	TDT43	I/O		W	Trace Data

Table 1.6.1 MB91FV360 I/O pins and their functions (Continued)

284	TDT36	I/O		W	Trace Data
285	TDT33	I/O		W	Trace Data
286	TDT28	I/O		W	Trace Data
287	TDT19	I/O		W	Trace Data
288	TDT13	I/O		W	Trace Data
289	TDT6	I/O		W	Trace Data
290	TDT5	I/O		W	Trace Data
291	X1			H	4 MHz Oscillator Pin
292	MONCLK	O		G	Clock Output for test purposes
293	MD0	I		T	Mode Pin 0
294	INT7	I/O	PK7	A	Ext. Interrupt 7
295	INT1	I/O	PK1	A	Ext. Interrupt 1
296	LED5	I/O	PJ5	J	LED Port 5
297	LTESTX	I		E	Test Input
298	ATGX	I/O	PI3	A	ADC Trigger
299	AVRL			R	Analog Reference Low
300	AVRH			R	Analog Reference High
301	DREQ1	I/O	PB3	A	DMA Request 1
302	AN12	I/O	PG4	B	ADC Input 12
303	AN11	I/O	PG3	B	ADC Input 11
304	WR0X	I/O	P84	S	Ext. Bus Control
305	RDY	I/O		S	Ext. Bus Control
306	A25	I/O		Q	Ext. Bus Address Bit 25
307	A16	I/O		Q	Ext. Bus Address Bit 16
308	A10	I/O		Q	Ext. Bus Address Bit 10
309	A6	I/O		Q	Ext. Bus Address Bit 6
310	A1	I/O		Q	Ext. Bus Address Bit 1
311	not connected				
312	D24	I/O		Q	Ext. Bus Data Bit 24
313	D12	I/O		Q	Ext. Bus Data Bit 12
314	not connected				
315	PWM2P3	I/O	PS6	K	SMC 3
316	HVSS				
317	HVSS				
318	not connected				
319	VDD5Q				

Table 1.6.1 MB91FV360 I/O pins and their functions (Continued)

320	TX2	I/O	PP4	Q	CAN 2 TX
321	TX0	I/O	PP0	Q	CAN 0 TX
322	OCPA6	I/O	PO6	A	PPG Output
323	VDD5M				
324	VDD5L				
325	not connected				
326	VDD5H				
327	TAD14	O		X	Trace Address
328	VSS3				
329	VSS3				
330	not connected				
331	VDD3C				
332	TDT46	I/O		W	Trace Data
333	TDT40	I/O		W	Trace Data
334	TDT38	I/O		W	Trace Data
335	VDD3B				
336	TDT29	I/O		W	Trace Data
337	TDT17	I/O		W	Trace Data
338	VDD3A				
339	TDT0	I/O		W	Trace Data
340	VSS				
341	VSS				
342	not connected				
343	VDD5E				
344	INT0	I/O	PK0	A	Ext. Interrupt 0
345	LED2	I/O	PJ2	J	LED Port 2
346	LED0	I/O	PJ0	J	LED Port 0
347	VDD5D				
348	AVSS				Analog VSS
349	DEOP2	I/O	PB7	A	DMA EOP 2
350	VDD5C				
351	CS2X	I/O	P96	A	Chip select 2
352	VSS				
353	VSS				
354	VDD5B				
355	not connected				

Table 1.6.1 MB91FV360 I/O pins and their functions (Continued)

356	A9	I/O		Q	Ext. Bus Address Bit 9
357	A3	I/O		Q	Ext. Bus Address Bit 3
358	VSS				
359	VSS				
360	VDD5T				
361	VSS				
362	VSS				
363	VSS				
364	not connected				
365	HVSS				
366	VSS				
367	VSS				
368	not connected				
369	VSS				
370	VSS				
371	not connected				
372	VSS				
373	VSS				
374	VSS				
375	VDD3D				
376	VSS3				
377	VSS3				
378	VSS3				
379	not connected				
380	VSS3				
381	VSS3				
382	not connected				
383	VSS3				
384	VSS3				
385	VSS3				
386	VDD5G				
387	VSS				
388	VSS				
389	VSS				
390	not connected				
391	VSS				

Table 1.6.1 MB91FV360 I/O pins and their functions (Continued)

392	VSS				
393	not connected				
394	VSS				
395	VSS				
396	VSS				
397	not connected				
398	VSS				
399	VSS				
400	VSS				
401	VDD5A				

1.6.2 MB91F362GB I/O Pins and Their Function

Table 1.6.2 MB91F362GB I/O pins and their function

Pin No.	Pin Name	I/O	General Purpose IO Port	Circuit Type	Function
1	D24	I/O		Q	Ext. Bus Data Bit 24
2	D25	I/O		Q	Ext. Bus Data Bit 25
3	D26	I/O		Q	Ext. Bus Data Bit 26
4	D27	I/O		Q	Ext. Bus Data Bit 27
5	D28	I/O		Q	Ext. Bus Data Bit 28
6	D29	I/O		Q	Ext. Bus Data Bit 29
7	D30	I/O		Q	Ext. Bus Data Bit 30
8	D31	I/O		Q	Ext. Bus Data Bit 31
9	A0	I/O		Q	Ext. Bus Address Bit 0
10	A1	I/O		Q	Ext. Bus Address Bit 1
11	A2	I/O		Q	Ext. Bus Address Bit 2
12	A3	I/O		Q	Ext. Bus Address Bit 3
13	A4	I/O		Q	Ext. Bus Address Bit 4
14	A5	I/O		Q	Ext. Bus Address Bit 5
15	A6	I/O		Q	Ext. Bus Address Bit 6
16	A7	I/O		Q	Ext. Bus Address Bit 7
17	A8	I/O		Q	Ext. Bus Address Bit 8
18	A9	I/O		Q	Ext. Bus Address Bit 9
19	A10	I/O		Q	Ext. Bus Address Bit 10
20	A11	I/O		Q	Ext. Bus Address Bit 11
21	A12	I/O		Q	Ext. Bus Address Bit 12
22	A13	I/O		Q	Ext. Bus Address Bit 13
23	A14	I/O		Q	Ext. Bus Address Bit 14
24	A15	I/O		Q	Ext. Bus Address Bit 15
25	VDD35				separated Ext.Bus VDD, 3.3 or 5.0 V
26	VSS				
27	A16	I/O		Q	Ext. Bus Address Bit 16
28	A17	I/O		Q	Ext. Bus Address Bit 17
29	A18	I/O		Q	Ext. Bus Address Bit 18
30	A19	I/O		Q	Ext. Bus Address Bit 19
31	A20	I/O		Q	Ext. Bus Address Bit 20
32	CS4X	I/O	P74	A	Chip Select 4

Table 1.6.2 MB91F362GB I/O pins and their function (Continued)

33	CS5X	I/O	P75	A	Chip Select 5
34	CS6X	I/O	P76	A	Chip Select 6
35	RDY	I/O		S	Ext. Bus Control
36	BGRNTX	I/O	P81	A	Ext. Bus Control
37	BRQ	I/O	P82	A	Ext. Bus Control
38	RDX	I/O		S	Ext. Bus Control
39	WR0X	I/O		S	Ext. Bus Control
40	WR1X	I/O		S	Ext. Bus Control
41	WR2X	I/O		S	Ext. Bus Control
42	WR3X	I/O		S	Ext. Bus Control
43	AS	I/O	P90	A	Ext. Bus Control
44	ALE	I/O	P91	A	Ext. Bus Control
45	CLK	I/O		A	Ext. Bus Clk
46	AH	I/O	P93	A	Ext. Bus Control Signal
47	CS0X	I/O	P94	A	Chip select 0
48	CS1X	I/O	P95	A	Chip Select 1
49	CS2X	I/O	P96	A	Chip Select 2
50	CS3X	I/O	P97	A	Chip Select 3
51	VDD35				separated Ext.Bus VDD, 3.3 or 5.0 V
52	VSS				
53	AN8	I/O	PG0	B	ADC Input 8
54	AN9	I/O	PG1	B	ADC Input 9
55	AN10	I/O	PG2	B	ADC Input 10
56	AN11	I/O	PG3	B	ADC Input 11
57	AN12	I/O	PG4	B	ADC Input 12
58	AN13	I/O	PG5	B	ADC Input 13
59	AN14	I/O	PG6	B	ADC Input 14
60	AN15	I/O	PG7	B	ADC Input 15
61	DREQ0	I/O	PB0	A	DMA Request 0
62	DACK0	I/O	PB1	A	DMA Acknowledge 0
63	DEOP0	I/O	PB2	A	DMA EOP 0
64	AVCC				Analog VCC
65	AVRH			R	Analog Reference High
66	AN0	I/O	PH0	B	ADC Input 0
67	AN1	I/O	PH1	B	ADC Input 1
68	AN2	I/O	PH2	B	ADC Input 2

Table 1.6.2 MB91F362GB I/O pins and their function (Continued)

69	AN3	I/O	PH3	B	ADC Input 3
70	AN4	I/O	PH4	B	ADC Input 4
71	AN5	I/O	PH5	B	ADC Input 5
72	AN6	I/O	PH6	B	ADC Input 6
73	AN7	I/O	PH7	B	ADC Input 7
74	AVSS, AVRL				Analog VSS, Analog Reference Low
75	DA0	O		C	DAC Output
76	DA1	O		C	DAC Output
77	ALARM	I		D	Alarm Comparator Input
78	VSS				
79	VDD				
80	ATGX	I/O	PI3	A	ADC Trigger Input
81	TESTX	I		E	Test Input (should be connected to VDD)
82	CPUTESTX	I		E	Test Input (should be connected to VDD)
83	LTESTX	I		E	Test Input (should be connected to VDD)
84	LED0	I/O	PJ0	J	LED Port 0
85	LED1	I/O	PJ1	J	LED Port 1
86	LED2	I/O	PJ2	J	LED Port 2
87	LED3	I/O	PJ3	J	LED Port 3
88	LED4	I/O	PJ4	J	LED Port 4
89	LED5	I/O	PJ5	J	LED Port 5
90	LED6	I/O	PJ6	J	LED Port 6
91	LED7	I/O	PJ7	J	LED Port 7
92	VDD				
93	VSS				
94	INT0	I/O	PK0	A	Ext. Interrupt 0
95	INT1	I/O	PK1	A	Ext. Interrupt 1
96	INT2	I/O	PK2	A	Ext. Interrupt 2
97	INT3	I/O	PK3	A	Ext. Interrupt 3
98	INT4	I/O	PK4	A	Ext. Interrupt 4
99	INT5	I/O	PK5	A	Ext. Interrupt 5
100	INT6	I/O	PK6	A	Ext. Interrupt 6
101	INT7	I/O	PK7	A	Ext. Interrupt 7
102	IN0	I/O	PL0	A	ICU Input 0
103	IN1	I/O	PL1	A	ICU Input 1
104	IN2	I/O	PL2	A	ICU Input 2

Table 1.6.2 MB91F362GB I/O pins and their function (Continued)

105	IN3	I/O	PL3	A	ICU Input 3
106	OUT0	I/O	PL4	A	OCU Output 0
107	OUT1	I/O	PL5	A	OCU Output 1
108	OUT2	I/O	PL6	A	OCU Output 2
109	OUT3	I/O	PL7	A	OCU Output 3
110	VSS				
111	MD0	I		T	Mode Pin 0
112	MD1	I		T	Mode Pin 1
113	MD2	I		T	Mode Pin 2
114	HSTX	I		E	Hardware Standby
115	INITX	I		U	Initial Pin
116	MONCLK	O		G	System Clock Output for evaluation purposes
117	SELCLK	I		F	Clock Selection, must be connected to VDD
118	VDD				
119	X0			H	4 MHz Oscillator Pin
120	X1			H	4 MHz Oscillator Pin
121	X0A	I		I	reserved - must be connected to VSS
122	X1A	O		I	reserved - should be left open
123	VSS				
124	CPO			C	reserved - should be left open
125	VCI			D	reserved - must be connected to VSS
126	SGO	I/O	PM0	A	Sound Generator SGO
127	SGA	I/O	PM1	A	Sound Generator SGA
128	SDA	I/O	PM2	Y	I^2C SDA
129	SCL	I/O	PM3	Y	I^2C SCL
130	SOT4	I/O	PN0	A	SIO Output
131	SIN4	I/O	PN1	A	SIO Input
132	SCK4	I/O	PN2	A	SIO Clock
133	SIN3	I/O	PN3	A	SIO Input
134	SOT3	I/O	PN4	A	SIO Output
135	SCK3	I/O	PN5	A	SIO Clock
136	OCPA0	I/O	PO0	A	PPG Output
137	OCPA1	I/O	PO1	A	PPG Output
138	OCPA2	I/O	PO2	A	PPG Output
139	OCPA3	I/O	PO3	A	PPG Output
140	OCPA4	I/O	PO4	A	PPG Output

Table 1.6.2 MB91F362GB I/O pins and their function (Continued)

141	OCPA5	I/O	PO5	A	PPG Output
142	OCPA6	I/O	PO6	A	PPG Output
143	OCPA7	I/O	PO7	A	PPG Output
144	VDD				
145	VSS				
146	TX0	I/O	PP0	Q	CAN 0 TX
147	RX0	I/O	PP1	Q	CAN 0 RX
148	TX1	I/O	PP2	Q	CAN 1 TX
149	RX1	I/O	PP3	Q	CAN 1 RX
150	TX2	I/O	PP4	Q	CAN 2 TX
151	RX2	I/O	PP5	Q	CAN 2 RX
152	SIN0	I/O	PQ0	A	UART 0 Input
153	SOT0	I/O	PQ1	A	UART 0 Output
154	SIN1	I/O	PQ2	A	UART 1 Input
155	SOT1	I/O	PQ3	A	UART 1 Output
156	SIN2	I/O	PQ4	A	UART 2 Input
157	SOT2	I/O	PQ5	A	UART 2 Output
158	VSS				
159	VCC3/C			C	Bypass Capacitor Pin
160	VDD				
161	HVSS				
162	PWM1P0	I/O	PR0	K	SMC 0
163	PWM1M0	I/O	PR1	K	SMC 0
164	PWM2P0	I/O	PR2	K	SMC 0
165	PWM2M0	I/O	PR3	M	SMC 0
166	HVDD				
167	PWM1P1	I/O	PR4	K	SMC 1
168	PWM1M1	I/O	PR5	K	SMC 1
169	PWM2P1	I/O	PR6	K	SMC 1
170	PWM2M1	I/O	PR7	M	SMC 1
171	HVSS				
172	PWM1P2	I/O	PS0	K	SMC 2
173	PWM1M2	I/O	PS1	K	SMC 2
174	PWM2P2	I/O	PS2	K	SMC 2
175	PWM2M2	I/O	PS3	M	SMC 2
176	HVDD				

Table 1.6.2 MB91F362GB I/O pins and their function (Continued)

177	PWM1P3	I/O	PS4	K	SMC 3
178	PWM1M3	I/O	PS5	K	SMC 3
179	PWM2P3	I/O	PS6	K	SMC 3
180	PWM2M3	I/O	PS7	M	SMC 3
181	HVSS				
182	VDD35				separated Ext.Bus VDD, 3.3 or 5.0 V
183	D0	I/O		Q	Ext. Bus Data Bit 0
184	D1	I/O		Q	Ext. Bus Data Bit 1
185	D2	I/O		Q	Ext. Bus Data Bit 2
186	D3	I/O		Q	Ext. Bus Data Bit 3
187	D4	I/O		Q	Ext. Bus Data Bit 4
188	D5	I/O		Q	Ext. Bus Data Bit 5
189	D6	I/O		Q	Ext. Bus Data Bit 6
190	D7	I/O		Q	Ext. Bus Data Bit 7
191	D8	I/O		Q	Ext. Bus Data Bit 8
192	D9	I/O		Q	Ext. Bus Data Bit 9
193	D10	I/O		Q	Ext. Bus Data Bit 10
194	D11	I/O		Q	Ext. Bus Data Bit 11
195	D12	I/O		Q	Ext. Bus Data Bit 12
196	D13	I/O		Q	Ext. Bus Data Bit 13
197	D14	I/O		Q	Ext. Bus Data Bit 14
198	VDD35				separated Ext.Bus VDD, 3.3 or 5.0 V
199	VSS				
200	D15	I/O		Q	Ext. Bus Data Bit 15
201	D16	I/O		Q	Ext. Bus Data Bit 16
202	D17	I/O		Q	Ext. Bus Data Bit 17
203	D18	I/O		Q	Ext. Bus Data Bit 18
204	D19	I/O		Q	Ext. Bus Data Bit 19
205	D20	I/O		Q	Ext. Bus Data Bit 20
206	D21	I/O		Q	Ext. Bus Data Bit 21
207	D22	I/O		Q	Ext. Bus Data Bit 22
208	D23	I/O		Q	Ext. Bus Data Bit 23

Note: If pins VDD35 (25, 51, 182, 198) are connected to 3.3V then the external bus interface (pins 1-52, 182-208) can be operated at 3.3V levels.

1.6.3 MB91F364G I/O Pins and Their Function

Table 1.6.3 MB91F364G I/O pins and their function

Pin No	Pin Name	I/O	General Purpose IO Port	Circuit Type	Function
1	AN0	I/O	PH0	B	ADC input 0
2	AN1	I/O	PH1	B	ADC input 1
3	AN2	I/O	PH2	B	ADC input 2
4	AN3	I/O	PH3	B	ADC input 3
5	AN4	I/O	PH4	B	ADC input 4
6	AN5	I/O	PH5	B	ADC input 5
7	AVSS, AVRL				AVSS, analog reference low
8	AVRH			R	analog reference high
9	AVCC				AVCC
10	AN6	I/O	PH6	B	ADC input 6
11	AN7	I/O	PH7	B	ADC input 7
12	AN8	I/O	PG0	B	ADC input 8
13	AN9	I/O	PG1	B	ADC input 9
14	AN10	I/O	PG2	B	ADC input 10
15	AN11	I/O	PG3	B	ADC input 11
16	VSS				
17	VDD				
18	SDA	I/O	PM2	YA	I2C SDA
19	SCL	I/O	PM3	YA	I2C SCL
20	SOT0	I/O	PQ1	A	UART 0 SOT
21	SIN0	I/O	PQ0	A	UART 0 SIN
22	HSTX	I		F	hardware standby
23	NMIX	I		E	non maskable interrupt
24	SELCLK	I		F	select RTC clock
25	VDD				
26	MONCLK	O		Q1	modulated clock output
27	VSS				
28	X1A	O		I	32 kHz oscillator pin
29	X0A	I		I	32 kHz oscillator pin
30	VDD				
31	X1	O		H	4 MHz oscillator pin
32	X0	I		H	4 MHz oscillator pin

Table 1.6.3 MB91F364G I/O pins and their function (Continued)

Pin No	Pin Name	I/O	General Purpose IO Port	Circuit Type	Function
33	VSS				
34	INT0	I/O	PK0	B	external interrupt 0
35	INT1	I/O	PK1	B	external interrupt 1
36	INT2	I/O	PK2	B	external interrupt 2
37	INT3	I/O	PK3	B	external interrupt 3
38	INT4	I/O	PK4	B	external interrupt 4
39	INT5	I/O	PK5	B	external interrupt 5
40	INT6	I/O	PK6	B	external interrupt 6
41	INT7	I/O	PK7	B	external interrupt 7
42	VDD				
43	VSS				
44	IN0	I/O	PL0	B	ICU input 0 (see note 1)
45	IN1	I/O	PL1	B	ICU input 1 (see note 1)
46	IN2	I/O	PL2	B	ICU input 2 (see note 1)
47	IN3	I/O	PL3	B	ICU input 3 (see note 1)
48	OUT0	I/O	PL4	B	OCU output 0
49	OUT1	I/O	PL5	B	OCU output 1
50	OUT2	I/O	PL6	B	OCU output 2
51	OUT3	I/O	PL7	B	OCU output 3
52	VDD				
53	VSS				
54	TESTX	I		E	test input
55	CPUTESTX	I		E	test input
56	ATGX	I/O	PI3	A	ADC trigger
57	MD0	I		T	mode pin 0
58	MD1	I		T	mode pin 1
59	MD2	I		T	mode pin 2
60	INITX	I		U	inital pin
61	VDD				
62	VCC3C				pins for regulator capacitance or for external supply of core voltage
63	VCC3C				
64	VSS (#)				Don't connect to VSS in first ES series ! Leave open! See note 3

Table 1.6.3 MB91F364G I/O pins and their function (Continued)

Pin No	Pin Name	I/O	General Purpose IO Port	Circuit Type	Function
65	VDDI				separate core supply
66	VDDI				
67	VDDI				
68	BREAKX	I	BREAKX	E	EDSU break pin
69	VDD				
70	VSS				
71	RX0	I/O	PP1	Q	CAN RX
72	TX0	I/O	PP0	Q	CAN TX
73	OCPA0	I/O	PO0	A	PPG output 0
74	OCPA1	I/O	PO1	A	PPG output 1
75	OCPA2	I/O	PO2	A	PPG output 2
76	OCPA3	I/O	PO3	A	PPG output 3
77	VSS				
78	SIN5	I/O	PT0	A	USART 5 SIN
79	SCK5	I/O	PT1	A	USART 5 SCK
80	SOT5	I/O	PT2	A	USART 5 SOT
81	SOT6	I/O	PT3	A	USART 6 SOT
82	SCK6	I/O	PT4	A	USART 6 SCK
83	SIN6	I/O	PT5	A	USART 6 SIN
84	VDD				
85	VSS				
86	SIN3	I/O	PN3	A	SIO SIN
87	SOT3	I/O	PN4	A	SIO SOT
88	SCK3	I/O	PN5	A	SIO SCK
89	VSS				
90	LTESTX	I	LTESTX	E	test pin
91	VDD				
92	PR0	I/O	PR0	A	port R 0
93	PR1	I/O	PR1	A	port R 1
94	PR2	I/O	PR2	A	port R 2
95	PR3	I/O	PR3	A	port R 3
96	PR4	I/O	PR4	A	port R 4
97	PR5	I/O	PR5	A	port R 5
98	PR6	I/O	PR6	A	port R 6

Table 1.6.3 MB91F364G I/O pins and their function (Continued)

Pin No	Pin Name	I/O	General Purpose IO Port	Circuit Type	Function
99	PR7	I/O	PR7	A	port R 7
100	VSS				
101	VDD				
102	LED0	I/O	PJ0	J	LED port 0
103	LED1	I/O	PJ1	J	LED port 1
104	LED2	I/O	PJ2	J	LED port 2
105	LED3	I/O	PJ3	J	LED port 3
106	VSS				
107	LED4	I/O	PJ4	J	LED port 4
108	LED5	I/O	PJ5	J	LED port 5
109	LED6	I/O	PJ6	J	LED port 6
110	LED7	I/O	PJ7	J	LED port 7
111	VSS				
112	VDD				
113	PO4	I/O	PO4	A	port O 4
114	PO5	I/O	PO5	A	port O 5
115	PO6	I/O	PO6	A	port O 6
116	PO7	I/O	PO7	A	port O 7
117	DA0	O		C	See note 2
118	DA1	O		C	See note 2
119	VSS				
120	VDD				

Note 1: If the port L function register bits are cleared, the ICU input lines are connected with the LSYNC outputs of the LIN-USARTs.

Note 2: The pins DA1 and DA0 are also used for digital test functions. To ensure proper system function, always write ’0’ to port P data direction register DDRP[3:2] and port P function register PFRP[3:2].

Note 3: Pin 064 (VSS) will be available after redesign. In the first ES series, this pin must be left open.

1.6.4 MB91F369GA I/O Pins and Their Function

Table 1.6.4 MB91F369GA I/O pins and their functions

Pin No.	Pin Name	I/O	General Purpose IO Port	Circuit Type	Function
1	A4	I/O		Q	Ext. Bus Address Bit 4
2	A5	I/O		Q	Ext. Bus Address Bit 5
3	A6	I/O		Q	Ext. Bus Address Bit 6
4	A7	I/O		Q	Ext. Bus Address Bit 7
5	A8	I/O		Q	Ext. Bus Address Bit 8
6	A9	I/O		Q	Ext. Bus Address Bit 9
7	A10	I/O		Q	Ext. Bus Address Bit 10
8	A11	I/O		Q	Ext. Bus Address Bit 11
9	VDD35				separated Ext.Bus VDD, 3.3 or 5.0 V
10	CLK	I/O		A	Ext. Bus CLock
11	VSS				
12	A12	I/O		Q	Ext. Bus Address Bit 12
13	A13	I/O		Q	Ext. Bus Address Bit 13
14	A14	I/O		Q	Ext. Bus Address Bit 14
15	A15	I/O		Q	Ext. Bus Address Bit 15
16	A16	I/O		Q	Ext. Bus Address Bit 16
17	A17	I/O		Q	Ext. Bus Address Bit 17
18	A18	I/O		Q	Ext. Bus Address Bit 18
19	A19	I/O		Q	Ext. Bus Address Bit 19
20	A20	I/O		Q	Ext. Bus Address Bit 20
21	VDD35				separated Ext.Bus VDD, 3.3 or 5.0 V
22	VSS				
23	CS4X	I/O	P74	A	Chip Select 4
24	CS5X	I/O	P75	A	Chip Select 5
25	CS6X	I/O	P76	A	Chip Select 6
26	RDX	I/O		S	Ext. Bus Control
27	BGRNTX	I/O	P81	A	Ext. Bus Control
28	BRQ	I/O	P82	A	Ext. Bus Control
29	AS	I/O	P90	A	Ext. Bus Control
30	ALE	I/O	P91	A	Ext. Bus Control
31	AH	I/O	P93	A	Ext. Bus Control Signal
32	CS0X	I/O	P94	A	Chip select 0

Table 1.6.4 MB91F369GA I/O pins and their functions (Continued)

33	CS1X	I/O	P95	A	Chip Select 1
34	CS2X	I/O	P96	A	Chip Select 2
35	CS3X	I/O	P97	A	Chip Select 3
36	DREQ0	I/O	PB0	A	DMA Request 0
37	DACK0	I/O	PB1	A	DMA Acknowledge 0
38	DEOP0	I/O	PB2	A	DMA EOP 0
39	VSS				
40	VDD35				separated Ext.Bus VDD, 3.3 or 5.0 V
41	AVRH			R	Analog Reference High
42	AVCC				Analog VCC
43	AVSS, AVRL				Analog VSS, Analog Reference Low
44	AN0	I/O	PH0	B	ADC Input 0
45	AN1	I/O	PH1	B	ADC Input 1
46	AN2	I/O	PH2	B	ADC Input 2
47	AN3	I/O	PH3	B	ADC Input 3
48	AN4	I/O	PH4	B	ADC Input 4
49	AN5	I/O	PH5	B	ADC Input 5
50	AN6	I/O	PH6	B	ADC Input 6
51	AN7	I/O	PH7	B	ADC Input 7
52	AN8	I/O	PG0	B	ADC Input 8
53	AN9	I/O	PG1	B	ADC Input 9
54	ALARM	I		D	Alarm Comparator Input
55	VSS				
56	VDD				
57	ATGX	I/O	PI3	A	ADC Trigger Input
58	MD0	I		T	Mode Pin 0
59	MD1	I		T	Mode Pin 1
60	MD2	I		T	Mode Pin 2
61	HSTX	I		E	Hardware Standby
62	INITX	I		U	Initial Pin
63	TESTX	I		E	Test Input (should be connected to VDD)
64	CPUTESTX	I		E	Test Input (should be connected to VDD)
65	LTESTX	I		E	Test Input (should be connected to VDD)
66	VDD				
67	VSS				
68	INT0	I/O	PK0	A	Ext. Interrupt 0

Table 1.6.4 MB91F369GA I/O pins and their functions (Continued)

69	INT1	I/O	PK1	A	Ext. Interrupt 1
70	INT2	I/O	PK2	A	Ext. Interrupt 2
71	INT3	I/O	PK3	A	Ext. Interrupt 3
72	INT4	I/O	PK4	A	Ext. Interrupt 4
73	INT5	I/O	PK5	A	Ext. Interrupt 5
74	INT6	I/O	PK6	A	Ext. Interrupt 6
75	INT7	I/O	PK7	A	Ext. Interrupt 7
76	SGO	I/O	PM0	A	Sound Generator SGO
77	SGA	I/O	PM1	A	Sound Generator SGA
78	SDA	I/O	PM2	Y	I^2C SDA
79	SCL	I/O	PM3	Y	I^2C SCL
80	SOT4	I/O	PN0	A	SIO Output
81	SIN4	I/O	PN1	A	SIO Input
82	SCK4	I/O	PN2	A	SIO Clock
83	SIN3	I/O	PN3	A	SIO Input
84	SOT3	I/O	PN4	A	SIO Output
85	SCK3	I/O	PN5	A	SIO Clock
86	VSS				
87	VDDI				Supply voltage for internal regulator
88	VDDI				Supply voltage for internal regulator
89	VDDI				Supply voltage for internal regulator
90	VDDI				Supply voltage for internal regulator
91	VCC3C				Capacitor pin for internal regulator
92	VSS				
93	TX0	I/O	PP0	Q	CAN 0 TX
94	RX0	I/O	PP1	Q	CAN 0 RX
95	TX1	I/O	PP2	Q	CAN 1 TX
96	RX1	I/O	PP3	Q	CAN 1 RX
97	SIN0	I/O	PQ0	A	UART 0 Input
98	SOT0	I/O	PQ1	A	UART 0 Output
99	VDD				
100	VSS				
101	OCPA0	I/O	PO0	A	PPG Output
102	OCPA1	I/O	PO1	A	PPG Output
103	OCPA2	I/O	PO2	A	PPG Output
104	OCPA3	I/O	PO3	A	PPG Output

Table 1.6.4 MB91F369GA I/O pins and their functions (Continued)

105	VDDX				
106	X0			H	4 MHz Oscillator Pin
107	X1			H	4 MHz Oscillator Pin
108	VSS				
109	VDD				
110	MONCLK	O		Q1	System Clock Output
111	VSS				
112	VDD35				separated Ext.Bus VDD, 3.3 or 5.0 V
113	VSS				
114	WR3X	I/O		S	Ext. Bus Control
115	WR2X	I/O		S	Ext. Bus Control
116	WR1X	I/O		S	Ext. Bus Control
117	WR0X	I/O		S	Ext. Bus Control
118	RDY	I/O		S	Ext. Bus Control
119	VDD35				separated Ext.Bus VDD, 3.3 or 5.0 V
120	VSS				
121	D0	I/O		Q	Ext. Bus Data Bit 0
122	D1	I/O		Q	Ext. Bus Data Bit 1
123	D2	I/O		Q	Ext. Bus Data Bit 2
124	D3	I/O		Q	Ext. Bus Data Bit 3
125	D4	I/O		Q	Ext. Bus Data Bit 4
126	D5	I/O		Q	Ext. Bus Data Bit 5
127	D6	I/O		Q	Ext. Bus Data Bit 6
128	D7	I/O		Q	Ext. Bus Data Bit 7
129	D8	I/O		Q	Ext. Bus Data Bit 8
130	D9	I/O		Q	Ext. Bus Data Bit 9
131	D10	I/O		Q	Ext. Bus Data Bit 10
132	D11	I/O		Q	Ext. Bus Data Bit 11
133	D12	I/O		Q	Ext. Bus Data Bit 12
134	D13	I/O		Q	Ext. Bus Data Bit 13
135	D14	I/O		Q	Ext. Bus Data Bit 14
136	D15	I/O		Q	Ext. Bus Data Bit 15
137	VDD35				separated Ext.Bus VDD, 3.3 or 5.0 V
138	VSS				
139	D16	I/O		Q	Ext. Bus Data Bit 16
140	D17	I/O		Q	Ext. Bus Data Bit 17

Table 1.6.4 MB91F369GA I/O pins and their functions (Continued)

141	D18	I/O		Q	Ext. Bus Data Bit 18
142	D19	I/O		Q	Ext. Bus Data Bit 19
143	D20	I/O		Q	Ext. Bus Data Bit 20
144	D21	I/O		Q	Ext. Bus Data Bit 21
145	D22	I/O		Q	Ext. Bus Data Bit 22
146	D23	I/O		Q	Ext. Bus Data Bit 23
147	D24	I/O		Q	Ext. Bus Data Bit 24
148	D25	I/O		Q	Ext. Bus Data Bit 25
149	D26	I/O		Q	Ext. Bus Data Bit 26
150	D27	I/O		Q	Ext. Bus Data Bit 27
151	D28	I/O		Q	Ext. Bus Data Bit 28
152	D29	I/O		Q	Ext. Bus Data Bit 29
153	D30	I/O		Q	Ext. Bus Data Bit 30
154	D31	I/O		Q	Ext. Bus Data Bit 31
155	VDD35				separated Ext.Bus VDD, 3.3 or 5.0 V
156	VSS				
157	A0	I/O		Q	Ext. Bus Address Bit 0
158	A1	I/O		Q	Ext. Bus Address Bit 1
159	A2	I/O		Q	Ext. Bus Address Bit 2
160	A3	I/O		Q	Ext. Bus Address Bit 3

Note: If pins VDD35 (9, 21, 40, 112, 119, 137, 155) are connected to a 3.3V supply the external bus interface (pins 1-40, 112-160) can be operated at 3.3V levels.

1.6.5 MB91F365GB/F366GB/F376G, MB91366GA I/O Pins

Table 1.6.5 MB91F365GB/F366GB/F376G, MB91366GA I/O Pins and their function

Pin No. QFP120	Pin Name	I/O	General Purpose I/O Port	Circuit Type on Flash Device	Circuit Type on ROM Device	Function
1	VDD					
2	VSS					
3	PJ4	I/O	PJ4	A	A	Digital IO-Port
4	PJ5	I/O	PJ5	A	A	Digital IO-Port
5	PJ6	I/O	PJ6	A	A	Digital IO-Port
6	PJ7	I/O	PJ7	A	A	Digital IO-Port
7	PI3	I/O	PI3	A	A	Digital IO-Port
8	VDD					
9	VSS					
10	SGO	I/O	PM0	A	A	Sound Gen. SGO
11	SGA	I/O	PM1	A	A	Sound Gen. SGA
12	SDA	I/O	PM2	Y	Y	I2C SDA (no internal pull-up!)
13	SCL	I/O	PM3	Y	Y	I2C SCL (no internal pull-up!)
14	VDD					
15	VSS					
16	AVRH			R	R	Analog Voltage Ref. high
17	AVCC					Analog VCC
18	AVSS/ AVRL					Ana.Volt.Ref.low/An.VSS
19	AN0	I/O	PH0	B	B	ADC input
20	AN1	I/O	PH1	B	B	ADC input
21	AN2	I/O	PH2	B	B	ADC input
22	AN3	I/O	PH3	B	B	ADC input
23	AN4	I/O	PH4	B	B	ADC input
24	AN5	I/O	PH5	B	B	ADC input
25	AN6	I/O	PH6	B	B	ADC input
26	AN7	I/O	PH7	B	B	ADC input
27	DA0	O		C	C	DAC Output (MB91F365GB)
	X0A	I		I	I	32 kHz Osc (MB91F366GB/366GA/ MB91F376G)

Table 1.6.5 MB91F365GB/F366GB/F376G, MB91366GA I/O Pins and their function (Continued)

Pin No. QFP120	Pin Name	I/O	General Purpose I/O Port	Circuit Type on Flash Device	Circuit Type on ROM Device	Function
28	DA1	O		C	C	DAC Output (MB91F365GB)
	X1A	O		I	I	32 kHz Osc. (MB91F366GB/366GA/ MB91F376G)
29	ALARM	I		D	D	Alarm Comparator Input
30	VSS					
31	BOOT	I/O	P93	A	A	BOOT pin (see remark)
32	TESTX	I		E	E	Test mode pin
33	CPUTESTX	I		E	E	Test mode pin
34	VDD					
35	X0	I		H	H	4 MHz Oscillator Pin
36	X1	O		H	H	4 MHz Oscillator Pin
37	VSS					
38	MONCLK	O		G	G	Clock output
39	INT0	I/O	PK0	A	A	Ext. Interrupt
40	INT1	I/O	PK1	A	A	Ext. Interrupt
41	INT2	I/O	PK2	A	A	Ext. Interrupt
42	INT3	I/O	PK3	A	A	Ext. Interrupt
43	INT4	I/O	PK4	A	A	Ext. Interrupt
44	INT5	I/O	PK5	A	A	Ext. Interrupt
45	INT6	I/O	PK6	A	A	Ext. Interrupt
46	INT7	I/O	PK7	A	A	Ext. Interrupt
47	VDD					supply pin for internal voltage regulator
48	VCC3/C					Capacitor pin for internal voltage reg.
49	VSS					
50	IN0	I/O	PL0	A	A	ICU input
51	IN1	I/O	PL1	A	A	ICU input
52	IN2	I/O	PL2	A	A	ICU input
53	IN3	I/O	PL3	A	A	ICU input
54	OUT0	I/O	PL4	A	A	OCU Output
55	OUT1	I/O	PL5	A	A	OCU Output
56	VDD					supply pin for internal voltage regulator
57	MD0	I		T	F	Mode Pin
58	MD1	I		T	F	Mode Pin

Table 1.6.5 MB91F365GB/F366GB/F376G, MB91366GA I/O Pins and their function (Continued)

Pin No. QFP120	Pin Name	I/O	General Purpose I/O Port	Circuit Type on Flash Device	Circuit Type on ROM Device	Function
59	MD2	I		T	F	Mode Pin
60	INITX	I		U	U	Initial
61	VDD					supply pin for internal voltage regulator
62	VSS					
63	SOT4	I/O	PN0	A	A	SIO output
64	SIN4	I/O	PN1	A	A	SIO input
65	SCK4	I/O	PN2	A	A	SIO clock
66	SIN3	I/O	PN3	A	A	SIO input
67	SOT3	I/O	PN4	A	A	SIO output
68	SCK3	I/O	PN5	A	A	SIO clock
69	VSS					
70	OCPA0	I/O	PO0	A	A	PPG output
71	OCPA1	I/O	PO1	A	A	PPG output
72	OCPA2	I/O	PO2	A	A	PPG output
73	OCPA3	I/O	PO3	A	A	PPG output
74	OCPA4	I/O	PO4	A	A	PPG output
75	OCPA5	I/O	PO5	A	A	PPG output
76	OCPA6	I/O	PO6	A	A	PPG output
77	OCPA7	I/O	PO7	A	A	PPG output
78	TX0	I/O	PP0	Q	Q	CAN TX output
79	RX0	I/O	PP1	Q	Q	CAN RX output
80	TX1	I/O	PP2	Q	Q	CAN TX output
81	RX1	I/O	PP3	Q	Q	CAN RX output
82	VDD					
83	VSS					
84	SIN0	I/O	PQ0	A	A	UART input
85	SOT0	I/O	PQ1	A	A	UART output
86	SIN1	I/O	PQ2	A	A	UART input
87	SOT1	I/O	PQ3	A	A	UART output
88	PG0	I/O	PG0	A	A	Digital IO-Port
89	PG1	I/O	PG1	A	A	Digital IO-Port
90	PG2	I/O	PG2	A	A	Digital IO-Port

Table 1.6.5 MB91F365GB/F366GB/F376G, MB91366GA I/O Pins and their function (Continued)

Pin No. QFP120	Pin Name	I/O	General Purpose I/O Port	Circuit Type on Flash Device	Circuit Type on ROM Device	Function
91	PG3	I/O	PG3	A	A	Digital IO-Port
92	PG4	I/O	PG4	A	A	Digital IO-Port
93	PG5	I/O	PG5	A	A	Digital IO-Port
94	VDD					
95	HVSS					SMC VSS
96	PWM1P0	I/O	PR0	K	K	SMC 0
97	PWM1M0	I/O	PR1	K	K	SMC 0
98	PWM2P0	I/O	PR2	K	K	SMC 0
99	PWM2M0	I/O	PR3	M	M	SMC 0
100	HVDD					SMC VDD
101	PWM1P1	I/O	PR4	K	K	SMC 1
102	PWM1M1	I/O	PR5	K	K	SMC 1
103	PWM2P1	I/O	PR6	K	K	SMC 1
104	PWM2M1	I/O	PR7	M	M	SMC 1
105	HVSS					SMC VSS
106	PWM1P2	I/O	PS0	K	K	SMC 2
107	PWM1M2	I/O	PS1	K	K	SMC 2
108	PWM2P2	I/O	PS2	K	K	SMC 2
109	PWM2M2	I/O	PS3	M	M	SMC 2
110	HVDD					SMC VDD
111	PWM1P3	I/O	PS4	K	K	SMC 3
112	PWM1M3	I/O	PS5	K	K	SMC 3
113	PWM2P3	I/O	PS6	K	K	SMC 3
114	PWM2M3	I/O	PS7	M	M	SMC 3
115	HVSS					
116	VDD					
117	PJ0	I/O	PJ0	A	A	Digital IO-Port
118	PJ1	I/O	PJ1	A	A	Digital IO-Port
119	PJ2	I/O	PJ2	A	A	Digital IO-Port
120	PJ3	I/O	PJ3	A	A	Digital IO-Port

Remark: (1) Pin 31 (BOOT) should be low by default (pull down resistor). To avoid disturbances in case of reset/boot it should preferably only be used as output by any application.

1.6.6 MB91F367GB/F368GB I/O Pins and Their Function

Table 1.6.6 MB91F367GB/F368GB I/O Pins and their functions

Pin No. QFP120	Pin Name	I/O	General Purpose I/O Port	Circuit Type	Function
1	VDD				
2	VSS				
3	PJ4	I/O	PJ4	A	Digital IO-Port
4	PJ5	I/O	PJ5	A	Digital IO-Port
5	PJ6	I/O	PJ6	A	Digital IO-Port
6	PJ7	I/O	PJ7	A	Digital IO-Port
7	PI3	I/O	PI3	A	Digital IO-Port
8	VDD				
9	VSS				
10	PM0	I/O	PM0	A	Digital IO-Port
11	PM1	I/O	PM1	A	Digital IO-Port
12	SDA	I/O	PM2	Y	I2C SDA (no internal pull-up!)
13	SCL	I/O	PM3	Y	I2C SCL (no internal pull-up!)
14	VDD				
15	VSS				
16	AVRH			R	Analog Voltage Ref. high
17	AVCC				Analog VCC
18	AVSS/ AVRL				Ana.Volt.Ref.low/An.VSS
19	AN0	I/O	PH0	B	ADC input
20	AN1	I/O	PH1	B	ADC input
21	AN2	I/O	PH2	B	ADC input
22	AN3	I/O	PH3	B	ADC input
23	AN4	I/O	PH4	B	ADC input
24	AN5	I/O	PH5	B	ADC input
25	AN6	I/O	PH6	B	ADC input
26	AN7	I/O	PH7	B	ADC input
27	X0A	I		I	32 KHz Oscillator Pin (MB91F368GB)
	N.C.				not connected (MB91F367GB)
28	X1A	O		I	32 KHz Oscillator Pin (MB91F368GB)
	N.C.				not connected (MB91F367GB)

Table 1.6.6 MB91F367GB/F368GB I/O Pins and their functions (Continued)

Pin No. QFP120	Pin Name	I/O	General Purpose I/O Port	Circuit Type	Function
29	ALARM	I		D	Alarm Comparator Input
30	VSS				
31	BOOT	I/O	P93	A	BOOT pin (see remark!)
32	TESTX	I		E	Test mode pin
33	CPUT-ESTX	I		E	Test mode pin
34	VDD				
35	X0	I		H	4 MHz Oscillator Pin
36	X1	O		H	4 MHz Oscillator Pin
37	VSS				
38	MONCLK	O		G	Clock output
39	INT0	I/O	PK0	A	Ext. Interrupt
40	INT1	I/O	PK1	A	Ext. Interrupt
41	INT2	I/O	PK2	A	Ext. Interrupt
42	INT3	I/O	PK3	A	Ext. Interrupt
43	INT4	I/O	PK4	A	Ext. Interrupt
44	INT5	I/O	PK5	A	Ext. Interrupt
45	INT6	I/O	PK6	A	Ext. Interrupt
46	INT7	I/O	PK7	A	Ext. Interrupt
47	VDD				supply pin for internal voltage regulator
48	VCC3/C				Capacitor pin for V. reg.
49	VSS				
50	IN0	I/O	PL0	A	ICU input
51	IN1	I/O	PL1	A	ICU input
52	IN2	I/O	PL2	A	ICU input
53	IN3	I/O	PL3	A	ICU input
54	OUT0	I/O	PL4	A	OCU Output
55	OUT1	I/O	PL5	A	OCU Output
56	VDD				supply pin for internal voltage regulator
57	MD0	I		T	Mode Pin
58	MD1	I		T	Mode Pin
59	MD2	I		T	Mode Pin
60	INITX	I		U	Initial

Table 1.6.6 MB91F367GB/F368GB I/O Pins and their functions (Continued)

Pin No. QFP120	Pin Name	I/O	General Purpose I/O Port	Circuit Type	Function
61	VDD				supply pin for internal voltage regulator
62	VSS				
63	SOT4	I/O	PN0	A	SIO output
64	SIN4	I/O	PN1	A	SIO input
65	SCK4	I/O	PN2	A	SIO clock
66	SIN3	I/O	PN3	A	SIO input
67	SOT3	I/O	PN4	A	SIO output
68	SCK3	I/O	PN5	A	SIO clock
69	VSS				
70	OCPA0	I/O	PO0	A	PPG output
71	OCPA1	I/O	PO1	A	PPG output
72	OCPA2	I/O	PO2	A	PPG output
73	OCPA3	I/O	PO3	A	PPG output
74	PO4	I/O	PO4	A	Digital IO-Port
75	PO5	I/O	PO5	A	Digital IO-Port
76	PO6	I/O	PO6	A	Digital IO-Port
77	PO7	I/O	PO7	A	Digital IO-Port
78	TX0	I/O	PP0	Q	CAN TX output
79	RX0	I/O	PP1	Q	CAN RX output
80	TX1	I/O	PP2	Q	CAN TX output
81	RX1	I/O	PP3	Q	CAN RX output
82	VDD				
83	VSS				
84	SIN0	I/O	PQ0	A	UART input
85	SOT0	I/O	PQ1	A	UART output
86	PQ2	I/O	PQ2	A	Digital IO-Port
87	PQ3	I/O	PQ3	A	Digital IO-Port
88	PG0	I/O	PG0	A	Digital IO-Port
89	PG1	I/O	PG1	A	Digital IO-Port
90	PG2	I/O	PG2	A	Digital IO-Port
91	PG3	I/O	PG3	A	Digital IO-Port
92	PG4	I/O	PG4	A	Digital IO-Port
93	PG5	I/O	PG5	A	Digital IO-Port

Table 1.6.6 MB91F367GB/F368GB I/O Pins and their functions (Continued)

Pin No. QFP120	Pin Name	I/O	General Purpose I/O Port	Circuit Type	Function
94	VDD				
95	VSS				
96	PR0	I/O	PR0	K	Digital IO-Port
97	PR1	I/O	PR1	K	Digital IO-Port
98	PR2	I/O	PR2	K	Digital IO-Port
99	PR3	I/O	PR3	M	Digital IO-Port
100	HVDD				VDD for ports R and S
101	PR4	I/O	PR4	K	Digital IO-Port
102	PR5	I/O	PR5	K	Digital IO-Port
103	PR6	I/O	PR6	K	Digital IO-Port
104	PR7	I/O	PR7	M	Digital IO-Port
105	VSS				
106	PS0	I/O	PS0	K	Digital IO-Port
107	PS1	I/O	PS1	K	Digital IO-Port
108	PS2	I/O	PS2	K	Digital IO-Port
109	PS3	I/O	PS3	M	Digital IO-Port
110	HVDD				VDD for ports R and S
111	PS4	I/O	PS4	K	Digital IO-Port
112	PS5	I/O	PS5	K	Digital IO-Port
113	PS6	I/O	PS6	K	Digital IO-Port
114	PS7	I/O	PS7	M	Digital IO-Port
115	VSS				
116	VDD				
117	PJ0	I/O	PJ0	A	Digital IO-Port
118	PJ1	I/O	PJ1	A	Digital IO-Port
119	PJ2	I/O	PJ2	A	Digital IO-Port
120	PJ3	I/O	PJ3	A	Digital IO-Port

Remark: Pin 31 (BOOT) should be low by default (pull down resistor). To avoid disturbances in case of reset/boot it should preferably only be used as output by any application.

1.6.7 I/O Circuit Type List

The circuit types, listed in the preceeding tables, have the following properties:

Table 1.6.7 I/O circuit type list

Circuit Type	Description
A	I/O, IOH=4 mA / IOL=4 mA, CMOS Automotive Schmitt-Trigger Input, STOP control
B	I/O, IOH=4 mA / IOL=4 mA, CMOS Automotive Schmitt-Trigger Input, Analog Input, STOP control
C	Analog Output
D	Analog Input
E	CMOS Schmitt-Trigger Input, 50K Pull-up
F	CMOS Schmitt-Trigger Input
G	Tristate Output, IOH=4 mA / IOL=4 mA
H	4 MHz Oscillator Pin
I	32 kHz Oscillator pin
J	I/O, IOH=14 mA / IOL = 24 mA, CMOS Automotive Schmitt-Trigger Input, STOP control (LED)
K	I/O, IOH=30 mA / IOL=30 mA, CMOS Automotive Schmitt-Trigger Input, STOP control,slew rate improved for EMC (SMC)
L	I/O, IOH=4 mA / IOL=4 mA, CMOS Input; 5 V or 3 V input
M	I/O, IOH=30 mA / IOL=30 mA, CMOS Automotive Schmitt-Trigger Input, Analog Input, STOP control, slew rate improved for EMC (SMC)
N	I/O, IOH=4 mA / IOL=4 mA, CMOS Input, 50K Pulldown; 5 V or 3 V input
O	CMOS Input, 50K Pulldown; 5 V or 3 V input
P	CMOS Input; 3 V input
Q	I/O, IOH=4 mA / IOL=4 mA, CMOS Input, STOP control
Q1	I/O, IOH=8 mA / IOL=8 mA, CMOS Input, STOP control
R	AVRL / AVRH Input
S	I/O, IOH=4 mA / IOL=4 mA, CMOS Input, STOP control, 10K Pull-up Resistor
T	CMOS Input, can withstand V_{ID} for flash programming
U	CMOS Schmitt-Trigger Input, 50K Pull-up, 3.3 V and 5 V inputs to core
W	I/O, IOH=4 mA / IOL=4 mA, CMOS Input; 3 V input
X	Tristate Output , IOH=4 mA / IOL=4 mA, 3 V
Y	I/O, IOH=3mA / IOL=3mA (I^2C), CMOS Input, STOP control
YA	I/O, IOH=3mA / IOL=3mA (I^2C), CMOS Schmitt Trigger Input, STOP control

1.7 I/O CIRCUIT TYPES

I/O circuit types for the terminals in chapter 1.6 are shown in Table 1.7

Table 1.7 I/O Circuit Types

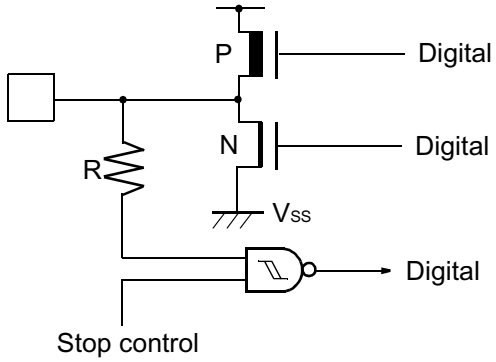
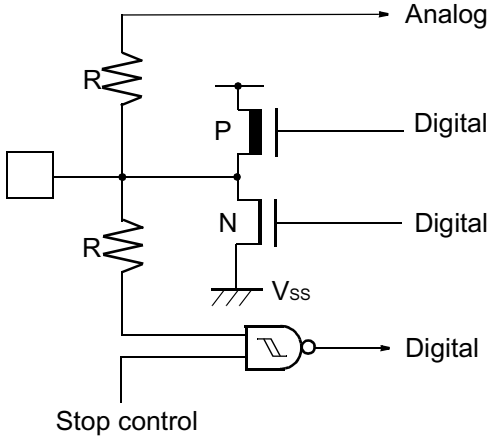
Type	Circuit type	Remarks
	Note: Symbols used in circuit types (Common to all circuit diagrams) P: P channel transistor N: N channel transistor R: Diffusion resistor	
A		• I/O, CMOS Automotive Schmitt-Trigger Input, STOP control, IOH = 4 mA, IOL = 4 mA
B		• I/O, CMOS Automotive Schmitt-Trigger Input, Analog Input, STOP control, IOH = 4 mA, IOL = 4 mA

Table 1.7 I/O Circuit Types (Continued)

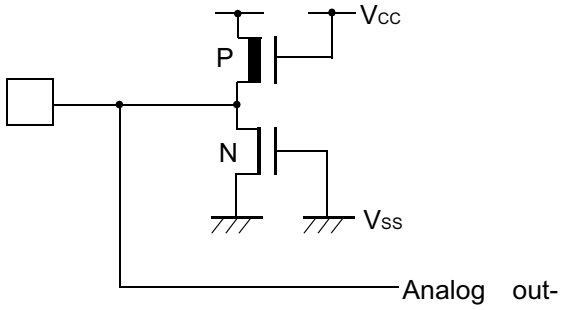
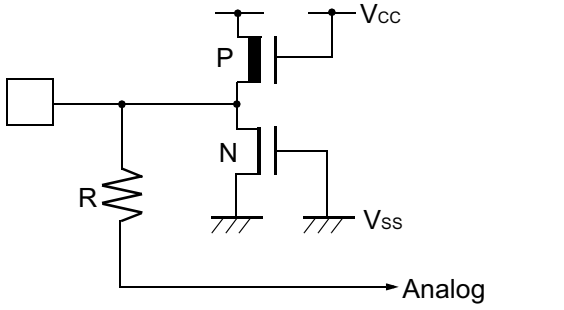
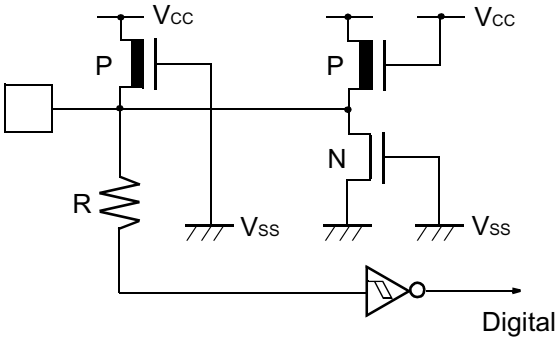
Type	Circuit type	Remarks
	Note: Symbols used in circuit types (Common to all circuit diagrams) P: P channel transistor N: N channel transistor R: Diffusion resistor	
C		• Analog output
D		• Analog Input
E		• CMOS Schmitt-Trigger Input, 50K Pullup

Table 1.7 I/O Circuit Types (Continued)

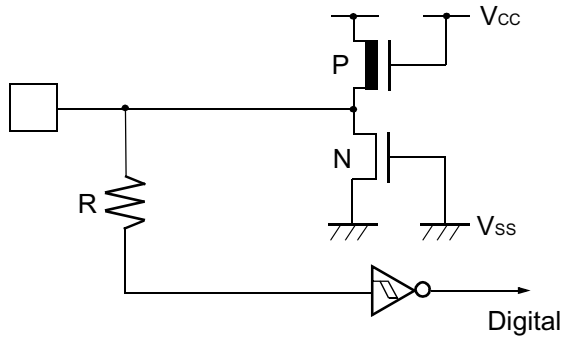
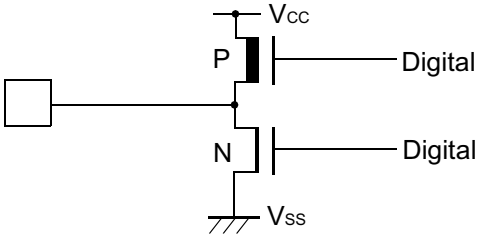
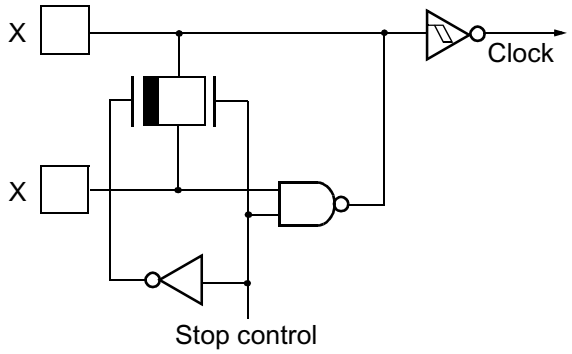
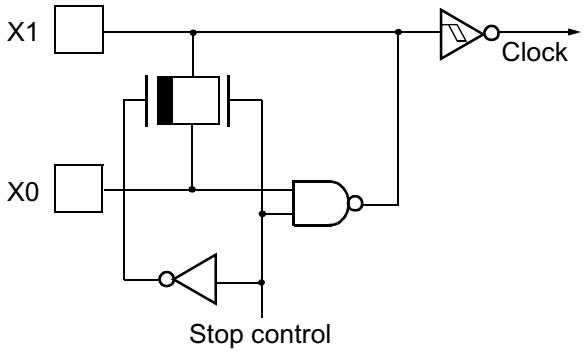
Type	Circuit type	Remarks
	Note: Symbols used in circuit types (Common to all circuit diagrams) P: P channel transistor N: N channel transistor R: Diffusion resistor	
F		• CMOS Schmitt-Trigger Input
G		• Tristate Output, IOH = 4 mA, IOL = 4 mA
H		• 4 MHz Oscillator Pin
I		• 32 kHz Oscillator Pin

Table 1.7 I/O Circuit Types (Continued)

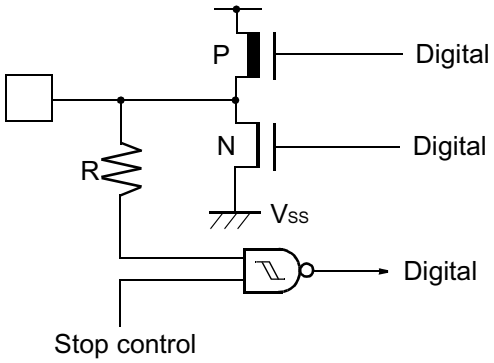
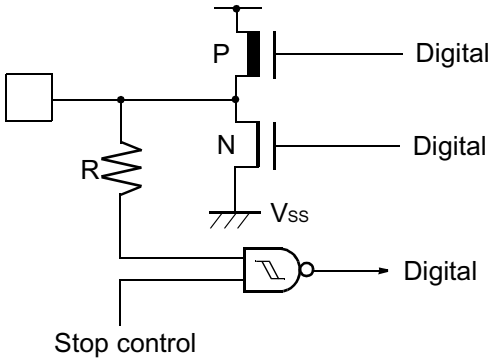
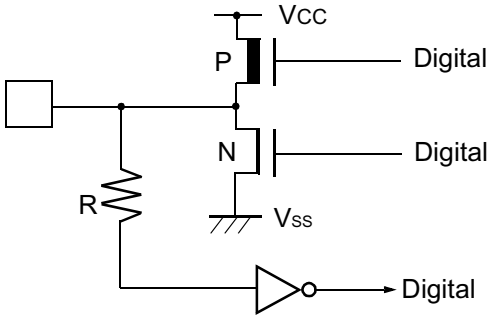
Type	Circuit type	Remarks
	Note: Symbols used in circuit types (Common to all circuit diagrams) P: P channel transistor N: N channel transistor R: Diffusion resistor	
J		• I/O, CMOS Automotive Schmitt-Trigger Input, STOP control (LED), IOH = 14 mA, IOL = 24 mA
K		• I/O, CMOS Automotive Schmitt-Trigger Input, STOP control (SMC), IOH = 30 mA, IOL = 30 mA • typ. slew rate of 40 ns
L		• I/O, CMOS Input; 5 V or 3 V input, IOH = 4 mA, IOL = 4 mA

Table 1.7 I/O Circuit Types (Continued)

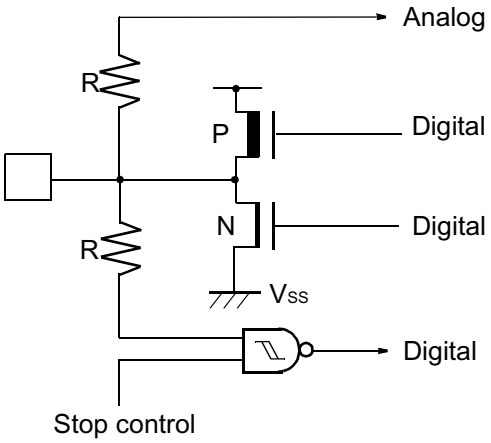
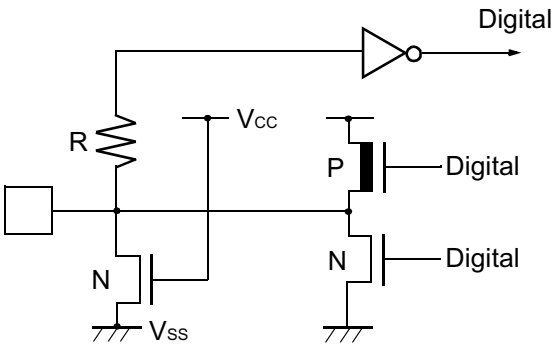
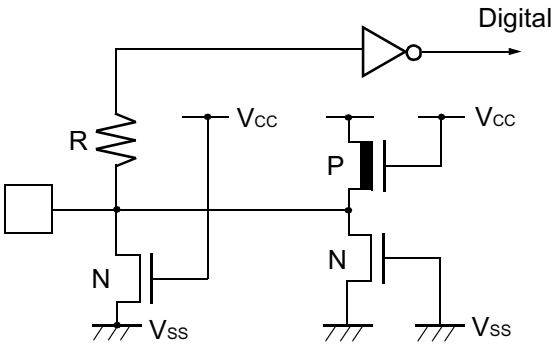
Type	Circuit type	Remarks
	Note: Symbols used in circuit types (Common to all circuit diagrams) P: P channel transistor N: N channel transistor R: Diffusion resistor	
M		• I/O, CMOS Automotive Schmitt-Trigger Input, Analog Input, STOP control (SMC), IOH = 30 mA, IOL = 30 mA • typ. slew rate of 40 ns
N		• I/O, CMOS Input, 50K Pulldown; 5 V or 3 V input, IOH = 4 mA, IOL = 4 mA
O		• CMOS Input, 50K Pulldown; 5 V or 3 V input

Table 1.7 I/O Circuit Types (Continued)

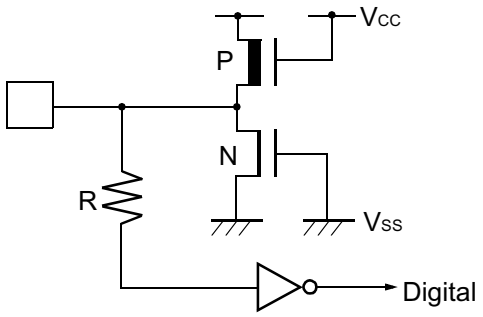
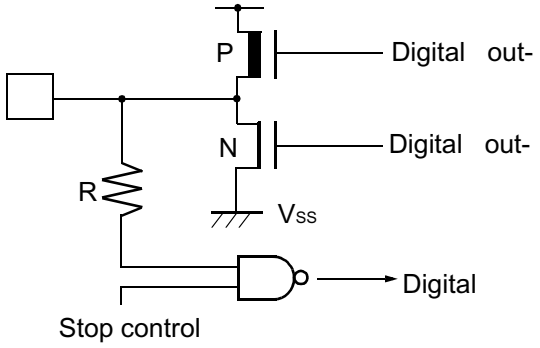
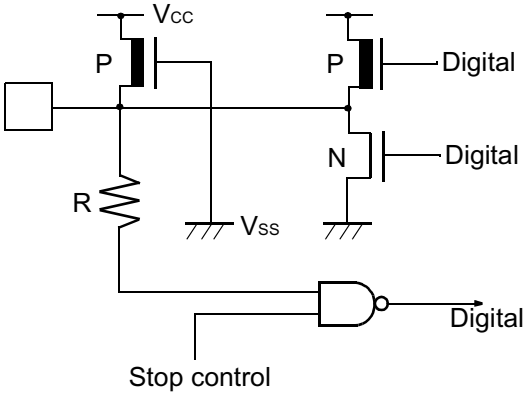
Type	Circuit type	Remarks
	Note: Symbols used in circuit types (Common to all circuit diagrams) P: P channel transistor N: N channel transistor R: Diffusion resistor	
P		• CMOS Input; 3 V input
Q/Q1		• Q: I/O CMOS Input, STOP control, IOH = 4 mA, IOL = 4 mA • • • Q1: I/O CMOS Input, STOP control, IOH = 8 mA, IOL = 8 mA
S		• I/O, CMOS Input, STOP control, 10K Pullup Resistor, IOH = 4 mA, IOL = 4 mA

Table 1.7 I/O Circuit Types (Continued)

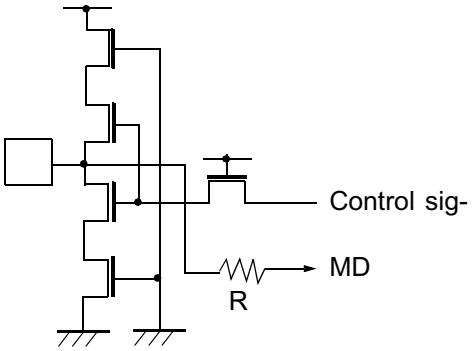
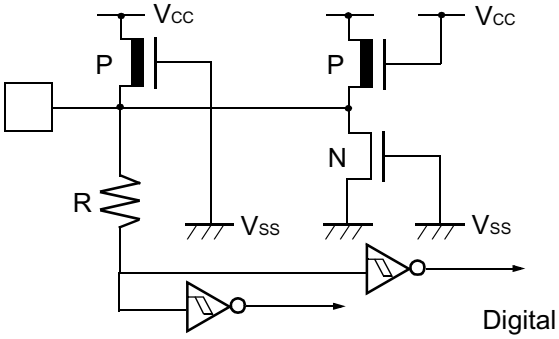
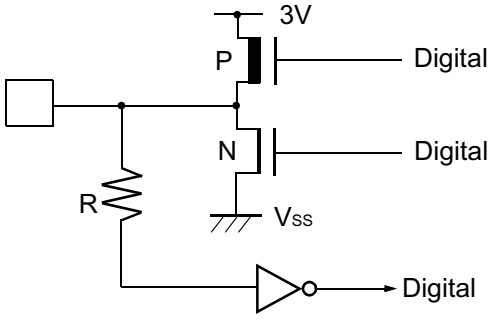
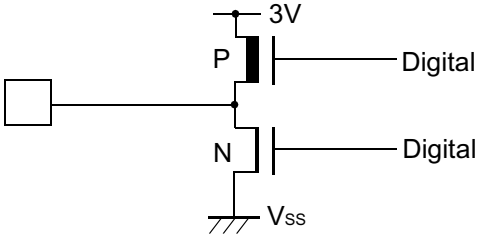
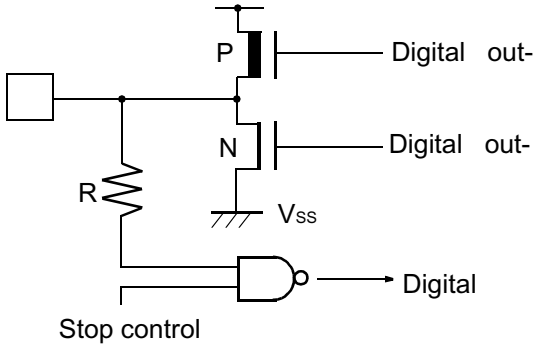
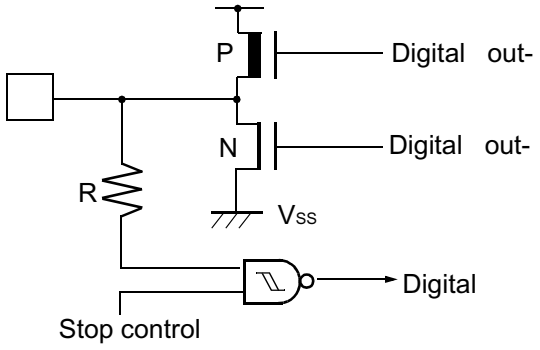
Type	Circuit type	Remarks
	Note: Symbols used in circuit types (Common to all circuit diagrams) P: P channel transistor N: N channel transistor R: Diffusion resistor	
T		• CMOS Input • can withstand high VID for flash programming
U		• CMOS Schmitt-Trigger Input, 50K Pullup 3V and 5V input to the core
W		• I/O, CMOS Input; 3 V input
X		• Tristate Output, 3 V

Table 1.7 I/O Circuit Types (Continued)

Type	Circuit type	Remarks
	Note: Symbols used in circuit types (Common to all circuit diagrams) P: P channel transistor N: N channel transistor R: Diffusion resistor	
Y		• I/O CMOS Input, STOP control , IOH = 3 mA, IOL = 3 mA, in I^2C mode operating as open drain outputs
YA		• I/O CMOS Schmitt-Trigger Input, STOP control , IOH = 3 mA, IOL = 3 mA, in I^2C mode operating as open drain outputs

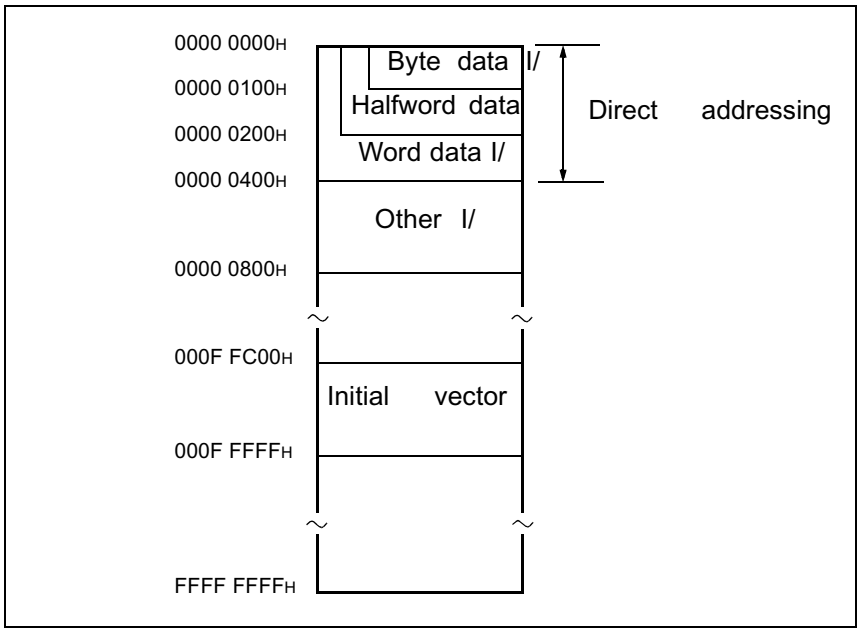
1.8 MEMORY SPACE

The FR50 series has a 4-Gbyte logical address space (2^{32} locations) available from the CPU by linear access.

■ Memory Map

Figure 1.8 shows the memory space of the MB91360.

Figure 1.8 MB91360 Memory Map



● Direct addressing areas

The address space areas listed below are the direct addressing areas used for I/O. You can specify addresses in these areas directly in instruction operands.

The direct area differs as follows depending on the size of the data accessed.

- Byte data access: 0 to 0FFH
- Halfword data access: 0 to 1FFH
- Word data access: 0 to 3FFH

1.9 HANDLING DEVICES

This section describes a number of precautions to observe when using the device.

■ **Preventing latch-up**

Latch-up may occur in a CMOS IC if a voltage greater than V_{DD} or less than V_{SS} is applied to an input or output pin or if the voltage applied between V_{DD} and V_{SS} exceeds the rating. If latch-up occurs, the power supply current increases rapidly resulting in thermal damage to circuit elements. Therefore, ensure that maximum ratings are not exceeded in circuit operation.

■ **Connecting unused pins**

Leaving unused input pins open may result in misbehaviour or latch up and possible permanent damage of the device. Therefore they must be tied to V_{DD} or V_{SS} through resistors. In this case those resistors should be more than 2 KOhm.

Unused bidirectional pins should be set to the output state and can be left open, or the input state with the above described connection.

The resistor of more than 2 KOhm is used to limit currents through the protection diodes. In case of voltages at the not used pin of 0.3 V or more below V_{SS} or 0.3 V or more above V_{DD} currents which could cause latch-up will flow through those diodes.

■ **External reset input**

When inputting an "L" level to the INITX pin, hold this low level at the INITX pin long enough so that after release of the low level at INITX and the passing of the built in waiting time stable oscillation of the oscillation circuit is achieved. INITX must be pulled low for at least 8 cycles of the 4 MHz oscillation clock.

■ **Power supply pins**

All V_{DD} pins should be connected to the same potential (exception can be the external bus interface on F362GA and F369GA). The analogue supply voltage (AVCC) must not be turned on before the digital supply voltage. If the external bus interface is supplied with 3.3 V this voltage also must not be turned on before the 5V digital voltage has been switched on. If the supply voltage to the external bus interface is switched off (it may not be tristate but should be pulled low) it must be made sure that all related signals do not have a voltage higher than this pulled down supply.

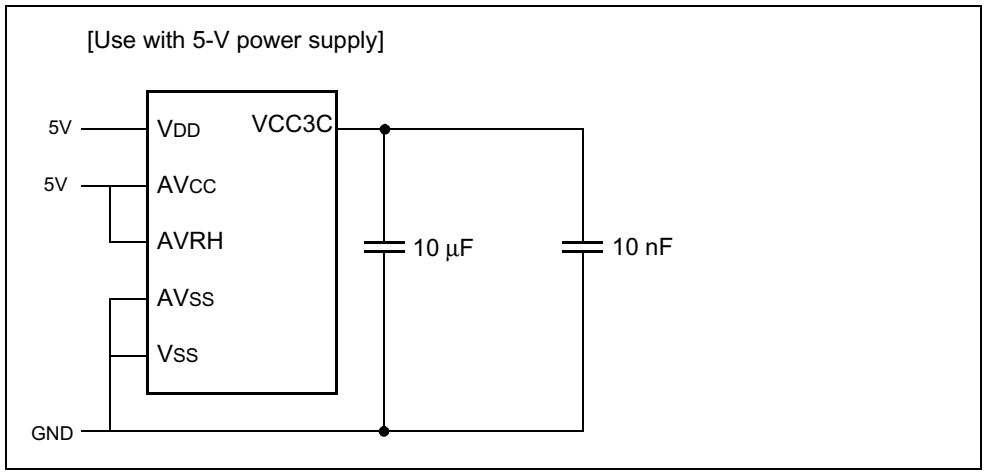
When multiple V_{DD} and V_{SS} pins are provided, be sure to connect all V_{DD} and V_{SS} pins to the power supply or ground externally. Although pins at the same potential are connected together in the internal device design so as to prevent malfunctions such as latch-up, connecting all V_{DD} and V_{SS} pins appropriately minimizes unwanted radiation, prevents malfunction of strobe signals due to increases in the ground level, and keeps the overall output current rating.

Also, take care to connect V_{DD} and V_{SS} to current source in the lowest possible impedance.

Connection of a ceramic bypass capacitor of approximately 0.1 µF between V_{DD} and V_{SS} close to the device is recommended.

The MB91360 contains a regulator. To use the device with the 5-V power supply, supply 5-V power to the Vcc pins and be sure to connect a bypass capacitor of 10 µF parallel to 10 nF to the VCC3/C pin for the regulator.

Figure 1.9a Example of Power Supply Connection



■ Crystal oscillator circuit

Noise in the vicinity of the X0 and X1 pins can be a cause of device malfunction. Design the circuit board so that X0, X1, the crystal oscillator (or ceramic oscillator), and the bypass capacitor to ground are located as close to the device as possible.

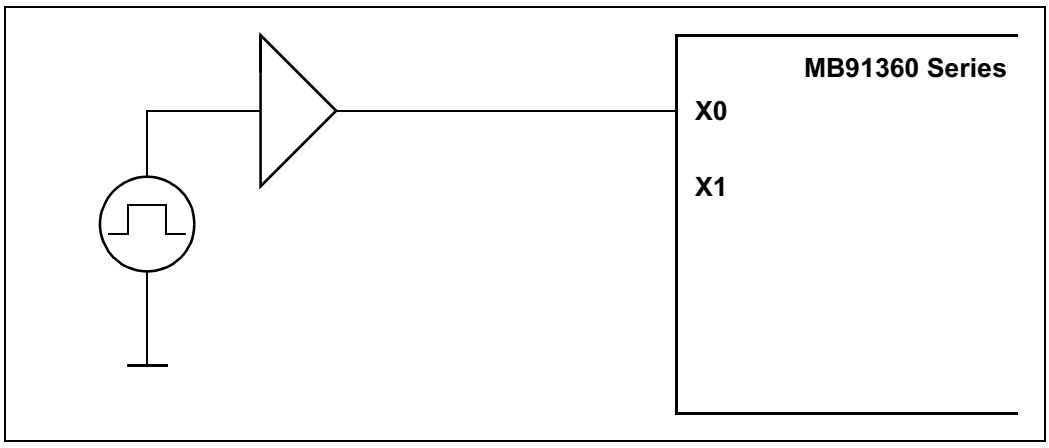
A printed circuit board design that surrounds the X0 and X1 pins with ground provides for stable operation and is strongly recommended.

■ Using an external clock

To use an external clock, drive X0 pin only and leave X1 pin open.

Below is a diagram of how to use external clock.

Figure 1.9b Example of using external clock



■ Mode pins

Connect the mode pins (MD0 to MD2) directly to V_{DD} or V_{SS}.

To prevent the device unintentionally entering test mode due to noise, lay out the printed circuit board so as to minimize the distance from the mode pins to V_{DD} or V_{SS} and to provide a low-impedance connection.

■ **Turning the power supply on**

Immediately after power on always execute INIT at the INITX pin (start with a low level at the INITX pin). Hold this low level at the INITX pin long enough so that after release of the low level at INITX and the passing of the built in waiting time stable oscillation of the oscillation circuit is achieved. INITX must be pulled low for at least 8 cycles of the 4 MHz oscillation clock.

The analogue supply voltage (AVCC) must not be turned on before the digital supply voltage. If the external bus interface is supplied with 3.3 V this voltage also must not be turned on before the 5V digital voltage has been switched on.

■ **Status during power-on**

As long as the minimum operating voltage has not been reached during power-on the output pin levels are not guaranteed.

■ **Note on during operation of PLL clock mode**

If the PLL clock mode is selected, the microcontroller attempt to be working with the self-oscillating circuit even when there is no external oscillator or external clock input is stopped. Performance of this operation, however, cannot be guaranteed.

■ **The function of the watchdog timer**

The watchdog timer in this model has the functions watching that the program performs the delay of reset within a fixed period and resetting the CPU when the delay of reset is not performed because of the program malfunction.

Therefore, once the function of the watchdog timer is enabled, the watchdog timer keeps on watching until the reset operation.

As a exceptional processing, the watchdog timer performs the delay of reset automatically under the condition in which the CPU program operation is stopped. Please refer to the explanation item of the function of the watchdog timer about the exceptional codition.

By the way, if above codition will be issued by the system program or hardware malfunction, a watchdog reset may be not performed. In this case please perform the reset operation (INIT) by using the external INITX pin.

CHAPTER 2 CPU

This chapter describes key information about the functions of the FR50 series CPU core including the architecture, specifications, and instructions.

2.1 CPU ARCHITECTURE

The FR50 CPU is a high performance CPU core with a RISC architecture. The CPU features additional high-level instructions suitable for embedded applications.

■ Features

- RISC architecture

  Basic instructions are executed at one instruction per cycle (high-speed processing using a five-stage pipeline).

- 32-bit architecture

  General-purpose registers : 16 × 32 bits

- 4-GByte linear memory space
- Internal multiplier

  32-bit × 32-bit multiplication: 5 cycles

  16-bit × 16-bit multiplication: 3 cycles

- Enhanced interrupt processing function

  Fast response speed (6 cycles)

  Supports multiple interrupts

  Level masking function (16 levels)

- Enhanced I/O manipulation instructions

  Memory-to-memory transfer instructions

  Bit manipulation instructions

- High code efficiency

  Basic instruction word length: 16 bits

- Low-power consumption

  Sleep mode and stop mode

■ FR50 CPU Specifications

Table 2.1 FR50 CPU Specifications

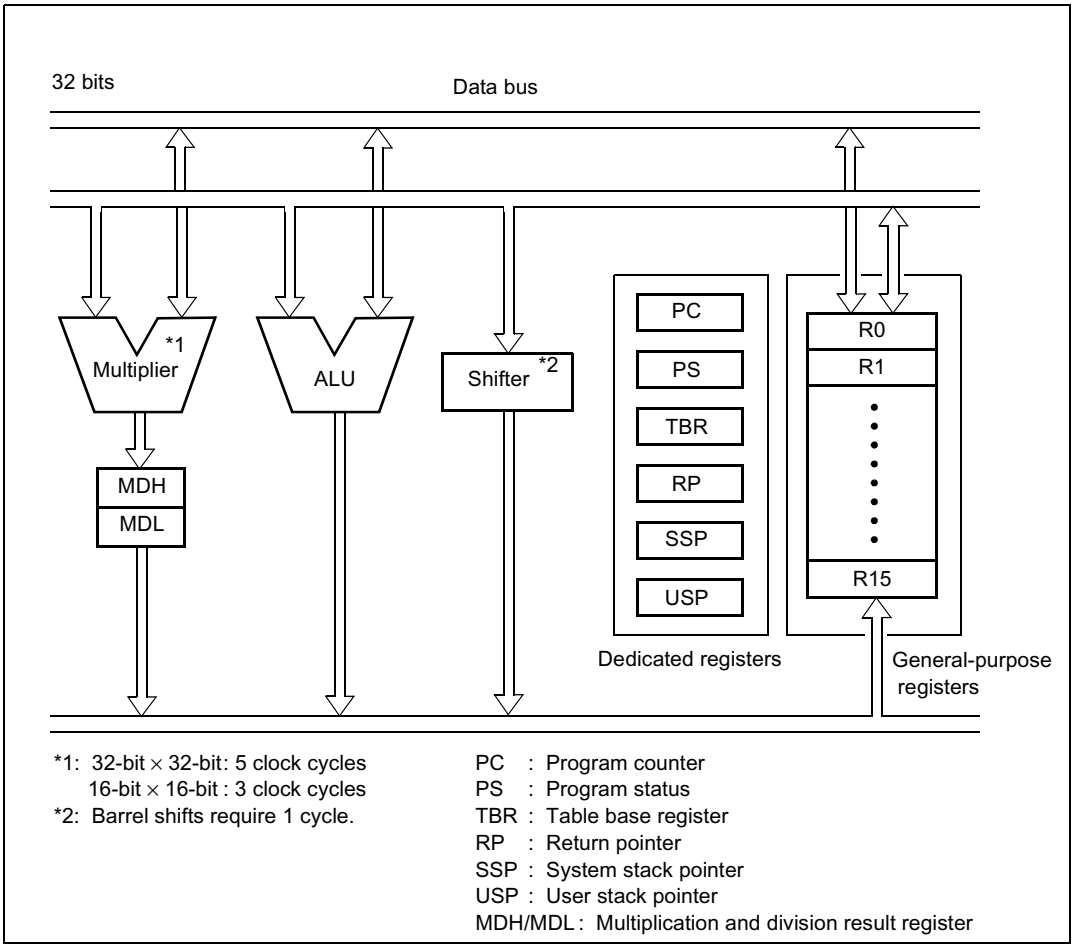
Parameter	Specification
Clock	up to 64 MHz (device dependent)
Minimum instruction execution time	1 clock
Pipeline stages	5 stages
Interlock control	Automatic pipeline interlock control by internal hardware
Internal bus	Harvard bus, instruction/data: 16 bits/32 bits
Data length	4 bits, 8 bits, 16 bits, or 32 bits
Memory space	4 GBytes (2^{32})
Number of registers	General-purpose registers R0 to R15: 16 × 32 bits Dedicated registers PC : 32 bits Program counter PS : 32 bits Flags and interrupt level TBR : 32 bits Interrupt vector table RP : 32 bits Stores the return address SSP : 32 bits System stack (shared with R15) USP : 32 bits User stack (shared with R15) MDH : 32 bits Multiplication and division data MDL : 32 bits Multiplication and division data
Number of instructions	165 instructions
Instruction length	16 bits (fixed) (excluding immediate value instructions)
Special instructions	Multiplication instruction: 32-bit × 32-bit = 64 bits, 5 cycles, signed or unsigned 16-bit × 16-bit = 32 bits, 3 cycles, signed or unsigned Division instruction: 32-bit / 32-bit = 32 bits, step operation Barrel shift instruction : Can shift by specified number of bits in 1 clock Bit manipulation instructions : Performs a logical operation on memory and transfers the result to memory. Delayed branch instructions : Automatically selected by the compiler Multi-register save and restore instructions Instructions to reserve and release dynamic variables Byte data swap instructions for semaphore control Memory-to-memory transfer instructions High-level internal peripheral access instructions
Exceptions	Reset, INT instruction, reserved instruction exception, maskable interrupts, Non-Maskable Interrupt NMI
Interrupt levels	16 levels (maskable interrupts)
Interrupt response time	6 clocks
Process	0.35 μm
Power supply	5 V / internal voltage regulator generates internal supply voltage of 3.3 V

2.2 CPU HARDWARE STRUCTURE

This section contains the FR50 CPU block diagram and gives an overview of the CPU functions.

■ CPU Block Diagram

Figure 2.2 CPU Block Diagram



■ CPU Function Description

- Internal bus structure

  The width of the internal data bus is 32 bits. 32-bit operations can be performed in one clock cycle.

  The instruction address (32 bits) and instruction data (16 bits), and data address (32 bits) and data (32 bits) buses have a Harvard bus interface structure.

- Registers

  The general-purpose registers consist of R0 to R15 (16 × 32-bit registers).

  The dedicated registers consist of the program counter (PC), program status (PS), table base register (TBR), return pointer (RP), system stack pointer (SSP), and user stack pointer (USP). All dedicated registers are 32 bits.

  Multiplication and division result registers MDH (32 bits) and MDL (32 bits) are also provided. These registers store 64-bit multiplication results, division result divisor, and the remainder and quotient of the result.

  General-purpose register R13 is used as the virtual accumulator and R14 as the frame pointer for special instructions. R15 can be selected within a program to function as the SSP or USP.

- Barrel shifter

  The barrel shifter can perform a shift of the specified number of bits on 32-bit data in one clock cycle.

  The barrel shifter is used in multiplication, division, average, and similar calculations containing power of two constants.

- ALU

  Used for operations such as 32-bit operations, logical operations, and multiplication and division.

- Multiplier

  A 32-bit × 32-bit integer multiplier circuit. The 64-bit result is stored in MDH and MDL.

- Write buffer

  The CPU contains a two-stage data store buffer.

- Delayed branch

  The compiler automatically selects delayed branch instructions.

- Interrupt control

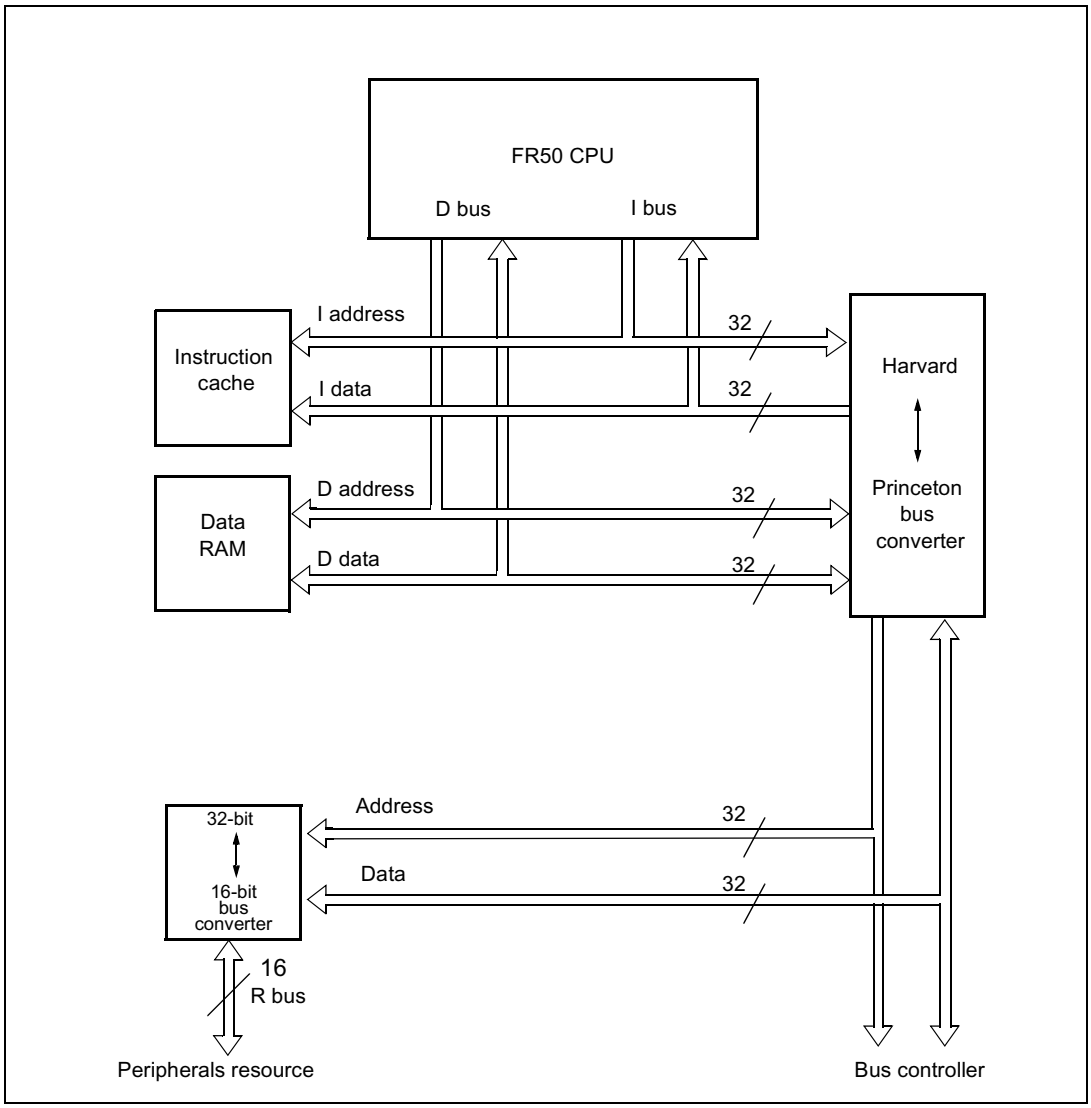
  The program can branch to the interrupt vector address in 6 clock cycles.

2.3 INTERNAL ARCHITECTURE OF THE FR50

The FR50 CPU has a Harvard architecture in which the data and instruction buses are independent.
The on-chip instruction cache is connected to the instruction bus (I bus). The data bus (D bus) connects to the Data RAM. The Harvard bus/Princeton bus converter is connected to both the I bus and D bus and acts as the interface between the CPU, the bus controller and the 32bit-16 bit bus converter.

■ **Internal Architecture of the MB91360**

Figure 2.3a Internal Architecture of the FR50



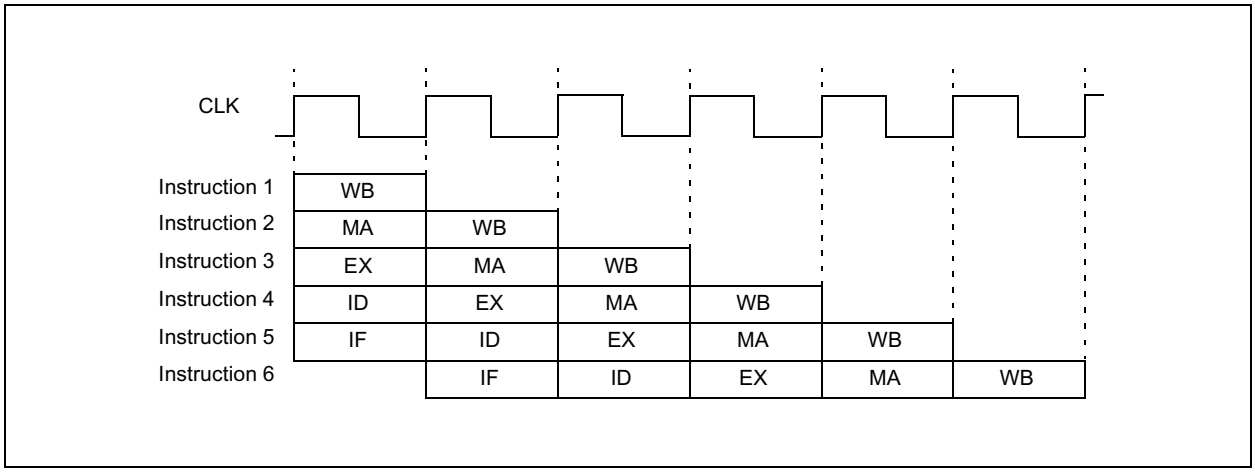
■ CPU Pipeline Operation

The CPU is a compact implementation of the 32-bit RISC FR50 architecture.

To execute one instruction per cycle, the CPU uses a five-stage instruction pipeline structure. The pipeline consists of the following stages.

- Instruction fetch (IF): Outputs the instruction address and fetches the instruction.
- Instruction decode (ID) : Decodes the fetched instruction. Also reads registers.
- Execution (EX) : Executes the operation.
- Memory access (MA) : Performs memory load or store accesses.
- Write back (WB) : Writes the operation result (or loaded memory data) to a register.

Figure 2.3b Instruction Pipeline



Instructions are not executed out of order. Therefore, if instruction A enters the pipeline ahead of instruction B, instruction A always reaches the writeback stage before instruction B.

The standard instruction execution speed is one instruction per cycle. However, load and store instructions that involve a memory wait, branch instructions without a delay slot, and multi-cycle instructions require more than one cycle to execute. The instruction execution speed also drops if delivery of instructions is slow.

■ Instruction Cache

The on-chip instruction cache enables a high-performance system to be implemented without the added cost of external high-speed memory and related control logic. Even if the external bus speed is low, instructions can be supplied to the CPU at high speed.

See chapter 3 "INSTRUCTION CACHE" on page 129 for further information on the instruction cache.

■ 32-bit ↔ 16-bit Bus Converter

Acts as an interface between the CPU core and the 16-bit R bus and enables the CPU to access data in the internal peripheral circuits.

When a 32-bit access from the CPU occurs, the bus converter converts the access into two 16-bit accesses on the R bus. Note that the access width of some internal peripheral circuits is restricted.

■ Harvard ↔ Princeton Bus Converter

Handles CPU instruction and data bus accesses and provides a smooth interface to the external bus.

The CPU has a Harvard architecture in which the instruction and data buses are separate. However, the bus controller that controls the external bus has a single-bus princeton architecture. The bus converter prioritizes CPU instruction and data bus accesses and controls access to the bus controller. This function continuously optimizes the order of external bus accesses.

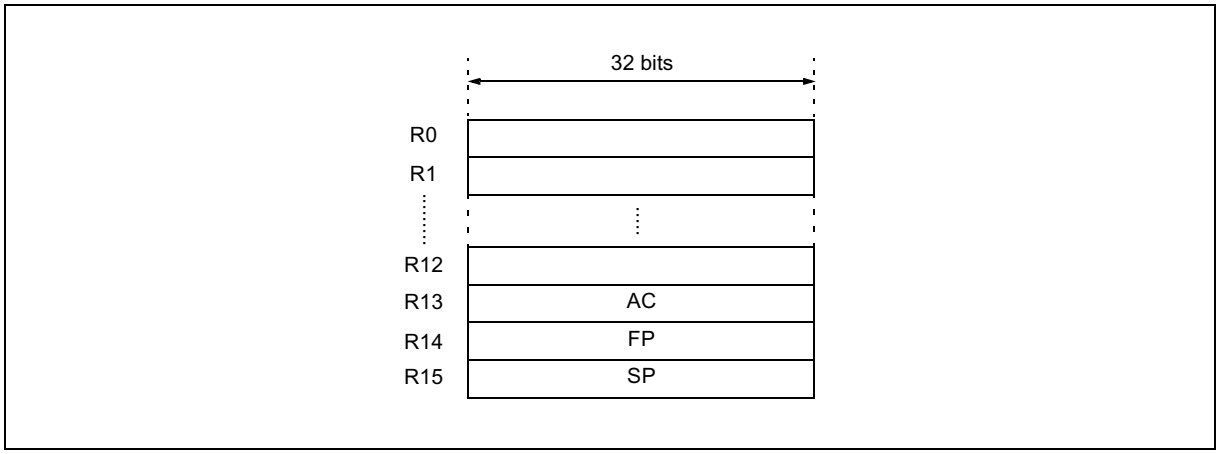
In order to eliminate times when the CPU is waiting on the bus, a two-word write buffer and one-word prefetch buffer for instruction fetches are provided.

2.4 PROGRAMMING MODEL

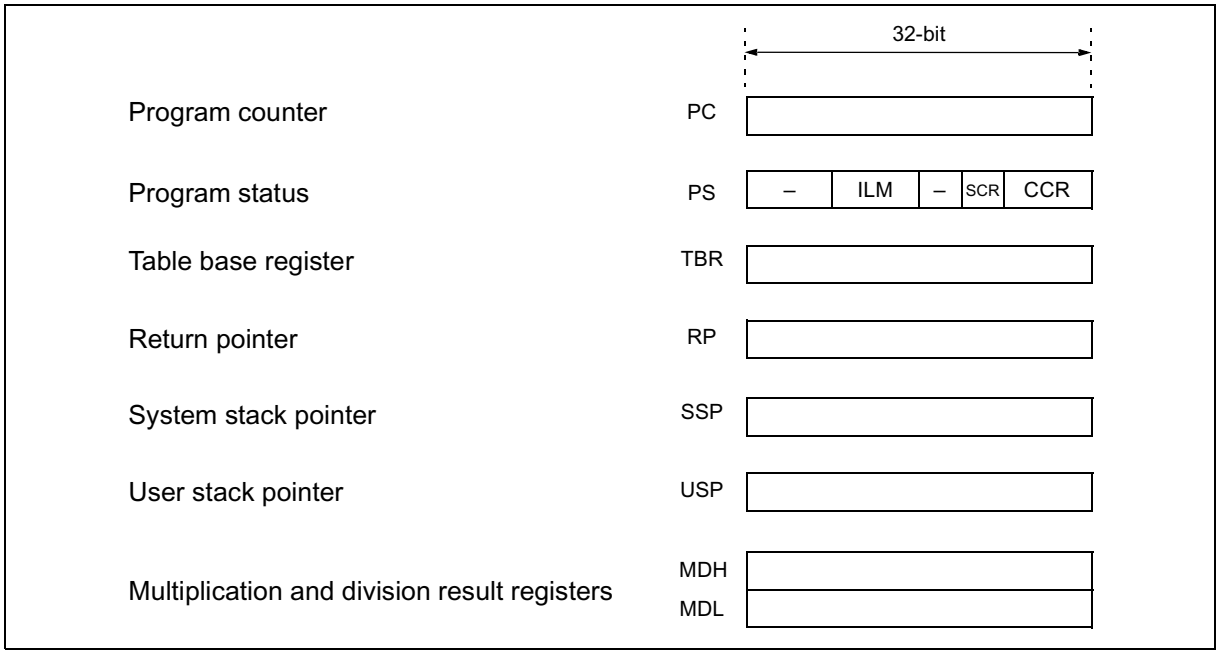
This section describes the main CPU registers used in programming.
The CPU has the following two types of registers.

- **General-purpose registers**
- **Dedicated registers**

■ General-purpose Registers



■ Dedicated Registers

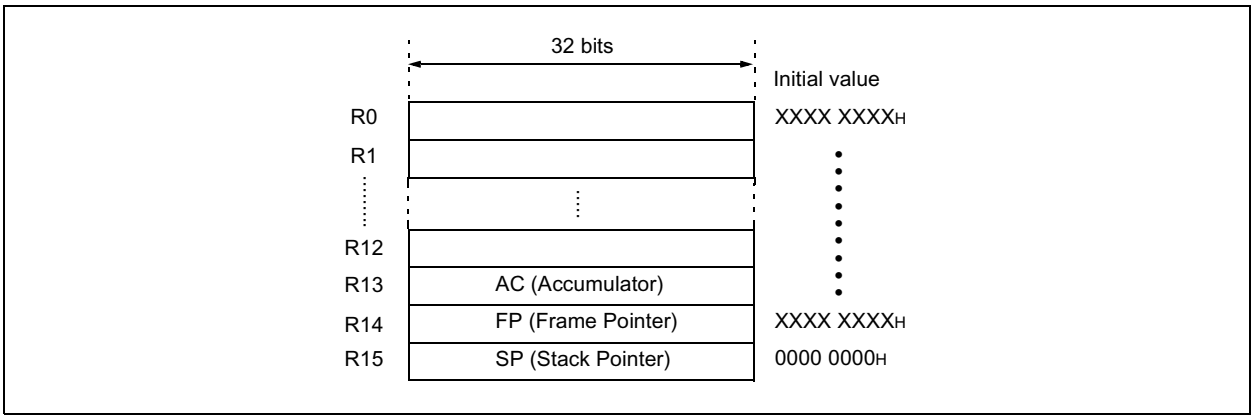


2.4.1 General-purpose Registers

The general-purpose registers are CPU registers R0 to R15. The register are used as the accumulator for operations and as pointers (a field indicating an address) for memory access. The user can specify the purpose for which the general-purpose registers are used.

■ Structure of the General-purpose Registers

Figure 2.4.1 Structure of the General-purpose Registers



Among 16 general-purpose registers, the following registers assume a special purpose. This enhances some instructions.

R13 : Virtual accumulator (AC)

R14 : Frame pointer (FP)

R15 : Stack pointer (SP)

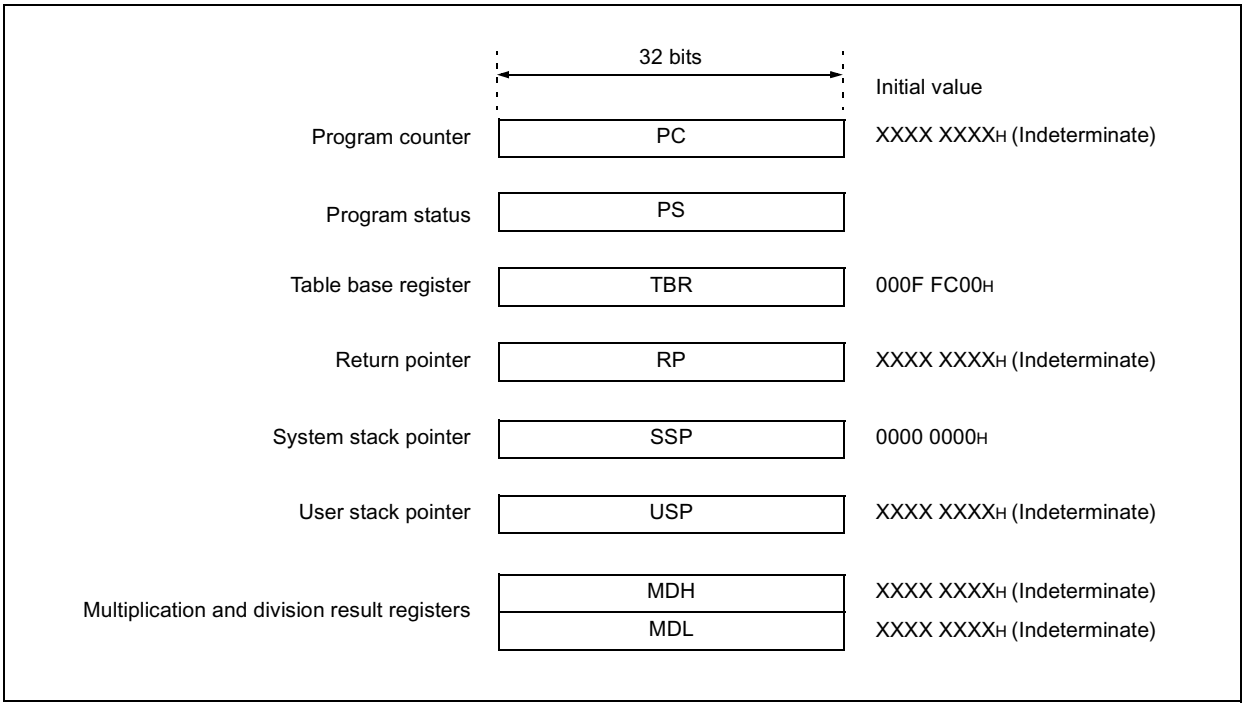
The initial value of R0 to R14 after a reset is indeterminate. The initial value of R15 is 00000000H (SSP value).

2.4.2 Dedicated Registers

Each of the dedicated registers is used for a particular purpose. The dedicated registers consist of the program counter (PC), program status (PS), table base register (TBR), return pointer (RP), system stack pointer (SSP), user stack pointer (USP), and multiplication and division result registers (MDH/MDL).

■ Structure of the Dedicated Registers

Figure 2.4.2 Structure of the Dedicated Registers



■ Descriptions of the Dedicated Registers

- PC (Program Counter)

  The program counter (PC) points to the address of the currently executing instruction.

  Bit 0 is set to "0" when updating the PC as part of instruction execution. Bit 0 can only have the value "1" when an odd-numbered address is specified as a branch destination address.

  However, even if bit 0 is "1", the bit 0 value is ignored and instructions must be located at addresses that are multiples of two.

  The initial value after a reset is indeterminate.

- PS (Program Status)

  Stores the program status and consists of the CCR, SCR, and ILM sections. See section 2.4.3 "Program Status Register (PS)" on page 88 for further information.

  All undefined bits are reserved bits and are always read as "0". Writing is ignored.

- TBR (Table Base Register)

  The table base register stores the top address of the vector table used for EIT(exception, interrupt, and trap) processing.

  Initialized to 000FFC00$_H$ by a reset.

- RP (Return Pointer)

  The return pointer stores the return address from subroutines.

  Executing the CALL instruction places the value of the PC in the RP.

  Executing the RET instruction places the contents of the RP in the PC.

  The initial value after a reset is indeterminate.

- SSP (System Stack Pointer)

  The SSP is the system stack pointer.

  Functions as R15 when the S flag is "0".

  The SSP can also be specified as the stack pointer (R15).

  The SSP can also be used as the stack pointer that specifies the stack on which to save the PS and PC when an EIT occurs.

  Initialized to 00000000$_H$ by a reset.

- USP (User Stack Pointer)

  The USP is the user stack pointer.

  Functions as the stack pointer (R15) when the S flag is "1".

  The USP can also be specified as the stack pointer (R15).

  The initial value after a reset is indeterminate.

- MDH/MDL : Multiplication and division result register

  These registers are used for multiplication and division. Both registers are 32 bits.

  The initial value after a reset is indeterminate.

  Execution of a multiplication

  For a 32-bit × 32-bit multiplication, the 64-bit result is stored in the multiplication and division result registers as shown below.

  MDH: Upper 32 bits

  MDL: Lower 32 bits

  For a 16-bit × 16-bit multiplication, the result is stored as follows.

  MDH: Indeterminate

  MDL: 32-bit result

  Execution of a division

  Stores the dividend in MDL when starting the calculation.

  Executes the DIV0S/DIV0U, DIV1, DIV2, DIV3, and DIV4S instructions to perform the division. This stores the result in MDH and MDL.

  MDH: Remainder

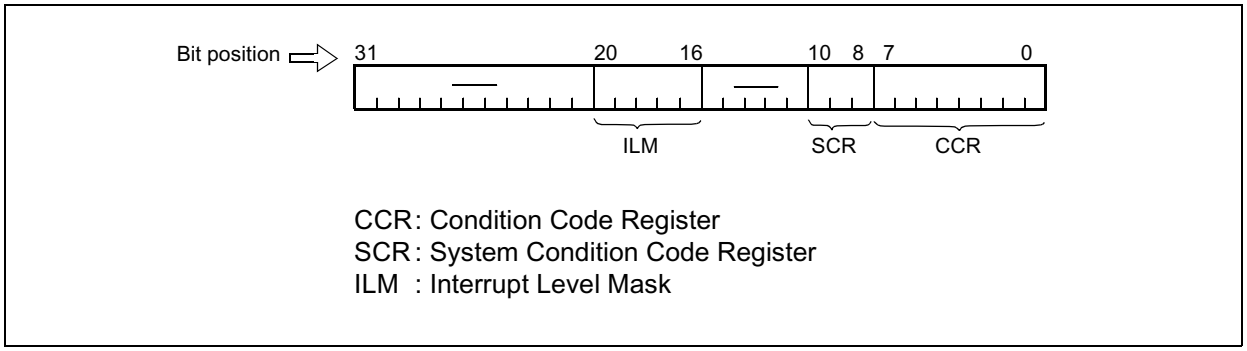
  MDL: Quotient

2.4.3 Program Status Register (PS)

The PS stores the program status and consists of the CCR, SCR, and ILM sections. All undefined bits are reserved bits and are always read as "0". Writing is ignored.

■ Structure of the Program Status Register (PS)

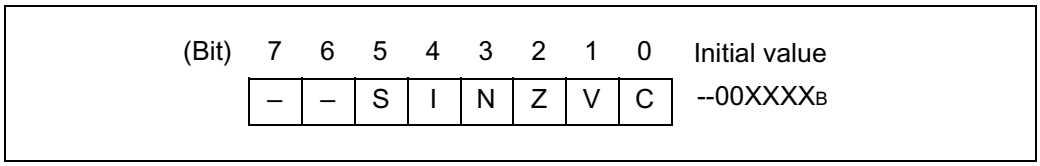
Figure 2.4.3a Structure of the Program Status Register (PS)



■ Condition Code Register (CCR)

● CCR structure

Figure 2.4.3b Structure of the Condition Code Register



● Functions of the CCR bits

[Bit 5] S: Stack flag

Specifies the stack pointer to use as R15.

Value	Function
0	SSP is used as R15. Automatically changes to "0" when an EIT occurs. However, the value saved on the stack is the value before the bit is cleared.
1	USP is used as R15.

Cleared to "0" by a reset.

[Bit 4] I: Interrupt enable flag

Controls the enabling and disabling of user interrupt requests.

Value	Function
0	User interrupts disabled Cleared to "0" by the INT instruction. However, the value saved on the stack is the value before the bit is cleared.
1	User interrupts enabled The ILM value controls masking of user interrupt requests.

Cleared to "0" by a reset.

[Bit 3] N: Negative flag

Specifies the sign of the operation result interpreted as a two's complement integer.

Value	Function
0	Indicates that the operation result was positive.
1	Indicates that the operation result was negative.

The initial value after a reset is indeterminate.

[Bit 2] Z: Zero flag

Specifies whether the operation result was zero or not.

Value	Function
0	Indicates that the operation result was non-zero.
1	Indicates that the operation result was zero.

The initial value after a reset is indeterminate.

[Bit 1] V: Overflow flag

Specifies whether or not an overflow occurred in the operation result. (When the operand used in the operation is interpreted as a two's complement integer.)

Value	Function
0	Indicates that no overflow occurred in the operation result.
1	Indicates that an overflow occurred in the operation result.

The initial value after a reset is indeterminate.

[Bit 0] C: Carry flag

Specifies whether a carry or borrow from the MSB occurred in the operation.

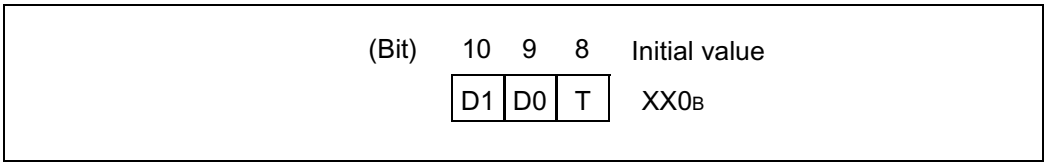
Value	Function
0	Indicates that neither a carry nor borrow occurred.
1	Indicates that a carry or borrow occurred.

The initial value after a reset is indeterminate.

■ System Condition Code Register (SCR)

● SCR structure

Figure 2.4.3c Structure of the System Condition Code Register



● Functions of the SCR bits

[Bit 10, 9] D1, D0: Step division flags

Stores intermediate data during execution of step divisions.

Do not modify these bits during a division operation.

If performing other operations during a step division, restarting of the division is assured if the value of the PS register is saved and restored.

The initial value after a reset is indeterminate.

Executing the DIV0S instruction references the dividend and divisor and sets the bits.

Executing the DIV0U instruction forcibly clears the bits.

[Bit 8] T: Step trace trap flag

This flag specifies whether the step trace trap is enabled or disabled.

Value	Function
0	Disable the step trace trap.
1	Enable the step trace trap. This disables the user NMI and all user interrupts.

Initialized to "0" by a reset.

The step trace trap function is used by the emulator. The function cannot be used by the user program when using the emulator.

■ Interrupt Level Mask Register (ILM)

- ILM structure

Figure 2.4.3d Structure of the Interrupt Level Mask Register

(Bit)	20	19	18	17	16	Initial value
	ILM4	ILM3	ILM2	ILM1	ILM0	01111_B

- Functions of the ILM bits

The ILM register stores the interrupt level mask value. The value stored in the ILM is used for the level mask.

Only interrupt requests with an interrupt level that has a higher priority than the level in the ILM are accepted.

Level 0 (00000_B) has the highest priority and level 31 (11111_B) has the lowest priority.

Restrictions apply to the values that can be set by the program. When the original value is between 16 and 31, only new values between 16 and 31 can be set. Executing an instruction to set a value between 0 and 15 transfers the value "specified value + 16".

If the original value is between 0 and 15, any value between 0 and 31 can be set.

Initialized to 15 (01111_B) by a reset.

2.5 DATA STRUCTURE

Data is allocated in the FR50 series as follows.

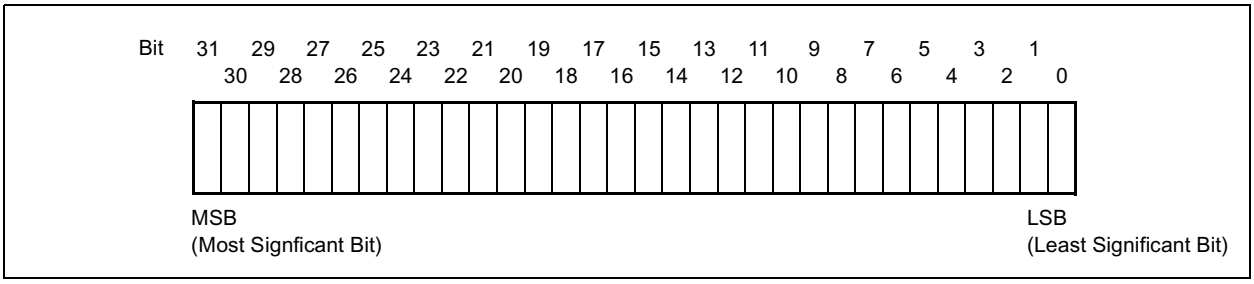
- **Bit ordering: Little endian**
- **Byte ordering: Big endian**

■ Bit Ordering

The FR50 uses little endian bit ordering.

In little endian bit ordering, bit 0 is located in the lowest position, and bits 1, 2, and so on are placed in successively higher positions.

Figure 2.5a Little Endian Bit Ordering

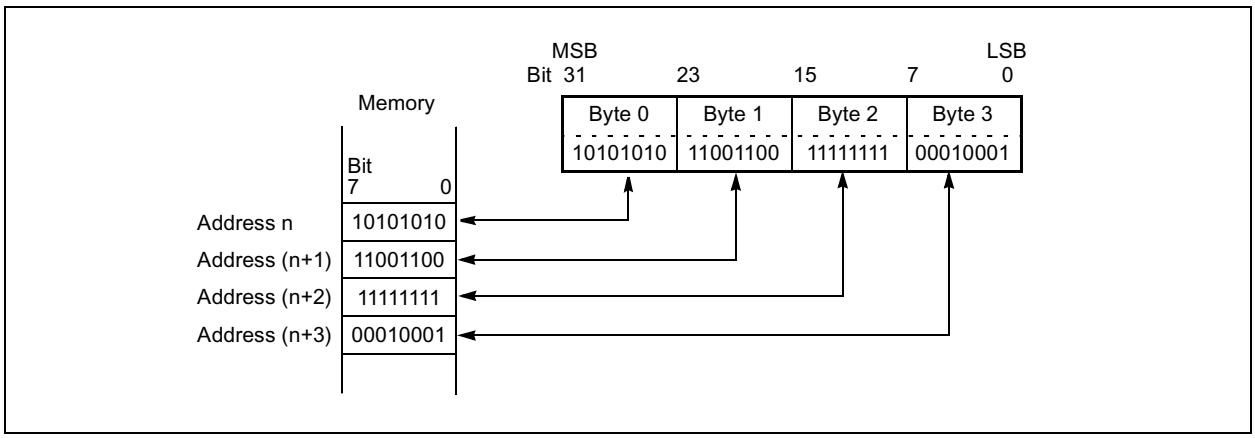


■ Byte Ordering

The FR50 uses big endian byte ordering.

For 32-bit data, big endian byte ordering places byte 0 at the highest position and byte 3 at the lowest position. Byte 0 is located in the lowest address, byte 1 is in the byte 0 address + 1, and so on. See Figure 2.5b.

Figure 2.5b Big Endian Byte Ordering



2.5.3 Word Alignment

Instructions and data are accessed by byte. Therefore, the address at which the instructions or data are located depends on the instruction or data size. This section describes address allocation for program access and data access.

■ Program Access

As the FR50 series uses 16-bit instructions (fixed-length), program must be located at addresses that are a multiple of two.

Bit 0 of the program counter (PC) is set to "0" when updating the PC as part of instruction execution. Bit 0 can only have the value "1" when an odd-numbered address is specified as a branch destination address. However, even if bit 0 is "1", the bit 0 value is ignored and instructions must be located at addresses that are multiples of two. There is no odd-numbered address exception.

■ Data Access

The address for data access in the FR50 series is forcibly aligned as follows depending on the data access size.

Word access: Address is a multiple of 4. (The lowest two bits are forcibly set to "00".)

Halfword access : Address is a multiple of 2. (The lowest bit is forcibly set to "0".)

Byte access: (Any address can be used.)

In word or halfword data access, it is the effective address that may have bits forcibly set to "0". For example, in the @(R13, Ri) addressing mode, the register values prior to the addition are used without change (even if the LSB is "1") and the LSB of the addition result is masked. The register values used in the calculation are not masked.

[Example] LD @(R13, R2), R0

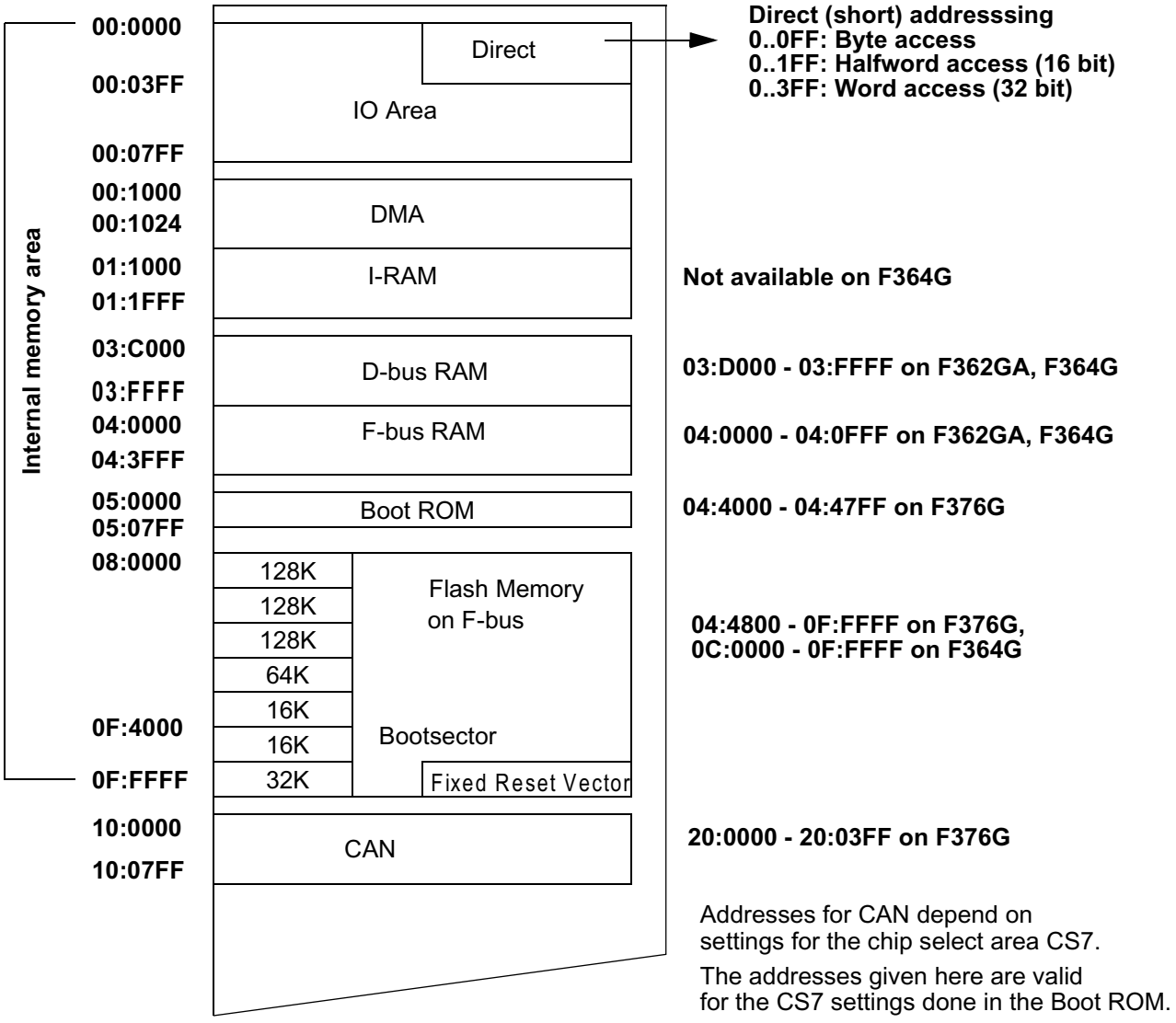
	R13	00002222H
+	R2	00000003H
Result of addition		00002225H
		⇩ Lower two bits forcibly masked
Address pins		00002224H

2.6 MEMORY MAP

This section shows the MB91360 memory map and the FR50 series Common Memory Map.

■ MB91360 Memory Map

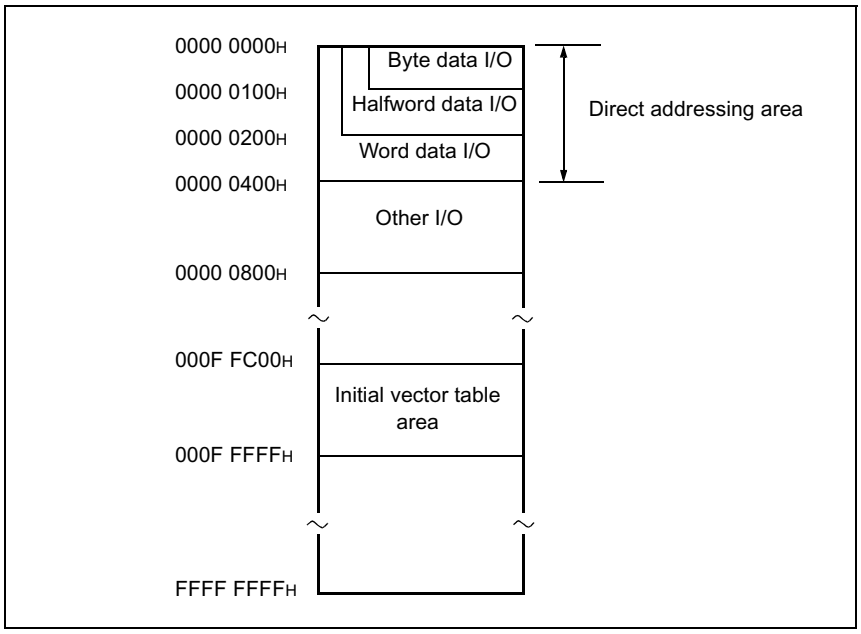
Figure 2.6a MB91360 Memory Map



■ FR50 Series Common Memory Map

The address space is a 32-bit linear logical address space giving a total area of 4 Gbytes (2^{32} locations).

Figure 2.6b FR50 series Common Memory Map



- Direct addressing area

  The following areas of the address space are I/O areas that can be accessed by direct addressing. Address operands can be specified directly in instructions.

  The size of directly addressable address areas depends the length of the data being accessed.

  - Byte data (8 bits) : 0 to 0FFH
  - Halfword data (16 bits) : 0 to 1FFH
  - Word data (32 bits) : 0 to 3FFH

- Initial vector table area

  The area between 000FFC00H and 000FFFFFH is the initial EIT vector table area.

  The vector table used in EIT processing can be set to a user-specified address by changing the table base register (TBR). However, after initialization by a reset, the vector table is located in this area.

  After execution of the code in the internal boot ROM TBR is set to 000FFC00H.

2.7 INSTRUCTIONS

The FR50 series instruction set adds easily controlled CISC-type instructions to the previous RISC processor instruction set. The instruction set is suitable for embedded applications and supports use of high-level languages.
This section describes the instruction set features, lists the basic instructions, and describes the branch instructions (delayed branch instructions) which have a large influence on the flow of pipeline operations.
See Appendix D "INSTRUCTIONS" on page 790 and the *FR50 series Instruction Manual* for further information about the instruction set.

■ Features

- Fixed, 16-bit instruction length (excluding immediate transfer instructions)
  Improved instruction code efficiency compared with high-level RISC processors
- Number of instructions: 165
- High-speed multiplication instruction and step division instruction
  32-bit × 32-bit: 5 clock cycles
  16-bit × 16-bit: 3 clock cycles
  32-bit / 32-bit: 33 clock cycles
  36 clock cycles (signed)
- Barrel shift instruction
- I/O control instructions optimized for embedded control
  Bit manipulation : Immediate or memory-to-memory operations
  Logical operations : Operations between general-purpose registers and memory
  Memory-to-memory transfer : Memory-to-memory data transfer
- Instructions optimized for compiler or realtime operating system
  Function call and return processing instructions
  Instructions to save and restore multiple consecutive general-purpose registers
  Delayed branch instructions
- Peripheral accelerator instructions

2.7.1 Overview of Instructions and Basic Instructions

In addition to standard RISC instructions, the FR50 provides logical operations optimized for embedded applications, bit manipulation instructions, and direct addressing instructions. As instructions are 16 bits (some instructions are 32 bits or 48 bits), the FR50 provides superior memory utilization.

■ Overview of Instructions

The instruction set can be divided into the following functional groups.

- Arithmetic operations
- Load and store
- Branch
- Logical operations and bit manipulation
- Direct addressing
- Others

● Arithmetic operations

The standard arithmetic operation instructions (add, subtract, compare) and shift instructions (logical shift, arithmetic operation shift) are provided. Addition and subtraction include operations with carry for use in multi-word operations and operations that do not change the flags (which are suitable for address calculations).

32-bit × 32-bit multiplication, 16-bit × 16-bit multiplication, and 32-bit / 32-bit step division instructions are provided. Immediate value transfer instructions (which set immediate values to registers) and register-to-register transfer instructions are also provided.

The arithmetic operation instructions all perform operations using the general-purpose registers and multiplication and division registers in the CPU.

● Load and store

Load and store instructions read data from or write data to external memory. The instructions are also used to read from and write to the internal peripheral circuits (I/O).

Load and store instructions are provided for byte (8 bit), halfword (16 bit), and word length (32 bit) access. In addition to standard register indirect memory addressing, some instructions support register indirect with displacement memory addressing and register indirect with register increment or decrement memory addressing.

- Branch

  The branch, call, interrupt, and return instructions.

  Branch instructions are divided into those that have a delay slot and those that do not. This allows optimization to suit the usage. See “Branch Instructions” on page 101 for further information on branch instructions.

- Logical operations and bit manipulation

  Logical operation instructions can perform AND, OR, or EOR logical operations between general-purpose registers or between general-purpose registers and memory (or I/O). The bit manipulation instructions can manipulate the contents of memory (or I/O) directly. These instructions use standard register indirect memory addressing.

- Direct addressing

  Direct addressing instructions are instructions used for addresses between I/O and general-purpose registers, and between I/O and memory. Specifying an I/O address directly in an instruction rather than by register indirect addressing provides high-speed, high-efficiency access. Some instructions support register indirect with register increment or decrement memory addressing.

- Other

  These include instructions to set the flags in the PS register, perform stack operations, and sign-extend or zero-extend. Function entry and exit processing instructions for high-level languages and multi-register loading and storing instructions are also provided.

■ Basic Instructions

Table 2.7.1 Basic Instructions

Addition and subtraction instructions

```
ADD   Rj,   Ri
*ADD  #s5,  Ri
ADDC  Rj,   Ri
ADDN  Rj,   Ri
*ADDN #s5,  Ri
SUB   Rj,   Ri
SUBC  Rj,   Ri
SUBN  Rj,   Ri
ADD   #u4,  Ri
ADD2  #u4,  Ri
ADDN  #u4,  Ri
ADDN2 #u4   Ri
```

Compare operation instructions

```
CMP   Rj,   Ri
*CMP  #s5,  Ri
CMP   #u4,  Ri
CMP2  #u4,  Ri
```

Logical operation instructions

```
AND   Rj,   Ri
AND   Rj,   @Ri
ANDH  Rj,   @Ri
ANDB  Rj,   @Ri
OR    Rj,   Ri
OR    Rj,   @Ri
ORH   Rj,   @Ri
ORB   Rj,   @Ri
EOR   Rj,   Ri
EOR   Rj,   @Ri
EORH  Rj,   @Ri
EPRB  Rj,   @Ri
```

Bit manipulation instructions

```
*BAND  #u8,  @Ri
 BANDL #u4,  @Ri
 BANDH #u4,  @Ri
*BOR   #u8,  @Ri
 BORL  #u4,  @Ri
 BORH  #u4,  @Ri
*BEOR  #u8,  @Ri
 BEORL #u4,  @Ri
 BEORH #u4,  @Ri
 BTSTL #u4,  @Ri
 BTSTH #u4,  @Ri
```

Multiplication and division instructions

```
MUL   Rj,   Ri
MULU  Rj,   Ri
MULH  Rj,   Ri
MULUH Rj,   Ri
*DIV  Ri
*DIVU Ri
 DIV0S Ri
 DIV0U Ri
 DIV1  Ri
 DIV2  Ri
 DIV3
 DIV4S
```

Shift instructions

```
 LSL   Rj,   Ri
*LSL   #u5,  Ri
 LSR   Rj,   Ri
*LSR   #u5,  Ri
 ASR   Rj,   Ri
*ASR   #u5,  Ri
 LSL   #u4,  Ri
 LSL2  #u4,  Ri
 LSR   #u4,  Ri
 LSR2  #u4,  Ri
 ASR   #u4,  Ri
 ASR2  #u4,  Ri
```

Register-to-register transfer instructions

```
MOV   Rj,   Ri
MOV   Rs,   Ri
MOV   Ri,   Rs
MOV   PS,   Ri
MOV   Ri,   PS
```

Memory load instructions

```
LD    @Rj, Ri
LD    @(R13,Rj), Ri
LD    @(R14,disp10), Ri
LD    @(R15,udisp6), Ri
LD    @R15+, Ri
LD    @R15+, Rs
LD    @R15+, PS
LDUH  @Rj, Ri
LDUH  @(R13,Rj), Ri
LDUH  @(R14,disp9), Ri
LDUB  @Rj, Ri
LDUB  @(R13,Rj), Ri
LDUB  @(R14,disp8), Ri
```

Memory store instructions

```
ST    Ri,   @Rj
ST    Ri,   @(R13,Rj)
ST    Ri,   @(R14,disp10)
ST    Ri,   @(R15,udisp6)
ST    Ri,   @–R15
ST    Rs,   @–R15
ST    PS,   @–R15
STH   Ri,   @Rj
STH   Ri,   @(R13,Rj)
STH   Ri,   @(R14,disp9)
STB   Ri,   @Rj
STB   Ri,   @(R13,Rj)
STB   Ri,   @(R14,disp8)
```

Immediate value transfer instructions

```
*LDI    # {i8 | i20 | i32},  Ri
 LDI:32 #i32,  Ri
 LDI:20 #i20,  Ri
 LDI:8  #i8,   Ri
```

Note: An asterisk(*) on the left of the mnemonic indicates that the instruction is an extended instruction for which instructions are expanded or added by the assembler.

Table 2.7.1 Basic Instructions *(Continued)*

Standard branch instructions *1	Other instructions	Direct addressing instructions
JMP @Ri	NOP	DMOV @dir10, R13
CALL label12	ANDCCR #u8	DMOV R13, @dir10
CALL @Ri	ORCCR #u8	DMOV @dir10, @R13+
RET	STILM #u8	DMOV @R13+, @dir10
INT #u8	ADDSP #u10	DMOV @dir10, @-R15
INTE	EXTSB Ri	DMOV @-R15, @dir10
RETI	EXTUB Ri	DMOVH @dir9, R13
BRA label9	EXTSH Ri	DMOVH R13, @dir9
BNO label9	EXTUH Ri	DMOVH @dir9, @R13+
BEQ label9	*LDM (reglist)	DMOVH @R13+, @dir9
BNE label9	LDM0 (reglist)	DMOVB @dir8, R13
BC label9	LDM1 (reglist)	DMOVB R13, @dir8
BNC label9	*STM (reglist)	DMOVB @dir8, @R13+
BN label9	STM0 (reglist)	DMOVB @R13+, @dir8
BP label9	STM1 (reglist)	
BV label9	ENTER #u10	
BNV label9	LEAVE	
BLT label9	XCHB @Rj, Ri	
BGE label9		
BLE label9		
BGT label9		
BLS label9		
BHI label9		
Delayed branch instructions *2		**Peripheral instructions**
JMP:D @Ri		LDRES @Ri+, #u4
CALL:D label12		STRES #u4, @Ri+
CALL:D @Ri		
RET:D		
BRA:D label9		
BNO:D label9		
BEQ:D label9		
BNE:D label9		
BC:D label9		
BNC:D label9		
BN:D label9		
BP:D label9		
BV:D label9		
BNV:D label9		
BLT:D label9		
BGE:D label9		
BLE:D label9		
BGT:D label9		
BLS:D label9		
BHI:D label9		

*1: Without delay slot *2: With delay slot

Note: An asterisk(*) on the left of the mnemonic indicates that the instruction is an extended instruction for which instructions are expanded or added by the assembler.

2.7.2 Branch Instructions

In the FR50 series, you can specify whether branch instructions operate with or without a delay slot. Operation with a delay slot means that the instruction placed after the branch instruction is executed.

■ Branch Instructions with a Delay Slot

The instructions with the notation listed below perform the branch operation with a delay slot.

JMP:D	@Ri	CALL:D	label12	CALL:D	@Ri	RET:D	
BRA:D	label9	BNO:D	label9	BEQ:D	label9	BNE:D	label9
BC:D	label9	BNC:D	label9	BN:D	label9	BP:D	label9
BV:D	label9	BNV:D	label9	BLT:D	label9	BGE:D	label9
BLE:D	label9	BGT:D	label9	BLS:D	label9	BHI:D	label9

- Operation

  Branching with a delay slot means that the branch operation occurs after executing the instruction placed immediately after the branch instruction (in what is called the delay slot).

  As the instruction in the delay slot is executed before the branch operation, the apparent execution speed is one cycle. However, if no useful instruction can be placed in the delay slot, a NOP instruction must be placed instead.

[Example]

```
;       Instruction sequence
        ADD     R1, R2  ;
        BRA:D   LABEL   ; Branch instruction
        M0V     R2, R3  ; Delay slot: Executed before branching
         :
LABEL   ST      R3, @R4 ; Branch destination
```

For conditional branch instructions, the instruction in the delay slot is executed whether or not the branch condition is satisfied.

Although delayed branch instructions appear to reverse the execution order of some instructions, this only applies to the updating of the program counter (PC). Other operations (such as updating or referencing registers) are executed in the order they appear in the program.

The following describes some specific examples.

1) The value of Ri referenced by the JMP:D @Ri or CALL:D @Ri instruction is not changed by any update of Ri by the instruction in the delay slot.

[Example]

```
LDI:32 #Label, R0
JMP:D  @R0          ;Branch to Label.
LDI:8  #0,     R0   ;Does not change the branch destination address.
:
```

2) The value of RP referenced by the RET:D instruction is not changed by any update of RP by the instruction in the delay slot.

[Example]

```
RET:D               ;Branch to the address previously set in RP.
MOV    R8,     RP   ;Does not affect the return operation.
:
```

3) The flags referenced by the Bcc:D rel instruction are not affected by the instruction in the delay slot.

[Example]

```
ADD    #1,     R0   ;Flag change
BC:D   Overflow     ;Branch depending on the result of the previous
                     instruction.
ANDCCR #0           ;This flag update does not reference the above
:                    branch instruction.
```

4) Referencing the RP by the instruction in the CALL:D instruction's delay slot reads the RP value updated by the CALL:D instruction.

[Example]

```
CALL:D Label        ;Update RP and branch.
MOV    RP,     R0   ;Transfers the RP value resulting from the
:                    execution of the above CALL:D.
```

- Restrictions

(1) Instructions that can be placed in the delay slot

Only instructions that meet the following criteria can be executed in the delay slot.

- 1-cycle instruction
- Not a branch instruction
- Instruction not affected by the order in which it is executed

"One-cycle instructions" are instructions with "1", "a", "b", "c", or "d" listed in the number of cycles column of the instruction list.

(2) Step trace trap

Step trace traps do not occur between execution of branch instructions with a delay slot and the delay slot.

(3) Interrupts/NMI

Interrupts and NMI are not accepted between execution of branch instructions with a delay slot and the delay slot.

(4) Undefined instruction exception

If the delay slot contains an undefined instruction, the undefined instruction exception is not generated. In this case, the undefined instruction is executed as an NOP instruction.

■ Branch Instructions without a delay slot

The instructions with the notation listed below perform the branch operation without a delay slot.

JMP	@Ri	CALL	label12	CALL	@Ri	RET	
BRA	label9	BNO	label9	BEQ	label9	BNE	label9
BC	label9	BNC	label9	BN	label9	BP	label9
BV	label9	BNV	label9	BLT	label9	BGE	label9
BLE	label9	BGT	label9	BLS	label9	BHI	label9

● Description of operation

Branching without a delay slot means that instructions are always executed in the order they appear in the program. The instruction following the branch instruction is never executed before branching.

[Example]

```
;          Instruction sequence
           ADD     R1, R2   ;
           BRA     LABEL    ; Branch instruction (without delay slot)
           MOV     R2, R3   ; Not executed
            :
LABEL      ST      R3, @R4  ; Branch destination
```

The number of cycles required to execute a branch instruction without a delay slot is two cycles if the branch occurs and one cycle if the branch does not occur.

As no instruction can be placed in the delay slot for a branch instruction without a delay slot, instruction code efficiency is increased compared with a branch instruction with a delay slot containing a NOP instruction.

Use a delay slot when there is a useful instruction to place in the delay slot and do not use a delay slot otherwise. This satisfies both execution speed and code efficiency.

2.8 EIT (EXCEPTIONS, INTERRUPT, TRAP)

The term EIT is a general name for exceptions, interrupts and traps. EITs interrupt execution of the current program when an event occurs and pass control to another program.
Exceptions are generated based on the execution context. Execution restarts from the instruction that caused the exception.
Interrupts are generated independently of the execution context and are triggered by hardware events.
Traps are generated based on the execution context. These include traps generated by an operation within the program like a system call. Execution restarts from the instruction following the instruction that caused the trap.
The context refers to the information required to completely define the state of the CPU when executing an instruction.

■ Features

- Multiple interrupt support
- Level mask function for interrupts (15 levels are available to the user.)
- Trap instruction (INT)
- EITs for activating the emulator (hardware/software)

■ EIT Triggers

The following items can generate an EIT.

- Reset
- User interrupt (internal peripherals, external interrupts)
- NMI (Non-Maskable Interrupt)
- Delayed interrupt
- Undefined instruction exception
- Trap instruction (INT)
- Trap instruction (INTE)
- Step trace trap

■ Returning from an EIT

- Use the RETI standard branch instruction to return from an EIT.

■ Interrupt Levels

Interrupt levels have the range 0 to 31 and are managed as a 5-bit value.
Table 2.8a, “Interrupt Levels,” on page 106 lists the assignment of each level.

Table 2.8a Interrupt Levels

Level		EIT Type	Remarks
Binary	Decimal		
00000 ⋮ 00011	0 ⋮ 3	(System reserved) ⋮ (System reserved)	If the original value of the ILM is between 16 and 31, values in this range cannot be set to the ILM by the program.
00100	4	INTE instruction Step trace trap	
00101 ⋮ 01110	5 ⋮ 14	(System reserved) ⋮ (System reserved)	
01111	15	NMI (user)	
10000	16	Interrupt	When set to the ILM, user interrupts are disabled.
10001 ⋮ 11110	17 ⋮ 30	Interrupt ⋮ Interrupt	
11111	31	—	When set to the ICR, the interrupt is disabled.

Levels 16 to 31 are available for the user.

The undefined instruction exception and INT instruction are not affected by the interrupt level. Similarly, they do not change the ILM.

■ I Flag

This flag enables or disables interrupts. The flag is located in bit 4 of the condition code register (CCR) in the PS.

Value	Function	
0	Interrupts disabled	Cleared to "0" by the INT instruction. However, the value saved on the stack is the value before the bit is cleared.
1	Interrupts enabled	Masking of interrupt requests is controlled by the value in ILM.

■ ILM (Interrupt Level Mask Register)

The ILM register is located in the PS and stores the interrupt level mask value.

Only interrupt requests to the CPU with an interrupt level that has a higher priority than the level in the ILM are accepted.

Level 0 (00000_B) has the highest priority and level 31 (11111_B) has the lowest priority.

Restrictions apply to the values that can be set by the program. When the original value is between 16 and 31, only new values between 16 and 31 can be set. Executing an instruction to set a value between 0 and 15 transfers the value "specified value + 16".

If the original value is between 0 and 15, any value between 0 and 31 can be set.

The SETILM instruction is used to set a value in the ILM register.

■ Level Mask for Interrupts/NMI

When an NMI or interrupt request occurs, the interrupt's priority level (Table 2.8a) is compared with the level mask value in the ILM.

If the following condition is satisfied, the interrupt is masked and the request not accepted:

Interrupt level of interrupt request ≥ Level mask value

■ ICR (Interrupt Control Register)

These are registers located in the interrupt controller and set the level for each interrupt request. ICR registers are provided for each interrupt input. The ICR registers are mapped in the I/O memory space and are accessed by the CPU via the bus.

● ICR structure

Figure 2.8a Structure of the Interrupt Control Registers

Bit	7	6	5	4	3	2	1	0	Initial value
	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	---11111
				R	R/W	R/W	R/W	R/W	⇦Access

R: Read, R/W: Read/write

● Functions of the ICR bits

Table 2.8b Functions of the Interrupt Control Register Bit

Bit		Function
No.	Name	
4	ICR4	Always "1"
3	ICR3	Lower 4 bits of the interrupt level for the interrupt Readable and writable Including bit 4, an ICR can have values in the range 16 to 31.
2	ICR2	
1	ICR1	
0	ICR0	

- ICR mapping

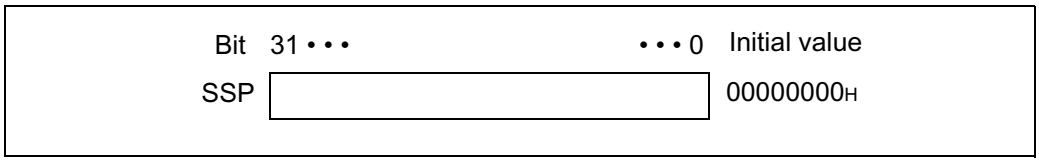
Table 2.8c Interrupt Control Register and Interrupt Vector for Each Interrupt

Interrupt	Interrupt control register		Corresponding interrupt vector		
	No.	Address	No.		Address
			Hexadecimal	Decimal	
IRQ00	ICR00	00000400H	10H	16	TBR + 3BCH
IRQ01	ICR01	00000401H	11H	17	TBR + 3B8H
IRQ02	ICR02	00000402H	12H	18	TBR + 3B4H
⋮	⋮	⋮	⋮	⋮	⋮
IRQ47	ICR47	0000042FH	3FH	63	TBR + 300H

Note: See chapter 9 "INTERRUPT CONTROLLER" on page 277ff .

■ SSP (System Stack Pointer)

Figure 2.8b System Stack Pointer Register



The SSP is used as the pointer to the stack used to save and restore data on receiving or returning from an EIT.

The value of the SSP is decremented by 8 by EIT processing and incremented by 8 on returning from the EIT by the RETI instruction.

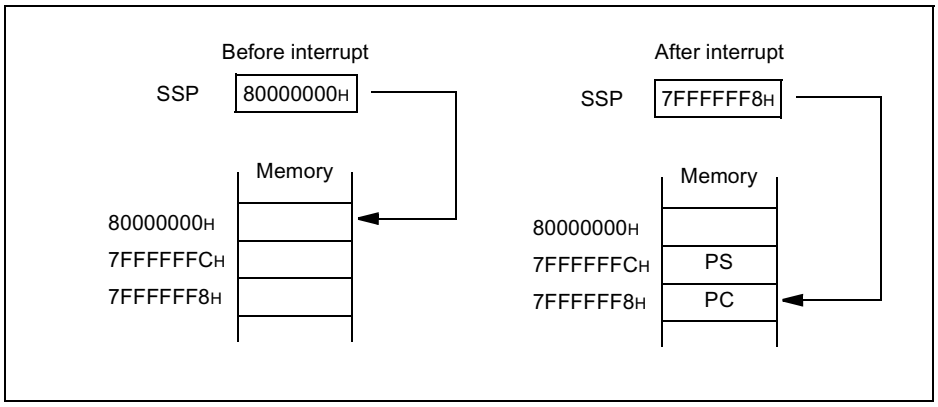
Initialized to 00000000H by a reset.

If during the execution of the internal boot ROM code, a valid boot condition is detected. SSP is set to 0003D3F8H.

■ Interrupt Stack

The area pointed to by the SSP and used to save and restore the PC and PS values. After an interrupt occurs, the PC is placed at the address pointed to by SSP and the PS at the address "SSP+4".

Figure 2.8c Example of Interrupt Stack Operation



■ TBR (Table Base Register)

This register contains the top address of the EIT vector table.

The vector address for each EIT is calculated by adding the offset value for that EIT to the TBR.

Initialized to 000FFC00H by a reset.

■ EIT Vector Table

The 1-Kbyte area starting from the address pointed to by TBR is the EIT vector area.

Each vector consists of four bytes. The following formula shows the relationship between the vector number and vector address.

vctadr =TBR + vctofs

=TBR + (3FCH – 4 × vct)

vctadr : Vector address

vctofs : Vector offset

vct : Vector number

The lower 2 bits of the addition result are always treated as "00".

The initial vector table area after a reset is the area between 000FFC00H and 000FFFFFH.

Special functions are assigned to some vectors. See table 2.8d for further information.

Table 2.8d Vector Table

Vector offset (Hexadecimal)	Vector number		Description
	Hexadecimal	Decimal	
3FC	00	0	Reset (*)
3F8	01	1	System reserved
3F4	02	2	System reserved
3F0	03	3	System reserved
⋮	⋮	⋮	⋮
3E0	07	7	System reserved
3DC	08	8	System reserved
3D8	09	9	INTE instruction
3D4	0A	10	System reserved
3D0	0B	11	System reserved
3CC	0C	12	Step trace trap
3C8	0D	13	System reserved
3C4	0E	14	Undefined instruction exception
3C0	0F	15	NMI (user)
3BC	10	16	Maskable interrupt #0
3B8	11	17	Maskable interrupt #1 (Note)
⋮	⋮	⋮	⋮
300	3F	63	Maskable interrupt/INT instruction
2FC	40	64	System reserved (used by REALOS)
2F8	41	65	System reserved (used by REALOS)
2F4	42	66	Maskable interrupt/INT instruction
⋮	⋮	⋮	⋮

Note: See Appendix B "INTERRUPT VECTORS" on page 781 for the MB91360 vector table.

*: Even if the TBR value is modified, the reset vector is always located at the fixed address 000FFFFC$_H$.

2.8.5 Multiple EIT Processing

When more than one EIT occurs at the same time, the CPU repetitively performs the following operations: select one EIT to accept, execute the EIT sequence, then detect the next EIT.
The CPU executes the handler instructions for the last EIT to be accepted when EIT detection finds no more EITs that can be accepted.
Therefore, the sequence for executing the handlers for multiple EITs that occur at the same time is determined by the following two factors.

- **The priority of the accepting EITs**
- **How other EITs are masked when an EIT is accepted.**

■ Priority for Accepting EITs

The EIT acceptance priority is the priority order for selecting which EIT to execute. The EIT sequence consists of saving the PS and PC, updating the PC (as required), and performing masking of other EITs.

Consequently, the handler of the first EIT to be accepted is not necessarily executed first.

Table 2.8.1a lists the priority order for accepting EITs.

Table 2.8.1a Priority Order for Accepting EITs and Masking of Other EITs

Priority for accepting EITs	EIT	Masking of other EITs
1	Reset	Other EITs are cleared.
2	Undefined instruction exception	Canceled
3	INT instruction	I flag = 0
4	User interrupt	ILM = Level of accepted interrupt
5	NMI (user)	ILM = 15
6	(INTE instruction)	ILM = 4 *1
7	NMI (emulator)	ILM = 4
8	Step trace trap	ILM = 4

*1: The priority becomes 6 only when an emulator NMI generated at the same time as the INTE instructions.
(The MB91360 uses the emulator NMI for a break by data access.)

After accepting an EIT and performing masking of other EITs, the priority for executing the handlers of EITs that occur at the same time is as shown in Table 2.8.1b .

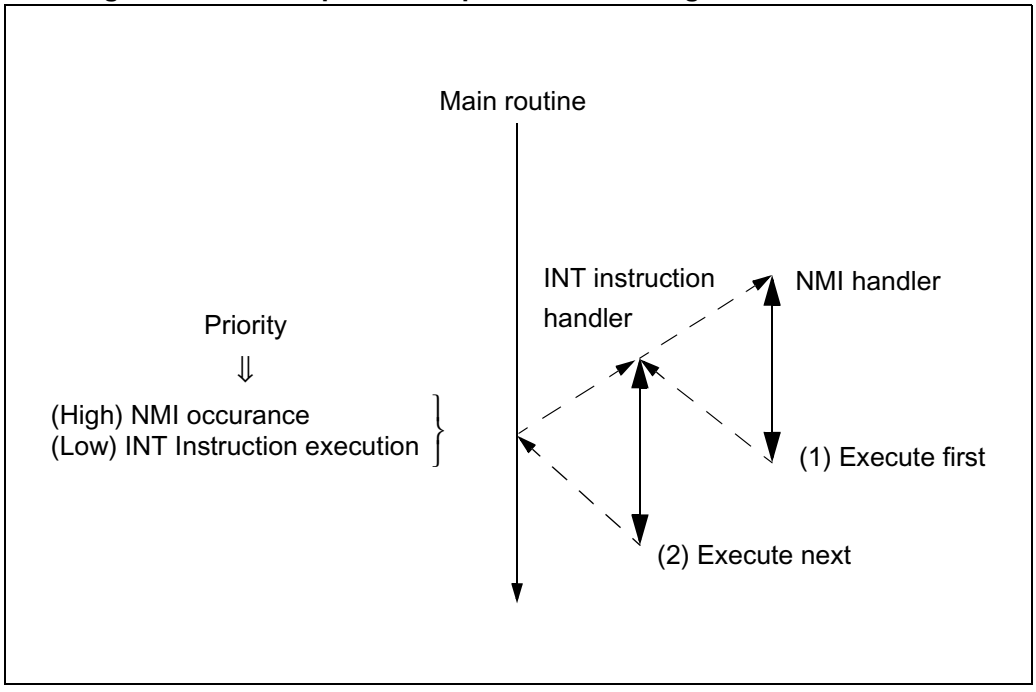
Table 2.8.1b Execution Priority for EIT Handlers

Handler execution priority	EIT
1	Reset *1
2	Undefined instruction exception
3	Step trace trap *2
4	INTE instruction *2
5	NMI (user)
6	INT instruction
7	User interrupt

*1:Other EITs are cleared.

*2:If the INTE instruction is executed by step execution, only the step trace trap EIT is generated. The INTE EIT is ignored.

Figure 2.8.1 Example of Multiple EIT Processing



2.8.2 EIT Operation

This section describes EIT operation.
In the following explanation, the transfer source "PC" contains the address of the instruction at which the EIT was detected.
Also, in the operation description, the "address of the next instruction" depends on the instruction at which the EIT was detected, as follows.

- **For LDI: 32..... PC+6**
- **For LDI: 20, COPOP, COPLD, COPST, COPSV..... PC+4**
- **For all other instructions..... PC+2**

■ Operation of User Interrupts and NMI

When a user interrupt or user non-maskable interrupt (NMI) request is generated, the following sequence determines whether or not the request can be accepted.

- Determining whether an interrupt request can be accepted

1) Compare the interrupt levels of any simultaneously occurring requests and select the interrupt with the highest priority level (smallest level value).

   For maskable interrupts, the level value used for the comparison is the value stored in the interrupt's ICR. For the NMI, a predefined constant is used.

2) If more than one interrupt request with the same level is generated, select the interrupt request with the lowest interrupt number.

3) Compare the interrupt level of the selected interrupt request with the level mask value set by the ILM.

   If the interrupt level ≥ the level mask value, the interrupt request is masked and not accepted.

   If the interrupt level < the level mask value, proceed to step 4).

4) If the selected interrupt request is a maskable interrupt and the I flag is "0", the interrupt request is masked and not accepted. If the I flag is "1", proceed to step 5).

   If the selected interrupt request is an NMI, proceed to step 5) regardless of the I flag value.

5) If the above conditions are satisfied, the interrupt request is accepted at the end of the current instruction processing.

If a user interrupt or NMI request is accepted when an EIT request is detected, the CPU operates as follows using the interrupt number corresponding to the accepted interrupt request.

( ) represents the address the register points to.

[Operation]

(1) SSP – 4 → SSP

(2) PS → (SSP)

(3) SSP – 4 → SSP

(4) address of next instruction → (SSP)

(5) interrupt level of accepted request → ILM

(6) "0" → S flag

(7) (TBR + vector offset of received interrupt request) → PC

After the interrupt sequence completes, the system checks whether any new EITs are present before executing the first instruction of the handler. If an EIT that can be accepted is present, the CPU enters the EIT processing sequence.

■ Operation of the INT Instruction

INT #u8

Branches to the interrupt handler at the vector indicated by u8.

[Operation]

(1) SSP – 4 → SSP

(2) PS → (SSP)

(3) SSP – 4 → SSP

(4) PC + 2 → (SSP)

(5) "0" → I flag

(6) "0" → S flag

(7) (TBR + $3FC_H$ – 4 × u8) → PC

■ Operation of the INTE Instruction

INTE

Branches to the interrupt handler pointed to by vector #9.

[Operation]

(1) SSP – 4 → SSP

(2) PS → (SSP)

(3) SSP – 4 → SSP

(4) PC + 2 → (SSP)

(5) "00100" → ILM

(6) "0" → S flag

(7) (TBR + $3D8_H$) → PC

Do not use the INTE instruction in the handler for the INTE instruction or step trace trap.

Also, the INTE does not generate an EIT during step execution.

■ Operation of the Step Trace Trap

If the T flag in the SCR of the PS is set (enabling the step trace function), a trap occurs after execution of each instruction. The trap causes execution to break.

[Conditions for detecting a step trace trap]

(1) T flag = 1

(2) Not a delayed branch instruction

(3) The currently executing program is not a handler for the INTE instruction or step trace trap.

(4) If the above conditions are satisfied, a break occurs at the end of the current instruction.

[Operation]

(1) SSP – 4 → SSP

(2) PS → (SSP)

(3) SSP – 4 → SSP

(4) address of next instruction → (SSP)

(5) "00100" → ILM

(6) "0" → S flag

(7) (TBR + $3CC_H$) → PC

The user NMI and user interrupts are disabled when the step trace trap is enabled by the T flag.

Similarly, the INTE instruction does not generate EITs.

■ Operation of the Undefined Instruction Exception

The undefined instruction exception is generated if an undefined instruction is detected at instruction decoding.

[Conditions for detecting an undefined instruction exception]

(1) An undefined instruction is detected at instruction decoding.

(2) The instruction is not in a delay slot (not located immediately after a delayed branch instruction).

(3) If the above conditions are satisfied, an undefined instruction exception is generated and execution breaks.

[Operation]

(1) SSP – 4 → SSP

(2) PS → (SSP)

(3) SSP – 4 → SSP

(4) PC → (SSP)

(5) "0" → S flag

(6) (TBR + $3C4_H$) → PC

The saved PC value is the address of the instruction that caused the undefined instruction exception.

■ Operation of the RETI Instruction

The RETI instruction returns from an EIT processing routine.

[Operation]

(1) (R15) → PC

(2) R15 + 4 → R15

(3) (R15) → PS

(4) R15 + 4 → R15

Note that the stack pointer used to restore the PS and PC is selected based on the contents of the S flag. It is recommended that you set the S flag to "1" to set USP as R15 if executing instructions that operate on R15 (stack pointer) in the interrupt handler. In this case, restore the S flag to "0" prior to the RETI instruction.

■ Precautions

- Delay slot

  Restrictions apply to EITs in the delay slot of branch instructions.
  See 2.7.2 "Branch Instructions" on page 101.

2.9 RESET (DEVICE INITIALIZATION)

2.9.1 Outline

This chapter explains reset operations that initialize the FR50 series of devices.

When a reset factor occurs, the FR50 series of devices stops all programs and hardware operations and initializes.

This state is called the reset state.

When the reset factor is released, the device starts the programs and hardware operations from the initial state.

The series of operations from this reset state to the operation start is called the reset sequence.

2.9.2 Reset Level

The reset operation of the FR50 series of devices is divided into two levels: setting initialization reset (INIT), and operation initialization reset (RST). The occurrence factor and initialization type of each reset level differ.

Each reset level is explained below.

■ Setting Initialization Reset (INIT)

The most powerful reset level, which initializes all settings, is called setting initialization reset (INIT).

INIT initializes the following settings:

- Device operation mode (settings of bus mode and external bus width)
- All settings related to internal clocks (clock source selection, PLL control, division rate setting)
- All settings of external bus extension interface
- All settings related to other pin states
- All settings to be initialized by operation initialization reset (RST)

For details, see each Function Description.

After power-on, always execute INIT at the INITX pin.

■ Operation Initialization Reset (RST)

The normal reset level, which initializes program operation, is called operation initialization reset (RST). When INIT occurs, RST occurs at the same time.

RST initializes the following settings:

- Program operation
- CPU and internal bus
- Register setting values of resources
- I/O port setting

- All settings related to CS0 areas of external bus

For details, see each function description.

2.9.3 Reset Source

This chapter explains each reset source in the FR50 series of devices and the reset levels. Past reset sources can be obtained by reading the reset source register (RSRR). For details of registers and flags that appear in each description, see section 5.8 "REGISTERS IN CLOCK GENERATION CONTROL BLOCK" on page 157.

■ Input to INITX Pin (Setting Initialization Reset Pin)

The INITX pin (external pin) functions as the setting initialization reset pin.

The setting initialization reset (INIT) request occurs when a Low-level signal is input to the INITX pin.

This request is released by inputting a High-level signal to the INIT pin.

When setting initialization reset (INIT) is executed according to the INIT request, the INIT bit (bit 15) in the reset source register (RSRR) is set.

INIT executed by this request is the most powerful reset of all reset sources; it takes priority over all inputs, operations, and states. Immediately after power-on, always execute INIT at the INITX pin. Also hold a Low-level input to the INITX pin to secure the oscillation stabilization waiting time of the requested oscillator circuit.

(Executing INIT at the INITX pin initializes the set stabilization oscillation waiting time to the minimum value.)

- Occurrence factor: Low-level input to external INITX pin
- Release factor: High-level input to external INITX pin
- Occurrence level: Setting initialization reset (INIT)
- Corresponding flag: INIT bit (bit 15)

■ Input to RSTX Pin (Operation Initialization Reset Pin)

The RSTX pin (external pin) functions as the operation initialization reset pin.

The operation initialization reset (RST) request occurs when a Low-level signal is input to the RSTX pin.

This request is released by inputting a High-level signal to the RSTX pin.

When operation initialization reset (RST) is executed according to the RSTX request, the ERST bit (bit 12) in the reset source register (RSRR) is set.

If the SYNCR bit (bit 7) in the time-base counter control register (TBCR) is set, RST is executed according to this request only after all bus access stops. For this reason, it may take some time RST until occurs depending on the bus usage state.

- Occurrence factor: Low-level input to external RSTX pin
- Release factor: High-level input to external RSTX pin

- Occurrence level: Operation initialization reset (RST)
- Corresponding flag: ERST bit (bit 12)

■ **Internal Power down reset (see chapter 22 "POWER DOWN RESET" on page 525)**

Like a low-level signal at the RSTX pin, the internal power down reset will cause an operation initialization reset (RST).

■ **STCR: Write to SRST Bit (Software Reset)**

When 0 is written to the SRST bit (bit 4) in the standby control register (STCR), a software reset request occurs.

The software reset request is the operation initialization reset (RST) request.

If RST occurs, the accepted software reset request is released.

When operation initialization reset (RST) is executed according to the software reset request, the SRST bit (bit 11) in the reset source register (RSRR) is set.

If the SYNCR bit (bit 7) in the time-base counter control register (TBCR) is set, RST is executed according to this request only after all bus access stops. For this reason, it may take some time RST until occurs depending on the bus usage state.

- Occurrence factor: 0 written to SRST bit (bit 4) in STCR
- Release factor: Execution of operation initialization reset (RST)
- Occurrence level: Operation initialization reset (RST)
- Corresponding flag: SRST bit (bit 11)

■ **Watchdog Reset**

Writing data to the watchdog timer control register (RSRR) starts the watchdog timer. If A5h and 5Ah are not written subsequently to the watchdog reset occurrence delay register (WPR) within the cycle set by the WT1 and WT0 bits (bits 9 and 8) in the RSRR, a watchdog reset request occurs. The watchdog reset request is the setting initialization reset (INIT) request. When INIT or RST occurs, the accepted watchdog reset request is released.

If INIT is executed according to the watchdog reset request, the WDOG bit (bit 13) in the reset source register (RSRR) is set. If INIT is executed according to the watchdog reset request, the set oscillation stabilization waiting time is not initialized.

- Occurrence factor: Elapse of set watchdog timer cycle
- Release factor: Execution of setting initialization reset (INIT) or operation initialization reset (RST)
- Occurrence level: Setting initialization reset (INIT)
- Corresponding flag: WDOG bit (bit 13)

■ **Input to HSTX Pin (Hardware Standby Pin)**

The HSTX pin (external pin) functions as the hardware standby pin. A hardware standby request occurs when a Low-level signal is input to the HSTX pin.

When the hardware standby request is accepted and the device switches to the hardware

standby state, setting initialization reset (INIT) occurs at the same time.

When a High-level signal is input to the HSTX pin, the hardware standby request is released. Setting initialization reset (INIT) is also released.

If the INIT request occurs at the external INITX pin when the device is in the hardware standby state, the hardware standby request is released but INIT remains in the state in which it occurred.

If INIT occurs when the device is in the hardware standby state, the HSTB bit (bit 14) in the reset source register (RSRR) is set.

If the hardware standby request occurs immediately after power-on, setting initialization reset (INIT) at the INITX pin has priority over other resets. However, if INIT at the INITX pin is released subsequently and the device switches to the hardware standby state, the set oscillation waiting time is initialized to the maximum value and the oscillation stabilization waiting time after the hardware standby request has been released is set to the maximum value.

- Occurrence factor: Low-level signal input to external HSTX pin
- Release factor: High-level signal input to external HSTX pin or setting initialization reset (INIT) at INITX pin
- Occurrence level: Setting initialization reset (INIT)
- Corresponding flag: HSTB bit (bit 14)

2.9.4 Reset Sequence

When the reset factor is released, the device starts execution of the reset sequence. The reset sequence operation differs for each reset level and is explained for each reset level below.

■ Setting Initialization Reset (INIT) Release Sequence

When the setting initialization reset (INIT) request is released, the device executes the following operations in sequence:

1) Releases INIT and switches to oscillation stabilization waiting state
2) Retains operation initialization reset (RST) state during oscillation stabilization waiting time (bits 3 and 2 (OS1 and OS0) in STCR) and stops internal clock
3) Sets operation initialization reset (RST) and starts internal clock operation
4) Releases RST and switches to normal operation state
5) Reads mode vector from 000FFFF8h address
6) Writes mode vector to mode register (MODR) at 000007FDh address
7) Reads reset vector from 000FFFFCh address
8) Writes reset vector to program counter (PC)
9) Starts program operation at address indicated by PC, in case of single chip mode this is the first address of the internal boot ROM (see chapter 4 "BOOT ROM / CONFIGURATION REGISTER" on page 139).

■ Operation Initialization Reset (RST) Release Sequence

When the operation initialization reset (RST) request is released, the device executes the

following operations in sequence:

1) Releases RST and switches to normal operation state
2) Reads reset vector from 000FFFFCh address
3) Writes reset vector to program counter (PC)
4) Starts program operation at address indicated by PC

■ **Hardware Standby Release Sequence**

When the hardware standby request or low-voltage detection standby request is released, the device executes the following operations in sequence:

1) Sets setting initialization reset (INIT), starts internal clock operation, releases pin high-impedance processing
2) Releases INIT and switches to oscillation stabilization waiting state
3) Retains operation initialization reset (RST) state during oscillation stabilization waiting time (bits 3 and 2 (OS1 and OS0) in STCR) and stops internal clock
4) Sets operation initialization reset (RST) and starts internal clock operation
5) Releases RST and switches to normal operation state
6) Reads mode vector from 000FFFF8h address
7) Writes mode vector to mode register (MODR) at 000007FDh address
8) Reads reset vector from 000FFFFCh address
9) Writes reset vector to program counter (PC)
10) Starts program operation at address indicated by PC

2.9.5 Oscillation Stabilization Waiting Time

The device automatically switches to the oscillation stabilization waiting state when it returns from a state in which the original oscillation stopped, or may have stopped.

This function disables use of unstable oscillator output after oscillation has started.

During the oscillation stabilization waiting time, supply of internal and external clock stops and only the built-in time-base counter operates. The device waits for the elapse of the stabilization waiting time set by the standby control register (STCR).

The oscillation stabilization waiting operation is detailed below.

■ **Oscillation Stabilization Waiting Factors**

The oscillation stabilization waiting occurrence factors are shown below.

1) When setting initialization reset (INIT) released.
   Immediately after INIT has been released, the device switches to the oscillation stabilization waiting state.
   After the oscillation stabilization waiting time has elapsed, the device switches to the operation initialization reset (RST) state.

2) At return from stop mode.
Immediately after the stop mode has been released, the device switches to the oscillation stabilization waiting state.
However, if the stop mode is released by the setting initialization reset (INIT) request, the device switches to the INIT state. It switches to the oscillation stabilization waiting state after INIT has been released.
After the oscillation stabilization waiting time has elapsed, the device switches to the state corresponding to the stop mode release factor.

At return due to input of valid external interrupt request (including NMI)

-> Device switches to normal operation state

At return due to INIT request

-> Device switches to RST state

At return due to the RST request

-> Device switches to RST state

3) At return due to error state occurrence (valid when PLL selected).
If a PLL control error(*) occurs when the PLL is operating as a source clock, the device switches automatically to the oscillation stabilization waiting time to secure the PLL lock time.
After the oscillation stabilization waiting time has elapsed, the device switches to the normal operation state.
(*): Change of division rate during use of PLL and PLL operation enable bit error, etc.

■ Selection of Oscillation Stabilization Waiting Time

The built-in time-base counter is used to count the oscillation stabilization waiting time.

When the device switches to the oscillation stabilization waiting state because an oscillation stabilization waiting factor occurs, the built-in time-base counter starts counting the oscillation stabilization waiting time after it has been initialized once.

The OS1 and OS0 bits (bits 3 and 2) in the standby control register (STCR) can be used to select and set one of the four oscillation stabilization waiting times.

The set oscillation stabilization waiting time can be initialized only by setting initialization reset (INIT) at the INTX pin (external pin). At other resets (e.g., watchdog reset, hardware standby, and operation initialization reset (RST)), the oscillation stabilization waiting time set before the reset is held.

The four oscillation stabilization waiting times assume the following uses:

1) OS1, OS0 = 00: No oscillation stabilization waiting time
(When PLL and oscillator not stopped in stop mode)

2) OS1, OS0 = 01: PLL lock waiting time (When oscillator not stopped in external clock input or in stop mode)

3) OS1, OS0 = 10: Oscillation stabilization waiting time (middle)
(When fast-stabilizing oscillator (e.g., ceramic oscillator) used)

4) OS1, OS0 = 11: Oscillation stabilization waiting time (long)
(When ordinary crystal oscillator, etc., used)

Immediately after power-on, always execute INIT at the INITX pin. Also hold Low-level input to the INITX pin to secure the oscillation stabilization waiting time of the oscillator circuit requested by the oscillator circuit.

(Executing INIT at the INITX pin initializes the set stabilization oscillation waiting time to the minimum value.)

If the hardware standby request occurs immediately after power-on, setting initialization reset (INIT) at the INITX pin has priority over other resets. However, if INIT at the INITX pin is released subsequently and the device switches to the hardware standby state, the set oscillation waiting time is initialized to the maximum value and the oscillation stabilization waiting time after the hardware standby request has been released is set to the maximum value.

2.9.6 Reset Mode

Operation initialization reset (RST) usually supports two modes: ordinary (asynchronous) reset mode, and synchronous reset mode. Use the SYNCR bit (bit 7) in the time-base counter control register (TBCR) to select the mode.

The set reset mode can be initialized only by setting the initialization reset (INIT).

INIT is always executed asynchronously.

Ordinary (asynchronous) reset and synchronous reset are explained below.

■ Ordinary (Asynchronous) Reset

Immediate switching to the RST state or the hardware standby state when the RST or hardware standby request occurs is called ordinary (asynchronous) reset.

When the RST or hardware standby request is accepted in the ordinary (asynchronous) reset mode, the device immediately switches to the RST or hardware standby state without reference to the internal bus access state.

The result of bus access being performed when this mode switches to another mode is not guaranteed. However, the bus access request is always accepted.

When the SYNCR bit (bit 7) in the TBCR is 0, the device operates in the ordinary (asynchronous) reset mode.

The initial value after INIT has occurred is the ordinary reset mode.

■ Synchronous Reset

Switching to the RST or hardware standby state after all bus accesses have stopped when the RST or hardware standby request occurs is called synchronous reset.

Even if the RST or hardware standby request is accepted in the synchronous reset mode, the device does not switch to the RST or hardware standby state if the internal bus is being accessed.

When the above request is accepted, a sleep request is issued to the internal buses. When each bus stops operation and switches to the sleep state, the device switches to the RST or hardware standby state. All bus accesses stop when the synchronous reset mode switches to another mode and the results of all bus accesses are guaranteed.

However, if bus access does not stop for some reason, each request cannot be accepted. (Even in this case, INIT becomes valid immediately.) Bus access does not stop in the following cases:

1) When BRQ (bus release acknowledge) valid, and internal bus issues new bus access request

2) When RDY (ready request) input continuously to external extension bus interface and bus wait valid. In the following case, the device switches finally to each state. However, this switching takes a long time.

Note: The DMA controller does not delay device switching to each state because it stops transfer when it receives a request.
When the SYNCR bit (bit 7) in the TBCR is 1, it indicates the synchronous reset mode.
After INIT has occurred, the initial value returns to the ordinary (asynchronous) reset mode.

2.10 OPERATING MODES

There are two operating modes; bus mode, and access mode.

Bus Mode		Access Mode
Single chip		
Internal ROM, external bus		32-bit bus width
	→	16-bit bus width
External ROM, external bus		8-bit bus width

1) Bus mode
In the bus mode, the internal ROM operations and external access functions are controlled. It is defined by the mode setup pins (**MD2**, **MD1**, and **MD0**) and the **ROMA** bit values in the mode data.

2) Access mode
In the access mode, the external data bus width is controlled. It is defined by WTH1 and WTH0 bits in the mode register and BW1 and BW0 bits in AMD0 to AMD7 (Area MoDe Register).

2.10.1 Bus mode

There are three bus modes in the FR50 series. For details, see section 2.10.2 "Mode Setting" on page 126.

■ Bus mode 0 (single chip mode)

This mode is valid for Fbus RAM in addition to the internal I/O. Ibus RAM and Dbus RAM. Access to other areas are invalid.

■ Bus mode 1 (internal ROM external bus mode)

This mode is valid for Fbus RAM (internal ROM etc.) in addition to the internal I/O, Ibus RAM and Dbus RAM.

Access to other areas is access to external space. A part of the external pins function as bus pins.

■ Bus mode 2 (external ROM external bus mode)

This mode is valid for the internal I/O, Ibus RAM and Dbus RAM.

It forbids access to Fbus RAM; all accesses are to external space. A part of the external pins function as bus pins.

2.10.2 Mode Setting

The FR50 series of devices uses mode pins (MD2 to MD0) and a mode register (MODR) to set the operation mode.

■ Mode Pins

Three mode pins (MD2 to MD0) are used to specify mode vector fetch and set a test mode.

Table 2.10.2 Mode Setting

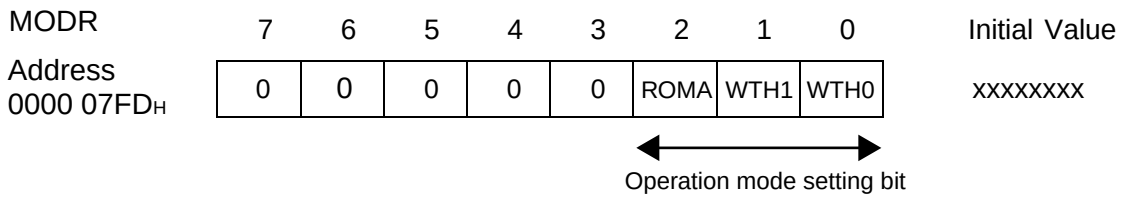
Mode Pins MD2 MD1 MD0	Mode name	Reset vector access area	Remarks
0 0 0	Internal ROM mode vector	Internal	
0 0 1	External ROM mode vector	External	The mode register is used to set the bus width.
remaining settings			Reserved, but see chapter 32 about flash memory mode.

■ Mode Register (MODR)

The data to be written to 0000_7FDH using mode vector fetch is called mode data.

MODR is located at 0000_07FDH. After an operation mode has been set in MODR, the device operates in this operation mode. MODR is set only when a reset factor (INIT level) occurs. User programs cannot write data to MODR.

< Mode Register (MODR)>



[Bits 7 to 3]: (Reserved bits)

Always set 00000 at bits 7 to 3. Operation is not guaranteed when other values are set.

[Bit 2]: ROMA (internal ROM enable bit)

The ROMA bit is used to set whether to validate the internal ROM area (Fbus memory area).

ROMA	Function	Remarks
0	External ROM mode	Access to the Fbus area is external.
1	Internal ROM mode	

[Bits 1 and 0]: WTH1 and WTH0 (bus width/single chip mode specifying bits)

The WTH1 and WTH0 bits are used to set the bus width (valid when operation mode is external bus mode) and the single chip mode. When the operation mode is the external bus mode, this value is set at the BW1 and BW0 bits of AMD0 (CS0 area).

WTH1	WTH0	Function	Remarks
0	0	8-bit bus width	External bus mode
0	1	16-bit bus width	External bus mode
1	0	32-bit bus width	External bus mode
1	1	Single chip mode	

2.10.3 Fixed Vector

If MB91360 devices are started in mode MD[2:0]=000, the internal fixed mode vector (FMV=0x06) and the fixed reset vector are used. The fixed reset vector points to the start address of the internal BootROM.

This enables access to the F-bus area, to the internal CAN modules and the internal flash memory.

See also chapter 4 "BOOT ROM / CONFIGURATION REGISTER" on page 139.

CHAPTER 3 INSTRUCTION CACHE

This chapter describes the instruction cache memory included in MB91360 family members and it operation. This only applies to MB91FV360GA.

3.1 GENERAL DESCRIPTION

The instruction cache is temporary memory. When an external low-speed memory accesses an instruction code,the instruction cache stores the single-accessed code to increase the second and subsequent access speeds.Setting this memory to the RAM mode enables software to directly read and write instruction cache data RAM and tag RAM. To turn on instruction cache once and then to turn it off, always use the subroutine described in Section 3.5.

3.2 MAIN BODY STRUCTURE

- FR basic instruction length: 2 bytes
- Block arrangement system: 2-way set associative system
- Block
  One way consists of 128 blocks.
  One block consists of 16 bytes (= 4 sub-blocks).
  One sub-block consists of 4 bytes (= 1 bus access unit).

Figure 3.2a Instruction Cache Structure

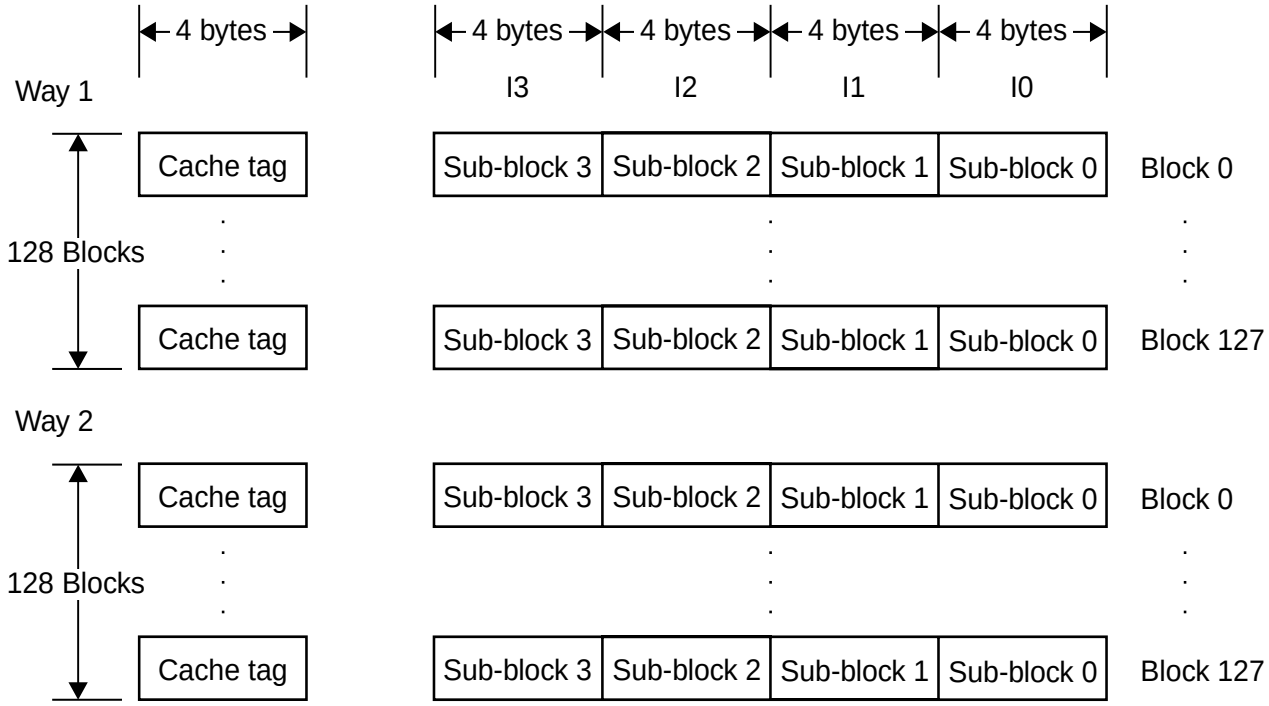
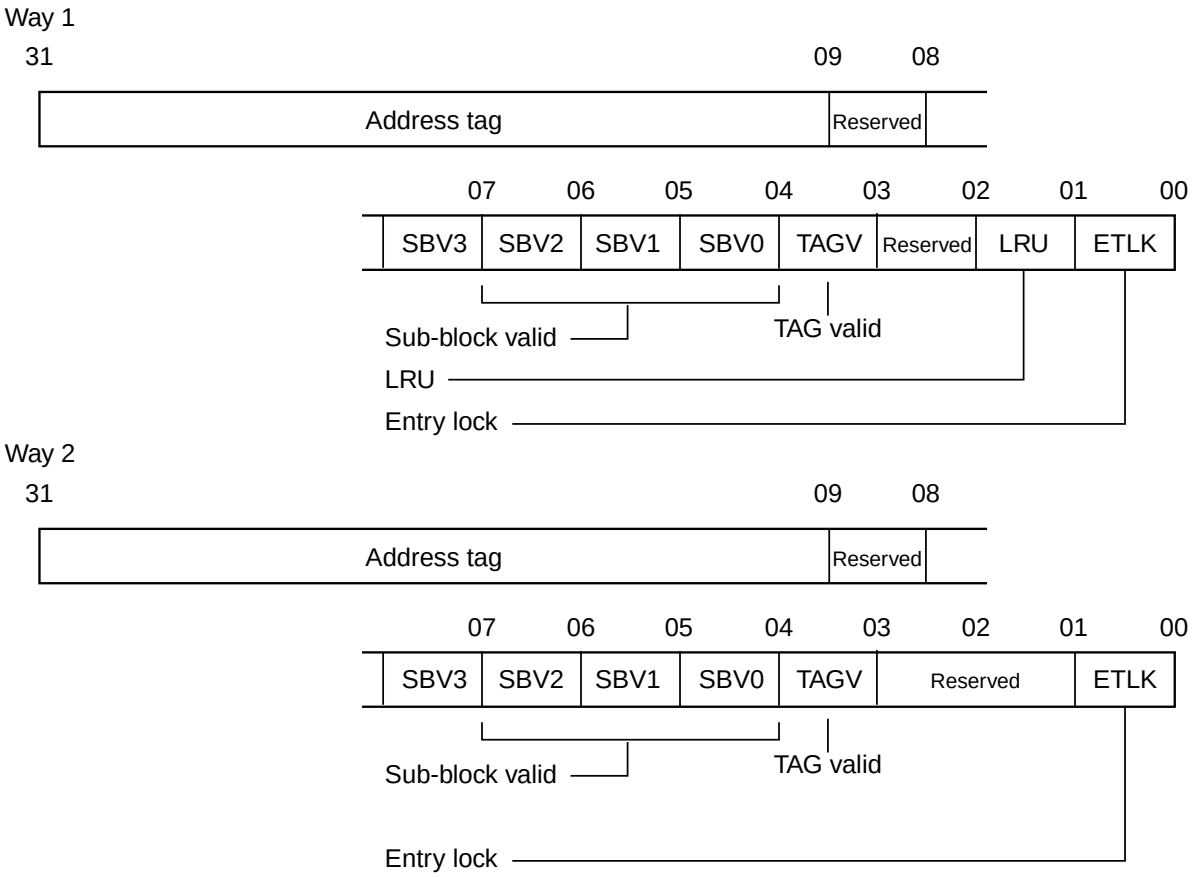


Figure 3.2b Instruction Cache Tag



[Bits 31 to 9] Address tag

The upper 23 bits of the memory address of the instruction cached in the associated block are stored.

The memory address **IA** of the instruction data stored in sub-block **k** of block **i** is as follows:

IA = address tag x $\mathbf{2^{11}}$ + **i** x $\mathbf{2^{4}}$ + **k** x $\mathbf{2^{2}}$

Performs coincidence checkout of instruction address that CPU requests access to.

The following operations are performed depending on the tag checkout results.

1) When the requested instruction data is in the cache (hit), the cache transfers the data to the CPU within one cycle.

2) When the requested instruction data is not in the cache (miss), the CPU and cache simultaneously acquire the data (1 sub-block) by external access.

[Bits 7 to 4] **SBV3** to **SBV0**: Sub-block valid bits

When **SBV*** = **1**, the current instruction data at the tagged address is entered in the associated sub-block. Normally, two instructions are stored in a sub-block (except for the immediate value transferinstruction).

[Bit 3] **TAGV**: **TAG** valid bit

This bit indicates whether address tag value is valid.

When this bit is **0**, this block is invalid irrespective of the sub-valid bit (flushed state).

[Bit 1] **LRU** bit (way 1 only)

Present only for way-1 instruction cache tag. Regarding the selected set, this bit indicates whether a way-1 or way-2 entry was accessed in the cache last. When **LRU = 1**, a way-1 set entry was not found in the cache last. Conversely, when **LRU = 0**, a way-2 set entry was not found in the cachelast.

[Bit 0] **ETLK**: Entry lock bit

This bit locks all tagged entries within block in cache.

When **ETLK = 1**, the locked state prevails so entries are not updated when they are not found inthe cache. However, invalid sub-blocks are updated. If the data is not found in the cache when way 1 and way 2 are both entry-locked, the external memory is accessed after the loss of one cycle of "cache miss" evaluation.

■ Control register structure

IRBS (32 bit)	31	30	29	28	27	26	25	24	Initial value
0000 0300H	0	0	0	0	0	0	0	0	0000 0000B
	R	R	R	R	R	R	R	R	
	23	22	21	20	19	18	17	16	
	0	0	0	0	0	0	0	1	0000 0001B
	R	R	R	R	R	R	R	R	
	15	14	13	12	11	10	9	8	
(0000 0302H)	IRBS	IRBS	IRBS	IRBS	-	-	-	-	0010 ----B
	R/W	R/W	R/W	R/W	-	-	-	-	
	7	6	5	4	3	2	1	0	
	-	-	-	-	-	-	-	-	
	-	-	-	-	-	-	-	-	

IRBS [bits 15 to 12]

These bits are used to set the base address of cache RAM at access in the RAM mode. Align cache RAM in units of 4K bytes. These bits are initial-ized by **INIT**. The initial value is the 00012000H address.

ISIZE (8 bit)	7	6	5	4	3	2	1	0	Initial value
0000 0307H	-	-	-	-	-	SIZE1	SIZE0	-	---- --11B
	-	-	-	-	-	R/W	R/W	-	

ISIZE [bits 1 to 0] : **SIZE1** and **SIZE0**

These bits are used to set the cache size. As shown in Figure I-CACHE-3, the cache size, IRAM size, and address map in the RAM mode change according to setting of these bits. Whenthe cache size is changed, always turn on cache after unlocking flush and entry lock.

CACHE Size Register

SIZE1	SIZE0	Size
0	0	1KB
0	1	2KB
1	0	Setting disabled
1	1	4 KB (initial value)

The **ICHCR** (I-CacHe Control Register) controls the instruction cache operations.

Writing to the **ICHCR** does not affect caching of instructions fetched within three subsequent cycles.

ICHCR (8 bit)	7	6	5	4	3	2	1	0	Initial value
0000 03E7H	RAM	-	GBLK	ALFL	EOLK	ELKR	FLSH	ENAB	0-00 0000B
	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	

If during the execution of the code in the internal boot ROM a valid boot condition is detected this register is set to "0x81".

[Bit 7] **RAM**: RAM Mode

Setting the RAM bit to 1 enables instruction cache to operate in the RAM mode.

If the ENAB bit is 1 (cache on) when this bit is set to 1, cache RAM is mapped to the area specified by IRBS.

[Bit 5] **GBLK**: Global lock bit

This bit locks all current entries in cache.

When GBLK = 1, valid entries in the cache are not updated at a "cache miss." However, invalid sub-blocks are updated. The resulting instruction data fetch operation is the same as for unlocked entries.

[Bit 4] **ALFL**: Auto lock fail bit

If locking a locked entry, the ALFL is set to 1. When entry-auto-locked entry is updated for a locked entry, the new entry is not locked in the cache despite theuser's intention. Referencing is done for program debugging.

This bit is cleared when 0 is written.

[Bit 3] **EOLK**: Entry auto lock bit

This bit specifies whether auto locking feature is enabled or disabled for indi-vidual entries in instruction cache.

Entries accessed (only at miss) while EOLK = 1, are locked when the entry lock bit in the cache tag is set to 1 by hardware. Locked entries are not subse-quently updated at a cache miss. However, invalid sub-blocks are updated.

To assure proper locking, first flush, and then set bit 3.

[Bit 2] **ELKR**: Entry lock release bit

This bit is used to clear entry lock bits at all cache tags.

When ELKR is set to 1, entry lock bits at all cache tags are cleared to 0 at the next cycle.

However, the value of this bit is held only for one clock cycle. It is cleared to 0 at the second and subsequent clock cycles.

[Bit 1] **FLSH**: Flush bit

This bit specifies flushing of instruction cache.

When FLSH = 1, the cache data is flushed. However, the value of this bit is held only for 1 clock cycle. It is cleared to 0 at the second and subsequent clock cycles.

[Bit 0] **ENAB**:Enable bit

This bit enables or disables instruction cache.

When ENAB = 0, the instruction cache is disabled so instruction accessing from the CPU is performed directly relative to the outside, without using the cache.

When the instruction cache is disabled, instructions in the cache are saved.

Figure 3.2c I-Cache Address Map

Figure I-CHCHE-3 RAM Address Map

Address	Cache off RAM off	Cache off RAM on	Cache 4K RAM off	Cache 4K RAM on	Cache 2K RAM off	Cache 2K RAM on	Cache 1K RAM off	Cache 1K RAM on
00010000		TAG (way0)		TAG (way0)		TAG (way0)		TAG(way0)
00010200								
00010400								
00010600								
00010800		TAG (way1)		TAG (way1)		TAG (way1)		TAG(way1)
00010A00								
00010C00								
00010E00								
00011000	IRAM (way0)	**IRAM (way0)**			IRAM (way0)	IRAM (way0)	IRAM (way0)	IRAM (way0)
00011200								
00011400					IRAM (way1)	IRAM (way1)	IRAM (way1)	IRAM (way1)
00011600								
00011800	IRAM (way1)	**IRAM (way1)**						
00011A00								
00011C00								
00011E00								
(IRBS) 00012000				$RAM (way0)		$RAM (way0)		$RAM(way0)
Initial Value 00012200								$RAM(way1)
00012400						$RAM (way1)		
00012600								
00012800				$RAM (way1)				
00012A00								
00012C00								
00012E00								
00013000								

If the IRBS setting value overlaps any other memory areas, IRBS memory ($RAM) has priority over other memo

is a mirror area.

TAG RAM			$RAM		
00010000		Entry at 00x address	00012000		Instruction (SBV0) at 000 address
00010004		Mirror at 00x address	00012004		Instruction (SBV1) at 004 address
00010008			00012008		Instruction (SBV2) at 008 address
0001000C			0001200C		Instruction (SBV3) at 00C address
00010010		Entry at 00x address	00012010		Instruction (SBV0) at 010 address
00010014		Mirror at 00x address	00012014		Instruction (SBV1) at 014 address
					Instruction (SBV2) at 018 address
					Instruction (SBV3) at 01C address
00010020					

Figure I-CACHE-4 Memory Assignment For Each Cache Size

Address	Cache 4K	Cache 2K	Cache 1K	Cache off
000	$RAM (WAY0)	IRAM	IRAM	IRAM
200				
400		$RAM	Not used	
600			$RAM	
800	$RAM (WAY1)	IRAM	IRAM	IRAM
A00				
C00		$RAM	Not used	
E00			$RAM	

Figure I-CACHE-5 Cache Areas

Address	ROMAB=0 (Without ROM)	ROMAB=1 (With ROM)	
00000000	Direct area	Direct area	
00010000	IRAM	IRAM	
00020000			(Even the D-bus RAM area is cached when it is transferred to the AI bus.)
00030000			
00040000	Cache area	Internal ROM	
00100000		Cache area	This area can be set as a non-cache area for each chip selection area.
FFFFFFFF			

3.3 VARIOUS OPERATING MODE CONDITIONS

■ **Cache status in various operating modes**

The table below indicates the prevailing state for disable and flush when the associated bit is changed by bit manipulation instruction, etc.

		Immediately after Reset	Disable	Flush
Cache Memory		The contents are undefined.	The preceding state is held. Rewriting is impossible while the cache is disabled.	The preceding state is held.
Tag	Address Tag	The contents are undefined.	The preceding state is held. Rewriting is impossible while the cache is disabled.	The preceding state is held.
	Sub-block Valid Bit	The contents are undefined.	The preceding state is held. Rewriting is impossible while the cache is disabled.	The preceding state is held.
	LRU	The contents are undefined.	The preceding state is held. Rewriting is impossible while the cache is disabled.	The preceding state is held.
	Entry Lock Bit	The contents are undefined.	The preceding state is held. Rewriting is impossible while the cache is disabled.	The preceding state is held. (entry lock release is required)
	TAG Valid Bit	The contents are undefined.	The preceding state is held. Flushingis possible while the cache is disabled.	All entries are invalid.
RAM			The preceding state is held. Flushing is possible while the cache is disabled.	The preceding state is held.
Control Register	Global Lock	Unlock	The preceding state is held. Rewriting is possible while the cache is disabled.	The preceding state is held.
	Auto lock Fail	No fail	The preceding state is held. Rewriting is possible while the cache is disabled.	The preceding state is held.
	Entry Auto Lock	Unlock	The preceding state is held. Rewriting is possible while the cache is disabled.	The preceding state is held.
	Entry Lock Release	No release	The preceding state is held. Rewriting is possible while the cache is disabled.	The preceding state is held.
	Enable	Disabled	Disabled	The preceding state is held.
	Flush	Not flushed	The preceding state is held. Rewriting is possible while the cache is disabled.	Flushed in cycle following memory accessing. Reverts to 0 subsequently.

■ **Cache Entry Update**

Cache entries are updated as shown in the following table.

	Unlock	Lock
Hit	Not updated	Not updated.
Miss	The memory data is loaded, and the cache entry data is updated.	Not updated at tag miss. Updated when sub-block invalid.

3.4 INSTRUCTION CACHE SPACE FOR CACHING

- Instruction cache can be cached only in external bus space.
- Even if the external memory data is updated by DMA transfer, coherency with cached instructions will not be maintained. If coherency needs must maintained, flush the cache.
- Instruction cache can be set as a non-cache area for each chip selection area. Even in this case, cache-on is subject to a 1-cycle penalty but cache-off is not.

3.5 SETUP FOR FR50 I-CACHE USAGE

1) Initializing

To use the I-Cache, first, clear the cache contents. Erase the old data by setting the register **FLSH** bit and **ELKR** bit to **1**.

```
Idi  #0x000003e7,r0// I-Cache control register address
Idi  #0B00000110,r1// FLSH bit (bit 1)
       //           ELKR bit (bit 2)
stb    r1,@r0// Writing to register
```

The cache is now initialized.

2) Enabling (turning ON) cache

To enable the I-Cache, set the ENAB bit to 1.

```
Idi  #0x000003e7,r0// I-Cache control register address
Idi  #0B00000001,r1// ENAB bit (bit 0)
stb    r1,@r0// Writing to register
```

All subsequently-accessed instructions will be cached.

Cache can be validated and initialized at the same time.

```
Idi  #0x000003e7,r0// I-Cache control register address
Idi  #0B00000111,r1// ENAB bit (bit 0)
       //           FLSH bit (bit 1)
       //           ELKR bit (bit 2)
stb    r1,@r0// Writing to register
```

3) Disabling (turning OFF) cache

To disable the I-Cache, set the ENAB bit to 0.

```
Idi  #0x000003e7,r0// I-Cache control register address
Idi  #0B00000000,r1// ENAB bit (bit 0)
stb  r1,@r0// Writing to register
```

In the resultant state (same as state prevailing after reset), there appears to be no cache. The cache can be turned off if the processing may experience problems due to cache overhead

4) Locking all cached instructions

To lock all the currently-cached instructions in the I-Cache, set the register **GBLK** bit to 1. The **ENAB** bit must also be set to 1. If it is not, the cache is turned OFF, so instructions locked in the cache cannot be used.

```
Idi  #0x000003e7,r0// I-Cache control register address
Idi  #0B00100001,r1// ENAB bit (bit 0)
     //           GBLK bit (bit 5)
stb  r1,@r0// Writing to register
```

5) Locking specific cached instructions

To lock a specific group of instructions (e.g., subroutines) in the cache, set the EOLK bit to 1 before executing such instructions.

Instructions locked in this manner are accessed rapidly as if using high-speed internal ROM.

```
Idi  #0x000003e7,r0// I-Cache control register address
Idi  #0B00001001,r1// ENAB bit (bit 0)
     //           EOLK bit (bit 3)
stb  r1,@r0// Writing to register
```

The above instruction lock becomes effective starting with the instruction next to the stb instruction although it depends on the memory wait count. Set the EOLK bit to 0 when the group of instructions which want to lock is ended.

```
Idi  #0x000003e7,r0// I-Cache control register address
Idi  #0B00000101,r1// ENAB bit (bit 0)
     //           EOLK bit (bit 3)
stb  r1,@r0// Writing to register
```

6) Unlocking cached instructions

To release the lock, proceed as follows.

```
Idi  #0x000003e7,r0// I-Cache control register address
Idi  #0B00000000,r1//Cash disabled
stb  r1,@r0// Writing to register
Idi  #0B00000100,r1// ELKR bit (bit 2)
stb  r1,@r0// Writing to register
```

Only lock information is released; locked instructions are replaced sequentially with new instructions according to the state of the LRU bit.

CHAPTER 4 BOOT ROM / CONFIGURATION REGISTER

This chapter describes the functionality of the embedded boot ROM which is used in single chip mode.

4.1 BOOT ROM

The BootROM is a fixed start-up routine which is located at FF000 (Reset entry) and will therefore be executed after every RST or INIT. The purpose of this ROM is to configure the device after a reset and to provide a simple serial bootloader for programming the embedded Flash memories.

The BootROM contains three logical parts :

■ Chip Initializations

Immediately after each reset, the following settings will be made :

CS0 : 200000…2FFFFF, 32Bit Bus, 1 wait-state (default external access)

CS7 :100000…10FFFF , 16Bit Bus, 1 wait-state (CAN)

In addition, the Table-Base Register will be initialized and the synchronous reset (see TBCR) will be enabled.

■ Check for Bootcondition

After the chip initializations, the "Security-Vector" will be checked (Vector #66).

This check is performed as follows:

Table 4.1a Security Vector Check

Device	Security Vector Check
MB91F362GA	Security vector is FFFFFFFFH
MB91F376G	Security vector is FFFFFFFFH or security vector is outside of 04:4800H - 13: FFFFH,
All other MB91360 series devices	Security vector is FFFFFFFFH or security vector is outside of 08: 0000H - 0F: FFFFH

The purpose of this feature is to disable the bootstraploader due to security reasons; only if the conditions in the table above apply the bootstraploader can be started.

The RSRR (reset cause register) will be read and saved. If no power-on reset (external INITX input, RSRR=0x80) is indicated, a branch to the user application will be initiated.

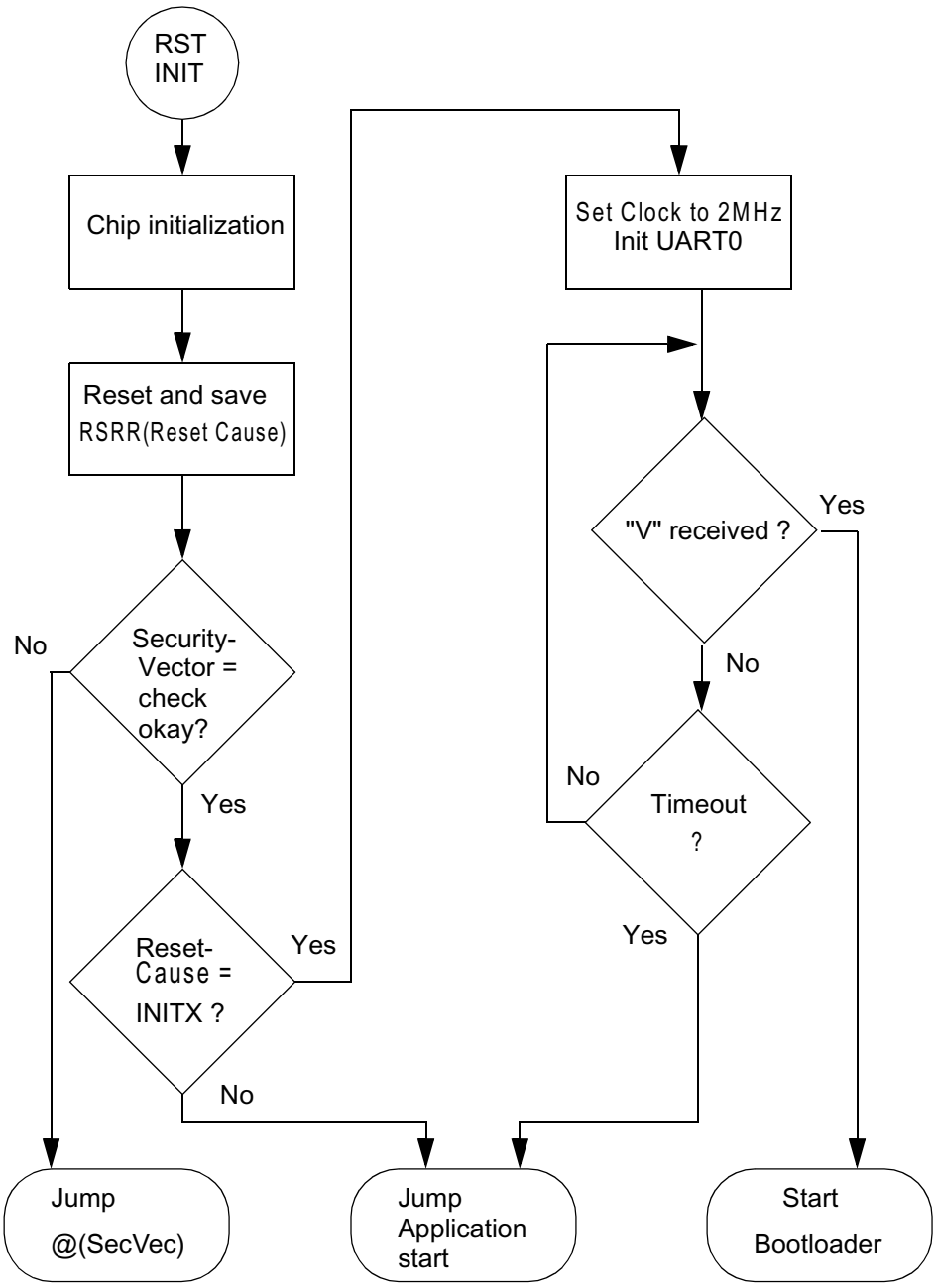
If INITX was detected and the "Security-Vector" check okay, the following conditions must be met in order to start the Bootstraploader:

Within a certain time, the start-up character "V" must be received via UART0 (9600,8N1). The timeout is set to 200ms. On MB91FV360GA the Bootstraploader will also be started, if the external Bootpin (Boot/P93) is high.

■ Bootstraploader

If the Bootcondition was met, an acknowledge character "F" will be transmitted via UART0 to indicate that the Bootloader is ready to accept commands. 4 different commands are possible :

- Receive and write to a specified memory block
- Dump the contents of a specified memory block
- Initiate a "CALL" to a certain location
- Re-dump a calculated checksum for verification



Note: The RSRR register can only be read once. After reading the RSRR-register, the contents

will be saved temporarily in RAM (3D500) and will be present in R4 (C-Compiler convention for parameters) after a branch to Application start or the address specified by the Security-Vector.
The BootROM initializes the chip and changes the settings of some registers (see table below).
The standard start-address (if no other address was specified by the Security-Vector) for user-programs is F4000.

■ Bootstraploader Description

If a valid bootcondition was detected, data can be written to or read from the MCU using a serial protocol. In addition, calls to a dedicated address or re-dumps of checksums (16-Bit length, calculated while receiving or transmitting data) are possible. The communication channel is UART0 (9600 baud ,8 data-bits, no parity,1 stop bit, no handshaking, binary transmission).

Note: Software to transfer data using this protocol is available from Fujitsu.

■ Serial Communication Protocol

Table 4.1b Boot ROM Serial Communication

Command	PC to MCU	MCU to PC	Remark
Read (2)	1 2 Adr (4 Bytes) Size (2 Bytes)	241 (0xF1) 130 (0x82) Binary Dump CS (2Bytes)	 Lo, MidLo, MidHi, Hi Lo, Hi Direct read and dump Bootloader sends 16-Bit check-sum
Write (3)	1 3 Adr (4 Bytes) Size (2 Bytes) Binary Dump	241 (0xF1) 131 (0x83) CS (2 Bytes)	 Lo, MidLo, MidHi, Hi Lo, Hi Receive and store Bootloader sends 16-Bit check-sum
Call (4)	1 4 Adr (4 Bytes)	241 (0xF1) 132 (0x84) **ret. Parameter**	**Calls specified Address and waits for a return. The returned parameter in R4 will be echoed to the PC**
Get Checksum (5)	1 5	241 (0xF1) 133 (0x85) CS (2 Bytes)	MCU re-dumps 16-bit checksum (Lo, Hi) calculated at last write or read operation

■ Registers modified by the BootROM

Table 4.1c Registers modified by Boot ROM

Register	Value	Address	Description
ASR0	0x20	0x640	Area select register 0
AMR0	0x0F	0x642	Area mask register 0
AMD0	0x11	0x660	Area mode register 0
ASR7	0x10	0x65C	Area select register 7
AMR7	0x00	0x65E	Area mask register 7
AMD7	0x29	0x667	Area mode register 7
CSE	0x81	0x668	CS-Select enable register
CMCR	0x0180	0x0164	CAN Clock Enable
TBR	0x0FFC00	-	Table Base Register
TBCR	0x03	0x482	Time-base counter/Sync-RST Register
RSRR	0 (see note)	0x480	Reset Source Register (visible in R4)
SP*	0x3D3F8	-	Stack pointer
CLKR*	0x0 (4MHz) 0x36 (32kHz)	0x484	Clock source control register
DIVR0*	0x0 (4Mhz) 0x77 (32kHz)	0x486	Clock division register 0 (CPU and RBus)
DIVR1*	0x0 (4Mhz) 0x70 (32kHz)	0x487	Clock division register 1 (ext. Bus)
PFRQ*	0x03	0x41A	Port function register Port Q (UART0)
SMR0*	0x31	0x63	UART0 Mode Register
SCR0*	0x13	0x62	UART0 Control Register
UTIMR0*	0x05	0x68	U-Timer0 Reload
UTIMC0*	0x82	0x6B	U-Timer0 Control
FMWT	0x13	0x00007004H	Flash Wait Control Register
FMCS	0x60	0x00007000H	Flash Control Status Register

Note: * these registers will only be modified if a check for valid bootcondition is performed

4.2 CONFIGURATION REGISTER (F362 MODE REGISTER F362MD)

This register is used to control which pins of the external bus interface are active, where the pins for the external DMA channel are located and which I²C module is used.

address	bit 15	bit14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
00001FEH	ADRSWAP	ASYMCLKT	HIZ_D_A	HIZ_ECLK	HIZ_D_23_16	HIZ_D_15_0	DMASWP	IICSEL
access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
initial value	0	0	0	0	0	0	0	0

Bit 15: ADRSWAP (only available on MB91F369GA)

0: Pin 23 = CS4X, Pin 24 = CS5X, Pin 25 = CS6X

1: Pin 23 = A [21], Pin 24 = A[22], Pin 25 = A[23]

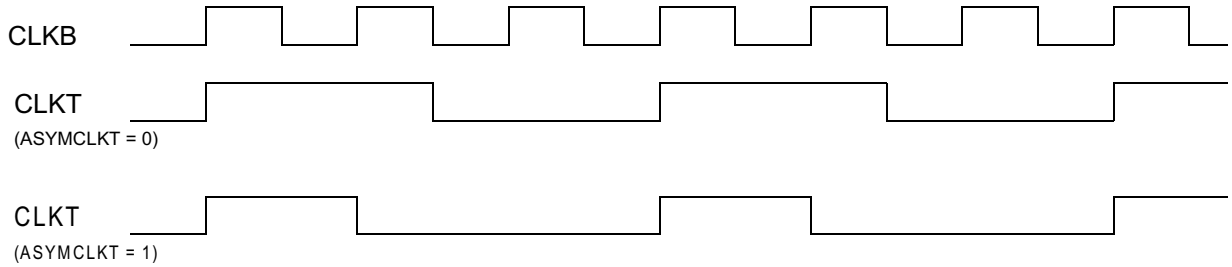
This is valid if the corresponding PFRs are set to output CSX or address signals, respectively.

Bit 14: ASYMCLKT (not available on MB91F364G)

0: duty cycle for CLKT is 1:1

1: for odd ratios between CLKB and CLKT (1:3, 1:5, ...) the high pulse of CLKT will be shortened by the length of 1 high pulse of CLKB

Example for CLKB:CLKT = 1:3



Bit 13: HIZ_D_A (only availabe on devices with external interface)

0: D[31:24], A[20:0], RDX, WR3X, WR2X, WR1X, WR0X outputs enabled

1: D[31:24], A[20:0], RDX, WR3X, WR2X, WR1X, WR0X outputs high-Z

Bit 12: HIZ_ECLK (only availabe on devices with external interface)

0: CLK output enabled

1: CLK output high-Z

Bit 11: HIZ_D_23_16 (only availabe on devices with external interface)

0: D[23:16] outputs enabled

1: D[23:16] outputs high-Z

Bit 10: HIZ_D_15_0 (only availabe on devices with external interface)

0: D[15:0] outputs enabled

1: D[15:0] outputs high-Z

Bit 9: DMASWP (only on MB91F362GB and MB91FV360GA)

0: external DMA channel 0 swapping disabled

1: external DMA channel 0 swapping enabled (see table below for pin locations)

DMA Pin Swap	MB91F362GB	MB91FV360GA
DREQ0 <-> BRQ	Pin 61 <-> Pin 37	Pin 193 <-> Pin 88
DACK0 <-> BGRNTX	Pin 62 <-> Pin 36	Pin 135 <-> Pin 139
DEOP0 <-> AS	Pin 63 <-> Pin 43	Pin 84 <-> Pin 87

Bit 8: IICSEL (only available on devices with 100 kHz **and** 400 kHz I^2C interfaces)

0: selection of 100 kHz I^2C interface (I^2C-1)

1: selection of 400 kHz I^2C interface (I^2C-2)

CHAPTER 5 CLOCK GENERATION AND DEVICE STATES

This chapter describes details about generation and control of the clock used to control the MB91360. In addition device states and low power modes are explained. This chapter assumes operation without subclock. For a description of subclock operation see the corresponding chapter.

5.1 CLOCK GENERATION OUTLINE

The MB91V360 generates internal operating clocks as follows:

- Base clock generation: Device scales clock source input by 2 (X clock) or oscillates base clock with PLL to generate basic clock (PLL clock)
- Generation of each internal clock: Device scales base clock to generate clocks supplied to each block

Generation and control of each clock are explained below.

For details of the registers and flags, see sections 5.7 "CLOCK GENERATION CONTROL" on page 156 and 5.8 "REGISTERS IN CLOCK GENERATION CONTROL BLOCK" on page 157.

Some devices allow the operation of the RTC module based on a separate 32 kHz subclock. See chapter 30 "SUBCLOCK" on page 651 about subclock operation for more details.

5.2 BASE CLOCK GENERATION

A 4 MHz crystal and capacitances need to be connected to pins X0 and X1.

All clocks including the external bus clock are generated by the device itself.

The built-in PLL (see block diagram in section 5.7) is used to scale the main clock.

An internal base clock is generated by selecting one of the following source clocks:

- Source clock generated by scaling main clock input by 2 (X clock = 2 Mhz)
- Source clock generated by multiplying main clock using PLL (PLL clock)

Source clock selection is controlled by the setting of the clock source control register (CLKR).

5.3 PLL CONTROL

Oscillation enable/disable and division rate is controlled by settings of the clock source control register (CLKR).

5.3.1 PLL Oscillation Enable/Disable

PLL oscillation enable/disable for the PLL is controlled according to the setting of the PLL1EN bit (bit 10) in the CLKR.

PLL1EN (bit 10) in the CLKR is initialized to 0 after setting initialization reset (INIT) has been executed and PLL oscillation is stopped. When PLL oscillation is stopped, PLL clock cannot be selected as the source clock.

When program operation is started, first, set the multiply rate for the PLL and enable PLL oscillation. Then, switch the source clock PLL clock after the PLL lock waiting time has elapsed. In this case, the time-base timer interrupt should be used to indicate the end of the PLL lock waiting time.

When PLL clock is selected as the source clock, the PLL operation cannot be stopped (write operations to the corresponding register bits are ignored). To stop the PLL e.g. when the device switches to the stop mode, reselect X clock as the source clock, then stop the PLL.

When the OSCD1 bit (bit 0) in the standby control register (STCR) is set so that oscillation stops in the stop mode, the PLLs also automatically stop when the device switches to the stop mode.

When the device subsequently returns from the stop mode, the PLLs automatically start oscillation.

5.3.2 PLL Multiply Rate

The PLL1S2, PLL1S1, and PLL1S0 bits (bits 14 to 12) in the clock source control register (CLKR) are used to set the multiply rate of PLL.

All these bits are initialized to 0 after setting initialization reset (INIT) has been executed.

■ PLL multiply rate setting

To change the PLL multiply rate setting at the beginning, set the rate before or when enabling PLL operation. After modifying the multiply rate, switch the source clock to PLL clock when the lock waiting time has elapsed. In this case, the time-base timer interrupt should be used to indicate that the PLL lock waiting time is over.

When changing the PLL multiply rate setting during operation, switch the source clock to X clock, then change the setting. After changing the multiply rate, as explained above, switch the source clock back to PLL clock when the lock waiting time has elapsed.

The PLL multiply rate setting can be changed even when the PLL is in use. However, in this case, the device switches automatically to the oscillation stabilization waiting state after the multiply rate setting has been changed.

Program operation stops until the oscillation stabilization waiting time has elapsed. If the clock source is switched to X clock, program operation does not stop.

5.4 WAITING TIMES

After enabling of the oscillators it will take some time before a stable clock signal from the oscillators becomes available. This time is called the oscillation stabilization waiting time.(see section 5.10 "CLOCK CONTROL SECTION RESOURCES" on page 176).

After startup of the PLLs again a certain time is needed before the set PLL output frequency stabilizes. This time is called PLL lock waiting time

The following lines give some details about when it is necessary to wait for the elapsing of these times.

■ Waiting time after power-on

After power-on, it is necessary to postpone operation until the oscillation stabilization waiting time of the oscillator circuit is over.

The oscillation stabilization waiting time is initialized to its default value by inputting a low-level signal to the INITX pin. See section 5.8.2 "STCR: Standby Control Register" on page 159.

In this initial state, the lock waiting time need not be considered because no PLL operation is enabled.

■ Waiting time after initialization reset

When the initialization reset (INIT) is released, the device switches to the oscillation stabilization waiting state. In this state the oscillation stabilization waiting time is generated internally.

In the oscillation stabilization waiting state caused by low-level signal input to the INITX pin at power -on, the time is initialized to its default value.

However, if a low-level signal is input to the HSTX pin (hardware standby pin) in this state, the device switches to the hardware standby state and the oscillator circuit stops. This causes the oscillation stabilization waiting time to be initialized to the maximum value for safety.

If INIT is caused by an event other than signal input to the INITX pin after program operation has been started, the oscillation stabilization waiting time set by the program is used when INIT is released.

In these states, the lock waiting time needs not to be considered because no PLL operation is enabled.

■ Waiting time after PLL operation enabled

When operation of the inactive PLL is enabled after program operation has started, PLL output cannot be used before the lock waiting time has elapsed.

If the PLL is not selected as the source clock, program operation can be executed even during the lock waiting time.

In this case, the time-base timer interrupt should be used to indicate that the PLL lock waiting time is over.

■ Waiting time after PLL multiply rate changed

When the multiply rate setting of the operating PLL is enabled after program operation has been started, PLL output cannot be used unless the lock waiting time has elapsed.

If the PLL is not selected as the source clock, program operation can be executed even during the lock waiting time.

In this case, the time-base timer interrupt should be used to show that the PLL lock waiting time has elapsed.

■ **Waiting time after return from stop mode**

When the device switches to the stop mode after program operation had been started, the oscillation stabilization waiting time set by the program is used at return from stop mode.

If the oscillator circuit is set to stop in the stop mode, the oscillation stabilization waiting time of the oscillator circuit or the lock waiting time of the PLL being used, whichever is longer, becomes necessary. Set the oscillation stabilization waiting time before switching the device to the stop mode.

If the oscillator circuit is set not to stop in the stop mode, the oscillation stabilization waiting time should be set to the minimum value before the device switches to the stop mode. (However, if the hardware standby request is input in the stop mode after the oscillator circuit stops, the oscillation stabilization stop time becomes necessary after return. However, if the oscillation stabilization waiting time is set to the minimum value, operation after return is not guaranteed because this waiting time cannot be acquired. In this case, set the oscillation stabilization waiting time for the oscillator circuit in advance.)

5.5 CLOCK DISTRIBUTION

Operating clocks for each function are created from the base clock generated from the source clock. The device has three main internal operating clocks. The division rate of each of these clocks can be set independently.

Each internal operating clock is explained below.

■ CPU clock (CLKB)

The CPU clock is used by the CPU, internal memory, and internal buses.

The following circuits use the CPU clock.

- CPU
- Instruction cache
- Built-in RAM, built-in Flash and ROM (see chapter 32 "FLASH MEMORY" on page 667 for related wait cycle setting)
- Bit search module
- I bus, D bus, X bus, F bus
- DMA controller
- DSU

The upper frequency limit is device dependent, see table in first chapter. Combinations of multiply and division rates that exceed this upper frequency limit cannot be set.

■ Resource clock (CLKP)

The resource clock is used by resources and peripheral buses.

The following circuits use the resource clock.

- Peripheral bus
- Clock control block (bus interface block only)
- Interrupt controller, External interrupt input
- I/O port bus, Resource I/O ports
- Stepper Motor Controllers
- UART modules, U-Timers
- SIO, SIO Prescalers
- 16-bit reload timer
- A/D converter, D/A converter
- IO-Timer
- Sound Generator
- PWM Timer
- I^2C Interface
- Alarm Comparator, Power-down reset

The operable upper frequency limit is 32 MHz. Combinations of multiply and division rates that exceed this upper frequency limit cannot be set.

■ **External bus clock (CLKT)**

The external bus clock is used by the external extension bus interface.

The following circuits use the external bus clock.

- External extension bus interface
- External CLK output
- Bus Interface I/O ports

The operable upper frequency limit is 32 MHz. Combinations of multiply and division rates that exceed this upper frequency limit cannot be set.

In addition to those three clocks there are:

■ **Clock for the CAN modules (CANCLK)**

This clock is directly taken from the output of PLL1. The rate for this clock is controlled by a progammable 8 bit prescaler (Register CMCR).
The value for CANCLK is PLL clock / (prescaler value + 1).

■ **Clock for the RTC (Real Time Clock module)**

The clock is directly either the output of the 4 MHz or the 32 KHz oscillator circuit - depending on the device and the SELCLK pin setting.

5.6 CLOCK PULSE DIVISION

The division rate of each internal operating clock can be set independently from the base clock. This function enables setting of the optimum operating frequency for each circuit.

The division rate is set by the basic clock division setting register 0 (DIVR0) and basic clock division setting register 1 (DIVR1). DIVR0 and DIVR1 each have 4 setting bits corresponding to each clock. Register setting value + 1 is the division rate for the base clock. Even if the set division rate is odd, the duty is always 50.

When the register setting value is changed, the new division rate becomes valid from the rising edge of the next clock after setting.

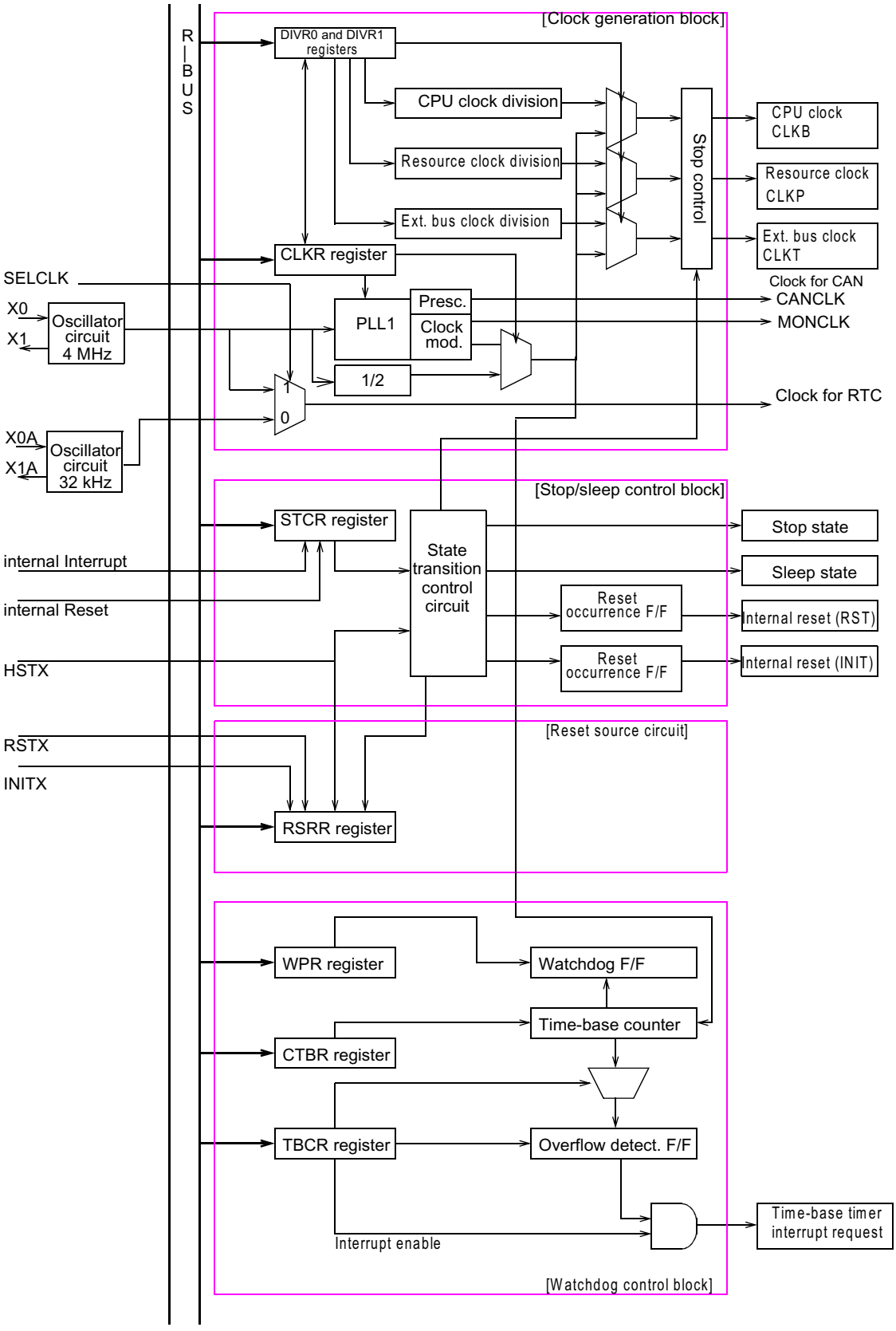
Division rate setting is not re-initialized if an operation initialization reset (RST) occurs, and the division rate setting before RST occurs is held. The division rate setting is initialized only when the setting initialization reset (INIT) occurs. In the initial state, the division rates of all clocks other than the resource clock (CLKP) are 1.

For this reason, always set a division rate before changing the source clock to PLL clock.

The operable upper frequency limit is specified for each clock. Operation is not guaranteed if the operable upper frequency limit is exceeded as a result of the combined source clock selection, PLL multiply rate setting, and division rate setting. Pay attention to this. (Take care not to make a mistake in the change setting sequence for source clock selection.)

5.7 CLOCK GENERATION CONTROL

The figure below shows the block diagram of the clock generation control block. For details of the registers, see section 5.8.



5.8 REGISTERS IN CLOCK GENERATION CONTROL BLOCK

5.8.1 RSRR: Reset Source Register, Watchdog Timer Control Register

bit	15	14	13	12	11	10	9	8
address : 0000 0480H	INIT	HSTB	WDOG	ERST	SRST	-	WT1	WT0
access	R	R	R	R	R	-	R/W	R/W
Initial Value (INITX)	1	0	0	0	0	-	0	0
Initial Value (INIT)	*	*	*	x	x	-	0	0
Initial Value (RST)	x	x	x	*	*	-	0	0
After Boot ROM **	0	0	0	0	0	-	0	0

*: varies with reset factor

x: not initialized

**: After execution of the program in the internal boot ROM the reset source is visible in R4.

RSRR is used to retain the preceding reset factor and to control the watchdog timer start. It is also used to set the watchdog timer cycle. The held reset factor is cleared when this register is read. If a reset occurs several times before this register is read, several reset factor flags are accumulated (set).

The watchdog timer is started when this register is written. Watchdog timer operation continues until RST occurs.

[Bit 15]: INIT (INITialize reset occurred)

This bit indicates whether rest (INIT) was caused by input to the INITX pin.

0	INIT not caused by input to INITX pin
1	INIT caused by input to INITX pin

This bit is set to 0 by reading it.
This bit is read only. Write does not affect other bit values.

[Bit 14]: HSTB (Hardware STandBy reset occurred)

This bit indicates whether reset (INIT) was caused by input to the HSTX pin.

0	INIT not caused by input to HSTX pin
1	INIT caused by input to HSTX pin

This bit is initialized to 0 when reset (INIT) is caused by input to the INITX pin or reading it.
This bit is read only. Write does not affect other bit values.

[Bit 13]: WDOG (WatchDOG reset occurred)

This bit indicates whether reset (INIT) was caused by the watchdog timer.

0	INIT not caused by watchdog timer
1	INIT caused by watchdog timer

This bit is initialized to 0 when reset (INIT) is caused by input to the INITX pin or by reading it.
This bit is read only. Write does not affect other bit values.

[Bit 12]: ERST (External ReSeT occurred)

This indicates whether reset (RST) was caused by input to the RSTX pin.

0	RST not caused by input to RSTX pin
1	RST caused by input to RSTX pin

This bit is initialized to 0 when reset (INIT) is caused by input to the INITX pin or by reading it.
This bit is read only. Write does not affect other bit values

[Bit 11]: SRST (Software ReSeT occurred)

This bit indicates whether reset (RST) was caused by data write (software reset) to the SRST bit of the STCR register.

0	RST not caused by software reset
1	RST caused by software reset

This bit is initialized to 0 when reset (INIT) is caused by input to the INITX pin or immediately after read.
This bit is read only. Write does not affect other bit values.

[Bits 9 and 8]: WT1, WT0 (Watchdog interval Time select)

These bits are used to set the watchdog timer cycle.
Select one of the four watchdog timer cycles from the table below by writing the values to these bits.

WT1	**WT0**	Minimum writing interval to WPR required for control of watchdog reset occurrence	Time between last 5AH writing to WPR and occurrence of watchdog reset
0	0	ϕ X 2^{20} (Initial Value)	ϕ X 2^{20} to ϕ X 2^{21}
0	1	ϕ X 2^{22}	ϕ X 2^{22} to ϕ X 2^{23}
1	0	ϕ X 2^{24}	ϕ X 2^{24} to ϕ X 2^{25}
1	1	ϕ X 2^{26}	ϕ X 2^{26} to ϕ X 2^{27}

(ϕ indicates system base clock cycle)

These bits are initialized to 00 by reset (RST). These bits are read only. Only the first write after reset (RST) is valid; subsequent writes are invalid.

5.8.2 STCR: Standby Control Register

bit	7	6	5	4	3	2	1	0
address : 0000 0481H	STOP	SLEEP	HIZ	SRST	OS1	OS0	OSCD2	OSCD1
access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value (INITX)	0	0	1	1	0	0	1	1
Initial Value (HSTX)*	0	0	1	1	1	1	1	1
Initial Value (INIT)	0	0	1	1	x	x	1	1
Initial Value (RST)	0	0	x	1	x	x	x	x

* : Valid only when this initialization is performed simultaneously with initialization by INITX: others same as INIT.

This register is used to control the device operation mode.
It switches the device to the stop or sleep mode (standby mode) and controls the pins in the stop mode and oscillation stop. It also sets the oscillation stabilization waiting time and issues a Software Reset instruction.

[Bit 7]: STOP (STOP mode)

This bit is used to instruct the device to switch to the stop mode. It has priority over bit 6 (SLEEP bit); when both the bits are 1, the device switches to the stop mode.

0	Device does not switch to stop mode (initial value)
1	Device switches to stop mode

This bit is initialized to 0 when reset (RST) or a stop return factor occurs.

This bit is readable and writable.

See also section 5.12.2 "Stop Mode/RTC Mode" on page 184 about STOP mode.

[Bit 6]: SLEEP (SLEEP mode)

This bit is used to instruct the device to switch to the sleep mode. Bit 7 (STOP bit) has priority over this bit; when both the bits are 1, the device switches to the stop mode.

0	Device does not switch to sleep mode (initial value)
1	Device switches to sleep mode

This bit is initialized to 0 when reset (RST) or a stop return factor occurs.

This bit is readable and writable.

See also section 5.12.1 "Sleep Mode" on page 182 about SLEEP mode.

[Bit 5]: HIZ (HIZ Mode)

This bit is used to control the pin state when the device is in the stop mode.

0	The pin state before the device switches to the stop mode is held.
1	When the device is in the stop mode, pin output enters the high-impedance state (initial value).

This bit is initialized to 1 when reset (INIT) occurs.

This bit is readable and writable.

[Bit 4]: SRST (Software ReSeT)

This bit is used to issue the Software Reset (RST) instruction.

0	Software Reset (RST) instruction issued
1	Software Reset (RST) instruction not issued (initial value)

This bit is initialized to 1 when a reset (RST) occurs.

This bit is readable and writable. 1 is always read.

[Bits 3,2]: OS1,OS0 (Oscillation Stabilization time select)

These bits set the oscillation stabilization wait time after a reset (INIT) or after a return to the stop mode.

The table below shows the combination of values written to these bits and the four associated oscillation stabilization wait times.

OS1	OS0	Oscillation stabilization wait time	4 MHz oscillation
0	0	$\phi \times 2^{16}$ (Initial value)	32 [msec]
0	1	ϕ to 2^{11}	1 [msec]
1	0	ϕ to 2^{16}	32 [msec]
1	1	ϕ to 2^{1}	1 [µsec]

(ϕ is the system-based clock cycle. In this case, it is half the original oscillation.)

These bits are initialized to 00 at reset (INIT) caused by input to the INITX pin.

However, when the reset (INIT) caused by input to the INITX pin and the reset (INIT) caused by input to the HSTX pin are valid at the same time, these bits are initialized to 11.

These bits are readable and writable.

[Bit 1]: OSCD2 (Oscillation disable mode for subclock oscillator)

For settings see chapter 30 "SUBCLOCK" on page 651 about subclock operation.

[Bit 0]: OSCD1 (OSCillation Disable mode for main oscillator)

This bit controls the stop mode oscillation stop for main oscillation input (XIN1).

0	Oscillation does not stop in the stop mode.
1	Oscillation stops in the stop mode (initial value).

This bit is initialized to 1 at reset (INIT).

These bits are readable and writable.

5.8.3 TBCR: Time-base counter control register

bit	15	14	13	12	11	10	9	8
address : 00000482H	TBIF	TBIE	TBC2	TBC1	TBC0	-	SYNCR	SYNCS
Initial value (INIT)	0	0	x	x	x	x	0	0
Initial value ((RST)	0	0	x	x	x	x	x	x
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

After execution of the code in the internal boot ROM the value of this register is "0x03".

This register controls time-base timer interrupts, etc.

It enables the time-base timer interrupt, selects the interrupt interval, and sets the reset operation option function.

[Bit 15]: TBIF (Time-base timer Interrupt Flag)

This bit is the time-base counter interrupt flag.

It indicates that the interval to which the time-base counter was set has passed (interval set by bits 13 to 11 (bits TBC2 to TBC0)).

With an interrupt enabled by bit 14 (TBIE bit) (TBIE=1), when bit 15 is set to 1, a time-base timer interrupt request is generated.

Clearing factor	0 written by instruction
Setting factor	Elapse of set (or specified) interval (falling edge of time-base counter output detected)

This bit is initialized to 0 at reset (RST).

These bits are readable and writable.

However, only 0 can be written. If 1 is written, the bit value does not change. 1 is always read by a read-modify-write instruction.

[Bit 14]: TBIE (Time-base timer Interrupt Enable)

This bit enables output of the time-base timer interrupt request.

It controls the interrupt request output caused by the elapse of the interval to which the time-base counter was set.

When this bit is set to 1, a time-base timer interrupt request is generated when bit 15 (TBIF bit) is set to 1.

0	Disables time-base timer interrupt request output (initial value)
1	Enables time-base timer interrupt request output

This bit is initialized to 1 at reset (RST).

These bits are readable and writable.

[Bits 13 to 11]: TBC2, TBC1, TBC0 (Time-base timer Counting time select)

These bits set the interval of the time-base counter used for the time-base timer.

The table below shows the combination of values written to these bits and their associated 8 intervals.

TBC2	TBC1	TBC0	Timer interval	If clock frequency is set to 48 MHz
0	0	0	$\phi \times 2^{11}$	42.6 [µsec]
0	0	1	$\phi \times 2^{12}$	85.3 [µsec]
0	1	0	$\phi \times 2^{13}$	170.6 [µsec]
0	1	1	$\phi \times 2^{22}$	87.3 [msec]
1	0	0	$\phi \times 2^{23}$	174.7 [msec]
1	0	1	$\phi \times 2^{24}$	349.5 [msec]
1	1	0	$\phi \times 2^{25}$	699 [msec]
1	1	1	$\phi \times 2^{26}$	1.4 [sec]

(ϕ is the system-based clock cycle.)

The initial values of these bits are undefined. Always set a value before enabling an interrupt.

These bits are readable and writable.

[Bit 10]: (reserved bit)

This bit is reserved.

The read value is undefined; write has no effect.

[Bit 09]: SYNCR (SYNChronous Reset enable)

This bit is a synchronous reset enable bit.

When an operation initialization reset (RST) request or a hardware standby request is generated, this bit selects an ordinary reset that immediately causes a reset (RST), or a hardware standby transition, or a synchronous reset that causes an operation initialization reset (RST), or a hardware standby transition after all bus access has stopped.

0	Causes ordinary reset
1	Causes synchronous reset

This bit is initialized to 0 at reset (INIT).

These bits are readable and writable.

[Bit 08]: SYNCS (SYNChronous Standby enable)

This bit enables synchronous standby operation.

When generating a standby request (sleep mode request or stop mode request), this bit selects either the ordinary standby operation in which a transition to the standby state is only caused by a write to the control bit of the STCR register, or the synchronous standby operation in which a transition to the standby state is caused by reading the STCR register after writing to the control bit of the STCR register.

0	Performs ordinary standby operation
1	Performs synchronous standby operation

This bit is initialized to 0 at reset (INIT).

These bits are readable and writable.

5.8.4 CTBR: Time-base counter clear register

bit	7	6	5	4	3	2	1	0
address : 00000483H	D7	D6	D5	D4	D3	D2	D1	D0
Initial value (INIT)	x	x	x	x	x	x	x	x
Initial value(RST)	x	x	x	x	x	x	x	x
	W	W	W	W	W	W	W	W

This register initializes the time-base counter.

When {A5h} and {5Ah} are written consecutively to this register, all bits of the time-base counter are cleared to 0 immediately after a write to {5Ah}. There is no time restriction between the {A5h} write and {5Ah} write.

However, when data other than {5Ah} is written after the {A5h} write, the time-base counter is not cleared even when {5Ah} is written unless {A5h} is written again.

The value read from this register is undefined.

Note: When the time-base counter is cleared using this register, the oscillation stabilization wait interval, watchdog timer cycle, and time-base timer cycle changes temporarily.

5.8.5 CLKR: Clock source control register

bit	15	14	13	12	11	10	9	8
address : 00000484H	PLL2S0	PLL1S2	PLL1S1	PLL1S0	PLL2EN	PLL1EN	CLKS1	CLKS0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value(INIT)	0	0	1	1	0	0	0	0
Initial value(RST)	x	x	x	x	x	x	x	x

If during the execution of the code in the internal boot ROM a valid bootcondition is found this register is set to either "0x00" (4 MHz mode) or to "0x36" (32 kHz mode).

This register selects the clock source used as the base clock for the system and controls the PLL.

This register selects one of three types of clock sources (two are available with this product). Also, it enables the PLL operation and multiply rate for the main system and subsystem.

[Bit 15]: PLL2S0 (PLL2 ratio Select 0)

This bit is not supported by this product.

[Bits 14 to 12]: PLL1S2, PLL1S1, PLL1S0 (PLL1 ratio Select 2-0)

These bits select the multiply rate for PLL1.

One of eight multiply rates is selected for PLL1 (six multiply rates are available for this product).

Rewrite of this bit is disabled while the PLL is selected as the clock source.

PLL1S2	PLL1S1	PLL1S0	PLL1 multiply rate	system-based clock cycle ϕ
0	0	0	reserved	n.a.
0	0	1	reserved	n.a.
0	1	0	(Multiplied by 6)	ϕ =41.7 [nsec] (24 [MHz])
0	1	1	(Multiplied by 4)	ϕ =62.5 [nsec] (16 [MHz])
1	0	0	(Multiplied by 8)	ϕ =31.3 [nsec] (32 [MHz])
1	0	1	(Multiplied by 10)	ϕ = 25.0 [nsec] (40 [MHz])
1	1	0	(Multiplied by 12)	ϕ =20.8 [nsec] (48 [MHz])
1	1	1	(Multiplied by 16)	ϕ =15.6 [nsec] (64 [MHz])

(ϕ is the system-based clock cycle.)

These bits are initialized to 000 at reset (INIT).

These bits are readable and writable.

[Bit 11]: PLL2EN (PLL2 ENable)

For settings see chapter 30 "SUBCLOCK" on page 651 about subclock operation.

[Bit 10]: PLL1EN (PLL1 ENable)

This bit enables the operation of both PLLs.

Rewrite of this bit is disabled while the PLL is selected as the clock source.

Also, while this bit is 0, selection of the PLL as the clock source is disabled.
(The clock source depends on the setting of bits 9, 8 (CLKS1, 0 bits).)

When bit 0 (OSCD1) of STCR is 1, the PLLs stop in the stop mode even when bit 10 is 1. The PLL operation is enabled after returning from the stop mode.

0	Stops PLLs (initial value)
1	Enables PLLs operation

This bit is initialized to 0 at reset (INIT).

These bits are readable and writable.

Set this bit to "1" before changing CLKS1, CLKS0 to select the PLL as clock source.

Set this bit to "1" only after all other clock settings have been done (But see also note following the description of DIVR0).

[Bits 9,8]: CLKS1, CLKS0 (CLocK source Select)

These bits set the clock source used for the FR50 core.

The table below shows the combination of values written to these bits and their associated clock sources.

(Two clock sources are available with this product.)

While bit 9 (CLKS1) is 1, the value of bit 8 (CLKS0) cannot be changed.

[forbidden changes]	[allowed changes]
"00"→"11"	"00"→"01" or "10"
"01"→"10"	"01"→"11" or "00"
"10"→"01" or "11"	"10"→"00"
"11"→"00" or "10"	"11"→"01"

.

CLKS1	CLKS0	Clock source setting
0	0	Oscillation input divided by 2 "X clock" (initial value)
0	1	Oscillation input divided by 2 "X clock"
1	0	PLL clock
1	1	reserved

These bits are initialized to 00 at reset (INIT).

These bits are readable and writable.

Setting these bits to "10" only works if the PLL has been enabled before (PLL1EN = "1").

5.8.6 WPR Watchdog reset generation postponement register

bit	7	6	5	4	3	2	1	0
Address : 00000485H	D7	D6	D5	D4	D3	D2	D1	D0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value (INIT)	x	x	x	x	x	x	x	x
Initial value(RST)	x	x	x	x	x	x	x	x

This register postpones generation of the watchdog reset. When {A5h} and {5Ah} are written consecutively to this register, the detection FF of the watchdog timer is cleared immediately after the {5Ah} write to postpone generation of the watchdog reset. There is no minimum time restriction between the {A5h} write and the {5Ah} write.

However, when data other than {5Ah} is written after the {A5h} write, the detection FF of the watchdog timer is not cleared even when {5Ah} is written, unless {A5h} is written again. Also, the watchdog reset is generated if writing of both data is not finished within the period specified in the table below.

The change shown in the table depends on the state of WT1 (bit 9) and WT0 (bit 8) of the RSRR register.

WT1	WT0	Maximum writing interval to WPR required for control of watchdog reset occurrence	Time between last 5AH writing to WPR and occurrence of watchdog reset
0	0	ϕ X 2^{20} (Initial Value)	ϕ X 2^{20} to ϕ X 2^{21}
0	1	ϕ X 2^{22}	ϕ X 2^{22} to ϕ X 2^{23}
1	0	ϕ X 2^{24}	ϕ X 2^{24} to ϕ X 2^{25}
1	1	ϕ X 2^{26}	ϕ X 2^{26} to ϕ X 2^{27}

(ϕ is the system-based clock cycle. WT1 (bit 9) and WT0 (bit 8) are used to set the cycle of the watchdog timer of RSRR.)

Although the CPU is not operating in the stop state, clearing is performed automatically during DMA transfer in the sleep state. Consequently, when these conditions occur, the watchdog reset is postponed automatically. See also section 5.10.1 "Time-base Counter" on page 176ff.

However, when a hold request for the external bus (BRQ) is accepted, the watchdog reset is not postponed, so, to hold the external bus for a long period, validate the sleep mode and then input the hold request (BRQ).

The value read from this register is undefined.

5.8.7 DIVR0: Base clock division setting register 0

bit	7	6	5	4	3	2	1	0
address : 00000486H	B3	B2	B1	B0	P3	P2	P1	P0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value (INIT)	0	0	0	0	0	0	1	1
Initial value ((RST)	x	x	x	x	x	x	x	x

If during the execution of the code in the internal boot ROM a valid bootcondition is found this register is set to "0x00."

This register controls the base clock division ratio of each internal clock.

It sets the division ratio of the clock for the CPU and internal buses (CLKB), and the clock for resources and peripheral buses (CLKP).

The upper operation frequency limit is defined for each clock. Note that operation is not assured if the upper frequency limit is exceeded by a combination of the source clock selection, PLL multiply rate setting, and division ratio setting. (Take care not to use the procedure to set the change of the source clock selection).

When this register setting is changed, the new division ratio is valid from the next clock rate.

[Bits 15 to 12]: B3, B2, B1, B0 (clkB divide select 3 to 0)

These bits set the division ratio for the CPU clock (CLKB), internal memory, and internal buses.

When these bits are set, one base clock division ratio (clock frequency) is selected from the 16 shown in the table below for the CPU clock and internal buses.

The upper frequency limit is 64 MHz. Do not set a division ratio causing a frequency exceeding the upper frequency limit

Important Note:

When dividing CLKB some precautions have to be taken with regard to the ratio between CLKB and CLKT. If CLKB needs to be divided it is necessary to follow the guidelines below:

- Startup with both clocks undivided.
- Enable the PLL.
- Wait for the PLL to be locked.
- Set clock source to PLL.
- Set division ratio for both clocks at the same time;
  write in parallel to DIVR0 and DIVR1 (half word access).

Always select a ratio such that the frequency of CLKB is higher or equal to the frequency of CLKT and the frequency of CLKB divided by the frequency of CLKT is an integer.

See following table for allowed settings:

Before going into stop mode set the division ratio for CLKB back to 1, deselect the PLL as

clock source, then go into the stop mode.

B3	B2	B1	B0	Clock division ratio	Clock frequency: When base clock is 48 MHz:
0	0	0	0	ϕ	48 [MHz] (Initial value)
0	0	0	1	ϕ X 2(2 division ratio)	24 [MHz]
0	0	1	0	ϕ X 3(3 division ratio)	16 [MHz]
0	0	1	1	ϕ X 4(4 division ratio)	12 [MHz]
0	1	0	0	ϕ X 5(5 division ratio)	9.6 [MHz]
0	1	0	1	ϕ X 6(6 division ratio)	8 [MHz]
0	1	1	0	ϕ X 7(7 division ratio)	6.8 [MHz]
0	1	1	1	ϕ X 8(8 division ratio)	6 [MHz]
...	...	...	...	...	...
1	1	1	1	ϕ X 16(16 division ratio)	3 [MHz]

(ϕ is the system-based clock cycle.)

These bits are initialized to 0000 at reset (INIT).

These bits are readable and writable.

[Bits 11 to 8]: P3, P2, P1, P0 (clkP divide select 3 to 0)

These bits set the division ratio for the peripheral clock (CLKP) and peripheral buses (CLKP).

When these bits are set, one base clock division ratio (clock frequency) is selected from the 16 types shown in the table below for the peripheral circuits and buses.

The upper frequency limit is 32 MHz. Do not set a division ratio causing a frequency that exceeds the upper frequency limit.

P3	P2	P1	P0	Clock division ratio	Clock frequency: When base clock is 48 MHz:
0	0	0	0	ϕ	48 [MHz]
0	0	0	1	ϕ X 2(2 division ratio)	24 [MHz]
0	0	1	0	ϕ X 3(3 division ratio)	16 [MHz]
0	0	1	1	ϕ X 4(4 division ratio)	12 [MHz] (Initial value)
0	1	0	0	ϕ X 5(5 division ratio)	9.6 [MHz]
0	1	0	1	ϕ X 6(6 division ratio)	8 [MHz]
0	1	1	0	ϕ X 7(7 division ratio)	6.8 [MHz]
0	1	1	1	ϕ X 8(8 division ratio)	6 [MHz]
...	...	...	...	...	...
1	1	1	1	ϕ X 16(16 division ratio)	3 [MHz]

(ϕ is the system-based clock cycle.)

These bits are initialized to 0011 at reset (INIT).

These bits are readable and writable.

Table 5.8.7a DIVR0 (B3 to 0), DIVR1 (T3 to 0) register combination

DIVR0 (B3 to 0)	DIVR1 (T3 to 0)	Divide CLKB:CLKT		DIVR0 (B3 to 0)	DIVR1 (T3 to 0)	Divide CLKB:CLKT	
0000	XXXX	1:X	Setting enabled				
0001	0000	2:1	SETTING DISABLED	0101	0000	6:1	SETTING DISABLED
0001	0001	2:2	Setting enabled	0101	0001	6:2	SETTING DISABLED
0001	0010	2:3	SETTING DISABLED	0101	0010	6:3	SETTING DISABLED
0001	0011	2:4	Setting enabled	0101	0011	6:4	SETTING DISABLED
0001	0100	2:5	SETTING DISABLED	0101	0100	6:5	SETTING DISABLED
0001	0101	2:6	Setting enabled	0101	0101	6:6	Setting enabled
0001	0110	2:7	SETTING DISABLED	0101	0110	6:7	SETTING DISABLED
0001	0111	2:8	Setting enabled	0101	0111	6:8	SETTING DISABLED
0001	1111	2:16	Setting enabled	0101	1111	6:16	SETTING DISABLED
0010	0000	3:1	SETTING DISABLED	0110	0000	7:1	SETTING DISABLED
0010	0001	3:2	SETTING DISABLED	0110	0001	7:2	SETTING DISABLED
0010	0010	3:3	Setting enabled	0110	0010	7:3	SETTING DISABLED
0010	0011	3:4	SETTING DISABLED	0110	0011	7:4	SETTING DISABLED
0010	0100	3:5	SETTING DISABLED	0110	0100	7:5	SETTING DISABLED
0010	0101	3:6	Setting enabled	0110	0101	7:6	SETTING DISABLED
0010	0110	3:7	SETTING DISABLED	0110	0110	7:7	Setting enabled
0010	0111	3:8	SETTING DISABLED	0110	0111	7:8	SETTING DISABLED
0010	1111	3:16	SETTING DISABLED	0110	1111	7:16	SETTING DISABLED
0011	0000	4:1	SETTING DISABLED	0111	0000	8:1	SETTING DISABLED

Table 5.8.7a DIVR0 (B3 to 0), DIVR1 (T3 to 0) register combination (Continued)

DIVR0 (B3 to 0)	DIVR1 (T3 to 0)	Divide CLKB:CLKT		DIVR0 (B3 to 0)	DIVR1 (T3 to 0)	Divide CLKB:CLKT	
0011	0001	4:2	SETTING DISABLED	0111	0001	8:2	SETTING DISABLED
0011	0010	4:3	SETTING DISABLED	0111	0010	8:3	SETTING DISABLED
0011	0011	4:4	Setting enabled	0111	0011	8:4	SETTING DISABLED
0011	0100	4:5	SETTING DISABLED	0111	0100	8:5	SETTING DISABLED
0011	0101	4:6	SETTING DISABLED	0111	0101	8:6	SETTING DISABLED
0011	0110	4:7	SETTING DISABLED	0111	0110	8:7	SETTING DISABLED
0011	0111	4:8	Setting enabled	0111	0111	8:8	Setting enabled
0011	1111	4:16	Setting enabled	0111	1111	8:16	Setting enabled
0100	0000	5:1	SETTING DISABLED	1111	0000	16:1	SETTING DISABLED
0100	0001	5:2	SETTING DISABLED	1111	0001	16:2	SETTING DISABLED
0100	0010	5:3	SETTING DISABLED	1111	0010	16:3	SETTING DISABLED
0100	0011	5:4	SETTING DISABLED	1111	0011	16:4	SETTING DISABLED
0100	0100	5:5	Setting enabled	1111	0100	16:5	SETTING DISABLED
0100	0101	5:6	SETTING DISABLED	1111	0101	16:6	SETTING DISABLED
0100	0110	5:7	SETTING DISABLED	1111	0110	16:7	SETTING DISABLED
0100	0111	5:8	SETTING DISABLED	1111	0111	16:8	SETTING DISABLED
0100	1111	5:16	SETTING DISABLED	1111	1111	16:16	Setting enabled

(X:1 or 0)

5.8.8 DIVR1: Base clock division setting register 1

bit	7	6	5	4	3	2	1	0
address : 00000487H	T3	T2	T1	T0	S3	S2	S1	S0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value (INIT)	0	0	0	0	0	0	0	0
Initial value(RST)	x	x	x	x	x	x	x	x

If during the execution of the code in the internal boot ROM a valid bootcondition is found this register is set to either "0x00" (4 MHz mode) or to "0x70" (32 kHz mode).

This register controls the base clock division ratio for each internal clock.

It sets the division ratio for the external extended bus interface (CLKT).

The upper frequency limit is defined for each clock. Note that operation is not assured if the upper frequency limit is exceeded by a combination of the source clock selection, PLL multiply rate setting, and division ratio setting. (Take care not to use the procedure to set the change of the source clock selection).

When this register setting is changed, the new division ratio is valid from the next clock rate.

[Bits 7 to 4]: T3, T2, T1, T0 (clkT divide select 3 to 0)

These bits set the clock division ratio for the external buses (CLKT). When these bits are set, one base clock division ratio (clock frequency) is selected from the 16 shown in the table below for the clock for the external extended bus interface.

The upper frequency limit is 32 MHz. Do not set a division ratio causing about a frequency that exceeds the upper frequency limit.

T3	T2	T1	T0	Clock division ratio	Clock frequency: When base clock is 48 MHz:
0	0	0	0	ϕ	48 [MHz] (Initial value)
0	0	0	1	ϕ X 2(2 division ratio)	24 [MHz]
0	0	1	0	ϕ X 3(3 division ratio)	16 [MHz]
0	0	1	1	ϕ X 4(4 division ratio)	12 [MHz]
0	1	0	0	ϕ X 5(5 division ratio)	9.6 [MHz]
0	1	0	1	ϕ X 6(6 division ratio)	8 [MHz]
0	1	1	0	ϕ X 7(7 division ratio)	6.8 [MHz]
0	1	1	1	ϕ X 8(8 division ratio)	6 [MHz]
...	...	...	...	...	...
1	1	1	1	ϕ X 16(16 division ratio)	3 [MHz]

(ϕ is the system-based clock cycle.)

These bits are initialized to 0000 at reset (INIT).

These bits are readable and writable.

[Bits 3 to 0]: S3, S2, S1, S0 (clkS divide select 3 to 0)

These bits are not used by this product.

5.8.9 CMCR: Clock Control for CAN Modules

The MB91360 family allows to use a dedicated clock for the CAN modules instead of the External Bus Clock (CLKT). CLKT may have to obey other requirements and may not always be fitting for the generation of the proper CAN bit timing. The CMCR register is used to activate the dedicated clock for the CAN modules CANCLK.

This separation of clocks also would allow to scale down the CPU clock and the External Bus Clock but still operate the CAN at e.g. 16 MHz.

For details, please see section 6.2.1 "Control Register (CMCR)" on page 195.

5.8.10 MONCLK pin

The MONCLK pin offers the possibility to monitor internal clock signals at an external pin.

Depending on the device, it is possible to select between PLL clock, oscillator signals and the CAN clock.

Two registers control the MONCLK pin functionality. The control register CMCR, enables and disables the pin. The Select Register CMLT1 determines which signal is observable at the MONCLK pin.

For details, please see sections 6.2.1 "Control Register (CMCR)" on page 195 and 6.2.4 "Random Number Generator and Observer Register (CMLT0..3)" on page 200.

5.9 SWITCHING FROM/TO CLOCK SOURCE PLL

5.9.1 Introduction

During operation the above mentioned devices will often switch from operation at low clock frequencies to operation at higher clock frequencies. This will happen each time when the clock source is changed from oscillator input to PLL e.g. after power-on or reset or after return from low power modes (STOP and RTC modes).

And there is the inverse situation: when going from PLL based operation to low frequency operation. Before going to STOP or RTC mode clock source must be set to oscillator input.

Clock source selection is done by writing the appropriate values into bits 8 and 9 of the CLKR register:

CLKR (clock source register at address 0x484		Clock Source Setting
CLKS1 (bit 9 of CLKR)	CLKS0 (bit 8 of CLKR)	
0	Don't care	Oscillation input divided by 2
1	0	PLL clock
1	1	Reserved setting

It has been observed that for both changes in operating frequency there is a temporary change of the supply voltage generated by the internal voltage regulator. When going to higher frequencies there is a voltage drop, when going to lower frequencies there is an increase of the internal voltage. The size of the drop or increase depends on the used operating frequency.

To avoid operation outside of the device specification or operation failures the items described in the following chapters must be considered. Especially for operation at 48 MHz and above certain counter measures are required to guarantee a fail free operation. For 64 MHz limitations with regard to maximum ambient temperature and minimum operation voltage have to be applied (see table below).

There is no need for counter measures in the case of transitions from/to SLEEP mode. Here only very small voltage drops and voltage increases have been observed.

5.9.2 Reduction of Internal Voltage Change

The recommended way of reducing the internal voltage drop or voltage surge is to add a

■ **larger Capacitance at VCC3C pin**

A capacitance of 10 μF connected in parallel with a capacitance of 10 nF at the VCC3C pin will result in a significant improvement with regard to the internal voltage change. Adding a capacitance only will allow operation up to a PLL frequency of 48 MHz (see also table below).

An additional external capacitor for reducing voltage drops or surges can be omitted if the below given procedures are used.

■ **Smooth start and stop of clocks**

Instead of switching all clocks (CLKB, CLKT and CLKP) directly from a low frequency setting to their target frequencies this can be done in several steps. Also the return to low frequency operation is done in several steps. The sleep mode is utilised to reduce the current consumption – and by this the voltage changes – during the transition between slow and fast clock operation.

Overview of recommended counter measures to avoid malfunction:

PLL frequency setting	Measures to be taken
64 MHz (see also device dependent limitations described in datasheets)	Maximum ambient temperature of 70 degr. C, Minimum operating voltage of 4.75 V and additional capacitance of 10 µF AND Startup/Shutdown routines
48 MHz	Additional capacitance of 10 µF OR Startup/Shutdown routines
32 MHz	Additional capacitance of 10 µF OR Shutdown routine
24 MHz	No special requirements, additional capacitance of 10 µF OR Startup/Shutdown routines recommended
16 MHz	No special requirements, additional capacitance of 10 µF OR Startup/Shutdown routines recommended

5.9.3 Clock Modulator

To avoid calibration errors the clock modulator may not be started during the voltage drop caused by the operation frequency changes described above.

It is recommended to insert a waiting time of 10 µs between that operation which sets the clock frequencies to their target values and the activation of the clock modulator.

About clock modulation, please see chapter 6 "CLOCK MODULATOR" on page 191.

5.9.4 Procedure

For the recommended way to implement a smooth start and stop procedure please see below:

Fujitsu provides a function to switch the PLL. This function basically implements the following command sequence:

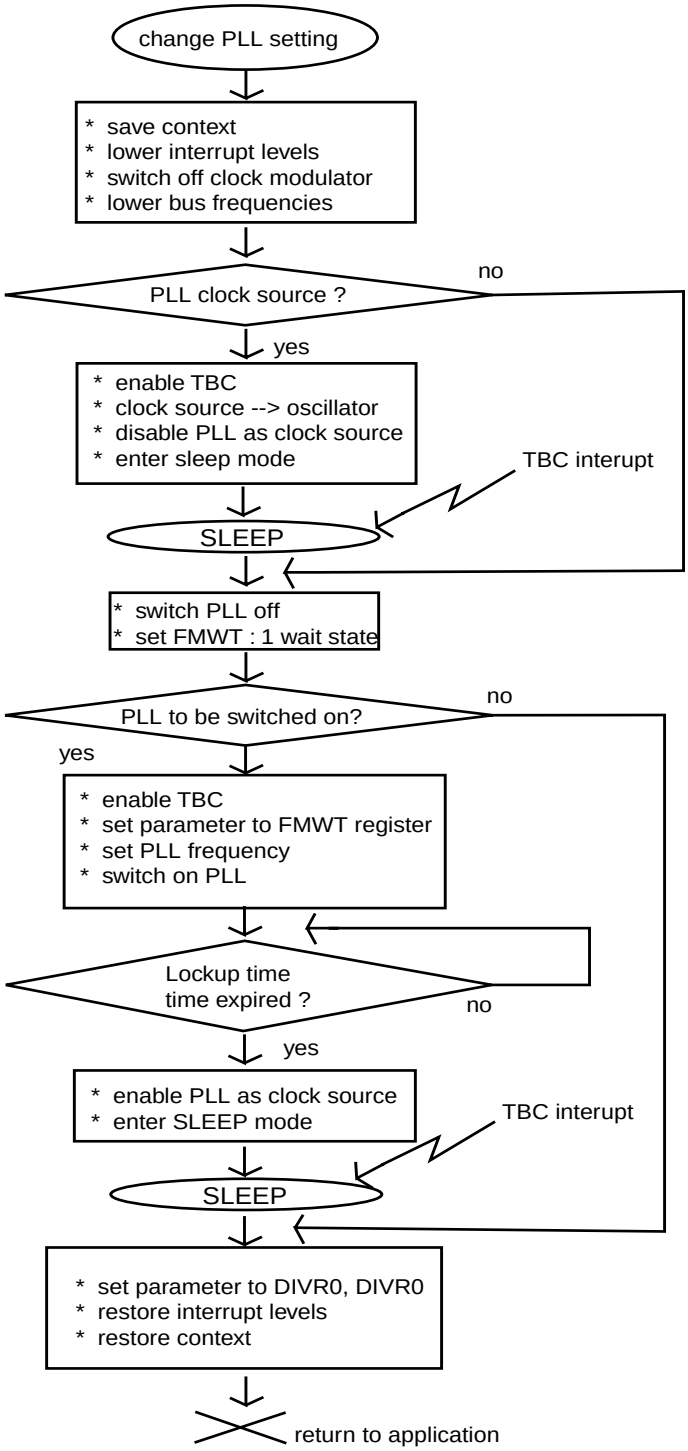
- reduce frequency of external bus and of resource bus
- initialize, enable interrupt and start of time base counter
- switch to sleep mode and immediately after select oscillator as clock source and switch off PLL
- wake up through time base counter interrupt

- disable time base timer interrupt
- initialize, enable interrupt and start of time base counter
- set multiply factor of PLL
- switch to sleep mode and select PLL as clock source
- wake up through time base counter interrupt
- disable time base timer interrupt

- increase frequency of external bus and of resource bus

At the moment when PLL is enabled and selected as clock source the controller is in sleep mode. After expiring of the time base timer it returns to run mode.

Please see also the following flow diagram.



For more details please ask for corresponding application note from Fujitsu.

5.10 CLOCK CONTROL SECTION RESOURCES

The resource function of the clock control section is explained below.

5.10.1 Time-base Counter

The clock control section has a 26-bit time-base counter that operates using system-based clocks.

The time-base counter is used to count the oscillation stabilization wait time as well as for the following uses:

- Watchdog timer

  The watchdog timer is used to detect malfunction of software or hardware by generating a setting initialization reset (INIT) request after a defined time.

- Time-base timer

  The time-base counter output is used to generate interval interrupts.

Those functions are explained below.

■ Watchdog Timer

The watchdog timer detects the program or hardware malfunction by using a time-base counter output. When the watchdog reset generation delay is no longer caused during the set (or specified) interval due to the program or hardware malfunction, the watchdog timer generates the setting initialization reset (INIT) request as a watchdog reset.

- Watchdog timer start and setting cycle

  The watchdog timer is started by a write to the 1st RSRR (reset factor register/watchdog timer control register) after a reset (RST). At this point, the watchdog timer interval is set using bits 09, 08 (bits WT1, WT0). Only the interval set by the first write is valid; all intervals set by succeeding writes are ignored.

- Watchdog reset generation delay

  After the watchdog timer is started, the program must write data to WPR (watchdog reset generation delay register) regularly in the order of {A5h}, {5Ah}. When data is written, the watchdog reset generation flag is initialized.

- Watchdog reset generation

  The watchdog reset generation flag is set by the falling edge of the time-base counter output at the set interval.

  When the flag is still set at detection of the second falling edge, the watchdog timer generates a setting initialization reset (INIT) request as a watchdog reset.

- Watchdog timer stop

  After the watchdog timer is started, it cannot be stopped until the operation initialization reset (RST) generates.

  In the following condition in which RST generates, the watchdog timer stops until it is restarted by the program:

  - Operation initialization reset (RST) state
  - Setting initialization reset (INIT) state

- Oscillation stabilization wait reset (RST) state
- Hardware standby state

- Watchdog timer temporary stop (delay of automatic generation)

  When the CPU program operation is stopped, the watchdog timer initializes the watchdog reset generation flag, delaying generation of the watchdog reset. Stop of the program operation means that the program is in one of the following states:

  - Sleep state
  - Stop state
  - Oscillation stabilization wait run state
  - Transferring DMA to either the I-Bus (instruction bus) or D-Bus (data bus) (MB91FV360GA, MB91F36[5-8]GB, MB91366GA, MB91F369G only)
  - Data access to the I-Bus area (I-RAM or I-Cache in IRAM mode) (MB91FV360GA, MB91F36[5-8]GB, MB91366GA, MB91F369G only)
  - Fetching an instruction from D-Bus RAM (MB91FV360GA, MB91F36[5-8]GB, MB91366GA, MB91F369G only)
  - Breaking the program by using the emulator debugger. (MB91FV360GA only)
  - Breaking the program by using the monitor debugger. (MB91FV360GA only)
  - Period from execution of INTE instruction until execution of RETI instruction (MB91FV360GA only)
  - Breaking at each instruction in step trace trap mode (T-flag of IP register = 1) (MB91FV360GA only)
  - Breaking the program by using the EDSU (MB91F364G, MB91F376G only)

  When the time-base counter is cleared, the watchdog reset generation flag is also initialized, delaying generation of a watchdog reset.

  If the states, listed above, will be issued by the system program or hardware malfunction, a watchdog reset may be not performed. In this case please perform the reset operation (INIT) by using the external INITX pin.

■ Time-base Timer

The time-base timer is an interval interrupt generation timer that uses the output of the time-base counter. The timer is suitable for counting a relatively long duration of up to {base clock x 2^{27}} cycles, such as PLL lock wait time, and subclock oscillation stabilization wait time.

When the falling edge of output of the time-base counter for the set interval is detected, the time-base timer generates a time-base timer interrupt request.

- Time-base timer start and setting interval

  The time-base timer sets an interval by using bits 13-11 (bits TBC2, TBC1, TBC0) of TBCR (time-base counter control register).

  The falling edge of output of the time-base counter for the set interval is always detected, so after setting an interval, clear bit 15 (TBIF bit), and then set bit 14 (TBIE bit) to 1 to enable the interrupt request output.

  When changing the interval, set bit 14 (TBIE bit) to 0 in advance to disable the interrupt request output.

  The time-base counter is always counting and is affected by these settings. To obtain an

accurate interval interrupt time, clear the time-base counter before enabling an interrupt. Otherwise, an interrupt request might occur immediately after an interrupt is enabled.

- Clearing time-base counter by program

  When data is written to CTBR, in order of {A5h} and {5Ah}, all bits of time-base counter are cleared to 0 immediately after {5Ah} writing. There are no limits of time between {A5h} and {5Ah} writing.

  However, if the data other than {5Ah} is written after {A5h} writing, {5Ah} writing does not make a clearing if {A5h} is written again.

  When the time-base counter is cleared, the watchdog reset generation flag is initialized, delaying generation of the watchdog reset.

- Clearing time-base counter at device state transition

  All bits of the time-base counter are cleared simultaneously to 0 at transition of the following device states:

  - Stop state
  - Setting initialization reset (INIT) state
  - Hardware standby state

  In particular, in the stop state, the time-base counter is used for counting the oscillation stabilization wait time, so the time-base timer interval interrupt might generate unintentionally. Therefore, before setting the stop mode, disable the time-base timer interrupt, and not use the time-base timer.

  In the other states, the operation initialization reset (RST) is generated, so the time-base timer interrupts are disabled automatically.

5.11 DEVICE STATE CONTROL

The states and control of the FR50 series of devices are explained below.

5.11.1 Device States and State Transition

The FR50 series of devices have the following operation states:

■ **Run State (Ordinary Operation)**

In the run state, the program runs; all the internal clocks are supplied and all the circuits operate.

However, the bus clock for the 16-bit peripheral bus stops when the peripheral bus is not accessed.

Transition requests for each state are accepted, but in the synchronous reset mode, the state transition response to transition requests is partly different from that in the ordinary reset mode. For details, see section 2.9.6 "Reset Mode" on page 123.

■ **Sleep State**

In the sleep state, the program stops. Program operation causes state transition.

In the sleep state, only CPU program execution is stopped and resources remain operational. The instruction cache is stopped, and various internal memory and internal and external buses are stopped unless requested by the DMA controller.

When a valid interrupt request is generated, the sleep state is cleared and a transition to the run state (ordinary operation) occurs.

When a setting initialization reset (INIT) request is generated, a transition to the setting initialization reset (INIT) state occurs.

When an operation initialization reset (RST) request is generated, a transition to the operation initialization reset (RST) state occurs.

When a hardware standby request is generated, a transition to the hardware standby state occurs.

■ **Stop State**

In the stop state, devices are stopped. Program operation causes state transition.

In the stop state, all internal circuits are stopped, all the internal clocks are stopped, and the oscillation circuit and PLL can be stopped by appropriate setting.

Also, by performing the appropriate setting, external pins (some external pins are excluded) can be put in the high-impedance state.

When a particular valid interrupt request (not used clock) is generated, a transition to the oscillation stabilization wait run state occurs.

When a setting initialization reset (INIT) request is generated, a transition to the setting initialization reset (INIT) state occurs.

When an operation initialization reset (RST) request is generated, a transition to the oscillation stabilization wait reset (RST) state occurs.

When a hardware standby request is generated, a transition to the hardware standby state occurs.

■ Hardware Standby State

In the hardware standby state, devices are stopped. When a Low level (hardware standby request) is input to the external HSTX pin the transition into hardware standby state occurs.

In the hardware standby state, all internal circuits are stopped, all internal clocks are stopped, and the oscillation circuits and PLLs are also stopped.

A setting initialization reset (INIT) is supplied to the internal circuits.

External pins (some external pins are excluded) are uniformly put in the high-impedance state.

When a high level is input to the external HSTX pin or when a low level is input to the external INITX pin, a transition to the setting initialization reset (INIT) state occurs.

■ Oscillation Stabilization Wait Run State

In the oscillation stabilization wait run state, devices are stopped. A transition to this state occurs after a return from the stop state.

All the internal circuits are stopped except the clock generation control section (time-base counter and device state control section). All the internal clocks are stopped, but the oscillation circuit and the operation-enabled PLL operate.

High-impedance control of external pins in the stop state, etc., is cleared.

When the set (or specified) oscillation stabilization wait time has elapsed, a transition to the run state (ordinary operation) occurs.

When a setting initialization reset (INIT) request is generated, a transition to the setting initialization reset (INIT) state occurs.

When an operation initialization reset (RST) request is generated, a transition to the oscillation stabilization wait reset (RST) state occurs.

When a hardware standby request is generated, a transition to the hardware standby state occurs.

■ Oscillation Stabilization Wait Reset (RST) State

In the oscillation stabilization wait reset (RST) state, devices are stopped. A transition to this state occurs after a return from the stop state or the setting initialization reset (INIT) state.

All the internal circuits are stopped except the clock generation control section (time-base counter and device state control section). All the internal clocks are stopped, but the oscillation circuit and the operation-enabled PLL operate.

High-impedance control of external pins in the stop state, etc., is cleared.

An operation initialization reset (RST) is output to internal circuits.

When the set oscillation stabilization wait time has elapsed, a transition to the oscillation stabilization wait reset (RST) state occurs.

When a setting initialization reset (INIT) request is generated, a transition to the setting initialization reset (INIT) state occurs.

When a hardware standby request is generated, a transition to the hardware standby state occurs.

■ **Operation Initialization Reset (RST) State**

In the operation initialization reset (RST) state, the program is in the initialization state. A transition to this state occurs when an operation initialization reset (RST) request is accepted, or the oscillation stabilization wait reset (RST) state is terminated.

The CPU program execution stops and the program counter is initialized. Most resources are initialized. All the internal clocks and oscillation circuits and the operation-enabled PLL operate.

An operation initialization reset (RST) is output to internal circuits.

When an operation initialization reset (RST) request is released, a transition to the run state (ordinary operation) occurs, executing the operation initialization reset sequence. After a return from the setting initialization reset (INIT) state, the setting initialization reset sequence is executed.

When a setting initialization reset (INIT) request is generated, a transition to the setting initialization reset (INIT) state occurs.

When a hardware standby request is generated, a transition to the hardware standby state occurs.

■ **Setting Initialization Reset (INIT) State**

In the setting initialization reset (INIT) state, all the settings are initialized. A transition to this state occurs when a setting initialization reset (INIT) request is accepted or the hardware standby state is terminated.

The CPU program execution stops and the program counter is initialized. All the resources are initialized. The oscillation circuit operates, but the PLL stops. All the internal clocks stop while a Low level is input to the external INITX pin, but they operate during other periods.

A setting initialization reset (INIT) and an operation initialization reset (RST) are output to the internal circuits.

When the setting initialization reset (INIT) request is released, this state is cleared and a transition to the oscillation stabilization wait reset (RST) state occurs. After this, a transition to the operation initialization reset (RST) state occurs, executing the setting initialization reset sequence.

5.11.2 Priority Of Each State Transition Request

In any state, each state transition request follows the priority below. However, some requests are only generated in a particular state, so they are only valid in that state.

[Highest] Setting initialization reset (INIT) request

Hardware standby request

Termination of oscillation stabilization wait time
(Only the oscillation stabilization wait reset state

and the oscillation stabilization wait run state occur,)

Operation initialization reset (RST) request

Valid interrupt request (Only the run, sleep, and stop states occur.)

Stop mode request (write to register) (Only the run state occurs.)

[Lowest] Sleep mode request (write to register) (Only the run state occurs.)

5.12 LOW POWER CONSUMPTION MODES

Each low power consumption mode and its use for the FR50 series of devices are explained.

The low power consumption modes for the FR50 series of devices are shown below.

- Sleep mode

  A write to the register changes the device to the sleep state.

- Stop mode

  A write to the register changes the device to the stop state.

- Hardware standby mode

  Input of a Low level to the external HSTX pin changes the device to the hardware standby state.

Each of the above modes is explained below.

5.12.1 Sleep Mode

When 1 is written to bit 6 (SLEEP bit) of STCR (standby control register), the sleep mode is enabled. The sleep state remains valid until a factor for returning from the sleep state occurs.

When 1 is written to both bit 7 (STOP bit) and bit 6 (SLEEP bit) of STCR (standby control register), bit 7 (STOP bit) takes precedence over bit 6 (SLEEP bit), causing a transition to the stop state.

For the sleep state, see section 5.11.1 "Device States and State Transition" on page 179.

■ **Circuits stopped in sleep state**

- CPU program execution
- Instruction cache
- Data cache
- Bit search module (This operates when a DMA transfer occurs.)
- Various internal memory (These operate when a DMA transfer occurs.)
- Internal/external bus (This operates when a DMA transfer occurs.)

■ **Circuits operating in sleep state**

- Oscillation circuit
- Operation-enabled PLLs
- Clock generation control section
- Interrupt controller
- Resources
- DMA controller
- DSU

■ **Factors for returning from sleep state**

- Valid interrupt request

  When an interrupt request higher than the level set by the CPU ILM is generated, the sleep mode is cleared, causing a transition to the run state (ordinary operation).

  When an interrupt request equal to or lower than the level set by the CPU ILM is generated, the sleep mode is not cleared.

- Setting initialization reset (INIT) request

  When a setting initialization reset (INIT) request is generated, a transition to the setting initialization reset (INIT) state occurs.

- Hardware standby request

  When a hardware standby request is generated, a transition to the hardware standby state occurs.

- Operation initialization reset (RST) request

  When an operation initialization reset (RST) request is generated, a transition to the operation initialization reset (RST) state occurs.

Note: For the priority of each factor, see section 5.11.2 "Priority Of Each State Transition Request" on page 181.

■ **Ordinary standby operation and synchronous standby operation**

When bit 8 (SYNCS bit) of TBCR (time-base counter control register) is set to 1, the synchronous standby operation is enabled. In this state, a write to the SLEEP bit does not cause a transition to the sleep state. A write to the SLEEP bit followed by a read from the STCR register causes a transition to the sleep state.

When the SYNCS bit is set to 0, the ordinary standby operation is enabled. In this state, a write to the SLEEP bit causes a transition to the sleep state.

With the ordinary standby operation enabled, when the division rate of the peripheral clock (CLKP) is a large value compared to the CPU clock (CLKB), many instructions are executed by the time a write is actually performed to the SLEEP bit.

So, after a write instruction to the SLEEP bit, if a NOP instruction is not placed that is equal to or higher than the {5 + (division rate of CPU clock)/(division rate of peripheral clock)} instructions, the succeeding instructions are executed before changing to the sleep state.

With the synchronous standby operation enabled, transition to the sleep state does not occur until a read from the STCR register is completed after a write to the SLEEP bit is actually performed. This is because the CPU uses the bus until the value read from the STCR register is stored by the CPU.

So, irrespective of the setting of the division rate of the CPU clock (CLKB) and peripheral clock (CLKP), when only two NOP instructions are placed after a write instruction to the SLEEP bit and a read instruction to the STCR register, execution of the succeeding instructions can be prevented before changing to the sleep state.

5.12.2 Stop Mode/RTC Mode

When 1 is written to bit 7 (STOP bit) of STCR (standby control register), the stop mode is enabled. The stop state remains valid until a factor for returning from the stop state occurs.

When 1 is written to both bit 7 (STOP bit) and bit 6 (SLEEP bit) of STCR (standby control register), bit 7 (STOP bit) takes precedence over bit 6 (SLEEP bit), causing a transition to the stop state.

For the stop state, see section 5.11.1 "Device States and State Transition" on page 179.

■ **Circuits stopped in stop state**

- Oscillation circuits set to be stopped

  When bit 0 (OSCD1 bit) of STCR (standby control register) is set to 1, the oscillation circuit for main clocks in the stop state enters the stop state.

- Non-enabled PLLs or PLLs connected to oscillation circuit set to be stopped

  When bit 0 (OSCD1 bit) of STCR (standby control register) is set to 1, even if bit 10 (PLL1EN bit) of CLKR (clock source control register) is set to 1, the PLLs enter the stop state.

- All internal circuits except following:

■ **Non-stopped circuits in stop state**

- Oscillation circuits set to be not-stopped

  When bit 1 (OSCD1 bit) of STCR (standby control register) is set to 0, the oscillation circuit does not stop in the stop state. This mode is also called RTC mode.

- Enabled PLLs or PLLs connected to oscillation circuit set to be non-stopped

  When bit 0 (OSCD1 bit) of STCR (standby control register) is set to 0, even if bit 10 (PLL1EN bit) of CLKR (clock source control register) is set to 1, the PLL enters the stop state.

■ **High impedance control of pins in stop state**

When bit 5 (HIZ bit) of STCR (standby control register) is set to 1, output from a pin in the stop state enters the high impedance state. For pins subject to high-impedance control, see Appendix C.

When bit 5 (HIZ bit) of STCR (standby control register) is set to 0, the pin output in the stop state holds the value before it changed to the stop state. For details, see Appendix C.

■ **Factors for returning from stop state**

- Occurrence of valid interrupt request requiring no clocks

  Only external interrupt input pins (INT0 - INT7) with request type set to 'H' level are valid.

  When an interrupt request higher than the level set by the CPU ILM is generated, the stop mode is cleared, causing a transition to the run state (ordinary operation).

  When an interrupt request equal to or lower than the level set by the CPU ILM is generated, the stop mode is not cleared.

- Setting initialization reset (INIT) request

  When a setting initialization reset (INIT) request is generated, a transition to the setting initialization reset (INIT) state occurs.

- Hardware standby request

  When a hardware standby request is generated, a transition to the hardware standby state

occurs.

- Operation initialization reset (RST) request

  When an operation initialization reset (RST) request is generated, a transition to the operation initialization reset (RST) state occurs.

Note: For the priority of each factor, see section 5.11.2 "Priority Of Each State Transition Request" on page 181

■ **Clock source selection in stop mode**

Select the main clock divided by 2 (X clock) as the source clock before setting the stop mode. For details, see section 5.2 "BASE CLOCK GENERATION" on page 149.

The division rate setting restrictions are the same as for ordinary operation.

■ **Ordinary standby operation and synchronous standby operation**

When bit 8 (SYNCS bit) of TBCR (time-base counter control register) is set to 1, the synchronous standby operation is enabled. In this state, a write to the STOP bit does not cause a transition to the sleep state. A write to the STOP bit followed by a read from the STCR register causes a transition to the stop state.

When the SYNCS bit is set to 0, the ordinary standby operation is enabled. In this state, a write to the STOP bit causes a transition to the stop state.

With the ordinary standby operation enabled, when the division rate of the peripheral clock (CLKP) is a large value compared to the CPU clock (CLKB), many instructions are executed by the time a write is actually performed to the STOP bit.

So, after a write instruction to the STOP bit, if not more NOP instructions are placed than the {5 + (division rate of CPU clock)/(division rate of peripheral clock)} instructions, the succeeding instructions are executed before changing to the stop state.

With the synchronous standby operation enabled, transition to the stop state does not occur until a read from the STCR register is completed after a write to the STOP bit is actually performed. This is because the CPU uses the bus until the value read from the STCR register is stored by the CPU.

So, irrespective of the setting of the division rate of the CPU clock (CLKB) and peripheral clock (CLKP), when only two NOP instructions are placed after a write instruction to the STOP bit and a read instruction to the STCR register, execution of the succeeding instructions can be prevented before changing to the stop state.

5.12.3 Hardware Standby Mode

When a Low level is input to the external HSTX pin, a hardware standby request is generated, causing a transition to the hardware standby state. After this, the hardware standby state is held while a Low level is input.

For the hardware standby state, see section 5.11.1 "Device States and State Transition" on page 179.

■ **Circuits stopped in hardware standby state**

- All oscillation circuits
- All PLLs
- All internal circuits

■ **Circuits not stopped in hardware standby state**

- None

■ **High-impedance control of pins in hardware standby state**

In the hardware standby state, output from pins enters the high impedance state. For pins subject to highimpedance control, see Appendix C.

■ **Hardware standby state return factors**

- Input of High level to external HSTX pin

  When the hardware standby request is released, a transition occurs to the setting initialization reset (INIT) state.

- Setting initialization reset (INIT) due to input of Low level to external INITX pin

  When a setting initialization reset (INIT) request is generated by the external INITX pin, a transition to the setting initialization reset (INIT) state occurs. Other setting initialization reset (INIT) requests are not generated in the hardware standby state.

Note: For the priority of each factor, see section 5.11.2 "Priority Of Each State Transition Request" on page 181

■ **Ordinary reset and synchronous reset**

When bit 9 (SYNCR bit) of TBCR (time-base counter control register) is set to 1, the synchronous reset is enabled. In this state, no transition to the hardware standby state occurs while the internal bus is being accessed even if a hardware standby request is accepted.

When the SYNCR bit is set to 0, the ordinary reset is enabled. In this state, when a hardware standby request is accepted, a transition to the hardware standby state occurs irrespective of the operation state of the internal bus access.

For details, see section 2.9.6 "Reset Mode" on page 123.

■ Operation in Standby Modes

Mode	Transition condition	Oscillator	PLL	Internal clock		Resources	Pins	How to wakeup
				CPU/ internal bus	DMA/ peripheral			
Run	—	○	○	○	○	○	Operating	—
Sleep	SLEEP= “1” of STCR	○	○	×	○	○	Operating	Interrupt request RST, INIT
RTC	STOP=”1”, OSCD1=”0” of STCR	○	!!!	×	×	×	*	Interrupt request RST, INIT
Stop	STOP= “1”, OSCD1=”1” of STCR	×	×	×	×	×	*	Interrupt request RST, INIT
Hardware standby	HSTX= “0”	×	×	×	×	×	Hi-Z	HSTX= “1”
Reset (RST)	RST	○	○	Reset (RST) is applied to registers			Hi-Z	
Reset (INIT)	INIT	○	×	Reset (INIT) is applied to registers			Hi-Z	

Note: ○: Operating, ×: Halted, Hi-Z: High impedance

RST: External reset pin (RSTX) = “0”

Software reset (SRST bit of STCR register = “0”)

Power-down reset

INIT: External reset pin (INITX) = “0”

Return from Hardware Standby

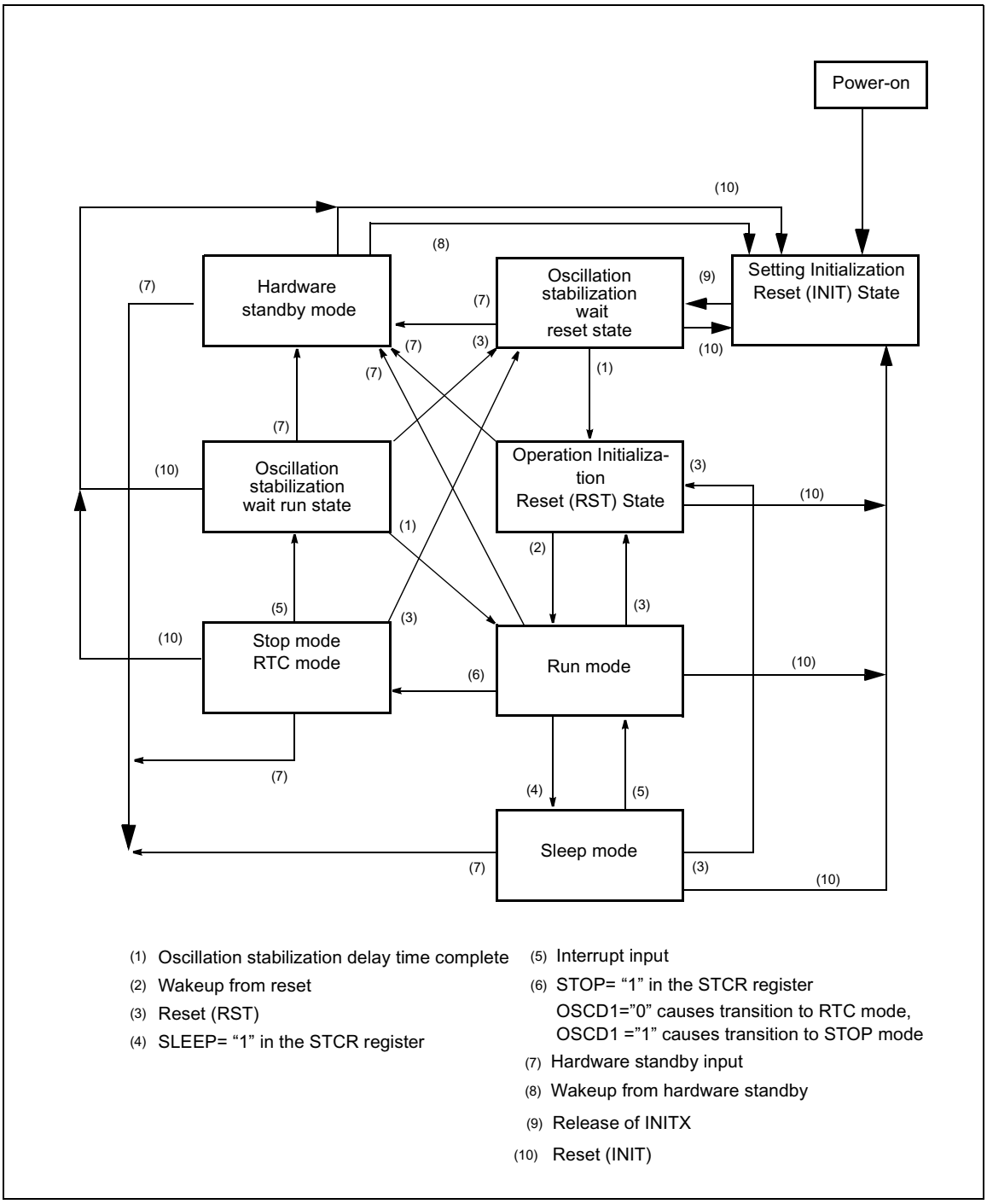
Watchdog reset

*: Hold previous states when HIZ of STCR is “0”. Go to Hi-Z when HIZ is “1”.

!!!: PLLs must be switched off before entering RTC state

■ State Transition Diagram for Standby Modes

Figure 5.12 State Transition Diagram for Standby Modes



5.12.4 Transition from 4MHz RTC Mode to RUN Mode

MB91360 devices contain two voltage regulators – one which is used for STOP and RTC mode operation, the other which is used for RUN mode operation.

When going from STOP/RTC mode to RUN mode the “RUN mode regulator” is switched on and after a delay of typically 12 µs the other regulator is switched off.

If there is a transition from STOP mode to RUN mode or from 32 kHz RTC mode to RUN mode the 4 MHz oscillator must be started and an appropriate oscillation stabilisation time must be applied. In these cases the internal clocks will start significantly after the RUN mode regulator has been switched on.

This can be different in case of a transition from 4 MHz RTC mode to RUN mode. In this case the 4 MHz oscillator is already running:

If the transition from 4 MHz RTC mode to RUN mode is done

- at a supply voltage below 4.75 V,
- with a capacitance of less than 4.7µF connected to the VCC3C pin,
- with the minimum oscillation stabilisation time of 1 µs set,

then the device may show wrong behaviour. This setting means that the internal clocks start running before the “RUN mode regulator” is properly switched on. A voltage drop can be seen at the VCC3C pin.

This problem can be avoided by making sure that not all the conditions given above apply:

- Use supply voltages equal or above 4.75 V

  or

- Connect a capacitance of 4.7µF or more to the VCC3C pin. A larger capacitance is also recommended to avoid problems when switching on the PLL in RUN mode

  or

- Select a larger oscillation stabilisation time e.g. by setting the OS1, OS0 bits in the STCR register to “01” which defines an oscillation stabilisation time of 1 ms.

CHAPTER 6 CLOCK MODULATOR

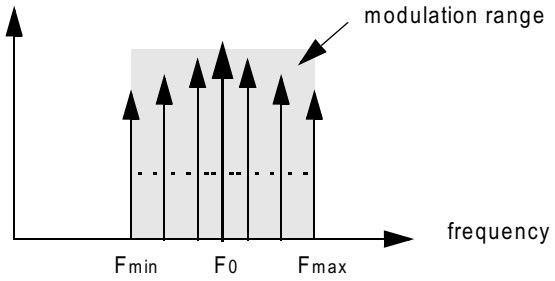
This chapter provides an overview of the Clock Modulator and its features. It describes the register structure and operation of the Clock Modulator.

6.1 OVERVIEW

The clock modulator is intended for the reduction of electromagnetic interference (EMI), by spreading the spectrum of the clock signal over a wide range of frequencies.

The module is fed with an unmodulated reference clock with frequency F_0, provided by the PLL circuit. This reference clock is frequency modulated, controlled by a random signal. I.e. the width of every clock pulse in the modulated clock is randomly determined.

Figure 6.1 frequency spectrum of the modulated clock (fundamentals only)



■ **modulation degree and frequency resolution**

Maximum and minimum frequencies (F_{max} and F_{min}) of the modulated clock are well defined by the modulation degree parameter. Furthermore the resolution of the modulation range is selectable in 3 steps from low to high. Higher resolution implies a finer granularity of discrete frequencies in the spectrum of the modulated clock but less possible modulation degrees.

In general the highest possible frequency resolution combined with the highest possible modulation degree results in the highest EMI reduction. But for some cases a lower modulation degree or a higher modulation degree at a lower resolution may result in a better EMI behaviour. Please refer to the table of possible settings in the appendix.

The mean frequency of the modulated clock is equal to the reference clock frequency F_0.

■ **calibration unit**

The modulator can adapt to input frequencies between 16MHz - 48 MHz. For this purpose the module contains a calibration unit. The calibration process can be triggered by software or automatically by hardware (recommended).

The calibration process must be started by software when the modulator is enabled. During normal operation, the automatic calibration feature cares for a calibrated modulator. The frequency of automatic calibration events is programmable via the automatic calibration reload timer value.

■ **CAN prescaler**

Because the CAN module requires an unmodulated operation clock, the modulator provides an unmodulated clock besides the modulated one.

This CAN-clock is scalable via the CAN clock prescaler.

■ **Observer**

Some internal clock signals can be observed at the external MONCLK pin. The modulator

module provides a multiplexer to select the desired signal.

For further information, please contact FUJITSU.

6.2 REGISTER DESCRIPTION

This section lists the clock modulator registers and describes the function of each register in detail.

Address	Register				Block
	+0	+1	+2	+3	
000164H	CMCR [R/W] 11111111 0000000		CMPR [R/W] ----1001 1---0001		Clock Modulation
000168H	CMLS0 [R/W] 01110111 1111111		CMLS1 [R/W] 01110111 1111111		
00016CH	CMLS2 [R/W] 01110111 1111111		CMLS3 [R/W] 01110111 1111111		
000170H	CMLT0 [R/W] -----100 00000010		CMLT1 [R/W] 11110100 00000010		
000174H	CMLT2 [R/W] -----100 00000010		CMLT3 [R/W] -----100 00000010		
000178H	CMAC [R/W] 11111111 1111111		CMTS [R] --000001 01111111		

6.2.1 Control Register (CMCR)

The Control Register (CMCR) has the following functions:

- **Modulator enable/disable**
- **Enable MONCLK pin**
- **Select output-clock (for testing only)**
- **Start calibration**
- **Initialize random number generator**
- **Rbus registers enable/disable**
- **CAN prescaler enable/disable**
- **CAN prescale value**

Precaution: Write the prescale value PRE (CMCRH-register) only, when the prescaler is disabled.

Control Register low byte, Address: 000165H

Bit no. ⇐	7	6	5	4	3	2	1	0	
	PRES	CDSBLE	IRNG	CAL	MSEL	MTST	SCLK	MSRT	CMCRL
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

■ BIT[0]: MSRT - Modulator Start

This bit enables / disables the modulator.

0	Modulator off (default)
1	Modulator on

Before the modulator can be started, all configuration registers (CMLT, CMLS, etc.) must be set.

When the modulator is started, the CAL-bit and the IRNG-bit must be set, in order to start a calibration process and initialize the random number generator.

In order to set the output clock to modulated clock, the SCLK-bit must be set to 1. If SCLK remains 0, the output clock is unmodulated, even if the modulator is switched on.

Caution: It is not allowed to switch off the modulator, when a software triggered (CAL-bit) calibration process is active.

The end of calibration which was triggered by software is indicated by the cleared CAL bit. After the software triggered calibration process is finished, the modulator can be switched off at any time.

Caution: The clock modulator must be switched off every time before
a) the PLL frequency will be changed
b) the PLL will be switched off (e.g. in power save modes)
The modulator can be switched on after the PLL delivers a stable clock signal (after PLL lock time).

■ **BIT[1]: SCLK - Select Clock**

This bit selects the output-clock.

0	Output = unmodulated clock (default)
1	Output = modulated clock

This bit should be used only for **test-reasons**, i.e. to run the modulator while the core is feeded with unmodulated clock.

A change of this bit, while the modulator is running, has no effect until the next calibration is triggered by soft- or by hardware.

For normal operation, SCLK should be set at the same time as the MSRT-bit for modulator-start or it should be already set when the modulator is started.

When the modulator is switched off, the output-clock changes automatically to unmodulated clock, independent of this bit.

■ **BIT[2]: MTST - Enable Testclk-pin MONCLK**

0	MONCLK pin is disabled (default)
1	MONCLK pin is enabled

The test-signal at the TESTCLK-pin can be selected with the MSEL-bit and the OBS-bits in CMLT1-register.

■ **BIT[3]: MSEL - Select Testclock**

0	Signal at testpin is modulated clock (default)
1	Signal at testpin is selected with OBS-bits in CMLT1-register

Note: The MSEL-signal is not synchronized. Therefore glitches can occur at the TESTCLK-pin during the switching.

■ **BIT[4]: CAL - Calibration trigger**

This bit starts a calibration (software calibration request) of the modulator and indicates the end of calibration

0	The delay-chain is calibrated (default)
1	Start calibration

CAL can be set only by software and is cleared by hardware at the end of calibration.

When the modulator is started (MSRT-bit), CAL must be set, in order to start a calibration process at the beginning of modulation.

Writing of a 0 by software is ignored. The read-cycle of RMW-instructions (read modify write) returns 0.

When the Rbus registers are locked by the CDSBLE-bit, SW calibration requests can not be executed.

Caution: It is not allowed to switch off the modulator, when a software triggered (CAL-bit) calibration process is active.

The CAL-bit indicates an active calibration process only if this process was triggered by

software. Calibration processes which are triggered automatically by hardware are not indicated by a set CAL bit.

During calibration, the output is set to the unmodulated clock automatically.

The duration of the calibration is 44xT_0. (T_0=period of input clock, F_0=frequency of input clock)

F_0 [MHz]	T_0 [ns]	duration of calibration
16	62.5	2.75 µs
32	31.25	1.375 µs
48	20.83	0.916 µs

■ **BIT[5]: IRNG - Initialize random number generator**

IRNG initializes the random number generator.

0	random number generator is initialized
1	random number generator will be initialized

This bit can be set only by software and is cleared by hardware after the initialization is finished.

When the modulator is started (MSRT-bit), IRNG must be set, in order to initialize the random number generator.

Writing of a 0 by software is ignored. The read-cycle of RMW-instructions (read modify write) returns 0.

■ **BIT[6]: CDSBLE - Rbus register enable / disable**

0	All Rbus registers are enabled (default)
1	All Rbus registers disabled, except for CDSBLE-bit in CMCR-register

The CDSBLE bit is a safety feature to avoid unintended register write access.

When the registers are disabled, only read access is possible, i.e. the registers are write protected.

Note: When the Rbus registers are locked, SW calibration requests can not be executed.

Caution: It is necessary to perform a software calibration request every time the modulator is started (see CAL-bit description). If the CDSBLE bit is set at the same time as the modulator is started and calibration is requested, the hardware can not reset the CAL-bit to 0 after calibration is finished -> CAL remains 1.

As a side effect, automatic (hardware) calibration can not be triggered and the modulator is not calibrated.

Therefore it is necessary to wait until the first calibration is finished before the CDSBLE bit can be set. The end of calibration which was triggered by software is indicated by the cleared CAL bit. After the software triggered calibration is finished, the register lock can be set at any time, even if automatic calibration is enabled.

■ BIT[7]: PRES - CAN Prescaler enable / disable

0	CAN prescaler disabled (default)
1	CAN prescaler enabled

The scaling factor for the CAN prescaler is set with the PRE-bits in CMCRH.

The CANCLK-output has the polarity '1' , if the prescaler is disabled.

Control Register high byte	15	14	13	12	11	10	9	8	⇐ Bit no.
Address: 000164H	PRE7	PRE6	PRE5	PRE4	PRE3	PRE2	PRE1	PRE0	CMCRH
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	

■ BIT[15:8]: PRE - CAN Prescaler Value

Value for CAN prescaler. The clock for the CAN-module is determined with the following equation:

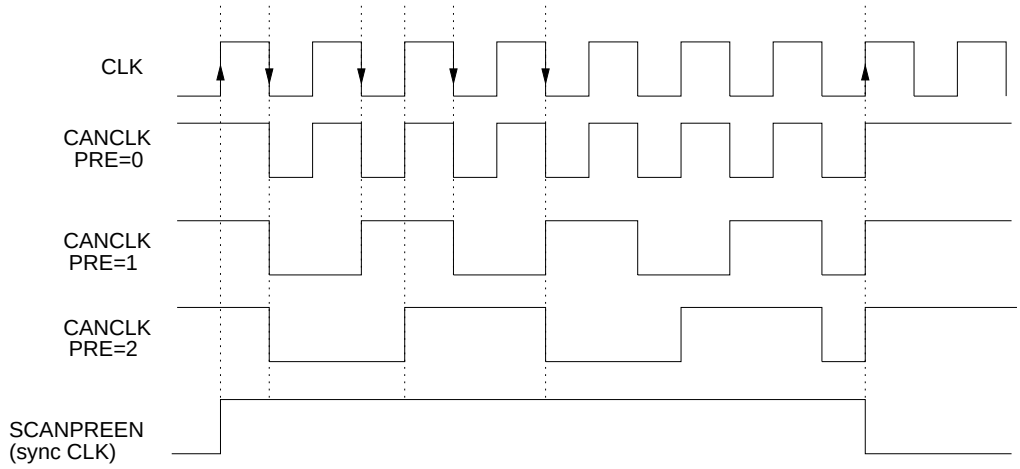
$$\left(CANCLK = \frac{F_0}{PRE + 1}\right)$$

Note: The CANCLK will be divided by 2 in the CAN modules!

F_0 = unmodulated input clock (output of PLL1 or X0 oscillator).
Default value: PRE=255, CANCLK=F_0/256

Precaution: Write the prescale value PRE (CMCRH-register) only, when the prescaler is disabled.

The CANCLK is enabled with the PRES - bit in CMCRL.
The CANCLK-output has the polarity '1' , if the prescaler is disabled.



falling edges of CANCLK occur only at falling edges of CLK
rising edges of CANCLK occur at rising or falling edges of CLK

CANCLK starts always with a falling edge

SCANPREEN:The internal synchronized PRES-bit of the CMCR register (BIT[7])
CAN prescaler enable

CLK: Unmodulated input clock (output of PLL1 or X0 oscillator)

6.2.2 Modulation Parameter Register (CMPR)

The Modulation Parameter Register (CMPR) determines the modulation degree.

Precaution: Write this register only, when the modulator is switched off. I.e. at first the modulator has to be configured, then it can be enabled with the MSRT-bit.

Depending on the input frequency and the frequency resolution, several different modulation degrees can be chosen. The appendix contains a list of possible register settings. In general the the highest possible frequency resolution combined with the highest possible modulation degree results in the highest EMI reduction. But for some cases a lower resolution with a higher modulation degree may result in a better EMI behaviour.

Register	Address	initial value	value to be programmed
CMPR	000166h	0xF9F1	refer to the appendix

6.2.3 Frequency Resolution Setting Register (CMLS0..3)

The CMLS registers determine the frequency resolution of the modulation range

Precaution: Write these registers only, when the modulator is switched off. I.e. at first the modulator has to be configured, then it can be enabled with the MSRT-bit.

The frequency resolution within the modulation range can be set in three steps from low to high. Higher resolution implies a finer granularity of discrete frequencies in the spectrum of the modulated clock but less possible modulation degrees.

In general the highest possible frequency resolution combined with the highest possible modulation degree results in the highest EMI reduction. But for some cases a lower resolution with a higher modulation degree may result in a better EMI behaviour.

CMLS0 and CMLS1 must be always set to 0xF800, resp. 0xFF04. The value for CMLS2 and CMLS3 depends on the desired frequency resolution. Please refer to the table of possible settings in the appendix.

Register	Address	initial value	value to be programmed
CMLS0	000168h	0x77FF	0xF800
CMLS1	00016Ah	0x77FF	0xFF04
CMLS2	00016Ch	0x77FF	refer to the appendix

CMLS3	00016Eh	0x77FF	refer to the appendix

6.2.4 Random Number Generator and Observer Register (CMLT0..3)

CMLT0 - CMLT3 configure the random number generator. In addition CMLT1 contains the observer control register.

Precaution: Write these registers only, when the modulator is switched off. I.e. at first the modulator has to be configured, then it can be enabled with the MSRT-bit.

The random number generator must be configured with the register values listed in the following table:

Register	Address	initial value	value to be programmed
CMLT0	000170h	0xFC02	0xF802
CMLT1	000172h	0xF402	0x**S**802
CMLT2	000174h	0xFC02	0xF802
CMLT3	000176h	0xFC02	0xF802

legend: **S** = OB**S**ERVER => control register described below

The observer control register is located in CMLT1.

CMLT1 high byte

	15	14	13	12	11	10	9	8	⇐ Bit no.
Address: 000172H	OBS3	OBS2	OBS1	OBS0	LT11	LT10	LT9	LT8	CMLT1H
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(1)	(1)	(1)	(1)	(0)	(1)	(0)	(0)	

CMLT1 low byte

	7	6	5	4	3	2	1	0	⇐ Bit no.
Address: 000173H	LT7	LT6	LT5	LT4	LT3	LT2	LT1	LT0	CMLT1L
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(1)	(0)	

■ **BIT[15:12] OBS - Observer**

These bits are implemented in CMLT1-register. Default value is b1111.

With OBS[3:0], the output at the MONCLK-pin is selected. To observe internal signals at the MONCLK-pin, this must be enabled with the MTST-bit in CMCR register.

If the OBS-bits shall determine which signal can be observed, MSEL, located in CMCR, must be set to 1. Otherwise the modulated clock is observable at MONCLK.

Note: The OBS-signal is not synchronized. Therefore glitches can occur at the MONCLK-pin during the switching.
The MONCLK-pin is driven by a pin driver. Therefore the frequency spectrum, measured at this pin, does not represent the spectrum of the internal clock signal.

The following internal signals are observable:

CMCR. MSEL	OBS[3:0]		signal		
	bin	dec	MB91F361G MB91F362GB, MB91F369GA	MB91FV360GA, MB91F364G, MB91F366GB, MB91F368GB	MB91F365GB, MB91F367GB
0	dddd		modulated clock	modulated clock	modulated clock
1	0000	0	unmodulated clock (PLL)	unmodulated clock (PLL)	unmodulated clock (PLL)
1	0001	1	reserved	4MHz oscillator	4MHz oscillator
1	0010	2	4MHz oscillator	32kHz oscillator	0
1	0011	3	CAN clock	CAN clock	CAN clock
1	else	4-14	reserved	reserved	reserved
1	1111	15	automatic calibration timer pulse	automatic calibration timer pulse	automatic calibration timer pulse

legend: d=don’t care

6.2.5 Automatic Calibration Reload Timer Value (CMAC)

This register contains the reload value of the automatic calibration reload timer.

Precaution: Write this register only, when the modulator is switched off. I.e. at first the modulator has to be configured, then it can be enabled with the MSRT-bit.

high byte

	15	14	13	12	11	10	9	8	⇐ Bit no.
Address: 000178H	RELF	RELE	RELD	RELC	RELB	RELA	REL9	REL8	CMACH
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	

low byte

	7	6	5	4	3	2	1	0	⇐ Bit no.
Address: 000179H	REL7	REL6	REL5	REL4	REL3	REL2	REL1	REL0	CMACL
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	

The automatic calibration reload timer triggers a new calibration of the modulator in intervals of the time tCAL. tCAL can be calculated with the following equation:

$$t_{CAL} = \frac{2047 \cdot reloadvalue}{F_0}$$

The following table shows the maximal possible calibration interval (reload value = 0xFFFF) for different input frequencies. Because the modulator switches to unmodulated clock during calibration, the calibration interval time tCAL should be not less than 1 sec.

F0 (input clock)	**maximum-tCAL**	**duration of calibration**
16MHz	8.39 sec	3 µs
32MHz	4.19 sec	1.5 µs
48MHz	2.8 sec	1 µs

6.2.6 Status Register (CMTS)

The Status Register contains the error flags ELF and EHF of the calibration unit

Precaution: All other bits are reserved and read only.

high byte

	15	14	13	12	11	10	9	8	⇐ Bit no.
Address: 00017AH	-	-	ELF	EHF	-	-	-	-	CMTSH
Read/write ⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
Default value⇒	(1)	(1	(0)	(0)	(0)	(0)	(0)	(1)	

low byte

	7	6	5	4	3	2	1	0	⇐ Bit no.
Address: 00017BH	-	-	-	-	-	-	-	-	CMTSL
Read/write ⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
Default value⇒	(0)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	

■ BIT[12] EHF - error flag high input frequency

The EHF error flag is set, if the modulator can not be calibrated due to high input frequency. Reduce the input frequency in order to avoid misbehavior of the clock modulator.

The condition is checked at the end of the calibration process. The flag is always cleared at the beginning of an new calibration process. The error flag has read only access.

0	no error occured during calibration
1	an error occured during calibration - input frequency is too high

■ BIT[13] ELF - error flag low input frequency

The ELF error flag is set, if the modulator can not be calibrated due to low input frequency. Increase the input frequency in order to avoid misbehavior of the clock modulator.

The condition is checked at the end of the calibration process. The flag is always cleared at the beginning of an new calibration process. The error flag has read only access.

0	no error occured during calibration
1	an error occured during calibration - input frequency is roo low

6.3 APPENDIX

The appendix provides application examples and tables with possible register settings.

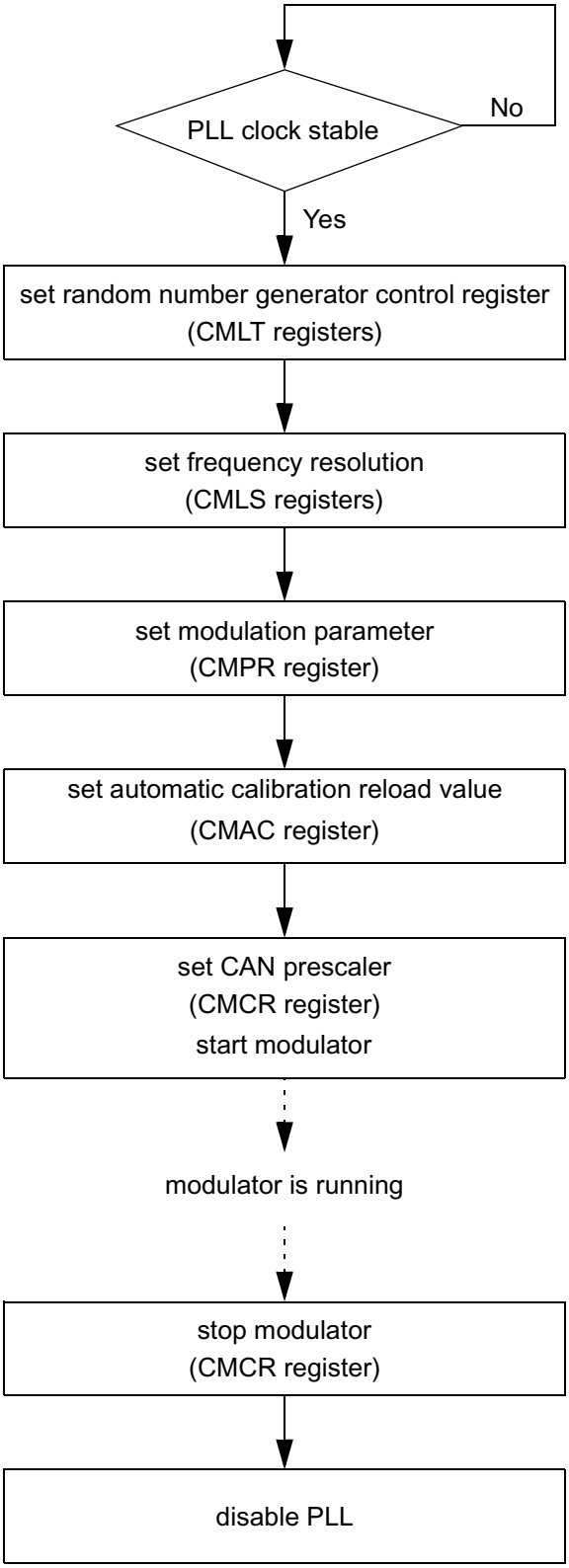
6.3.1 Modulation Parameter selection

It is not possible to recommend a particular modulation parameter setting to achieve a particular reduction in EMI. The best setting depends much on the actual application, the whole system and the requirements.

In order to find the optimal modulation parameter setting, the following approach is recommended.

1)	define the required PLL frequency based on performance needs	e.g. 32 MHz
2)	choose the setting with the highest resolution and modulation degree.	e.g. resolution:medium, degree:2
3)	perform EMI measurements	
4)	if the EMI measuerments does not fulfill the requirements, you may either reduce the modulation degree at the same frequency resolution, or increase the modulation degree at a lower frequency resolution	e.g. resolution:medium, degree:1 or e.g. resolution:low, degree:5
5)	repeat item 3) with the new setting and continue until the best settings is identified	

6.3.2 Configuration Flowchart



6.3.3 Example program

The following example assembler program shows, how to configure and start the clock modulator. It is important to begin with the configuration of the modulator and start it in the last step.

- First, the CMLT control registers are set to their required values. The observer selector remains the initial value 0xF.
- The frequency resolution is set to medium.
- Corresponding to the selected frequency resolution there are 3 possible modulation degree settings for 16MHz input clock, 2 settings for 24MHz, 2 settings for 32MHz and 1 setting for 40MHz (refer to the table of possible settings in the appendix). Assuming the input clock is 32MHz, modulation degree 2 is chosen (frequency range from 23.273MHz to 51.2MHz).
- The automatic calibration timer reload value is set to 0xFFFF what corresponds to an calibration interval of 4.192 sec. at 32MHz input clock.
- Finally, the clock modulator is started. At the same time, the CAN-prescaler is set to 3 (CANCLK=input_clock/4) and the MONCLK pin remains disabled. Note that the calibration is triggered (CAL bit) and the random number generator is initialized (IRNG bit) together with the start of the modulator.

```
// control register CMLT
     ldi       #0xF802,r0
     ldi       #CMLT0,r1
     ldi       #CMLT1,r2
     ldi       #CMLT2,r3
     ldi       #CMLT3,r4
     sth       r0,@r1
     sth       r0,@r2
     sth       r0,@r3
     sth       r0,@r4



// frequency resolution register CMLS
     ldi       #0xF800,r0
     ldi       #0xFF04,r1
     ldi       #0xF813,r2
     ldi       #0x0000,r3
     ldi       #CMLS0,r4
     ldi       #CMLS1,r5
     ldi       #CMLS2,r6
     ldi       #CMLS3,r7
```

```
     sth       r0,@r4
     sth       r1,@r5
     sth       r2,@r6
     sth       r3,@r7

// modulation parameter register CMPR
ldi  #0x0A82,r0
     ldi       #CMPR,r1
     sth       r0,@r1

// calibration reload timer CMAC = 0xFFFF
     ldi       #0xFFFF,r0
     ldi       #CMAC,r1
     sth       r0,@r1

// control register CMCR
// set CAN-prescaler = 3 (CANCLK=input_CLK/4), enable CAN clock
// disable MONCLK pin
// start clock modulator
     ldi       #0x03B3,r0
     ldi       #CMCR,r1
     sth       r0,@r1

   .
   .
   .

// stop clock modulator
// CAN clock remains enabled
     ldi       #0x0380,r0
     ldi       #CMCR,r1
     sth       r0,@r1
```

6.3.4 Possible Settings

F_0= input-frequency

F_{max}= upper modulation range border

F_{min}= lower modulation range border

Note: The actual highest frequency occurring in the modulated clock can be higher than the listed frequency F_{max} . Respectively the lowest occurring frequency can be lower than F_{min}. The absolute highest frequency occurring in the modulated clock is below 64 MHz.

F0 [MHz]	frequency resolution	modulation degree	F_{min} [MHz]	F_{max} [MHz]	CMLS2	CMLS3	CMPR
16	low	1	15.059	17.067	0x0000	0x0000	0x0F81
16	low	2	14.222	18.286	0x0000	0x0000	0x0E82
16	low	3	13.474	19.692	0x0000	0x0000	0x0D83
16	low	4	12.800	21.333	0x0000	0x0000	0x0C84
16	low	5	12.190	23.273	0x0000	0x0000	0x0B85
16	low	6	11.636	25.600	0x0000	0x0000	0x0A86
16	low	7	11.130	28.444	0x0000	0x0000	0x0987
16	low	8	10.667	32.000	0x0000	0x0000	0x0888
16	low	9	10.240	36.571	0x0000	0x0000	0x0789
16	low	10	9.846	42.667	0x0000	0x0000	0x068A
16	medium	1	13.474	19.692	0xF813	0x0000	0x0D81
16	medium	2	11.636	25.600	0xF813	0x0000	0x0A82
16	medium	3	10.240	36.571	0xF813	0x0000	0x0783
16	high	1	11.130	28.444	0xF813	0xFF84	0xF9F1
24	low	1	22.588	25.600	0x0000	0x0000	0x0F81
24	low	2	21.333	27.429	0x0000	0x0000	0x0E82
24	low	3	20.211	29.538	0x0000	0x0000	0x0D83
24	low	4	19.200	32.000	0x0000	0x0000	0x0C84
24	low	5	18.286	34.909	0x0000	0x0000	0x0B85
24	low	6	17.455	38.400	0x0000	0x0000	0x0A86
24	low	7	16.696	42.667	0x0000	0x0000	0x0987
24	low	8	16.000	48.000	0x0000	0x0000	0x0888
24	medium	1	20.211	29.538	0xF813	0x0000	0x0D81
24	medium	2	17.455	38.400	0xF813	0x0000	0x0A82
24	high	1	16.696	42.667	0xF813	0xFF84	0xF9F1

32	low	1	30.118	34.133	0x0000	0x0000	0x0F81
32	low	2	28.444	36.571	0x0000	0x0000	0x0E82
32	low	3	26.947	39.385	0x0000	0x0000	0x0D83
32	low	4	25.600	42.667	0x0000	0x0000	0x0C84
32	low	5	24.381	46.545	0x0000	0x0000	0x0B85
32	low	6	23.273	51.200	0x0000	0x0000	0x0A86
32	medium	1	26.947	39.385	0xF813	0x0000	0x0D81
32	medium	2	23.273	51.200	0xF813	0x0000	0x0A82
40	low	1	37.647	42.667	0x0000	0x0000	0x0F81
40	low	2	35.556	45.714	0x0000	0x0000	0x0E82
40	low	3	33.684	49.231	0x0000	0x0000	0x0D83
40	medium	1	33.684	49.231	0xF813	0x0000	0x0D81
48	low	1	45.176	51.200	0x0000	0x0000	0x0F81
48	low	2	42.667	54.857	0x0000	0x0000	0x0E82

legend:

initial value

CHAPTER 7 I/O PORTS

This chapter provides an overview of I/O ports, lists the registers, and describes conditions for using external pins as I/O ports.

7.1 I/O PORTS AND REGISTER CONFIGURATION

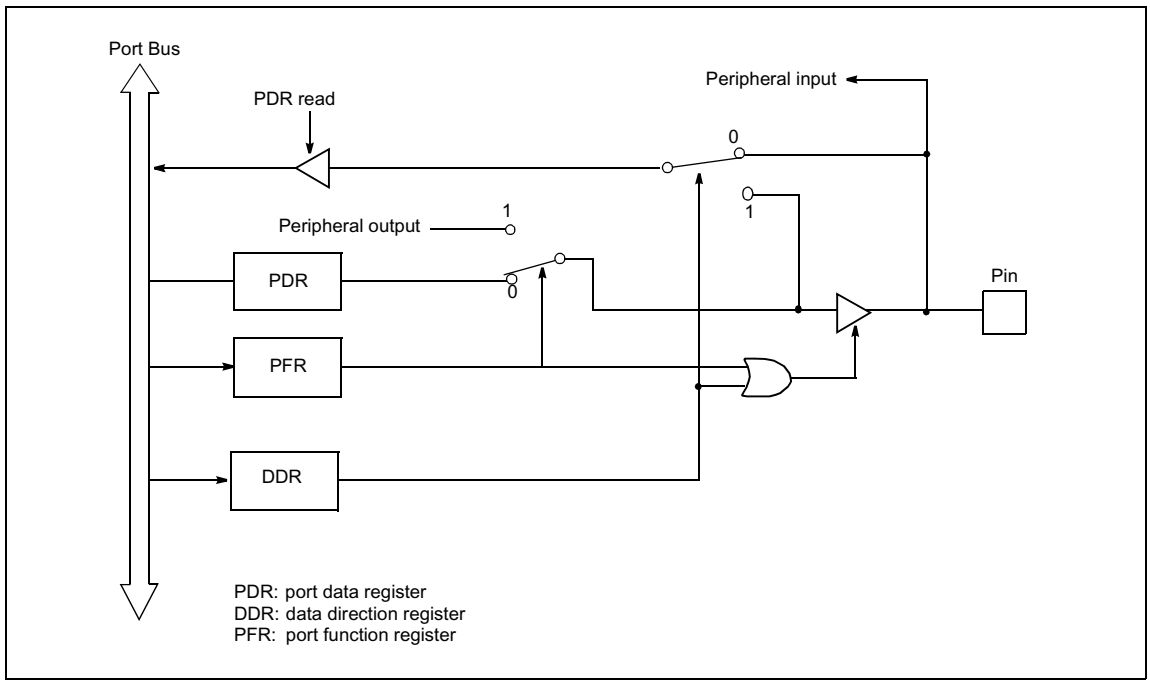
In the MB91360, pins not used by peripherals can be used as I/O ports. This section shows the basic port block diagram (structure diagram) and lists the register configuration.

The MB91360 has the following three types of I/O port registers:

- **Port data and Port function registers (PDR and PFR)**
- **Data direction registers (DDR)**

■ Basic Block Diagram of Port

Figure 7.1a Basic Block Diagram of Port



■ Register Description and Configuration

The I/O port registers consist of the "port data registers (PDR)", the "data direction registers (DDR)" and the "port function registers (PFR)".

The bits in PDRs correspond to the bits in DDRs and PFRs. Similarly, the register bits correspond to the port pins.

The port data registers contain the port I/O data and the data direction registers specify whether the corresponding bits (pins) are inputs or outputs. Bits set to "0" are inputs and bits set to "1" are outputs.The port function registers specify whether the port is used as peripheral port or as "I/O" port. Usually bits set to "0" mean I/O port and bits set to "1" mean functional port.

In case of analog peripherals there is additional circuitry to ensure that the digital logic is not

disturbed by the analog signals. If the analog input function e.g. ADC is enabled the digital input is fixed to "0".

- Input mode (DDR = "0")
  - PDR read.... : Reads the level on the corresponding external pin.
  - PDR write ... : Writes the PDR setting value.
- Output mode (DDR = "1")
  - PDR read.... : Reads the PDR value.
  - PDR write ... : Outputs the PDR value to the corresponding external pins.

7.1.2 Port Data Registers

		7	6	5	4	3	2	1	0	Initial Value	Access
PDR7											
Address:	00000007H	P77	P76	P75	P74	P73	P72	P71	P70	1111XXXXB	R/W
PDR8											
Address:	00000008H	P87	P86	P85	P84	P83	P82	P81	P80	XXXXXXXXB	R/W
PDR9											
Address:	00000009H	P97	P96	P95	P94	P93	P92	P91	P90	XXXXXXX1B	R/W
PDRB											
Address:	0000000BH	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	XXXXXXXXB	R/W
PDRG											
Address:	00000010H	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0	XXXXXXXXB	R/W
PDRH											
Address:	00000011H	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0	XXXXXXXXB	R/W
PDRI											
Address:	00000012H	PI7	---	---	---	PI3	---	---	---	X---X---B	R/W
PDRJ											
Address:	00000013H	PJ7	PJ6	PJ5	PJ4	PJ3	PJ2	PJ1	PJ0	XXXXXXXXB	R/W
PDRK											
Address:	00000014H	PK7	PK6	PJK	PK4	PK3	PK2	PJK	PK0	XXXXXXXXB	R/W
PDRL											
Address:	00000015H	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0	XXXXXXXXB	R/W
PDRM											
Address:	00000016H	---	---	---	---	PM3	PM2	PM1	PM0	----XXXXB	R/W

PDRN	7	6	5	4	3	2	1	0	Initial Value	Access
Address: 00000017H	---	---	PN5	PN4	PN3	PN2	PN1	PN0	--XXXXXXB	R/W

PDRO	7	6	5	4	3	2	1	0	Initial Value	Access
Address: 00000018H	PO7	PO6	PO5	PO4	PO3	PO2	PO1	PO0	XXXXXXXXB	R/W

PDRP	7	6	5	4	3	2	1	0	Initial Value	Access
Address: 00000019H	PP7	PP6	PP5	PP4	PP3	PP2	PP1	PP0	XXXXXXXXB	R/W

PDRQ	7	6	5	4	3	2	1	0	Initial Value	Access
Address: 0000001AH	---	---	PQ5	PQ4	PQ3	PQ2	PQ1	PQ	--XXXXXXB	R/W

PDRR	7	6	5	4	3	2	1	0	Initial Value	Access
Address: 0000001BH	PR7	PR6	PR5	PR4	PR3	PR2	PR1	PR0	XXXXXXXXB	R/W

PDRS	7	6	5	4	3	2	1	0	Initial Value	Access
Address: 0000001CH	PS7	PS6	PS5	PS4	PS3	PS2	PS1	PS0	XXXXXXXXB	R/W

PDRT	7	6	5	4	3	2	1	0	Initial Value	Access
Address: 0000001DH	---	---	PT5	PT4	PT3	PT2	PT1	PS0	--XXXXXXB	R/W

Figure 7.1.2 Port Data Registers

7.1.3 Data Direction Registers (DDR)

DDR7	7	6	5	4	3	2	1	0	Initial Value	Access
Address: 00000607H	P77	P76	P75	P74	P73	P72	P71	P70	00000000B	R/W

DDR8	7	6	5	4	3	2	1	0	Initial Value	Access
Address: 00000608H	P87	P86	P85	P84	P83	P82	P81	P80	00000000B	R/W

DDR9	7	6	5	4	3	2	1	0	Initial Value	Access
Address: 00000609H	P97	P96	P95	P94	P93	P92	P91	P90	00000000B	R/W

DDRB	7	6	5	4	3	2	1	0	Initial Value	Access
Address: 0000060BH	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	00000000B	R/W

DDRG	7	6	5	4	3	2	1	0	Initial Value	Access
Address: 00000400H	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0	00000000B	R/W

DDRH	7	6	5	4	3	2	1	0	Initial Value	Access
Address: 00000401H	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0	00000000B	R/W

DDRI	7	6	5	4	3	2	1	0	Initial Value	Access
Address: 00000402H	---	---	---	---	PI3	---	---	---	----0---B	R/W

DDRJ	7	6	5	4	3	2	1	0	Initial Value	Access
Address: 00000403H	PJ7	PJ6	PJ5	PJ4	PJ3	PJ2	PJ1	PJ0	00000000B	R/W

DDRK	7	6	5	4	3	2	1	0	Initial Value	Access
Address: 00000404H	PK7	PK6	PK5	PK4	PK3	PK2	PK1	PK0	00000000B	R/W

DDRL	7	6	5	4	3	2	1	0	Initial Value	Access
Address: 00000405H	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0	00000000B	R/W

		7	6	5	4	3	2	1	0	Initial Value	Access
DDRM											
Address:	00000406H	---	---	---	---	PM3	PM2	PM1	PM0	----0000B	R/W
DDRN		7	6	5	4	3	2	1	0	Initial Value	Access
Address:	00000407H	---	---	PN5	PN4	PN3	PN2	PN1	PN0	--000000B	R/W
DDRO		7	6	5	4	3	2	1	0	Initial Value	Access
Address:	00000408H	PO7	PO6	PO5	PO4	PO3	PO2	PO1	PO0	00000000B	R/W
DDRP		7	6	5	4	3	2	1	0	Initial Value	Access
Address:	00000409H	PP7	PP6	PP5	PP4	PP3	PP2	PP1	PP0	00000000B	R/W
DDRQ		7	6	5	4	3	2	1	0	Initial Value	Access
Address:	0000040AH	---	---	PQ5	PQ4	PQ3	PQ2	PQ1	PQ0	--000000B	R/W
DDRR		7	6	5	4	3	2	1	0	Initial Value	Access
Address:	0000040BH	PR7	PR6	PR5	PR4	PR3	PR2	PR1	PR0	00000000B	R/W
DDRS		7	6	5	4	3	2	1	0	Initial Value	Access
Address:	0000040CH	PS7	PS6	PS5	PS4	PS3	PS2	PS1	PS0	00000000B	R/W
DDRT		7	6	5	4	3	2	1	0	Initial Value	Access
Address:	0000040DH	---	---	PT5	PT4	PT3	PT2	PT1	PS0	--000000B	R/W

Figure 7.1.3 Data Direction Registers

7.1.4 Port Function Registers (PFR)

PFR7	7	6	5	4	3	2	1	0	Initial Value	Access
Address: 00000617H	P77	P76	P75	P74	P73	P72	P71	P70	00001111B	R/W

PFR8	7	6	5	4	3	2	1	0	Initial Value	Access
Address: 00000618H	P87	P86	P85	P84	P83	P82	---	---	111110--B	R/W

PFR9	7	6	5	4	3	2	1	0	Initial Value	Access
Address: 00000619H	P97	P96	P95	P94	P93	P92	P91	P90	11110101B	R/W

PFRB	7	6	5	4	3	2	1	0	Initial Value	Access
Address: 0000061BH	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	00000000B	R/W

PFR27	7	6	5	4	3	2	1	0	Initial Value	Access
Address: 00000627H	P77	P76	P75	P74	P73	P72	P71	P70	1111-00-B	R/W

PFRG	7	6	5	4	3	2	1	0	Initial Value	Access
Address: 00000410H	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0	00000000B	R/W

PFRH	7	6	5	4	3	2	1	0	Initial Value	Access
Address: 00000411H	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0	00000000B	R/W

PFRI	7	6	5	4	3	2	1	0	Initial Value	Access
Address: 00000412H	---	---	---	---	PI3	---	---	---	----0---B	R/W

PFRJ	7	6	5	4	3	2	1	0	Initial Value	Access
Address: 00000413H	PJ7	PJ6	PJ5	PJ4	PJ3	PJ2	PJ1	PJ0	00000000B	R/W

PFRK	7	6	5	4	3	2	1	0	Initial Value	Access
Address: 00000414H	PK7	PK6	PK5	PK4	PK3	PK2	PK1	PK0	00000000B	R/W

		7	6	5	4	3	2	1	0	Initial Value	Access
PFRL											
Address:	00000415H	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0	00000000B	R/W
PFRM											
Address:	00000416H	---	---	---	---	PM3	PM2	PM1	PM0	----0000B	R/W
PFRN											
Address:	00000417H	---	---	PN5	PN4	PN3	PN2	PN1	PN0	--000000B	R/W
PFRO											
Address:	00000418H	PO7	PO6	PO5	PO4	PO3	PO2	PO1	PO0	00000000B	R/W
PFRP											
Address:	00000419H	PP7	PP6	PP5	PP4	PP3	PP2	PP1	PP0	00000000B	R/W
PFRQ											
Address:	0000041AH	---	---	PQ5	PQ4	PQ3	PQ2	PQ1	PQ0	--000000B	R/W
PFRR											
Address:	0000041BH	PR7	PR6	PR5	PR4	PR3	PR2	PR1	PR0	00000000B	R/W
PFRS											
Address:	0000041CH	PS7	PS6	PS5	PS4	PS3	PS2	PS1	PS0	00000000B	R/W
PFRT											
Address:	0000041DH	---	---	PT5	PT4	PT3	PT2	PT1	PT0	--000000B	R/W

Figure 7.1.4 Port Function Registers

7.2 PORT FUNCTION REGISTER SETTINGS

The following table lists the initial values and functions of PFR registers.

Note: About PFR7, PFR27, PFR8, PFR9 and PFRB see also section
8.5 "USING THE BUS INTERFACE AS GENERAL I/O PORTS" on page 271.

Table 7.2a Port Function Registers PFR7, PFR27

Register Name		Bit Name		Bit		Function
--	PFR7	--	P70	-	0	Versatile port
				-	1	Address output A24(Initial status)
PFR27(P271)	PFR7(P71)	P271	P71	0	0	Versatile port
				0	1	Address output A25(Initial status)
				1	0	Setting disabled
				1	1	IOWX
PFR27(P272)	PFR7(P72)	P272	P72	0	0	Versatile port
				0	1	Address output A26 (Initial status)
				1	0	Setting disabled
				1	1	IORX
--	PFR7	--	P73	-	0	Versatile port
				-	1	Address output A27 (Initial status)
PFR27(P274)	PFR7(P74)	P274	P74	0	0	Versatile port
				0	1	Address output A28
				1	0	1-output (Initial status)
				1	1	CS4 output (see also section 4.2)
PFR27(P275)	PFR7(P75)	P275	P75	0	0	Versatile port
				0	1	Address output A29
				1	0	1-output (Initial status)
				1	1	CS5 output (see also section 4.2)
PFR27(P276)	PFR7(P76)	P276	P76	0	0	Versatile port
				0	1	Address output A30
				1	0	1-output (Initial status)
				1	1	CS6 output (see also section 4.2)
PFR27(P277)	PFR7(P77)	P277	P77	0	0	Versatile port
				0	1	Address output A31
				1	0	1-output (Initial status)
				1	1	CS7 output

Table 7.2b Port Function Registers PFR8 ... PFRT

Register Name	Bit Name	Bit	Function
PFR8			
PFR8 (BGRNTX)	–	–	(Becomes a dedicated BGRNTX pin if the P82 bit is set to the BRO pin can be used as versatile port otherwise.)
PFR8 (BRQ)	P82	0	Versatile port (Initial status)
		1	BRQ
PFR8 (RDX)	P83	1	Must be set to 1
PFR8 (WR0X)	P84	1	Must be set to 1
PFR8 (WR1X)	P85	1	Must be set to 1
PFR8 (WR2X)	P86	0	Versatile port
		1	WR2X output (Initial status)
PFR8 (WR3X)	P87	0	Versatile port
		1	WR3X output (Initial status)
PFR9			
PFR9 (AS)	P90	0	Versatile port
		1	AS (Initial status)
PFR9 (ALE)	P91	0	Versatile port (Initial value)
		1	ALE is not implemented (output 0)
PFR9 (CLK)	P92	0	Versatile port, see section 8.5 !
		1	CLK Output (Initial value)
PFR9 (AH/BOOT)	P93	0	AH is not implemented (Initial status: 0)
		1	Versatile port
PFR9 (CS0X)	P94	0	Versatile port
		1	Should be set to 1 when an external bus is used. (Initial value: 1)
PFR9 (CS1X)	P95	0	Versatile port
		1	CS1 output (Initial status)
PFR9 (CS2X)	P96	0	Versatile port
		1	CS2 output (Initial status)
PFR9 (CS3X)	P97	0	Versatile port
		1	CS3 output (Initial status)

Table 7.2b Port Function Registers PFR8 ... PFRT (Continued)

Register Name	Bit Name	Bit	Function
PFRB			
PFRB (PB0), PFRB (PB3)	PB0, PB3	0	Should be set to 0 during normal operation. (Initial value: 0)
PFRB (PB1), PFRB (PB4), PFRB (PB6)	PB1, PB4, PB6	0	Versatile port (Initial status)
		1	DACK0, 1, 2
PFRB (bit2), DDRB (PB2)	bit2, PB2	0,0	Versatile port input (Initial status)
		0,1	Versatile port output
		1,0	DMAC: DSTP input
		1,1	DMAC: DEOP output
PFRB (bit5), DDRB (PB5)	bit5, PB5	0,0	Versatile port input (Initial status)
		0,1	Versatile port output
		1,0	DMAC: DSTP input
		1,1	DMAC: DEOP output
PFRB (bit7), DDRB (PB7)	bit7, PB7	0,0	Versatile port input (Initial status)
		0,1	Versatile port output
		1,0	DMAC: DSTP input
		1,1	DMAC: DEOP output
PFRG ... PFRK			
PFRG	PG7-PG0	0	I/O port (Initial value)
		1	AN15 - AN8 (ADC inputs)
PFRH	PH7-PH0	0	I/O Port (Initial value)
		1	AN7 - AN0 (ADC inputs)
PFRI	PI3	0	I/O Port (Initial value)
		1	ATGX (ADC trigger input)
	PI7		The value of SELCLK can be read at PI7 (PDR7[7])
PFRJ	PJ7-PJ0	0	I/O Port (Initial value)
		1	LED7-LED0 (LED port outputs)
PFRK	PK7-PK0	0	I/O Port (Initial value)
		1	INT7 - INT0 (ext. interrupts inputs)

Table 7.2b Port Function Registers PFR8 ... PFRT (Continued)

Register Name	Bit Name	Bit	Function
PFRL			
PFRL	PL0-PL3	0	I/O Port (Initial value)
		1	IN0-IN3 (ICU inputs)
	PL4-PL7	0	I/O Port (Initial value)
		1	OUT0-OUT3 (OCU outputs)
PFRM			
PFRM	PM0	0	I/O Port (Initial value)
		1	SGO (Sound Gen. output)
	PM1	0	I/O Port (Initial value)
		1	SGA (Sound Gen. output)
	PM2	0	I/O Port (Initial value)
		1	SDA (I^2C data)
	PM3	0	I/O Port (Initial value)
		1	SCL (I^2C clock)
PFRN			
PFRN	PN0	0	I/O Port (Initial value)
		1	SOT4 (SIO output)
	PN1	0	I/O Port (Initial value)
		1	SIN4 (SIO input)
	PN2	0	I/O Port (Initial value)
		1	SCK4 (SIO clock)
	PN3	0	I/O Port (Initial value)
		1	SIN3 (SIO input)
	PN4	0	I/O Port (Initial value)
		1	SOT3 (SIO output)
	PN5	0	I/O Port (Initial value)
		1	SCK3 (SIO clock)
PFRO			
PFRO	PO0-PO7	0	I/O Port (Initial value)
		1	OCPA0 - OCPA7 (PPG outputs)

Table 7.2b Port Function Registers PFR8 ... PFRT (Continued)

Register Name	Bit Name	Bit	Function
PFRP			
PFRP	PP0	0	I/O Port (Initial value)
		1	TX0 (CAN 0)
	PP1	0	I/O Port (Initial value)
		1	RX0 (CAN 0)
	PP2	0	I/O Port (Initial value)
		1	TX1 (CAN 1)
	PP3	0	I/O Port (Initial value)
		1	RX1 (CAN 1)
	PP4	0	I/O Port (Initial value)
		1	TX2 (CAN 2)
	PP5	0	I/O Port (Initial value)
		1	RX2 (CAN 2)
	PP6	0	I/O Port (Initial value)
		1	TX3 (CAN 3)
	PP7	0	I/O Port (Initial value)
		1	RX3 (CAN 3)
PFRQ			
PFRQ	PQ0	0	I/O Port (Initial value)
		1	SIN0 (UART 0) (value after boot ROM [1])
	PQ1	0	I/O Port (Initial value)
		1	SOT0 (UART 0) (value after boot ROM [1])
	PQ2	0	I/O Port (Initial value)
		1	SIN1 (UART 1)
	PQ3	0	I/O Port (Initial value)
		1	SOT1 (UART 1)
	PQ4	0	I/O Port (Initial value)
		1	SIN2 (UART 2)
	PQ5	0	I/O Port (Initial value)
		1	SOT2 (UART 2)

Table 7.2b Port Function Registers PFR8 ... PFRT (Continued)

Register Name	Bit Name	Bit	Function
PFRR			
PFRR	PR0	0	I/O Port (Initial value)
		1	PWM1P0 (Stepper Motor Control)
	PR1	0	I/O Port (Initial value)
		1	PWM1M0 (Stepper Motor Control)
	PR2	0	I/O Port (Initial value)
		1	PWM2P0 (Stepper Motor Control)
	PR3	0	I/O Port (Initial value)
		1	PWM2M0 (Stepper Motor Control)
	PR4	0	I/O Port (Initial value)
		1	PWM1P1 (Stepper Motor Control)
	PR5	0	I/O Port (Initial value)
		1	PWM1M1 (Stepper Motor Control)
	PR6	0	I/O Port (Initial value)
		1	PWM2P1 (Stepper Motor Control)
	PR7	0	I/O Port (Initial value)
		1	PWM2M1 (Stepper Motor Control)

Table 7.2b Port Function Registers PFR8 ... PFRT (Continued)

Register Name	Bit Name	Bit	Function
PFRS			
PFRS	PS0	0	I/O Port (Initial value)
		1	PWM1P2 (Stepper Motor Control)
	PS1	0	I/O Port (Initial value)
		1	PWM1M2 (Stepper Motor Control)
	PS2	0	I/O Port (Initial value)
		1	PWM2P2 (Stepper Motor Control)
	PS3	0	I/O Port (Initial value)
		1	PWM2M2 (Stepper Motor Control)
	PS4	0	I/O Port (Initial value)
		1	PWM1P3 (Stepper Motor Control)
	PS5	0	I/O Port (Initial value)
		1	PWM1M3 (Stepper Motor Control)
	PS6	0	I/O Port (Initial value)
		1	PWM2P3 (Stepper Motor Control)
	PS7	0	I/O Port (Initial value)
		1	PWM2M3 (Stepper Motor Control)

Table 7.2b Port Function Registers PFR8 ... PFRT (Continued)

Register Name	Bit Name	Bit	Function
PFRT (MB91F364G only)			
PFRT	PT0		I/O Port (Initial value)
			SIN5 (LIN-UART 5)
	PT1		I/O Port (Initial value)
			SCK5 (LIN-UART 5)
	PT2		I/O Port (Initial value)
			SOT5 (LIN-UART 5)
	PT3		I/O Port (Initial value)
			SOT6 (LIN-UART 6)
	PT4		I/O Port (Initial value)
			SCK6 (LIN-UART 6)
	PT5		I/O Port (Initial value)
			SIN6 (LIN-UART 6)

Note: 1) Only if a valid boot condition is detected during the execution of the boot code, this value will be set.

CHAPTER 8 EXTERNAL BUS INTERFACE

This Chapter describes in detail basic information about the external bus interface, the register structure and functions, bus operation basics and bus timing.

8.1 BUS INTERFACE

The external bus interface controls the interfaces with the external memory and external I/Os.

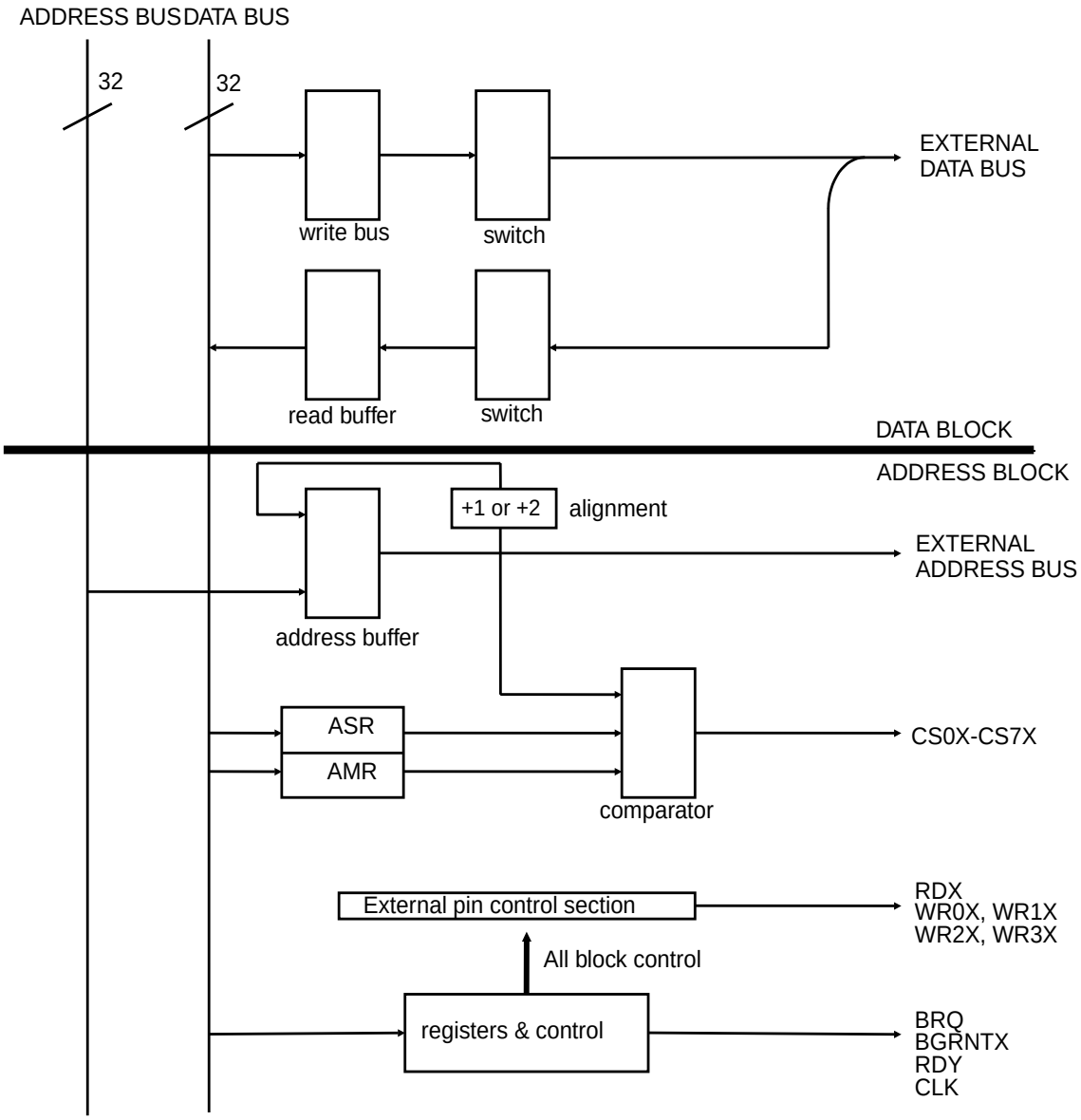
8.1.1 Features

- Up to 32-bit (4 GB) address output.
- Up to eight independent banks provided by chip-select function. [1)]
- The banks can be set in 64-KB (minimum) at any position in the logic address space.
- Can be set to no area.
- 32/16/8 bit bus width setup can be performed for each chip-select area.
- Programmable automatic memory wait (up to 7 cycles) insertion.
- Unused address/data pins can be used as I/O ports.[2)]
- Can handle DMAC fly-by

Note: [1)] Chip Select Areas CS7 and CS1 are used for the internal CAN modules and Flash module (F361G only) respectively. The necessary register settings are done by an internal boot routine. Take care not to overwrite register bits related to those CS area.

Note: [2)] If the CAN macros which are connected internally to the external bus (also called User Logic Bus) are used, a certain number of data , address and control ports of the external bus interface cannot be configured as general purpose IO ports.
For details see sections 7.2 "PORT FUNCTION REGISTER SETTINGS" on page 220 and 8.5 "USING THE BUS INTERFACE AS GENERAL I/O PORTS" on page 271.

8.1.2 Block Diagram



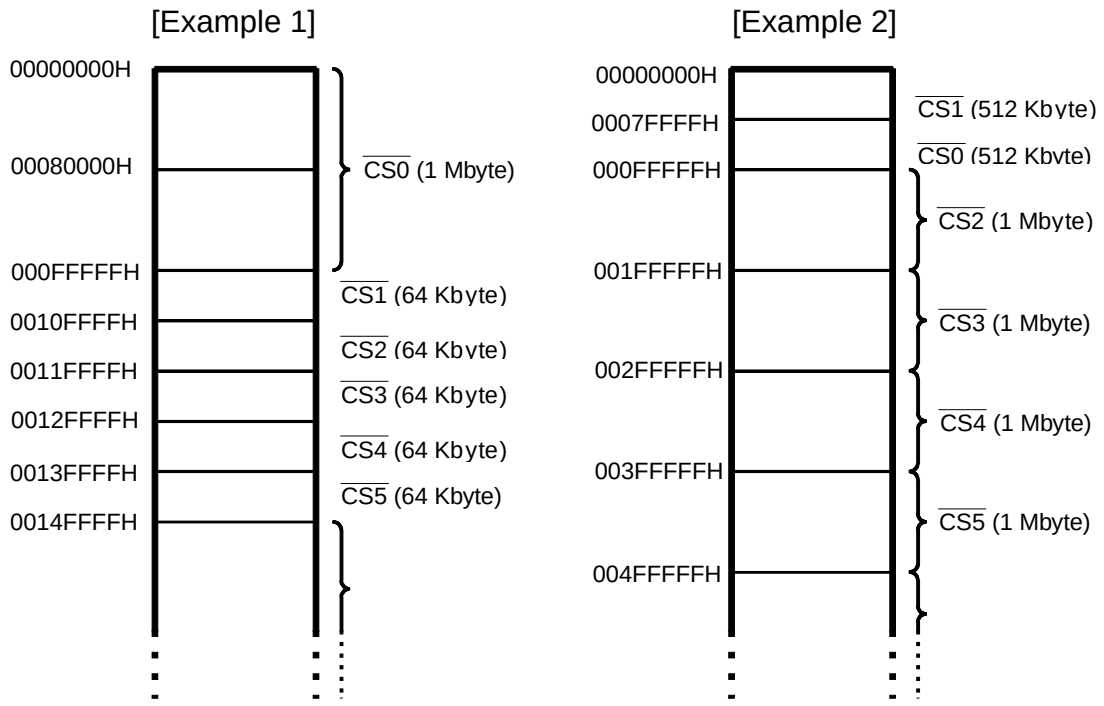
8.1.3 Overview

■ Area

A total of eight types of chip-select areas are provided for the bus interface.

Individual areas can be positioned in 64-KB (minimum) within a 4-GB space using area select registers ASR0 to ASR7 and area mask registers AMR0 to AMR7. External bus access to areas designated by the ASR0 to ASR7 registers causes the corresponding chip-select signals CS0X to CS7X to be active 'L'.

Example 1 shows how area 0 is arranged at addresses 00000000H to 000FFFFFH, and areas 1 to 5 are arranged at addresses 00100000H to 0014FFFFH in 64-KB increments. Example 2 shows how area 0 is arranged at addresses 00080000H to 000FFFFFH, and area 1 is arranged at addresses 00000000H and 0007FFFFH in 512-KB increments and areas 2 to 5 are arranged in 1-MB increments at addresses 00100000H to 004FFFFFH.



■ Bus Size Designation

The bus width can be specified for all areas by performing register setup.

For area 0, after clearing the setting initialization reset (INIT) and operation initialization reset (RST), the bus size is specified by the setting written to the mode vector (MODR) using mode vector fetch.

8.1.4 Register List

Address	31 to 24	23 to 16	15 to 8	7 to 0
0000 0640H	ASR0(Area Select Reg. 0)		AMR0(Area Mode Reg. 0)	
0000 0644H	ASR1(Area Select Reg. 1)		AMR1(Area Mode Reg. 1)	
0000 0648H	ASR2(Area Select Reg. 2)		AMR2(Area Mode Reg. 2)	
0000 064CH	ASR3(Area Select Reg. 3)		AMR3(Area Mode Reg. 3)	
0000 0650H	ASR4(Area Select Reg. 4)		AMR4(Area Mode Reg. 4)	
0000 0654H	ASR5(Area Select Reg. 5)		AMR5(Area Mode Reg. 5)	
0000 0658H	ASR6(Area Select Reg. 6)		AMR6(Area Mode Reg. 6)	
0000 065CH	ASR7(Area Select Reg. 7)		AMR7(Area Mode Reg. 7)	
0000 0660H	AMD0	AMD1	AMD2	AMD3
0000 0664H	AMD4	AMD5	AMD6	AMD7
0000 0668H	CSE	-	-	-
0000 0670H	CHE	-	-	-
0000 07FCH		MODR		

ASR (Area Select Register)

AMR (Area Mask Register)

AMD (Area Mode Register)

CSE (Chip Select Enable Register)

CHE (CacHe Enable Register)

MODR(MODe Register)

8.1.5 Description of Registers

■ **Area Select Registers (ASR0 to ASR7)**

									Initial value		
ASR0	15	14	13	12 · · ·	2	1	0		INIT	RST	Access
0000 0640 H	A31	A30	A29	· · · · · ·	A18	A17	A16		0000 H	0000 H	W
									Initial value		Access
ASR1	15	14	13	12 · · ·	2	1	0		INIT	RST	
0000 0644 H	A31	A30	A29	· · · · · ·	A18	A17	A16		0000 H	xxxx H	W
									Initial value		Access
ASR2	15	14	13	12 · · ·	2	1	0		INIT	RST	
0000 0648 H	A31	A30	A29	· · · · · ·	A18	A17	A16		0000 H	xxxx H	W
									Initial value		Access
ASR3	15	14	13	12 · · ·	2	1	0		INIT	RST	
0000 064C H	A31	A30	A29	· · · · · ·	A18	A17	A16		0000 H	xxxx H	W
									Initial value		Access
ASR4	15	14	13	12 · · ·	2	1	0		INIT	RST	
0000 0650 H	A31	A30	A29	· · · · · ·	A18	A17	A16		0000 H	xxxx H	W
									Initial value		Access
ASR5	15	14	13	12 · · ·	2	1	0		INIT	RST	
0000 0654 H	A31	A30	A29	· · · · · ·	A18	A17	A16		0000 H	xxxx H	W
									Initial value		Access
ASR6	15	14	13	12 · · ·	2	1	0		INIT	RST	
0000 0658 H	A31	A30	A29	· · · · · ·	A18	A17	A16		0000 H	xxxx H	W
									Initial value		Access
ASR7	15	14	13	12 · · ·	2	1	0		INIT	RST	
0000 065CH	A31	A30	A29	· · · · · ·	A18	A17	A16		0000 H	xxxx H	W

Note: After execution of the code in the internal boot ROM ASR0 is set to "0x20" and ASR7 to "0x10".

■ Area Mask Registers (AMR0 to AMR7)

		15	14	13	12	···	2	1	0	Initial value INIT	Initial value RST	Access
AMR0	0000 0642 H	A31	A30	A29	···	···	A18	A17	A16	FFFF H	FFFF H	W
AMR1	0000 0646 H	A31	A30	A29	···	···	A18	A17	A16	0000 H	xxxx H	W
AMR2	0000 064A H	A31	A30	A29	···	···	A18	A17	A16	0000 H	xxxx H	W
AMR3	0000 064E H	A31	A30	A29	···	···	A18	A17	A16	0000 H	xxxx H	W
AMR4	0000 0652 H	A31	A30	A29	···	···	A18	A17	A16	0000 H	xxxx H	W
AMR5	0000 0656 H	A31	A30	A29	···	···	A18	A17	A16	0000 H	xxxx H	W
AMR6	0000 065A H	A31	A30	A29	···	···	A18	A17	A16	0000 H	xxxx H	W
AMR7	0000 065E H	A31	A30	A29	···	···	A18	A17	A16	0000 H	xxxx H	W

Note: After execution of the code in the internal boot ROM AMR0 is set to "0x0F" and AMR7 to "0x00"

Area select registers **ASR0** to **ASR7** and area mask registers **AMR0** to **AMR7** specify the address space range for chip-select areas 0 to 7.

The **ASR0** to **ASR7** registers specify the 16 high-order address bits **A31** to **A16**.

The **AMR0** to **AMR7** registers mask the associated address bits. Regarding the **AMR0** to **AMR7** bits, the value 0 means **care** and the value 1 means **don't care**. **care** indicates address space 0 when the **ASR** setting is 0, or address space 1 when the **ASR** setting is **1**. **don't care** indicates address spaces 0 and 1 irrespective of the **ASR** setting.

ASR0, **AMR0**: Initialization is performed at reset, setting all the areas in the address space to the specification of area 0.

ASR1 to **7**, **AMR1** to **7**: Initialization is performed at INIT. At reset, the previous value is retained.

An **ASR/AMR** combination example of individual chip select area designations is shown below.

Example 1:
```
ASR1    =   00000000  00000011  B
AMR1    =   00000000  00000000  B
```

When the above setup is used, the **AMR1** bit related to the bit setting **1** for the **ASR1** is 0. Therefore, the area-1 address space is 64 KB as shown below.

```
00000000  00000011 00000000  00000000 B   (00030000 H)
                      to
00000000  00000011 11111111  11111111 B   (0003FFFF H)
```

Example 2:
```
ASR2    =   00001111  11111111  B
AMR2    =   00000000  00000011  B
```

When the above setup is used, the **ASR2** setting related to the bit setting 0 for the **AMR2** is **care** (**1** for **1** or **0** for **0**), whereas the **ASR2** bit related to the bit setting **1** for the **AMR2** is **don't care** (**0** or **1**). Therefore, the area-2 address space is 256 KB as shown below.

```
00001111  11111100 00000000  00000000 B   (0DDC0000 H)
                      to
00001111  11111111 11111111  11111111 B   (0FFFFFFF H)
```

The address spaces for areas 1 to 5 can be arranged as desired in minimum 64-KB unit within a 4-GB space using the **ASR1** to **ASR5** registers, and **AMR1** to **AMR5** registers. When bus access is gained to an area designated by such registers, the associated chip select pin (CS0 to CS7) delivers a Low output.

At reset, area 0 is allocated to all areas.

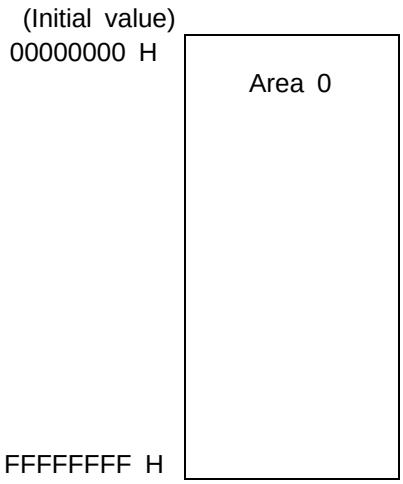
If a continuous area is required there is also a certain maximum for a chip select area. The table below shows some examples for this:

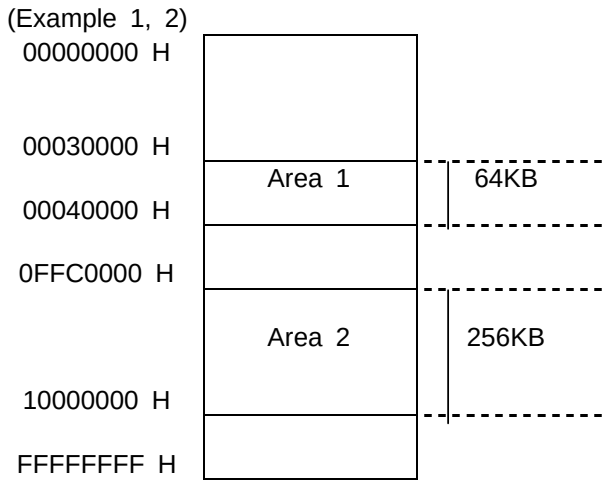
Table 8.1.5 CS Area Examples

ASR	AMR	CS Area
0020H	001FH	1 MB: 0020:0000H to 002F:FFFFH
0200H	01FFH	16 MB: 0200:0000H to 02FF:FFFFH
2000H	1FFFH	256 MB: 2000:0000H to 2FFF:FFFFH

Note: Chip-select areas must be set so that they do not overlap.

When chip select areas overlap, the chip select area with the smaller number has priority.
Both chip select signals will be active but the settings for bus width, auto wait cycles, RDY enable of the area with the smaller number will be used.





■ Area Mode Registers (AMD0 to AMD7)

The **AMD0** to **AMD7** registers specify the memory access operation mode for individual chip-select areas.

AMD0	7	6	5	4	3	2	1	0	Initial value INIT	Initial value RST	Access
0000 0660 H	-	-	RDYE	BW1	BW0	WTC2	WTC1	WTC0	- 0000111 B	- 00xx111 B	R/W

After execution of the code in the internal boot ROM this register is set to "0x11".

The **AMD0** specifies the operating mode for chip-select area 0. When a reset is performed, area 0 is selected for all areas.

[bit6] = (reserved)

Reserved bit

At ordinary operation, always set this bit to 0.

[bit5] = RDYE (ReaDY input Enable bit)

RDYE controls RDY input for area 0 as shown below.

RDYE	
0	Invalidates RDY input (initial value)
1	Validates RDY input

[bit4,3] = BW1,0 (Bus Width bit)

The **BW1** and **BW0** bits specify the bus width for area 0.

BW1	BW0	Bus width
0	0	8 bits
0	1	16 bits
1	0	32 bits
1	1	reserved

After resetting INIT, the mode-fetched value (WTH bit of mode register) is reflected to BW bit immediately after mode-fetching.

[bit2 to 0] = WTC2 to 0 (Wait Cycle bit)

These bits specify the automatic wait insertion count for the normal bus interface.

WTC2	WTC1	WTC0	Insertion Wait Cycle Count
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7 (Initial value)

When reset, the **WTC2** to **WTC0** bits of the **AMD0** are set to **111**. For bus access immediately after the reset state is cleared, 7-cycle wait insertion is performed automatically.

AMD1	7	6	5	4	3	2	1	0	Initial value INIT	Initial value RST	Access
0000 0661 H	-	-	RDYE	BW1	BW0	WTC2	WTC1	WTC0	- 0000000 B	- xxxxxxx B	R/W

The **AMD1** specifies the operating mode for chip-select area 1 (area designated by **ASR1** and **AMR1**).

[bit6] = (reserved)

Reserved bit

At ordinary operation, always set this bit to 0.

[bit5] = RDYE (ReaDY input Enable bit)

RDYE controls RDY input for area 1 as shown below.

RDYE	
0	Invalidates RDY input
1	Validates RDY input

[bit4,3] = BW1,0 (Bus Width bit)

These bits specify the area-1 bus width.

BW1	BW0	Bus width
0	0	8 bits
0	1	16 bits
1	0	32 bits
1	1	reserved

[bit2 to 0] = WTC2 to 0 (Wait Cycle bit)

These bits specify the automatic insertion wait cycle count for normal bus interface and time division I/O interface operations. These bits operate in the same manner as the **WTC2** to **WTC0** bits of the **AMD0**. However, at reset, they are initialized to **000**, so the insertion wait cycle count is **0**.

Note: This product type does not support the time division I/O interface.

									Initial value		
AMD2	7	6	5	4	3	2	1	0	INIT	RST	Access
0000 0662 H	-	-	RDYE	BW1	BW0	WTC2	WTC1	WTC0	--000000 B	--xxxxxx B	R/W

AMD2 specifies the operation mode for chip select area 2.

[bit5] = RDYE (ReaDY input Enable bit)

RDYE controls RDY input for area 2 as shown below.

RDYE	
0	Invalidates RDY input
1	Validates RDY input

[bit4,3] = BW1,0 (Bus Width bit)

BW1, 0 specify the bus width for area 2.

BW1	BW0	Bus width
0	0	8 bits
0	1	16 bits
1	0	32 bits
1	1	reserved

[bit2 to 0] = WTC2 to 0(Wait Cycle bit)

WTC2 to WTC0 specify the auto-insert wait cycle count for the normal bus interface operation.

WTC2	WTC1	WTC0	Auto-insert wait cycle count
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

WTC2 to WTC0 have a similar function to those for the other AMD registers and are initialized to 000 at setting initialization reset (INIT), and the insert wait cycle count is 0.

									Initial value		
AMD3	7	6	5	4	3	2	1	0	INIT	RST	Access
0000 0663 H	-	-	RDYE	BW1	BW0	WTC2	WTC1	WTC0	--000000 B	--xxxxxx B	R/W

AMD3 specifies the operation mode for chip select area 3.

[bit5] = RDYE (ReaDY input Enable bit)

RDYE controls RDY input for area 3 as shown below.

RDYE	
0	Invalidates RDY input
1	Validates RDY input

[bit 4, 3] = BW1, 0 (Bus Width bit)

BW1, 0 specify the bus width for area 3.

BW1	BW0	Bus width
0	0	8 bits
0	1	16 bits
1	0	32 bits
1	1	reserved

[bit2 to bit0] = WTC2 to WTC0 (Wait Cycle bit)

WTC2 to WTC0 specify the auto-insert wait cycle count for the normal bus interface operation.

WTC2	WTC1	WTC0	Auto-insert wait cycle count
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

WTC2 to WTC0 have a similar function to those for the other AMD registers and are initialized to 000 at setting initialization reset (INIT), and the insert wait cycle count is 0.

									Initial value		
AMD4	7	6	5	4	3	2	1	0	INIT	RST	Access
0000 0664 H	-	-	RDYE	BW1	BW0	WTC2	WTC1	WTC0	--000000 B	--xxxxxx B	R/W

AMD4 specifies the operation mode for chip select area 4.

[bit5] = RDYE (ReaDY input Enable bit)

RDYE controls RDY input for area 4 as shown below.

RDYE	
0	Invalidates RDY input
1	Validates RDY input

[bit4,3] = BW1,0 (Bus Width bit)

BW1, 0 specify the bus width for area 4.

BW1	BW0	Bus width
0	0	8 bits
0	1	16 bits
1	0	32 bits
1	1	reserved

[bit2 to bit0] = WTC2 to WTC0 (Wait Cycle bit)

WTC2 to WTC0 specify the auto-insert wait cycle count for the ordinary bus interface operation.

WTC2	WTC1	WTC0	Auto-insert wait cycle count
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

WTC2 to WTC0 have the function similar to those of the other AMD registers and are initialized to "000" at setting initialization reset (INIT), and the insert wait cycle count is '0'.

									Initial value		
AMD5	7	6	5	4	3	2	1	0	INIT	RST	Access
0000 0665 H	-	-	RDYE	BW1	BW0	WTC2	WTC1	WTC0	--000000 B	--XXXXXX B	R/W

AMD5 specifies the operation mode for chip select area 5.

[bit5] = RDYE (ReaDY input Enable bit)

RDYE controls RDY input for area 5 as shown below.

RDYE	
0	Invalidates RDY input
1	Validates RDY input

[bit4,3] = BW1,0 (Bus Width bit)

BW1, 0 specify the bus width for area 5.

BW1	BW0	Bus width
0	0	8 bits
0	1	16 bits
1	0	32 bits
1	1	reserved

[bit2 to bit0] = WTC2 to WTC0 (Wait Cycle bit)

WTC2 to WTC0 specify the auto-insert wait cycle count for the normal bus interface operation.

WTC2	WTC1	WTC0	Auto-insert wait cycle count
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

WTC2 to WTC0 have a similar function to those for the other AMD registers and are initialized to 000 at setting initialization reset (INIT), and the insert wait cycle count is 0.

									Initial value		
AMD6	7	6	5	4	3	2	1	0	INIT	RST	Access
0000 0666 H	-	-	RDYE	BW1	BW0	WTC2	WTC1	WTC0	--000000 B	--xxxxxx B	R/W

AMD6 specifies the operation mode for chip select area 6 (the area specified by ASR6 and AMR6).

[bit6] = (reserved)

Reserved bit. At ordinary operation, always set this bit to 0.

[bit5] = RDYE (ReaDY input Enable bit)

RDYE controls RDY input for area 6 as shown below.

RDYE	
0	Invalidates RDY input
1	Validates RDY input

[bit4,3] = BW1,0 (Bus Width bit)

BW1, 0 specify the bus width for area 6.

BW1	BW0	Bus width
0	0	8 bits
0	1	16 bits
1	0	32 bits
1	1	reserved

[bit2 to bit0] = WTC2 to WTC0 (Wait Cycle bit)

WTC2 to WTC0 specify the auto-insert wait cycle count for the memory access to area 6.WTC2 to WTC0 have a similar function to those for the other AMD registers and are initialized to 000 at reset, and the insert wait cycle count is 0.

									Initial value		
AMD7	7	6	5	4	3	2	1	0	INIT	RST	Access
0000 0667 H	-	-	RDYE	BW1	BW0	WTC2	WTC1	WTC0	--000000 B	--xxxxxx B	R/W

After execution of the code in the internal boot ROM this register is set to "0x29" (only on F361G).

AMD7 specifies the bus mode for chip select area 7 (the area specified by ASR7 and AMR7).

Each bit here has the same meaning as for AMD6.

■ **CHE (CacHe Enable register)**

	15	14	13	12	11	10	9	8	Initial value	Access
0000 0670 H	CHE7	CHE6	CHE5	CHE4	CHE3	CHE2	CHE1	CHE0	11111111 B	R/W

[bit15 to bit8]

These bits specify the cache/non-cache area in each CS area (CS0 to CS7).

0: Non-cache area

1: Cache area

CHE bit	Associated CS
CHE0	CS0
CHE1	CS1
CHE2	CS2
CHE3	CS3
CHE4	CS4
CHE5	CS5
CHE6	CS6
CHE7	CS7

CHE0 is associated with the CS0 area, and CHE1 is associated with the CS1 area.

When a non-cache area specification and a cache area specification are overlapped, the non-cache area specification has priority over the cache area specification.

■ **CSE (Chip Select Enable register)**

	15	14	13	12	11	10	9	8	Initial value	Access
0000 0668 H	CSE7	CSE6	CSE5	CSE4	CSE3	CSE2	CSE1	CSE0	00000001 B	R/W

After execution of the code in the internal boot ROM this register is set to "0x81".

[bit15 to bit8]

These bits are the enable bits for the CS0 to CS7 chip select areas.

The initial value is 00000001 B; only CSE0 is enabled.

Writing 1 to CSE0 to CSE7 validates the setting of ASR0 to ASR7 and AMR0 to AMR7.

CSE0 to CSE7	Setting of ASR0 to ASR7, AMR0 to AMR7
0	disable
1	enable

CSE bit	Associated CS
CSE0	CS0
CSE1	CS1
CSE2	CS2
CSE3	CS3
CSE4	CS4
CSE5	CS5
CSE6	CS6
CSE7	CS7

CSE0 is associated with the CS0 area, and CSE1 is associated with the CS1 area.

■ MODR (MODe Register)

See Section 2.10.2 "Mode Setting" on page 126.

8.2 BUS OPERATION

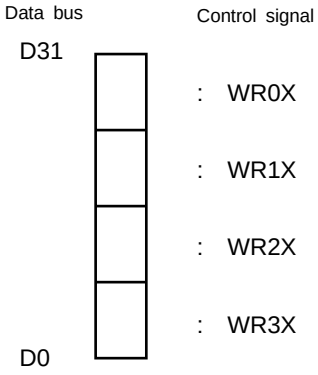
8.2.1 Relationship between Data Bus Width and Control Signal

The WRX to WR3X, control signals always have a one-to-one correspondence with the data bus byte positions irrespective of big/little endian mode or data bus width.

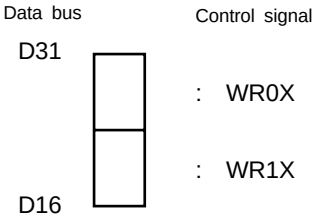
The following summarizes the relationships between the control signals and the MB91360 data bus byte positions used for preselected data bus widths in various bus modes.

■ **Normal bus interface**

a)32bit bus width

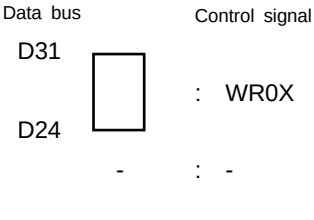


b)16bit bus width



(D15 to D0: not used)

b)8bit bus width



(D23 to D0: not used)

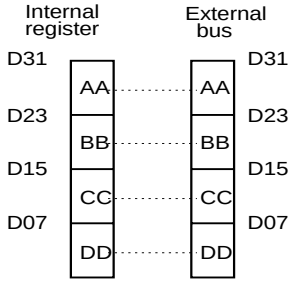
8.2.2 Big-endian Bus Access

FR50 series devices perform external access using big endian.

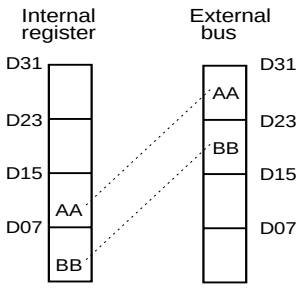
■ **Data format**

The relationship between the internal register and external data bus is as follows.

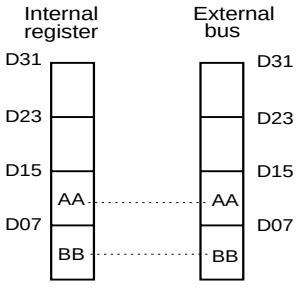
Word access Halfword access
(when **LD/ST** instruction executed) (when **LDUH/STH** instruction executed)



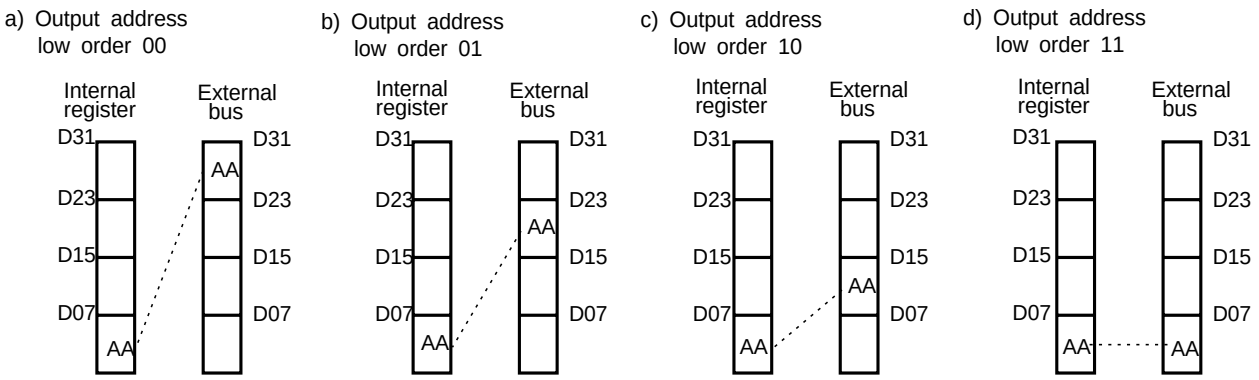
a) Output address low order 00



b) Output address low order 10

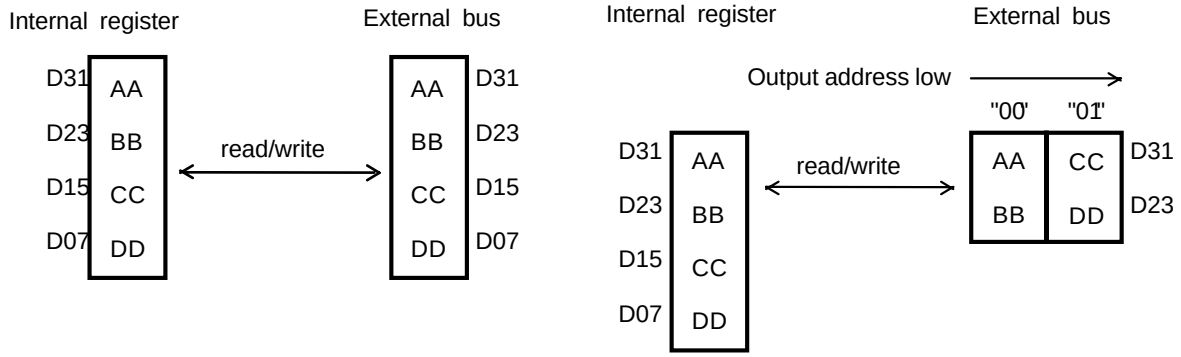


Byte access (when **LDUB/STB** instruction executed)

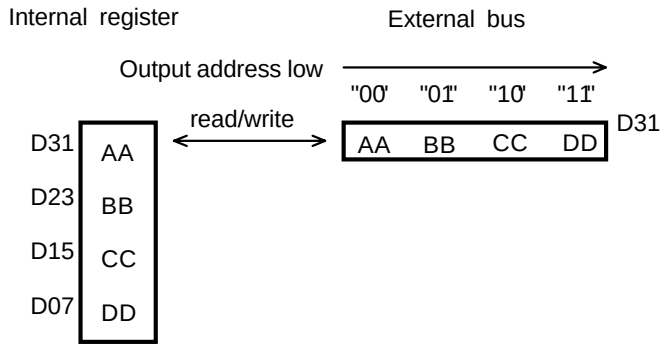


■ Data bus width

32-bit bus width16-bit bus width



8-bit bus width



■ External bus access

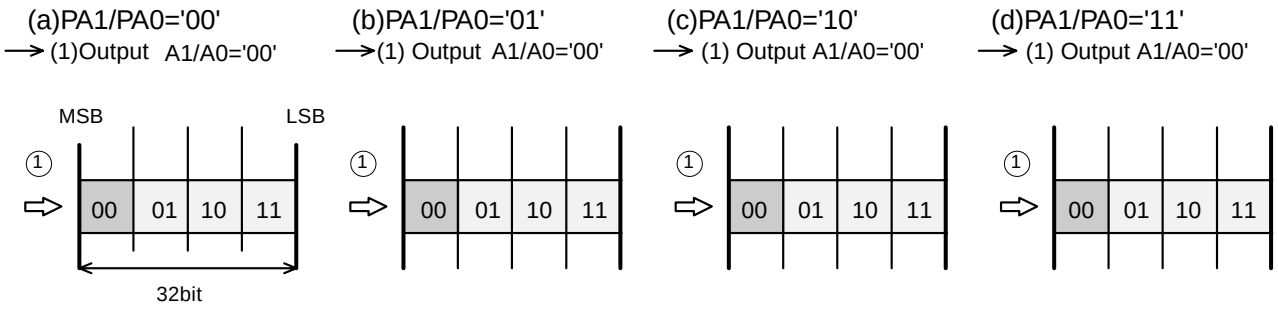
To summarize the external bus access, this section describes the access byte position, program address and output address, and bus access count under the 32-bit/16-bit/8-bit bus width and word/halfword/byte access categories.

PA1/PA0	Program-designated rwo low-order address bits
Output A1/A0	Two low order address bits to be output
□ + □	Starting byte position of address to be output
	Data byte position to access
(1) to (4)	Bus access count

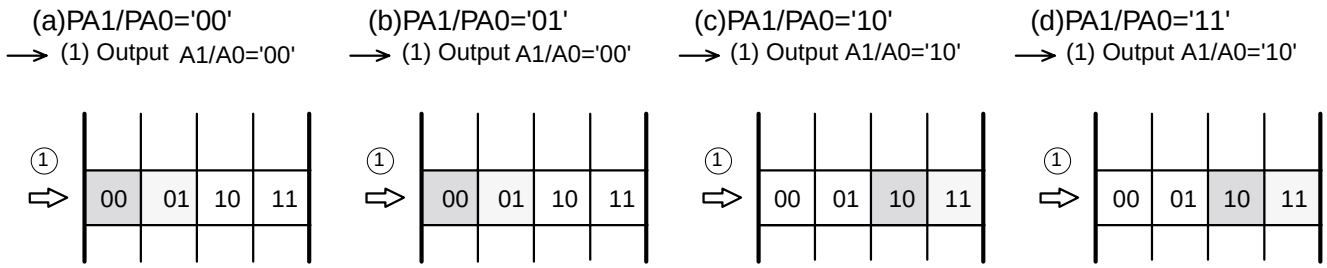
Therefore, at word access, the output address two low-order bits are all 00 even when the program-designated address two low-order bits are **00**, **01**, **10**, or **11**. On the other hand, at halfword access, the output address two low-order bits are **00** when the program-designated address two low-order bits are **00** or **01**, or **10** when the program-designated address two low-order bits are **10** or **11**.

- 32-bit bus width

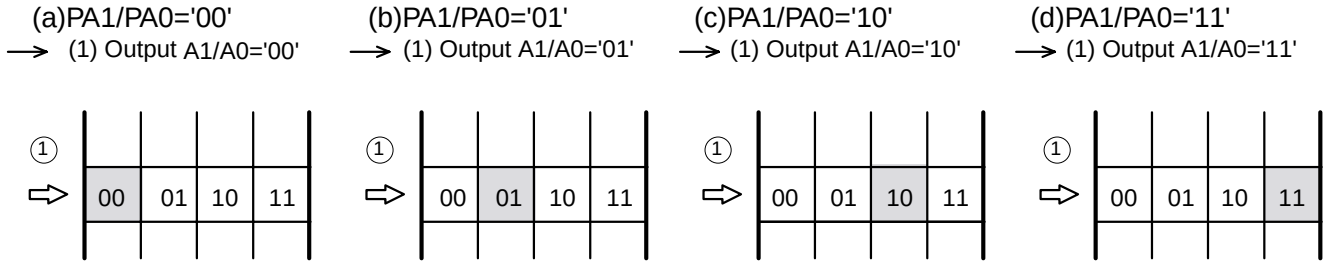
(A) word access



(B) half word access



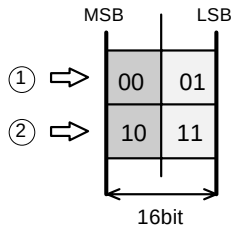
(C) byte access



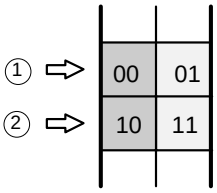
- 16bit bus width

(A) word access

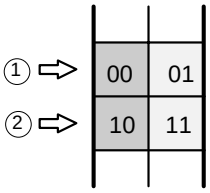
(a)PA1/PA0='00'
→ (1) Output A1/A0='00'
(2) Output A1/A0='10'



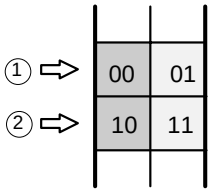
(b)PA1/PA0='01'
→(1) Output A1/A0='00'
(2) Output A1/A0='10'



(c)PA1/PA0='10'
→ (1) Output A1/A0='00'
(2) Output A1/A0='10'

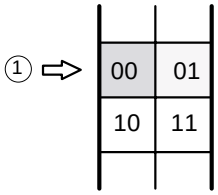


(d)PA1/PA0='11'
→ (1) Output A1/A0='00'
(2) Output A1/A0='10'

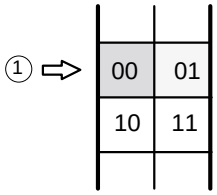


(B) half word access

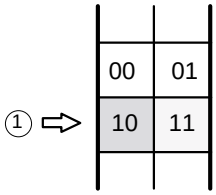
(a)PA1/PA0='00'
(1) Output A1/A0='00'



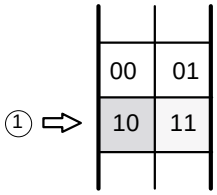
(b)PA1/PA0='01'
(1) Output A1/A0='00'



(c)PA1/PA0='10'
(1) Output A1/A0='10'

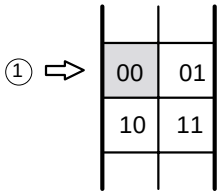


(d)PA1/PA0='11'
(1) Output A1/A0='10'

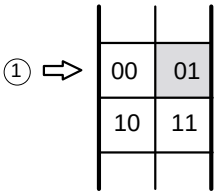


(C) byte access

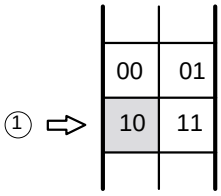
(a)PA1/PA0='00'
→ (1) Output A1/A0='00'



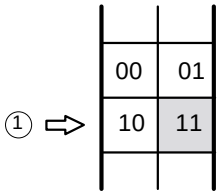
(b)PA1/PA0='01'
→ (1) Output A1/A0='01'



(c)PA1/PA0='10'
→ (1) Output A1/A0='10'



(d)PA1/PA0='11'
→ (1) Output A1/A0='10'



- 8bit bus width

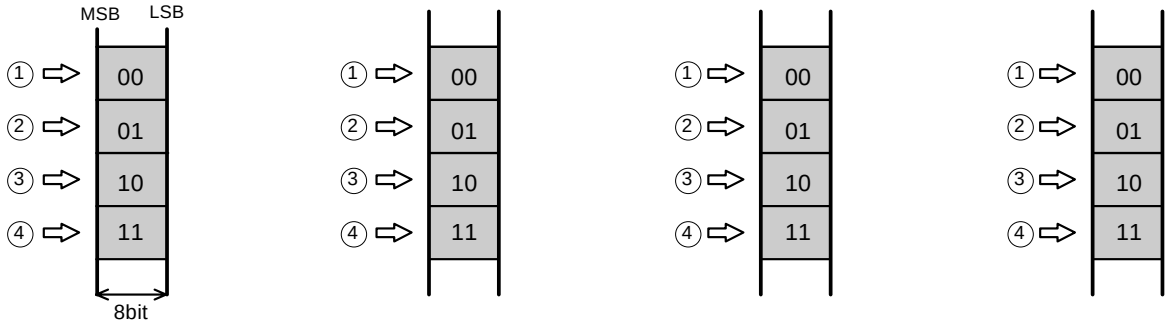
(A) word access

(a)PA1/PA0='00'
→ (1) Output A1/A0='00'
(2) Output A1/A0='01'
(3) Output A1/A0='10'
(4) Output A1/A0='11'

(b)PA1/PA0='01'
→ (1) Output A1/A0='00'
(2) Output A1/A0='01'
(3) Output A1/A0='10'
(4) Output A1/A0='11'

(c)PA1/PA0='10'
→ (1) Output A1/A0='00'
(2) Output A1/A0='01'
(3) Output A1/A0='10'
(4) Output A1/A0='11'

(d)PA1/PA0='11'
→ (1) Output A1/A0='00'
(2) Output A1/A0='01'
(3) Output A1/A0='10'
(4) Output A1/A0='11'



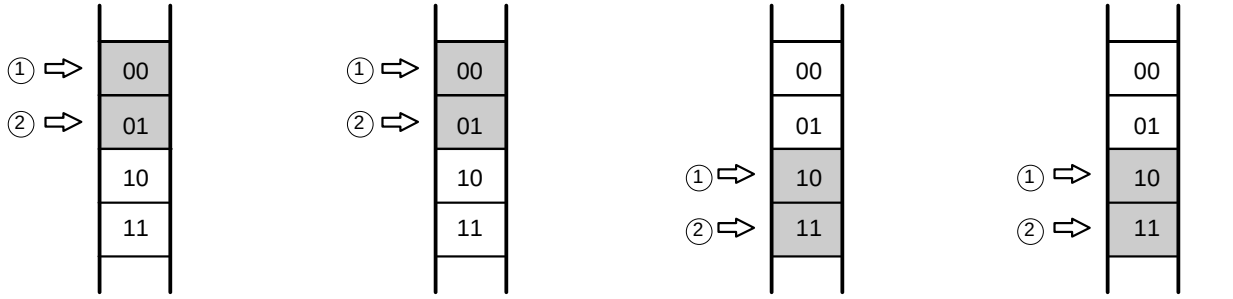
(B) half word access

(a)PA1/PA0='00'
→ (1) Output A1/A0='00'
(2) Output A1/A0='01'

(b)PA1/PA0='01'
→ (1) Output A1/A0='00'
(2) Output A1/A0='01'

(c)PA1/PA0='10'
→ (1) Output A1/A0='10'
(2) Output A1/A0='11'

(d)PA1/PA0='11'
→ (1) Output A1/A0='10'
(2) Output A1/A0='11'



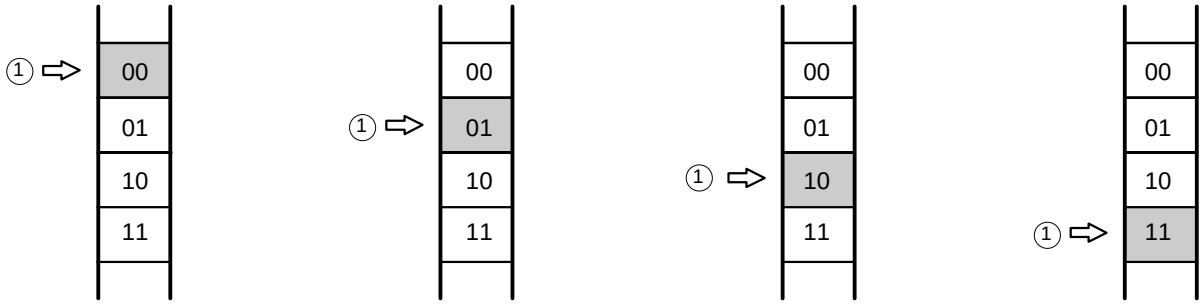
(C) byte access

(a)PA1/PA0='00'
→ (1) Output A1/A0='00'

(b)PA1/PA0='01'
→ (1) Output A1/A0='01'

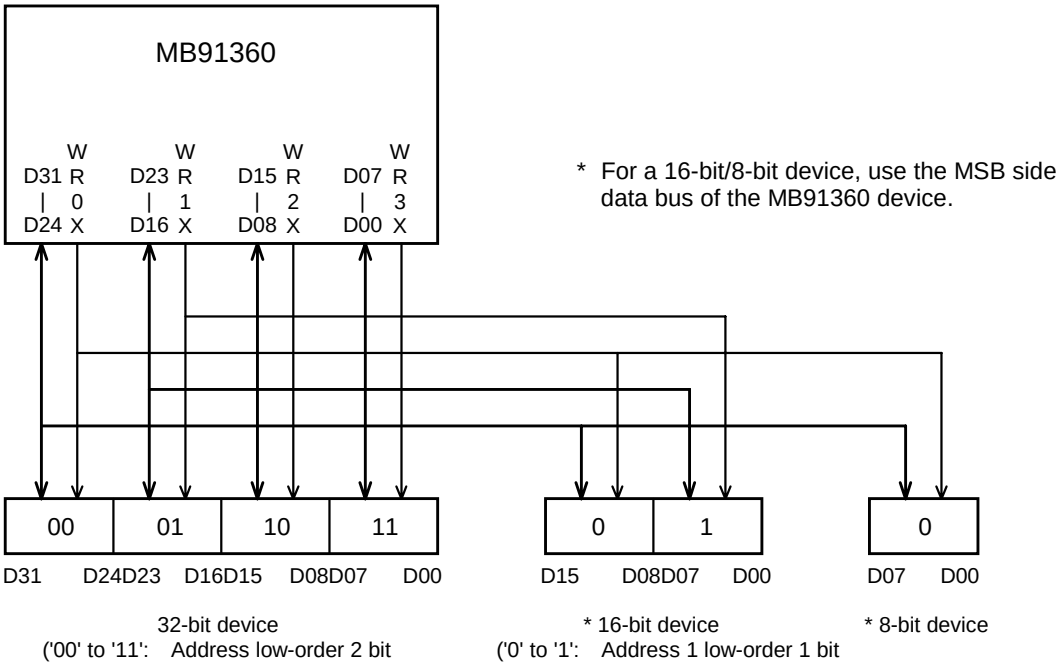
(c)PA1/PA0='10'
→ (1) Output A1/A0='10'

(d)PA1/PA0='11'
→ (1) Output A1/A0='11'



■ Typical connection to external device

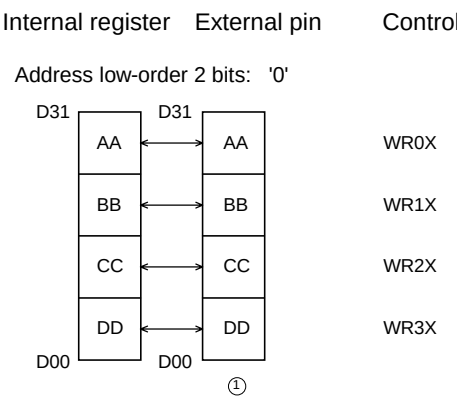
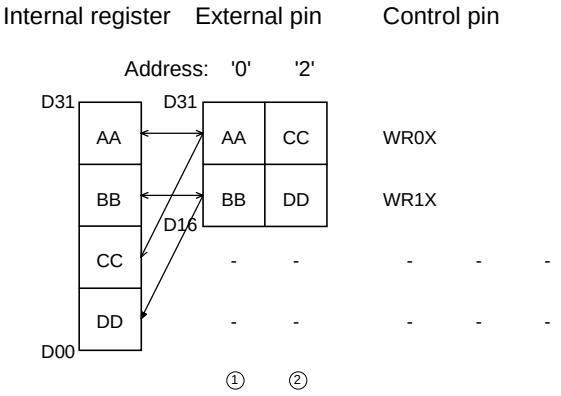
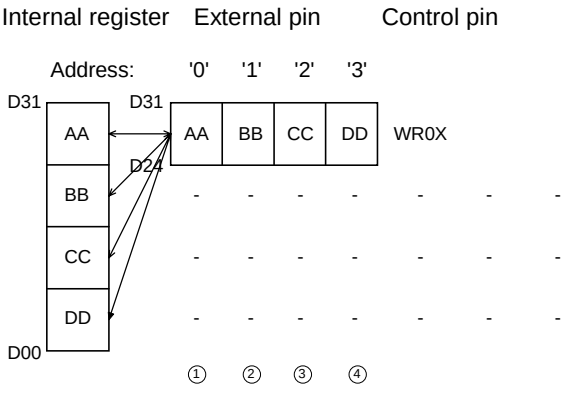
Figure 8.2.2 Typical connection to external device



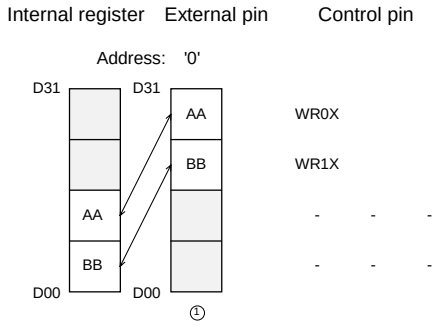
8.2.3 External Bus Access

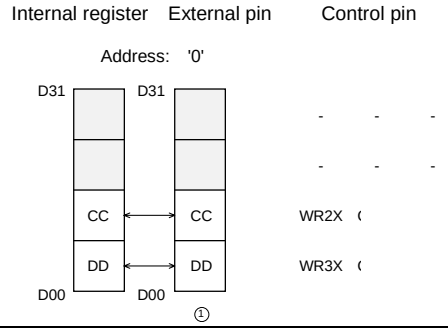
The following diagram explains in some more detail the bus access for different bus widths and word widths.

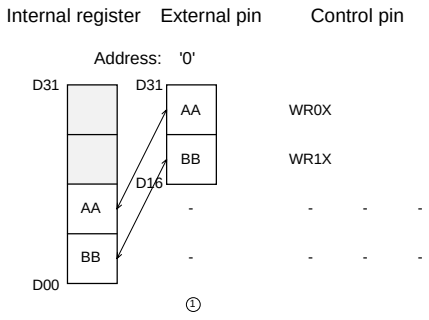
■ **Word access**

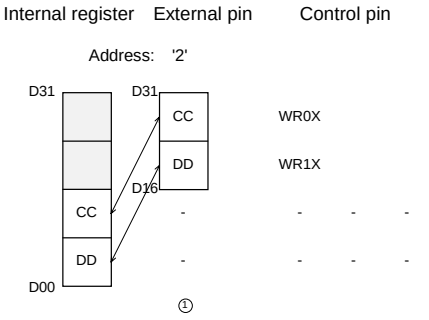
<table>
<tr><th></th><th>big endian mode</th></tr>
<tr><td>32-bit Bus width</td><td> 
</td></tr>
<tr><td>16-bit Bus width</td><td> 
</td></tr>
<tr><td>8-bit Bus width</td><td> 
</td></tr>
</table>

■ Half word access

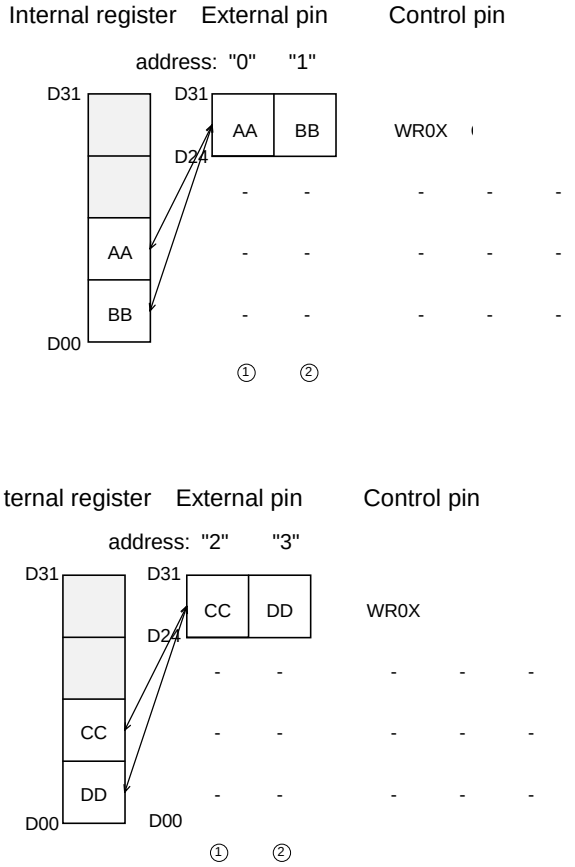
<table>
<tr><th></th><th>big endian mode</th></tr>
<tr><td>32-bit Bus width</td><td>




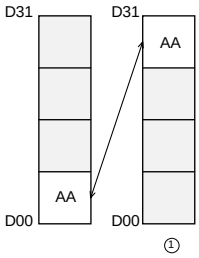
</td></tr>
<tr><td>16-bit Bus width</td><td>




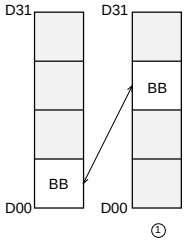
</td></tr>
</table>

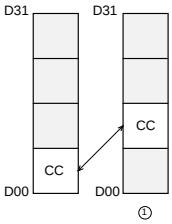
	big endian mode
8-bits Bus Width	

■ Byte access

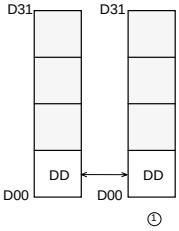
<table>
<tr><th></th><th>big endian mode</th></tr>
<tr><td rowspan="2">32-bit
Bus
Width</td><td>
Internal register External pin Control pin
address: "0"


WR0X
- - -
- - -
- - -

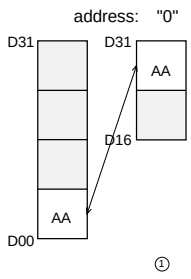
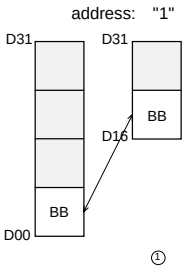
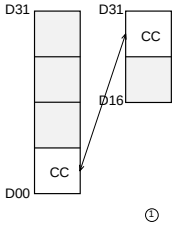
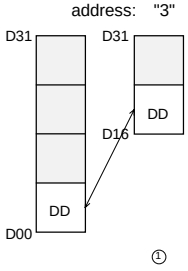
Internal register External pin Control pin
address: "1"


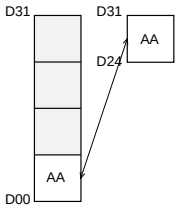
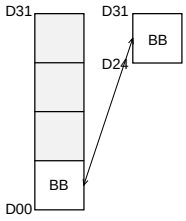
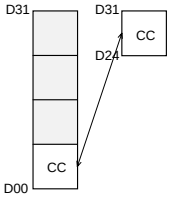
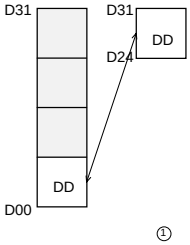
- - -
WR1X
- - -
- - -
</td></tr>
<tr><td>
Internal register External pin Control pin
address: "2"


- - -
- - -
WR2X
- - -

Internal register External pin Control pin
address: "3"


- - -
- - -
- - -
WR3X CAS3 WE3
</td></tr>
</table>

	big endian mode
16-bit Bus Width	Internal register External pin Control pin address: "0"  Internal register External pin Control pin address: "1" 
	Internal register External pin Control pin address: "2"  Internal register External pin Control pin address: "3" 

	big endian mode
8-bit Bus Width	Internal register External pin Control pin address: "0"  Internal register External pin Control pin address: "1" 
	Internal register External pin Control pin address: "2"  Internal register External pin Control pin address: "3" 

8.3 BUS SIGNALS

Table 8.3 Bus signals

Signal	Function	Related edge of CLKT	Read	Write	Ext. Bus Request	Note
A[31:0]	Address	rising edge	Output	Output	Z	
AH	Test Signal	n.a.	Output / 0	Output / 0	Output / 0	(1)
ALE	Bus Control	n.a.	Output / 0	Output / 0	Output / 0	(2)
AS	Address Strobe	rising edge	Output	Output	Output / 0	
BGRNTX	Bus Request Acknowledge	rising edge	Output / 1	Output / 1	Output	
BRQ	Bus Request	rising edge	Input	Input	Input	
CLK	Clock	rising/falling edge	Output	Output	Output	
CSnX	Chip Select	rising edge	Output	Output	Output / 1	
D[31:0]	Data	write: rising edge read: falling edge	Input	Output	Z	
RDX	Read	falling edge	Output	Output / 1	Z	
RDY	Ready	falling edge	Input	Input	Input	
WRX[3:0]	Write	falling edge	Output / 1	Output	Z	

Note: (1) Bus related function not implemented, can be used as IO port.

Note: (2) Bus control port for multiplex bus access (Address and data are using the same bus ports). Not implemented in MB91360 family.
Initial function: Port in input direction. After setting PFR9[1] output / 0.

8.4 BUS TIMING

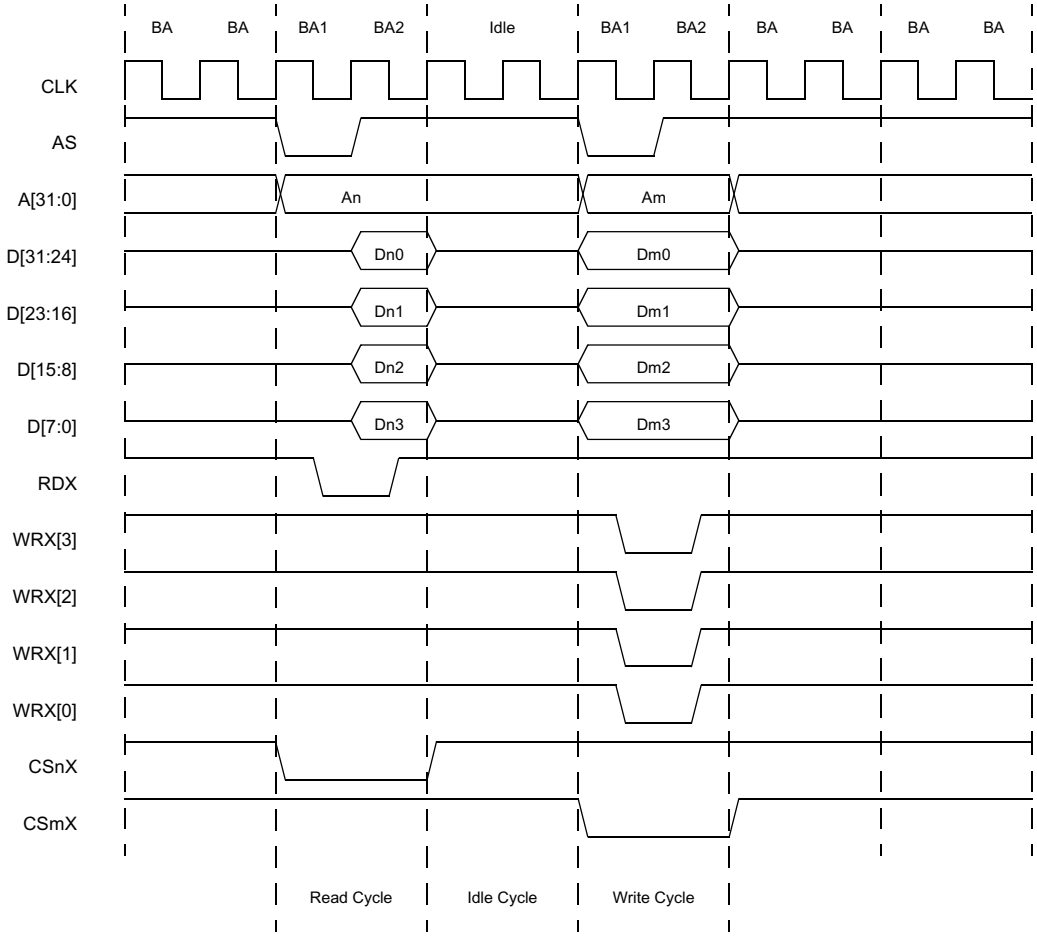
This section details the bus access operations in various modes. Please refer to section 34.8 "AC CHARACTERISTICS" on page 741 for detailed information about timing.

8.4.1 Normal Bus Access

When the normal bus interface is used, the basic bus cycle consists of ***two clocks*** at both read and write accesses.

Note: The term **BW** denotes the bus width. The term **access** used below refers to word/halfword/byte access for read/write operation.

■ **Basic read/write cycle**



- The **CLK** is generated by an internal clock (CLKT).
- AS outputs L only during the bus cycle start (BA1) period.
- The external **A31** to **A00** outputs the word/halfword/byte access starting byte position address at the beginning of the bus cycle.
- The **CS0X** to **CS7X** (area chip select) signal outputs are the same timing as the **A31** to **A00**.
- In the read cycle, the external **D31** to **D00** input is acquired at the rising edge of the external **RDX** output signal.
- In the read cycle, the external **D31** to **D00** input is entirely acquired according to the **RD** signal irrespective of the bus width or word/halfword/byte access, and the MB91360 internally checks whether the acquired data is valid.

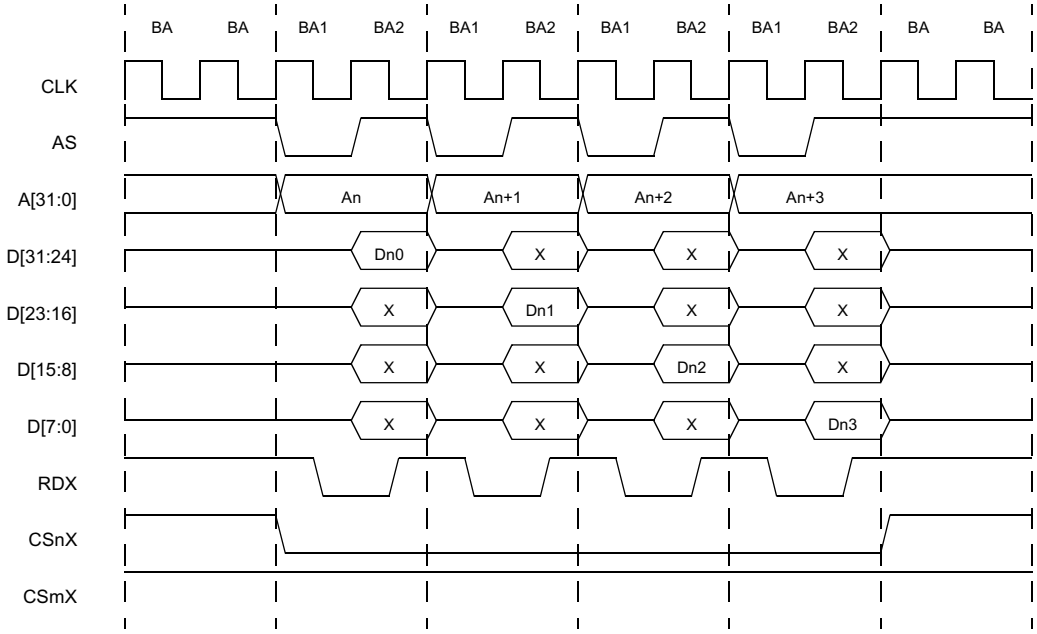
- In the write cycle, write data output generation starts at the beginning of the bus cycle (**BA1**), and the **WR0X**, **WR1X**, **WR2X** and **WR3X** control the control signals related to the individual data bus byte positions.

The examples on the following pages are based on CPU operation (CLKB) at 16 MHz and operation of the external bus interface (CLKT) at 4 MHz. Please refer also to section 8.4.4 "Idle Cycles" on page 268 for idle cycles which will be inserted depending on the clock ratio between CPU clock and bus interface clock and also depending on the operation sequence.

8.4.2 32 Bit Bus Width

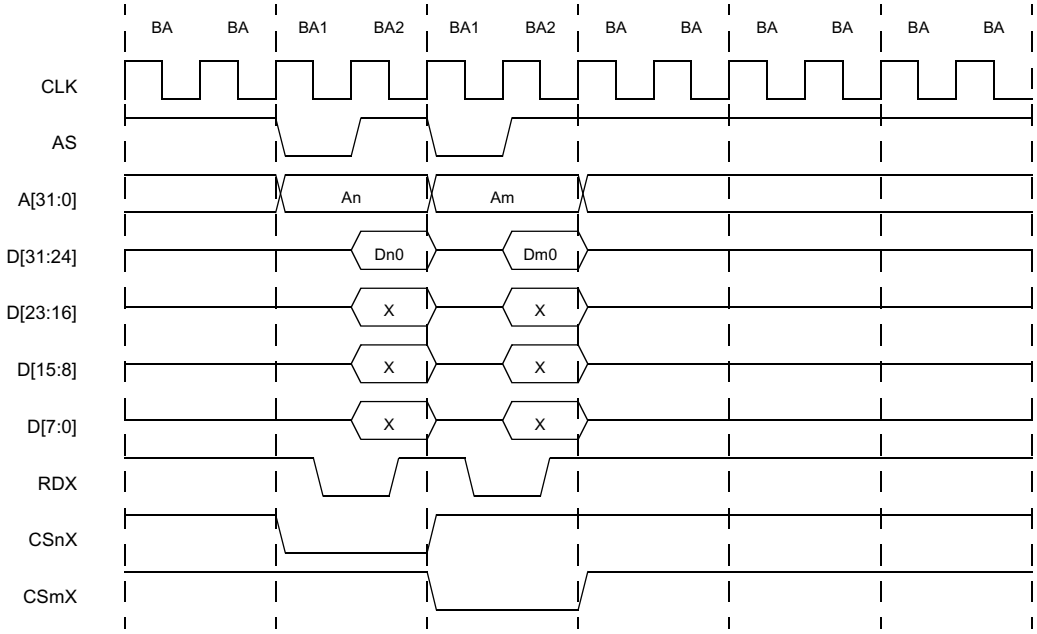
■ 8 Bit Access

CLKB/CLKT = 16/4 / External bus width 32 / Read 8 bit - Read 8 bit / Continues addresses

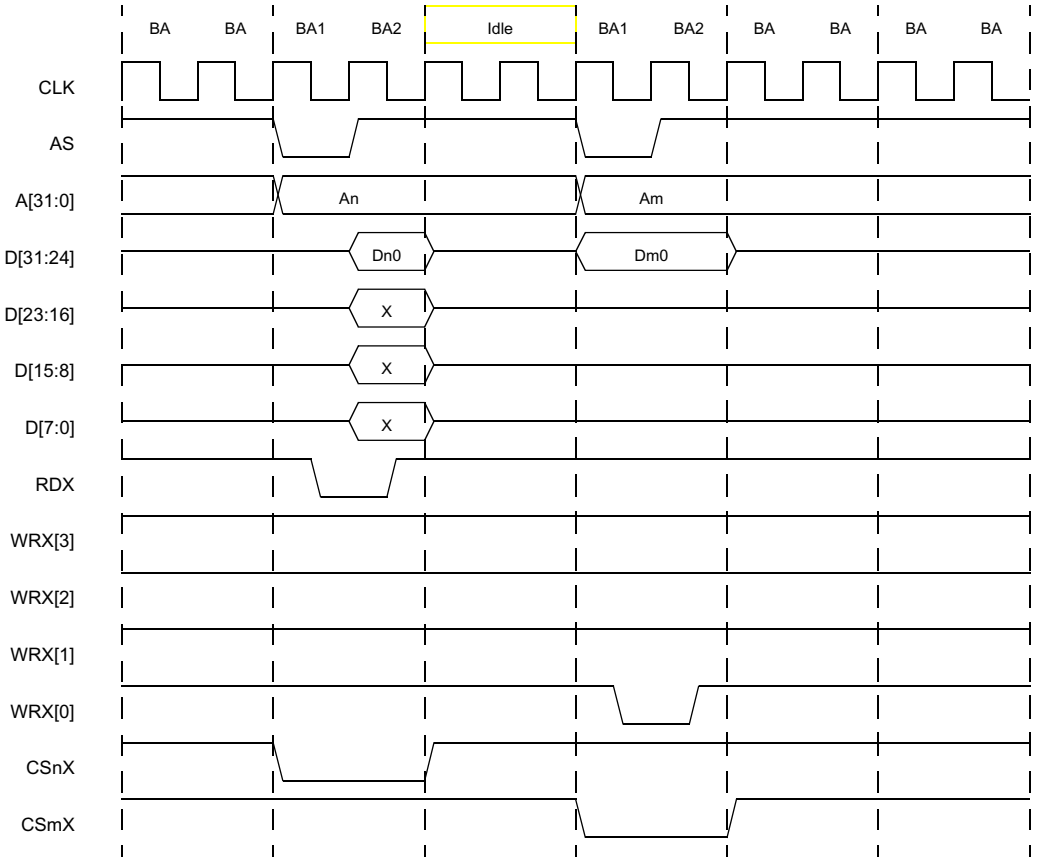


- For byte read access, high, mid high, mid low or low order 8 bits are used.

ClkB/ClkT = 16/4 / External bus width 32 / Read 8 bit - Read 8 bit / Different addresses

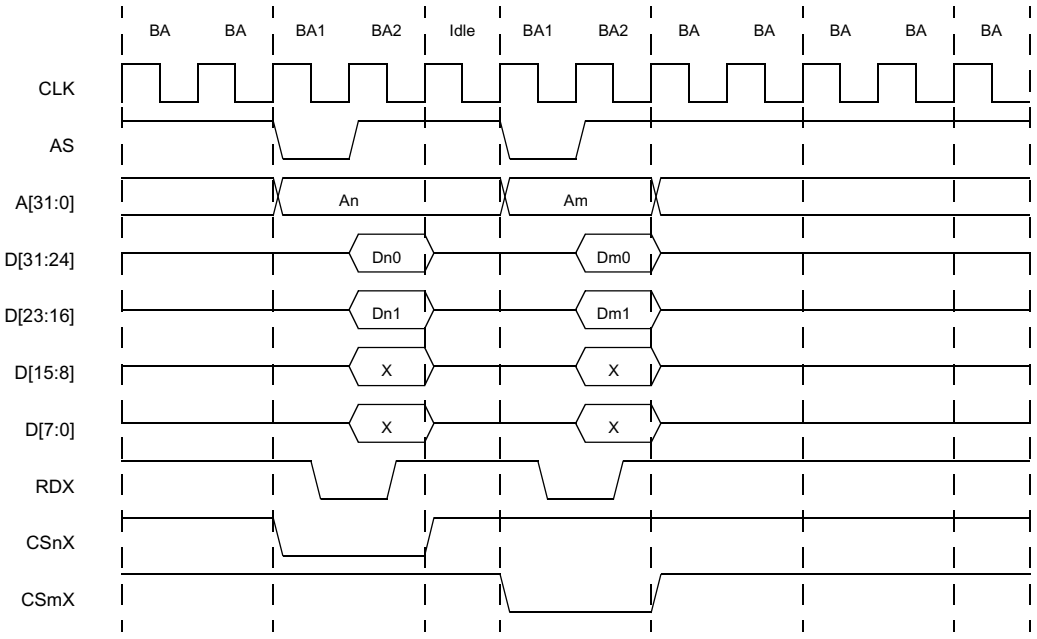


ClkB/ClkT = 16/4 / External bus width 32 / Read 8 bit - Write 8 bit / Different addresses



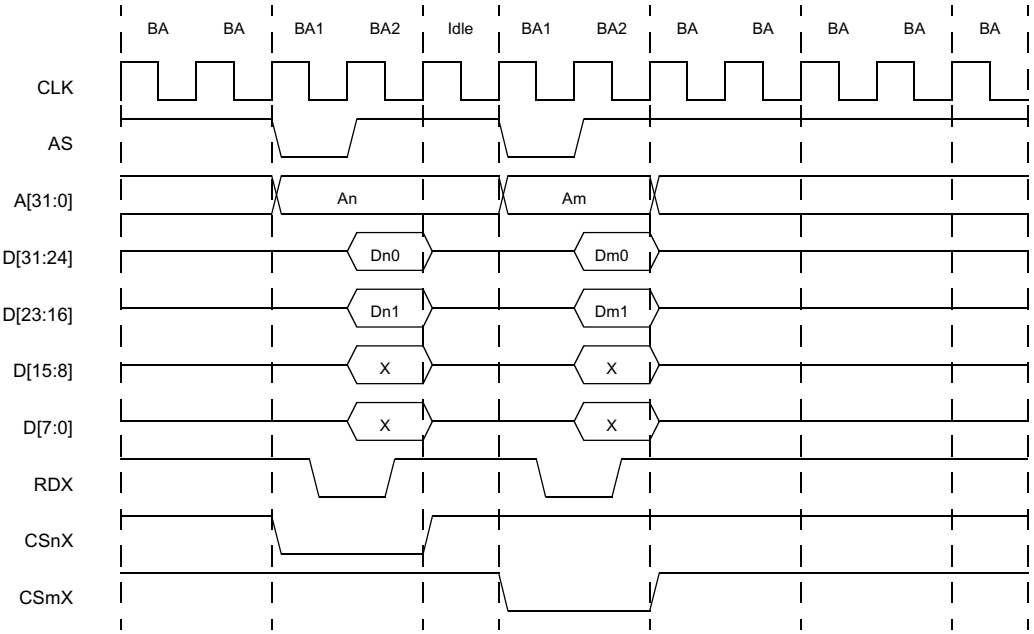
■ 16 Bit Access

ClkB/ClkT = 16/4 / External bus width 32 / Read 16 bit - Read 16 Bit / Continues addresses

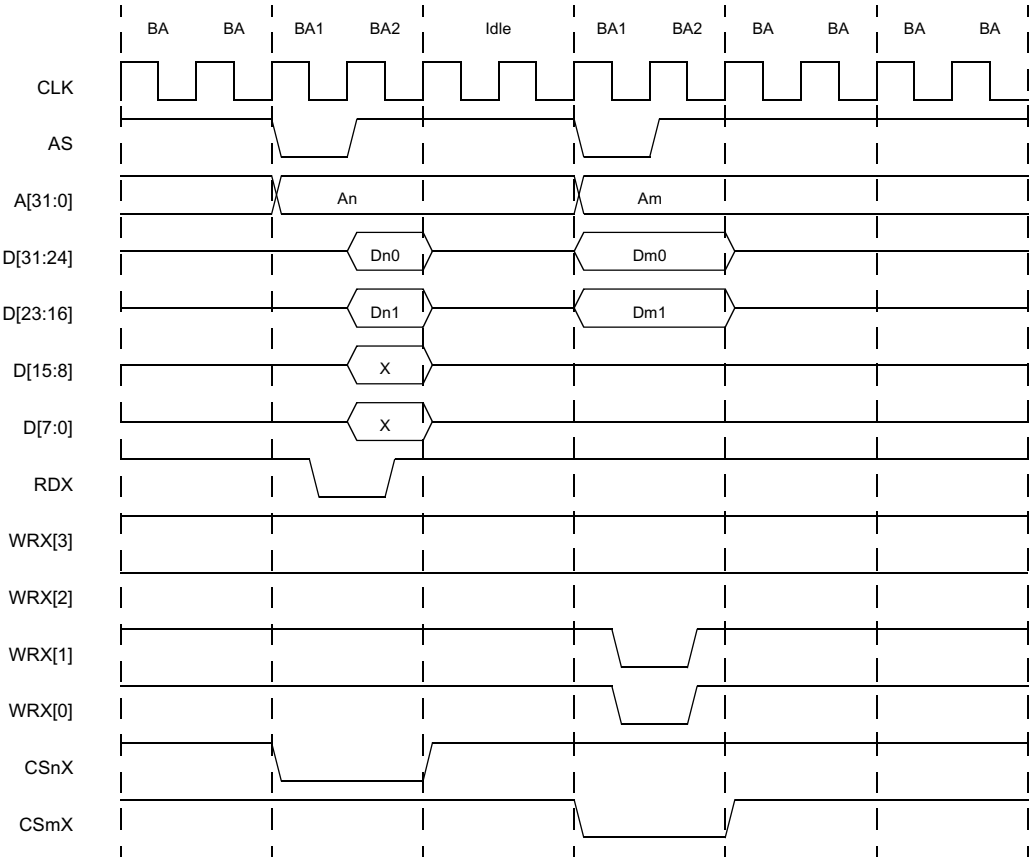


- For halfword read access, high or low order 16 bits are used.

ClkB/ClkT = 16/4 / External bus width 32 / Read 16 bit - Read 16 Bit / Different addresses



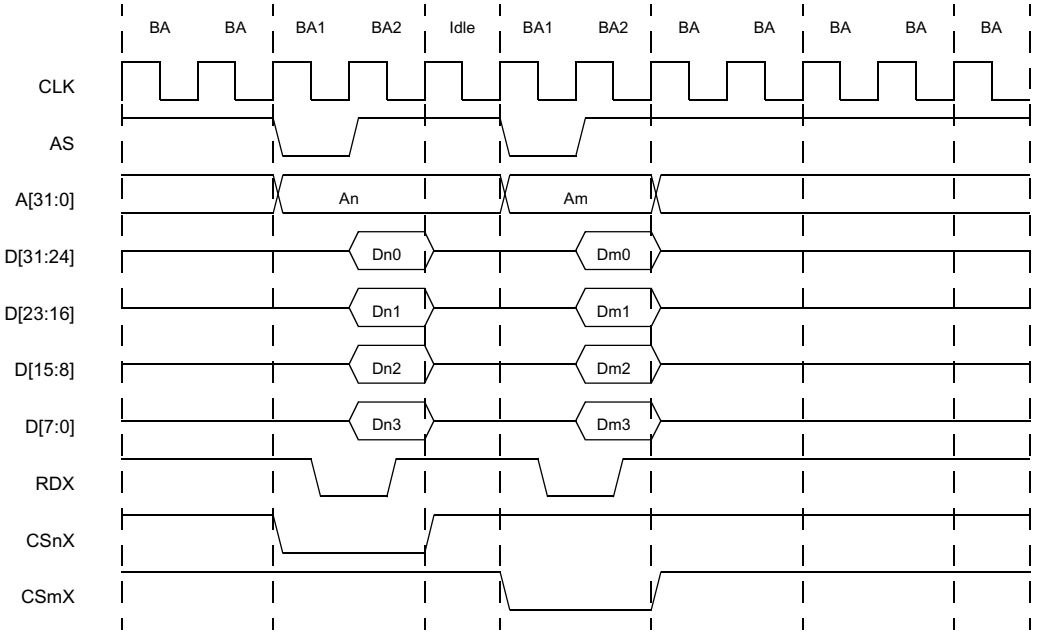
ClkB/ClkT = 16/4 / External bus width 32 / Read 16 bit - Write 16 bit / Different addresses



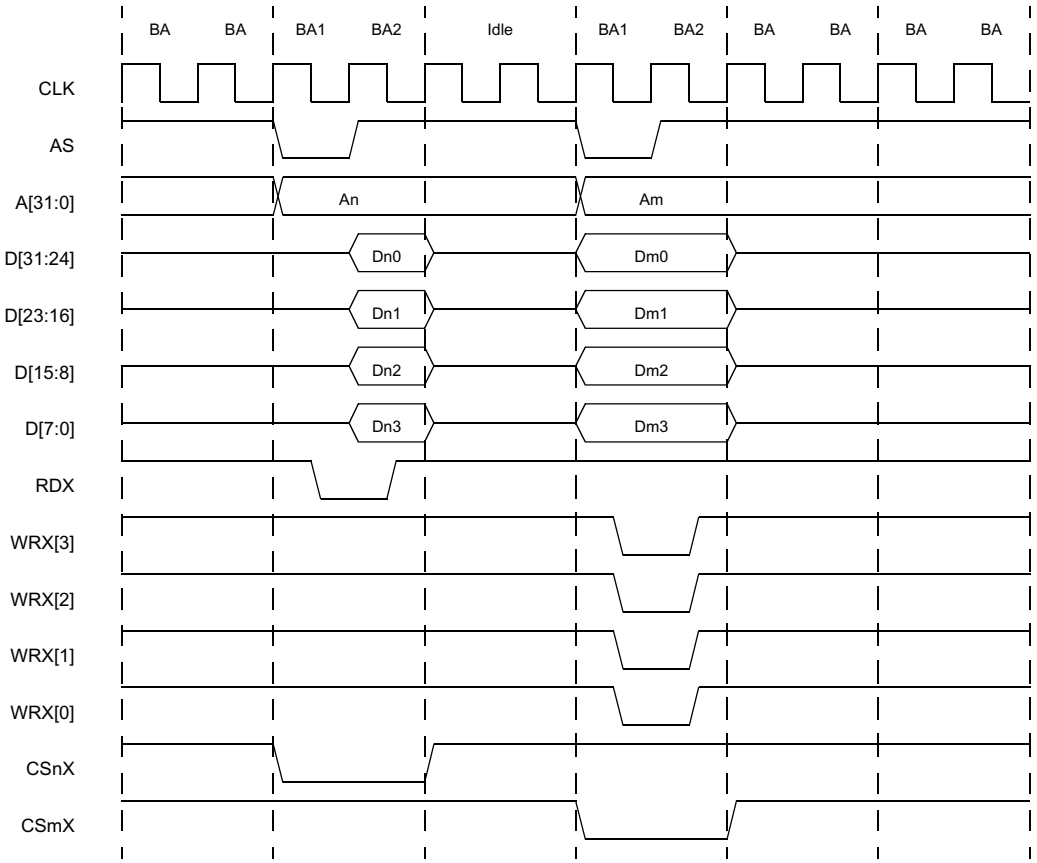
■ 32 Bit Access

- When the bus width is 32 bits, external D31 to D00 are used.

ClkB/ClkT = 16/4 / External bus width 32 / Read 32 bit - Read 32 bit / Different addresses



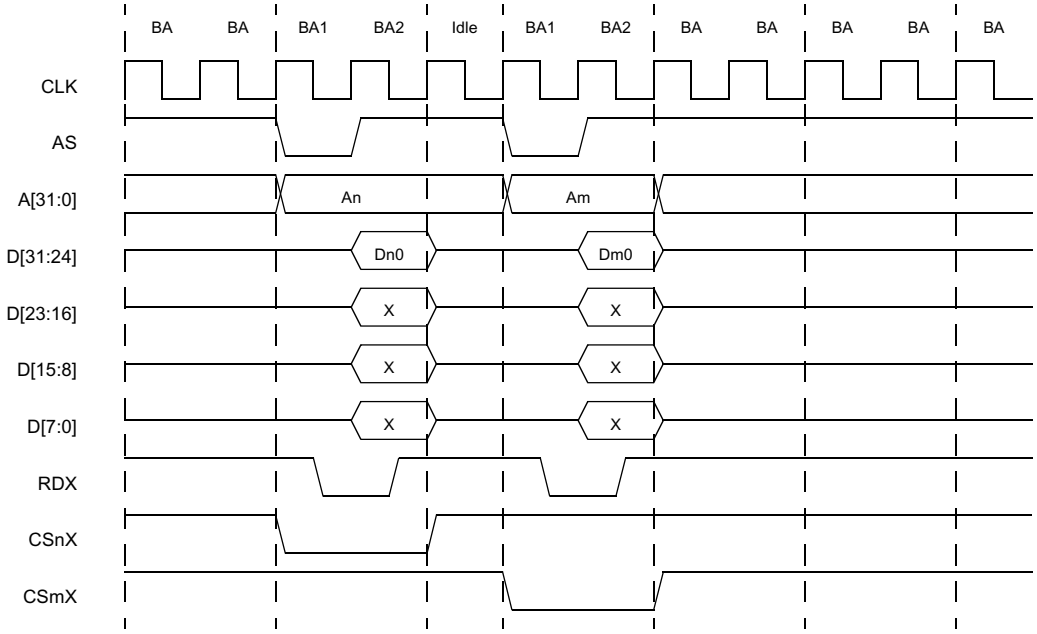
ClkB/ClkT = 16/4 / External bus width 32 / Read 32 bit - Write 32 bit / Different addresses



8.4.3 8 Bit Bus Width

■ **8 Bit Access**

ClkB/ClkT = 16/4 / External bus width 8 / Read 8 bit - Read 8 bit / Different addresses



- In the case of 8-/16-bit bus width, the unused data bus/**WRX** signal can be used for I/O port function.

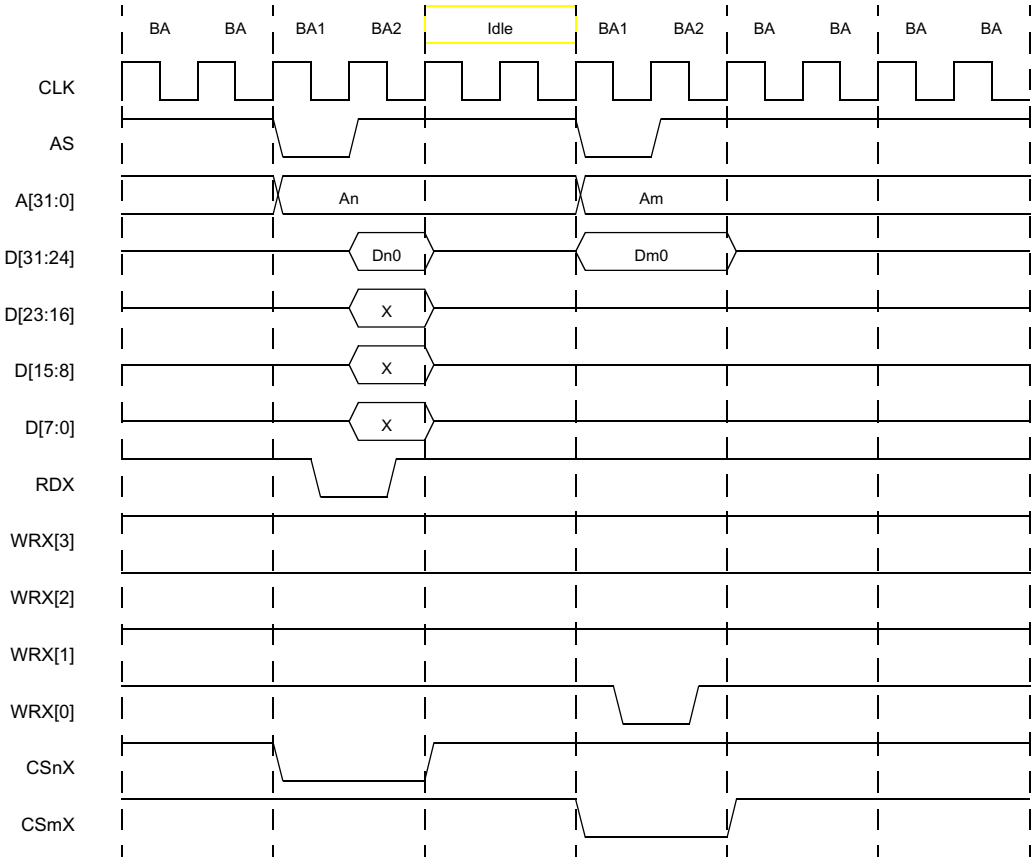
Pin Bus Width	D31-24 WR0X	D23-16 WR1X	D15-8 WR2X	D7-0 WR3X
32-bit	X	X	X	X
16-bit	X	X	O	O
8-bit	X	O	O	O

X: I/O port use disabled
O:I/O port use enabled

Note that either the data bus or **WR1X** to **WR3X** use as an I/O port depends on the maximum bus width of areas 0 to 5. For example, when the bus widths of areas 0 to 7 are all 8 bits, the **D23** to **D00** and **WR1X** to **WR3X** can be used as I/O ports. However, if only area 1 is 16 bits and the other areas are all 8 bits, the maximum bus width is 16 bits; therefore, the **D23** to **D16** and **WR1X** cannot be used as I/O ports even during 8-bit area accessing.

In this case, D15 to D00 and **WR2X/WR3X** can be used as I/O ports.

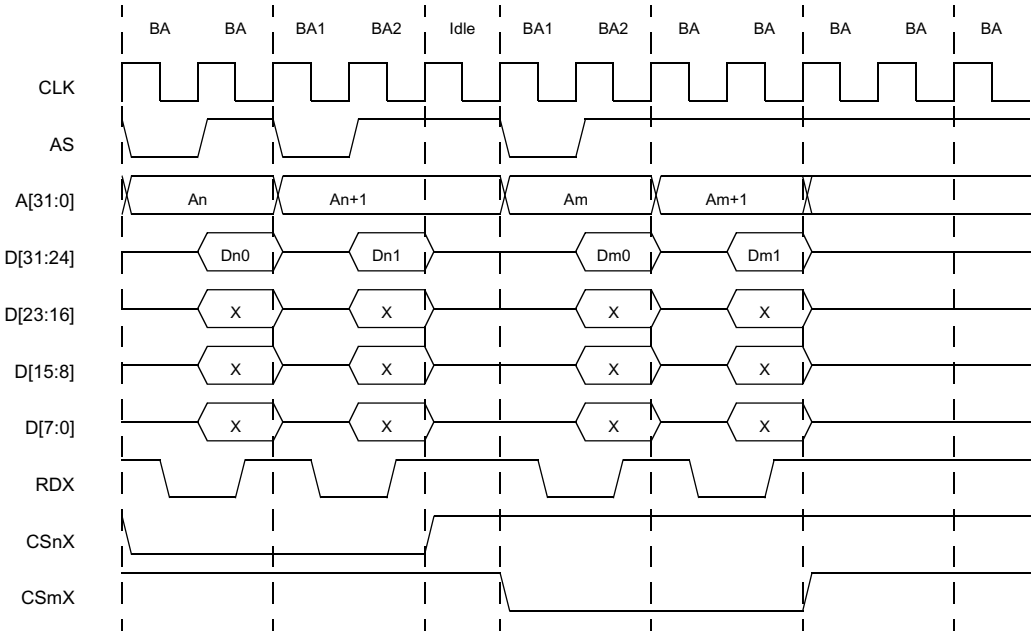
ClkB/ClkT = 16/4 / External bus width 8 / Read 8 bit - Write 8 bit / Different addresses



- When the bus width is 8 bits, the input values at low order D23 to D00 are ignored.

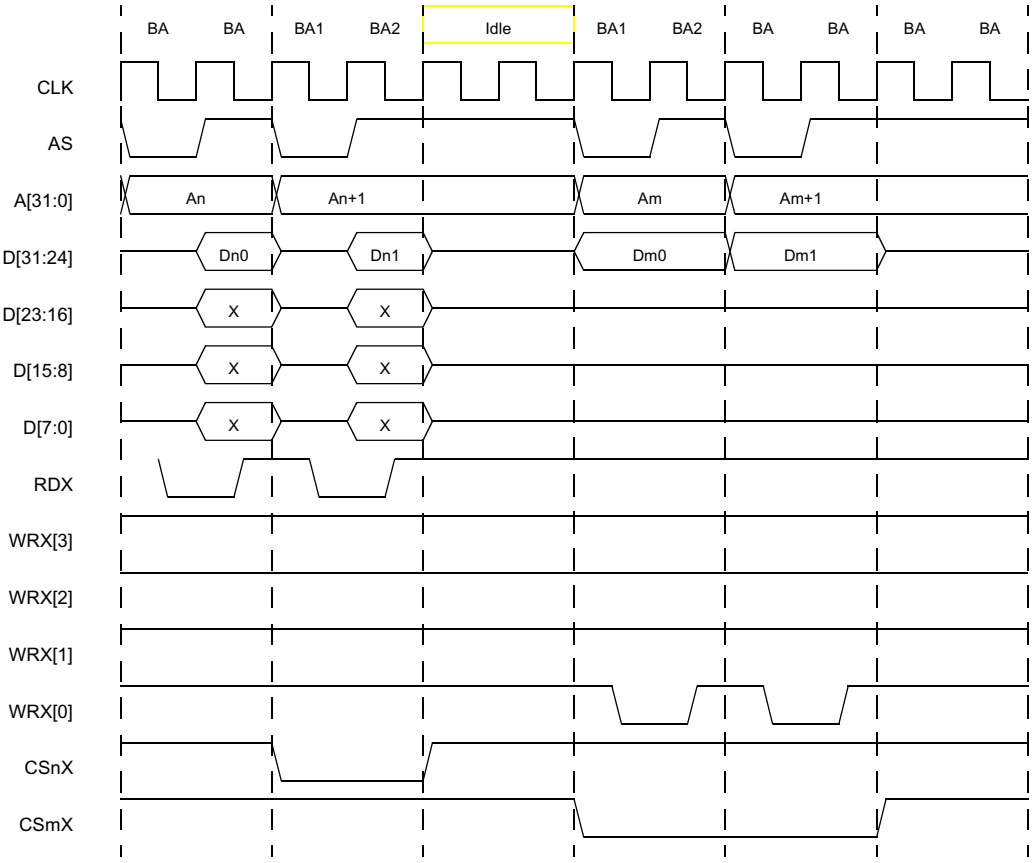
■ 16 Bit Access

ClkB/ClkT = 16/4 / External bus width 8 / Read 16 bit - Read 16 bit / Different addresses



- When the access size exceeds the data bus width, the addresses are accessed sequentially beginning with the **MSB**.

ClkB/ClkT = 16/4 / External bus width 8 / Read 16 bit - Write 16 bit / Different addresses



- When the bus width is 16 bits, the input values at low order D15 to D00 are ignored.

8.4.4 Idle Cycles

Idle cycles are caused by internal synchronization between CLKT and CLKB. The number of idle cycles depends on

- the clock ratio and
- the access type

The following table shows the minimum number idle cycles:

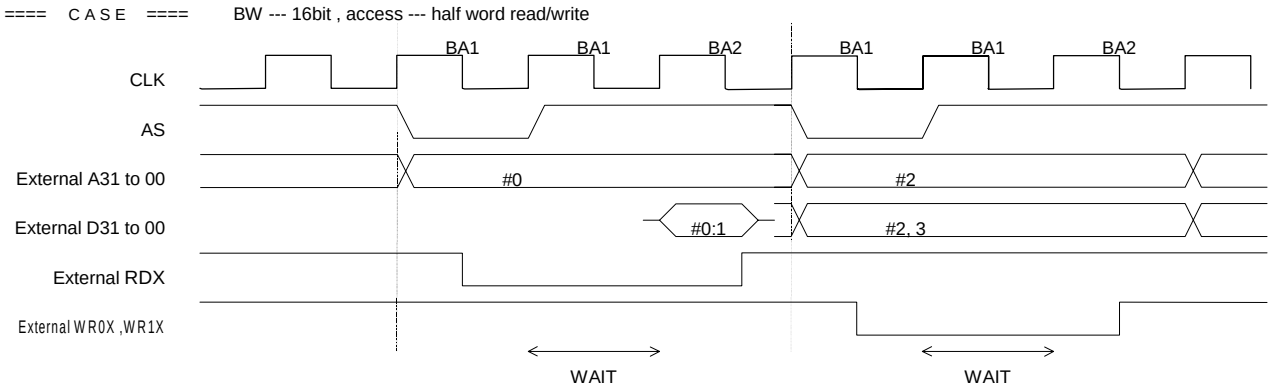
Table 8.4.4 Minimum number of idle cycles

CLKB:CLKT	read/read	read/write	write/*
1 : 1	1	2	0
1 : 2	1	1	0
1 : 3	0	1	0
1 : 4	0	1	0
1 : 8+	0	0	0

8.4.5 Wait Cycle Operation

There are two different wait cycles; automatic wait cycle provided by the **AMD WTC** and external wait cycle based on external RDY pin use.

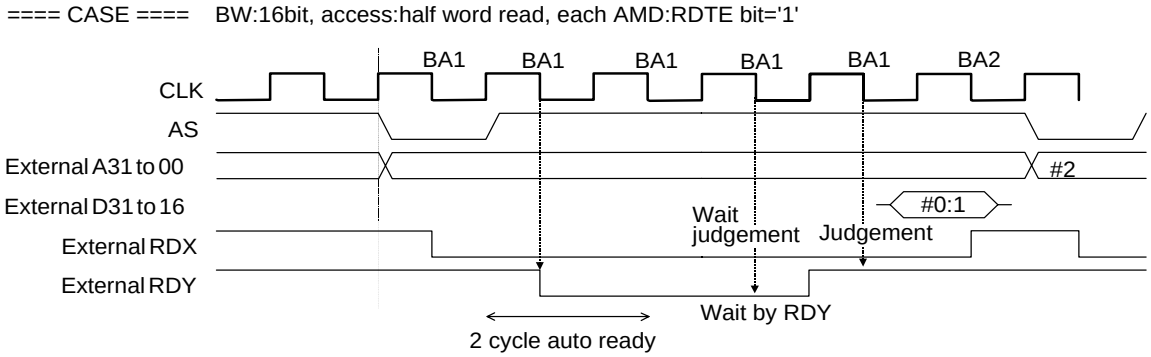
■ Automatic wait cycle



- The automatic wait cycle is achieved at setup of each **AMD WTC** bit.
- The above diagram shows a bus cycle where the **WTC** is **001**, in other words, one wait is inserted, and the bus cycle normally consists of a total of three clocks composed of two clocks and one wait cycle.

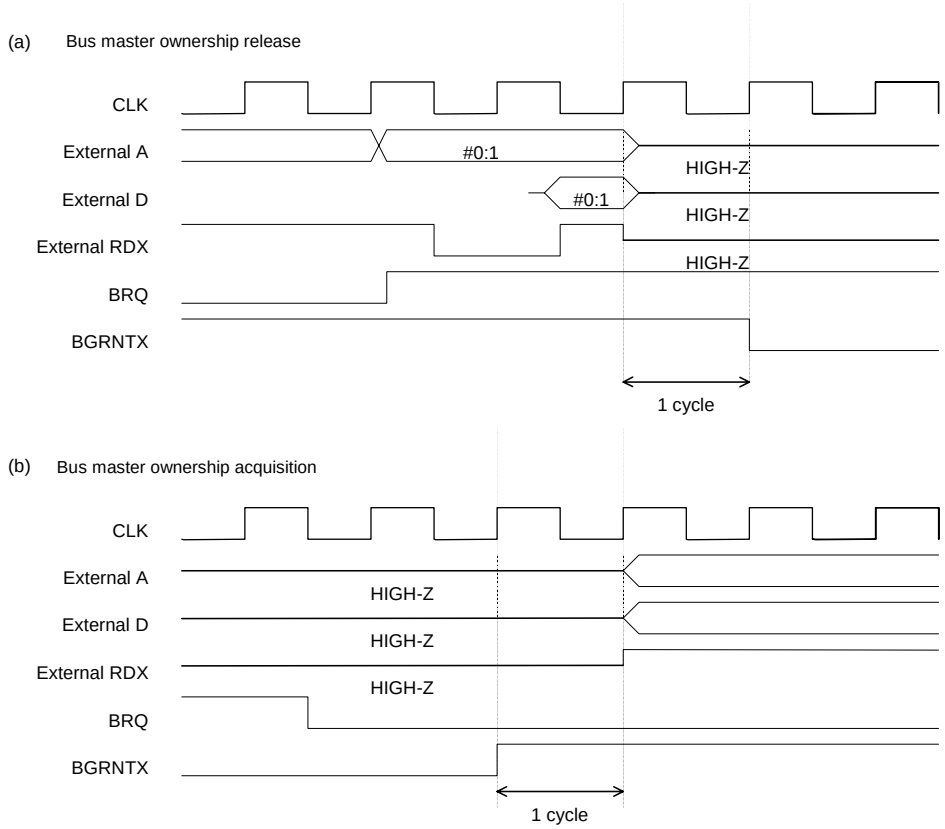
The automatic wait setup may consist of up to 7 clock cycles (a bus cycle consisting of 9 clock cycles).

■ External wait cycle



- The external wait cycle can be achieved by setting the each **AMD RDYE** bit to **1** and enabling the external **RDY** pin input.
- When using the external **RDY**, always set an automatic wait cycle consisting of two or more clocks.

8.4.6 External Bus Request



- When the **PFR8 P82** bit* is set to **1**, bus arbitration can be conducted by the **BRQ** and **BGRNTX**.
- At bus master ownership release, the pin is set to High-Z, and the **BGRNTX** is asserted one cycle later.
- At bus master ownership acquisition, the **BGRNTX** is negated, and each pin is rendered active one clock later.

*: See the chapter 7 "I/O PORTS" on page 211.

8.5 USING THE BUS INTERFACE AS GENERAL I/O PORTS

8.5.1 Introduction

This application note describes how to use the I/O pins of the external bus interface as general purpose I/O ports ("Port Mode"). It is applicable for the 32-bit microcontroller devices of MB91360 series: MB91FV360GA, MB91F362GB and MB91F369GA.

If the devices are initialized to sigle chip mode (MD2=0, MD1=0, MD0=0), then the mode register MODR is set to the fixed vector 0x06 (internal ROM, 32 bit external interface). It is not possible to change the content of MODR by software. Please refer also to sections 2.10.2 "Mode Setting" on page 126 and 2.10.3 "Fixed Vector" on page 127.

This section describes two software workarounds to use parts of the external interface as general ports:

- Don’t use CAN and configure 7 ports (48 pins on MB91F362GB) to general I/O function,
- Use CAN and configure 4 ports (32 pins on MB91F362GB) to general I/O function.

8.5.2 Precausions

For both software workarounds, the states of the external interface during power-up should be considered, because the external interface becomes active during boot sequence:

■ Oscillation Stabilisation Time

After initialisation reset, the device counts the oscillation stabilisation time (32 ms). All external interface ports are in Hi-Z state for this time.

■ Boot ROM Execution

At the end of the oscillation stabilisation time, the fixed mode vector and reset vector are fetched. This enables the external interface before starting Boot ROM software execution.

The external interface ports have the following values there:

Table 8.5.2 External interface during boot sequence

I/O Pin	Input / Output	Value
A[21:0]	output	0x0FFFF8
D[31:0]	input	Hi-Z
CS6X, CS5X, CS4X, CS3X, CS2X, CS1X, CS0X	output	1
RDX, WR3X, WR2X, WR1X, WR0X	output	1
CLK	output	CLKT pulse
ALE, AS, AH, BRQ, BGRNTX, RDY	input	Hi-Z

If the Boot ROM software finds an valid entry in the flash security vector, the jump to flash entry address is performed. With a clock frequency of 4 MHz at X0, it takes 144 µs from Boot ROM start until flash software start. To keep the active state of external interface short, it is recommended to place the following software workarounds at the beginning of the flash program.

8.5.3 Used Registers

The following registers are used to access the external interface ports:

Table 8.5.3 External Interface Port Registers

Address	Register				Block
	+0	+1	+2	+3	
000000H	PDR0 [R/W] XXXX XXXX	PDR1 [R/W] XXXX XXXX	PDR2 [R/W] XXXX XXXX	PDR3 [R/W] XXXX XXXX	Port Data Register
000004H	PDR4 [R/W] XXXX XXXX	PDR5 [R/W] XXXX XXXX	PDR6 [R/W] - - - X XXXX	PDR7 [R/W] - 111 - - - -	
000008H	PDR8 [R/W] XXXX XXXX	PDR9 [R/W] XXXX XXX1	--	PDRB [R/W] - - - - - - XXX	
000600H	DDR0 [R/W] 0000 0000	DDR1 [R/W] 0000 0000	DDR2 [R/W] 0000 0000	DDR3 [R/W] 0000 0000	Data Direction Register
000604H	DDR4 [R/W] 0000 0000	DDR5 [R/W] 0000 0000	DDR6 [R/W] 0000 0000	DDR7 [R/W] 0000 0000	
000608H	DDR8 [R/W] 0000 0000	DDR9 [R/W] 0000 0000	--	DDRB [R/W] 0000 0000	
000610H	PFR0 [R/W] 0000 0000	PFR1 [R/W] 0000 0000	PFR2 [R/W] 0000 0000	PFR3 [R/W] 0000 0000	Port Function Register
000614H	PFR4 [R/W] 0000 0000	PFR5 [R/W] 0000 0000	PFR6 [R/W] 1111 1111	PFR7 [R/W] 0000 1111	
000618H	PFR8 [R/W] 1111 10-0	PFR9 [R/W] 1111 0101	--	PFRB [R/W] 0000 0000	
000624H	--	--	--	PFR27 [R/W] 1111 -00-	

The port data registers (PDR), data direction registers (DDR) and port function registers (PFR) work the same way as described for the normal I/O ports in section 7.1 "I/O PORTS AND REGISTER CONFIGURATION" on page 212ff.

The registers of ports 7, 8, 9 and B are explained in sections 7.1 and 7.2.

8.5.4 Software Workaround, if CAN is not used

■ **Configurable Ports**

If the CAN(s) are not used, it is possible to configure 7 Ports with 48 pins (MB91F362GB) to general port mode:

Table 8.5.4 External interface in port mode if CAN is not used

I/O Pins	Port	Comment
A[20:16]	P6[4:0]	can be set to port mode
A[15:0]	P5[7:0] P4[7:0]	remains address output
D[31:24]	P3[7:0]	remains bidirectional data bus
D[23:0]	P2[7:0] P1[7:0] P0[7:0]	can be set to port mode
CS6X, CS5X, CS4X	P7[6:4]	can be set to port mode
WR3X, WR2X, WR1X, WR0X, RDX, BRQ, BGRNTX, RDY	P8[7:3] P8[2:0]	can be set to port mode
CS3X, CS2X, CS1X, CS0X, AH, CLK, ALE, AS	P9[7:4] P9[3:0]	can be set to port mode

■ **Setup sequence**

It is recommended to place this setup at the beginning of the flash software.

- Set CLKB & CLKP & CLKT & CLKS to 1/1 (don’t start PLL here)
- Disable CAN clock (was enabled by boot ROM)
- Disable CS7 (CAN) and CS0 in CSE register
- Re-configure external interface to 8 bit / 0 wait states for CS0 and CS7 area
- Clear the port function registers for Port 0 ... Port 9 (has no effect to Port 3, 4, 5)
- Start Your application

■ **Assembler example**

```
;;; ***************************************************************************
            .section main,code,locate=_PC_ADDRESS;; 0x00F4000 (in Flash)

;; --------------- manual startup... ---------------
// ** set CLKB & CLKP & CLKT & CLKS to 1/1 (no PLL here)
            ldi:20 #0x0000,r1       // CLK[BPTS]=1/1
            ldi:32 #_DIVR0,r12      // set DIVR address
            sth    r1,@r12          //

_MAIN00:
```

```
; ########################################################################
; Setup ext. I/F to port mode:
; ########################################################################

; ------ Disable CAN clock (was enabled by boot ROM) -----------
            LDI    #_CMCR,R12
            LDI    #0x000, R0       ; disable CAN clock (CPU/2)
            STH    R0,@R12
; ------ Disable CS7 (CAN) and CS0 ----------------------------------
            LDI    #_CSE,R12        ; Chip select enable register
            LDI    #0x00,R1         ; disable CS7, CS0
            STB    R1,@R12

;; -------- ext. I/F to 8 bit -------------------------------------
            LDI    #_AMD0,R12       ; Area Mode reg   AMD0
;;          LDI    #0x11,R1         ; 32-Bit Bus, 1 wait state, no ready-pin
;;                                  ;                          (done by BootROM)
            LDI    #0x00,R1         ;  8-Bit Bus, 0 wait state, no ready-pin
            STB    R1,@R12

            LDI    #_AMD7,R12       ; Area Mode reg   AMD7 (CAN)
;;          LDI    #0x29,R1         ; 16-Bit Bus, 1 wait state, ready-pin enabled
;;                                  ;                               (done by BootROM)
            LDI    #0x00,R1         ;  8-Bit Bus, 0 wait state, no ready-pin
            STB    R1,@R12

;; -------- Clear all PFR's ----------------------------------------
            LDI    #0x0000,R0       ;; clear all PFR's
            LDI    #_PFR27,R1       ;;
            stb    R0,@R1           ;; Port 7 PFR2, Addr=0x0627
            LDI    #_PFR0,R14       ;;
            stb    r0,@(r14,0x0)    ;; Port 0 PFR, Addr=0x0610
            stb    r0,@(r14,0x1)    ;; Port 1 PFR, Addr=0x0611
            stb    r0,@(r14,0x2)    ;; Port 2 PFR, Addr=0x0612
            stb    r0,@(r14,0x3)    ;; Port 3 PFR, Addr=0x0613
            stb    r0,@(r14,0x4)    ;; Port 4 PFR, Addr=0x0614
            stb    r0,@(r14,0x5)    ;; Port 5 PFR, Addr=0x0615
            stb    r0,@(r14,0x6)    ;; Port 6 PFR, Addr=0x0616
            stb    r0,@(r14,0x7)    ;; Port 7 PFR, Addr=0x0617
            stb    r0,@(r14,0x8)    ;; Port 8 PFR, Addr=0x0618
            stb    r0,@(r14,0x9)    ;; Port 9 PFR, Addr=0x0619


Start_Application:

;; ========== Start Your application here...
```

8.5.5 Software Workaround, if CAN is used

■ **Configurable Ports**

If the CAN(s) will be used, it is possible to configure 5 Ports with 33 pins to general port mode:

Table 8.5.5 External interface in port mode if CAN is to be used

I/O Pins	Port	Comment
A[20:16]	P6[4:0]	can be set to port mode
A[15:0]	P5[7:0] P4[7:0]	remains address output
D[31:16]	P3[7:0] P2[7:0]	remains bidirectional data bus (16 bit needed for CAN at ext. I/F)
D[15:0]	P1[7:0] P0[7:0]	can be set to port mode
CS6X, CS5X, CS4X	P7[6:4]	can be set to port mode
WR3X, WR2X, WR1X, WR0X, RDX BRQ, BGRNTX RDY	P8[7:3] P8[2:1] P8[0]	remain WRX[3:0], RDX output; can be set to port mode; remains RDY input
CS3X, CS2X, CS1X, CS0X AH CLK ALE, AS	P9[7:4] P9[3] P9[2] P9[1:0]	can be set to port mode; can be set to port mode; CLK remains CLKT output; can be set to port mode;

■ **Setup sequence**

It is recommended to place this setup at the beginning of the flash software.

- Set CLKB & CLKP & CLKT & CLKS to 1/1 (don’t start PLL here)
- Disable CS0 in CSE register
- Re-configure external interface to 16 bit / 0 wait states for CS0 area
- Clear the port function registers for Port 0 ... Port 7 (has no effect to Port 2, 3, 4, 5) and the usable bits in the PFR of Ports 8 / 9.
- Start Your application

■ **Assembler example**

```
;;; ************************************************************************
            .section main,code,locate=_PC_ADDRESS;; 0x00F4000 (in Flash)

;; --------------- manual startup... ---------------
// ** set CLKB & CLKP & CLKT & CLKS to 1/1 (no PLL here)
            ldi:20 #0x0000,r1       // CLK[BPTS]=1/1
            ldi:32 #_DIVR0,r12      // set DIVR address
            sth    r1,@r12          //
_MAIN00:
; ######################################################################
; Setup ext. I/F to port mode:
; ######################################################################
; ------ DON'T Disable CAN clock (was enabled by boot ROM) -----------

; ------ Disable  CS0 ---------------------------------
            LDI    #_CSE,R12            ; Chip select enable register
            LDI    #0x80,R1             ; disable CS0, CS7 is for CAN
            STB    R1,@R12

;; -------- ext. I/F to 16bit (8 bit makes no sense) -----------------

            LDI    #_AMD0,R12       ; Area Mode reg   AMD0
;;          LDI    #0x11,R1         ; 32-Bit Bus, 1 wait state, no ready-pin
;;                                  ;                    (done by BootROM)
            LDI    #0x08,R1       ; 16-Bit Bus, 0 wait state, no ready-pin
            STB    R1,@R12

;; ----- DON'T change AMD7 mode (is prepared by BootROM) -------------

;; -------- Clear all PFR's ----------------------------------------

            LDI    #0x0000,R0       ;; clear all PFR's
            LDI    #_PFR27,R1       ;;
            stb    R0,@R1           ;; Port 7 PFR2, Addr=0x0627
            LDI    #_PFR0,R14       ;;
            stb    r0,@(r14,0x0)    ;; Port 0 PFR, Addr=0x0610
            stb    r0,@(r14,0x1)    ;; Port 1 PFR, Addr=0x0611
            stb    r0,@(r14,0x2)    ;; Port 2 PFR, Addr=0x0612
            stb    r0,@(r14,0x3)    ;; Port 3 PFR, Addr=0x0613
            stb    r0,@(r14,0x4)    ;; Port 4 PFR, Addr=0x0614
            stb    r0,@(r14,0x5)    ;; Port 5 PFR, Addr=0x0615
            stb    r0,@(r14,0x6)    ;; Port 6 PFR, Addr=0x0616
            stb    r0,@(r14,0x7)    ;; Port 7 PFR, Addr=0x0617

            ldi    #0xF9,R0         ;; PFR8: WRX[3:0],RDX,RDY needed.
            stb    r0,@(r14,0x8)    ;; Port 8 PFR, Addr=0x0618

            ldi    #0x04,R0         ;; PFR9: CLK needed.
            stb    r0,@(r14,0x9)    ;; Port 9 PFR, Addr=0x0619


Start_Application:
;; ========== Start Your application here...
```

CHAPTER 9 INTERRUPT CONTROLLER

This Chapter provides an overview of the interrupt controller, describes the register structure and functions, and describes the interrupt controller operation

9.1 OVERVIEW

An interrupt controller controls interrupt acceptance and arbitration processing.

■ Hardware configuration

This module consists of the following:

- ICR register
- Interrupt priority evaluation circuit
- Interrupt level and interrupt number (vector) generator
- Hold request cancel request generator

■ Major functions

This module has the following major functions:

- Detecting an NMI request or interrupt request
- Priority evaluation (using the level or number)
- Transferring the level of the interrupt cause in the evaluation result (to the CPU)
- Transferring the number of the interrupt cause in the evaluation result (to the CPU)
- Instructing recovery from stop mode due to an NMI or interrupt level other than 11111 (to the CPU)
- Generating a hold request cancel request for the bus master

9.2 LIST OF REGISTERS

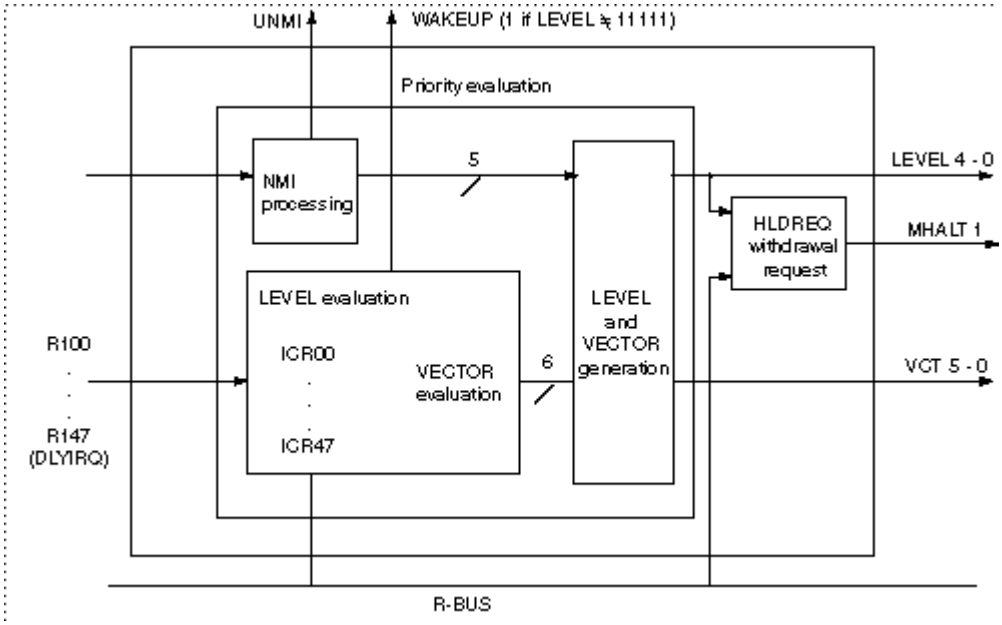
	bit	7	6	5	4	3	2	1	0	
Address: 00000440	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR00
Address: 00000441	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR01
Address: 00000442	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR02
Address: 00000443	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR03
Address: 00000444	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR04
Address: 00000445	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR05
Address: 00000446	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR06
Address: 00000447	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR07
Address: 00000448	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR08
Address: 00000449	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR09
Address: 0000044A	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR10
Address: 0000044B	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR11
Address: 0000044C	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR12
Address: 0000044D	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR13
Address: 0000044E	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR14
Address: 0000044F	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR15
Address: 00000450	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR16
Address: 00000451	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR17
Address: 00000452	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR18
Address: 00000453	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR19
Address: 00000454	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR20
Address: 00000455	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR21
Address: 00000456	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR22
Address: 00000457	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR23
Address: 00000458	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR24
Address: 00000459	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR25
Address: 0000045A	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR26
Address: 0000045B	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR27
Address: 0000045C	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR28
Address: 0000045D	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR29
Address: 0000045E	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR30
Address: 0000045F	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR31
					R	R/W	R/W	R/W	R/W	

	bit	7	6	5	4	3	2	1	0	
Address: 00000460	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR32
Address: 00000461	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR33
Address: 00000462	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR34
Address: 00000463	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR35
Address: 00000464	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR36
Address: 00000465	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR37
Address: 00000466	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR38
Address: 00000467	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR39
Address: 00000468	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR40
Address: 00000469	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR41
Address: 0000046A	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR42
Address: 0000046B	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR43
Address: 0000046C	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR44
Address: 0000046D	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR45
Address: 0000046E	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR46
Address: 0000046F	H	–	–	–	ICR4	ICR3	ICR2	ICR1	ICR0	ICR47
					R	R/W	R/W	R/W	R/W	

Address: 00000045	H	MHALTI	–	–	LVL4	LVL3	LVL2	LVL1	LVL0	HRCL
		R/W			R	R/W	R/W	R/W	R/W	

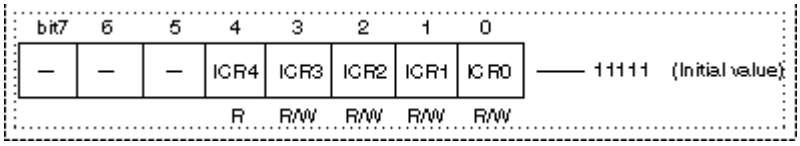
9.3 BLOCK DIAGRAM

Figure 9.3 Interrupt Controller Block diagram



9.4 DETAILED EXPLANATION OF REGISTERS

9.4.1 ICR (Interrupt Control Register)



This section explains the interrupt control registers. One interrupt control register is provided for each interrupt input and defines the interrupt level of the corresponding interrupt request.

[bits4 to 0] ICR4 to 0

Each of these interrupt level setting bits specifies the interrupt level of the corresponding interrupt request.

If the interrupt level defined in this register is greater than the level mask value that you have defined (or that is predefined) in the ILM register of the CPU, the interrupt request is masked by the CPU. A register, when reset, is initialized to 11111B.

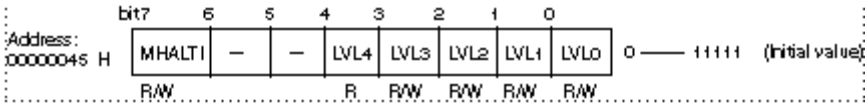
Table 9.4.1 shows the correspondence between a definable interrupt level setting bit and an interrupt level.

Table 9.4.1 Interrupt Levels

ICR4	ICR3	ICR2	ICR1	ICR0	Interrupt level	
0	0	0	0	0	0	Reserved by system
0	1	1	1	0	14	
0	1	1	1	1	15	NMI
1	0	0	0	0	16	Strongest level that can be defined
1	0	0	0	1	17	(Strong)
1	0	0	1	0	18	
1	0	0	1	1	19	
1	0	1	0	0	20	
1	0	1	0	1	21	
1	0	1	1	0	22	
1	0	1	1	1	23	
1	1	0	0	0	24	
1	1	0	0	1	25	
1	1	0	1	0	26	
1	1	0	1	1	27	
1	1	1	0	0	28	
1	1	1	0	1	29	
1	1	1	1	0	30	(Weak)
1	1	1	1	1	31	Interrupt disabled

ICR4 is always 1 and cannot be rewritten to 0.

9.4.2 HRCL (Hold Request Cancel Level register)



This register defines the level at which a hold request cancel request is generated.

[bit 7] MHALTI

The MHALTI bit indicates an NMI request. This bit is set to 1 by an NMI request. The NMI request is cleared if this bit is set to 0.

[bits 4 to 0] LVL4 to 0

These bits define the interrupt level at which a hold request cancel request is generated for the bus master.

If an interrupt request with a stronger interrupt level than specified in this register is generated, a hold request cancel request is sent to the bus master.

The LVL4 bit is always 1 and cannot be rewritten to 0.

9.5 EXPLANATION OF OPERATION

9.5.1 Priority evaluation

This module selects the interrupt cause with the highest priority among those simultaneously generated and outputs the level and number of this interrupt cause to the CPU.

The priority of an interrupt cause can be evaluated according to the following criteria:

1 NMI

2 An interrupt cause meeting the following conditions:

- The level of an interrupt cause with a value other than 31 (31 disables interrupts.)
- An interrupt cause with the smallest level
- One of the above interrupt causes with the smallest interrupt number

If no interrupt cause is selected according to the above evaluation criteria, interrupt level 31 (11111B) is output. At this time, the interrupt number is undefined.

The table in Appendix B shows the correspondence of interrupt causes, interrupt numbers, and interrupt levels.

9.5.2 NMI (Non-Maskable Interrupt)

An NMI has the highest priority among the interrupt causes handled by this module.

Thus, an NMI, if generated at the same time as other interrupt causes, is always handled before others.

■ **If an NMI is generated, the following information is conveyed to the CPU.**

Interrupt level: 15 (01111_B)

Interrupt number: 15 (0001111_B)

■ **NMI detection**

An NMI can be defined or detected in an external interrupt or NMI module. This module only generates an interrupt level and number and MHALTI according to an NMI request.

9.5.3 Hold request cancel request (HRCL)

To process an interrupt with a high priority during CPU hold, request the hold request generator to cancel the request. In the HRCL register, define the interrupt level at which a cancel request is to be generated.

■ **Generation conditions**

If an interrupt cause with a stronger interrupt level than specified in the HRCL register is generated, a hold request cancel request is generated.

Interrupt level of the HRCL register > Interrupt level after the priority evaluation → Cancel request generated

Interrupt level of the HRCL register ≤ Interrupt level after the priority evaluation → Cancel request not generated

Unless the interrupt cause that generated the cancel request is cleared, the cancel request remains valid and prevents any DMA transfers from occurring. Be sure to clear the corresponding interrupt cause.

While an NMI is used, the MHALTI bit of the HRCL register is set to 1. As long as this bit is not

cleared, the cancel request is valid. To put the CPU in the hold status, clear the MHALTI bit.

■ **Definable level**

The HRCL register can be set to 10000B to 11111B as with the ICR.

This bit, if set to 11111B, generates a cancel request for all the interrupt levels. This bit, if set to 10000B, generates a cancel request only for an NMI.

Table 9.5.3 shows how to define the interrupt level at which a hold request cancel request is generated.

Table 9.5.3 Interrupt level at which a cancel request is generated

HRCL register	Interrupt level at which a cancel request is generated
16	NMI only
17	NMI and Interrupt Level 16
18	NMI and Interrupt Levels 16 and 17
~	~
31	NMI and Interrupt Levels 16 through 30 (Initial value)

After resetting, the DMA transfer is disabled at all interrupt levels. No DMA transfer is performed even though an interrupt occurs. In this case, set the HRCL register to the required value.

9.5.4 Recovery from standby mode (stop/sleep)

This module provides a function for recovering from stop mode if an interrupt request is generated. If at least one interrupt request from a peripheral resource (with an interrupt level other than 11111), including an NMI, is generated, a request for recovery from stop mode is sent to the clock controller.

Since the priority evaluation unit restarts when the clock is re-supplied after recovery from stop mode, the CPU executes instructions until the priority evaluation unit provides a result.

Recovery from sleep mode is the same as described above.

The registers of this module can be accessed even during sleep mode.

■ **Precautions**

- The NMI request also initiates recovery from stop mode. However, configure the NMI so that a valid input is detected even in stop mode.
- For an interrupt cause that you do not want to initiate recovery from stop or sleep mode, set the interrupt level to 11111 in the control register of the corresponding peripheral resource.

9.5.5 Example of using the HRCL function

To have the CPU perform processing with a high priority during the DMA transfer, have DMA cancel the hold request, thus canceling the hold status of the CPU. This section explains how to use an interrupt to have DMA cancel the hold request, i.e., enable the CPU implementing a higher priority operation.

■ **Control register**

- HRCL (Hold Request Cancel Level setting register):

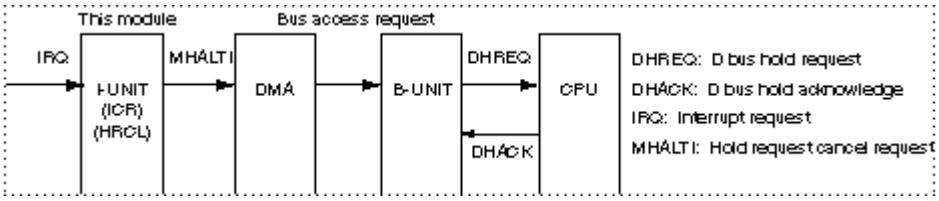
If an interrupt with an interrupt level stronger than specified in this register is generated, a hold request cancel request is sent to DMA. This register defines the reference level for this purpose.

- ICR:

  In the ICR corresponding to the interrupt cause to be used, define a level stronger than that specified in the HRCL register.

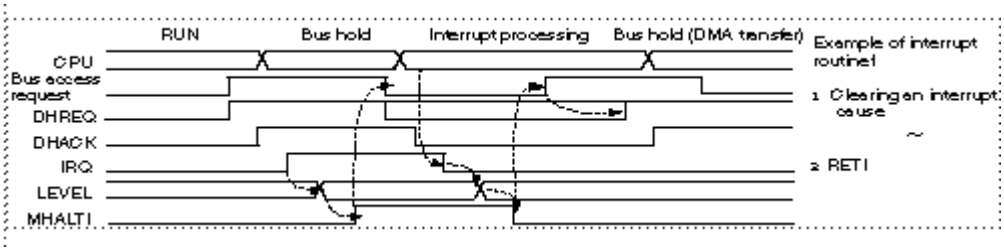
■ Hardware configuration

Signal flow is as follows:



■ Sequence

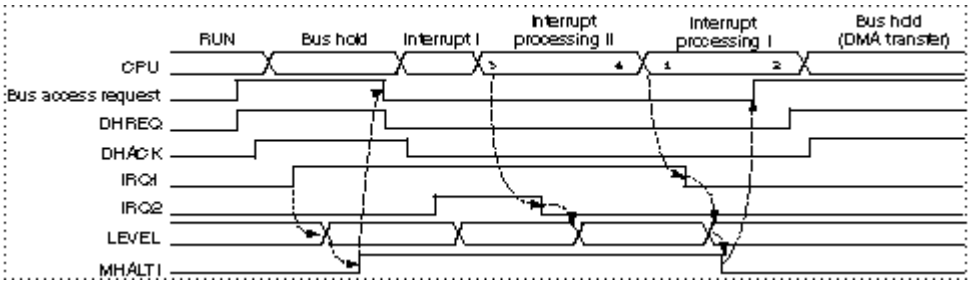
Figure 9.5.5a Itnerrupt level HRCL > a



If an interrupt request occurs, it changes the interrupt level. The interrupt level, if it is stronger than that specified in the HRCL register, makes MHALTI active for DMA. This will cause DMA to set the access request low and the CPU to recover from hold status to perform the interrupt processing.

The following shows the case of multiple interrupts.

Figure 9.5.5b Interrupt level HRCL > a > b



Example of interrupt routine

1, 3 Clearing an interrupt cause

2, 4 RETI

The above example shows that, while Interrupt Routine I is being executed, an interrupt with a higher priority occurs. While a higher interrupt level than that specified in the HRCL register exists, DHREQ remains low.

■ Precautions

Be especially careful of the relationship between the interrupt levels defined in the HRCL register and those define in the ICR.

CHAPTER 10 EXTERNAL INTERRUPT/NMI CONTROLLER

This chapter provides an overview of the external interrupt/NMI controller, describes the register structure and functions, and describes the operation of the external interrupt/NMI controller.

10.1 OVERVIEW OF THE EXTERNAL INTERUPT CONTROLLER

The external interrupt/NMI controller controls external interrupt requests input from the NMI and INT0 to INT7 pins.
Detection of "H" levels, "L" levels, rising edges, or falling edges can be selected (except for the NMI).
The external interrupt/NMI controller can also be used for DMA requests.
This section lists the registers of the controller and provides its block diagram.

■ Register Configuration of the External Interrupt and NMI Controller

Register	Name	Bit structure							
		7	6	5	4	3	2	1	0
Enable interrupt request register	ENIR	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0
		15	14	13	12	11	10	9	8
External interrupt request register	EIRR	ER7	ER6	ER5	ER4	ER3	ER2	ER1	ER0
		15	14	13	12	11	10	9	8
External level register	ELVR	LB7	LA7	LB6	LA6	LB5	LA5	LB4	LA4
		7	6	5	4	3	2	1	0
		LB3	LA3	LB2	LA2	LB1	LA1	LB0	LA0

Figure 10.1a Register Configuration of the External Interrupt and NMI Controller

■ Block Diagram of the External Interrupt and NMI Controller

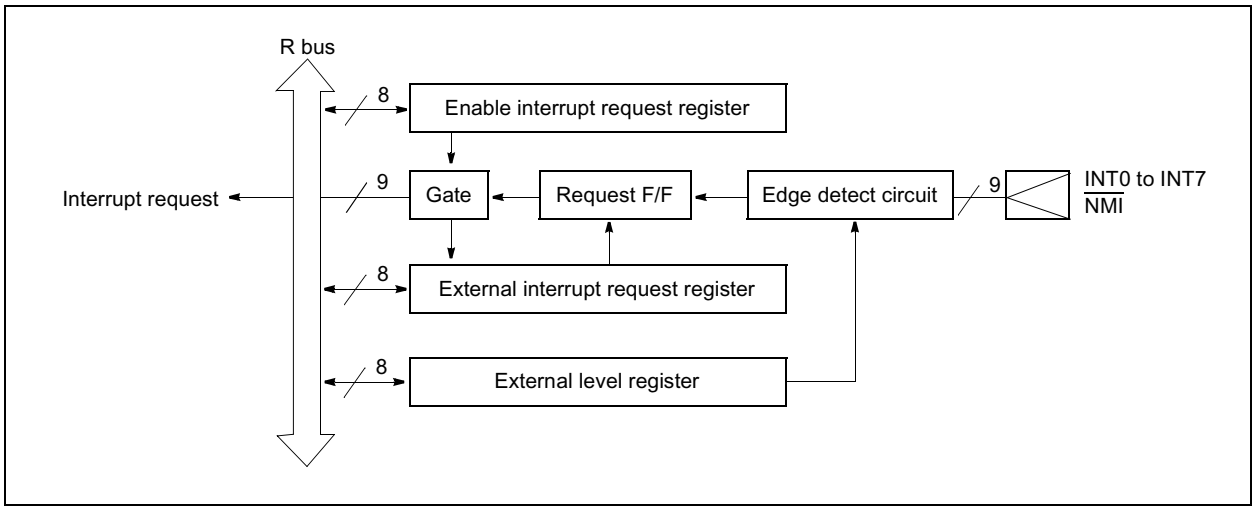


Figure 10.1b Block Diagram of the External Interrupt and NMI Controller

10.2 EXTERNAL INTERRUPT CONTROLLER REGISTERS

This section describes the enable interrupt request register (ENIR), external interrupt request register (EIRR), and external level register (ELVR).

■ **Enable Interrupt Request Register (ENIR)**

		Bits									
	Address	7	6	5	4	3	2	1	0	Initial value	Access
ENIR	000041H	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0	00000000	R/W

Figure 10.2a Structure of the Enable Interrupt Request Register

The enable interrupt request register (ENIR) controls masking of external interrupt requests.

Output of interrupt requests corresponding to bits set to "1" in this register is enabled. (EN0 controls whether INT0 is enabled.) Interrupt requests are output to the interrupt controller.

Interrupt conditions are held for pins corresponding to bits set to "0" in this register but no request is output to the interrupt controller.

No mask bit is provided for the non-maskable interrupt (NMI).

■ **External Interrupt Request Register (EIRR)**

		Bits									
	Address	15	14	13	12	11	10	9	8	Initial value	Access
EIRR	000040H	ER7	ER6	ER5	ER4	ER3	ER2	ER1	ER0	00000000	R/W

Figure 10.2b Structure of the External Interrupt Request Register

Reading the external interrupt request register (EIRR) indicates whether corresponding external interrupt requests are present. Writing to the register clears the flipflops used to store the requests.

Bits that return "1" when read indicate that an external interrupt request is present on the corresponding pin.

Similarly, writing "0" to a bit clears the flipflop used to store the request corresponding to that bit. Writing "1" has no meaning.

Register bits are always read as "1" by read-modify-write instructions.

The NMI flag cannot be read or written by the user.

■ External Level Register (ELVR)

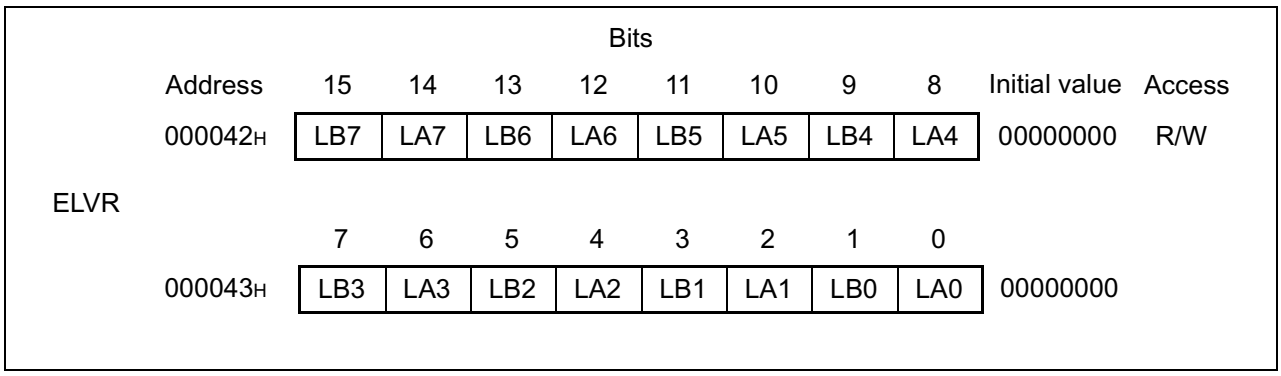


Figure 10.2c Structure of the External Level Register

The external level register (ELVR) specifies how to detect requests.

Two bits each are provided for INT0 to INT7. Table 7.2 lists the settings.

When a request input is set to level detect, the corresponding EIRR bit is immediately set again if cleared when the input is at the active level.

Table 10.2 External Interrupt Request Level Settings

LBx	LAx	Operation
0	0	Generate a request for an "L" level.
0	1	Generate a request for an "H" level.
1	0	Generate a request on a rising edge.
1	1	Generate a request on a falling edge.

The NMI only detects a falling edge on the NMI input (except in stop mode).
In stop mode, the NMI detects the "L" level.

10.3 OPERATION OF THE EXT. INTERRUPT CONTROLLER

This section describes the operation of external interrupts, wakeup from stop mode, the operation procedure for external interrupts, external interrupt request levels, and the NMI.

■ Operation of External Interrupts

After the request level and enable register have been set, input of a request of the type specified by the ELVR register to the interrupt's pin causes the external interrupt controller to generate an interrupt request signal to the interrupt controller. The interrupt controller evaluates the priority of any simultaneously occurring interrupts and generates the interrupt with the highest priority.

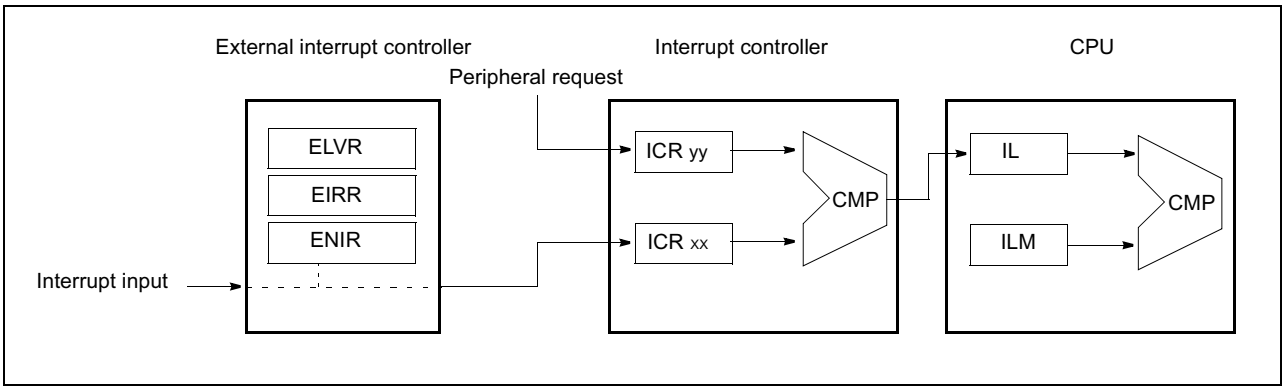


Figure 10.3a Signal Flow for External Interrupts

■ Waking Up from Stop Mode

When using an external interrupt to wakeup from a stop mode in which the clock is halted, set the input request type to "H" level request. Using "L" level requests may cause malfunction.
Edge detection requests cannot be used to wakeup from a stop mode in which the clock is halted.

■ Operation Procedure for External Interrupts

Use the following procedure to set the external interrupt controller registers.

1) Set the enable register bit for the interrupt to disabled.
2) Set the external level register bits for the interrupt.
3) Clear the request bit for the interrupt.
4) Set the enable register bit for the interrupt to enabled.

Steps 3) and 4) can be performed as a single 16-bit write.

Always disable the interrupt in the enable register before setting registers. Also, always clear the request register before enabling the interrupt in the enable register. This is to prevent an unintended interrupt request from being generated when setting registers or enabling the interrupt.

■ External Interrupt Request Levels

When the request type is set to edge detect, a pulse width of at least three machine cycles (peripheral clock machine cycles) is required to ensure that the edge is detected.

If an external request input is applied and then removed when the request type is set to level detect, an internal circuit stores the interrupt condition and therefore the request to the interrupt controller remains active after the external input is removed. The request register must be cleared to clear the request to the interrupt controller.

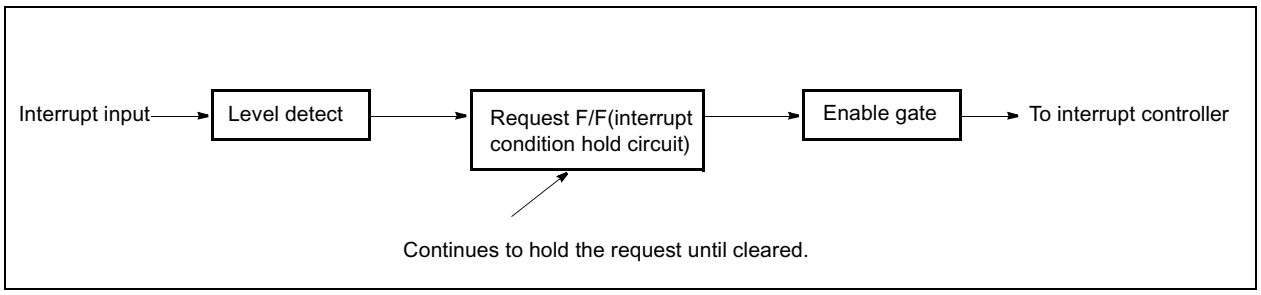


Figure 10.3b Clearing the Interrupt Condition Hold Circuit when the Interrupt Type Is Set to Level Detect

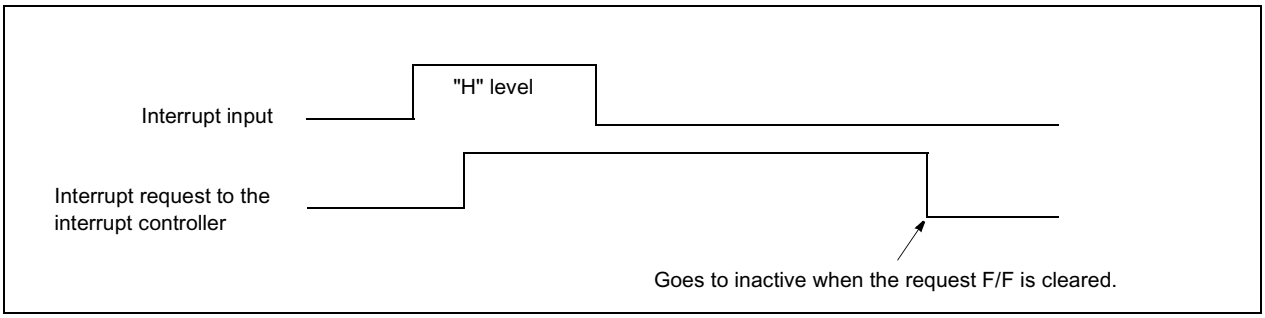


Figure 10.3c Interrupt Condition and Interrupt Request to the Interrupt Controller when the Interrupt Is Enabled

■ Non-maskable Interrupt (NMI)

● The NMI has the highest priority of all user interrupts and cannot be masked. However, if a NMI is detected after reset and before the ILM is set, then the NMI is not accepted by the CPU. In the case, the NMI request is latched and accepted immediately after setting the ILM.Therefore, after a reset, please see NMI after setting the ILM to a value beyond 15.

● The NMI is detected as follows.

Normal operation: Falling edge
Stop mode: "L" level

● The NMI can be used to wakeup from stop mode. Inputting an "L" level during stop mode causes the device to wakeup from stop mode and delay for the oscillation stabilization delay time.

The NMI request detector has an NMI bit. The bit is set by an NMI request; it is cleared when the interrupt is accepted by NMI itself or when a reset occurs. The bit cannot be read or written.

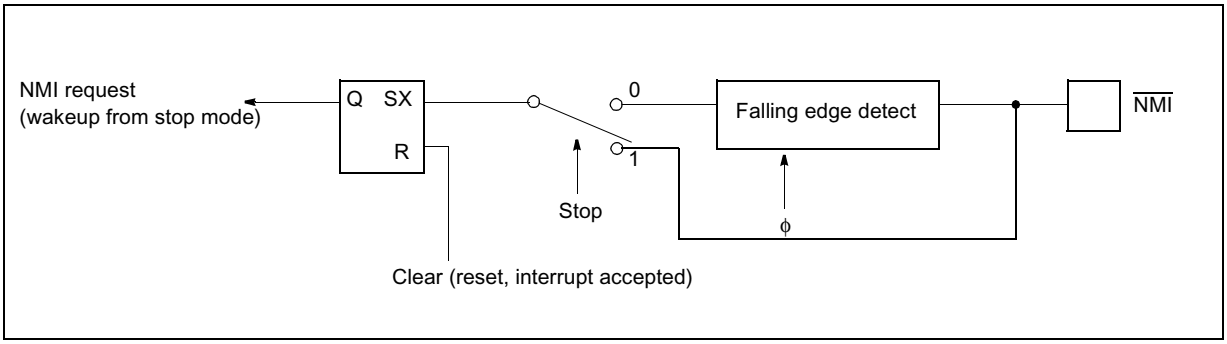


Figure 10.3d NMI Request Detector

CHAPTER 11 DMA CONTROLLER (DMAC)

This Chapter provides an overview of the DMA controller (DMAC), describes the register structure and functions, and describes the operation of the DMA controller.

11.1 OUTLINE

The DMAC module is used to implement direct memory access (DMA) transfer in FR50 series devices.

In a DMA transfer controlled by this module, various types of data can be transferred at high speed without involving the CPU, thus increasing system performance.

■ Hardware configuration

The following are the main components of the DMAC module:

- Five independent DMA channels
- 5-channel independent access control circuit
- 32-bit address registers (Reload can be specified: Two registers for each channel.)
- 16-bit transfer count registers (Reload can be specified: One register for each channel.)
- 4-bit block count registers (One register for each channel)
- External transfer request input pins DREQ0, DREQ1, and DREQ2 (only channels 0, 1, and 2)
- External transfer request acceptance output pins DACK0, DACK1, and DACK2 (only channels 0, 1, and 2)
- DMA termination output pins DEOP0, DEOP1, and DEOP2 (only channels 0, 1, and 2)
- Fly-by transfer (memory to I/O, memory to memory) (Note: IOR and IOW can be supported using ports if necessary.)
  An access signal on the I/O side is generated by decoding the DACK, RDX, WRX and CAS signals.
- Two-cycle transfer

■ Main functions

The following are the main functions of data transfer performed by the module:

● Independent data transfer in multiple channels is enabled (5 channels).

- Priority (channel 0 > channel 1 > channel 2 > channel 3 > channel 4)
- Priority can be alternated between channel 0 and channel 1.
- DMAC start cause

  External-only pin input (edge detection/level detection channels 0 to 2 only)

  Internal peripheral request (interrupt request is shared, including external interrupts)

  Software request (register write)

- Transfer mode

  Demand transfer, burst transfer, step transfer, block transfer

  Addressing mode 32-bit full address specification (increase, decrease, fixed)
  (An address increment/decrement size of -255 to +255 can be specified.)

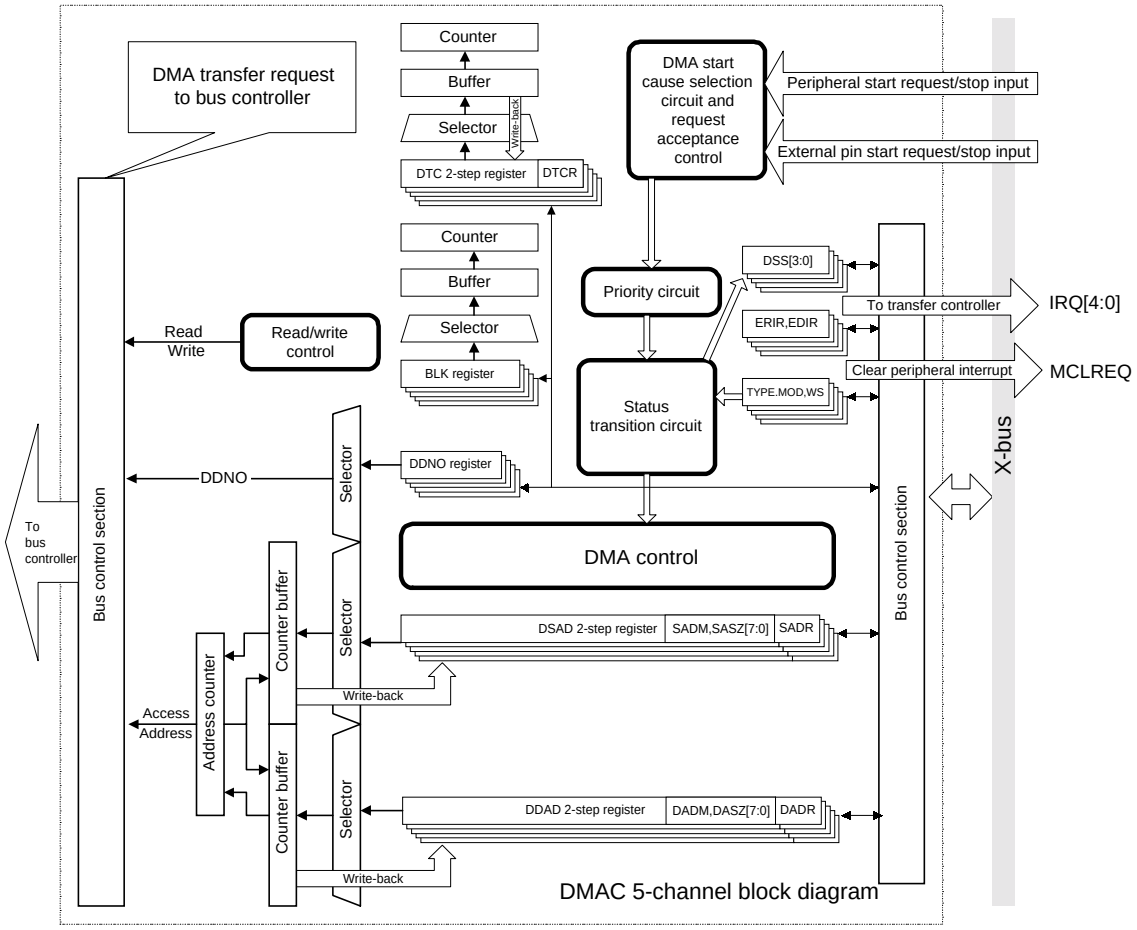
  Data types of byte, halfword, and word lengths

  Single-shot/reload selectable

11.2 OUTLINE OF REGISTERS

Register		Address	bit 31-24 23-16 15-8 7-0
Channel 0 control/status register A	DMACA0	00000200H	
Channel 0 control/status register B	DMACB0	00000204H	
Channel 1 control/status register A	DMACA1	00000208H	
Channel 1 control/status register B	DMACB1	0000020CH	
Channel 2 control/status register A	DMACA2	00000210H	
Channel 2 control/status register B	DMACB2	00000214H	
Channel 3 control/status register A	DMACA3	00000218H	
Channel 3 control/status register B	DMACB3	0000021CH	
Channel 4 control/status register A	DMACA4	00000220H	
Channel 4 control/status register B	DMACB4	00000224H	
Overall control register	DMACR	00000240H	
Channel 0 transfer source address register	DMASA0	00001000H	
Channel 0 transfer destination address register	DMADA0	00001004H	
Channel 1 transfer source address register	DMASA1	00001008H	
Channel 1 transfer destination address register	DMADA1	0000100CH	
Channel 2 transfer source address register	DMASA2	00001010H	
Channel 2 transfer destination address register	DMADA2	00001014H	
Channel 3 transfer source address register	DMASA3	00001018H	
Channel 3 transfer destination address register	DMADA3	0000101CH	
Channel 4 transfer source address register	DMASA4	00001020H	
Channel 4 transfer destination address register	DMADA4	00001028H	

11.3 BLOCK DIAGRAM



DMAC 5-channel block diagram

11.4 DETAILED EXPLANATION OF REGISTERS

11.4.1 Notes on setting registers

When the DMAC is set, certain bits need to be set while the DMA is stopped. If these bits are set during operation (transfer), normal operation is not assured.

A asterisk (*) indicates a bit whose setting, if done during DMAC transfer, may affect operation. The DMAC transfer must be stopped (start-prohibited status or temporarily stopped status) before the bits are rewritten.

If these bits are set with DMA transfer in the start-prohibited status
(DMACR:DMAE = 0 or DMACA:DENB = 0), the setting takes effect when starting is allowed.

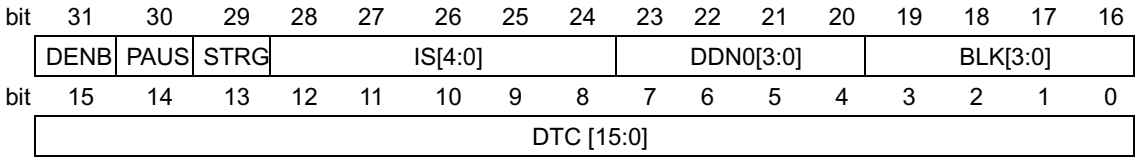
If these bits are set with the DMA transfer in the temporarily stopped status
(DMACR:DMAH[3:0] ≠ 0000 or DMACA:PAUS = 1), the setting takes effect when the temporarily stopped status is released.

11.4.2 DMAC - Channel 0,1,2,3,4 control/status register A

■ **DMACA0 to DMACA4**

These registers control the operation of the DMAC channels. There is one register for each channel.

The function of each bit is explained below.



(Initial value: 00000000_0000XXXX_XXXXXXXX_XXXXXXXX b)

[Bit 31] DENB (DMA Enable): DMA operation enable bit

This bit enables or disables DMA transfer start for a transfer channel.

The activated channel starts DMA transfer when a transfer request is generated and is accepted.

All transfer requests issued for a channel for which start has not been enabled are invalid .

When the transfer of the activated channel ends because the specified count is reached, this bit becomes 0 and the transfer stops.

When 0 is written to this bit, the transfer is forcibly stopped. Before transfer is forcibly stopped by writing 0, the DMA must be placed in the temporarily stopped status by the PAUS bit [DMACA bit 30]. If a forced stop without a temporary stop occurs, the DMA stops but the data transfer is not assured. Use the DSS[2:0] bits [DMACB bits 18 to 16] to check the type of stop.

DENB	Function
0	DMA operation disabled for the corresponding channel (initial value)
1	DMA operation enabled for the corresponding channel

- When reset or stop request is accepted: Initialized to 0.

- Can be read and written.
- When all channel operations are disabled by the bit15:DMAE bit of the overall DMAC control register (DMACR), 1 written to this bit has no effect and the stopped status is retained. When the operation is enabled by this bit and the operation is disabled by the bit15:DMAE bit, this bit becomes 0 and transfer is suspended (forcibly stopped).

[Bit 30] PAUS (Pause): Temporary stop instruction

This bit controls temporary stopping of a DMA transfer for the corresponding channel. Once this bit has been set, DMA transfer is not performed until it is cleared. (During a DMA stop, the DSS bits become 1xx.)

When the PAUS bit is set before start, the temporarily stopped status continues.

New transfer requests issued while this bit is set are accepted but transfer does not start until the bit is cleared.
See section 11.5.11 "Accepting a transfer request and performing transfer" on page 326.

PAUS	Function
0	DMA operation enabled for the corresponding channel (initial value)
1	DMA temporarily stopped for the corresponding channel

- When reset: Initialized to 0.
- Can be read and written.

[Bit 29] STRG (Software Trigger): Transfer request

This bit generates a DMA transfer request for the corresponding channel. When 1 is written to this bit, a transfer request is generated at the point at which writing to the register terminates and transfer for the corresponding channel starts.

However, if the corresponding channel has been activated, operations on this bit have no effect.

Note: When activation by writing to the DMAE bit and a transfer request generated by this bit occur simultaneously, the transfer request is valid and transfer starts. When a transfer request generated by this bit occurs simultaneously with writing 1 to the PAUS bit, the transfer request is valid but the DMA transfer does not start until the PAUS bit is restored to 0.

STRG	Function
0	Invalid
1	DMA start request

- When reset: Initialized to 0.
- The read value is always 0.
- As a written value, only 1 is valid. A 0 has no effect on operation.

[Bits 28 to 24] IS4 to IS0 (Input Select)*: Transfer cause select

These bits select a transfer request cause as shown below. A software transfer request by the STRG bit function is valid regardless of the setting of these bits.

IS	Function	Transfer stop request
00000	Activation by hardware prohibited	Not present
00001 ↓ 01101	Setting prohibited ↓ Setting prohibited	
01110	External DMA - pin H level or ↑ edge	
01111	External DMA - pin L level or ↓ edge	
10000	UART 0 RX Interrupt	Present
10001	UART 0 TX Interrupt	Not present
10010	UART 1 RX Interrupt	
10011	UART 1 TX Interrupt	
10100	External Interrupt 0	
10101	External Interrupt 1	
10110	Reload Timer 0 Interrupt	
10111	Reload Timer 1 Interrupt	
11000	External Interrupt 2	
11001	External Interrupt 3	
11010	UART 2 RX Interrupt	
11011	UART 2 TX Interrupt	
11100	SIO 0 Interrupt *1)	
11101	I^2C Interrupt	
11110	A/D Converter Interrupt	
11111	SIO 1 Interrupt *1)	

- When reset: Initialized to 00000.
- Can be read and written.
- *1): Implemented only on F369GA.

Note: When demand transfer mode is selected, only IS[4:0] = 01110, 01111 can be set. Activation by other causes is prohibited.

Note: Input of an external request is valid only for channels 0, 1, and 2. External request input cannot be selected for channels 3 and 4. Whether detection is level detection or edge detection is determined by the mode setting. (Level detection is used for demand transfer and edge detection for other modes.)

[Bits 23 to 20] DDNO3 to DDNO0 (Direct Access Number)*: Fly-by to internal peripheral

These bits specify the transfer destination/source internal peripheral registers used by the corresponding channel.

- When reset: Initialized to 0000.
- Can be read and written.

Note: MB91360 series does not support this function. Written data is ignored.

[Bits 19 to 16] BLK3 to BLK0 (Block Size): Block size specification

These bits specify the block size of a block transfer for the corresponding channel. The value set in these bits becomes the number of words for one transfer unit (more accurately, the repeat count for data size setting). If block transfer is not performed, set 01H (size 1). (For demand transfer, the value of this register is ignored; size 1 is used.)

BLK	Function
XXXX	Block size specification for the corresponding channel

- When reset: Not initialized.
- Can be read and written.
- When all bits are set to 0, the block size is 16 words.
- When there is a read, the block size (reload value) is always read.

[Bits 15 to 00] DTC (DMA Terminal Count Register)*: Transfer count register

This register stores the transfer count. Each register (one for each DMACA) consists of 16 bits.

Each register has a dedicated reload register. If a data transfer count register is used for a channel that allows it to be reloaded, the initial register value is automatically restored at the end of transfer.

DTC	Function
XXXX	Transfer count specification for the corresponding channel

When DMA transfer starts, the register data is stored in the counter buffer of the DMA dedicated transfer count counter and the counter is decremented for each transfer unit. At the end of the DMA transfer, the contents of the counter buffer are written back to the register and the DMA terminates. As a result, the transfer count value cannot be read during DMA operation.

- When reset: Not initialized.
- Can be read and written. **Halfword** length or **word** length must be used for DTC access.
- The read value is the count value. The reload value cannot be read.
- When reset: Not initialized.

11.4.3 DMAC - Channel 0,1,2,3,4 control/status register B

■ DMACB0 to DMACB4

These registers control the operation of the DMAC channels. There is one register for each channel.

The function of each bit is explained below.

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TYPE[1:0]		MD[1:0]		WS[1:0]		SADM	DADM	DTCR	SADR	DADR	ERIE	EDIE	DSS[2:0]		
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SASZ[7:0]								DASZ[7;0]							

(Initial value: 00000000_00000000_XXXXXXXX_XXXXXXXX b)

[Bits 31 to 30] TYPE (TYPE)*: Transfer type setting

These bits set the operation type of the corresponding channel as follows.

Two-cycle transfer mode: In this mode, the transfer source address (DMASA) and transfer destination address (DMADA) are set, and transfer is performed by repeating the read and write operations as many times as the value of the transfer count. All areas (32-bit addressing) can be specified for both the transfer source and transfer destination.

Fly-by transfer mode: In this mode, external area ⇔ external area transfer is performed in one cycle by setting the memory address in the transfer destination address (DMADA). An external area (excluding SDRAM) must be specified as the memory address.

TYPE	Function
00	Two-cycle transfer (initial value)
01	Fly-by: Memory to I/O transfer
10	Fly-by: I/O to memory transfer
11	Setting prohibited

- When reset: Initialized to 00.
- Can be read and written.

[Bits 29 to 28] MODE (MODE)*: Transfer mode setting

These bits set the operation mode of the corresponding channel as follows.

MOD	Function
00	Block/step transfer mode (initial value)
01	Burst transfer mode
10	Demand transfer mode
11	Setting prohibited

- When reset: Initialized to 00.
- Can be read and written.

[Bits 27 to 26] WS (Word Size): Transfer data size selection

These bits select the transfer data size of the corresponding channel.

Transfer is performed with the data size set in this register for the specified count.

WS	Function
00	Transfer in byte units (initial value)
01	Transfer in halfword units
10	Transfer in word units
11	Setting prohibited

- When reset: Initialized to 00.
- Can be read and written.

[Bit 25] SADM (Source-Addr. Count-Mode Select)*: Transfer source address count mode specification

This bit specifies the processing for each transfer from the transfer source address for the corresponding channel.

The incrementing and decrementing of the address conforms to the specified transfer source address count size (SASZ). The address is incremented or decremented after each transfer.

When the transfer ends, the next access address is written in the corresponding address register (DMASA).

Therefore, the transfer source address register is not updated until the DMA transfer ends.

To fix the address, an increment or decrement must be specified for this register and the address counter size must be 0.

SADM	Function
0	The transfer source address is incremented. (Initial value)
1	The transfer source address is decremented.

- When reset: Initialized to 0.
- Can be read and written.

[Bit 24] DADM (Destination-Addr. Count-Mode Select)*: Transfer destination address count mode specification

This bit specifies the processing for each transfer to the transfer destination address for the corresponding channel.

The incrementing and decrementing of the address conforms to the specified transfer destination address count size (DASZ). The address is incremented or decremented after each transfer. When the transfer ends, the next access address is written in the corresponding address register (DMADA).

As a result, the transfer destination address register is not updated until the DMA transfer terminates.

To fix the address, an increment or decrement must be specified for this register and the address counter size must be 0.

DADM	Function
0	The transfer destination address is incremented. (Initial value)
1	The transfer destination address is decremented.

- When reset: Initialized to 0.
- Can be read and written.

[Bit 23] DTCR (DTC-Reg. Reload)*: Transfer count register reload specification

This bit controls the reload function of the transfer count register for the corresponding channel.

When reloading is enabled by this bit, the count register value is initialized and transfer is restarted at the end of the transfer.

When count counter reloading is disabled, the transfer is a single-shot operation that stops at the end of the transfer even though reloading has been specified in the address register.

DTCR	Function
0	Transfer count register reloading is disabled (initial value).
1	Transfer count register reloading is enabled.

- When reset: Initialized to 0.
- Can be read and written.

[Bit 22] SADR (Source-Addr.-Reg. Reload)*: Transfer source address register reload specification

This bit controls the reload function of the address register for the corresponding channel.

When reloading is enabled by this bit, the transfer source address register value is initialized at the end of the transfer.

When count counter reloading is disabled, the transfer is a single-shot operation that stops at the end of the transfer even though reloading has been specified in the address register. In this case, the address register value stops in the status when the initial value has been reloaded.

When reloading is disabled by this bit, the address register value at the end of the transfer becomes the next access address to the end address (incremented address, if address incrementing is specified).

SADR	Function
0	Transfer source address register reloading is disabled (initial value).
1	Transfer source address register reloading is enabled.

- When reset: Initialized to 0.
- Can be read and written.

[Bit 21] DADR (Dest.-Addr.-Reg. Reload)*: Transfer destination address register reload specification

This bit controls the reload function of the address register for the corresponding channel.

When reloading is enabled by this bit, the transfer destination address register value is initialized at the end of the transfer.

Other functional details are equivalent to the those of bit17:DRLD.

DADR	Function
0	Transfer destination address register reloading is disabled (initial value).
1	Transfer destination address register reloading is enabled.

- When reset: Initialized to 0.
- Can be read and written.

[Bit 20] ERIE (Error Interrupt Enable)*: Error interrupt output enable

This bit controls generation of an interrupt for termination caused by an error. DSS2 to DSS0 indicate the type of error that has occurred. Note that this interrupt is not generated by all termination causes. An interrupt is generated by a specific termination cause. (See the explanation of the DSS2 to DSS0 bits.)

ERIE	Function
0	Error interrupt request output is disabled (initial value).
1	Error transfer request output is enabled.

- When reset: Initialized to 0.
- Can be read and written.

[Bit 19] EDIE (End Interrupt Enable)*: End interrupt output enable

This bit controls generation of an interrupt when termination is normal.

EDIE	Function
0	End interrupt request output is disabled (initial value).
1	End transfer request output is enabled.

- When reset: Initialized to 0.
- Can be read and written.

[Bits 18 to 16] DSS2 to DSS0 (DMA Stop Status)*: Display of cause of transfer stop

These bits cause a 3-bit code (end code) that indicates the reason that DMA transfer for the corresponding channel stopped or ended to be displayed. The contents of the end code are as follows.

DSS	Function	Interrupt generated
000	Initial value	None
x01	Address error (underflow/overflow)	Error
x10	Transfer stop request	Error
x11	Normal termination	Termination
1xx	DMA temporarily stopped (DMAH, PAUS bit, interrupt, etc.)	None

A transfer stop request is set only when the peripheral request and external pin DSTP function are used.

Note: The "Interrupt generated" column indicates the types of interrupt requests that can be generated.

- When reset: Initialized to 000.
- Cleared when 000 is written.
- Can be read and written. However, the only valid value written to this bit is 000.

[Bits 15 to 8] SASZ (Source Addr Count Size)*: Transfer source address count size specification

These bits specify the increment or decrement for each transfer for the transfer source address (DMASA) for the corresponding channel. The value set for this bit becomes the increment or decrement of the address for one transfer unit. The incrementing and decrementing of the address conforms to the specification of the transfer source address count mode (SADM).

SASZ	Function
XXXX	Specifies the increment or decrement size for the transfer source address. 0 to 255.

- When reset: Not initialized.
- Can be read and written.

[Bits 7 to 0] DASZ (Des Addr Count Size)*: Transfer destination address count size specification

These bits specify the increment or decrement for each transfer for the transfer destination address (DMADA) for the corresponding channel. The value set for this bit becomes the increment and decrement of the address for one transfer unit. The incrementing and decrementing of the address conforms to the specification of the transfer destination address count mode (DADM).

DASZ	Function
XXXX	Specifies the increment or decrement size for the transfer source address. 0 to 255.

- When reset: Not initialized.
- Can be read and written.

11.4.4 DMAC - Ch.0-4 transfer source/destination address reg.

■ **DMASA0 to DMASA4/DMADA0 to DMADA4**

These registers control the operation of the DMAC channels. There is one register for each channel.

The function of each bit is explained below.

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DMASA[31:16]															
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DMASA[15:0]															

(Initial value: XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXXXX b)

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DMADA[31:16]															
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DMADA[16:0]															

(Initial value: XXXXXXXX_XXXXXXXX_XXXXXXXX_XXXXXXXX b)

The transfer source/destination addresses are stored in this register group. Each register consists of 32 bits.

[Bits 31 to 0] DMASA (DMA Source Addr)*: Transfer source address setting

These bits specify the transfer source address.

[Bits 31 to 0] DMADA (DMA Destination Addr)*: Transfer destination address setting

These bits specify the transfer destination address.

When DMA transfer starts, this register data is stored in the counter buffer of the DMA dedicated address counter and the address is counted, based on the setting, for each transfer. At the end of the DMA transfer, the contents of the counter buffer are written back to the register and the DMA terminates. As a result, the address counter value cannot be read during DMA operation.

Each register has a dedicated reload register. If a source or destination address register is used for a channel that allows it to be reloaded, the initial register value is automatically restored at the end of transfer. Other address registers are not affected.

- When reset: Not initialized.
- Can be read and written. **Word length** must be used for this register.
- The read value becomes the pre-transfer address value during transfer and becomes the next access address value at the end of the transfer. The reload value cannot be read. Therefore, the transfer address cannot be read on a realtime basis.

Note: Do not use this register to set the DMAC register. DMA transfer to the DMAC register itself is not possible.

11.4.5 DMAC - Channel 0,1,2,3,4 overall control register

■ **DMACR**

This register controls the overall operation of the five DMAC channels. Byte access must be used for this register.

The function of each bit is as follows.

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DMAE	-	-	PM01	DMAH[3:0]				-	-	-	-	-	-	-	-
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

(Initial value: 0XX00000_XXXXXXXX_XXXXXXXX_XXXXXXXX bit)

[Bit 31] DMAE (DMA Enable): DMA operation permission

This bit controls the operation of all DMA channels.

When DMA operation is disabled by this bit, transfer over all channels is disabled regardless of the start/stop setting for each channel and operating status. For a channel in the midst of a transfer, the request is canceled and the transfer stops at the block boundary. Any start operation to be performed for a channel in the disabled status is invalid.

Although start/stop operations are allowed for each channel when DMA operation is enabled by this bit, this is not enough to activate a channel.

When 0 is written to this bit, transfer is forcibly stopped. Before transfer is forcibly stopped by writing 0, the DMA must be placed in the temporarily stopped status by the DMAH[3:0] bits [DMACR bits 27 to 24]. If a forced stop without a temporary stop occurs, the DMA stops but the data transfer is not assured. Use the DSS[2:0] bits [DMACB bits 18 to 16] to check the type of stop.

DMAE	Function
0	DMA transfer disabled for all channels (initial value)
1	DMA transfer enabled for all channels

- When reset: Initialized to 0.
- Can be read and written.

[Bit 28] PM01 (Priority mode ch0,1 robin): Channel priority alternation

Set this bit to alternate the priority of channel 0 and channel 1 for each transfer.

PM01	Function
0	Priority fixed (channel 0 > channel 1) (initial value)
1	Priority alternated (channel 1 > channel 0)

- When reset: Initialized to 0.
- Can be read and written.

[Bits 27 to 24] DMAH (DMA Halt): DMA temporary stop

These bits control temporary stopping of all DMA channels. If one of these bits has been set, DMA transfer is not performed on any channel until all bits are cleared.

When these bits are set before activation, all channels remain in the temporarily stopped status.

All transfer requests generated on a channel for which DMA transfer is enabled (DENB = 1) while this bit is set are valid. Transfer starts when this bit is cleared.

DMAH	Function
0000	DMA operation enabled for all channels (initial value)
not 0000	DMA temporarily stopped for all channels

- When reset: Initialized to 0.
- Can be read and written.

[Bits 30, 29, 23 to 0] (Reserved): Unused bits.

- The read value is undefined.

11.5 OPERATION

11.5.1 Outline

The DMAC block is a multifunctional DMA controller, built into FR50 series products, that controls high-speed data transfer without CPU intervention.

■ Main operation

- Each transfer channel can individually specify various functions.
- When activation has been enabled, a channel does not start a transfer operation until it detects a transfer request that has been set.
- When a channel detects the transfer request, it issues a DMA transfer request to the bus controller. Then, it obtains the right to use the bus from the bus controller and starts transfer.
- Data is transferred in the sequence defined by the mode set for the specific channel.

■ Transfer mode

Each DMA controller channel transfers data according to the transfer mode set with the MOD[1:0] bits of the DMACB register for the channel.

● Block/step transfer

When a transfer request is received, the DMA controller transfers only 1 block of data. The DMA controller does not issue a transfer request to the bus controller until it receives the next transfer request.

Block transfer unit: Block size that has been set (DMACA:BLK[3:0])

● Burst transfer

When a transfer request is received, the DMA controller transfers data continuously using the specified transfer count.

Transfer count specified: Block size x transfer count (DMACA:BLK[3:0] x DMACA:DTC[15:0])

● Demand transfer

The DMA controller transfers data continuously using the specified transfer count or transfers data continuously until an external device stops issuing transfer requests (DREQ pin level detection). The specified transfer count equals the transfer count (DMACA:DTC[15:0]) set in demand transfer mode. The block size is fixed at 1, and the register value is ignored.

■ Transfer type

● Two-cycle transfer (regular transfer)

One DMA controller operation consists of a read operation and a write operation.

The DMA controller reads data from the address in the transfer source register and writes the data to the address in the transfer destination register.

- Fly-by transfer (memory to I/O)

  The DMA controller operation consists of a read operation.

  To transfer data when fly-by transfer is set, the DMA controller issues a fly-by transfer (read) request to the bus controller. The bus controller then causes the external interface to transfer data in a fly-by transfer (read).

- Fly-by transfer (I/O to memory)

  The DMA controller operation consists of a write operation.

  Other operations are the same as those for a memory-to-I/O fly-by transfer.

  The area accessed to write data in a fly-by transfer must be an external area.

■ Transfer address

The address is specified as explained below. A separate address is specified for each channel transfer source and channel transfer destination.

Specification of the address setting registers (DMASA and DMADA) in a two-cycle transfer is different from that for a fly-by transfer.

- Specifying addresses in a two-cycle transfer

  The value read from a register (DMASA or DMADA) in which an address has already been set is used as the address to access data.

  When the DMA controller receives a transfer request, it stores the address from the register in the temporary storage buffer and starts data transfer.

  The counter generates (incrementing decrementing, and fixed-selection) the address to be used for the next access for each transfer (access) and then returns the generated address to the temporary memory buffer. Data in the temporary memory buffer is returned to the register (DMASA or DMADA) whenever a 1-block transfer terminates.

  Since, therefore, the address register (DMASA or DMADA) value is updated for each block transfer, the address used during transfer cannot be obtained.

- Specifying the address in a fly-by transfer

  In this transfer type, the value read from the transfer destination address register (DMADA) is used as the address to access data. The transfer source address register (DMASA) is ignored. The address must be set in an external area.

  When the DMA controller receives a transfer request, it stores the address from the register in the temporary storage buffer and starts data transfer.

  The counter generates (incrementing decrementing, and fixed-selection) the address to be used for the next access for each transfer (access) and then returns the generated address to the temporary memory buffer. Data in this temporary memory buffer is returned to the register (DMADA) whenever a 1-block transfer terminates.

  Since, therefore, the address register (DMADA) value is updated for each block transfer, the address used during transfer cannot be obtained.

■ **Transfer count and transfer termination**

● Transfer count

The transfer count register is decremented by 1 whenever a 1-block transfer terminates. When the transfer count register becomes 0, data transfer terminates for the specified transfer count. The controller then displays the termination code and stops or restarts (*1) data transfer.

Note: *1 If reloading of the transfer count register has been disabled, the controller stops data transfer. If it is enabled, the controller initializes the register value and waits for a data transfer request (DMACB:DTCR)

Like the address register, the transfer count register value is updated for each 1-block transfer.

● Transfer termination

The causes of transfer termination are listed below. A termination code, which indicates the cause, is displayed when transfer terminates (DMACB:DSS[2:0]).

- Data transfer termination for the specified transfer count (DMACA:BLK[3:0] x DMACA:DTC[15:0]) => Normal termination
- Issuing a transfer stop request from a peripheral circuit or external pin (DSTP) => Error
- Occurrence of an address error => Error
- Occurrence of a reset => Reset

The transfer stop cause is displayed (DSS) for each termination cause, and a transfer stop interrupt or error interrupt may occur.

11.5.2 Setting a transfer request

There are three types of transfer requests to start DMA controller transfer, as listed below. A software request can always be used regardless of whether other requests have been set.

■ **External transfer request pin**

The input to an input pin provided for each channel generates a transfer request.

Only channels 0 to 3 can be used (DREQ0, 1, and 2).

Setting the transfer type and start cause selects the following causes.

Edge detection

When the transfer type is block, step, or burst transfer, edge detection is selected.

- Rising edge detection: Set in the transfer cause selection register when DMACB:IS[4:0] is 01110
- Falling edge detection: Set in the transfer cause selection register when DMACB:IS[4:0] is 01111

When the transfer type is demand transfer, level detection is selected.

- H-level detection: Set in the transfer cause selection register when DMACB:IS[4:0] is 01110
- L-level detection: Set in the transfer cause selection register when DMACB:IS[4:0] is 01111

■ **Internal peripheral request**

An interrupt generated by an internal peripheral circuit generates a transfer request.

The peripheral circuit generating an interrupt that causes a transfer request to be issued is set for each channel. (DMACB:IS[4:0] = 1xxxx)

This type of request cannot be used as an external transfer request.

Note: Since an interrupt request used as a transfer request is taken as an interrupt request for the CPU, set the interrupt controller to interrupts disabled (ICR register).

■ **Software request**

Writing data to the trigger bit of the register issues the transfer request (DMACA:STRG).

This request, which is separate from the two transfer requests explained above, is the one usually used.

If the software request is issued simultaneously with activation (transfer enable), the DMA transfer request is immediately issued to the bus controller and the transfer starts.

11.5.3 Transfer sequence

The transfer type and transfer mode that determine the operation sequence after DMA data transfer has started can be set separately for each channel (DMACB:TYPE [1:0] and MOD[1:0] settings).

■ **Selecting the transfer sequence**

The following sequences are determined by the register setting:

- Burst two-cycle transfer
- Demand two-cycle transfer
- Block/step two-cycle transfer
- Burst fly-by transfer
- Demand fly-by transfer
- Block/step fly-by transfer

■ **Burst two-cycle transfer**

When a transfer request is received, the DMA controller continuously transfers data for the specified transfer count. In a two-cycle transfer, a 32-bit transfer source address and transfer destination address can be specified for all areas.

A peripheral transfer request, software transfer request, or external pin (DREQ) edge input detection request can be selected as the transfer request.

Specifying the transfer source address	Direction	Specifying the transfer destination address
All areas can be specified using 32 bits.	⇒	All areas can be specified using 32 bits.

(List of transfer addresses that can be specified)

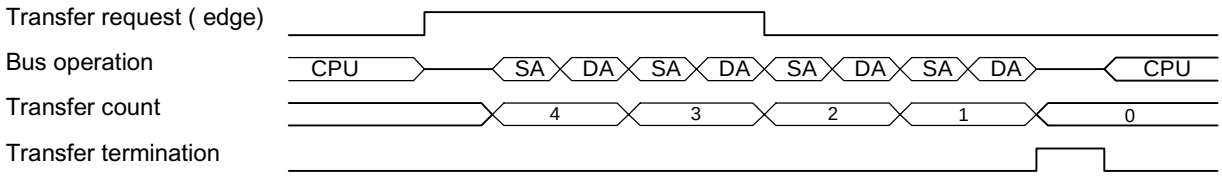
- Characteristics of burst transfer

  When a transfer request is received, the DMA controller transfers data continuously until the transfer count register becomes 0.
  The transfer count is obtained as follows: Transfer count = Block size x Transfer count. (DMACA:BLK[3:0] x DMACA:DTC[15:0])

  If another transfer request is issued during data transfer, it is ignored.

  If reloading of the transfer count register has been enabled, the next transfer request is accepted after data transfer ends.

  If a higher-priority transfer request is accepted from another channel during data transfer, the channel is switched to the other channel when one block of data has been transferred. Control is not returned to the original channel until the data transfer for the higher-priority transfer request has finished.



(Example of burst transfer when burst transfer is activated at the external pin signal rising edge, the block count is 1, and the transfer count is 4)

■ Burst fly-by transfer

This transfer is same as two-cycle transfer except for the following except that the transfer area is an external area only and the transfer operation consists of only a read (memory to I/O) or write (I/O to memory) operation.

Specifying the transfer source address	Direction	Specifying the transfer destination address
Specification not required (invalid)	None	External area

(List of transfer addresses that can be specified)

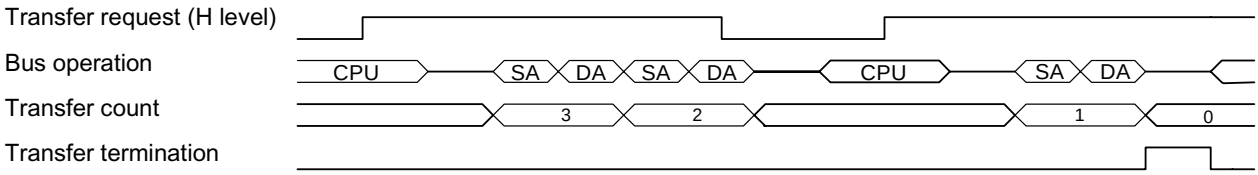
■ Demand two-cycle transfer

The demand transfer sequence is selected only when the external pin H or L level is selected as a transfer request. (DMACA:IS[3:0] is set for the level selection.)

- Characteristics of continuous transfer

  The transfer request is checked every time. If the external input level is the same as the transfer request level, data is transferred continuously without the external transfer request being accepted. If the external input level changes, the original transfer request is canceled, and data transfer is stopped when one block of data is transferred. This operation continues for the specified transfer count.

Other operations are the same as those for burst transfer.



(Example of demand transfer when the external pin H level is activated, the block count is 1, and the transfer count is 3)

Transfer source address	Direction	Transfer destination address
External area	⇒	External area
External area	⇒	Internal I/O
External area	⇒	Internal RAM or ROM
Internal I/O	⇒	External area
Internal RAM or ROM	⇒	External area

(List of transfer addresses that can be specified)

Note: When demand transfer is selected, make sure to set the address of an external area in the transfer source or transfer destination register, or in both registers. In demand transfer mode, since the DMA transfer is set adjusted to the external bus timing, an external area must be accessed.

■ Demand fly-by transfer

This transfer is same as two-cycle transfer except for the following except that the transfer area is an external area only and the transfer operation consists of only a read (memory to I/O) or write (I/O to memory) operation.

Specifying the transfer source address	Direction	Specifying the transfer destination address
Specification not required (invalid)	None	External area

(List of transfer addresses that can be specified)

■ Step/block two-cycle transfer

For a step/block transfer (data is transferred for the specified block count for each 1-block transfer request), a 32-bit transfer source or transfer destination address can be specified for all areas.

Specifying the transfer source address	Direction	Specifying the transfer destination address
All areas can be specified using 32 bits.	⇒	All areas can be specified using 32 bits.

(List of transfer addresses that can be specified)

■ Step transfer

When 1 is set as the block size, the step transfer sequence is selected.

- Characteristics of step transfer

  When a transfer request is received, the DMA controller transfers data only once. It then cancels the transfer request and stops transferring data. (The DMA controller cancels the DMA transfer request from the bus controller.)

  If another transfer request is issued during data transfer, it is ignored.

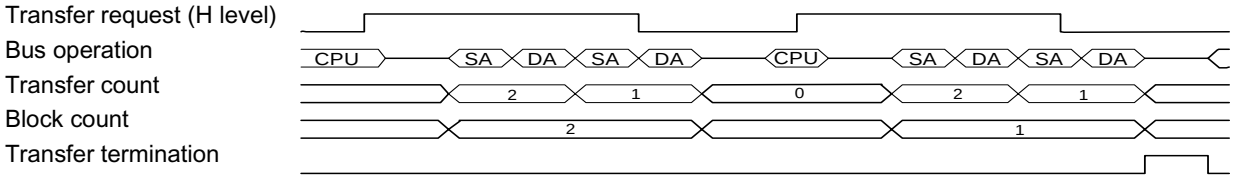
  If a higher-priority transfer request is accepted from another channel during data transfer, the channel is switched to start another data transfer when the data transfer ends. The priority in the step transfer is valid when the transfer requests are issued at the same time.

■ Block transfer

If a value other than 1 is set as the block size, the block transfer sequence is selected.

- Characteristics of block transfer

  This transfer is the same operation as step transfer except that one data transfer consists of two or more (block count) transfer cycles.



(Example of block transfer when block transfer is activated at the external pin signal rising edge, the block count is 2, and the transfer count is 2)

■ Step/block two-cycle fly-by transfer

This transfer is same as the two-cycle transfer except that the transfer area is an external area only and the transfer operation consists of only a read (memory to I/O) or write (I/O to memory) operation..

Specifying the transfer source address	Direction	Specifying the transfer destination address
Specification not required (invalid)	None	External area

(List of transfer addresses that can be specified)

11.5.4 General DMA transfer

■ **Block size**

- The quantity of data to be transferred at one time equals the value set in the block size specification register (x data size).
- Since the quantity of data to be transferred in a transfer cycle is fixed at the value specified for the data size, the data to be transferred at one time consists of the number of transfer cycles for the block size specification value.
- If a higher-priority transfer request is accepted during data transfer or a data transfer temporary stop request is issued, data transfer is stopped when the data block unit is transferred. Data blocks that are not expected to be divided or temporarily stopped are assured. A large block size may degrade response.
- Data transfer stops immediately only when the reset signal is detected. However, the data being transferred is not assured.

■ **Reload operation**

The DMAC module can set the three types of reload functions explained below for each channel:

1) Transfer count register reload function

After data transfer ends for the specified count, this function sets the initial setting value again in the transfer count register and waits for activation to be received.

This function is set to repeat all transfer sequences.

When this function is not specified, the count register value becomes and remains zero after data transfer ends for the specified count, and subsequent data is not transferred.

2) Transfer source address register reload function

After data transfer ends for the specified count, this function sets the initial setting value again in the transfer source address register.

This function is set to continuously transfer data from a fixed area in the transfer source address area.

When this function is not specified, the transfer source address register value becomes the address after the address used when data transfer ended for the specified count. This function is used when the address area is not fixed.

3) Transfer destination address register reload function

After data transfer ends for the specified count, this function sets the initial setting value again in the transfer destination address register.

This function is set to continuously transfer data to a fixed area in the transfer destination address area.

(The following is the same as (2).)

- After data transfer ends for the specified count, simply enabling the transfer source or transfer destination register reload function does not restart data transfer. Rather, each address register value is only set again.

Note: Special example of operation mode and reload operation:

- If the transfer count register reload function is used when data is transferred in continuous transfer mode by detecting the external pin input level, this function reloads the data as is,

even though data transfer ended during input, and continues transfer. In this case, a termination code is set.

- To start data transfer again from input detection, do not specify the reload function after data transfer is temporarily stopped due to transfer termination.
- When the data transfer is interrupted if data is transferred in burst, block, or step transfer mode, the reload operation is executed to terminate data transfer. Data is not transferred until the transfer request input is detected again.

11.5.5 Addressing mode

The transfer destination or transfer source address for each transfer channel is specified individually.

Address specification is explained below. Specify addresses in the transfer sequence.

■ Specifying the address register

- In two-cycle transfer mode, set the transfer source address in the transfer source address setting register (DMASA) and set the transfer destination address in the transfer destination address setting register (DMADA).
- In fly-by transfer mode, set the memory address in the transfer destination address setting register (DMASA). The value of the transfer destination address setting register (DMADA) at this time is ignored.

■ Characteristics of the address register

The register can have up to 32 bits. When a 32-bit length register is used, all spaces in the memory map can be accessed.

■ Address register function

The address is read for every access and transferred to the address bus.

At the same time, the address counter calculates the address to be used for the next access and the address register is updated with the new address.

The transfer destination or transfer source address is calculated for each channel individually using an increment or a decrement. The size of the increment or decrement used to calculate addresses is determined by the address count size specification register value. (DMACB:SASZ, DASZ)

If the reload function has not been enabled, the address resulting from address calculation remains as the last address in the address register when data transfer ends.

If the reload function has been enabled, the initial address value is reloaded.

Note: 1. If overflow or underflow occurs as a result of 32-bit address calculation, the error is detected as an address error and the data transfer by the channel is stopped.
(See items related to termination codes.)

Note: 2. Do not set the DMAC register address in the address register.

Note: 3. In demand transfer, make sure to set the address of an external area in the transfer source address setting register or transfer destination address setting register, or in both registers.

Note: 4. Do not allow the DMAC to transfer data to a DMAC register.

11.5.6 Data type

The data length (data size) in one transfer is selected from the following:

- byte
- halfword
- word

In DMA data transfer, word boundary specifications are observed. Therefore, if an address with a different data length is specified for the transfer destination or transfer source address, different low-order bits for the data length are ignored.

- word: The access start address consists of four bytes the two low-order bits of which are 00.
- halfword: The access start address consists of two bytes the low-order bit of which is 0.
- byte: The access start address and the specified address match.

If the lower-order bits of the transfer source address differ from those of the transfer destination address, the address that was set is output on the internal address bus. However, the address is corrected according to the specifications, and the data is accessed.

11.5.7 Controlling the transfer count

The transfer count is a 16-bit specification (1 to 65536). The transfer count specification value is set in the transfer count register (DMACA:DTC).

The register value is stored in the temporary storage buffer when data transfer is started and is decremented by 1 in the transfer count counter. When this counter value becomes 0, transfer end for the specified count is detected, and the data transfer by the channel is stopped or restart acceptance (reload) is awaited.

■ Characteristics of transfer count register group

Each register consists of 16 bits.

Each register waits for its own reload register.

If the register value is zero when data transfer is started, data is transferred 65,536 times.

■ Reload operation

This function is effective only when the register has the reload function and reloading has been enabled.

The initial value of the count register is saved in the reload register when data transfer is started.

When the transfer count counter becomes zero, end of transfer is posted, and the initial value is read from the reload register and written to the count register.

11.5.8 Controlling the CPU

When DMA accepts a transfer request, it sends the transfer request to the bus controller.

The bus controller gives DMA the right to use the internal bus at the bus operation interval. Data transfer then starts.

■ **DMA transfer and interrupt**

During DMA data transfer, interrupts are not usually accepted until the data transfer ends.

If a DMA transfer request is issued during an interrupt, the transfer request is accepted and the interrupt is stopped until the data transfer ends.

As an exception, if an NMI request or an interrupt request which is higher than the hold suppression level set by the interrupt controller is issued, the DMAC temporarily cancels the transfer request to the bus controller at 1-block intervals. The DMAC then temporarily stops data transfer until the requested interrupt terminates. The transfer request is internally kept during the interrupt. When the interrupt terminates, the DMAC reissues the transfer request to the bus controller, obtains the right to use the bus, and restarts the DMA transfer.

■ **DMA suppression**

If a higher-priority interrupt cause occurs during DMA transfer, the FR50 series DMAC stops DMA transfer and control branches to the corresponding interrupt routine. This feature is valid as long as the interrupt request is being processed. However, if the interrupt cause is cleared, this feature does not work and DMA transfer is restarted by the interrupt routine. Therefore, if the processing routine for the interrupt cause that interrupts DMA transfer suppresses starting of DMA retransfer after the interrupt cause is cleared, use the DMA suppression function.
Writing a value other than zero to overall DMAC control register DMAH[3:0] bits starts the DMA suppression function. Writing zero to this register stops the DMA suppression function.

This function is mainly used in the interrupt routine. Before the interrupt routine clears the interrupt cause, this function increments the value in the DMA suppression register by 1. This increment will not cause a subsequent DMA transfer. After the interrupt has been handled, the value of the DMAH[3:0] bits is decremented by 1 before control is returned. If simultaneous interrupts occur, the value of the DMAH[3:0] bits does not become zero, and DMA transfer is continuously suppressed. If simultaneous interrupts do not occur, the value of DMAH[3:0] bits becomes zero, and the DMA request is enabled immediately.

Note: The number of bits in the register is four. If the interrupt level exceeds 15, this function cannot be used because of simultaneous interrupts.

Note: Increase the DMA task priority higher by 15 levels compared with other interrupt levels.

11.5.9 Hold intervention

When a device operates in external bus extended mode, the hold function can be used from external pins. The relationship between the external hold request and DMA transfer request for the DMAC module is explained below.

■ **DMA transfer request during external hold**

If the external bus area is accessed after DMA transfer has started., the DMA transfer is temporarily stopped at that point. The DMA transfer is restarted after the external hold is released.

■ **External hold request during DMA transfer**

The hold function is used from an external pin. If, however, the external bus area is

accessed because of DMA transfer, the DMA transfer is temporarily stopped at that point. The DMA transfer is restarted after the external hold is released.

■ **Simultaneous occurrence of DMA transfer request and external hold request**

The hold function is used from an external pin, and the DMA transfer is internally started. If, however, the external bus area is accessed because of DMA transfer, the DMA transfer is temporarily stopped at that point. The DMA transfer is restarted after the external hold is released.

11.5.10 Operation start

Although each channel individually controls the starting of DMA transfer, all channels must have been enabled for operation.

■ **Enabling operation for all channels**

Before DMAC channels are started, operation by all channels must be enabled with the DMA operation enable bit (DMACR:DMAE). All start settings and transfer requests are invalid if channels have not been enabled.

■ **Transfer start**

The operation enable bit in the control register for each channel is used to start the transfer operation. When a transfer request for the activated channel is accepted, DMA transfer operation is started in the mode that has been set.

■ **Starting the channel after it has been temporarily stopped**

When a channel has been stopped temporarily before DMA transfer is started, it continues to temporarily stop even though the channel transfer operation has started. If a transfer request is received while the channel keeps stopping, the request is accepted and held. The channel starts transferring data when the temporary stop is released.

11.5.11 Accepting a transfer request and performing transfer

After a channel is started, sampling of the transfer request set to each channel is started.

If a transfer request is detected when edge detection is selected as an external pin start cause, the DMAC holds the request until the conditions for clearing the transfer request are satisfied. (when an external pin start cause is selected in a block, step, or burst transfer.)

If level detection or peripheral interrupt start is selected because of the external pin start cause, the DMAC continues data transfer until the transfer request is cleared. If the transfer request is cleared, the DMAC stops transferring data after the transfer unit (demand transfer/peripheral interrupt start).
Since a peripheral interrupt is assumed to be level detection, use the DMA to clear the interrupt.

A transfer request is always accepted, even when a transfer request from another channel is accepted, and data is transferred. The order in which channels transfer data for the transfer unit is determined according to priority.

11.5.12 Clearing a peripheral interrupt by the DMA

The FR50 series DMA provides a function for clearing peripheral interrupts. This function operates when peripheral interrupt is selected as the DMA start cause (when IS[4:0] is 1xxxx).

The peripheral interrupt is cleared for the start cause that has been set. Only peripheral functions set by IS[4:0] are cleared.

Interrupt clear generation timing

When clearing of an interrupt is generated is determined by the transfer mode. (See section11.6

"OPERATION FLOW" on page 334.)

■ **Block/step transfer**

When block transfer is selected, the clear signal is generated every time 1 block (step) of data is transferred.

■ **Burst transfer**

When burst transfer is selected, the clear signal is generated after data is transferred for the specified transfer count.

■ **Demand transfer**

Since only a start request from an external pin is supported in demand transfer, a clear signal is not generated.

11.5.13 Temporary stop

DMA transfer temporarily stops in the cases listed below.

■ **Setting temporary stop by writing data to the control register (can be set only for all channels at once)**

When temporary stop is set using the temporary stop bits, the all channels stop transferring data until temporary stop release is set. Use the DSS bits to check the temporary stop.

When the temporary stop is released, the channels restart data transfer.

■ **During processing of an NMI or hold suppression level interrupt**

If an NMI request or interrupt request that is higher than the hold suppression level occurs, all the channels that are transferring data are stopped temporarily after the transfer unit. The channels then relinquish the right to use the bus, giving the NMI or hold suppression level interrupt processing priority. Transfer requests accepted during NMI or interrupt processing are held as is, and the channels wait for NMI processing to terminate.

The channels holding requests restart data transfer after the NMI or hold suppression level interrupt processing has been completed.

11.5.14 Operation termination/stop

Although termination of DMA transfer is controlled individually for each channel, the operation of all channels can be disabled.

■ **Transfer termination**

If the transfer count register becomes zero when reloading has not been enabled, data transfer stops. The termination code indicating normal termination is displayed, and subsequent transfer requests are invalid. (The DMACA:DENB bit is cleared.)

If the transfer count register becomes zero when reloading has been enabled, the initial value is reloaded. The termination code indicating normal termination is displayed, and waiting for the transfer request starts. (The DMACA:DENB bit is not cleared.)

■ **Disabling the operation of all channels**

If the operation of all channels is disabled using the DMA operation enable bit DMAE, the DMAC operation of all channels, including a channel in current use, is stopped. Even though DMA operation for all channels is enabled again, data is not transferred unless DMA operation is restarted for specific channels. No interrupt will occur in this case.

11.5.15 Stop due to an error

If a channel stops transferring data for a reason other than normal termination (that is, for various errors) when data is being transferred for the specified count, channel operation stops or is forcibly stopped.

■ **Transfer stop request issued from a peripheral circuit**

Some peripheral circuits that issue a transfer request issue a transfer stop request if an error is detected. (Example: A reception or transfer error in a communication peripheral circuit)

The DMAC that accepted the transfer stop request displays the termination code indicating the transfer stop request and stops transferring data over the channel.

IS	Function	Transfer stop request
00000	Hardware	↑
↓	↓	None
01111	External pin L level or ↓ edge	↓
10000		↑
↓	↓	Issued
10010		↓
10011		↑
↓	↓	None
11111		↓

Note: For more information about the conditions under which a transfer stop request is issued, see the specifications for each peripheral circuit.

■ **Occurrence of an address error**

If an incorrect address is set in an address mode as shown below, the setting is detected as an address error. (When a 32-bit address is specified, an overflow or underflow occurs in the address counter.)

When an address error is detected, the DMAC displays the termination code indicating that an address error has occurred and stops transferring data over the channel.

11.5.16 DMAC interrupt control

The following interrupts can be output for each DMAC channel in addition to peripheral interrupts that are regarded as transfer requests.

- Transfer termination interrupt:Occurs only when data transfer ends normally
- Error interrupt:Transfer stop request issued from a peripheral circuit
  (an error due to a peripheral circuit),
  Occurrence of an address error (an error due to software)

These interrupts are all generated based on the contents of the termination code.

000 is written to DMACS DSS2 to DSS0 (termination code) to clear the interrupt request. When the channel restarts data transfer, make sure to write 000 as the termination code to clear the interrupt request.

If reloading has been enabled, the channel automatically restarts data transfer. At this time, the termination code is not cleared, and is held until a new termination code is written when the next data transfer terminates.

Only one type of termination cause can be displayed with a termination code. If there are two or more simultaneous causes, the priority is determined and the code with higher priority is displayed. The interrupt that is to occur follows the display of the termination code.

The following lists the priority for displaying the termination codes in descending order of priority:

- Reset
- Clearing the request by writing 000
- Peripheral stop request or stop request by external pin input (DSTP)
- Normal termination
- Stop when an address error is detected
- Selecting and controlling channels

11.5.17 DMA transfer during sleep

The DMAC also operates in sleep mode.

Note the following for operation of the DMAC in sleep mode.

Note: 1) The DMAC register cannot be rewritten because the CPU stops. The register must be set in advance before the DMAC an operate in sleep mode.

Note: 2) Because sleep mode is released by an interrupt, if a peripheral interrupt is selected due to DMAC start, the interrupt must be disabled by the interrupt controller.
If sleep mode is not to be released by a DMAC termination interrupt, disable the interrupt.

11.5.18 Channel selection and control

Up to five transfer channels can be specified for simultaneous operation. The functions for each channel can set separately.

■ Channel priority

Because only one channel can be used for DMA transfer, channels are assigned a priority.
Fixed mode or alternation mode is used to assign priority. A mode is selected for each channel group, which will be explained later.

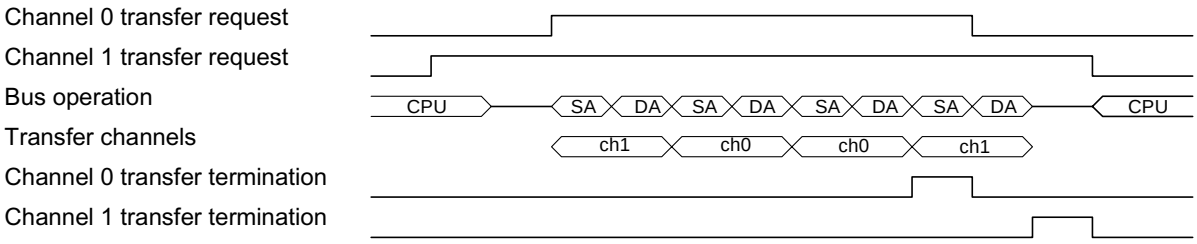
- Fixed mode

  The order of channels is fixed in ascending order by channel number.

  (Channel 0 > channel 1 > channel 2 > channel 3 > channel 4)

  If a higher-priority transfer request is accepted during data transfer, the transfer channel is switched to the channel with the higher priority when the data transfer for the block unit (the value set in the block size specification register x the data size) is completed.

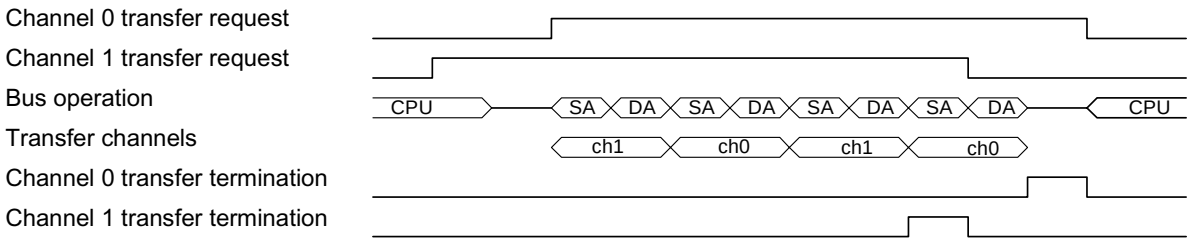
  When the data transfer for the higher-priority transfer request is completed, data transfer for the original channel is restarted.



- Alternation mode (only between channel 0 and channel 1)

  When operation has been enabled, channels are set in the same order as (1). However, the channel order is reversed each time a data transfer ends. If the simultaneous transfer requests are issued, the channel is switched for each transfer.

  This mode is valid when continuous or burst transfer is set.



■ Channel group

Channel priority is selected by mode as shown in the following table.

Mode	Priority	Remarks
Fixed	ch0>ch1	
Alternation	ch0>ch1 ↑ ↓ ch0<ch1	The channels are initially set in above order. The order is reversed after a channel transfers data in the above order.

11.5.19 External Pin and Internal Operation Timing

This is a supplementary note related to external pin and internal operation timing.

■ **Minimum effective pulse size of DREQ external pin (only channels 0, 1, and 2)**

When DMA transfer operates in all modes (such as burst, step, block, and demand transfer), the minimum size required is five system clock cycles (=1/2θ, two CPU system clock cycles).

Note: DACK output does not indicate acceptance of DREQ input. If DMA is enabled and data has not been transferred yet, DREQ input is always accepted. Therefore, DREQ input must not be held until DACK output is asserted (demand transfer mode excepted).

■ **Negate timing of DREQ external pin when demand transfer request is stopped**

● For two-cycle transfer

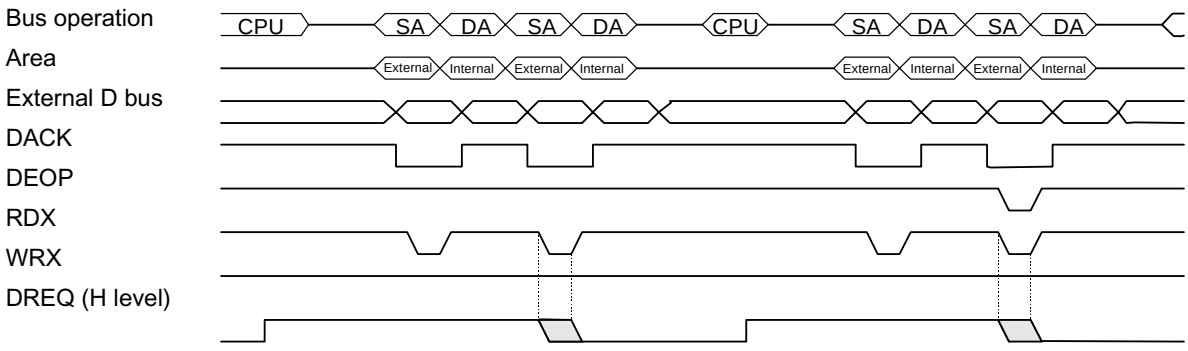
In demand transfer, make sure to set the address of an external area in the transfer source address setting register or transfer destination address setting register, or in both registers.

• When data is transferred between external areas

Negate the external WRX pin output while it is low when the transfer source is accessed in the last DMA transfer (between DACK=L & WRX=L). If DREQ is negated after the above negation, the next data may be transferred.

• When data is transferred between an external area and internal area

Negate the external RDX pin output while it is low when the transfer source is accessed in the last DMA transfer (between DACK=L & RDX=L). If DREQ is negated after the above negation, the next data may be transferred.



(Example of the negate timing of the DREQ external pin when data is transferred in a two-cycle transfer from an external area to an internal area)

• When data is transferred between an internal area and external area

Negate the external WRX pin output while it is low when the transfer source is accessed in the last DMA transfer (between DACK=L & WRX=L). If DREQ is negated after the above negation, the next data may be transferred.

● For fly-by (read/write) transfer

In demand transfer, make sure to set the address of an external area in the transfer destination register.

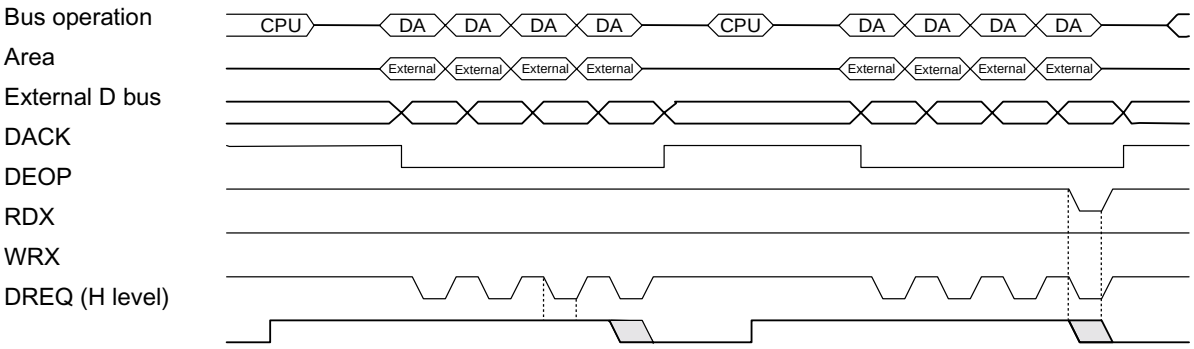
• fly-by (read)

Negate the external RDX pin output while it is low when the transfer source is accessed in

the last DMA transfer (between DACK=L & RDX=L). If DREQ is negated after the above negation, the nest data may be transferred.

- fly-by (write)

  Negate the external WRX pin output while it is low when the transfer source is accessed in the last DMA transfer (between DACK=L & RDX=L). If DREQ is negated after the above negation, the next data may be transferred.



(Example of the negate timing of DREQ external pin when data is transferred in a fly-by transfer (write))

- Timing of DREQ external pin to continuously transfer data over the same channel

  Data transfer in burst, step, block, or demand transfer mode
  The DREQ external pin input cannot assure continuous data transfer over the same channel. Even if DREQ is asserted again at the fastest timing to clear the request held internally at the transfer termination, detection of a transfer request from another channel is valid for one system clock cycle (for one cycle of CLK output). Therefore, if another channel has a higher priority, data transfer for that channel is started.
  Even if DREQ is asserted before above operation, the assertion is ignored because data transfer has not ended yet. If the transfer request is not issued from another channel, reassert DREQ when DACK pin output is asserted. Data transfer is restarted over the same channel.

- Timing of DACK pin output

  DACK output the FR50 series DMAC indicates that data has been transferred for a transfer request that has been accepted.

  DACK output basically synchronizes with the address output according to the external bus access timing. For DACK output to be used, it must be enabled in the port register.

- Timing of DEOP pin output

  DEOP output by FR50 series DMA indicates that DMA data has been transferred for the specified transfer count over the channel for which the transfer request was accepted.

  DEOP is output when access of the external area for the last block to be transferred is started. Therefore, if a value other than 1 is set (block transfer mode) for the block size, DEOP is output when the last data of the last block is transferred. In this case, if DACK pin output is asserted even while the data transfer is continuing (before DEOP is output), acceptance of the next DREQ is started.

  DEOP output synchronizes with RDX and WRX according to the external bus access timing. However, if the transfer source or transfer destination register is internally accessed, DEOP is not output. For DEOP output to be used, DEOP output must be enabled in the port register.

- Timing of DSTP external pin

When DMA transfer operates in all modes (such as burst, step, block, and demand transfer), the minimum size required is five system clock cycles (=1/2θ, two CPU system clock cycles).

To use DSTP input timing, synchronize it with external access in the same way as for DREQ.
(Use the signal decoded between DACK output and RDX or WRX.)

This timing is used to forcibly stop DMA data transfer. The use of this pin can forcibly stop data transfer. However, the status register (DMACB:DSS[2:0]) indicates a transfer stop request and is used to indicate an error. If the interrupt is enabled, it will occur.

Since this function is shared by the DEOP pin and DSTP pin, both pins cannot use the function simultaneously. Use the port register to switch the function.

- When an external pin transfer request is reinput during data transfer

  Data transfer in burst, step, or block transfer mode
  The next transfer request is not valid even though it is input before the DACK signal is asserted in the DMAC. However, the external bus control unit does not completely synchronize with DMAC operation. Therefore, the output of DACK and DEOP must initialize the circuit creating the DREQ external pin input to enable a transfer request by DREQ input.

  Data transfer in demand transfer mode
  If reloading the transfer count register is specified when data is transferred by transfer count, the transfer request can be accepted again.

- Another transfer request is issued during block transfer

  Another transfer request is not detected until the specified block has been completely transferred. Evaluate the transfer request accepted when data is transferred in block units and use the channel with the highest priority to transfer data.

- Data transfer between an external I/O device and external memory

  The DMAC does not distinguish between external I/O and external memory when transferring. Set the external I/O as a fixed external address.

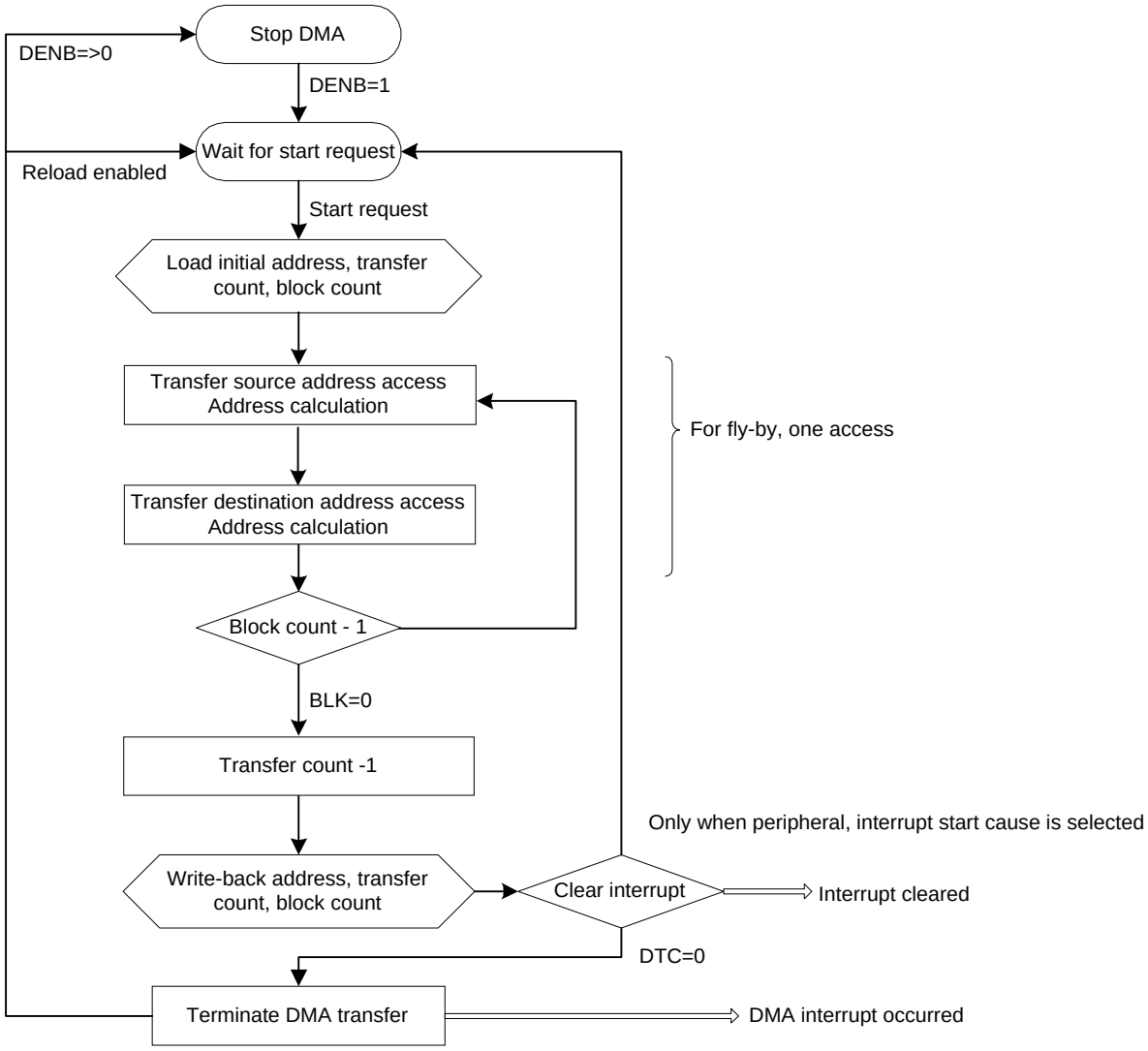
  To transfer data in fly-by mode, set the address of the external memory in the transfer destination address register. For an external I/O, use the signal decoded between the DACK output and RDX or WRX.

- DMAC AC characteristics

  The DREQ external pin, DACK pin output, and DEOP pin output are the external pins related to the DMAC. The output timing is synchronized with external bus access.

11.6 OPERATION FLOW

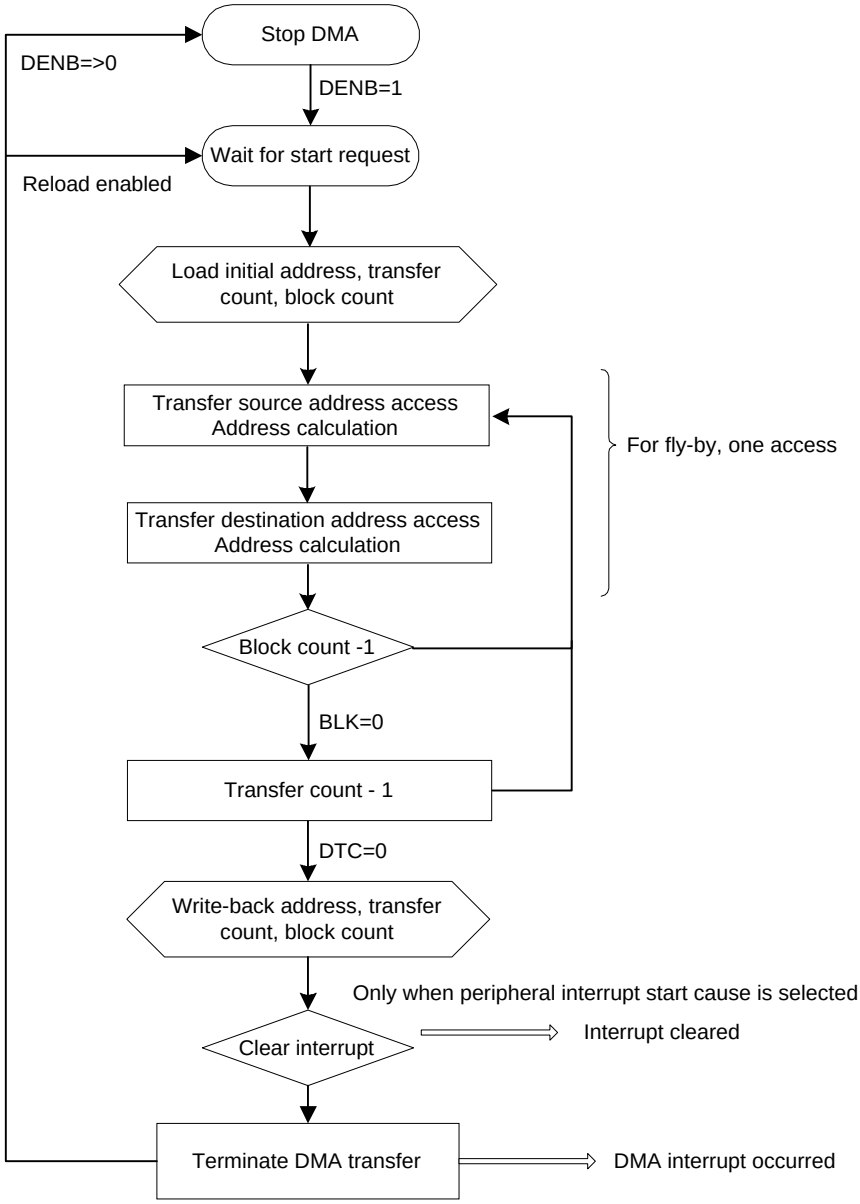
11.6.1 Block transfer



■ **Block transfer**

- Can be started by all start causes (select).
- Can access all areas.
- The block count can be set.
- Issues an interrupt clear when transfer specified by the block count terminates.
- Issues a DMA interrupt when transfer terminates for the specified count.

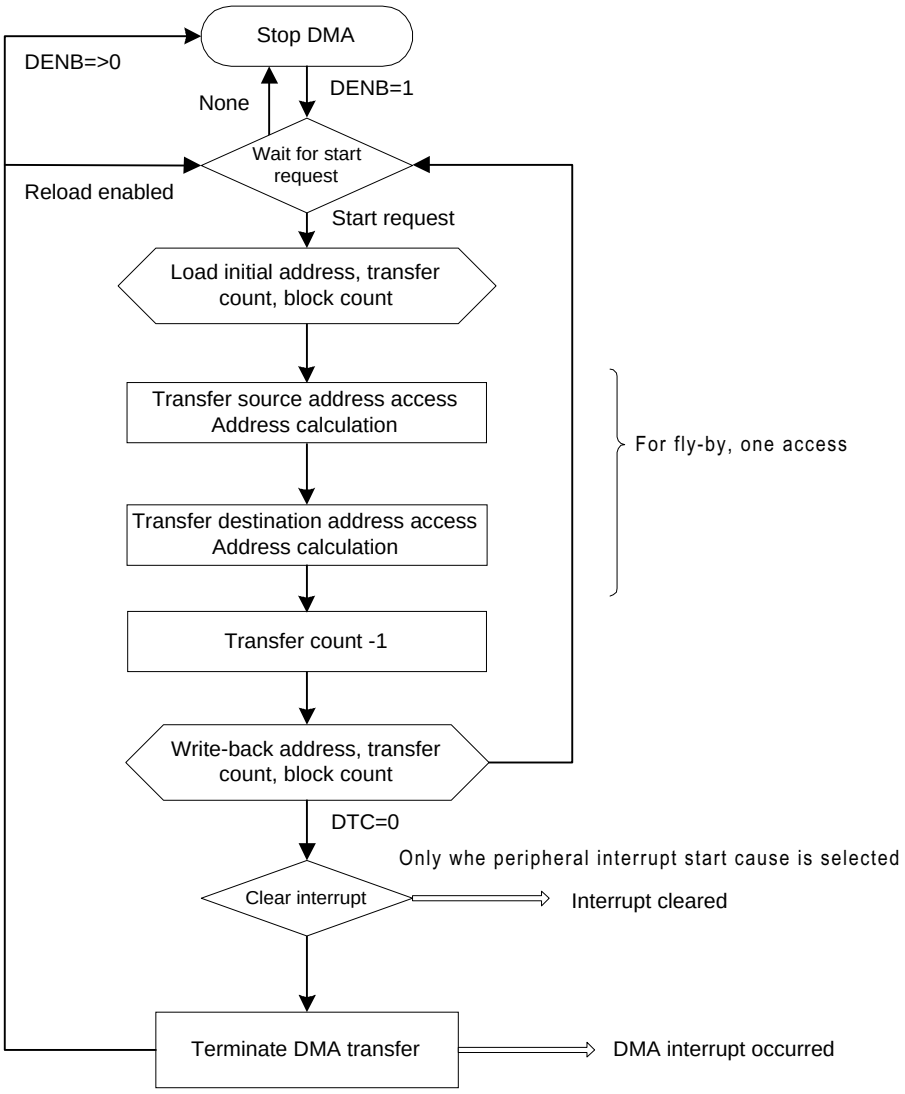
11.6.2 Burst transfer



■ **Burst transfer**

- Can be started by all start causes (select).
- Can access all areas.
- The block count can be set.
- Clears the interrupt and issues a DMA interrupt when transfer terminates for the specified count.

11.6.3 Demand transfer



■ **Demand transfer**

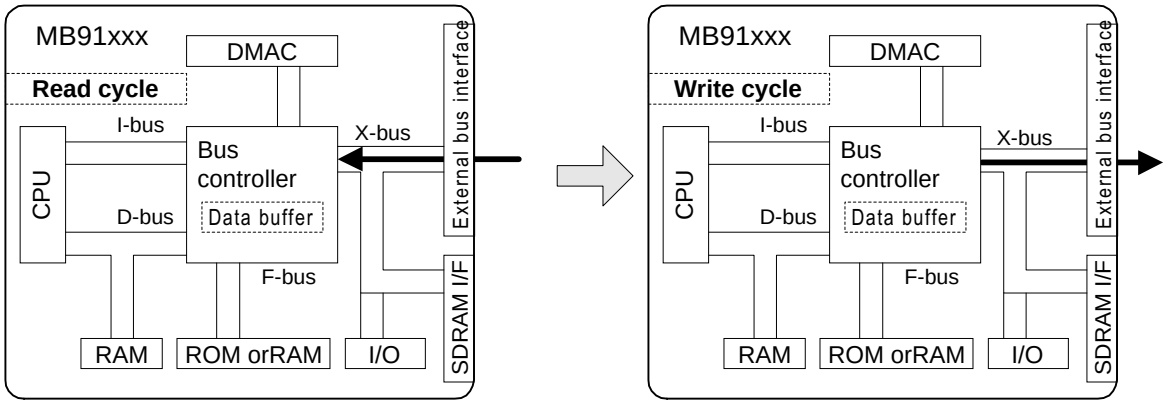
- Accepts only requests (level detection) from the external pin (DREQ). Starting by other causes is not allowed.
- An external area access is a required condition (because external area access becomes the next start cause).
- The block count is always 1 regardless of the setting.
- Clears the interrupt and issues a DMA interrupt when transfer terminates for the specified count.

11.7 DATA BUS

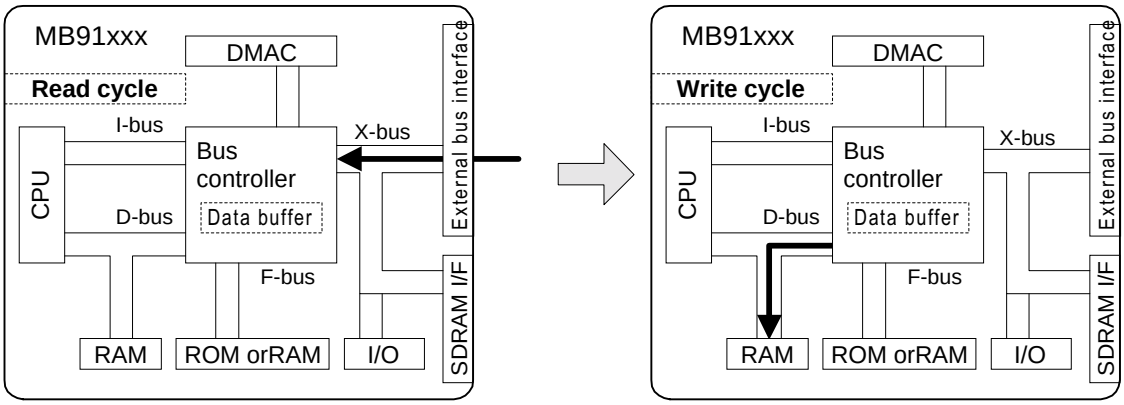
11.7.1 Data flow for two-cycle transfer

Five transfer examples are shown below with diagrams. (Other combinations are omitted.)

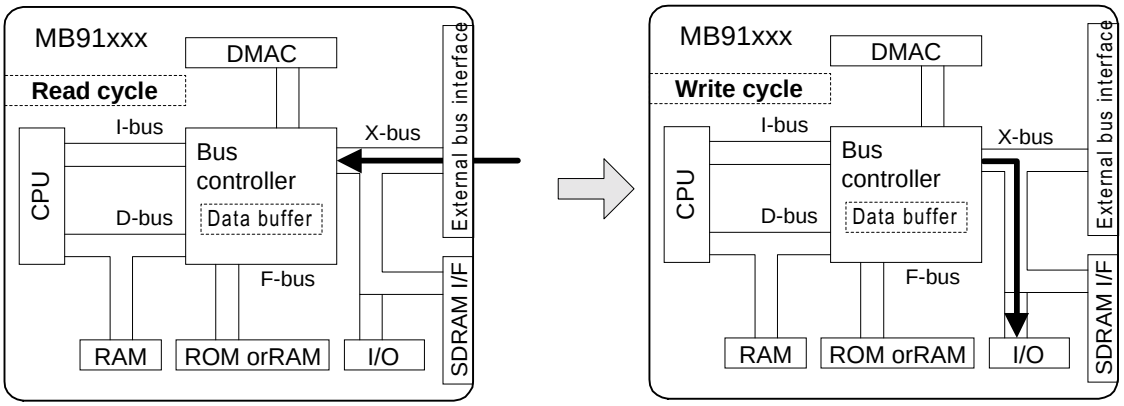
External area → external area transfer



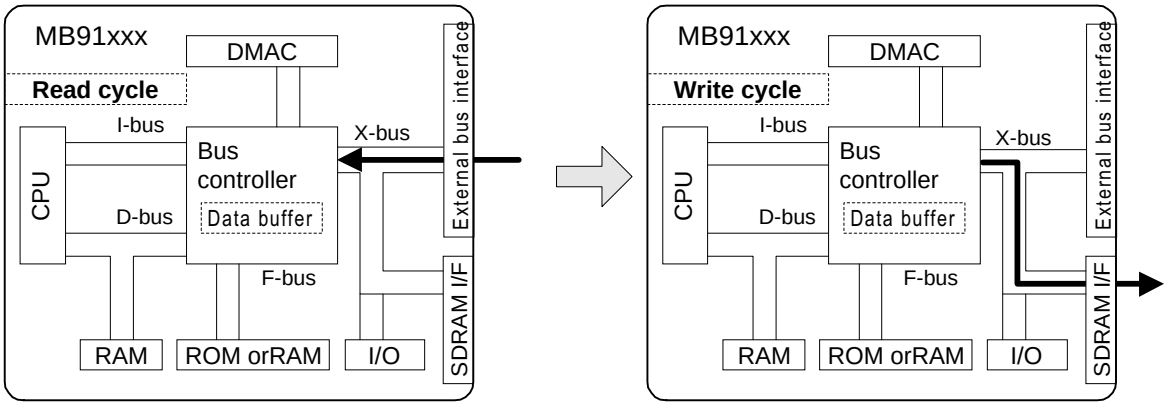
External area → internal RAM area transfer



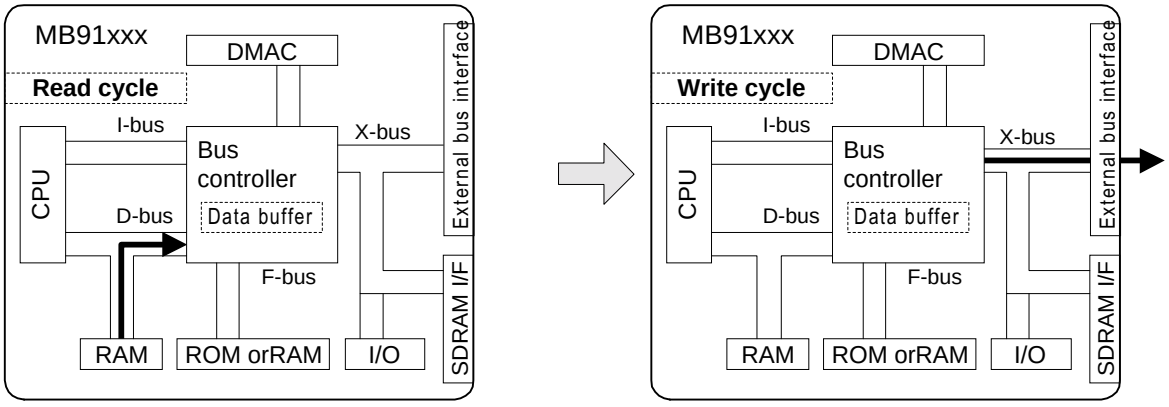
External area → internal I/O area transfer



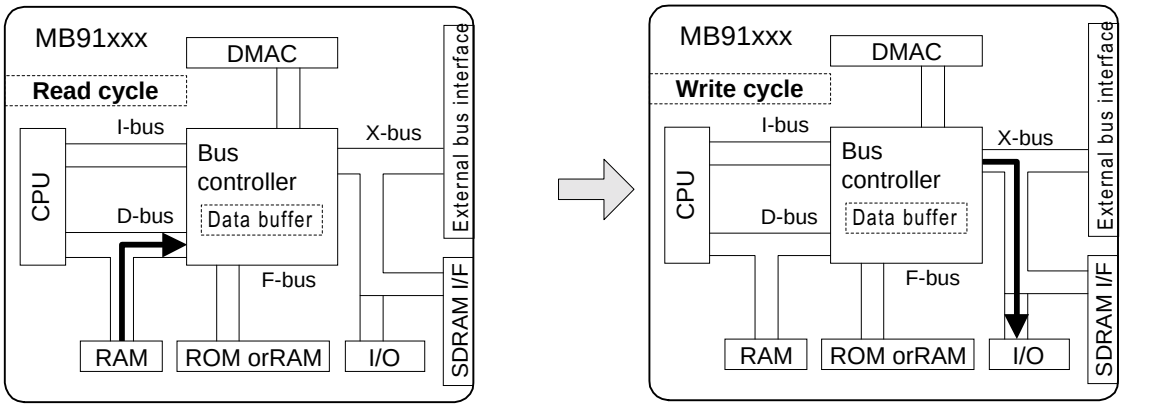
External area → SDRAM area transfer



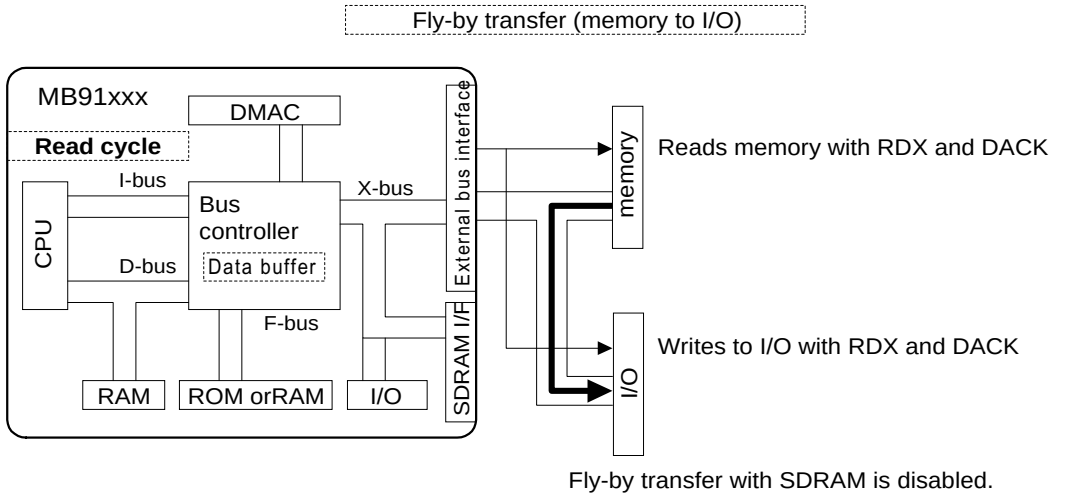
Internal RAM area → external area transfer

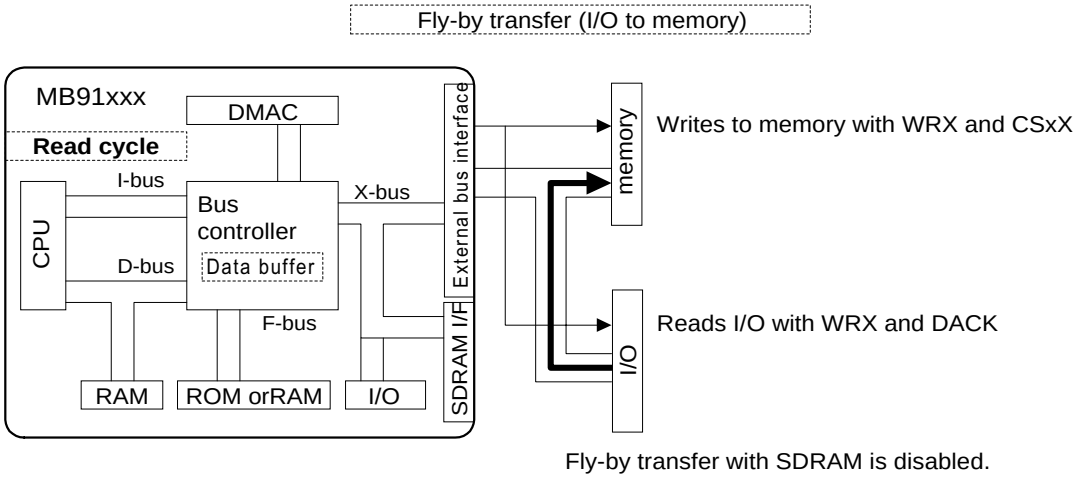


Internal RAM area → internal I/O area transfer



11.7.2 Data flow for fly-by transfer





For Fly-by transfer in addition to those signals shown above also IOWX (A26) and IORX (A25) can be used. In case of transfer from memory to I/O IOWX equals RDX, in case of transfers from I/O to memory IORX equals WRX.

11.8 EXTERNAL DMA SIGNALS

The external DMA transfer uses the external bus interface for transmitting adresses and data. Please see the related chapter for an explanation of these signals. In addition to the bus interface signals the following DMA control signals are used:

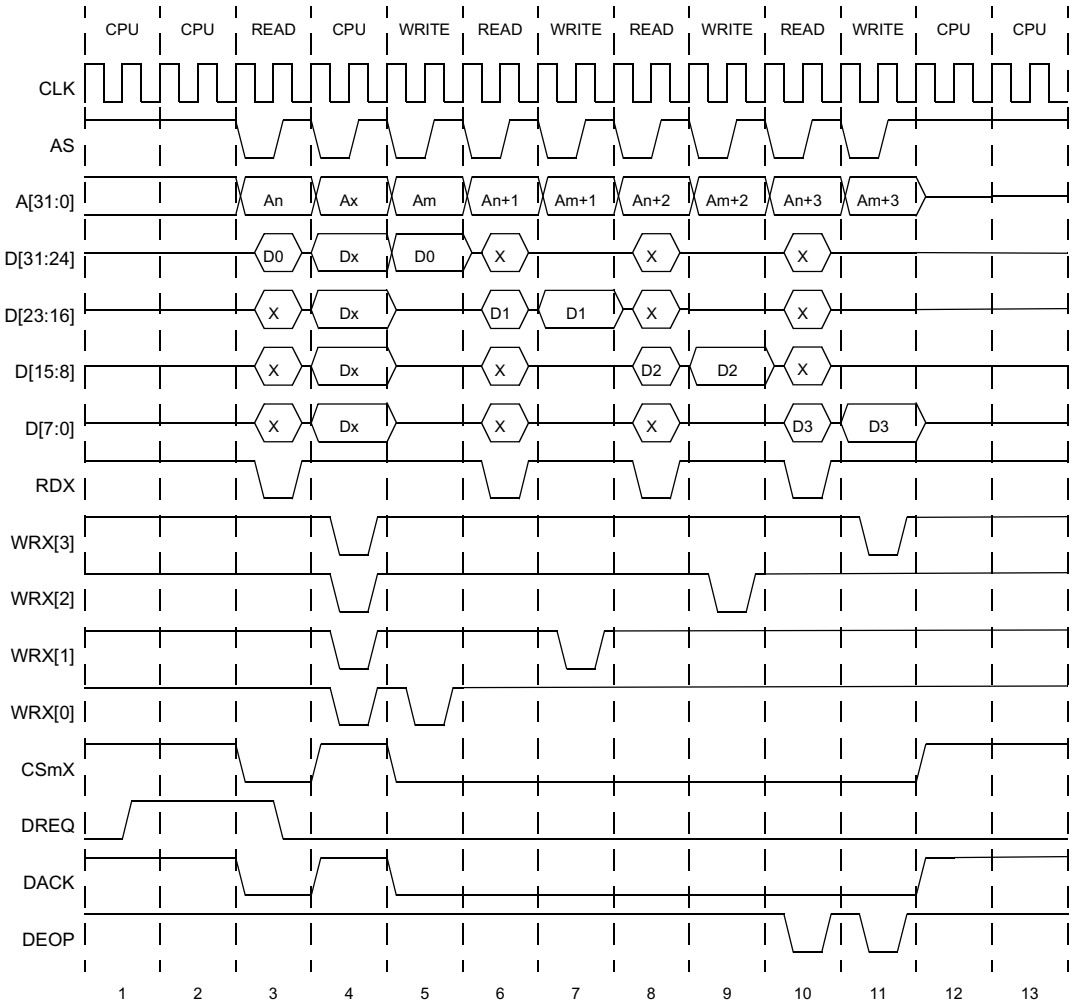
Table 11.8 External DMA Signals

Signal	Function	Related edge of CLKT	Direction	Notes
DREQ	External DMA request	rising edge	Input	
DACK	External DMA acknowledge	rising edge	Output	
DEOP	External DMA End of Process	falling edge	Output	1)
DSTP	External DMA STOP	rising edge	Input	1)

Note: DEOP and DSTP share the same pin, the required functionality must be defined via PFR.

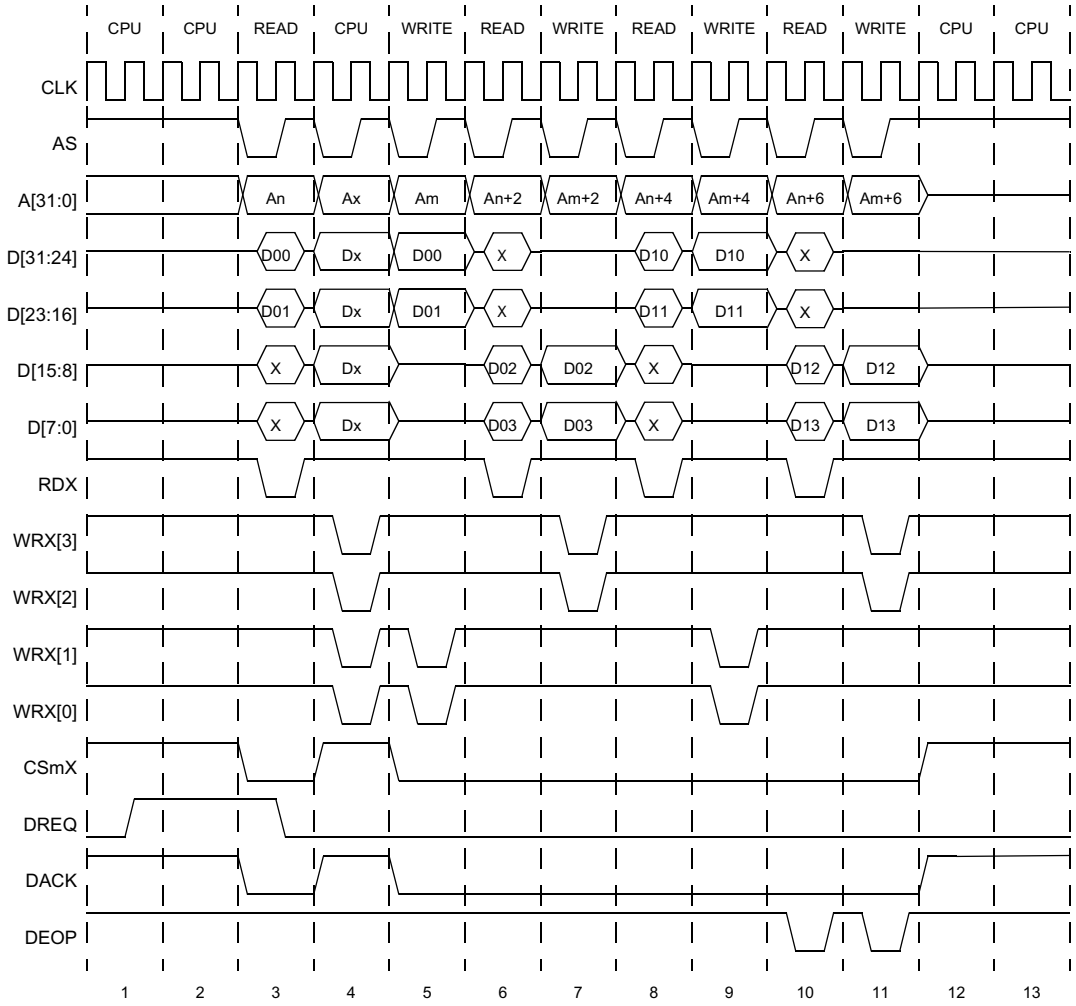
11.9 EXAMPLES

DMA Block Two Cycle Byte Transfer



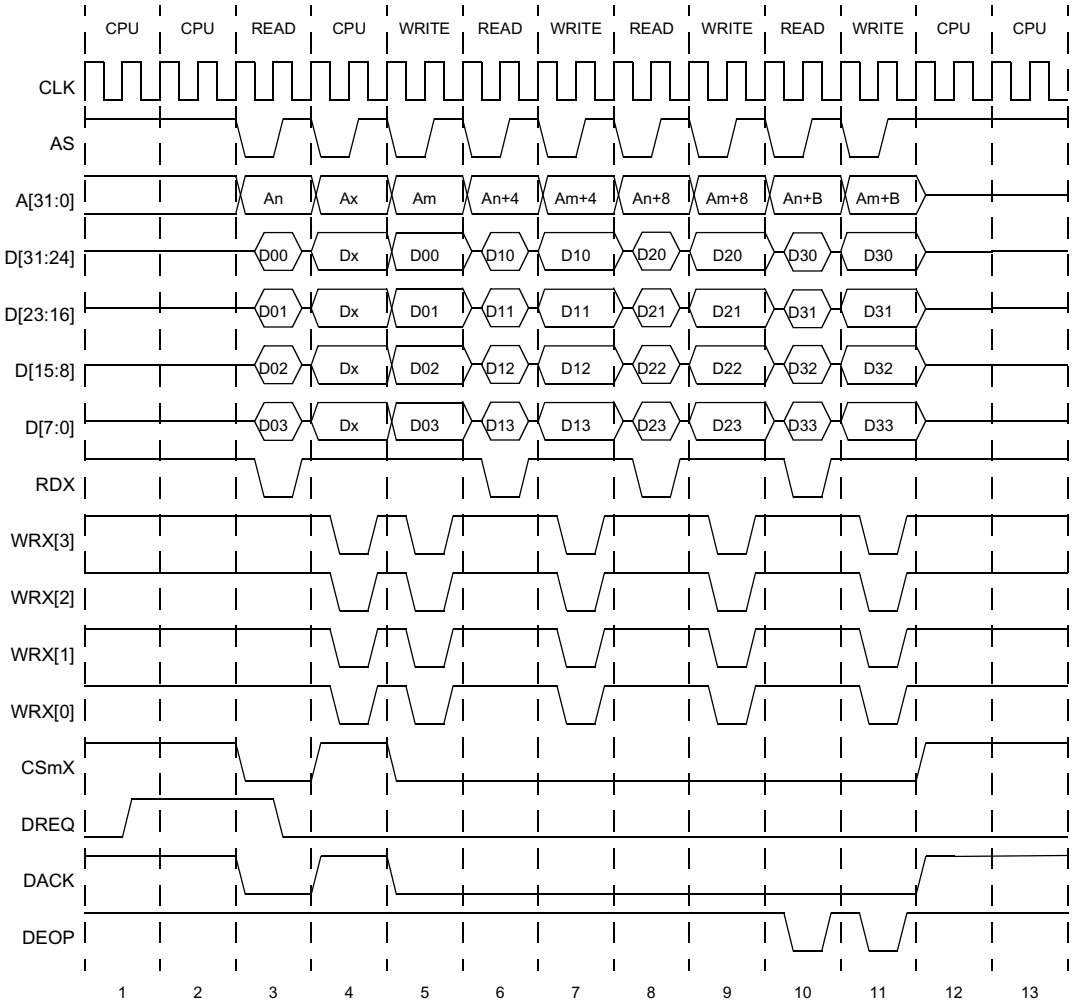
Section 1: Bus operation is CPU access (DACK=1). External DREQ becomes active to start the DMA transfer.
Section 2: Bus operation is still CPU access (DACK=1).
Section 3: Bus operation changes to DMA access (DACK=0). First byte (D0) is read (RDX=0) from DMA source address (An).
Section 4: DMA transfer is interrupted by CPU (DACK=1) write access.
Section 5: First byte (D0) is written (WRX[0]=0) to DMA destination address (Am).
Section 6: Second byte (D1) is read (RDX=0) from DMA source address (An+1).
Section 7: Second byte (D1) is written (WRX[1]=0) to DMA destination address (Am+1).
Section 8: Third byte (D2) is read (RDX=0) from DMA source address (An+2).
Section 9: Third byte (D2) is written (WRX[2]=0) to DMA destination address (Am+2).
Section 10: Forth byte (D3) is read (RDX=0) from DMA source address (An+3). This is the last DMA read access (DEOP=0).
Section 11: Forth byte (D3) is written (WRX[3]=0) to DMA destination address (Am+3). This is the last DMA write access (DEOP=0).
Section 12: DMA transfer is finished. Bus operation is CPU access (DACK=1).

DMA Burst Two Cycle Halfword Transfer



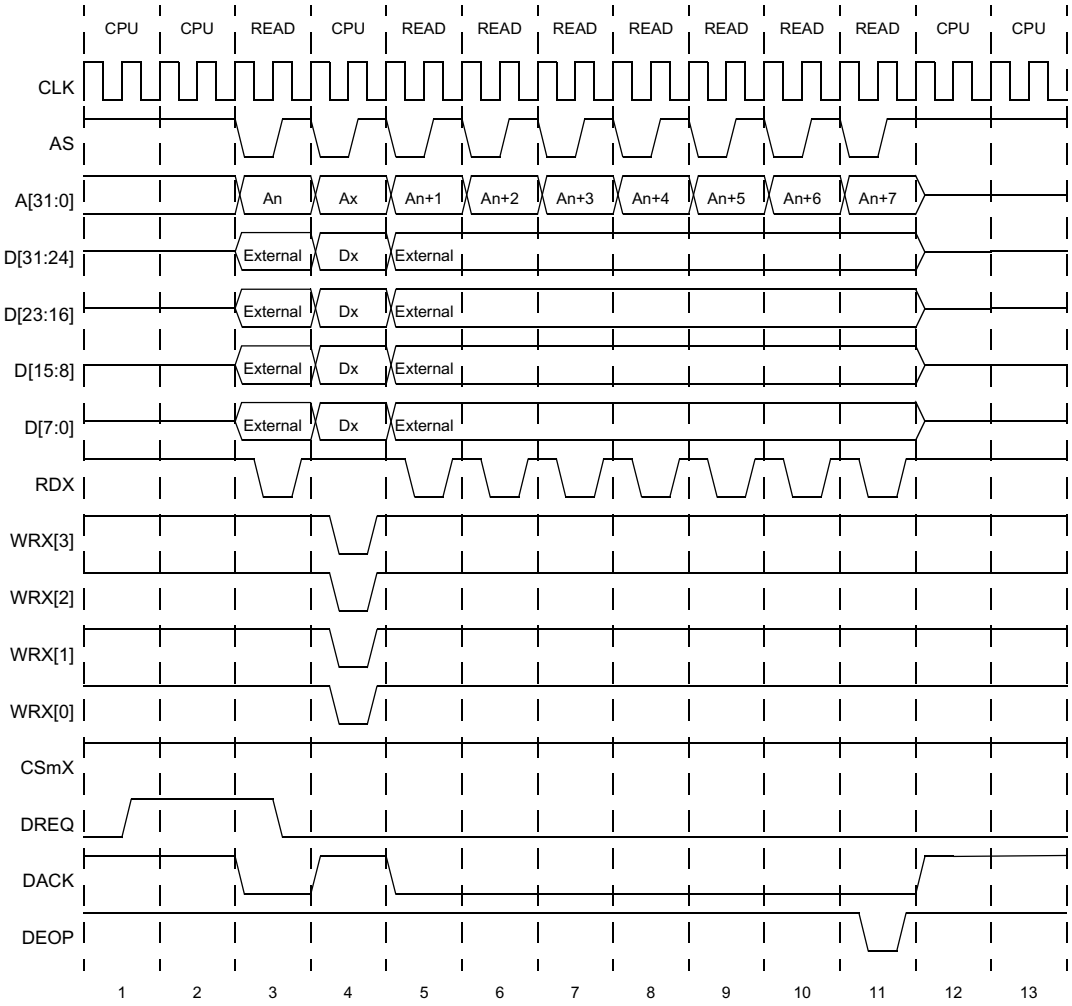
Section 1: Bus operation is CPU access (DACK=1). External DREQ becomes active to start the DMA transfer.
Section 2: Bus operation is still CPU access (DACK=1).
Section 3: Bus operation changes to DMA access (DACK=0). First halfword (D00, D01) is read (RDX=0) from DMA source address (An).
Section 4: DMA transfer is interrupted by CPU (DACK=1) write access.
Section 5: First halfword (D00, D01) is written (WRX[1:0]=00) to DMA destination address (Am).
Section 6: Second halfword (D02, D03) is read (RDX=0) from DMA source address (An+2).
Section 7: Second halfword (D02, D03) is written (WRX[3:2]=00) to DMA destination address (Am+2).
Section 8: Third halfword (D10, D11) is read (RDX=0) from DMA source address (An+4).
Section 9: Third halfword (D10, D11) is written (WRX[1:0]=00) to DMA destination address (Am+4).
Section 10: Forth halfword (D12, D13) is read (RDX=0) from DMA source address (An+6). This is the last DMA read access (DEOP=0).
Section 11: Forth halfword (D12, D13) is written (WRX[3:2]=00) to DMA destination address (Am+6). This is the last DMA write access (DEOP=0).
Section 12: DMA transfer is finished. Bus operation is CPU access (DACK=1).

DMA Burst Two Cycle Word Transfer



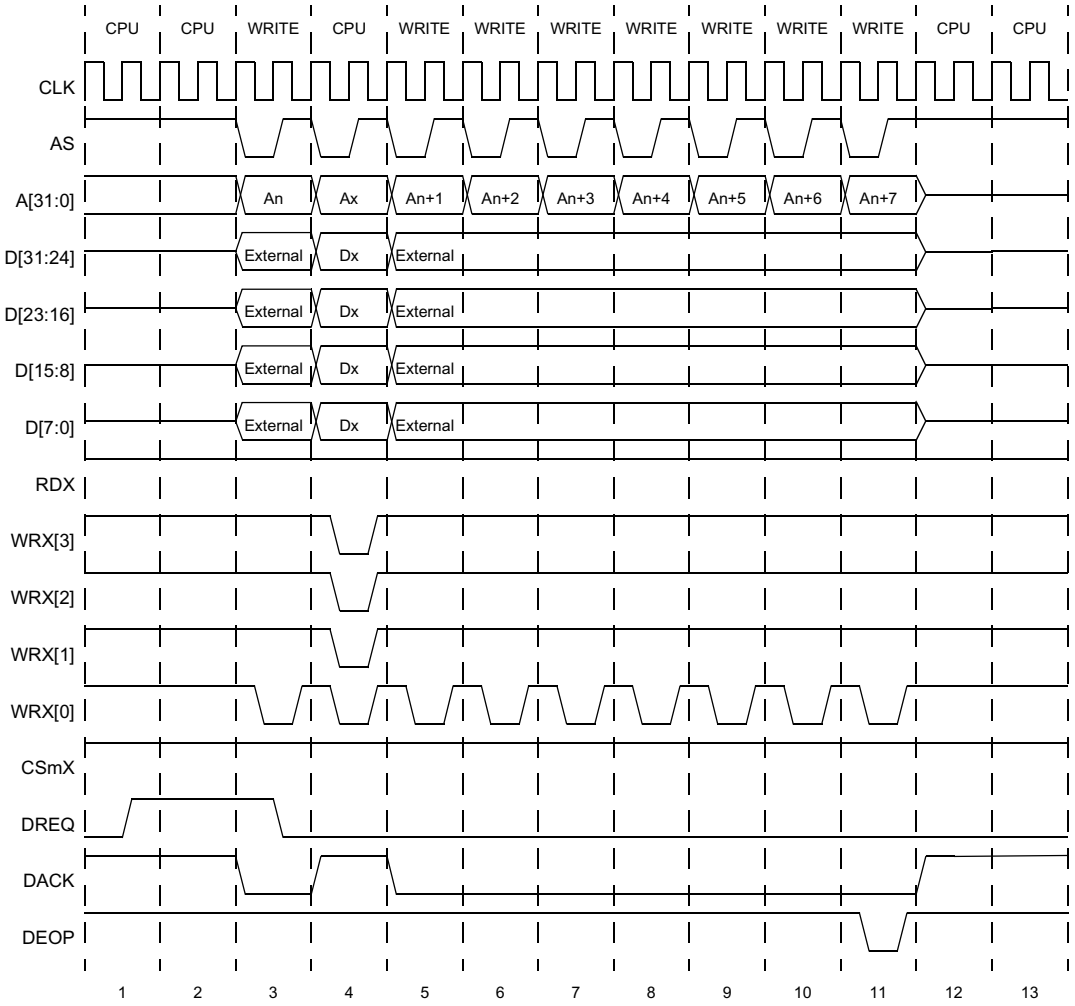
Section 1: Bus operation is CPU access (DACK=1). External DREQ becomes active to start the DMA transfer.
Section 2: Bus operation is still CPU access (DACK=1).
Section 3: Bus operation changes to DMA access (DACK=0). First word (D00, D01, D02, D03) is read (RDX=0) from DMA source address (An).
Section 4: DMA transfer is interrupted by CPU (DACK=1) write access.
Section 5: First word (D00, D01, D02, D03) is written (WRX[3:0]=0000) to DMA destination address (Am).
Section 6: Second word (D10, D11, D12, D13) is read (RDX=0) from DMA source address (An+4).
Section 7: Second word (D10, D11, D12, D13) is written (WRX[3:0]=0000) to DMA destination address (Am+4).
Section 8: Third word (D20, D21, D22, D23) is read (RDX=0) from DMA source address (An+8).
Section 9: Third word (D20, D21, D22, D23) is written (WRX[3:0]=0000) to DMA destination address (Am+8).
Section 10: Forth word (D30, D31, D32, D33) is read (RDX=0) from DMA source address (An+B). This is the last DMA read access (DEOP=0).
Section 11: Forth word (D30, D31, D32, D33) is written (WRX[3:0]=0000) to DMA destination address (Am+B). This is the last DMA write access (DEOP=0).
Section 12: DMA transfer is finished. Bus operation is CPU access (DACK=1).

DMA Fly-by Byte Transfer (Memory to IO)



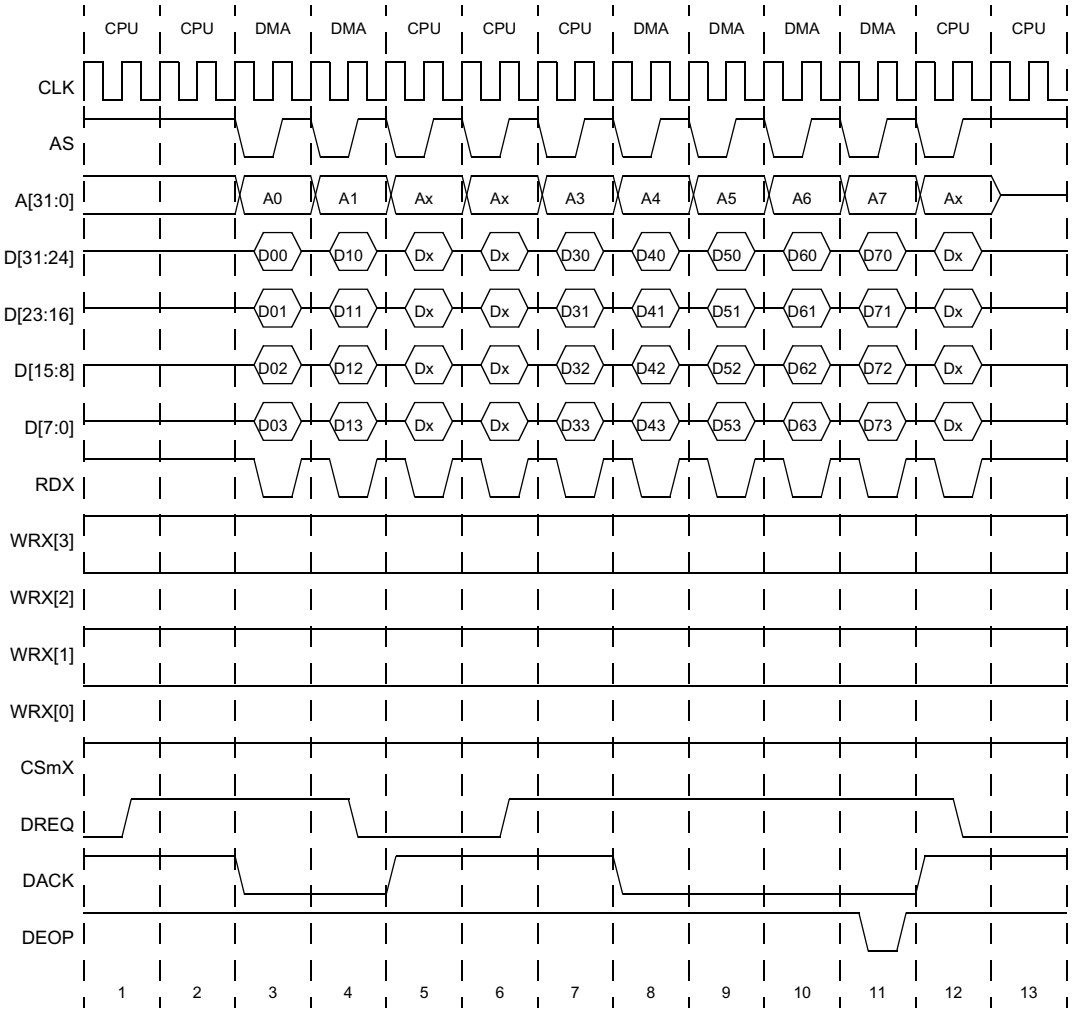
Section 1: Bus operation is CPU access (DACK=1). External DREQ becomes active to start the DMA transfer.
Section 2: Bus operation is still CPU access (DACK=1).
Section 3: Bus operation changes to DMA access (DACK=0). First byte is read (RDX=0) from DMA source address (An). Data bus is assigned to external component.
Section 4: DMA transfer is interrupted by CPU (DACK=1) write access.
Section 5: Same as section 3.
Section 6-10: Bus operation is DMA access (DACK=0).
Section 11: This is the last DMA read access (DEOP=0).

DMA Fly-by Byte Transfer (IO to Memory)



Section 1: Bus operation is CPU access (DACK=1). External DREQ becomes active to start the DMA transfer.
Section 2: Bus operation is still CPU access (DACK=1).
Section 3: Bus operation changes to DMA access (DACK=0). First byte is written (WRX[0]=0) to DMA destination address (An). Data bus is assigned to external component.
Section 4: DMA transfer is interrupted by CPU (DACK=1) write access.
Section 5: Same as section 3.
Section 6-10: Bus operation is DMA access (DACK=0).
Section 11: This is the last DMA write access (DEOP=0).

DMA Demand Word Transfer



Section 1: Bus operation is CPU access. External DREQ becomes active to start the DMA transfer.
Section 2: Bus operation is still CPU access.
Section 3: Bus operation changes to DMA access. This is flaged by DACK -> 0. First DMA address (A0) and data (D0) are read (RDX=0).
Section 4: Second DMA address (A1) and data (D1) are read. DMA transfer is interrupted by external DREQ -> 0.
Section 5: Bus operation is CPU access (DACK=1).
Section 6: External DREQ becomes active again to continue DMA transfer.
Section 7: Same as section 2.
Section 8: Same as section 3.
Section 9: DMA transfer.
Section 10: DMA transfer.
Section 11: Last DMA transfer. This is flaged by DEOP pulses 0.
Section 12: DMA is finished. DREQ becomes inactive as reaction on DACK. Bus operation is CPU access.

CHAPTER 12 MODULES FOR OS SUPPORT

This Chapter provides an overview of the delayed interrupt and the bit search module and describes their register structure and functions, operation, and save and restore processing for the search module.

12.1 DELAYED INTERRUPT

■ Delayed Interrupt Control Register (DICR)

The delayed interrupt control register (DICR) is a delayed interrupt generator register and is used to generate the task switching interrupt.

● Structure of the DICR

	Bits								
Address	7	6	5	4	3	2	1	0	Initial value
00000044H	–	–	–	–	–	–	–	DLYI	-------0
								R/W	⇦ Access

Figure 12.1a Structure of the Delayed Interrupt Control Register

● Functions of the DICR bits

[Bit 0] DLYI

DLYI	Description
0	Clear delayed interrupt (no request). (Initial value)
1	Generate delayed interrupt.

This bit controls generating and clearing the delayed interrupt.

● Operation:

Delayed interrupts are generated for task switching. Software can use the delayed interrupt function to generate or cancel interrupt requests to the CPU.

(1) Interrupt number

Delayed interrupts are assigned to the interrupt source having the highest interrupt number.

The MB91360 assigns interrupt number 63 (3FH) to delayed interrupts.

(2) DLY1 bit in the DICR register

Setting the DLY1 bit in the DICR register to "1" generates a delayed interrupt. Setting the bit to "0" cancels the delayed interrupt.

This bit has the same effect as the interrupt source flag for general interrupts. In the interrupt routine, clear the bit and specify task switching.

12.2 BIT SEARCH MODULE

This chapter provides an overview of the bit search module and describes the register structure and functions, bit search module operation, and save and restore processing.

12.2.1 Overview of the Bit Search Module

The bit search module searches for a "0", "1", or change-point in the data written to the input register and returns the position of the detected bit.

■ **Register Configuration of the Bit Search Module**

Register name	Name	Address	Register structure 31	0
Data register for detecting zeros	BSD0	0000 03F0H		
Data register for detecting ones	BSD1	0000 03F4H		
Data register for detecting change-points	BSDC	0000 03F8H		
Detection result register	BSRR	0000 03FCH		

Figure 12.2.1a Register Configuration of the Bit Search Module

■ **Block Diagram of the Bit Search Module**

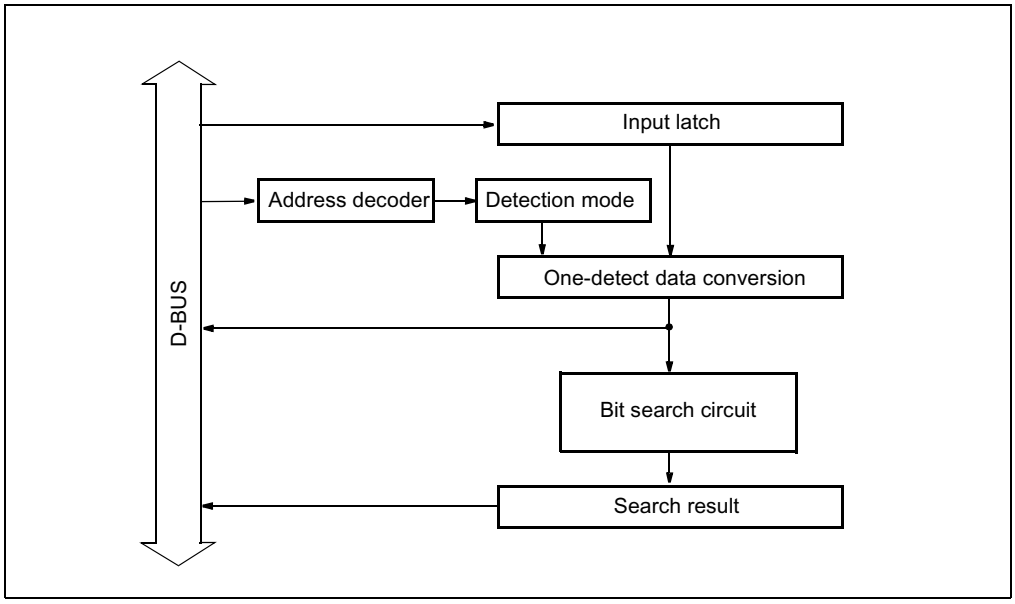


Figure 12.2.1b Block Diagram of the Bit Search Module

12.2.2 Bit Search Module Registers

This section describes the data register for detecting zeros (BSD0), data register for detecting ones (BSD1), data register for detecting change-points (BSDC), and detection result register (BSRR).

■ **Data Register for Detecting Zeros (BSD0)**

Address	31 Register structure 0	Initial value	Access
0000 03F0H		Indeterminate	W

Figure 12.2.2a Structure of the Data Register for Detecting Zeros

Used to detect zeros in the value written to the register.

The initial value after a reset is indeterminate.

The read value is indeterminate.

Only use 32-bit length data transfer instructions to transfer data to the register. (Do not use 8-bit length or 16-bit length data transfer instructions.)

■ **Data Register for Detecting Ones (BSD1)**

Address	31 Register structure 0	Initial value	Access
0000 03F4H		Indeterminate	R/W

Figure 12.2.2b Structure of the Data Register for Detecting Ones

Only use 32-bit length data transfer instruction to transfer data to the register. (Do not use 8-bit length or 16-bit length data transfer instructions.)

- Writing

  Used to detect ones in the value written to the register.

- Reading

  Reads data for saving the internal state of the bit search module.

  Use this register to save and restore the state of the bit search module when using the module in an interrupt handler or similar. Saving and restoring can be performed using this register even when performing zero-detection, change-point detection, and writing data to a data register.

  The initial value after a reset is indeterminate.

■ **Data Register for Detecting Change-Points (BSDC)**

Address	31	Register structure	0	Initial value	Access
0000 03F8H				Indeterminate	W

Figure 12.2.2c Structure of the Data Register for Detecting Change-points

Used to detect the change-point in the value written to the register.

The initial value after a reset is indeterminate.

The read value is indeterminate.

Only use 32-bit length data transfer instructions to transfer data to the register. (Do not use 8-bit length or 16-bit length data transfer instructions.)

■ **Detection Result Register (BSRR)**

Address	31	Register structure	0	Initial value	Access
0000 03FCH				Indeterminate	R

Figure 12.2.2d Structure of the Detection Result Register

The result of zero-detection, one-detection, or change-point detection can be read from this register.

The most recent data register to be written to determines the type of detection result read from this register.

12.2.3 Bit Search Module Operation and Save and Restore Processing

---

This section describes the operation of zero-detection, one-detection and change-point detection, and save and restore processing.

■ **Zero-detection**

The bit search module scans data written to the data register for detecting zeros from the MSB (most siginificant bit) to the LSB (least Siginificant bit) and returns the position of the first "0".

The search result is obtained by reading the detection result register.

Table 12.2.3 "Bit Position and Returned Value (Decimal)" on page 353 lists the relationship between the detected position and the returned value.

The bit search module returns the value "32" if the data does not contain any zeros (in other words, if the value is "FFFFFFFFH").

[Execution example]

Data written to the register	Read value (decimal)

11111111111111111111000000000000B (FFFFF000H)	⇨	20
11111000010010011110000010101010B (F849E0AAH)	⇨	5
10000000000000101010101010101010B (8002AAAAH)	⇨	1
11111111111111111111111111111111B (FFFFFFFFH)	⇨	32

■ One-detection

The bit search module scans data written to the data register for detecting ones from the MSB (most siginificant bit) to the LSB (least Siginificant bit) and returns the position of the first "1".

The search result is obtained by reading the detection result register.

Table 15.3 lists the relationship between the detected position and the returned value.

The bit search module returns the value "32" if the data does not contain any ones (in other words, if the value is "0000000H").

[Execution example]

Data written to the register		Read value (decimal)
00100000000000000000000000000000B (20000000H)	⇨	2
00000001001000110100010101100111B (01234567H)	⇨	7
00000000000000111111111111111111B (0003FFFFH)	⇨	14
00000000000000000000000000000001B (00000001H)	⇨	31
00000000000000000000000000000000B (00000000H)	⇨	32

■ Change-point Detection

The bit search module scans data written to the data register for detecting change-points from bit 30 to the LSB and compares each bit with the MSB value.

The module returns the position of the first bit with a different value to the MSB.

The search result is obtained by reading the detection result register.

Table 12.2.3 "Bit Position and Returned Value (Decimal)" on page 353 lists the relationship between the detected position and the returned value.

The bit search module returns the value "32" if the data does not contain a change-point.

[Execution example]

Data written to the register		Read value (decimal)
00100000000000000000000000000000B (20000000H)	⇨	2
00000001001000110100010101100111B (01234567H)	⇨	7
00000000000000111111111111111111B (0003FFFFH)	⇨	14
00000000000000000000000000000001B (00000001H)	⇨	31
00000000000000000000000000000000B (00000000H)	⇨	32
11111111111111111111000000000000B (FFFF000H)	⇨	20
11111000010010011110000010101010B (F849E0AAH)	⇨	5
10000000000000101010101010101010B (8002AAAAH)	⇨	1
11111111111111111111111111111111B (FFFFFFFFH)	⇨	32

Table 12.2.3 Bit Position and Returned Value (Decimal)

Detected bit position	Returned value	Detected bit position	Returned value	Detected bit position	Returned value	Detected bit position	Returned value
31	0	23	8	15	16	7	24
30	1	22	9	14	17	6	25
29	2	21	10	13	18	5	26
28	3	20	11	12	19	4	27
27	4	19	12	11	20	3	28
26	5	18	13	10	21	2	29
25	6	17	14	9	22	1	30
24	7	16	15	8	23	0	31
–		–		–		None	32

■ Save/Restore Processing

Use the following procedure if you need to save/restore the internal state of the bit search module, such as when using the bit search module in an interrupt handler.

1) Read the data register for detecting ones and save the contents (save).
2) Use the bit search module.
3) Write the data saved in step 1) to the data register for detecting ones (restore).

As a result of this procedure, the next time the detection result register is read, the read value is the value resulting from the contents written to the bit search module prior to step 1) above.

Even if the last data register to be written to was the data register for detecting zeros or the data register for detecting change-points, the above procedure correctly restores the module to its original state.

CHAPTER 13 PWM TIMER

This chapter provides an overview of the PWM timer, describes the register structure and functions, and describes the operation of the PWM timer.

13.1 OVERVIEW OF THE PWM TIMER

The PWM (Pulse Width Modulation) timer can output high-precision pulse waves at an arbitrary cycle and pulse width (duty ratio).
The MB91360 contains eight PWM timer channels. Each of the channels consists of a 16-bit down-counter, cycle setting register, duty setting register, and pin controller.
The control status register for each channel is used to indicate the operation status of the PWM timer. General control registers 1 and 2 are common registers shared by four channels, serving for input and software triggering.

■ Features

- The count clock for the 16-bit down-counter can be selected from among the following four types:

  Internal clocks: ϕ, ϕ/4, ϕ/16, ϕ/64 (ϕ: Clock for peripherals, CLKP)

- The counter can be initialized to "FFFF$_H$" by a reset or underflow.
  The 16-bit down-counter causes an underflow when it changes from "0000$_H$" to "FFFF$_H$".

- Each channel has PWM outputs.

  Eight channels: Eight output pins

- Registers

  Cycle setting register: Data reload register with buffer

  Data transfer from the buffer is performed either when an activation trigger is detected or when the down-counter causes an underflow (cycle match). The output is inverted at a cycle match.

  Duty setting register: Compare register with buffer

  The value set in this register is compared to the counter value. The output is inverted when the values match (duty match).

- Pin control

  A duty match causes a reset to "1" (given priority).

  An underflow causes a reset to "0".

  The output value fix mode enables output of all "L" or all "H".

  The polarity can also be specified.

- Interrupt requests can be generated by selecting the following interrupt sources:

  Activation of the PWM timer (software trigger or trigger input)

  Occurrence of an underflow (cycle match)

  Occurrence of a duty match

  Occurrence of an underflow (cycle match) or duty match

- You can set simultaneous activation of two or more channels using software or another interval timer. You can also set restarting the PWM timer during operation.

■ PWM Timer Block Diagram

- Configuration diagram of the entire PWM timer

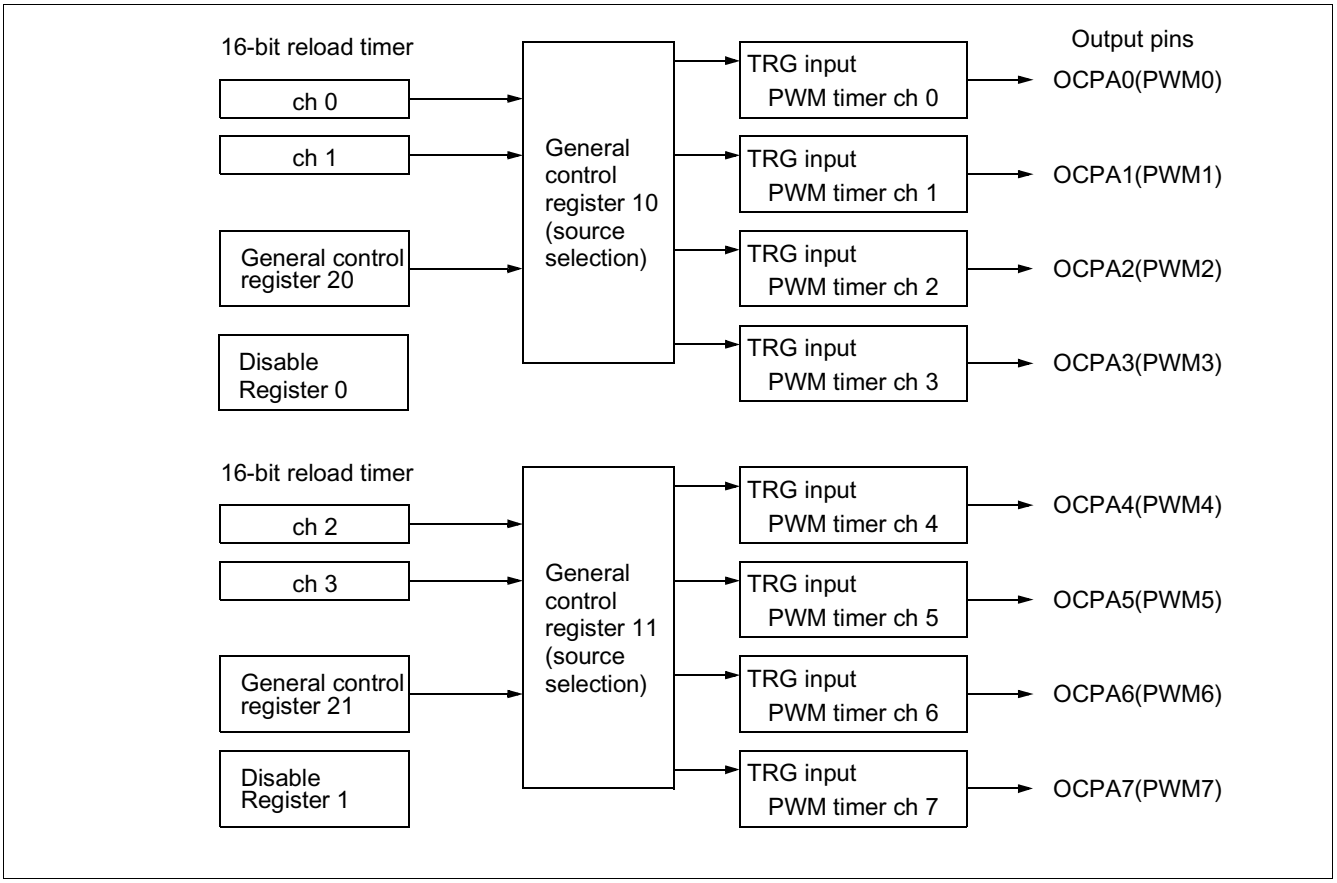


Figure 13.1a Configuration Diagram of the Entire PWM Timer

- Configuration diagram of PWM timer channel 1

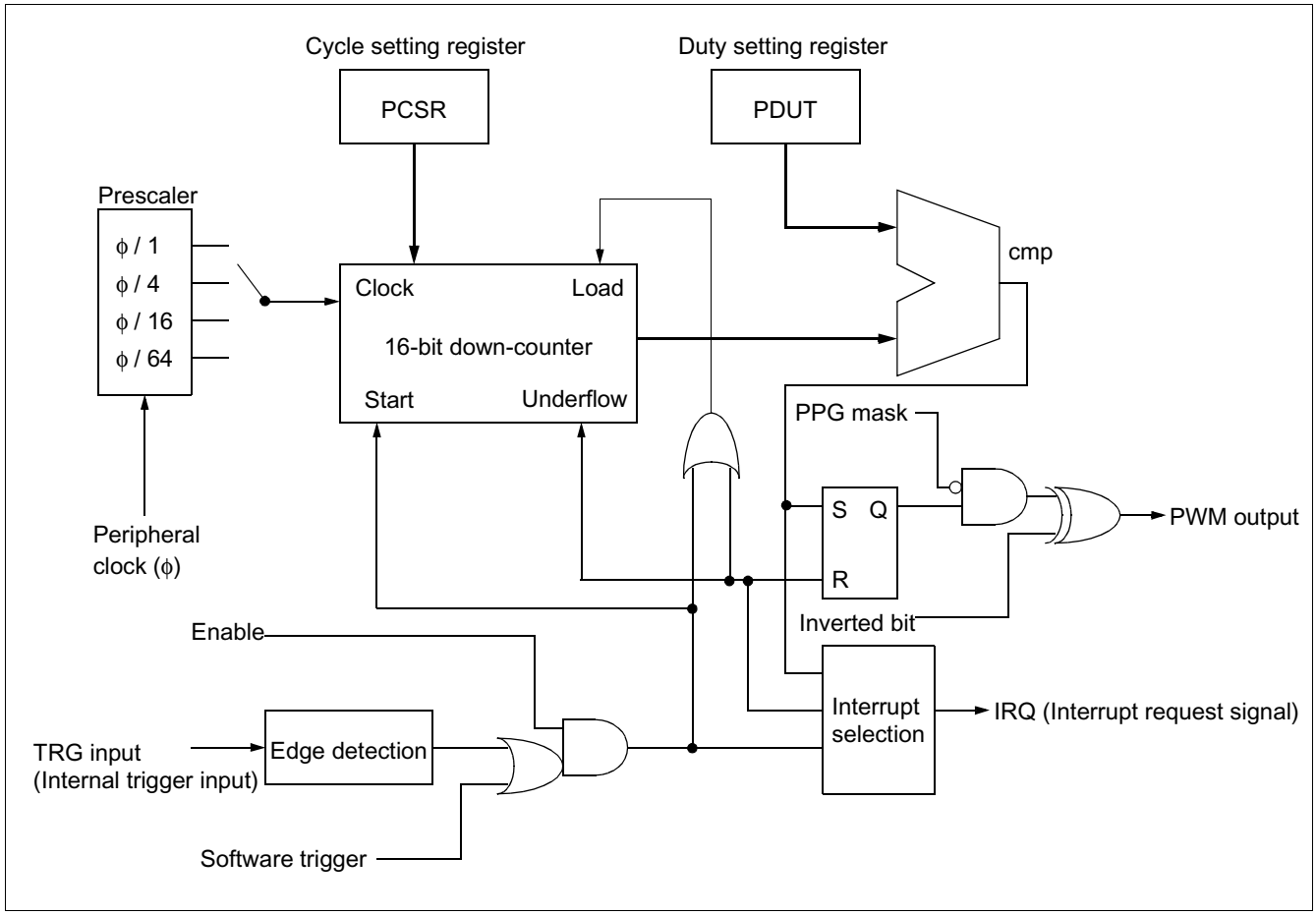


Figure 13.1b Configuration Diagram of PWM Timer Channel 1

13.2 PWM TIMER REGISTERS

This section lists the PWM timer registers and details their functions.

■ PWM Timer Registers for channels 0 - 3

Address	Bits 15 8	7 0	Access	Register name
0000 0118H	GCN10		R/W	General control register 10
0000 011AH	PDBL0	GCN20	R/W	Disable/General control register 20
PWM timer ch 0				
0000 0120H	PTMR		R	ch 0 timer register
0000 0122H	PCSR		W	ch 0 cycle setting register
0000 0124H	PDUT		W	ch 0 duty setting register
0000 0126H	PCNH	PCNL	R/W	ch 0 control status registers
PWM timer ch 1				
0000 0128H	PTMR		R	ch 1 timer register
0000 012AH	PCSR		W	ch 1 cycle setting register
0000 012CH	PDUT		W	ch 1 duty setting register
0000 012EH	PCNH	PCNL	R/W	ch 1 control status registers
PWM timer ch 2				
0000 0130H	PTMR		R	ch 2 timer register
0000 0132H	PCSR		W	ch 2 cycle setting register
0000 0134H	PDUT		W	ch 2 duty setting register
0000 0136H	PCNH	PCNL	R/W	ch 2 control status registers
PWM timer ch 3				
0000 0138H	PTMR		R	ch 3 timer register
0000 013AH	PCSR		W	ch 3 cycle setting register
0000 013CH	PDUT		W	ch 3 duty setting register
0000 013EH	PCNH	PCNL	R/W	ch 3 control status registers

■ PWM Timer Registers for channels 4 - 7

Address	Bits 15 – 8	Bits 7 – 0	Access	Register name
0000 011CH	GCN11		R/W	General control register 11
0000 011EH	PDBL1	GCN21	R/W	Disable/General control register 21
PWM timer ch 4				
0000 0140H	PTMR		R	ch 4 timer register
0000 0142H	PCSR		W	ch 4 cycle setting register
0000 0144H	PDUT		W	ch 4 duty setting register
0000 0146H	PCNH	PCNL	R/W	ch 4 control status registers
PWM timer ch 5				
0000 0148H	PTMR		R	ch 5 timer register
0000 014AH	PCSR		W	ch 5 cycle setting register
0000 014CH	PDUT		W	ch 5 duty setting register
0000 014EH	PCNH	PCNL	R/W	ch 5 control status registers
PWM timer ch 6				
0000 0150H	PTMR		R	ch 6 timer register
0000 0152H	PCSR		W	ch 6 cycle setting register
0000 0154H	PDUT		W	ch 6 duty setting register
0000 0156H	PCNH	PCNL	R/W	ch 6 control status registers
PWM timer ch 7				
0000 0158H	PTMR		R	ch 7 timer register
0000 015AH	PCSR		W	ch 7 cycle setting register
0000 015CH	PDUT		W	ch 7 duty setting register
0000 015EH	PCNH	PCNL	R/W	ch 7 control status registers

Figure 13.2 PWM Timer Registers

13.2.1 Control Status Registers (PCNH, PCNL)

The PWM timer control status registers (PCNH and PCNL) is a register for controlling operation of the PWM timer and indicating the status.

■ **Structure of the Control Status Registers (PCNH, PCNL)**

PCNH	Address	Bits 15	14	13	12	11	10	9	8	Initial value	Access
		CNTE	STGR	MDSE	RTRG	CKS1	CKS0	PGMS	–		
ch 0	0000 0126H									0000000-B	R/W
ch 1	0000 012EH										
ch 2	0000 0136H										
ch 3	0000 013EH										
ch 4	0000 0146H										
ch 5	0000 014EH										
ch 6	0000 0156H										
ch 7	0000 015EH										

PCNL	Address	7	6	5	4	3	2	1	0	Initial value	Access
		EGS1	EGS0	IREN	IRQF	IRS1	IRS0	-	OSEL		
ch 0	0000 0127H									000000-0B	R/W
ch 1	0000 012FH										
ch 2	0000 0137H										
ch 3	0000 013FH										
ch 4	0000 0147H										
ch 5	0000 014FH										
ch 6	0000 0157H										
ch 7	0000 015FH										

Figure 13.2.1 Configuration of the Control Status Registers

■ **Functions of the PCNH/PCNL Bits**

[Bit 15] CNTE: Timer operation enable bit

This bit enables or disables operation of the 16-bit down-counter.

0	Disable timer operation.	[Initial value]
1	Enable timer operation.	

[Bit 14] STGR: Software trigger bit

Setting this bit to "1" causes a software trigger.

The bit returns a value of "0" whenever read.

[Bit 13] MDSE: Operation mode select bit

This bit selects PWM operation for generating a continuous stream of pulses or one-shot operation for generating a single pulse.

0	PWM operation	[Initial value]
1	One-shot operation	

[Bit 12] RTRG: Restart enable bit

This bit enables or disables restarting the PWM timer using a software trigger or trigger input.

0	Disables restarting. [Initial value]
1	Enables restarting.

[Bits 11, 10] CKS1, CKS0: Counter clock select bits

These bits select the count clock for the 16-bit down-counter.

Table 13.2.1a Selecting the Count Clock

CKS1	CKS0	Count clock source
0	0	ϕ/1 [Initial value]
0	1	$\phi/2^2$
1	0	$\phi/2^4$
1	1	$\phi/2^6$

ϕ : Clock for peripherals (CLKP)

[Bit 9] PGMS: PWM output mask select bit

When set to "1", this bit masks the PWM output to "0" or "1" regardless of the mode, cycle, and duty settings.

Table 13.2.1b PWM Output with PGMS set to "1"

Polarity	PWM output
Normal polarity	"L" level output [Initial value]
Inverted polarity	"H" level output

For all "H" output with normal polarity or all "L" output with inverted polarity, set the cycle setting and duty setting registers to the same value to produce the inverted output of the above mask value.

For normal and inverted polarities, see the description of [Bit 0].

[Bit 8]: Unused bit

[Bits 7, 6] EGS1, EGS0: Trigger input edge select bits

These bits select the effective edge for the trigger input source selected by general control register 1 (GCN1). Setting the software trigger bit (STGR) to "1" enables software triggering regardless of the currently selected mode.

Table 13.2.1c Selecting the Trigger Input Edge

EGS1	EGS0	Trigger input edge
0	0	Invalid [Initial value]
0	1	Rising edge
1	0	Falling edge
1	1	Both (rising/falling) edges

[Bit 5] IREN: Interrupt request enable bit

This bit enables or disables interrupt requests.

0	Disable interrupt requests.	[Initial value]
1	Enable interrupt requests.	

[Bit 4] IRQF: Interrupt request flag

When the interrupt source selected by bits 3 and 2 (IRS1 and IRS0) is generated with bit 5 (IREN) enabling interrupt requests, this bit is set, generating an interrupt to the CPU.

If activation of DMA transfer has been selected, DMA transfer is activated.

This bit is cleared by writing "0" to it or using the clear signal from the DMAC.

Writing "1" to this bit does not change the bit value.

When read by a read modify write instruction, the bit returns "1" regardless of the bit value.

[Bits 3, 2] IRS1, IRS0: Interrupt source select bits

These bits select the interrupt source that sets bit 4 (IRQF).

Table 13.2.1d Selecting the Interrupt Source

IRS1	IRS0	Interrupt source
0	0	Generation of a software trigger or trigger input[Initial value]
0	1	Occurrence of an underflow (cycle match)
1	0	Occurrence of a duty match
1	1	Occurrence of an underflow (cycle match) or duty match

[Bit 0] OSEL: PWM output polarity select bit

This bit sets the PWM output polarity.

The bit is used in combination with bit 9 (PGMS) to specify the following:

Table 13.2.1e Specifying the PWM Output Polarity and Edge

PGMS	OSEL	PWM output
0	0	Normal polarity [Initial value]
0	1	Inverted polarity
1	0	Fixed at "L" output
1	1	Fixed at "H" output

Polarity	After reset	Duty match	Under-flow
Normal polarity	"L" output		
Inverted polarity	"H" output		

13.2.2 PWM Cycle Setting Register (PCSR)

The PWM cycle setting register (PCSR) is a 16-bit reload register with buffer for setting the count value required for output of a pulse signal with an arbitrary cycle.

■ **Structure of the PWM Cycle Setting Register (PCSR)**

	Address	Bits 15	14	13	12	11	10	9	8	Initial value	Access
ch 0	0000 0122H	D15	D14	D13	D12	D11	D10	D09	D08	XXXXXXXXB	W
ch 1	0000 012AH										
ch 2	0000 0132H										
ch 3	0000 013AH	7	6	5	4	3	2	1	0		
ch 4	0000 0142H	D07	D06	D05	D04	D03	D02	D01	D00	XXXXXXXXB	W
ch 5	0000 014AH										
ch 6	0000 0152H										
ch 7	0000 015AH										

Figure 13.2.2 Structure of the PWM Cycle Setting Register

■ **Functions of the PCSR**

The PWM cycle setting register (PCSR) sets the counter value at which the 16-bit down-counter starts counting. The set value is loaded into the counter when the counter activation trigger is detected and when the counter causes an underflow (from "0000H" to "FFFFH").

The down-counter causes an underflow (cycle match) when it reaches "the set value + 1" count after starting counting, when the output pulse signal cycle is determined. The cycle is the value obtained by multiplying "the count clock cycle" by "the set value + 1" count.

For initializing or updating the cycle, be sure to write to the duty setting register after writing to the cycle setting register.

To write to this register, access it using 16-bit data.

13.2.3 PWM Duty Setting Register (PDUT)

The PWM duty setting register (PDUT) is a 16-bit compare register with buffer for setting the compare value required for output of a signal with an arbitrary pulse width.

■ **Structure of the PWM Duty Setting Register (PDUT)**

	Address	Bits 15	14	13	12	11	10	9	8	Initial value	Access
ch 0	0000 0124H	D15	D14	D13	D12	D11	D10	D09	D08	XXXXXXXXB	W
ch 1	0000 012CH										
ch 2	0000 0134H										
ch 3	0000 013CH										
ch 4	0000 0144H	D07	D06	D05	D04	D03	D02	D01	D00	XXXXXXXXB	W
ch 5	0000 014CH										
ch 6	0000 0154H										
ch 7	0000 015CH										

7	6	5	4	3	2	1	0	

Figure 13.2.3 Structure of the PWM Duty Setting Register

■ **Functions of the PDUT**

The PWM duty setting register (PDUT) sets the counter value for determining the PWM output pulse width, which is compared to the count value in the down-counter. When the values match, the PWM output polarity is inverted.

The down-counter continues counting with the same polarity. When it reaches "the set value + 1" count to cause an underflow (cycle match), the polarity is inverted and the pulse width is determined. The pulse width is the value obtained by multiplying "the count clock cycle" by "the set value + 1" count.

Setting the cycle setting and duty setting registers to the same value produces all "H" output with normal polarity or all "L" output with inverted polarity.

The PCSR and PDUT register values should be set as "PCSR > PDUT". Setting the register values as "PCSR < PDUT" results in undefined PWM output.

To write to this register, access it using 16-bit data.

13.2.4 PWM Timer Register (PTMR)

The PWM timer register (PTMR) is used to read the 16-bit down-counter value.

■ **Structure of the PWM Timer Register (PTMR)**

	Address	Bits 15	14	13	12	11	10	9	8	Initial value	Access
ch 0	0000 0120H	D15	D14	D13	D12	D11	D10	D09	D08	111111111B	R
ch 1	0000 0128H										
ch 2	0000 0130H										
ch 3	0000 0138H	7	6	5	4	3	2	1	0		
ch 4	0000 0140H	D07	D06	D05	D04	D03	D02	D01	D00	11111111B	R
ch 5	0000 0148H										
ch 6	0000 0150H										
ch 7	0000 0158H										

Figure 13.2.4 Structure of the PWM Timer Register

■ **Functions of the PTMR**

The PWM timer register (PTMR) is a 16-bit data register from which the down-counter value is read.

To read a value from this register, access it using 16-bit data.

13.2.5 General Control Register 1 (GCN10,GCN11)

General control register 1 (GCN10,GCN11) selects the PWM timer trigger input source. Four bits are assigned for each channel.

■ **Structure of general control register 1 (GCN10,GCN11)**

Address	Bits 15	14	13	12	11	10	9	8	Initial value	Access
ch0-ch3: 0000 0118H ch4-ch7: 0000 011CH	TSEL33 to 30				TSEL23 to 20				00110010B	R/W
	7	6	5	4	3	2	1	0		
	TSEL13 to 10				TSEL03 to 00				00010000B	R/W

Figure 13.2.5 Structure of General Control Register 1

■ **Functions of the GCN10**

[Bits 15 to 12] TSEL33 to TSEL30: ch 3 input select bits

These bits select the trigger input to PWM timer channel 3.

Table 13.2.5a Selecting the PWM Timer ch 3 Input

TSEL33 to TSEL30				ch 3 trigger input
15	14	13	12	
0	0	0	0	GCN20 EN0 bit
0	0	0	1	GCN20 EN1 bit
0	0	1	0	GCN20 EN2 bit
0	0	1	1	GCN20 EN3 bit[Initial value]
0	1	0	0	16-bit reload timer ch 0
0	1	0	1	16-bit reload timer ch 1
0	1	1	×	Setting prohibited
1	0	×	×	Setting prohibited
1	1	×	×	Setting prohibited

[Bits 11 to 8] TSEL23 to TSEL20: ch 2 trigger input select bits

These bits select the trigger input to PWM timer channel 2.

Table 13.2.5b Selecting the PWM Timer ch 2 Trigger Input

TSEL23 to TSEL20				ch 2 trigger input
11	10	9	8	
0	0	0	0	GCN20 EN0 bit
0	0	0	1	GCN20 EN1 bit
0	0	1	0	GCN20 EN2 bit[Initial value]
0	0	1	1	GCN20 EN3 bit
0	1	0	0	16-bit reload timer ch 0
0	1	0	1	16-bit reload timer ch 1
0	1	1	×	Setting prohibited
1	0	×	×	Setting prohibited
1	1	×	×	Setting prohibited

[Bits 7 to 4] TSEL13 to TSEL10: ch 1 trigger input select bits

These bits select the trigger input to PWM timer channel 1.

Table 13.2.5c Selecting the PWM Timer ch 1 Trigger Input

TSEL13 to TSEL10				ch 1 trigger input
7	6	5	4	
0	0	0	0	GCN20 EN0 bit
0	0	0	1	GCN20 EN1 bit[Initial value]
0	0	1	0	GCN20 EN2 bit
0	0	1	1	GCN20 EN3 bit
0	1	0	0	16-bit reload timer ch 0
0	1	0	1	16-bit reload timer ch 1
0	1	1	×	Setting prohibited
1	0	×	×	Setting prohibited
1	1	×	×	Setting prohibited

[Bits 3 to 0] TSEL03 to TSEL00: ch 0 trigger input select bits

These bits select the trigger input to PWM timer channel 0.

Table 13.2.5d Selecting the PWM Timer ch 0 Trigger Input

TSEL03 to TSEL00				ch 0 trigger input
3	2	1	0	
0	0	0	0	GCN20 EN0 bit[Initial value]
0	0	0	1	GCN20 EN1 bit
0	0	1	0	GCN20 EN2 bit
0	0	1	1	GCN20 EN3 bit
0	1	0	0	16-bit reload timer ch 0
0	1	0	1	16-bit reload timer ch 1
0	1	1	×	Setting prohibited
1	0	×	×	Setting prohibited
1	1	×	×	Setting prohibited

■ **Functions of the GCN11**

[Bits 15 to 12] TSEL33 to TSEL30: ch 7 input select bits

These bits select the trigger input to PWM timer channel 7.

Table 13.2.5e Selecting the PWM Timer ch 7 Input

TSEL33 to TSEL30				ch 7 trigger input
15	14	13	12	
0	0	0	0	GCN21 EN0 bit
0	0	0	1	GCN21 EN1 bit
0	0	1	0	GCN21 EN2 bit
0	0	1	1	GCN21 EN3 bit[Initial value]
0	1	0	0	16-bit reload timer ch 2
0	1	0	1	16-bit reload timer ch 3
0	1	1	×	Setting prohibited
1	0	×	×	Setting prohibited
1	1	×	×	Setting prohibited

[Bits 11 to 8] TSEL23 to TSEL20: ch 6 trigger input select bits

These bits select the trigger input to PWM timer channel 6.

Table 13.2.5f Selecting the PWM Timer ch 6 Trigger Input

TSEL23 to TSEL20				ch 6 trigger input
11	10	9	8	
0	0	0	0	GCN21 EN0 bit
0	0	0	1	GCN21 EN1 bit
0	0	1	0	GCN21 EN2 bit[Initial value]
0	0	1	1	GCN21 EN3 bit
0	1	0	0	16-bit reload timer ch 2
0	1	0	1	16-bit reload timer ch 3
0	1	1	×	Setting prohibited
1	0	×	×	Setting prohibited
1	1	×	×	Setting prohibited

[Bits 7 to 4] TSEL13 to TSEL10: ch 5 trigger input select bits

These bits select the trigger input to PWM timer channel 5.

Table 13.2.5g Selecting the PWM Timer ch 5 Trigger Input

TSEL13 to TSEL10				ch 5 trigger input
7	6	5	4	
0	0	0	0	GCN21 EN0 bit
0	0	0	1	GCN21 EN1 bit[Initial value]
0	0	1	0	GCN21 EN2 bit
0	0	1	1	GCN21 EN3 bit
0	1	0	0	16-bit reload timer ch 2
0	1	0	1	16-bit reload timer ch 3
0	1	1	×	Setting prohibited
1	0	×	×	Setting prohibited
1	1	×	×	Setting prohibited

[Bits 3 to 0] TSEL03 to TSEL00: ch 4 trigger input select bits

These bits select the trigger input to PWM timer channel 4.

Table 13.2.5h Selecting the PWM Timer ch 4 Trigger Input

TSEL03 to TSEL00				ch 4 trigger input
3	2	1	0	
0	0	0	0	GCN21 EN0 bit[Initial value]
0	0	0	1	GCN21 EN1 bit
0	0	1	0	GCN21 EN2 bit
0	0	1	1	GCN21 EN3 bit
0	1	0	0	16-bit reload timer ch 2
0	1	0	1	16-bit reload timer ch 3
0	1	1	×	Setting prohibited
1	0	×	×	Setting prohibited
1	1	×	×	Setting prohibited

13.2.6 Disable/General Control Register 2 (GCN20, GCN21)

General control register 2 (GCN20, GCN21) is used to generate an activation trigger using software. The Disable registers (PDBL0,PDBL1) can be used to selectively disable the clock for one or more timer and channels.

■ Structure of general control register 2 (GCN20,GCN21)

	Bits									
Address	7	6	5	4	3	2	1	0	Initial value	Access
GCN20: 0000 011BH	–	–	–	–	EN3	EN2	EN1	EN0	00000000B	R/W
GCN21: 0000 011FH										

Figure 13.2.6a Structure of General Control Register 2

■ Functions of the GCN20

If the EN0 to EN3 bits in this register are selected by general control register 1 (GCN1), the register value is passed to the PWM timer trigger input as it is.

Multiple PWM timer channels can be activated at the same time by using software to generate the edge selected by bits 7 and 6 (EGS1 and EGS0) in the control status register (PCNL).

Be sure to write "0" to bits 7 to 4 in this register.

■ Functions of the GCN21

If the EN0 to EN3 bits in this register are selected by general control register 1 (GCN11), the register value is passed to the PWM timer trigger input as it is.

Multiple PWM timer channels can be activated at the same time by using software to generate the edge selected by bits 7 and 6 (EGS1 and EGS0) in the control status register (PCNL).

Be sure to write "0" to bits 7 to 4 in this register.

■ Structure of the Disable Registers (PDBL0, PDBL1)

	Bits									
Address	7	6	5	4	3	2	1	0	Initial value	Access
PDBL0: 0000 011AH	–	–	–	–	DBL3	DBL2	DBL1	DBL0	----0000B	R/W
	7	6	5	4	3	2	1	0		
PDBL1: 0000 011EH	–	–	–	–	DBL7	DBL6	DBL5	DBC4	----0000B	R/W

Figure 13.2.6b Structure of General Control Register 2

■ Functions of the PDBL0, PDBL1

Setting one of the bits to "1" will disable the clock for the corresponding pwm channel.

13.3 PWM TIMER OPERATION

This section describes the operation of the PWM timer.
There are two PWM output operation modes: PWM operation and one-shot operation modes.

■ PWM Timer Operation

The PWM operation mode outputs a continuous stream of pulses; the one-shot operation mode outputs a single pulse.

This section also discusses interrupts and how to produce all "L" and all "H" PWM outputs. The items covered are shown below:

- PWM operation
- One-shot operation
- Interrupts
- All "L" and all "H" PWM outputs

13.3.1 PWM Operation

The PWM operation mode outputs a continuous stream of pulses.

■ PWM Operation

The PWM operation mode outputs a continuous stream of pulses from the time at which an activation trigger is detected.

The output pulse cycle can be controlled by changing the value in the cycle setting register (PCSR).

The output pulse width can also be controlled by changing the value in the duty setting register (PDUT).

When setting the output pulse cycle and width (duty ratio), be sure to write data to the PDUT register after writing data to the PCSR register.

■ PWM Output Timing

When the activation trigger is detected and the down-counter is reset, the value set in the cycle setting register (PCSR) is loaded into the counter and the down-counter starts counting.

The counter continues counting and the value set in the duty setting register (PDUT) is compared to the counter value. When the values match (duty match), the PWM output polarity is inverted.

If the counter causes an underflow in this state when trigger restarting is disabled, the value set in the PCSR register is loaded into the counter (cycle match) and the PWM output polarity is inverted. Now the pulse signal with the set cycle and pulse width has been output. From then on, the pulse signal is output continuously by the same procedure.

When trigger restarting has been enabled, the restarting trigger is enabled. The value set in the PCSR register is reloaded into the counter and the counter restarts counting.

Figure 13.3.1a shows a timing example in which the PWM output has normal polarity and trigger restarting has been disabled. Figure 13.3.1b shows a timing example in which the PWM output has normal polarity and trigger restarting has been enabled.

[Equations for calculating the PWM output signal cycle and pulse width]

Pm = T (m + 1) µs	Pm: Output pulse cycle
Pn = T (n + 1) µs	Pn: Output pulse width
	T: Count clock cycle
	m: Value set in the cycle setting register (PCSR)
	n: Value set in the duty setting register (PDUT)

- PWM operation timing example 1 (Trigger restarting disabled, PWM output: Normal polarity)

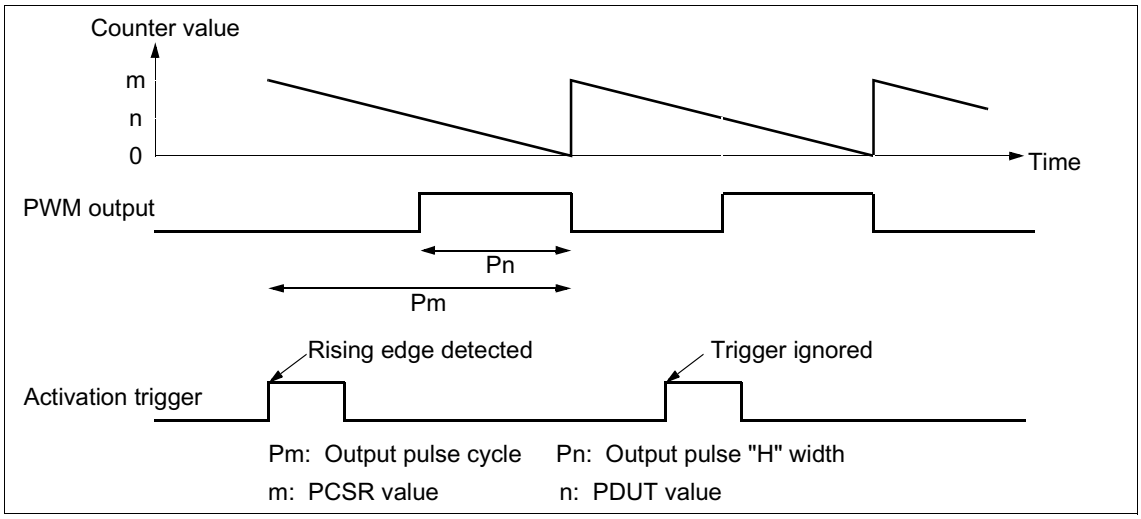


Figure 13.3.1a PWM Operation Timing Example 1 (Trigger Restarting Disabled, PWM Output: Normal Polarity)

- PWM operation timing example 2 (Trigger restarting enabled, PWM output: Normal polarity)

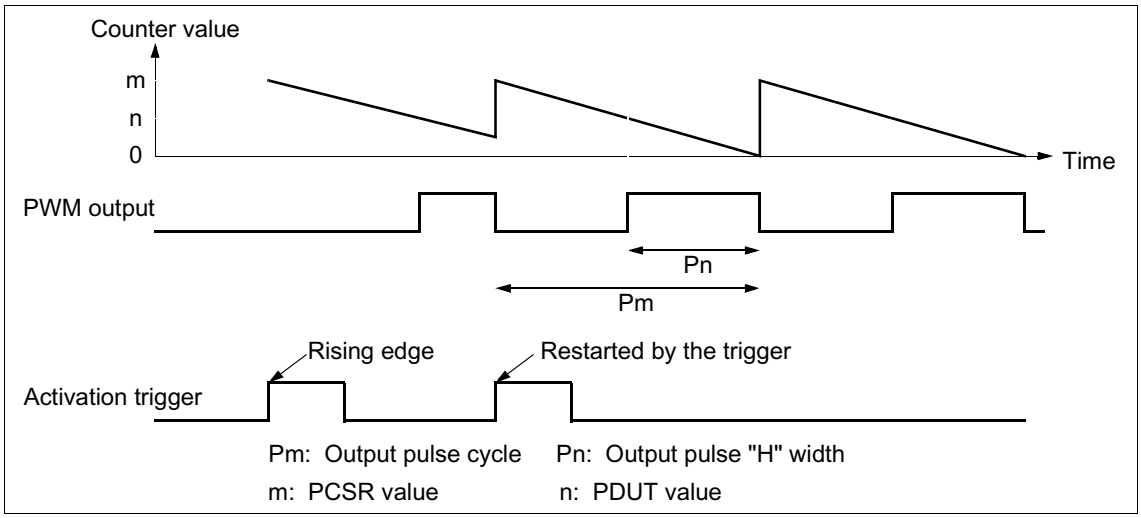


Figure 13.3.1b PWM Operation Timing Example 2 (Trigger Restarting Enabled, PWM Output: Normal Polarity)

13.3.2 One-Shot Operation

The one-shot operation mode outputs a single pulse

■ One-Shot Operation

The one-shot operation mode outputs a single pulse upon detection of an activation trigger.

The output pulse cycle can be controlled by changing the value in the cycle setting register (PCSR).

The output pulse width can also be controlled by changing the value in the duty setting register (PDUT).

When setting the output pulse cycle and width (duty ratio), be sure to write data to the PDUT register after writing data to the PCSR register.

One-shot operation also depends on whether restarting has been disabled or enabled.

Although the PWM output timing is the same as in the PWM operation mode, one-shot operation stops after output of a single pulse when the counter causes an underflow.

The equations for calculating the cycle and pulse width of the PWM output signal cycle are also the same as those for the PWM operation mode.

- One-shop operation timing example 1 (Trigger restarting disabled, PWM output: Normal polarity)

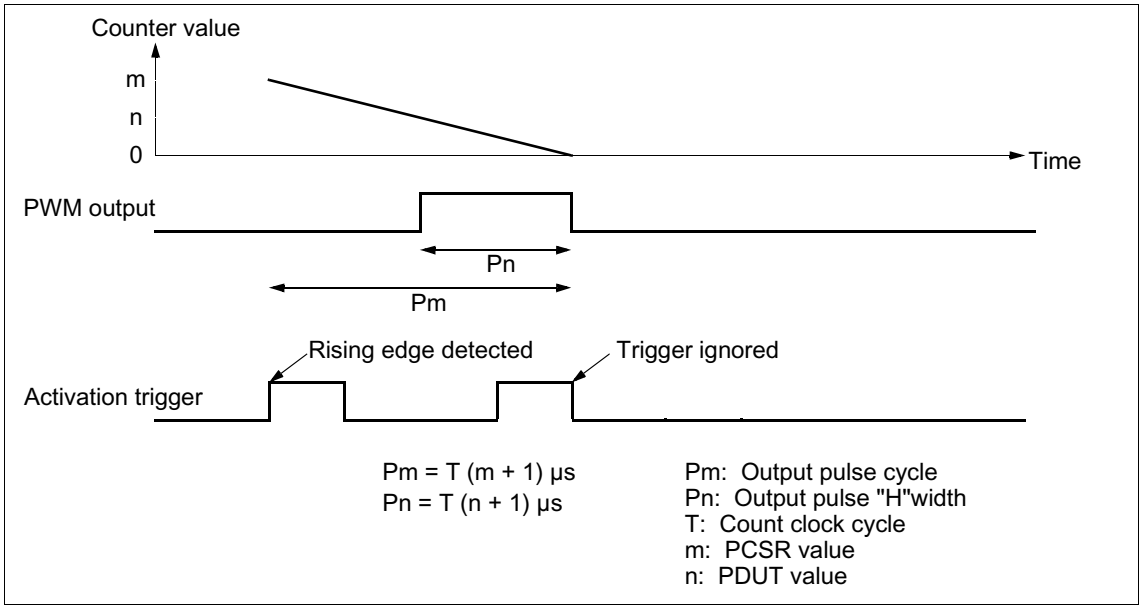


Figure 13.3.2a One-Shot Operation Timing Example 1 (Trigger Restarting Disabled, PWM Output: Normal Polarity)

- One-shot operation timing example 2 (Trigger restarting enabled, PWM output: Normal polarity)

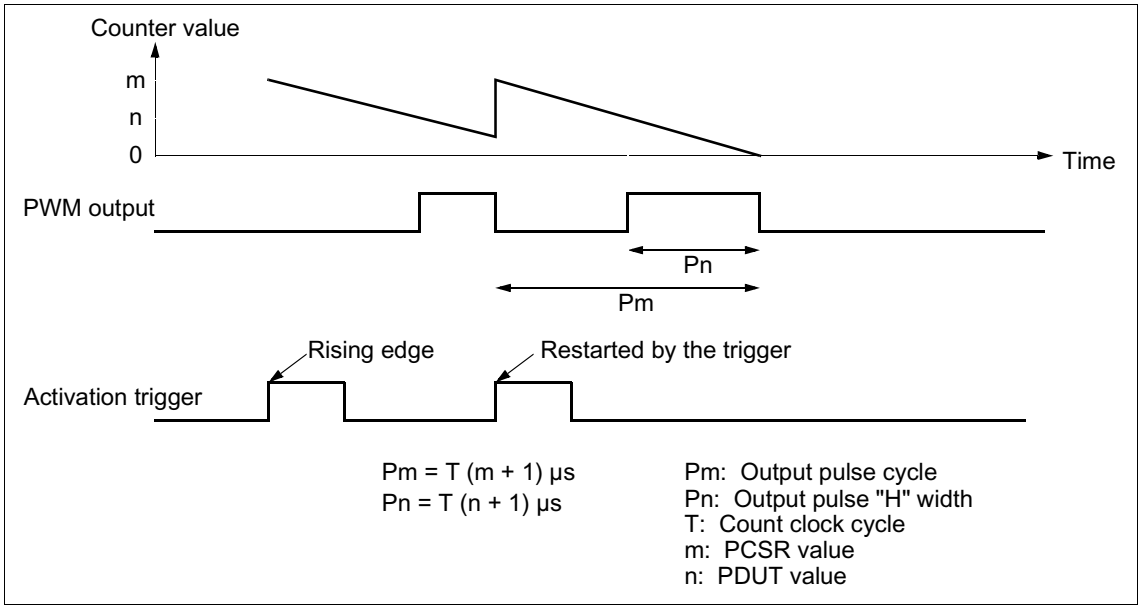


Figure 13.3.2b PWM Operation Timing Example 2 (Trigger Restarting Enabled, PWM Output: Normal Polarity)

13.3.3 Interrupts

Interrupt request signals can be generated by selecting the following interrupt sources:

- **Activation of the PWM timer (software trigger or trigger input)**
- **Occurrence of an underflow (cycle match)**
- **Occurrence of a duty match**
- **Occurrence of an underflow (cycle match) or duty match**

■ Interrupt Operation

The interrupt request signal to the CPU is generated by setting bits in the control status register (PCNL) as follows:

- Use bit 5 as the interrupt enable bit (IREN) to enable interrupts.
- Use bits 3 and 2 as the interrupt source select bits (IRS1 and IRS0) to select the desired interrupt source.
- When the above conditions are set, bit 4 as the interrupt request flag (IRQF) is set, generating the interrupt request signal to the CPU.

Figure 13.3.3 shows an interrupt timing diagram.

● Interrupt Source Timing

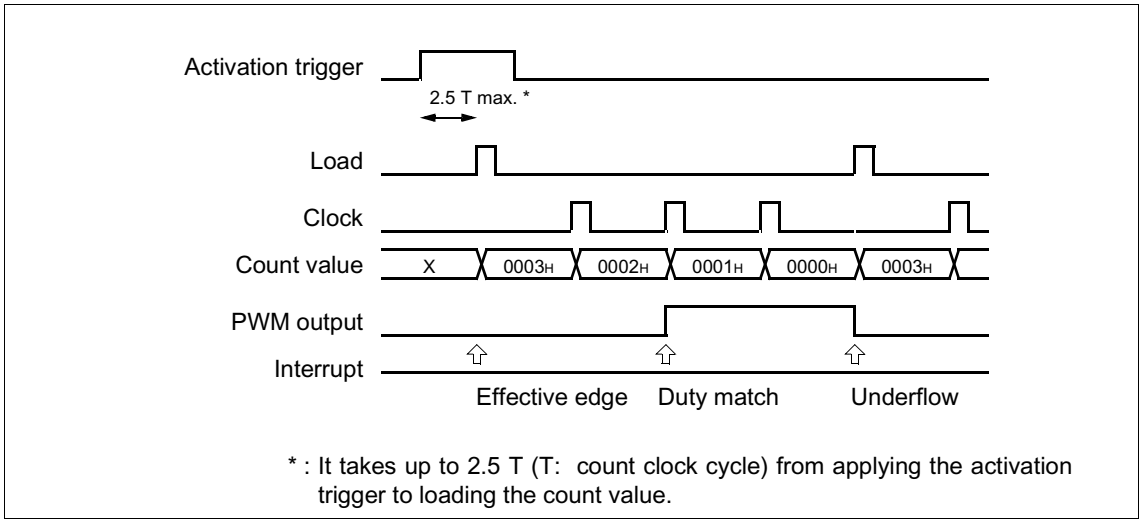


Figure 13.3.3 Interrupt Source Timing

13.3.4 All "L" and All "H" PWM Outputs

This section provides examples of producing all "L" and all "H" PWM outputs.

■ All "L" or All "H" PWM Output

Writing "1" to bit 9, or the PWM output mask select bit (PGMS), in the control status register (PCNH) masks the PWM output to the "L" level (normal polarity) or "H" level (inverted polarity) regardless of the mode, cycle, and duty settings.

If you want to output signals at all "H" level voltage with normal polarity or at all "L" voltage with inverted polarity, set the cycle and duty setting registers to the same value to produce the inverted output of the above mask value.

- Example of producing all "L" PWM output

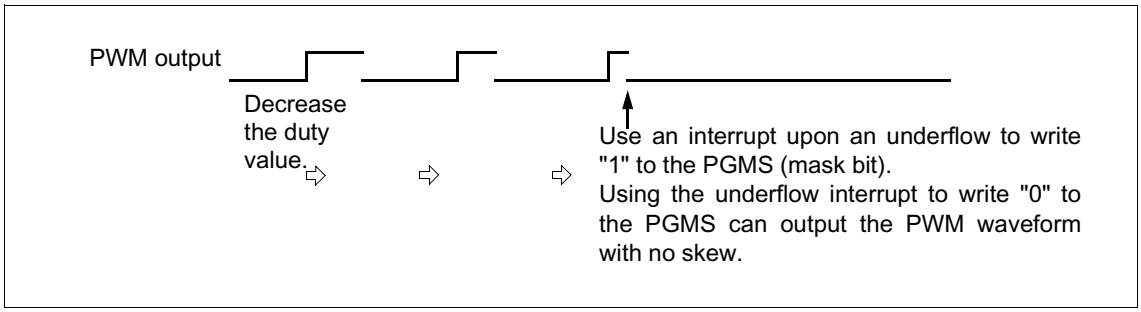


Figure 13.3.4a Example of Producing All "L" PWM Output

- Example of producing all "H" PWM output

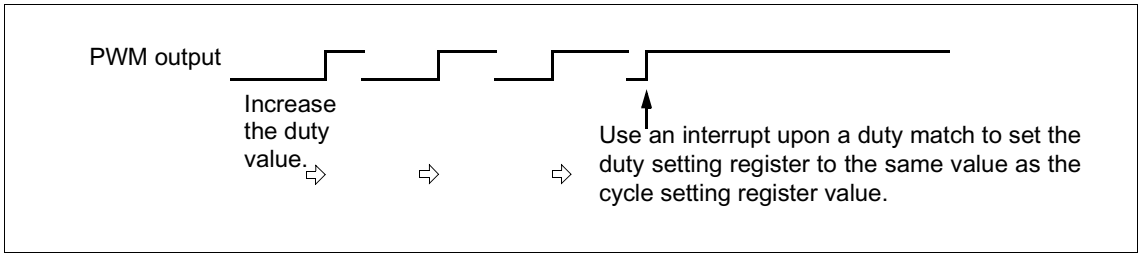


Figure 13.3.4b Example of Producing All "H" PWM Output

13.3.5 Activating Multiple PWM Timer Channels

General control registers 1 and 2 (GCN1x and GCN2x) can be used to activate multiple PWM timer channels.
Selecting the activation trigger using the GCN1x register allows multiple channels to be activated at the same time.
This section provides an example of activating PWM timer channels by means of software using the GCN2x register and an example of activating them using the GCN1x register to select the trigger input from the 16-bit reload timer.

■ Activating Multiple PWM Timer Channels Using Software

● Setting procedure

(1) Set the cycle setting register (PCSR) to the cycle setting counter value.

(2) Set the duty setting register (PDUT) to the pulse width setting counter value.

Note: Be sure to write data to the PDUT register after writing data to the PCSR register.

(3) Set the GCN1 register to determine the trigger input source for each channel you want to activate.
Since this example uses the GCN2 register, leave the initial settings intact.

To activate ch 0: EN0
To activate ch 1: EN1
To activate ch 2: EN2
To activate ch 3: EN3

(4) Set the control status registers (PCNL, PCNH) corresponding to the channels to be activated.
In this example, make the following settings:

Bit			Function
No.	Abbreviation	Value	
15	CNTE	1	Enable timer operation.
14	STGR	0	Disable software triggering because GCN2 is used for activation.
13	MDSE	0	PWM operation
12	RTRG	0	Disable restarting.
11	CKS1	0	Count clock: ϕ
10	CKS0	0	
9	PGMS	0	No output mask
8	–	0	Unused bit (Can be either 0 or 1.)
7	EGS1	0	Activate at rising edge
6	EGS0	1	
5	IREN	1	Enable interrupt requests.
4	IRQF	0	Clear interrupt source
3	IRS1	0	Generate interrupt request on occurrence of underflow.
2	IRS0	1	
0	OSEL	0	Normal polarity

(5) Write data to general control register 2 (GCN2x) to generate the activation trigger.
To activate PWM timer ch 0 and ch 1 at the same time using the above settings, write "1" to the EN0 and EN1 bits in the GCN20 register.

A rising edge will be generated and the PWM timer output pins OCPA0 and OCPA1 will output pulses (PWM0 and PWM1).

■ **Activation using a Trigger from the 16-Bit Reload Timer**

In step (3) in the above setting procedure, use general control register 1 (GCN10) to select the trigger input source.

Select the 16-bit reload timer as the trigger input source. In step (5), activate the 16-bit reload timer in place of general control register 2 (GCN20).

Set bits 12, 7, and 6 in the control status register (PCNH) as shown below for toggling the 16-bit reload timer output, and the PWM timer can be restarted at fixed intervals.

Bit 12 (RTRG): "1" ⇨ Enable restarting.

Bits 7, 6 (EGS1, EGS0): "11" ⇨ Activate at both rising and falling edges.

CHAPTER 14 A/D CONVERTER

This chapter provides an overview of the A/D converter, describes the register structure and functions, and describes the operation of the A/D converter.

For the electrical specification, please see section
34.7 "CONVERTER CHARACTERISTICS" on page 737.

14.1 OVERVIEW

The A/D Converter converts analog input voltage into digital values, and provides the following features.

- Conversion time : minimum 178 cycles (32MHz: 5.6 μs, 24 MHz: 7.4 μs, 16 MHz: 11.2 μs) per channel
- RC type successive approximation conversion with sample & hold circuit
- 10-bit resolution
- Program selection analog input from 16 channels
- Single conversion mode : conversion of one selected channel
- Scan conversion mode : continuous conversion of multiple channels, programmable for up to 16 channels

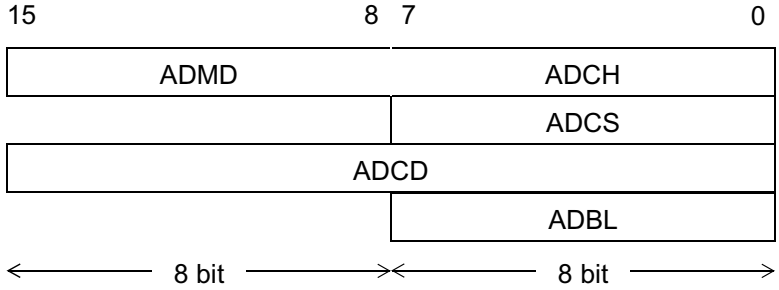
  Single conversion mode: Convert the specified channel once only.

  Continuous mode: Repeatedly convert the specified channels.

  Stop mode: Convert one channel then temporarily halt until the next activation. (Enables synchronization of the conversion start timing.)
- A/D conversion can be followed by an A/D conversion interrupt request to CPU. This interrupt, an option that is ideal for continuous processing can be used to start a DMA transfer of the results of A/D conversion to memory.
- Startup may be by software, external trigger (falling edge) or timer (rising edge) on devices where timer channel 4 is available.

14.2 REGISTER LIST

The A/D converter has the following registers.



	bit	7	6	5	4	3	2	1	0	
Address : 00009DH		ANS3	ANS2	ANS1	ANS0	ANE3	ANE2	ANE1	ANE0	Channel setting register ADCH
	bit	15	14	13	12	11	10	9	8	
Address : 00009CH		–	–	–	–	MOD1	MOD0	STS1	STS0	Mode register ADMD
	bit	7	6	5	4	3	2	1	0	
Address : 00009FH		BUSY	INT	INTE	PAUS	–	–	STRT	Reserved	Control status register ADCS
	bit	7	6	5	4	3	2	1	0	
Address : 0000A1H		D7	D6	D5	D4	D3	D2	D1	D0	Data register ADCD
	bit	15	14	13	12	11	10	9	8	
Address : 0000A0H		–	–	–	–	–	–	D9	D8	
	bit	15	14	13	12	11	10	9	8	
Address : 0000A3H		–	–	–	–	–	–	–	DBL	Disable register ADBL

14.3 BLOCK DIAGRAM

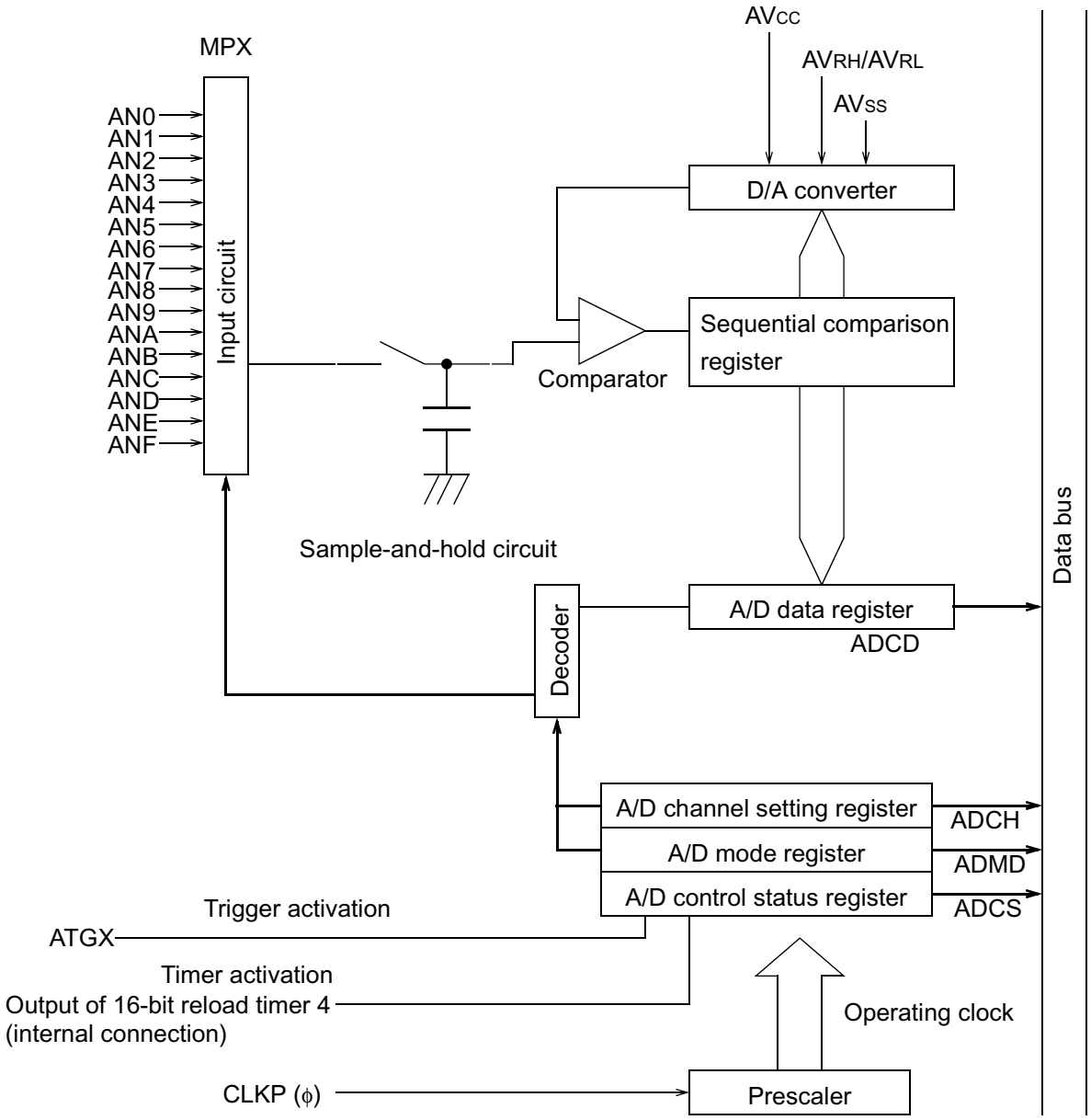


Figure 14.3 ADC Block Diagram

14.4 DETAILED REGISTER DESCRIPTIONS

14.4.1 ADCH (A/D Channel Setting Register)

This register specifies the channels for the A/D converter to convert.

	7	6	5	4	3	2	1	0	⇐Bit no.
ADCH Address : 00009DH	ANS3	ANS2	ANS1	ANS0	ANE3	ANE2	ANE1	ANE0	
Initial value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	
Read/write⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	

Do not update ADCH when the A/D converter is operating.

[bits 7, 6, 5, 4] ANS3, ANS2, ANS1, ANS0 (Analog start channel set)

These bits set the start channel for A/D conversion.

Activating the A/D converter starts A/D conversion from the channel specified by these bits.

ANS3	ANS2	ANS1	ANS0	Start Channel
0	0	0	0	AN0
0	0	0	1	AN1
0	0	1	0	AN2
0	0	1	1	AN3
0	1	0	0	AN4
0	1	0	1	AN5
0	1	1	0	AN6
0	1	1	1	AN7
1	0	0	0	AN8
1	0	0	1	AN9
1	0	1	0	ANA
1	0	1	1	ANB
1	1	0	0	ANC
1	1	0	1	AND
1	1	1	0	ANE
1	1	1	1	ANF

Reading these bits during A/D conversion reads the channel number currently being converted.

Reading these bits while A/D conversion is halted reads the most recently converted channel.

Reset by writing "0" or by a reset.

[bits 3, 2, 1, 0] ANE3, ANE2, ANE1, ANE0 (Analog end channel set)

These bits set the end channel for A/D conversion.

ANE3	ANE2	ANE1	ANE0	End channel
0	0	0	0	AN0
0	0	0	1	AN1
0	0	1	0	AN2
0	0	1	1	AN3
0	1	0	0	AN4
0	1	0	1	AN5
0	1	1	0	AN6
0	1	1	1	AN7
1	0	0	0	AN8
1	0	0	1	AN9
1	0	1	0	ANA
1	0	1	1	ANB
1	1	0	0	ANC
1	1	0	1	AND
1	1	1	0	ANE
1	1	1	1	ANF

Setting the same channel as in ANS3 to ANS0 specifies conversion for that channel only. (Single conversion)

In continuous or stop mode, conversion is performed up to the channel specified by these bits. Conversion then starts again from the start channel specified by ANS3 to ANS0.

If ANS > ANE, conversion starts with the channel specified by ANS, continues up to channel F, starts again from channel 0, and ends with the channel specified by ANE.

Reset by writing "0" or by a reset.

Example:Channel setting ANS = 0xE, ANE = 0x3, single conversion mode

Operation:Conversion channel $E \rightarrow F \rightarrow 0 \rightarrow 1 \rightarrow 2 \rightarrow 3$ end

14.4.2 ADMD (A/D Mode Register)

This register sets the operation mode and activation source for the A/D converter.

	15	14	13	12	11	10	9	8	⇐Bit no.
ADMD Address : 00009CH	–	–	–	Reserved	MOD1	MOD0	STS1	STS0	
Initial value⇒	(–)	(–)	(–)	(X)	(0)	(0)	(0)	(0)	
Read/write⇒	(–)	(–)	(–)	(W)	(R/W)	(R/W)	(R/W)	(R/W)	

Do not update ADMD when the A/D converter is operating.

[bit 12] A reserved bit. Always set to "0" when writing. The read value is indeterminate.

[bits 11, 10] MOD1, MOD0 (A/D converter mode set)

These bits set the operation mode.

MOD1	MOD0	Operating mode
0	0	Single mode; all restarts conversion during operation enabled
0	1	Single mode; restarts conversion during operation disabled
1	0	Continuous mode; restarts conversion during operation disabled
1	1	Stop mode; restarts conversion during operation disabled

Single mode: Continuous A/D conversion from selected channel(s) ANS3 to ANS0 to selected channel(s) ANE3 to ANE0 with a pause after every conversion cycle.

Continuous mode: Repeated A/D conversion cycles from selected channels ANS3 to ANS0 to selected channels ANE3 to ANE0.

Stop mode: A/D conversion for each channel from selected channels ANS3 to ANS0 to selected channels ANE3 to ANE0, followed by a pause. Restart is determined by the occurrence of a start source.

- When A/D conversion is started in continuous mode or stop mode, conversion operation continued until stopped by the BUSY bit.
- Conversion is stopped by writing '0' to the BUSY bit.
- All restarts are disabled for any of the timer, external trigger and software start sources in single, continuous and stop modes.

Restarts conversion : Activating the A/D converter and then applying another activation while the A/D is still operating is called restarts conversion.

[bits 9, 8] STS1, STS0 (Start source select)

Reset by writing "0" or by a reset.

These bits select the A/D activation source.

STS1	STS0	Function
0	0	Software activation
0	1	External trigger pin activation and software activation
1	0	Timer activation and software activation
1	1	External trigger pin activation, timer activation, and software activation

In multiple-activation modes, the first activation to occur starts A/D conversion.

The activation source changes immediately on writing to the register. Therefore, care is required when switching activation modes during A/D operation.

The A/D converter detects falling edges on the external trigger pin and rising edges on the timer.

Selecting the timer selects the output of timer 4.

- When using the A/D activation source bits (STS1 and STS0) in the ADMD register to select the external trigger or internal timer to activate the A/D converter, also set the external trigger or internal timer input value to inactive.
- If the input is active, an activation will occur when you write to the STS1 and STS0 bits.
- When setting the STS1 and STS0 bits, ensure that the ATGX input = "1" and the internal timer output (timer 4) = "0".

14.4.3 ADCS (A/D Control Status Register)

This is the control status register of the A/D converter.

	7	6	5	4	3	2	1	0	⇐Bit no.
ADCS Address : 00009FH	BUSY	INT	INTE	PAUS	–	–	STRT	Reserved	
Initial value⇒	(0)	(0)	(0)	(0)	(–)	(–)	(0)	(0)	
Read/write⇒	(R/W)	(R/W)	(R/W)	(R/W)	(–)	(–)	(W)	(R/W)	

[bit 7] BUSY (Busy flag and stop)

On reading: A/D converter operation indication bit. Set on activation of A/D conversion and cleared on completion.

On writing: Writing "0" to this bit during A/D conversion forcibly terminates conversion. Use to forcibly terminate in continuous and stop modes. In normal operation, write "1" to this bit.

RMW instructions read the bit as "1".

Cleared on the completion of A/D conversion in single conversion mode.

In continuous and stop mode, the flag is not cleared until conversion is terminated by writing "0".

Initialized to "0" by a reset.

Do not specify forcible termination and software activation (BUSY="0" and STRT="1") at the same time.

[bit 6] INT (Interrupt)

Data indication bit. This bit is set when conversion data is stored in ADCD.

If bit 5 (INTE) is "1" when this bit is set, an interrupt request is generated or, if activation of DMA is enabled, DMA is activated. Writing "1" to the bit has no meaning.

Only clear this bit by writing "0" when A/D conversion is halted.

0	Conversion not complete.
1	Conversion complete and result stored in data register.

The bit is cleared by writing "0" or by the DMA interrupt clear signal.

Initialized to "0" by a reset.

[bit 5] INTE (Interrupt enable)

This bit enables or disables the conversion completion interrupt.

0	Disable interrupt.
1	Enable interrupt.

Always set this bit when using DMA.

DMA is activated by generation of an interrupt request.

Cleared by writing "0" or by a reset.

[bit 4] PAUS (A/D converter pause)

This bit is set when A/D conversion temporarily halts.

The A/D converter has only one register to store the conversion result. Therefore, the previous conversion result is lost if it is not transferred by DMA when performing continuous conversion.

To avoid this problem, the next conversion data is not stored in the data register until the previous value has been transferred by DMA. A/D conversion halts during this time.

A/D conversion restarts when DMA transfer completes.

This bit is only meaningful when using DMA.

- See the description of the conversion data protection function in the section 14.5 "ADC OPERATION" on page 394.

0	No temporary halt occurred.
1	A temporary halt occurred.

Cleared by writing "0" or by a reset.

Only clear this bit by writing "0" when A/D conversion is halted.

[bit 1] STRT (Start)

Writing "1" to this bit starts A/D conversion.

Write "1" again to restart conversion.

Restarting does not function in stop mode.

The bit is always read as "0".

Do not specify forcible termination and software activation (BUSY="0" and STRT="1") at the same time.

0	No meaning
1	Activate the A/D.

Initialized to "0" by a reset.

[bit 0] Test bit. Always write "0" to this bit.

14.4.4 ADCD (A/D Data Register)

This register stores the conversion result of the A/D converter.

	7	6	5	4	3	2	1	0	⇐Bit no.
Address : 0000A1H	D7	D6	D5	D4	D3	D2	D1	D0	
Initial value⇒	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	
Read/write⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	

	15	14	13	12	11	10	9	8	⇐Bit no.
Address : 0000A0H	–	–	–	–	–	–	D9	D8	
Initial value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(X)	(X)	
Read/write⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	

These registers store the digital result of A/D conversion.

The register value is updated at the completion of each conversion. The register normally stores the result of the previous conversion.

The contents of the register are indeterminate after a reset.

Bits 15 to 10 of ADCD are read as "0".

The A/D converter has a conversion data protection function. See the section 14.5 "ADC OPERATION" on page 394 for further information.

Do not write to this register.

14.4.5 ADBL (A/D Disable Register)

This register allows the disabling of the clock for the ADC.

	15	14	13	12	11	10	9	8	⇐Bit no.
Address : 0000A3H	–	–	–	–	–	–	–	DBL	
Initial value⇒	(-)	(-)	(-)	(-)	(-)	(-)	(-)	(0)	
Read/write⇒	(-)	(-)	(-)	(-)	(-)	(-)	(-)	(R/W)	

Setting the DBL bit to"1" disables the clock for the ADC.

14.5 ADC OPERATION

The A/D converter operates using the successive approximation method with 10-bit resolution. As only one 16-bit register is provided to store conversion results, the conversion data register (ADCD) is updated each time conversion completes.

Therefore, as the A/D converter on its own is not suitable for performing continuous conversion, it is recommended that you use the DMA service function to transfer the conversion data to memory during conversion operation.

The following pages describe the operating modes.

14.5.1 Single Mode

In single conversion mode, the analog input signals selected by the ANS bits and ANE bits are converted in order until the completion of conversion on the end channel determined by the ANE bits. A/D conversion then ends. If the start channel and end channel are the same (ANS=ANE), only a single channel conversion is performed.

Examples: ANS=0000, ANE=0011
⇒ Start → AN0 → AN1 → AN2 → AN3 → End

ANS=0010, ANE=0010
⇒ Start → AN2 → End

14.5.2 Continuous Mode

In continuous mode the analog input signals selected by the ANS bits and ANE bits are converted in order until the completion of conversion on the end channel determined by the ANE bits, then the converter returns to the ANS channel for analog input and repeats the process continuously. When the start and end channels are the same (ANS = ANE), conversion is performed continuously for that channel.

Examples: ANS=0000, ANE=0011
⇒ Start → AN0 → AN1 → AN2 → AN3 → AN0 ... → repeat

ANS=0010, ANE=0010
⇒ Start → AN2 → AN2 → AN2 ... → repeat

In continuous mode, conversion is repeated until '0' is written to the BUSY bit. (Writing '0' to the BUSY bit forcibly stops the conversion operation.)

Note that forcibly terminating operation halts the current conversion during mid-conversion.

(If operation is forcibly terminated, the value in the conversion register is the result of the most recently completed conversion.)

14.5.3 Stop Mode

In stop mode the analog input signals selected by the ANS bits and ANE bits are converted in order, but conversion operation pauses for each channel. The pause is released by applying another start signal.

At the completion of conversion on the end channel determined by the ANE bits, the converter returns to the ANS channel for analog input signal and repeats the conversion process continuously.

When the start and end channels are the same (ANS = ANE), only a single channel conversion is performed.

Examples: ANS=0000, ANE=0011
⇒ Start → AN0 → stop → start → AN1 → stop → start → AN2
→ stop → start → AN3 → stop → start → AN0 ... → repeat

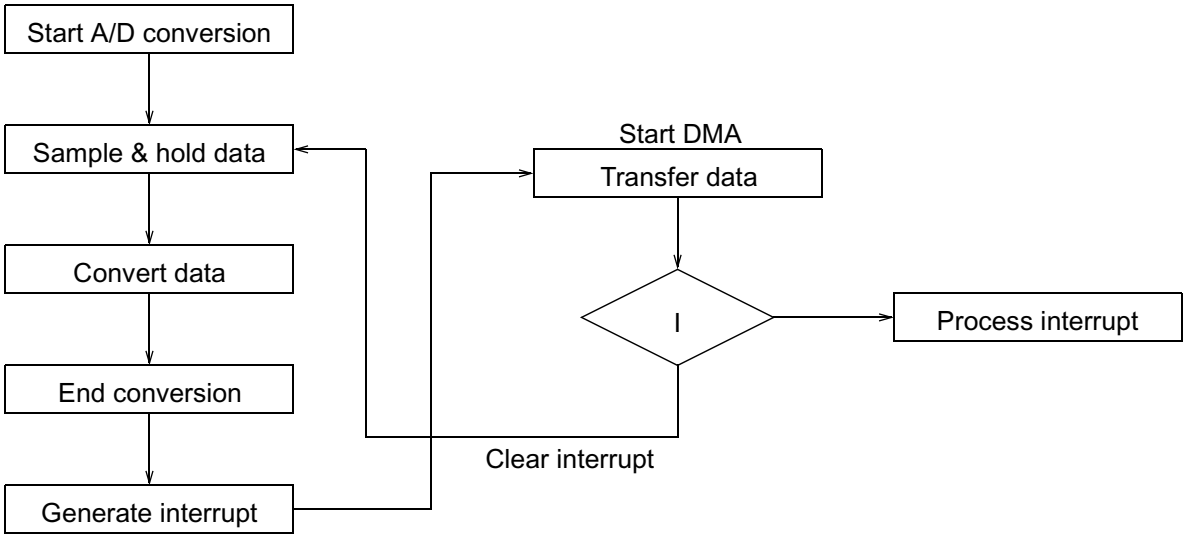
ANS=0010, ANE=0010
⇒ Start → AN2 → stop → start → AN2 → stop → start → AN2 ... → repeat

In stop mode the startup source is only the source determined by the STS1, STS0 bits.

This mode enables synchronization of the conversion start signal.

14.5.4 Conversion Operations Using DMA

The following chart illustrates the flow of operation from the start of A/D conversion through the transfer of conversion data, in continuous conversion mode.



I Indicates action determined by DMA setting.

14.5.5 Conversion Data Protection Function

A feature of the A/D converter is a conversion data protection function that allows the unit to protect the results of continuous conversion and multiple data sets.

Because there is only one conversion data register, continuous A/D conversion means that the last data in the register is lost each time conversion results are stored after a conversion cycle ends. To protect against loss of data, the A/D converter has a function that causes it to pause without storing the new data in the register even if the current conversion cycle ends until the previous data has been transferred to memory by DMA.

The pause is cancelled when the DMA transfer to memory is completed, and conversion resumes.

If the previous data is transferred to memory in time, A/D conversion continues without pause.

Caution: This function is related to the INT and INTE bits in the ADCS1 register.

The data protection function has been designed to operate only in interrupt-enabled (INTE=1) state.

If the interrupt is disabled (INTE=0), this function will not operate and continuous

A/D conversion processing will cause the previous set of conversion data to be lost when the next set is stored in the register.

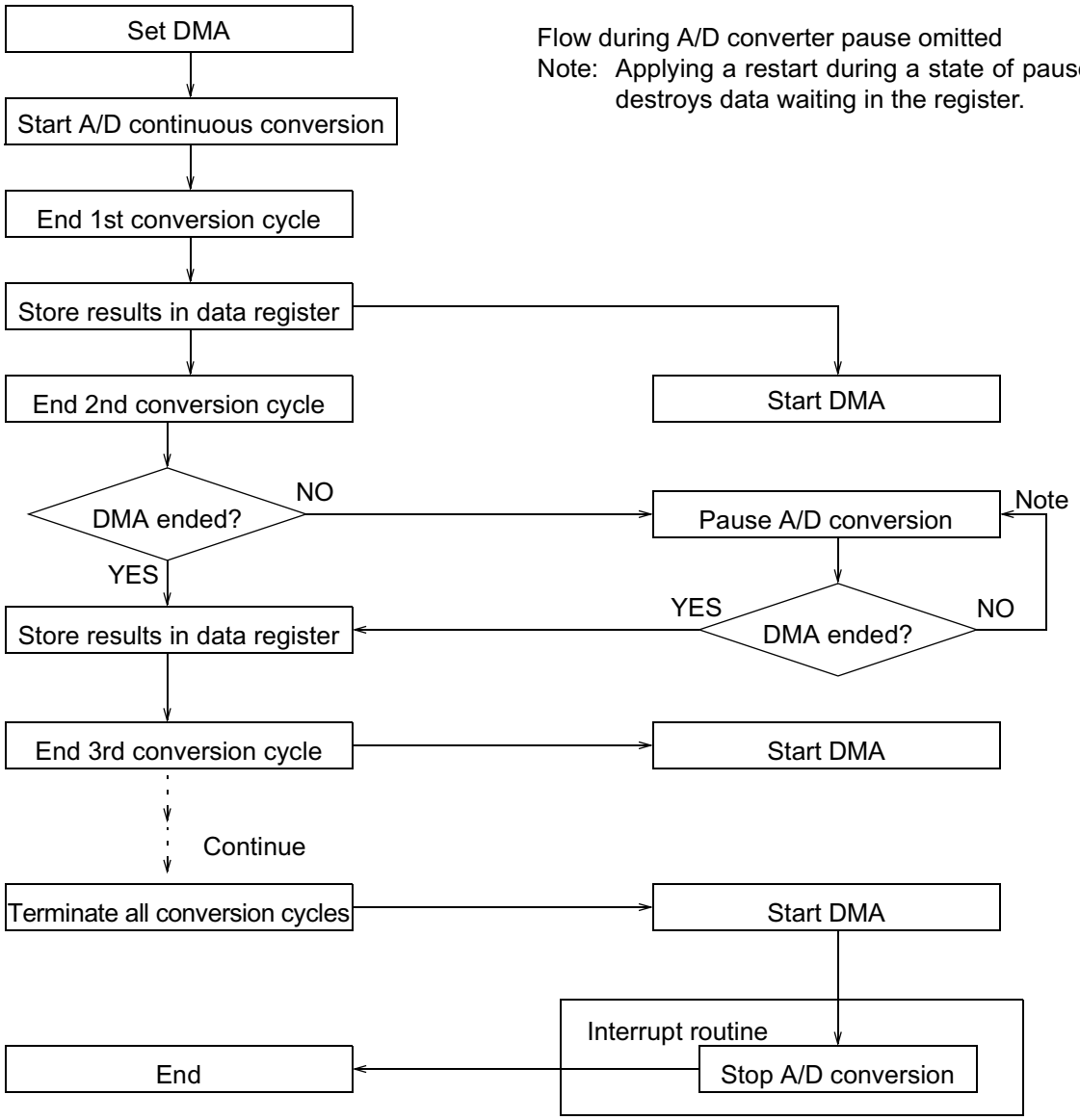
Also, in interrupt-enabled mode (INTE=1), the INT bit cannot be cleared without using DMA so that the data protection function may force A/D conversion to stay in a state of pause. In this case, an interrupt sequence is used to clear the INT bit and release the pause.

If interrupts are disabled while the A/D converter is in a state of pause during DMA transfer, it is possible that the A/D converter will resume operation and the contents of the conversion data register may be lost before transfer can be completed.

Also, applying a restart during a state of pause destroys data waiting in the register.

.

Flow of the data preservation function (when using DMA)



14.6 OTHER PRECAUTIONARY INFORMATION

Pulses input to the ATGX pin must be longer than the minimum input pulse width shown below.

Pulse width: 2 peripheral clock (CLKP) cycles

However, input pulses shorter than the above specification may be recognized as valid pulses in some cases.

The A/D converter's external trigger inputs does not have a filter function in the MB91360 series. If required, use a filter or similar circuit externally.

CHAPTER 15 16-BIT RELOAD TIMER

This chapter provides an overview of the 16-bit reload timer, describes the register structure/functions, and describes the operation of the 16-bit reload timer.

15.1 OVERVIEW OF THE 16-BIT RELOAD TIMER

Each 16-bit reload timer consists of a 16-bit down-counter, a 16-bit reload register, a prescaler for generating the internal count clock, and a control register.
The 16-bit reload timer can also activate DMA transfer using interrupts.
The MB91360 contains six 16-bit reload timer channels.

■ Functions

The input clock can be selected from three internal clocks
(the peripheral clock CLKP divided by 2/8/32).

■ 16-bit Reload Timer Register Configuration

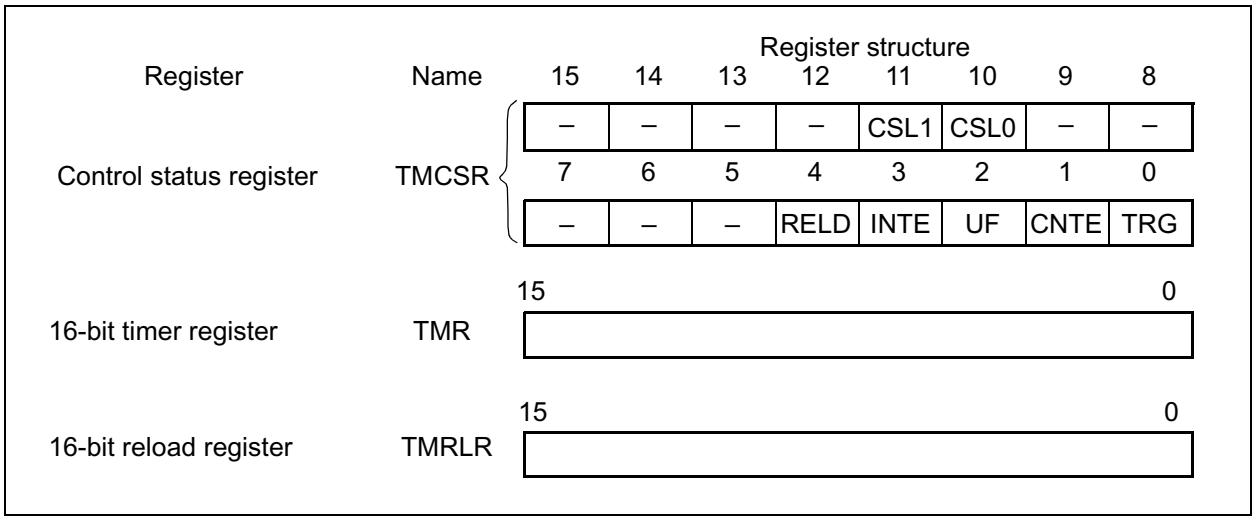


Figure 15.1a 16-bit Reload Timer Register Configuration

■ Block Diagram of the 16-Bit Reload Timer

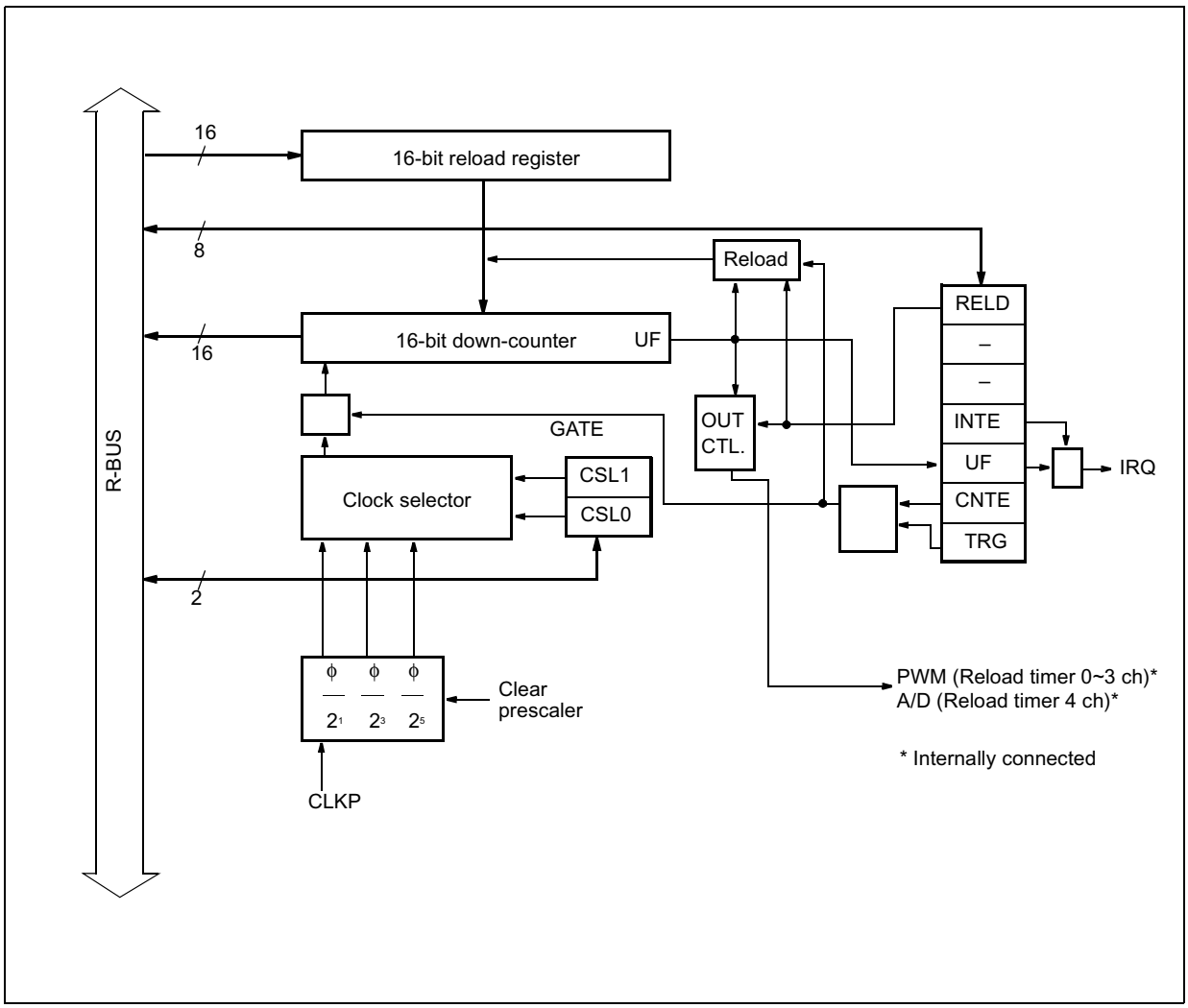


Figure 15.1b Block Diagram of the 16-bit Reload Timer

15.2 16-BIT RELOAD TIMER REGISTERS

This section describes the 16-bit reload timer registers listed below.

- **Control status register (TMCSR)**
- **16-bit timer register (TMR)**
- **16-bit reload register (TMRLR)**

15.2.1 Control Status Register (TMCSR)

Controls the operation mode and interrupts for the 16-bit reload timer.

Only change the value of bits other than UF, CNTE, and TRG when CNTE = "0".

The bits can be written simultaneously.

■ **TMCSR structure**

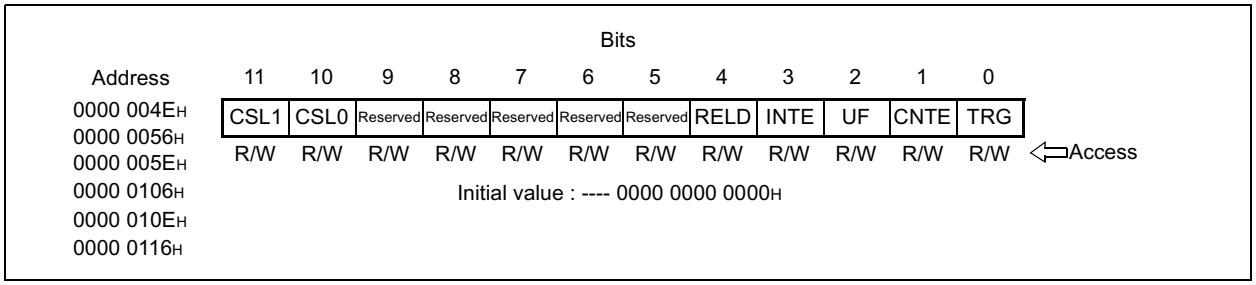


Figure 15.2.1 Structure of the Control Status Register

■ **Functions of the TMCSR bits**

[Bits 11, 10] CSL1, CSL0 (Count clock SeLect)

The count clock select bits.

Table 15.2.1 lists the clock source selections.

Table 15.2.1 CSL Bit Clock Source Settings

CSL1	CSL0	Clock source (ϕ : CLKP)
0	0	$\phi / 2^1$
0	1	$\phi / 2^3$
1	0	$\phi / 2^5$
1	1	Setting disabled

[Bits 9 to 5] Reserved

Always set to "00000".

[Bit 4] RELD

This bit enables reload operations.

When RELD is "1", the timer operates in reload mode. In this mode, the timer loads the reload register contents into the counter and continues counting whenever an underflow occurs (when the counter value changes from 0000_H to $FFFF_H$).

When RELD is "0", the count operation stops when an underflow occurs due to the counter value changing from 0000_H to $FFFF_H$.

[Bit 3] INTE

The interrupt request enable bit.

When INTE is "1", an interrupt request is generated when the UF bit changes to "1".

When INTE is "0", no interrupt requests are generated.

[Bit 2] UF

The timer interrupt request flag.

UF is set to "1" when an underflow occurs (when the counter value changes from 0000_H to $FFFF_H$).

Writing "0" clears the bit. Writing "1" has no meaning. Read as "1" by read-modify-write instructions.

[Bit 1] CNTE

The timer count enable bit.

Writing "1" sets the timer to wait for a trigger.

Writing "0" stops count operation.

[Bit 0] TRG

Software trigger bit.

Writing "1" to TRG applies a software trigger, causing the timer to load the reload register contents to the counter and start counting.

Writing "0" has no meaning. Reading always returns "0".

Applying a trigger using this register is only valid when CNTE = "1". Writing "1" has no effect if CNTE = "0".

15.2.2 16-bit Timer Register (TMR)

Reading this register reads the count value of the 16-bit timer.

The initial value is indeterminate.

Always read this register using 16-bit data transfer instructions.

■ **TMR structure**

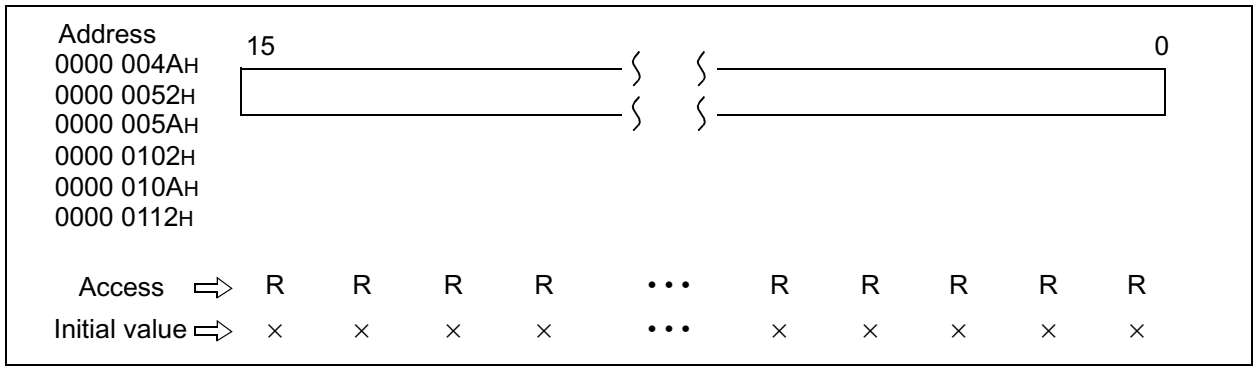


Figure 15.2.2 Structure of the 16-bit Timer Register

15.2.3 16-bit Reload Register (TMRLR)

The 16-bit reload register stores the initial count value.

The initial value is indeterminate.

Always write to this register using 16-bit data transfer instructions.

■ **TMRLR structure**

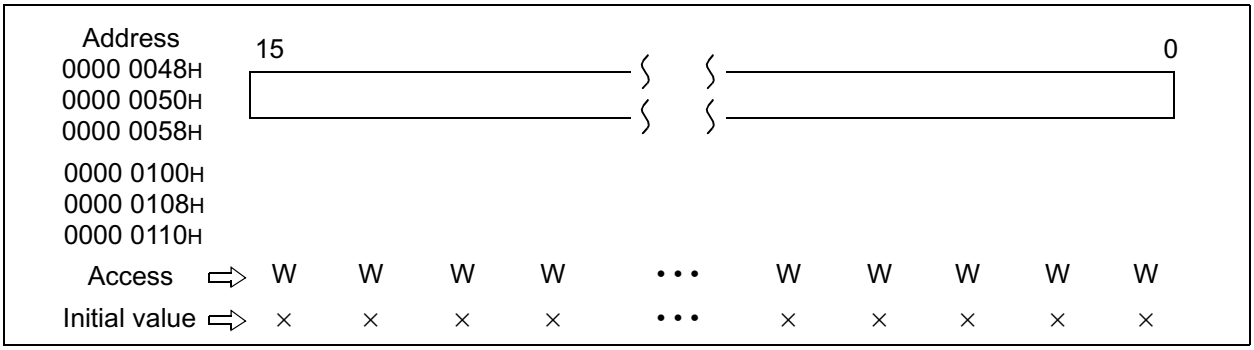


Figure 15.2.3 Structure of the 16-bit Reload Register

15.3 OPERATION OF THE 16-BIT RELOAD TIMER

This section describes the following operations of the 16-bit reload timer.

- **Internal clock operation**
- **Underflow operation**

15.3.1 Internal Clock Operation

The peripheral clock CLKP divided by 2, 8 or 32 can be selected as the clock source when operating the timer from an internal clock.

Writing "1" to both the CNTE and TRG bits in the control status register enables and starts counting simultaneously.

Using the TRG bit as a trigger input is always available when the timer is enabled (CNTE = "1"), regardless of the operation mode.

Figure 15.3.1 shows counter activation and counter operation.

A time ϕ (ϕ: CLKP machine cycle) is required from the counter start trigger being input until the reload register data is loaded into counter.

■ **Counter activation and operation timing**

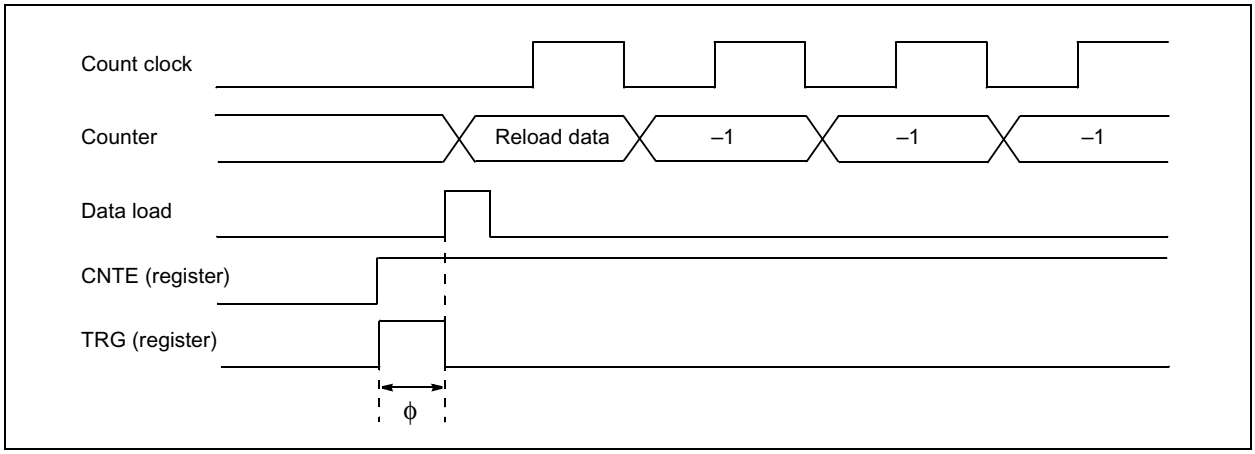


Figure 15.3.1 Counter Activation and Operation Timing

15.3.2 Underflow Operation

An underflow occurs when the counter value changes from 0000_H to $FFFF_H$. Therefore, an underflow occurs after "reload register setting + 1" counts.

If the RELD bit in the control register is "1" when the underflow occurs, the contents of the reload register are loaded into the counter and counting continues. When RELD is "0", counting stops with the counter at $FFFF_H$.

The UF bit in the control register is set when the underflow occurs. If the INTE bit is "1" at this time, an interrupt request is generated.

Figure 15.3.2 shows the operation when an underflow occurs.

■ **Underflow operation timing**

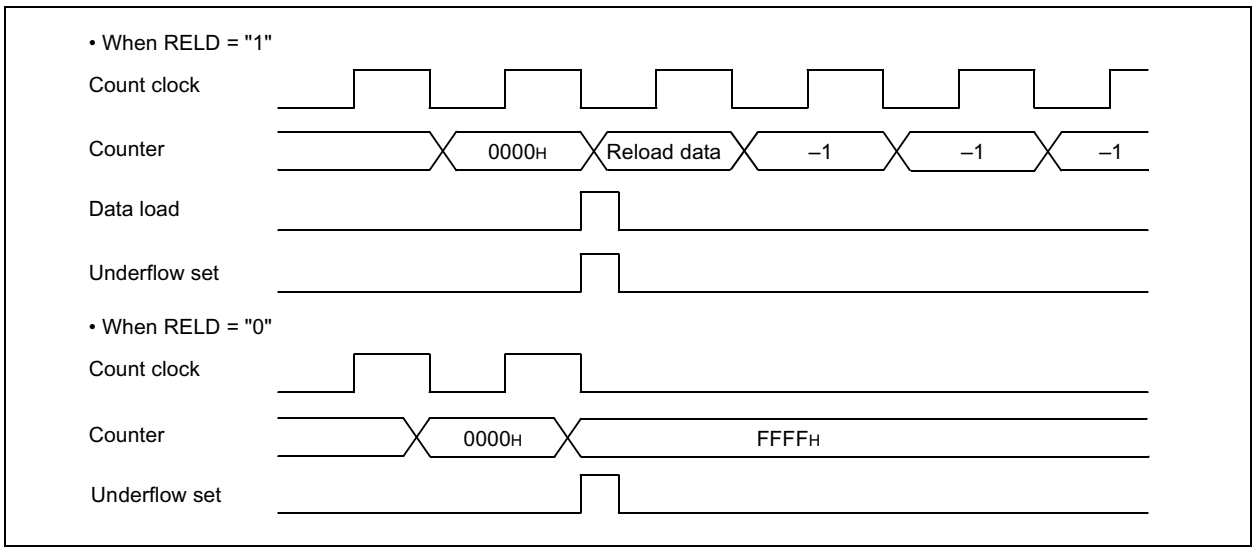


Figure 15.3.2 Underflow Operation Timing

15.3.3 Counter Operation States

The counter state is determined by the CNTE bit in the control register and the internal WAIT signal. The available states are CNTE = "0" and WAIT = "1" (STOP state: operation halted), CNTE = "1" and WAIT = "1" (WAIT state: waiting for a trigger), and CNTE = "1" and WAIT = "0" (RUN state: operating).

Figure 15.3.3 shows the transitions between each state.

■ **Counter state transitions**

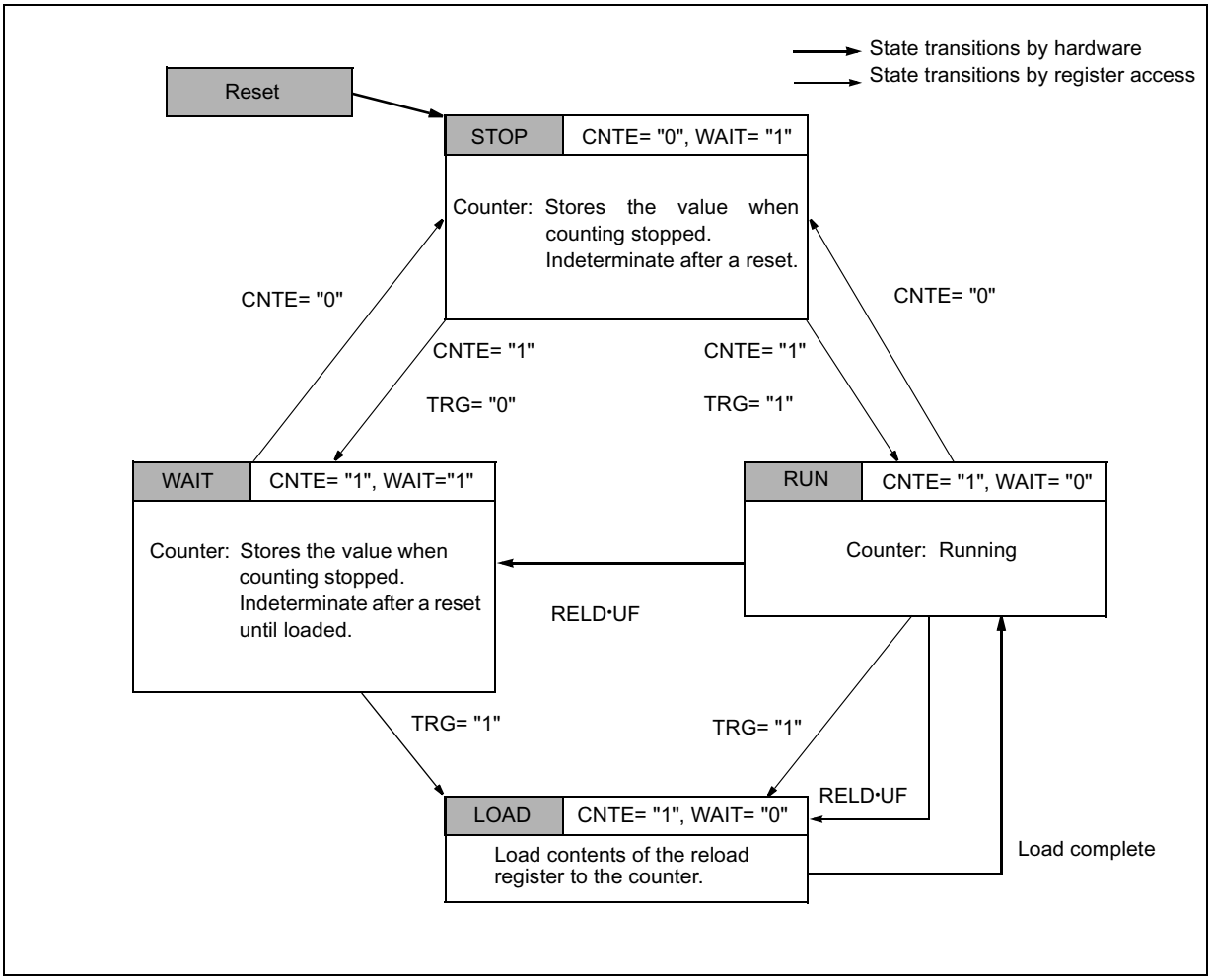


Figure 15.3.3 Counter State Transitions

15.3.4 Other Operations

1. The interrupt request signal of reload timer channel 0 and channel 1 can be used to activate DMA transfer.

   The DMA controller clears the interrupt flag of the reload timer when it acknowledges the transfer request.

2. The output of reload timer channel 4 is connected internally to the A/D converter. This can be used to activate A/D conversion periodically at the interval set in the reload register.

3. The outputs of reload timer channels 0 to 3 are connected internally to the PWM timers and can be used to trigger those timers.

CHAPTER 16 CAN CONTROLLER

This Chapter provides an overview of the CAN Interface, describes the register structure and functions, and describes the operation of the CAN Interface.

The CAN controller is a module built into a MB91360. The CAN (Controller Area Network) is the standard protocol for serial communication between automobile controllers and is widely used in industrial applications.

The CAN controller has the following features:

- Conforms to CAN Specification Version 2.0 Part A and B
  - Supports transmission/reception in standard frame and extended frame formats
- Supports transmitting of data frames by receiving remote frames
- 16 transmitting/receiving message buffers
  - 29-bit ID and 8-byte data
  - Multi-level message buffer configuration
- Supports full-bit comparison, full-bit mask and partial bit mask filtering.
  - Two acceptance mask registers in either standard frame format or extended frame formats
- Bit rate programmable from 10 Kbits/s to 1 Mbits/s (when input clock is at 16 MHz)

The following chapters only describe CAN 0. For the addresses of the registers of the other CAN channels see the IO-Map.

The addresses shown assume that the CS7 area is defined as described in the chapter about the internal Boot ROM.

16.1 LIST OF CONTROL REGISTERS

Table 16.1a List of Control Registers (1)

Address CAN0	Register	Abbreviation	Access	Initial Value
100000H	Message buffer valid register	BVALR0	R/W	00000000 00000000
100001H				
100002H	Transmit request register	TREQR0	R/W	00000000 00000000
100003H				
100004H	Transmit cancel register	TCANR0	W	00000000 00000000
100005H				
100006H	Transmit complete register	TCR0	R/W	00000000 00000000
100007H				
100008H	Receive complete register	RCR0	R/W	00000000 00000000
100009H				
10000AH	Remote request receiving register	RRTRR0	R/W	00000000 00000000
10000BH				
10000CH	Receive overrun register	ROVRR0	R/W	00000000 00000000
10000DH				
10000EH	Receive interrupt enable register	RIER0	R/W	00000000 00000000
10000FH				
100010H	Control status register	CSR0	R/W, R	00---000 0----0-1
100011H				
100012H	Last event indicator register	LEIR0	R/W	-------- 000-0000
100013H				
100014H	Receive/transmit error counter	RTEC0	R	00000000 00000000
100015H				
100016H	Bit timing register	BTR0	R/W	-1111111 11111111
100017H				
100018H	**IDE** register	IDER0	R/W	XXXXXXXX XXXXXXXX
100019H				

Table 1:

Table 16.1b List of Control Registers (2)

Address CAN0	Register	Abbreviation	Access	Initial Value
10001AH	Transmit **RTR** register	TRTRR0	R/W	00000000
10001BH				00000000
10001CH	Remote frame receive waiting register	RFWTR0	R/W	XXXXXXXX
10001DH				XXXXXXXX
10001EH	Transmit interrupt enable register	TIER0	R/W	00000000
10001FH				00000000
100020H	Acceptance mask select register	AMSR0	R/W	XXXXXXXX
100021H				XXXXXXXX
100022H				XXXXXXXX
100023H				XXXXXXXX
100024H	Acceptance mask register 0	AMR0	R/W	XXXXXXXX
100025H				XXXXXXXX
100026H				XXXXX---
100027H				XXXXXXXX
100028H	Acceptance mask register 1	AMR0	R/W	XXXXXXXX
100029H				XXXXXXXX
10002AH				XXXXX---
10002BH				XXXXXXXX

16.2 MESSAGE BUFFERS

Write access to the registers listed in tables16.2a to16.2f is not possible in byte mode. Use 16-bit access when writing to these addresses.

Table 16.2a List of Message Buffers (ID Registers) (1)

Address CAN0	Register	Abbreviation	Access	Initial Value
10002CH to 10004BH	General-purpose RAM	--	R/W	XXXXXXXX to XXXXXXXX
10004CH	**ID** register 0	IDR00	R/W	XXXXXXXX XXXXXXXX
10004DH				
10004EH				XXXXX--- XXXXXXXX
10004FH				
100050H	**ID** register 1	IDR10	R/W	XXXXXXXX XXXXXXXX
100051H				
100052H				XXXXX--- XXXXXXXX
100053H				
100054H	**ID** register 2	IDR20	R/W	XXXXXXXX XXXXXXXX
100055H				
100056H				XXXXX--- XXXXXXXX
100057H				
100058H	**ID** register 3	IDR30	R/W	XXXXXXXX XXXXXXXX
100059H				
10005AH				XXXXX--- XXXXXXXX
10005BH				
10005CH	**ID** register 4	IDR40	R/W	XXXXXXXX XXXXXXXX
10005DH				
10005EH				XXXXX--- XXXXXXXX
10005FH				
100060H	**ID** register 5	IDR50	R/W	XXXXXXXX XXXXXXXX
100061H				
100062H				XXXXX--- XXXXXXXX
100063H				

Table 16.2a List of Message Buffers (ID Registers) (1)

Address CAN0	Register	Abbreviation	Access	Initial Value
100064H	**ID** register 6	IDR60	R/W	XXXXXXXX XXXXXXXX
100065H				
100066H				XXXXX--- XXXXXXXX
100067H				

Table 16.2b List of Message Buffers (ID Registers) (2)

Address CAN0	Register	Abbreviation	Access	Initial Value
100068H	**ID** register 7	IDR70	R/W	XXXXXXXX XXXXXXXX
100069H				
10006AH				XXXXX--- XXXXXXXX
10006BH				
10006CH	**ID** register 8	IDR80	R/W	XXXXXXXX XXXXXXXX
10006DH				
10006EH				XXXXX--- XXXXXXXX
10006FH				
100070H	**ID** register 9	IDR90	R/W	XXXXXXXX XXXXXXXX
100071H				
100072H				XXXXX--- XXXXXXXX
100073H				
100074H	**ID** register 10	IDR100	R/W	XXXXXXXX XXXXXXXX
100075H				
100076H				XXXXX--- XXXXXXXX
100077H				
100078H	**ID** register 11	IDR110	R/W	XXXXXXXX XXXXXXXX
100079H				
10007AH				XXXXX--- XXXXXXXX
10007BH				

Table 16.2b List of Message Buffers (ID Registers) (2)

Address CAN0	Register	Abbreviation	Access	Initial Value
10007CH	**ID** register 12	IDR120	R/W	XXXXXXXX XXXXXXXX
10007DH				
10007EH				XXXXX--- XXXXXXXX
10007FH				
100080H	**ID** register 13	IDR130	R/W	XXXXXXXX XXXXXXXX
100081H				
100082H				XXXXX--- XXXXXXXX
100083H				
100084H	**ID** register 14	IDR140	R/W	XXXXXXXX XXXXXXXX
100085H				
100086H				XXXXX--- XXXXXXXX
100087H				

Table 16.2c List of Message Buffers (ID Registers) (3)

Address CAN0	Register	Abbreviation	Access	Initial Value
100088H	**ID** register 15	IDR150	R/W	XXXXXXXX XXXXXXXX
100089H				
10008AH				XXXXX--- XXXXXXXX
10008BH				

Table 16.2d List of Message Buffers (DLC Registers and Data Registers) (1)

Address CAN0	Register	Abbreviation	Access	Initial Value
10008CH	**DLC** register 0	DLCR00	R/W	----XXXX
10008DH				
10008EH	**DLC** register 1	DLCR10	R/W	----XXXX
10008FH				

Table 16.2d List of Message Buffers (DLC Registers and Data Registers) (1)

Address CAN0	Register	Abbreviation	Access	Initial Value
100090H 100091H	**DLC** register 2	DLCR20	R/W	----XXXX
100092H 100093H	**DLC** register 3	DLCR30	R/W	----XXXX
100094H 100095H	**DLC** register 4	DLCR40	R/W	----XXXX
100096H 100097H	**DLC** register 5	DLCR50	R/W	----XXXX
100098H 100099H	**DLC** register 6	DLCR60	R/W	----XXXX
10009AH 10009BH	**DLC** register 7	DLCR70	R/W	----XXXX
10009CH 10009DH	**DLC** register 8	DLCR80	R/W	----XXXX
10009EH 10009FH	**DLC** register 9	DLCR90	R/W	----XXXX
1000A0H 1000A1H	**DLC** register 10	DLCR100	R/W	----XXXX

Table 16.2e List of Message Buffers (DLC Registers and Data Registers) (2)

Address CAN0	Register	Abbreviation	Access	Initial Value
1000A2H 1000A3H	**DLC** register 11	DLCR110	R/W	----XXXX
1000A4H 1000A5H	**DLC** register 12	DLCR120	R/W	----XXXX
1000A6H 1000A7H	**DLC** register 13	DLCR130	R/W	----XXXX

Table 16.2e List of Message Buffers (DLC Registers and Data Registers) (2)

Address CAN0	Register	Abbreviation	Access	Initial Value
1000A8H 1000A9H	**DLC** register 14	DLCR140	R/W	----XXXX
1000AAH 1000ABH	**DLC** register 15	DLCR150	R/W	----XXXX
1000ACH to 1000B3H	Data register 0 (8 bytes)	DTR00	R/W	XXXXXXXX to XXXXXXXX
1000B4H to 1000BBH	Data register 1 (8 bytes)	DTR10	R/W	XXXXXXXX to XXXXXXXX
1000BCH to 1000C3H	Data register 2 (8 bytes)	DTR20	R/W	XXXXXXXX to XXXXXXXX
1000C4H to 1000CBH	Data register 3 (8 bytes)	DTR30	R/W	XXXXXXXX to XXXXXXXX
1000CCH to 1000D3H	Data register 4 (8 bytes)	DTR40	R/W	XXXXXXXX to XXXXXXXX
1000D4H to 1000DBH	Data register 5 (8 bytes)	DTR50	R/W	XXXXXXXX to XXXXXXXX
1000DCH to 1000E3H	Data register 6 (8 bytes)	DTR60	R/W	XXXXXXXX to XXXXXXXX
1000E4H to 1000EBH	Data register 7 (8 bytes)	DTR70	R/W	XXXXXXXX to XXXXXXXX
1000ECH to 1000F3H	Data register 8 (8 bytes)	DTR80	R/W	XXXXXXXX to XXXXXXXX
1000F4H to 1000FBH	Data register 9 (8 bytes)	DTR90	R/W	XXXXXXXX to XXXXXXXX

Table 16.2f List of Message Buffers (DLC Registers and Data Registers) (3)

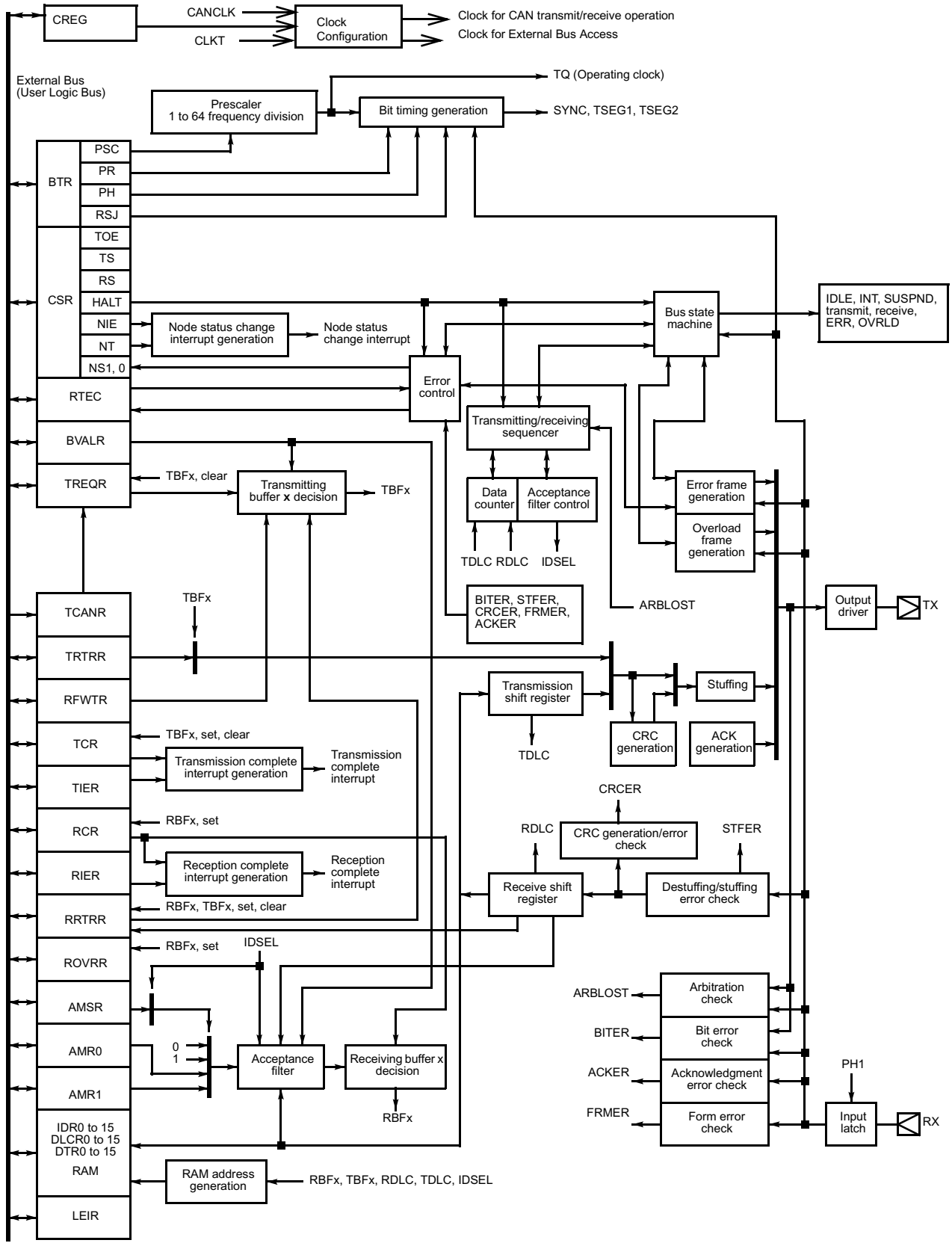
Address CAN0	Register	Abbreviation	Access	Initial Value
1000FCH to 100103H	Data register 10 (8 bytes)	DTR100	R/W	XXXXXXXX to XXXXXXXX
100104H to 10010BH	Data register 11 (8 bytes)	DTR110	R/W	XXXXXXXX to XXXXXXXX
10010CH to 100113H	Data register 12 (8 bytes)	DTR120	R/W	XXXXXXXX to XXXXXXXX
100114H to 10011BH	Data register 13 (8 bytes)	DTR130	R/W	XXXXXXXX to XXXXXXXX
10011CH to 100123H	Data register 14 (8 bytes)	DTR140	R/W	XXXXXXXX to XXXXXXXX
100124H to 10012BH	Data register 15 (8 bytes)	DTR150	R/W	XXXXXXXX to XXXXXXXX

Table 16.2g Configuration Register (CREG)

Address CAN0	Register	Abbreviation	Access	Initial Value
10012CH 10012DH	Configuration register	CREG0	R/W	00000000 00000110

16.3 BLOCK DIAGRAM

Figure 16.3 Block Diagram



16.4 OVERALL CONTROL REGISTER

16.4.1 CSR: Control Status Register

This register is prohibited from executing any bit manipulation instructions (Read-Modify-Write instructions).

	15	14	13	12	11	10	9	8
Address: 100010H (CAN0)	TS	RS	—	—	—	NT	NS1	NS0
Read/write:	(R)	(R)	(—)	(—)	(—)	(R/W)	(R)	(R)
Initial value:	(0)	(0)	(—)	(—)	(—)	(0)	(0)	(0)

	7	6	5	4	3	2	1	0
Address: 100011H (CAN0)	TOE	—	—	—	—	NIE	—	HALT
Read/write:	(R/W)	(—)	(—)	(—)	(—)	(R/W)	(—)	(R/W)
Initial value:	(0)	(—)	(—)	(—)	(—)	(0)	(—)	(1)

[Bit 15] **TS**: Transmit status bit

This bit indicates whether a message is being transmitted.

0: Message not transmitted

1: Message transmitted

This bit is **0** even while error and overload frames are transmitted.

[Bit 14] **RS**: Receive status bit

This bit indicates whether a message is being received.

0: Message not received

1: Message received

While a message is on the bus, this bit becomes **1**. Therefore, this bit is also **1** while a message is being transmitted. This bit does not necessarily indicates whether a receiving message passes through the acceptance filter.

As a result, when this bit is **0**, it implies that the bus operation is stopped (**HALT** = **0**); the bus is in the intermission/bus idle or a error/overload frame is on the bus.

[Bit 10] **NT**: Node status transition flag

If the node status is changed to increment, or from Bus Off to Error Active, this bit is set to **1**.

In other words, the **NT** bit is set to **1** if the node status is changed from Error Active (**00**) to Warning (**01**), from Warning (**01**) to Error Passive (**10**), from Error Passive (**10**) to Bus Off (**11**), and from Bus Off (**11**) to Error Active (**00**). Numbers in parentheses indicate the values of **NS1** and **NS0** bits.

When the node status transition interrupt enable bit (**NIE**) is **1**, an interrupt is generated. Writing **0** sets the **NT** bit to **0**. Writing **1** to the **NT** bit is ignored. **1** is read when a Read Modify Write instruction is read.

[Bits 9 to 8] **NS1** and **NS0**: Node status bits 1 and 0

These bits indicate the current node status.

Correspondence between NS1 and NS0 and Node Status

Table 16.4a Correspondence between NS1, NS0 and Node Status

NS1	NS0	Node Status
0	0	Error active
0	1	Warning (active)
1	0	Error possible
1	1	Bus off

Note: Warning (error active) is included in the error active in CAN Specification 2.0B for the node status, however, indicates that the transmit error counter or receive error counter has exceeded 96. The node status change diagram is shown in figure 16.4a.

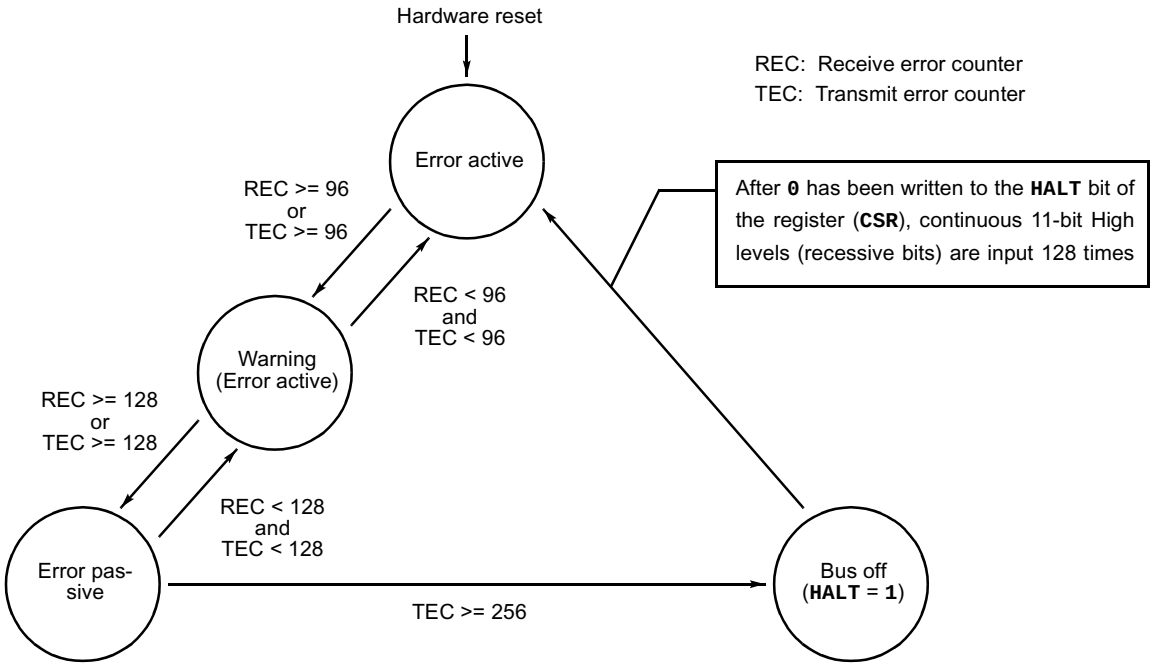


Figure 16.4a Node Status Transition Diagram

[Bit 7] **TOE**: Transmit output enable bit

Writing **1** to this bit switches from a general-purpose port pin to a transmit pin of the CAN controller.

0: General-purpose port pin

1: Transmit pin of CAN controller

[Bit 2] **NIE**: Node status transition interrupt enable bit

This bit enables or disables a node status transition interrupt (when **NT** = **1**).

0: Node status transition interrupt disabled

1: Node status transition interrupt enabled

[Bit 0] **HALT**: Bus operation stop bit

This bit sets or cancels bus operation stop, or displays its state.

(a) Conditions for setting bus operation stop (**HALT** = **1**)

• After hardware reset

• When node status changed to bus off

• By writing **1** to **HALT**

Notes:

1)The bus operation should be stopped by writing **1** to **HALT** before the MB9136x is changed in low-power consumption mode (stop mode, RTC mode, and hardware stand-by mode).

2)If transmission is in progress when **1** is written to **HALT**, the bus operation is stopped (**HALT** = **1**) after transmission is terminated. If reception is in progress when **1** is written to **HALT**, the bus operation is stopped immediately (**HALT** = **1**). If received messages are being stored in the message buffer (**x**), stop the bus operation (**HALT** = **1**) after storing the messages.

3)To check whether the bus operation has stopped, always read the **HALT** bit.

(b) Conditions for canceling bus operation stop (**HALT** = **0**)

• By writing **0** to **HALT**

Notes:

1)Canceling the bus operation stop ***after hardware reset*** or by ***writing 1 to HALT*** as above conditions is performed after **0** is written to **HALT** and continuous 11-bit High levels (recessive bits) have been input to the receive input pin (**RX**) (**HALT** = **0**).

2)Canceling the bus operation stop when ***the node status is changed to bus off*** as above conditions is performed after **0** is written to **HALT** and continuous 11-bit High levels (recessive bits) have been input 128 times to the receive input pin (**RX**) (**HALT** = **0**). Then, the values of both transmit and receive error counters reach 0 and the node status is changed to error active.

(c) State during bus operation stop (**HALT** = **1**)

• The bus does not perform any operation, such as transmission and reception.

• The transmit output pin (**TX**) outputs a High level (recessive bit).

• The values of other registers and error counters are not changed.

Note:The bit timing register (**BTR**) should be set during bus operation stop (**HALT** = **1**).

16.4.2 LEIR: Last Event Indicator Register

This register indicates the last event.

The **NTE**, **TCE**, and **RCE** bits are exclusive. When the bit of the last event is set to **1**, other bits are set to **0**s.

Address: 100013H (CAN0)

	7	6	5	4	3	2	1	0
	NTE	TCE	RCE	—	MBP3	MBP2	MBP1	MBP0
Read/write	(R/W)	(R/W)	(R/W)	(—)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(0)	(0)	(0)	(—)	(0)	(0)	(0)	(0)

[Bit 7] **NTE**: Node status transition event bit

When this bit is **1**, it indicates that node status transition is the last event.

This bit is set to **1** at the same time as the **NT** bit of the control status register (**CSR**). This bit is also set to **1**, irrespective of the setting of the node status transition interrupt enable bit (**NIE**) of the **CSR**.

Writing **0** to this bit sets the **NT** bit to **0**. Writing **1** to this bit is ignored.

1 is read when a Read Modify Write instruction is read.

[Bit 6] **TCE**: Transmit completion event bit

When this bit is **1**, it indicates that transmit completion is the last event.

This bit is set to **1** at the same time as any one of the bits of the transmit completion register (**TCR**). This bit is also set to **1**, irrespective of the settings of the bits of the transmit interrupt enable register (**TIER**).

Writing **0** sets this bit to **0**. Writing **1** to this bit is ignored.

1 is read when a Read Modify Write instruction is read.

When this bit is set to **1**, the **MBP3** to **MBP0** bits are used to indicate the numbers of the message buffers completing the transmit operation.

[Bit 5] **RCE**: Receive completion event bit

When this bit is **1**, it indicates that receive completion is the last event.

This bit is set to **1** at the same time as any one of the bits of the receive complete register (**RCR**). This bit is also set to **1** irrespective of the settings of the bits of the receive interrupt enable register (**RIER**).

Writing **0** sets this bit to **0**. Writing **1** to this bit is ignored.

1 is read when a Read Modify Write instruction is read.

When this bit is set to **1**, the **MBP3** to **MBP0** bits are used to indicate the numbers of the message buffers completing the receive operation.

[Bits 3 to 0] **MBP3** to **MBP0**: Message buffer pointer bits

When the **TCE** or **RCE** bit is set to **1**, these bits indicate the corresponding numbers of the message buffers (0 to 15). If the **NTE** bit is set to **1**, these bits have no meaning.

Writing **0** sets these bits to **0**s. Writing **1** to these bits is ignored.

1s are read when a Read Modify Write instruction is read.

If LEIR is accessed within an CAN interrupt handler, the event causing the interrupt in not necessarily the same as indicated by LEIR. In the time from interrupt request to the LEIR access within the interrupt handler there may occur other CAN events.

16.5 RTEC: RECEIVE AND TRANSMIT ERROR COUNTERS

	15	14	13	12	11	10	9	8
Address: 100014H (CAN0)	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0
Read/write:	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)
Initial value:	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

	7	6	5	4	3	2	1	0
Address: 100015H (CAN0)	REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0
Read/write:	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)
Initial value:	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

[Bits 15 to 8] **TEC7** to **TEC0**: Transmit error counter

These are transmit error counters.

REC7 to **REC0** values indicate **0** to **7** when the counter value is more than 256, and the subsequent increment is not counted for counter value. In this case, Error Passive is indicated for the node status (**NS1** and **NS0** of control status register **CSR** = **11**).

[Bits 7 to 0] **REC7** to **REC0**: Receive error counter

These are receive error counters.

REC7 to **REC0** values indicate **0** to **7** when the counter value is more than 256, and the subsequent increment is not counted for counter value. In this case, Bus Off is indicated for the node status (**NS1** and **NS0** of control status register **CSR** = **10**).

16.6 BTR: BIT TIMING REGISTER

This register sets the prescaler and bit timing.

	15	14	13	12	11	10	9	8
Address: 100016H (CAN0)	—	TS2.2	TS2.1	TS2.0	TS1.3	TS1.2	TS1.1	TS1.0
Read/write:	(—)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(—)	(1)	(1)	(1)	(1)	(1)	(1)	(1)

	7	6	5	4	3	2	1	0
Address: 100017H (CAN0)	RSJ1	RSJ0	PSC5	PSC4	PSC3	PSC2	PSC1	PSC0
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)

Note: This register should be set during bus operation stop (**HALT** = **1**).

[Bits 14 to 12] **TS2.2** to **TS2.0**: Time segment 2 setting bits 2 to 0

These bits cause the time quanta (**TQ**) to undergo [(**TS2.2** to **TS2.0**) +1] frequency division to determine time segment 2 (**TSEG2**). The time segment 2 is equal to the phase buffer segment 2 (**PHASE_SEG2**) in the CAN specification.

[Bits 11 to 8] **TS1.3** to **TS1.0**: Time segment 1 setting bits 3 to 0

These bits cause the time quanta (**TQ**) to undergo [(**TS1.3** to **TS1.0**) +1] frequency division to determine time segment 1 (**TSEG1**). The time segment 1 is equal to the propagation segment (**PROP_SEG**) + phase buffer segment 1 (**PHASE_SEG1**) in the CAN specification.

[Bits 7 and 6] **RSJ1** and **RSJ0**: Resynchronization jump width setting bits 1 and 0

These bits cause the time quanta (**TQ**) to undergo [(**RSJ1** to **RSJ0**) +1] frequency division to determine the resynchronization jump width.

[Bits 5 to 0] **PSC5** to **PSC0**: Prescaler setting bits 5 to 0

These bits cause the input clock to undergo [(**PSC5** to **PSC0**) +1] frequency division to determine the time quanta (**TQ**) of the CAN controller.

The bit time segments in the CAN specification, CAN controller are shown in figures 16.6a and 16.6b respectively.

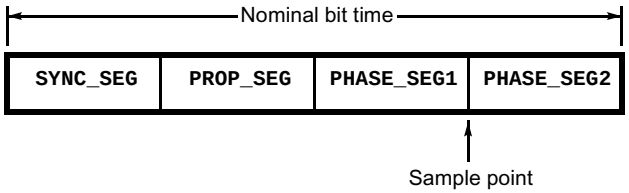


Figure 16.6a Bit Time Segment in CAN Specification

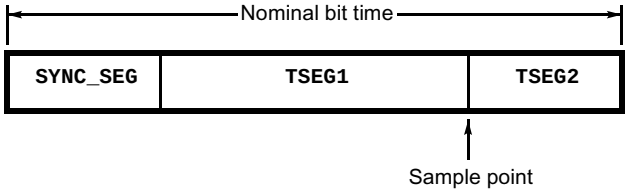


Figure 16.6b Bit Time Segment in CAN Controller

The relationship between **PSC** = **PSC5** to **PSC0**, **TSI** = **TS1**.3 to **TS1**.0, **TS2** = **TS2.2** to **TS1.0**, and **RSJ** = **RSJ1** and **RSJ0** when the input clock (**CLK**), time quanta (**TQ**), bit time (**BT**), synchronous segment (**SYNC_ SEG**), time segment 1 and 2 (**TSEG1** and **TSEG2**), and resynchronization jump width [(**RSJ1** and **RSJ0**) +1] frequency division is shown below.

TQ = (PSC + 1) × CLK
BT = SYNC_SEG + TSEG1 + TSEG2
= (1 + (TS1 + 1) + (TS2 + 1)) × TQ
= (3 + TS1 + TS2) TQ
RSJW = (RSJ + 1) × TQ

For correct operation of the CAN controller, the following conditions should be met:

- Devices with "G" suffix:

```
For 1 <= PSC <= 63:
  TSEG1 >= 2TQ
  TSEG1 >= RSJW
  TSEG2 >= 2TQ
  TSEG2 >= RSJW

For PSC = 0:
  TSEG1 >= 5TQ
  TSEG2 >= 2TQ
  TSEG2 >= RSJW
```

- Devices without "G" suffix:

```
For 1 <= PSC <= 63:
  TSEG1 >= RSJW
  TSEG2 >= RSJW + 2TQ

For PSC = 0:
  TSEG1 >= 5TQ
  TSEG2 >= RSJW + 2TQ
```

In order to meet the Bit Timing requirements defined in the CAN Specification, additional conditions have to be met, e.g. the propagation delay has to be considered.

16.7 MESSAGE BUFFER CONTROL REGISTERS

16.7.1 BVALR: Message Buffer Valid Register

	15	14	13	12	11	10	9	8
Address: 100000H (CAN0)	BVAL15	BVAL14	BVAL13	BVAL12	BVAL11	BVAL10	BVAL9	BVAL8
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

	7	6	5	4	3	2	1	0
Address: 100001H (CAN0)	BVAL7	BVAL6	BVAL5	BVAL4	BVAL3	BVAL2	BVAL1	BVAL0
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

This register sets the validity of the message buffer (**x**) or displays its state.

0: Message buffer (**x**) invalid

1: Message buffer (**x**) valid

If the message buffer (**x**) is set to invalid, it will not transmit or receive messages.

- Operation for suppressing transmission request:

  Don't use BVAL bit for suppressing transmission request, use TCAN bit instead of it.

- Operation for composing transmission message:

  For composing a transmission message, it is necessary to disable the message buffer by BVAL bit to change contents of ID and IDE registers. In this case, BVAL bit should reset (BVAL=0) after checking if TREQ bit is 0 or after completion of the previous message transmission (TC=1).

If the buffer is set to invalid during reception operating, it immediately becomes invalid (**BVALx** = **0**). If received messages are stored in a message buffer (**x**), the message buffer (**x**) is invalid after storing the messages.

Notes:

1) **x** indicates a message buffer number (**x** = 0 to 15).

2) When invaliding a message buffer (**x**) by writing **0** to a bit (**BVALx**), execution of a bit manipulation instruction is prohibited until the bit is set to **0**.

16.7.2 IDER: IDE register

	15	14	13	12	11	10	9	8
Address: 100018H (CAN0)	IDE15	IDE14	IDE13	IDE12	IDE11	IDE10	IDE9	IDE8
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)

	7	6	5	4	3	2	1	0
Address: 100019H (CAN0)	IDE7	IDE6	IDE5	IDE4	IDE3	IDE2	IDE1	IDE0
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)

This register sets the frame format used by the message buffer (**x**) during transmission/ reception.

0: The standard frame format (**ID11** bit) is used for the message buffer (**x**).

1: The extended frame format (**ID29** bit) is used for the message buffer (**x**).

Note: This register should be set when the message buffer (**x**) is invalid (**BVALx** of the message buffer valid register (**BVALR**) = **0**). Setting when the buffer is valid (**BVALx** = **1**) may cause unnecessary received messages to be stored.

16.7.3 TREQR: Transmission Request Register

	15	14	13	12	11	10	9	8
Address: 100002H (CAN0)	TREQ15	TREQ14	TREQ13	TREQ12	TREQ11	TREQ10	TREQ9	TREQ8
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

	7	6	5	4	3	2	1	0
Address: 100003H (CAN0)	TREQ7	TREQ6	TREQ5	TREQ4	TREQ3	TREQ2	TREQ1	TREQ0
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

This register sets a transmission request to the message buffer (**x**) or displays its state.

When **1** is written to **TREQx**, transmission to the message buffer (**x**) starts. If **RFWTx** of the remote frame receiving wait register (**RFWTR**)*1 is **0**, transmission starts immediately. However, if **RFWTx** = **1**, transmission starts after waiting until a remote frame is received (**RRTRx** of the remote request receiving register (**RRTRR**)*1 becomes **1**). Transmission starts*2 immediately even when **RFWTx** = **1**, if **RRTRx** is already **1** when **1** is written to **TREQx**.

*1: For **RFWTR** and **TRTRR**, see 16.7.5 "RFWTR: Remote Frame Receiving Wait Register" on page 429 and 16.7.4 "TRTRR: Transmission RTR Register" on page 429.

*2: For cancellation of transmission, see 16.7.6 "TCANR: Transmission Cancel Register" on page 430 and16.10 "TRANSMISSION" on page 444

Writing **0** to **TREQx** is ignored.

0 is read when a Read Modify Write instruction is read.

If clearing (to **0**) at completion of the transmit operation and setting by writing **1** are concurrent, clearing is preferred.

If **1** is written to more than one bit, transmission is performed, starting with the lower-numbered message buffer (**x**).

TREQx is **1** while transmission is pending, and becomes **0** when transmission is completed or canceled.

16.7.4 TRTRR: Transmission RTR Register

	15	14	13	12	11	10	9	8
Address: 10001AH (CAN0)	TRTR15	TRTR14	TRTR13	TRTR12	TRTR11	TRTR10	TRTR9	TRTR8
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

	7	6	5	4	3	2	1	0
Address: 10001BH (CAN0)	TRTR7	TRTR6	TRTR5	TRTR4	TRTR3	TRTR2	TRTR1	TRTR0
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

This register sets the **RTR** (Remote Transmission Request) bit during transmission by the message buffer (**x**).

0: Data frame transmitted

1: Remote frame transmitted

16.7.5 RFWTR: Remote Frame Receiving Wait Register

	15	14	13	12	11	10	9	8
Address: 10001CH (CAN0)	RFWT15	RFWT14	RFWT13	RFWT12	RFWT11	RFWT10	RFWT9	RFWT8
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)

	7	6	5	4	3	2	1	0
Address: 10001DCH (CAN0)	RFWT7	RFWT6	RFWT5	RFWT4	RFWT3	RFWT2	RFWT1	RFWT0
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)

This register sets the condition for starting transmission when a request for data frame transmission is set (**TREQx** of the transmission request register (**TREQR**) is **1** and **TRTRx** of the transmitting **RTR** register (**TRTRR**) is **0**).

0: Transmission starts immediately

1: Transmission starts after waiting until remote frame received (**RRTRx** of remote request receiving register (**RRTRR**) becomes **1**)

Notes:

1) Transmission starts immediately if **RRTRx** is already **1** when a request for transmission is set.

2) For remote frame transmission, do not set **RFWTx** to **1**.

16.7.6 TCANR: Transmission Cancel Register

	15	14	13	12	11	10	9	8
Address: 100004H (CAN0)	TCAN15	TCAN14	TCAN13	TCAN12	TCAN11	TCAN10	TCAN9	TCAN8
Read/write:	(W)	(W)	(W)	(W)	(W)	(W)	(W)	(W)
Initial value:	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

	7	6	5	4	3	2	1	0
Address: 100005H (CAN0)	TCAN7	TCAN6	TCAN5	TCAN4	TCAN3	TCAN2	TCAN1	TCAN0
Read/write:	(W)	(W)	(W)	(W)	(W)	(W)	(W)	(W)
Initial value:	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

When **1** is written to **TCANx**, this register cancels a pending request for transmission to the message buffer (**x**).

At completion of transmission, **TREQx** of the transmission request register (**TREQR**) becomes **0**. Writing **0** to **TCANx** is ignored.

This is a write-only register and its read value is always **0**.

16.7.7 TCR: Transmission Complete Register

	15	14	13	12	11	10	9	8
Address: 100006H (CAN0)	TC15	TC14	TC13	TC12	TC11	TC10	TC9	TC8
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

	7	6	5	4	3	2	1	0
Address: 100007H (CAN0)	TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

At completion of transmission by the message buffer (**x**), the corresponding **TCx** becomes **1**.

If **TIEx** of the transmission complete interrupt enable register (**TIER**) is **1**, an interrupt occurs.

Conditions for **TCx** = **0**

- Write **0** to **TCx**.

- Write **1** to **TREQx** of the transmission request register (**TREQR**).

After the completion of transmission, write **0** to **TCx** to set it to **0**. Writing **1** to **TCx** is ignored.

1 is read when a Read Modify Write instruction is read.

Note: If setting (to **1**) at completion of the transmit operation and clearing by writing **0** are concurrent, setting is preferred.

16.7.8 TIER: Transmission Interrupt Enable Register

	15	14	13	12	11	10	9	8
Address: 10001EH (CAN0)	TIE15	TIE14	TIE13	TIE12	TIE11	TIE10	TIE9	TIE8
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

	7	6	5	4	3	2	1	0
Address: 10001FH (CAN0)	TIE7	TIE6	TIE5	TIE4	TIE3	TIE2	TIE1	TIE0
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

This register enables or disables the transmission interrupt by the message buffer (**x**). The transmission interrupt is generated at transmission completion (when **TCx** of the transmission complete register (**TCR**) is **1**).

0: Transmission interrupt disabled

1: Transmission interrupt enabled

16.7.9 RCR: Reception Complete Register

	15	14	13	12	11	10	9	8
Address: 100008H (CAN0)	RC15	RC14	RC13	RC12	RC11	RC10	RC9	RC8
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

	7	6	5	4	3	2	1	0
Address: 100009H (CAN0)	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

At completion of storing received messages in the message buffer (**x**), **RCx** becomes **1**.

If **RIEx** of the reception complete interrupt enable register (**RIER**) is **1**, an interrupt occurs.

Conditions for **RCx** = **0**

- Write **0** to **RCx**.

After completion of storing received messages, write **0** to **RCx** to set it to **0**. Writing **1** to **RCx** is ignored.

1 is read when a Read Modify Write instruction is read.

Note: If setting (to **1**) at completion of the receive operation and clearing by writing **0** are concurrent, setting is preferred.

16.7.10 RRTRR: Remote Request Receiving Register

	15	14	13	12	11	10	9	8
Address: 10000AH (CAN0)	RRTR15	RRTR14	RRTR13	RRTR12	RRTR11	RRTR10	RRTR9	RRTR8
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

	7	6	5	4	3	2	1	0
Address: 10000BH (CAN0)	RRTR7	RRTR6	RRTR5	RRTR4	RRTR3	RRTR2	RRTR1	RRTR0
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

After a received remote frame is stored in the message buffer (**x**), **RRTRx** becomes **1** (at the same time as **RCx** setting to **1**).

Conditions for **RRTRx** = **0**

- Write **0** to **RRTRx**.
- After a received data frame is stored in the message buffer (**x**) (at the same time as **RCx** setting to **1**).
- Transmission by the message buffer (**x**) is completed (**TCx** of the transmission complete register (**TCR**) is **1**).

Writing **1** to **RRTRx** is ignored.

1 is read when a Read Modify Write instruction is read.

Note: If setting (to **1**) and clearing by writing **0** are concurrent, setting is preferred.

16.7.11 ROVRR: Receive Overrun Register

	15	14	13	12	11	10	9	8
Address: 10000CH (CAN0)	ROVR15	ROVR14	ROVR13	ROVR12	ROVR11	ROVR10	ROVR9	ROVR8
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

	7	6	5	4	3	2	1	0
Address: 10000DH (CAN0)	ROVR7	ROVR6	ROVR5	ROVR4	ROVR3	ROVR2	ROVR1	ROVR0
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

If **RCx** of the reception complete register (**RCR**) is **1** when completing storing of a received message in the message buffer (**x**), **ROVRx** becomes **1**, indicating that reception has overrun.

Writing **0** to **ROVRx** results in **ROVRx** = **0**. Writing **1** to **ROVRx** is ignored. After checking that reception has overrun, write **0** to **ROVRx** to set it to **0**.

1 is read when a Read Modify Write instruction is read.

Note: If setting (to **1**) and clearing by writing **0** are concurrent, setting is preferred.

16.7.12 RIER: Reception Interrupt Enable Register

	15	14	13	12	11	10	9	8
Address: 10000EH (CAN0)	RIE15	RIE14	RIE13	RIE12	RIE11	RIE10	RIE9	RIE8
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

	7	6	5	4	3	2	1	0
Address: 10000FH (CAN0)	RIE7	RIE6	RIE5	RIE4	RIE3	RIE2	RIE1	RIE0
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

This register enables or disables the reception interrupt by the message buffer (**x**).

The reception interrupt is generated at reception completion (when **RCx** of the reception completion register (**RCR**) is **1**).

0: Reception interrupt disabled

1: Reception interrupt enabled

16.7.13 AMSR: Acceptance Mask Select Register

BYTE1

	15	14	13	12	11	10	9	8
Address: 100020H (CAN0)	AMS7.1	AMS7.0	AMS6.1	AMS6.0	AMS5.1	AMS5.0	AMS4.1	AMS4.0
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)

BYTE0

	7	6	5	4	3	2	1	0
Address: 100021H (CAN0)	AMS3.1	AMS3.0	AMS2.1	AMS2.0	AMS1.1	AMS1.0	AMS0.1	AMS0.0
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)

BYTE3

	15	14	13	12	11	10	9	8
Address: 100022H (CAN0)	AMS15.1	AMS15.0	AMS14.1	AMS14.0	AMS13.1	AMS13.0	AMS12.1	AMS12.0
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)

BYTE2

	7	6	5	4	3	2	1	0
Address: 100023H (CAN0)	AMS11.1	AMS11.0	AMS10.1	AMS10.0	AMS9.1	AMS9.0	AMS8.1	AMS8.0
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)

This register selects a mask (acceptance mask) for comparison between the received message **ID** and the message buffer (**x**) **ID**.

Table 16.7.13 Selection of Acceptance Mask

AMSx.1	AMSx.0	Acceptance Mask
0	0	Full-bit comparision
0	1	Full-bit mask
1	0	Acceptance mask register 0 (**AMR0**)
1	1	Acceptance mask register 1 (**AMR1**)

Note: **AMSx.1** and **AMSx.0** should be set when the message buffer (**x**) is invalid (**BVALx** of the message buffer valid register (**BVALR**) is **0**). Setting when the buffer is valid (**BVALx** = **1**) may cause unnecessary received messages to be stored.

16.7.14 AMR0 and AMR1: Acceptance Mask Registers 0 and 1

AMR0 BYTE1	15	14	13	12	11	10	9	8
Address: 100024H (CAN0)	AM20	AM19	AM18	AM17	AM16	AM15	AM14	AM13
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)

AMR0 BYTE0	7	6	5	4	3	2	1	0
Address: 100025H (CAN0)	AM28	AM27	AM26	AM25	AM24	AM23	AM22	AM21
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)

AMR0 BYTE3	15	14	13	12	11	10	9	8
Address: 100026H (CAN0)	AM4	AM3	AM2	AM1	AM0	—	—	—
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(—)	(—)	(—)
Initial value:	(X)	(X)	(X)	(X)	(X)	(—)	(—)	(—)

AMR0 BYTE2	7	6	5	4	3	2	1	0
Address: 100027H (CAN0)	AM12	AM11	AM10	AM9	AM8	AM7	AM6	AM5
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)

AMR1 BYTE1	15	14	13	12	11	10	9	8
Address: 100028H (CAN0)	AM20	AM19	AM18	AM17	AM16	AM15	AM14	AM13
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)

AMR1 BYTE0	7	6	5	4	3	2	1	0
Address: 100029H (CAN0)	AM28	AM27	AM26	AM25	AM24	AM23	AM22	AM21
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)

AMR1 BYTE3

	15	14	13	12	11	10	9	8
Address: 10002AH (CAN0)	AM4	AM3	AM2	AM1	AM0	—	—	—
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(—)	(—)	(—)
Initial value:	(X)	(X)	(X)	(X)	(X)	(—)	(—)	(—)

AMR1 BYTE2

	7	6	5	4	3	2	1	0
Address: 10002BH (CAN0)	AM12	AM11	AM10	AM9	AM8	AM7	AM6	AM5
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)

There are two acceptance mask registers, **AMR0** and **AMR1**, both of which are available either in the standard frame format or extended frame format.

AM28 to **AM18** (11 bits) are used for acceptance masks in the standard frame format and **AM28** to **AM0** (29 bits) are used for acceptance masks in the extended format.

0: Compare
Compare the bit of the acceptance code (**ID** register **IDRx** for comparing with the received message **ID**) corresponding to this bit with the bit of the received message **ID**. If there is no match, no message is received.

1: Mask
Mask the bit of the acceptance code **ID** register (**IDRx**) corresponding to this bit. No comparison is made with the bit of the received message **ID**.

Note: **AMR0** and **AMR1** should be set when all the message buffers (**x**) selecting **AMR0** and **AMR1** are invalid (**BVALx** of the message buffer valid register (**BVALR**) is **0**). Setting when the buffers are valid (**BVALx** = **1**) may cause unnecessary received messages to be stored.

16.8 MESSAGE BUFFERS

- There are 16 message buffers.
- One message buffer **x** (**x** = 0 to 15) consists of an **ID** register (**IDRx**), **DLC** register (**DLCRx**), and data register (**DTRx**).
- The message buffer (**x**) is used both for transmission and reception.
- The lower-numbered message buffers are assigned higher priority.
  - At transmission, when a request for transmission is made to more than one message buffer, transmission is performed, starting with the lowest-numbered message buffer (See section 16.10 "TRANSMISSION" on page 444).
  - At reception, when the received message **ID** passes through the acceptance filter (mechanism for comparing the acceptance-masked **ID** of received message and message buffer) of more than one message buffer, the received message is stored in the lowest-numbered message buffer (See 16.11 "RECEPTION" on page 446).
- When the same acceptance filter is set in more than one message buffer, the message buffers can be used as a multi-level message buffer. This provides allowance for receiving time (See 16.12.7 "Setting Configuration of Multi-level Message Buffer" on page 452).

Notes:

1) A write operation to message buffers and general-purpose RAM areas should be performed in words to even addresses only. A write operation in bytes causes undefined data to be written to the upper byte at writing to the lower byte. Writing to the upper byte is ignored.
2) When the **BVALx** bit of the message buffer valid register (**BVALR**) is **0** (Invalid), the message buffers **x** (**IDRx**, **DLCRx**, and **DTRx**) can be used as general-purpose RAM. However, during the receive operation, there may be a maximum waiting time of 64 machine cycles. The same is true for the general-purpose RAM areas (**CAN0**: addresses **001A00H** to **001A1FH** and **CAN1**: addresses **001B00H** to **001B1FH**).

16.8.1 IDRx: ID Register x (x = 0 to 15)

BYTE1	15	14	13	12	11	10	9	8
Address: 10004CH + 4x (CAN0)	ID20	ID19	ID18	ID17	ID16	ID15	ID14	ID13
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)

BYTE0	7	6	5	4	3	2	1	0
Address: 10004DH + 4x (CAN0)	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)

BYTE3	15	14	13	12	11	10	9	8
Address: 10004EH + 4x (CAN0)	ID4	ID3	ID2	ID1	ID0	—	—	—
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(—)	(—)	(—)
Initial value:	(X)	(X)	(X)	(X)	(X)	(—)	(—)	(—)

BYTE2	7	6	5	4	3	2	1	0
Address: 10004FH + 4x (CAN0)	ID12	ID11	ID10	ID9	ID8	ID7	ID6	ID5
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)

This is the **ID** register for message buffer (**x**).

When using the message buffer (**x**) in the standard frame format (**IDEx** of the **IDE** register (**IDER**) = **0**), use 11 bits of **ID28** to **ID18**. When using the buffer in the extended frame format (**IDEx** = **1**), use 29 bits of **ID28** to **ID0**.

ID28 to **ID0** have the following functions:

- Set acceptance code (**ID** for comparing with the received message **ID**).
- Set transmitted message **ID**.

Note: In the standard frame format, setting **1**s to all bits of **ID28** to **ID22** is prohibited).

- Store the received message **ID**.

Note: Received message **ID**s are also stored in acceptance-masked bits. In the standard frame format, **ID17** to **ID0** are undefined.

Notes:

1) A write operation to this register should be performed in words. A write operation in bytes causes undefined data to be written to the upper byte at writing to the lower byte. Writing to the upper byte is ignored.
2) This register should be set when the message buffer (**x**) is invalid (**BVALx** of the message

buffer valid register (**BVALR**) is **0**). Setting when the buffer is valid (**BVALx** = **1**) may cause unnecessary received messages to be stored.

16.8.2 DLCRx: DLC Register x (x = 0 to 15)

	7	6	5	4	3	2	1	0
Address: 10008DH + 2**x** (CAN0)	—	—	—	—	DLC3	DLC2	DLC1	DLC0
Read/write:	(—)	(—)	(—)	(—)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(—)	(—)	(—)	(—)	(X)	(X)	(X)	(X)

This is the **DLC** register for message buffer **x**.

■ **Transmission**

- Set the data length (byte count) of a transmitted message when a data frame is transmitted (**TRTRx** of the transmitting **RTR** register (**TRTRR**) is **0**).
- Set the data length (byte count) of a requested message when a remote frame is transmitted (**TRTRx** = **1**).

Note: Setting other than **0000** to **1000** (0 to 8 bytes) is prohibited.

■ **Reception**

- Store the data length (byte count) of a received message when a data frame is received (**RRTRx** of the remote frame request receiving register (**RRTRR**) is **0**).
- Store the data length (byte count) of a requested message when a remote frame is received (**RRTRx** = **1**).

Note: A write operation to this register should be performed in words. A write operation in bytes causes undefined data to be written to the upper byte at writing to the lower byte, and causes undefined data to be written to the lower byte at writing to the upper byte.

16.8.3 DTRx: Data Register x (x = 0 to 15)

BYTE1

Address: 1000ACH + 8x (CAN0)	15	14	13	12	11	10	9	8
	D7	D6	D5	D4	D3	D2	D1	D0
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)

BYTE0

Address: 1000ADH + 8x (CAN0)	7	6	5	4	3	2	1	0
	D7	D6	D5	D4	D3	D2	D1	D0
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)

BYTE3

Address: 1000AEH + 8x (CAN0)	15	14	13	12	11	10	9	8
	D7	D6	D5	D4	D3	D2	D1	D0
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)

BYTE2

Address: 1000AFH + 8x (CAN0)	7	6	5	4	3	2	1	0
	D7	D6	D5	D4	D3	D2	D1	D0
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)

BYTE5

Address: 1000B0H + 8x (CAN0)	15	14	13	12	11	10	9	8
	D7	D6	D5	D4	D3	D2	D1	D0
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)

BYTE4

Address: 1000B1H + 8x (CAN0)	7	6	5	4	3	2	1	0
	D7	D6	D5	D4	D3	D2	D1	D0
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)

BYTE7

	15	14	13	12	11	10	9	8
Address: 1000B2H + 8**x** (CAN0)	D7	D6	D5	D4	D3	D2	D1	D0
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)

BYTE6

	7	6	5	4	3	2	1	0
Address: 1000B3H + 8**x** (CAN0)	D7	D6	D5	D4	D3	D2	D1	D0
Read/write:	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value:	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)

This is the data register for message buffer (**x**).

This register is used only in transmitting and receiving a data frame but not in transmitting and receiving a remote frame.

- Sets transmitted message data (any of 0 to 8 bytes).
  Data is transmitted in the order of **BYTE0**, **BYTE1**, ..., **BYTE7**, starting with the MSB.
- Stores received message data.
  Data is stored in the order of **BYTE0**, **BYTE1**, ..., **BYTE7**, starting with the MSB.
  Even if the received message data is less than 8 bytes, the remaining bytes of the data register (**DTRx**), to which data are stored, are undefined.

Note: A write operation to this register should be performed in words. A write operation in bytes causes undefined data to be written to the upper byte at writing to the lower byte. Writing to the upper byte is ignored.

16.9 CREG: INTERFACE CONTROL REGISTER

The **CREG** register is intended for configuration of CAN transmit/receive operation clock and CAN User Logic Bus (External Bus) access, when there are different clocks for the CANCLK and the User Logic Bus clock (CLKT). See also chapter 5 "CLOCK GENERATION AND DEVICE STATES" on page 147ff and 6.2.1 "Control Register (CMCR)" on page 195ff.

Note: The CANCLK prescaler value is set by the CMCR register in the clock modulator. Inside the CAN module, CANCLK will be divided by 2 additionally!.

For proper operation the User Logic Bus connected to the CAN controllers has to be programmed at Chip Select Area 7, 16-bit Bus Access, 1 Waitstate and RDY enabled.

The default of the CREG setting uses User Logic Bus clock for CAN register access and CANCLK as CAN transmit/receive operation clock with synchronisation enabled.

The setting with BOOTROM uses the User Logic Bus clock as CAN clock with synchronisation disabled.

		15	14	13	12	11	10	9	8
Address:	10012CH (CAN0)	DTR_W	DTR_B	IDR_W	IDR_B	AMR_W	AMR_B	AMS_W	AMS_B
	Read/write	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
	Initial / Bootrom value	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

		7	6	5	4	3	2	1	0
Address:	10012DH (CAN0)	E_INT	S_INT	C_INV	L_INV	C_CLK	L_CLK	SYNCH	CDSBLE
	Read/write	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
	Inital/ Bootrom value	(0)	(0)	(0)	(0)	(0)	(0/1)	(0/1)	(0)

[Bit 15]...[Bit 8] **Byte ordering**

These bits can be used to change the byte ordering of the following registers :
AMS, AMR0, AMR1, IDRx[10:0], and DTRx[15:0].

If xxx_B=1 and xxx_W=0 byte ordering is changed

from	Byte0	Byte1	Byte2	Byte3	to	Byte1	Byte0	Byte3	Byte2

If xxx_W=1 and xxx=B=0 byte ordering is changed

from	Byte0	Byte1	Byte2	Byte3	to	Byte2	Byte3	Byte0	Byte1

If xxx_W=1 and xxx_B=1 byte ordering is changed

from	Byte0	Byte1	Byte2	Byte3	to	Byte3	Byte2	Byte1	Byte0

[Bit 7]...[Bit 4] **E_INT, S_INT, C_INV, L_INV**

These bits are for special CPU configurations. Proper operation can not be guaranteed if they are changed from the default values.

[Bit 3:2] **C_CLK**: CANCLK, **L_CLK**: User Logic Bus Clock (CLKT)

These bits select the clock for CAN User Logic Bus access and CAN transmit/receive operation.

C_CLK	L_CLK	Mode
0	0	CANCLK (divided by 2) is used for CAN operation, CLKT is used for the access to CAN registers connected to the User Logic Bus; synchronization mode (SYNCH = 0) is strongly recommended
0	1	reserved mode for testing purposes
1	0	reserved mode for testing purposes
1	1	reserved mode for testing purposes

[Bit 1] **SYNCH**: Synchronisation enable bit

This bit enables or disables synchronisation of CPU main clock and User Logic Bus clock.

0: Synchronisation enabled

1: Synchronisation disabled

[Bit 0] **CDSBLE**: Clock disable bit

This bit enables or disables the CAN clock and CAN bus operation.

0: Clock and CAN bus operation enabled

1: Clock and CAN bus operation disabled

Notes:

1) For CAN operation frequency with User Logic Bus clocking the BOOTROM settings can be used.

2) The CAN operation frequency that is dividable by the CAN prescaler register (**BTR[5:0]**) is always the selected clock and its frequency divided by 2.

16.10 TRANSMISSION

■ Starting Transmission

When **1** is written to **TREQx** of the transmission request register (**TREQR**), transmission by the message buffer (**x**) starts. At this time, **TREQx** becomes **1** and **TCx** of the transmission complete register (**TCR**) becomes **0**.

If **RFWTx** of the remote frame receiving wait register (**RFWTR**) is **0**, transmission starts immediately. If **RFWTx** is **1**, transmission starts after waiting until a remote frame is received (**RRTRx** of the remote request receiving register (**RRTRR**) becomes **1**).

■ Transmitting

If a request for transmission is made to more than one message buffer (more than one **TREQx** is **1**), transmission is performed, starting with the lowest-numbered message buffer.

Message transmission to the CAN bus (by the transmit output pin **TX**) starts when the bus is idle.

If **TRTRx** of the transmission **RTR** register (**TRTRR**) is **0**, a data frame is transmitted. If **TRTRx** is **1**, a remote frame is transmitted.

If the message buffer competes with other CAN controllers on the CAN bus for transmission and arbitration fails, or if an error occurs during transmission, the message buffer waits until the bus is idle and repeats retransmission until it is successful.

■ Canceling transmission request

- Canceling by transmission cancel register (**TCANR**)

  A transmission request for message buffer (**x**) having not executed transmission during transmission pending can be canceled by writing **1** to **TCANx** of the transmission cancel register (**TCANR**). At completion of cancellation, **TREQx** becomes **0**.

- Canceling by storing received message

  The message buffer (**x**) having not executed transmission despite transmission request also performs reception.

  If the message buffer (**x**) has not executed transmission despite a request for transmission of a data frame (**TRTRx** = **0** or **TREQx** = **1**), the transmission request is canceled after storing received data frames passing through the acceptance filter (**TREQx** = **0**).

  Note: A transmission request is not canceled by storing remote frames (**TREQx** = **1** remains unchanged).

  If the message buffer (**x**) has not executed transmission despite a request for transmission of a remote frame (**TRTRx** = **1** or **TREQx** = **1**), the transmission request is canceled after storing received remote frames passing through the acceptance filter (**TREQx** = **0**).

  Note: The transmission request is canceled by storing either data frames or remote frames.

■ Completing transmission

When transmission is successful, **RRTRx** becomes **0**, **TREQx** becomes **0**, and **TCx** of the transmission complete register (**TCR**) becomes **1**.

If the transmission complete interrupt is enabled (**TIEx** of the transmission complete interrupt enable register (**TIER**) is **1**), an interrupt occurs.

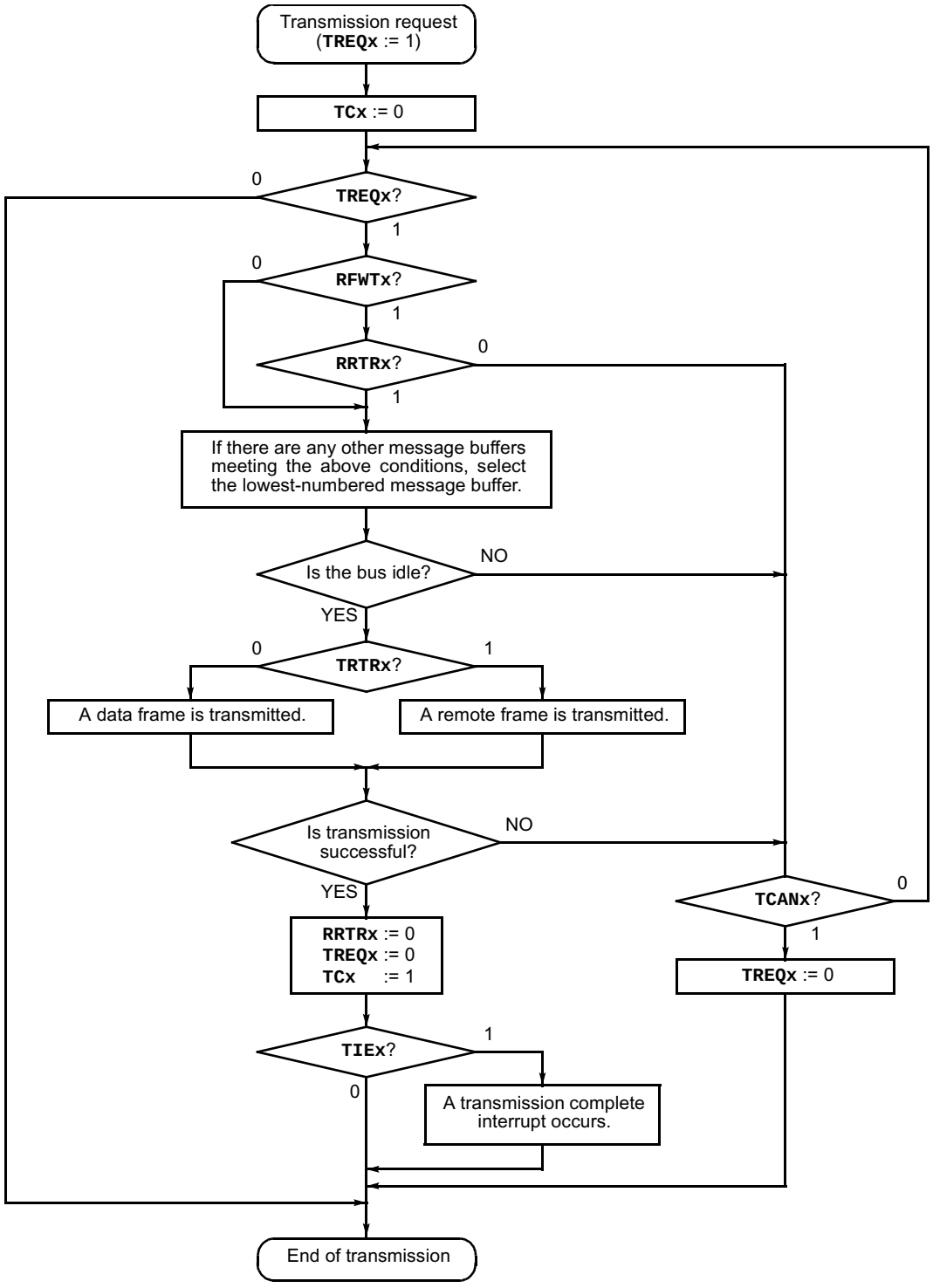


Figure 16.10 Transmission Flowchart

16.11 RECEPTION

■ **Starting reception**

Reception starts when the start of data frame or remote frame (SOF) is detected on the CAN bus.

■ **Acceptance filtering**

The received message in the standard frame format is compared with the message buffer (**x**) set (**IDEx** of the **IDE** register (**IDER**) is **0**) in the standard frame format. The received message in the extended frame format is compared with the message buffer (**x**) set (**IDEx** is **1**) in the extended frame format.

If all the bits set to ***Compare*** by the acceptance mask agree after comparison between the received message **ID** and acceptance code (**ID** register (**IDRx**) for comparing with the received message ID), the received message passes to the acceptance filter of the message buffer (**x**).

■ **Storing received message**

When the receive operation is successful, received messages are stored in a message buffer **x** including **ID**s passed through the acceptance filter.

When receiving data frames, received messages are stored in the **ID** register (**IDRx**), **DLC** register (**DLCRx**), and data register (**DTRx**).

Even if received message data is less than 8 bytes, the data is stored in the remaining bytes of the **DTRx** and its value is undefined.

When receiving remote frames, received messages are stored only in the **IDRx** and **DLCRx**, and the **DTRx** remains unchanged.

If there is more than one message buffer including **ID**s passed through the acceptance filter, the message buffer **x** in which received messages are to be stored is determined according to the following rules.

- The order of priority of the message buffer **x** (**x** = 0 to 15) rises as its number lower; in other words, message buffer 0 is given the highest and the message buffer 15 is given the lowest priority.
- Basically, message buffers with the **RCx** bit of the receive completion register (**RCR**) set to **0** are preferred in storing received messages.
- If the bits of the acceptance mask select register (**AMSR**) are set to All Bits Compare (for message buffers with the **AMSx.1** and **AMSx.0** bits set to **00**), received messages are stored irrespective of the setting of the **RCx** bit of the **RCR**.
- If there are message buffers with the **RCx** bit of the **RCR** set to **0**, or with the bits of the **AMSR** set to All Bits Compare, received messages are stored in the lowest-number (highest-priority) message buffer **x**.
- If there are no message buffers above-mentioned, received messages are stored in a lower-number message buffer **x**.

Figure 16.11a shows the flowchart in determining the message buffer **x** where received messages are to be stored.

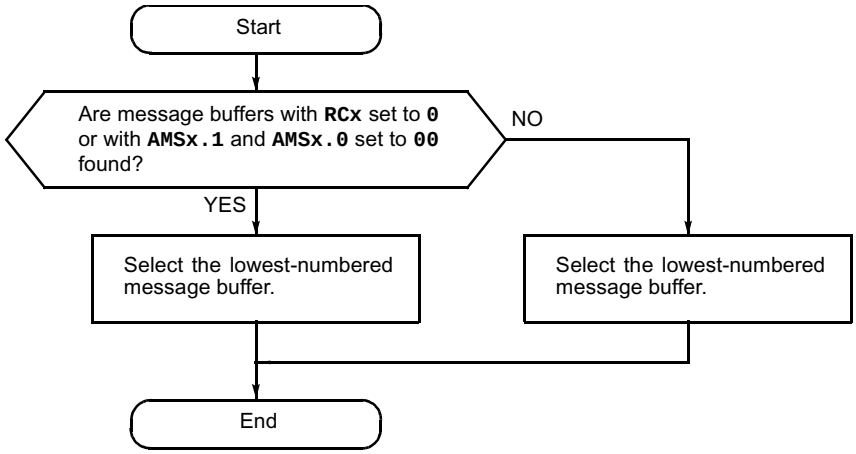


Figure 16.11a Flowchart Determining Message Buffer (x) where Received Messages Stored

Message buffers should be arranged in ascending numeric order; that is, message buffer with the bits of the **AMSR** set to All Bits Compare, message buffer using **AMR0** or **AMR1**, and message buffer with the bits of the **AMSR** set to All Bits Mask.

■ Receive overrun

When the **RCx** bit of the receive complete register (**RCR**) corresponding to the message buffer **x** where received messages are to be stored is set to **1** and storing of received messages in the message buffer **x** is completed, the **ROVRx** bit of the receive overrun register (**ROVRR**) is set to **1**, indicating receive overrun.

■ Processing for reception of data frame and remote frame

- Processing for reception of data frame

  RRTRx of the remote request receiving register (**RRTRR**) becomes **0**.

  TREQx of the transmission request register (**TREQR**) becomes **0** (immediately before storing the received message). A transmission request for message buffer (**x**) having not executed transmission will be canceled.

  Note: A request for transmission of either a data frame or remote frame is canceled.

- Processing for reception of remote frame

  RRTRx becomes **0**.

  If **TRTRx** of the transmitting **RTR** register (**TRTRR**) is **1**, **TREQx** becomes **0**. A request for transmitting remote frame to message buffer having not executed transmission will be canceled.

  Notes:

  1) A request for data frame transmission is not canceled.

  2) For cancellation of a transmission request, see section 16.10 "TRANSMISSION" on page 444.

■ Completing reception

RCx of the reception complete register (**RCR**) becomes **1** after storing the received message.

If a reception interrupt is enabled (**RIEx** of the reception interrupt enable register (**RIER**) is **1**), an interrupt occurs.

Note: This CAN controller will not receive any messages transmitted by itself.

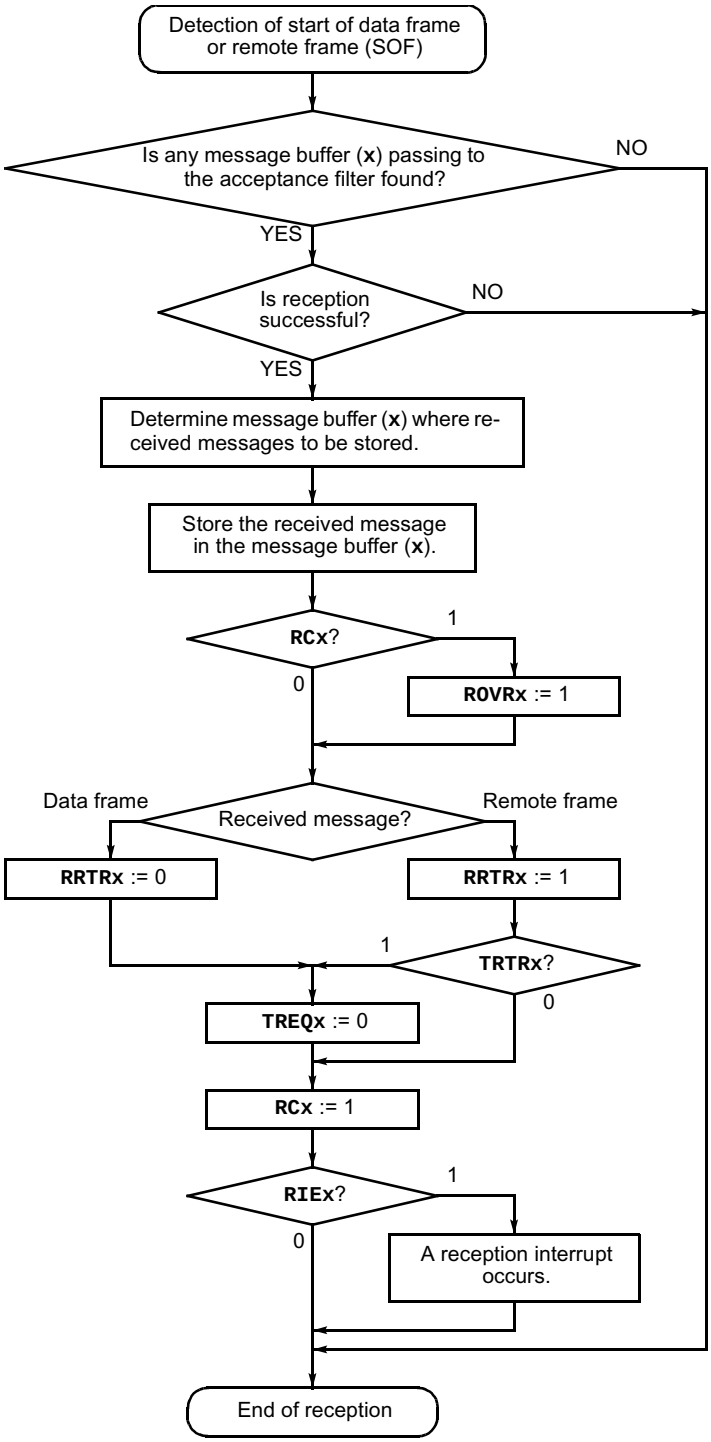


Figure 16.11b Reception Flowchart

16.12 USAGE PROCEDURE

16.12.1 Setting Bit Timing

The bit timing register (**BTR**) should be set during bus operation stop (when the bus operation stop bit (**HALT**) of the control status register (**CSR**) is **1**).

After the setting completion, write **0** to **HALT** to cancel bus operation stop.

16.12.2 Setting Frame Format

Set the frame format used by the message buffer (**x**). When using the standard frame format, set **IDEx** of the **IDE** register (**IDER**) to **0**. When using the extended frame format, set **IDEx** to **1**.

This setting should be made when the message buffer (**x**) is invalid (**BVALx** of the message buffer valid register (**BVALR**) is **0**). Setting when the buffer is valid (**BVALx** = **1**) may cause unnecessary received messages to be stored.

16.12.3 Setting ID

Set the message buffer (**x**) **ID** to **ID28** to **ID0** of **ID** register (**IDRx**). The message buffer (**x**) **ID** need not be set to **ID11** to **ID0** in the standard frame format. The message buffer (**x**) **ID** is used as a transmission message at transmission and is used as an acceptance code at reception.

This setting should be made when the message buffer (**x**) is invalid (**BVALx** of the message buffer valid register (**BVALR**) is **0**). Setting when the buffer is valid (**BVALx** = **1**) may cause unnecessary received messages to be stored.

16.12.4 Setting Acceptance Filter

The acceptance filter of the message buffer (x) is set by an acceptance code and acceptance mask set. It should be set when the acceptance message buffer (x) is invalid (BVALx of the message buffer enable register (BVALR) is 0). Setting when the buffer is valid (BVALx = 1) may cause unnecessary received messages to be stored.

Set the acceptance mask used in each message buffer (x) by the acceptance mask select register (AMSR). The acceptance mask registers (AMR0 and AMR1) should also be set if used (For the setting details, see 16.7.13 and 16.7.14).

The acceptance mask should be set so that a transmission request may not be canceled when unnecessary received messages are stored. For example, it should be set to a full-bit comparison if receiving only the message of the same **ID** as transmission **ID**.

16.12.5 Procedure for Transmission by Message Buffer (x)

After the completion of setting 16.12.2 to 16.12.4, set **BVALx** to **1** to make the message buffer (**x**) valid.

■ **Setting transmit data length code**

Set the transmit data length code (byte count) to **DLC3** to **DLC0** of the **DLC** register (**DLCRx**).

For data frame transmission (when **TRTRx** of the transmission **RTR** register (**TRTRR**) is **0**), set the data length of the transmitted message.

For remote frame transmission (when **TRTRx** = **1**), set the data length (byte count) of the requested message.

Note: Setting other than **0000** to **1000** (0 to 8 bytes) is prohibited.

■ **Setting transmit data (only for transmission of data frame)**

For data frame transmission (when **TRTRx** of the transmission register (**TRTRR**) is **0**), set data as the count of byte transmitted in the data register (**DTRx**).

Note: Transmit data should be rewritten with the **TREQx** bit of the transmission request register (**TREQR**) set to **0**. There is no need for setting the **BVALx** bit of the message buffer valid register (**BVALR**) to **0**. Setting the **BVALx** bit to **0** disables remote frame receiving.

■ **Setting transmission RTR register**

For data frame transmission, set **TRTRx** of the transmission **RTR** register (**TRTRR**) to **0**.

For remote frame transmission, set **TRTRx** to **1**.

■ **Setting conditions for starting transmission (only for transmission of data frame)**

Set **RFWTx** of the remote frame receiving wait register (**RFWTR**) to **0** to start transmission immediately after a request for data frame transmission is set (**TREQx** of the transmission request register (**TREQR**) is **1** and **TRTRx** of the transmission **RTR** register (**TRTRR**) is **0**).

Set **RFWTx** to **1** to start transmission after waiting until a remote frame is received (**RRTRx** of the remote request receiving register (**RRTRR**) becomes **1**) after a request for data frame transmission is set (**TREQx** = **1** and **TRTRx** = **0**).

Remote frame transmission can not be made, if **RFWTx** is set to **1**.

■ **Setting transmission complete interrupt**

When generating a transmission complete interrupt, set **TIEx** of the transmission complete interrupt enable register (**TIER**) to **1**.

When not generating a transmission complete interrupt, set **TIEx** to **0**.

■ **Setting transmission request**

For a transmission request, set **TREQx** of the transmission request register (**TREQR**) to **1**.

■ **Canceling transmission request**

When canceling a pending request for transmission to the message buffer (**x**), write **1** to **TCANx** of the transmission cancel register (**TCANR**).

Check **TREQx**. For **TREQx** = **0**, transmission cancellation is terminated or transmission is completed. Check **TCx** of the transmission complete register (**TCR**). For **TCx** = **0**, transmission cancellation is terminated. For **TCx** = **1**, transmission is completed.

■ **Processing for completion of transmission**

If transmission is successful, **TCx** of the transmission complete register (**TCR**) becomes **1**.

If the transmission complete interrupt is enabled (**TIEx** of the transmission complete interrupt enable register (**TIER**) is **1**), an interrupt occurs.

After checking the transmission completion, write **0** to **TCx** to set it to **0**. This cancels the transmission complete interrupt.

In the following cases, the pending transmission request is canceled by receiving and storing a message.

- Request for data frame transmission by reception of data frame
- Request for remote frame transmission by reception of data frame
- Request for remote frame transmission by reception of remote frame

Request for data frame transmission is not canceled by receiving and storing a remote frame. **ID** and **DLC**, however, are changed by the **ID** and **DLC** of the received remote frame. Note that the **ID** and **DLC** of data frame to be transmitted become the value of received remote frame.

16.12.6 Procedure for Reception by Message Buffer (x)

After setting 16.12.2 to 16.12.4, set as follows.

■ **Setting reception interrupt**

When a reception interrupt is enabled, set **RIEx** of the reception interrupt enable register (**RIER**) to **1**.

When a reception interrupt is disabled, set **RIEx** to **0**.

■ **Starting reception**

When starting reception after setting, set **BVALx** of the message buffer valid register (**BVALR**) to **1** to make the message buffer (**x**) valid.

■ **Processing for reception completion**

If reception is successful after passing to the acceptance filter, the received message is stored in the message buffer (**x**) and **RCx** of the reception complete register (**RCR**) becomes **1**. For data frame reception, **RRTRx** of the remote request receiving register (**RRTRR**) becomes **0**. For remote frame reception, **RRTRx** becomes **1**.

If a reception interrupt is enabled (**RIEx** of the reception interrupt enable register (**RIER**) is **1**), an interrupt occurs.

After checking the reception completion (**RCx** = **1**), process the received message.

After completion of processing the received message, check **ROVRx** of the reception overrun register (**ROVRR**).

For **ROVRx** = **0**, the processed received message is valid. Write **0** to **CRx** to set it to **0** (the reception complete interrupt is also canceled) to terminate reception.

For **ROVRx** = **1**, a reception overrun occurred and the new received message may overwrite the processed received message. In this case, received messages should be processed again after setting the **ROVRx** bit to **0** by writing **0** to it.

Figure 16.12.6a shows an example of receive interrupt handling.

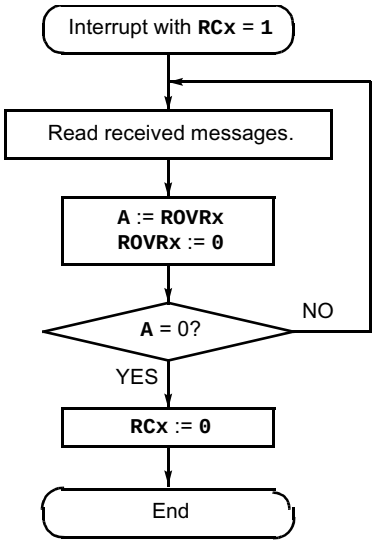


Figure 16.12.6a Example of Receive Interrupt Handling

16.12.7 Setting Configuration of Multi-level Message Buffer

If the receptions are performed frequently, or if an unspecified count of message is received, in other words, if there is insufficient time for receiving messages, more than one message buffer can be combined into a multi-level message buffer to provide allowance for processing time of the received message by CPU.

To provide a multi-level message buffer, the same acceptance filter must be set in the combined message buffers.

If the bits of the acceptance mask select register (**AMSR**) are set to All Bits Compare ((**AMSx.1**, **AMSx.0**) = (**0**, **0**)), multi-level message configuration of message buffers is not allowed. This is because All Bits Compare causes received messages to be stored irrespective of the value of the **RCx** bit of the receive completion register (**RCR**), so received messages are always stored in lower-numbered (lower-priority) message buffers even if All Bits Compare and identical acceptance code (**ID** register (**IDRx**)) are specified for more than one message buffer. Therefore, All Bits Compare and identical acceptance code should not be specified for more than one message buffer.

Figure 16.12.7a shows operational examples of multi-level message buffers.

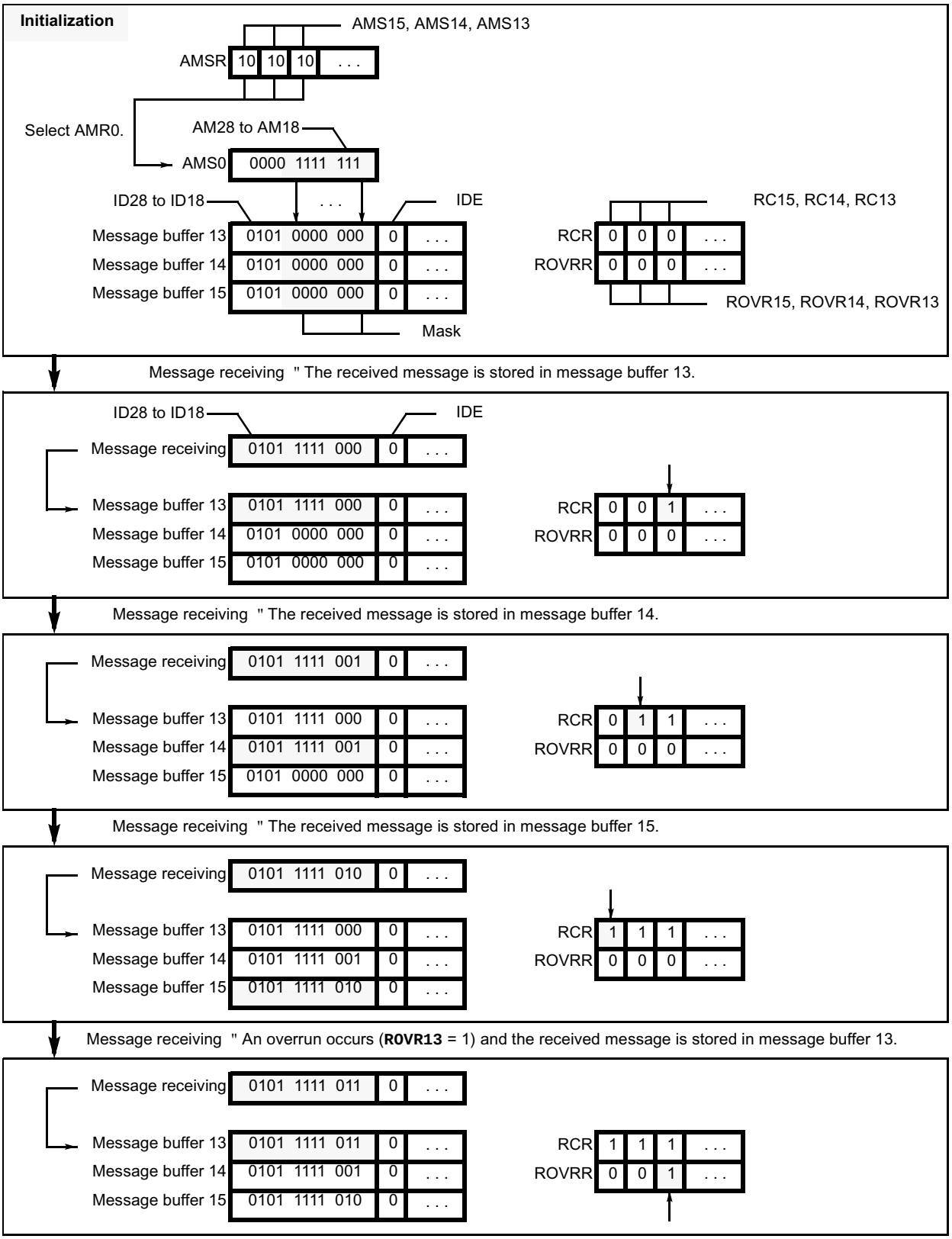


Figure 16.12.7a Examples of Operation of Multi-level Message Buffer

Note: Four messages are received with the same acceptance filter set in message buffers 13, 14 and 15.

CHAPTER 17 D/A CONVERTER

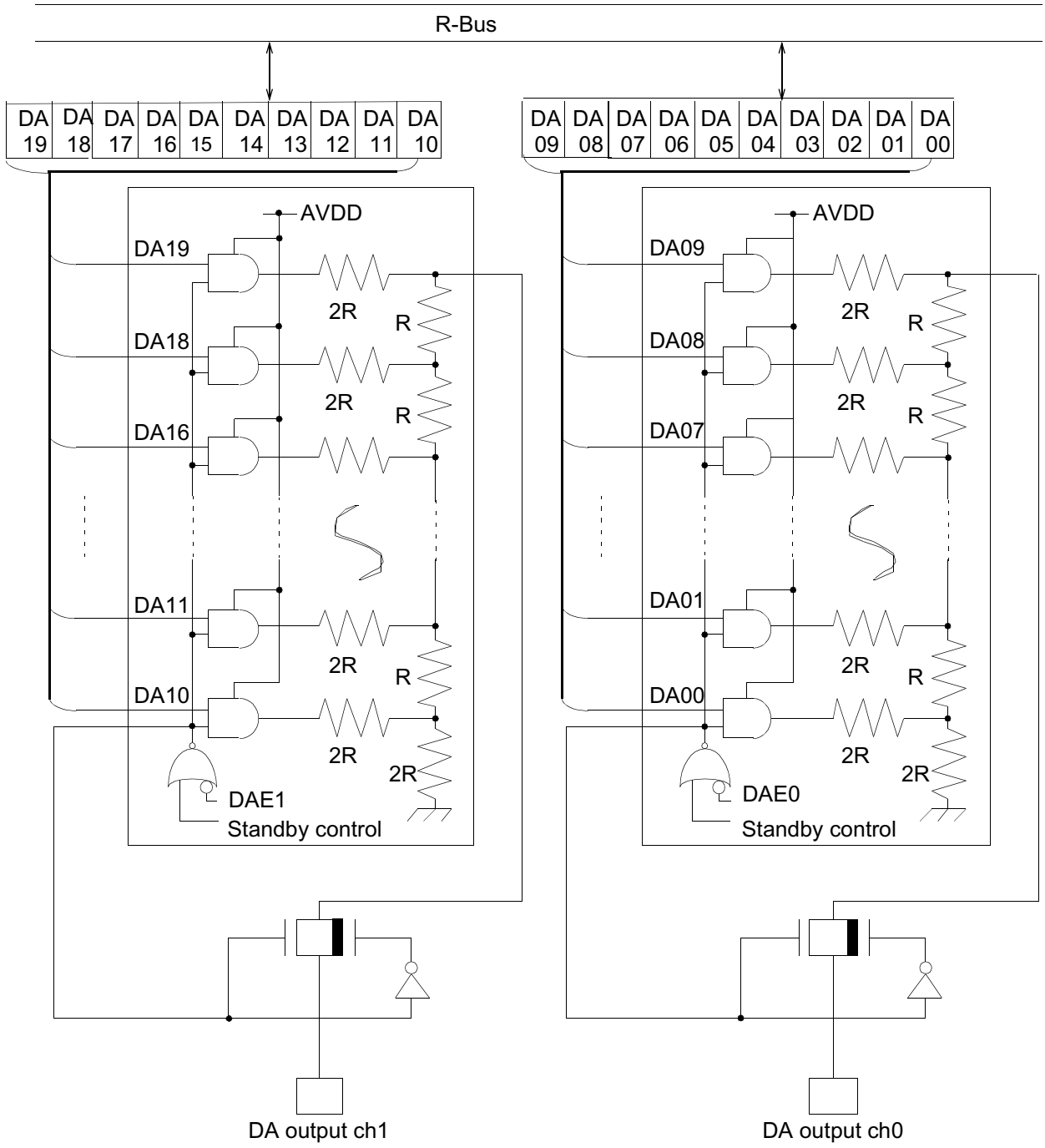
This Chapter provides an overview of the D/A converter, describes the register structure and functions, and describes the operarton of the D/A converter.

This block is an R-2R format D/A converter, having ten-bit resolution. The D/A converter has two channels. Output control can be performed independently for the two channels using the D/A control register.

For the electrical specification, please see section
34.7 "CONVERTER CHARACTERISTICS" on page 737.

17.1 BLOCK DIAGRAM

Figure 17.1a Block diagram of the D/A converter.



17.2 REGISTERS

	bit	7	6	5	4	3	2	1	0	
DACR Address : 0000A5H		------	------	------	------	------	MODE	DAE1	DAE0	D/A control register
	bit	15	14	13	12	11	10	9	8	
DADR0 Address : 0000A6H		------	------	------	------	------	------	DA09	DA08	D/A converter data register (ch 0)
	bit	7	6	5	4	3	2	1	0	
DADR0 Address : 0000A7H		DA07	DA06	DA05	DA04	DA03	DA02	DA01	DA00	D/A converter data register (ch 0)
	bit	15	14	13	12	11	10	9	8	
DADR1 Address : 0000A8H		------	------	------	------	------	------	DA19	DA18	D/A converter data register (ch 1)
	bit	7	6	5	4	3	2	1	0	
DADR1 Address : 0000A9H		DA17	DA16	DA15	DA14	DA13	DA12	DA11	DA10	D/A converter data register (ch 1)
	bit	7	6	5	4	3	2	1	0	
DDBL Address : 0000ABH		------	------	------	------	------	------	------	DBL	D/A clock control

17.3 REGISTER DETAILS

17.3.1 DACR (D/A control register)

	bit	7	6	5	4	3	2	1	0	
DACR Address : 0000A5H		—	—	—	—	—	MODE	DAE1	DAE0	Initial value -----000B
							R/W	R/W	R/W	

[bits 1,0] DAE1 and DAE0

These bits are used to enable or disable the D/A converter output. DAE1 controls channel 1 output, while DAE0 controls channel 0 output.

When '1' is written to these bits, D/A output is enabled. When '0' is set, D/A output is disabled.

These bits are initialized to '0' upon a reset. These bits are readable and writable.

[bit 2] MODE

This bit is used to control the mode of the D/A converter.

When '1' is written to this bit, the D/A converter is operating with 8 bit resolution. When '0' is set, the D/A converter is operating with 10 bit resolution.

This bit is initialized to '0' upon a reset. This bit is readable and writable.

17.3.2 DADR (D/A data register)

DADR0 bit
Address : 0000A6H

15	14	13	12	11	10	9	8
------	------	------	------	------	------	DA09	DA08
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value XXXXXXXXB

DADR0 bit
Address : 0000A7H

7	6	5	4	3	2	1	0
DA07	DA06	DA05	DA04	DA03	DA02	DA01	DA00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value XXXXXXXXB

DADR1 bit
Address : 0000A8H

15	14	13	12	11	10	9	8
------	------	------	------	------	------	DA19	DA18
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value XXXXXXXXB

DADR1 bit
Address : 0000A9H

7	6	5	4	3	2	1	0
DA17	DA16	DA15	DA14	DA13	DA12	DA11	DA10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value XXXXXXXXB

DADR0 [bits 9 to 0] DA09 to DA00

These bits are used to set the output voltage of D/A converter ch0.

These bits are not initialized upon a reset. These bits are readable and writable.

DADR1 [bits 9 to 0] DA19 to DA10

These bits are used to set the output voltage of D/A converter ch1.

These bits are not initialized upon a reset. These bits are readable and writable.

When reading those registers in the 8 bit resolution mode, the value written to DAx7...DAx0 will appear at DAx9 ...DAx2.

17.3.3 DDBL (D/A clock control register)

DDBL bit
Address : 0000ABH

7	6	5	4	3	2	1	0
------	------	------	------	------	------	------	DBL
							R/W

Initial value -------0B

[bit 0] DBL

This bit is used to control the clock for the D/A converter module.

When '1' is written to this bit, the clock for the D/A converter module is disabled. When '0' is set, a clock is supplied to the D/A converter module.

This bit is initialized to '0' upon a reset. This bit is readable and writable.

17.4 OPERATIONS

D/A output is started by writing a desired D/A output value to the D/A data register (DADR) and setting '1' to the enable bit for the corresponding D/A output channel in the D/A control register (DACR).

Disabling D/A output turns off the analog switch that is inserted serially into the output of each D/A converter channel. In addition, the D/A converter is internally cleared to '0' and the path of the DC current is shut down. This also applies in stop mode.

Figure 14.4.1 shows the theoretical values of D/A converter output voltages

The D/A converter output voltages are between 0 V and (255/256) V × AVCC in 8 bit mode, and (1023/1024) V x AVCC in 10 bit mode respectively. The output voltage range is changed by regulating the AVCC voltage externally.

The D/A converter output does not have an internal buffer amplifier. Since an analog switch (max. 100 Ω) is serially inserted into the output, allow sufficient settling time when applying an external output load.

Table 17.4a Theoretical values of D/A converter output voltages for 8 bit resolution

Values written to DA07 to DA00 and DA17 to DA10	Theoretical values of output voltages
00H	AVSS + 0 * 1 LSB
01H	AVSS + 1 * 1 LSB
02H	AVSS + 2 * 1 LSB
⋮	⋮
FDH	AVSS + 253 * 1 LSB
FEH	AVSS + 254 * 1 LSB
FFH	AVSS + 255 * 1 LSB

1 LSB = (AVCC-AVSS)/256

Table 17.4b Theoretical values of D/A converter output voltages for 10 bit resolution

Values written to DA09 to DA00 and DA19 to DA10	Theoretical values of output voltages
000H	AVSS + 0 * 1 LSB
001H	ACSS + 1 * 1 LSB
002H	AVSS + 2 * 1 LSB
⋮	⋮
3FDH	AVSS + 1021 * 1 LSB
3FEH	AVSS + 1022 * 1 LSB
3FFH	AVSS + 1023 * 1 LSB

1 LSB = (AVCC-AVSS)/1024

CHAPTER 18 100 kHz I²C INTERFACE

This section describes the functions and operation of the MB91360 series basic I²C interface. This interface allows operation up to 100 kHz and 8-bit-addressing.

18.1 I²C INTERFACE OVERVIEW

The I²C interface is a serial I/O port supporting the Inter IC bus, operating as a master/slave device on the I²C bus.

■ I²C Interface Features

The MB91360 series microcontroller includes a built-in one-channel I²C interface. The I²C interface has the following features.

- Master/slave sending and receiving functions
- Arbitration function
- Clock synchronization function
- Slave address/general call address detection function
- Transfer direction detection function
- Repeated start condition generation and detection function
- Bus error detection function

■ I²C Interface Registers

● Bus Status Register (IBSR)

Bus status register

	7	6	5	4	3	2	1	0	⇐ Bit no.
Address : 000095H	BB	RSC	AL	LRB	TRX	AAS	GCA	FBT	IBSR
Read/write ⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

● Bus Control Register (IBCR)

Bus control register

	15	14	13	12	11	10	9	8	⇐ Bit no.
Address : 000094H	BER	BEIE	SCC	MSS	ACK	GCAA	INTE	INT	IBCR
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

● Clock control register (ICCR)

Clock control register

	7	6	5	4	3	2	1	0	⇐ Bit no.
Address : 000097H	–	–	EN	CS4	CS3	CS2	CS1	CS0	ICCR
Read/write ⇒	(–)	(–)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(–)	(–)	(0)	(X)	(X)	(X)	(X)	(X)	

● Address Register (IADR)

Address register

	15	14	13	12	11	10	9	8	⇐ Bit no.
Address : 000096H	–	A6	A5	A4	A3	A2	A1	A0	IADR
Read/write ⇒	(–)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(–)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	

- Data Register (IDAR)

Data register	7	6	5	4	3	2	1	0	⇐ Bit no.
Address : 000099H	D7	D6	D5	D4	D3	D2	D1	D0	IDAR
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	

- Clock Disable Register (IDBL)

Data register	7	6	5	4	3	2	1	0	⇐ Bit no.
Address : 00009BH	---	---	---	---	---	---	---	DBL	IDBL
Read/write ⇒	(-)	(-)	(-)	(-)	(-)	(-)	(-)	(R/W)	
Default value⇒	(-)	(-)	(-)	(-)	(-)	(-)	(-)	(0)	

18.2 I²C INTERFACE BLOCK DIAGRAM

Figure 18.2a shows the I²C interface block diagram.

■ **I²C Interface Block Diagram**

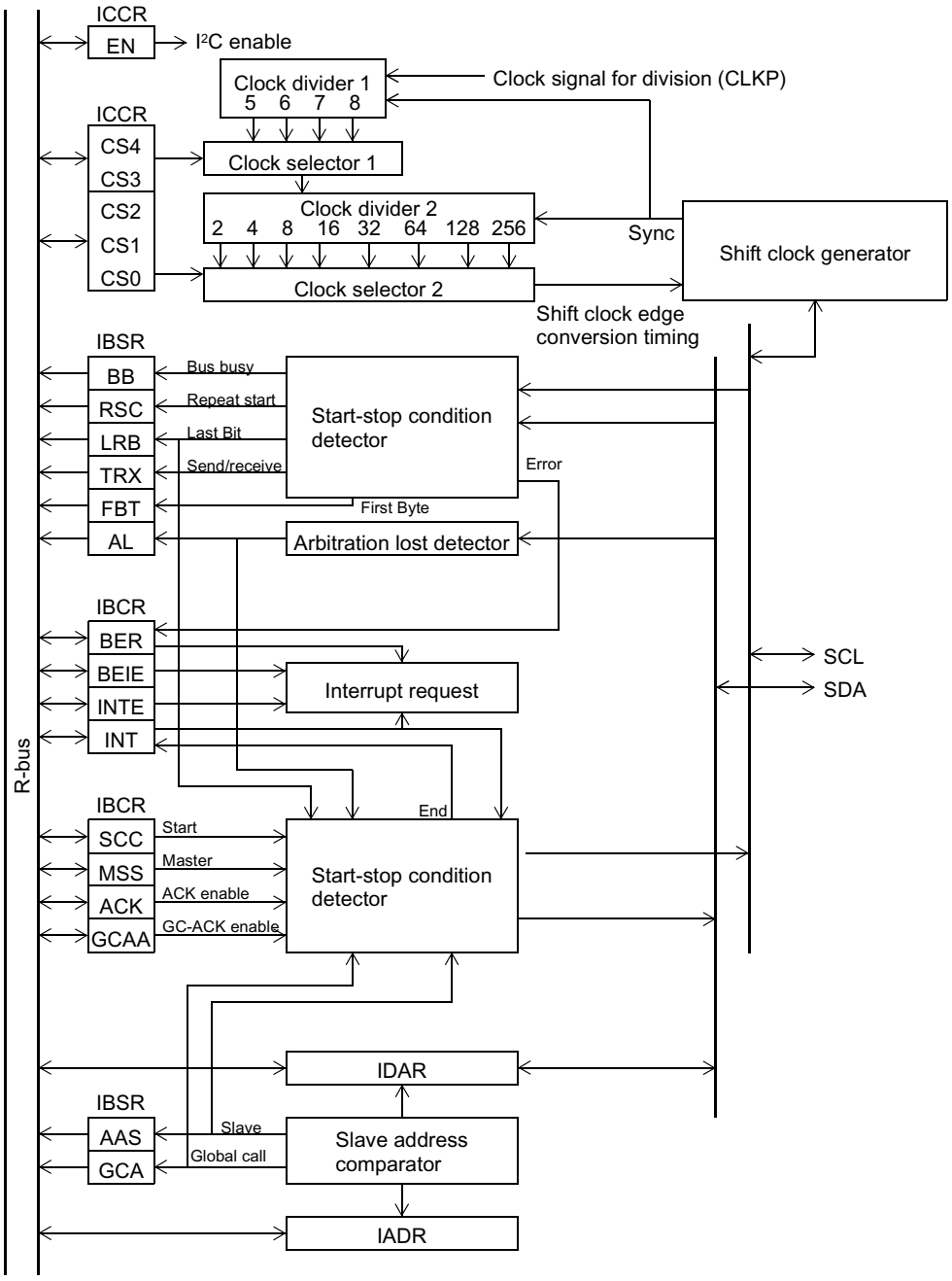


Figure 18.2a I²C Interface Block Diagram

18.3 I²C REGISTERS

18.3.1 Bus Status Register (IBSR)

The bus status register (IBSR) has the following functions.

- **Repeated start condition detection**
- **Arbitration lost detection**
- **Acknowledge detection**
- **Data transfer**
- **Addressing detection**
- **General call address detection**
- **First byte detection**

■ Register Configuration

Bus status register	7	6	5	4	3	2	1	0	⇐ Bit no.
Address : 000095H	BB	RSC	AL	LRB	TRX	AAS	GCA	FBT	IBSR
Read/write ⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

■ Bit Description

[bit 7] BB (Bus Busy)

This bit indicates the status of the I²C bus.

0	Stop condition detected.
1	Start condition detected (bus in use).

[bit 6] RSC (Repeated Start Condition)

This bit indicates detection of a repeated start condition.

0	Repeated start condition not detected.
1	Bus in use, restart condition detected.

This bit is cleared by writing "0" to the INT bit and detection of a bus pause condition or stop condition when not addressed in slave mode.

[bit 5] AL (Arbitration Lost)

This bit detects an arbitration lost condition.

0	No arbitration last condition detected.
1	Arbitration lost event occurred during master sending, or "1" was written to MSS bit while another bus system was using the bus.

This bit is cleared by writing "0" to the INT bit.

[bit 4] LRB (Last Received Bit)

This bit is used to store acknowledge message from the receiving side.

It is cleared by a start condition or stop condition.

[bit 3] TRX (Transfer/Receive)

This bit indicates either sending or receiving operation during data transfer.

0	Receiving operation
1	Sending operation

[bit 2] AAS (Addressed As Slave)

This bit indicates detection of an addressing.

0	Not addressed as slave
1	Addressed as slave

This bit is cleared by a start condition or stop condition.

[bit 1] GCA (General Call Address)

This bit indicates detection of a general call address (00H).

0	General call address not received as slave.
1	General call address received as slave.

This bit is cleared by a start condition or stop condition.

[bit 0] FBT (First Byte Transfer)

This bit indicates detection of a first byte.

0	Incoming data is not a first byte.
1	Incoming data is a first byte (address data).

Even when set to "1" by a start condition, this bit is cleared by writing "0" to the INT bit, or when not addressed in slave mode.

18.3.2 Bus Ccontrol Register (IBCR)

The bus control register (IBCR) has the following functions.

- **Interrupt request/interrupt enabling**
- **Start condition generation**
- **Master/slave selection**
- **Acknowledge generation enabling**

■ Register Configuration

Bus control register — Address : 000094H

15	14	13	12	11	10	9	8	⇐ Bit no.
BER	BEIE	SCC	MSS	ACK	GCAA	INTE	INT	IBCR
(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	Read/write ⇒
(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	Default value⇒

■ Bit Configuration

[bit 15] BER (Bus ERror)

This bit is the bus error interrupt flag.

(Write access)

0	Clear bus error interrupt flag.
1	No effect

(Read access)

0	No bus error detected
1	Data transfer in progress, illegal start, or stop condition detected

When this bit is set, the EN bit in the CCR register is cleared, the I^2C interface goes to pause status, and data transfer is interrupted.

[bit 14] BEIE (Bus Error Interrupt Enable)

This bit enables the bus error interrupt.

0	Bus error interrupt disabled
1	Bus error interrupt enabled

Setting this bit to "1" enables an interrupt to be generated when the BER bit value is set to "1."

[bit 13] SCC (Start Condition Continue)

This bit is used to generate a start condition.

0	No effect
1	Generate restart condition during master transfer.

The read value of this bit is always "0."

[bit 12] MSS (Master Slave Select)

This is the master/slave select bit.

0	Go to slave mode after a stop condition is generated and transfer ends.
1	Go to master mode, generate start condition and start transfer.

This bit is cleared and the mode goes to slave mode if an arbitration lost event occurs during master sending.

[bit 11] ACK (ACKnowledge)

This is the acknowledge generation enable bit on receiving the data.

0	No acknowledge generation.
1	Acknowledge generation.

This bit is not valid when receiving address data in slave mode.

[bit 10] GCAA (General Call Address Acknowledge)

This bit enables acknowledge generation when a general call address is received.

0	No acknowledge generation.
1	Acknowledge generation.

[bit 9] INTE (INTerrupt Enable)

This bit is the interrupt enable.

0	Interrupt disabled
1	Interrupt enabled

Setting this bit to "1" enables interrupt generation when the INT bit is set to "1."

[bit 8]: INT (INTerrupt)

This bit is the transfer end interrupt request flag.

(Write access)

0	Clear transfer end interrupt request flag.
1	No effect

(Read access)

0	Transfer not ended.
1	Set at end of a 1-byte transfer including an acknowledge bit, under the following conditions: • Device is bus master. • Device is addressed as slave. • General call address received. • Arbitration loss event occurred. • Start condition attempted while another system was using bus.

While this bit is set to "1" the SCL line will hold an "L" level signal. Writing "0" to this bit clears the setting, releases the SCL line, and executes transfer of the next byte. In addition, this bit is cleared to "0" when a start condition or stop condition occurs when in master mode.

■ SCC, MSS, INT Bit Competition

Simultaneously writing to the SCC, MSS and INT bits causes competition to transfer the next byte, generate a start condition, and generate a stop condition. In this case, the order of priority is as follows.

1) Next byte transfer and stop condition generation.

When "0" is written to the INT bit and "0" is written to the MSS bit, the MSS bit takes priority and a stop condition is generated.

2) Next byte transfer and start condition generation.

When "0" is written to the INT bit and "1" is written to the SCC bit, the SCC bit takes priority and a start condition is generated.

3) Start condition generation and stop condition generation

It is prohibited to simultaneously write "1" to the SCC bit and "0" to the MSS bit.

18.3.3 Clock Control Register (ICCR)

The clock control register (ICCR) has the following functions.

- **Enabling I²C interface operation**
- **Setting the serial clock frequency**

■ Register Configuration

Clock control register

	7	6	5	4	3	2	1	0	⇐ Bit no.
Address : 000097H	–	–	EN	CS4	CS3	CS2	CS1	CS0	ICCR
Read/write ⇒	(–)	(–)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(–)	(–)	(0)	(X)	(X)	(X)	(X)	(X)	

■ Bit Description

[bit 7], [bit 6] Not used.

[bit 5] EN (ENable)

This bit enables I²C interface operation.

0	Operation disabled
1	Operation enabled

When this bit is set to "0" all bits in the BSR register and BCR register (except the BER, BEIE bits) are cleared.

[bit 4] - [bit 0] CS4-0 (Clock Period Select 4-0)

These bits select the serial clock frequency. The shift clock frequency fsck is determined by the following formula.

$$fsck = \frac{\phi}{m \times n + 4}$$ ϕ : peripheral clock (CLKP)

Note: The addition of 4 cycles is the minimum overhead needed to check whether the SCL pin output level has changed. If the SCL pin startup delay is large, or the slave device has extended the clock cycle, the resulting value will be increased.

The values m and n for the CS4-0 bits are determined as shown in Table 18.3.4a.

Table 18.3.4a Serial Clock Frequency Settings

m	CS4	CS3
5	0	0
6	0	1
7	1	0
8	1	1

n	CS2	CS1	CS0
4	0	0	0
8	0	0	1
16	0	1	0
32	0	1	1
64	1	0	0
128	1	0	1
256	1	1	0
512	1	1	1

This I^2C module is compatible to the standard I^2C mode: It can run at frequencies up to 100 kHz.

18.3.5 Address Register (IADR) / Data Register (IDAR)

The address register (IADR) designates slave addresses, and the data register (IDAR) is used in serial data transfer.

■ Address Register (IADR)

Address register	15	14	13	12	11	10	9	8	⇐ Bit no.
Address : 000096H	–	A6	A5	A4	A3	A2	A1	A0	IADR
Read/write ⇒	(–)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(–)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	

[bit 14] - [bit 8] Slave address bits (A6-A0)

This register is used to designate slave addresses. In slave mode, address data is received and compared with the DAR register. If a match is detected, and acknowledge signal is sent to the master device.

■ Data Register (IDAR)

Data register	7	6	5	4	3	2	1	0	⇐ Bit no.
Address : 000099H	D7	D6	D5	D4	D3	D2	D1	D0	IDAR
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	

[bit 7] - [bit 0] Data bits (D7-D0)

The data register is used in serial data transfer, and transfers data MSB-first. During data receiving (TRX=0), the data output value is "1."

This register is double buffered on the write side, so that when the bus is in use (BB=1), write data can be loaded to the register for serial transfer as each byte is transferred. When read, the serial transfer register is read directly, therefore receiving data values in this register are valid only when the INT bit is set.

■ Disable Register (IDBL)

Disable register	7	6	5	4	3	2	1	0	⇐ Bit no.
Address : 00009BH	---	---	---	---	---	---	---	DBL	IDBL
Read/write ⇒	(-)	(-)	(-)	(-)	(-)	(-)	(-)	(R/W)	
Default value⇒	(-)	(-)	(-)	(-)	(-)	(-)	(-)	(0)	

[bit 0] DBL

This bit is used to control the clock for the I^2C module.

When "1" is written to this bit the module is disabled. When "0" is set, a clock is supplied to the I^2C module.

This bit is initialized to"0" upon reset. This bit is readbale and writable.

18.4 I²C INTERFACE OPERATION

The I²C bus executes communication using two bi-directional bus lines, the serial data line (SDA) and serial clock line (SCL). The I²C interface has two open-drain I/O pins (SDA, SCL) corresponding to these lines, enabling wired logic applications.

■ I²C Interface Operation

- Start Conditions

  With the bus in open status (BB=0, MSS=0), writing "1" to the MSS bit places the I²C interface in master mode and at the same time generates a start condition. In master mode even when the bus is in use (BB=1), another start condition can be generated by writing "1" to the SCC bit.

  Start conditions can be generated under the following 2 conditions:

  - Writing "1" to the MSS bit when the bus is not in use (MSS=0, BB=0, INT=0, AL=0).
  - Writing "1" to the SCC bit when in bus master mode and interrupt status (MSS=1, BB=1, INT=1, AL=0).

  Writing "1" to the MSS bit while another system is using the bus (in idle mode) causes the AL bit to be set to "1." Writing "1" to the MSS bit or SCC bit in any other situation has no significance.

- Stop Conditions

  Writing "0" to the MSS bit in master mode (MSS=1) generates a stop condition and places the device in slave mode.

  Stop condition can be generated under the following condition.

  - Writing "0" to the MSS bit when in bus master mode and interrupt status (MSS=1, BB=1, INT=1, AL=0).

  Writing "0" to the MSS bit in any other situation has no significance.

- Addressing

  In master mode, after a start condition is generated the BB and TRX bits are set to "1" and the contents of the IDAR register are output in MSB order. After address data is sent and an acknowledge signal received from the slave device, bit 0 of the sent data (bit 0 of the IDAR register after sending) is inverted and stored in the TRX bit.

  In slave mode, after a start condition is generated the BB and TRX bits are set to "1" and data sent from the master device is received into the IDAR register. After address data is received, the contents of the IDAR register and IADR register are compared. If a match results, the AAS bit is set to "1" and an acknowledge signal is sent to the master. Then bit 0 of the received data (bit 0 of the IDAR register after receiving) is stored in the TRX bit.

- Arbitration

  During sending in master mode, if another master device is sending data at the same time, arbitration is performed. If a device is sending the data value "1" and the data on the SDA line has an "L" level value, the device is considered to have lost arbitration, and the AL bit is set to "1." Also, the AL bit is set to "1" if the bus is in use as just described and the device attempts to generate a start condition. Setting the AL bit to "1" clears both the MSS and TRX bit to "0" and places the device in slave mode.

- Acknowledgment

  Acknowledgment signals are sent from the receiving side to the sending side. The ACK bit can be used to select whether to send an acknowledgment when data is received. When data is sent acknowledge signals from the receiver are stored in the LRB bit.

  When sending in slave mode, if no acknowledgment is received from the receiving master the TRX bit is set to "0" and the device goes to slave receiving mode. This enables the master to generate a stop condition as soon as the slave has released the SCL line.

- Bus Errors

  Bus errors are assumed in the following situations, and the I^2C interface goes into pause mode.

  - Detection of a fundamental rule violation on the I^2C bus during data transfer (including acknowledgment signals).
  - Detection of a stop condition while in master mode.
  - Detection of a fundamental rule violation on the I^2C bus during bus idle mode.

CHAPTER 19 400 kHz I²C INTERFACE

This section describes the functions and operation of the fast I²C interface.

19.1 I²C INTERFACE OVERVIEW

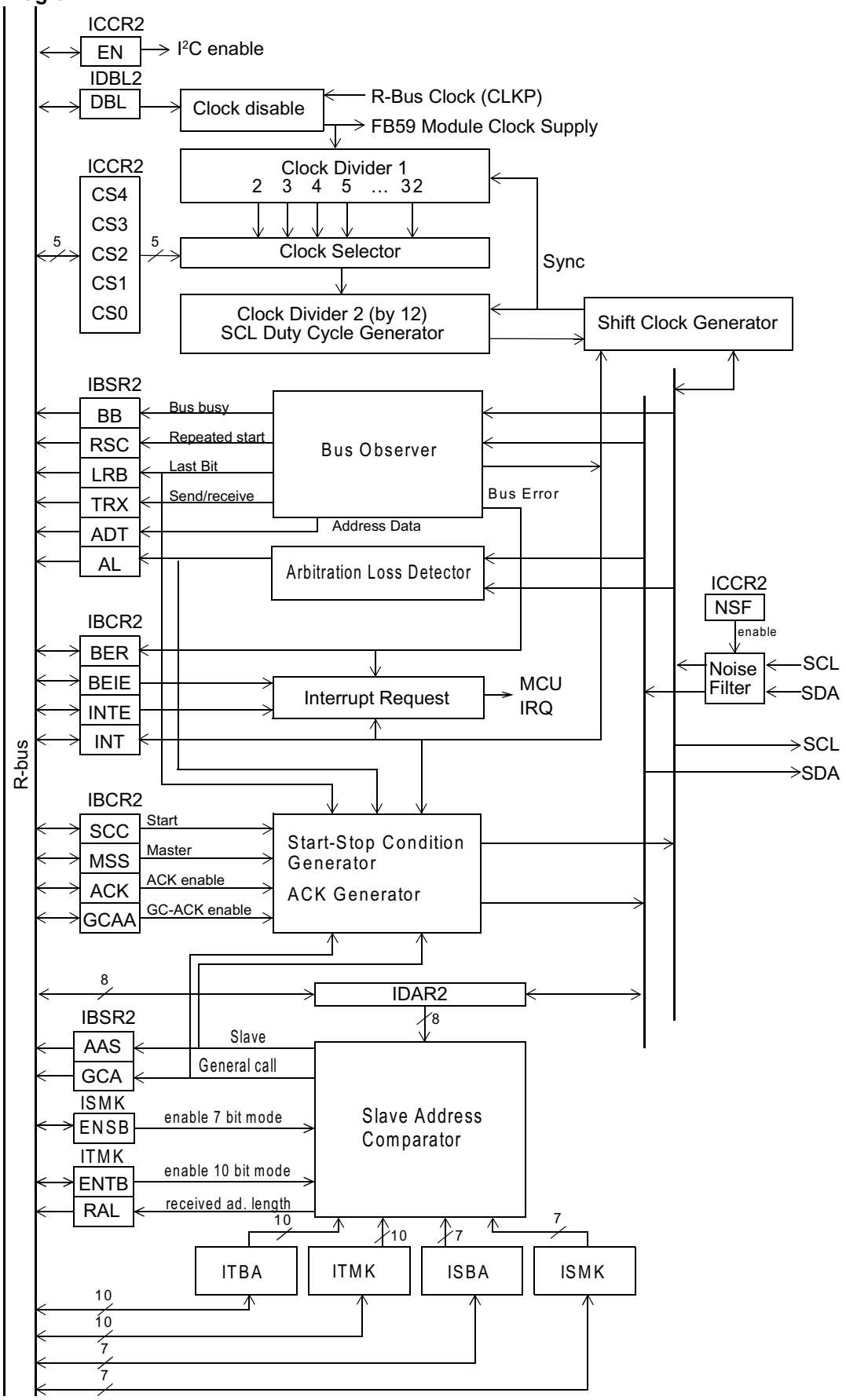
The I²C interface is a serial I/O port supporting the Inter IC bus, operating as a master/ slave device on the I²C bus.

■ Features

- Master/slave transmitting and receiving functions
- Arbitration function
- Clock synchronization function
- General call addressing support
- Transfer direction detection function
- Repeated start condition generation and detection function
- Bus error detection function
- 7 bit addressing as master and slave
- 10 bit addressing as master and slave
- Possibility to give the interface a seven *and* a ten bit slave address
- Acknowledging upon slave address reception can be disabled (Master-only operation)
- Address masking to give interface several slave addresses (in 7 and 10 bit mode)
- Up to 400 kBit transfer rate
- Possibility to use built-in noise filters for SDA and SCL
- Can receive data at 400 kBit if R-Bus-Clock is higher than 6 MHz regardless of prescaler setting
- Can generate MCU interrupts on transmission and bus error events
- Supports being slowed down by a slave on bit and byte level

The I²C interface does not support SCL clock stretching on bit level since it can receive the full 400 kBit datarate if the R-Bus-Clock (CLKP) is higher than 6 MHz regardless of the prescaler setting. However, clock stretching on byte level is performed since SCL is pulled low during an interrupt (INT='1' in IBCR2 register).

■ Block Diagram



19.2 I²C INTERFACE REGISTERS

This section describes the function of the I²C interface registers in detail.

Bus Control Register (IBCR2)

Bus control register	15	14	13	12	11	10	9	8	⇐ Bit no.
Address : 000184H	BER	BEIE	SCC	MSS	ACK	GCAA	INTE	INT	IBCR2
Read/write ⇒	(R/W)	(R/W)	(W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

Bus Status Register (IBSR2)

Bus status register	7	6	5	4	3	2	1	0	⇐ Bit no.
Address : 000185H	BB	RSC	AL	LRB	TRX	AAS	GCA	ADT	IBSR2
Read/write ⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

Ten Bit slave Address register (ITBA)

Ten Bit Address high byte	15	14	13	12	11	10	9	8	⇐ Bit no.
Address : 000186H	---	---	---	---	---	---	TA9	TA8	ITBAH
Read/write ⇒	(-)	(-)	(-)	(-)	(-)	(-)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

Ten Bit Address low byte	7	6	5	4	3	2	1	0	⇐ Bit no.
Address : 000187H	TA7	TA6	TA5	TA4	TA3	TA2	TA1	TA0	ITBAL
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

Ten bit slave address MasK register (ITMK)

Ten Bit Address Mask high byte	15	14	13	12	11	10	9	8	⇐ Bit no.
Address : 000188H	ENTB	RAL	---	---	---	---	TM9	TM8	ITMKH
Read/write ⇒	(R/W)	(R)	(-)	(-)	(-)	(-)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(1)	(1)	(1)	(1)	(1)	(1)	

Ten Bit Address Mask low byte	7	6	5	4	3	2	1	0	⇐ Bit no.
Address : 000189H	TM7	TM6	TM5	TM4	TM3	TM2	TM1	TM0	ITMKL
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	

Seven Bit slave Address register (ISBA)

Seven Bit Address register	7	6	5	4	3	2	1	0	⇐ Bit no.
Address : 00018BH	---	SA6	SA5	SA4	SA3	SA2	SA1	SA0	ISBA
Read/write ⇒	(-)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

Seven bit slave address MasK register (ISMK)

Seven Bit Address Mask register	15	14	13	12	11	10	9	8	⇐ Bit no.
Address : 00018AH	ENSB	SM6	SM5	SM4	SM3	SM2	SM1	SM0	ISMK
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	

Data Register (IDARH)

Data register high byte	15	14	13	12	11	10	9	8	⇐ Bit no.
Address : 00018CH	---	---	---	---	---	---	---	---	IDARH
Read/write ⇒	(-)	(-)	(-)	(-)	(-)	(-)	(-)	(-)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

Data register	7	6	5	4	3	2	1	0	⇐ Bit no.
Address : 00018DH	D7	D6	D5	D4	D3	D2	D1	D0	IDAR2
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

Clock control register (ICCR2)

Clock Control register	15	14	13	12	11	10	9	8	⇐ Bit no.
Address : 00018EH	---	NSF	EN	CS4	CS3	CS2	CS1	CS0	ICCR2
Read/write ⇒	(-)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(1)	(1)	(1)	(1)	(1)	

Clock Disable Register (IDBL2)

Clock Disable register	7	6	5	4	3	2	1	0	⇐ Bit no.
Address : 00018FH	---	---	---	---	---	---	---	DBL	IDBL2
Read/write ⇒	(-)	(-)	(-)	(-)	(-)	(-)	(-)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

19.2.1 Bus Status Register (IBSR2)

The bus status register (IBSR2) has the following functions:

- **Bus busy detection**
- **Repeated start condition detection**
- **Arbitration loss detection**
- **Acknowledge detection**
- **Data transfer direction indication**
- **Addressing as slave detection**
- **General call address detection**
- **Address data transfer detection**

This register is read-only, all bits are controlled by the hardware. All bits are cleared if the interface is not enabled (EN='0' in ICCR2).

Bus status register — Address : 000185H — ⇐ Bit no. — IBSR2

	7	6	5	4	3	2	1	0
	BB	RSC	AL	LRB	TRX	AAS	GCA	ADT
Read/write ⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

[bit 7] BB (Bus Busy)

This bit indicates the status of the I²C bus.

0	Stop condition detected (bus idle).
1	Start condition detected (bus in use).

This bit is set to '1' if a start condition is detected. It is reset upon a stop condition.

[bit 6] RSC (Repeated Start Condition)

This bit indicates detection of a repeated start condition.

0	Repeated start condition not detected.
1	Bus in use, repeated start condition detected.

This bit is cleared at the end of an address data transfer (ADT='0') or detection of a stop condition.

[bit 5] AL (Arbitration Loss)

This bit indicates an arbitration loss.

0	No arbitration loss detected.
1	Arbitration loss occurred during master sending.

This bit is cleared by writing ‘0’ to the INT bit or by writing ‘1’ to the MSS bit in the IBCR2 register.

An arbitration loss occurs if:

the data sent does not match the data read on the SDA line at the rising SCL edge

a repeated start condition is generated by another master in the first bit of a data byte

the interface could not generate a start or stop condition because another slave pulled the SCL line low before

[bit 4] LRB (Last Received Bit)

This bit is used to store the acknowledge message from the receiving side at the transmitter side.

0	Receiver acknowledged.
1	Receiver did not acknowledge.

It is changed by the hardware upon reception of bit 9 (acknowledge bit) and is also cleared by a start or stop condition.

[bit 3] TRX (Transferring data)

This bit indicates data sending operation during data transfer.

0	Not transmitting data.
1	Transmitting data.

It is set to ‘1’:

- if a start condition was generated in master mode at end of a first byte transfer and read access as slave or sending data as master

It is set to ‘0’ if:

- the bus is idle (BB=‘0’ in IBCR2)
- an arbitration loss occured
- a ‘1’ is written to the SCC bit during master interrupt (MSS=‘1’ and INT=‘1’)
- the MSS bit being cleared during master interrupt (MSS=‘1’ and INT=‘1’)
- the interface is in slave mode and the last transferred byte was not acknowledged
- the interface is in slave mode and it is receiving data
- the interface is in master mode and is reading data from a slave

[bit 2] AAS (Addressed As Slave)

This bit indicates detection of a slave addressing.

0	Not addressed as slave.
1	Addressed as slave.

This bit is cleared by a (repeated-) start or stop condition. It is set if the interface detects its seven and/or ten bit slave address.

[bit 1] GCA (General Call Address)

This bit indicates detection of a general call address (0x00).

0	General call address not received as slave.
1	General call address received as slave.

This bit is cleared by a (repeated-) start or stop condition.

[bit 0] ADT (Address Data Transfer)

This bit indicates the detection of an address data transfer.

0	Incoming data is not address data (or bus is not in use).
1	Incoming data is address data.

This bit is set to ‘1’ by a start condition. It is cleared after the second byte if a ten bit slave address header with write access is detected, else it is cleared after the first byte.

"After" the first/second byte means:

- a ‘0’ is written to the MSS bit during a master interrupt (MSS=‘1’ and INT=‘1’ in IBCR2)
- a ‘1’ is written to the SCC bit during a master interrupt (MSS=‘1’ and INT=‘1’ in IBCR2)
- the INT bit is being cleared
- the beginning of every byte transfer if the interface is not involved in the current transfer as master or slave

19.2.2 Bus Control Register (IBCR2)

The bus control register (IBCR2) has the following functions:

- **Interrupt enabling flags**
- **Interrupt generation flag**
- **Bus error detection flag**
- **Repeated start condition generation**
- **Master / slave mode selection**
- **General call acknowledge generation enabling**
- **Data byte acknowledge generation enabling**

Write access to this register should only occur while the INT='1' or if a transfer is to be started. The user should not write to this register during an ongoing transfer since changes to the ACK or GCAA bits could result in bus errors. All bits in this register except the BER and the BEIE bit are cleared if the interface is not enabled (EN='0' in ICCR2).

Bus control register ⇐ Bit no.

Address : 000184H IBCR2

15	14	13	12	11	10	9	8
BER	BEIE	SCC	MSS	ACK	GCAA	INTE	INT
(R/W)	(R/W)	(W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

Read/write ⇒
Default value⇒

[bit 15] BER (Bus ERror)

This bit is the bus error interrupt flag. It is set by the hardware and cleared by the user. It always reads '1' in a Read-Modify-Write access.

(Write access)

0	Clear bus error interrupt flag.
1	No effect.

(Read access)

0	No bus error detected.
1	One of the error conditions described below detected.

When this bit is set, the EN bit in the ICCR2 register is cleared, the I^2C interface goes to pause status, data transfer is interrupted and all bits in the IBSR2 and the IBCR2 registers except BER and BEIE are cleared. *The BER bit must be cleared before the interface may be reenabled.*

This bit is set to '1' if:

- start or stop conditions are detected at wrong places: during an address data transfer or during the transfer of the bits two to nine (acknowledge bit)
- a ten bit address header with read access is received before a ten bit write access

- a stop condition is detected while the interface is in master mode

The detection of the first two of the above conditions is enabled after the reception of the first stop condition to prevent false bus error reports if the interface is being enabled during an ongoing transfer.

[bit 14] BEIE (Bus Error Interrupt Enable)

This bit enables the bus error interrupt. It can only be changed by the user.

0	Bus error interrupt disabled.
1	Bus error interrupt enabled.

Setting this bit to '1' enables MCU interrupt generation when the BER bit is set to '1'.

[bit 13] SCC (Start Condition Continue)

This bit is used to generate a repeated start condition. It is write only - it always reads '0'.

0	No effect.
1	Generate repeated start condition during master transfer.

A repeated start condition is generated if a '1' is written to this bit while an interrupt in master mode (MSS='1' and INT='1') and the INT bit is cleared automatically.

[bit 12] MSS (Master Slave Select)

This is the master/slave mode selection bit. It can only be set by the user, but it can be cleared by the user and the hardware.

0	Go to slave mode.
1	Go to master mode, generate start condition and send address data byte in IDAR2 register.

It is cleared if an arbitration loss event occurs during master sending.

If a '0' is written to it during a master interrupt (MSS='1' and INT='1'), the INT bit is cleared automatically, a stop condition will be generated and the data transfer ends. Note that the MSS bit is reset immediately, the generation of the stop condition can be checked by polling the BB bit in the IBSR2 register.

If a '1' is written to it while the bus is idle (MSS='0' and BB='0'), a start condition is generated and the contents of the IDAR2 register (which should be address data) is sent.

If a '1' is written to the MSS bit while the bus is in use (BB='1' and TRX='0' in IBSR2; MSS='0' in IBCR2), the interface waits until the bus is free and then starts sending.

If the interface is addressed as slave with write access (data *reception*) in the meantime, it will start sending after the transfer ended and the bus is free again. If the interface is *sending* data as slave in the meantime (AAS='1' and TRX='1' in IBSR2), it will not start sending data if the bus of free again. It is important to check whether the interface was addressed as slave (AAS='1' in IBSR2), sent the data byte successfully (MSS='1' in IBCR2) or failed to send the data byte (AL='1' in IBSR2) at the next interrupt!

[bit 11] ACK (ACKnowledge)

This is the acknowledge generation on *data* byte reception enable bit. It can only be changed by the user.

0	The interface will not acknowledge on data byte reception.
1	The interface will acknowledge on data byte reception.

This bit is not valid when receiving *address* bytes in slave mode - if the interface detects its 7 or 10 bit slave address, it will acknowledge if the corresponding enable bit (ENTB in ITMK or ENSB in ISMK) is set.

Write access to this bit should occur during an interrupt (INT='1') or if the bus is idle (BB='0' in the IBSR2 register) write access to this bit is only possible if the interface is enabled (EN='1' in ICCR2) and if there is no bus error (BER='0' in IBCR2) only.

[bit 10] GCAA (General Call Address Acknowledge)

This bit enables acknowledge generation when a general call address is received. It can only be changed by the user.

0	The interface will not acknowledge on general call address byte reception.
1	The interface will acknowledge on general call address byte reception.

Write access to this bit should occur during an interrupt (INT='1') or if the bus is idle (BB='0' in IBSR2 register) write access to this bit is only possible if the interface is enabled (EN='1' in ICCR2) and if there is no bus error (BER='0' in IBCR2) only.

[bit 9] INTE (INTerrupt Enable)

This bit enables the MCU interrupt generation. It can only be changed by the user.

0	Interrupt disabled.
1	Interrupt enabled.

Setting this bit to '1' enables MCU interrupt generation when the INT bit is set to '1' (by the hardware).

[bit 8]: INT (INTerrupt)

This bit is the transfer end interrupt request flag. It is changed by the hardware and can be cleared by the user. It always reads '1' in a Read-Modify-Write access.

(Write access)

0	Clear transfer end interrupt request flag.
1	No effect.

(Read access)

0	Transfer not ended or not involved in current transfer or bus is idle.
1	Set *at the end of a 1-byte* data *transfer or reception* including the acknowledge bit under the following conditions: • Device is bus master. • Device is addressed as slave. • General call address received. • Arbitration loss occurred. Set *at the end of an address data reception* (after first byte if seven bit address received, after second byte if ten bit address received) including the acknowledge bit if the device is addressed as slave.

While this bit is '1' the SCL line will hold an 'L' level signal. Writing '0' to this bit clears the setting, releases the SCL line, and executes transfer of the next byte or a repeated start or stop condition is generated. Additionally, this bit is cleared if a '1' is written to the SCC bit or the MSS bit is being cleared.

■ SCC, MSS And INT Bit Competition

Simultaneously writing to the SCC, MSS and INT bits causes a competition to transfer the next byte, to generate a repeated start condition or to generate a stop condition. In these cases the order of priority is as follows:

Next byte transfer and stop condition generation.
When '0' is written to the INT bit and '0' is written to the MSS bit, the MSS bit takes priority and a stop condition is generated.

Next byte transfer and start condition generation.
When '0' is written to the INT bit and '1' is written to the SCC bit, the SCC bit takes priority. A repeated start condition is generated and the contents of the IDAR2 register is sent.

Repeated start condition generation and stop condition generation.
When a '1' is written to the SCC bit and '0' to the MSS bit, the MSS bit clearing takes priority. A stop condition is generated and the interface enters slave mode.

19.2.3 Ten Bit Slave Address Register (ITBA)

This register (ITBAH / ITBAL) designates the ten bit slave address.

Write access to this register is only possible if the interface is disabled (EN='0' in ICCR2).

Ten Bit Address high byte	15	14	13	12	11	10	9	8	⇐ Bit no.
Address : 000186H	---	---	---	---	---	---	TA9	TA8	ITBAH
Read/write ⇒	(-)	(-)	(-)	(-)	(-)	(-)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

Ten Bit Address low byte	7	6	5	4	3	2	1	0	⇐ Bit no.
Address : 000187H	TA7	TA6	TA5	TA4	TA3	TA2	TA1	TA0	ITBAL
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

[bit 15] - [bit 10] Not used.

These bits always read '0'.

[bit 9] - [bit 0] TBA - Ten Bit slave Address (TA9-TA0)

When address data is received in slave mode, it is compared to the ITBA register if the ten bit address is enabled (ENTB='1' in the ITMK register). An acknowledge is sent to the master after reception of a ten bit address header with write access[1]. Then, the second incoming byte is compared to the TBAL register. If a match is detected, an acknowledge signal is sent to the master device and the AAS bit is set.

Additionally, the interface acknowledges upon the the reception of a ten bit header with read access[2] after a *repeated* start conditon.

All bits of the slave address may be masked using the ITMK register. The received ten bit slave address is written back to the ITBA register, it is only valid while the AAS bit in the IBSR2 register is '1'.

1. Note: a ten bit header (write access) consists of the following bit sequence: 11110, TA9, TA8, 0.
2. Note: a ten bit header (read access) consists of the following bit sequence: 11110, TA9, TA8, 1.

19.2.4 Ten Bit Address Mask Register (ITMK)

This register contains the ten bit slave address mask and the ten bit slave address enable bit.

Ten Bit Address Mask high byte	15	14	13	12	11	10	9	8	⇐ Bit no.
Address : 000188H	ENTB	RAL	---	---	---	---	TM9	TM8	ITMKH
Read/write ⇒	(R/W)	(R)	(-)	(-)	(-)	(-)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(1)	(1)	(1)	(1)	(1)	(1)	

Ten Bit Address Mask low byte	7	6	5	4	3	2	1	0	⇐ Bit no.
Address : 000189H	TM7	TM6	TM5	TM4	TM3	TM2	TM1	TM0	ITMKL
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	

[bit 15] ENTB - EnaBle Ten Bit slave address

This bit enables the ten bit slave address (and the acknowleding upon its reception). Write access to this bit is only possible if the interface is disabled (EN='0' in ICCR2).

0	Ten bit slave address disabled.
1	Ten bit slave address enabled.

[bit 14] RAL - Received slave Address Length

This bit indicates whether the interface was addressed as a seven or ten bit slave. It is read-only.

0	Addressed as seven bit slave.
1	Addressed as ten bit slave.

This bit can be used to determine whether the interface was addressed as a seven or ten bit slave if both slave addresses are enabled (ENTB='1' and ENSB='1'). Its contents is only valid if the AAS bit in the IBSR2 register is '1'. This bit is also reset if the interface is disabled (EN='0' in ICCR2).

[bit 13] - [bit 10] Not used.

These bits always read '1'.

[bit 9] - [bit 0] TMK - Ten bit slave address MasK (TM9-TM0).

This register is used to mask the ten bit slave address of the interface. Write access to these bits is only possible if the interface is disabled (EN='0' in ICCR2).

0	Bit is not used in slave address comparision.
1	Bit is used in slave address comparision.

This can be used to make the interface acknowledge on multiple ten bit slave addresses. Only the bits set to '1' in this register are used in the ten bit slave address comparision. The received slave address is written back to the ITBA register and thus may be determined by reading the ITBA register if the AAS bit in the IBSR2 register is '1'.

Note: If the address mask is changed after the interface had been enabled, the slave address should also be set again since it could have been overwritten by a previously received slave address.

19.2.5 Seven Bit Slave Address Register (ISBA)

This register designates the seven bit slave address.

Write access to this register is only possible if the interface is disabled (EN='0' in ICCR2).

Seven Bit Address register — Address : 00018BH — ISBA

	7	6	5	4	3	2	1	0	⇐ Bit no.
	---	SA6	SA5	SA4	SA3	SA2	SA1	SA0	
Read/write ⇒	(-)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

[bit 7] Not used.

This bit always reads '0'.

[bit 6] - [bit 0] Seven Bit slave Address (SA6-SA0)

When address data is received in slave mode, it is compared to the ISBA register if the seven bit address is enabled (ENSB='1' in the ISMK register). If a match is detected, an acknowledge signal is sent to the master device and the AAS bit is set.

All bits of the slave address may be masked using the ISMK register. The received seven bit slave address is written back to the ISBA register, it is only valid while the AAS bit in the IBSR2 register is '1'.

The interface does not compare the contents of this register to the incoming data if a ten bit header or a general call is received.

19.2.6 Seven Bit Slave Address Mask Register (ISMK)

This register contains the seven bit slave address mask and the seven bit mode enable bit. Write access to this register is only possible if the interface is disabled (EN='0' in ICCR2).

Seven Bit Address Mask register	15	14	13	12	11	10	9	8	⇐ Bit no.
Address : 00018AH	ENSB	SM6	SM5	SM4	SM3	SM2	SM1	SM0	ISMK
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	

[bit 15] ENSB - EnaBle Seven Bit slave address

This bit enables the seven bit slave address (and the acknowleding upon its reception).

0	Seven bit slave address disabled.
1	Seven bit slave address enabled.

[bit 14] - [bit 8] SMK - Seven bit slave address MasK (SM6-SM0)

This register is used to mask the seven bit slave address of the interface.

0	Bit is not used in slave address comparision.
1	Bit is used in slave address comparision.

This can be used to make the interface acknowledge on multiple seven bit slave addresses. Only the bits set to '1' in this register are used in the seven bit slave address comparision. The received slave address is written back to the ISBA register and may thus may be determined by reading the ISBA register if the AAS bit in the IBSR2 register is '1'.

Note: If the address mask is changed after the interface had been enabled, the slave address should also be set again since it could have been overwritten by a previously received slave address.

19.2.7 Data Register (IDAR2)

Data register high byte	15	14	13	12	11	10	9	8	⇐ Bit no.
Address : 00018CH	---	---	---	---	---	---	---	---	IDARH
Read/write ⇒	(-)	(-)	(-)	(-)	(-)	(-)	(-)	(-)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

Data register	7	6	5	4	3	2	1	0	⇐ Bit no.
Address : 00018DH	D7	D6	D5	D4	D3	D2	D1	D0	IDAR2
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

[bit 15] - [bit 8] Not used.

These bits always read ‘0’.

[bit 7] - [bit 0] Data bits (D7-D0)

The data register is used in serial data transfer, and transfers data MSB-first. This register is double buffered on the write side, so that when the bus is in use (BB=‘1’), write data can be loaded to the register for serial transfer. The data byte is loaded into the internal transfer register if the INT bit in the IBCR2 register is being cleared or the bus is idle (BB=‘0’ in IBSR2). In a read access, the internal register is read directly, therefore *received data values in this register are only valid if INT=‘1’ in the IBCR2 register*.

19.2.8 Clock Control Register (ICCR2)

The clock control register (ICCR2) has the following functions:

- **Enable testmode**
- **Enable IO pad noise filters**
- **Enable I²C interface operation**
- **Setting the serial clock frequency**

Clock Control register — Address : 00018EH — ICCR2

	15	14	13	12	11	10	9	8	⇐ Bit no.
	---	NSF	EN	CS4	CS3	CS2	CS1	CS0	
Read/write ⇒	(-)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(1)	(1)	(1)	(1)	(1)	

[bit 15] Not used.

This bit always reads ‘0’.

[bit 14] IO pad NoiSe Filter enable.

This bit enables the noise filters built into the SDA and SCL IO pads.
The noise filter will suppress single spikes with a pulse width of 0 ns (minimum) and between 1 and 1.5 cycles of R-bus (maximum). The maximum depends on the phase relationship between I²C signals (SDA, ACL) and R-bus clock.
It should be set to ‘1’ if the interface is transmitting or receiving at datarates above 100 kBit.

[bit 13] EN (ENable)

This bit enables the I²C interface operation. It can only be set by the user but may be cleared by the user and the hardware.

0	Interface disabled.
1	Interface enabled.

When this bit is set to ‘0’ all bits in the IBSR2 register and IBCR2 register (except the BER and BEIE bits) are cleared and the module is disabled and the I²C lines are left open. It is cleared by the hardware if a bus error occurs (BER=‘1’ in IBCR2).

Warning: The interface *immediately* stops transmitting or receiving if is it is being disabled. This might leave the I²C bus in an undesired state!

[bit 12] - [bit 8] CS4-0 (Clock preScaler)

These bits select the serial bitrate. They can only be changed if the interface is disabled (EN='0') or the EN bit is being cleared in the same write access.

It is determined by the following formula:

$$\text{Bitrate} = \frac{\phi}{n*12 + 18}$$ **Noise filter disabled**

n>0; ϕ : R-Bus clock CLKP (set by DIVR0 register)

$$\text{Bitrate} = \frac{\phi}{n*12 + 19\ (+1)}$$ **Noise filter enabled**

n>0; ϕ : R-Bus clock CLKP (set by DIVR0 register)
(+1): Unaccurancy caused by noise filter operation

Note: Because of the noise filter (depending on relationship between external signal and internal clock it will cause different delays) the divider in the second formular can vary between (12n + 19) and (12n + 20).

■ **Prescaler settings:**

Table 19.2.8a I2C Prescaler Settings

n	CS4	CS3	CS2	CS1	CS0
1	0	0	0	0	1
2	0	0	0	1	0
3	0	0	0	1	1
...					
31	1	1	1	1	1

Do not use n=0 prescaler setting, it violates SDA/SCL timings!

The table below shows SCL frequency measurement results for the most common R-bus clock settings and the recommended related pre-scaler settings for 100 Kbit and 400 Kbit operation.

Table 19.2.8b I2C Sending Bitrates

R-Bus Clock (CLKP) [MHz]	100 kBit (Noise filter disabled)		400 kBit (Noise filter enabled)	
	n	Bitrate [kBit]	n	Bitrate [kBit]
32			5	387.5
24	19	97.5	4	352.5
16	12	98	2	372
8	6	89	1	266.5

Note: It should be noted that the measured values have been determined by examining the last 8 cycles of a transfer. This was done because the first cycle of all address or data transfers is longer than the other cycles. To be more precise: In case of an address transfer this first cycle is 3 prescaler periods longer than the other cycles, in case of a data transfer it is 4 prescaler periods longer (see figure below).

■ **SCL Waveforms**

Figure 19.2.8a SCL Waveforms

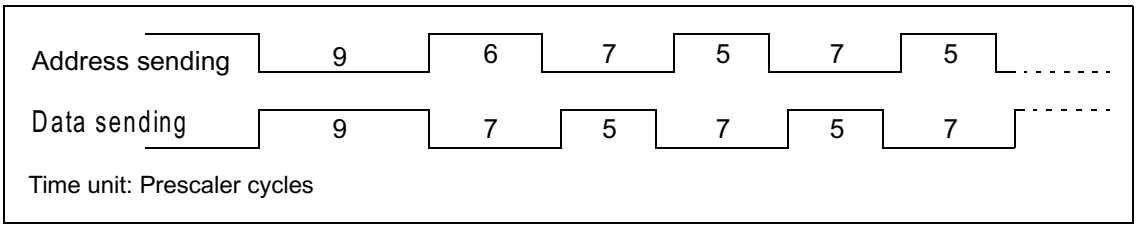


Figure 19.2.8a shows the SCL waveform for sending of address and data bits. The timings given in the figure are prescaler periods (e.g. '9' means 9 times the prescaler count based on the R-Bus clock). The timings in the figure are only valid if no other device on the I2C bus influences the SCL timing.

19.2.9 Clock Disable Register (IDBL2)

Clock disable register	7	6	5	4	3	2	1	0	⇐ Bit no.
Address : 00018FH	---	---	---	---	---	---	---	DBL	IDBL2
Read/write ⇒	(-)	(-)	(-)	(-)	(-)	(-)	(-)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

[bit 0] DBL Clock Disable

This bit is used to control the clock supply for the I²C module. When this bit is '1', clock supply to the module is disabled (e.g. for power saving) and the I²C lines are left open. This bit is initialized to '0' upon reset.

Warning: The interface *immediately* stops transmitting or receiving if the clock supply is it is being disabled by this bit. This might leave the I²C bus in an undesired state!

If the interface is disabled by the DBL bit (DBL='1'), read access to any I²C register except IDBL2 will return undefined values and write access to any I²C register except IDBL2 has no effect.

19.3 I²C INTERFACE OPERATION

The I²C bus executes communication using two bi-directional bus lines, the serial data line (SDA) and serial clock line (SCL). The I²C interface has two open-drain I/O pins (SDA/SCL) corresponding to these lines, enabling wired logic applications.

■ Start Conditions

When the bus is free (BB='0' in IBSR2, MSS='0' in IBCR2), writing '1' to the MSS bit places the I²C interface in master mode and generates a start condition.
If a '1' is written to it while the bus is idle (MSS='0' and BB='0'), a start condition is generated and the contents of the IDAR2 register (which should be address data) is sent.
Repeated start conditions can be generated by writing '1' to the SCC bit when in bus master mode and interrupt status (MSS='1' and INT='1' in IBCR2).

If a '1' is written to the MSS bit while the bus is in use (BB='1' and TRX='0' in IBSR2; MSS='0'and INT='0'in IBCR2), the interface waits until the bus is free and then starts sending.
If the interface is addressed as slave with write access (data reception) in the meantime, it will start sending after the transfer ended and the bus is free again. If the interface is sending data as slave in the meantime, it will not start sending data if the bus of free again. It is important to check whether the interface was addressed as slave (MSS='0' in IBCR2 and AAS='1' in IBSR2), sent the data byte successfully (MSS='1' in IBCR2) or failed to send the data byte (AL='1' in IBSR2) at the next interrupt!

Writing '1' to the MSS bit or SCC bit in any other situation has no significance.

■ Stop Conditions

Writing '0' to the MSS bit in master mode (MSS='1' and INT='1' in IBCR2) generates a stop condition and places the device in slave mode. Writing '0' to the MSS bit in any other situation has no significance.
After clearing the MSS bit, the interface tries to generate a stop condition which might fail if another master pulls the SCL line low before the stop condition has been generated. *This will generate an interrupt after the next byte has been transferred!*

■ Slave Address Detection

In slave mode, after a start condition is generated the BB is set to '1' and data sent from the master device is received into the IDAR2 register.
After the reception of eight bits, the contents of the IDAR2 register is compared to the ISBA register using the bit mask stored in ISMK if the ENSB bit in the ISMK register is '1'. If a match results, the AAS bit is set to '1' and an acknowledge signal is sent to the master. Then bit 0 of the received data (bit 0 of the IDAR2 register) is inverted and stored in the TRX bit.
If the ENTB bit in the ITMK register is '1' and a ten bit address header (11110, TA1, TA0, write access) is detected, the interface sends an acknowledge signal to the master and stores the inverted last data bit in the TRX register. No interrupt is generated. Then, the next transferred byte is compared (using the bit mask stored in ITMK) to the lower byte of the ITBA register. If a match is found, an acknowledge signal is sent to the master, the AAS bit is set and an interrupt is generated.
If the interface was addressed as slave and detectes a repeated start condition, the AAS bit is set after reception of the ten bit address header (11110, TA1, TA0, read access) and an interrupt

is generated.
Since there are seperate registers for the ten and seven bit address and their bitmasks, it is possible to make the interface acknowledge on both addresses by setting the ENSB (in ISMK) and ENTB (in ITMK) bits. The received slave address length (seven or ten bit) may be determined by reading the RAL bit in the ITMK register (this bit is valid if the AAS bit is set only).
It is also possible to give the interface no slave address by setting both bits to '0' if it is only used as a master.
All slave address bits may be masked with their corresponding mask register (ITMK or ISMK).

■ Slave Address Masking

Only the bits set to '1' in the mask registers (ITMK / ISMK) are used for address comparision, all other bits are ignored. The received slave address can be read from the ITBA (if ten bit address received, RAL='1') or ISBA (if seven bit address received, RAL='0') register if the AAS bit in the IBSR2 register is '1'.
If the bitmasks are cleared, the interface can be used as a bus monitor since it will always be addressed as slave. Note that this is not a real bus monitor because it *acknowledges* upon any slave address reception, even if there is no other slave listening.

■ Addressing Slaves

In master mode, after a start condition is generated the BB and TRX bits are set to '1' and the contents of the IDAR2 register is sent in MSB first order. After address data is sent and an acknowledge signal was received from the slave device, bit 0 of the sent data (bit 0 of the IDAR2 register after sending) is inverted and stored in the TRX bit. Acknowledgement by the slave may be checked using the LRB bit in the IBSR2 register. This procedure also applies to a repeated start condition.
In order to address a ten bit slave for write access, two bytes have to be sent. The first one is the ten bit address header which consists of the bitsequence '1 1 1 1 0 A9 A8 0', it is followed by the second byte containing the lower eight bits of the ten bit slave address (A7 - A0).
A ten bit slave is accessed for reading by sending the above byte sequence and generating a repeated start condition (SCC bit in IBCR2) followed by a ten bit address header with read access (1 1 1 1 0 A9 A8 1).
Summary of the address data bytes:

7 bit slave, write access: Start condition - A6 A5 A4 A3 A2 A1 A0 0.

7 bit slave, read access: Start condition - A6 A5 A4 A3 A2 A1 A0 1.

10 bit slave, write access: Start condition - 1 1 1 1 0 A9 A8 0 - A7 A6 A5 A4 A3 A2 A1 A0.

10 bit slave, read access: Start condition - 1 1 1 1 0 A9 A8 1 - A7 A6 A5 A4 A3 A2 A1 A0 - repeated start - 1 1 1 1 0 A9 A8 1.

■ Arbitration

During sending in master mode, if another master device is sending data at the same time, arbitration is performed. If a device is sending the data value '1' and the data on the SDA line has an 'L' level value, the device is considered to have lost arbitration, and the AL bit is set to '1.'
Also, the AL bit is set to '1' if a start conditon is detected at the first bit of a data byte but the interface did not want to generate one or the generation of a start or stop condition failed by some reason.
Arbitration loss detection clears both the MSS and TRX bit and immediately places the device in slave mode so it is able to acknowledge if its own slave address is being sent.

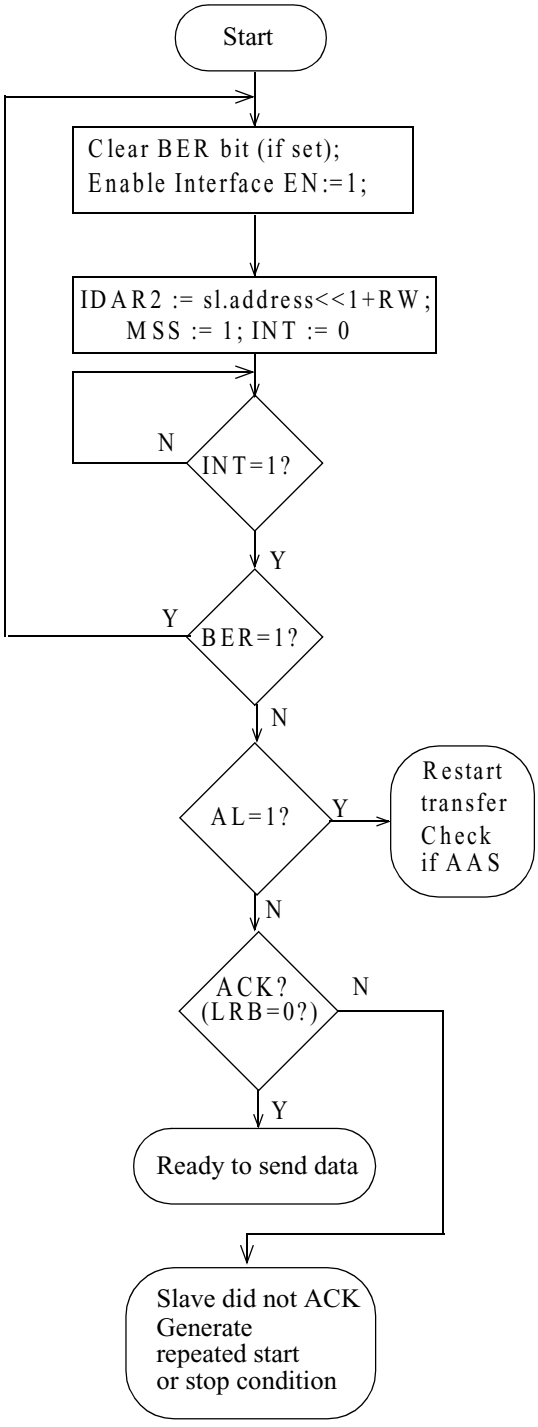
■ Acknowledgement

Acknowledge bits are sent from the receiver to the transmitter. The ACK bit in the IBCR2 register can be used to select whether to send an acknowledgment when data bytes are received.
When data is send in slave mode (read access from another master), if no acknowledgement is received from the master, the TRX bit is set to ‘0’ and the device goes to receiving mode. This enables the master to generate a stop condition as soon as the slave has released the SCL line.
In master mode, acknowledgement by the slave may be checked by reading the LRB bit in the IBSR2 register.

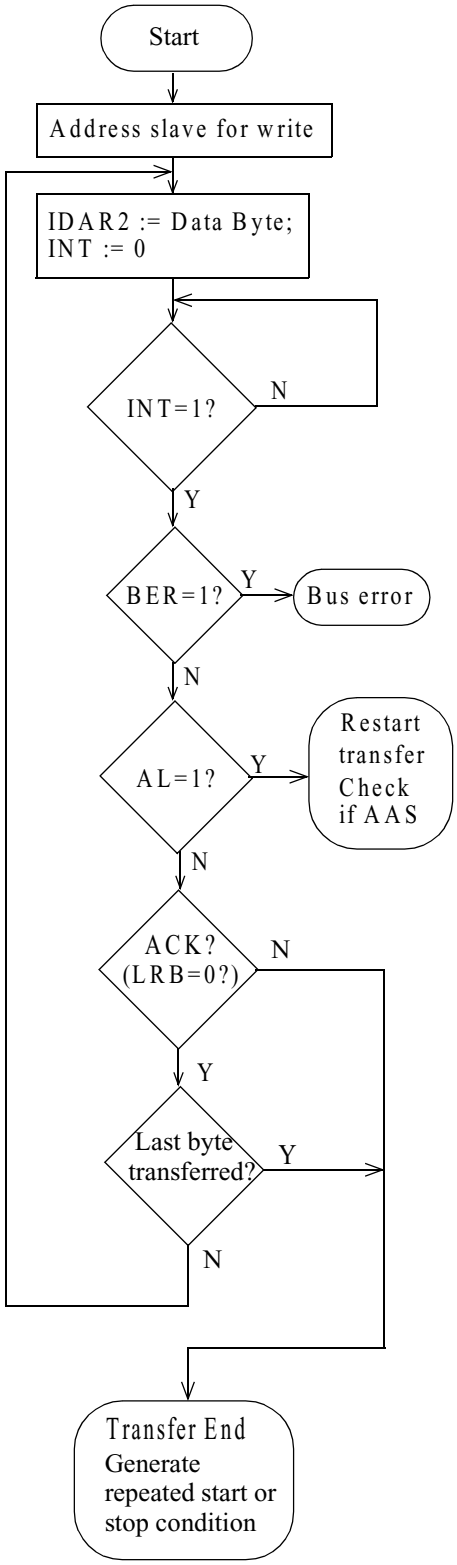
19.4 PROGRAMMING FLOW CHARTS

■ **Example Of Slave Addressing And Sending Data**

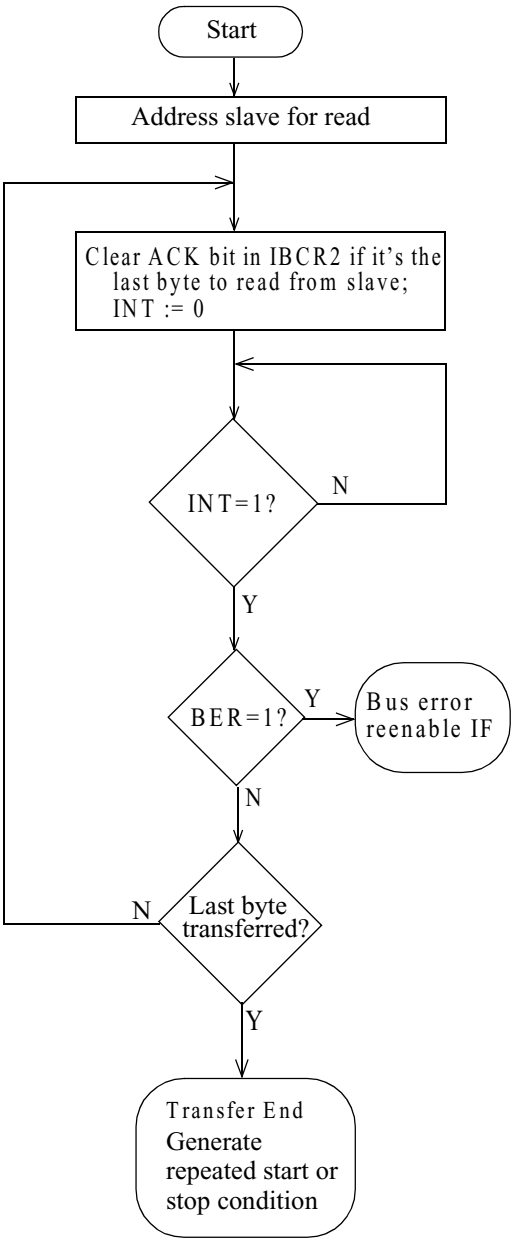
Addressing a 7 bit slave



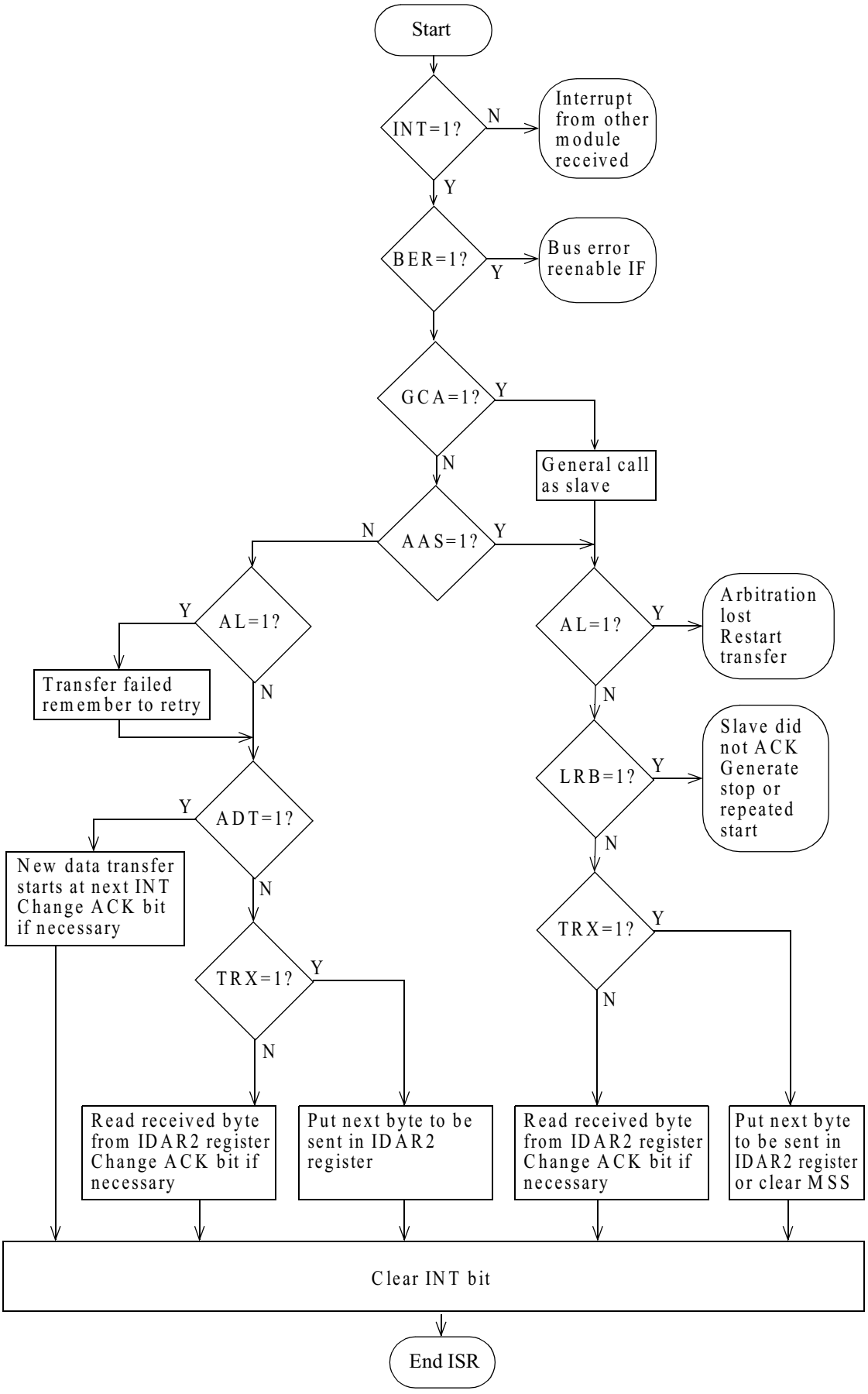
Sending data



■ Example Of Receiving Data



■ Example Of An Interrupt Handler



CHAPTER 20 16-BIT I/O TIMER

The MB91360 Series contains two 16-bit free-running timer modules, two output compare modules, and two input capture modules and supports four input channels and four output channels.

The following sections only describes the 16-bit free-running timer, Output Compare 0/1 and Input Capture 0/1. The remaining modules have the identical functions and the register addresses should be found in the I/O map.

20.1 FUNCTION OVERVIEW

■ **16-bit free-running timer**

The 16-bit free-run timer consists of a 16-bit up counter, control register, and prescaler. The values output from this timer counter are used as the base timer for input capture and output compare.

- Four counter clocks are available.
  Peripheral clock CLKP : ∅/4, ∅/16, ∅/32, ∅/64
- An interrupt can be generated upon a counter overflow or a match with compare register 0.
- The counter value can be initialized to "0000H" upon a reset, software clear, or match with compare register 0.

■ **Output compare (2 channels per one module)**

The output compare module consists of two 16-bit compare registers, compare output latch, and control register.

When the 16-bit free-running timer value matches the compare register value, the output level is reversed and an interrupt is issued.

- The two compare registers can be used independently.
  Output pins and interrupt flags corresponding to compare registers
- Output pins can be controlled based on pairs of the two compare registers.
  Output pins can be reversed by using the two compare registers.
- Initial values for output pins can be set.
- Interrupts can be generated upon a compare match.

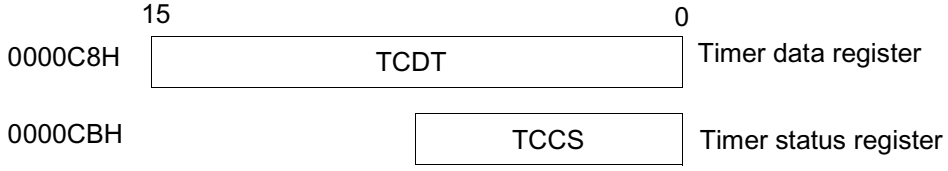
■ **Input capture (2 channels per one module)**

The input capture module consists of two 16-bit capture registers and control registers corresponding to two independent external input pins. The 16-bit free-running timer value can be stored in the capture register and an interrupt is issued simultaneously upon detection of an edge of a signal input from an external input pin.

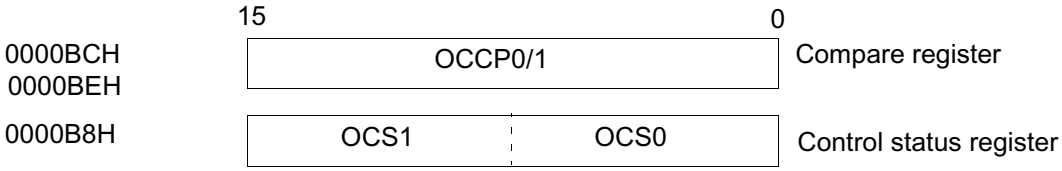
- The detection edge of an external input signal can be specified.
  Rising, falling, or both edges
- Two input channels can operate independently.
- An interrupt can be issued upon a valid edge of an external input signal.

20.2 REGISTERS

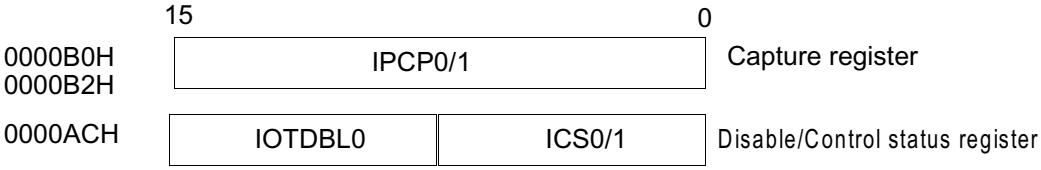
■ **16-bit free-running timer**



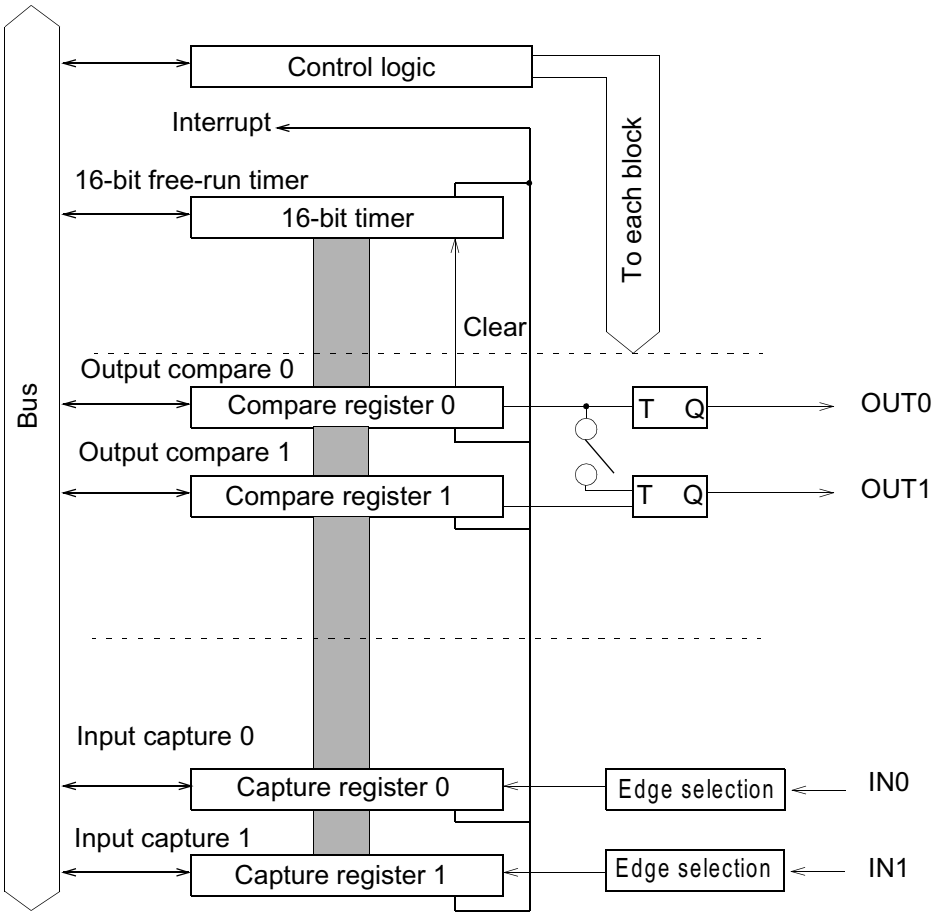
■ **16-bit output compare**



■ **16-bit input capture**



20.3 BLOCK DIAGRAM

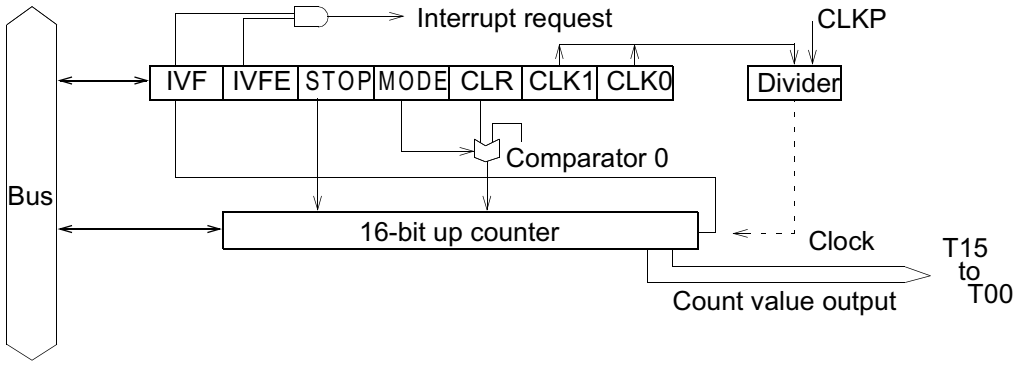


20.4 16-BIT FREE-RUNNING TIMER

The 16-bit free-running timer consists of a 16-bit up counter and a control status register. The count values of this timer are used as the base timer for the output compares and input captures.

- Four counter clock frequencies are available.
- An interrupt can be generated upon a counter value overflow.
- The counter value can be initialized upon a match with compare register 0, depending on the mode.

20.4.1 16-bit Free-Running Timer Block Diagram



20.4.2 16-bit Free-Running Timer Data register (TCDT)

Address: 0000C8H

bit	15	14	13	12	11	10	9	8
	T15	T14	T13	T12	T11	T10	T09	T08
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	0	0	0	0	0	0	0	0

←Attribute
←Initial value

Address: 0000C9H

bit	7	6	5	4	3	2	1	0
	T07	T06	T05	T04	T03	T02	T01	T00
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	0	0	0	0	0	0	0	0

←Attribute
←Initial value

The data register can read the count value of the 16-bit free-running timer. The counter value is cleared to "0000" upon a reset. The timer value can be set by writing a value to this register. However, ensure that the value is written while the operation is stopped (STOP=1).

The data register must be accessed by the word access instructions.

The 16-bit free-running timer is initialized upon the following factors:

- Reset
- Clear bit (CLR) of control status register
- A match between compare register 0 and the timer counter value.

20.4.3 16-bit F.-R. Timer Control/Status Register (TCCS)

Address: 0000CBH

bit	7	6	5	4	3	2	1	0	
	Reserved	IVF	IVFE	STOP	MODE	CLR	CLK1	CLK0	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	←Attribute
	0	0	0	0	0	0	0	0	←Initial value

[bit 7] Reserved bit

Always write "0" to this bit.

[bit 6] IVF

This bit is an interrupt request flag of the 16-bit free-running timer.

If the 16-bit free-running timer overflows, or if the counter is cleared by a match with compare register 0, "1" is set to this bit.

An interrupt is issued if the interrupt request enable bit (bit 5: IVFE) is set.

This bit is cleared by writing "0." Writing "1" has no effect.

"1" is always read by a read-modify-write instruction.

0	No interrupt request (default)
1	Interrupt request

[bit 5] IVFE

IVFE is an interrupt enable bit of the 16-bit free-run timer. While this bit is "1", an interrupt is issued if "1" is set to the interrupt flag (bit 5: IVF).

0	Interrupt disabled (default)
1	Interrupt enabled

[bit 4] STOP

The STOP bit is used to stop the 16-bit free-running timer.

Writing "1" to this bit stops the timer. Writing "0" starts the timer.

0	Counter enabled (operation) (default)
1	Counter disabled (stop)

* The output compare operation stops when the 16-bit free-running timer stops.

[bit 3] MODE

The MODE bit is used to set the reset condition of the 16-bit free-running timer.

When "0" is set, the counter value can be initialized by RESET or a clear bit (bit 2: CLR).

When "1" is set, the counter value can be initialized by a match with compare register 0 in addition to RESET and a clear bit (bit 2: CLR).

0	Initialization by reset or clear bit (default)
1	Initialization by reset, clear bit, or compare register 0

* The clear bit and the match with compare register initializes the timer when the timer value changes.

[bit 2] CLR

The CLR bit initializes the operating 16-bit free-running timer value to "0000."

When "1" is set, the counter value is initialized to "0000." Writing "0" has no effect. "0" is always read from this bit. The counter value is initialized when the count value changes.

0	No effect (default)
1	The counter value is initialized to "0000".

* To initialize the counter value while the timer is stopped, write "0000" to the data register.

[bits 1 and 0] CLK1 and CLK0

CLK1 and CLK0 are used to select the count clock for the 16-bit free-run timer. The clock is updated immediately after a value is written to these bits. Therefore, ensure that the output compare and input capture operations are stopped before a value is written to these bits.

CLK1	CLK0	Count clock	∅=16 MHz	∅=8 MHz	∅=4 MHz	∅=1 MHz
0	0	∅/4	0.25 µs	0.5 µs	1 µs	4 µs
0	1	∅/16	1 µs	2 µs	4 µs	16 µs
1	0	∅/32	2 µs	4 µs	8 µs	32 µs
1	1	∅/64	4 µs	8 µs	16 µs	64 µs

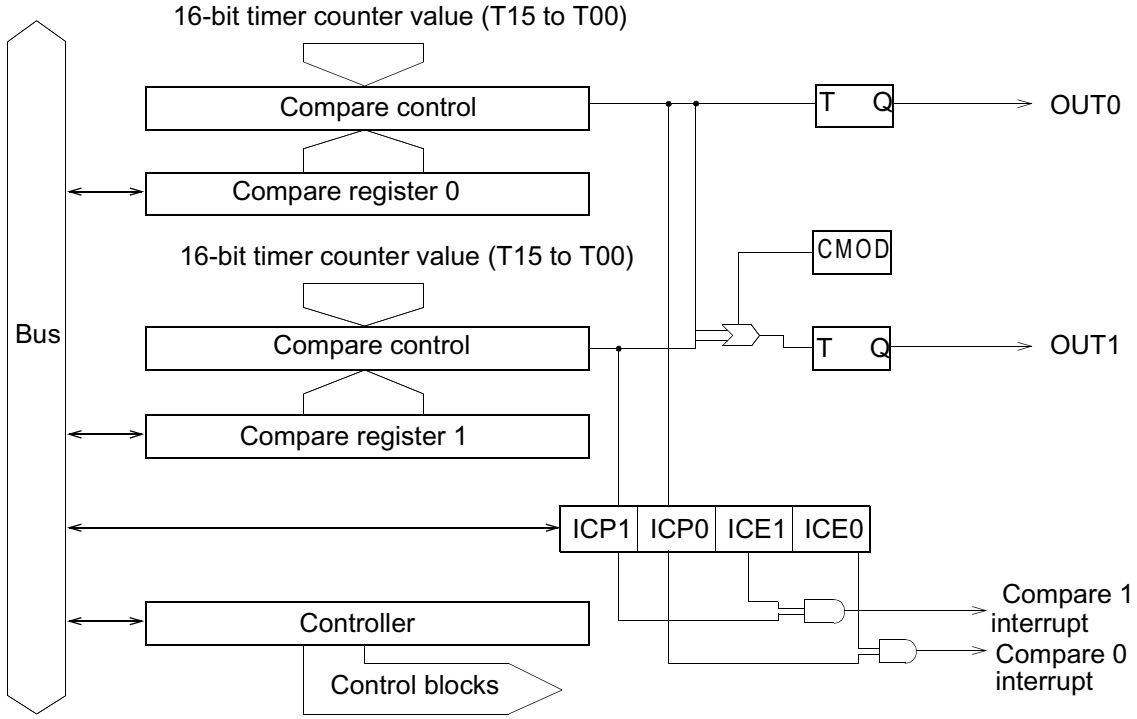
* ∅ = Peripheral clock CLKP

20.5 OUTPUT COMPARE

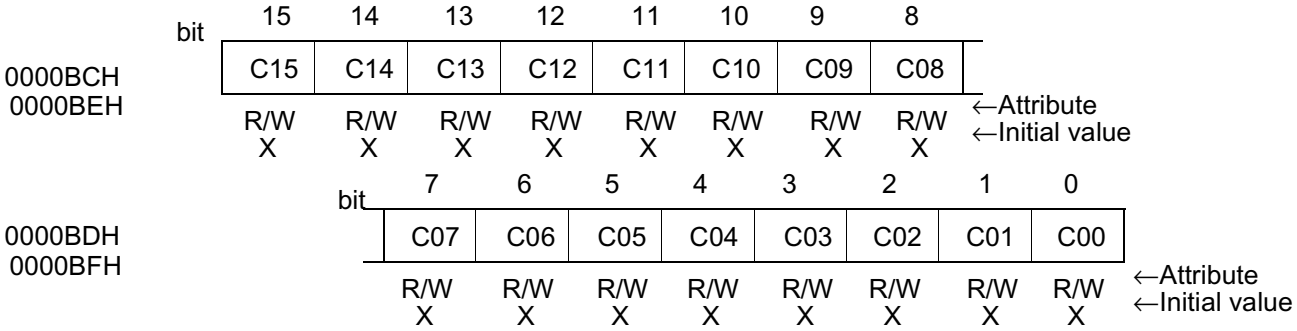
The output compare module consists of two 16-bit compare registers, two compare output pins, and control register. If the value written to the compare register of this module matches the 16-bit free-running timer value, the output level of the pin can be reversed and an interrupt can be issued.

- Two compare registers exist that can be used independently. Depending on the setting, the two compare registers can be used to control pin outputs.
- The initial value for the pin output can be specified.
- An interrupt can be issued upon a match as a result of comparison.

20.5.1 Output Compare Block Diagram



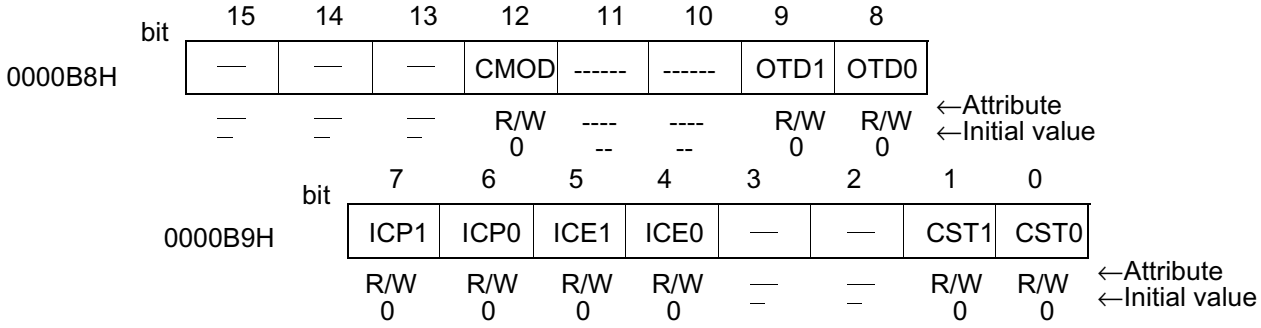
20.5.2 Output Compare Comparision Register (OCCP)



These 16-bit compare registers are compared with the 16-bit free-running timer. Since the initial register values are undefined, set appropriate value before enabling the operation. These registers must be accessed by the word access instructions. When the value of the register

matches that of the 16-bit free-running timer, a compare signal is generated and the output compare interrupt flag is set. If output is enabled, the output level corresponding to the compare register is reversed.

20.5.3 Output Compare Control status register (OCS01)



[bits 15, 14, and 13] Unused bits

[bit 12] CMOD

CMOD is used to switch the pin output level reverse mode upon a match.

- When CMOD=0 (default), the output level of the pin corresponding to the compare register is reversed.

  OUT0: The level is reversed upon a match with compare register 0.

  OUT1: The level is reversed upon a match with compare register 1.

- When CMOD=1, the output level is reversed for the compare register 0 in the same manner as for CMOD=0. The output level of the pin corresponding to compare register 1 (OUT1), however, is reversed upon a match with compare register 0 or 1. If compare registers 0 and 1 have the same value, the same operation as with a single compare register is performed.

  OUT0: The level is reversed upon a match with compare register 0.

  OUT1: The level is reversed upon a match with compare register 0 or 1.

[bits 11 and 10] Unused bits

[bits 9 and 8] OTD1 and OTD0

These bits are used to change the pin output level when the output compare pin output is enabled. The initial value of the compare pin output is "0." Ensure that the compare operation is stopped before a value is written. When read, these bits indicate the output compare pin output value.

0	Sets "0" for the compare pin output. (default)
1	Sets "1" for the compare pin output.

* OTD1: Corresponds to output compare 1. OTD0: Corresponds to output compare 0

[bits 7 and 6] ICP1 and ICP0

These bits are used as output compare interrupt flags. "1" is set to these bits when the compare register value matches the 16-bit free-run timer value. While the interrupt request bits (ICE1 and ICE0) are enabled, an output compare interrupt occurs when the ICP1 and ICP0 bits are set. These bits are cleared by writing "0."

Writing "1" has no effect. "1" is always read by a read-modify-write instruction.

0	No compare match (default)
1	Compare match

* ICP1: Corresponds to output compare 1. ICP0: Corresponds to output compare 0

[bits 5 and 4] ICE1 and ICE0

These bits are used as output compare interrupt enable flags. While the "1" is written to these bits, an output compare interrupt occurs when an interrupt flag (ICP1 or ICP0) is set.

0	Output compare interrupt disabled (default)
1	Output compare interrupt enabled

* ICE1: Corresponds to output compare 1. ICE0: Corresponds to output compare 0.

[bits 3 and 2] Unused bits

[bits 1 and 0] CST1 and CST0

These bits are used to enable the comparison with 16-bit free-run timer.

0	Compare operation disabled (default)
1	Compare operation enabled

Ensure that a value is written to the compare register before the compare operation is enabled.

* CST1: Corresponds to output compare 1. CST0: Corresponds to output compare 0.

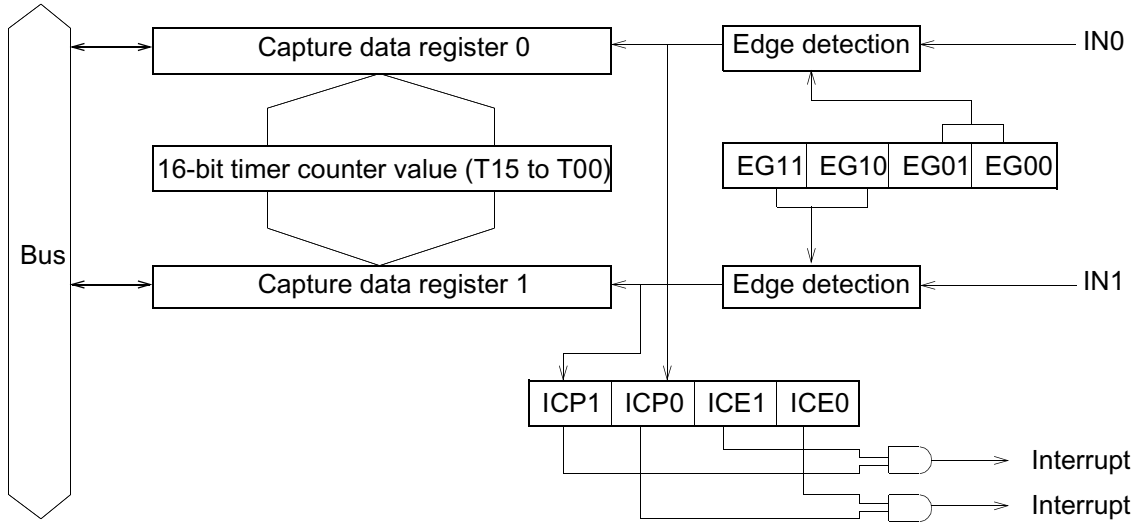
Since output compare is synchronized with the 16-bit free-running timer clock, stopping the 16-bit free-running timer stops compare operation.

20.6 INPUT CAPTURE

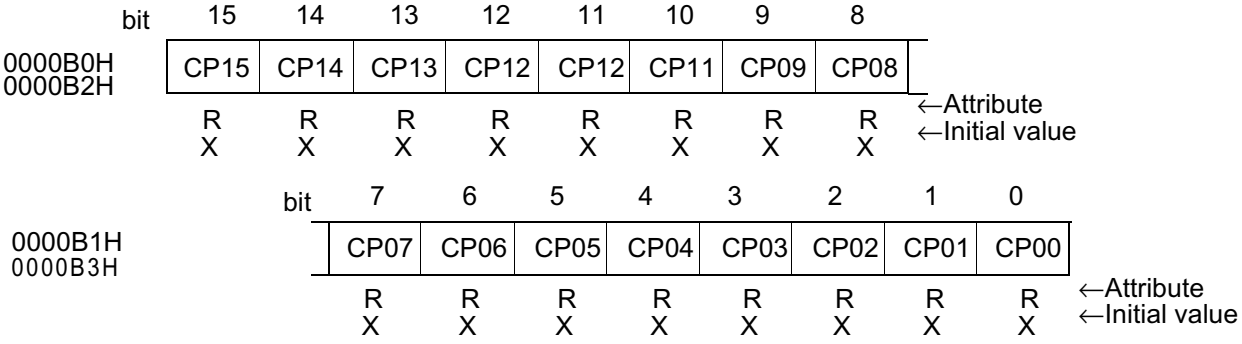
This module detects a rising or falling edge or both edges of an external input signal and stores the 16-bit free-running timer value in a register. In addition, this module can generate an interrupt upon detection of an edge. The input capture module consists of two input capture data registers and one control register. Each input capture has a corresponding external input pin.

- The detection edge of an external input can be selected from three types:
  Rising edge, falling edge, or both edges
- An interrupt can be generated upon detection of a valid edge of an external input.

20.6.1 Input Capture Block Diagram



20.6.2 Input Capture Data Register (IPCP)



This register stores the 16-bit timer value when a valid edge of the corresponding external pin input waveform is detected. (This register must be accessed in word mode. No value can be written to this register.)

20.6.3 Input Capture Control Status Register (ICS)

bit	7	6	5	4	3	2	1	0	
0000ADH	ICP1	ICP0	ICE1	ICE0	EG11	EG10	EG01	EG00	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	←Attribute
	0	0	0	0	0	0	0	0	←Initial value

[bits 7 and 6] ICP1 and ICP0

These bits are used as input capture interrupt flags. "1" is set to this bit upon detection of a valid edge of an external input pin. While the interrupt enable bits (ICE0 and ICE1) are set, an interrupt can be generated upon detection of a valid edge.

These bits are cleared by writing "0." Writing "1" has no effect. "1" is always read by a read-modify-write instruction.

0	No valid edge detection (default)
1	Valid edge detection

* ICP0: Corresponds to input capture 0. ICP1: Corresponds to input capture 1.

[bits 5 and 4] ICE1 and ICE0

These bits are used to enable input capture interrupts. While these bits are "1", an input capture interrupt is generated when the interrupt flag (ICP0 or ICP1) is set.

0	Interrupt disabled (default)
1	Interrupt enabled

* ICE0: Corresponds to input capture 0. ICE1: Corresponds to input capture 1.

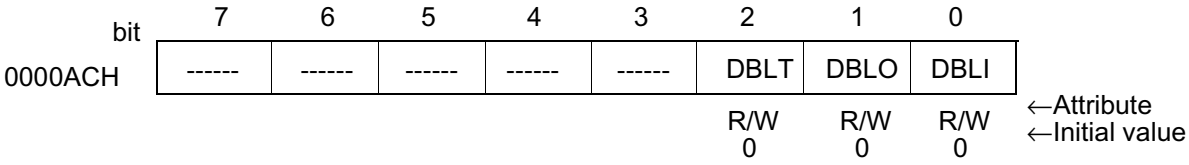
[bits 3, 2, 1, and 0] EG11, EG10, EG01, and EG00

These bits are used to specify the valid edge polarity of the external inputs. These bits are also used to enable input capture operation.

EG11 EG01	**EG10 EG00**	**Edge detection polarity**
0	0	No edge detection (stop) (default)
0	1	Rising edge detection ⇧
1	0	Falling edge detection ⇩
1	1	Both edge detection ⇧ ⇩

* EG01 and EG00: Correspond to input capture 0. EG11 and EG10: Correspond to input capture 1.

20.6.4 Input Capture Disable Register (IOTDBL)



[bit 2] DBLT

This bit can be used to disable the clock for the Free Running Timer Module. "1" means the clock is disabled.

[bit 1] DBLO

This bit can be used to disable the clock for the Output Compare Module (both channels). "1" means the clock is disabled.

[bit 0] DBLI

This bit can be used to disable the clock for the Input Capture Module (both channels). "1" means the clock is disabled.

20.7 OPERATIONS

20.7.1 16-bit Free-Running Timer

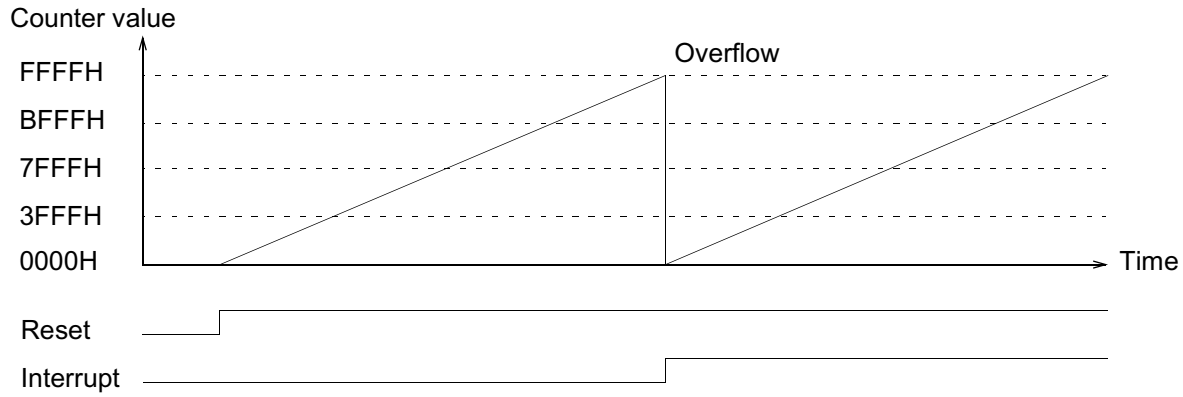
The 16-bit free-running timer starts counting from counter value "0000" after the reset is released. The counter value is used as the reference time for the 16-bit output compare and 16-bit input capture operations.

The counter value is cleared in the following conditions:

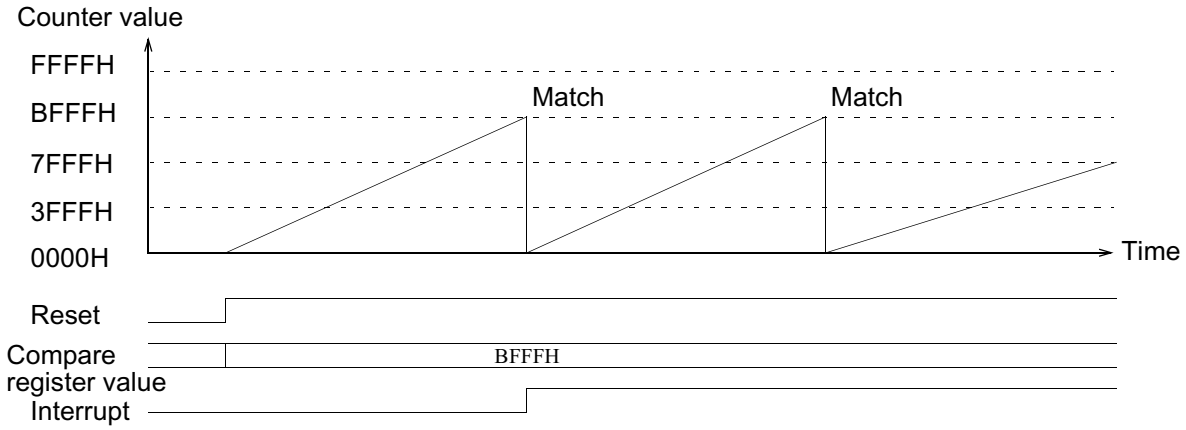
- When an overflow occurs.
- When a match with the output compare register 0 occurs. (This depends on the mode.)
- When "1" is written to the CLR bit of the TCCS register during operation.
- When "0000" is written to the TCDC register during stop.
- Reset

An interrupt can be generated when an overflow occurs or when the counter is cleared by a match with the compare register 0. (Compare match interrupts can be used only in an appropriate mode.

■ **Clearing the counter by an overflow**



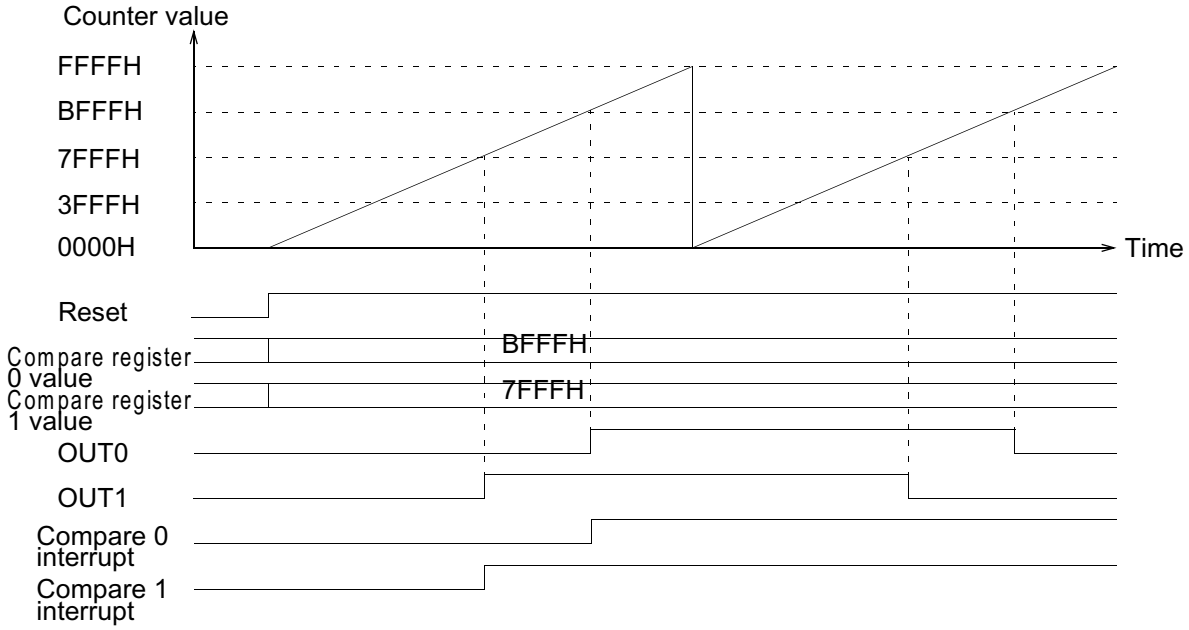
■ **Clearing the counter upon a match with output compare register**



20.7.2 16-bit Output Compare

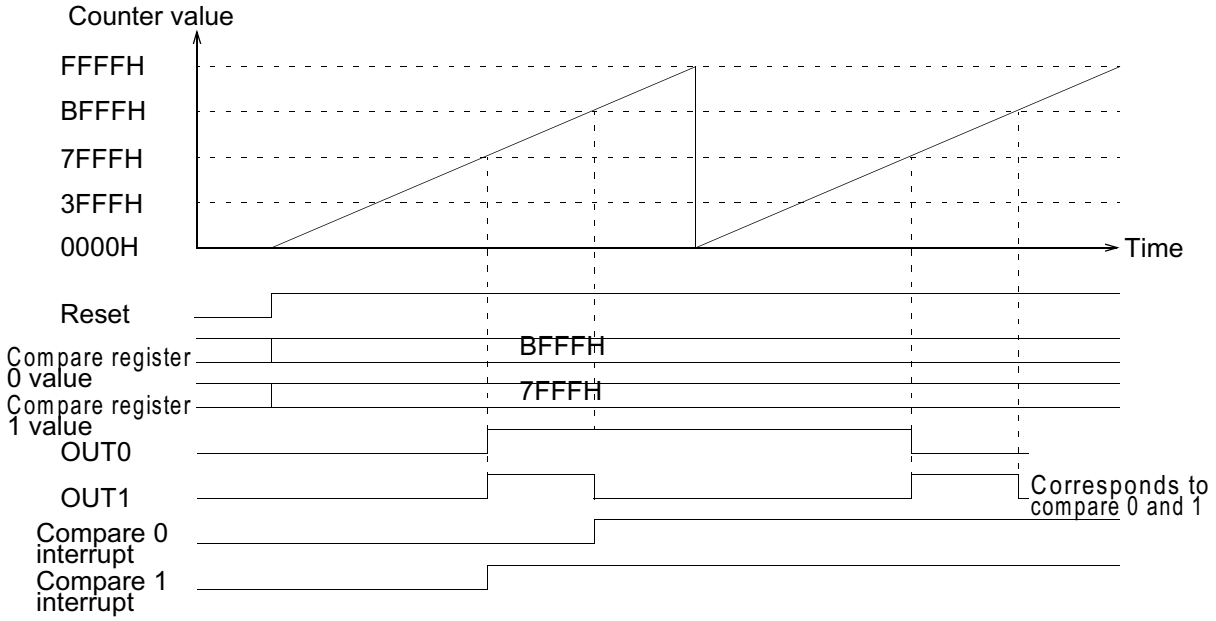
In the 16-bit output compare operation, an interrupt request flag can be set and the output level can be reversed when the specified compare register value matches the 16-bit free-run timer value.

■ **Sample of output waveform when compare registers 0 and 1 are used (The initial output value is 0.)**



The output level can be changed using two compare registers (when CMOD=1).

■ **Sample of a output waveform with two compare registers (The initial output value is "0.")**



20.7.3 16-bit Input Capture

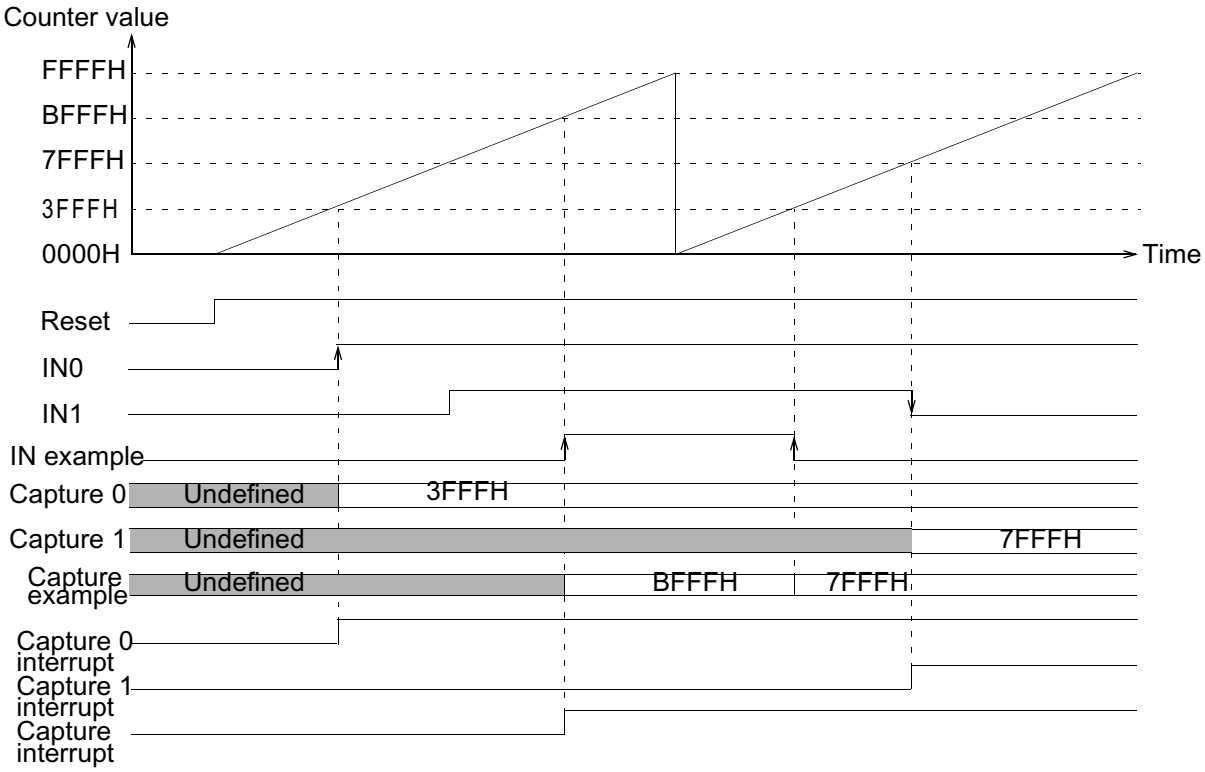
In 16-bit input capture operation, an interrupt can be generated upon detection of at the specified edge, fetching the 16-bit free-run timer value and writing it to the capture register.

■ **Sample of input capture fetch timing**

Capture 0: Rising edge

Capture 1: Falling edge

Capture example: Both edges

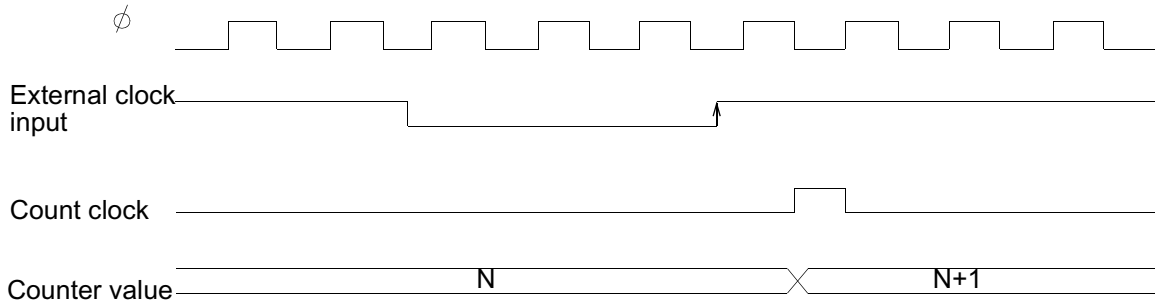


20.8 TIMING

20.8.1 16-bit Free-Running Timer Count Timing

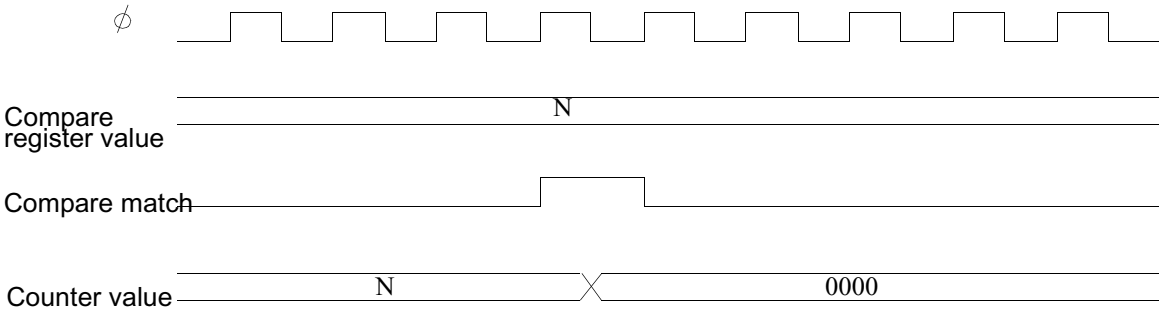
The 16-bit free-run timer is incremented based on the input clock (internal or external clock). When external clock is selected, the 16-bit free-run timer is incremented at the rising edge.

■ **Free-running timer count timing**



The counter can be cleared upon a reset, software clear, or a match with the compare register 0. By a reset or software clear, the counter is immediately cleared. By a match with compare register 0, the counter is cleared in synchronization with the count timing.

■ **Free-running timer clear timing (match with the compare register 0)**

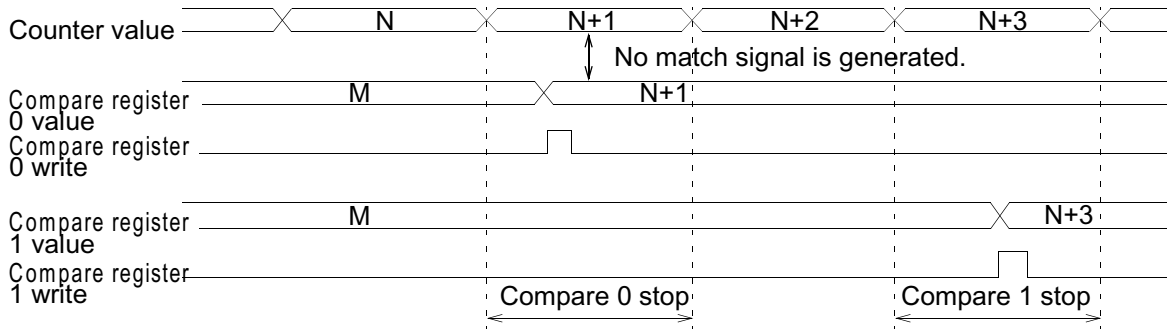


20.8.2 Output Compare Timing

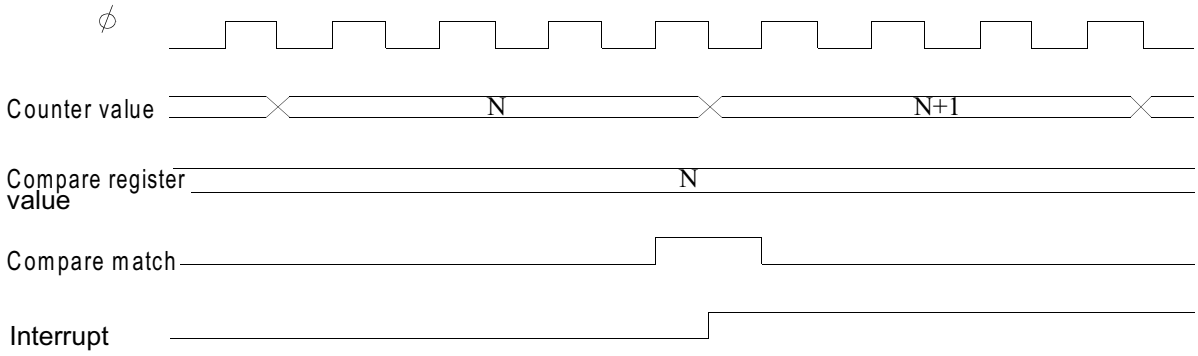
In output compare operation, a compare match signal is generated when the free-running timer value matches the specified compare register value. The output value can be reversed and an interrupt can be issued. The output reverse timing upon a compare match is synchronized with the counter count timing.

■ Compare operation upon update of compare register

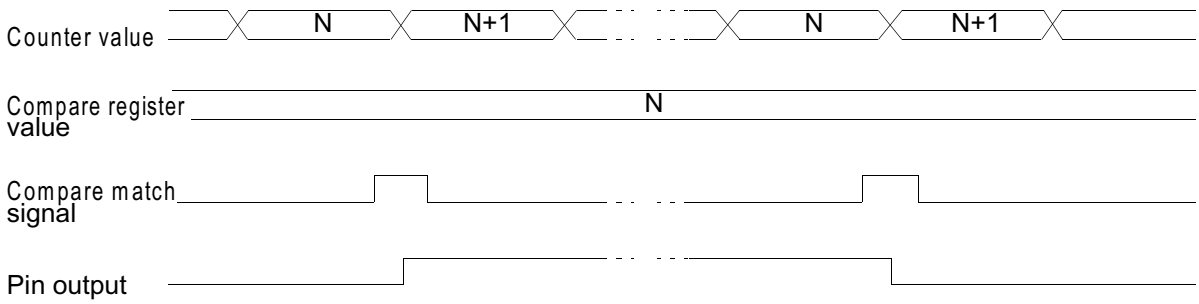
When the compare register is updated, comparison with the counter value is not performed.



■ Interrupt timing

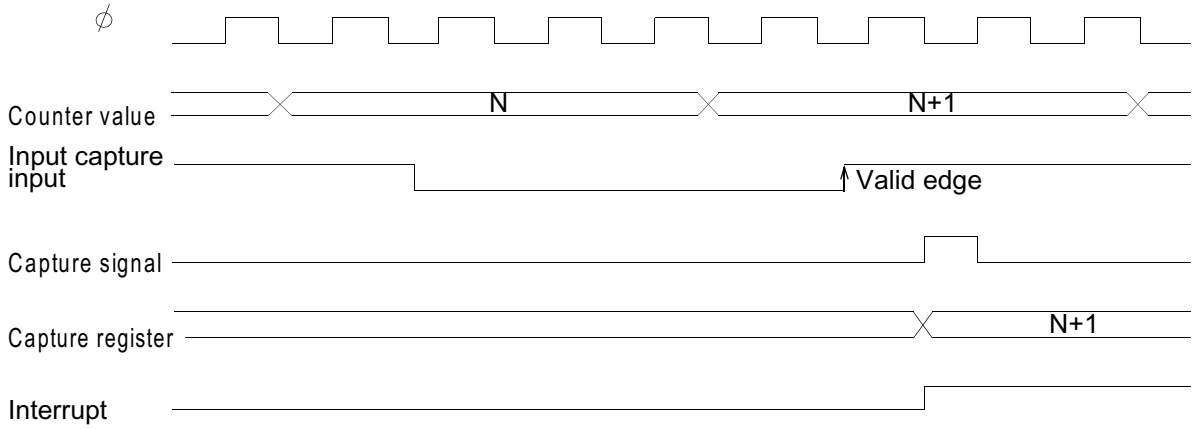


■ Output pin change timing



20.8.3 Input Capture Input Timing

■ **Capture timing for input signals**



CHAPTER 21 ALARM COMPARATOR

This chapter provides an overview of the Alarm Comparator (also called Under/ Overvoltage Detection), describes the register structure and functions, and describes the operation of the Alarm Comparator.

21.1 BLOCK DIAGRAM

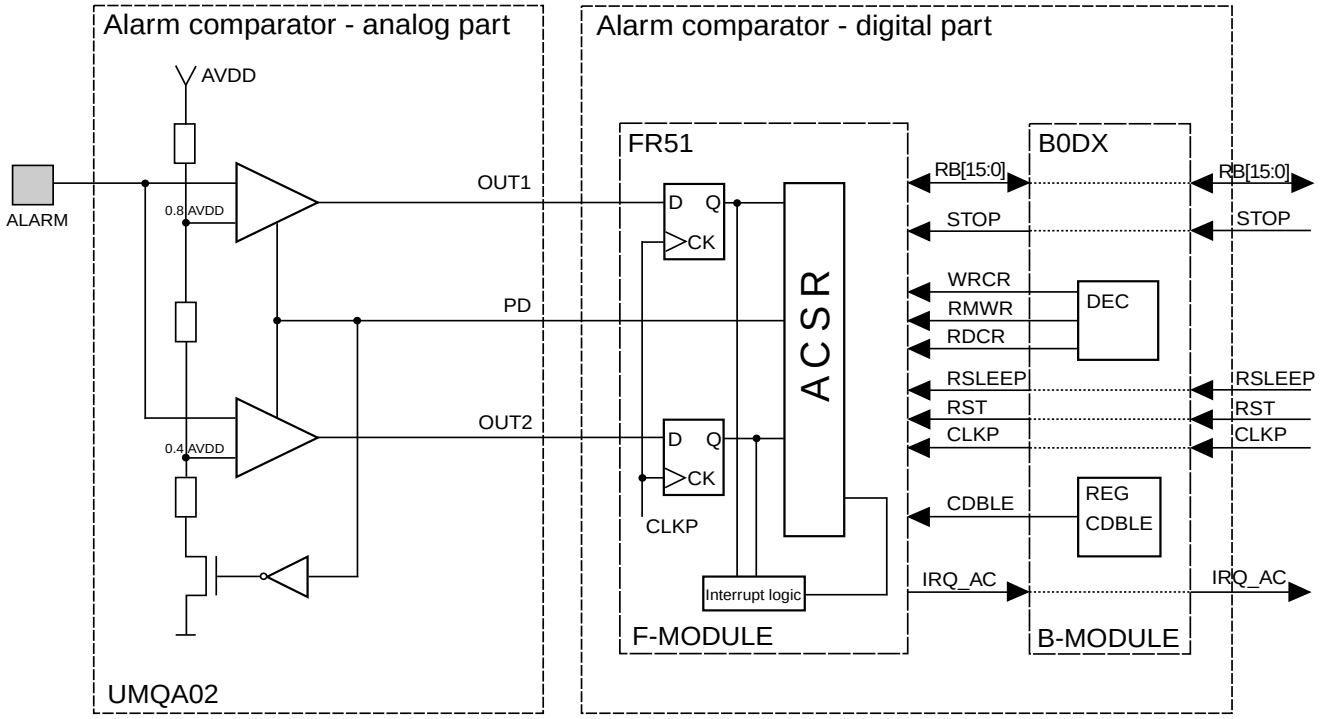


Figure 21.1 Alarmcomparator (simplified circuit)

21.2 REGISTERS

21.2.1 Alarm Comparator Clock Disable Register (ACCDBL)

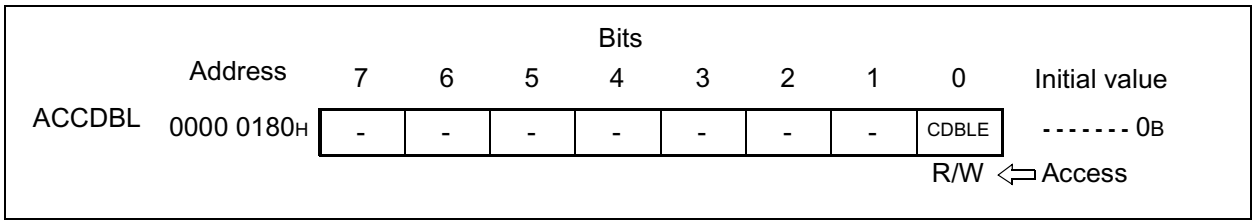


Figure 21.2.1 Structure of Alarm comparator clock disable register

Bit 0: CDBLE clock disable bit.

1	Clock disable (digital part of the alarm comparator)
0	Clock enable (digital part of the alarm comparator) [Initial value]

21.2.2 Alarm Comparator Status Disable Register (ACSR)

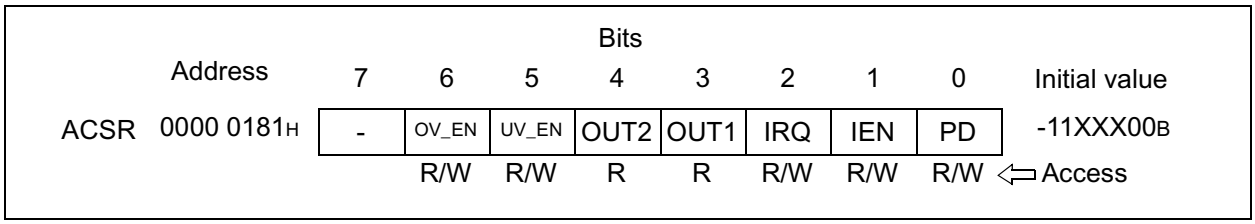


Figure 21.2.2 Structure of Alarm comparator status register

Bit 6:OV_EN Overvoltage Enable

1	Interrupt enabled in case of overvoltage. [Initial value]
0	No interrupt in case of overvoltage

Bit 5: UV_EN Undervoltage Enable

1	Interrupt enabled in case of undervoltage [Initial value]
0	No interrupt in case of undervoltage.

Bit 4: OUT2 synchronized output of Alarm comparator UV output.

0	analog input voltage < 2/5 VDDA.
1	analog input voltage > 2/5 VDDA

Bit 3: OUT1 synchronized output of Alarm comparator OV output.

1	analog input voltage > 4/5 VDDA.
0	analog input voltage < 4/5 VDDA

Bit 2: IRQ Interrupt bit.

1	Under- or overvoltage condition detected
0	Normal operation

Bit 1: IEN Interrupt enable bit.

1	Interrupt assertion enabled
0	Interrupt assertion disabled [Initial value]

Bit 0: PD Power down bit.

1	Power down (analog part)
0	Runmode (analog part) [Initial value]

21.3 OPERATION MODES

The alarm comparator circuit can operate in interrupt or polling mode. The internal interrupt logic will detect each interrupt event independent from setting of the IEN bit.

21.3.1 Interrupt Mode (IEN=1)

The following truth table describes the valid interrupt events

Table 2:

OUT2	OUT1	IRQ	analog input voltage range
1	1	1	Vin > 0.8 AVDD (overvoltage)
1	0	0	0.4 AVDD < Vin < 0.8 AVDD (normal operation)
0	0	1	0.4 AVDD > Vin (undervoltage)

The interrupt Bit IRQ will be set with the next positive transition of CLKP after detecting an interrupt event. If IEN=1 this will create an interrupt request to the CPU. In order to determine the reason for the asserted interrupt - if both interrupts are enabled - it is necessary to read the ACSR register immediately inside the interrupt service routine. OUT2 and OUT1 always contain the actual status of the comparator outputs, i.e. the interrupt trigger event will not be stored.

21.3.2 Polling Mode (IEN=0)

The IRQ register bit will be set by an active interrupt event and can be reset by writing to the ACSR register.The ACSR can be polled continuously in order to monitor the input voltage which is feed to the AC comparator inputs.

21.3.3 Setting and Resetting of IRQ-Flagbit

The IRQ bit of the ACSR register can be reset to zero by writing a “0” to it. Writing an “1” to the IRQ bit of ACSR register has no effect. IRQ can only be set to “1” by hardware, i.e. by the outputs of the comparator circuits. IRQ will remain active as long as an active interrupt status is detected, even if a “0” is written to it.

A bitset command performed on the ACSR register will result in a RMW access on the R-Bus. Every read access during performing a RMW command will return a “1” for the IRQ flag to the CPU. That avoids any loss in detecting interrupt events due to software setting of IRQ-Flag Bit.

21.3.4 Power Down Modes of the Alarm Comparator

The alarm comparator circuit has the following power down modes:

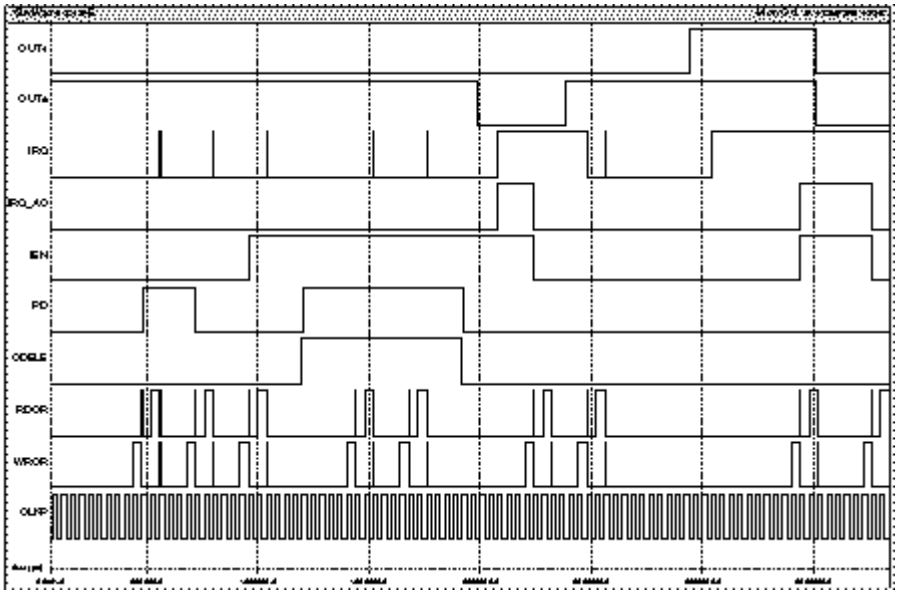
Table 21.3.4 Alarm Comparator power down modes

STOP	PD	CDBLE	analog part(UMQA01	digital part(FR51)
0	0	0	run mode	run mode
0	0	1	power down mode	power down mode
0	1	0	power down mode	run mode
0	1	1	power down mode	power down mode
1	0	0	power down mode	power down mode
1	0	1	power down mode	power down mode
1	1	0	power down mode	power down mode
1	1	1	power down mode	power down mode

Precaution: The outputs of the alarm comparator (analog parts) will remain undefined for at least 3 us after power on and also after reentering the runmode. You have to make sure whether the IRQ is correct set before enabling the alarm comparator interrupt source.

21.4 SIMULATION OF THE VERILOG MODEL OF ALARM

The following sketch displays some typical signals of the AC-module.

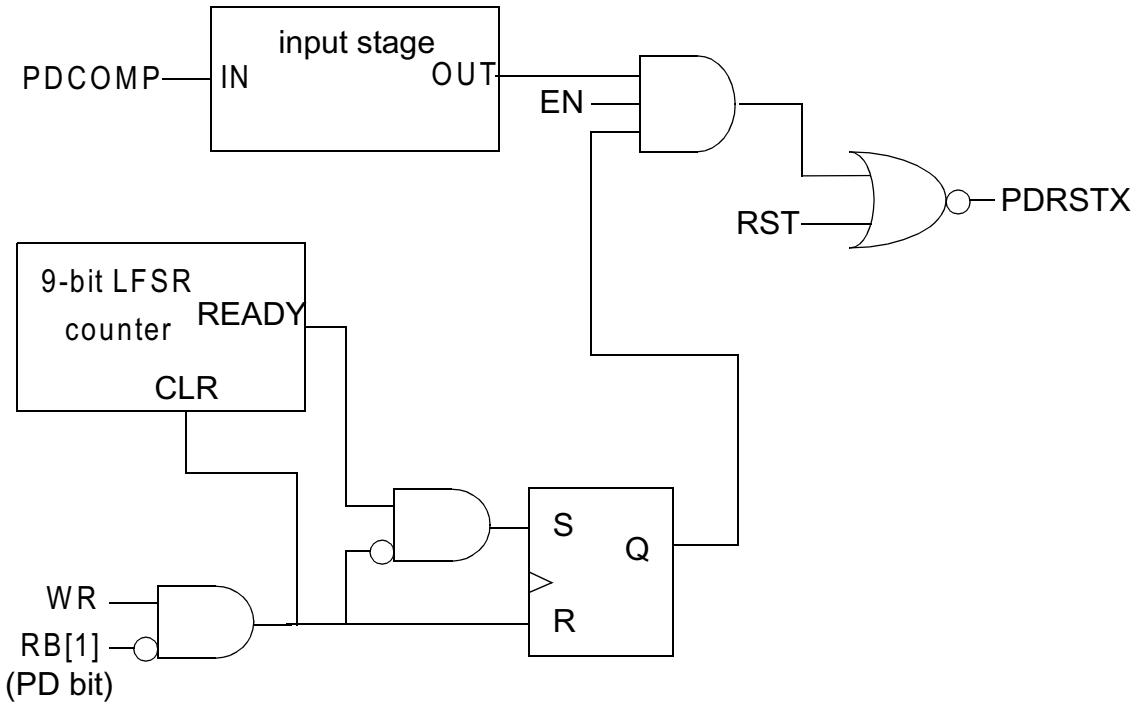


CHAPTER 22 POWER DOWN RESET

This Chapter provides an overview of the Power Down Reset, describes the register structure and functions, and describes the operation of the Power Down Reset Module

22.1 OVERVIEW

The power down reset module performs a system reset when VCC goes below a threshold voltage. The reset signal is be disabled and enabled by setting the power down reset control register PDRCR. For low power applications the digital and the analog part of the power down reset control circuit can be disabled.



■ Features

The power down reset circuit has following features:

- monitors the 5V power supply voltage and generates a reset signal after 2.27μs (typ) in case of a voltage lower than the threshold of 4.0V (typ)
- internal band gap circuit for reference voltage
- hysteresis input 50mV (typ) for monitored voltage
- reduced current consumption by 260 μA (typ) if the analog part of the power down reset circuit is switched to power down mode

22.2 REGISTER

PDRCR	7	6	5	4	3	2	1	0
Address: 00017DH	–	–	–	–	–	CDSBLE	PD	EN
access	–	–	–	–	–	R/W	R/W	R/W
initial value (INIT)	–	–	–	–	–	0	0	0
initial value (RST)	X	X	X	X	X	X	X	X

[Bit 2] CDSBLE: Clock Disable

Disables the clock of the digital part of the power down reset comparator.

0	Clock is enabled.	[Initial value]
1	Clock is disabled.	

[Bit 1] PD: Power Down

Controls the low power mode of the analog part of the power down reset comparator.

0	Low power mode is disabled.	[Initial value]
1	Low power mode is enabled.	

[Bit 0] EN: Enable

Enables the power down reset comparator.

0	Power down reset circuit is disabled	[Initial value]
1	Power down reset circuit is enabled.	

22.3 OPERATION MODES

22.3.1 Run Modes

After initial reset the EN bit of the PDRCR is disabled. To activate the power down reset control circuit it must be enabled.

If the VDD voltage falls below threshold, the PDCOMP signal becomes high. To avoid metastable states caused by the asynchronous event of voltage drop the input stage delays the PDCOMP signal by two clock cycles. Then the PDRSTX signal becomes low active.

If the PD bit of the PDRCR is switched from 1 to 0 during low voltage of VDD, the PDCOMP signal is in unknown state for 3.19 µs. To avoid undefined behaviour of the circuit, a 9-bit counter is reset after write access to the PD bit. The counter enables the power down reset control circuit after 511 clock cycles. Assuming the fastest specified frequency for CLKP=32 MHz this will result in a delay of 21µs.

22.3.2 Low Power Modes

The digital part and the analog part of the power down reset circuit can be set in low power modes. Additional to the clock disabling of the FR52, the analog circuit UMQA01 can be set in low power mode.

PD	CDSBLE	Analog Part UMQA01	Digital Part FR52
0	0	run	run
0	1	run	low power
1	1	low power	low power

CHAPTER 23 SERIAL I/O INTERFACE (SIO)

This Chapter provides an overview of the Serial I/O Interface (SIO), describes the register structure and functions, and describes the operation of the SIO.

23.1 BLOCK DIAGRAM

This block is a serial I/O interface that allows data transfer using clock synchronization. The interface consists of a single eight-bit channel. Data can be transferred from the LSB or MSB.

MB91360 contains two Serial I/O units SIO0 and SIO1. This chapter only describes SIO0. Please see the IO-Map for the register addresses of SIO1.

The serial I/O interface operates in two modes:

- Internal shift clock mode: Data is transferred in synchronization with the internal clock.
- External shift clock mode: Data is transferred in synchronization with the clock supplied via the external pin (SCK). By manipulating the general-purpose port sharing the external pin (SCK), data can also be transferred by a CPU instruction in this mode.

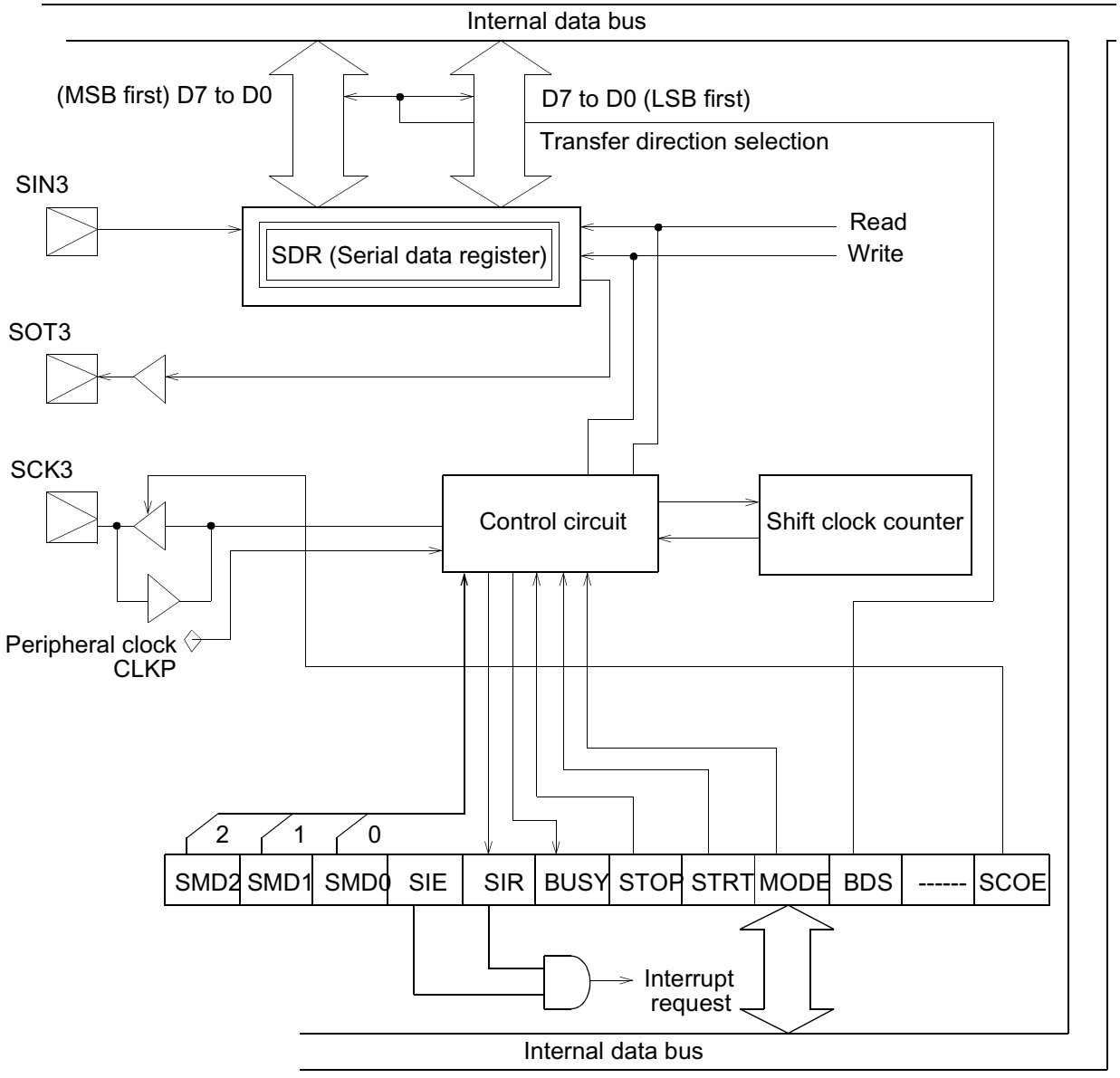
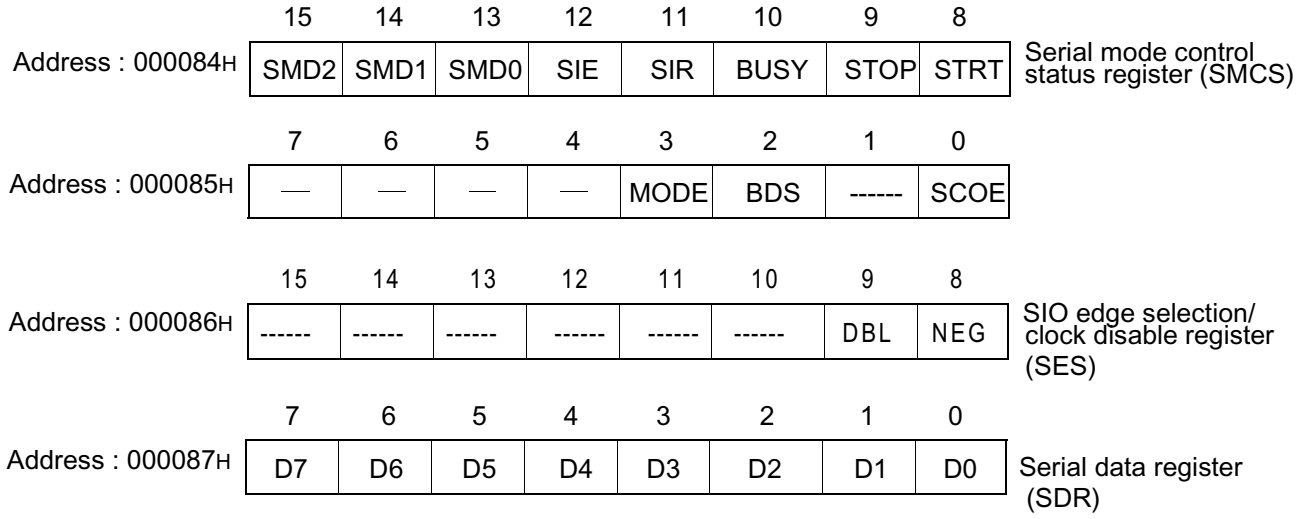


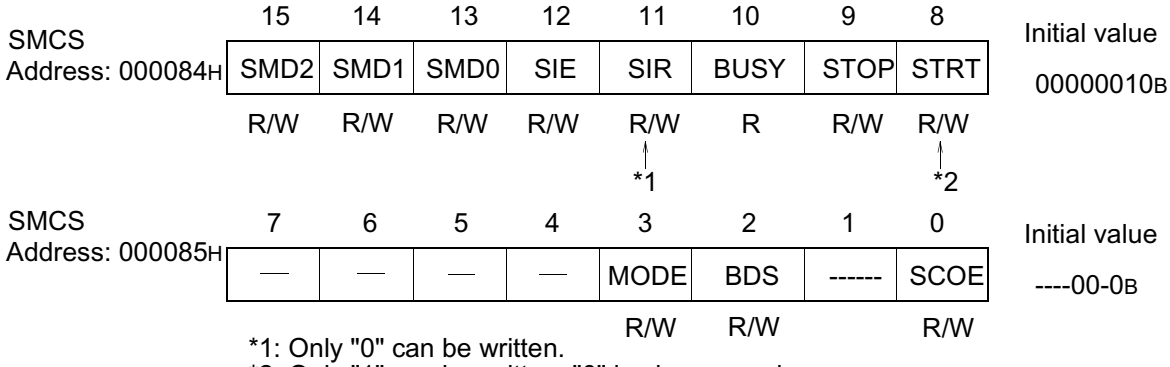
Figure 23.1 Serial I/O interface block diagram

23.2 REGISTERS



23.3 REGISTER DETAILS

23.3.1 Serial Mode Control Status Register (SMCS)



*1: Only "0" can be written.
*2: Only "1" can be written. "0" is always read.

This register controls the serial I/O transfer mode. The functions of the bits are explained below.

[bit 3] Serial mode selection bit (MODE)

The serial mode selection bit is used to select the conditions to start the transfer operation from the stop state. This bit must not be updated during operation.

MODE	Operation
0	Transfer starts when STRT=1. [Default]
1	Transfer starts when the serial data register is read or written to.

This bit is initialized to a "0" upon a reset, and can be read or written to. To activate the intelligent I/O service, ensure that "1" is written to this bit.

[bit 2] Bit order select bit (BDS)

This bit specifies the bit ordering of the data transfer. The data can be transferred from the least significant bit (LSB first mode) or from the most significant bit (MSB first mode).

BDS	Operation
0	LSB first [default]
1	MSB first

<Note>
Specify the bit ordering before any data is written to SDR.

[bit 1] Unused bit

[bit 0] Shift clock output enable bit (SCOE: SCK1 output enable)

The shift clock output enable bit controls the output from the shift clock I/O output external pin as described below:

SCOE	Operation
0	Shift clock input pin [default]
1	Shift clock output pin

Ensure that "0" is written to this bit when data is transferred for each instruction in external shift clock mode.

This bit is initialized to "0" upon a reset. This bit is readable and writable.

[Bits 15, 14, and 13] Shift clock selection bits (SMD2, SMD1, SMD0: Serial shift clock mode)

The shift clock selection bits are used to select the serial shift clock mode as described below.

SMD2	SMD1	SMD0	CLKP= 16 MHz, div=16	CLKP= 8 MHz, div=8	CLKP= 4 MHz, div=8
0	0	0	1 MHz	1 MHz	500 kHz
0	0	1	500 kHz	500 kHz	250 kHz
0	1	0	125 kHz	125 kHz	62.5 kHz
0	1	1	62.5 kHz	62.5 kHz	31.25 kHz
1	0	0	31.25 kHz	31.25 kHz	15.625 kHz
1	0	1	External shift clock mode		
1	1	0	Reserved		
1	1	1	Reserved		

div	Serial I/O prescaler Div3 - Div0				Recommended CLKP frequency
6	1	1	0	1	6 MHz
8	1	1	0	0	8 MHz
10	1	0	1	1	10 MHz
12	1	0	1	0	12 MHz
14	1	0	0	1	14 MHz
16	1	0	0	0	16 MHz

Setting of the Serial I/O prescaler (CDCR)
* For details, see 23.4 "SERIAL I/O PRESCALER" on page 535.

These bits are initialized to "000" upon a reset. These bits must not be updated during data transfer.

Five types of internal shift clock and an external shift clock are available. Do not set 110 or 111 in SMD2, SMD1, and SMD0 as these values are reserved.

If the external shift clock mode is selected, one bit shift operation can be performed by software. This can be done by changing the output level of the general purpose I/O sharing the shift clock input.

[bit 12] Serial I/O interrupt enable bit (SIE: Serial I/O interrupt enable)

The serial I/O interrupt enable bit controls the serial I/O interrupt request as described below.

SIE	Operation
0	Serial I/O interrupt disabled [default]
1	Serial I/O interrupt enabled

This bit is initialized to "0" upon a reset. This bit is readable and writable.

[bit 11] Serial I/O interrupt request bit (SIR: Serial I/O interrupt request)

When serial data transfer is completed, "1" is set to this bit. If this bit is set while interrupts are enabled (SIE=1), an interrupt request is issued to the CPU. The clear condition varies with the MODE bit.

When "0" is written to the MODE bit, the SIR bit is cleared by writing "0". When "1" is written to the MODE bit, the SIR bit is cleared by reading or writing to SDR. When the system is reset or "1" is written to the STOP bit, the SIR bit is cleared regardless of the MODE bit value.

Writing "1" to the SIR bit has no effect. "1" is always read by a read operation of a read-modify-write instruction.

[bit 10] Transfer status bit (BUSY)

The transfer status bit indicates whether serial transfer is being executed.

BUSY	Operating
0	Stopped, or standing by for serial data register R/W [default]
1	Serial transfer

This bit is initialized to "0" upon a reset. This is a read-only bit.

[bit 9] Stop bit (STOP)

The stop bit forcibly terminates serial transfer. When "1" is written to this bit, the transfer is stopped.

STOP	Operating
0	Normal operation
1	Transfer stop by STOP=1 [default value]

This bit is initialized to "1" upon a reset. This bit is readable and writable.

[bit 8] Start bit (STRT: Start)

The start bit activates serial transfer. Writing "1" to this bit starts the data transfer when the MODE bit is set to 0. When the MODE bit is set to 1 and the STRT bit is set to 1, writing the data into serial data register starts the transfer.

Writing "1" is ignored while the system is performing serial transfer or standing by for a serial shift register read or write. Writing "0" has no effect."0" is always read.

23.3.2 Serial Shift Data Register (SDR)

SDR
Address : 000087H

7	6	5	4	3	2	1	0
D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value XXH
(undefined)

This serial data register stores the serial I/O transfer data. During transfer, the SDR must not be read or written to.

23.3.3 SIO Edge Selection / Clock Disable Register (SES) .

SES
Address : 000086H

7	6	5	4	3	2	1	0
—	—	—	—	—	—	DBL	NEG
						R/W	R/W

Initial value
------00B

[bit 1] DBL (Disable)

DBL	Operation
0	Normal operation [default]
1	The clock for the SIO module is disabled

[bit 0] NEG (Shift clock negation)

NEG	Operation
0	Normal operation [default]
1	The shift clock is inverted

See also 23.5.7 "Negative Clock Operation" on page 541.

23.4 SERIAL I/O PRESCALER

The Serial I/O Prescaler provides the shift clock for the Serial I/O.

CDC.R
Address: 00008CH

15	14	13	12	11	10	9	8
MD	—	—	—	DIV3	DIV2	DIV1	DIV0
R/W				R/W	R/W	R/W	R/W

Initial value
0---1111B

The operation clock for the Serial I/O is obtained by dividing the peripheral clock CLKP. The Serial I/O is designed so that a constant baud rate can be obtained for a variety of CLKP clocks by the user of the communication prescaler. The CDCR register controls the CLKP clock division.

[bit 15] MD (Clock divide mode select):

This bit is used to control the operation of the communication prescaler.

MD	Operation
0	The Serial I/O Prescaler is disabled and reset.
1	The Serial I/O Prescaler is enabled.

Note: Setting MD to 0 resets the prescaler. This feature can be used to generate SPI-Timing. See 23.5.5 "SPI-Timing" on page 541.

[bits 11, 10, 9, and 8] DIV3 to DIV0 (Divide 3 to 0):

These bits are used to determine the CLKP clock division ratio.

DIV3 to 0	CLKP division ratio
1101B	6
1100B	8
1011B	10
1010B	12
1001B	14
1000B	16

When the division ratio is changed, allow two cycles for the clock to stabilize before starting communication.

23.5 OPERATIONS

23.5.1 Outline

The serial I/O consists of the serial mode control status register (SMCS) and shift register (SDR), and is used for input and output of 8-bit serial data. The bits in the shift register are serially output via the serial output pin (SOT1 pin) at the falling edge of the serial shift clock (external clock or internal clock). The bits are serially input to the shift register (SDR) via the serial input pin (SIN1 pin) at the rising edge of the serial shift clock. The shift direction (transfer from MSB or LSB) is specified by the direction specification bit (BDS) of the serial mode control status register (SMCS).

At the end of serial data transfer, this block is stopped or stands by for a read or write of the data register according to the MODE bit of the serial mode control status register (SMCS). To start transfer from the stop or standby state, follow the procedure below.

- To resume operation from the stop state, write "0" to the STOP bit and "1" to the STRT bit. (The STO and STRT bits can be set simultaneously.)
- To resume operation from the serial shift data register R/W standby state, read or write to the data register.

23.5.2 Shift Clock

There are two modes of shift clock: internal or external shift clock. These two modes are selected by setting the SMCS. To switch the modes, ensure that serial I/O transfer is stopped. To check whether the serial I/O transfer is stopped, read the BUSY bit.

■ **Internal shift clock mode**

In internal shift clock mode, data transfer is based on the peripheral clock CLKP. As a synchronization timing output, a shift clock of 50% duty ratio can be output from the SCK pin. Data is transferred at one bit per clock. The transfer speed is expressed as follows:

$$\text{Transfer speed (s)} = \frac{A}{\text{CLKP machine cycle (Hz)}}$$

"A" is the division ratio indicated by the SMD bits of SMCS. The value can be 2^1, 2^2, 2^3, 2^4, or 2^5.

■ **External shift clock mode**

In external shift clock mode, the data transfer is based on the external clock supplied via the SCK pin. Data is transferred at one bit per clock.

The transfer speed can be between close to 0 MHz and 1/(5 CLKP cycles). For example, the transfer speed can be up to 2 MHz when 1 CLKP cycle is equal to 0.1 µs.

A data bit can also be transferred by software, which is enabled as described below.

Select external shift clock mode, and write "0" to the SCOE bit of SMCS. Then, write "1" to the direction register for the port sharing the SCK pin, and place the port in output mode. Then, when "1" and "0" are written to the data register (PDR) of the port, the port value output via the SCK pin is fetched as the external clock and transfer starts. Ensure that the shift clock starts from "H."

<Note>
The SMCS or SDR must not be written to during serial I/O operation.

23.5.3 Serial I/O operation

There are four serial I/O operation status: STOP, halt, SDR R/W standby, and transfer.

■ **STOP**

The STOP state is initiated upon RESET or when "1" is written to the STOP bit of SMCS. The shift counter is initialized, and "0" is written to SIR.

To resume operation from the STOP state, write "0" to STOP and "1" to STRT. (These two bits can be written to simultaneously.) Since the STOP bit overrides the STRT bit, transfer cannot be started by writing "1" to STRT while "1" is written to STOP.

■ **Halt**

When transfer is completed while the MODE bit is "0," "0" is set to BUSY and "1" is set to SIR of the SMCS, the counter is initialized, and the system stops. To resume operation from the stop state, write "1" to STRT.

■ **Serial data register R/W standby**

When transfer is completed while the MODE bit is "1," "0" is set to BUSY and "1" is set to SIR of the SMCS, and the system enters the serial data register R/W standby state. If the interrupt enable flag is set, an interrupt signal is output from this block.

To resume operation from R/W standby state, read or write to the serial data register. This sets the BUSY bit to "1" and starts data transfer.

■ **Transfer**

"1" is set to the BUSY bit and serial transfer is being performed. According to the MODE bit, the halt state or R/W standby state comes next.

The figures below are diagrams of the operation transitions.

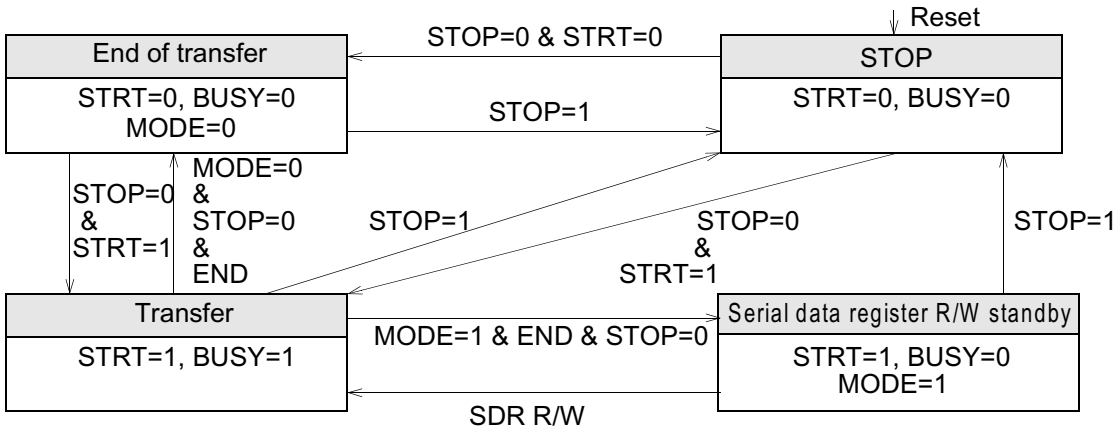


Figure 23.5.3a Extended I/O serial interface operation transitions

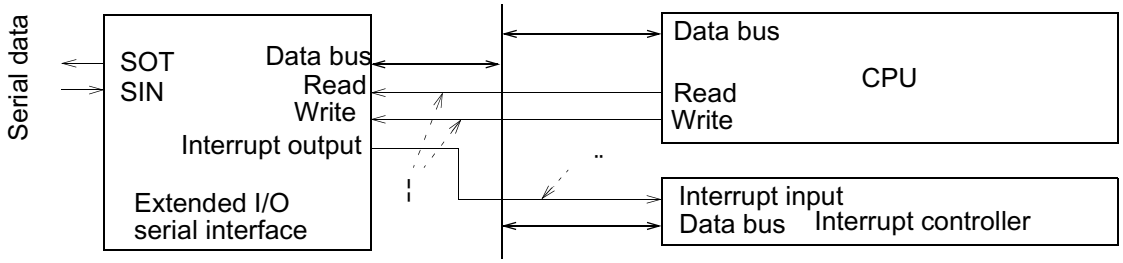


Figure 23.5.3b Serial data register read/write

- If "1" is written to MODE, transfer ends according to the shift clock counter. The read/write standby state starts when "1" is written to SIR. If "1" is written to the SIE bit, an interrupt signal is generated. No interrupt signal is generated when SIE is inactive or transfer has been terminated by writing "1" to STOP.
- ¦Reading or writing to the serial data register clears the interrupt request and starts serial transfer.

23.5.4 Shift Operation Start/Stop Timing and I/O Timing

Start: Write "0" to the STOP bit and "1" to the STRT bit of SMCS.

Stop: The system may stop at the end of transfer or when "1" is written to STOP.

Stop by STOP=1 ⇨ The system stops with SIR=0 regardless of the MODE bit.

Stop by end of transfer ⇨ The system stops with SIR=1 regardless of the MODE bit.

Regardless of the MODE bit, the BUSY bit becomes "1" during serial transfer and becomes "0" during stop or R/W standby state. To check the transfer status, read this bit.

(a) Internal shift clock mode (LSB first)

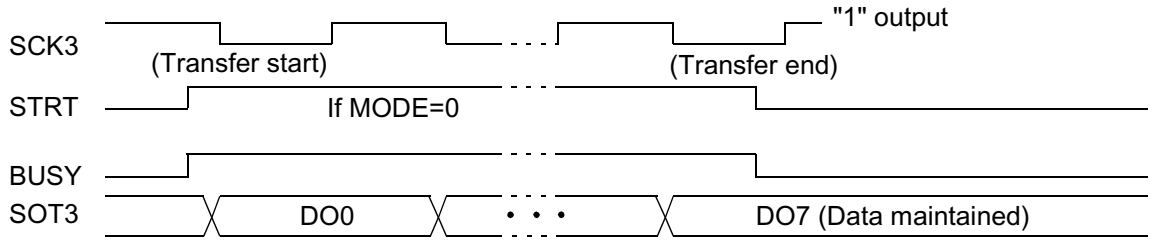


Figure 23.5.4a Shift operation start/stop timing (internal clock)

(b)-1 External shift clock mode (LSB first)

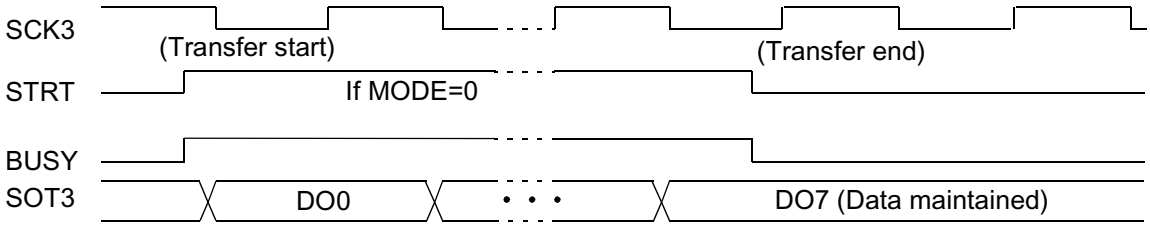


Figure 23.5.4b Shift operation start/stop timing (external clock)

(b)-2 External shift clock mode with instruction shift (LSB first)

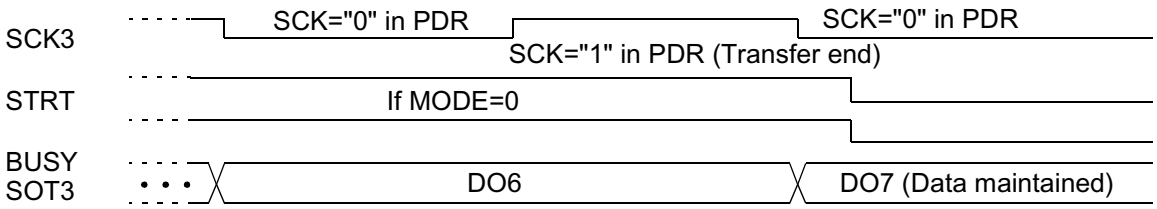


Figure 23.5.4c Shift operation start/stop timing

External shift clock mode with instruction shift:

* For an instruction shift, "H" is output when "1" is written to the bit corresponding to SCK of PDR, and "L" is output when "0" is written. (When SCOE=0 in external shift clock mode)

c) Stop by STOP=1 (LSB first, internal clock)

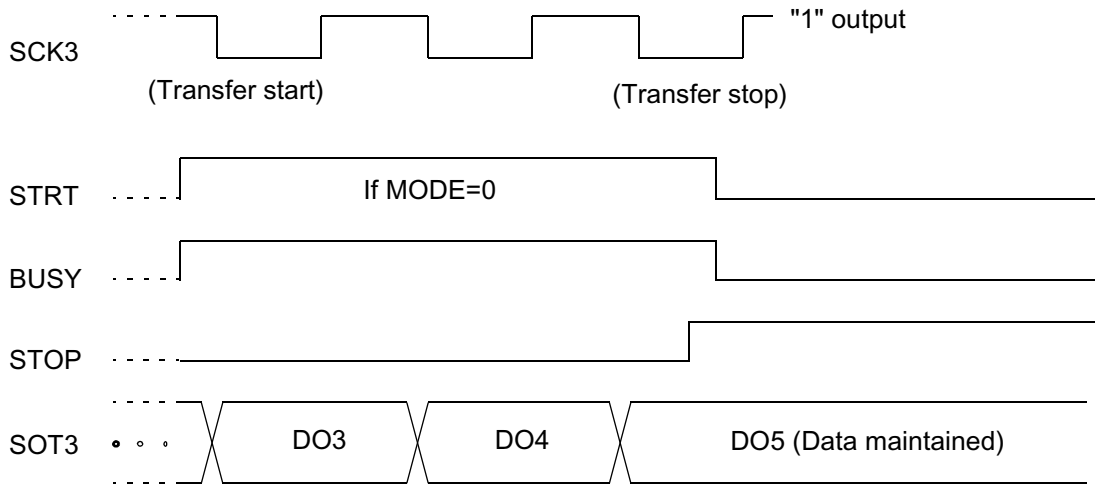
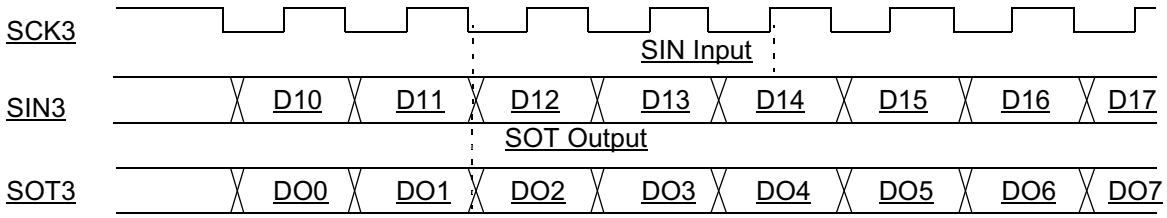


Figure 23.5.4d Stop timing when "1" is written to the STOP bit

<Note>
DO7 to DO0 indicate output data.

During serial data transfer, data is output from the serial output pin (SOT2) at the falling edge of the shift clock, and input from the serial input pin (SIN) at the rising edge.

m LSB first (When the BDS bit is "0")



m MSB first (When the BDS bit is "1")

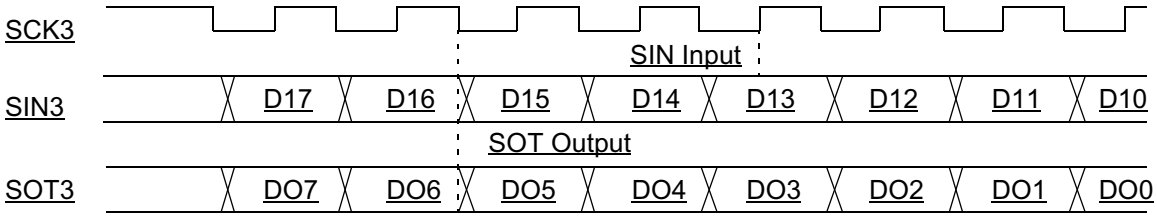


Figure 23.5.4e I/O shift timing

23.5.5 SPI-Timing

23.5.6 Interrupt Function

This block can issue an interrupt request to the CPU. At the end of data transfer, the SIR bit is set as an interrupt flag. When "1" is written to the interrupt enable bit (SIE bit) of SMCS, an interrupt request is issued to the CPU.

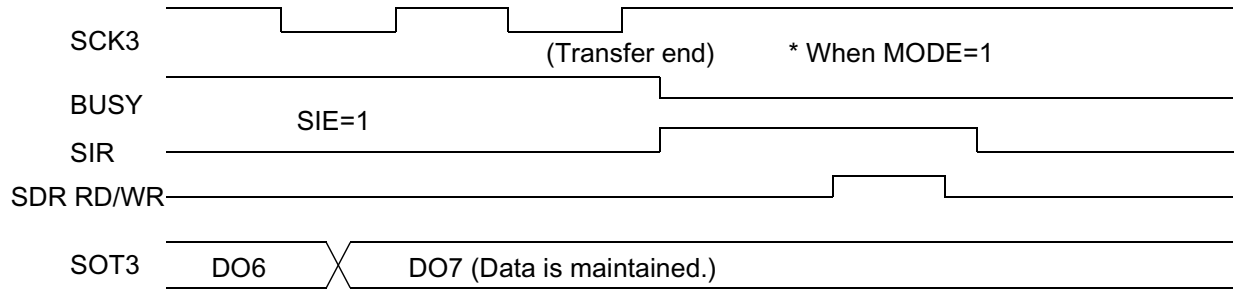


Figure 23.5.6a Interrupt signal output timing

23.5.7 Negative Clock Operation

The MB91360 series supports the negative clock operation of the Serial I/O. In this operation, the shift clock signal is simply negated by a inverter. Therefore the definition of the shift clock signal in the preceeding sections of the Serial I/O is inverted from the logic low level to logic high level, from the negative edge to the positive edge and vise-versa. This is the same for both the serial clock input and output.

The Edge Selector register is prepared for this purpose.

SES
Address : 000086H

7	6	5	4	3	2	1	0
—	—	—	—	—	—	DBL	NEG
						R/W	R/W

Initial value
------00B

Setting the NEG bit:

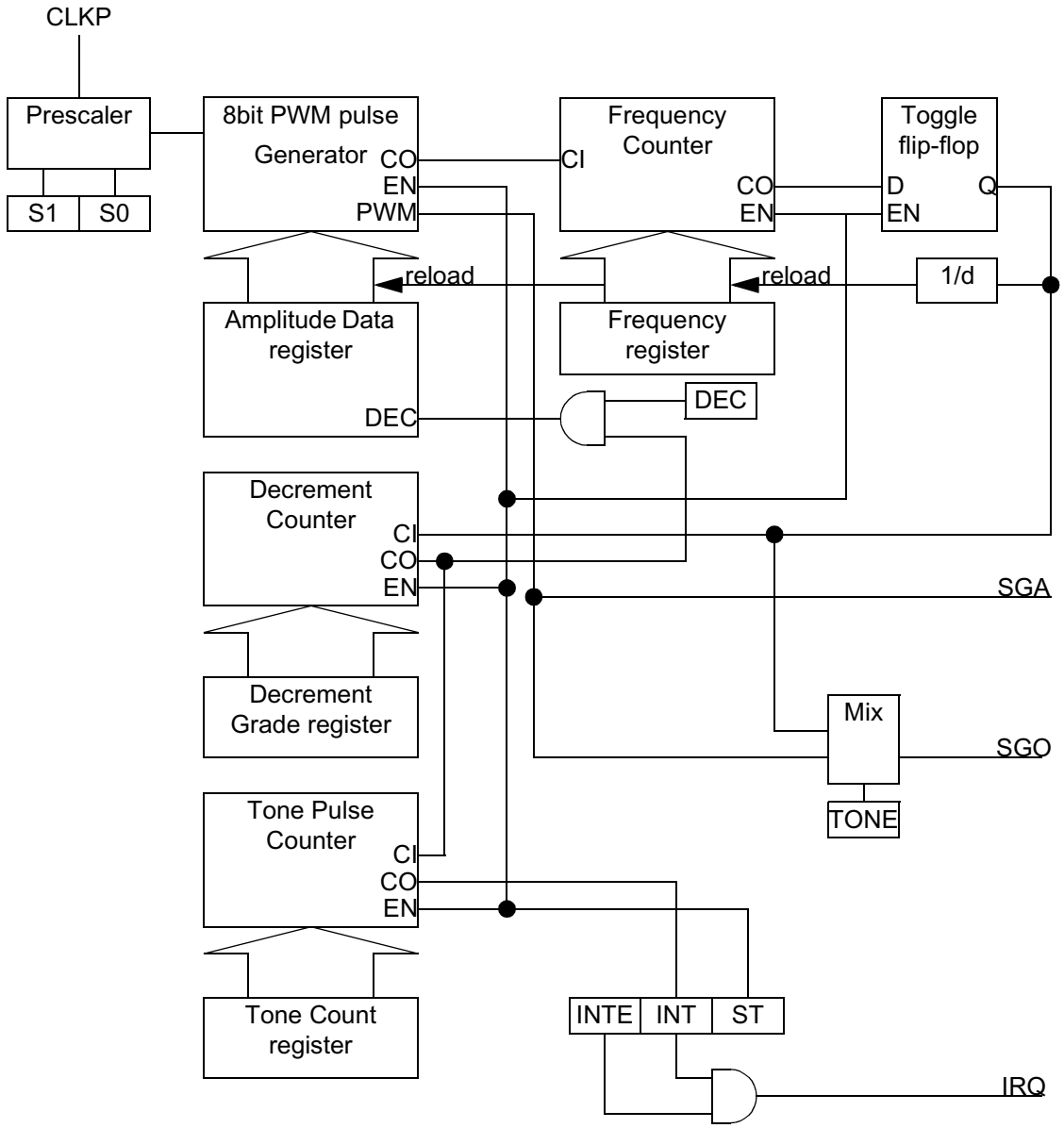
NEG	Operation
0	Normal operation [default]
1	The shift clock signal is inverted

CHAPTER 24 SOUND GENERATOR

This Chapter provides an overview of the Sound Generator, describes the register structure and functions, and describe the operation of the Sound Generator.

The Sound Generator consists of the Sound Control register, Frequency Data register, Amplitude Data register, Decrement Grade register, Tone Count register, Sound Disable register, PWM pulse generator, Frequency counter, Decrement counter and Tone Pulse counter.

24.1 BLOCK DIAGRAM



24.2 REGISTERS

Sound Control register
Address: 0000EF$_H$

Bit number	7	6	5	4	3	2	1	0
SGCR	S1	S0	TONE	—	—	INTE	INT	ST
Read/write	(R/W)	(R/W)	(R/W)	—	—	(R/W)	(R/W)	(R/W)
Initial value	(0)	(0)	(0)	—	—	(0)	(0)	(0)

Address: 0000EE$_H$

Bit number	15	14	13	12	11	10	9	8
SGCR	TST	—	—	—	—	—	BUSY	DEC
Read/write	(R/W)	—	—	—	—	—	(R)	(R/W)
Initial value	(0)	—	—	—	—	—	(0)	(0)

Frequency Data register
Address: 0000F1$_H$

Bit number	7	6	5	4	3	2	1	0
SGFR	D7	D6	D5	D4	D3	D2	D1	D0
Read/write	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)

Amplitude Data register
Address: 0000F0$_H$

Bit number	15	14	13	12	11	10	9	8
SGAR	D7	D6	D5	D4	D3	D2	D1	D0
Read/write	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

Decrement Grade register
Address: 0000F3$_H$

Bit number	7	6	5	4	3	2	1	0
SGDR	D7	D6	D5	D4	D3	D2	D1	D0
Read/write	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)

Tone Count register
Address: 0000F2$_H$

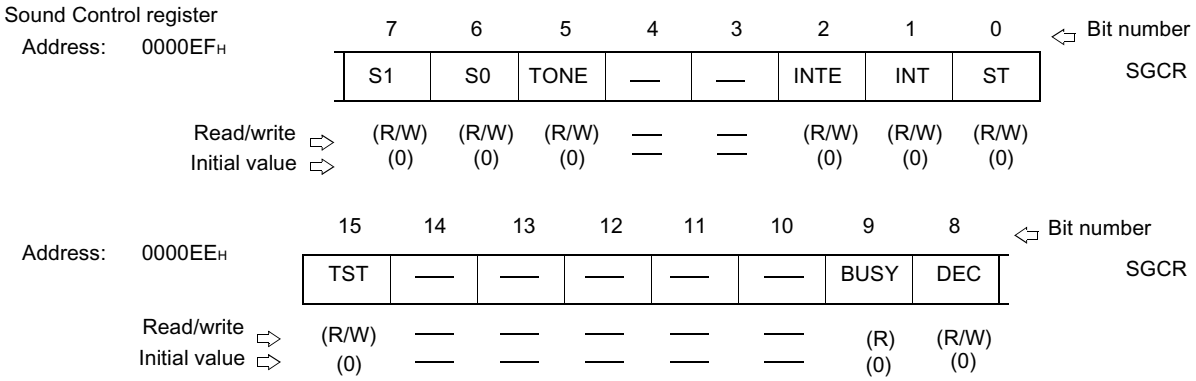
Bit number	15	14	13	12	11	10	9	8
SGTR	D7	D6	D5	D4	D3	D2	D1	D0
Read/write	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)

Sound Disable register
Address: 0000ED$_H$

Bit number	7	6	5	4	3	2	1	0
SGDBL	—	—	—	—	—	—	—	DBL
Read/write	—	—	—	—	—	—	—	(R/W)
Initial value	—	—	—	—	—	—	—	(0)

24.3 REGISTER DETAILS

24.3.1 Sound Control Register (SGCR)



[bit 15] TST : Test bit

This bit is prepared for the device test. In any user applications, it should be set to "0".

[bit 9] BUSY : Busy bit

This bit indicates whether the Sound Generator is in operation. This bit is set to "1" upon the ST bit is set to "1". It is reset to "0" when the ST bit is reset to "0" and the operation is completed at the end of one tone cycle. Any write instructions performed on this bit has no effect

[bit 8] DEC : Auto-decrement enable bit

The DEC bit is prepared for an automatic de-gradation of the sound in conjunction with the Decrement Grade register.

If this bit is set to "1", the stored value in the Amplitude Data register is decremented by 1(one), every time when the Decrement counter counts the number of tone pulses from the toggle flip-flop specified by the Decrement Grade register.

[bits 7 to 6] S1 to S0 : Operation clock select bits

These bits specify the clock input signal for the Sound Generator.

S1	S0	Clock input
0	0	CLKP
0	1	1/2 CLKP
1	0	1/4 CLKP
1	1	1/8 CLKP

[bit 5] TONE : Tone output bit

When this bit is set to "1", the SGO signal becomes a simple square-waveform (tone pulses) from the toggle flip-flop. Otherwise it is the mixed (AND logic) signal of the tone and PWM pulses.

[bit 2] INTE : Interrupt enable bit

This bit enables the interrupt signal of the Sound Generator. When this bit is "1" and the INT bit is set to "1", the Sound Generator signals an interrupt.

[bit 1] INT : Interrupt bit

This bit is set to "1" when the Tone Pulse counter counts the number of the tone pulses specified by the Tone Count register and Decrement Grade register.

This bit is reset to "0" by writing "0". Writing "1" has no effect and Read-Modify-Write instructions always result in reading "1".

[bit 0] ST : Start bit

This bit is for starting the operation of the Sound Generator. While this bit is "1", the Sound Generator perform its operation.

When this bit is reset to "0", the Sound Generator stops its operation at the end of the current tone cycle. The BUSY bit indicates whether the Sound Generator is fully stopped.

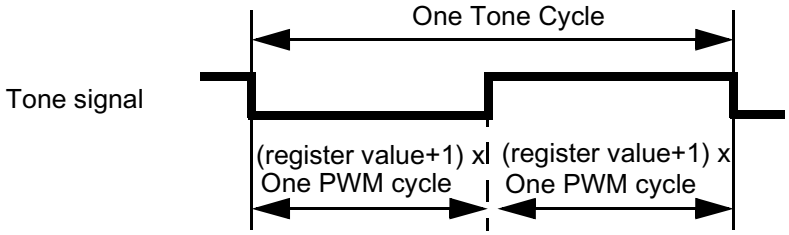
24.3.2 Frequency Data Register (SGFR)

Frequency Data register
Address: 0000F1H

	7	6	5	4	3	2	1	0	⇦ Bit number
	D7	D6	D5	D4	D3	D2	D1	D0	SGFR
Read/write ⇨	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Initial value ⇨	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	

The Frequency Data register stores the reload value for the Frequency counter. The stored value represents the frequency of the sound (or the tone signal from the toggle flip-flop). The register value is reloaded into the counter at every transition of the toggle signal.

The following figure shows the relationship between the tone signal and the register value.



It should be noted that modifications of the register value while operation may alter the duty cycle of 50% depending on the timing of the modification.

24.3.3 Amplitude Data Register (SAGR)

Amplitude Data register
Address: 0000F0H

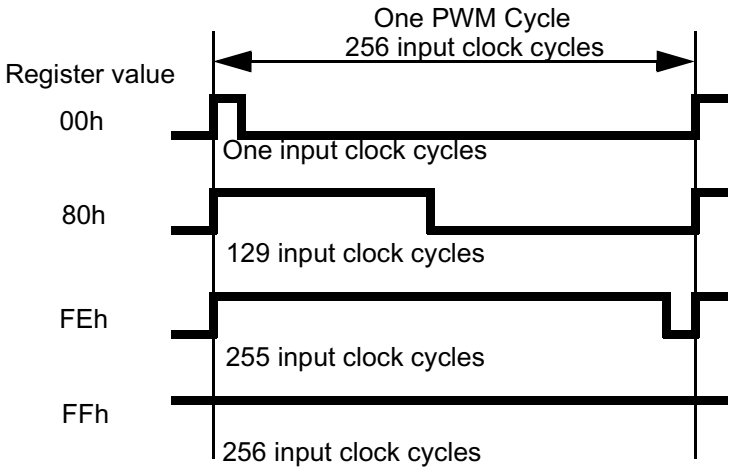
	15	14	13	12	11	10	9	8	⇦ Bit number
	D7	D6	D5	D4	D3	D2	D1	D0	SGAR
Read/write ⇨	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Initial value ⇨	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

The Amplitude Data register stores the reload value for the PWM pulse generator. The register value represents the amplitude of the sound. The register value is reloaded into the PWM pulse generator at the end of every tone cycle.

When the DEC bit is "1" and the Decrement counter reaches its reload value, this register value is decremented by 1(one). And when the register value reaches "00", further decrements are

not performed. However the sound generator continues its operation until the ST bit is cleared.

The following figure shows the relationship between the register value and the PWM pulse.



When the register value is set to "FF", the PWM signal is always "1".

24.3.4 Decrement Grade Register (SGDR)

Decrement Grade register

Address: 0000F3H

	7	6	5	4	3	2	1	0	
	D7	D6	D5	D4	D3	D2	D1	D0	⇦ Bit number SGDR
Read/write ⇨	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Initial value ⇨	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	

The Decrement Grade register stores the reload value for the Decrement counter. They are prepared to automatically decrement the stored value in the Amplitude Data register.

When the DEC bit is "1" and the Decrement counter counts the number of tone pulses up to the reload value, the stored value in the Amplitude Data register is decremented by 1 (one) at the end of the tone cycle.

This operation realizes automatic de-gradation of the sound with fewer number of CPU interventions.

It should be noted that the number of the tone pulses specified by this register equals to "register value +1". When the Decrement Grade register is set to "00", the decrement operation is performed every tone cycle.

24.3.5 Tone Count Register (SGTR)

Tone Count register

Address: 0000F2H

	15	14	13	12	11	10	9	8	
	D7	D6	D5	D4	D3	D2	D1	D0	⇦ Bit number SGTR
Read/write ⇨	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Initial value ⇨	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	

The Tone Count register stores the reload value for the Tone Pulse counter. The Tone Pulse counter accumulate the number of tone pulses (or number of decrement operations) and when it reaches the reload value it sets the INT bit. They are intended to reduce the frequency of interrupts.

The count input of the Tone Pulse counter is connected to the carry-out signal from the Decrement counter. And when the Tone count register is set to "00", the Tone Pulse counter

sets the INT bit every carry-out from the Decrement counter. Thus the number of accumulated tone pulses is:
((Decrement Grade register) +1) x ((Tone Count register) +1)

i.e. When the both registers are set to "00", the INT bit is set every tone cycle.

24.3.6 Sound Disable Register (SGDBL)

Sound Disable register

Address: 0000ED$_H$

	7	6	5	4	3	2	1	0
Bit number								SGDBL
	—	—	—	—	—	—	—	DBL
Read/write	—	—	—	—	—	—	—	(R/W)
Initial value	—	—	—	—	—	—	—	(0)

This bit is used to control the clock for the Sound Generator.

When "1" is written to this bit, the clock for the Sound Generator module is disabled.

When "0" is set, a clock is supplied to the Sound Generator module.

This bit is initialized to "0" upon reset. This bit is readable and writable.

CHAPTER 25 STEPPER MOTOR CONTROLLER

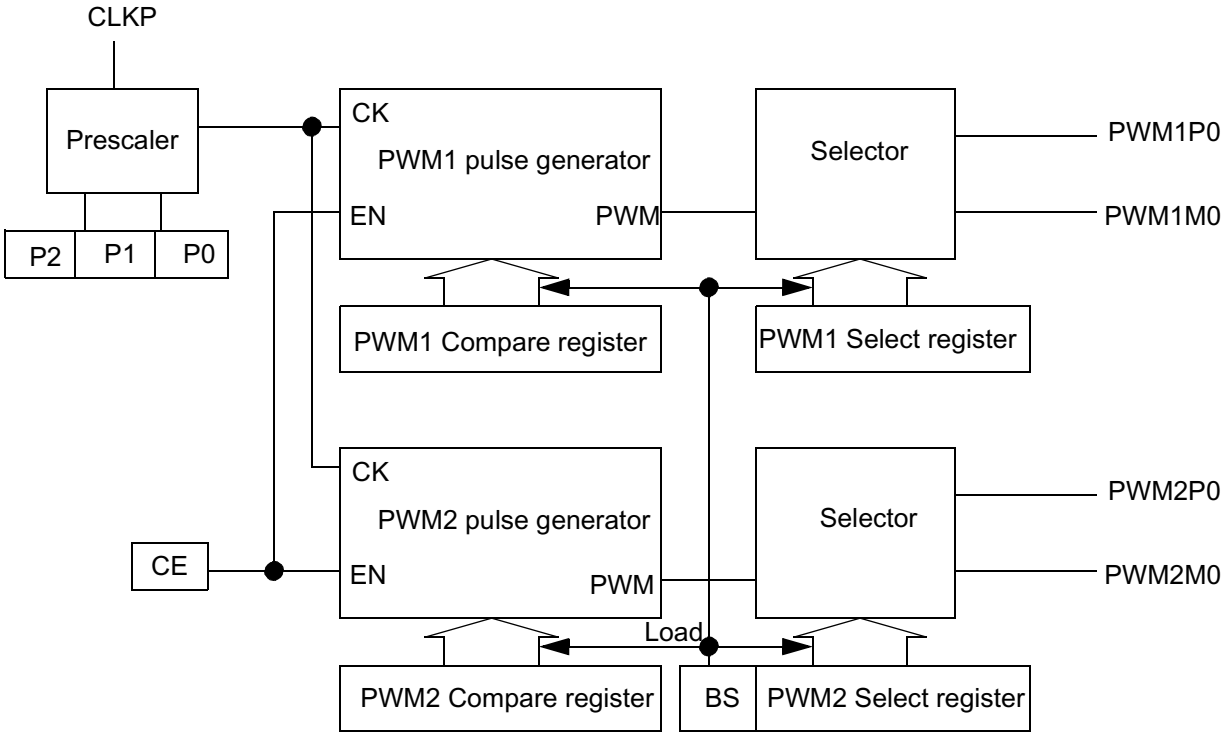
This Chapter provides an overview of the Stepper Motor Control Module, describe the register structure and functions, and described the operation of the Stepper Motor Control Module.

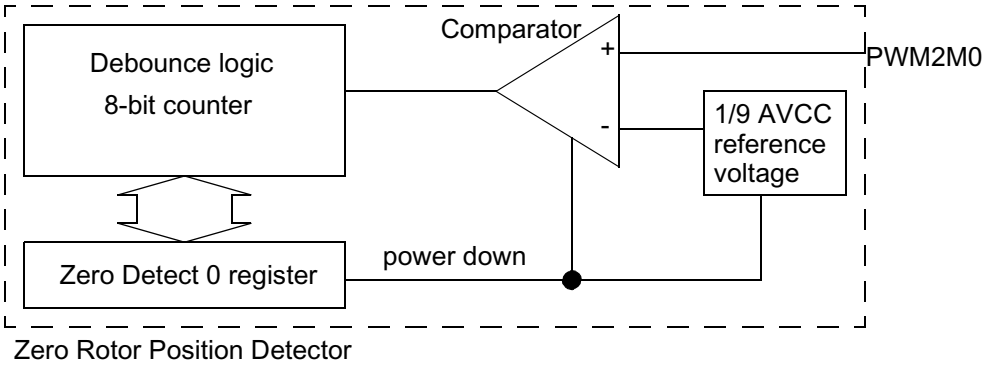
25.1 OVERVIEW

The Stepping Motor Controller consists of two PWM Pulse Generators, four motor drivers, Selector Logic and the Zero Rotor Position Detector. The four motor drivers have high output drive capabilities and they can be directly connected to the four ends of two motor coils. The combination of the PWM Pulse Generators and Selector Logic is designed to control the rotation of the motor. A Synchronization mechanism assures the synchronous operations of the two PWMs. The Zero Rotor Position Detector helps CPU obtain feed back information of the rotor movements. The following sections describe the Stepping Motor Controller 0 only. The other controllers have the same functions. The register addresses are found in the I/O map.

Note: The Rotor Zero Position Detection capability is protected by a patent from Siemens VDO Automotive AG and may only be used with VDO's prior approval.

25.2 BLOCK DIAGRAM





25.3 REGISTERS

PWM Control 0 register
Address: 0000D1H

Bit number	7	6	5	4	3	2	1	0
PWC0	S2	P2	P1	P0	CE	—	—	TST
Read/write	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	—	—	(R/W)
Initial value	(0)	(0)	(0)	(0)	(0)	—	—	(0)

Zero Detect 0 register
Address: 0000D0H

Bit number	15	14	13	12	11	10	9	8
ZPD0	S1	S0	TS	T2	T1	T0	PD	RS
Read/write	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R)
Initial value	(0)	(0)	(0)	(0)	(0)	(0)	(1)	(0)

PWM1 Compare 0 register
Address: 0000D9H

Bit number	7	6	5	4	3	2	1	0
PWC10	D7	D6	D5	D4	D3	D2	D1	D0
Read/write	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)

PWM2 Compare 0 register
Address: 0000D8H

Bit number	15	14	13	12	11	10	9	8
PWC20	D7	D6	D5	D4	D3	D2	D1	D0
Read/write	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)

PWM1 Select register
Address: 0000DBH

Bit number	7	6	5	4	3	2	1	0
PWS10	—	—	P2	P1	P0	M2	M1	M0
Read/write	—	—	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value	—	—	(0)	(0)	(0)	(0)	(0)	(0)

PWM2 Select register
Address: 0000DAH

Bit number	15	14	13	12	11	10	9	8
PWS20	—	BS	P2	P1	P0	M2	M1	M0
Read/write	—	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value	—	(0)	(0)	(0)	(0)	(0)	(0)	(0)

PWM Clock Disable register
Address: 0000E8H

Bit number	15	14	13	12	11	10	9	8
SMDBL0	—	—	—	—	—	—	—	DBL
Read/write	—	—	—	—	—	—	—	(R/W)
Initial value	—	—	—	—	—	—	—	(0)

25.4 REGISTER DETAILS

25.4.1 PWM Control 0 Register (PWC0)

PWM Control 0 register
Address: 0000D1H

Bit number	7	6	5	4	3	2	1	0
PWC0	S2	P2	P1	P0	CE	—	—	TST
Read/write	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	—	—	(R/W)
Initial value	(0)	(0)	(0)	(0)	(0)	—	—	(0)

[bit 7] S2: Debounce clock select bit 2, see 25.4.2 "Zero Detect 0 register (ZPD0)" on page 556

[bits 6 to 4] P2 to P0 : Operation clock select bits

These bits specify the clock input signal for the PWM pulse generators.

P2	P1	P0	Clock input
0	0	0	Peripheral clock (CLKP)
0	0	1	1/2 CLKP
0	1	0	1/4 CLKP
0	1	1	1/8 CLKP
1	0	0	Reserved
1	0	1	1/5 CLKP
1	1	0	1/6 CLKP
1	1	1	Reserved

[bits 3] CE : Count enable bit

This bit enables the operation of the PWM pulse generators. When it is set to "1", the PWM pulse generators start their operation. Note that the PWM2 pulse generator starts the operation one CLKP cycle after the PWM1 pulse generators is started. This is to help reduce the switching noise from the output drivers.

[bits 0] TST : Test bit

This bit is for the device test. In user applications, it should always be set to "0".

25.4.2 Zero Detect 0 register (ZPD0)

Zero Detect 0 register
Address: 0000D0H

15	14	13	12	11	10	9	8	⇦ Bit number
S1	S0	TS	T2	T1	T0	PD	RS	ZPD0
(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R)	⇨ Read/write
(0)	(0)	(0)	(0)	(0)	(0)	(1)	(0)	⇨ Initial value

[bits 15 to 14] S1 to S0 : Debounce clock select bit

These bits specify the clock frequency used for the Debounce logic. The Debounce logic samples the output of the comparator with the specified clock frequency.

S2	S1	S0	Clock input
0	0	0	Peripheral clock (CLKP)
0	0	1	1/2 CLKP
0	1	0	1/4 CLKP
0	1	1	1/8 CLKP
1	0	0	Reserved
1	0	1	1/5 CLKP
1	1	0	1/6 CLKP
1	1	1	Reserved

Note: The bit S2 is located in the PWM Control 0 Register.

[bit 13] TS : Time slice bit

This bit enables the operation of the Zero Rotor Position Detector. While this bit is "1", the Zero Rotor Position Detector compares the input voltage at the PWM2M0 pin with the reference voltage and sets the RS bit if the input voltage exceed the reference voltage.*

* For the comparator a settling time of 3 μs should be allowed.

[bits 12 to 10] T2 to T0 : Number of samples

These bits specifies the number of samples for the Debounce logic. The Debounce logic samples the output of the comparator the specified number of times. The output of the Debounce logic becomes "1" when all the sampled values are "1".

T2	T1	T0	Number of samples
0	0	0	1
0	0	1	2
0	1	0	3
0	1	1	4
1	X	X	5

[bit 9] PD : Power down bit

When this bit is set to "1", the power supply to the analog components (comparator and reference voltage source) is switched off.

[bit 8] RS : Result bit

The RS bit indicates whether the input voltage at the PWM2M0 pin exceeded the reference voltage.

The RS bit is set to "1" if the output of the Debounce logic becomes "1". While TS bit is "0", the RS bit always indicates "0".

25.4.3 PWM1&2 Compare Registers (PWC10, PWC20)

PWM1 Compare 0 register

Address: 0000D9H

	7	6	5	4	3	2	1	0	
	D7	D6	D5	D4	D3	D2	D1	D0	PWC10
Read/write	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Initial value	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	

Bit number

PWM2 Compare 0 register

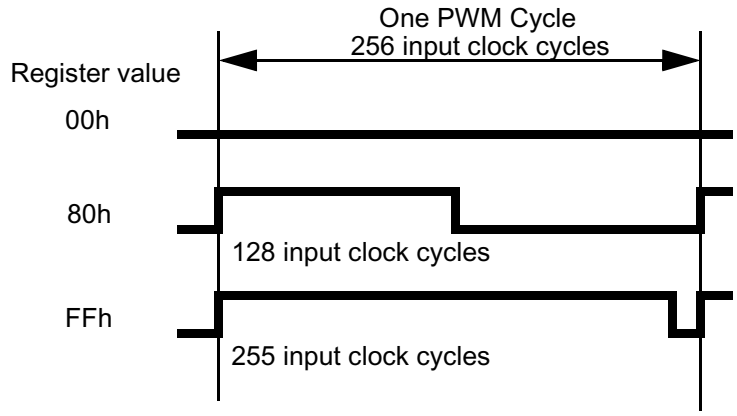
Address: 0000D8H

	15	14	13	12	11	10	9	8	
	D7	D6	D5	D4	D3	D2	D1	D0	PWC20
Read/write	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Initial value	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)	

Bit number

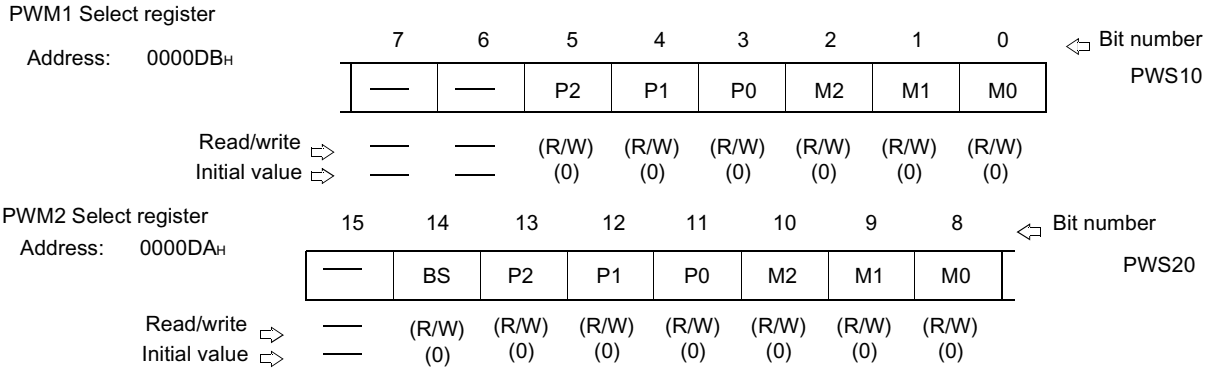
The contents of the two 8-bit compare registers determine the widths of PWM pulses.

The stored value of "00H" represents the PWM duty of 0% and "FFH" represents the duty of 99.6%.



These registers are accessible at any time, however the modified values are reflected to the pulse width at the end of the current PWM cycle after the BS bit of the PWM2 Select register is set to "1".

25.4.4 PWM1&2 Select registers (PWS10, PWS20)



[bit 14] BS : Update bit

This bit is prepared to synchronise the settings for the PWM outputs. Any modifications in the two compare registers and two select registers are not reflected to the output signals until this bit is set.

When this bit is set to "1", the PWM pulse generators and selectors load the register contents at the end of the current PWM cycle. The BS bit is reset to "0" automatically at the beginning of the next PWM cycle. If the BS bit is set to "1" by software at the same time as this automatic reset, the BS bit is set to "1" (or remains unchanged) and the automatic reset is cancelled.

[bits 13 to 11] P2 to P0 : Output Select bits

These bits selects the output signal at PWM2P0.

[bits 10 to 8] M2 to M0 : Output Select bits

These bits selects the output signal at PWM2M0.

[bits 5 to 3] P2 to P0 : Output Select bits

These bits selects the output signal at PWM1P0.

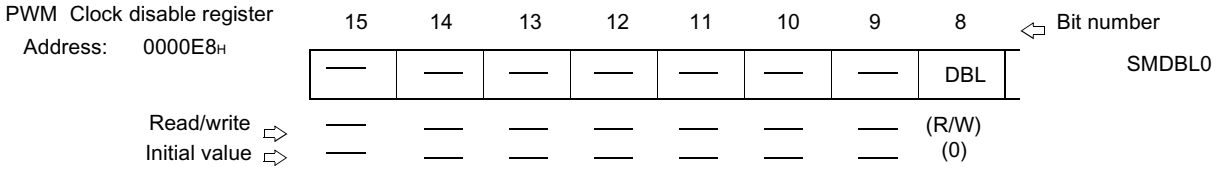
[bits 2 to 0] M2 to M0 : Output Select bits

These bits selects the output signal at PWM1M0.

The following table shows the relationship between the output levels and select bits.

P2	P1	P0	PWMnP0	M2	M1	M0	PWMnM0
0	0	0	L	0	0	0	L
0	0	1	H	0	0	1	H
0	1	X	PWM pulses	0	1	X	PWM pulses
1	X	X	High impedance	1	X	X	High impedance

25.4.5 PWM Clock Disable Register (SMDBL)



[bit 8] DBL : Clock disable bit

When this bit is set to "1", the clock for the SMC module is disabled. If set to "0" a clock is supplied to the SMC module. This bit is initialized to "0". This bit is readable and writable

Remark: For SMC1 and SMC3 this bit is at bit position 0.

CHAPTER 26 U-TIMER

The U-timer (U-TIMER) is a 16-bit timer used to generate the baud rate for the UART. This chapter provides an overview of the U-timer, describes the register structure and functions, and describes the operation of the U-timer.

26.1 OVERVIEW OF THE U-TIMER

The U-timer (U-TIMER) is a 16-bit timer used to generate the baud rate for the UART. The operating frequency of the chip and the U-TIMER reload value can be combined to set a user-defined baud rate. As the timer can generate a count underflow interrupt, the U-TIMER can also be used as an interval timer.
The MB91360 contains three U-TIMER channels. The intervaltimers can count for a maximum of $2^{16} \times$ CLKP.

■ U-TIMER Block Diagram

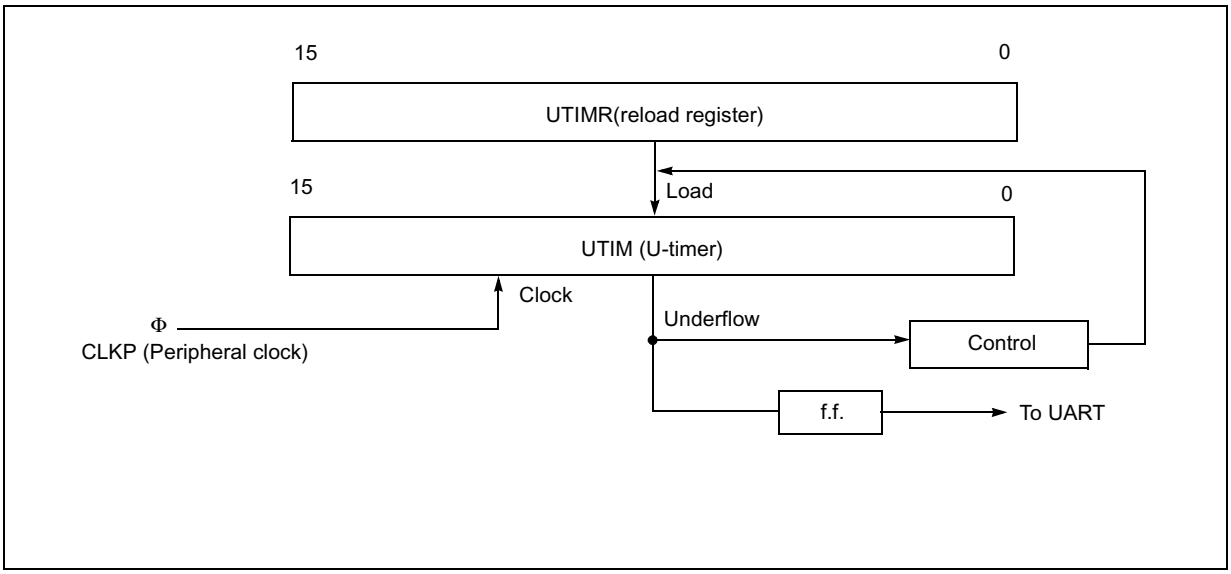


Figure 26.1 U-timer Block Diagram

26.2 U-TIMER REGISTERS

This section describes the three U-TIMER registers listed below.

- **U-timer register (UTIM)**
- **Reload register (UTIMR)**
- **U-TIMER control register (UTIMC)**

■ **Register Configuration of the U-timer (U-TIMER)**

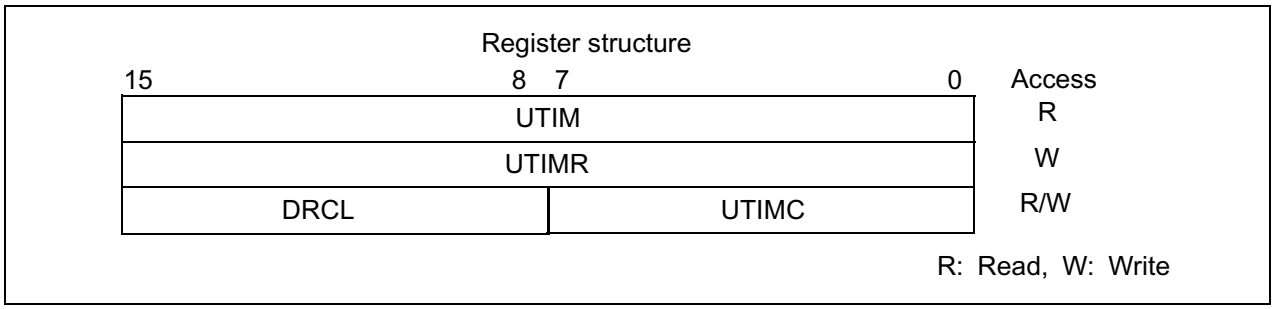


Figure 26.2a Register Configuration of the U-timer (U-TIMER)

26.2.2 U-Timer Register (UTIM)

UTIM holds the timer value. Access using 16-bit transfer instructions.

UTIM	Address	Bits 15	14	...	2	1	0	Initial value	Access
0-ch	0000 0068H	b15	b14		b2	b1	b0	0	R
1-ch	0000 0074H								
2-ch	0000 0080H								

Figure 26.2.2a Structure of the U-timer Register

26.2.3 U-Timer Reload Register (UTIMR)

The UTIMR register stores the reload value loaded to UTIM when UTIM underflows. Access using 16 bit transfer instructions.

UTIMR	Address	Bits 15	14	...	2	1	0	Initial value	Access
0-ch	0000 0068H	b15	b14		b2	b1	b0	0	W
1-ch	0000 0074H								
2-ch	0000 0080H								

Figure 26.2.3a Structure of the U-timer Reload Register

If a valid boot condition is detected during execution of the code in the internal boot ROM the UTIMR0 register is set to "0x05".

26.2.4 U-Timer Control Register (UTIMC)

UTIMC controls the U-TIMER operation.

■ **Structure of the U-timer control register**

UTIMC	Address	7	6	5	4	3	2	1	0	Initial value	Access
0-ch	0000 006B$_H$	UCC1	–	–	–	UNDR	Reserved	UTST	UTCR	0---0001	R/W
1-ch	0000 0077$_H$										
2-ch	0000 0083$_H$										

Figure 26.2.4a Structure of the U-timer Control Register

If a valid boot condition is detected during execution of the code in the internal boot ROM the UTIMC0 register is set to "0x82".

■ **Functions of the UTIMC bits**

[Bit 7] UCC1(U-timer Count Control 1): U-timer count control 1

UCC1 controls how U-TIMER counts.

UCC1	Operation
0	Normal operation $\alpha = 2n + 2$ [Initial value]
1	+1 mode $\alpha = 2n + 3$

n : UTIMR setting value
α : Cycle of the output clock to the UART

In addition to outputting the standard 2(n+1) cycle clock to the UART, an odd-numbered divide ratio can also be set for the U-TIMER.

Setting UCC1 to "1" generates a clock with a cycle "2n + 3".

Example settings: (1) UTIMR = "5", UCC1 = "0" $\rightarrow$ Clock cycle = 2n + 2 = 12 cycles

(2) UTIMR = "25", UCC1 = "1" $\rightarrow$ Clock cycle = 2n + 3 = 53 cycles

(3) UTIMR = "60", UCC1 = "0" $\rightarrow$ Clock cycle = 2n + 2 = 122 cycles

Set UCC1 value to "0" when using U-TIMER as an interval timer.

[Bits 6, 5 and 4] Reserved

Always set these bits to "000" when writing to this register.

[Bit 3] UNDR (UNDeR flow flag): Underflow flag

UNDR is a flag that indicates when an underflow occurs. An underflow interrupt is generated if UNDR is set when UTIE is "1". UNDR is cleared by a reset or by writing "0".

The bit is always read as "1" by read-modify-write instructions.

Writing "1" to UNDR has no meaning.

[Bit 2] Reserved

Always set this bit to "0".

[Bit 1] UTST(U-TIMER STart): STart U-timer

Operation enable bit for the U-TIMER.

0	Halted: Writing "0" during operation halts the timer. [Initial value]
1	Operating: Operation continues if "1" is written during operation.

[Bit 0] UTCR(U-TIMER CleaR): Clear U-timer

Writing "0" to UTCR clears the U-TIMER to 0000H.

The underflow flipflop (f.f. in the block diagram) is also cleared to "0".

The bit is always read as "1".

Precautions:

(1) The counter is automatically reloaded if the start bit (UTST) is set when the timer is halted.

(2) Setting the clear bit (UTCR) and start bit (UTST) at the same time when the timer is halted clears the counter to zero and generates an underflow at the next down-count.

(3) Setting the clear bit (UTCR) when the timer is operating clears the counter to zero. This may cause a short spike pulse on the output waveform and result in malfunction of the UART. If using the output clock, do not clear the counter using the clear bit while the timer is operating.

26.2.5 DMA Interrupt Clear Register (DRCL)

This register is used for preparing DMA transfers.

■ **Structure of the DRCL register**

DRCL	Address	15	14	13	12	11	10	9	8	Initial value	Access
0-ch	0000 006AH	--	–	–	--	--	--	--	--	--------	W
1-ch	0000 0076H										
2-ch	0000 0082H										

Figure 26.2.5a Structure of the U-timer DRCLRegister

■ **Function of the DRCL Register**

Write once to this register before you use DMA for the first time.

26.3 U-TIMER OPERATION

This section describes the baud rate calculation of the U-Timer.

26.3.1 Baud Rate Calculation

The UART uses the underflow flipflop (f.f in the diagram) of the corresponding U-TIMER (U-TIMER0 for UART0, U-TIMER1 for UART1, and U-TIMER2 for UART2) as the clock source for the baud rate.

■ Asynchronous (start bit synchronization) mode

The UART uses the output of the U-TIMER divided by 16.

$$bps = \frac{\Phi}{(2n+2) \times 16} \quad \cdots\cdots\cdots \text{When UCC1 = "0"}$$

$$bps = \frac{\Phi}{(2n+3) \times 16} \quad \cdots\cdots\cdots \text{When UCC1 = "1"}$$

n: UTIMR (reload value)

Φ : Frequency of the peripheral clock CLKP (depends on the gear function)

■ Clock-synchronous mode

The clock-synchronous mode of the UARTs is not supported in the MB91360 series.

CHAPTER 27 UART

The UART is a serial I/O port for performing asynchronous (start bit synchronization) communications. This chapter provides an overview of the UART, describes the register structure and functions, and describes the operation of the UART.

27.1 OVERVIEW OF THE UART

The UART is a serial I/O port for performing asynchronous (start bit synchronization) communications. The MB91360 series contains up to three UART channels.

■ Features

- Full-duplex, double buffering
- Supports asynchronous (start bit synchronization) communications
- Supports multi-processor mode
- Fully programmable baud rate The baud rate can be set using an internal timer. (See the chapter 26 "U-TIMER" on page 561.)
- Supports flexible baud rate setting using an external clock
- Error detection function (parity, framing, overrun)
- Non return to zero (NRZ) transfer signal
- Supports DMA transfer activation using an interrupt

■ UART Block Diagram

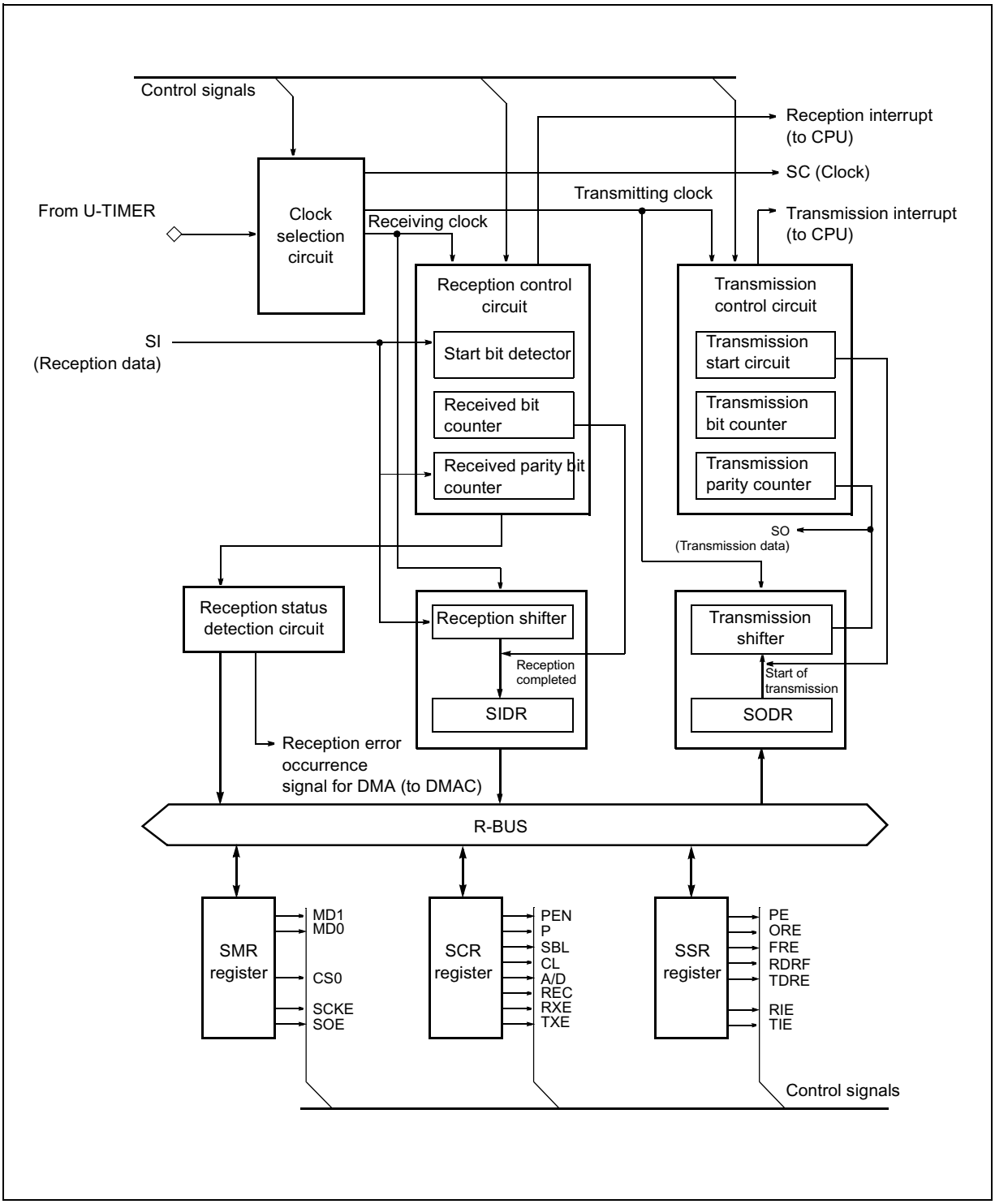


Figure 27.1 Overall Block Diagram of the UART

27.2 UART REGISTERS

This section lists the UART registers (Figure 27.2a) and describes the function of each register in detail.

■ Register Configuration of the UART

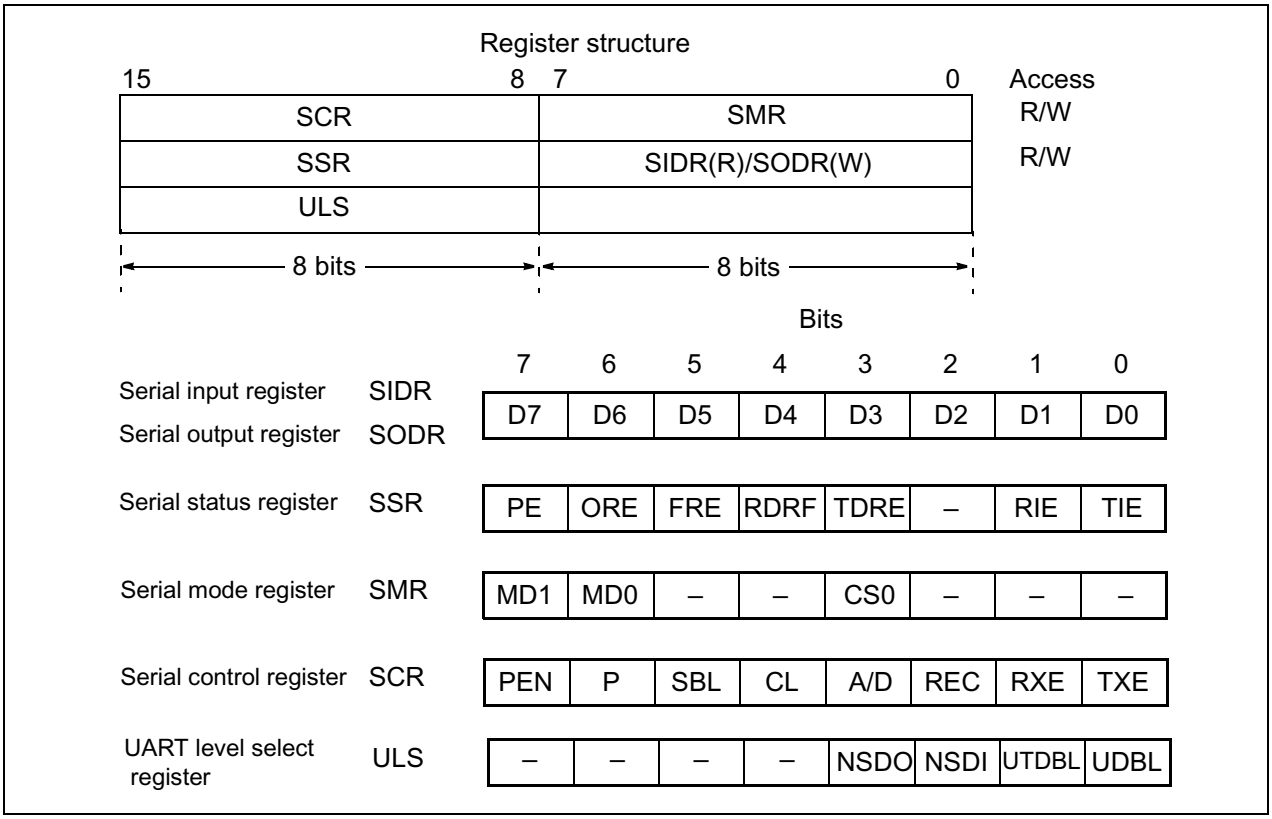


Figure 27.2a Register Configuration of the UART

27.2.2 Serial Mode Register (SMR)

The serial mode register (SMR) specifies the UART operation mode.
Set the operation mode when the UART is halted. Do not write to this register during UART operation.

■ Structure of the Serial Mode Register (SMR)

		Bits								
	Address	7	6	5	4	3	2	1	0	Initial value
SMR	0000 0063H 0000 006FH 0000 007BH	MD1	MD0	Reserved	Reserved	CS0	Reserved	Reserved	Reserved	00--0-00B
		R/W	R/W			W				⇦ Access

Figure 27.2.2a Structure of the Serial Mode Register

If a valid boot condition is detected during execution of the code in the internal boot ROM the SMR0 register is set to "0x31".

■ Functions of the SMR Bits

[Bits 7 and 6] MD1, MD0(MoDe select): Mode select

Selects the UART operation mode.

Mode	MD1	MD0	Operation mode
0	0	0	Asynchronous (start bit synchronization), normal mode [Initial value]
1	0	1	Asynchronous (start bit synchronization), multi-processor mode
2	1	0	CLK-synchronous mode (not supported by MB91360 series)
–	1	1	Setting not available

Precautions:

CLK asynchronous mode 1 (multi-processor mode) is used to connect a number of slave CPUs to a single host CPU. As the UART cannot determine the format of received data, the MB91360 only supports operation as the host (master) CPU in multi-processor mode.

Also, as the parity check function cannot be used, set the PEN bit of the SCR register to "0".

[Bits 5 and 4] Reserved

Always set to "1".

[Bit 3] CS0 (Clock Select): Clock select

Selects the UART operating clock.

0	Internal timer (U-TIMER) [Initial value]
1	External clock (not supported by MB91360)

[Bit 2, 1, 0] Reserved

Always set to "000". Bit 1 is SCKE (Serial Clock Enable), bit 0 is SOE (Serial Out Enable). Both functions are not supported on MB91360 series. To enable serial output, set the attached port function register bit.

27.2.3 Serial Control Register (SCR)

The serial control register (SCR) controls the transmission protocol used for serial communications.

■ Structure of the Serial Control Register (SCR)

		Bits								
	Address	7	6	5	4	3	2	1	0	Initial value
SCR	0000 0062H 0000 006EH	PEN	P	SBL	CL	A/D	REC	RXE	TXE	00000100B
	0000 007AH	R/W	R/W	R/W	R/W	R/W	W	R/W	R/W	⇦ Access

Figure 27.2.3a Structure of the Serial Control Register

If a valid boot condition is detected during execution of the code in the internal boot ROM the SCR0 register is set to "0x13".

■ Functions of the SCR Bits

[Bit 7] PEN (Parity ENable): Parity enable

Specifies whether or not to append a parity bit to the data when performing serial communications.

0	Do not use parity. [Initial value]
1	Use parity.

Precautions:

A parity bit can only be appended in normal mode (mode 0) of asynchronous (start bit synchronization) communications mode. A parity bit cannot be appended for multi-processor mode (mode 1).

[Bit 6] P: Parity

Specifies whether to use even/odd parity when appending a parity bit to the data when performing serial communications.

0	Even parity [Initial value]
1	Odd parity

[Bit 5] SBL (Stop Bit Length): Stop bit length

Specifies the number of stop bits to use as a frame end mark in asynchronous (start bit synchronization) communications mode.

0	1 stop bit	[Initial value]
1	2 stop bits	

[Bit 4] CL (Character Length): Data length

Specifies the number of data bits in a transmission or reception frame.

0	7 data bits	[Initial value]
1	8 data bits	

Precautions:

Only normal mode (mode 0) of asynchronous (start bit synchronization) communications mode can handle 7-bit data. Set 8 data bits when using multi-processor mode (mode 1).

[Bit 3] A/D (Address/Data): Address/data frame

Specifies the format of transmission or reception frames for multi-processor mode (mode 1) of asynchronous (start bit synchronization) communications.

0	Data frame	[Initial value]
1	Address frame	

[Bit 2] REC (Receiver Error Clear): Receiver error clear

Clears the error flags (PE, ORE, and FRE) in the SSR register.

0	Clear the error flags (PE, ORE, and FRE) in the SSR register.
1	Writing "1" is ignored. Reading always returns "1".

[Bit 1] RXE (Receiver Enable): Receiver enable

Controls UART reception.

0	Disable reception.	[Initial value]
1	Enable reception.	

Precautions:

If reception is disabled during a receive operation (when data is being input to the reception shifter), the UART does not halt reception until it completes reception of the frame and stores the received data in the receive data buffer SIDR register.

[Bit 0] TXE (Transmitter Enable): Transmitter enable

Controls UART transmission.

0	Disable transmission. [Initial value]
1	Enable transmission.

Precautions:

If transmission is disabled during a transmit operation (when data is being output from the transmission register), the UART does not halt transmission until there is no more data in the transmission data buffer SODR register.

27.2.4 Serial Input / Output Register (SIDR / SODR)

The serial input register (SIDR) and serial output register (SODR) are the reception/ transmission data buffer registers.
The MSB (D7) is ignored when the number of data bits is set to 7 bits.
Only write to the SODR register when the TDRE bit of the SSR register is "1".

■ **Structure of the Serial Input Register (SIDR) and Serial Output Register (SODR)**

		Bits									
	Address	7	6	5	4	3	2	1	0	Initial value	Access
SIDR	0000 0061H 0000 006DH 0000 0079H	D7	D6	D5	D4	D3	D2	D1	D0	Indeterminate	R
SODR	0000 0061H 0000 006DH 0000 0079H	D7	D6	D5	D4	D3	D2	D1	D0	Indeterminate	W

Precautions: Writing to these addresses writes to the SODR register and reading reads the SIDR register.

Figure 27.2.4a Structure of the Serial Input Register and Serial Output Register

27.2.5 Serial Status Register (SSR)

The serial status register (SSR) contains the flags that indicate the operating state of the UART.

■ Structure of the Serial Status Register (SSR)

		Bits								
	Address	7	6	5	4	3	2	1	0	Initial value
SSR	0000 0060H 0000 006CH 0000 0078H	PE	ORE	FRE	RDRF	TDRE	–	RIE	TIE	00001-00B
		R	R	R	R	R		R/W	R/W	⇦ Access

Figure 27.2.5a Structure of the Serial Status Register

■ Functions of the SSR Bits

[Bit 7] PE (Parity Error): Parity error

Set when a parity error occurs during reception. PE is an interrupt request flag.

To clear the flag once it has been set, write "0" to REC bit of the SCR register.

The contents of the SIDR register are not valid when this bit is set.

0	No parity error	[Initial value]
1	Parity error occurred.	

[Bit 6] ORE (Over Run Error): Overrun error

Set when an overrun error occurs during reception. ORE is an interrupt request flag.

To clear the flag once it has been set, write "0" to REC bit of the SCR register.

The contents of the SIDR register are not valid when this bit is set.

0	No overrun error	[Initial value]
1	Overrun error occurred.	

[Bit 5] FRE (FRaming Error): Framing error

Set when a framing error occurs during reception. FRE is an interrupt request flag.

To clear the flag once it has been set, write "0" to REC bit of the SCR register.

The contents of the SIDR register are not valid when this bit is set.

0	No framing error	[Initial value]
1	Framing error occurred.	

[Bit 4] RDRF (Receiver Data Register Full): Receive data register full

Indicates that received data is present in the SIDR register. RDRF is an interrupt request flag.

The bit is set when received data is loaded into the SIDR register and automatically cleared when the SIDR register is read.

0	No received data	[Initial value]
1	Received data present.	

[Bit 3] TDRE (Transmitter Data Register Empty): Transmit data register empty

Indicates that transmission data can be written to the SODR register. TDRE is an interrupt request flag.

Writing transmission data to the SODR register clears the bit. The bit is set again when the data written to the register is loaded into the transmission shifter and data transfer starts. This indicates that you can now write the next data to the SODR register.

0	Writing transmission data is prohibited.	
1	Writing transmission data is enabled.	[Initial value]

[Bit 2] Reserved

[Bit 1] RIE(Receive Interrupt Enable): Receive interrupt enable

Controls the reception interrupt.

0	Interrupt disabled.	[Initial value]
1	Interrupt enabled.	

Precautions:

Reception interrupts are triggered by PE, ORE, or FRE when an error occurs as well as by RDRF when data is received normally.

[Bit 0] TIE(Transmitter Interrupt Enable): Transmit interrupt enable

Controls the transmission interrupt.

0	Interrupt disabled.	[Initial value]
1	Interrupt enabled.	

Precautions:

Transmission interrupts are triggered by TDRE (transmission request).

27.2.6 UART Level Select Register (ULS)

The UART level select register controls the level of SDI and SDO. It is also used to control the clock for the UART and the U-Timer

■ **Structure of the UART Level Select Register (ULS)**

		Bits								
	Address	7	6	5	4	3	2	1	0	Initial value
ULS	0000 0064H 0000 0070H 0000 007CH	----	----	----	----	NSDO	NSDI	UTDBL	UDBL	----0000B
						R/W	R/W	R/W	R/W	⇦ Access

■ **Functions of the ULS Bits**

[Bit 3] NSDO: Negate SO

If this bit is set the output signal on SO will be inverted.

This bit is initialized to "0" upon reset. This bit is readable and writable.

[Bit 2] NSDI: Negate SI

If this bit is set the input signal on SI will be inverted.

This bit is initialized to "0" upon reset. This bit is readable and writable.

[Bit 1] UTDBL: U-Timer Disable

This bit is used to control the clock for the U-Timer.

When "1" is written to this bit, the clock for the U-Timer module is disabled. When "0" is set, a clock is supplied to the U-Timer module.

This bit is initialized to "0" upon reset. This bit is readable and writable.

[Bit 0] UDBL: UART Disable

This bit is used to control the clock for the UART.

When "1" is written to this bit, the clock for the UART module is disabled. When "0" is set, a clock is supplied to the UART module.

This bit is initialized to "0" upon reset. This bit is readable and writable.

27.3 UART OPERATION

This section describes the following items relating to the UART operation. The section first describes the operation modes and clock selection, then describes each mode in turn.

- **UART operation mode/UART clock selection**
- **Asynchronous (start bit synchronization) mode**
- **Interrupt generation and flag set timings**

■ Operation Modes

The UART has the operation modes listed in Table 11.3. You can switch between modes by setting the SMR and SCR registers.

Mode	Parity	Number of data bits	Operation mode	Number of stop bits
0	Yes/ No	7	Normal asynchronous (start bit synchronization) mode	1 or 2 bits
	Yes/ No	8		
1	None	8 + 1	Multi-processor asynchronous (start bit synchronization) mode	

However, the number of stop bits in asynchronous (start bit synchronization) mode can only be specified for transmission. Reception always operates assuming only one stop bit. The UART does not operate in modes other than those listed above. Do not set other modes.

■ UART Clock Selection

- Internal timer

  The U-TIMER and the reload value set for the U-TIMER determines the baud rate. The baud rate in this case is calculated by the following formula.

  Asynchronous (start bit synchronization) mode: $\Phi / (16 \times \beta)$
  CLK-synchronous mode: Φ / β

  Φ: Frequency of the peripheral machine clock (CLKP)
  β : Cycle set in U-TIMER (2n+2 or 2n+3, where n is the reload value)

  For asynchronous (start bit synchronization) mode, transfer can be performed with a baud rate in the range –1% to +1% of the specified baud rate.

- CLK-synchronous mode is not supported in MB91360 series.

27.3.1 Asynchronous (Start Bit Synchronization) Mode

This section describes asynchronous (start bit synchronization) communications.

■ Transfer Data Format

The UART internally only handles NRZ (Non Return to Zero) format data. Figure 27.3.1a shows the data format. If you want to use a different data format use the appropriate settings in the UART Level Select register (ULS).

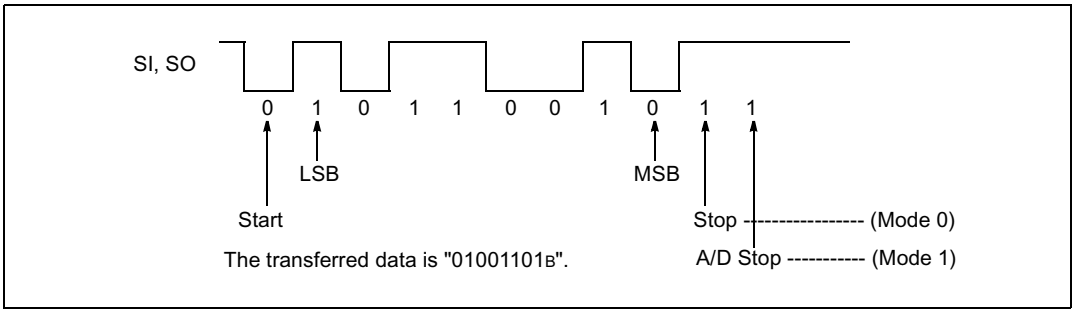


Figure 27.3.1a Transfer Data Format for Asynchronous (start bit Synchronization) Mode (Modes 0 and 1)

As Figure 27.3.1a shows, the transferred data starts with a start bit ("L" level data) which is followed by the specified number of data bits (LSB-first). The data format ends with the stop bit ("H" level data).

When an external clock is selected, input the clock continuously.

The number of data bits can be set to 7 or 8 in normal mode (mode 0).

In multi-processor mode (mode 1), the number of data bits must be set to 8. Also, a parity bit cannot be appended in multi-processor mode but the A/D bit is always appended.

■ Reception

Reception operates continuously when bit 1 (RXE) of the SCR is "1".

After detecting a start bit on the receive line, the UART receives the frame of data using the data format specified in SCR. If an error occurs after the frame has been received, the error flag is set and then the RDRF flag (bit 4 of SSR) is set.

If bit 1 (RIE) of SSR is set to "1" at this time, a reception interrupt is output to the CPU.

Check the SSR register flags and read the SIDR register if reception was normal or perform any required processing if an error occurred.

Reading the SIDR register clears the RDRF flag.

■ **Transmission**

Write the transmission data to SODR when the TDRE flag (bit 3) of SSR is "1". If bit 0 (TXE) of SCR is "1", the data is transmitted.

The data set in SODR is loaded into the transmission shift register and transmission starts. This sets the TDRE flag again indicating that the next transmission data can be set to SODR.

If bit 0 (TIE) of SSR is set to "1" at this time, a transmission interrupt is output to the CPU to request that transmission data be set to SODR.

Setting data to SODR clears the TDRE flag.

27.3.2 Interrupt Generation and Flag Set Timings

This section explaines the interrupt generation of the UART.

The UART has five flags and two interrupts.

The five flags are PE / ORE / FRE / RDRF / TDRE. PE indicates a parity error, ORE an overrun error, and FRE a framing error. These flags are set when a receive error occurs. Writing "0" to the REC bit of SCR clears these flags.

RDRF is an interrupt flag and indicates that received data is present in the SIDR register. The flag is set when received data is loaded into the SIDR register and cleared when the register is read. Note that the parity detection function is not available in mode 1 and the parity and framing error detection functions are not available in mode 2.

TDRE is an interrupt flag and indicates that transmit data can be written to the SODR register. The flag is set when the SODR register becomes empty (and therefore writing to the register becomes possible). Writing to the SODR register clears the flag.

The two interrupts are the reception interrupt and transmission interrupt. When receiving, an interrupt request is generated by PE / ORE / FRE / RDRF. When transmitting, an interrupt request is generated by TDRE. The following shows the interrupt flag set timings for each operation mode.

■ **Interrupt Flag Set Timing When Receiving in Mode 0**

The PE, ORE, FRE, and RDRF flags are set and an interrupt request output to the CPU when data reception completes and the last stop bit is detected.

The data in SIDR is invalid if PE, ORE, or FRE is set.

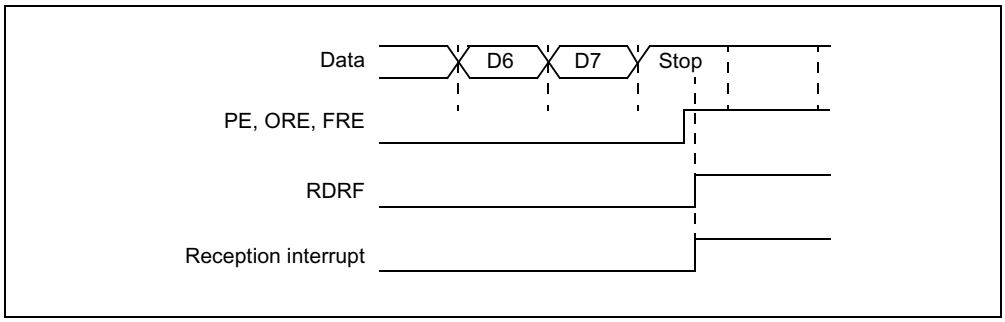


Figure 27.3.2a Interrupt Flag Set Timing When Receiving in Mode 0

■ Interrupt Flag Set Timing When Receiving in Mode 1

The ORE, FRE, and RDRF flags are set and an interrupt request output to the CPU when data reception completes and the last stop bit is detected. As only 8 data bits can be received, the final bit (bit 9) indicating whether the frame is address/data is treated as invalid data.

The data in SIDR is invalid if ORE or FRE is set.

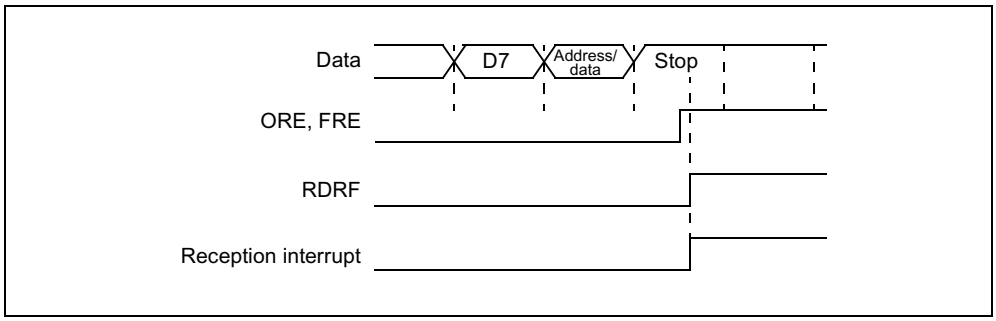


Figure 27.3.2b Interrupt Flag Set Timing When Receiving in Mode 1

■ Interrupt Flag Set Timing When Receiving in Mode 2

The ORE and RDRF flags are set and an interrupt request output to the CPU when data reception completes and the last data bit (D7) is detected.

The data in SIDR is invalid if ORE is set.

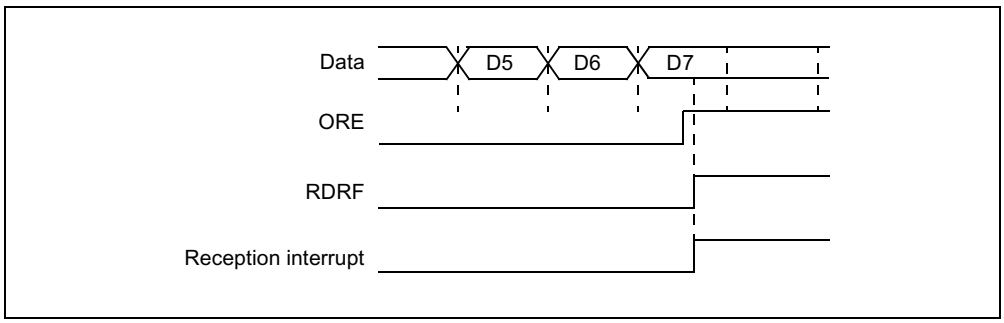


Figure 27.3.2c Interrupt Flag Set Timing When Receiving in Mode 2

■ Interrupt Flag Set Timing When Transmitting in Mode 0, 1, 2

Writing to SODR clears the TDRE flag and transfers the data to the internal shift register. The TDRE flag is set and an interrupt request output to the CPU when the next data is able to be written to SODR.

Writing "0" to the TXE bit in SCR (or to the RXE bit in mode 2) while transmission is in progress, disables UART transmission after the TDRE bit in SSR goes to "1" and the transmission shifter halts. Data written to SODR after writing "0" to the TXE bit in SCR (or to the RXE bit in mode 2) but before transmission halts is transmitted.

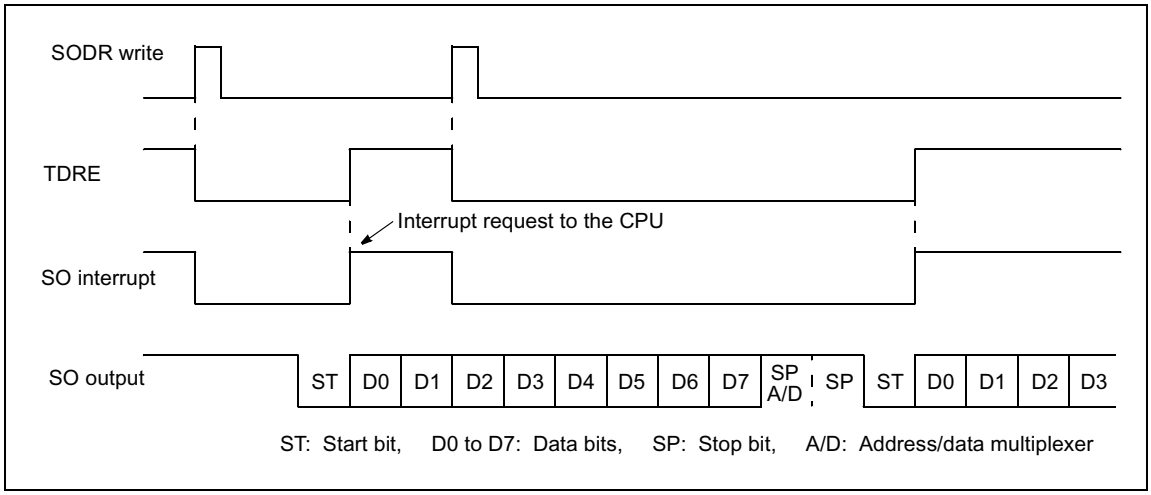


Figure 27.3.2d Interrupt Flag Set Timing When Transmitting in Mode 0 and 1

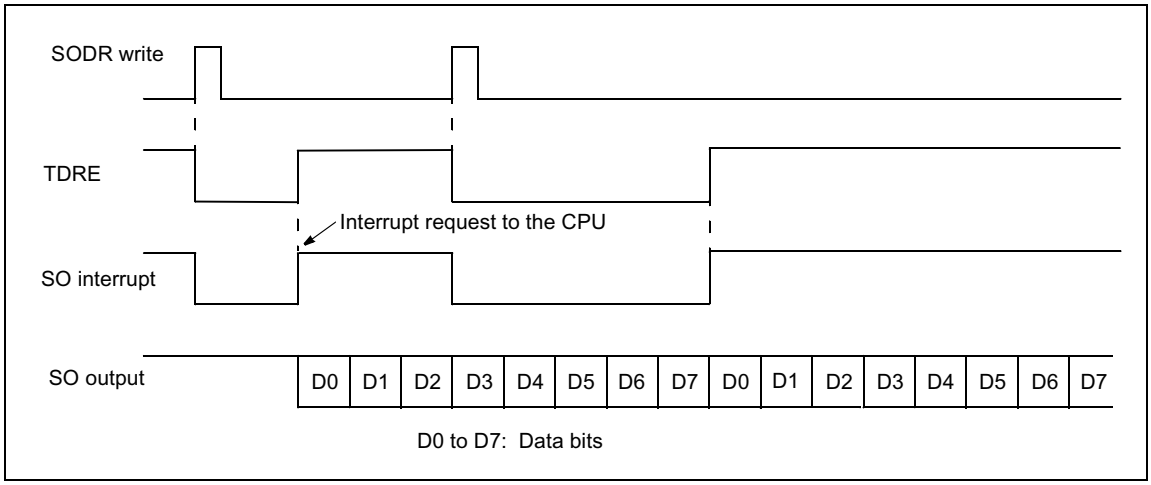


Figure 27.3.2e Interrupt Flag Set Timing When Transmitting in Mode 2

27.3.3 Other Items

This section describes precautions when using the UART, example applications, and examples of setting the baud rate and U-TIMER reload value.

■ Precautions When Using the UART

Only set the operation mode when the UART is halted.

The transmission and reception data when the mode is set is not guaranteed.

■ Example Application

Mode 1 is used to connect a single host CPU to a number of slave CPUs. (See figure 27.3.3a.)

The UART only supports the communications interface for the host CPU.

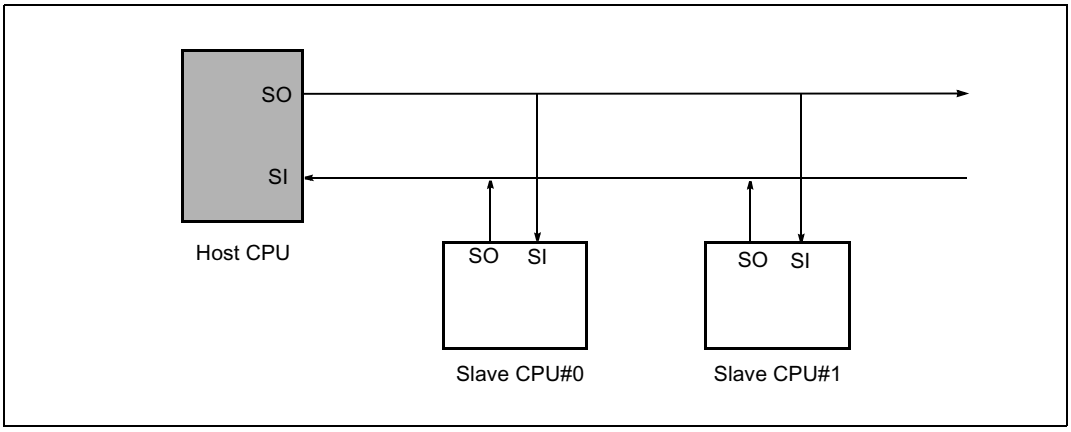


Figure 27.3.3a Example System Structure Using Mode 1

Communications starts with the host CPU transmitting address data.

Data transmitted when the A/D bit of SCR is "1" is address data. The address data selects which slave CPU to communicate with and enables communications with the host CPU.

Data transmitted when the A/D bit of SCR is "0" is normal data.

Figure 27.3.5b shows a flowchart of this operation.

As the parity check function is not available in this mode, set the PEN bit of SCR to "0".

■ Communications Flowchart for Mode 1

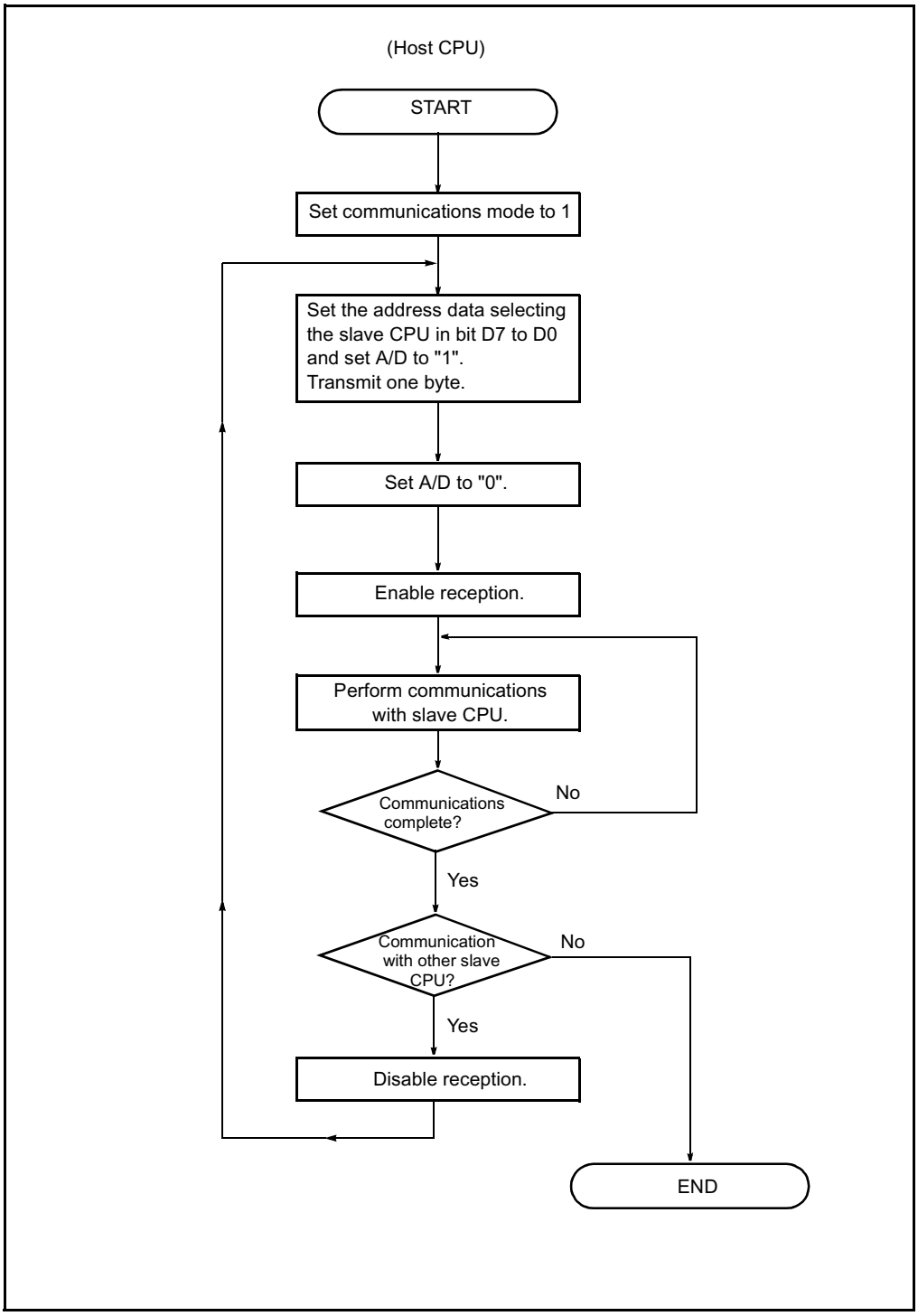


Figure 27.3.3b Communications Flowchart for Mode 1

■ Example of Setting the Baud Rate and U-timer Reload Value

The following shows an example of setting the baud rate and U-timer reload value.

The frequencies listed in the table represent the peripheral clock (CLKP) frequency. UCC1 is the value set in the UCC1 bit of the U-timer control register (UTIMC).

Setting options marked by "–" in the table indicate that the error exceeds ±1% and therefore cannot be used.

- Asynchronous (start bit synchronization) mode

Baud rate	µs	24 MHz	16 MHz	12 MHz	8 MHz
1200	833.33	624 (UCC1=0)	415 (UCC1=1)	311 (UCC1=1)	207 (UCC1=1)
2400	416.67	311 (UCC1=1)	207 (UCC1=1)	155 (UCC1=1)	103 (UCC1=0)
4800	208.33	155 (UCC1=0)	103 (UCC1=0)	77 (UCC1=0)	51 (UCC1=0)
9600	104.17	77 (UCC1=0)	51 (UCC1=0)	38 (UCC1=0)	25 (UCC1=0)
19200	52.08	38 (UCC1=0)	25 (UCC1=0)	18 (UCC1=1)	12 (UCC1=0)
38400	26.04	18 (UCC1=1)	12 (UCC1=0)	-	5 (UCC1=1)
57600	17.36	12 (UCC1=1)	7 (UCC1= 1)	5 (UCC1= 1)	-
10400	96.15	71 (UCC1=0)	47 (UCC1=0)	35 (UCC1=0)	23 (UCC1=0)
31250	32.00	23 (UCC1=0)	15 (UCC1=0)	11 (UCC1=0)	7 (UCC1=0)
62500	16.00	11 (UCC1=0)	7 (UCC1=0)	5 (UCC1= 0)	3 (UCC1=0)

CHAPTER 28 USART WITH LIN-FUNCTIONALITY

This chapter explains the functions and operation of USART. The USART with LIN (Local Interconnect Network) - Function is a general-purpose serial data communication interface for performing synchronous or asynchronous communication with external devices.

Note: This chapter only lists the registers and addresses of USART #5. Please refer to the IO-Map for the addresses of the other USARTs.

28.1 OVERVIEW OF USART

The USART with LIN (Local Interconnect Network) - Function is a general-purpose serial data communication interface for performing synchronous or asynchronous communication with external devices. USART provides bidirectional communication function (normal mode), master-slave communication function (multiprocessor mode in master/slave systems), and special features for LIN-bus systems (working both as master or as slave device).
Please note that USART is not software compatible to the other UARTs

■ USART Functions

USART is a general-purpose serial data communication interface for transmitting serial data to and receiving data from another CPU and peripheral devices. It has the functions listed in table 28.1a.

Table 28.1a USART functions

Item	Function
Data buffer	Full-duplex
Serial Input	5 times oversampling in asynchronous mode
Transfer mode	- Clock synchronous (start-stop synchronization and start-stop-bit-option) - Clock asynchronous (using start-, stop-bits)
Baud rate	- A dedicated baud rate generator is provided, which consists of a 15-bit-reload counter - An external clock can be input and also be adjusted by the reload counter
Data length	- 7 bits (not in synchronous or LIN mode) - 8 bits
Signal mode	Non-return to zero (NRZ) and return to zero (RZ)
Start bit timing	Clock synchronization to the falling edge of the start bit in asynchronous mode
Reception error detection	- Framing error - Overrun error - Parity error
Interrupt request	- Reception interrupt (reception complete, reception error detect, Bus-Idle, LIN-Synch-break detect) - Transmission interrupt (transmission complete)
Master-slave communication function (multiprocessor mode)	One-to-n communication (one master to n slaves) (This function is supported both for master and slave system).
Synchronous mode	Function as Master- or Slave-USART

Table 28.1a USART functions (Continued)

Transceiving pins	Direct access possible
LIN bus options	- Operation as master device - Operation as slave device - Generation of LIN-Sync-break - Detection of LIN-Sync-break - Detection of start/stop edges in LIN-Sync-field connected to ICU 0 and 2
Synchronous serial clock	The synchronous serial clock can be output continuously on the SCK pin for synchronous communication with start & stop bits
Clock delay option	Special synchronous Clock Mode for delaying clock (useful for SPI-compliance)

■ USART operation modes

The USART operates in four different modes, which are determined by the MD0- and the MD1-bit of the Serial mode register (SMR5). Mode 0 and 2 are used for bidirectional serial communication, mode 1 for master/slave communication and mode 3 for LIN master/slave communication.

Table 28.1b USART operation modes

Operation mode		Data length		Synchroniza-tion of mode	Length of stop bit	data bit directio n*
		parity disabled	parity enabled			
0	normal mode	7 or 8		asynchronous	1 or 2	L/M
1	multiprocessor	7 or 8 + 1**	-	asynchronous	1 or 2	L/M
2	normal mode	8		synchronous	0, 1 or 2	L/M
3	LIN mode	8	-	asynchronous	1	L

* means the data bit transfer format: LSB or MSB first.

** "+1" means the indicator bit of the address/data selection in the multiprocessor mode, instead of parity.

Note: Mode 1 operation is supported both for master or slave operation of USART in a master-slave connection system. In Mode 3 the USART function is locked to 8N1-Format, LSB first.

If the mode is changed, USART cuts off all possible transmission or reception and awaits then new action.

The MD1 and MD0 bit of the Serial Mode Register (SMR5) determine the operation mode of USART as shown in the following table:

Table 28.1c Mode Bit Setting

MD1	MD0	Mode	Description
0	0	0	Asynchronous (normal mode)
0	1	1	Asynchronous (multiprocessor mode)
1	0	2	Synchronous (normal mode)
1	1	3	Asynchronous (LIN mode)

■ USART Interrupts

Table 28.1d USART interrupts

Interrupt cause	Interrupt number	Interrupt control register		Interrupt Vector	
		Register name	Address	Offset	Default address
USART5 reception interrupt	#55 (37H)	ICR39	0467H	320H	000FFF20H
USART5 transmission interrupt	#56 (38H)	ICR40	0468H	31CH	000FFF1CH
USART6 reception interrupt	#57 (39H)	ICR41	0469H	318H	000FFF18H
USART6 transmission interrupt	#58 (3AH)	ICR42	046AH	314H	000FFF14H

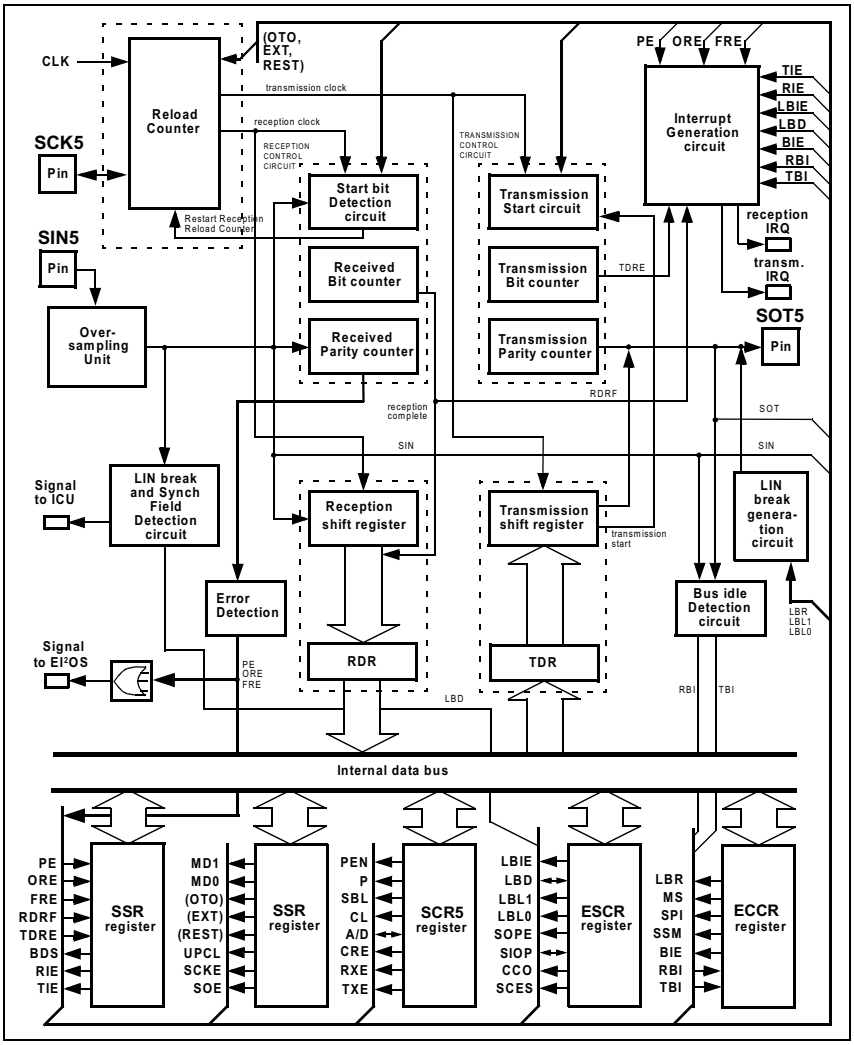
28.2 CONFIGURATION OF USART

This section provides a short overview on the building blocks of USART.

■ **USART consists of the following blocks:**

- Reload Counter
- Reception Control Circuit
- Reception Shift Register
- Reception Data Register
- Transmission Control Circuit
- Transmission Shift Register
- Transmission Data Register
- Error Detection Circuit
- Oversampling Unit
- Interrupt Generation Circuit
- LIN Break and Synch Field Detection
- Bus Idle Detection Circuit
- Serial Mode Register (SMR5)
- Serial Control Register (SCR5)
- Serial Status Register (SSR5)
- Serial Control Register (SCR5)
- Extended Com. Contr. Reg. (ECCR5)
- Extended Status/Contr. Reg. (ESCR5)

Figure 28.2a USART Block Diagram



■ Explanation of the different blocks

- Reload Counter

  The reload counter functions as the dedicated baud rate generator. It can select external input clock or internal clock for the transmitting and receiving clocks. The reload counter has a 15 bit register for the reload value. The actual count of the transmission reload counter can be read via the BGR0/1.

- Reception Control Circuit

  The reception control circuit consists of a received bit counter, start bit detection circuit, and received parity counter. The received bit counter counts reception data bits. When reception of one data item for the specified data length is complete, the received bit counter sets the Reception data register full flag. The start bit detection circuit detects start bits from the serial input signal and sends a signal to the reload counter to synchronize it to the falling edge of these start bits. The reception parity counter calculates the parity of the reception data.

- Reception Shift Register

  The reception shift register fetches reception data input from the SIN5 pin, shifting the data bit by bit. When reception is complete, the reception shift register transfers receive data to

the RDR5 register.

- Reception Data Register

  This register retains reception data. Serial input data is converted and stored in this register.

- Transmission Control Circuit

  The transmission control circuit consists of a transmission bit counter, transmission start circuit, and transmission parity counter.The transmission bit counter counts transmission data bits. When the transmission of one data item of the specified data length is complete, the transmission bit counter sets the Transmission data register full flag. The transmission start circuit starts transmission when data is written to TDR5. The transmission parity counter generates a parity bit for data to be transmitted if parity is enabled.

- Transmission Shift Register

  The transmission shift register transfers data written to the TDR5 register to itself and outputs the data to the SOT5 pin, shifting the data bit by bit.

- Transmission Data Register

  This register sets transmission data. Data written to this register is converted to serial data and output.

- Error Detection Circuit

  The error detection circuit checks if there was any error during the last reception. If an error has occurred it sets the corresponding error flags.

- Oversampling Unit

  The oversampling unit oversamples the incoming data at the SIN5 pin for five times. It is switched off in synchronous operation mode.

- Interrupt Generation Circuit

  The interrupt generation circuit administers all cases of generating a reception or transmission interrupt. If a corresponding enable flag is set and an interrupt case occurs the interrupt will be generated immediately.

- LIN Break and Synchronization Field Detection Circuit

  The LIN break and LIN synchronization field detection circuit detects a LIN break, if a LIN master node is sending a message header. If a LIN break is detected a special flag bit is generated. The first and the fifth falling edge of the synchronization field is recognized by this circuit by generating an internal signal for the Input Capture Unit to measure the actual serial clock time of the transmitting master node.

- LIN Break Generation Circuit

  The LIN break generation circuit generates a LIN break of a determined length.

- Bus Idle Detection circuit

  The bus idle detection circuit recognizes if neither reception nor transmission is going on. In this case the circuit generates a special flag bit.

- Serial Mode Register

  This register performs the following operations:

  • Selecting the USART operation mode

  • Selecting a clock input source

  • Selecting if an external clock is connected “one-to-one” or connected to the reload counter

- Resetting the USART (preserving the settings of the registers)
- Specifying whether to enable serial data output to the corresponding pin
- Specifying whether to enable clock output to the corresponding pin

● Serial Control Register

This register performs the following operations:

- Specifying whether to provide parity bits
- Selecting parity bits
- Specifying a stop bit length
- Specifying a data length
- Selecting a frame data format in mode 1
- Clearing the error flags
- Specifying whether to enable transmission
- Specifying whether to enable reception

● Serial Status Register

This register checks the transmission and reception status and error status, and enables and disables transmission and reception interrupt requests.

● Extended Status/Control Register

This register provides several LIN functions, direct access to the SIN5 and SOT5 pin and setting for the USART synchronous clock mode.

● Extended Communication Control Register

The extended communication control register provides bus idle recognition interrupt settings, synchronous clock settings, and the LIN break generation.

28.3 USART PINS

This section describes the USART pins and provides a pin block diagram.

■ USART Pins

The USART pins also serve as general ports. Table 28.3a lists the pin functions, I/O formats, and settings required to use USART5.

Table 28.3a USART5 Pins

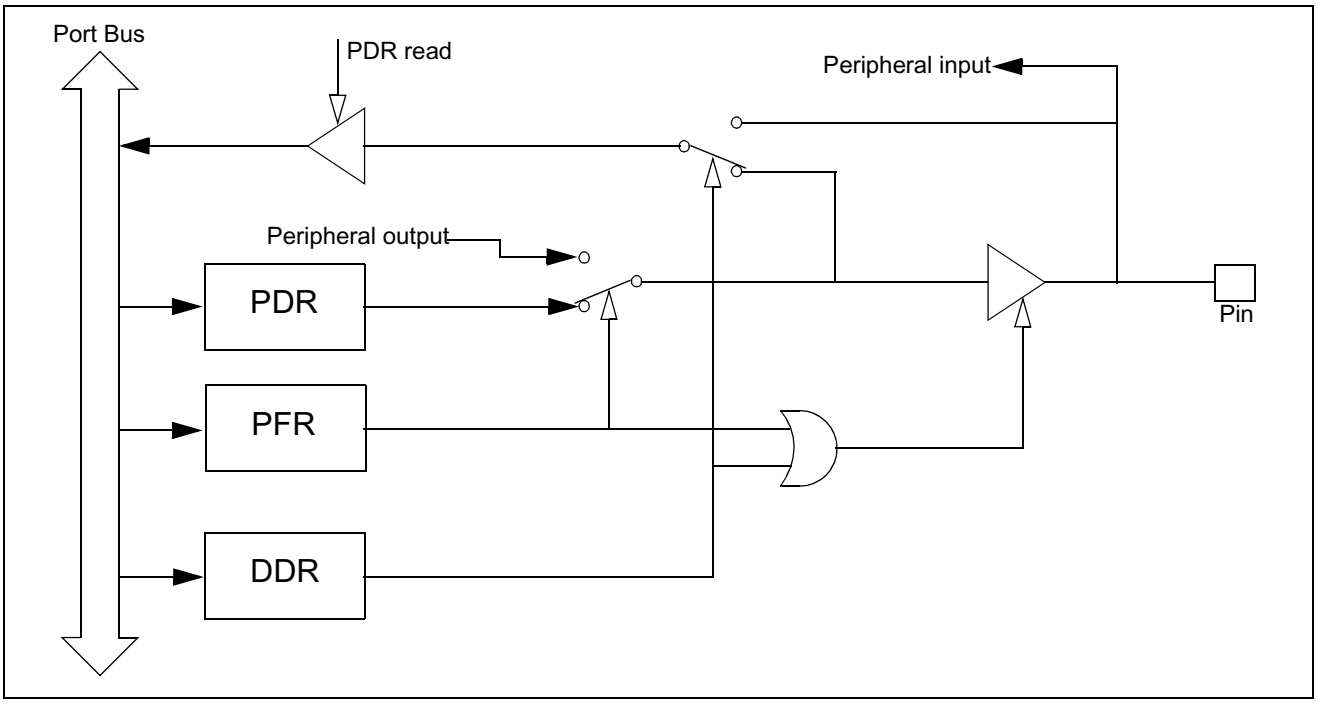
Pin name MB91F364G	Pin function	I/O format	Pull-up	Standby control	Setting required to use pin
SIN5	Port T I/O or serial data input	CMOS output and CMOS Automotive hysteresis input	Nothing	Provided	Set port function mode: PFRT: bit0 = 1
SOT5	Port T I/O or serial data output				Set to output enable mode: SMR5: SOE = 1, Set port function mode: PFRT: bit2 = 1
SCK5	Port T I/O or serial clock input/ output				Set port function mode: PFRT: bit1 = 1 Set to output enable mode when a clock is output SMR5: SCKE = 1

MB91F364G containes a second USART (USART6). It is connected to the following pins:

Table 28.3b USART6 Pins

Pin name MB91F364G	Pin function	I/O format	Pull-up	Standby control	Setting required to use pin
SIN6	Port T I/O or serial data input	CMOS output and CMOS Automotive hysteresis input	Nothing	Provided	Set port function mode: PFRT: bit5 = 1
SOT6	Port T I/O or serial data output				Set to output enable mode: SMR6: SOE = 1, Set port function mode: PFRT: bit3 = 1
SCK6	Port T I/O or serial clock input/ output				Set port function mode: PFRT: bit4 = 1 Set to output enable mode when a clock is output SMR6: SCKE = 1

Figure 28.3a Block Diagram of USART pins



28.4 USART REGISTERS

The following table shows the USART registers (MB91F364G).

Table 28.4 USART Registers

Address	bit 15 bit 8	bit 7 bit 0
0198H, 0199H	SCR5 (Serial Control Register)	SMR5 (Serial Mode Register)
019AH, 019BH	SSR5 (Serial Status Register)	RDR5/TDR5 (Rx, Tx Data Register)
019CH, 019DH	ESCR5 (Extended Status/Control Reg.)	ECCR5 (Extended Comm. Contr. Reg.)
019EH, 019FH	BGR15 (Baud Rate Generator Reg. 1)	BGR05 (Baud Rate Generator R. 0)
01A0H, 01A1H	SCR6 (Serial Control Register)	SMR6 (Serial Mode Register)
01A2H, 01A3H	SSR6 (Serial Status Register)	RDR6/TDR6 (Rx, Tx Data Register)
01A4H, 01A5H	ESCR6 (Extended Status/Control Reg.)	ECCR6 (Extended Comm. Contr. Reg.)
01A6H, 01A7H	BGR16 (Baud Rate Generator Reg. 1)	BGR06 (Baud Rate Generator R. 0)

28.4.1 Serial Control Register 5 (SCR5)

This register specifies parity bits, selects the stop bit and data lengths, selects a frame data format in mode 1, clears the reception error flag, and specifies whether to enable transmission and reception.

Figure 28.4.1 Serial Control Register 5 (SCR5)

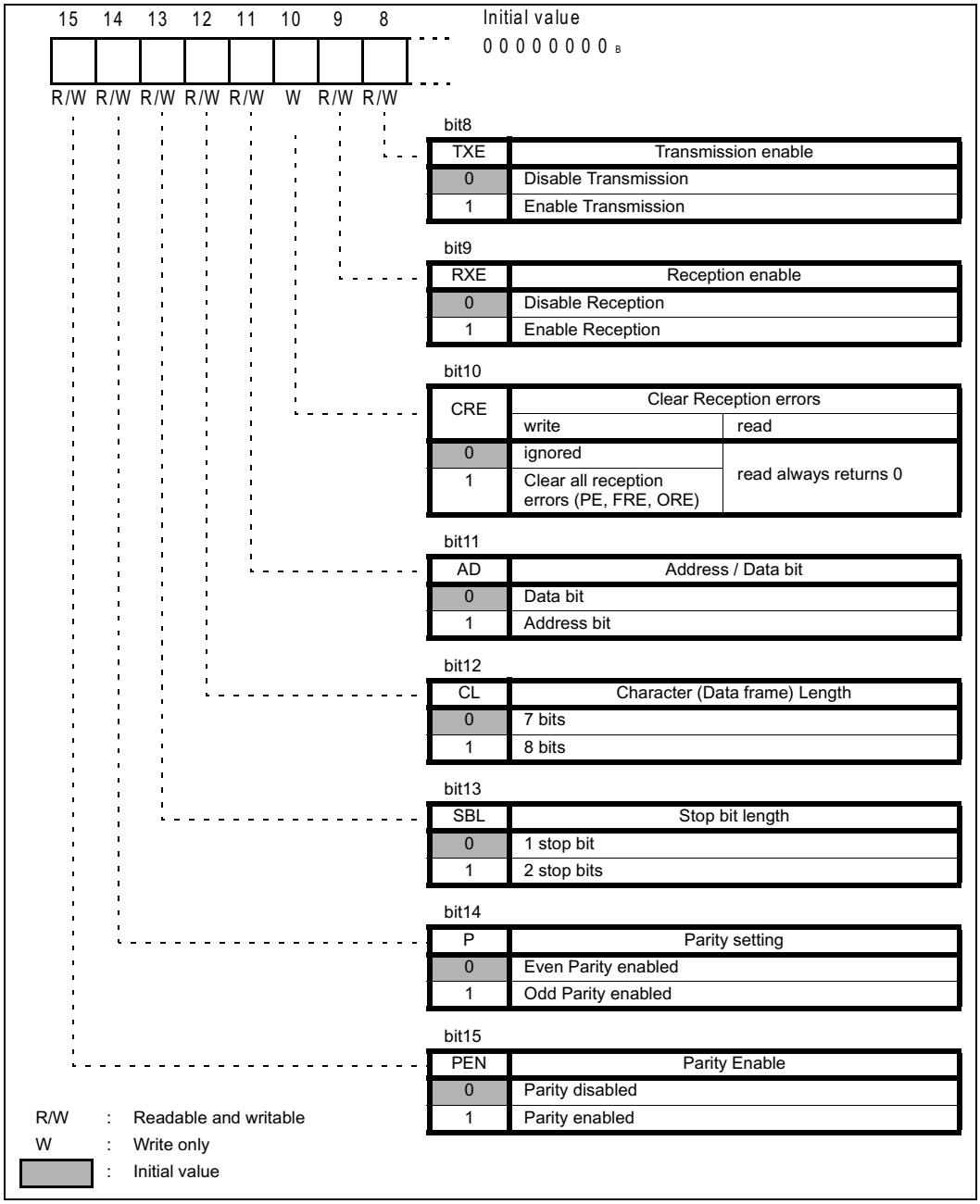


Table 28.4.1a Functions of each bit of control register 5 (SCR5)

Bit name		Function
bit15	PEN: Parity enable bit	This bit selects whether to add a parity bit during transmission in serial asynchronous mode or detect it during reception. Parity is only provided in mode 0 and in mode 2 if SSM of the ECCR5 is selected. This bit is fixed to 0 (no parity) in mode 3 (LIN).
bit14	P: Parity selection bit	When parity is provided and enabled this bit selects even (0) or odd (1) parity
bit13	SBL: Stop bit length selection bit	This bit selects the length of the stop bit of an asynchronous data frame or a synchronous frame if SSM of the ECCR5 is selected. This bit is fixed to 0 (1 stop bit) in mode 3 (LIN).
bit12	CL: Data length selection bit	This bit specifies the length of transmission or reception data. This bit is fixed to 1 (8 bits) in mode 2 and 3.
bit11	AD: Address/Data selection bit *	This bit specifies the data format in multiprocessor mode 1. Writing to this bit determines an address or data frame to be sent next, reading from it returns the last received kind of frame. A 1 indicates an address frame, a 0 indicates a usual data frame. **Note:** During a RMW-Read cycle the AD bit returns the value to be sent instead of the last received AD bit. see table below*
bit10	CRE: Clear reception error flags bit	This bit clears the FRE, ORE, and PE flag of the Serial Status Register (SSR5). This bit also clears a possible reception interrupt caused of errors. Writing a 1 to it clears the error flag. Writing a 0 has no effect. Reading from it always returns 0.
bit9	RXE: Reception enable bit	This bit enables USART reception. If this bit is set to 0, USART disables the reception of data frames. The LIN break detection in mode 0 or 3 remains unaffected.
bit8	TXE: Transmission enable bit	This bit enables USART transmission. If this bit is set to 0, USART disables the transmission of data frames.

* see table 28.4.1b for R/W options

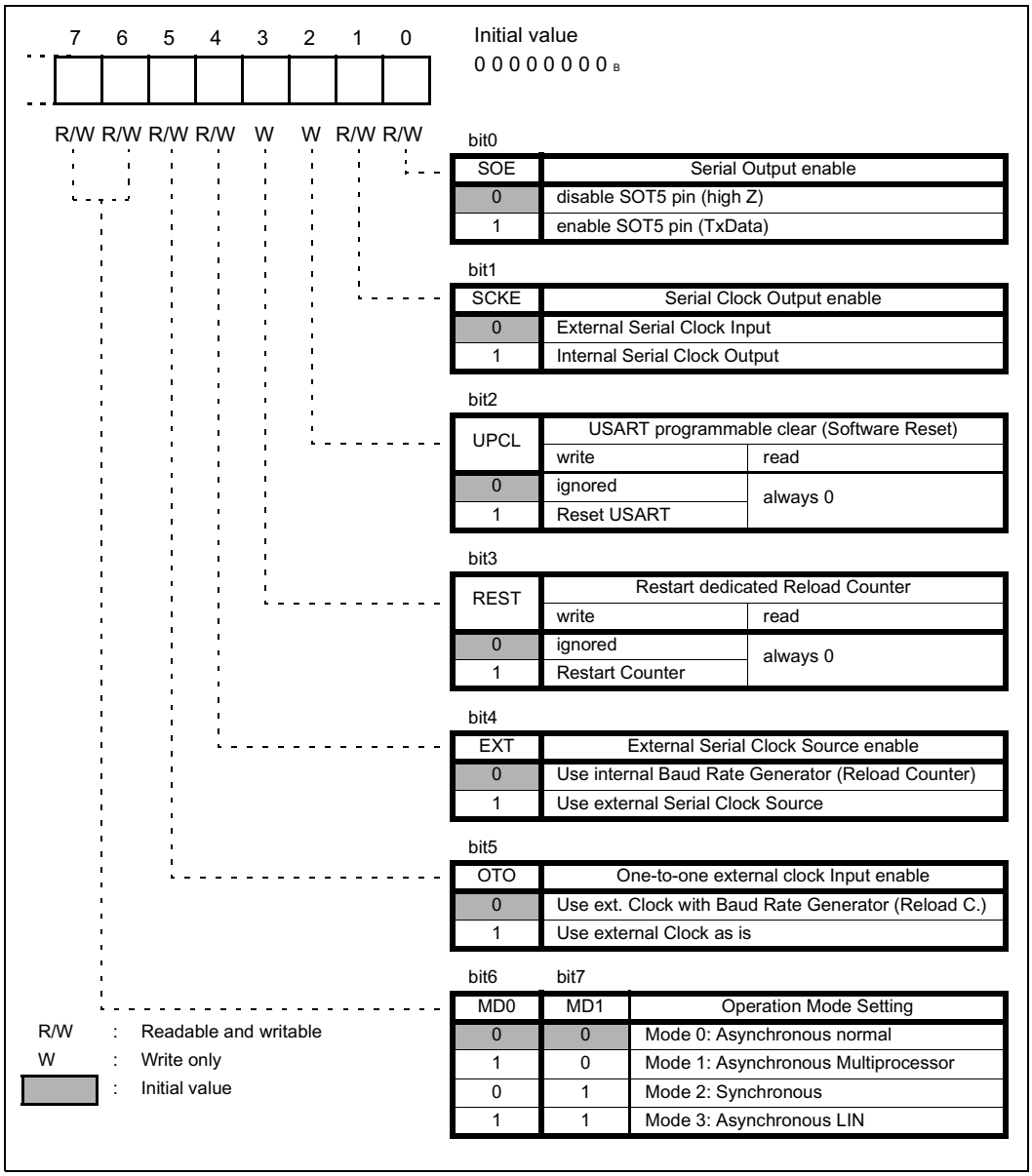
Table 28.4.1b * Read/Write options of AD-Bit

Cycle	Action
Write	Write data to be sent to AD-Bit
Normal Read	Read received AD-Bit
RMW-Read	Read data to be sent from AD-Bit

28.4.2 Serial Mode Register 5 (SMR5)

This register selects an operation mode and baud rate clock and specifies whether to enable output of serial data and clocks to the corresponding pin.

Figure 28.4.2 Configuration of the Serial Mode register 5 (SMR5)



bit0

SOE	Serial Output enable
0	disable SOT5 pin (high Z)
1	enable SOT5 pin (TxData)

bit1

SCKE	Serial Clock Output enable
0	External Serial Clock Input
1	Internal Serial Clock Output

bit2

UPCL	USART programmable clear (Software Reset)	
	write	read
0	ignored	always 0
1	Reset USART	

bit3

REST	Restart dedicated Reload Counter	
	write	read
0	ignored	always 0
1	Restart Counter	

bit4

EXT	External Serial Clock Source enable
0	Use internal Baud Rate Generator (Reload Counter)
1	Use external Serial Clock Source

bit5

OTO	One-to-one external clock Input enable
0	Use ext. Clock with Baud Rate Generator (Reload C.)
1	Use external Clock as is

bit6 MD0	bit7 MD1	Operation Mode Setting
0	0	Mode 0: Asynchronous normal
1	0	Mode 1: Asynchronous Multiprocessor
0	1	Mode 2: Synchronous
1	1	Mode 3: Asynchronous LIN

Table 28.4.2a Bit function of the Serial Mode register 5 (SMR5)

Bit name		Function
bit7 bit6	MD1 and MD0: Operation mode selection bits	These two bits sets the USART operation mode.
bit5	OTO: One-to-one external clock selection bit	This bit sets an external clock directly to the USART's serial clock. This function is used for synchronous slave mode operation
bit4	EXT: External clock selection bit	This bit executes internal or external clock source for the reload counter
bit3	REST: Restart of transmission reload counter bit	If a 1 is written to this bit the reload counter is restarted. Writing 0 to it has no effect. Reading from this bit always returns 0.
bit2	UPCL: USART programmable clear bit (Software reset)	Writing a 1 to this bit resets USART immediately. The register settings are preserved. Possible reception or transmission will cut off. All error flags are cleared and the Reception Data Register (RDR5) contains 00h. Writing 0 to this bit has no effect. Reading from it always returns 0.
bit1	SCKE: Serial clock output enable	• This bit controls the serial clock input-output ports. • When this bit is 0, the P94/SCK5 pin operate as general input-output port (P94) or serial clock input pin. When this bit is 1, the pin operates as serial clock output pin. <Caution> • When using the P94/SCK5 pin as serial clock input (SCKE=0) pin, set the P94 as input port. Also, select external clock (EXT = 1) using the external clock selection bit. <Reference> When the SCK5 pin is assigned to serial clock output (SCKE=1), it functions as the serial clock output pin regardless of the status of the general input-output ports.
bit0	SOE: Serial data output enable bit	• This bit enables or disables the output of serial data. • When this bit is 0, the P95/SOT5 pin operates as general input-output pin (P95). When this bit is 1, the P95/SOT5 pin operates as serial data output pins (SOT5). <Reference> When serial data is output (SOE=1), the enabled, the P95/SOT5 pin functions as SOT5 pins regardless of the status of general input-output ports (P95)

28.4.3 Serial Status Register 5 (SSR5)

This register checks the transmission and reception status and error status, and enables and disables the transmission and reception interrupts.

Figure 28.4.3 Configuration of the Serial Status register 5 (SSR5)

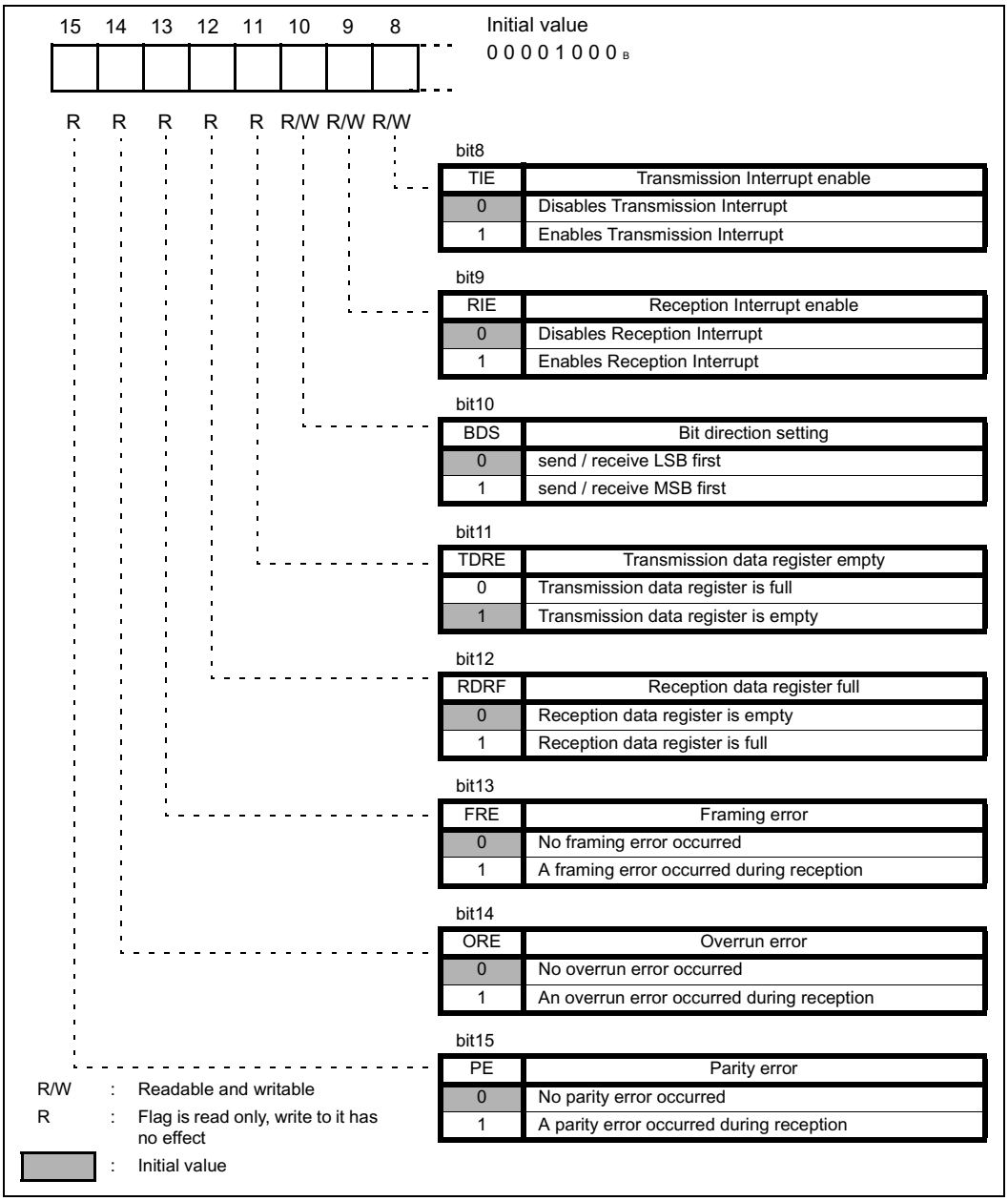


Table 28.4.3a Functions of each bit of status register 5 (SSR5)

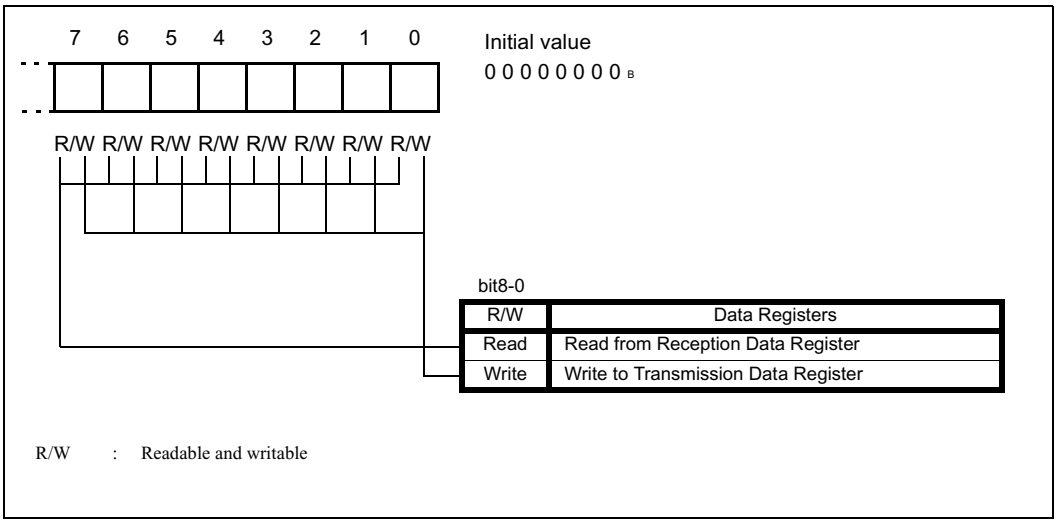
Bit name		Function
bit15	PE: Parity error flag bit	• This bit is set to 1 when a parity error occurs during reception and is cleared when 0 is written to the CRE bit of the serial mode register (SMR5). • A reception interrupt request is output when this bit and the RIE bit are 1. • Data in the reception data register (RDR5) is invalid when this flag is set.
bit14	ORE: Overrun error flag bit	• This bit is set to 1 when an overrun error occurs during reception and is cleared when 0 is written to the CRE bit of the serial mode register (SMR5). • A reception interrupt request is output when this bit and the RIE bit are 1. • Data in the reception data register (RDR5) is invalid when this flag is set.
bit13	FRE: Framing error flag bit	• This bit is set to 1 when a framing error occurs during reception and is cleared when 0 is written to the CRE bit of the serial mode register 1 (SMR5). • A reception interrupt request is output when this bit and the RIE bit are 1. • Data in the reception data register (RDR5) is invalid when this flag is set.
bit12	RDRF: Receive data full flag bit	• This flag indicates the status of the reception data register (RDR5). • This bit is set to 1 when reception data is loaded into RDR5 and can only be cleared to 0 when the reception data register (RDR5) is read. • A reception interrupt request is output when this bit and the RIE bit are 1.
bit11	TDRE: Transmission data empty flag bit	• This flag indicates the status of the transmission data register (TDR5). • This bit is cleared to 0 when transmission data is written to TDR5 and is set to 1 when data is loaded into the transmission shift register and transmission starts. • A transmission interrupt request is generated if this bit and the RIE bit are 1. <Caution> This bit is set to 1 (TDR5 empty) as its initial value.
bit10	BDS: Transfer direction selection bit	• This bit selects whether to transfer serial data from the least significant bit (LSB first, BDS=0) or the most significant bit (MSB first, BDS=1). <Caution> The high-order and low-order sides of serial data are interchanged with each other during reading from or writing to the serial data register. If this bit is set to another value after the data is written to the RDR5 register, the data becomes invalid. This bit is fixed to 0 in mode 3 (LIN)
bit9	RIE: Reception interrupt request enable bit	• This bit enables or disables input of a request for transmission interrupt to the CPU. • A reception interrupt request is output when this bit and the reception data flag bit (RDRF) are 1 or this bit and one or more error flag bits (PE, ORE, and FRE) are 1.
bit8	TIE: Transmission interrupt request enable bit	• This bit enables or disables output of a request for transmission interrupt to the CPU. • A transmission interrupt request is output when this bit and the TDRE bit are 1.

28.4.4 Reception and Transmission Data Register (RDR5 / TDR5)

The reception data register (RDR5) holds the received data. The transmission data register (TDR5) holds the transmission data. Both RDR5 and TDR5 registers are located at the same address.

Note: TDR5 is a write-only register and RDR5 is a read-only register. These registers are located in the same address, so the read value is different from the write value. Therefore, instructions that perform a read-modify-write (RMW) operation, such as the INC/DEC instruction, cannot be used.

Figure 28.4.4 Transmission and Reception Data registers 5 (RDR5 / TDR5)



■ **Reception:**

RDR5 is the register that contains reception data. The serial data signal transmitted to the SIN5 pin is converted in the shift register and stored there. When the data length is 7 bits, the uppermost bit (D7) contains 0. When reception is complete the data is stored in this register and the reception data full flag bit (SSR5: RDRF) is set to 1. If a reception interrupt request is enabled at this point, a reception interrupt occurs.

Read RDR5 when the RDRF bit of the status register (SSR5) is 1. The RDRF bit is cleared automatically to 0 when RDR5 is read. Also the reception interrupt is cleared if it is enabled and no error has occurred.

Data in RDR5 is invalid when a reception error occurs (SSR5: PE, ORE, or FRE = 1).

■ **Transmission:**

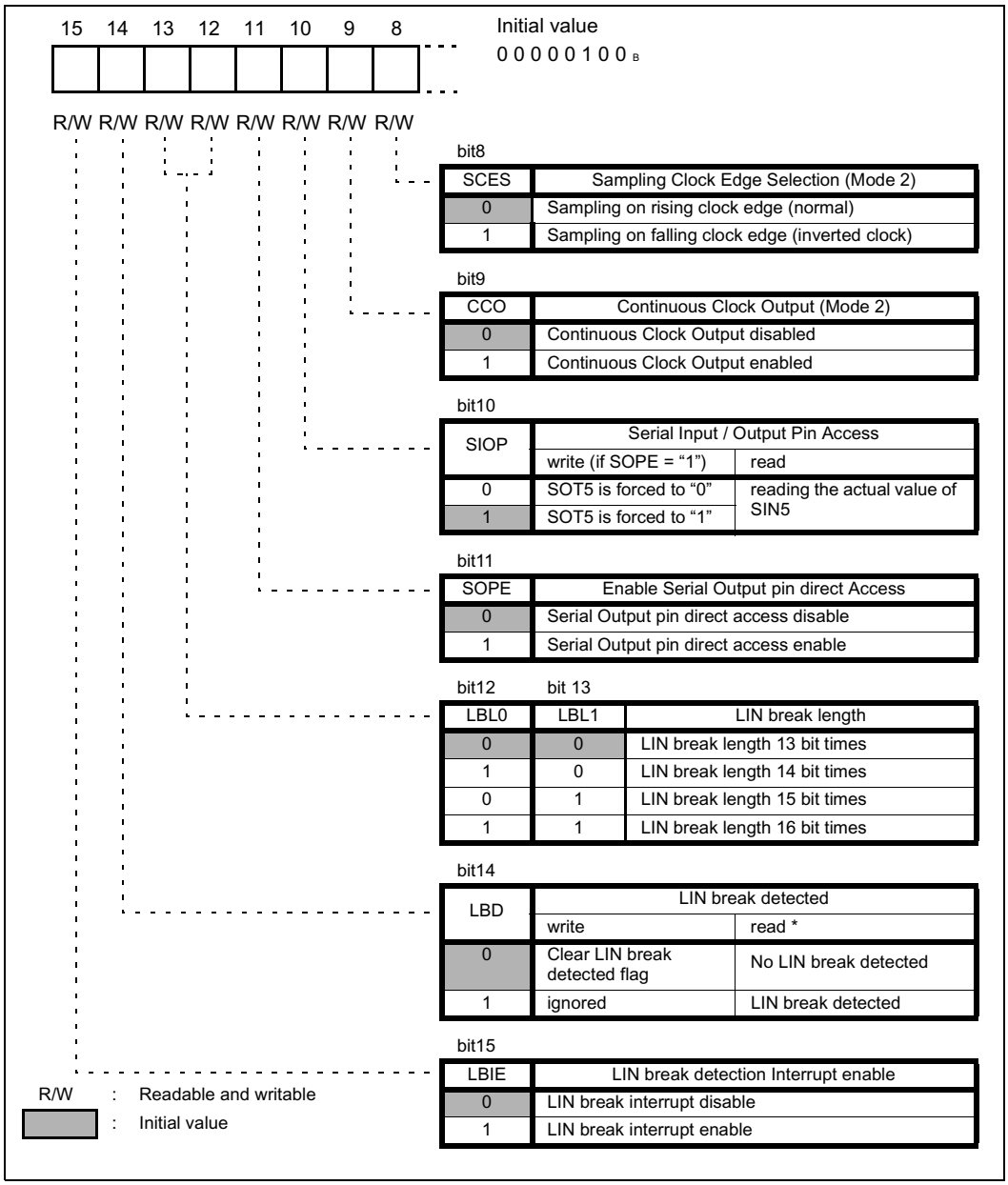
When data to be transmitted is written to the transmission data register in transmission enable state, it is transferred to the transmission shift register, then converted to serial data, and transmitted from the serial data output terminal (SOT5 pin). If the data length is 7 bits, the uppermost bit (D7) is not sent.

When transmission data is written to this register, the transmission data empty flag bit (SSR5: TDRE) is cleared to 0. When transfer to the transmission shift register is complete, the bit is set to 1. When the TDRE bit is 1, the next part of transmission data can be written. If output transmission interrupt requests have been enabled, a transmission interrupt is generated. Write the next part of transmission data when a transmission interrupt is generated or the TDRE bit is 1.

28.4.5 Extended Status/Control Register (ESCR5)

This register provides several LIN functions, direct access to the SIN5 and SOT5 pin and setting for USART synchronous clock mode.

Figure 28.4.5 Configuration of the Extended Status/Control Register (ESCR5)



bit8

SCES	Sampling Clock Edge Selection (Mode 2)
0	Sampling on rising clock edge (normal)
1	Sampling on falling clock edge (inverted clock)

bit9

CCO	Continuous Clock Output (Mode 2)
0	Continuous Clock Output disabled
1	Continuous Clock Output enabled

bit10

SIOP	Serial Input / Output Pin Access	
	write (if SOPE = “1”)	read
0	SOT5 is forced to “0”	reading the actual value of SIN5
1	SOT5 is forced to “1”	

bit11

SOPE	Enable Serial Output pin direct Access
0	Serial Output pin direct access disable
1	Serial Output pin direct access enable

bit12 bit 13

LBL0	LBL1	LIN break length
0	0	LIN break length 13 bit times
1	0	LIN break length 14 bit times
0	1	LIN break length 15 bit times
1	1	LIN break length 16 bit times

bit14

LBD	LIN break detected	
	write	read *
0	Clear LIN break detected flag	No LIN break detected
1	ignored	LIN break detected

bit15

LBIE	LIN break detection Interrupt enable
0	LIN break interrupt disable
1	LIN break interrupt enable

* see table 28.4.5a for RMW access!

Table 28.4.5a Function of each bit of the Extended Status/Control Register (ESCR5)

Bit name		Function
bit15	LBIE: LIN break detection interrupt enable bit	This bit enables a reception interrupt, if a LIN break was detected.
bit14	LBD: LIN break detected flag	This bit goes 1 if a LIN break was detected. Writing a 0 to it clears this bit and the corresponding interrupt, if it is enabled. **Note:** RMW instructions always return "1". In this case, the value "1" does not indicate a LIN-Break.
bit13 bit12	LBL1/0: LIN break length selection	These two bits determine how many serial bit times the LIN break is generated by USART. Receiving a LIN break is always fixed to 131bit times.
bit11	SOPE: Serial Output pin direct access enable*	Setting this bit to 1 enables the direct write to the SOT5 pin, if SOE = 1 (SMR5).*
bit10	SIOP: Serial Input/ Output Pin direct access*	Normal read instructions always return the actual value of the SIN5 pin. Writing to it sets the bit value to the SOT5 pin, if SOPE = 1. During a Read-Modify-Write instruction the bit returns the SOT5 value in the read cycle.*
bit9	CCO: Continuos Clock Output enable bit	This bit enables a continuos serial clock at the SCK5 pin if USART operates in master mode 2 (synchronous) and the SCK5 pin is configured as a clock output.
bit8	SCES: Serial clock edge selection bit	This bit inverts the internal serial clock in mode 2 (synchronous) and the output clock signal, if USART operates in master mode 2 (synchronous) and the SCK5 pin is configured as a clock output. In slave mode 2 the sampling time turns from rising edge to falling edge.

* see table 28.4.5b for SOPE and SIOP interaction

Table 28.4.5b * Description of the interaction of SOPE and SIOP:

SOPE	SIOP	Writing to SIOP	Reading from SIOP
0	R/W	has no effect on the SOT5 pin but holds the written value.	returns current value of SIN5
1	R/W	write "0" or "1" to SOT5	returns current value of SIN5
1	RMW		returns current value of SOT5 and writes it back

28.4.6 Extended Communication Control Register (ECCR5)

The extended communication control register provides bus idle recognition interrupt settings, synchronous clock settings, and the LIN break generation.

Figure 28.4.6 Configuration of the Extended Communication Control Register (ECCR5)

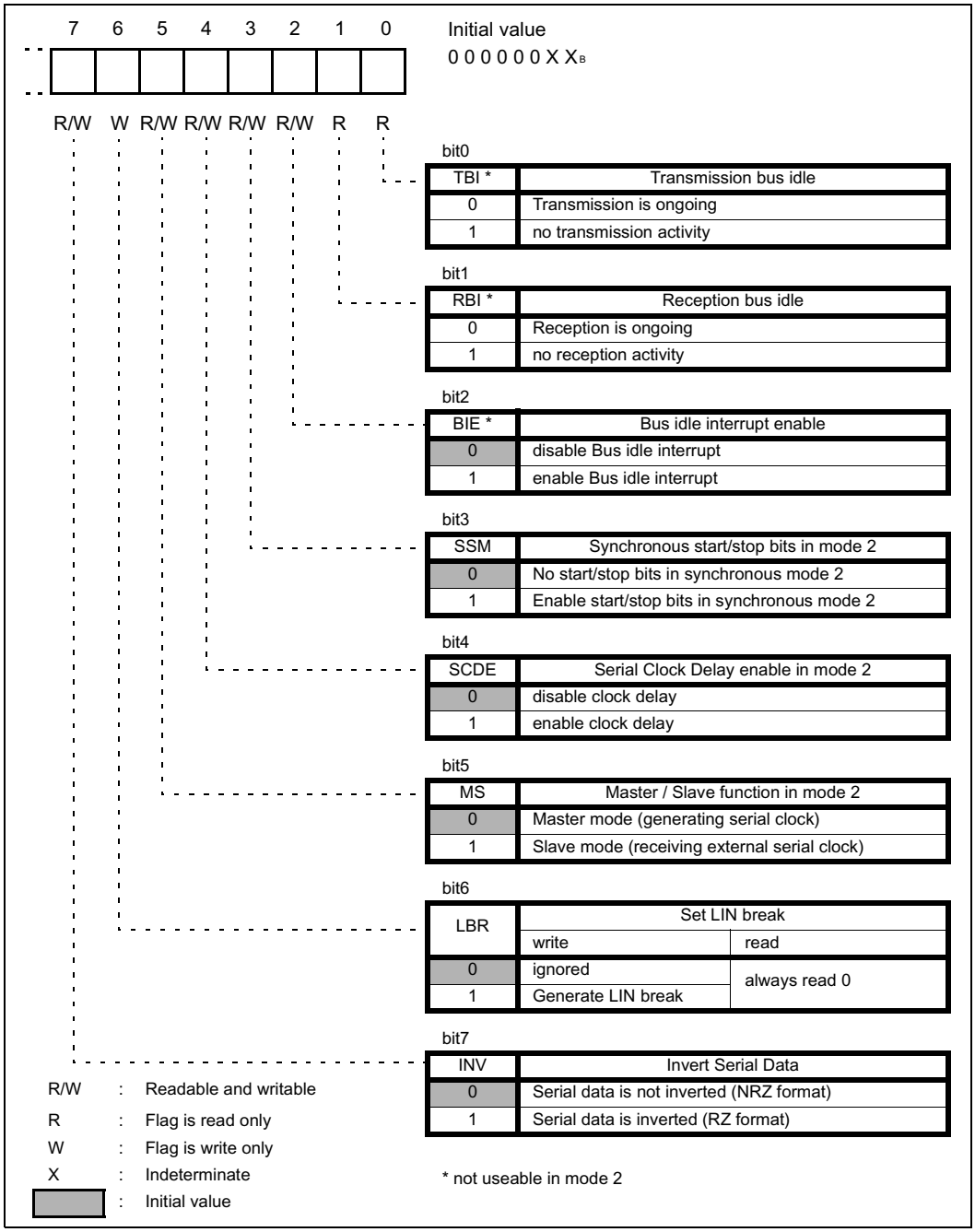


Table 28.4.6a Function of each bit of the Extended Communication Control Register (ECCR5)

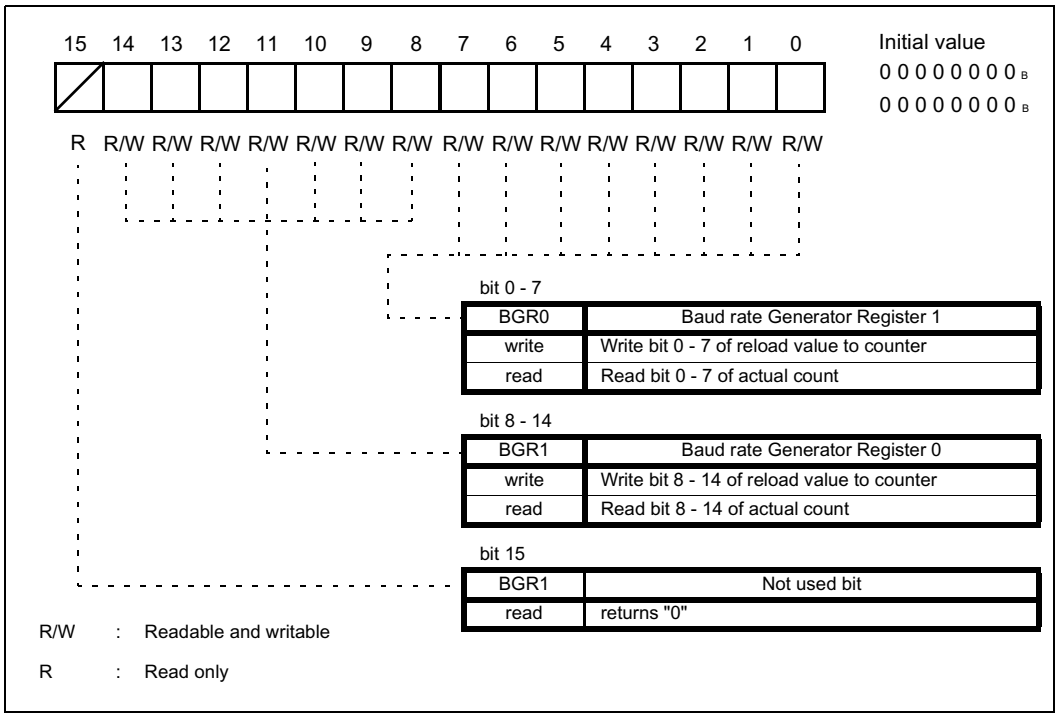
Bit name		Function
bit7	INV: Invert serial data	This bit inverts the serial data at SIN5 and SOT5 pin. SCK5 is not affected (see ESCR5: SCES). Writing "0": The serial data format is NRZ (default) Writing "1": The serial data is inverted (RZ format) RMW instructions do not affect this bit.
bit6	LBR: Set LIN break bit	Writing a 1 to this bit generates a LIN break of the length selected by the LBL0/1 bits of the ESCR5, if operation mode 0 or 3 is selected.
bit5	MS: Master/Slave mode selection bit	This bit selects master or slave mode of USART in synchronous mode 2. If master is selected USART generates the synchronous clock by itself. If slave mode is selected USART receives external serial clock. <Caution> If slave mode is selected, the clock source must be external and set to “One-to-One” (SMR5: SCKE = 0, EXT = 1, OTO = 1).
bit4	SCDE: Serial clock delay enable bit	If this bit is set, the serial output clock is delayed by 1 CLKP cycle (or half of its period in SPI-compliance*). This only applies, if USART operates in master mode 2.
bit3	SSM: Start/Stop bit mode enable	This bit adds start and stop bits to the synchronous data format in operation mode 2. It is ignored in mode 0, 1, and 3.
bit2	BIE: Bus idle interrupt enable	This bit enables a reception interrupt, if there is neither reception nor transmission ongoing (RBI = 1, TBI = 1). **Note:** Do not use BIE in mode 2.
bit1	RBI: Reception bus idle flag bit	This bit is “1” if there is no reception activity on the SIN5 pin. **Note:** Do not use this flag in mode 2.
bit0	TBI: Transmission bus idle flag bit	This bit is “1” if there is no transmission activity on the SOT5 pin. **Note:** Do not use this flag in mode 2.

* The USART in MB91360 devices cannot be made SPI-compliant.

28.4.7 Baud Rate / Reload Counter Register 0 and 1 (BGR0 / 1)

The baud rate / reload counter registers set the division ratio for the serial clock. Also the actual count of the transmission reload counter can be read.

Figure 28.4.7 Baudrate Reload Counter Register 0 and 1 (BGR0 / 1)



The Baud Rate / Reload Counter Registers determine the division ratio for the serial clock.

Both registers can be read or written via byte or word access.

28.5 USART INTERRUPTS

The USART uses both reception and transmission interrupts. An interrupt request can be generated for either of the following causes:

- **Receive data is set in the Reception Data Register (RDR5), or a reception error occurs.**
- **Transmission data is transferred from the Transmission Data Register (TDR5) to the transmission shift register.**
- **A LIN break is detected**
- **No bus activity (neither reception nor transmission)**

■ **USART Interrupts**

Table 28.5 Interrupt control bits and interrupt causes of USART

Reception/ transmission/ ICU	Interrupt request flag bit	Flag Register	Operation mode				Interrupt cause	Interrupt cause enable bit	How to clear the Interrupt Request
			0	1	2	3			
Reception	RDRF	SSR5	x	x	x	x	receive data is written to RDR	SSR5 : RIE	Receive data is read
	ORE	SSR5	x	x	x	x	Overrun error		"1" is written to clear rec. error bit (SSR5: CRE)
	FRE	SSR5	x	x	*	x	Framing error		
	PE	SSR5	x		*		Parity error		
	LBD	ESCR5	x			x	LIN synch break detected	ESCR5 : LBIE	"0" is written to ESCR5 : LBD
	TBI & RBI	ESCR5	x	x		x	no bus activity	ECCR5 : BIE	Receive data / Send data
Transmission	TDRE	SSR5	x	x	x	x	Empty transmission register	SSR5 : TIE	Transfer data is written
Input Capture Unit	ICP0	ICS01/23	x			x	1st falling edge of LIN synch field	ICS01/23 : ICE0	disable ICE0 temporary
	ICP0	ICS01/23	x			x	5th falling edge of LIN synch field	ICS01/23 : ICE0	disable ICE0

x : Used

* : Only available if ECCR5/SSM = 1

■ Reception Interrupt

If one of the following events occurs in reception mode, the corresponding flag bit of the Serial Status Register (SSR5) is set to "1":

- - Data reception is complete, i. e. the received data was transferred from the serial input shift register to the Reception Data Register (RDR5) and data can be read: **RDRF**
- - Overrun error, i. e. RDRF = 1 and RDR5 was not read by the CPU: **ORE**
- - Framing error, i. e. a stop bit was expected, but a "0"-bit was received: **FRE**
- - Parity error, i. e. a wrong parity bit was detected: **PE**

If at least one of these flag bits above go "1" and the reception interrupt is enabled (SSR5: RIE = 1), a reception interrupt request is generated.

If the Reception Data Register (RDR5) is read, the RDRF flag is automatically cleared to "0". Note that this is the *only* way to reset the RDRF flag. The error flags are cleared to "0", if a "1" is written to the Clear Reception Error (CRE) flag bit of the Serial Control Register (SCR5). The RDR5 contains only valid data if the RDRF flag is "1" and no error bits are set.

Note, that the CRE flag is "write only" and by writing a "1" to it, it is internally held to "1" for one CPU clock cycle.

■ Transmission Interrupt

If transmission data is transferred from the Transmission Data Register (TDR5) to the transfer shift register (this happens, if the shift register is empty and transmission data exists), the Transmission Data Register Empty flag bit (TDRE) of the Serial Status Register (SSR5) is set to "1". In this case an interrupt request is generated, if the Transmission Interrupt Enable (TIE) bit of the SSR5 was set to "1" before.

Note, that the initial value of TDRE (after hardware or software reset) is "1". So an interrupt is generated immediately then, if the TIE flag is set to "1". Also note, that the *only* way to reset the TDRE flag is writing data to the Transmission Data Register (TDR5).

■ LIN Synchronization Break Interrupt

This paragraph is only relevant, if USART operates in mode 0 or 3 as a LIN slave.

If the bus (serial input) goes "0" (dominant) for more than 11 bit times, the LIN Break Detected (LBD) flag bit of the Extended Status/Control Register (ESCR5) is set to "1". Note, that in this case after 9 bit times the reception error flags are set to "1", therefore the RIE flag has to set to "0" or the RXE flag has to set to "0", if only a LIN synch break detect is desired. In the other case a reception error interrupt would be generated first, and the interrupt handler routine has then to wait for LBD = 1.

The interrupt and the LBD flag are cleared after writing a "1" to the LBD flag. This makes it sure, that the CPU has detected the LIN synch break, because of the following procedure of adjusting the serial clock to the LIN master.

■ LIN Synchronization Field Edge Detection Interrupts

This paragraph is only relevant, if USART operates in mode 0 or 3 as a LIN slave. After a LIN break detection the next falling edge of the reception bus is indicated by USART. Simultaneously an internal signal connected to the ICU is set to "1". This signal is reset to "0" after the fifth falling edge of the LIN Synchronization Field. In both cases the ICU1/5 generates an interrupt, if "both edge detection" and the ICU1/5 interrupt are enabled. The difference of the ICU1/5 counter values is the serial clock multiplied by 8. Dividing it by 8 results in the new detected and calculated baud rate for the dedicated reload counter. This value - 1 has then to be written to the Baud Rate Generator Registers (BGR1/0).There is no need to restart the reload counter,

because it is automatically reset if a falling edge of a start bit is detected.

■ Bus Idle Interrupt

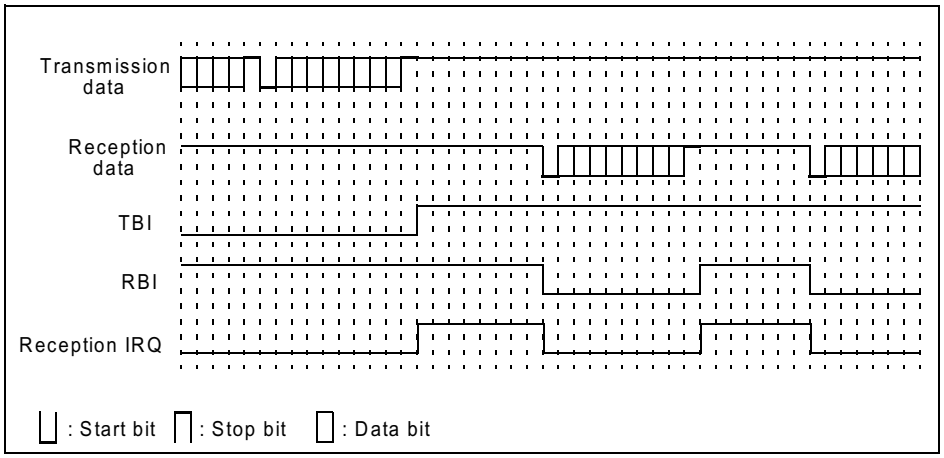
If there is no reception activity on the SIN5 pin, the RBI flag bit of the ECCR5 goes "1". The TBI flag bit respectively goes "1", when no data is transmitted. If the Bus Idle Interrupt Enable bit (BIE) of the ECCR5 is set and **both** bus idle flag bits (TBI **and** RBI) are "1", an interrupt is generated.

Note: The TBI flag goes also "0" if there is no bus activity, but a "0" is written to the SIOP bit, if SOPE is "1".

Note: TBI nd RBI cannot be used in mode 2 (synchronous communication).

Figure 28.5 illustrates how the bus idle interrupt is generated

Figure 28.5 Bus idle interrupt generation



28.5.1 Reception Interrupt Generation and Flag Set Timing

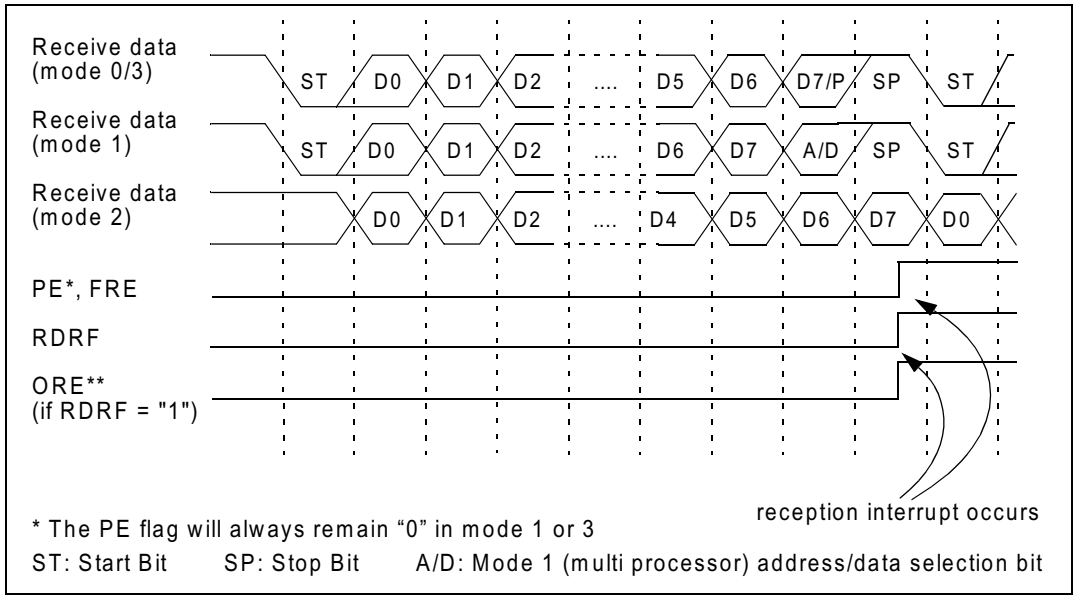
The following are the reception interrupt causes: Completion of reception (SSR5: RDRF) and occurrence of a reception error (SSR5: PE, ORE, or FRE).

■ **Reception Interrupt Generation and Flag Set Timing**

Generally a reception interrupt is generated, if the received data is complete (RDRF = 1) and the Reception Interrupt Enable (RIE) flag bit of the Serial Status Register (SSR5) was set to "1". This interrupt is generated if the first stop bit is detected in mode 0, 1, 2 (if SSM = 1), 3, or the last data bit was read in mode 2 (if SSM = 0).

Note: If a reception error has occurred, the Reception Data Register (RDR5) contains invalid data in each mode.

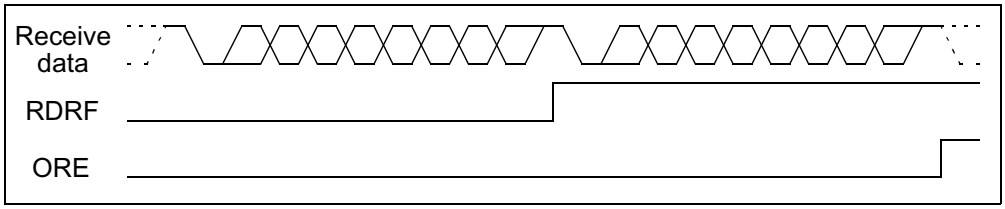
Figure 28.5.1a Reception operation and flag set timing



Note: The example in figure 28.5.1a does not show all possible reception options for mode 0 and 3. Here it is: "7p1" and "8N1" (p = "E" [even] or "O" [odd]), all in NRZ data format (ECCR5: INV = 0).

**ORE only occurs, if the reception data is not read by the CPU (RDRF = 1) and another data frame is read.

Figure 28.5.1b ORE set timing



28.5.2 Transmission Interrupt Generation and Flag Set Timing

A transmission interrupt is generated when the next data to be sent is ready to be written to the output data register (TDR5).

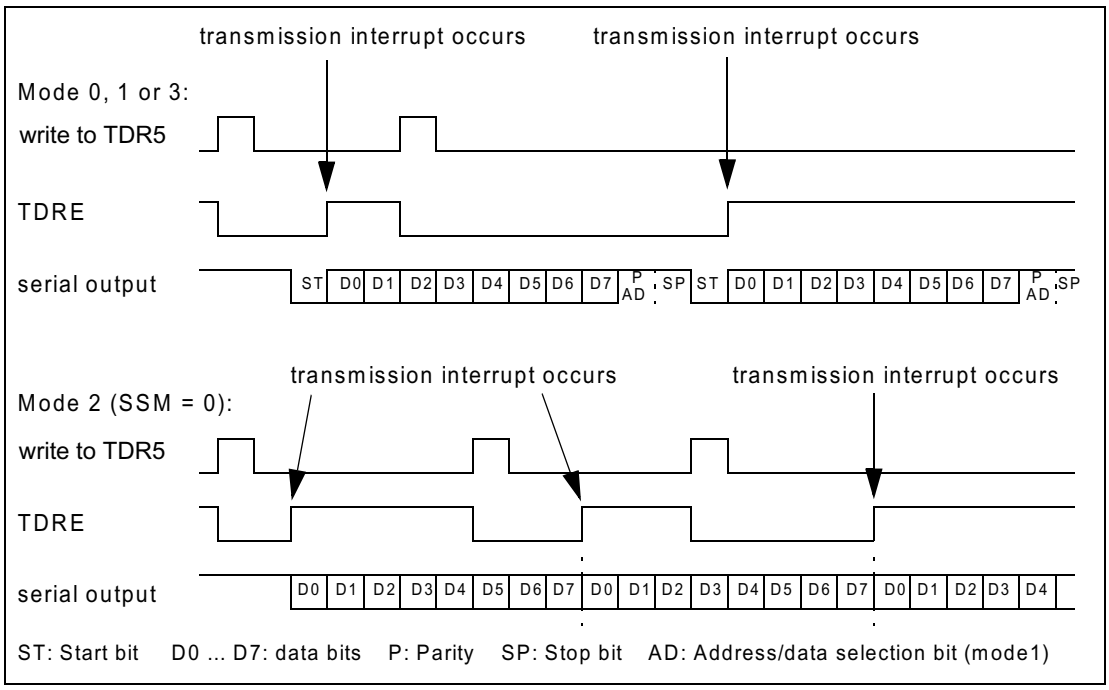
■ Transmission Interrupt Generation and Flag Set Timing

A transmission interrupt is generated, when the next data to be send is ready to be written to the Transmission Data Register (TDR5), i. e. the TDR5 is empty, and the transmission interrupt is enabled by setting the Transmission Interrupt Enable (TIE) bit of the Serial Status Register (SSR5) to "1".

The Transmission Data Register Empty (TDRE) flag bit of the SSR5 indicates an empty TDR5. Because the TDRE bit is "read only", it only can be cleared by writing data into TDR5.

The following figure demonstrates the transmission operation and flag set timing for the four modes of USART.

Figure 28.5.2a Transmission operation and flag set timing



Note: The example in figure 28.5.2a does not show all possible transmission options for mode 0. Here it is: "8p1" (p = "E" [even] or "O" [odd]), ECCR5: INV = 0. Parity is not provided in mode 3 or 2, if SSM = 0.

■ Transmission Interrupt Request Generation Timing

If the TDRE flag is set to 1 when a transmission interrupt is enabled (SSR5: TIE=1), transmission interrupt request is generated.

<Check>

A transmission completion interrupt is generated immediately after the transmission interrupt is enabled (TIE=1) because the TDRE bit is set to 1 as its initial value. TDRE is a read-only bit that can be cleared only by writing new data to the output data register (TDR5). Carefully specify the transmission interrupt enable timing.

28.6 USART BAUD RATES

One of the following can be selected for the USART serial clock source:

- **Dedicated baud rate generator (Reload Counter)**
- **External clock as it is (clock input to the SCK5 pin)**
- **External clock connected to the baud rate generator (Reload Counter)**

■ USART Baud Rate Selection

The baud rate selection circuit is designed as shown below. One of the following three types of baud rates can be selected:

- Baud Rates Determined Using the Dedicated Baud Rate Generator (Reload Counter)

  USART has two independent internal reload counters for transmission and reception serial clock. The baud rate can be selected via the 15-bit reload value determined by the Baud Rate Generator Register 0 and 1 (BGR0/1).

  The reload counter divides the peripheral clock by the value set in the Baud Rate Generator Register 0 and 1.

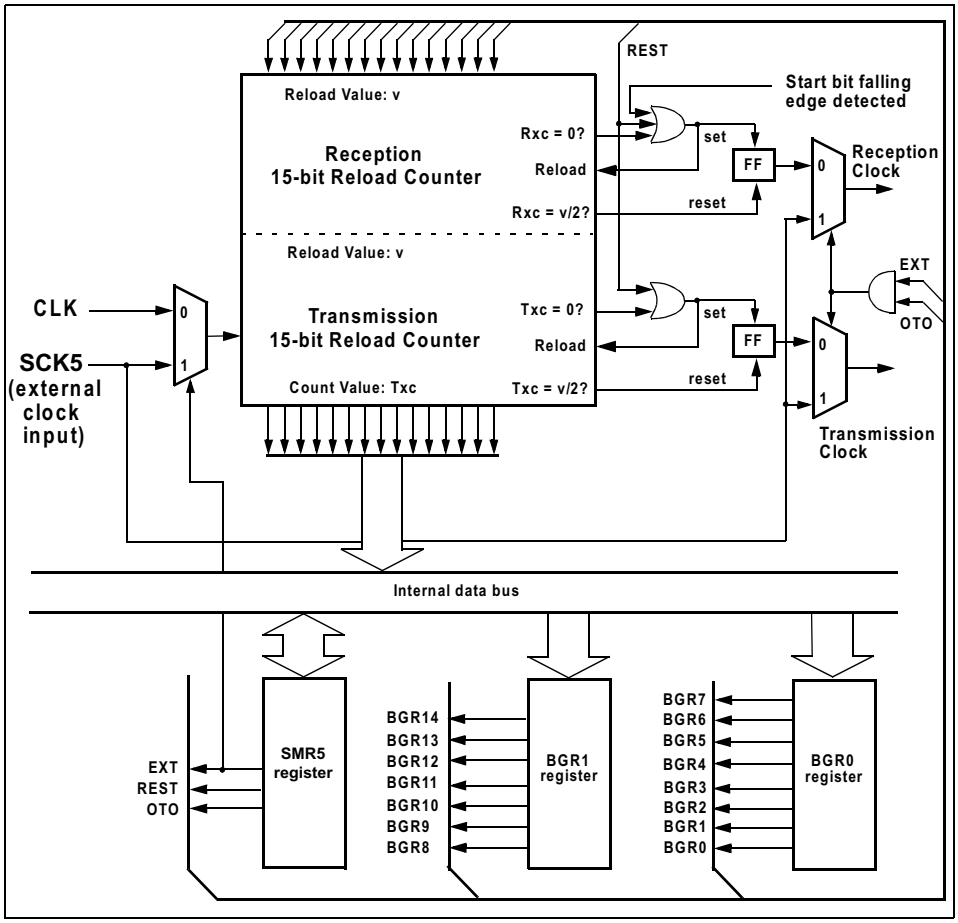
- Baud Rates determined using external clock (one-to-one mode)

  The clock input from USART clock pulse input pins (SCK5/P94) is used as it is (synchronous). Any baud rate less than the peripheral clock divided by 4 and is divisible can be set externally

- Baud Rates determined using the dedicated baud rate generator with external clock

  An external clock source can also be connected internally to the reload counter. In this mode it is used instead of the internal peripheral clock. This was designed to use quartz oscillators with special frequencies and having the possibility to divide them.

Figure 28.6 Baud rate selection circuit (reload counter)



28.6.1 Setting the Baud Rate

This section describes how the baud rates are set and the resulting serial clock frequency is calculated.

■ **Calculating the baud rate**

The both 15-bit Reload Counters are programmed by the Baud Rate Generator Registers 1 and 0 (BGR1, 0). The following calculation formula should be used to set the wanted baud rate:

Reload Value:

$$v = [\Phi / b] - 1 \quad ,$$

where Φ is the resource clock (CLKP), b the baud rate and [ ] gaussian brackets (mathematical rounding function).

■ **Example of Calculation**

If the CPU clock is 16 MHz and the desired baud rate is 19200 baud then the reload value v is:

$v = [16*10^6 / 19200] - 1 =$ **832**

The exact baud rate can then be recalculated: $b_{exact} = \Phi / (v + 1)$, here it is: $16*10^6 / 833 = 19207.6831$

Note: Setting the reload value to 0 stops the reload counter. For this reason the minimum division ratio is 2.

■ Suggested Division Ratios for different machine speeds and baud rates

The following settings are suggested for different MCU clock speeds and baud rates:

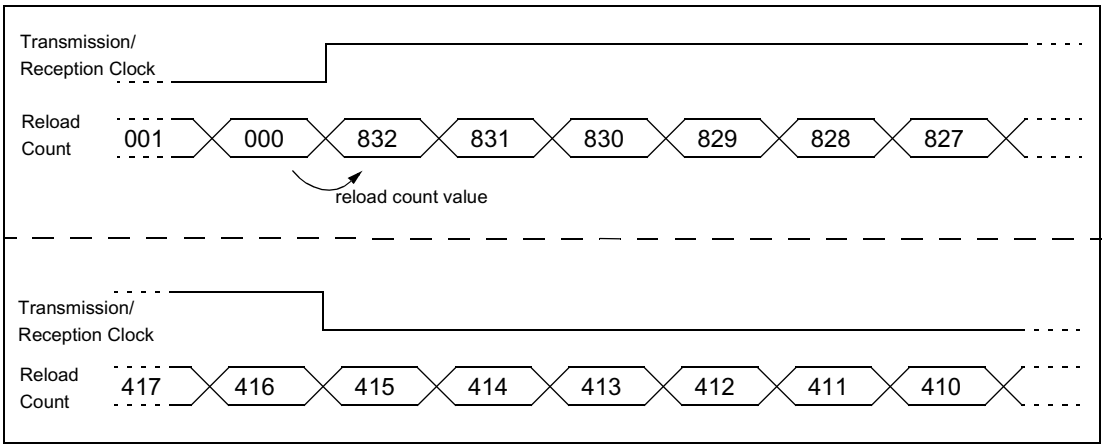
Table 28.6.1 Suggested Baud Rates and reload values at different machine speeds.

Baud rate	8 MHz		10 MHz		16 MHz		20 MHz		24 MHz		32 MHz	
	value	% dev.	value	% dev.	value	% dev.	value	% dev.	value	% dev.	value	% dev.
2M	3	0	4	0	7	0	9	0	11	0	15	0
1M	7	0	9	0	15	0	19	0	23	0	31	0
500000	15	0	19	0	31	0	39	0	47	0	63	0
460800	-	-	-	-	-	-	-	-	51	-0.16	-	-
250000	31	0	39	0	63	0	79	0	95	0	127	0
230400	-	-	-	-	-	-	86	0.22	103	-0.16	138	-0.08
153600	51	-0.16	64	-0.16	103	-0.16	129	-0.16	155	-0.16	207	0.16
125000	63	0	79	0	127	0	159	0	191	0	255	0
115200	-	-	86	0.22	138	0.08	173	0.22	207	-0.16	278	-0.08
76800	103	-0.16	129	-0.16	207	-0.16	259	-0.16	311	-0.16	416	0.08
57600	138	0.08	173	0.22	277	0.08	346	-0.06	416	0.08	555	0.08
38400	207	-0.16	259	-0.16	416	0.08	520	0.03	624	0	832	-0.04
28800	277	0.08	346	<0.01	554	-0.01	693	-0.06	832	-0.03	1110	-0.01
19200	416	0.08	520	0.03	832	-0.03	1041	0.03	1249	0	1666	0.02
10417	767	<0.01	959	<0.01	1535	<0.01	1919	<0.01	2303	<0.01	3071	<0.01
9600	832	0.04	1041	0.03	1666	0.02	2083	0.03	2499	0	3332	0.01
7200	1110	<0.01	1388	<0.01	2221	<0.01	2777	<0.01	3332	<0.01	4443	-0.01
4800	1666	0.02	2082	-0.02	3332	<0.01	4166	<0.01	4999	0	6666	<0.01
2400	3332	<0.01	4166	<0.01	6666	<0.01	8332	<0.01	9999	0	13332	<0.01
1200	6666	<0.01	8334	0.02	13332	<0.01	16666	<0.01	19999	0	26666	<0.01
600	13332	<0.01	16666	<0.01	26666	<0.01	-	-	-	-		
300	26666	<0.01	-	-	-	-	-	-	-	-		

■ Counting Example

Assume the reload value is 832. The figure 28.6.1 demonstrates the behavior of the both Reload Counters:

Figure 28.6.1 Counting example of the reload counters



Note: The falling edge of the Serial Clock Signal always occurs after | (v + 1) / 2 |.

28.6.2 Restarting the Reload Counter

The Reload Counter can be restarted of the following reasons:

Transmission and Reception Reload Counter:

- Global MCU Reset
- USART programmable clear (SMR5:UPCL bit)
- User programmable restart (SMR5: REST bit)

Reception Reload Counter:

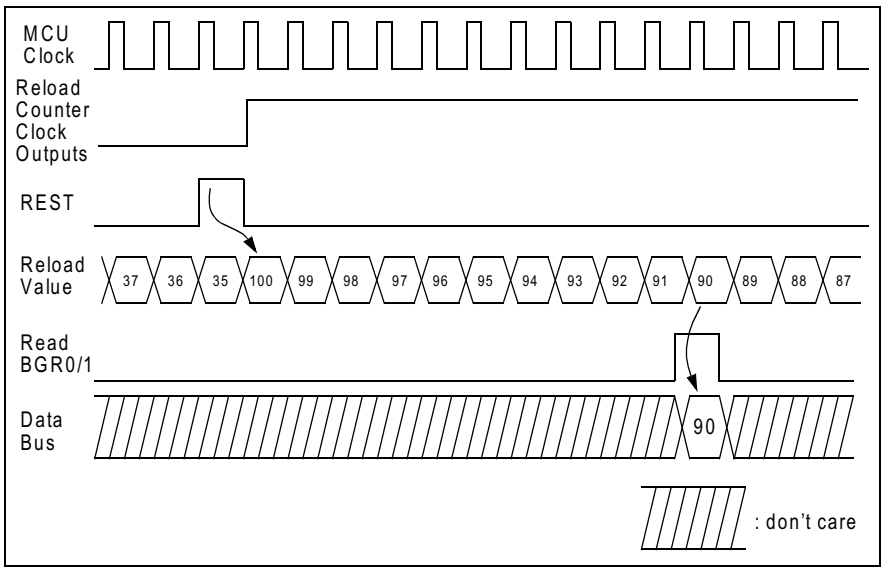
- Start bit falling edge detection in asynchronous mode

■ Programmable Restart

If the REST bit of the Serial Mode Register (SMR5) is set by the user, both Reload Counters are restarted at the next clock cycle. This feature is intended to use the Transmission Reload Counter as a small timer.

The following figure illustrates a possible usage of this feature (assume that the reload value is 100.)

Figure 28.6.2a Reload Counter Restart example



In this example the number of MCU clock cycles (cyc) after REST is then:

***cyc* = *v* - *c* + 1 = 100 - 90 + 1 = 11**,

where *v* is the reload value and *c* is the read counter value.

Note: If USART is reset by setting SMR5:UPCL, the Reload Counters will restart too.

■ Automatic Restart

In asynchronous UART mode if a falling edge of a start bit is detected the Reception Reload Counter is restarted. This is intended to synchronize the serial input shifter to the incoming serial data stream.

28.7 OPERATION OF USART

USART operates in operation mode 0 for normal bidirectional serial communication, in mode 2 and 3 in bidirectional communication as master or slave, and in mode 1 as master or slave in multiprocessor communication.

■ Operation of USART

● Operation modes

There are four USART operation modes: modes 0 to 3. As listed in table 28.7, an operation mode can be selected according to the inter-CPU connection method and data transfer mode.

Table 28.7 USART operation mode

Operation mode		Data length		Synchronization of mode	Length of stop bit	data bit direction*
		parity disabled	parity enabled			
0	normal mode	7 or 8		asynchronous	1 or 2	L/M
1	multiprocessor	7 or 8 + 1**	-	asynchronous	1 or 2	L/M
2	normal mode	8		synchronous	0, 1 or 2	L/M
3	LIN mode	8	-	asynchronous	1	L

* means the data bit transfer format: LSB or MSB first

** "+1" means the indicator bit of the address/data selection in the multiprocessor mode, instead of parity.

Note: Mode 1 operation is supported both for master or slave operation of USART in a master-slave connection system. In Mode 3 the USART function is locked to 8N1-Format, LSB first.

If the mode is changed, USART cuts off all possible transmission or reception and awaits then new action.

■ Inter-CPU Connection Method

External Clock One-to-one connection (normal mode) and master-slave connection (multiprocessor mode) can be selected. For either connection method, the data length, whether to enable parity, and the synchronization method must be common to all CPUs. Select an operation mode as follows:

- In the one-to-one connection method, operation mode 0 or 2 must be used in the two CPUs. Select operation mode 0 for asynchronous transfer mode and operation mode 2 for synchronous transfer mode.

  Note, that one CPU has to set to the master and one to the slave in synchronous mode 2.

- Select operation mode 1 for the master-slave connection method and use it either for the master or slave system.

■ Synchronization Methods

In asynchronous operation USART reception clock is automatically synchronized to the falling edge of a received start bit.

In synchronous mode the synchronization is performed either by the clock signal of the master device or by USART itself if operating as master.

■ Signal Mode

USART can treat data in non-return to zero (NRZ) and return to zero (RZ) format. For this option the ECCR: INV bit is provided.

■ Operation Enable Bit

USART controls both transmission and reception using the operation enable bit for transmission (SCR5: TXE) and reception (SCR5: RXE). If each of the operations is disabled, stop it as follows:

- If reception operation is disabled during reception (data is input to the reception shift register), finish frame reception and read the received data of the reception data register (RDR5). Then stop the reception operation.
- If the transmission operation is disabled during transmission (data is output from the transmission shift register), wait until there is no data in the transmission data register (TDR5) before stopping the transmission operation.

28.7.1 Operation in Asynchronous Mode (Op. Modes 0 and 1)

When USART is used in operation mode 0 (normal mode) or operation mode 1 (multiprocessor mode), the asynchronous transfer mode is selected.

■ Transfer data format

Generally each data transfer in the asynchronous mode operation begins with the start bit (low-level on bus) and ends with at least one stop bit (high-level). The direction of the bit stream (LSB first or MSB first) is determined by the BDS-Bit of the Serial Status Register (SSR5). The parity bit (if enabled) is always placed between the last data bit and the (first) stop bit.

In operation mode 0 the length of the data frame can be 7 or 8 bits, with or without parity, and 1 or 2 stop bits.

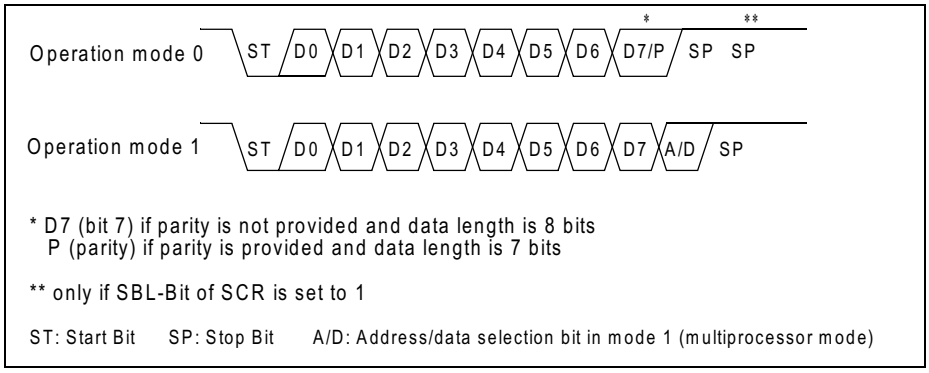
In operation mode 1 the length of the data frame can be 7 or 8 bits with a following address-/ data-selection bit instead of a parity bit. 1 or 2 stop bits can be selected.

The calculation formula for the bit length of a transfer frame is:

Length = 1 + d + p + s

(d = number of data bits [7 or 8], p = parity [0 or 1], s = number of stop bits [1 or 2]

Figure 28.7.1 Transfer data format (operation modes 0 and 1)



Note: If BDS-Bit of the Serial Status Register (SSR5) is set to "1" (MSB first), the bit stream processes as: D7, D6, ... , D1, D0, (P).

During Reception both stop bits are detected, if selected. But the Reception data register full (RDRF) flag will go "1" at the first stop bit. The bus idle flag (RBI of ECCR5) goes "1" after the second stop bit if no further start bit is detected. (The second stop bit belongs to "bus activity", although it is just mark level.)

■ Transmission Operation

If the Transmission Data Register Empty (TDRE) flag bit of the Serial Status Register (SSR5) is "1", transmission data is allowed to be written to the Transmission Data Register (TDR5). When data is written, the TDRE flag goes "0". If the transmission operation is enabled by the TXE-Bit ("1") of the Serial Control Register (SCR5), the data is written next to the transmission shift register and the transmission starts at the next clock cycle of the serial clock, beginning with the start bit. Thereby the TDRE flag goes "1", so that new data can be written to the TDR5.

If transmission interrupt is enabled (TIE = 1), the interrupt is generated by the TDRE flag. Note, that the initial value of the TDRE flag is "1", so that in this case if TIE is set to "1" an interrupt will occur immediately.

■ Reception Operation

Reception operation is performed every time it is enabled by the Reception Enable (RXE) flag bit of the SCR5. If a start bit is detected, a data frame is received according to the format specified by the SCR5. By occurring errors, the corresponding error flags are set (PE, ORE, FRE). However after the reception of the data frame the data is transferred from the serial shift register to the Reception Data Register (RDR5) and the Receive Data Register Full (RDRF) flag bit of the SSR5 is set. The data then has to be read by the CPU. By doing so, the RDRF flag is cleared. If reception interrupt is enabled (RIE = 1), the interrupt is simply generated by the RDRF.

Note: Only when the RDRF flag bit is set and no errors have occurred the Reception Data Register (RDR5) contains valid data.

■ Stop Bit, Error Detection, and Parity

For transmission, 1 or 2 stop bits can be selected. During reception, if selected, both stop bits are checked, to set the reception bus idle (RBI) flag of ECCR5 correctly after the second stop bit.

In mode 0 parity, overrun, and framing errors can be detected.

In mode 1, overrun and framing errors can be detected. Parity is not provided.

By setting the Parity Enable (PEN) bit of the Serial Control Register (SCR5) the USART provides parity calculation (during transmission) and parity detection and check (during reception) in mode 0 (and mode 2 if the SSM bit of ECCR5 is set).

Even parity is set, if the P bit of SCR5 is cleared, odd parity if the flag bit is set.In mode 1, overrun and framing errors can be detected. Parity is not provided.

■ Signal mode NRZ and RZ

To set USART to the NRZ data format set the ECCR5:INV bit to 0 (initial value).
RZ data format is set, if the ECCR5:INV bit was set to 1.

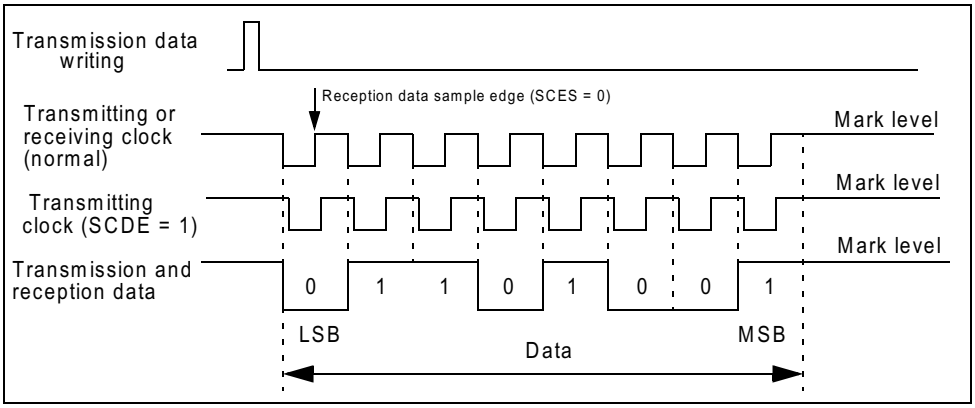
28.7.2 Operation in Synchronous Mode (Operation Mode 2)

The clock synchronous transfer method is used for USART operation mode 2 (normal mode).

■ Transfer data format (standard synchronous)

In the synchronous mode, 8-bit data is transferred with no start or stop bits if the SSM bit of the Extended Communication Control Register (ECCR5) is 0. To the data format in mode 2 belongs a special clock signal. The figure below illustrates the data format during a transmission in the synchronous operation mode

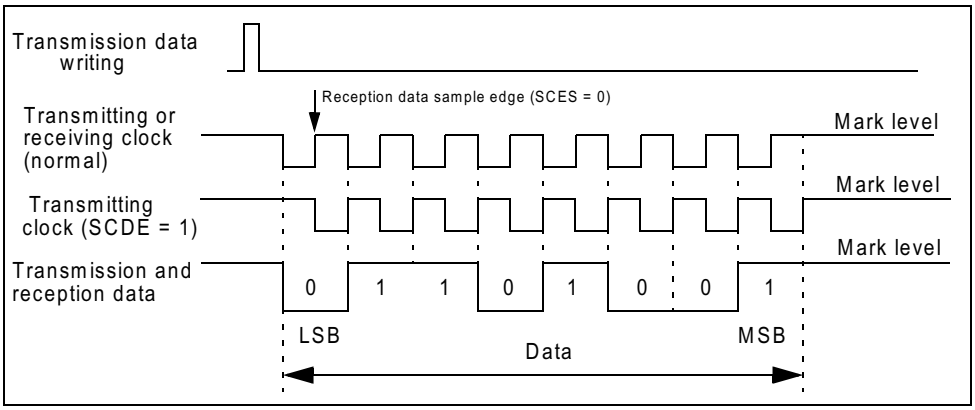
Figure 28.7.2a Transfer data format (operation mode 2).



■ Transfer data format (SPI, not supported in MB91360 Series)

In the synchronous mode, 8-bit data is transferred with no start or stop bits if the SSM bit of the Extended Communication Control Register (ECCR5) is 0. To the data format in mode 2 belongs a special clock signal. The figure below illustrates the data format during a transmission in the synchronous operation mode.

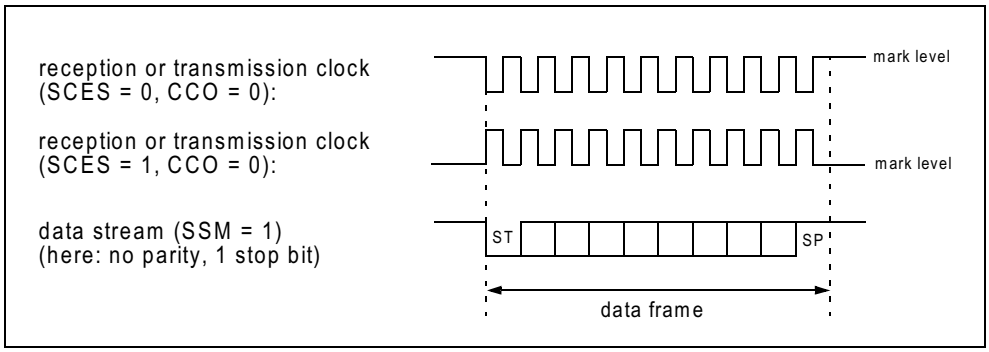
Figure 28.7.2b SPI Transfer data format (operation mode 2)



■ Clock inversion and start/stop bits in mode 2

If the SCES bit of the Extended Status/Control Register (ESCR5) is set the serial clock is inverted. Therefore in slave mode USART samples the data bits at the falling edge of the received serial clock. Note, that in master mode if SCES is set the clock signal's mark level is "0". If the SSM5 bit of the Extended Communication Control Register (ECCR5) is set the data format gets additional start and stop bits like in asynchronous mode

Figure 28.7.2c Transfer data format with clock inversion



■ Clock Supply

In clock synchronous (normal) mode (I/O extended serial), the number of the transmission and reception bits has to be equal to the number of clock cycles. Note, that if start/stop bits communication is enabled, the number of clock cycles has to match with the quantity for the additional start and stop bit(s).

If the internal clock (dedicated reload counter) is selected, the data receiving synchronous clock is generated automatically if data is transmitted.

If external clock is selected, be sure, that the transmission side of the Transmission Data Register contains data and then clock cycles for each bit to sent have to be generated and supplied from outside. The mark level ("H") must be retained before transmission starts and after it is complete if SCES is “0”.

Setting the SCDE bit of ECCR delays the transmitting clock signal by 1 CLKP cycle (or half a clock period in SPI). This will make sure, that the transmission data is valid and stable at any falling clock edge. (Necessary, if the receiving device samples the data at falling clock edge). This function is disabled when CCO is enabled.

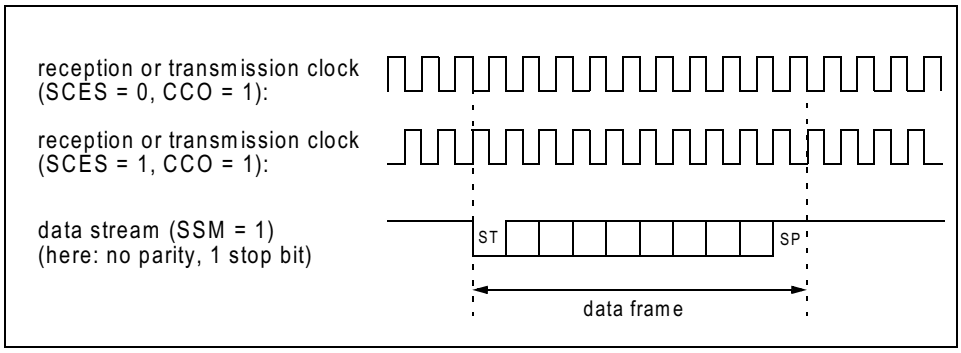
Note: The USART in MB91360 devices cannot be made SPI-compliant.

If the Serial Clock Edge Select (SCES) bit of the ESCR is set, the USARTs clock is inverted and thus samples the reception data at the falling clock edge. In this case, the sending device must make sure that the serial data is valid at the falling serial clock edge.

When both the SCES and the SCDE bit are set, data is stable at the rising clock edge, as in the case of SCES = SCDE = 0. However, the marker value for idle state is inverted (low).

If the CCO bit of the Extended Status/Control Register (ESCR5) is set, the serial clock on the SCK5 pin in master mode is continuously clocked out. It is strongly recommended to use start and stop bits in this mode to signalize the receiver, when a data frame begins and when it stops. Figure 28.7.2d illustrates this.

Figure 28.7.2d Continuous clock output in mode 2



■ Data signal mode

NRZ data format is selected, if ECCR5: INV = 0, otherwise the signal mode for the serial data input and output pin is RZ.

■ Error Detection

If no Start/Stop bits are selected (ECCR5: SSM = 0) only overrun errors are detected.

■ Communication

For initialization the synchronous mode, the following settings have to be done:

- Baud Rate Generator Registers (BGR0/1):

  Set the desired reload value for the dedicated Baud Rate Reload Counter

- Serial Mode Control Register (SMR5):
  - MD1, MD0: "10b" (Mode 2)
  - SCKE: “1” for dedicated Baud Rate Reload Counter
    “0” for external clock input
  - SOE: “1” for transmission and reception
    “0” for reception only
- Serial Control Register (SCR5):
  - RXE, TXE: one of these flag bit is set to "1"
  - PEN: no parity provided - Value: don’t care
  - P, SBL, A/D: no parity, no stop bit(s), no Address/Data selection - Value: don’t care
  - CL: automatically fixed to 8-bit data - Value: don’t care
  - CRE: "1" (the error flag is cleared for initialization, possible transmission or reception will cut off)
- Serial Status Register (SSR5):
  - BDS: "0" for LSB first,
    "1" for MSB first
  - RIE: "1" if interrupts are used
    "0" if not
  - TIE: "1" if interrupts are used
    "0" if not

- Extended Communication Register (ECCR5):
  - SSM: "0" if no start/stop bits are desired (normal)
    "1" for adding start/stop bits (special)
  - MS: "0" for master mode (USART generates the serial clock);
    "1" for slave mode (USART receives serial clock from the master device)

To start the communication, write data into the Transmission Data Register (TDR5).

To receive data, disable the Serial Output Enable (SOE) bit of the SMR5 and write dummy data to TDR5.

Note: Setting continuous clock and start-/stop-bit mode, duplex transception is possible like in asynchronous modes.

28.7.3 Operation with LIN Function (Operation Mode 3)

USART can be used either for LIN-Master devices or LIN-Slave devices. For this LIN function a special mode (3) is provided. Setting the USART to mode 3, configure the data format to 8N1-LSB-first format.

■ USART as LIN master

In LIN master mode, the master determines the baud rate of the whole sub bus. Therefore, slave devices have to synchronize to the master and the desired baud rate remains fixed in master operation after initialization.

Writing a "1" into the LBR bit of the Extended Status/Communication Register (ECCR5) generates a 13 - 16 bit times low-level on the SOT5 pin, which is the LIN synchronization break and the start of a LIN message. Thereby the TDRE flag of the Serial Status Register (SSR5) goes "0" and is reset to "1" after the break, and generates a transmission interrupt for the CPU (if TIE of SSR5 is "1").

The length of the Synchronization break to be sent can be determined by the LBL1/0 bits of the ESCR5 as follows:

Table 28.7.3 LIN break length

LBL1	LBL0	Length of Break
0	0	13 Bit times
0	1	14 Bit times
1	0	15 Bit times
1	1	16 Bit times

The Synch Field can be sent as a simple 0x55-Byte after the LIN break. To prevent a transmission interrupt, the 0x55 can be written to the TDR5 just after writing the "1" to the LBR bit, although the TDRE flag is "0". The internal transmission shifter waits until the LIN break has finished and shifts the TDR5 value out afterwards. In this case no interrupt is generated after the LIN break and before the start bit.

■ USART as LIN slave

In LIN slave mode USART has to synchronize to the master's baud rate. If Reception is disabled (RXE = 0) but LIN break Interrupt is enabled (LBIE = 1) USART will generate an reception interrupt, if a synchronization break of the LIN master is detected, and indicates it with the LBD flag of the ESCR5. Writing a "0" to this bit clears the interrupt. The next step is to analyze the baud rate of the LIN master. The first falling edge of the Synch Field is detected by USART. The USART signals it then to the Input Capture Unit (ICU1/5) via a rising edge of an internal connection. The fifth falling edge resets the ICU signal. Therefore the ICU has to be configured for the LIN input capture (ICE01/45) and its interrupts have to be enabled (ICS01/45). The values of th ICU counter register after the first Interrupt (a) and after the second interrupt (b) yield the BGR value:

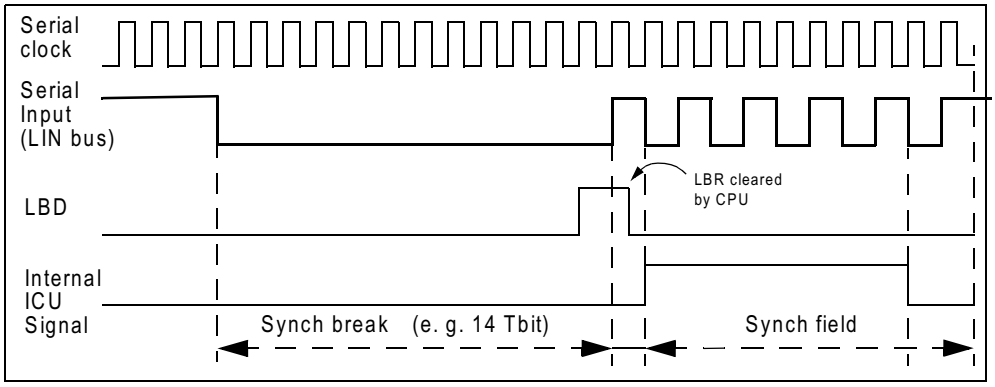
without timer overflow: BGR value = (b - a) / 8 ,

with timer overflow: BGR value = (max - b + a) / 8 ,

where max is the timer maximum value at which the overflow occurs.

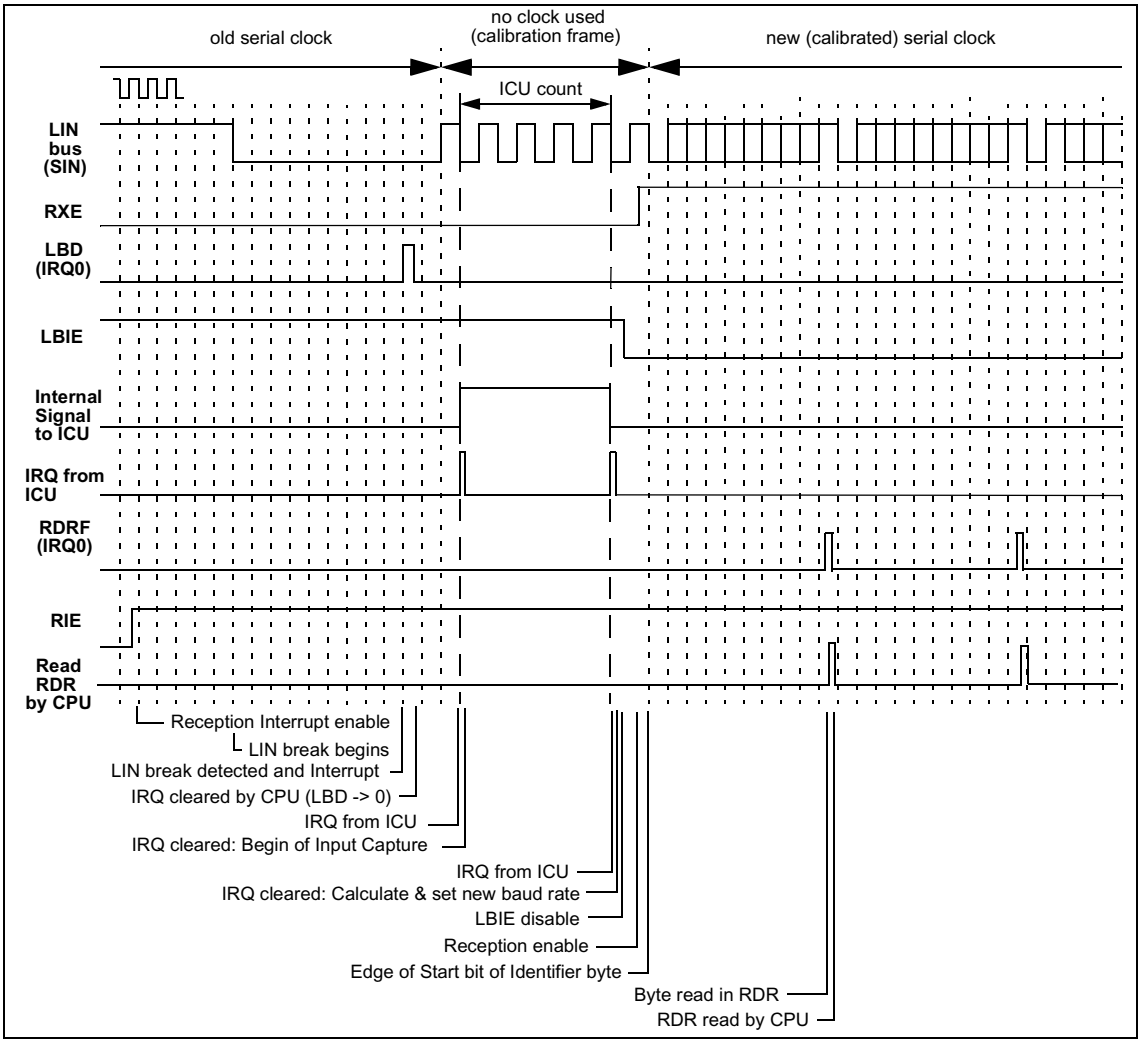
The figure 28.7.3a shows a typical start of a LIN message frame and the behavior of the USART.

Figure 28.7.3a USART behavior as slave in LIN mode



■ LIN bus timing

Figure 28.7.3b LIN bus timing and USART signals



28.7.4 Direct Access to Serial Pins

USART allows the user directly to access to the transmission pin (SOT5) or the reception pin (SIN5).

■ USART Direct Pin Access

The USART provides the ability for the Programmer to access directly to serial input or output pin. The software can always monitor the incoming serial data by reading the SIOP bit of the ESCR5. If setting the Serial Output Pin direct access Enable (SOPE) bit of the ESCR5 the software can force the SOT5 pin to a desired value. Note, that this access is only possible, if the transmission shift register is empty (i. e. no transmission activity).

In LIN mode this function can be used for reading back the own transmission and is used for error handling if something is physically wrong with the single-wire LIN-bus.

Note: Write the desired value to the SIOP pin **before** enabling the output pin access, to prevent undesired peaks. The peaks can occur because SIOP holds the last written value.

<check>

During a Read-Modify-Write operation the SIOP bit returns the actual value of the SOT5 pin in the read cycle instead the value of SIN5 during a normal read instruction.

28.7.5 Bidirectional Communication Function (Normal Mode)

In operation mode 0 or 2, normal serial bidirectional communication is available. Select operation mode 0 for asynchronous communication and operation mode 2 for synchronous communication.

■ Bidirectional Communication Function

The settings shown in figure 28.7.5a are required to operate USART in normal mode (operation mode 0 or 2).

Figure 28.7.5a Settings for USART operation mode 0 and 2

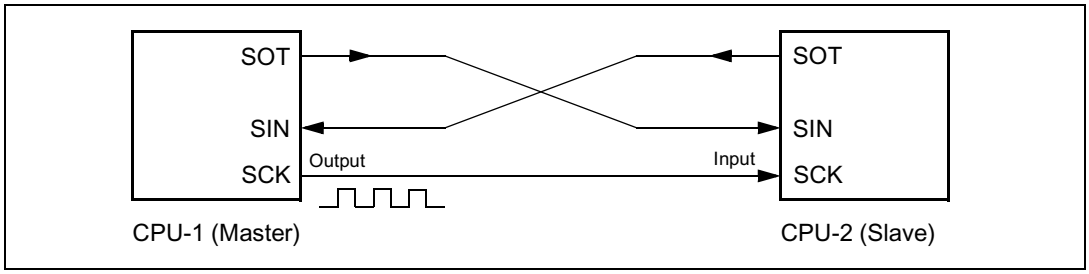
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR5,SMR5	PEN	P	SBL	CL	AD	CRE	RXE	TXE	MD1	MD0	OTO	EXT	REST	UPCL	SCKE	SOE
Mode 0	◎	◎	◎	◎	x	0	◎	◎	0	0	x	0	0	0	◎	◎
Mode 2	□	□	□	x	x	0	◎	◎	1	0	◎	◎	0	0	◎	◎
SSR5, TDR5/RDR5	PE	ORE	FRE	RDRF	TDRE	BDS	RIE	TIE	Set transmission data (during writing) Retain reception data (during reading)							
Mode 0	◎	◎	◎	◎	◎	◎	◎	◎								
Mode 2	□	◎	□	◎	◎	◎	◎	◎								
ESCR5,ECCR5	LBIE	LBD	LBL1	LBL0	SOPE	SIOP	CCO	SECS	-	LBR	MS	SCDE	SSM	BIE	RBI	TBI
Mode 0	x	x	x	x	x	x	x	x		x	x	x	x	x	x	x
Mode 2	x	x	x	x	x	x	◎	◎		x	◎	◎	◎	x	x	x

◎ Bit is used
x Bit is not used
0 / 1 Set bit to 0 / 1
□ Bit is used if SSM = 1 (Synchronous start-/stop-bit mode)

■ Inter-CPU Connection

As shown in figure 28.7.5b, interconnect two CPUs in USART mode 2

Figure 28.7.5b Connection example of USART mode 2 bidirectional communication



28.7.6 Master-Slave Communication Function (Multiprocessor Mode)

USART communication with multiple CPUs connected in master-slave mode is available for both master or slave systems.

■ Master-slave Communication Function

The settings shown in figure 28.7.6a are required to operate USART in multiprocessor mode (operation mode 1).

Figure 28.7.6a Settings for USART operation mode 1

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR5,SMR5	PEN	P	SBL	CL	AD	CRE	RXE	TXE	MD1	MD0	OTO	EXT	REST	UPCL	SCKE	SOE
Mode 1	x	x	◎	◎	◎	0	◎	◎	0	1	x	0	0	0	1	◎

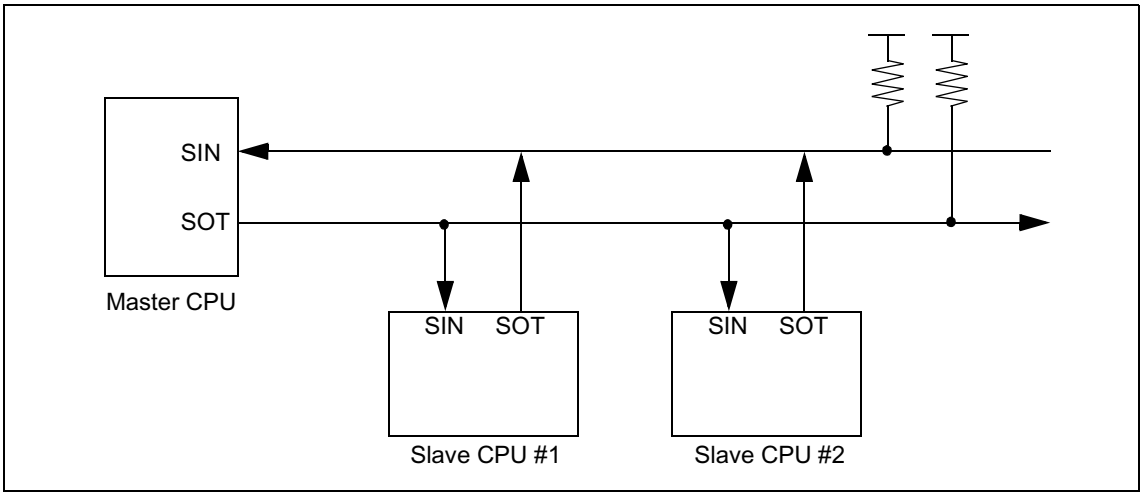
SSR5, TDR5/RDR5	PE	ORE	FRE	RDRF	TDRE	BDS	RIE	TIE	Set transmission data (during writing) Retain reception data (during reading)
Mode 1	x	◎	◎	◎	◎	◎	◎	◎	

◎ Bit is used
x Bit is not used
0 / 1 Set bit to 0 / 1

■ Inter-CPU Connection

As shown in figure 28.7.6b, a communication system consists of one master CPU and multiple slave CPUs connected to two communication lines. USART can be used for the master or slave CPU.

Figure 28.7.6b Connection example of USART master-slave communication



■ Function Selection

Select the operation mode and data transfer mode for master-slave communication as shown in table 28.7.6c.

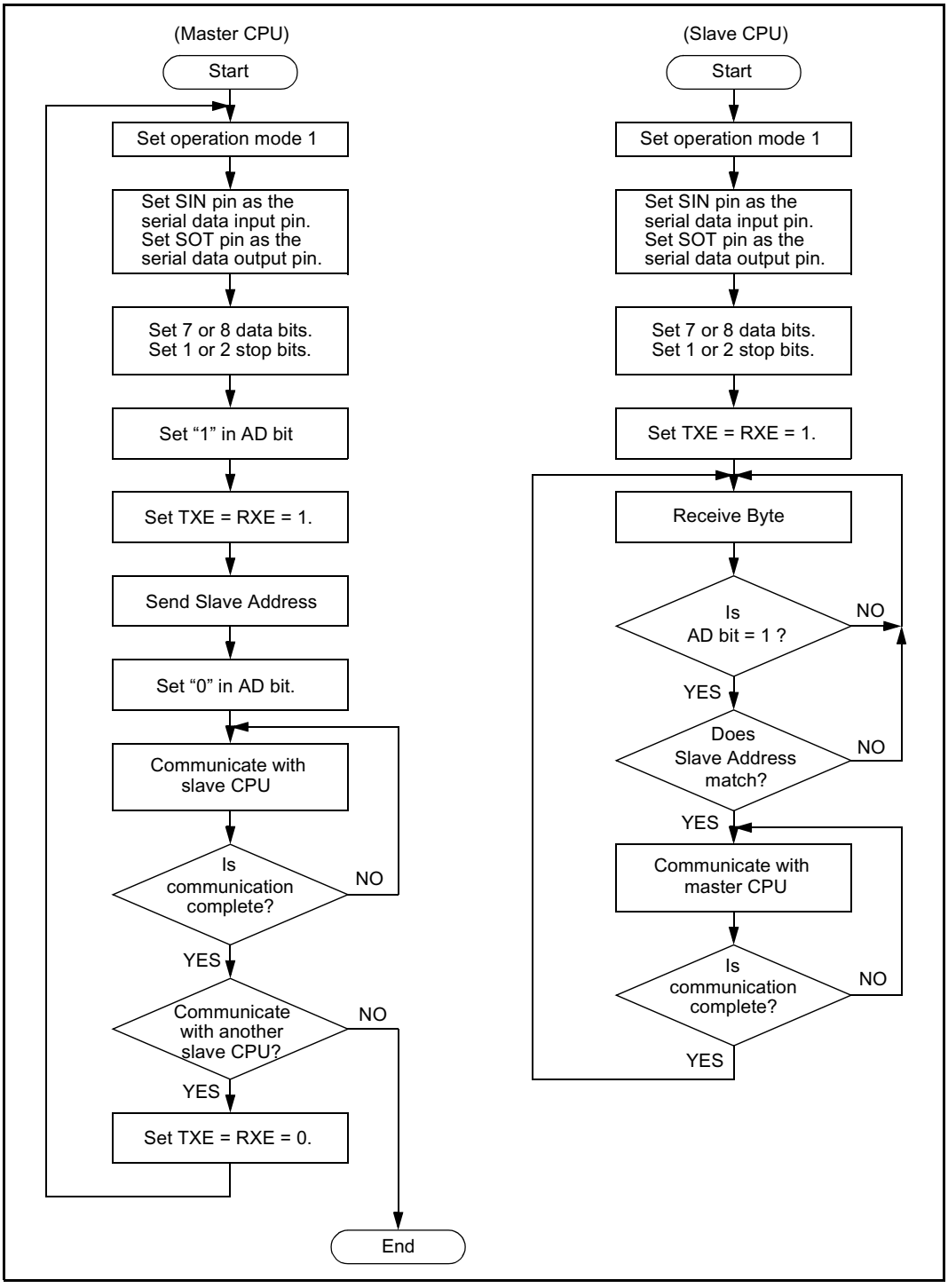
Table 28.7.6c Selection of the master-slave communication function

	Operation mode		Data	Parity	Synchronization method	Stop bit	Bit direction
	Master CPU	Slave CPU					
Address transmission and reception	Mode 1 (send AD-bit)	Mode 1 (receive AD-bit)	AD="1" + 7- or 8-bit address	None	Asynchronous	1 or 2 bits	LSB or MSB first
Data transmission and reception			AD="0" + 7- or 8-bit data				

■ Communication Procedure

When the master CPU transmits address data, communication starts. The A/D bit in the address data is set to 1, and the communication destination slave CPU is selected. Each slave CPU checks the address data using a program. When the address data indicates the address assigned to a slave CPU, the slave CPU communicates with the master CPU (ordinary data). Figure 28.7.6d shows a flowchart of master-slave communication (multiprocessor mode)

Figure 28.7.6d Master-slave communication flowchart



28.7.7 LIN Communication Function

USART communication with LIN devices is available for both LIN master or LIN slave systems.

■ LIN-Master-Slave Communication Function

The settings shown in the figure below are required to operate USART in LIN communication mode (operation mode 3).

Figure 28.7.7a Settings for USART

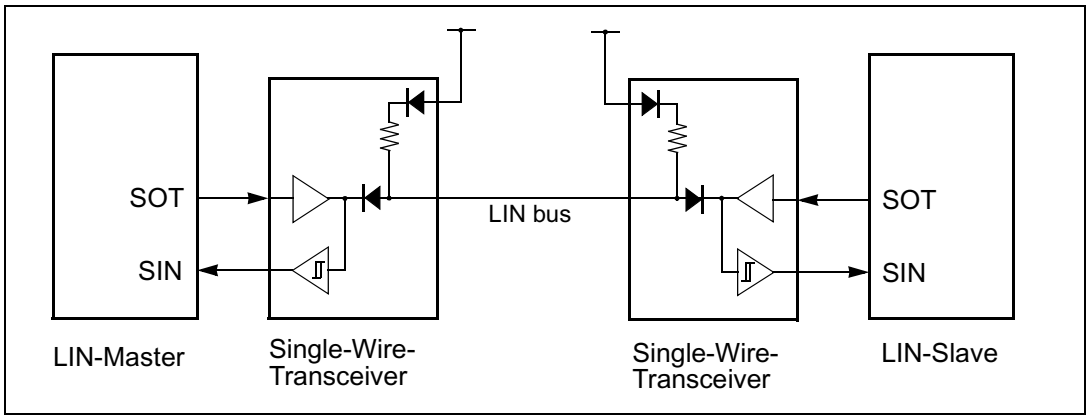
bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR5,SMR5	PEN	P	SBL	CL	AD	CRE	RXE	TXE	MD1	MD0	OTO	EXT	REST	UPCL	SCKE	SOE
Mode 3	x	x	+	+	x	0	◎	◎	1	1	x	0	0	0	1	◎
SSR5, TDR5/RDR5	PE	ORE	FRE	RDRF	TDRE	BDS	RIE	TIE	Set transmission data (during writing) Retain reception data (during reading)							
Mode 3	x	◎	◎	◎	◎	+	◎	◎								
ESCR5,ECCR5	LBIE	LBD	LBL1	LBL0	SOPE	SIOP	CCO	SECS	-	LBR	MS	SCDE	SSM	BIE	RBI	TBI
Mode 3	◎	◎	◎	◎	x	x	x	x		◎	x	x	x	x	x	x

◎ Bit is used
x Bit is not used
0 / 1 Set bit to 0 / 1
\+ Bit is automatically set to the correct value

■ LIN device connection

As shown in the Figure below, a communication system of one LIN-Master device and a LIN-Slave device. USART can operate both as LIN-Master or LIN-Slave.

Figure 28.7.7b Connection example of a small LIN-Bus system

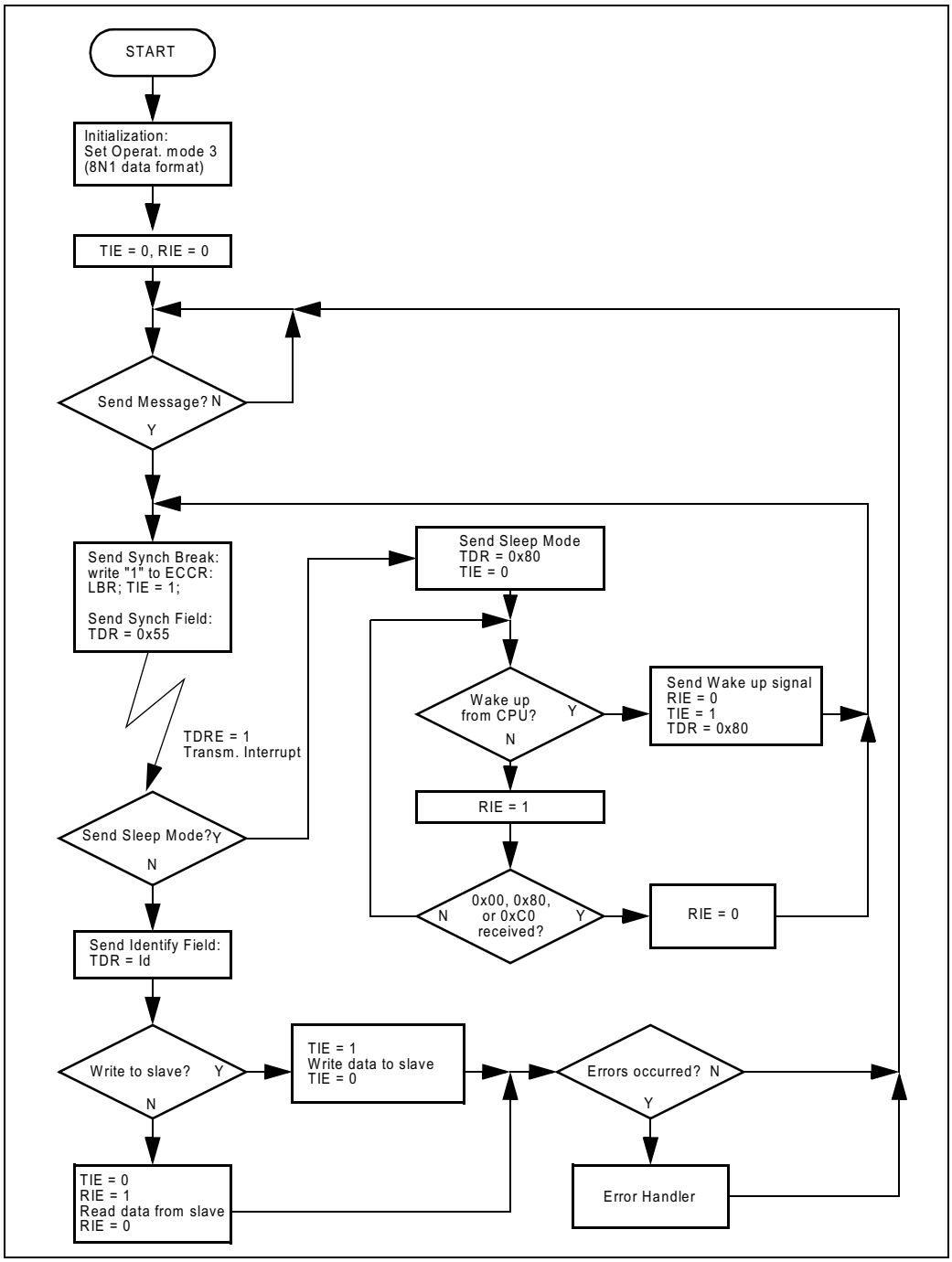


28.7.8 Sample Flowcharts for USART in LIN Communication (Operation Mode 3)

This section contains sample flowcharts for USART in LIN communication.

■ **USART as master device**

Figure 28.7.8a USART LIN master flow chart



■ USART as slave device

Figure 28.7.8b USART LIN slave flow chart (part1)

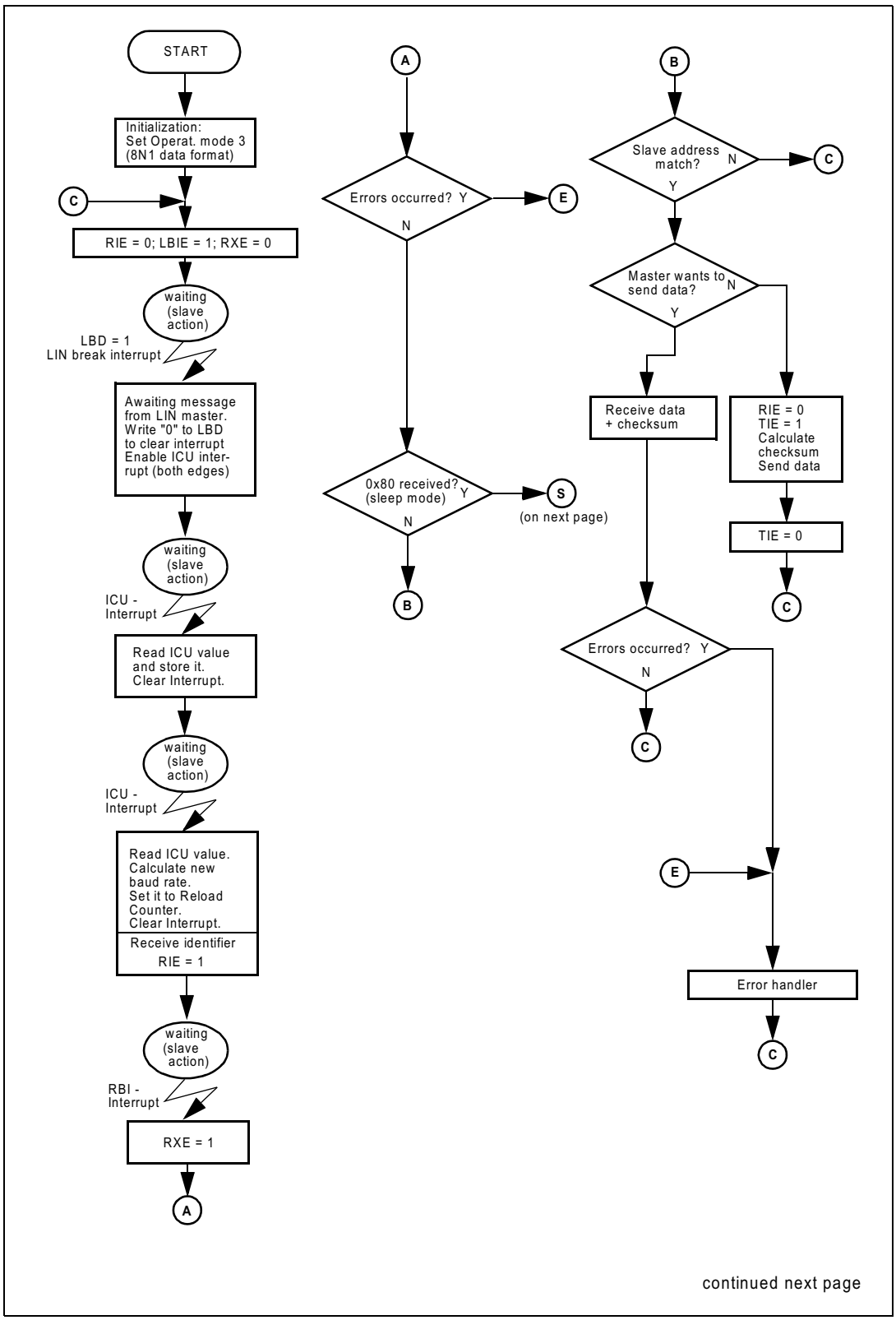
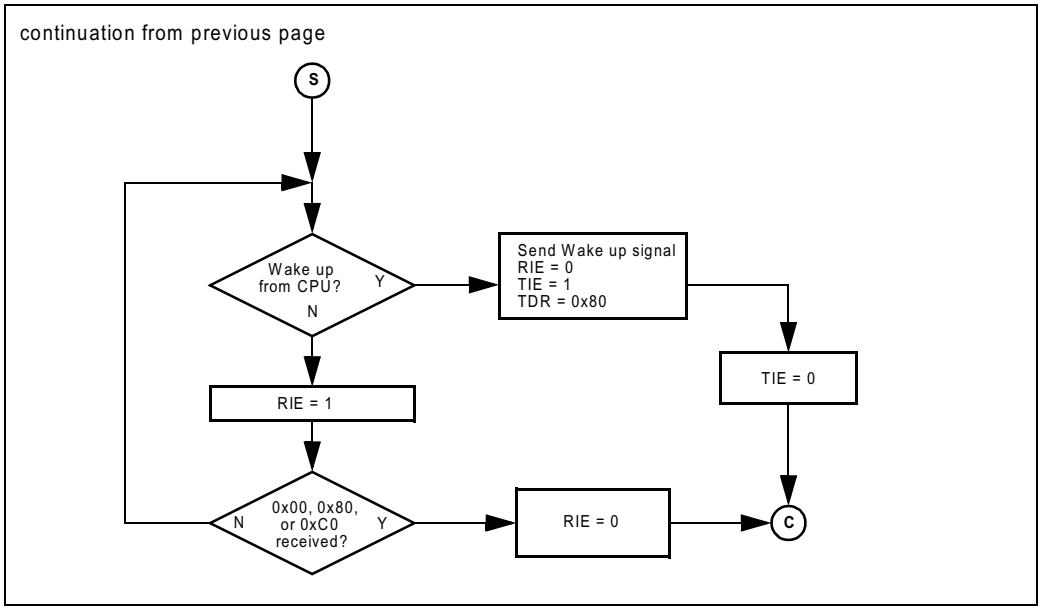


Figure 28.7.8c USART LIN slave flow chart (part 2)



28.8 NOTES ON USING USART

Notes on using USART are given below.

■ Enabling Operations

In USART, the control register (SCR5) has TXE (transmission) and RXE (reception) operation enable bits. Both, transmission and reception operations, must be enabled before the transfer starts because they have been disabled as the default value (initial value). The transfer can also be canceled by disabling its operation as required.

In single wire bus systems like ISO 9141 (LIN bus system) because of the mono directional communication it is highly recommended to enable only one of these two bits at the same time. Because of the automatic reception the sent data by USART would be received by USART too.

■ Communication Mode Setting

Set the communication mode while the system is not operating. If the mode is changed during transmission or reception, the transmission or reception is stopped and possible data will be get lost.

■ Transmission Interrupt Enabling Timing

The default (initial value) of the transmission data empty flag bit (SSR5: TDRE) is “1” (no transmission data and transmission data write enable state). A transmission interrupt request is generated as soon as the transmission interrupt request is enabled (SSR5: TIE=1). Be sure to set the TIE flag to “1” after setting the transmission data to avoid an immediate interrupt.

■ Using LIN operation mode 3

The LIN features are also available in mode 0 (transmitting, receiving break), but using mode 3 sets the USART data format automatically to LIN format (8N1, LSB first). So, break features are appliable for bus protocols other than LIN in mode 0. Note, that the transmission time of the break is variable, but the detection is specified to a minimum of 11 serial bit times.

■ Changing Operation Settings

It is strongly recommended to reset USART after changing operation settings. Particularly in synchronous mode 2 if (for example) start-/stop-bits added to or removed from the data format.

<Caution>

If settings in the Serial Mode Register (SMR5) are desired, it is not useful to set the UPCL bit at the same time to reset USART. The correct operation settings are *not* guaranteed in this case. Thus it is recommended to set the bits of the SMR5 and *then* to set them again plus the UPCL bit.

■ LIN slave settings

To initiate USART for LIN slave make sure to set the baudrate before receiving the first LIN synchronization break. This is needed to detect safely the minimum of 11 bit times of a LIN synch break.

■ Software compatibility

Although USART is similar to older Fujitsu-UARTs it is **not** software compatible to them. The programming models may be the same, but the structure of the registers differ. Furthermore the

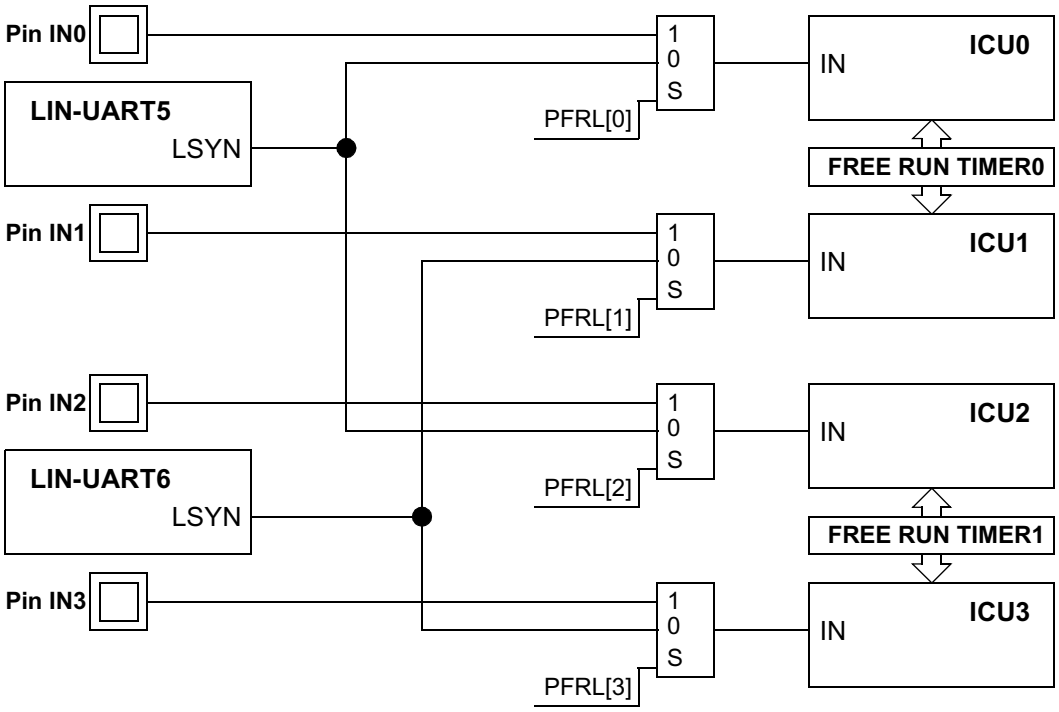
setting of the baud rate is now determined by a reload value instead of selecting a preset value.

■ Bus Idle Function

The Bus Idle Function cannot be used in synchronous mode 2.

■ Baud Rate Detection Using the Input Capture Units

The USARTs provide the signal LSYN that can be connected to the ICU so that LSYN's pulse length can be measured to derive the baud rate. The connection of the LSYN signals to the ICUs is controlled by the Port L function register PFRL (address 0415H), bits PFRL[3:0]:



If the PFR bit is set, the ICU is connected to its corresponding input pin IN.
If the PFR bit is cleared, the pin IN is in port mode (PortL[3:0]), and the USARTs are connected to the ICU. The user has to take into account that:

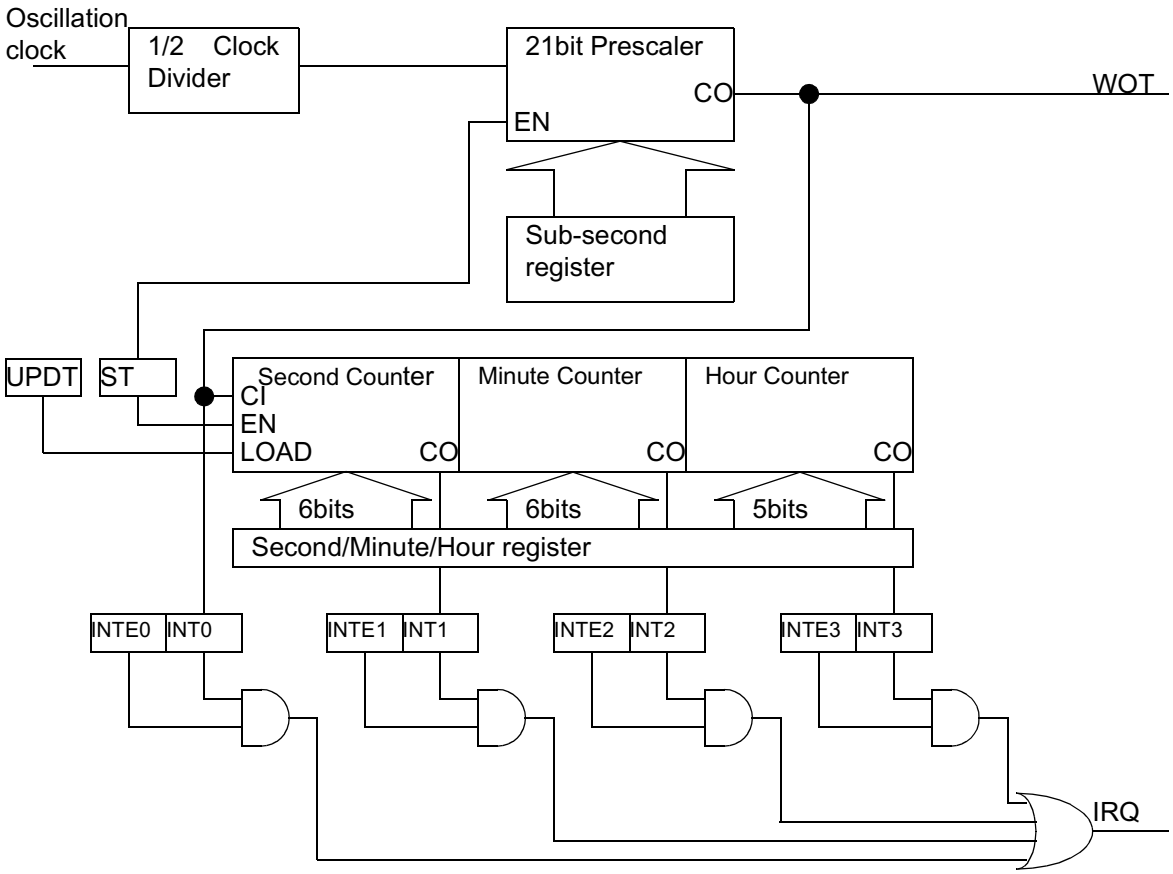
- ICU0 and ICU1 share one free running timer (prescaler), ICU2 and ICU3 share the other one.
- The free running timers can be cleared by enabling this function in OCU0/OCU2 !

CHAPTER 29 REAL TIME CLOCK

This Chaper provides an overview of the Real Time Clock (also called Watchtimer), describes the register structure and functions , and describes the operation of RTC module.

The Real Time Clock (Watch Timer) consists of the Timer Control register, Sub-second register, Second/Minute/Hour registers, 1/2 clock divider, 21bit prescaler and Second/ Minute/Hour counters. The Real Time Clock operates as the real-world timer and provides the real-world time information.

29.1 BLOCK DIAGRAM



29.2 REGISTERS

Timer control register
Address: 0000F7H

Bit number	7	6	5	4	3	2	1	0
WTCR	TST2	TST1	TST0	—	RUN	UPDT	—	ST
Read/write	(R/W)	(R/W)	(R/W)	—	(R)	(R/W)	—	(R/W)
Initial value	(0)	(0)	(0)	—	(0)	(0)	—	(0)

Timer control register
Address: 0000F6H

Bit number	15	14	13	12	11	10	9	8
WTCR	INTE3	INT3	INTE2	INT2	INTE1	INT1	INTE0	INT0
Read/write	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

Sub-second register
Address: 0000FBH

Bit number	7	6	5	4	3	2	1	0
WTBR	D7	D6	D5	D4	D3	D2	D1	D0
Read/write	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)

Sub-second register
Address: 0000FAH

Bit number	15	14	13	12	11	10	9	8
WTBR	D15	D14	D13	D12	D11	D10	D9	D8
Read/write	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value	(X)	(X)	(X)	(X)	(X)	(X)	(X)	(X)

Sub-second register
Address: 0000F9H

Bit number	7	6	5	4	3	2	1	0
WTBR	—	—	—	D20	D19	D18	D17	D16
Read/write	—	—	—	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value	—	—	—	(X)	(X)	(X)	(X)	(X)

Second register
Address: 0000FEH

Bit number	15	14	13	12	11	10	9	8
WTSR	—	—	S5	S4	S3	S2	S1	S0
Read/write	—	—	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value	—	—	(X)	(X)	(X)	(X)	(X)	(X)

Minute register
Address: 0000FDH

Bit number	7	6	5	4	3	2	1	0
WTMR	—	—	M5	M4	M3	M2	M1	M0
Read/write	—	—	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value	—	—	(X)	(X)	(X)	(X)	(X)	(X)

Hour register
Address: 0000FCH

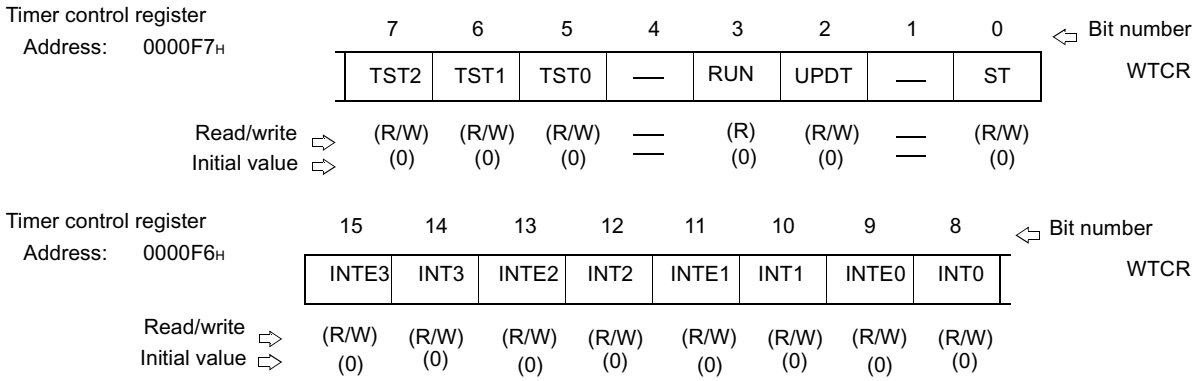
Bit number	15	14	13	12	11	10	9	8
WTHR	—	—	—	H4	H3	H2	H1	H0
Read/write	—	—	—	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)
Initial value	—	—	—	(X)	(X)	(X)	(X)	(X)

Timer disable register
Address: 0000F5H

	7	6	5	4	3	2	1	0	⇦ Bit number
	—	—	—	—	—	—	—	DBL	WTDBL
Read/write ⇨	—	—	—	—	—	—	—	(R/W)	
Initial value ⇨	—	—	—	—	—	—	—	(0)	

29.3 BREGISTER DETAILS

29.3.1 Timer Control Register (WTCR)



[bits 15 to 8] INT3 to 0, INTE3 to 0 : Interrupt flags and Interrupt enable flags

INT0 to INT3 are the interrupt flags. They are set when the second counter, minute counter and hour counter overflow respectively. If a INT bit is set while the corresponding INTE bit is "1", the Watch Timer signals an interrupt. These flags are intended to signal an interrupt every second/minute/hour/day.

Writing "0" to the INT bits clears the flags and writing "1" does not have any effect. Any read-modify-write instruction performed on the INT bit results reading "1".

[bits 7 to 5] TST2 to 0 : Test bits

These bits are prepared for the device test. In any user applications, they should be set to "000".

[bit 3] RUN : Flag

This bit can be read only and if "1" is read it indicates that the RTC module is actively operating.

[bit 2] UPDT : Update bit

The UPDT bit is prepared for modifying the Second/Minute/Hour counter values.

To modify the counter values, write the modified data in the Second/Minute/Hour registers. Then set the UPDT bit to "1". The register values are loaded to the counter at the next CO signal from the 21-bit prescaler. The UPDT bit is reset by the hardware when the counter values are updated. However, if the set operation by software and the reset operation by hardware occur at the same time, the UPDT bit will not be reset. This will only work, if the R-bus clock has a higher frequency than the RTC clock (oscillation clock).

Note: See 29.4.1 "Using the Update Bit (UPDT)" on page 650 for a workaround.

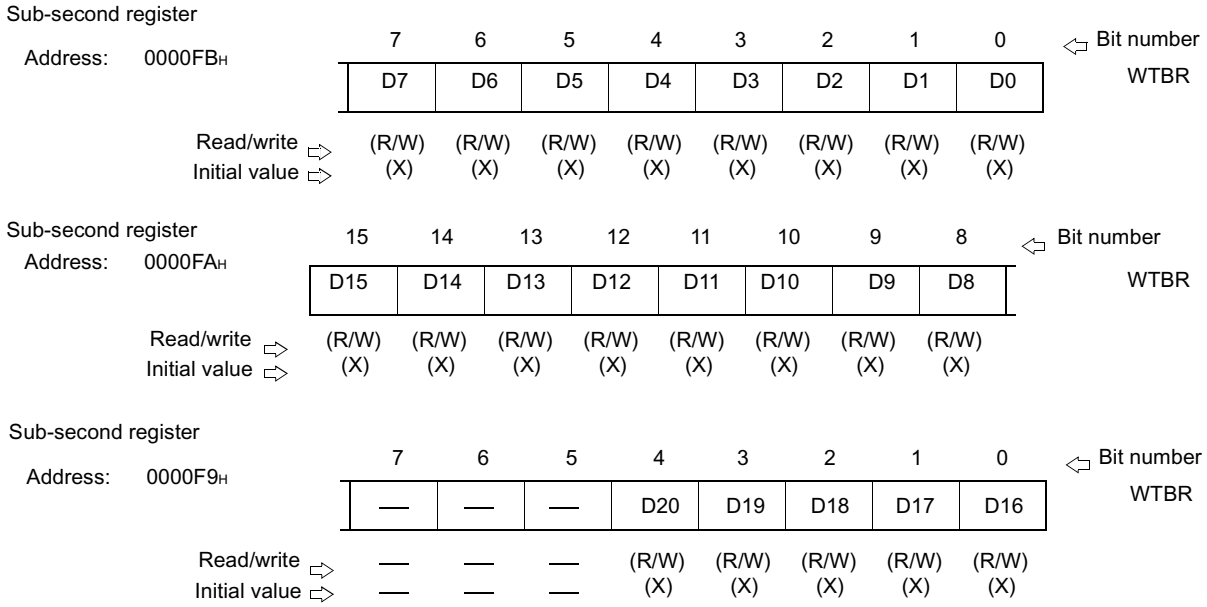
Writing "0" to the UPDT bit has no effect and a read-modify-write instruction results in reading "0".

[bit 0] ST : Start bit

When the ST bit is set to "1", the Watch Timer loads Second/Minute/Hour values from the registers and starts its operation. When it is reset to "0", all the counters and the prescalers are reset to "0" and halts.

This bit can also be used for updating the counter values. Set ST to "0", wait for RUN to go to "0", update the counter values and set ST to "1".

29.3.2 Sub-Second Registers (WTBR)



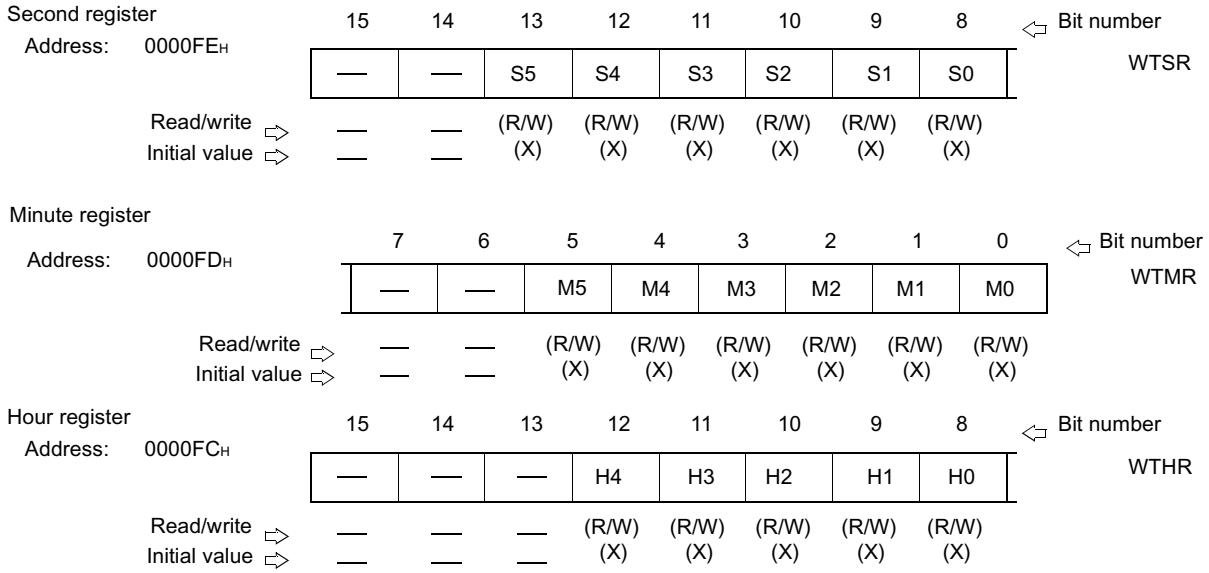
[bit 20 to 0] D20 to D0

The Sub-second register stores the reload value for the 21bit prescaler. This value is reloaded after the reload counter reaches "0". Note that when modifying the all three bytes, make sure the reload operation will not be performed in between the write instructions. Otherwise the 21-bit prescaler loads the incorrect value of the combination of new data and old data bytes. It is generally recommended that the Sub-Second register are updated while the ST bit is "0". If the sub-second registers are set to "0", the 21-bit prescaler does not operate at all.

The input clock frequency always equals the oscillation clock frequency and it is intended to be 4MHz or 32.768 KHz. For 4 MHz the reload value of the 21bit prescaler is typically set to Hex 1E847F which equals to "$2^7 * 5^6$-1". For 32.768 KHz the typical value for the prescaler would be: Hex 4000.

Therefore the combination of these two prescalers is intended to provide a clock signal of exact one second.

29.3.3 Second/Minute/Hour Registers (WTSR,WTMR,WTHR)



The Second/Minute/Hour registers store the time information. It is a binary representation of the second, minute and hour.

Reading these registers simply returns the counter values. These registers are write associable however, the written data is loaded in the counters after the UPDT bit is set to "1".

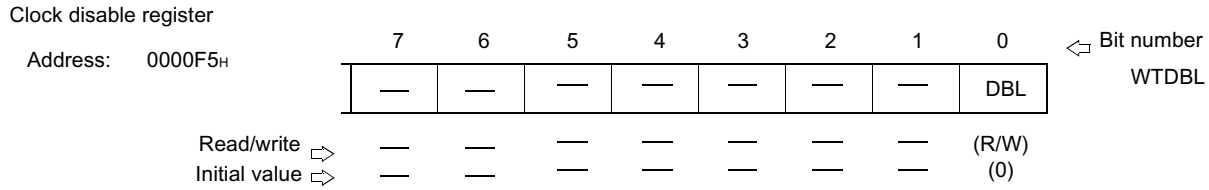
Since there are three byte-registers, make sure the obtained values from the registers are consistent.

i.e. Obtained value of "1 hour, 59 minute, 59 second" could be "0 hour 59 minute, 59 second" or "1 hour, 0 minute, 0 second" or "2 hour, 0 minute, 0 second".

If reading is done at the moment of the counter overflow it is possible to read wrong values. So reading should be either triggered by an interrupt of the RTC module (data will be stable, when the interrupt is seen by the CPU) or the following procedure should be followed:

- Clear interrupt flags of the RTC module
- Read registers
- If flags are set after reading (time overflow occurred during reading) read again

29.3.4 Clock Disable Register (WTDBL)



[bit 0] DBL: clock disable

If this bit is set to "1" the clock for the RTC module is disabled. Set this bit to "0" for normal operation. This bit is initialized to "0". This bit is readable and writable.

29.4 NOTES ON USING THE RTC

29.4.1 Using the Update Bit (UPDT)

■ Introduction

It should be noted that for the case where the RTC clock is faster than the R-bus clock (e.g. RTC clock = 4 MHz; R-Bus clock CLKP = 2 MHz) the usage of the update bit is only possible on MB91F364G and MB91F369GA. For all other MB91360 devices the updates in this special case must be done by using the ST and RUN control bits.

■ Update Bit Timing

When using the update bit on the other devices, the following procedure should be followed:

- Write new values into the Second/Minute/Hour registers (WTSR,WTMR,WTHR).
- Set update bit UPDT bit in WTCR register to "1".

  The new values are loaded into the RTC counter at the next counter overflow (second interrupt). After the counter update the UPDT bit is reset by the hardware. Only then the update operation is completely finished. This update operation requires the RTC clock and the R-Bus clock CLKP. So to ensure that this operation can be successfully finished.
- Wait for the UPDT bit to become "0" before going into RTC or STOP mode.

CHAPTER 30 SUBCLOCK

This section describes the functions and operation of the subclock

Note: Subclock operation is only possible in certain devices. Please refer to section 1.2 "MB91360 PRODUCT LINEUP" on page 4.

30.1 OVERVIEW OF THE SUBCLOCK SYSTEM

The Subclock System provides various power saving modes. The key of the concept is to supply the 32KHz clock signal only to the Real Time Clock RTC) Module, while the rest of the MCU is provided with 4MHz clock signal in order to achieve lower power supply current in the RTC32K mode.

This behaviour can be altered by the configuration input, SELCLK pin to switch the RTC module to operate with the 4 MHz clock. The following chapters describe the operation with SELCLK connectd to "0" and SELCLK connected to "1" respectively.

Note: On MB91F362GB SELCLK should always be connected to "1", subclock operation is not implemented on those devices.

30.2 OPERATION OF SUBCLOCK (SELCLK = 0)

The next table summarises the operation states of the components related to the Subclock System.To simplify this table SLEEP modes are not listed but the operation is the same as for RUN modes except that the CPU is stopped

.

Table 30.2a Operation modes

Mode	Power dissipation	Operation of components				
		4M Osc.	32K Osc.	RTC	CPU & Peripheral	PLL
RUN	High	Run	Run	Run	Run	Stop/Run
RTC4M32K	Medium Low	Run	Run	Run	Stop	Stop
RTC32K	Low	Stop	Run	Run	Stop	Stop
STOP	Lowest	Stop	Stop	Stop	Stop	Stop

The RTC4M32K mode brings the advantage of a faster reactivation time from this mode to the RUN mode. This advantage may particulary improve the reaction time to CAN bus events due to shorter oscillation stabilization time.

The following table summarises those operation modes and necessary software settings

.

Table 30.2b Operation mode software settings

Mode	Software Setting					
	STOP	PLL1EN	PLL2EN	OSCD1	OSCD2	RTC32
RUN	0	0 or 1	1	Don’t Care	Don’t Care	Don’t Care
RTC4M32K	1	Don’t Care	1	0	0	Don’t Care
RTC32K	1	Don’t Care	1	1	0	1
STOP	1	Don’t Care	Don’t Care	1	1	Don’t Care

The STOP, OSCD1 and OSDC2 bits reside in bit 7, 1 and 0 of the STCR register (address 000481H) in the Clock Generation Control module. The PLL2EN and PLL1EN bits reside in bit 11 and 10 of the CLKR register (address 000484H) also in the Clock Generation Control module.

The RTC32 bit resides in bit 8 of the CLKR2 register (address 000046H).

It is recommended that PLL2EN is set to "1" after the initialization to start the 32KHz oscillation and this bit should be kept at "1" during the operation. Otherwise the 32 kHz oscillator does not start. Also bits 9 and 10 of the CLKR register (address 0046H) should always be set to "0" during operation.

30.3 4MHz REAL TIME CLOCK CONFIGURATION (SELCLK=1)

When the SELCLK pad is connected logic level 1, the 32KHz oscillation is disabled regardless of the software setting. In this configuration, the Real Time Clock Module is supplied with the 4MHz oscillation clock signal.

The following table summarises the modes available in this configuration.

.

Table 30.3a Operation modes with SELCLK=1

Mode	Power dissipation	Operation of components				
		4M Osc.	32K Osc.	RTC	CPU & Peripheral	PLL
RUN	High	Run	Stop	Run	Run	Stop/Run
RTC4M	Medium Low	Run	Stop	Run	Stop	Stop
STOP	Lowest	Stop	Stop	Stop	Stop	Stop

Table 30.3b Software settings with SELCLK=1

Mode	Software Setting					
	STOP	PLL1EN	PLL2EN	OSCD1	OSCD2	RTC32
RUN	0	0 or 1	Don’t Care	Don’t Care	Don’t Care	Don’t Care
RTC4M	1	Don’t Care	Don’t Care	0	Don’t Care	Don’t Care
STOP	1	Don’t Care	Don’t Care	1	Don’t Care	Don’t Care

30.4 USE OF REAL TIME CLOCK MODULE

There is some additional consideration needed to operate the RTC module to achieve the desired functionality.

Because the RTC module is directly connected to the 32kHz oscillation clock, the oscillation stabilization time has to be taken care of by the software. This can be achieved by using another timer (e.g the Time Base Timer) to trigger the software to start the RTC module (Setting of ST bit to "1").

It is also important to stop the RTC module before entering the STOP mode. Otherwise, the reactivation from STOP mode results in unpredictable operation of the RTC module.

After the reactivation, the oscillation stabilization time has to be measured again by the software, then the RTC module can be restarted.

Remarks:

When switching from RTC4M to RUN mode a wait time from at least 20 µs must be performed.

CHAPTER 31 32kHz CLOCK CALIBRATION UNIT

The 32kHz Clock Calibration Module provides possibilities to calibrate the 32kHz oscillation clock with respect to the 4MHz oscillation clock. This chapter gives an overview of the calibration unit, describes the registers and provides some application notes.

31.1 OVERVIEW

The 32kHz Clock Calibration Module provides possibilities to calibrate the 32kHz oscillation clock with respect to the 4MHz oscillation clock.

31.1.1 Description

This hardware allows the software to measure time generated by the 32kHz clock with the 4MHz clock.

By utilizing this hardware in conjunction with software processing, the accuracy of the 32kHz clock can come closer to that of the 4MHz clock. The measurement result from the 32kHz Clock Calibration Module can be processed by the software and the setting required for the Real Time Clock Module can be obtained.

This module consists of two timers, one operating with the 32kHz clock and the other operating with the 4MHz clock. The 32kHz timer triggers the 4MHz timer and resulting 4MHz timer value is stored in a register. The value stored in this register can be used for the subsequent software processing to calculate the desired Real Time Clock module's setting.

31.1.2 Block Diagram

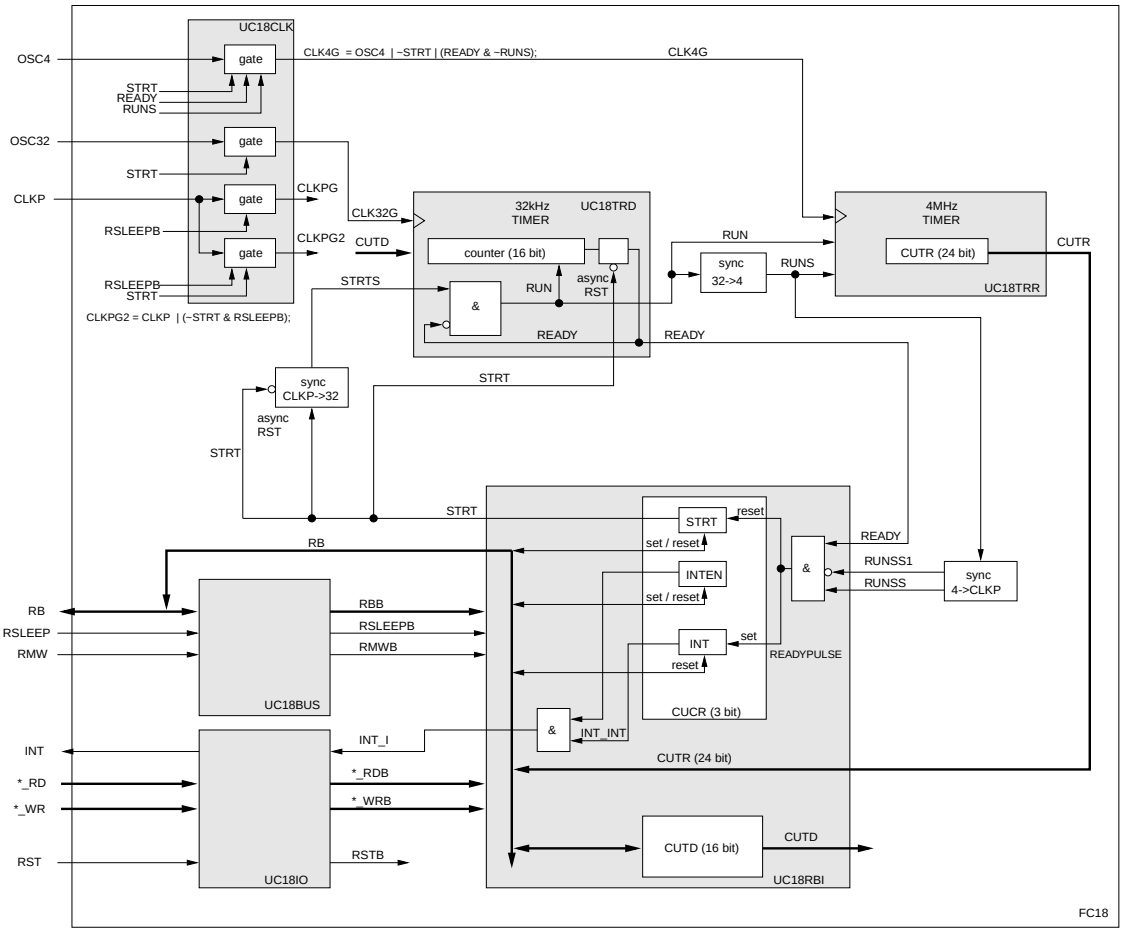


Figure 31.1.2 Block Diagram of the calibration unit

31.1.3 Timing

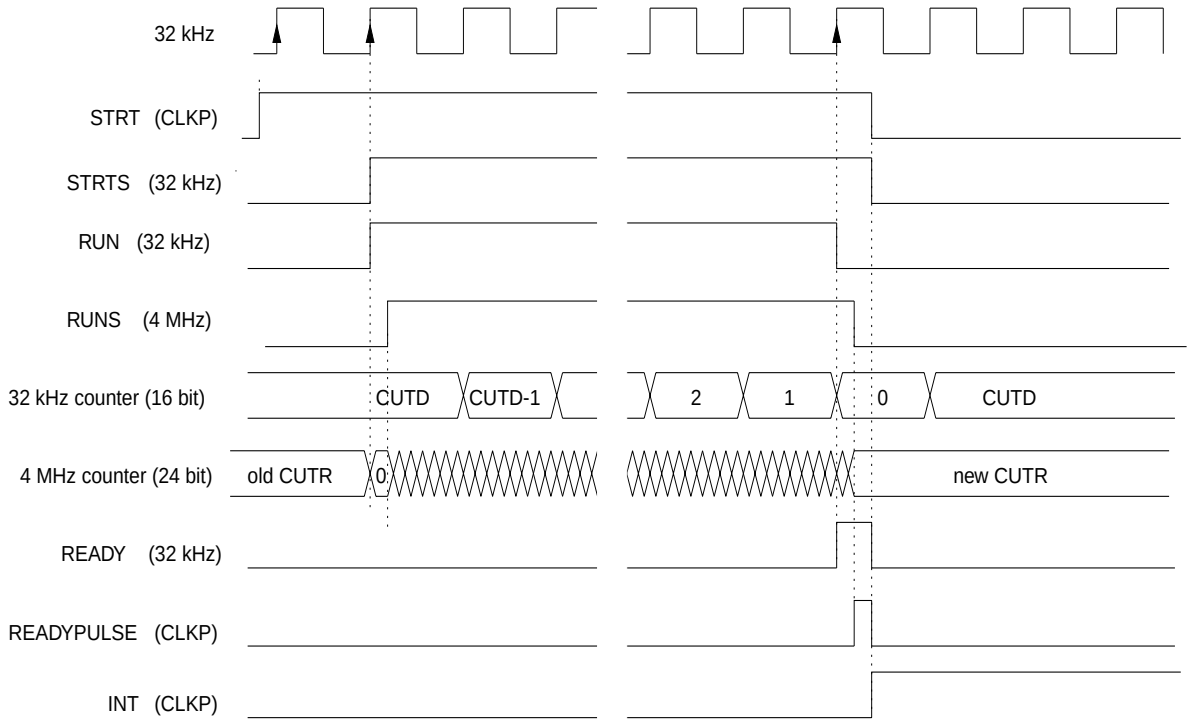


Figure 31.1.3 Timing of the measurement process

31.1.4 Clocks

The module operates with 3 different clocks: The 4 MHz clock OSC4, the 32 kHz clock OSC32 and the peripheral clock CLKP. Synchronization circuits adapt the different domains.

All 3 clocks are gated. The 32kHz and the 4MHz clock are switched off if STRT is 0. CLKPG is gated by RSLEEP and CLKPG2 by RSLEEP and STRT for the 2 bits, which are set/reset by hardware.

The clock frequencies have to fulfill the following requirements:

- Clock ratio

  $T_{OSC32} > 2 \times T_{OSC4} + 3 \times T_{CLKP}$

  $T_{OSC4} < 1/2 \times T_{OSC32} - 3/2 \times T_{CLKP}$

  $T_{CLKP} < 1/3 \times T_{OSC32} - 2/3 \times T_{OSC4}$

- The input frequencies must not exceed the values given in Table 31.1.5a.

Table 31.1.5a Maximum operation frequencies

	CLKP		OSC32		OSC4	
maximum	32 MHz	31.25 ns	4 MHz	250 ns	13 MHz	76.9 ns

Table 31.1.5b Example of valid clock ratios which fulfill requirements 1 and 2

	OSC32		OSC4		CLKP	
maximum operation speed	4 MHz	250 ns	13 MHz	76.9 ns	32 MHz	31.25 ns
standard TDIR mode	500 kHz	2000 ns	4 MHz	250 ns	4 MHz	250 ns
normal operation	32 kHz	31.25 us	4 MHz	250 ns	>2 MHz	500 ns

31.2 REGISTER DESCRIPTION

This section lists the registers of the calibration unit and describes the function of each register in detail.

■ **Calibration Unit Control Register (CUCR)**

Control Register low byte	7	6	5	4	3	2	1	0	⇐ Bit no.
Address : 000191H	-	-	-	STRT	-	-	INT	INTEN	CUCRL
Read/write ⇒	(R)	(R)	(R)	(R/W)	(R)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

■ 32kHz Timer Data Register (CUTD)

32kHz Timer Data Register high byte

	15	14	13	12	11	10	9	8	⇐Bit no.
Address : 000192H	TDD15	TDD14	TDD13	TDD12	TDD11	TDD10	TDD9	TDD8	CUTDH
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(1)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

32kHz Timer Data Register low byte

	7	6	5	4	3	2	1	0	⇐Bit no.
Address : 000193H	TDD7	TDD6	TDD5	TDD4	TDD3	TDD2	TDD1	TDD0	CUTDL
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

■ 4MHz Timer Data Register (CUTR)

4MHz Timer Data Register1 high byte

	15	14	13	12	11	10	9	8	⇐Bit no.
Address : 000194H	-	-	-	-	-	-	-	-	CUTR1H
Read/write ⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

4MHz Timer Data Register1 low byte

	7	6	5	4	3	2	1	0	⇐Bit no.
Address : 000195H	TDR23	TDR22	TDR21	TDR20	TDR19	TDR18	TDR17	TDR16	CUTR1L
Read/write ⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

4MHz Timer Data Register2 high byte

	15	14	13	12	11	10	9	8	⇐Bit no.
Address : 000196H	TDR15	TDR14	TDR13	TDR12	TDR11	TDR10	TDR9	TDR8	CUTR2H
Read/write ⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

4MHz Timer Data Register2 low byte

	7	6	5	4	3	2	1	0	⇐Bit no.
Address : 000197H	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0	CUTR2L
Read/write ⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

Address	Register				Block
	+0	+1	+2	+3	
000190H	CUCR [R/W] -------- ---0--00		CUTD [R/W] 10000000 00000000		Calibration Unit of 32kHz oscillator
000194H	CUTR1 [R] -------- 00000000		CUTR2 [R] 00000000 00000000		

31.2.1 Calibration Unit Control Register (CUCR)

The Control Register (CUCR) has the following functions:

- **start / stop calibration measurement**
- **enable / disable interrupt**
- **indicates the end of the calibration measurement**

Control Register low byte	7	6	5	4	3	2	1	0	⇐ Bit no.
Address : 000191H	-	-	-	STRT	-	-	INT	INTEN	CUCRL
Read/write ⇒	(R)	(R)	(R)	(R/W)	(R)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

BIT[0]: INTEN - Interrupt enable

0	interrupt disabled (default)
1	interrupt enabled

This is the interrupt enable bit corresponding to the INT bit. When this bit is set to 1 and the INT bit is set by the hardware, the calibration module signals an interrupt to the CPU. The INT-bit itself is not affected by the INTEN bit and is set by hardware even if interrupts are disabled (INTEN=0).

BIT[1]: INT - Interrupt

0	calibration ongoing / module inactive (default)
1	calibration completed

This bit indicates the end of the calibration. When the 32KHz timer reaches zero after the start of calibration, the 4MHz Timer Data Register stores the last 4MHz timer value and the INT bit is set to 1.

The read-modify-write operation to this bit results in reading 1 and writing 0 to this bit clears this flag(INT=0). Writing 1 to this bit has no effect.

The interrupt flag INT is not reset by hardware. Therefor it must be reset by software before starting a new calibration. Otherwise the end of the calibration process is only signalized by the STRT bit (the INT flag stays 1 also during calibration).

BIT[4]: STRT - Calibration Start

0	calibration stopped, module switched off (default)
1	start calibration

When the STRT bit is set to 1 by the software, the calibration starts. The 32kHz Timer starts counting down from the value stored in the 32KHz Timer Data Register and the 4MHz Timer starts counting up from zero.

When the 32KHz Timer reaches zero, this bit is reset to 0 by the hardware.

If 0 is written into this bit by the software during the calibration process, the calibration is immediately stopped. If writing 0 by the software and reset to 0 by the hardware happens at the same time, the hardware operation supersede the software operation. This means the calibration is properly completed and the INT bit is set to “1”. Writing 1 to this bit during the calibration has no effect.

31.2.2 32KHz Timer Data Register (16 bit) (CUTD)

The 32kHz Timer Data Register (CUTD) holds the value which determines the duration of calibration (32kHz reload value)

32kHz Timer Data Register high byte	15	14	13	12	11	10	9	8	⇐Bit no.
Address : 000192H	TDD15	TDD14	TDD13	TDD12	TDD11	TDD10	TDD9	TDD8	CUTDH
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(1)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

32kHz Timer Data Register low byte	7	6	5	4	3	2	1	0	⇐Bit no.
Address : 000193H	TDD7	TDD6	TDD5	TDD4	TDD3	TDD2	TDD1	TDD0	CUTDL
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

Default value is 0x8000 which corresponds to a measurement duration of 1 second, if a 32.768 kHz crystal is used.

This register should be written only when the calibration is inactive (STRT=0).

The 32kHz Timer Register stores the value to specify the duration of the calibration. When the calibration is started, the stored value is loaded into the 32kHz Timer and the timer starts counting down until it reaches zero.

If CUTD is initialized with 0000 an underflow will occur and the measurement will take (FFFF hex + 1) * T_{osc32}

The 32kHz timer operates with the 32kHz oscillation clock.

In order to achieve a measurement duration of 1 second, the CUTD register has to be load with 0x8000 = 32768 dec. This number results from the exact oscillation frequency of the crystal, which is F_{osc}=32768 Hz. The ideal values of the measurement results (if a 4.00 MHz crystal is used) are shown in the following table.

Table 31.2.2 Ideal measurement results depending on measurement duration

duration of calibration	CUTD value	CUTR value
2 sec	0x0000	0x7A1200
1.75 sec	0xE000	0x6ACFC0
1.5 sec	0xC000	0x5B8D80
1.25 sec	0xA000	0x4C4B40
1 sec	0x8000	0x3D0900
0.75 sec	0x6000	0x2DC6C0
0.5 sec	0x4000	0x1E8480
0.25 sec	0x2000	0x0F4240

The duration of the whole process from writing a 1 into the STRT bit until STRT is reset by hardware is longer than the actual calibration measurement time, due to synchronization between the different clock domains. Process Duration < (CUTD + 3)*T_{osc32} .

The calibration measurement time is exact CUTD * T_{osc32}.

31.2.3 4MHz Timer Data Register (24 bits) (CUTR)

The Timer Data Register (CUTR) holds the value of the calibration result (4MHz counter)

Precaution: Reading this register during calibration, results in random values.
The end of calibration is indicated by the INT-bit and the STRT-bit in the CUCR-register.
After INT has changed from 0 to 1 / STRT has changed from 1 to 0, the value of CUTR is valid.

4MHz Timer Data Register1 high byte

	15	14	13	12	11	10	9	8	⇐Bit no.
Address : 000194H	-	-	-	-	-	-	-	-	CUTR1H
Read/write ⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

4MHz Timer Data Register1 low byte

	7	6	5	4	3	2	1	0	⇐Bit no.
Address : 000195H	TDR23	TDR22	TDR21	TDR20	TDR19	TDR18	TDR17	TDR16	CUTR1L
Read/write ⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

4MHz Timer Data Register2 high byte

	15	14	13	12	11	10	9	8	⇐Bit no.
Address : 000196H	TDR15	TDR14	TDR13	TDR12	TDR11	TDR10	TDR9	TDR8	CUTR2H
Read/write ⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

4MHz Timer Data Register2 low byte

	7	6	5	4	3	2	1	0	⇐Bit no.
Address : 000197H	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0	CUTR2L
Read/write ⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

The 4MHz Timer Data Register stores the result of the calibration. When the calibration is started, the 4MHz Timer starts counting up from zero. When the 32kHz Timer reaches zero, the 4MHz Timer stops counting and the register holds the calibration result until the next calibration is triggered by software.

Reading this register during calibration, results in random values.

The end of calibration is indicated by the INT-bit and the STRT-bit in CUCR-register. After these bits changed from 0 to 1, resp. 1 to 0, the value of CUTR is valid.

Writing into this register by software has no effect.

The 4MHz Timer operates with the 4MHz oscillation clock.

31.3 APPLICATION NOTE

This section lists application notes concerning accuracy of the calibration, power dissipation and measurement duration.

The setting of the 32KHz Timer Data Register can be calculated in the following way.

If the duration of 1 second is desired for the calibration, 8000Hex = 32768Dec should be set in the 32kHz Timer Data Register and it represents 32,768 pulses of the 32.768kHz oscillation clock.

This setting should result in the stored value of approximately 0x3D0900Hex in the 4MHz Time Data Register. This value represents 4,000,000 pulses of the 4MHz oscillator.

Table 31.3 Ideal measurement results (CUTR) with 32.768kHz and 4.0MHz oscillators

duration of calibration	CUTD value	CUTR value
2 sec	0x0000	0x7A1200
1.75 sec	0xE000	0x6ACFC0
1.5 sec	0xC000	0x5B8D80
1.25 sec	0xA000	0x4C4B40
1 sec	0x8000	0x3D0900
0.75 sec	0x6000	0x2DC6C0
0.5 sec	0x4000	0x1E8480
0.25 sec	0x2000	0x0F4240

The key to the use of the calibration module is power dissipation as well as accuracy of the calibration.

■ **Accuracy:**

The accuracy of the calibration is dependent on the clock frequency used by the 4MHz Timer and duration of the calibration. The maximum error of the 4MHz timer is +/- 1 digit. If the clock frequency is 4MHz and duration of the calibration is 1 second, the achieved accuracy is calculated in the following way:

0.25us (Clock cycle time) / 1 second (duration)=0.25 ppm.

In general:

Accuracy = (Clock cycle time of 4MHz Timer) / (Duration of calibration)

■ **Power dissipation:**

Suppose the current consumption I_{RUN} in run mode is 20 times the consumption I_{RTC} in RTC mode ($I_{RUN} = 20 \times I_{RTC}$).

If the MCU is woken up from RTC mode by software to trigger the calibration measurement every minute and the duration of the calibration is set to 1 second, the increase in the power dissipation can be $20 \times I_{RTC}/60 = 1/3 * I_{RTC}$.

Therefore the software has to make sure that the increase should not affect the hardware limitations coming from the system requirements. For example, the software has to be designed to trigger the calibration least frequently.

It is generally recommended that the theoretical increase in power dissipation is not more than 5% particularly in the RTC mode of the MCU.

■ **Measurement limits:**

The limit to the duration of the calibration is roughly 2 seconds if the 32kHz Timer is operating with 32kHz clock. On the other hand, the 4MHz Timer can measure time up to 4 seconds if it is working with a 4MHz clock.

CHAPTER 32 FLASH MEMORY

MB91360 devices feature 256 KB, 512 KB or 768 KB of embedded flash memory. It is connected to the F-bus.

Note: When referring to pin names in flash memory mode, the following chapter uses pin names as on MB91F362GB. Please check section 32.3 for the corresponding pin names of other devices.

Note: This chapter explaines the 512 KByte flash macro only. For the differences in addressing of the 256 KB and 768 KB macros, see the data sheets of MB91F364G and MB91F376G, respectively.

32.1 OUTLINE OF FLASH MEMORY

The Flash Memory consists of a flash memory unit derived from the MBM29LV400C and a flash memory interface circuit.

Flash Memory:

- 768 KWord x 8 bit / 384 KWord x 16 bit / 192 KWord x 32 bit
  ((64 KByte*5 + 32 KByte + 8 KByte*2 + 16 KByte) x 2) sectors (MB91F376G only)
- 512 KWord x 8 bit / 256 KWord x 16 bit / 128 KWord x 32 bit
  ((64 KByte*3 + 32 KByte + 8 KByte*2 + 16 KByte) x 2) sectors
- 256 KWord x 8 bit / 128 KWord x 16 bit / 64 KWord x 32 bit
  ((64 KByte + 32 KByte + 8 KByte*2 + 16 KByte) x 2) sectors (MB91F364G only)
- Uses automatic program algorithm (Embedded Algorithm™)
- Erase pause/restart function
- Detects completion of writing/erasing using data polling or toggle bit functions
- Detects completion of writing/erasing by RY/BY pin
- Compatible with JEDEC standard commands
- Performs minimum of 10,000 write/erase operations
- Sector erase function (any combination of sectors)
- Sector protect function
- Temporary sector protect cancellation function
- The flash memory interface circuit allows to write/erase the flash memory both under control of external pins by flash writer ("flash memory mode") and under control of internal bus by CPU ("CPU mode").

Embedded Algorithm™ is a registered trademark of Advanced Micro Devices, Inc.

Note: This chapter explaines the 512 KByte flash macro only. For the differences in addressing of the 256 KB and 768 KB macros, see the data sheets of MB91F364G and MB91F376G, respectively.

32.2 BLOCK DIAGRAMS OF FLASH MEMORY

Figure 32.2.1 shows the block diagram of the flash memory unit. Figure 32.2.2 shows the entire block diagram of the Flash Memory with the flash memory interface circuit. Table 32.2.3a shows the sector configuration of the Flash Memory.

32.2.1 Block Diagram of Flash Memory

Figure 32.2.1 shows the block diagram of the fash memory unit, which has almost the same configuration as the MBM29LV400C.

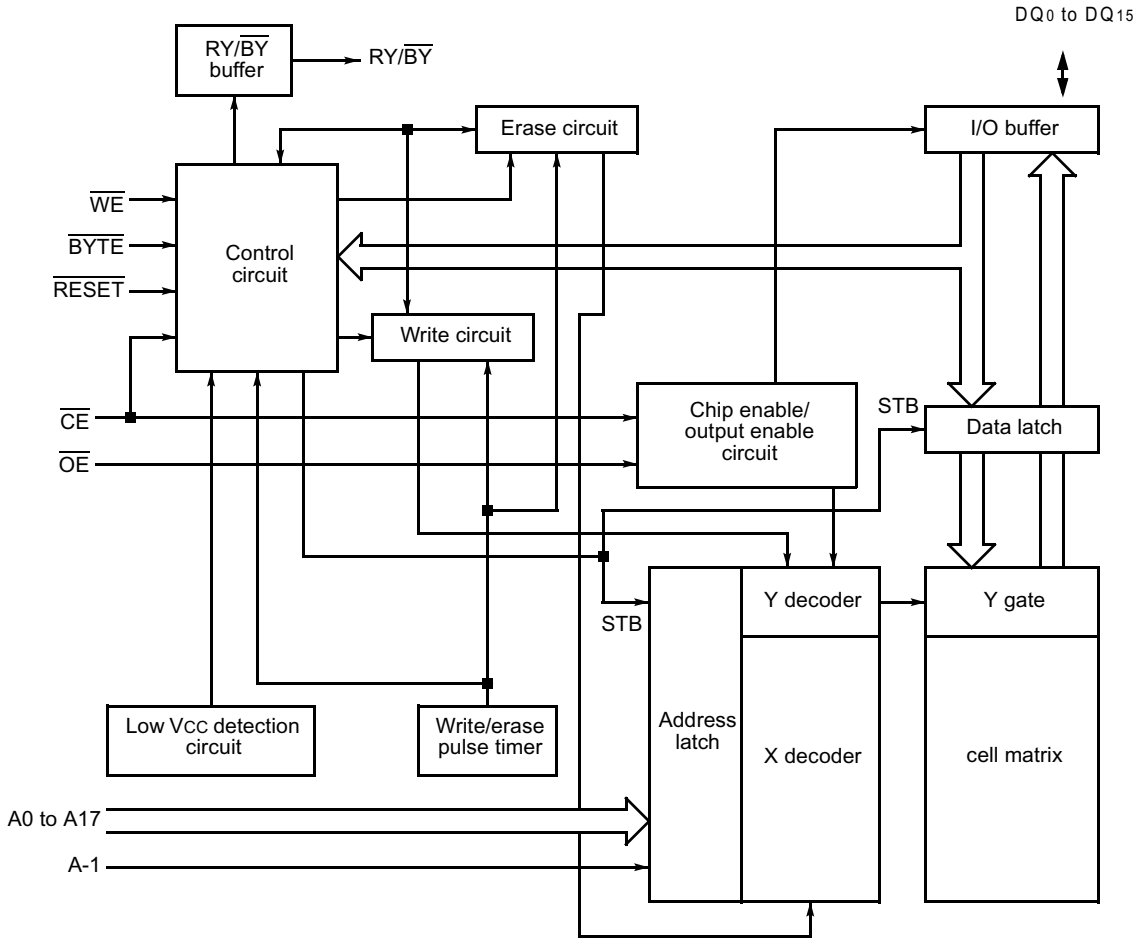
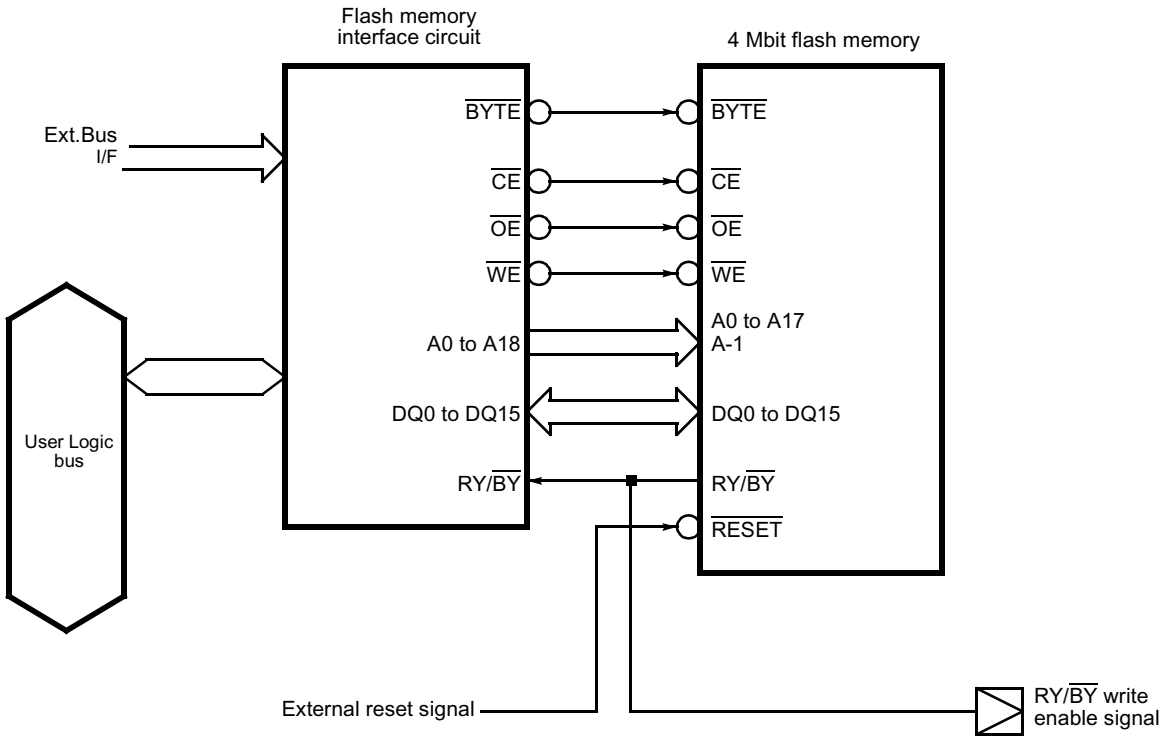


Figure 32.2.1 Block Diagram of Flash Memory

32.2.2 Entire Block Diagram of Flash Memory

Figure 32.2.2 shows the entire block diagram of the Flash Memory with the flash memory interface circuit.

Figure 32.2.2 Entire Block Diagram of Flash Memory



32.2.3 Sector Configuration

Please see also appendix A "I/O MAP" on page 749 for the addressing in CPU mode.

Table 32.2.3a Half word read/write, byte read

8 bit x 2 15 0		Flash Memory mode	CPU mode	8 bit x 2 15 0		Flash Memory mode	CPU mode
Sector 6:	16KB	3FFFFH 3C000H	FFFFCH F8000H	Sector 13:	16KB	7FFFFH 7C000H	FFFFEH F8002H
Sector 5:	8KB	3BFFFH 3A000H	F7FFCH F4000H	Sector 12:	8KB	7BFFFH 7A000H	F7FFEH F4002H
Sector4:	8KB	39FFFH 38000H	F3FFCH F0000H	Sector 11:	8KB	79FFFH 78000H	F3FFEH F0002H
Sector 3:	32KB	37FFFH 30000H	EFFFCH E0000H	Sector 10:	32KB	77FFFH 70000H	EFFFEH E0002H
Sector 2:	64KB	2FFFFH 20000H	EFFFCH C0000H	Sector 9:	64KB	6FFFFH 60000H	EFFFEH C0002H
Sector 1:	64KB	1FFFFH 10000H	BFFFCH A0000H	Sector 8:	64KB	5FFFFH 50000H	BFFFEH A0002H
Sector 0:	64KB	0FFFFH 00000H	9FFFCH 80000H	Sector 7:	64KB	4FFFFH 40000H	9FFFEH 80002H

Note: In 16-bit CPU mode, address N and N+1 apply to sectors 0...6, whereas address N+2 and N+3 apply to sectors 7...13.

Table 32.2.3b Long word read

MSB 8 bit x 2 31 16		LSB 8 bit x 2 15 8		Flash Memory mode	CPU mode
Sector 13	16KB	Sector 6	16KB	7FFFFH 78000H	FFFFFH F8000H
Sector 12	8KB	Sector 5	8KB	77FFFH 74000H	F7FFFH F4000H
Sector 11	8KB	Sector 4	8KB	73FFFH 70000H	F3FFFH F0000H
Sector 10	32KB	Sector 3	32KB	6FFFFH 60000H	EFFFFH E0000H
Sector 9	64KB	Sector 2	64KB	5FFFFH 40000H	DFFFFH C0000H
Sector 8	64KB	Sector 1	64KB	3FFFFH 20000H	BFFFFH A0000H
Sector 7	64KB	Sector 0	64KB	1FFFFH 00000H	9FFFFH 80000H

32.3 WRITE / ERASE MODES

The flash memory can be accessed in two different ways; the flash memory mode allowing write/ erase directly from the external pins, and the CPU mode allowing write/erase from the CPU via the internal bus. These modes are selected by the external mode pins.

32.3.1 Flash Memory mode

The CPU stops when the mode pins are set to **111** while the INITX signal is asserted. The flash memory interface circuit is directly connected to the external bus interface, allowing direct control by the external pins. This mode makes the MCU seem like a standard flash memory at the external pins, and write/erase can be performed using a flash memory programmer.

In the flash memory mode all the operations supported by the flash memory automatic algorithm can be used.

32.3.2 CPU mode

The flash memory is located in the F-Bus area of the CPU memory space and like ordinary mask ROM can be read-accessed and program-accessed from the CPU through the flash memory interface circuit.

Writing/erasing the flash memory is performed by instructions from the CPU via the flash memory interface circuit. Therefore, this mode allows rewriting even when the MCU is soldered on the target board.

The sector protect operations can not be performed in these modes.

32.3.3 Control signals of flash memory

Table Table 32.3.3a, “Pins used in flash memory mode,” on page 673 lists the flash memory control signals in the flash memory mode.

There is almost a one-to-one correspondence between the flash memory control signals and the external pins of the MBM29lV400TA. The V_{ID} (12 V) pins required by the sector protect operations are MD0, MD1 and MD2 instead of A9, $\overline{\text{RESET}}$ and $\overline{\text{OE}}$ for the MBMLV400C.

In the flash memory mode, the width of the external data bus can be 8 or 16 bit.

The following tables show the pins which are required for the programming procedure and also describe the states for the pins not used in flash memory mode. Most of the not used pins are in their reset state (high-Z outputs, enabled inputs). To prevent misbehaviour or damage these pins must be tied to VDD or VSS through resistors - see following tables for details.

Aside from the functional pins described below, all power pins should be connected to a power supply in the specified range, capacitances should be connected to the VCC3C pin as recommended.

■ , MB91F362GB

Table 32.3.3a Pins used in flash memory mode

MB91F362GB			MBM29LV400C	Notes
Pin number	Normal function	Flash Memory mode		
1 to 8	D24 to D31	D24 to D31	DQ8 to DQ15	
9	A0	A0	A-1	
10 to 24	A1 to A15	A1 to A15	A0 to A14	
27 to 30	A16 to A18	A16 to A18	A15 to A17	
32	CS4X	CS4X	WE	
33	CS5X	CS5X	BYTE	
35	RDY	RDY	OE	
36	BGRNTX	BGRNTX	CE	
37	BRQ	BRQ	RY/$\overline{BY}$	
111	MD0	VDA9	A9 (V_{ID})	
112	MD1	VDRS	$\overline{RESET}$ (V_{ID})	
113	MD2	VDOE	OE (V_{ID})	
115	INITX	INITX	RESET	
201 to 208	D16 to D23	D16 to D23	DQ0 to DQ7	
183 to 197, 200	D0 to D15	Output when reading from flash	-	Pull up
34	CS6X	TMODEX	-	Must be pulled up
30 to 31	A19 to A20	A19 to A20	-	Pull up or pull down

Table 32.3.3b Pins not used in flash memory mode

MB91F362GB			Notes
Pin number	Normal function	Pin state	
75 to 76	DA0, DA1	output	leave open
77	ALARM	input	pull down
81 to 83	TESTX, CPUTESTX, LTESTX	input	pull up or leave open (internal pull-up)
114	HSTX	input	pull up or leave open (internal pull-up)
116	MONCLK	output	leave open
117	SELCLK	input	pull up
119, 121	X0, X0A	input	pull down
120, 122	X1, X1A	output	leave open
124	CPO	output	leave open
125	VCI	input	pull down
all other signals		input	pull up

■ MB91F369GA

MB91F369GA				MBM29LV400C
Pin number	**Normal function**	**Flash Memory mode**	**Function in Flash Memory mode**	
58	MD0	HVDA9	High Volt. A9 (c)	A9 (V_{ID})
59	MD1	HVDR5	High Volt. RESET (c)	$\overline{RESET}$ (V_{ID})
60	MD2	HVDOE	High Volt. OE (c)	$\overline{OE}$ (VID)
62	INITX	RSTX	Hardware Reset	RESET
68	INT0	RY/BYX	Ready/Busy	RY/$\overline{BY}$
69	INT1	CEX	Chip Enable	CE
70	INT2	TSTX (a)	Flash Test	
71	INT3	BYTEX	switch 8/16 bit mode	BYTE
72	INT4	WEX	Write Enable	WE
73	INT5	OEX	Output Enable	OE
74	INT6	ATDIN (b)	Access Signal ATD	
75	INT7	EQIN (b)	Access Signal EQ	
121 to 136	D0 to D15	AQ19 downto AQ4	Address input	A19 downto A4
139 to 154	D16 to D31	DQ16 to DQ31	Data input/outpu	DQ0 to DQ15
157 to 160	A0 to A3	AQ0 to AQ3	Address input	A0 to A3

a. Pin 70 must be pulled high in Flash Memory Mode.
b. Pins 74 and 75 must be pulled low in Flash Memory Mode
c. Functionality as described in the Data Sheet of MBM29LV400C.

AVRH must be tight to a high level, ALARM to a low level. All other Pins can be left open during Flash Memory Mode.

■ **MB91FV360GA**

MB91FV360GA			MBM29LV400C
Pin number	**Normal function**	**Flash Memory mode**	
202	A0	A0	A-1
310	A1	A1	A0
201	A2	A2	A1
357	A3	A3	A2
257	A4	A4	A3
144	A5	A5	A4
309	A6	A6	A5
256	A7	A7	A6
200	A8	A8	A7
356	A9	A9	A8
308	A10	A10	A9
92	A11	A11	A10
44	A12	A12	A11
255	A13	A13	A12
143	A14	A14	A13
199	A15	A15	A14
307	A16	A16	A15
91	A17	A17	A16
142	A18	A18	A17
140	CS4X	CS4X	WE
196	CS5X	CS5X	BYTE
89	CS6X	TMODX	-
305	RDY	RDY	OE
139	BGRNTX	BGRNTX	CE
88	BRQ	BRQ	RY/$\overline{BY}$
293	MD0	VDA9	A9 (V_{ID})
31	MD1	VDRS	$\overline{RESET}$ (V_{ID})
239	MD2	VDOE	$\overline{OE}$ (V_{ID})
30	INITX	INITX	RESET
46	D16	D16	DQ0

MB91FV360GA			MBM29LV400C
Pin number	Normal function	Flash Memory mode	
95	D17	D17	DQ1
1	D18	D18	DQ2
148	D19	D19	DQ3
205	D20	D20	DQ4
45	D21	D21	DQ5
94	D22	D22	DQ6
260	D23	D23	DQ7
312	D24	D24	DQ8
204	D25	D25	DQ9
147	D26	D26	DQ10
93	D27	D27	DQ11
259	D28	D28	DQ12
203	D29	D29	DQ13
146	D30	D30	DQ14
258	D31	D31	DQ15

■ MB91F365GB/MB91F366GB

Table 32.3.3c Pins used in flash memory mode

MB91F365GB/F366GB			MBM29LV400C	Notes
Pin number	Normal function	Flash Memory mode		
31	BOOT	WE	WE	
32	TESTX	BYTE	BYTE	
33	CPUTESTX	TMODX		pull up
38	MONCLK	RY/$\overline{BY}$	RY/$\overline{BY}$	
39-46	INT0-INT7	D24 to D31	DQ8 to DQ15	
50	IN0	CE	CE	
51	IN1	OE	OE	
52	IN2	D20	DQ4	
53	IN3	D21	DQ5	
54	OUT0	D22	DQ6	
55	OUT1	D23	DQ7	
57	MD0	VDA9	A9 (V_{ID})	
58	MD1	VDRS	$\overline{RESET}$ (V_{ID})	
59	MD2	VDOE	OE (V_{ID})	
60	INITX	RESET	RESET	
91-93	PG3-PG5	A16-A18	A15-A17	
96	PWM1P0	A0	A-1	
97	PWM1M0	A1	A0	
98	PWM2P0	A2	A1	
99	PWM2M0	A3	A2	
101	PWM1P1	A4	A3	
102	PWM1M1	A5	A4	
103	PWM2P1	A6	A5	
104	PWM2M1	A7	A6	
106	PWM1P2	A8	A7	
107	PWM1M2	A9	A8	
108	PWM2P2	A10	A9	
109	PWM2M2	A11	A10	
111	PWM1P3	A12	A11	

Table 32.3.3c Pins used in flash memory mode

MB91F365GB/F366GB			MBM29LV400C	Notes
Pin number	Normal function	Flash Memory mode		
112	PWM1M3	A13	A12	
113	PWM2P3	A14	A13	
114	PWM2M3	A15	A14	
117 to 120	PJ0-PJ3	D16 to D19	DQ0 to DQ3	

Table 32.3.3d Pins not used in Flash Memory Mode

MB91F365GB/F366GB			
Pin number	Normal function	Pin State	Notes
35	X0	input	pull up
36	X1	output	leave open
66	SIN3	output	leave open
67	SOT3	output	leave open
68	SCK3	output	leave open
27	DA0 / X0A	output / input	leave open / pull up
28	DA1/X1A	output/output	leave open
29	ALARM	input	pull up
all other signals		input	pull up

■ **MB91F367GB/MB91F368GB**

Table 32.3.3e Pins used in flash memory mode

MB91F367GB/F368GB			MBM29LV400C	Notes
Pin number	Normal function	Flash Memory mode		
31	BOOT	WE	WE	
32	TESTX	BYTE	BYTE	
33	CPUTESTX	TMODX		pull up
38	MONCLK	RY/$\overline{BY}$	RY/$\overline{BY}$	
39-46	INT0-INT7	D24 to D31	DQ8 to DQ15	
50	IN0	CE	CE	
51	IN1	OE	OE	
52	IN2	D20	DQ4	
53	IN3	D21	DQ5	
54	OUT0	D22	DQ6	
55	OUT1	D23	DQ7	
57	MD0	VDA9	A9 (V_{ID})	
58	MD1	VDRS	$\overline{RESET}$ (V_{ID})	
59	MD2	VDOE	OE (V_{ID})	
60	INITX	RESET	RESET	
91-93	PG3-PG5	A16-A18	A15-A17	
96	PR0	A0	A-1	
97	PR1	A1	A0	
98	PR2	A2	A1	
99	PR3	A3	A2	
101	PR4	A4	A3	
102	PR5	A5	A4	
103	PR6	A6	A5	
104	PR7	A7	A6	
106	PS0	A8	A7	
107	PS1	A9	A8	
108	PS2	A10	A9	
109	PS3	A11	A10	
111	PS4	A12	A11	

Table 32.3.3e Pins used in flash memory mode

MB91F367GB/F368GB			MBM29LV400C	Notes
Pin number	Normal function	Flash Memory mode		
112	PS5	A13	A12	
113	PS6	A14	A13	
114	PS7	A15	A14	
117 to 120	PJ0-PJ3	D16 to D19	DQ0 to DQ3	

Table 32.3.3f Pins not used in Flash Memory Mode

MB91F367GB/F368GB			
Pin number	Normal function	Pin State	Notes
35	X0	input	pull up
36	X1	output	leave open
66	SIN3	output	leave open
67	SOT3	output	leave open
68	SCK3	output	leave open
27	X0A	input	pull up
28	X1A	output	leave open
29	ALARM	input	pull up
all other signals		input	pull up

■ MB91F364G

Table 32.3.3g Pins used in Flash Memory Mode

MB91F364G				MBM29LV400C
Pin number	Normal function	Flash Memory mode	Function in Flash Memory mode	
1	AN0	ATDIN [a]	Access Signal ATD	
2	AN1	BYTEX	switch 8/16 bit mode	$\overline{BYTE}$
34 to 41	INT0 to INT7	DQ8 to DQ15	Data input/output	DQ8 to DQ15
44 to 47	IN0 to IN3	DQ0 to DQ3	Data input/output	DQ0 to DQ3
48 to 51	OUT0 to OUT3	DQ4 to DQ7	Data input/output	DQ4 to DQ7
54	TESTX	OEX	Output Enable	$\overline{OE}$
55	CPUTESTX	CEX	Chip Enable	$\overline{CE}$
56	ATGX	RY/BYX	Ready/Busy output	RY/$\overline{BY}$
57	MD0	HVDA9	High Volt. A9 [c]	A9 (V_{ID})
58	MD1	HVDR5	High Volt. RESET [c]	$\overline{RESET}$ (V_{ID})
59	MD2	HVDOE	High Volt. OE [c]	$\overline{OE}$ (VID)
60	INITX	RSTX	Hardware Reset	$\overline{RESET}$
90	LTESTX	EQIN [a]	Access Signal EQ	
92 to 99	PR0 to PR7	AQ0 to AQ7	Address input	A0 to A7
102 to 105	LED0 to LED3	AQ8 to AQ11	Address input	A8 to A11
107 to 110	LED4 to LED7	AQ12 to AQ15	Address input	A12 to A15
113 to 115	PO4 to PO6	AQ16 to AQ18	Address input	A16 to A18
116	PO7	WEX	Write Enable	$\overline{WE}$
117	PP2	AQ 20	Address input	A 20
118	PP3	TSTX [b]	Flash Test	

a) Pins 1 and 90 must be pulled low in Flash Memory Mode.
b) Pin 118 must be pulled high in Flash Memory Mode.
c) Functionality as described in the Data Sheet of MBM29LV400C.

■ **MB91F376G**

MB91F376G			MBM29LV800C	Notes
Pin number	**Normal function**	**Flash Memory mode**		
31	BOOT	WE	WE	
32	TESTX	BYTE	BYTE	
33	CPUTESTX	TMODX		pull up
38	MONCLK	RY/$\overline{BY}$	RY/$\overline{BY}$	
39-46	INT0-INT7	D24 to D31	DQ8 to DQ15	
50	IN0	CE	CE	
51	IN1	OE	OE	
52	IN2	D20	DQ4	
53	IN3	D21	DQ5	
54	OUT0	D22	DQ6	
55	OUT1	D23	DQ7	
57	MD0	VDA9	A9 (V_{ID})	
58	MD1	VDRS	$\overline{RESET}$ (V_{ID})	
59	MD2	VDOE	OE (V_{ID})	
60	INITX	RESET	RESET	
91-93	PG3-PG5	A16-A18	A15-A17	
88	PG0	A20		pull up
89	PG1	A19	A18	
96	PWM1P0	A0	A-1	
97	PWM1M0	A1	A0	
98	PWM2P0	A2	A1	
99	PWM2M0	A3	A2	
101	PWM1P1	A4	A3	
102	PWM1M1	A5	A4	
103	PWM2P1	A6	A5	
104	PWM2M1	A7	A6	
106	PWM1P2	A8	A7	
107	PWM1M2	A9	A8	
108	PWM2P2	A10	A9	
109	PWM2M2	A11	A10	

MB91F376G			MBM29LV800C	Notes
Pin number	Normal function	Flash Memory mode		
111	PWM1P3	A12	A11	
112	PWM1M3	A13	A12	
113	PWM2P3	A14	A13	
114	PWM2M3	A15	A14	
117 to 120	PJ0-PJ3	D16 to D19	DQ0 to DQ3	

Pins not used in flash mode

MB91F376G			
Pin number	Normal function	Pin State	Notes
35	X0	input	pull up
36	X1	output	leave open
66	SIN3	output	leave open
67	SOT3	output	leave open
68	SCK3	output	leave open
27	DA0 / X0A	output / input	leave open / pull up
28	DA1/X1A	output/output	leave open
29	ALARM	input	pull up
all other signals		input	pull up

32.4 FLASH CONTROL STATUS REGISTER (FMCS)

Flash Memory Macros used in devices:

Normal Flash Macro used in : MB91F362GB

Fast Flash Macro used in : all other MB91360 devices.

address	bit 7	bit6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
00007000H	FACCEN	----	----	RDYEG	RDY	RDYI	WE	LPM
access	R/W	R/W	R/W	R	R	R/W	R/W	R/W
initial value	1 / 0 (Note 1)	1	1	0	X	0	0	0
value after Boot ROM	0	1	1	0	X	0	0	0

Note 1: In MB91F376G and MB91F364G the initial value is 0, in other devices 1.

Bit 7: FACCEN: FACC Output Enable

Normal Flash Macro :
- 0: FACC Signal enabled - recommended setting
- 1: FACC Signal disabled

Fast Flash Macro :
- 0: Synchronous read access using ATDIN and EQIN signals - recommended setting
- 1: Asynchronous read access

Bits 6,5: reserved

When writing to these bits always write "11"

Bit 4: RDYEG: Reserved bit

When the auto algorithm of flash memory is finished, this bit is set to '1'.

This bit is cleared by reading it.

0: Auto algorithm not yet finished

1: Auto algorithm finished

Bit 3: RDY:

The state of auto algorithm

0 : The state of the auto algorithm is WRITE/READ. Can't accept WRITE/READ/ERASE.

1 : It is possible to accept WRITE/READ/ERASE.

Bit 2: RDYI: Reserved bit

Bit 1: WE:

This bit is used to control writing and reading to flash memory in CPU mode

0: writing to flash memory is disabled, read access is 32 bit wide

1: writing enabled, read access 16 bit wide, auto algorithm can be used

This bit can only be written to if RDY is 1.

Bit 0: LPM:

0: normal mode

1: low power mode, can be used when CPU frequency is below 5 MHz

32.5 READ/WRITE ACCESS

In the flash memory mode, read/write access to the flash memory must be under control of the external pins. However, with the CPU access, there are no special timing constraints on read/write access because the flash memory is controlled by the flash memory interface circuit.

In this section, “write access” does not directly mean “program flash memory”. It implies “activation of the flash commands”.

32.5.1 Read/Write Access in Flash Memory Mode

Table 32.5.1 gives the setting of pins for read/write access in the Flash Memory mode. There is no special problem with control of these pins if connected to a flash memory writer. However, in other cases, timing specifications must be met. For the timing specifications, see sections 32.11 and 32.12.

Table 32.5.1 Setting Conditions of Pins for Read/Write Access in Flash Memory Mode

Operations	BG$\underline{RN}$TX ($\underline{CE}$)	RDY ($\overline{OE}$)	CS4X ($\overline{WE}$)	A0 to A18	D16 to D31	INITX
Read	L	L	H	Read address	DOUT	H
Write	L	H	L	Write address	DIN	H
Output disable	L	H	H	x	High-Z	H
Standby	H	x	x	x	High-Z	H
Hardware reset	x	x	x	x	High-Z	L

Note: This table uses pin names from F362GA. Check section 32.3 for corresponding pin names of other devices.

32.5.2 CPU Read Access, Flash Wait Control Reg. (FMWT)

■ **Flash Wait Control Register (FMWT)**

address	bit 7	bit6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
00007004H	----	----	FAC1	FAC0	EQINH	WTC2	WTC1	WTC0
access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
initial value		0	0	0/1 (Note 1)	0	0	1	1
value after Boot ROM Normal Flash Macro		0	0	0	0	0	1	1
value after Boot ROM Fast Flash Macro		0	0	1	0	0	1	1

Note 1: In MB91F376G and MB91F364G the initial value is 1, in other devices 0.

Bit 6: This bit is reserved, always set this bits to "0" when writing to this register.

Bits 5,4: FAC1, FAC0

Normal Flash Macro : These bits control the length of the low pulse for the FACC signal

Fast Flash Macro : These bits control the length of the high pulse for the ATDIN signal

FAC 1	FAC 0	length of low pulse for FACC/ length of high pulse for ATDIN
0	0	0.5 cycles of CLKB
0	1	1 cycle of CLKB
1	0	1.5 cycles of CLKB
1	1	2 cycles of CLKB

Bit 3: EQINH: This bit controlls the falling edge of the EQIN signal.

Normal Flash Macro : Always write ’0’ when writing to this register bit.

Fast Flash Macro :

0: falling edge of EQIN at falling edge of FWAITR;
1: falling edge of EQIN half cycle after falling edge of FWAITR;

Bit 2,1,0: WTC2,1,0: Wait Cycle bits

WTC2-0 are used to insert auto wait cycles for flash memory access.

WTC 2	WTC 1	WTC 0	wait cycles
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

Normal Flash Macro : Recommended settings:

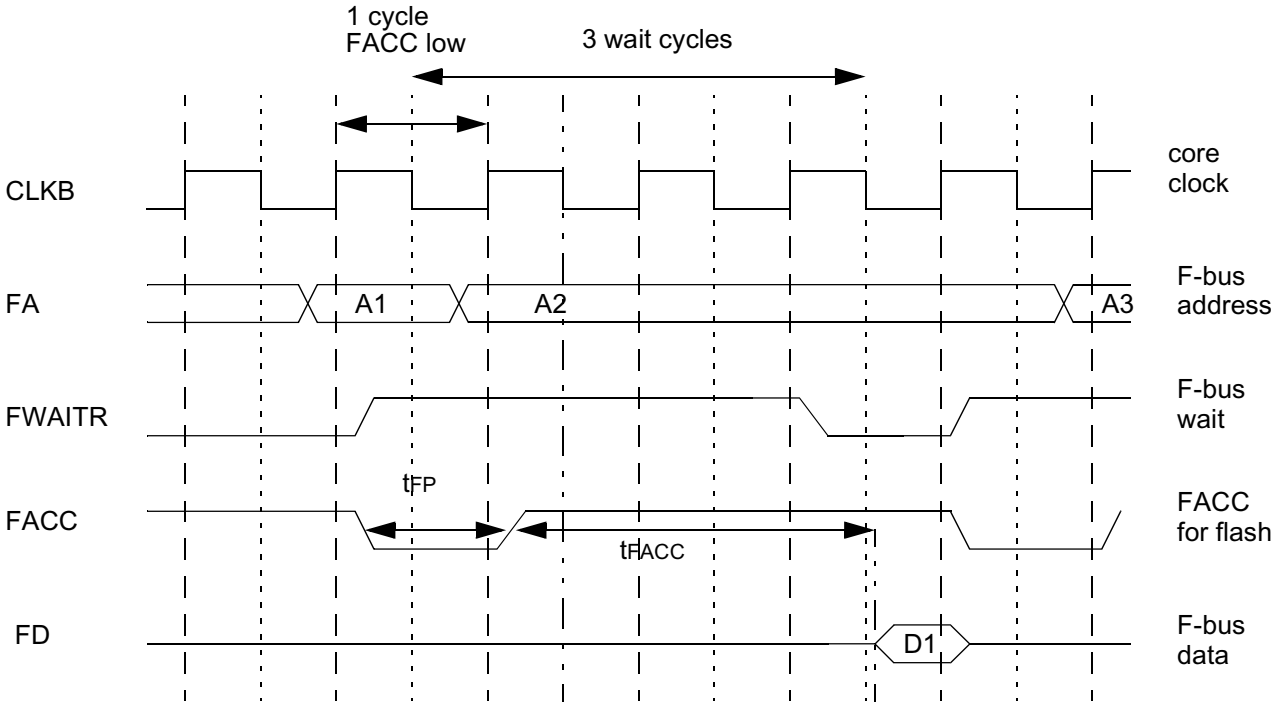
- Without applying clock modulation:

CLKB unmodulated core clock frequency [MHz]	FAC1	FAC0	EQINH	WTC2	WTC1	WTC0	FACC low cycles/wait cycles	FMWT
64	0	1	0	0	1	1	1 / 3	13H
48	0	1	0	0	1	1	1 / 3	13H
40	0	1	0	0	1	0	1 / 2	12H
32	0	0	0	0	1	0	0.5 / 2	02H
24	0	0	0	0	0	1	0.5 / 1	01H
16	0	0	0	0	0	1	0.5 / 1	01H

- When applying clock modulation:

CLKB core clock frequency [MHz]	Peak max. frequency	FAC1	FAC0	EQINH	WTC2	WTC1	WTC0	FACC low cycles/wait cycles	FMWT
48	64	0	1	0	0	1	1	1 / 3	13H
32	48	0	1	0	0	1	1	1 / 3	13H
24	40	0	1	0	0	1	0	1 / 2	12H
24	32	0	0	0	0	1	0	0.5 / 2	02H
16	24	0	0	0	0	0	1	0.5 / 1	01H

Example for flash memory read access with 1 cycle for the low time of FACC and 3 wait cycles:



The minimum value for t_{FP} is 15 ns, for t_{FACC} it is 40 ns.

Fast Flash Macro : Recommended settings:

- Without applying clock modulation:

CLKB unmodulated core clock frequency [MHz]	FAC1	FAC0	EQINH	WTC2	WTC1	WTC0	ATDIN high cycles/wait cycles	FMWT
64	0	1	0	0	1	1	1 / 3	13H
48	0	0 (1)	0 (1)	0	1	0	0.5/ 2	02H
40	0	0 (1)	0 (1)	0	1	0	0.5/ 2	02H
32	0	0	1	0	0	1	0.5 / 1	09H
24	0	0	0 (1)	0	0	1	0.5 / 1	01H
16	0	0	0	0	0	1	0.5 / 1	01H

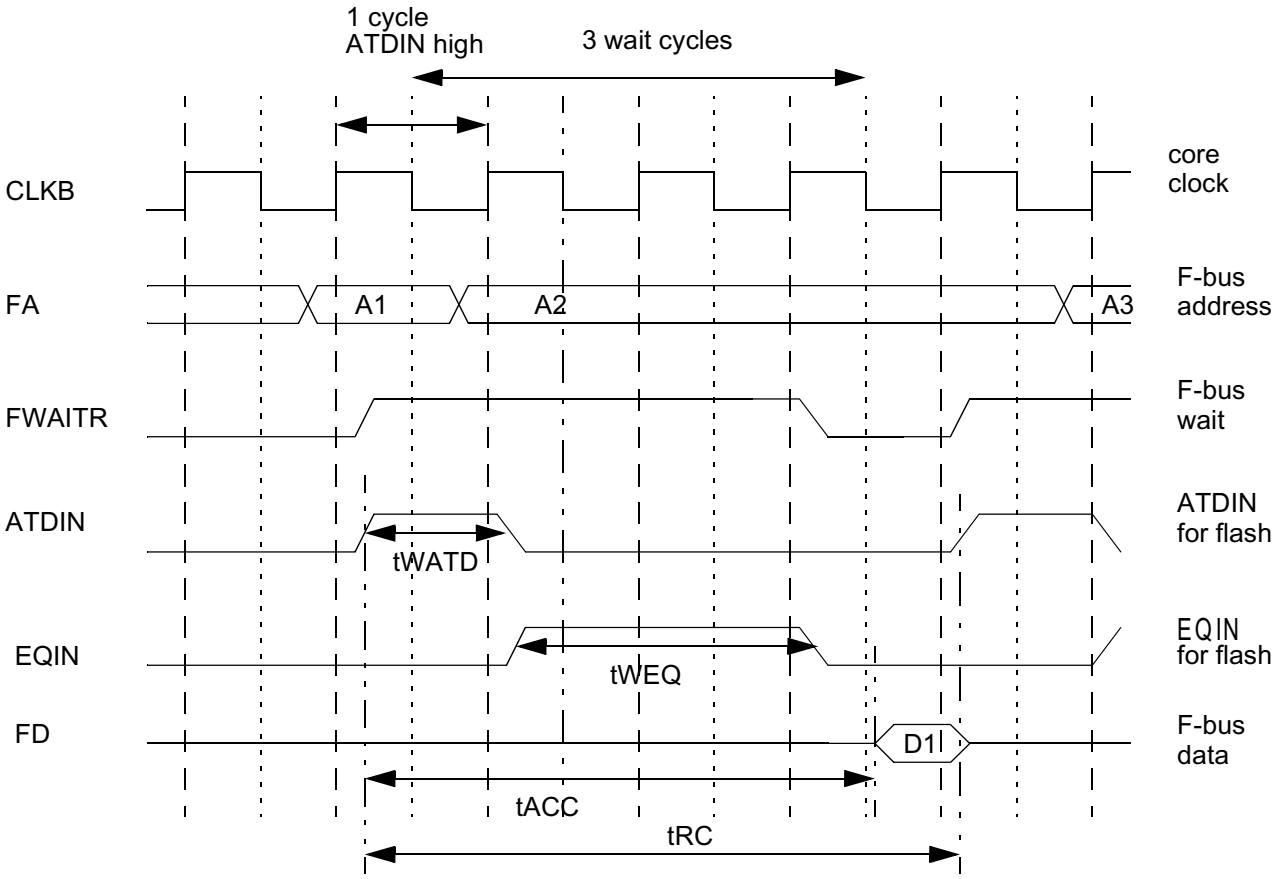
Note: (1) For MB91F376G, recommended value is 1.

- When applying clock modulation:

CLKB core clock frequency [MHz]	Peak max. frequency	FAC1	FAC0	EQINH	WTC2	WTC1	WTC0	ATDIN high cycles/wait cycles	FMWT
48	64	0	1	0	0	1	1	1 / 3	13H
32	48	0	0 (1)	0 (1)	0	1	0	0.5 / 2	02H
24	40	0	0 (1)	0 (1)	0	1	0	0.5 / 2	12H
24	32	0	0	1	0	0	1	0.5 / 1	09H
16	24	0	0	0 (1)	0	0	1	0.5 / 1	01H

Note: (1) For MB91F376G, recommended value is 1.

Example for flash memory read access with 1 cycle for the high time of ATDIN and 3 wait cycles:



The minimum value for tWATD is 10 ns, the minimum value for tWEQ is 20 ns.

The minimum value for tRC is 40 ns.

The maximum value for tACC is tWATD+tWEQ+5 ns.

32.5.3 CPU Write Access

Recommended settings for WTC2 to WTC0 for write access to the flash memory:

- FACCEN of FMCS should be set to 1 for writing, so FAC1, FAC0, EQINH register settings then have no meaning for the write operation:

■ **Without applying clock modulation:**

CLKB unmodulated core clock frequency [MHz]	WTC2	WTC1	WTC0	Wait cycles	FMWT
64	settings not allowed for writing				
48	1	0	0	4	X4H
40	1	0	0	4	X4H
32	0	1	0	2	X2H
24	0	1	0	2	X2H
16	0	0	1	1	X1H

■ **When applying clock modulation:**

CLKB core clock frequency [MHz]	Peak max. frequency	WTC2	WTC1	WTC0	Wait cycles	FMWT
48	64	setting not allowed for writing				
32	48	1	0	0	4	X4H
24	40	1	0	0	4	X4H
24	32	0	1	0	2	X2H
16	24	0	1	0	2	X2H

32.6 AUTOMATIC WRITE/ERASE

Irrespective of the Flash Memory mode or CPU mode, writing to/erasing the flash memory unit is performed by starting the flash memory automatic algorithm.

To start the automatic algorithm, various sequences of write accesses are executed in 1 to 6 cycles. They are called Flash commands.

32.6.1 Flash Commands

There are four commands for starting the automatic algorithm of the Flash Memory unit; Read/Reset, Write, Chip Erase, and Sector Erase. There are also Erase Suspend and Erase Resume commands for the sector erase operation.

Tables 32.6.1a and ***32.6.1b*** give the command sequence lists in the flash memory and CPU modes.

■ **Command sequence**

Tables 32.6.1a and ***32.6.1b*** list the commands available for the flash memory unit.

All data is written to command registers by byte access but should be written by half word access with the CPU access. Upper data bytes are ignored.

The flash memory mode permits selection of byte access or word access using the external pin $\overline{\text{BYTE}}$.

Table 32.6.1a Command Sequence List (CPU mode)

Command Sequence	Write Cycle of Bus	Write Cycle of First Bus		Write Cycle of Second Bus		Write Cycle of Third Bus		Read/Write Cycle of Fourth Bus		Write Cycle of Fifth Bus		Write Cycle of Sixth Bus	
		Address	Data	Address	Data	Address	Data	Address	Data	Address	Data	Address	Data
Read/Reset*	1	**xxxx	xxF0	—	—	—	—	—	—	—	—	—	—
Read/Reset*	4	**5554	xxAA	**aaa8	xx55	**5554	xxF0	RA	RD	—	—	—	—
Write	4	**5554	xxAA	**aaa8	xx55	**5554	xxA0	PA (even)	PD (half word)	—	—	—	—
Chip Erase	6	**5554	xxAA	**aaa8	xx55	**5554	xx80	**5554	xxAA	**aaa8	xx55	**5554	xx10
Sector Erase	6	**5554	xxAA	**aaa8	xx55	**5554	xx80	**5554	xxAA	**aaa8	xx55	SA (even)	xx30
Sector Erase Suspend		Input of address ****xxxx** or data (**xxB0H**) suspends sector erasing.											
Sector Erase Resume		Input of address ****xxxx** or data (**xx30H**) suspends and resumes sector erasing.											

Addresses in the table are the values in the CPU memory space. All addresses and data are hexadecimal values, where **x** is any value and ****** may be 08-0F.

RA: Read address
PA: Write address. Only even addresses can be specified.
SA: Sector address (See table 32.6.1c). Only even addresses can be specified.
RD: Read data
PD: Write data. Only half word data can be specified.

Note*: Two Read/Reset commands reset flash memory to the read mode.

Table 32.6.1b Command Sequence List (Flash Memory Mode)

Command Sequence	Write Cycle of Bus	Write Cycle of First Bus		Write Cycle of Second Bus		Write Cycle of Third Bus		Read/Write Cycle of Fourth Bus		Write Cycle of Fifth Bus		Write Cycle of Sixth Bus	
		Address	Data	Address	Data	Address	Data	Address	Data	Address	Data	Address	Data
Read/Reset*	1	*xxxx	F0	—	—	—	—	—	—	—	—	—	—
Read/Reset*	4	*aaaa	AA	*5554	55	*aaaa	F0	RA	RD	—	—	—	—
Write	4	*aaaa	AA	*5554	55	*aaaa	A0	PA (even)	PD (word)	—	—	—	—
Chip Erase	6	*aaaa	AA	*5554	55	*aaaa	80	*aaaa	AA	*5554	55	*aaaa	10
Sector Erase	6	*aaaa	AA	*5554	55	*aaaa	80	*aaaa	AA	*5554	55	SA (even)	30
Sector Erase Suspend		Input of address ***xxxx** or data (**B0H**) suspends sector erasing.											
Sector Erase Resume		Input of address ***xxxx** or data (**30H**) suspends and resumes sector erasing.											

Addresses in the table are values for writer addresses. All addresses and data are hexadecimal values, where **x** is any value and * may be 0-7.

RA: Read address
PA: Write address
SA: Sector address (See table 32.6.1c)
RD: Read data
PD: Write data. Only half word data can be specified.

Note*: Two Read/Reset commands reset flash memory to the read mode.

■ Read/Reset command

There are two Read/Reset commands; one is executed in one bus operation, and the other is executed at three bus operations; the command sequence of both is essentially the same.

The flash memory unit is in the read/reset state at default and always enters this state at power-on and at the normal termination of command execution. The read/reset state can be the wait state for input of another command.

In the read/reset state, data can be read by normal read access. When the CPU is in a normal mode, like mask ROM, program access from the CPU is enabled.

Consequently, this command is not needed for reading data and is used mainly to reset the automatic algorithm when command execution has not terminated for some reason.

■ Write command

The Write command is executed in four bus operations. In the command sequence, two **unlock** cycles are executed, followed by the **Write Setup** command and write data cycle. Automatic writing is started at the end of the 4th write cycle.

When the CPU is in a normal mode, only even addresses can be specified in the write data

cycle. Specifying odd addresses disables correct writing. Therefore, writing in a normal mode is performed in half words at even addresses. There are no such limitations in the Flash Memory mode.

After the command sequence for the automatic write algorithm has been executed, the flash memory unit generates correct write pulses created automatically to verify the margins of written data. The end of automatic writing can be determined using the data polling function (See section 32.6.2 "Execution State of Automatic Algorithm" on page 699). After the completion of writing, the flash memory unit returns to the read/reset state.

All commands are ignored during writing. If a hardware reset occurs during writing, data being written to the address become invalid.

Writing is possible in any address order or even beyond sector boundaries. However, execution of one Write command, results in writing one half word of data with the CPU access, and one byte or one half word one data of in the Flash Memory mode.

Data **0** cannot be returned to data **1** by writing. When data **1** is written to data **0**, the flash memory unit may hang up or possibliy complete the write operation without any error. However when the data is read in the read/reset state, it remains **0**. Only erasing enables data **0** to be set to data **1**.

Figure 32.6.1a shows the writing procedure using the Write command.

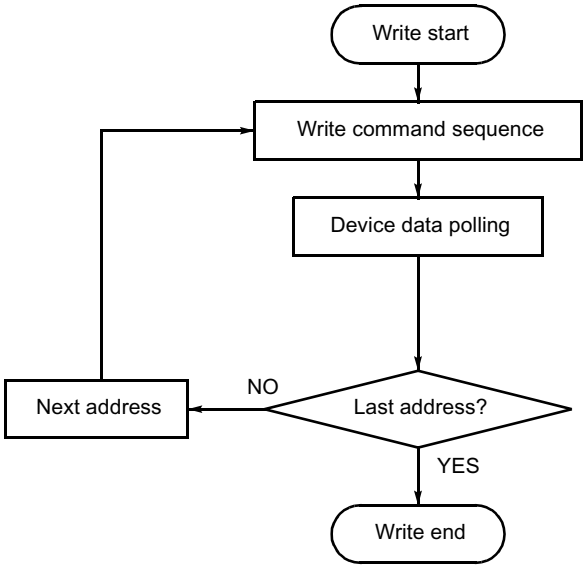


Figure 32.6.1a Writing Procedure Using Write Command

■ Chip Erase command

The Chip Erase command is executed in six bus operations. In the command sequence, two `unlock` cycles are executed, followed by the **`Write Setup`** command. The two same **`unlock`** cycles are further continued to write the Chip Erase command. Chip erasing is started at completion of the 6th write cycle.

Before chip erasing, the user need not perform writing to the flash memory unit. During execution of the automatic erase algorithm, the flash memory unit writes **0** patterns for verification before automatically erasing all cells.

The end of automatic erasing can be determined using the data polling function (See section 32.6.2 "Execution State of Automatic Algorithm" on page 699). Figure 32.6.1b shows the chip erasing procedure using the Chip Erase command.

■ Sector Erase command

The Sector Erase command is executed in six bus operations. In the command sequence, two **unlock** cycles are executed, followed by the Write Setup command. The two same **unlock** cycles are continued to write a write sector erase code to a sector address in the 6th cycle, and waiting 50 μs for sector erasing.

When erasing more than one sector, the sector erase code (30H) is written to the sector address, following the above sequence. Sector erasing is started after a 50-μs period of waiting for sector erasing is completed after the last sector erase code has been written. That is, when erasing more than one sector simultaneously, the next sector to be erased must be input within 50 μs and subsequent sectors may not be accepted. The validity of the next sector erase code can be monitored by the sector erase timer flag (see 32.6.2). When the commands other than the Sector Erase or the Sector Erase Suspend command are accepted during waiting for sector erasing, the flash memory unit returns to the read/reset state, ignoring the previous command sequence. In this case, erasing is completed by erasing the sector again.

Although sector addresses can be selected in any combination and order, selected sectors are erased in ascending order.

Before sector erasing, the user need not perform writing to the flash memory unit. Writing is performed to all memory within the sectors to be erased automatically. Other sectors are not affected during sector erasing.

The end of automatic sector erasing can be determined using the data polling function (See 32.6.2). After the completion of erasing, the flash memory unit returns to the read/reset state. Other commands are ignored during sector erasing.

Figure 32.6.1b shows the sector erasing procedure using the Sector Erase command.

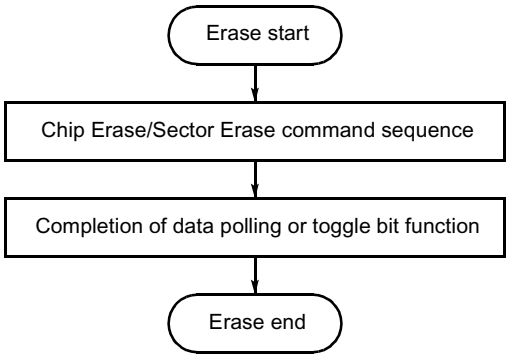


Figure 32.6.1b Chip Erasing/Sector Erasing Procedure Using Chip Erase/Sector Erase Command

- Sector Address

A sector address is an arbitary address representing each sector. Any accessible address within the sector and mode is sufficient.

Table 32.6.1c Sector Address for half word mode

Sector	A[18:13] in Flash Memory mode						Address range in Flash Memory mode	Address in CPU mode
	A18	A17	A16	A15	A14	A13		
SA13	1	1	1	1	1		7C000H to 7FFFFH	F8002H
SA12	1	1	1	1	0	1	7A000H to 7BFFFH	F4002H
SA11	1	1	1	1	0	0	78000H to 79FFFH	F0002H
SA10	1	1	1	0			70000H to 77FFFH	E0002H
SA9	1	1	0				60000H to 6FFFFH	C0002H
SA8	1	0	1				50000H to 5FFFFH	A0002H
SA7	1	0	0				40000H to 4FFFFH	80002H
SA6	0	1	1	1	1		3C000H to 3FFFFH	F8000H
SA5	0	1	1	1	0	1	3A000H to 3BFFFH	F4000H
SA4	0	1	1	1	0	0	38000H to 39FFFH	F0000H
SA3	0	1	1	0			30000H to 37FFFH	E0000H
SA2	0	1	0				20000H to 2FFFFH	C0000H
SA1	0	0	1				10000H to 1FFFFH	A0000H
SA0	0	0	0				00000H to 0FFFFH	80000H

■ Sector Erase Suspend command

The Sector Erase Suspend command suspends sector erasing by the automatic algorithm to enable reading of data from the sectors not being erased (writing is not possible). This command is enabled only during sector erasing including the period of waiting for sector erasing, and is ignored during chip erasing and writing. This command is executed by writing the sector erase suspend code (B0H). In this case, any address in the flash memory area is specified.

As soon as this command is input during the period of waiting for sector erasing, the flash memory unit completes waiting for sector erasing to suspend erasing and enters the erase suspend state.

When the Sector Erase Suspend command is input during sector erasing after the period of waiting for sector erasing, the flash memory unit enters the erase suspend state after a maximum of 20 μs.

Whether the flash memory unit has stopped erasing and is in the erase suspend state can be determined using the data polling function or toggle bit function (See 32.6.2). Input of the Sector Erase Suspend command in the erase suspend state is ignored.

In the erase suspend state, data can be read by normal read access to the sectors not being erased.

■ **Sector Erase Resume command**

The sector erase resume command resumes sector erasing suspended by the sector erase suspend command. This command is executed by writing the erase resume code (30H). In this case, any address in the flash memory area is specified.

In the erase resume state, the flash memory unit ignores input of the sector erase resume command but accepts input of the sector erase suspend command.

■ **Fast Mode**

The flash memory has fast mode function. This mode dispenses with the initial two unlock cyclesrequired in the standard program command sequence by writing fast mode command into the command register.In this mode, the required bus cycle for programming is two cycles instead of four bus cycles in standard programcommand. (Do not write erase command in this mode.) The read operation is also executed after exiting this mode. To exit this mode, it is necessary to write fast mode Reset command into the command register. (Refer to the figure 32.8a "Extended Sector Protection Algorithm" on page 707.) The Vcc active current is required even $\overline{CE}$=V_{IH} during fast mode.

■ **Fast Programming**

During fast mode, the programming can be excecuted with two bus cycles operation. The automatic algorithm is executed by writing program set-up command (A0H) and data write cycles (PA/PD). (Refer to the figure 32.8a "Extended Sector Protection Algorithm" on page 707.)

■ **Extended Sector Protection**

In addition to normal sector protection, extended sector protection is available. This function allows to protect sector by forcing V_{ID} on MD1 pin and write a command sequence. Unlike conventional procedure, it is not necessary to force V_{ID} and control timing for control pins. Only the MD1 pin requires V_{ID} for sector protection in this mode. With this condition, the operation is initiated by writing the set-up command (60H) into the command register. Then, the sector address pins (A_{18}, A_{17}, A_{16}, A_{15},A_{14} and A_{13}) and (A_6, A_1, A_0) = (0,1,0) should be set to the address of the sector to be protected (recommend to set V_{IL} for the other addresses pins), and the extended sector protect command (60H) must be written. A sector typically is protected in 150μs. To verify programming of the protection circuitry, the sector address pins (A_{18}, A_{17}, A_{16}, A_{15},A_{14} and A_{13}) and (A_6, A_1, A_0) = (0,1,0) should be set and command (40H) must be written. Following the write command, a logical “1” at device output D16 will produced for the protected sector in the read operation. If the output data is logical “0”, please repeat to write extended sector protect command (60H) again.To terminate the operation, it is neessary to set MD1 pin to V_{IH}.

32.6.2 Execution State of Automatic Algorithm

Since the writing / erasing flow is controlled by the automatic algorithm, the flash memory unit has a hardware sequence flag and ready / busy signal to inform units outside the flash memory unit of its internal operation and operation completion.

■ **List of hardware sequence flags**

The hardware sequence flags consist of the 4 outputs D23, D22, D21, and D19. They are a data polling flag, toggle bit flag, timing limit exceeding flag, and sector erase timer flag.

Hardware sequence flags can be referenced by read-accessing addresses in the flash memory area after starting the automatic algorithm. This is common to both the CPU access and flash memory mode.

Table 32.6.2a lists the functions of hardware sequence flags.

Table 32.6.2a List of Functions of Hardware Sequence Flags

State		D23	D22	D21	D19
State change in normal operation	Write => Write completed (when write address specified)	$\overline{D23}$ => D23	Toggle => Stop	0 =>1	0
	Chip/Sector Erase => Erase completed	0 =>1	Toggle => Stop	0 =>1	1
	Sector Erase wait => Erase started	0	Toggle	0	0 =>1
	Erase => Sector Erase suspended (Sector being erased)	0 =>1	Toggle =>1	0	1 =>0
	Sector Erase Suspend => Resume (Sector being erased)	1 => 0	1 =>Toggle	0	0 =>1
	Sector Erase Suspend (Sector not being erased)	DATA	DATA	DATA	DATA
Abnormal operation	Write	$\overline{D23}$	Toggle	1	0
	Chip/Sector Erase	0	Toggle	1	1

Note: D20 is used for the device test and D18 to D16 are reserved for future use.

■ Data polling flag (D23)

The data polling flag is mainly used to notify that the automatic algorithm is executing or has been terminated. This flag is valid for write, chip erase, or sector erase.

- Writing

  Read access during execution of the writing algorithm causes the flash memory unit to output the reverse data of bit 7 last written, irrespective of the value at the address specified by the address signal.

  Read access at the end of the writing algorithm causes the flash memory unit to output bit 7 of the read value at the address specified by the address signal.

- Chip/sector erasing

  Read access during execution of the chip/sector erasing algorithm causes the flash memory unit to output **0**, irrespective of the value at the address specified by the address signal. Read access at the end of the chip/sector erasing algorithm causes the flash memory unit to output **1**.

- Sector erasing suspension

  Read access during sector erasing suspension causes the flash memory unit to output **1** if the address specified by the address signal belongs to the sector being erased. The flash memory unit outputs bit 7 of the read value at the address specified by the address signal if the address specified by the address signal does not belong to the sector being erased.

  Referencing this flag together with the toggle bit flag permits a decision on whether the flash memory unit is in the erase suspended state and which sector is being erased.

Read access to the specified address while the automatic algorithm is ignored. At data reading, other bits can be output correct data after the end of data polling by bit 7. Data reading after the end of the automatic algorithm should be performed after completion of data polling has been checked.

■ Toggle bit flag (D22)

Like the data polling flag, the toggle bit flag is mainly used to notify that the automatic algorithm is executing or has been terminated using the toggle bit function. This flag is valid for Write, Chip Erase, or Sector Erase.

- Writing, chip and sector erasing

  Continuous read access during execution of the writing and chip/sector erasing algorithms causes the flash memory unit to toggle the **1** or **0** state at every read cycle, irrespective of the value at the address specified by the address signal.

  Continuous read access at the end of the writing and chip/sector erasing algorithms causes the flash memory unit to stop toggling bit 6 to output bit 6 of the value at the address specified by the address signal.

  At writing, if the sector where data is to be written is rewrite-protected, the toggle bit terminates the toggle operation of about 2 μs and then terminates without rewriting data. At erasing, if all the selected sectors are write-protected, the toggle bit performs toggling for about 100 μs and then returns to the read/reset state without rewriting data.

- Sector erasing suspension

  Read access during sector erase suspension causes the flash memory unit to output **1** if the address specified by the address signal belongs to the sector being erased. The flash memory unit outputs the value of bit 6 of the value at the address specified by the address signal if the address specified by the address signal does not belong to the sector being erased.

■ Time limit exceeding flag (D21)

The time limit exceeding flag is used to notify that the automatic algorithm has executed beyond the predetermined time (internal pulse count) in the flash memory unit. This flag is enabled for Write, Chip Erase, or Sector Erase.

Read access after the write, chip erase, or sector erase command causes the flash memory unit to output **0** if the automatic algorithm is executed within the specified time (time required for writing/erasing) and to output **1** if the algorithm is executed beyond the specified time, irrespective of the value at the address specified by the address signal. This is done irrespective of whether the automatic algorithm is executing or has terminated, making it possible to determine whether writing/erasing was successful or unsuccessful (See ***Examples of using hardware sequence flags*** on page 702). That is, when this flag outputs **1**, writing or erasing can be determined to have been unsuccessful, if the automatic algorithm is regarded as still being executed by the data polling function or toggle bit function.

For example, if the user writes **1** to the address where **0** is written, the fail state occurs. In this case, the flash memory hangs up and execution of the automatic algorithm is not terminated. Consequently, valid data cannot be output from D23. Bit 6 exceeds the time limit without stopping the toggle operation, and bit 5 outputs **1**. It should be noted that this state indicates that flash memory is not defective, but has not been used properly.

If this state occurs, the Reset command should be executed.

■ Sector erase timer flag (D19)

The sector erase timer flag is used to notify that the automatic algorithm is executed during the period of waiting for sector erasing after the sector erase command has started. This flag is valid with sector erase.

- Sector erasing

  Read access after the sector erase command causes the flash memory unit to output **0** if the

automatic algorithm is executed within the period of waiting for sector erasing and to output **1** if the algorithm is executed beyond the period of waiting for sector erasing, irrespective of the value at the address specified by the address signal.

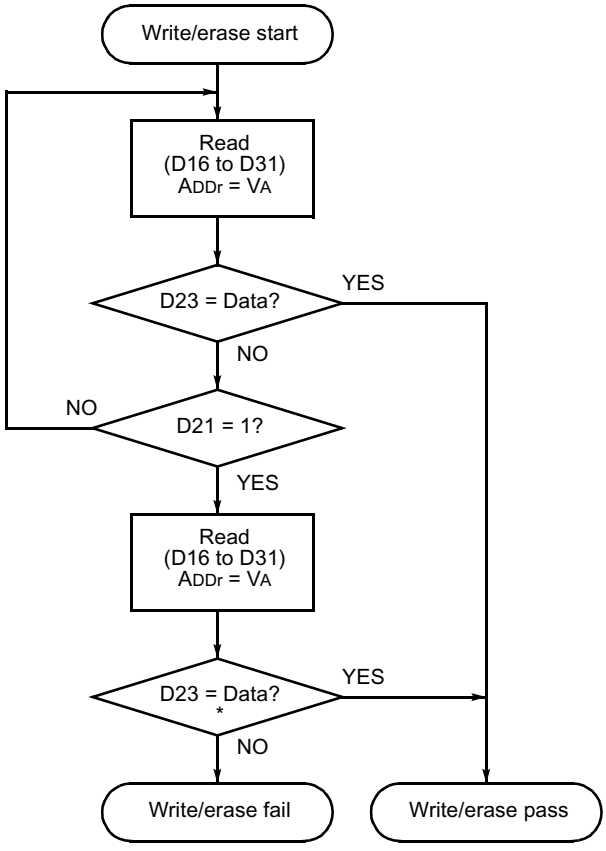
When the data polling function or toggle bit function indicates that the erasing algorithm is executing, internally-controlled erasing has already started if this flag is **1**. Continuous writing of sector erase codes or commands other than the sector erase suspend command are ignored until the data polling function or toggle bit function indicates the end of erasing. If this flag is **0**, the flash memory unit accepts writing of additional sector erase codes. To ensure this, the state of this flag should be checked by software before continuing to write sector erase codes. If this flag is **1** after the second state check, writing of additional sector erase codes may not be accepted.

■ Sector erase suspension

Read access during sector erase suspend causes the flash memory unit to output **1**, if the address specified by the address signal belongs to the sector being erased. The flash memory unit outputs bit 3 of the value at the address specified by the address signal, if the address specified by the signal address does not belong to the sector being erased.

■ Examples of using hardware sequence flags

Use of the previously-mentioned hardware sequence flags permits determination of the state of the automatic algorithm in the flash memory unit. Figures 32.6.2a and 32.6.2b give the flowcharts of decision on writing/erasing using the data polling function and toggle bit function.



VA= Write address
= Address of sector erased during sector erasing
= Address of sector not protected during chip erasing

Figure 32.6.2a Flowchart of Decision on Writing/Erasing Using Data Polling Function

Note: **D23** is changed as **D21** is changed. **D23** must be re-checked, even if **D21** = **1**.

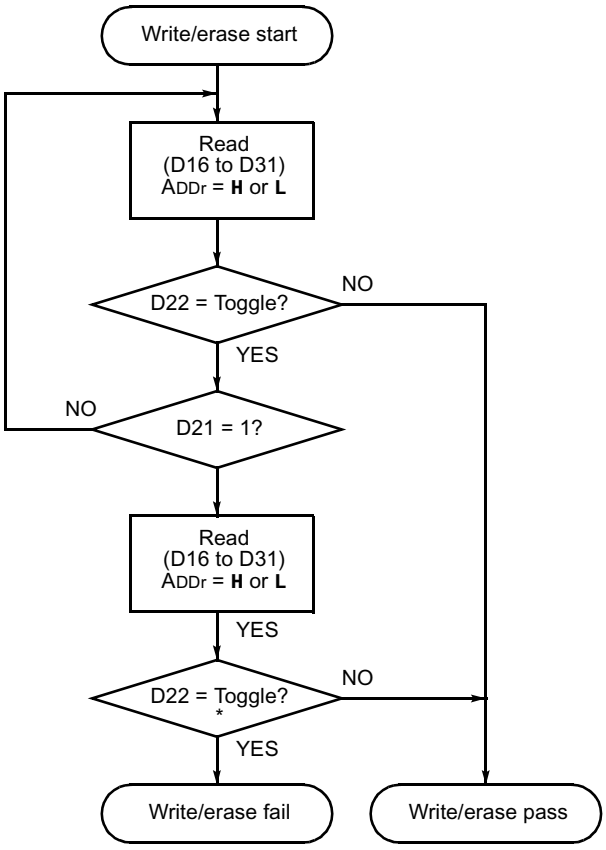


Figure 32.6.2b . Flowchart of Decision on Writing/Erasing Using Toggle Bit Function

Note: D22 stops toggling as D21 is changed to 1. D22 must be re-checked, even if D21 = 1.

■ Ready/busy signal

The flash memory unit has hardware sequence flags, such as a data polling flag and toggle bit flag, as well as a hardware signal, such as a ready/busy signal to notify that the automatic algorithm is executing or terminated.

This signal is for usage in the flash memory mode only:

The ready/busy signal asynchronously supplies an RY/$\overline{BY}$ signal from the flash memory unit outside the chip as an open-drain output BRQ pin. Connecting a pull-up resistor to V_{CC} allows parallel connection of several BRQ pins.

When the output of the BRQ pin is **0**, the flash memory unit is writing or erasing and busy. Neither Write nor Erase commands can be accepted. When the output of the BRQ pin is **1** with an external pull-up resistor connected, the flash memory unit is ready for reading/writing. In the erase suspend mode, the output of the BRQ pin is **1** with an external pull-up resistor connected. The BRQ pin indicates the ready state during external reset.

32.7 SECTOR PROTECT OPERATION

The flash memory unit has the sector protect function to disable illegal writing/erasing in sectors. Once protected, sectors remain unchanged unless the device is damaged. If sectors are protected temporarily, the protection can be canceled for writing/erasing. This operation is performed using the sector protect operations.

The sector protect operation does not have an automatic algorithm like the writing/erasing algorithm. It can be executed only in the flash memory mode. Therefore, this operation can only be performed using a flash memory writer

■ List of sector protect operations

There are three sector protect operations; Enable sector protect, verify sector protect, and temporary sector protect cancel.

Note: Set address pins **A18** to **A11** corresponding to sectors listed in Table 32.6.1c "Sector Address for half word mode" on page 698.

■ Enable Sector Protect

Enable Sector Protect provides writing to the protection circuit in the flash memory unit. This operation disables writing and erasing in any combination of the 13 sectors.

At this operation, the sector addresses (**A18**, **A17**, **A16**, **A15**, **A14**, **A13**,) of the sectors to be protected must be set to the address signals and **A9** must be set to **0**. See 32.6.1c for the correspondence between sectors and sector addresses.

After V_{ID} (12 V) has been applied to **MD2** and **MD0** to set BGRNTX ($\overline{CE}$) to **0**, writing to the protection circuit is started at the fall of the CS4X ($\overline{WE}$)pulse and terminated at the rise of the CS4X ($\overline{WE}$) pulse. Sector addresses must be kept constant while the CS4X ($\overline{WE}$) pulse is applied.

Once applied, sector protection cannot be canceled. This disables subsequent writing/erasing to/from protected sectors.

■ Verify Sector Protect

Verify Sector Protect verifies writing to the protection circuit in the flash memory unit.

At this operation, BGRNTX($\overline{CE}$) and RDY($\overline{OE}$)are set to **0**s and V is applied to **MD0** with CS4X($\overline{WE}$)set to **1** (margin mode). When a sector address (any one of **A18, A17, A16**, **A15**, **A14**, **A13**) is set to an address signal and read under the condition of (**A7**, **A2**, **A1**) = (**0**, **1**, **0**), **1** is output to **D16** in the protected sectors. **00H** is read in the unprotected sectors.

Figure 32.7a shows the sector protection algorithm using Enable Sector Protect and Verify Sector Protect.

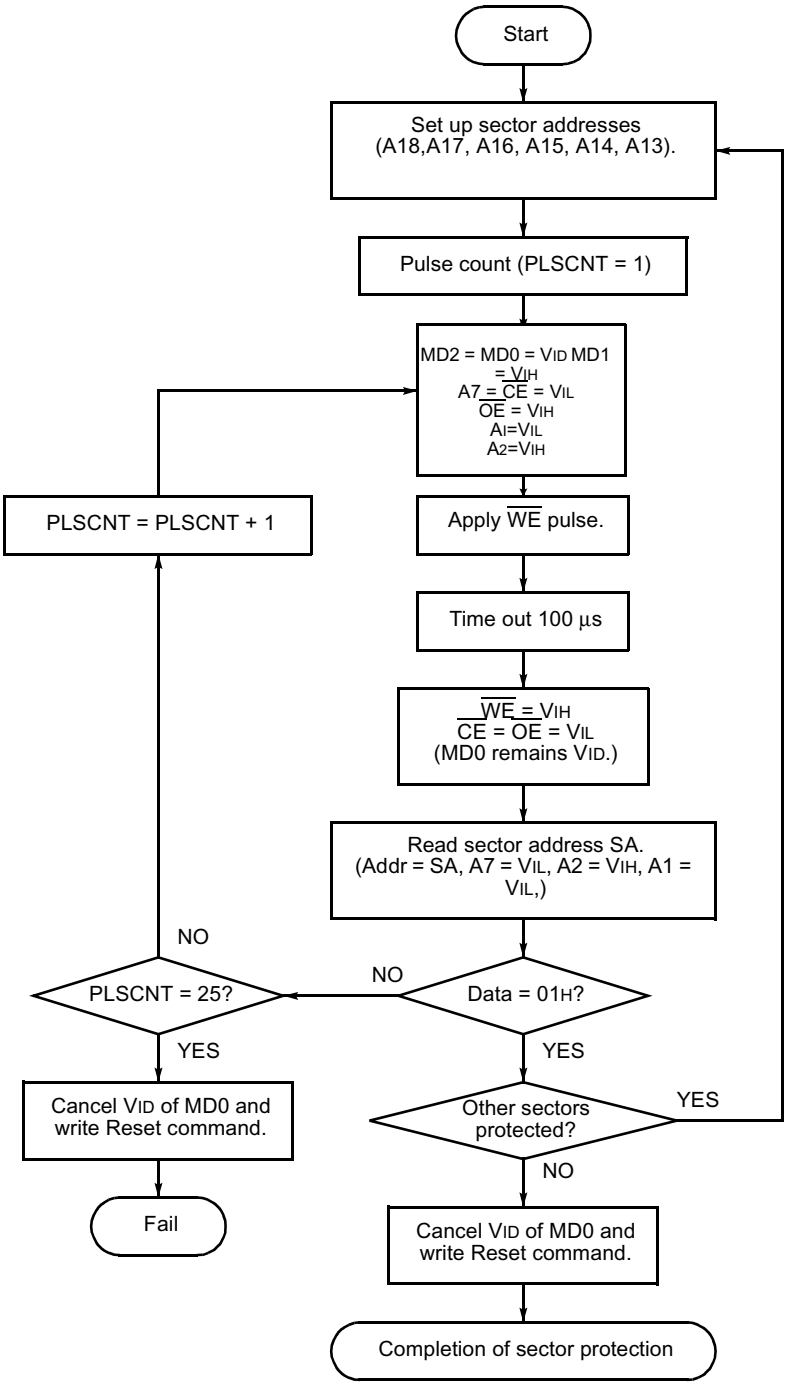


Figure 32.7a Sector Protection Algorithm Using Enable Sector Protect and Verify Sector Protect

■ Temporary Sector Protect Cancel

Writing/erasing to/from sectors protected by Enable Sector Protect is impossible unless the device is damaged. Temporary Sector Protect Cancel allows temporary cancellation of information on sector protection set previously. This operation is set by continuously-applying V_{ID} to **MD1**. At this time, all data on sector protection set previously are ignored, writing/erasing to/from all sectors is enabled.

When **MD1** is set to **1** (5 V), this operation is canceled and all previously-protected sectors are protected again.

Figure 32.7b shows the algorithm for temporary sector protect cancellation.

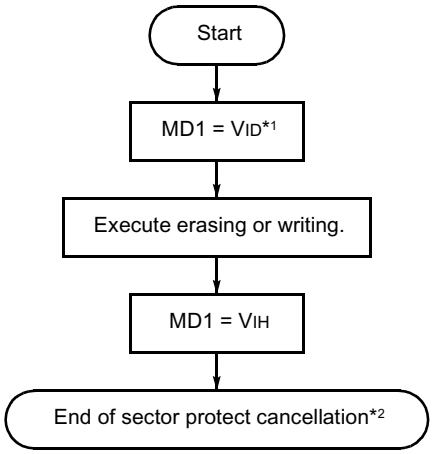


Figure 32.7b Algorithm for Temporary Sector Protect Cancellation

*1: Protection of all protected sectors is canceled.

*2: Previously-protected sectors are protected again.

32.8 EXTERNAL COMMAND

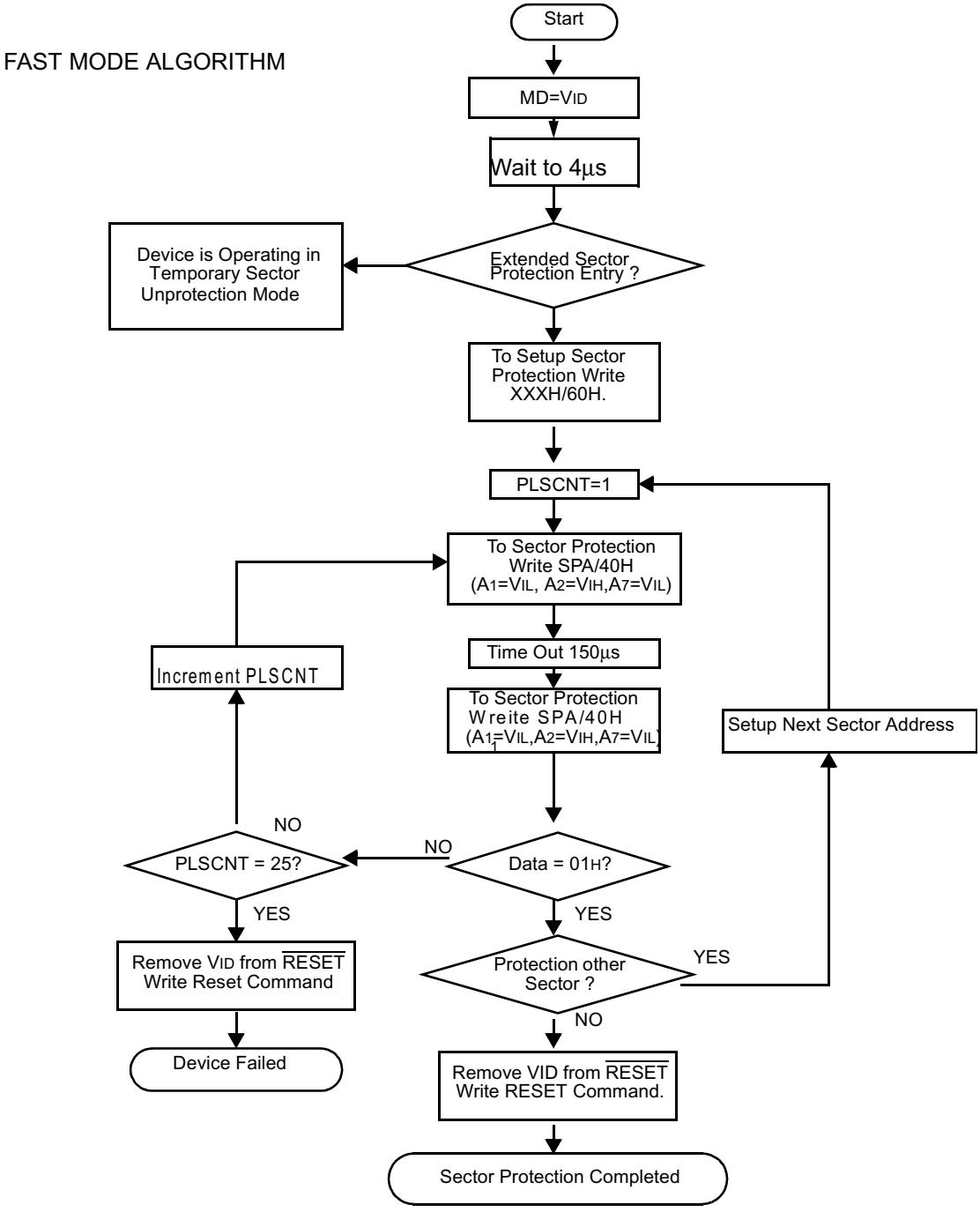


Figure 32.8a Extended Sector Protection Algorithm

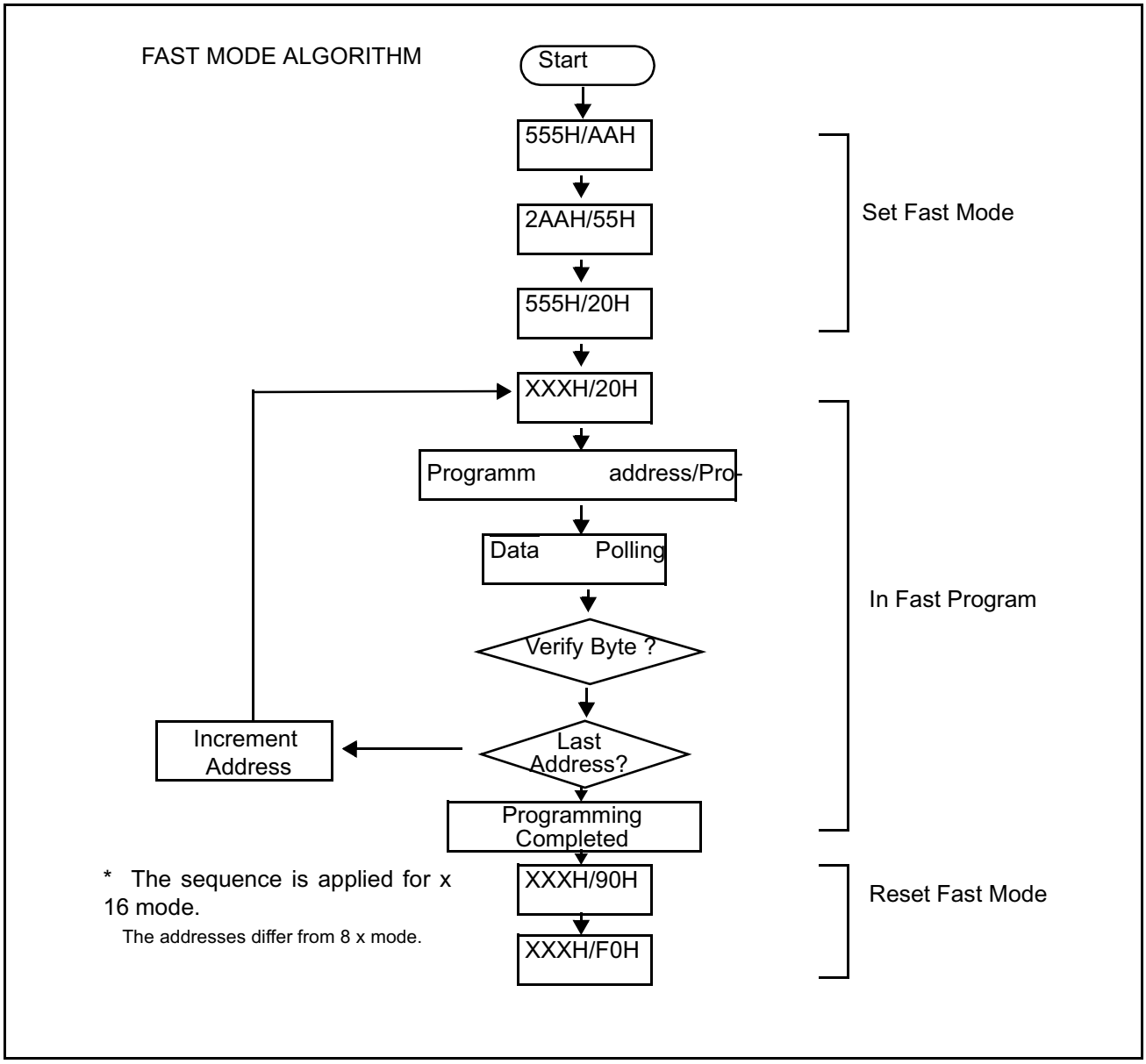


Figure 32.8b Embedded Programm™ Algorithm for Fast Mode

32.9 CONNECTION TO FLASH MEMORY

The Flash Memory mode of the MB91360 Series is intended mainly for external connection to a flash memory writer. As indicated in table 32.3.3a "Pins used in flash memory mode" on page 673, there is a slight difference between the external pins of the MB91360 Series and the MBM29LV400C (4 Mbit flash memory). Connection to an MBM29LV400C writer requires the socket adapter shown in figure 32.9a.

The external pins of the MBM29LV400C (4 Mbit flash memory) correspond to those of the MB91360 Series in the Flash Memory mode as indicated in table 32.3.3a. Connection to an flash writer requires a socket adapter with the same pin arrangement.

As indicated in table 32.3.3a, there is also a difference between the pins for supply of VID voltage (12 V) of the MBM29LV400C and the MB91360 Series. To allocate the normal digital voltage and VID voltage supplied from the writer to each pin of the MB91360 Series, the diode must be clamped by the socket adapter.

Figure 32.9a shows an example of connection from a flash memory writer to some MB91360 series devices using the socket adapter. Connections not shown in this figure should be made as indicated in table 32.3.3a.

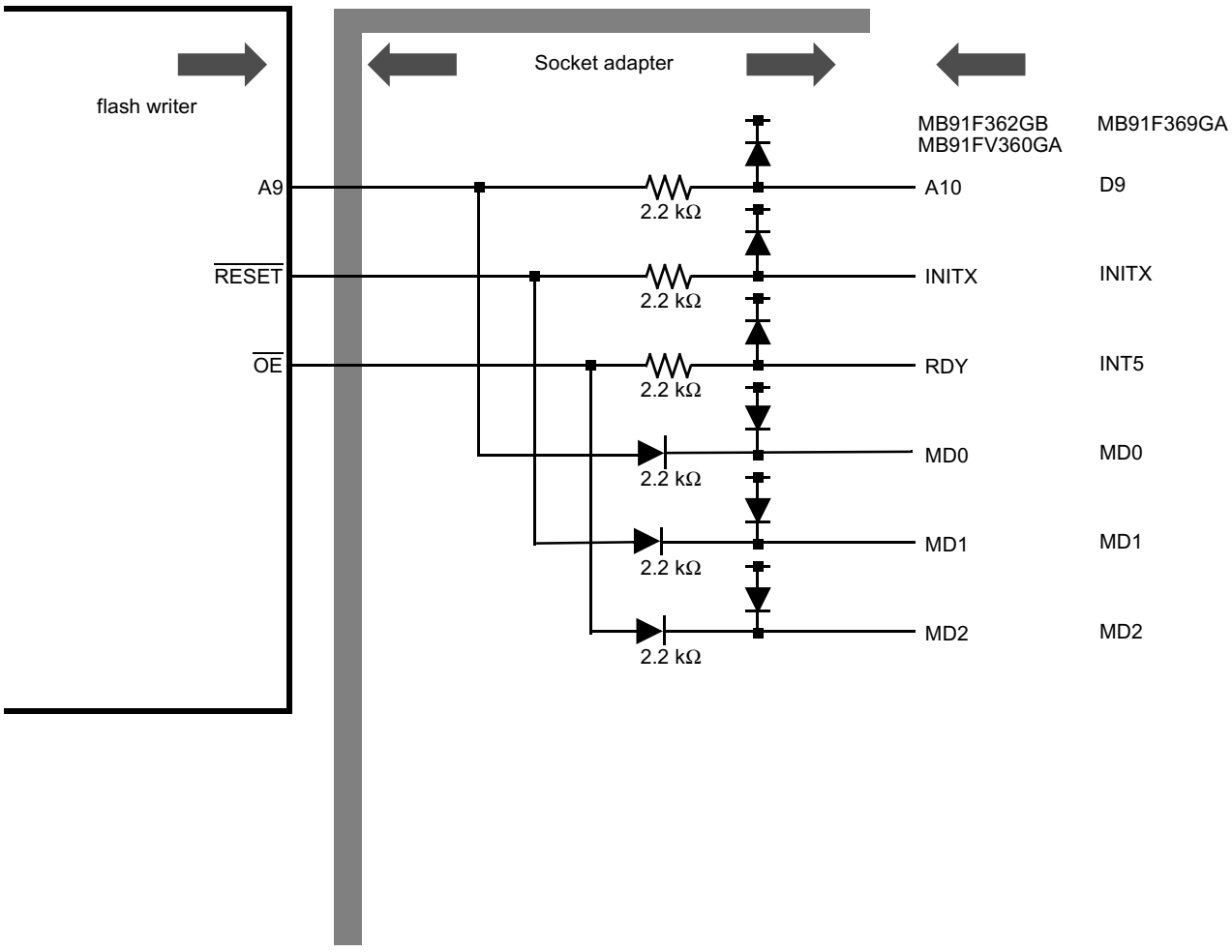


Figure 32.9a Example of Connection to Flash Memory Writer

32.10 NOTES FOR USE OF FLASH MEMORY

Notes on the Flash Memory in MB91360 devices are given below.

■ **Input of hardware reset (INITX)**

To input a hardware reset when the automatic algorithm is not started, where reading is in progress, a minimum of 500 ns should be taken at a low-level width. In this case, a maximum of 500 ns is required until data can be read from the flash memory after a hardware reset has been activated.

Similarly, to input a hardware reset when the automatic algorithm is activated, where writing/ erasing is in progress, a minimum of 50 ns should be taken in a low-level width. In this case, 20 μs are required until data can be read after the executing operation has been terminated to initialize the flash memory.

A hardware reset during writing undefines data being written. A hardware reset during erasing may make the sector being erased unusable.

Canceling software reset, watchdog timer reset, and hardware standby

When writing/erasing the flash memory with the CPU access and if reset conditions occur while the automatic algorithm is active, the CPU may run away. This occurs because these reset conditions cause the automatic algorithm to continue without initializing the flash memory unit, possibly preventing the flash memory unit from entering the read state when the CPU starts the sequence after the reset has been deasserted. These reset conditions should be inhibited during writing/erasing the Flash Memory.

■ **Program access to Flash Memory**

When the automatic algorithm is operating, read access to the flash memory is disabled. With the memory access mode of the CPU set to the internal ROM mode, writing/erasing should be started after switching the program area to another area such as RAM.

In this case, when sectors containing interrupt vectors are erased, interrupt processing cannot be executed.

For the same reason, all interrupt sources should be disabled while the automatic algorithm is operating.

■ **Hold function**

When the CPU accepts a hold request, the Write signal $\overline{WE}$ of the flash memory unit may be skewed and many cause erroneous writing/erasing. When the acceptance of a hold request is enabled,ensure that the **WE** bit of the control status register (**FMCS**) is **0**.

■ **Applying V_{ID}**

Applying V_{ID} required for the sector protect operation should always be started and terminated when the supply voltage is on.

32.11 TIMING DIAGRAMS IN FLASH MODE

Each timing diagram for the external pins of the MB91360 Series in the Flash Memory mode is shown below. For the respective AC specifications, see figure 32.11a.

■ **Data read by read access**

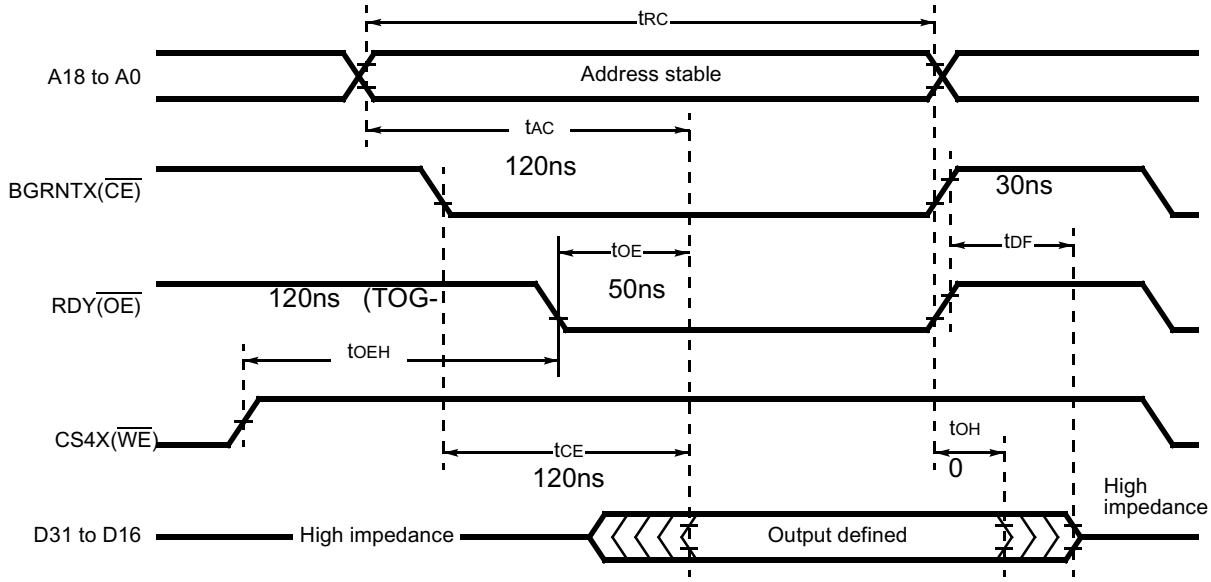
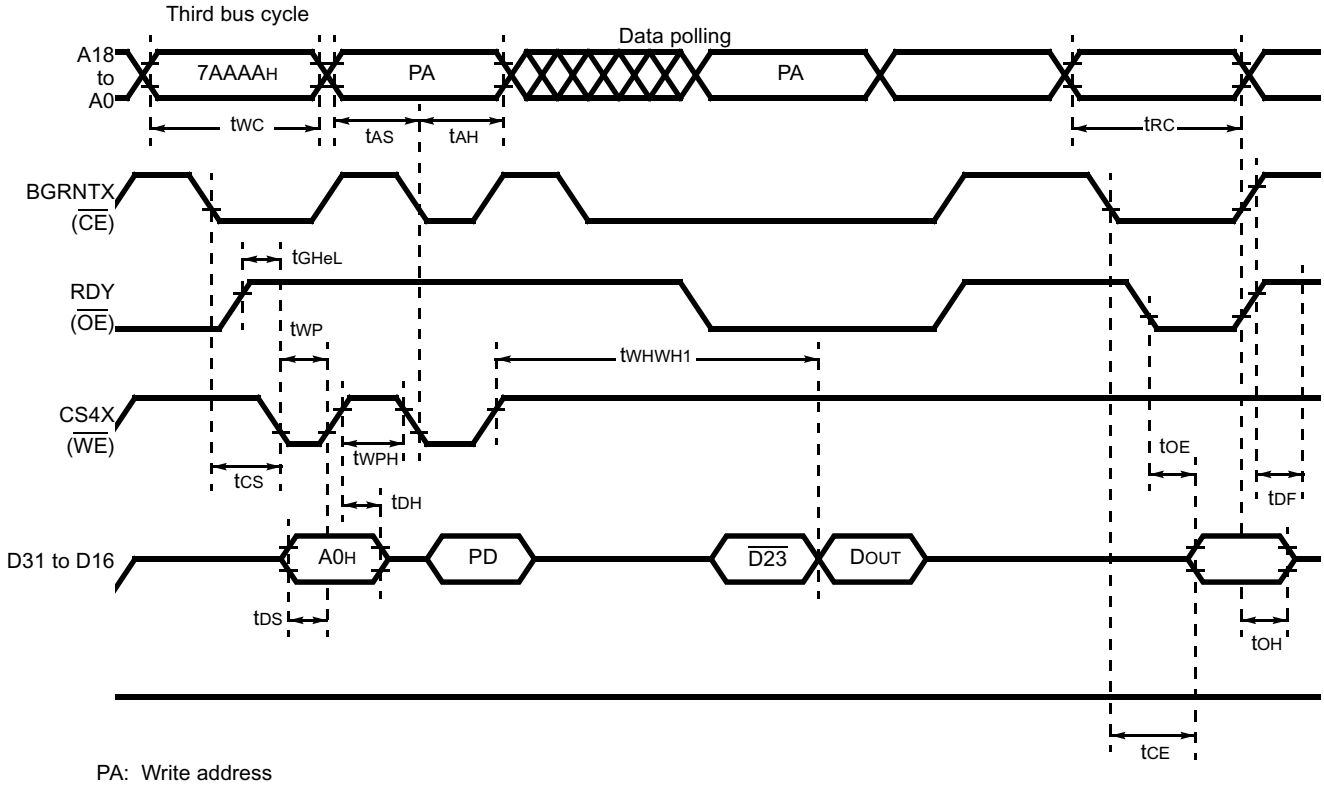


Figure 32.11a Timing Diagram for Read Access

■ Write Data polling Read ($\overline{WE}$ control)

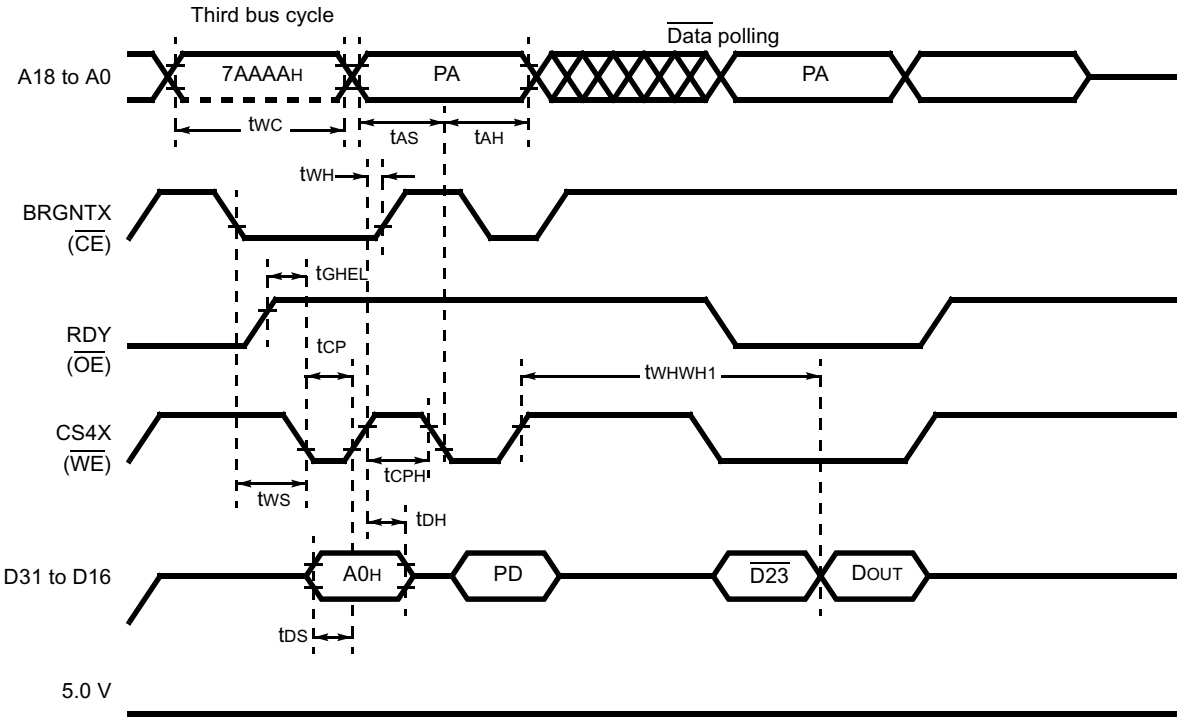


PA: Write address
PD: Write data
$\overline{D23}$: Reverse output of write data
DOUT: Output of write data

Figure 32.11b Timing Diagram for Write Access ($\overline{WE}$ Control)

Note: The last two bus cycle sequences out of the four are described.

■ Write Data polling Read ($\overline{CE}$ control)



PA: Write address
PD: Write data
D23: Reverse output of write data
DOUT: Output of write data

Figure 32.11c Timing Diagram for Write Access ($\overline{CE}$ Control)

Note: The last two bus cycle sequences out of the four are described.

■ **Chip erase/sector erase command sequence**

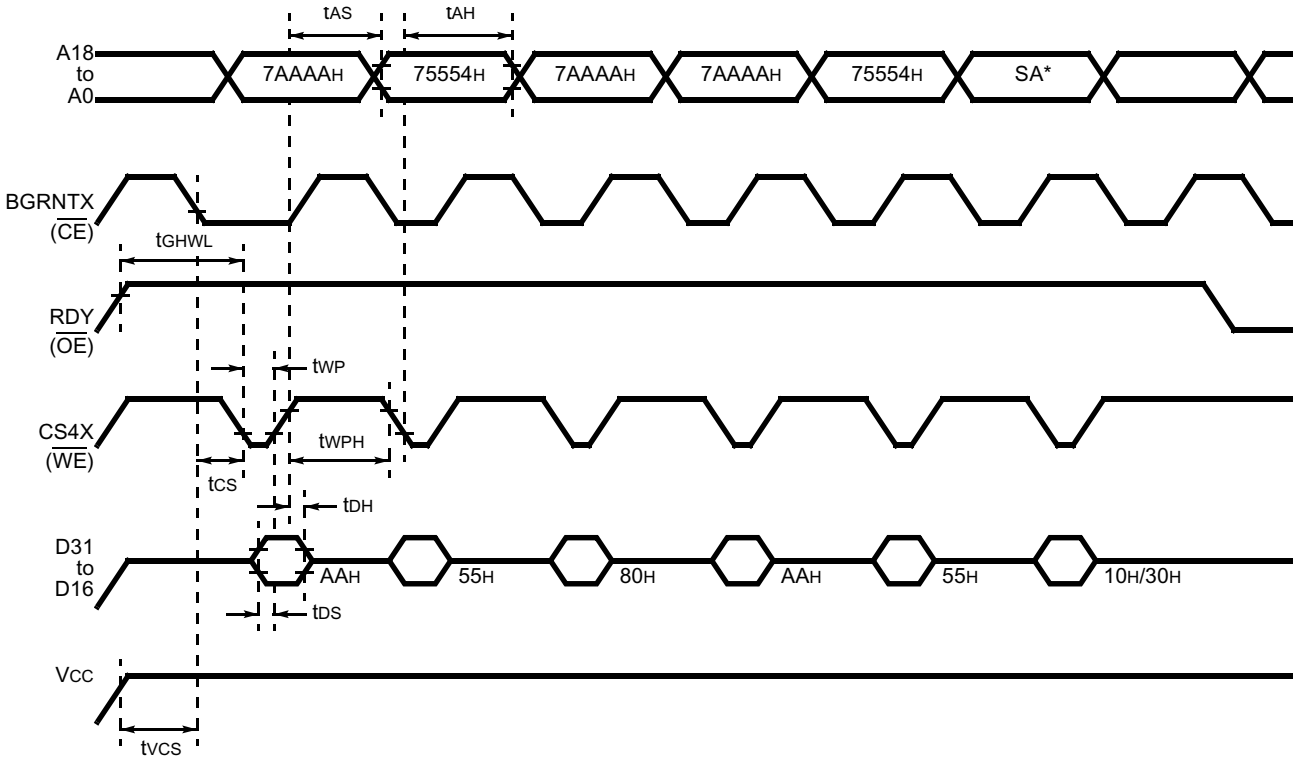


Figure 32.11d Timing Diagram for Write Access (Chip Erasing/Sector Erasing)

Note: **SA** is the sector address at sector erasing. **7AAAAH** (or **6AAAAH**) is the address at chip erasing.

■ **Data polling**

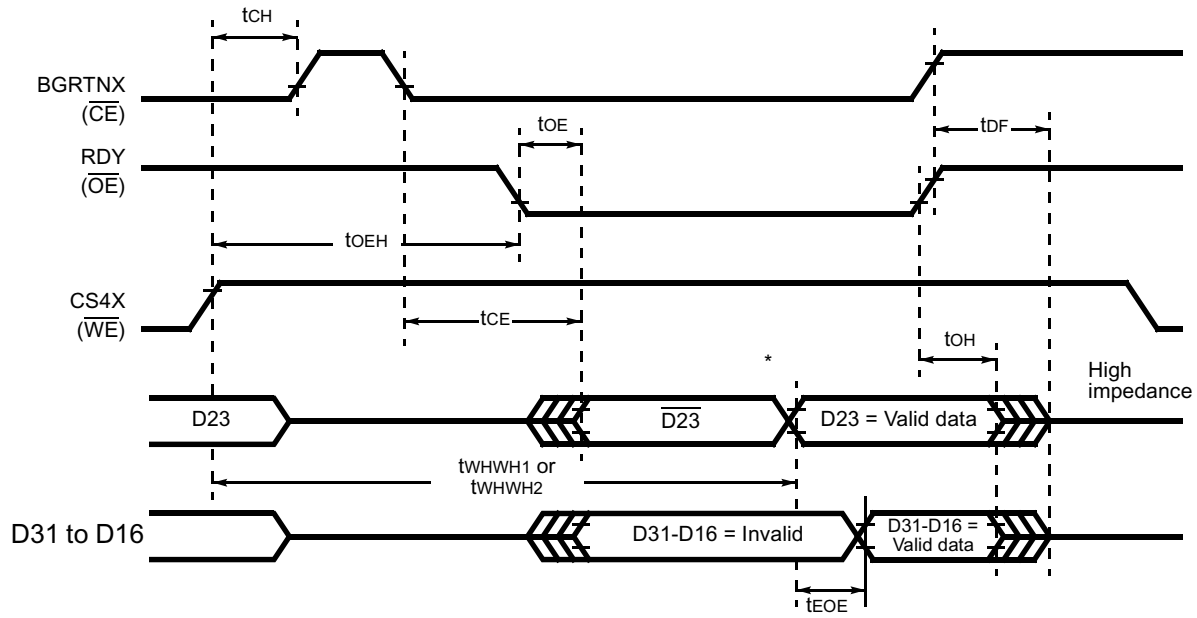


Figure 32.11e Timing Diagram for Data Polling

Note: **DQ7** is valid data (The device terminates automatic operation).

■ **Toggle bit**

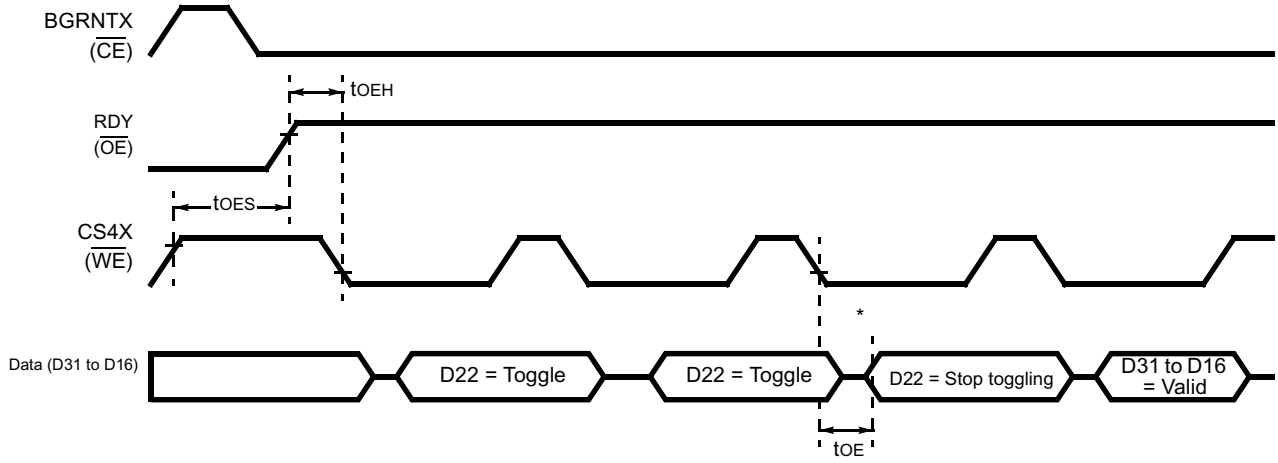


Figure 32.11f Timing Diagram for Toggle Bit

Note: **DQ6** stops toggling (The device terminates automatic operation).

■ **RY/$\overline{BY}$ timing during writing/erasing**

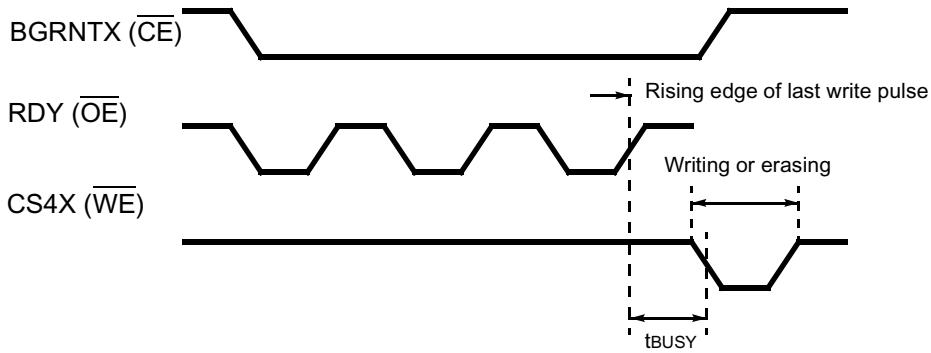


Figure 32.11g Timing Diagram for Output of RY/$\overline{BY}$ Signal during Writing/Erasing

■ **INITX and RY/$\overline{BY}$ timing**

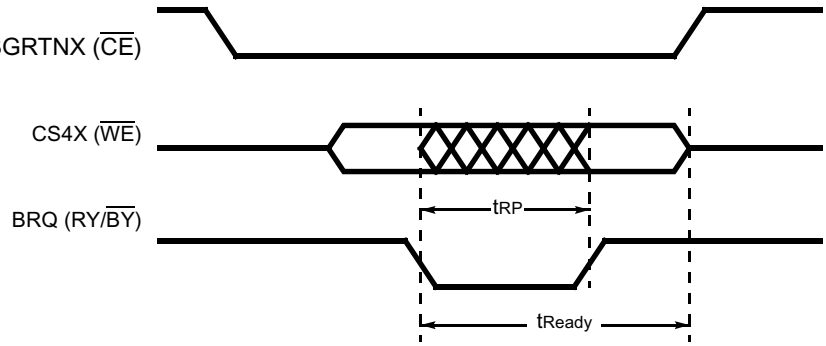


Figure 32.11h Timing Diagram for Output of RY/$\overline{BY}$ Signal at Hardware Reset

■ **Enable sector protect/verify sector protect**

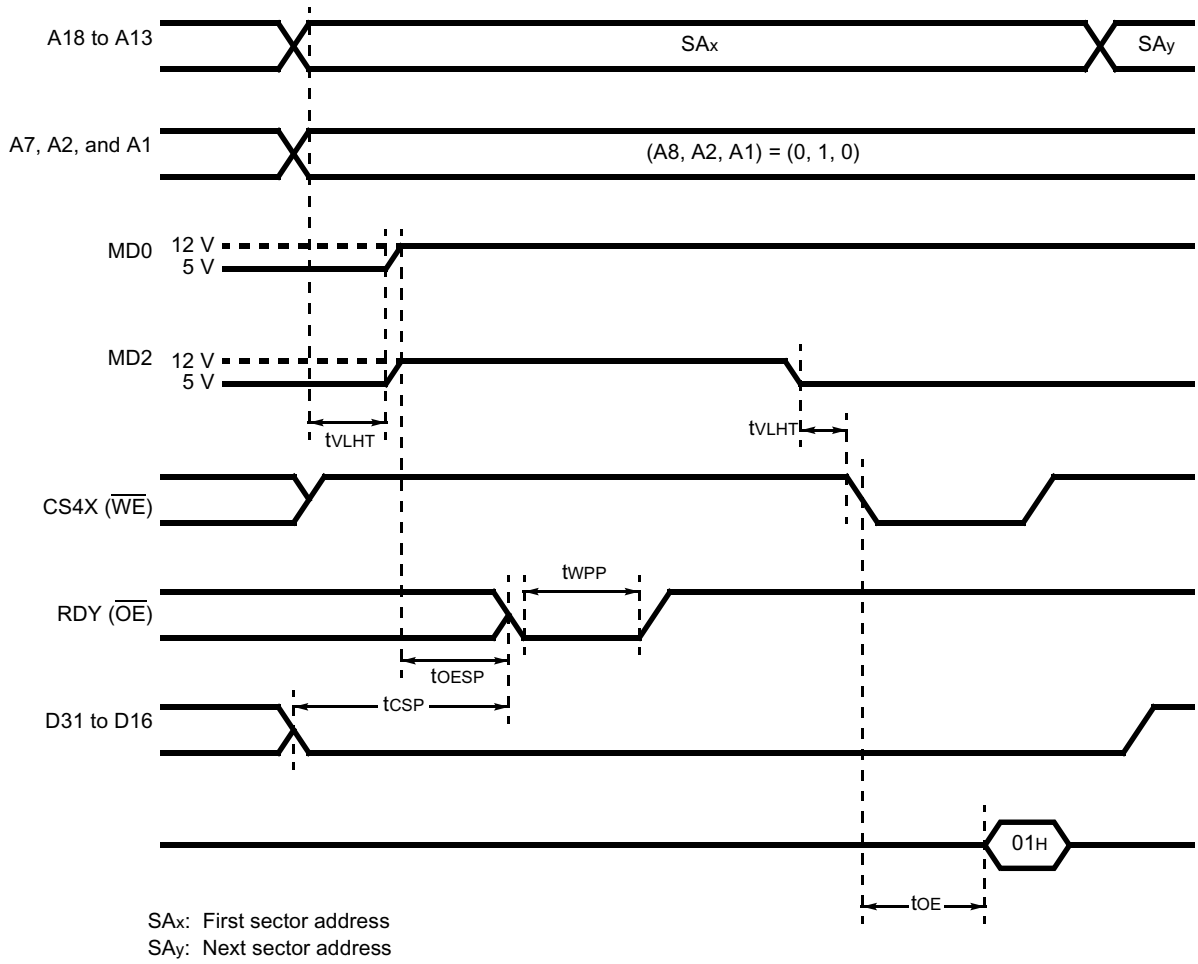


Figure 32.11i Enable Sector Protect/Verify Sector Protect

■ **Temporary sector protect cancellation**

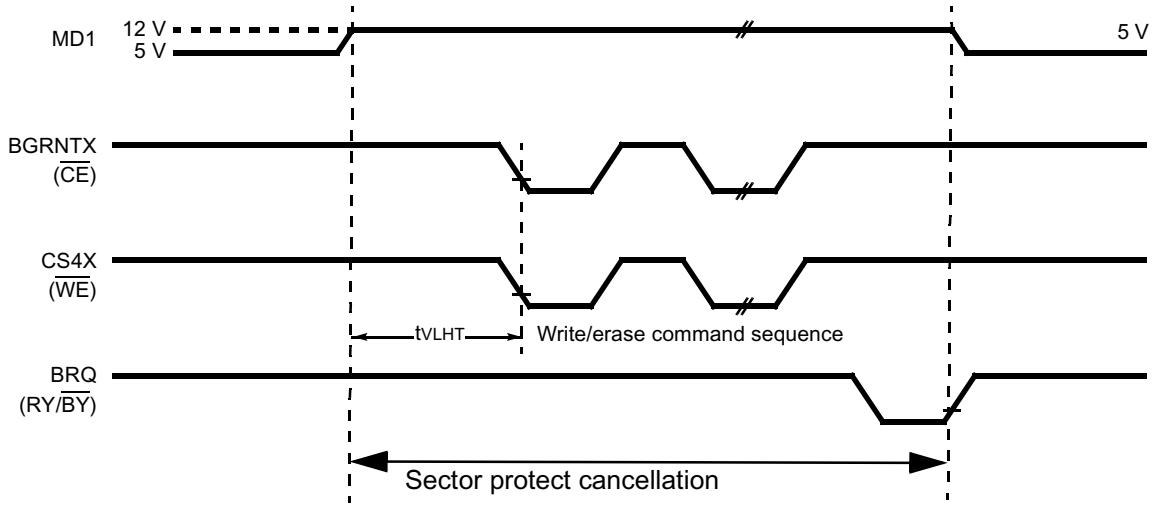


Figure 32.11j Temporary Sector Protect Cancellation

32.12 AC CHARACTERISTICS IN FLASH MEMORY MODE

The AC specifications for the external pins of the MB91360 Series in the Flash Memory mode are shown below. They apply to the case where the user performs read/write access in the Flash Memory mode. They are not needed for access in the normal mode and for use of a flash memory writer.

The values are subject to change without prior notice.

■ Read access

Table 32.12a AC Characteristics for Read Access

(Under recommended conditions)

Parameter	Symbol	Test Conditions	Value			Unit
			Min.	Typ.	Max.	
Read cycle time	t_{RC}	—	130	—	—	ns
Address access time	t_{ACC}	$\overline{CE}$ = VIL $\overline{OE}$ = VIL	—	—	130	ns
$\overline{CE}$ to data output	t_{CE}	$\overline{OE}$ = VIL	—	—	130	ns
$\overline{OE}$ to data output	t_{OE}	—	—	—	50	ns
$\overline{CE}$ to output floating	t_{DF}	—	—	—	30	ns
$\overline{OE}$ to output floating	t_{DF}	—	—	—	30	ns
Previous cycle data output hold time	t_{OH}	—	0	—	—	ns
INITX pin to return to read mode	t_{Ready}	—	—	—	20	µs

Note: sampled, not 100% tested

■ Write [write/erase command] access ($\overline{WE}$ control)

Table 32.12b AC Characteristics for Write Access ($\overline{WE}$ Control)

(Under recommended conditions)

Parameter		Symbol	Value			Unit
			Min.	Typ.	Max.	
Write cycle time		t_{WC}	130	—	—	ns
Address setup time		t_{AS}	0	—	—	ns
Address hold time		t_{AH}	50	—	—	ns
Data setup time		t_{DS}	50	—	—	ns
Data hold time		t_{DH}	0	—	—	ns
Output enable setup time		t_{OES}	0	—	—	ns
Output enable hold time	Read	t_{OEH}	0	—	—	ns
	Toggle and data polling		10	—	—	ns
Read recovery time before write		t_{GHWL}	0	—	—	ns
$\overline{CE}$ setup time		t_{CS}	0	—	—	ns
$\overline{CE}$ hold time		t_{CH}	0	—	—	ns
Write pulse width		t_{WP}	50	—	—	ns
Write pulse width High level		t_{WPH}	20	—	—	ns
Write continuation time		t_{WHWH1}	—	16	—	µs
Sector erase continuation time*1		t_{WHWH2}	—	1.5	30	sec
V_{CC} setup time		t_{VCS}	50	—	—	µs
Voltage transition time*2		t_{VLHL}	4	—	—	µs
Write pulse width*2		t_{WPP}	100	—	—	µs
OE setup time for validating $\overline{WE}$*2		t_{OESP}	4	—	—	µs
CE setup time for validating $\overline{WE}$*2		t_{CSP}	4	—	—	µs
$\overline{INITX}$ pulse width		t_{RP}	500	—	—	ns
RY/$\overline{BY}$ delay until write/erase is enabled		t_{BUSY}	50	—	—	ns

Note:

*1: The internal preprogramming time before erasing is not included.

*2: Applies only to sector protection

Characteristics are sampled, not 100% tested

■ Write [write/erase command] access ($\overline{CE}$ control)

Table 32.12c AC Characteristics for Write Access ($\overline{CE}$ Control)

(Under recommended conditions)

Parameter		Symbol	Value			Unit
			Min.	Typ.	Max.	
Write cycle time		t_{WC}	120	—	—	ns
Address setup time		t_{AS}	0	—	—	ns
Address hold time		t_{AH}	50	—	—	ns
Data setup time		t_{DS}	50	—	—	ns
Data hold time		t_{DH}	0	—	—	ns
Output enable setup time		t_{OES}	0	—	—	ns
Output enable hold time	Read	t_{OEH}	0	—	—	ns
	Toggle and data polling		10	—	—	ns
Read recovery time before write		t_{GHWL}	0	—	—	ns
$\overline{WE}$ setup time		t_{WS}	0	—	—	ns
$\overline{WE}$ hold time		t_{WH}	0	—	—	ns
$\overline{CE}$ pulse width		t_{CP}	50	—	—	ns
$\overline{CE}$ pulse width High level		t_{CPH}	20	—	—	ns
Write continuation time		t_{WHWH1}	—	16	—	µs
Sector erase continuation time*		t_{WHWH2}	—	1.5	30	sec
V_{CC} setup time		t_{VCS}	50	—	—	µs
$\overline{INITX}$ pulse width		t_{RP}	500	—	—	ns
RY/$\overline{BY}$ delay until write/erase is enabled		t_{BUSY}	50	—	—	ns

Note: The internal preprogramming time before erasing is not included.
Characteristics are sampled, not 100% tested

CHAPTER 33 EDSU

The Embedded Debug Support Unit (EDSU) is a module which enables basic in circuit debugging support functions on single chip FLASH MCU devices.

33.1 EDSU OVERVIEW

The Embedded Debug Support Unit (EDSU) is a module which enables basic in circuit debugging support functions on single chip FLASH MCU devices. It is targeted as intermediate solution for customers which did not want to by cost intensive In Circuit Emulation (ICE) systems. ICE systems are based on an Evaluation chip (EVA) coming with a special Debug Support Unit (DSU).

Another reason could be that physical dimensions of the target system make ICE adaption impossible but in system debugging features are required.

Hardware support is only implemented for break detection which causes a non maskable exception. A background debug monitor software can be implemented to handle these exceptions (interrupts) and to configure the EDSU accordingly. EDSU interfaces only to Execution Unit (E-Unit) and D-Bus internally. No additional MCU ports are required, but one UART channel is used during debugging. Program execution trace functionality is not supported.

Advantages are debug possibilities on the target device, the application software can be directly debugged on the 'real' target device without the need to switch on the emulation system. This enables a real in-system debugging, without the disadvantage of mounting a special (expensive) adaptor on the target system PCB to connect to the emulation system.

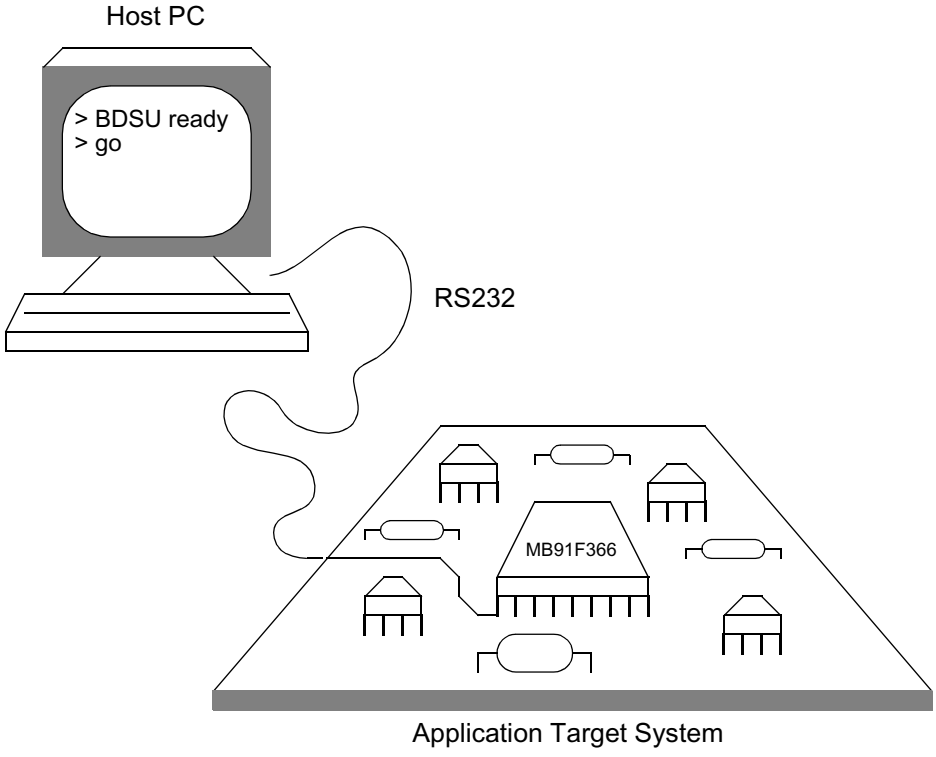
Even devices with new (or custom) macros implemented which are not available on the emulation device can be debugged.

These advantages make this solution very attractive to customers who need fast in-system development/debugging or don't want to use the (expensive) emulation system.

33.1.1 EDSU Application System

A typical application system using the Embedded debug support unit looks like shown in figure 33.1.1.

Figure 33.1.1 EDSU Application System



A host PC with RS232 interface running the background debug monitor software is connected via a serial cable directly with the MCU with Embedded debug support feature (an RS232 to CMOS converter can be implemented outside the target system PCB, e.g. in the serial cable).

33.1.2 Debug Systems

Table 33.1.2 compares the available debug systems.

Table 33.1.2 Comparison with available Debugging Systems

EVA Chip with DSU	EDSU	Software Monitor Debugger
Adapter to ICE system required	Single Chip, real in system debugging[a]	External RAM required
Debug software communicates via ICE, no ressource blocked during debugging	Needs one UART channel for debug software communication	Needs UART for debug software communication. Currently only software for external UART 16550 device available.
No monitor debugger kernel required	Monitor debugger kernel or debug functionality linked to the application required. Disadvantage: reserved FLASH area	Complicated boot procedure to start monitor debugger kernel. Disadvantage: reserved FLASH area, program upload to external RAM
Code could be located in and executed from FLASH	Code could be located in and executed from FLASH	Code located in external RAM which is mapped to FLASH area. FLASH disabled during debugging.
Real time debugging with hardware breakpoints	Real time debugging with hardware breakpoints	No real time debugging, software breakpoints
8 Instruction breakpoints 2 Operand breakpoints Mask option Data value break option Tracing of instructions/data	4 Instruction breakpoints 2 Operand breakpoints Range and mask options Data value break option	Instruction breakpoints limited by software only

Note: a) The debugging environment is exactly the same as for the target application. No additional adapters and auxiliary circuitry required (e.g. important for EMI measurements).

33.1.3 EDSU Main Functions

Embedded Debug Support Unit (EDSU) for MB91360 series

■ **Main functions:**

- 4 Instruction Address Breakpoints
- 2 Instruction Address Breakpoint Masks
- Instruction Breakpoint Address Range Function

- 2 Operand Address Breakpoints (programmable on datasize and access type)
- 1 Operand Address Breakpoint Mask
- Operand Breakpoint Address Range Function

- 2 Operand Data Value Breakpoints
- 1 Operand Data Value Breakpoint Mask

- Break Trigger programmable on resource interrupts

33.1.4 More Information About EDSU

Please refer to the documentation about the Background Debugging Software.

CHAPTER 34 ELECTRICAL SPECIFICATION

This Chapter provides information on maximum ratings, operating conditions and AC specifications.

34.1 ABSOLUTE MAXIMUM RATINGS

WARNING: Semiconductor devices can be permanently damaged by application of stress (voltage, current, temperature, etc.) in excess of absolute maximum ratings. Do not exceed these ratings.

Table 34.1a Absolute maximum ratings

Parameter	Symbol	min.	max.	Unit	Condition
Digital supply voltage	VDD-VSS	-0.3	6.0	V	
Stepper motor control supply voltage	HVDD-HVSS	-0.3	6.5	V	
Storage temperature	Tstg	-55	125	°C	
Power consumption	P_{TOT}	--	1)	W	T_A = +25°C
Digital input voltage	V_{IDIG}	-0.3 2)	5.8	V	VSS=0V, VDD=5V
Analog input voltage	V_{IA}	-0.3	5.8	V	AVSS=0V, AVCC=5V
Analog supply voltage	AVCC-AVSS	-0.3	5.8	V	AVSS=0V
Analog reference voltage	$V_{REFH/L}$ - VSSA	-0.3	5.8	V	AVSS=0V
Static DC current into digital I/O	$I_{I/ODC}$	-2	2	mA	$\Sigma I_{I/ODC} < I_{SRUN}$

1) dependent on family member

2) making full use of the allowed static DC correct into digital I/Os will lead to lower values for V_{IDIG} min.

Table 34.1b Maximum power consumption

Device	Maximum Power Consuption
MB91FV360GA	2.5 W
MB91F362GB	2.5 W
MB91F364G	1.2 W
MB91F365GB	1.3 W
MB91F366GB	1.3 W
MB91F367GB	1.3 W
MB91F368GB	1.3 W
MB91366GA	1.3 W
MB91F376G	1.2 W
MB91F369GA	2.5 W

34.2 RECOMMENDED OPERATING CONDITIONS

WARNING: The recommended operating conditions are required in order to ensure the normal operation of the semiconductor device. All of the device's electrical characteristics are warranted when the device is operated within these ranges.

Always use semiconductor devices within their recommended operating condition ranges. Operation outside these ranges may adversely affect reliability and could result in device failure.

No warranty is made with respect to uses, operating conditions, or combinations not represented on the data sheet. Users considering application outside the listed conditions are advised to contact their FUJITSU representatives beforehand.

Table 34.2a Recommended operating conditions

Parameter	Symbol	min.	typ.	max.	Unit	Condition
Operating temperature	Ta	-40		85	°C	
Supply voltage						
- Digital supply	VDD-VSS	4.25 [1)]	5	5.25	V	Internal voltage reg. VDD_{CORE}=3.3V
- Stepper motor control supply	HVDD-HVSS	4.75	5	5.25	V	HVSS=0V
- Analog supply	AVCC-AVSS	4.9	5	5.1	V	AVSS=0V
RAM data retention voltage	VDD-VSS	3.0			V	

[1)] this is only valid if the integrated power-down reset circuit is switched-off, else a reset can be triggered at voltages less or equal than 4.5 V (see spec items for power-down reset)

34.3 OPERATING CONDITIONS

Table 34.3a Operating Conditions

Parameter			Symbol	Value			Unit	Condition
				Min.	Typ.	Max.		
Current consumption	Run mode		I_{srun}	--	--	*1	mA	T_A = 25°C
	RTC mode		I_{sRTC}	--	0.5 --	1.25 500	mA µA	f_{clk} = 4 MHz at T_A = 25°C f_{clk} = 32 kHz at T_A = 25°C
	Stop mode		I_{sstop}	--	10	200	µA	f_{clk} = 0 at T_A = 25°C
Stepper motor control	H-port output voltage		V_{OHH}	HV_{DD}-500	--	HV_{DD}-125	mV	I_{ol} = ±30 mA, T_C = 25°C
			V_{OHL}	HV_{SS}+125	--	HV_{SS}+500	mV	I_{ol} = ±30 mA, T_C = 25°C
			V_{OHH}	HV_{DD}-500	--	HV_{DD}-125	mV	I_{ol} = ±27 mA, T_C = 85°C
			V_{OHL}	HV_{SS}+125	--	HV_{SS}+500	mV	I_{ol} = ±27 mA, T_C = 85°C
			V_{OHH}	HV_{DD}-500	--	HV_{DD}-125	mV	I_{ol} = ±30 mA, T_C = –40°C
			V_{OHL}	HV_{SS}+125	--	HV_{SS}+500	mV	I_{ol} = ±30 mA, T_C = –40°C
	SMC comparator threshold voltage		V_{THcomp}	HV_{DD}/9 -70	HV_{DD}/9	HV_{DD}/9 +70	mV	
	Slew rate		--	--	40	--	ns	Cload = 0 pF
Alarm comparator	Threshold voltage	Over-voltage	V_{TAH}	4/5 V_{DDA}-5%	4/5 V_{DDA}	4/5 V_{DDA}+5%	V	(external 4 : 1 divider)
		Under-voltage	V_{TAL}	2/5 V_{DDA}-5%	2/5 V_{DDA}	2/5 V_{DDA}+5%	V	
	Switching hysteresis		V_{TAHYS}	12.5	25	50	mV	
	Alarm sense time		t_{AS}	--	--	10	µs	
	Input resistance		R_{in}	5	--	--	MΩ	at V_{TAH}, V_{TAL}
Power down Reset	Threshold voltage		V_{TPOR}	3.5	4.0	4.5	V	
	Switching hysteresis		$V_{TPO-RHYS}$	20	50	80	mV	
	Reset sense time		t_{RS}	--	--	10	µs	
Digital outputs	Output “H” voltage		V_{OH}	V_{DD}-0.5	--	V_{DD}	V	I_{load} = 4mA
	Output “L” voltage		V_{OL}	V_{SS}	--	V_{SS}+0.4	V	I_{load} = –4mA

Table 34.3a Operating Conditions (Continued)

Parameter			Symbol	Value			Unit	Condition
				Min.	Typ.	Max.		
Digital Inputs*2	CMOS (Types : Q, S, Y, T)	High voltage range	V_{IH}	0.65 × V_{DD}	--	V_{DD}	V	
		Low voltage range	V_{IL}	V_{SS}	--	0.25 × V_{DD}	V	
	CMOS Schmitt-Trigger (Types : E, F, U)	High voltage range	V_{IH}	0.8 × V_{DD}	--	V_{DD}	V	
		Low voltage range	V_{IL}	V_{SS}	--	0.2 × V_{DD}	V	
	CMOS Automotive Schmitt-Trigger (Types : A, B, K1, M1, J)	High voltage range	V_{IH}	0.8 × V_{DD}	--	V_{DD}	V	
		Low voltage range	V_{IL}	V_{SS}	--	0.5 × V_{DD} 0.6 × V_{DD}	V V	V_{DD}min = 4.25 V V_{DD}min = 4.75 V
		hysteresis voltage	--	--	0.5	--	V	
	CMOS 3/5 V (Type : L, N, O)	High voltage range	V_{IH}	0.65 × V_{DD}	--	V_{DD}	V	
		Low voltage range	V_{IL}	V_{SS}	--	0.25 × V_{DD}	V	
	CMOS 3 V (Type : P, W)	High voltage range	V_{IH}	0.65 × V_{DD}	--	V_{DD}	V	
		Low voltage range	V_{IL}	V_{SS}	--	0.25 × V_{DD}	V	
	Input capacitance		C_{IN}	--	--	16	pF	
	Input leakage current		I_{IL}	−1	--	+1	µA	T_A = 25°C
	Pull up resistor		R_{up1} R_{up2}	--	50 10	--	kΩ kΩ	Types : E, U Type : S
ADC inputs	Reference voltage input		V_{REFH} V_{REFL}	V_{REFL}+3 V_{SSA}	--	V_{DDA} V_{REFH}-3	V V	
	Input voltage range		V_{imax} V_{imin}	V_{REFL} --	--	V_{REFH} --	V V	
	Input resistance		R_I	--	--	3.6	kΩ	
	Input capacitance		C_I	--	--	30	pF	
	Input leakage current		I_{IL}	−1	--	1	µA	T_A = 25°C
	Impedance of external output driving the ADC input		--	--	--	4.0	kΩ	at sampling time of 1.6 µs
DAC analog outputs	Output voltage		V_{out}	V_{SSA}	--	V_{DDA}	V	
	Output impedance		R_{out}	2.0	2.9	4.0	kΩ	external voltage follower required
	Output capacitance		C_{out}	--	--	20	pF	

Table 34.3a Operating Conditions (Continued)

Parameter		Symbol	Value			Unit	Condition
			Min.	Typ.	Max.		
Sound generator	Output voltage	$V_{outHIGH}$ V_{outLOW}	V_{DD}-0.5 V_{SS}	--	V_{DD} V_{SS}+0.4	V V	
	Output current	I_{out}	4	--	--	mA	
PPG	Output voltage	$V_{outHIGH}$ V_{outLOW}	V_{DD}-0.5 V_{SS}	--	V_{DD} V_{SS}+0.4	V V	
	Output current	I_{out}	4	--	--	mA	
LED	Output voltage	$V_{outHIGH}$ V_{outLOW}	V_{DD}-0.8 --	--	-- V_{SS}+0.8	V V	$I_{outHIGH}$ = 14 mA I_{outLOW} = 24 mA
I^2C Bus Interface (Open Drain Output)	Output voltage	$V_{outHIGH}$ V_{outLOW}	-- V_{SS}	--	V_{DD} V_{SS}+0.4	V V	
	Output current	I_{out}	3	--	--	mA	I_{outLOW} = 3 mA
Lock-up time PLL1 (4 MHz →16 MHz to 64 MHz)		--	--	0.1	1	ms	
ESD Protection (Human body model MIL883-B compliant)		V_{surge}	2	--	--	kV	$R_{discharge}$ = 1.5 kΩ $C_{discharge}$ = 100 pF

*1) See 34.4 "RUN MODE CURRENT / POWER CONSUMPTION" on page 731.

*2) Valid for bidirectional tristate I/O PAD cell

34.4 RUN MODE CURRENT / POWER CONSUMPTION

The power dissipation during normal operation is determined by the total power dissipation of the internal logic P_C, the dissipation from analog modules P_A and the power dissipation P_{IO} of the I/O buffers. Among the I/O buffers the dissipation caused by the stepper motor drivers P_{SMC} should be taken into special consideration.

So the overall power consumption P_D will be calculated as a sum of $P_C + P_A + P_{SMC} + P_{IO}$.

34.4.1 Logic Power Consumption

The following formul can be used to calculate the maximum core current consumption when the PLL is used depending on the frequency settings for the internal clocks:

I_{CC} = B * CLKB[MHz] + P * CLKP[MHz] + T * CLKT[MHz] + 35.5 mA

The factors B, P and T depend of the device, see table 34.4.1a.

If clock modulation is used the following value must be added to this result:

0.24[mA/MHz] * CLKB[MHz].

Table 34.4.1a Current consumption factors

Device	B [mA/MHz]	P [mA/MHz]	T [mA/MHz]	Remarks
MB91FV360GA MB91F362GB	3.45	2.52	0.72	
MB91F364G	1.25	1.70	0.40	
MB91F365GB MB91F366GB MB91F367GB MB91F368GB	2.30	2.70	0.50	
MB91366GA	2.30	2.70	0.50	
MB91F376G	1.25	1.70	0.40	
MB91F369GA	2.30	2.70	0.50	

For example, with MB91F362GB this results in the following values:

Table 34.4.1 Logic Power Consumption (Example: MB91F362GB)

Clock frequencies [MHz]			Maximum Core Current Consumption [mA]	Logic Power Consumption P_C at 5.25 V [W]	Remarks
CLKB	CLKP	CLKT			
64	16	16	308	1.62	no clock modulation possible
48	24	24	290	1.52	
48	16	16	264	1.39	
32	32	32	257	1.35	

Table 34.4.1 Logic Power Consumption (Example: MB91F362GB) (Continued)

Clock frequencies [MHz]			Maximum Core Current Consumption [mA]	Logic Power Consumption Pc at 5.25 V [W]	Remarks
32	16	16	205	1.08	
24	24	24	202	1.06	
24	12	12	163	0.86	
16	16	16	146	0.77	
2	2	2	40	0.21	no PLL, no clock modulation
0.125	0.125	0.125	30	0.16	no PLL, no clock modulation

In addition to this power consumption of the MCU core logic the following contributions to the overall power consumption have to be considered:

- Analog power consumtion
- I/O and SMC power consumption

See the following sections.

34.4.2 Analog Power Consumption

Table 34.4.2 Analog Power Consumption

Module	Typical Current Consumption	Maximum Current Consumption	Remark
DAC		1 mA / channel	current at AVCC
ADC	3 mA	7 mA	current at AVCC
	1.6 mA	2.6 mA	current at AVRH
Power down reset	0.26 mA	0.5 mA	current at VDD
Alarm Comparator	0.31 mA	0.5 mA	current at AVCC
Zero point detection	0.13 mA	0.25 mA	current at AVCC

To calculate the analog power consumption P_A, the current contributions of the active modules have to be multiplied by the maximum analog supply voltage of 5.1 V - or by the maximum digital supply voltage as in case of the Power down reset.

34.4.3 I/O and SMC Power Consumption

■ **SMC drivers**

The average current consumption per SMC channel is 38.2 mA, for four channels this results in 152.8 mA.

At 2 * 0.5 V this results in 153 mW power consumption P_{SMC} for four channels of stepper motor drivers.

■ **Other I/O Buffers**

The power dissipation (P_{IO}) (at 5.25 V) of the I/O buffers is represented as the sum of the dynamic power dissipation (P_{AC5V}, P_{AC3V}) and the static power consumption (P_{DC}).

$P_{IO} = P_{AC5V} * 1.1 + P_{AC3V} * 1.2 + P_{DC}$

The following table lists values for the calculation of P_{AC5V} and P_{AC3V}:

Table 34.4.3 Calculation of P_{AC5V} and P_{AC3V}

Buffer Type	Power Consumption P_{IB} / P_{OB} @ 5V to calculate P_{AC5V}	Power Consumption P_{IB} / P_{OB} @ 3.3V to calculate P_{AC3V}	Unit
Normal Input	12.4	12.4	µW/MHz (C_L in pF)
Bidirectional Input			
4 mA Bidirectional Output	194 + 25 C_L	85.5 + 11 C_L	
4 mA Output			
8 mA Bidirectional Output	353 + 25 C_L	154 + 11 C_L	
8 mA Output			

P_{AC} = P_{IB} * In * f * operating rate + P_{OB} * On * f * operating rate

P_{IB}: Power Consumption of Input Buffers and Bidirectional Inputs

P_{OB}: Power Consumption of Output Buffers and Bidirectional Outputs

In: Total number of input buffers and bidirectional buffer inputs

On: Total number of output buffers and bidirectional buffer outputs

f: System frequency

Operating rate: 1.0 if all buffers are switched simultaneously at system frequency

P_{DC} is the caused by off chip loads which are drawing static currents.

P_{DC} = VO * IO * DC_N

VO: Output voltage drop - usually 0.4 V

IO: Output current - usually 4 mA

DC_N: Number of output buffers and bidirectional buffers driving off chip loads causing static currents.

34.4.4 Packages and Maximal Allowed Power Consumption

The maximal allowed power consumption depends on the ambient temperature and the thermal resistance of the package.

Table 34.4 Package thermal resistance and max. allowed power consumption

Package	Thermal Resistance [°C/W]				Maximum allowed power consumption [W] (see Note)
	ϑja (junction to ambient)			ϑjc (junction to case)	
	0 m/s	1 m/s	3 m/s		
FPT-120P-M21	30	27	25	5	1.33
FPT-160P-M15	16	13	11	2.5	2.5
FPT-208P-M04	16	13	11	2.5	2.5
PGA-401C-A02	16	8.5	5.5	--	2.5

Note: The maximum allowed ambient temperature is 85°C, the maximum allowed junction temperature is 125°C. Under these conditions, the allowed maximum power consumption will be

P_{MAX} = (125°C - 85°C) / ϑja (K/W)

ϑja is the thermal resistance of this package at 0 m/s when used on a muli-layer board with separate power and ground planes.

34.5 CLOCK SETTINGS

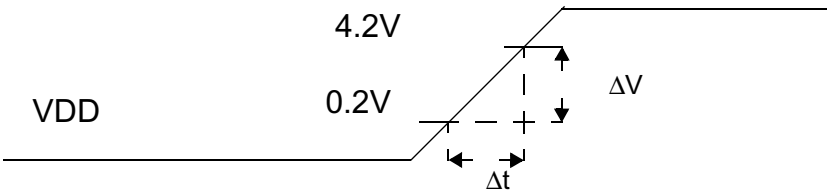
Table 34.5 Clock Settings

Clock domain	Clock name	Max. frequency setting	Remark
Core	CLKB	64 MHz	under normal operating conditions
		32 MHz	for supply voltage between 4.25 and 3.5 V
Resource bus	CLKP	32 MHz	
Ext. Bus	CLKT	32 MHz	
Clock for CAN	CANCLK	32 MHz	

34.6 THE TIME FOR POWER SUPPLY

Table 34.6 Power On time

Parameter	Symbol	min	typ.	max.	unit
Power supply raising slope	ΔV /Δt	-----	-----	0.05	V/μs
Power supply raising slope	t_R	80	-----	-----	μs



34.7 CONVERTER CHARACTERISTICS

34.7.1 A/D Converter Characteristics

Table 34.7.1 A/D Converter charactersitics

Parameter	Symbol	Rating			Unit	Remark
		Minimum	Typical	Maximum		
Resolution	--	--	--	10	Bit	
Conversion error	--	--	--	+/- 5.0	LSB	overall error
Non-linearity	--	--	--	+/-2.5	LSB	
Differantial Non-linearity	--	--	--	+/-1.9	LSB	
Zero Reading voltage	V_{0T}	AVRL -3.5	AVRL+0.5	AVRL+4.5	LSB	
Full scale reading voltage	V_{FST}	AVRH-5.5	AVRH-1.5	AVRH+2.5	LSB	
Input current	IA@AVCC	--	3.0	7.0	mA	
Reference voltage current	IR	--	1.6	2.6	mA	
Conversion time	--	178 cycles CLKP	--	1ms		
Ripple of supply voltage	--	--	--	+/- 5.0	mV	

34.7.2 A/D Converter Glossary

■ **Resolution**

The smallest change in analog voltage detected by A/D converter.

■ **Linearity error**

A deviation of actual conversion characteristic from a line connecting the zero-traction point (between “00 0000 0000” ↔ “00 0000 0001”) to the full-scale transition point (between “11 1111 1110” ↔ “11 1111 1111”) .

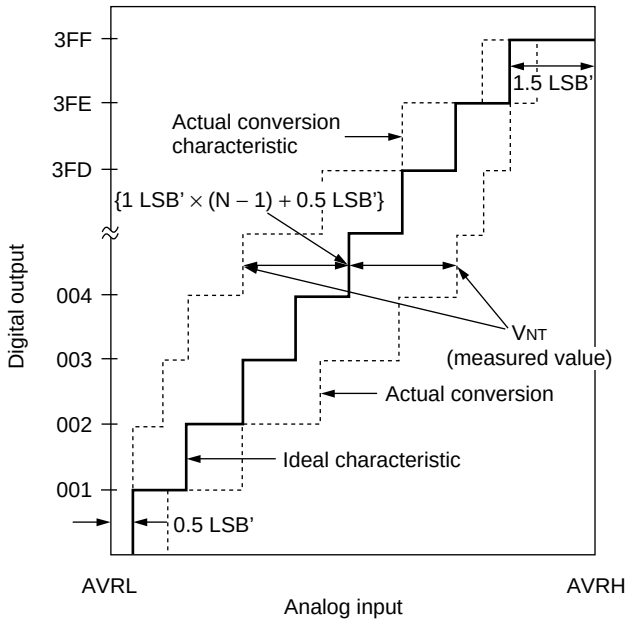
■ **Differential linearity error**

A deviation of a step voltage for changing the LSB of output code from ideal input voltage.

■ **Total error**

A difference between actual value and theoretical value. The overall error includes zero-transition error, full-scale transition error and linearity error.

Total error

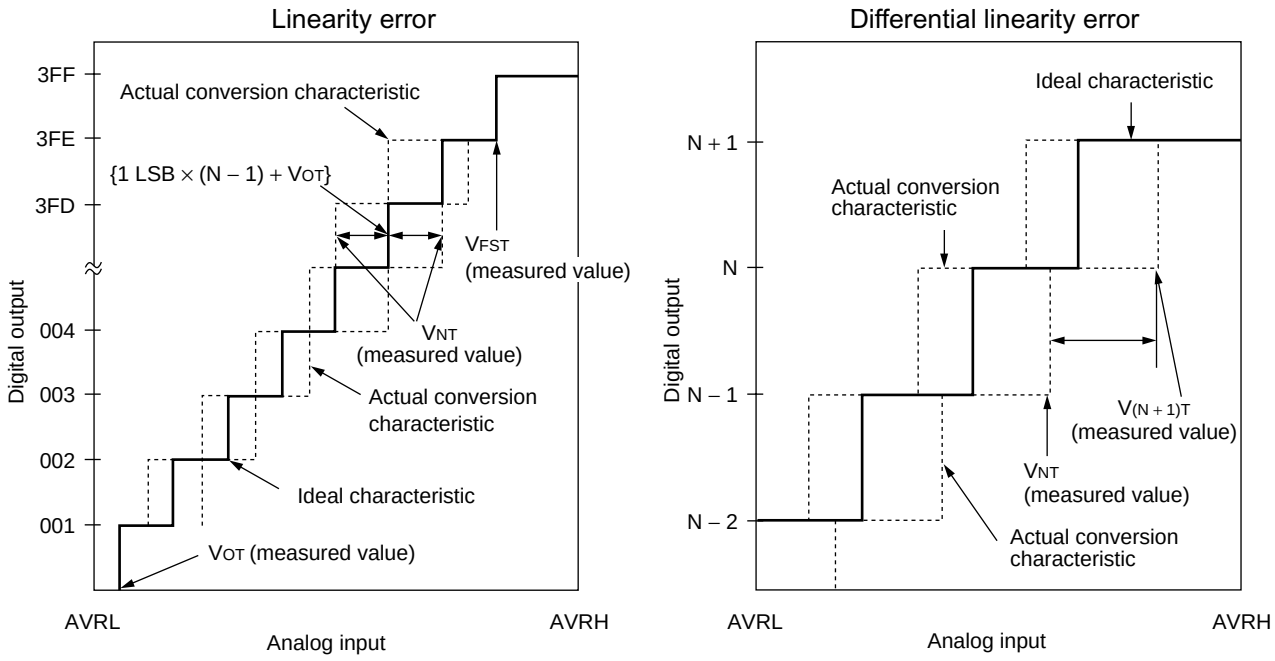


$$\text{Total error of digital output N} = \frac{V_{NT}\text{-}\{1\ LSB' \times (N\text{-}1) + 0.5\ LSB'\}}{1\ LSB'}\ [LSB]$$

V_{OT} ' (Ideal value) = AVRL+0.5 LSB' [V]

V_{FST} ' (Ideal value) = AVRH-1.5 LSB' [V]

V_{NT} : A voltage for causing transition of digital output from (N- 1) to N



$$\text{Linearity error of digital output N} = \frac{V_{NT}-\{1\ \text{LSB} \times (N-1) + V_{OT}\}}{1\ \text{LSB}}\ [\text{LSB}]$$

$$\text{Differential linearity error of digital output N} = \frac{V_{(N+1)T}-V_{NT}}{1\ \text{LSB}} - 1\ [\text{LSB}]$$

$$1\ \text{LSB} = \frac{V_{FST}-V_{OT}}{1022}\ [\text{V}]$$

$$1\ \text{LSB'}\ (\text{ideal value}) = \frac{\text{AVRH}-\text{AVRL}}{1022}\ [\text{V}]$$

V_{OT} : A voltage for causing transition of digital output from (000) H to (001) H

V_{FST} : A voltage for causing transition of digital output from (3FE) H to (3FF) H

V_{NT} : A voltage for causing transition of digital output from (N-1) H to N

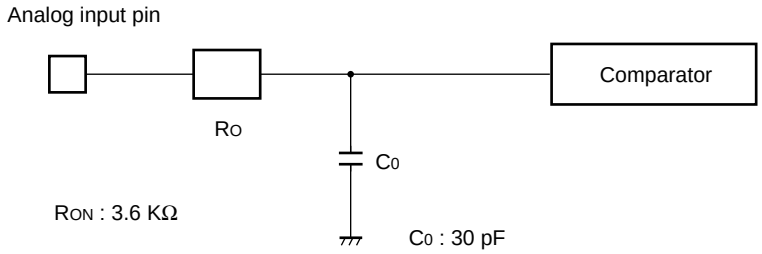
■ Error

As the absolute value of AVRH decreases, relative error increases.

34.7.3 Notes on Using A/D Converter

Output impedance of external circuit of analog input under following conditions;
Output impedance of external circuit < 4 kΩ.
If output impedance of external circuit is too high, analog voltage sampling time may be too short for accurate sampling.

- **Analog input Equivalent Circuit**



34.7.4 D/A Converter

Table 34.7.4 D/A Converter characteristics

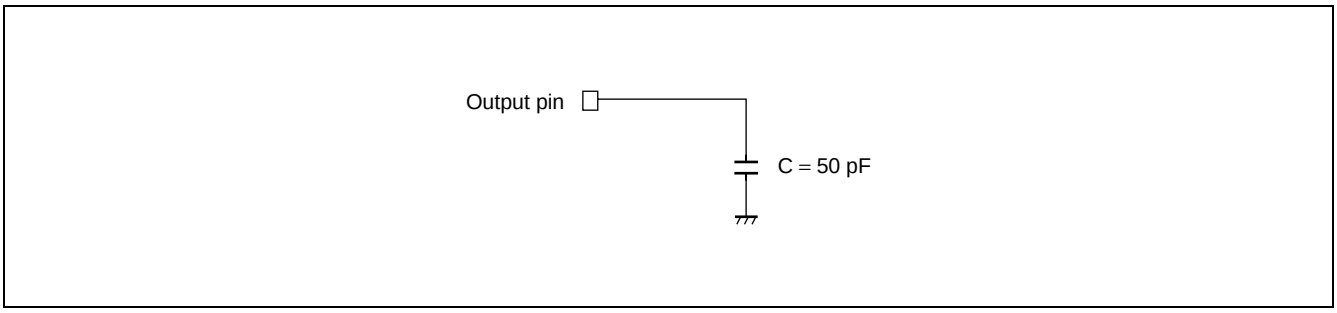
Parameter	Symbol	Rating			Unit	Remark
		Minimum	Typical	Maximum		
Resolution	--	--	--	10	Bit	
Differential linearity error	--	-0.9	--	+0.9	LSB	
Conversion time	--	--	3	--	µs	100pF external load

34.8 AC CHARACTERISTICS

34.8.1 Measurement conditions

Parameter	Symbol	Value	Unit	Conditions
“H” level input voltage	V_{IH}	according to I/O spec	V	V_{DD} = 4.25 to 5.25 V, T_A = –40 to +85°C
“L” level input voltage	V_{IL}		V	
“H” level output voltage	V_{OH}	0.5 × V_{DD}	V	
“L” level output voltage	V_{OL}	0.5 × V_{DD}	V	
“H” level input voltage	V_{IH}	3.0	V	V_{DD} = 3.0 to 3.6 V, T_A = –40 to +85°C
“L” level input voltage	V_{IL}	0	V	
“H” level output voltage	V_{OH}	0.5 × V_{DD}	V	
“L” level output voltage	V_{OL}	0.5 × V_{DD}	V	

■ **Load conditions**

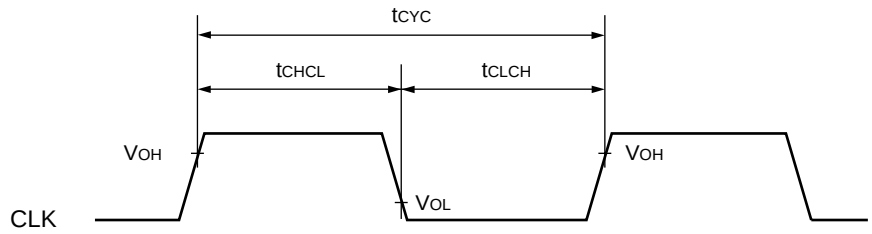


34.8.2 External bus clock

(V_{DD} = 4.25 V to 5.25 V, T_A = –40°C to +85°C)

Signal	Symbol	Pin name	Value		Unit
			Min.	Max.	
CLK cycle	t_{CYC}	CLK	t_{CPT}	--	ns
CLK rise → CLK fall	t_{CHCL}	CLK	t_{CYC}/2-10	t_{CYC}/2+10	ns
CLK fall → CLK rise	t_{CLCH}	CLK	t_{CYC}/2-10	t_{CYC}/2+10	ns

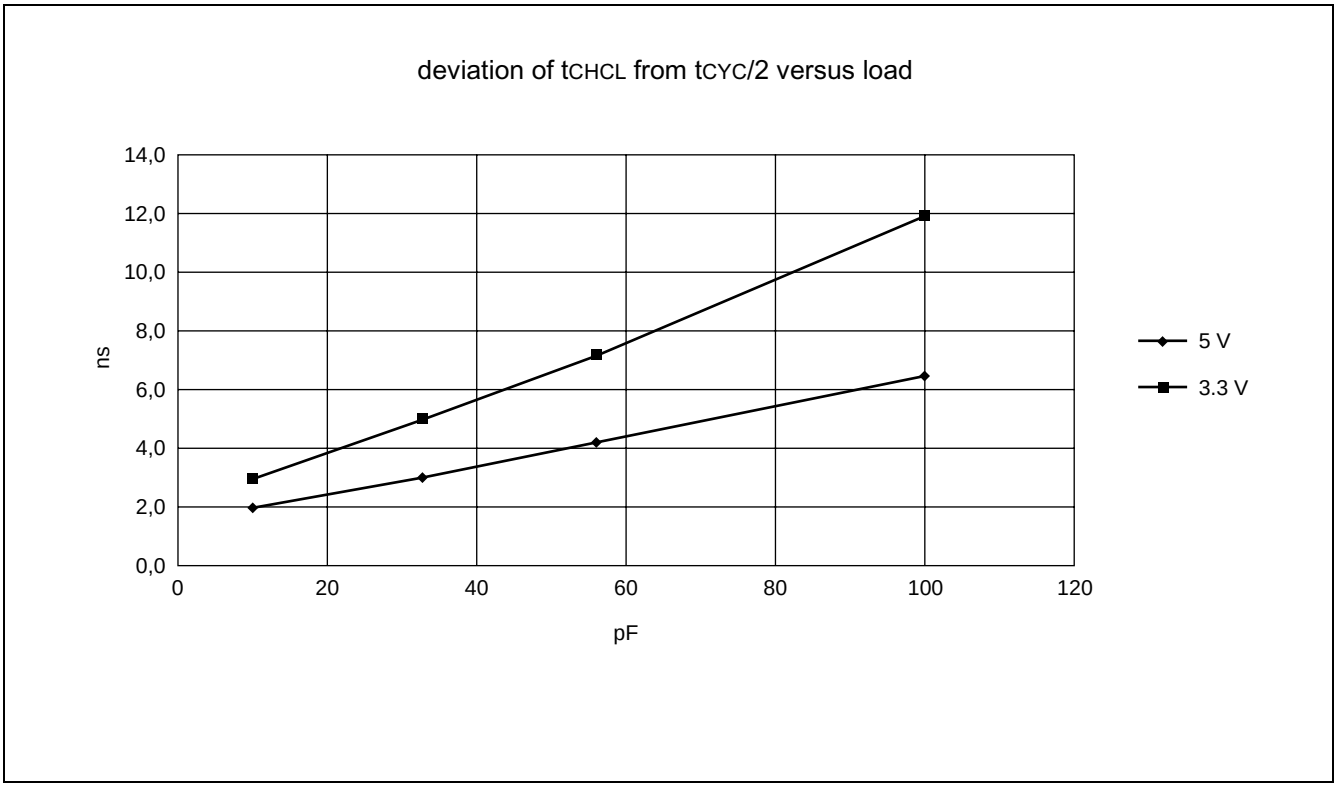
Note: This is only valid for operation without clock modulator



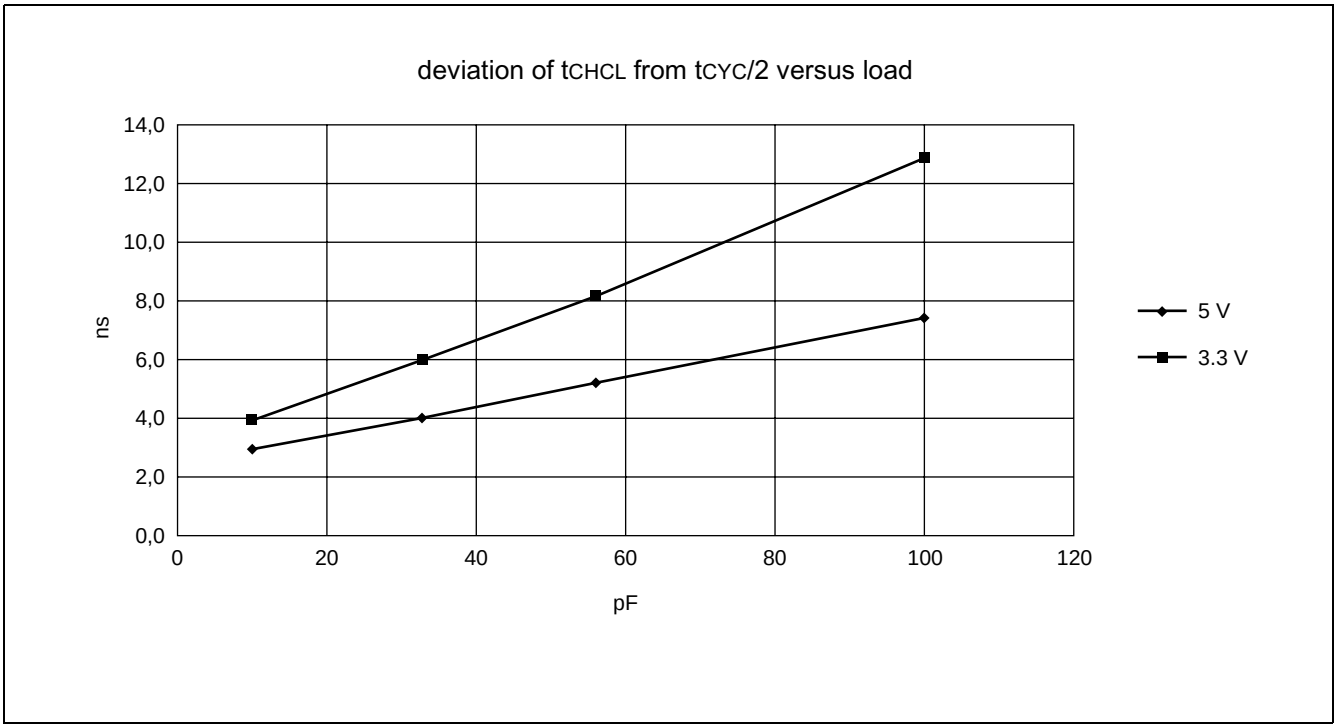
The values for t_{CHCL} and t_{CLCH} are heavily dependent on the load connected to the CLK pin. The following diagrams show this dependency for the worst case situation. The first diagram shows the situation for even division ratios between CLKB and CLKT, the second diagram shows this for odd division ratios between CLKB and CLKT (ASYMCLKT bit is not set) .

It has to note that when the combination of CLK frequency and load at CLK pin is such that rise or fall times are longer than t_{CYC}/2 the duty ratio can get worse.

■ **Even CLKB/CLKT division ratios :**



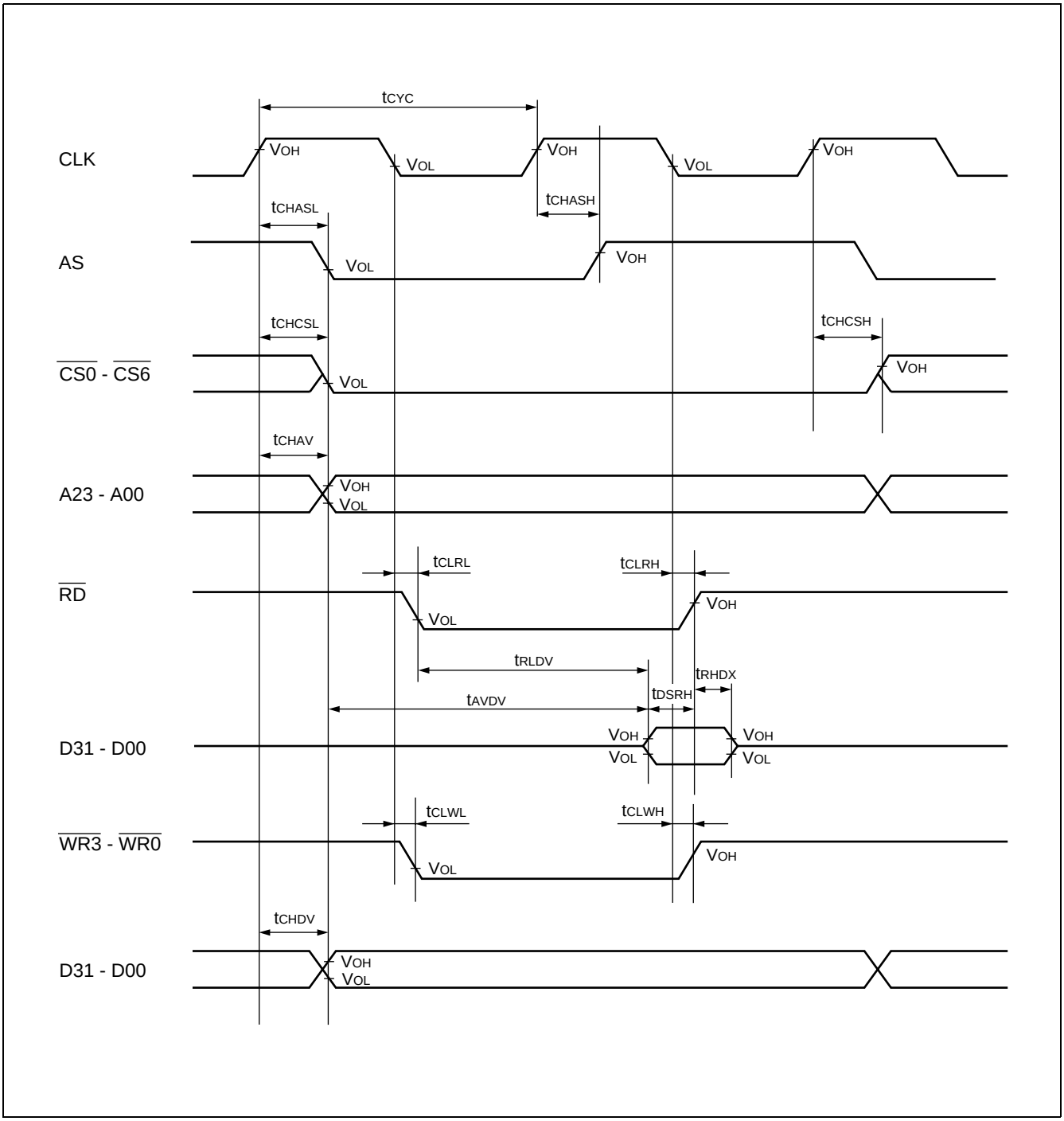
■ **Odd CLKB/CLKT division ratios :**



34.8.3 External bus interface

(V_{DD} = 4.25 V to 5.25 V, T_A = –40°C to +85°C)

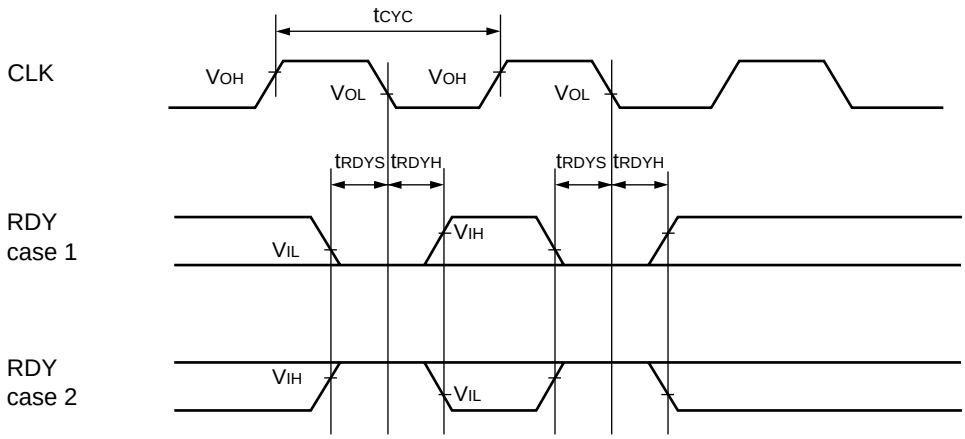
Signal	Symbol	Pin name	Value		Unit
			Min.	Max.	
$\overline{CS6}$ to $\overline{CS0}$ delay time	t_{CHCSL}	CLK $\overline{CS6}$ to $\overline{CS0}$	--	15	ns
$\overline{CS6}$ to $\overline{CS0}$ delay time	t_{CHCSH}		--	15	ns
Address delay time	t_{CHAV}	CLK A20 to A0	--	20	ns
Data delay time	t_{CHDV}	CLK D31 to D0	--	16	ns
$\overline{RD}$ delay time	t_{CLRL}	CLK $\overline{RD}$	--	15	ns
$\overline{RD}$ delay time	t_{CLRH}		--	15	ns
$\overline{WR3}$ to $\overline{WR0}$ delay time	t_{CLWL}	CLK $\overline{WR3}$ to $\overline{WR0}$	--	15	ns
$\overline{WR3}$ to $\overline{WR0}$ delay time	t_{CLWH}		--	15	ns
Effective address ⇒ Effect data input time	t_{AVDV}	A20 to A0 D31 to D0	--	3/2 × t_{CYC}-30	ns
$\overline{RD}$ (fall) → Effect data input time	t_{RLDV}	RD D31 to D0	--	t_{CYC}-20	ns
Data set up → $\overline{RD}$ (rise) time	t_{DSRH}		25	--	ns
$\overline{RD}$ (rise) → Data hold time	t_{RHDX}		0	--	ns
AS delay time	t_{CHASL}	AS	--	15	ns
AS delay time	t_{CHASH}	AS	--	15	ns


tCYC
CLK
VOH
VOL
tCHASH
tCHASL
AS
tCHCSL
tCHCSH
CS0 - CS6
tCHAV
A23 - A00
tCLRL
tCLRH
RD
tRLDV
tRHDX
tAVDV
tDSRH
D31 - D00
tCLWL
tCLWH
WR3 - WR0
tCHDV
D31 - D00

34.8.4 RDY

(V_{DD} = 4.25 V to 5.25 V, T_A = –40°C to +85°C)

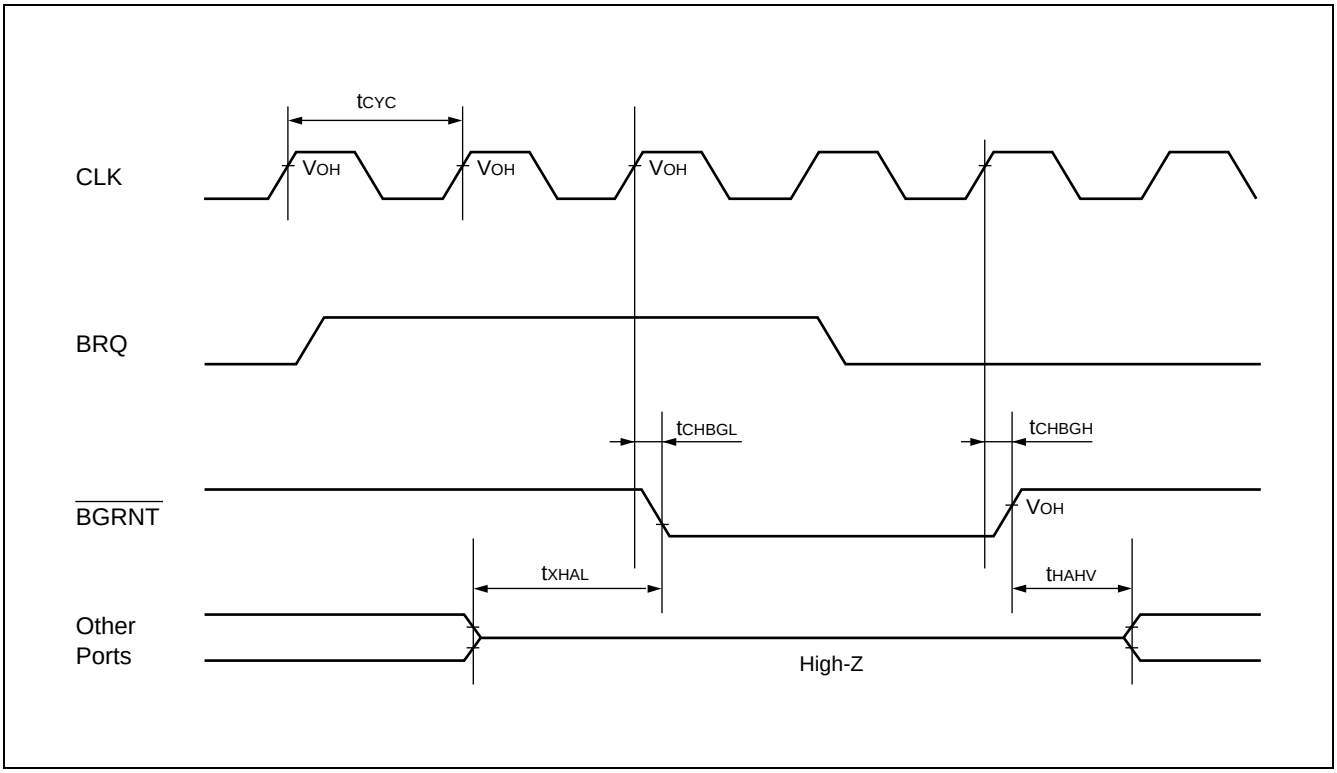
Signal	Symbol	Pin name	Value		Unit
			Min.	Max.	
RDY setup	t_{RDYS}	CLK RDY	16	--	ns
RDY hold	t_{RDYH}	CLK RDY	0	--	ns



34.8.5 BGRNT

(V_{DD} = 4.25 V to 5.25 V, T_A = –40°C to +85°C)

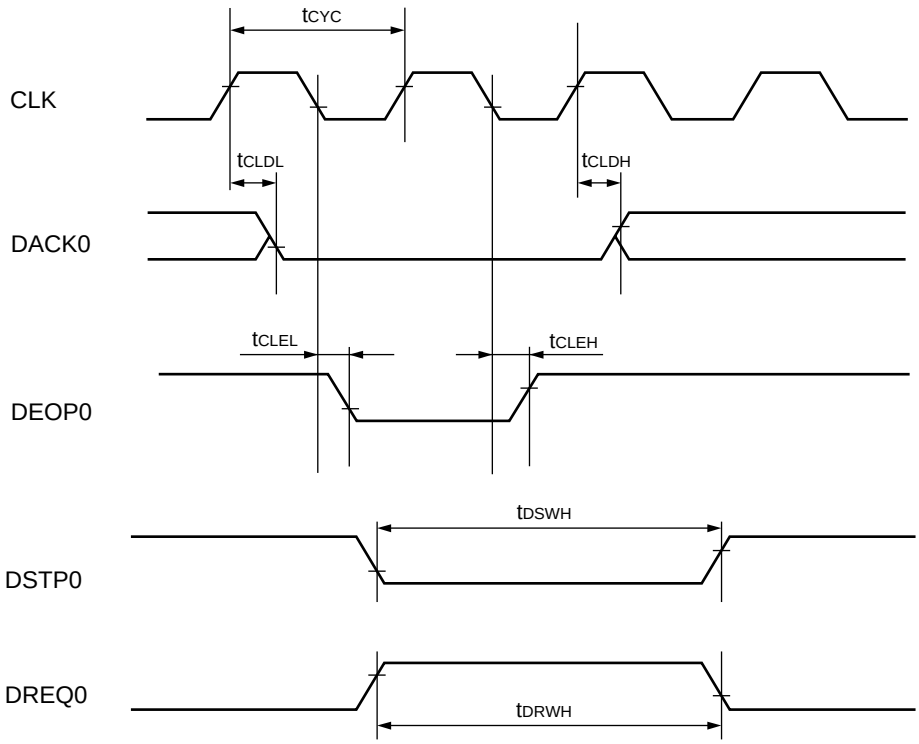
Signal	Symbol	Pin name	Value		Unit
			Min.	Max.	
$\overline{\text{BGRNT}}$	t_{CHBGL}	CLK	--	10	ns
$\overline{\text{BGRNT}}$	t_{CHBGH}	$\overline{\text{BGRNT}}$	--	10	ns
Bus access enabled $\overline{\text{BGRNT}}$ falling	t_{XHAL}	$\overline{\text{BGRNT}}$	tcyc – 15	tcyc + 15	ns
Bus access disabled $\overline{\text{BGRNT}}$ rising	t_{HAHV}		tcyc – 15	tcyc + 15	ns



34.8.6 DMA

(V_{DD} = 4.25 V to 5.25 V, T_A = –40°C to +85°C)

Signal	Symbol	Pin name	Value		Unit
			Min.	Max.	
DREQ	t_{DRWH}	DREQ0	$5t_{CYC}$	--	ns
DSTP	t_{DSWH}	DSTP0*	$5t_{CYC}$	--	ns
DACK	t_{CLDL}	CLK DACK0	--	20	ns
	t_{CLDH}		--	20	
DEOP	t_{CLEL}	CLK DEOP0	--	20	ns
	t_{CLEH}		--	20	



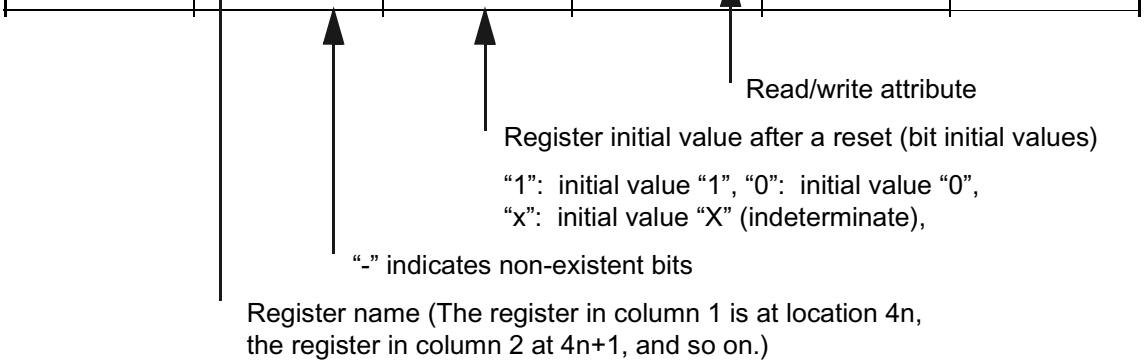
*) DSTP and DEOP share a pin. The pin is possible to change DSTP and DEOP functions using a port function register.

APPENDIX A I/O MAP

Table A lists the addresses for the registers used by the internal peripheral functions of the MB91360 devices.

■ How to Read the I/O Map

Address	Register				Internal peripheral
	+0	+1	+2	+3	
000014_H	PDRG [R/W] XXXXXX - -	PDRH [R/W] XXXXXXXX	PDRI [R/W] - - - - XXXX	—	Port data register



Location of far left of register (+0), +1, +2, and +3 each increment the location by one. When performing word access, the register in column 1 is placed at the MSB end of the data.

■ Precautions:

- Do not use RMW instructions on registers containing write-only (W) bits.

  RMW instructions(RMW:read-modify-write):

```
AND   Rj, @Ri     OR   Rj, @Ri     EOR   Rj, @Ri
ANDH  Rj, @Ri     ORH  Rj, @Ri     EORH  Rj, @Ri
ANDB  Rj, @Ri     ORB  Rj, @Ri     EORB  Rj, @Ri
BANDL #u4,@Ri     BORL #u4,@Ri     BEORL #u4,@Ri
BANDH #u4,@Ri     BORH #u4,@Ri     BEORH #u4, @Ri
```

- The data in reserved areas and areas marked "—" is indeterminate.
  Do not use those areas !

Address	Register				Block
	+0	+1	+2	+3	
000000_H	Reserved	Reserved	Reserved	Reserved	T-unit Port Data Register
000004_H	Reserved	Reserved	Reserved	PDR7 [R/W] -111 - - - -	
000008_H	PDR8 [R/W] XXXXXXXX	PDR9 [R/W] XXXXXXX1	————————	PDRB [R/W] - - - - - XXX	
$00000C_H$	————————				
000010_H	PDRG [R/W] XXXXXXXX	PDRH [R/W] XXXXXXXX	PDRI [R/W] X - - - X - - -	PDRJ [R/W] XXXXXXXX	R-bus Port Data Register
000014_H	PDRK [R/W] XXXXXXXX	PDRL [R/W] XXXXXXXX	PDRM [R/W] - - - - XXXX	PDRN [R/W] - - XXXXXX	
000018_H	PDRO [R/W] XXXXXXXX	PDRP [R/W] - - XXXXX	PDRQ [R/W] - - XXXXX	PDRR [R/W] XXXXXXXX	
$00001C_H$	PDRS [R/W] XXXXXXXX	PDRT [R/W] --XXXXXX			
000020_H | $00003C_H$					Reserved
000040_H	EIRR [R/W] 00000000	ENIR [R/W] 00000000	ELVR [R/W] 00000000 00000000		Ext int/NMI
000044_H	DICR [R/W] - - - - - - - 0	HRCL [R/W, R] 0 - - 11111	CLKR2 [R/W] - - - - - 000	reserved	DLYI/I-unit RTC
000048_H	TMRLR0 [W] XXXXXXXX XXXXXXXX		TMR0 [R] XXXXXXXX XXXXXXXX		Reload Timer 0
$00004C_H$	————————		TMCSR0 [R/W] - - - - 0000 - - - 00000		
000050_H	TMRLR1 [W] XXXXXXXX XXXXXXXX		TMR1 [R] XXXXXXXX XXXXXXXX		Reload Timer 1
000054_H	————————		TMCSR1 [R/W] - - - - 0000 - - - 00000		
000058_H	TMRLR2 [W] XXXXXXXX XXXXXXXX		TMR2 [R] XXXXXXXX XXXXXXXX		Reload Timer 2
$00005C_H$	————————		TMCSR2 [R/W] - - - - 0000 - - - 00000		

Address	Register				Block
	+0	+1	+2	+3	
000060_H	SSR0 [R/W, R] 00001 - 00	SIDR0 [R/W] XXXXXXXX	SCR0 [R/W, W] 00000100	SMR0 [R/W, W] 00 - - 0 - 0 -	UART0
000064_H	ULS0 [R/W] - - - - 0000				
000068_H	UTIM0/UTIMR0 [R/W] 00000000 00000000		DRCL0 [W] - - - - - - - -	UTIMC0 [R/W] 0 - - - 0 - 01	U-TIMER 0
$00006C_H$	SSR1 [R/W, R] 00001 - 00	SIDR1 [R/W] XXXXXXXX	SCR1 [R/W, W] 00000100	SMR1 [R/W, W] 00 - - 0 - 0 -	UART1
000070_H	ULS1 [R/W] - - - - 0000				
000074_H	UTIM1/UTIMR1 [R/W] 00000000 00000000		DRCL1 [W] - - - - - - - -	UTIMC1 [R/W] 0 - - - - - 01	U-TIMER 1
000078_H	SSR2 [R/W, R] 00001 - 00	SIDR2 [R/W] XXXXXXXX	SCR2 [R/W, W] 00000100	SMR2 [R/W, W] 00 - - 0 - 0 -	UART2
$00007C_H$	ULS2 [R/W] - - - - 0000				
000080_H	UTIM2/UTIMR2 [R/W] 00000000 00000000		DRCL2 [W] - - - - - - - -	UTIMC2 [R/W] 0 - - - 0 - 01	U-TIMER2
000084_H	SMCS0 [R/W, R] 00000010 - - - - 00-0		SES0 [R/W] - - - - - - 00	SDR0 [R/W] 00000000	SIO 0
000088_H	SMCS1 [R/W, R] 00000010 - - - - 00 - 0		SES1 [R/W] - - - - - - 00	SDR1 [R/W] 00000000	SIO 1
$00008C_H$	CDCR0 [R/W] 0 - - - 1111	Reserved	CDCR1 [R/W] 0 - - - 1111	Reserved	SIO 0/1 Prescaler
000090_H					Reserved
000094_H	IBCR [R/W] 00000000	IBSR [R] 00000000	IADR [R/W] -XXXXXXX	ICCR [R/W] - - 0XXXXX	I^2C (old) -> new I^2C from addr 0x184
000098_H		IDAR [R/W] XXXXXXXX		IDBL [R/W] -------0	
$00009C_H$	ADMD [R/W,W] - - - - 0000	ADCH [R/W] 00000000		ADCS [R/W,W] 0000 - - 00	A/D Converter
$0000A0_H$	ADCD [R/W] 000000XX XXXXXXXX			ADBL [R/W] - - - - - - - 0	

Address	Register				Block
	+0	+1	+2	+3	
$0000A4_H$		DACR [R/W] - - - - - 000	DADR0 [R/W] - - - - - - XX XXXXXXXX		DAC
$0000A8_H$	DADR1 [R/W] - - - - - - XX XXXXXXXX			DDBL [R/W] - - - - - - - 0	
$0000AC_H$	IOTDBL0 [R/W] - - - - - 000	ICS01 [R/W] 00000000	IOTDBL1 [R/W] - - - - - 000	ICS23 [R/W] 00000000	Input Capture 0,1,2,3
$0000B0_H$	IPCP0 [R] XXXXXXXX XXXXXXXX		IPCP1 [R] XXXXXXXX XXXXXXXX		
$0000B4_H$	IPCP2 [R] XXXXXXXX XXXXXXXX		IPCP3 [R] XXXXXXXX XXXXXXXX		
$0000B8_H$	OCS01 [R/W] - - - 0 - - 00 0000 - - 00		OCS23 [R/W] - - - 0 - - 00 0000 - - 00		Output Compare 0,1,2.3
$0000BC_H$	OCCP0 [R/W] XXXXXXXX XXXXXXXX		OCCP1 [R/W] XXXXXXXX XXXXXXXX		
$0000C0_H$	OCCP2 [R/W] XXXXXXXX XXXXXXXX		OCCP3 [R/W] XXXXXXXX XXXXXXXX		
$0000C4_H$					Reserved
$0000C8_H$	TCDT0 [R/W] XXXXXXXX XXXXXXXX		————————	TCCS0 [R/W] - 0000000	Free Running Counter 0 for ICU/ OCU
$0000CC_H$	TCDT1 [R/W] XXXXXXXX XXXXXXXX		————————	TCCS1 [R/W] - 0000000	Free Running Counter 1 for ICU/OCU
$0000D0_H$	ZPD0 [R/W,R] 00000010	PWC0 [R/W] 0000 0 - - 0	ZPD1 [R/W,R] 00000010	PWC1 [R/W] 0000 0 - - 0	SMC 0,1
$0000D4_H$	ZPD2 [R/W,R] 00000010	PWC2 [R/W] 0000 0 - - 0	ZPD3 [R/W,R] 00000010	PWC3 [R/W] 0000 0 - - 0	SMC 2,3
$0000D8_H$	PWC20 [R/W] XXXXXXXX	PWC10 [R/W] XXXXXXXX	PWS20 [R/W] - 0000000	PWS10 [R/W] - - 000000	SMC 0
$0000DC_H$	PWC21 [R/W] XXXXXXXX	PWC11 [R/W] XXXXXXXX	PWS21 [R/W] - 0000000	PWS11 [R/W] - - 000000	SMC 1
$0000E0_H$	PWC22 [R/W] XXXXXXXX	PWC12 [R/W] XXXXXXXX	PWS22 [R/W] - 0000000	PWS12 [R/W] - - 000000	SMC 2
$0000E4_H$	PWC23 [R/W] XXXXXXXX	PWC13 [R/W] XXXXXXXX	PWS23 [R/W] - 0000000	PWS13 [R/W] - - 000000	SMC 3
$0000E8_H$	SMDBL0 [R/W] - - - - - - - 0	SMDBL1 [R/W] - - - - - - 0	SMDBL2 [R/W] - - - - - - - 0	SMDBL3 [R/W] - - - - - - - 0	SMC 0,1,2,3

Address	Register				Block
	+0	+1	+2	+3	
$0000EC_H$		SGDBL [R/W] - - - - - - - 0	SGCR [R/W, R] 0 - - - - - 00 000 - - 000		Sound generator
$0000F0_H$	SGAR [R/W] 00000000	SGFR [R/W] XXXXXXXX	SGTR [R/W] XXXXXXXX	SGDR [R/W] XXXXXXXX	
$0000F4_H$		WTDBL [R/W] - - - - - - - 0	WTCR [R/W, R] 00000000 000 - 0000		Real Time Clock (Watch-Timer)
$0000F8_H$		WTBR [R/W] - - XXXXXX XXXXXXXX XXXXXXXX			
$0000FC_H$	WTHR [R/W] - - - 00000	WTMR [R/W] - - 000000	WTSR [R/W] - - 000000		
000100_H	TMRLR3 [W] XXXXXXXX XXXXXXXX		TMR3 [R] XXXXXXXX XXXXXXXX		Reload Timer 3
000104_H	_____		TMCSR3 [R/W] - - - - XX - - - - - XXXXX		
000108_H	TMRLR4 [W] XXXXXXXX XXXXXXXX		TMR4 [R] XXXXXXXX XXXXXXXX		Reload Timer 4
$00010C_H$	_____		TMCSR4 [R/W] - - - - XX - - - - - XXXXX		
000110_H	TMRLR5 [W] XXXXXXXX XXXXXXXX		TMR5 [R] XXXXXXXX XXXXXXXX		Reload Timer 5
000114_H	_____		TMCSR5 [R/W] - - - - XX - - - - - XXXXX		
000118_H	GCN10 [R/W] 00110010 00010000		PDBL0 [R/W] - - - - 0000	GCN20 [R/W] - - - - 0000	PWM Control 0
$00011C_H$	GCN11 [R/W] 00110010 00010000		PDBL1 [R/W] - - - - 0000	GCN21 [R/W] - - - - 0000	PWM Control 1
000120_H	PTMR0 [R] 11111111 11111111		PCSR0 [W] XXXXXXXX XXXXXXXX		PWM0
000124_H	PDUT0 [W] XXXXXXXX XXXXXXXX		PCNH0 [R/W] 0000000 -	PCNL0 [R/W] 000000 - 0	
000128_H	PTMR1 [R] 11111111 11111111		PCSR1 [W] XXXXXXXX XXXXXXXX		PWM1
$00012C_H$	PDUT1 [W] XXXXXXXX XXXXXXXX		PCNH1 [R/W] 0000000 -	PCNL1 [R/W] 000000 - 0	
000130_H	PTMR2 [R] 11111111 11111111		PCSR2 [W] XXXXXXXX XXXXXXXX		PWM2
000134_H	PDUT2 [W] XXXXXXXX XXXXXXXX		PCNH2 [R/W] 0000000 -	PCNL2 [R/W] 000000 - 0	

Address	Register				Block
	+0	+1	+2	+3	
000138_H	PTMR3 [R] 11111111 11111111		PCSR3 [W] XXXXXXXX XXXXXXXX		PWM3
$00013C_H$	PDUT3 [W] XXXXXXXX XXXXXXXX		PCNH3 [R/W] 0000000 -	PCNL3 [R/W] 000000 - 0	
000140_H	PTMR4 [R] 11111111 11111111		PCSR4 [W] XXXXXXXX XXXXXXXX		PWM4
000144_H	PDUT4 [W] XXXXXXXX XXXXXXXX		PCNH4 [R/W] 0000000 -	PCNL4 [R/W] 000000 - 0	
000148_H	PTMR5 [R] 11111111 11111111		PCSR5 [W] XXXXXXXX XXXXXXXX		PWM5
$00014C_H$	PDUT5 [W] XXXXXXXX XXXXXXXX		PCNH5 [R/W] 0000000 -	PCNL5 [R/W] 000000 - 0	
000150_H	PTMR6 [R] 11111111 11111111		PCSR6 [W] XXXXXXXX XXXXXXXX		PWM6
000154_H	PDUT 6 [W] XXXXXXXX XXXXXXXX		PCNH6 [R/W] 0000000 -	PCNL6 [R/W] 000000 - 0	
000158_H	PTMR7 [R] 11111111 11111111		PCSR7 [W] XXXXXXXX XXXXXXXX		PWM7
$00015C_H$	PDUT7 [W] XXXXXXXX XXXXXXXX		PCNH7 [R/W] 0000000 -	PCNL7 [R/W] 000000 - 0	
000160_H					Reserved
000164_H	CMCR [R/W] 11111111 0000000		CMPR [R/W, R] ----1001 1---0001		Clock Modulation
000168_H	CMLS0 [R/W] 01110111 1111111		CMLS1 [R/W] 01110111 1111111		
$00016C_H$	CMLS2 [R/W] 01110111 1111111		CMLS3 [R/W] 01110111 1111111		
000170_H	CMLT0 [R/W, R] -----100 00000010		CMLT1 [R/W, R] 11110100 00000010		
000174_H	CMLT2 [R/W, R] -----100 00000010		CMLT3 [R/W, R] -----100 00000010		
000178_H	CMAC [R/W] 11111111 1111111		CMTS [R] --000001 01111111		
$00017C_H$		PDRCR [R/W] - - - - - 000			Power down reset
000180_H	ACCDBL[R/W] - - - - - - - 0	ACSR [R/W, R] - - - XXX00			Alarm comparator

Address	Register				Block
	+0	+1	+2	+3	
000184_H	IBCR2 [R/W] 00000000	IBSR2 [R] 00000000	ITBAH [R/W] - - - - - - 00	ITBAL [R/W] 00000000	I^2C (new) (*) old and new I^2C share this bit!
000188_H	ITMKH [R/W, R] 00 - - - - 11	ITMKL [R/W] 11111111	ISMK [R/W] 01111111	ISBA [R/W] - 0000000	
$00018C_H$	IDARH [-] 00000000	IDAR2 [R/W] 00000000	ICCR2 [R/W] - 0011111	IDBL2(*) [R/W] - - - - - - - 0	
000190_H	CUCR [R/W, R] -------- ---0--00		CUTD [R/W] 10000000 00000000		Calibration Unit of 32kHz oscillator
000194_H	CUTR1 [R] -------- 00000000		CUTR2 [R] 00000000 00000000		
000198_H	SCR5 [R/W,W] 00000000	SMR5 [R/W,W] 00000000	SSR5 [R/W,R] 00001000	RDR5/TDR5 [R/W] 00000000	USART5
$00019C_H$	ESCR5 [R/W} 00000X00	ECCR5 [R/W,R,W] -00000XX	BGR15 [R/W] -0000000	BGR05 [R/W] 00000000	
$0001A0_H$	SCR6 [R/W,W] 00000000	SMR6 [R/W,W] 00000000	SSR6 [R/W,R] 00001000	RDR6/TDR6 [R/W] 00000000	USART6
$0001A4_H$	ESCR6 [R/W} 00000X00	ECCR6 [R/W,R,W] -00000XX	BGR16 [R/W] -0000000	BGR06 [R/W] 00000000	
$0001A8_H$ | $0001F8_H$					Reserved
$0001FC_H$			F362MD [R/W] 00000000	SFR [R/W] -------0 SFR_BDEN	F362 Mode Reg / Special function reg. (EDSU enable)

Address	Register				Block
	+0	+1	+2	+3	
000200_H	DMACA0 [R/W] 00000000 0000XXXX XXXXXXXX XXXXXXXX				DMAC
000204_H	DMACB0 [R/W] 00000000 00000000 XXXXXXXX XXXXXXXX				
000208_H	DMACA1 [R/W] 00000000 0000XXXX XXXXXXXX XXXXXXXX				
$00020C_H$	DMACB1 [R/W] 00000000 00000000 XXXXXXXX XXXXXXXX				
000210_H	DMACA2 [R/W] 00000000 0000XXXX XXXXXXXX XXXXXXXX				
000214_H	DMACB2 [R/W] 00000000 00000000 XXXXXXXX XXXXXXXX				
000218_H	DMACA3 [R/W] 00000000 0000XXXX XXXXXXXX XXXXXXXX				
$00021C_H$	DMACB3 [R/W] 00000000 00000000 XXXXXXXX XXXXXXXX				
000220_H	DMACA4 [R/W] 00000000 0000XXXX XXXXXXXX XXXXXXXX				
000224_H	DMACB4 [R/W] 00000000 00000000 XXXXXXXX XXXXXXXX				
000228_H | $00023C_H$	_____				
000240_H	DMACR [R/W] 0--00000 -------- -------- --------				
000244_H | $0002FC_H$	_____				Reserved
000300_H	IRBS [R/W, R] 00000000 00000001 00100000 --------				Instruction Cache
000304_H	_____			ISIZE [R/W] ------11	
000308_H | $0003E0_H$	_____				Reserved
$0003E4_H$	_____			ICHRC [R/W] 0-000000	Instruction Cache

Address	Register				Block
	+0	+1	+2	+3	
$0003E8_H$ | $0003EC_H$	_____				Reserved
$0003F0_H$	BSD0 [W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				Bit Search Module
$0003F4_H$	BSD1 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$0003F8_H$	BSDC [W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$0003FC_H$	BSRR [R] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
000400_H	DDRG [R/W] 00000000	DDRH [R/W] 00000000	DDRI [R/W] ----0---	DDRJ [R/W] 00000000	R-bus Port Direction Register
000404_H	DDRK [R/W] 00000000	DDRL [R/W] 00000000	DDRM [R/W] ----0000	DDRN [R/W] --000000	
000408_H	DDRO [R/W] 00000000	DDRP [R/W] ----0000	DDRQ [R/W] --000000	DDRR [R/W] 00000000	
$00040C_H$	DDRS [R/W] 00000000	DDRT [R/W] --000000			
000410_H	PFRG [R/W] 00000000	PFRH [R/W] 00000000	PFRI [R/W] ----0---	PFRJ [R/W] 00000000	R-bus Port Function Register
000414_H	PFRK [R/W] 00000000	PFRL [R/W] 00000000	PFRM [R/W] ----0000	PFRN [R/W] --000000	
000418_H	PFRO [R/W] 00000000	PFRP [R/W] 00000000	PFRQ [R/W] --000000	PFRR [R/W] 00000000	
$00041C_H$	PFRS [R/W] 00000000	PFRT [R/W] --000000			
000420_H | $00043C_H$	_____				Reserved

Address	Register				Block
	+0	+1	+2	+3	
000440_H	ICR00 [R/W, R] ---11111	ICR01 [R/W, R] ---11111	ICR02 [R/W, R] ---11111	ICR03 [R/W, R] ---11111	Interrupt Control unit
000444_H	ICR04 [R/W, R] ---11111	ICR05 [R/W, R] ---11111	ICR06 [R/W, R] ---11111	ICR07 [R/W, R] ---11111	
000448_H	ICR08 [R/W, R] ---11111	ICR09 [R/W, R] ---11111	ICR10 [R/W, R] ---11111	ICR11 [R/W, R] ---11111	
$00044C_H$	ICR12 [R/W, R] ---11111	ICR13 [R/W, R] ---11111	ICR14 [R/W, R] ---11111	ICR15 [R/W, R] ---11111	
000450_H	ICR16 [R/W, R] ---11111	ICR17 [R/W, R] ---11111	ICR18 [R/W, R] ---11111	ICR19 [R/W, R] ---11111	
000454_H	ICR20 [R/W, R] ---11111	ICR21 [R/W, R] ---11111	ICR22 [R/W, R] ---11111	ICR23 [R/W, R] ---11111	
000458_H	ICR24 [R/W, R] ---11111	ICR25 [R/W, R] ---11111	ICR26 [R/W, R] ---11111	ICR27 [R/W, R] ---11111	
$00045C_H$	ICR28 [R/W, R] ---11111	ICR29 [R/W, R] ---11111	ICR30 [R/W, R] ---11111	ICR31 [R/W, R] ---11111	
000460_H	ICR32 [R/W, R] ---11111	ICR33 [R/W, R] ---11111	ICR34 [R/W, R] ---11111	ICR35 [R/W, R] ---11111	
000464_H	ICR36 [R/W, R] ---11111	ICR37 [R/W, R] ---11111	ICR38 [R/W, R] ---11111	ICR39 [R/W, R] ---11111	
000468_H	ICR40 [R/W, R] ---11111	ICR41 [R/W, R] ---11111	ICR42 [R/W, R] ---11111	ICR43 [R/W, R] ---11111	
$00046C_H$	ICR44 [R/W, R] ---11111	ICR45 [R/W, R] ---11111	ICR46 [R/W, R] ---11111	ICR47 [R/W, R] ---11111	
000470_H | $00047C_H$	_____				Reserved
000480_H	RSRR [R/W, R] 10000-00	STCR [R/W] 00110011	TBCR [R/W] 00XXX-00	CTBR [W] XXXXXXXX	Clock Control unit
000484_H	CLKR [R/W] 00000000	WPR [W] XXXXXXXX	DIVR0 [R/W] 00000011	DIVR1 [R/W] 00000000	
000488_H | $0005FC_H$	_____				Reserved

Address	Register				Block
	+0	+1	+2	+3	
000600_H	_____	_____	_____	_____	T-unit Port Direction Register
000604_H	_____	_____	_____	DDR7 [R/W] 00000000	
000608_H	DDR8 [R/W] 00000000	DDR9 [R/W] 00000000	_____	DDRB [R/W] 00000000	
$00060C_H$	_____				
000610_H	_____	_____	_____	_____	T-unit Port Function Register
000614_H	_____	_____	_____	PFR7 [R/W] 00001111	
000618_H	PFR8 [R/W] 111110-0	PFR9 [R/W] 11110101	_____	PFRB [R/W] 00000000	
$00061C_H$	_____				
000620_H	_____				
000624_H	_____			PFR27 [R/W] 1111-00-	
000628_H | $00063F_H$	_____				Reserved
000640_H	ASR0 [W] 00000000 00000000		AMR0 [W] 11111000 11111111		T-unit
000644_H	ASR1 [W] 00000000 00000000		AMR1 [W] 00000000 00000000		
000648_H	ASR2 [W] 00000000 00000000		AMR2 [W] 00000000 00000000		
$00064C_H$	ASR3 [W] 00000000 00000000		AMR3 [W] 00000000 00000000		
000650_H	ASR4 [W] 00000000 00000000		AMR4 [W] 00000000 00000000		
000654_H	ASR5 [W] 00000000 00000000		AMR5 [W] 00000000 00000000		
000658_H	ASR6 [W] 00000000 00000000		AMR6 [W] 00000000 00000000		

Address	Register				Block
	+0	+1	+2	+3	
$00065C_H$	ASR7 [W] 00000000 00000000		AMR7 [W] 00000000 00000000		
000660_H	AMD0 [R/W] -00XX111	AMD1 [R/W] -XXXXXXX	AMD2 [R/W] --XXXXXX	AMD3 [R/W] --XXXXXX	
000664_H	AMD4 [R/W] --XXXXXX	AMD5 [R/W] --XXXXXX	AMD6 [R/W] -XXXXXXX	AMD7 [R/W] -XXXXXXX	
000668_H	CSE 11000011	————————	————————	————————	
$00066C_H$	————————		————————		
000670_H	CHE 11111111	————————	————————		
000674_H | $0007F8_H$	————————				Reserved
$0007FC_H$	————————	MODR [W] XXXXXXXX	————————	————————	Mode Register
000800_H | 000834_H	EDSU Registers, on MB91F364G only				EDSU
000838_H | $000AFC_H$	————————				Reserved

Address	Register				Block
	+0	+1	+2	+3	
$000B00_H$	ESTS0 X0000000	ESTS1 XXXXXXXX	ESTS2 XXXXXXXX	————————	DSU
$000B04_H$	ECTL0 0X000000	ECTL1 00000000	ECTL2 000X0000	ECTL3 00000X11	
$000B08_H$	ECNT0 XXXXXXXX	ECNT1 XXXXXXXX	EUSA XXX0000X	EDTC 0000XXXX	
$000B0C_H$	EWPT XXXXXXXX XXXXXXXX		————————		
$000B10_H$	EDTR0 XXXXXXXX XXXXXXXX		EDTR1 XXXXXXXX XXXXXXXX		
$000B14_H$ | $000B1C_H$	————————				
$000B20_H$	EIA0 XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000B24_H$	EIA1 XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000B28_H$	EIA2 XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000B2C_H$	EIA3 XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000B30_H$	EIA4 XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000B34_H$	EIA5 XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000B38_H$	EIA6 XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000B3C_H$	EIA7 XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000B40_H$	EDTA XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000B44_H$	EDTM XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000B48_H$	EOA0 XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000B4C_H$	EOA1 XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000B50_H$	EPCR XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				

Address	Register				Block
	+0	+1	+2	+3	
$000B54_H$	EPSR XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000B58_H$	EIAM0 XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000B5C_H$	EIAM1 XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000B60_H$	EOAM0/EODM0 XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000B64_H$	EOAM1/EODM1 XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000B68_H$	EOD0 XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$000B6C_H$	EOD1 XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
001000_H	DMASA0 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				DMAC
001004_H	DMADA0 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
001008_H	DMASA1 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$00100C_H$	DMADA1 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
001010_H	DMASA2 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
001014_H	DMADA2 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
001018_H	DMASA3 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$00101C_H$	DMADA3 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
001020_H	DMASA4 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
001024_H	DMADA4 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
001028_H | $003FFC_H$	_____				Reserved

Address	Register				Block
	+0	+1	+2	+3	
004000_H | $006FFF_H$	————————				Reserved
007000_H	FMCS [R/W] 1110X000	————————	————————	————————	Flash Memory Control Register
007004_H	FMWT [R/W] --000011	————————	————————	————————	
007008_H | $00FFFC_H$	————————				Reserved
010000_H | $010FFC_H$	For exact address range see chapter 3 "INSTRUCTION CACHE" on page 129. Cache only available on MB91FV360GA!				I-Cache 4 KB
011000_H | $011FFC_H$	Not available on MB91F364G. On MB91FV360, the cache can be used in I-RAM mode.				I-RAM 4 KB
012000_H | $01FFFC_H$	————————				Reserved
020000_H | $03BFFC_H$	————————				Reserved
$03C000_H$ | $03FFFC_H$	MB91F362GB, MB91F364G: Only 12KB ($03D000_H$ - $03FFC_H$)				User RAM 16 KB (D-Bus)
040000_H | $043FFC_H$	MB91F362GB, MB91F364G: Only 4 KB (040000_H - $040FFC_H$)				Fast RAM 16 KB (F-Bus)
044000_H | 0FEFFC	————————				Reserved
050000_H | $0507FC_H$	MB91F376G: 044000_H - $0447FFC_H$				Boot ROM 2 KB (F-Bus)
050800_H | $07FFF4_H$	————————				reserved

Address	Register				Block
	+0	+1	+2	+3	
080000_H | $09FFFC_H$	Sector 0 64 KB		Sector 7 64 KB		Flash Memory 512 KB [1)] on F-Bus
$0A0000_H$ | 0BFFFC	Sector 1 64 KB		Sector 8 64 KB		
$0C0000_H$ | 0DFFFC	Sector 2 64 KB		Sector 9 64 KB		
$0E0000_H$ | 0EFFFC	Sector 3 32 KB		Sector 10 32 KB		
$0F0000_H$ | $0F3FFC_H$	Sector 4 8 KB		Sector 11 8 KB		
$0F4000_H$ | $0F7FFC_H$	Sector 5 8 KB		Sector 12 8 KB		
$0F8000_H$ | $0FFFF4_H$	Sector 6 16 KB		Sector 13 16 KB		
$0FFFF8_H$	FMV [R] $06\ 00\ 00\ 00_H$				Mode Vector
$0FFFFC_H$	FRV [R] $00\ 05\ 00\ 00_H$ (MB91F376G: $00\ 04\ 40\ 00_H$)				Fixed Reset Vector

Write operations to address $0FFFF8_H$ and $0FFFFC_H$ are not possible. When reading these addresses, the values shown above will be read.

Note 1): For the 256 KB Flash macro, please refer to the data sheet of MB91F364G.
For the 768 KB flash macro, please refer to the data sheet of MB91F376G.

Address	Register				Block
	+0	+1	+2	+3	
100000_H	BVALR0 [R/W] 00000000 00000000		TREQR0 [R/W] 00000000 00000000		CAN 0 Remark: Address range for CAN 0 to CAN 3 depends on chip select range. Mentioned addresses are default values, determined by boot ROM contents.
100004_H	TCANR0 [W] 00000000 00000000		TCR0 [R/W] 00000000 00000000		
100008_H	RCR0 [R/W] 00000000 00000000		RRTRR0 [R/W] 00000000 00000000		
$10000C_H$	ROVRR0 [R/W] 00000000 00000000		RIER0 [R/W] 00000000 00000000		
100010_H	CSR0 [R/W, R] 00000000 00000001			LEIR0 [R/W] 000-0000	
100014_H	RTEC0 [R] 00000000 00000000		BTR0 [R/W] -1111111 11111111		
100018_H	IDER0 [R/W] XXXXXXXX XXXXXXXX		TRTRR0 [R/W] 00000000 00000000		
$10001C_H$	RFWTR0 [R/W] XXXXXXXX XXXXXXXX		TIER0 [R/W] 00000000 00000000		
100020_H	AMSR0 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
100024_H	AMR00 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100028_H	AMR10 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
$10002C_H$ | 100048_H	GENERAL PURPOSE RAM [R/W]				
$10004C_H$	IDR00 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100050_H	IDR10 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100054_H	IDR20 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100058_H	IDR30 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
$10005C_H$	IDR40 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100060_H	IDR50 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100064_H	IDR60 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				

Address	Register				Block
	+0	+1	+2	+3	
100068_H	IDR70 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				CAN 0
$10006C_H$	IDR80 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100070_H	IDR90 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100074_H	IDR100 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100078_H	IDR110 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
$10007C_H$	IDR120 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100080_H	IDR130 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100084_H	IDR140 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100088_H	IDR150 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
$10008C_H$	DLCR00 [R/W] -------- ----XXXX		DLCR10 [R/W] -------- ----XXXX		
100090_H	DLCR20 [R/W] -------- ----XXXX		DLCR30 [R/W] -------- ----XXXX		
100094_H	DLCR40 [R/W] -------- ----XXXX		DLCR50 [R/W] -------- ----XXXX		
100098_H	DLCR60 [R/W] -------- ----XXXX		DLCR70 [R/W] -------- ----XXXX		
$10009C_H$	DLCR80 [R/W] -------- ----XXXX		DLCR90 [R/W] -------- ----XXXX		
$1000A0_H$	DLCR100 [R/W] -------- ----XXXX		DLCR110 [R/W] -------- ----XXXX		
$1000A4_H$	DLCR120 [R/W] -------- ----XXXX		DLCR130 [R/W] -------- ----XXXX		
$1000A8_H$	DLCR140 [R/W] -------- ----XXXX		DLCR150 [R/W] -------- ----XXXX		
$1000AC_H$	DTR00 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$1000B4_H$	DTR10 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				

Address	Register				Block
	+0	+1	+2	+3	
$1000BC_H$	DTR20 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				CAN 0
$1000C4_H$	DTR30 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$1000CC_H$	DTR40 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$1000D4_H$	DTR50 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$1000DC_H$	DTR60 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$1000E4_H$	DTR70 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$1000EC_H$	DTR80 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$1000F4_H$	DTR90 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$1000FC_H$	DTR100 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
100104_H	DTR110 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$10010C_H$	DTR120 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
100114_H	DTR130 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$10011C_H$	DTR140 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
100124_H	DTR150 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$10012C_H$	CREG0 [R/W] 00000000 00000110				

Address	Register				Block
	+0	+1	+2	+3	
100200_H	BVALR1 [R/W] 00000000 00000000		TREQR1 [R/W] 00000000 00000000		CAN 1 Remark: Address range for CAN 0 to CAN 3 depends on chip select range. Mentioned addresses are default values, determined by boot ROM contents.
100204_H	TCANR1 [W] 00000000 00000000		TCR1 [R/W] 00000000 00000000		
100208_H	RCR1 [R/W] 00000000 00000000		RRTRR1 [R/W] 00000000 00000000		
$10020C_H$	ROVRR1 [R/W] 00000000 00000000		RIER1 [R/W] 00000000 00000000		
100210_H	CSR1 [R/W, R] 00000000 00000001			LEIR1 [R/W] 000-0000	
100214_H	RTEC1 [R] 00000000 00000000		BTR1 [R/W] -1111111 11111111		
100218_H	IDER1 [R/W] XXXXXXXX XXXXXXXX		TRTRR1 [R/W] 00000000 00000000		
$10021C_H$	RFWTR1 [R/W] XXXXXXXX XXXXXXXX		TIER1 [R/W] 00000000 00000000		
100220_H	AMSR1 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
100224_H	AMR01 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100228_H	AMR11 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
$10022C_H$ | 100248_H	GENERAL PURPOSE RAM [R/W]				
$10024C_H$	IDR01 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100250_H	IDR11 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100254_H	IDR21[R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100258_H	IDR31 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX-				
$10025C_H$	IDR41 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100260_H	IDR51 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100264_H	IDR61 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				

Address	Register				Block
	+0	+1	+2	+3	
100268_H	IDR71 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				CAN 1
$10026C_H$	IDR81 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100270_H	IDR91 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100274_H	IDR101 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100278_H	IDR111 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
$10027C_H$	IDR121 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXX---				
100280_H	IDR131 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100284_H	IDR141 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100288_H	IDR151 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
$10028C_H$	DLCR01 [R/W] -------- ----XXXX		DLCR11 [R/W] -------- ----XXXX		
100290_H	DLCR21 [R/W] -------- ----XXXX		DLCR31 [R/W] -------- ----XXXX		
100294_H	DLCR41 [R/W] -------- ----XXXX		DLCR51 [R/W] -------- ----XXXX		
100298_H	DLCR61 [R/W] -------- ----XXXX		DLCR71 [R/W] -------- ----XXXX		
$10029C_H$	DLCR81[R/W] -------- ----XXXX		DLCR91 [R/W] -------- ----XXXX		
$1002A0_H$	DLCR101 [R/W] -------- ----XXXX		DLCR111 [R/W] -------- ----XXXX		
$1002A4_H$	DLCR121 [R/W] -------- ----XXXX		DLCR131 [R/W] -------- ----XXXX		
$1002A8_H$	DLCR141 [R/W] -------- ----XXXX		DLCR151 [R/W] -------- ----XXXX		
$1002AC_H$	DTR01 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$1002B4_H$	DTR11 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				

Address	Register				Block
	+0	+1	+2	+3	
$1002BC_H$	DTR21 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				CAN 1
$1002C4_H$	DTR31 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$1002CC_H$	DTR41 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$1002D4_H$	DTR51 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$1002DC_H$	DTR61 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$1002E4_H$	DTR71 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$1002EC_H$	DTR81 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$1002F4_H$	DTR91 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$1002FC_H$	DTR101 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
100304_H	DTR111 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$10030C_H$	DTR121 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
100314_H	DTR131 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$10031C_H$	DTR141 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				

Address	Register				Block
	+0	+1	+2	+3	
100324_H	DTR151 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$10032C_H$	CREG1 [R/W] 00000000 00000110				

Address	Register				Block
	+0	+1	+2	+3	
100400_H	BVALR2 [R/W] 00000000 00000000		TREQR2 [R/W] 00000000 00000000		CAN 2 Remark: Address range for CAN 0 to CAN 3 depends on chip select range. Mentioned addresses are default values, determined by boot ROM contents.
100404_H	TCANR2 [W] 00000000 00000000		TCR2 [R/W] 00000000 00000000		
100408_H	RCR2 [R/W] 00000000 00000000		RRTRR1 [R/W] 00000000 00000000		
$10040C_H$	ROVRR2 [R/W] 00000000 00000000		RIER2 [R/W] 00000000 00000000		
100410_H	CSR2 [R/W, R] 00000000 00000001			LEIR2 [R/W] 000-0000	
100414_H	RTEC2 [R] 00000000 00000000		BTR2 [R/W] -1111111 11111111		
100418_H	IDER2 [R/W] XXXXXXXX XXXXXXXX		TRTRR2 [R/W] 00000000 00000000		
$10041C_H$	RFWTR2 [R/W] XXXXXXXX XXXXXXXX		TIER2 [R/W] 00000000 00000000		
100420_H	AMSR2 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
100424_H	AMR02 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100428_H	AMR12 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
$10042C_H$ | 100448_H	GENERAL PURPOSE RAM [R/W]				
$10044C_H$	IDR02 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100450_H	IDR12 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100454_H	IDR22[R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100458_H	IDR32 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX-				
$10045C_H$	IDR42 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100460_H	IDR52 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100464_H	IDR62 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				

Address	Register				Block
	+0	+1	+2	+3	
100468_H	IDR72 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				CAN 2
$10046C_H$	IDR82 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100470_H	IDR92 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100474_H	IDR102 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100478_H	IDR112 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
$10047C_H$	IDR122 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXX---				
100480_H	IDR132 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100484_H	IDR142 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100488_H	IDR152 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
$10048C_H$	DLCR02 [R/W] -------- ----XXXX		DLCR12 [R/W] -------- ----XXXX		
100490_H	DLCR22 [R/W] -------- ----XXXX		DLCR32 [R/W] -------- ----XXXX		
100494_H	DLCR42 [R/W] -------- ----XXXX		DLCR52 [R/W] -------- ----XXXX		
100498_H	DLCR62 [R/W] -------- ----XXXX		DLCR72 [R/W] -------- ----XXXX		
$10049C_H$	DLCR82[R/W] -------- ----XXXX		DLCR92 [R/W] -------- ----XXXX		
$1004A0_H$	DLCR102 [R/W] -------- ----XXXX		DLCR112 [R/W] -------- ----XXXX		
$1004A4_H$	DLCR122 [R/W] -------- ----XXXX		DLCR132 [R/W] -------- ----XXXX		
$1004A8_H$	DLCR142 [R/W] -------- ----XXXX		DLCR152 [R/W] -------- ----XXXX		
$1004AC_H$	DTR02 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$1004B4_H$	DTR12 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				

Address	Register				Block
	+0	+1	+2	+3	
$1004BC_H$	DTR22 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				CAN 2
$1004C4_H$	DTR32 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$1004CC_H$	DTR42 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$1004D4_H$	DTR52 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$1004DC_H$	DTR62 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$1004E4_H$	DTR72 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$1004EC_H$	DTR82 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$1004F4_H$	DTR92 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$1004FC_H$	DTR102 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
100504_H	DTR112 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$10050C_H$	DTR122 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
100514_H	DTR132 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$10051C_H$	DTR142 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				

Address	Register				Block
	+0	+1	+2	+3	
100524_H	DTR152 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$10052C_H$	CREG2 [R/W] 00000000 00000110				

Address	Register				Block
	+0	+1	+2	+3	
100600_H	BVALR3 [R/W] 00000000 00000000		TREQR3 [R/W] 00000000 00000000		CAN 3 Remark: Address range for CAN 0 to CAN 3 depends on chip select range. Mentioned addresses are default values, determined by boot ROM contents.
100604_H	TCANR3 [W] 00000000 00000000		TCR3 [R/W] 00000000 00000000		
100608_H	RCR3 [R/W] 00000000 00000000		RRTRR31 [R/W] 00000000 00000000		
$10060C_H$	ROVRR3 [R/W] 00000000 00000000		RIER3 [R/W] 00000000 00000000		
100610_H	CSR3 [R/W, R] 00000000 00000001			LEIR3 [R/W] 000-0000	
100614_H	RTEC3 [R] 00000000 00000000		BTR3 [R/W] -1111111 11111111		
100618_H	IDER3 [R/W] XXXXXXXX XXXXXXXX		TRTRR3 [R/W] 00000000 00000000		
$10061C_H$	RFWTR3 [R/W] XXXXXXXX XXXXXXXX		TIER3 [R/W] 00000000 00000000		
100620_H	AMSR3 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
100624_H	AMR03 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100628_H	AMR13 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
$10062C_H$ | 100648_H	GENERAL PURPOSE RAM [R/W]				
$10064C_H$	IDR03 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100650_H	IDR13 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100654_H	IDR23[R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100658_H	IDR33 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX-				
$10065C_H$	IDR43 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100660_H	IDR53 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100664_H	IDR63 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				

Address	Register				Block
	+0	+1	+2	+3	
100668_H	IDR73 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				CAN 3
$10066C_H$	IDR83 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100670_H	IDR93 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100674_H	IDR103 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100678_H	IDR113 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
$10067C_H$	IDR123 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXX---				
100680_H	IDR133 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100684_H	IDR143 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
100688_H	IDR153 [R/W] XXXXXXXX XXXXXXXX XXXXX--- XXXXXXXX				
$10068C_H$	DLCR033 [R/W] -------- ----XXXX		DLCR13 [R/W] -------- ----XXXX		
100690_H	DLCR233 [R/W] -------- ----XXXX		DLCR33 [R/W] -------- ----XXXX		
100694_H	DLCR43 [R/W] -------- ----XXXX		DLCR53 [R/W] -------- ----XXXX		
100698_H	DLCR63 [R/W] -------- ----XXXX		DLCR73 [R/W] -------- ----XXXX		
$10069C_H$	DLCR83[R/W] -------- ----XXXX		DLCR93 [R/W] -------- ----XXXX		
$1006A0_H$	DLCR103 [R/W] -------- ----XXXX		DLCR113 [R/W] -------- ----XXXX		
$1006A4_H$	DLCR123 [R/W] -------- ----XXXX		DLCR133 [R/W] -------- ----XXXX		
$1006A8_H$	DLCR143 [R/W] -------- ----XXXX		DLCR153 [R/W] -------- ----XXXX		
$1006AC_H$	DTR03 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$1006B4_H$	DTR13 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				

Address	Register				Block
	+0	+1	+2	+3	
$1006BC_H$	DTR23 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				CAN 3
$1006C4_H$	DTR33 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$1006CC_H$	DTR43 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$1006D4_H$	DTR53 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$1006DC_H$	DTR63 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$1006E4_H$	DTR73 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$1006EC_H$	DTR83 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$1006F4_H$	DTR93 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$1006FC_H$	DTR103 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
100704_H	DTR113 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$10070C_H$	DTR123 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
100714_H	DTR133 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$10071C_H$	DTR143 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				

Address	Register				Block
	+0	+1	+2	+3	
100724_H	DTR153 [R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
$10072C_H$	CREG3 [R/W] 00000000 00000110				

APPENDIX B INTERRUPT VECTORS

This appendix lists the interrupt vector table.
The interrupt vector table lists the interrupt vectors and interrupt control registers assigned to each MB91360 interrupt.

Interrupt	Interrupt number		Interrupt level *1		Interrupt vector *2		
	Decimal	Hexa-decimal	Setting Register	Register address	Offset	Default Vector address	RN
Reset	0	00	-	-	0x3FC	0x000FFFFC	
Mode vector	1	01	-	-	0x3F8	0x000FFFF8	
System reserved	2	02	-	-	0x3F4	0x000FFFF4	
System reserved	3	03	-	-	0x3F0	0x000FFFF0	
System reserved	4	04	-	-	0x3EC	0x000FFFEC	
System reserved	5	05	-	-	0x3E8	0x000FFFE8	
System reserved	6	06	-	-	0x3E4	0x000FFFE4	
Co-processor fault trap *4	7	07	-	-	0x3E0	0x000FFFE0	
Co-processor error trap *4	8	08	-	-	0x3DC	0x000FFFDC	
INTE instruction *4	9	09	-	-	0x3D8	0x000FFFD8	
Instruction break exception *4	10	0A	-	-	0x3D4	0x000FFFD4	
Operand break trap *4	11	0B	-	-	0x3D0	0x000FFFD0	
Step trace trap *4	12	0C	-	-	0x3CC	0x000FFFCC	
NMI interrupt(tool)*4	13	0D	-	-	0x3C8	0x000FFFC8	
Undefined instruction exception	14	0E	-	-	0x3C4	0x000FFFC4	
NMI request	15	0F	F_H fixed		0x3C0	0x000FFFC0	
External Interrupt 0	16	10	ICR00	0x440	0x3BC	0x000FFFBC	4
External Interrupt 1	17	11	ICR01	0x441	0x3B8	0x000FFFB8	5
External Interrupt 2	18	12	ICR02	0x442	0x3B4	0x000FFFB4	8
External Interrupt 3	19	13	ICR03	0x443	0x3B0	0x000FFFB0	9
External Interrupt 4	20	14	ICR04	0x444	0x3AC	0x000FFFAC	
External Interrupt 5	21	15	ICR05	0x445	0x3A8	0x000FFFA8	

External Interrupt 6	22	16	ICR06	0x446	0x3A4	0x000FFFA4	
External Interrupt 7	23	17	ICR07	0x447	0x3A0	0x000FFFA0	
Reload Timer 0	24	18	ICR08	0x448	0x39C	0x000FFF9C	6
Reload Timer 1	25	19	ICR09	0x449	0x398	0x000FFF98	7
Reload Timer 2	26	1A	ICR10	0x44A	0x394	0x000FFF94	
CAN 0 RX	27	1B	ICR11	0x44B	0x390	0x000FFF90	
CAN 0 TX/NS	28	1C	ICR12	0x44C	0x38C	0x000FFF8C	
CAN 1 RX	29	1D	ICR13	0x44D	0x388	0x000FFF88	
CAN 1 TX/NS	30	1E	ICR14	0x44E	0x384	0x000FFF84	
CAN 2 RX	31	1F	ICR15	0x44F	0x380	0x000FFF80	
CAN 2 TX/NS	32	20	ICR16	0x450	0x37C	0x000FFF7C	
CAN 3 RX *5	33	21	ICR17	0x451	0x378	0x000FFF78	
CAN 3 TX/NS *5	34	22	ICR18	0x452	0x374	0x000FFF74	
PPG 0/1	35	23	ICR19	0x453	0x370	0x000FFF70	
PPG 2/3	36	24	ICR20	0x454	0x36C	0x000FFF6C	
PPG 4/5	37	25	ICR21	0x455	0x368	0x000FFF68	
PPG 6/7	38	26	ICR22	0x456	0x364	0x000FFF64	
Reload Timer 3	39	27	ICR23	0x457	0x360	0x000FFF60	
Reload Timer 4	40	28	ICR24	0x458	0x35C	0x000FFF5C	
Reload Timer 5	41	29	ICR25	0x459	0x358	0x000FFF58	
ICU 0/1	42	2A	ICR26	0x45A	0x354	0x000FFF54	
OCU 0/1	43	2B	ICR27	0x45B	0x350	0x000FFF50	
ICU 2/3	44	2C	ICR28	0x45C	0x34C	0x000FFF4C	
OCU 2/3	45	2D	ICR29	0x45D	0x348	0x000FFF48	
ADC	46	2E	ICR30	0x45E	0x344	0x000FFF44	14
Timebase Overflow	47	2F	ICR31	0x45F	0x340	0x000FFF40	
Free Running Counter 0	48	30	ICR32	0x460	0x33C	0x000FFF3C	
Free Running Counter 1	49	31	ICR33	0x461	0x338	0x000FFF38	
SIO 0 *6	50	32	ICR34	0x462	0x334	0x000FFF34	12
SIO 1 *6	51	33	ICR35	0x463	0x330	0x000FFF30	15
Sound Generator	52	34	ICR36	0x464	0x32C	0x000FFF2C	
UART 0 RX	53	35	ICR37	0x465	0x328	0x000FFF28	0
UART 0 TX	54	36	ICR38	0x466	0x324	0x000FFF24	1

UART 1 RX / USART5 RX *7	55	37	ICR39	0x467	0x320	0x000FFF20	2*8
UART 1 TX/ USART5 TX *7	56	38	ICR40	0x468	0x31C	0x000FFF1C	3*8
UART 2 RX/ USART6 RX *7	57	39	ICR41	0x469	0x318	0x000FFF18	10*8
UART 2 TX/ USART6 TX *7	58	3A	ICR42	0x46A	0x314	0x000FFF14	11*8
I^2C	59	3B	ICR43	0x46B	0x310	0x000FFF10	13
Alarm Comparator	60	3C	ICR44	0x46C	0x30C	0x000FFF0C	
RTC (Watchtimer) / Calibration Unit	61	3D	ICR45	0x46D	0x308	0x000FFF08	
DMA	62	3E	ICR46	0x46E	0x304	0x000FFF04	
Delayed interrupt activation bit	63	3F	ICR47	0x46F	0x300	0x000FFF00	
System reserved *3	64	40	-	-	0x2FC	0x000FFEFC	
System reserved *3	65	41	-	-	0x2F8	0x000FFEF8	
Security vector	66	42			0x2F4	0x000FFEF4	
System reserved	67	43	(ICR51)	0x473	0x2F0	0x000FFEF0	
System reserved	68	44	(ICR52)	0x474	0x2EC	0x000FFEEC	
System reserved	69	45	(ICR53)	0x475	0x2E8	0x000FFEE8	
System reserved	70	46	(ICR54)	0x476	0x2E4	0x000FFEE4	
System reserved	71	47	(ICR55)	0x477	0x2E0	0x000FFEE0	
System reserved	72	48	(ICR56)	0x478	0x2DC	0x000FFEDC	
System reserved	73	49	(ICR57)	0x479	0x2D8	0x000FFED8	
System reserved	74	4A	(ICR58)	0x47A	0x2D4	0x000FFED4	
System reserved	75	4B	(ICR59)	0x47B	0x2D0	0x000FFED0	
System reserved	76	4C	(ICR60)	0x47C	0x2CC	0x000FFECC	
System reserved	77	4D	(ICR61)	0x47D	0x2C8	0x000FFEC8	
System reserved	78	4E	(ICR62)	0x47E	0x2C4	0x000FFEC4	
System reserved	79	4F	(ICR63)	0x47F	0x2C0	0x000FFEC0	
Used by the INT instruction.	80 to 255	50 to FF	-	-	0x2BC to 0x000	0x000FFEBC to 0x000FFC00	

*1 The ICRs are located in the interrupt controller and set the interrupt level for each interrupt request. An ICR is provided for each interrupt request.

*2 The vector address for each EIT (exception, interrupt or trap) is calculated by adding the listed offset to the table base register value (TBR). The TBR specifies the top of the EIT vector table. The addresses listed in

the table are for the default TBR value (0x000FFC00). The TBR is initialized to this value by a reset.After execution of the internal boot ROM TBR is set to 0_x00FFC00.

*3 Used by REALOS

*4 System reserved

*5 Only available on MB91FV360GA

*6 DMA to / from SIO is only implemented on MB91F369GA.

*7 USART5/6 in MB91F364G, in all other devices UART1/2.

*8 DMA to/from the USARTs in MB91F364G is not implemented.

Remarks:

The 1-Kbyte area from the address specified in TBR is the EIT vector area.

Each vector consists of four bytes. The following formula shows the relationship between the vector number and vector address.

vctadr = TBR + vctofs

= TBR + ($3FC_H - 4 \times$ vct)

vctadr : Vector address
vctofs : Vector offset
vct : Vector number

APPENDIX C PIN STATES IN EACH CPU STATE

This appendix describes the following items.

- **Data bus and memory operation modes**
- **Terminology used in the table of pin states**
- **Pin states in each CPU state**

■ Operation Modes

This appendix describes the pin states for the following mode:

Internal ROM mode, 32-bit data bus.

■ Terminology Used in the Table of Pin States

The following describes the main terminology used in the table of pin states.

For details see the pin state table.

■ Operation mode terminology

Table Ca Operation Mode Terminology

Term	Description
Sleep mode	Writing "1" to the SLEEP bit of the STCR register sets the device to sleep mode (STOP bit is "0"). STCR: Standby control register in the clock controller
Stop mode (HIZ = "0")	Writing "1" to the STOP bit of the STCR register when HIZ = "0" sets the device to stop mode. In this stop mode, the outputs pins hold their states prior to entering stop mode.The inputs are fixed at 0.
Stop mode (HIZ = "1")	Writing "1" to the STOP bit of the STCR register when HIZ = "1" sets the device to stop mode. In this stop mode, the I/O pins go to high impedance (inital value). The inputs are fixed at 0.
RTC mode	Writing "1" to the STOP bit of the STCR register when bit OSCD1 of this register is "0" sets the device to RTC mode. In this mode the selected oscillator is still running and the Real Time Clock (RTC) can be operated. Pins states are as in Stop mode.
Bus release (BGRNTX = "0")	The BRQ input is enabled when bit 2 of the PFR8 register is set to "1". In this case, inputting an "H" level signal to the BRQ pin releases the external bus and outputs an "L" level signal from the BGRNTX pin.
Setting Initialization Reset (INIT)	The following conditions trigger a INIT reset. (1) Input from the external reset pin (INITX) (2) Release of hardware standby (3) Watchdog timer timeout (4) Power-down reset
Operation Initialization Reset (RST)	The following conditions trigger a RST reset. (1) Input from the external reset pin (RSTX) (2) Software reset: Setting the SRST bit of the STCR register to “1”.
Hardware standby	While an “L” level is applied to the HSTX input pin, the oscillation halts and all I/O pins go to high impendance. The inputs are fixed to 0.

■ Operation state terminology

Table Cb Operation State Terminology

Term	Description
Input enabled	The input function is enabled.
Input fixed at 0	The external input is disconnected at the input gate for the pin and the input state is fixed at "0" internally.
Hi-Z output	The pin drive transistor is disabled and the pin goes to high impedance.
Maintain output	Indicates that pins maintain the output states that they had prior to entering the specified mode. That is, outputs from internal peripherals that have not halted continue to operate under the control of the internal peripheral. Pins set as port outputs maintain their output states.
Maintain previous state	For output pins, indicates that the pins maintain the output states that they had prior to entering the specified mode. For input pins, indicates the input remains enabled.

■ Pin States in Each CPU State

Table Cc Pin States in Each CPU State

<table>
<tr><th colspan="2">Pin function</th><th rowspan="2">Sleep mode</th><th colspan="2">Stop Mode / RTC Mode</th><th rowspan="2">INIT Reset</th><th rowspan="2">RST Reset</th><th rowspan="2">Hardware standby</th><th rowspan="2">Bus release (BGRNTX=0)</th></tr>
<tr><th>Port</th><th>Function</th><th>HIZX=0</th><th>HIZX=1</th></tr>
<tr><td></td><td>D0-D7</td><td rowspan="39">Maintain previous state</td><td rowspan="37">Maintain outputs, Inputs fixed at 0</td><td rowspan="37">Hi-Z outputs, Inputs fixed at 0</td><td rowspan="30">Hi-Z outputs, Inputs enabled (but see chapter about I/O ports for details of initial status)</td><td rowspan="30">Maintain previous state (but see chapter about I/O ports for details of initial status)</td><td rowspan="37">Hi-Z outputs, Inputs fixed at 0</td><td rowspan="20">Hi-Z outputs, Inputs enabled</td></tr>
<tr><td></td><td>D8-D15</td></tr>
<tr><td></td><td>D16-D23</td></tr>
<tr><td></td><td>D24-D31</td></tr>
<tr><td></td><td>A0-A7</td></tr>
<tr><td></td><td>A8-A15</td></tr>
<tr><td></td><td>A16-A20 (-A23)</td></tr>
<tr><td>P70-P72</td><td>A24-A26</td></tr>
<tr><td>P73</td><td>DREQ2</td></tr>
<tr><td>P74-P76</td><td>CS4X-CS6X</td></tr>
<tr><td></td><td>CS7X</td></tr>
<tr><td></td><td>RDY</td></tr>
<tr><td>P81</td><td>BGRNTX</td><td>Low</td></tr>
<tr><td>P82</td><td>BRQ</td><td>High</td></tr>
<tr><td></td><td>RDX</td><td rowspan="3">Hi-Z outputs, Inputs enabled</td></tr>
<tr><td></td><td>WR0X-WR3X</td></tr>
<tr><td>P90</td><td>AS</td></tr>
<tr><td>P91</td><td>ALE</td><td rowspan="2">Clock Out</td></tr>
<tr><td></td><td>CLK</td></tr>
<tr><td>P93</td><td>AH/BOOT</td><td>prev. state</td></tr>
<tr><td>P94-P97</td><td>CS0X-CS3X</td><td>Hi-Z out</td></tr>
<tr><td>PB0</td><td>DREQ0</td><td rowspan="24">Maintain previous state</td></tr>
<tr><td>PB1</td><td>DACK0</td></tr>
<tr><td>PB2</td><td>DEOP0</td></tr>
<tr><td>PB3</td><td>DREQ1</td></tr>
<tr><td>PB4</td><td>DACK1</td></tr>
<tr><td>PB5</td><td>DEOP1</td></tr>
<tr><td>PB6</td><td>DACK2</td></tr>
<tr><td>PB7</td><td>DEOP2</td></tr>
<tr><td>PG0-PG7</td><td>AN8-AN15</td><td rowspan="9">Hi-Z outputs, Inputs enabled, (port mode)</td><td rowspan="9">Hi-Z outputs, Inputs enabled, (port mode)</td></tr>
<tr><td>PH0-PH7</td><td>AN0-AN7</td></tr>
<tr><td></td><td>DA0-DA1</td></tr>
<tr><td>PI3</td><td>ATGX</td></tr>
<tr><td>PJ0-PJ7</td><td>LED0-LED7</td></tr>
<tr><td>PK0-PK7</td><td>INT0-INT7</td><td>Inputs enabled</td><td>Inputs enabled</td><td>Inputs enabled</td></tr>
<tr><td>PL0-PL3</td><td>IN0-IN3</td><td rowspan="9">Maintain outputs, Inputs fixed at 0</td><td rowspan="9">Hi-Z outputs, Inputs fixed at 0</td><td rowspan="9">Hi-Z outputs, Inputs fixed at 0</td></tr>
<tr><td>PL4-PL7</td><td>OUT0-OUT3</td></tr>
<tr><td>PM0</td><td>SGO</td></tr>
<tr><td>PM1</td><td>SGA</td></tr>
<tr><td>PM2</td><td>SDA</td></tr>
<tr><td>PM3</td><td>SCL</td></tr>
<tr><td>PN0</td><td>SOT4</td></tr>
<tr><td>PN1</td><td>SIN4</td></tr>
</table>

Table Cc Pin States in Each CPU State (Continued)

Pin function		Sleep mode	Stop Mode / RTC Mode		INIT Reset	RST Reset	Hardware standby	Bus release (BGRNTX=0)
Port	Function		HIZX=0	HIZX=1				
PN2	SCK4	Maintain previous state	Maintain outputs, Inputs fixed at 0	Hi-Z outputs, Inputs fixed at 0	Hi-Z outputs, inputs enabled (port operation)	Hi-Z outputs, inputs enabled (port operation)	Hi-Z outputs, Inputs fixed at 0	Maintain previous state
PN3	SIN3							
PN4	SOT3							
PN5	SCK3							
PO0-PO7	OCPA0-OCPA7							
PP0	TX0							
PP1	RX0							
PP2	TX1							
PP3	RX1							
PP4	TX2							
PP5	RX2							
PP6	TX3							
PP7	RX3							
PQ0	SIN0							
PQ1	SOT0							
PQ2	SIN1							
PQ3	SOT1							
PQ4	SIN2							
PQ5	SOT2							
PR0-PR7	SMC0,SMC1							
PS0-PS7	SMC2,SMC3							

Exceptions for MB91FV360GA, MB91F362GB, MB91F369GA:

If the PFR is set to "1" the following pins do not automatically enter Hi-Z state in Stop/RTC mode when HIZX="1" or in Hardware standby:

Resource	Port	Affected pins	Comment
I^2C	M	SDA,SCL	Only if I^2C bus is driving active low
SIO	N	SOT4,SCK4,SOT3,SCK3	SCK3,SCK4 only if they are configured to be output
CAN	P	TX0,TX1,TX2,TX3	
UART	Q	SOT2,SOT1,SOT0	
SMC	R,S	PWM1P0 to PWM1P3, PWM2P0 to PWM2P3, PWM1M0 to PWM1M3, PWM2M0 to PWM2M3	Only if Hiz-mode is not set in PWM1/2 select registers

APPENDIX D INSTRUCTIONS

This appendix lists the FR50 series instructions. First, the appendix describes the following items to help you interpret the instruction list.

- **How to read the instruction list**
- **Addressing mode symbols**
- **FR50 series instruction word format**

■ How to Read the Instruction List

Instruction operands are XXX src, dest (src → dest).

Mnemonic	Class	OP	Cycles	NZVC	Operation	Remarks
ADD Rj,Ri	A	A6	1	CCCC	Ri+Rj → Ri	
*ADD #s5,Ri	C’	A4	1	CCCC	Ri+s5 → Ri	—
.	.	.	.	.	.	—
.	.	.	.	.	.	
↓ (1)	↓ (2)	↓ (3)	↓ (4)	↓ (5)	↓ (6)	↓ (7)

(1) Specifies the instruction name.

An asterisk (*) on the left of the mnemonic indicates that the instruction is not a CPU instruction but is an extended instruction for which instructions are expanded or added by the assembler.

(2) The symbols represent the addressing modes that can be specified in operands.

See "Addressing Mode Symbols" for the symbol meanings.

(3)Indicates the class name of the instruction word format.

See "FR30 Family Instruction Word Format" for information about instruction word formats.

(4) Lists the hexadecimal value of the instruction code.

(5) Indicates the number of machine cycles.

a:Memory access cycles. Can be extended by the Ready function.

b:Memory access cycles. Can be extended by the Ready function.

However, if the next instruction references the register operated on by an LD operation, an interlock occurs and the number of execution cycles increases by one.

c:If the next instruction reads or writes to R15, SSP, or USP or is an instruction of instruction word format A, an interlock is applied and the cycle count is incremented by 1, resulting in two cycles.

d:If the next instruction references MDH or MDL, an interlock occurs and the number of execution cycles increases by one (for a total of two cycles).

The minimum values for a, b, c, and d are one cycle.

(6)Indicates flag changes.

Flag change	
C	Change
–	No change
0	Clear
1	Set

Flag meanings	
N	Nagative flag
Z	Zero flag
V	Over flag
C	Carry flag

(7)Indicates the operation of the instruction.

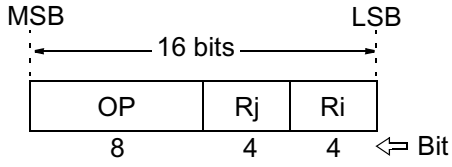
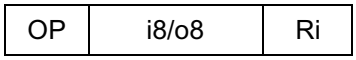
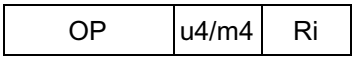
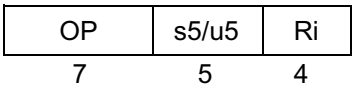
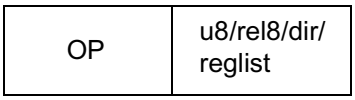
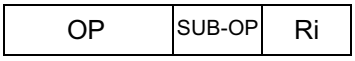
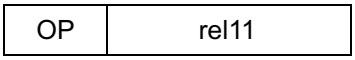
■ Addressing Mode Symbols

Table Da Explanation of Addressing Mode Symbols

Symbol	Meaning
Ri	Register direct (R0 to R15, AC, FP, SP)
Rj	Register direct (R0 to R15, AC, FP, SP)
R13	Register direct (R13, AC)
ps	Register direct (program status register)
Rs	Register direct (TBR, RP, SSP, USP, MDH, MDL)
CRi	Register direct (CR0 to CR15)
CRj	Register direct (CR0 to CR15)
#i8	Unsigned 8-bit immediate (–128 to 255), –128 to –1 are treated as 128 to 255.
#i20	Unsigned 20-bit immediate (–0X80000 to 0XFFFFF), –0X7FFFF to –1 are treated as 0X7FFFF to 0XFFFFF.
#i32	Unsigned 32-bit immediate (–0X80000000 to 0XFFFFFFFF), –0X80000000 to –1 are treated as 0X80000000 to 0XFFFFFFFF.
#s5	Signed 5-bit immediate (–16 to 15)
#s10	Signed 10-bit immediate (–512 to 508, only multiples of 4)
#u4	Unsigned 4-bit immediate (0 to 15)
#u5	Unsigned 5-bit immediate (0 to 31)
#u8	Unsigned 8-bit immediate (0 to 255)
#u10	Unsigned 10-bit immediate (0 to 1024, only multiples of 4)
@dir8	Unsigned 8-bit direct address (0 to 0XFF)
@dir9	Unsigned 9-bit direct address (0 to 0X1FE, only multiples of 2)
@dir10	Unsigned 10-bit direct address (0 to 0X3FC, only multiples of 4)
label9	Signed 9-bit branch address (–0X100 to 0XFC, only multiples of 2)
label12	Signed 12-bit branch address (–0X800 to 0X7FC, only multiples of 2)
label20	Signed 20-bit branch address (–0X80000 to 0X7FFFF)
label32	Signed 32-bit branch address (–0X80000000 to 0X7FFFFFFF)
@Ri	Register indirect (R0 to R15, AC, FP, SP)
@Rj	Register indirect (R0 to R15, AC, FP, SP)
@(R13,Rj)	Register relative indirect (Rj: R0 to R15, AC, FP, SP)
@(R14,disp10)	Register relative indirect (disp10: –0X200 to 0X1FC, only multiples of 4)
@(R14,disp9)	Register relative indirect (disp9: –0X100 to 0XFE, only multiples of 2)
@(R14,disp8)	Register relative indirect (disp8: –0X80 to 0X7F)
@(R15,udisp6)	Register relative indirect (udisp6: 0 to 60, only multiples of 4)
@Ri+	Register indirect with post-increment (R0 to R15, AC, FP, SP)
@R13+	Register indirect with post-increment (R13, AC)
@SP+	Stack push
@-SP	Stack pop
(reglist)	Register list

■ FR50 series Instruction Word Format

Table Db FR50 series Instruction Word Format

Type	Instruction Word Format
A	
B	
C	
C’	Only the ADD, ADDN, CMP, LSL, LSR and ASR instructions 
D	
E	
F	

D.1 FR50 SERIES INSTRUCTIONS

This section lists the FR50 series instructions in the following order.

Table D.1a Addition and Subtraction Instructions

Mnemonic	Class	OP	Cycles	NZVC	Operation	Remarks
ADD Rj, Ri	A	A6	1	CCCC	Ri + Rj → Ri	
*ADD #s5, Ri	C'	A4	1	CCCC	Ri + s5 → Ri	In the assembler, the MSB is the sign.
ADD #u4, Ri	C	A4	1	CCCC	Ri + extu(i4)→ Ri	Zero extended
ADD2 #u4, Ri	C	A5	1	CCCC	Ri + extu(i4)→ Ri	Minus extended
ADDC Rj, Ri	A	A7	1	CCCC	Ri + Rj + c → Ri	Add with carry
ADDN Rj, Ri	A	A2	1	----	Ri + Rj → Ri	
*ADDN #s5, Ri	C'	A0	1	----	Ri + s5 → Ri	In the assembler, the MSB is the sign.
ADDN #u4, Ri	C	A0	1	----	Ri + extu(i4)→ Ri	Zero extended
ADDN2 #u4, Ri	C	A1	1	----	Ri + extu(i4)→ Ri	Minus extended
SUB Rj, Ri	A	AC	1	CCCC	Ri – Rj → Ri	
SUBC Rj, Ri	A	AD	1	CCCC	Ri – Rj – c → Ri	Subtract with carry
SUBN Rj, Ri	A	AE	1	----	Ri – Rj → Ri	

Table D.1b Comparison Instructions

Mnemonic	Class	OP	Cycles	NZVC	Operation	Remarks
CMP Rj, Ri	A	AA	1	CCCC	Ri – Rj	
*CMP #s5, Ri	C'	A8	1	CCCC	Ri – s5	In the assembler, the MSB is the sign.
CMP #u4, Ri	C	A8	1	CCCC	Ri – extu(i4)	Zero extended
CMP2 #u4, Ri	C	A9	1	CCCC	Ri – extu(i4)	Minus extended

Table D.1c Logical Instructions

Mnemonic	Class	OP	Cycles	NZVC	Operation	Remarks
AND Rj, Ri	A	82	1	CC--	Ri &= Rj	Word
AND Rj, @Ri	A	84	1+2a	CC--	(Ri) &= Rj	Word
ANDH Rj, @Ri	A	85	1+2a	CC--	(Ri) &= Rj	Halfword
ANDB Rj, @Ri	A	86	1+2a	CC--	(Ri) &= Rj	Byte
OR Rj, Ri	A	92	1	CC--	Ri | = Rj	Word
OR Rj, @Ri	A	94	1+2a	CC--	(Ri) | = Rj	Word
ORH Rj, @Ri	A	95	1+2a	CC--	(Ri) | = Rj	Halfword
ORB Rj, @Ri	A	96	1+2a	CC--	(Ri) | = Rj	Byte
EOR Rj, Ri	A	9A	1	CC--	Ri ^ = Rj	Word
EOR Rj, @Ri	A	9C	1+2a	CC--	(Ri) ^ = Rj	Word
EORH Rj, @Ri	A	9D	1+2a	CC--	(Ri) ^ = Rj	Halfword
EORB Rj, @Ri	A	9E	1+2a	CC--	(Ri) ^ = Rj	Byte

Table D.1d Bit Manipulation Instructions

Mnemonic	Class	OP	Cycles	NZVC	Operation	Remarks
BANDL #u4, @Ri	C	80	1+2a	----	(Ri)&=(0xF0+u4)	Operates on the lower 4 bits.
BANDH #u4, @Ri	C	81	1+2a	----	(Ri)&=((u4<<4)+0x0F)	Operates on the upper 4 bits.
*BAND #u8, @Ri (Note 1)				----	(Ri)&=u8	
BORL #u4, @Ri	C	90	1+2a	----	(Ri) | = u4	Operates on the lower 4 bits.
BORH #u4, @Ri	C	91	1+2a	----	(Ri) | = (u4<<4)	Operates on the upper 4 bits.
*BOR #u8, @Ri (Note 2)				----	(Ri) | = u8	
BEORL #u4, @Ri	C	98	1+2a	----	(Ri) ^ = u4	Operates on the lower 4 bits.
BEORH #u4, @Ri	C	99	1+2a	----	(Ri) ^ = (u4<<4)	Operates on the upper 4 bits.
*BEOR #u8, @Ri (Note 3)				----	(Ri) ^ = u8	
BTSTL #u4, @Ri	C	88	2+a	0C--	(Ri) & u4	Tests the lower 4 bits.
BTSTH #u4, @Ri	C	89	2+a	CC--	(Ri) & (u4<<4)	Tests the upper 4 bits.

Notes:1) The assembler generates BANDL if a bit is set in u8&0x0F, and BANDH if a bit is set in u8&0xF0. In some cases, both BANDL and BANDH are generated.

2) The assembler generates BORL if a bit is set in u8&0x0F, and BORH if a bit is set in u8&0xF0. In some cases, both BORL and BORH are generated.

3) The assembler generates BEORL if a bit is set in u8&0x0F, and BEORH if a bit is set in u8&0xF0. In some cases, both BEORL and BEORH are generated.

Table D.1e Multiplication and Division Instructions

Mnemonic	Class	OP	Cycles	NZVC	Operation	Remarks
MUL Rj,Ri	A	AF	5	CCC-	Ri * Rj → MDH,MDL	32bit*32bit=64bit
MULU Rj,Ri	A	AB	5	CCC-	Ri * Rj → MDH,MDL	Unsigned
MULH Rj,Ri	A	BF	3	CC--	Ri * Rj → MDL	16bit*16it=32bit
MULUH Rj,Ri	A	BB	3	CC--	Ri * Rj → MDL	Unsigned
DIV0S Ri	E	97-4	1	----		Step operation
DIV0U Ri	E	97-5	1	----		32bit/32bit=32bit
DIV1 Ri	E	97-6	d	-C-C		
DIV2 Ri	E	97-7	1	-C-C		
DIV3	E	9F-6	1	----	MDL / Ri → MDL , MDL % Ri → MDH	
DIV4S	E	9F-7	1	----	MDL / Ri → MDL , MDL % Ri → MDH	
*DIV Ri (Note 1)			36	-C-C		
*DIVU Ri (Note 2)			33	-C-C		

Notes:1) Generates, DIV0S, DIV1 × 32, DIV2, DIV3, and DIV4S. The length of the instruction code is 72 bytes.

2) Generates DIV0U and DIV1 × 32. The length of the instruction code is 66 bytes.

Table D.1f Shift Instructions

Mnemonic	Class	OP	Cycles	NZVC	Operation	Remarks
LSL Rj, Ri	A	B6	1	CC-C	Ri << Rj → Ri	Logical shift
*LSL #u5, Ri (u5:0 to 31)	C'	B4	1	CC-C	Ri << u5 → Ri	
LSL #u4, Ri	C	B4	1	CC-C	Ri << u4 → Ri	
*LSL2 #u4, Ri	C	B5	1	CC-C	Ri << (u4+16) → Ri	
LSR Rj, Ri	A	B2	1	CC-C	Ri >> Rj → Ri	Logical shift
*LSR #u5, Ri (u5:0 to 31)	C'	B0	1	CC-C	Ri >> u5 → Ri	
LSR #u4, Ri	C	B0	1	CC-C	Ri >> u4 → Ri	
*LSR2 #u4, Ri	C	B1	1	CC-C	Ri >> (u4+16) → Ri	
ASR Rj, Ri	A	BA	1	CC-C	Ri >> Rj → Ri	Arithmetic shift
*ASR #u5, Ri (u5:0 to 31)	C'	B8	1	CC-C	Ri >> u5 → Ri	
ASR #u4, Ri	C	B8	1	CC-C	Ri >> u4 → Ri	
*ASR2 #u4, Ri	C	B9	1	CC-C	Ri >> (u4+16) → Ri	

Table D.1g Immediate Set, 16-bit, and 32-bit Immediate Transfer Instructions

Mnemonic	Class	OP	Cycles	NZVC	Operation	Remarks
LDI:32 #i32, Ri	E	9F-8	3	----	i32 → Ri	
LDI:20 #i20, Ri	C	9B	2	----	i20 → Ri	Upper 12 bits are zero extended.
LDI:8 #i8, Ri	B	C0	1	----	i8 → Ri	
*LDI #{i8 | i20 | i32}, Ri (Note)					{i8 | i20 | i32} → Ri	Upper 24 bits are zero extended.

Note: If the immediate value is an absolute value, the assembler automatically selects i8, i20, or i32. If the immediate value is a relative value or includes an external reference symbol, the assembler selects i32.

Table D.1h Memory Load Instructions

Mnemonic	Class	OP	Cycles	NZVC	Operation	Remarks
LD @Rj, Ri	A	04	b	----	(Rj)→Ri	
LD @(R13,Rj), Ri	A	00	b	----	(R13+Rj)→Ri	
LD @(R14,disp10), Ri	B	20	b	----	(R14+disp10)→Ri	
LD @(R15,udisp6), Ri	C	03	b	----	(R15+udisp6)→Ri	
LD @R15+, Ri	E	07-0	b	----	(R15)→Ri,R15+=4	
LD @R15+, Rs	E	07-8	b	----	(R15)→Rs,R15+=4	Rs: Special register
LD @R15+, PS	E	07-9	1+a+b	CCCC	(R15)→PS,R15+=4	(Note)
LDUH @Rj, Ri	A	05	b	----	(Rj)→Ri	Zero extended
LDUH @(R13,Rj), Ri	A	01	b	----	(R13+Rj)→Ri	Zero extended
LDUH @(R14,disp9), Ri	B	40	b	----	(R14+disp9)→Ri	Zero extended
LDUB @Rj, Ri	A	06	b	----	(Rj)→Ri	Zero extended
LDUB @(R13,Rj), Ri	A	02	b	----	(R13+Rj)→Ri	Zero extended
LDUB @(R14,disp8), Ri	B	60	b	----	(R14+disp8)→Ri	Zero extended

Note: The assembler calculates values to set in the o8 and o4 fields in the hardware specifications as follows:
disp10/4→o8, disp9/2→o8, disp8→o8 disp10, disp9, disp8 are signed.
udisp6/4→o4 udisp6 is unsigned.

Table D.1i Memory Store Instructions

Mnemonic	Class	OP	Cycles	NZVC	Operation	Remarks
ST Ri, @Rj	A	14	a	----	Ri→(Rj)	Word
ST Ri, @(R13,Rj)	A	10	a	----	Ri→(R13+Rj)	Word
ST Ri, @(R14,disp10)	B	30	a	----	Ri→(R14+disp10)	Word
ST Ri, @(R15,udisp6)	C	13	a	----	Ri→(R15+udisp6)	
ST Ri, @–R15	E	17-0	a	----	R15–=4,Ri→(R15)	
ST Rs, @–R15	E	17-8	a	----	R15–=4,Rs→(R15)	Rs: Special register
ST PS, @–R15	E	17-9	a	----	R15–=4,PS→(R15)	(Note)
STH Ri, @Rj	A	15	a	----	Ri→(Rj)	Halfword
STH Ri, @(R13,Rj)	A	11	a	----	Ri→(R13+Rj)	Halfword
STH Ri, @(R14,disp9)	B	50	a	----	Ri→(R14+disp9)	Halfword
STB Ri, @Rj	A	16	a	----	Ri→(Rj)	Byte
STB Ri, @(R13,Rj)	A	12	a	----	Ri→(R13+Rj)	Byte
STB Ri, @(R14,disp8)	B	70	a	----	Ri→(R14+disp8)	Byte

Note: The assembler calculates values to set in the o8 and o4 fields in the hardware specifications as follows.
disp10/4→o8, disp9/2→o8, disp8→o8 disp10, disp9, disp8 are signed.
udisp6/4→o4 udisp6 is unsigned.

Table D.1j Register-to-Register Transfer Instructions

Mnemonic	Class	OP	Cycles	NZVC	Operation	Remarks
MOV Rj, Ri	A	8B	1	----	Rj → Ri	General-purpose register-to-register transfer
MOV Rs, Ri	A	B7	1	----	Rs → Ri	Rs: Special register (Note)
MOV Ri, Rs	A	B3	1	----	Ri → Rs	Rs: Special register (Note)
MOV PS, Ri	E	17-1	1	----	PS → Ri	
MOV Ri, PS	E	07-1	c	CCCC	Ri → PS	

Note: Special register Rs: TBR, RP, USP, SSP, MDH, MDL

Table D.1k Standard Branch Instructions (No Delay)

Mnemonic		Class	OP	Cycles	NZVC	Operation	Remarks
JMP	@Ri	E	97-0	2	----	Ri → PC	
CALL	label12	F	D0	2	----	PC+2→RP ,PC+2+(label12–PC–2)→PC	
CALL	@Ri	E	97-1	2	----	PC+2→RP, Ri→PC	
RET		E	97-2	2	----	RP → PC	Return
INT	#u8	D	1F	3+3a	----	SSP–=4,PS→(SSP),SSP–=4,PC+2→(SSP), 0→I flag, 0 → S flag, (TBR+0x3FC–u8×4)→PC	
INTE		E	9F-3	3+3a	----	SSP–=4,PS→(SSP),SSP–=4,PC+2→(SSP), 0→S flag, (TBR+0x3D8)→PC	Used by the emulator
RETI		E	97-3	2+2a	CCCC	(R15)→PC,R15+=4,(R15)→PS,R15+=4	
BRA	label9	D	E0	2	---	PC+2+(label9–PC–2)→PC	
BNO	label9	D	E1	1	---	No branch	
BEQ	label9	D	E2	2/1	---	if(Z==1) then PC+2+(label9–PC–2)→PC	
BNE	label9	D	E3	2/1	---	↑s/Z==0	
BC	label9	D	E4	2/1	---	↑s/C==1	
BNC	label9	D	E5	2/1	---	↑s/C==0	
BN	label9	D	E6	2/1	---	↑s/N==1	
BP	label9	D	E7	2/1	---	↑s/N==0	
BV	label9	D	E8	2/1	---	↑s/V==1	
BNV	label9	D	E9	2/1	---	↑s/V==0	
BLT	label9	D	EA	2/1	---	↑s/V xor N==1	
BGE	label9	D	EB	2/1	---	↑s/V xor N==0	
BLE	label9	D	EC	2/1	---	↑s/(V xor N) or Z==1	
BGT	label9	D	ED	2/1	---	↑s/(V xor N) or Z==0	
BLS	label9	D	EE	2/1	---	↑s/C or Z==1	
BHI	label9	D	EF	2/1	---	↑s/C or Z==0	

Notes:

- 2/1 in the number of cycles column means two cycles when a branch occurs and 1 cycle when no branch occurs.
- The assembler calculates values to set in the rel11 and rel8 fields in the hardware specifications as follows:

  (label12 – PC – 2)/2 → rel11, (label9 – PC – 2)/2 → rel8 label12 and label9 are signed.

Table D.1l Delayed Branch Instructions

Mnemonic	Class	OP	Cycles	NZVC	Operation	Remarks
JMP:D @Ri	E	9F-0	1	----	Ri → PC	
CALL:D label12	F	D8	1	----	PC+4→RP ,PC+2+(label12–PC–2)→PC	
CALL:D @Ri	E	9F-1	1	----	PC+4→RP, Ri→PC	
RET:D	E	9F-2	1	----	RP → PC	Return
BRA:D label9	D	F0	1	----	PC+2+(label9–PC–2)→PC	
BNO:D label9	D	F1	1	----	No branch	
BEQ:D label9	D	F2	1	----	if(Z==1) then PC+2+(label9–PC–2)→PC	
BNE:D label9	D	F3	1	----	↑s/Z==0	
BC:D label9	D	F4	1	----	↑s/C==1	
BNC:D label9	D	F5	1	----	↑s/C==0	
BN:D label9	D	F6	1	----	↑s/N==1	
BP:D label9	D	F7	1	----	↑s/N==0	
BV:D label9	D	F8	1	----	↑s/V==1	
BNV:D label9	D	F9	1	----	↑s/V==0	
BLT:D label9	D	FA	1	----	↑s/V xor N==1	
BGE:D label9	D	FB	1	----	↑s/V xor N==0	
BLE:D label9	D	FC	1	----	↑s/(V xor N) or Z==1	
BGT:D label9	D	FD	1	----	↑s/(V xor N) or Z==0	
BLS:D label9	D	FE	1	----	↑s/C or Z==1	
BHI:D label9	D	FF	1	----	↑s/C or Z==0	

Notes:

- The assembler calculates values to set in the rel11 and rel8 fields in the hardware specifications as follows.
- (label12 – PC – 2)/2 → rel11, (label9 – PC – 2)/2 → rel8 label12 and label9 are signed.
- Delayed branch instructions always branch after execution of the next instruction (delay slot).
- One cycle instructions can be placed in the delay slot. Multi-cycle instructions cannot be placed in the delay slot.

Table D.1m Other Instructions

Mnemonic	Class	OP	Cycles	NZVC	Operation	Remarks
NOP	E	9F-A	1	----	No change occurs.	
ANDCCR #u8 ORCCR #u8	D D	83 93	c c	CCCC CCCC	CCR and u8 → CCR CCR or u8 → CCR	
STILM #u8	D	87	1	----	i8 → ILM	ILM immediate set
ADDSP #s10	D	A3	1	----	R15 += s10	ADD SP instruction (Note 1)
EXTSB Ri	E	97-8	1	----	Sign extended 8→32bit	
EXTUB Ri	E	97-9	1	----	Zero extended 8→32bit	
EXTSH Ri	E	97-A	1	----	Sign extended 16→32bit	
EXTUH Ri	E	97-B	1	----	Zero extended 16→32bit	
LDM0 (reglist)	D	8C		----	(R15)→reglist, R15 increment	Multi-load R0 to R7
LDM1 (reglist)	D	8D		----	(R15)→reglist, R15 increment	Multi-load R8 to R15
*LDM (reglist)				----	(R15)→reglist, R15 increment	Multi-load R0 to R15 (Note 2)
STM0 (reglist)	D	8E		----	R15 decrement reglist→(R15)	Multi-store R0 to R7
STM1 (reglist)	D	8F		----	R15 decrement reglist→(R15)	Multi-store R8 to R15
*STM (reglist)				----	R15 decrement reglist→(R15)	Multi-store R0 to R15 (Note 3)
ENTER #u10	D	0F	1+a	----	R14 → (R15 – 4), R15 – 4 → R14, R15 – u10 → R15	Function entry processing (Note 4)
LEAVE	E	9F-9	b	----	R14 + 4 → R15, (R15 – 4) → R14	Function exit processing
XCHB @Rj, Ri	A	8A	2a	----	Ri → TEMP (Rj) → Ri TEMP → (Rj)	For semaphore control Byte data

Notes: 1) The assembler calculates s10/4 and sets as an s8 value. s10 is signed.

2) • Generates LDM0 if reglist contains a specification for any of R0 to R7, and generates LDM1 if reglist contains a specification for any of R8 to R15. LDM0 and LDM1 may both be generated.

• The number of execution cycles for LDM0 (reglist) and LDM1 (reglist) depends on the number of specified registers n, as follows:
a*(n–1)+b+1 cycles

3) • Generates STM0 if reglist contains a specification for any of R0 to R7, and generates STM1 if reglist contains a specification for any of R8 to R15. STM0 and STM1 may both be generated.

• The number of execution cycles for STM0 (reglist) and STM1 (reglist) depends on the number of specified registers n, as follows:
a*n+1 cycles

4) The assembler calculates u10/4 and sets as a u8 value. u10 is unsigned.

Table D.1n 20-bit Standard Branch Macro-instructions

Mnemonic		Operation	Remarks
*CALL20	label20,Ri	Address of next instruction → RP, label20 → PC	Ri: Temporary register (Note 1)
*BRA20	label20,Ri	label20 → PC	Ri: Temporary register (Note 2)
*BEQ20	label20,Ri	if(Z==1) then label20→PC	Ri: Temporary register (Note 3)
*BNE20	label20,Ri	↑ s/Z==0	↑
*BC20	label20,Ri	↑ s/C==1	↑
*BNC20	label20,Ri	↑ s/C==0	↑
*BN20	label20,Ri	↑ s/N==1	↑
*BP20	label20,Ri	↑ s/N==0	↑
*BV20	label20,Ri	↑ s/V==1	↑
*BNV20	label20,Ri	↑ s/V==0	↑
*BLT20	label20,Ri	↑ s/V xor N==1	↑
*BGE20	label20,Ri	↑ s/V xor N==0	↑
*BLE20	label20,Ri	↑ s/(V xor N) or Z==1	↑
*BGT20	label20,Ri	↑ s/(V xor N) or Z==0	↑
*BLS20	label20,Ri	↑ s/C or Z==1	↑
*BHI20	label20,Ri	↑ s/C or Z==0	↑

Note 1: CALL20

- The following instruction is generated when label20 – PC – 2 is between – 0x800 and +0x7fe.

```
CALL    label12
```

- The following instruction is generated when label20 – PC – 2 is outside the above range or contains an external reference symbol.

```
LDI:20  #label20,Ri
CALL    @Ri
```

Note 2: BRA20

- The following instruction is generated when label20 – PC – 2 is between – 0x100 and +0xfe.

```
BRA     label9
```

- The following instruction is generated when label20 – PC – 2 is outside the above range or contains an external reference symbol.

```
LDI:20  #label20,Ri
JMP     @Ri
```

Note 3: Bcc20

- The following instruction is generated when label20 – PC – 2 is between – 0x100 and +0xfe.

```
Bcc     label9
```

- The following instructions are generated when label20 – PC – 2 is outside the above range or contains an external reference symbol.

```
Bxcc    false   xcc is the inverse condition of cc

LDI:20  #label20,Ri
JMP     @Ri

false:
```

Table D.1o 20-bit Delayed Branch Macro-instructions

Mnemonic	Operation	Remarks
*CALL20:D label20,Ri	Address of next instruction + 2 → RP, label20 → PC	Ri: Temporary register (Note 1)
*BRA20:D label20,Ri	label20→PC	Ri: Temporary register (Note 2)
*BEQ20:D label20,Ri	if(Z==1) then label20→PC	Ri: Temporary register (Note 3)
*BNE20:D label20,Ri	↑ s/Z==0	↑
*BC20:D label20,Ri	↑ s/C==1	↑
*BNC20:D label20,Ri	↑ s/C==0	↑
*BN20:D label20,Ri	↑ s/N==1	↑
*BP20:D label20,Ri	↑ s/N==0	↑
*BV20:D label20,Ri	↑ s/V==1	↑
*BNV20:D label20,Ri	↑ s/V==0	↑
*BLT20:D label20,Ri	↑ s/V xor N==1	↑
*BGE20:D label20,Ri	↑ s/V xor N==0	↑
*BLE20:D label20,Ri	↑ s/(V xor N) or Z==1	↑
*BGT20:D label20,Ri	↑ s/(V xor N) or Z==0	↑
*BLS20:D label20,Ri	↑ s/C or Z==1	↑
*BHI20:D label20,Ri	↑ s/C or Z==0	↑

Note 1: CALL20:D

- The following instruction is generated when label20 – PC – 2 is between – 0x800 and +0x7fe.

```
CALL:D  label12
```

- The following instructions are generated when label20 – PC – 2 is outside the above range or contains an external reference symbol.

```
LDI:20  #label20,Ri
CALL:D  @Ri
```

Note 2: BRA20:D

- The following instruction is generated when label20 – PC – 2 is between – 0x100 and +0xfe.

```
BRA:D   label9
```

- The following instructions are generated when label20 – PC – 2 is outside the above range or contains an external reference symbol.

```
LDI:20  #label20,Ri
JMP:D   @Ri
```

Note 3: Bcc20:D

- The following instruction is generated when label20 – PC – 2 is between – 0x100 and +0xfe.

```
Bcc:D   label9
```

- The following instructions are generated when label20 – PC – 2 is outside the above range or contains an external reference symbol.

```
Bxcc    false   xcc is the inverse condition of cc
LDI:20  #label20,Ri
JMP:D   @Ri

false:
```

Table D.1p 32-bit Standard Branch Macro-instructions

Mnemonic		Operation	Remarks
*CALL32	label32,Ri	Address of next instruction → RP, label32 → PC	Ri: Temporary register (Note 1)
*BRA32	label32,Ri	label32→PC	Ri: Temporary register (Note 2)
*BEQ32	label32,Ri	if(Z==1) then label32→PC	Ri: Temporary register (Note 3)
*BNE32	label32,Ri	↑ s/Z==0	↑
*BC32	label32,Ri	↑ s/C==1	↑
*BNC32	label32,Ri	↑ s/C==0	↑
*BN32	label32,Ri	↑ s/N==1	↑
*BP32	label32,Ri	↑ s/N==0	↑
*BV32	label32,Ri	↑ s/V==1	↑
*BNV32	label32,Ri	↑ s/V==0	↑
*BLT32	label32,Ri	↑ s/V xor N==1	↑
*BGE32	label32,Ri	↑ s/V xor N==0	↑
*BLE32	label32,Ri	↑ s/(V xor N) or Z==1	↑
*BGT32	label32,Ri	↑ s/(V xor N) or Z==0	↑
*BLS32	label32,Ri	↑ s/C or Z==1	↑
*BHI32	label32,Ri	↑ s/C or Z==0	↑

Note 1: CALL32

- The following instruction is generated when label32 – PC – 2 is between – 0x800 and +0x7fe.

```
CALL    label12
```

- The following instructions are generated when label32 – PC – 2 is outside the above range or contains an external reference symbol.

```
LDI:32  #label32,Ri
CALL    @Ri
```

Note 2: BRA32

- The following instruction is generated when label32 – PC – 2 is between – 0x100 and +0xfe.

```
BRA     label9
```

- The following instructions are generated when label32 – PC – 2 is outside the above range or contains an external reference symbol.

```
LDI:32  #label32,Ri
JMP     @Ri
```

Note 3: Bcc32

- The following instruction is generated when label32 – PC – 2 is between – 0x100 and +0xfe.

```
Bcc     label9
```

- The following instructions are generated when label32 – PC – 2 is outside the above range or contains an external reference symbol.

```
Bxcc    false   xcc is the inverse condition of cc
LDI:32  #label32,Ri
JMP     @Ri

false:
```

Table D.1q 32-bit Delayed Branch Macro-instructions

Mnemonic	Operation	Remarks
*CALL32:D label32,Ri	Address of next instruction+2 → RP, label32 → PC	Ri: Temporary register (Note 1)
*BRA32:D label32,Ri	label32 → PC	Ri: Temporary register (Note 2)
*BEQ32:D label32,Ri	if(Z==1) then label32 → PC	Ri: Temporary register (Note 3)
*BNE32:D label32,Ri	↑ s/Z==0	↑
*BC32:D label32,Ri	↑ s/C==1	↑
*BNC32:D label32,Ri	↑ s/C==0	↑
*BN32:D label32,Ri	↑ s/N==1	↑
*BP32:D label32,Ri	↑ s/N==0	↑
*BV32:D label32,Ri	↑ s/V==1	↑
*BNV32:D label32,Ri	↑ s/V==0	↑
*BLT32:D label32,Ri	↑ s/V xor N==1	↑
*BGE32:D label32,Ri	↑ s/V xor N==0	↑
*BLE32:D label32,Ri	↑ s/(V xor N) or Z==1	↑
*BGT32:D label32,Ri	↑ s/(V xor N) or Z==0	↑
*BLS32:D label32,Ri	↑ s/C or Z==1	↑
*BHI32:D label32,Ri	↑ s/C or Z==0	↑

Note 1: CALL32:D

- The following instruction is generated when label32 – PC – 2 is between – 0x800 and +0x7fe.

```
CALL:D  label12
```

- The following instructions are generated when label32 – PC – 2 is outside the above range or contains an external reference symbol.

```
LDI:32  #label32,Ri
CALL:D  @Ri
```

Note 2: BRA32:D

- The following instruction is generated when label32 – PC – 2 is between – 0x100 and +0xfe.

```
BRA:D   label9
```

- The following instructions are generated when label32 – PC – 2 is outside the above range or contains an external reference symbol.

```
LDI:32  #label32,Ri
JMP:D   @Ri
```

Note 3: Bcc32:D

- The following instruction is generated when label32 – PC – 2 is between – 0x100 and +0xfe.

```
Bcc:D   label9
```

- The following instructions are generated when label32 – PC – 2 is outside the above range or contains an external reference symbol.

```
Bxcc    false   xcc is the inverse condition of cc
LDI:32  #label32,Ri
JMP:D   @Ri

false:
```

Table D.1r Direct Addressing Instructions

Mnemonic	Class	OP	Cycles	NZVC	Operation	Remarks
DMOV @dir10, R13	D	08	b	----	(dir10) → R13	Word
DMOV R13, @dir10	D	18	a	----	R13 → (dir10)	Word
DMOV @dir10, @R13+	D	0C	2a	----	(dir10) → (R13),R13+=4	Word
DMOV @R13+, @dir10	D	1C	2a	----	(R13) → (dir10),R13+=4	Word
DMOV @dir10, @-R15	D	0B	2a	----	R15–=4,(R15) → (dir10)	Word
DMOV @R15+, @dir10	D	1B	2a	----	(R15) → (dir10),R15+=4	Word
DMOVH @dir9, R13	D	09	b	----	(dir9) → R13	Halfword
DMOVH R13, @dir9	D	19	a	----	R13 → (dir9)	Halfword
DMOVH @dir9, @R13+	D	0D	2a	----	(dir9) → (R13),R13+=2	Halfword
DMOVH @R13+, @dir9	D	1D	2a	----	(R13) → (dir9),R13+=2	Halfword
DMOVB @dir8, R13	D	0A	b	----	(dir8) → R13	Byte
DMOVB R13, @dir8	D	1A	a	----	R13 → (dir8)	Byte
DMOVB @dir8, @R13+	D	0E	2a	----	(dir8) → (R13),R13++	Byte
DMOVB @R13+, @dir8	D	1E	2a	----	(R13) → (dir8),R13++	Byte

- The assembler calculates and sets the dir8, dir9, and dir10 field values as follows.
  dir8 → dir, dir9/2 → dir, dir10/4 → dir dir8, dir9, and dir10 are unsigned.

Table D.1s Peripheral Instructions

Mnemonic	Class	OP	Cycles	NZVC	Operation	Remarks
LDRES @Ri+,#u4	C	BC	a	----	(Ri) → u4 peripheral Ri+=4	u4: Channel number
STRES #u4, @Ri+	C	BD	a	----	u4 peripheral → (Ri) Ri+=4	u4: Channel number

INDEX

W

Z

CM71-10104-6EA

FUJITSU SEMICONDUCTOR • CONTROLLER MANUAL

FR50

32-BIT MICROCONTROLLER

MB91360 Series

HARDWARE MANUAL

November 2002, the sixth edition

Published **FUJITSU LIMITED** Electronic Devices

Edited Technical Information Dept. / EMDC