zilog

Embedded in Life

An IXYS Company

ZNEO32! Family of Microcontrollers

Z32F1281 MCU

Product Specification

PS034504-0617

PRELIMINARY

ZNEO32!

32 Bit Microcontrollers

Warning: DO NOT USE THIS PRODUCT IN LIFE SUPPORT SYSTEMS.

LIFE SUPPORT POLICY

ZILOG'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS PRIOR WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF ZILOG CORPORATION.

As used herein

Life support devices or systems are devices which (a) are intended for surgical implant into the body, or (b) support or sustain life and whose failure to perform when properly used in accordance with instructions for use provided in the labeling can be reasonably expected to result in a significant injury to the user. A critical component is any component in a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system or to affect its safety or effectiveness.

Document Disclaimer

 Information in this publication concerning the devices, applications, or technology described is intended to suggest possible uses and may be superseded. ZILOG, INC. DOES NOT ASSUME LIABILITY FOR OR PROVIDE A REPRESENTATION OF ACCURACY OF THE INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED IN THIS DOCUMENT. ZILOG ALSO DOES NOT ASSUME LIABILITY FOR INTELLECTUAL PROPERTY INFRINGEMENT RELATED IN ANY MANNER TO USE OF INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED HEREIN OR OTHERWISE. The information contained within this document has been verified according to the general principles of electrical and mechanical engineering.

ZNEO32! is a trademark or registered trademark of Zilog, Inc. All other product or service names are the property of their respective owners.

Revision History

Each instance in this document's revision history reflects a change from its previous edition. For more details, refer to the corresponding page(s) or appropriate links furnished in the table below.

Date	Revision Level	Description	Page
Jun 2017	04	Updated part numbers to include the Cortex M identifier.	All
May 2016	03	Added Quadrature Encoder Interface information.	122
Apr 2016	02	Added timing information for peripherals; global edits for clarity.	All
Nov 2015	01	Original issue.	

1. Overview

Introduction

Zilog's Z32F1281 MCU, a member of the ZNEO32! Family of microcontrollers is a cost-effective and high-performance 32-bit microcontroller. The Z32F1281 MCU provides 3-phase PWM generator units which are suitable for inverter bridges, including motor drive systems. The two built-in channels of these generators control two inverter motors simultaneously.

Three 12-bit high speed ADC units with 16-channel analog multiplexed inputs are included to gather information from the motor. The Z32F1281 MCU can control up to two inverter motors or one inverter motor and the Power Factor Correction (PFC) function simultaneously. Four on-chip operational AMPs and four analog comparators are available to measure analog input signals. The operational amplifier can amplify the input signal to the proper signal range and transfer it to the ADC input channel. The comparator monitors external signals and helps create an internal emergency signal. Multiple powerful external serial interface engines communicate with on-board sensors.

Figure 1.1 shows a block diagram of the Z32F1281 MCU.

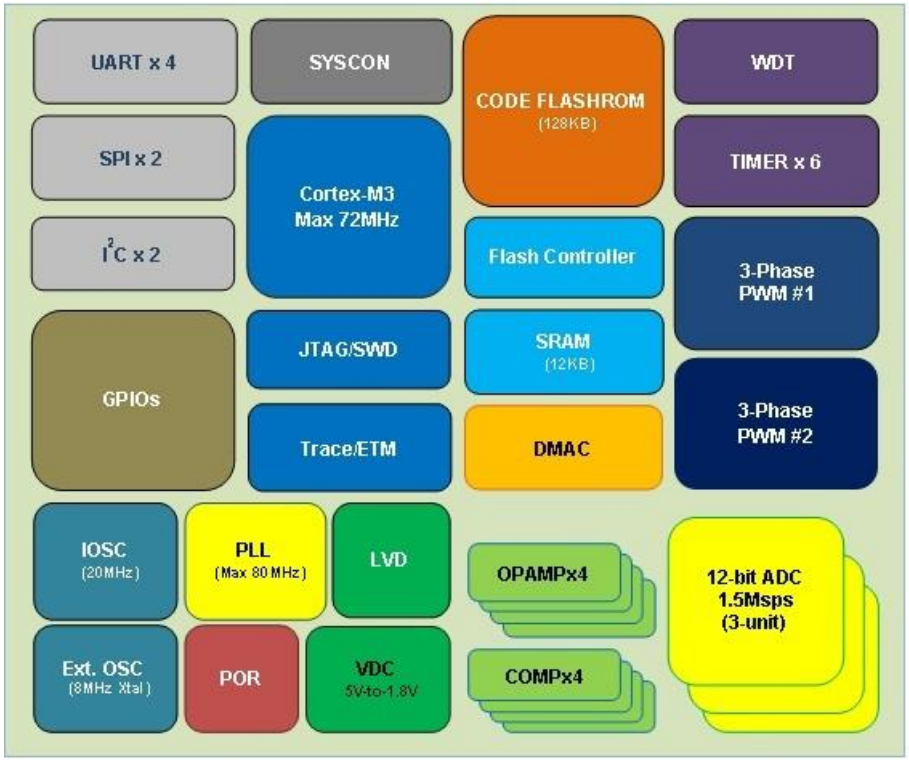


Figure 1.1. Z32F1281 MCU Block Diagram

Figure 1.2 and Figure 1.3 show the pin layouts.

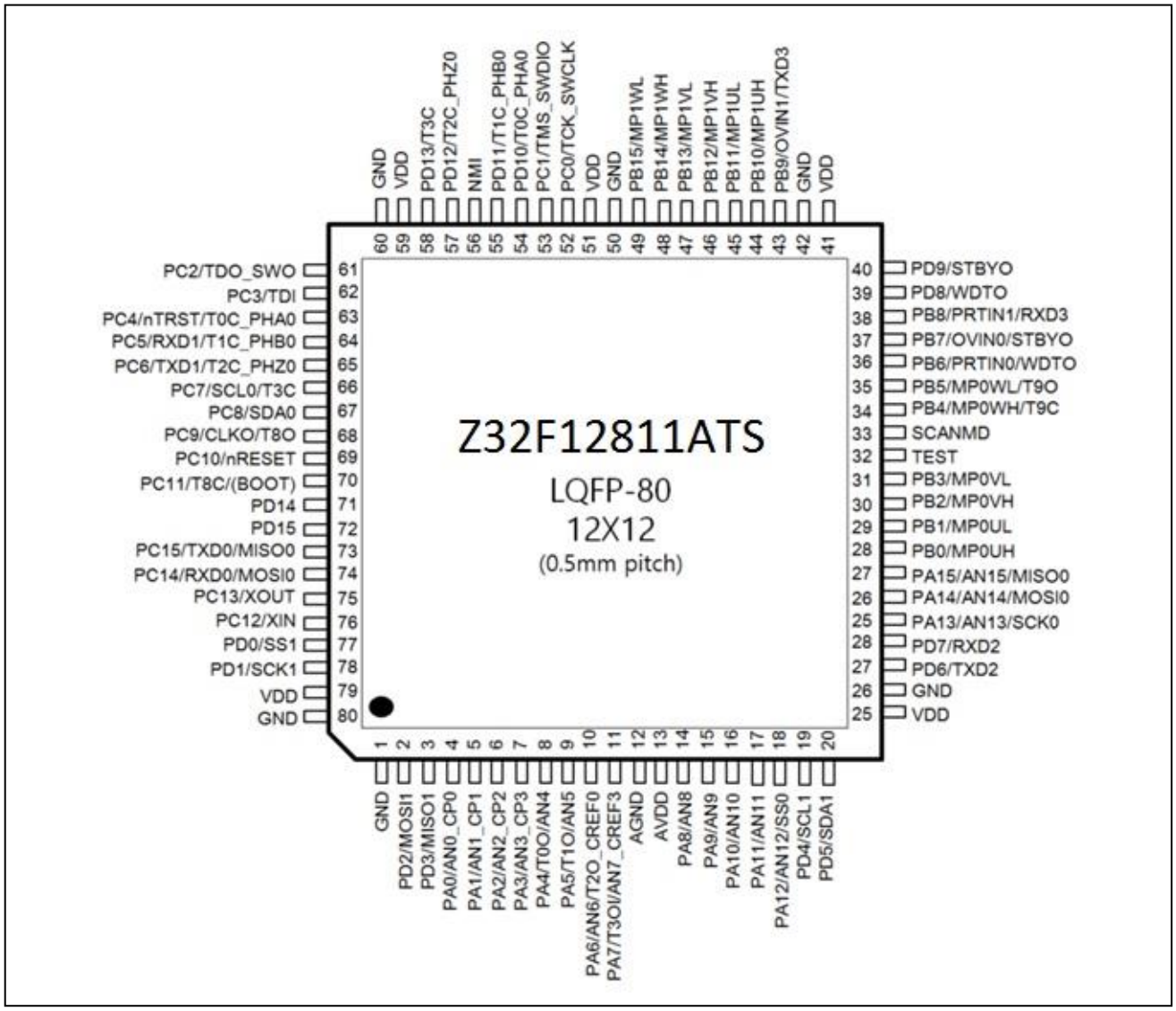


Figure 1.2. Pin Layout (LQFP-80)

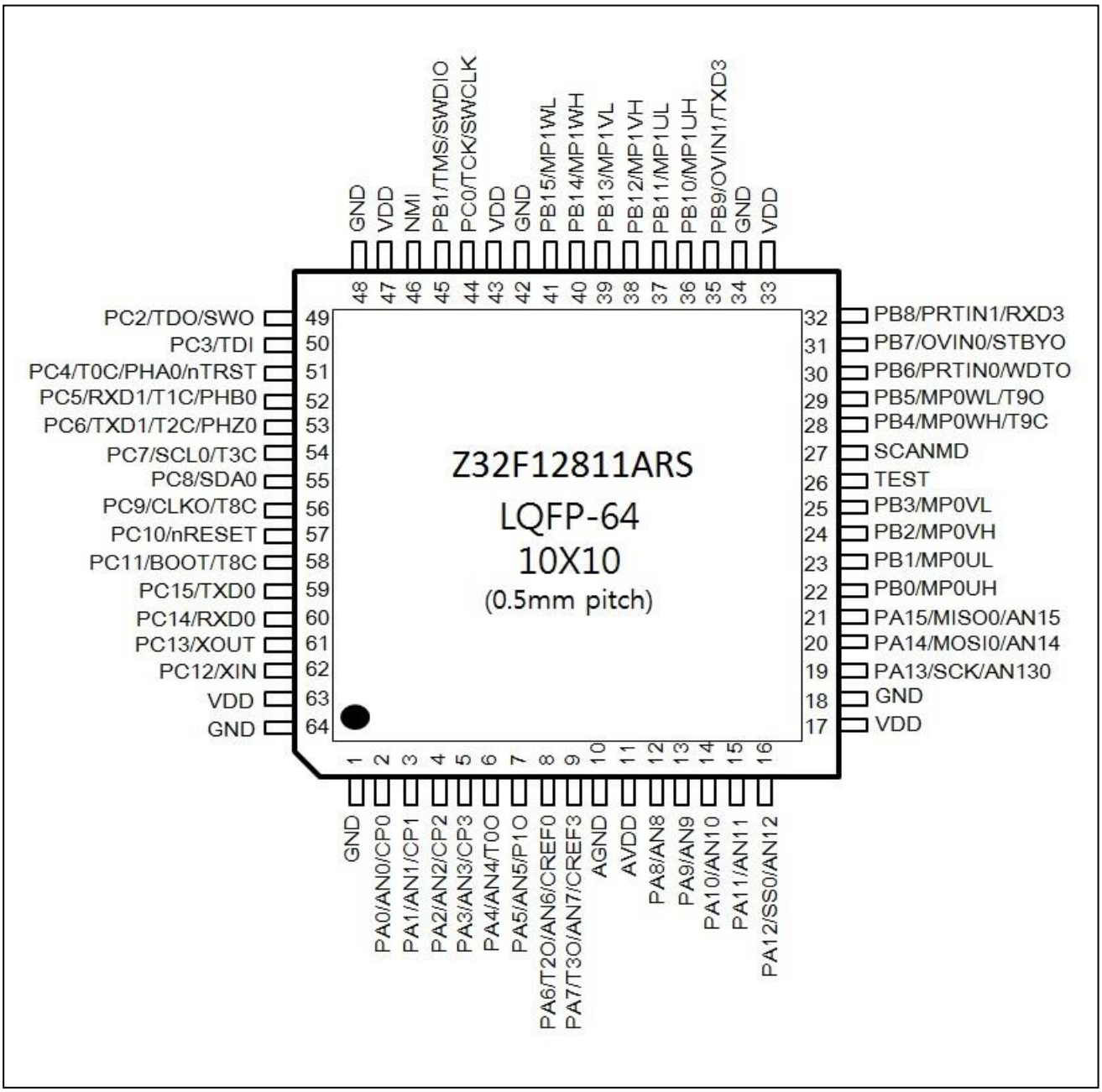


Figure 1.3. Pin Layout (LQFP-64)

Product Features

The Z32F1281 MCU offers the following features:

- High performance low-power Cortex-M3 core
- 128 KB code Flash memory with cache function
- 12 KB SRAM
- 3-Phase Motor PWM with ADC triggering function
  - 2 channels
- 1.5Msps high-speed ADC with burst conversion function
  - 2 or 3 units with 16 channel input
- Built-in Programmable Gain Amplifier (PGA) for ADC inputs
  - 4 channels
    - 3 channels for 3 shunt resistor configuration
    - 1 channel for 1 shunt resistor configuration
- Built-in analog comparator
  - 4 channels
    - 3 channels for 3 shunt resistor configuration
    - 1 channel for 1 shunt resistor configuration
- System fail-safe function by clock monitoring
- XTAL OSC fail monitoring
- Precision internal oscillator clock (20MHz ±3%)
- Watchdog timer
- Six general purpose timers
- Quadrature encoder interface counter
- External communication ports: 4 UARTs, 2 I^2Cs, 2 SPIs
- High current driving port for UART photo couplers
- Debug and emergency stop function
- Real-time monitoring function support for more effective development
- JTAG and Serial Wire Debug (SWD) in-circuit debugger
- Various memory size and package options
  - LQFP-80, LQFP-64
- Industrial grade operating temperature (-40 ~ +85℃)

Table 1.1. Device Type

Part Number	Flash	SRAM	UART	SPI	I2C	MPWM	ADC	I/O PORT	PKG
Z32F12811ATS	128KB	12KB	4	2	2	2	3-unit 16 ch	68	LQFP-80
Z32F12811ARS			2	2	1	2		48	LQFP-64

Architecture

Block Diagram

An internal block diagram of the Z32F1281 MCU is shown in Figure 1.4.

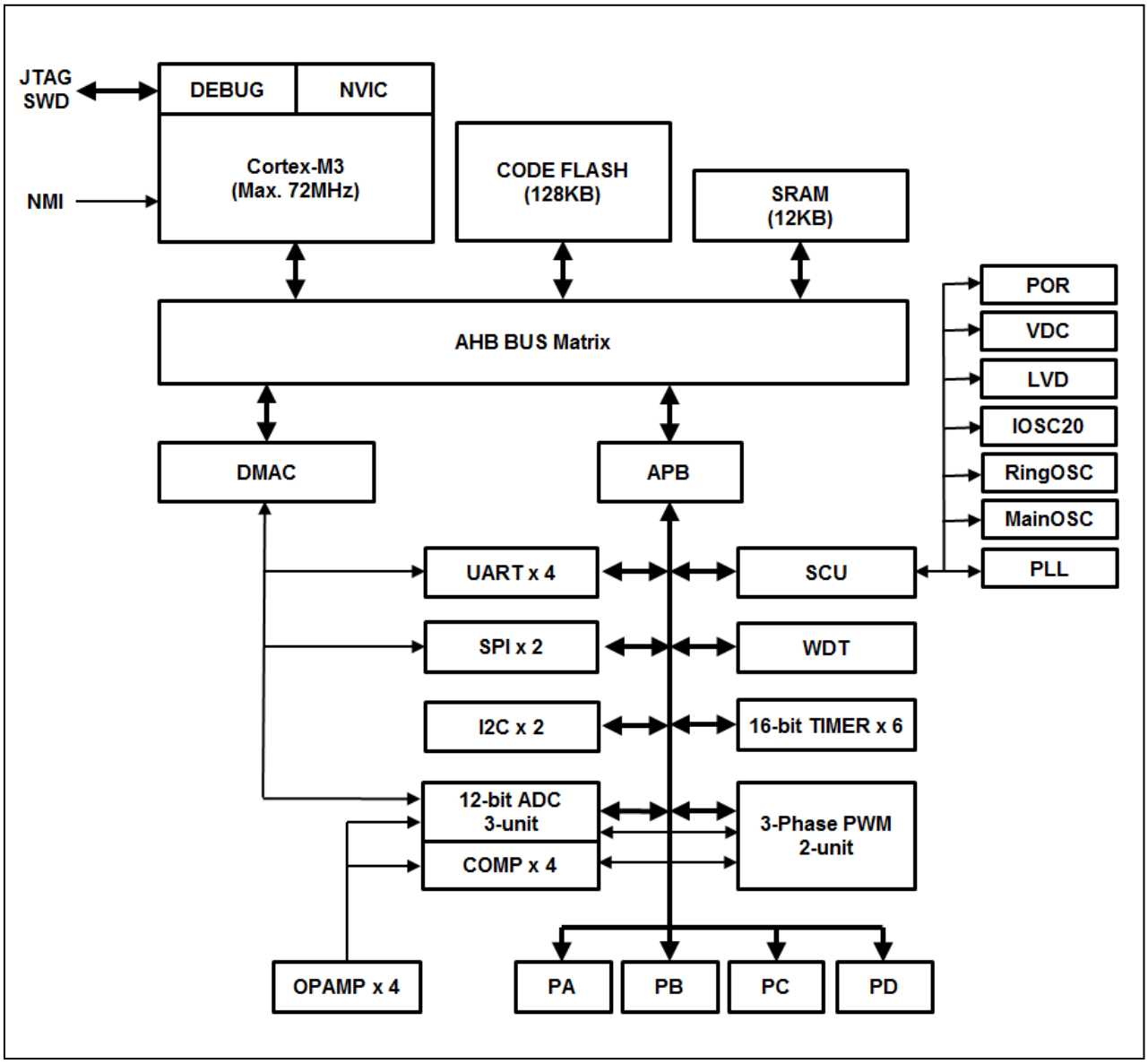


Figure 1.4. Internal Block Diagram

Functional Description

The following section provides an overview of the features of the Z32F1281 microcontroller.

ARM Cortex-M3

- ARM powered Cortex-M3 Core based on v7M architecture, which is optimized for small size and low-power systems. On core system timer (SYSTICK) provides a simple 24-bit timer that makes it easy to manage the system operations
- Thumb-compatible Thumb-2 only instruction set processor core makes code high-density
- Hardware division and single-cycle multiplication
- Integrated Nested Vectored Interrupt Controller (NVIC) provides deterministic interrupt handling
- Full featured debug solutions – JTAG and SWD, FPB, DWT, ITM, and TPIU
- Maximum 72 MHz operating frequency with zero wait execution

Nested Vector-Interrupt Controller (NVIC)

- The ARM Nested Vectored Interrupt Controller (NVIC) on the ARM Cortex-M3 core handles all internal and external exceptions. When an interrupt condition is detected, the processor state is automatically stored to the stack and automatically restored from the stack at the end of the interrupt service routine.
- The vector is fetched in parallel to the state saving, which enables efficient interrupt entry.
- The processor supports tail-chaining, which enables back-to-back interrupts to be performed without the overhead of state saving and restoring.

128 KB Internal Code Flash Memory

- The Z32F1281 MCU provides internal 128 KB code Flash memory and its controller. This is enough to program the motor algorithm and control the system. Self-programming is available and ISP and JTAG programming is also supported in Boot or Debugging Mode.
- Instruction and data cache buffer overcome the limitations of low-bandwidth Flash memory. The CPU can execute from Flash memory with zero wait state up to 72 MHz bus frequency.

12 KB Zero-wait Internal SRAM

- On chip 12 KB zero-wait SRAM can be used for working memory space and program code can be loaded on this SRAM.

Boot Logic

- The smart boot logic supports Flash programming. The Z32F1281 MCU can be entered by external boot pin and UART and SPI programming are available in Boot Mode. UART0 or SPI0 is used in Boot mode communication.

System Control Unit (SCU)

- The SCU block manages internal power, clock, reset and operation mode. It also controls analog blocks (INTOSC, VDC and LVD).

32-bit Watchdog Timer (WDT)

- The watchdog timer performs the system monitoring function. It generates an internal reset or interrupt to notice an abnormal status of the system.

Multi-purpose 16-bit Timer

- Six-channel 16-bit general purpose timers support:
  - Periodic timer mode
  - Counter mode
  - PWM mode
  - Capture mode

PWM Generator

- Two channels of the 3-phase PWM generator are implemented. 16 bit up/down counter with prescaler supports triangular and saw tooth waveforms.
- The PWM generates an internal ADC trigger signal to measure the signal on time.
- Dead time insertion and emergency stop functionality ensure that the chip and system operate under safe conditions.

Serial Peripheral Interface (SPI)

- Synchronous serial communication is provided by the SPI block. The Z32F1281 MCU has 2 channel SPI modules. The DMA function is supported by the DMA controller. Transfer data is moved to/from the memory area without CPU operation.
- Boot mode uses this SPI block to download the Flash program.

Inter-Integrated Circuit Interface (I^2C)

- The Z32F1281 MCU has a 2-channel I^2C block and it supports up to 400 kHz I^2C communication. Master and the slave modes are supported.

Universal Asynchronous Receiver/Transmitter (UART)

- The Z32F1281 MCU includes a 4-channel UART block. For accurate baud rate control, a fractional baud rate generator is provided.
- The DMA function is supported by the DMA controller. Transfer data is moved to/from memory area without CPU operation.

General PORT I/Os

- 16-bit PA, PB, PC, PD ports are available and provide multiple functionality:
- General I/O port
- Independent bit set/clear function
- External interrupt input port
- Pull-up/Open-drain
- On chip debounce Filter

12-bit Analog-to-Digital Converter (ADC)

- 3 built-in ADCs can convert analog signal up to 1usec conversion rate. 16-channel analog mux and OP-AMP provides various combinations from external analog signals.

Analog Front End (AFE)

- Operational Amplifier (OPAMP)
  - 4 built-in OPAMPs amplify analog signals up to x8.74 gain
- Analog Comparator (COMP)
  - 4 built-in analog comparators

Pin Description

The pin configurations are shown in Table 1.2. 16 pins are reserved for power/ground pair and dedicated pins.

Table 1.2. Pin Description

Pin Name		Type	Description		Remark
LQFP80	LQFP64				
79	63	VDD	P	VDD	
80	64	GND	P	Ground	
1	1	GND	P	Ground	
2	-	PD2	IOUS	PORT D Bit 2 Input/Output	
		MOSI1	I/O	SPI Channel 1 Master Out / Slave In	
3	-	PD3*	IOUS	PORT D Bit 3 Input/Output	
		MISOI1	I/O	SPI Channel 1 Master In / Slave Out	
4	2	PA0*	IOUS	PORT A Bit 0 Input/Output	
		AN0	IA	Analog Input 0	
		COMP0	IA	Comparator 0 Input	
5	3	PA1*	IOUS	PORT A Bit 1 Input/Output	
		AN1	IA	Analog Input1	
		COMP1	IA	Comparator 1 Input	
6	4	PA2*	IOUS	PORT A Bit 2 Input/Output	
		AN2	IA	Analog Input 2	
		COMP2	IA	Comparator 2 Input	
7	5	PA3*	IOUS	PORT A Bit 3 Input/Output	
		AN3	IA	Analog Input 3	
		COMP3	IA	Comparator 3 Input	
8	6	PA4*	IOUS	PORT A Bit 4 Input/Output	
		T0O	Output	Timer 0 Output	
		AN4	IA	Analog Input 4	
9	7	PA5*	IOUS	PORT A Bit 5 Input/Output	
		T1O	Output	Timer 1 Output	
		AN5	IA	Analog Input 5	
10	8	PA6*	IOUS	PORT A Bit 6 Input/Output	
		T2O	Output	Timer 2 Output	
		AN6	IA	Analog Input 6	
		CREF0	IA	Comparator 0 Reference Input	
11	9	PA7*	IOUS	PORT A Bit 7 Input/Output	
		TRACED3	Output	ETM Trace Data 3	
		T3O	Output	Timer 3 Output	
		AN7	IA	Analog Input 7	
		CREF1	IA	Comparator 1 Reference Input	
12	10	AGND	P	Analog Ground	
13	11	AVDD	P	Analog VDD	
14	12	PA8*	IOUS	PORT A Bit 8 Input/Output	
		TRACECLK	Output	ETM Trace Clock	
		AD0O	Output	ADC0 Start Signal	
		AN8	IA	Analog Input 8	
15	13	PA9*	IOUS	PORT A Bit 9 Input/Output	
		TRACED0	Output	ETM Trace Data 0	
		AD1O	Output	ADC1 Start Signal	
		AN9	IA	Analog Input 9	
16	14	PA10*	IOUS	PORT A Bit 10 Input/Output	
		TRACED1	Output	ETM Trace Data 1	
		AD2O	Output	ADC2 Start Signal	
		AN10	IA	Analog Input 10	
17	15	PA11*	IOUS	PORT A Bit 11 Input/Output	
		TRACED2	Output	ETM Trace Data 2	

		AN11	IA	Analog Input 11	
18	16	PA12*	IOUS	PORT A Bit 12 Input/Output	
		SS0	I/O	SPI0 Slave Select signal	
		AD2I	Input	ADC2 Start Input signal	
		AN12	IA	Analog Input 12	
19	-	PD4	IOUS	PORT D Bit 4 Input/Output	
		SCL1	Output	I^2C Channel 1 SCL In/Out	
20	-	PD5	IOUS	PORT D Bit 5 Input/Output	
		SDA1	Output	I^2C Channel 1 SDA In/Out	
21	17	VDD	P	VDD	
22	18	GND	P	Ground	
23	-	PD6*	IOUS	PORT D Bit 6 Input/Output	
		TXD2	Output	UART Channel 2 TxD Input	
		AD0I	Input	ADC0 Start Input signal	
24	-	PD7*	IOUS	PORT D Bit 7 Input/Output	
		RXD2	Input	UART Channel 2 RxD Input	
		AD1I	Input	ADC1 Start Input signal	
25	19	PA13*	IOUS	PORT A Bit 13 Input/Output	
		SCK0	I/O	SPI0 Data Clock Input/Output	
		AN13	IA	Analog Input 13	
26	20	PA14*	IOUS	PORT A Bit 14 Input/Output	
		MOSI0	I/O	SPI0 Master-Output/Slave-Input Data signal	
		AN14	IA	Analog Input 14	
27	21	PA15*	IOUS	PORT A Bit 15 Input/Output	
		MISO0	I/O	SPI0 Master-Input/Slave-Output Data signal	
		AN15	IA	Analog Input 15	
28	22	PB0	IOUS	PORT B Bit 0 Input/Output	
		PWM0H0	Output	PWM0 H0 Output	
29	23	PB1	IOUS	PORT B Bit 1 Input/Output	
		PWM0L0	Output	PWM0 L0 Output	
30	24	PB2	IOUS	PORT B Bit 0 Input/Output	
		PWM0H1	Output	PWM0 H1 Output	
31	25	PB3	IOUS	PORT B Bit 1 Input/Output	
		PWM0L1	Output	PWM0 L1 Output	
32	26	TEST	Input	Test-mode Input (Always tied ‘L’)	Pull-down
33	27	SCANMD	Input	Scan-mode Input (Always tied ‘L’)	Pull-down
34	28	PB4	IOUS	PORT B Bit 4 Input/Output	
		PWM0H2	Output	PWM0 H2 Output	
		T9C	I/O	Timer 9 Clock/Capture Input	
35	29	PB5	IOUS	PORT B Bit 5 Input/Output	
		PWM0L2	Output	PWM0 L2 Output	
		T9O	I/O	Timer 9 Output	
36	30	PB6	IOUS	PORT B Bit 6 Input/Output	
		PRTIN0	Input	PWM0 Protection Input signal 0	
		WDTO	Output	WDT Output	
37	31	PB7	IOUS	PORT B Bit 7 Input/Output	
		OVIN0	Input	PWM0 Over-voltage put signal 1	
		STBYO	Output	Power-down mode indication signal	
38	32	PB8	IOUS	PORT B Bit 8 Input/Output	
		PRTIN1	Input	PWM1 Protection Input signal 0	
		RXD3	Input	UART3 RXD Input	

39	-	PD8	IOUS	PORT D Bit 8 Input/Output	
		WDTO	Output	WDT Output	
30	-	PD9	IOUS	PORT D Bit 9 Input/Output	
		STBYO	Output	Power-down mode indication signal	
41	33	VDD	P	VDD	
42	34	GND	P	Ground	
43	35	PB9	IOUS	PORT B Bit 9 Input/Output	
		OVIN1	Input	PWM1 Over-voltage Input signal 1	
		TXD3	Output	UART3 TXD Output	
44	36	PB10	IOUS	PORT B Bit 10 Input/Output	
		PWM1H0	Output	PWM Channel 1 Phase 0 H-side Output	
45	37	PB11	IOUS	PORT B Bit 11 Input/Output	
		PWM1L0	Output	PWM Channel 1 Phase 0 L-side Output	
46	38	PB12	IOUS	PORT B Bit 12 Input/Output	
		PWM1H1	Output	PWM Channel 1 Phase 1 H-side Output	
47	39	PB13	IOUS	PORT B Bit 13 Input/Output	
		PWM1L1	Output	PWM Channel 1 Phase 1 L-side Output	
48	40	PB14	IOUS	PORT B Bit 14 Input/Output	
		PWM1H2	Output	PWM Channel 1 Phase 2 H-side Output	
49	41	PB15	IOUS	PORT B Bit 15 Input/Output	
		PWM1L2	Output	PWM Channel 1 Phase 2 L-side Output	
50	42	GND	P	Ground	
51	43	VDD	P	VDD	
52	44	PC0	IOUS	PORT C Bit 0 Input/Output	
		TCK/SWCK	Input	JTAG TCK, SWD Clock Input	
53	45	PC1	IOUS	PORT C Bit 1 Input/Output	
		TMS/SWDIO	I/O	JTAG TMS, SWD Data Input/Output	
54	-	PD10	IOUS	PORT D Bit 10 Input/Output	
		AD0SOC	Output	ADC0 Start-of-Conversion	
		T0C/PHA	Input	Timer 0 Clock/Capture/Phase-A Input	
55	-	PD11	IOUS	PORT D Bit 10 Input/Output	
		AD0EOC	Output	ADC0 End-of-Conversion	
		T1C/PHB	Input	Timer 1 Clock/Capture/Phase-B Input	
56	46	NMI	Input	Non-maskable Interrupt Input	
57	-	PD12	IOUS	PORT D Bit 12 Input/Output	
		AD1SOC	Output	ADC1 Start-of-Conversion	
		T2C/PHZ0	Input	Timer 2 Clock/Capture/Phase-Z Input	
58	-	PD13	IOUS	PORT D Bit 13 Input/Output	
		AD1EOC	Output	ADC1 End-of-Conversion	
		T3C	Input	Timer 3 Clock/Capture Input	
59	47	VDD	P	VDD	
60	48	GND	P	Ground	
61	49	PC2	IOUS	PORT C Bit 2 Input/Output	
		TDO/SWO	Output	JTAG TDO, SWO Output	
62	50	PC3	IOUS	PORT C Bit 3 Input/Output	
		TDI	Input	JTAG TDI Input	
63	51	PC4	IOUS	PORT C Bit 4Input/Output	
		nTRST	Input	JTAG nTRST Input	
		T0C/PHA	Input	Timer 0 Clock/Capture/Phase-A Input	
64	52	PC5	IOUS	PORT C Bit 5Input/Output	
		RXD1	Input	UART1 RXD Input	
		T1C/PHB	Input	Timer 1 Clock/Capture/Phase-B Input	
65	53	PC6	IOUS	PORT C Bit 6Input/Output	

		TXD1	Output	UART1 TXD Output	
		T2C/PHZ	Input	Timer 2 Clock/Capture/Phase-Z Input	
66	54	PC7	IOUS	PORT C Bit 7Input/Output	
		SCL0	Output	I^2C Channel 0 SCL In/Out	
		T3C	Input	Timer 3 Clock/Capture input	
67	55	PC8	IOUS	PORT C Bit 8 Input/Output	
		SDA0	Output	I^2C Channel 0 SDA In/Out	
68	56	PC9	IOUS	PORT C Bit 9 Input/Output	
		CLKO	Output	System Clock Output	
		T8O	Output	Timer 8 Output	
69	57	PC10	IOUS	PORT C Bit 10 Input/Output	
		nRESET	Input	External Reset Input	Pull-up
70	58	PC11	IOUS	PORT C Bit 11 Input/Output	
		BOOT	Input	Boot mode Selection Input	
		T8C	Input	Timer 8 Clock/Capture Input	
71	-	PD14	IOUS	PORT D Bit 14 Input/Output	
		AD2SOC	Output	ADC2 Start-of-Conversion Output signal	
72	-	TD15	IOUS	PORT D Bit 15 Input/Output	
		AD2EOC	Output	ADC2 Start-of-Conversion Output signal	
73	59	PC15	IOUS	PORT C Bit 14 Input/Output	
		TXD0	Output	UART0 TXD Output	
		MISO0	I/O	SPI0 Master-Input/Slave-Output	
74	60	PC14	IOUS	PORT C Bit 14 Input/Output	
		RXD0	Input	UART0 RXD Input	
		MOSI0	I/O	SPI0 Master-Output/Slave-Input	
		VMARGIN	OA	Not used. (test purpose)	
75	61	PC13	IOUS	PORT C Bit 13 Input/Output	
		XOUT	OA	External Crystal Oscillator Output	
76	62	PC12	IOUS	PORT C Bit 12 Input/Output	
		XIN	IA	External Crystal Oscillator Input	
77	-	PD0	IOUS	PORT D Bit 0 Input/Output	
		SS1	I/O	SPI1 Slave Select	
78	-	PD1	IOUS	PORT D Bit 1 Input/Output	
		SCK1	I/O	SPI1 Clock Input/Output	

*Notation: I=Input, O=Output, U=Pull-up, D=Pull-down,
S=Schmitt-Trigger Input Type, C=CMOS Input Type, A=Analog, P=Power
(*) Selected pin function after reset condition
Pin order may be changed with revision notice

Memory Map

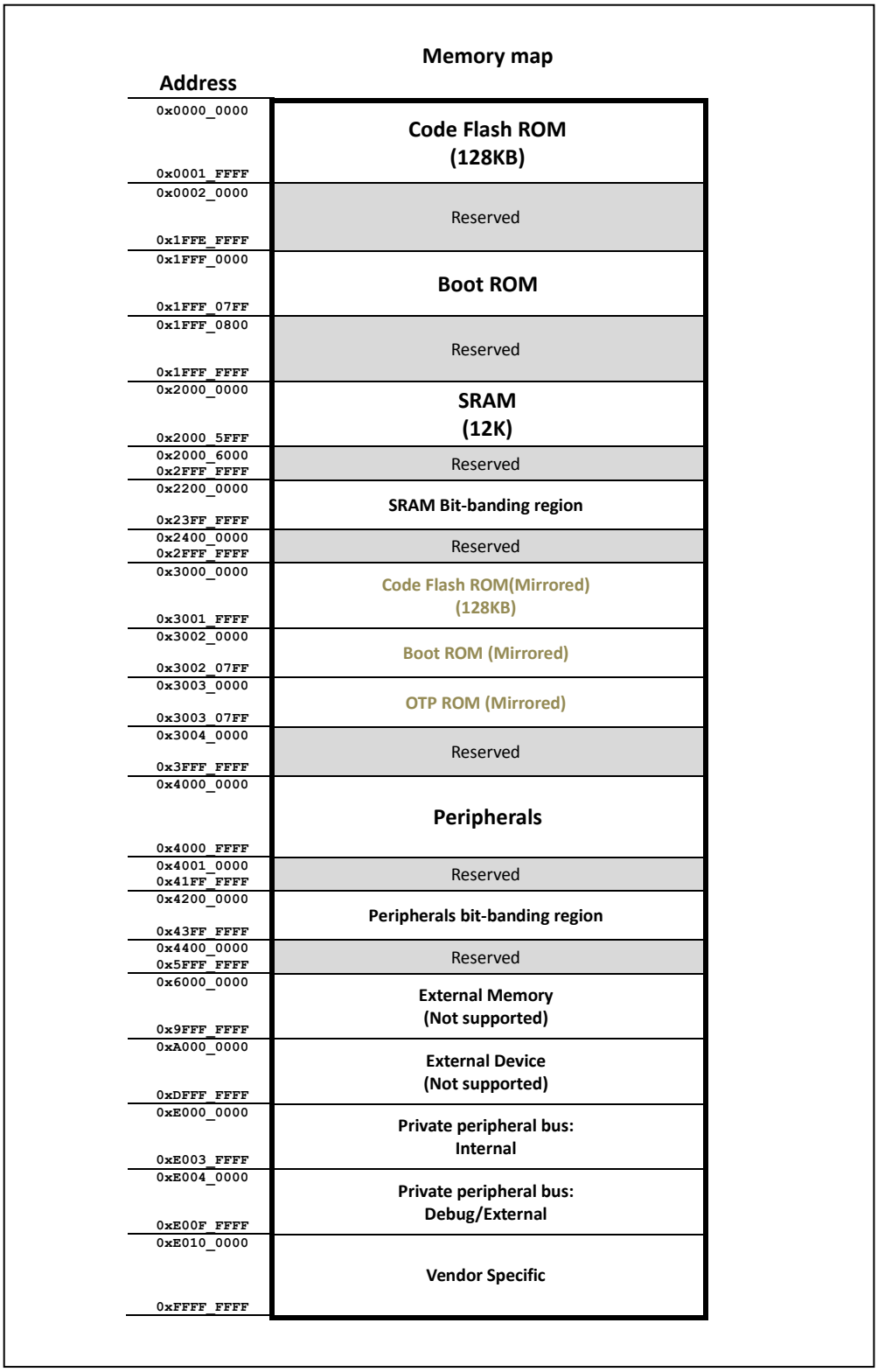


Figure 1.5. Main Memory Map

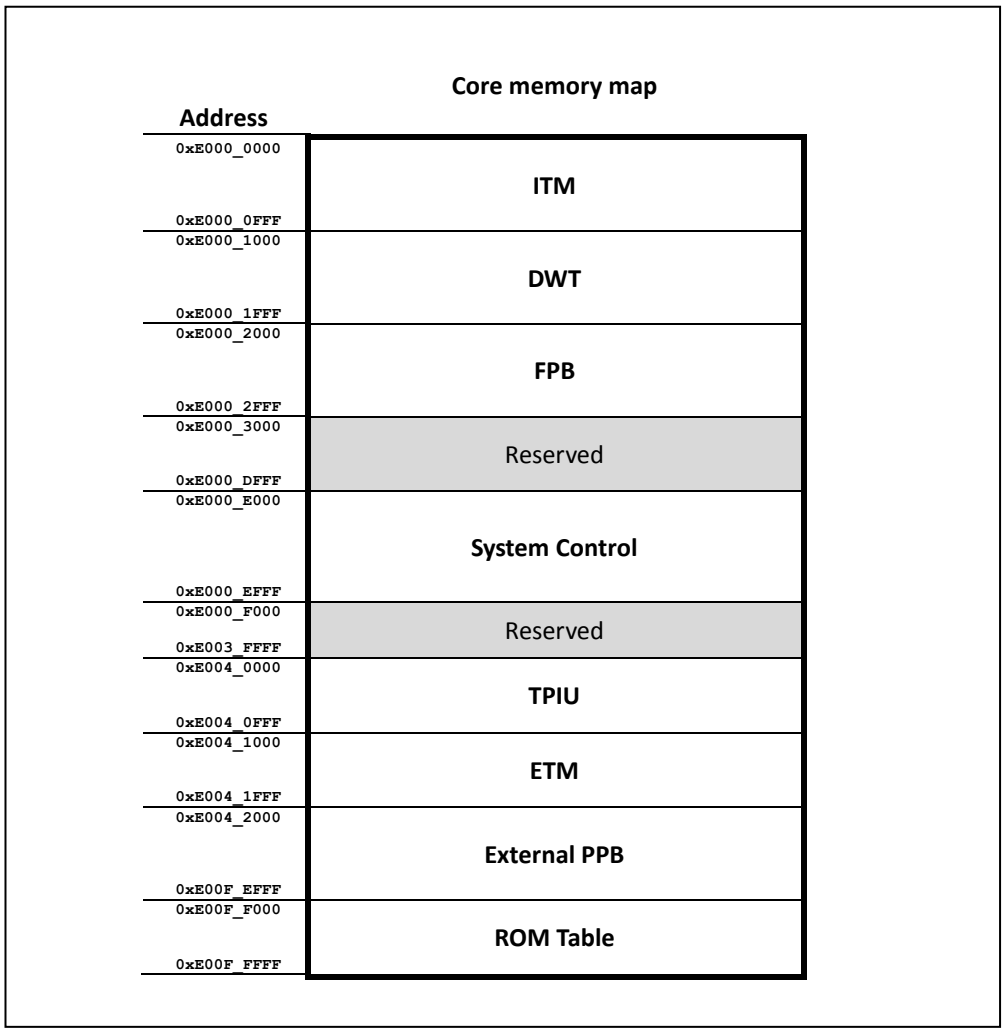


Figure 1.6. Cortex-M3 Private Memory Map

Note: For more information about the memory maps, refer to document number DDI337 from ARM.

Address	Peripheral map
0x4000_0000	SCU
0x4000_0100	FMC
0x4000_0200	WDT
0x4000_0300	Reserved
0x4000_0400	DMAC(15)
0x4000_0500	Reserved
0x4000_1000	PCU
0x4000_2000	GPIO(A,B,C,D)
0x4000_3000	TIMER(6)
0x4000_4000	MPWM0
0x4000_5000	MPWM1
0x4000_6000	Reserved
0x4000_8000	UART0
0x4000_8100	UART1
0x4000_8200	UART2
0x4000_8300	UART3
0x4000_8600	Reserved
0x4000_9000	SPI0
0x4000_9100	SPI1
0x4000_9200	Reserved
0x4000_A000	I^2C0
0x4000_A100	I^2C1
0x4000_A200	Reserved
0z4000_B000	ADC0
0x4000_B100	ADC1
0x4000_B200	ADC2
0x4000_B300	AFE
0x4000_B400 0x4000_FFFF	Reserved

Figure 1.7. Peripheral Memory Map

2. CPU

Cortex-M3 Core

The CPU core is supported by the ARM Cortex-M3 processor which provides a high-performance, low-cost platform. For more information about Cortex-M3, refer to document number DDI337 from ARM.

System Timer

The System Timer (SYSTICK) is a 24-bit timer and is part of the Cortex-M3 core. The system timer can be configured either through the registers (see the Cortex-M3 Technical Reference Manual) or through the provided functions defined in `core_cm3.h`. There is an interrupt vector for the system timer. To configure the system timer, call `SysTickConfig()` with the number of system clocks in between interrupt intervals (up to a maximum of 24 bits).

Interrupt Controller

The Nested Vectored Interrupt Controller (NVIC) is part of the core Cortex-M3 MCU. The NVIC controls system exceptions and peripheral interrupts and is closely coupled with the core to provide low latency and efficient processing of late arriving interrupts. The NVIC maintains knowledge of the nested interrupts to enable tail-chaining of interrupts.

The Z32F1281 MCU supports 64 peripheral interrupts (of which 25 are not used) and 16 system interrupts. The NVIC also allows setting software interrupts and resetting the system.

Interrupts can be assigned a PRIORITY GROUP (common interrupts with the same priorities) as well as individual priorities. There are 8 priority levels available. For an interrupt to be active, it must be enabled in the peripheral and the NVIC registers. For more information on NVIC, see the Cortex M3 Technical Reference Manual.

The system includes functions to set the NVIC registers which are defined in `core_cm3.h`.

Table2.1. Interrupt Vector Map

Interrupt Number	Vector Address	Interrupt Source
-16	0x0000_0000	Stack Pointer
-15	0x0000_0004	Reset Address
-14	0x0000_0008	NMI Handler
-13	0x0000_000C	Hard Fault Handler
-12	0x0000_0010	MPU Fault Handler
-11	0x0000_0014	BUS Fault Handler
-10	0x0000_0018	Usage Fault Handler
-9	0x0000_001C	Reserved
-8	0x0000_0020	
-7	0x0000_0024	
-6	0x0000_0028	
-5	0x0000_002C	SVCall Handler
-4	0x0000_0030	Debug Monitor Handler
-3	0x0000_0034	Reserved
-2	0x0000_0038	PenSV Handler
-1	0x0000_003C	SysTick Handler
0	0x0000_0040	LVDDETECT
1	0x0000_0044	SCLKFAIL
2	0x0000_0048	XOSCFAIL
3	0x0000_004C	WDT
4	0x0000_0050	Reserved
5	0x0000_0054	TIMER0
6	0x0000_0058	TIMER1
7	0x0000_005C	TIMER2
8	0x0000_0060	TIMER3
9	0x0000_0064	Reserved
10	0x0000_0068	

11	0x0000_006C	
12	0x0000_0070	
13	0x0000_0074	TIMER8
14	0x0000_0078	TIMER9
15	0x0000_007C	Reserved
16	0x0000_0080	GPIOAE
17	0x0000_0084	GPIOAO
18	0x0000_0088	GPIOBE
19	0x0000_008C	GPIOBO
20	0x0000_0090	GPIOCE
21	0x0000_0094	GPIOCO
22	0x0000_0098	GPIODE
23	0x0000_009C	GPIODO
24	0x0000_00A0	MPWM0
25	0x0000_00A4	MPWM0PROT
26	0x0000_00A8	MPWM0OVV
27	0x0000_00AC	MPWM1
28	0x0000_00B0	MPWM1PROT
29	0x0000_00B4	MPWM1OVV
30	0x0000_00B8	Reserved
31	0x0000_00BC	Reserved
32	0x0000_00C0	SPI0
33	0x0000_00C4	SPI1
34	0x0000_00C8	Reserved
35	0x0000_00CC	
36	0x0000_00D0	I2C0
37	0x0000_00D4	I2C1
38	0x0000_00D8	UART0
39	0x0000_00DC	UART1
40	0x0000_00E0	UART2
41	0x0000_00E4	UART3
42	0x0000_00E8	Reserved
43	0x0000_00EC	ADC0
44	0x0000_00F0	ADC1
45	0x0000_00F4	ADC2
46	0x0000_00F8	COMP0
47	0x0000_00FC	COMP1
48	0x0000_0100	COMP2
49	0x0000_0104	COMP3
50	0x0000_0108	Reserved
51	0x0000_010C	Reserved
52	0x0000_0110	Reserved
53	0x0000_0114	Reserved

54	0x0000_0118	Reserved
55	0x0000_011C	Reserved
56	0x0000_0120	Reserved
57	0x0000_0124	Reserved
58	0x0000_0128	Reserved
59	0x0000_012C	Reserved
60	0x0000_0130	Reserved
61	0x0000_0134	Reserved
62	0x0000_0138	Reserved
63	0x0000_013C	Reserved

3. Boot Mode

Boot Mode Pins

The Z32F1281 MCU has a Boot Mode option to program internal Flash memory. When the BOOT pin is pulled low, the system will start up in the BOOT area (0x1FFF_0000) instead of the default Flash area (0x0000_0000). This provides the ability to flash the part using either UART or SPI interfaces. The BOOT pin has an internal pull up resistor. Therefore, when the BOOT pin is not connected, it rides high (normal state).

Boot Mode uses the UART0 port and the SPI0 ports for the interface. The JTAG and SW interfaces can also be used, which provide the ability to recover from a bad Flash update that prevents the JTAG or SW debugger from attaching.

The pins for Boot Mode are listed in Table 3.1.

Table 3.1. Boot Mode Pin List

Block	Pin Name	Dir	Description
SYSTEM	**nRESET/PC10**	I	Reset Input signal
	BOOT/PC11	I	'0' to enter Boot mode
UART0	**RXD0/PC14**	I	UART Boot Receive Data
	TXD0/PC15	O	UART Boot Transmit Data
SPI0	**SS0/PA12**	I	SPI Boot Slave Select
	SCK0/PA13	I	SPI Boot Clock Input
	MOSI0/PA14	I	SPI Boot Data Input
	MISO0/PA15	O	SPI Boot Data Output

Boot Mode Connections

The target board can be designed using either of the Boot Mode ports – UART or SPI.

Figure 3.1 and Figure 4.1Figure 3.2 show sample connection diagrams in Boot Mode.

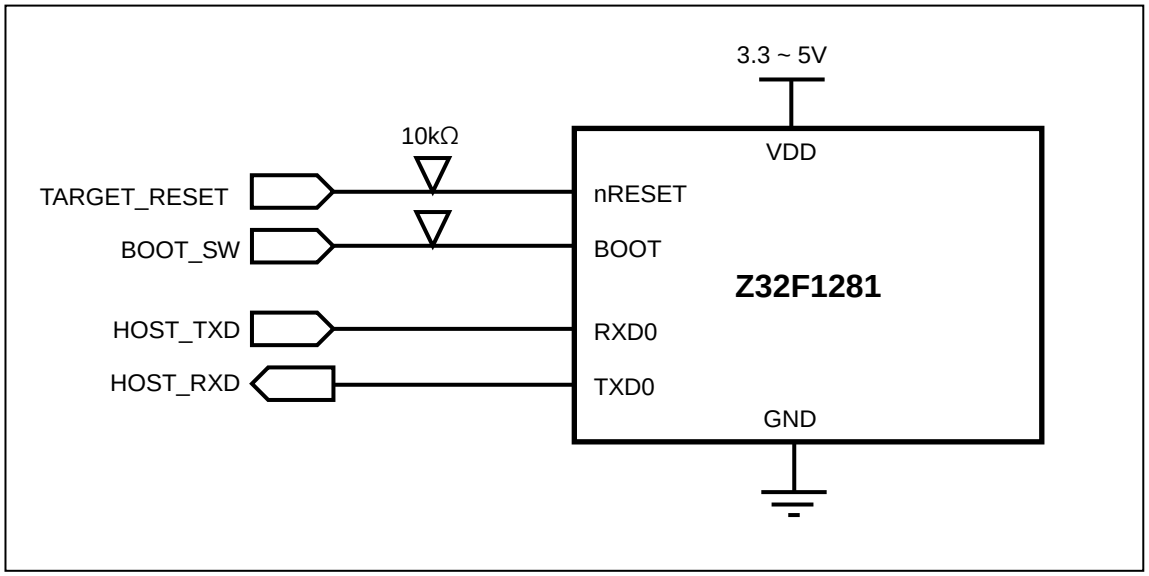


Figure 3.1. Connection Diagram of UART Boot

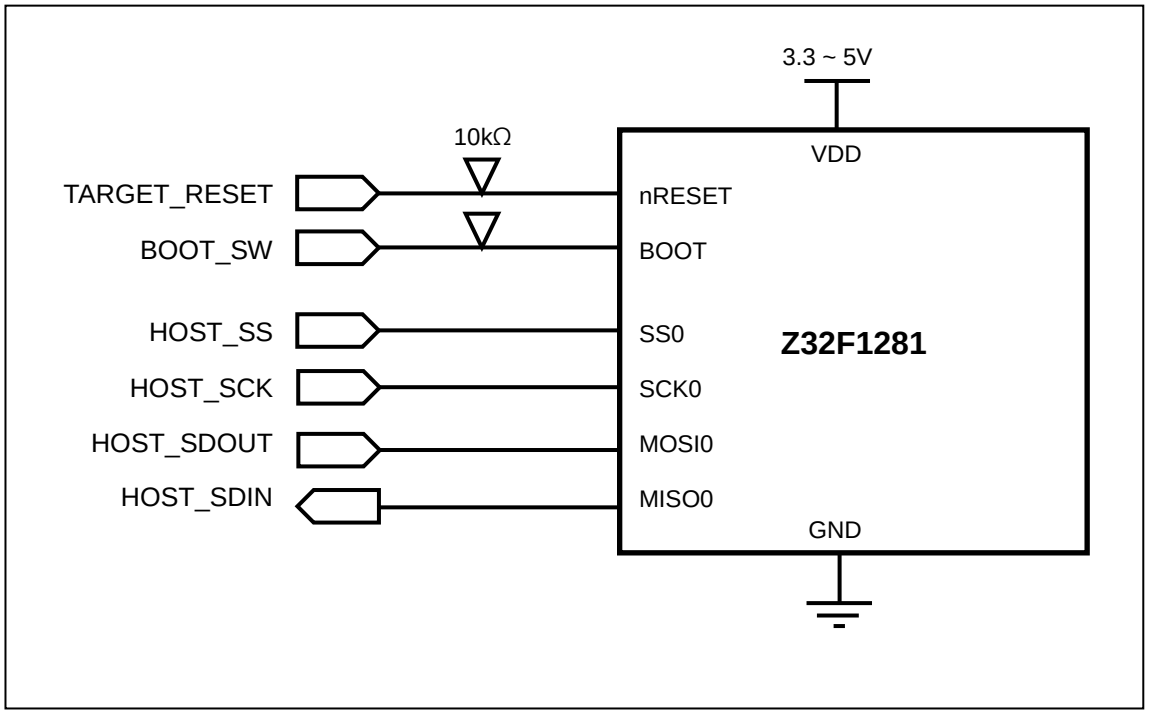


Figure 3.2. Connection Diagram of SPI Boot

4. System Control Unit

Overview

The Z32F1281 MCU has a built-in intelligent power control block which manages system analog blocks and operating modes. Internal reset and clock signals are controlled by the SCU block to maintain optimal system performance and power dissipation.

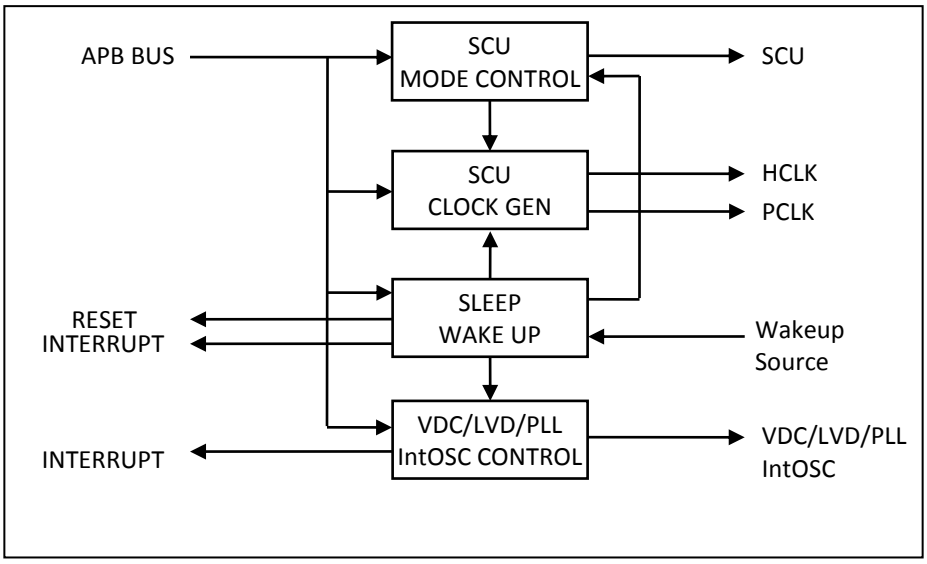


Figure 4.1. SCU Block Diagram

Clock System

The Z32F1281 MCU contains two main operating clocks – HCLK, which supplies the clock to the CPU and the AHB bus system; and PCLK, which supplies the clock to the peripheral systems. Users can control the clock system variation by software. Figure 4.2 shows the clock system of the chip. Table 4.1 lists the clock source descriptions.

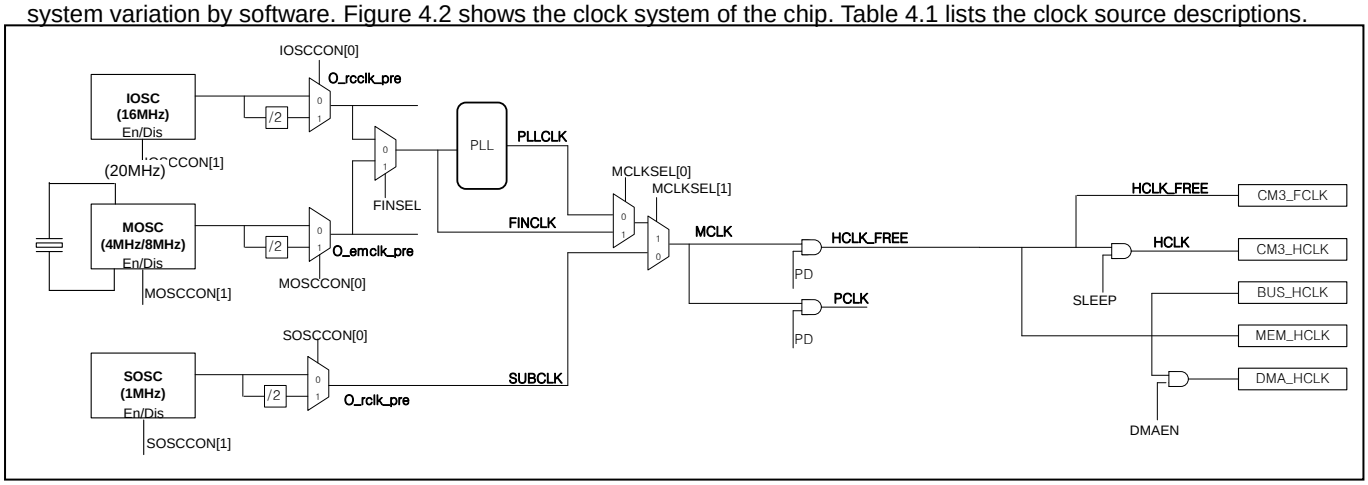


Figure 4.2. System Clock Configuration

Each of the registers to switch the clock source has a glitch-free circuit. Therefore, the clock can be switched without the risk of glitches.

Table 4.1. Clock Sources

Clock name	Frequency	Description
IOSC20	20MHz	Internal OSC
MOSC	XTAL(4MHz~8MHz)	External Crystal IOSC
PLL Clock	8MHz ~ 80MHz	On Chip PLL
ROSC	1MHz	Internal RING OSC

The PLL can synthesize the PLLCLK clock up to 80 MHz with the FIN reference clock. It also has an internal pre-divider and post-divider.

HCLK Clock Domain

The HCLK clock feeds the clock to the CPU and AHB bus. The Cortex-M3 CPU requires two clocks related with HCLK clock – FCLK and HCLK. FCLK is the free running clock and it is always running except in Power-down mode. HCLK can be stopped in Idle mode.

Miscellaneous Clock Domain for Cortex-M3

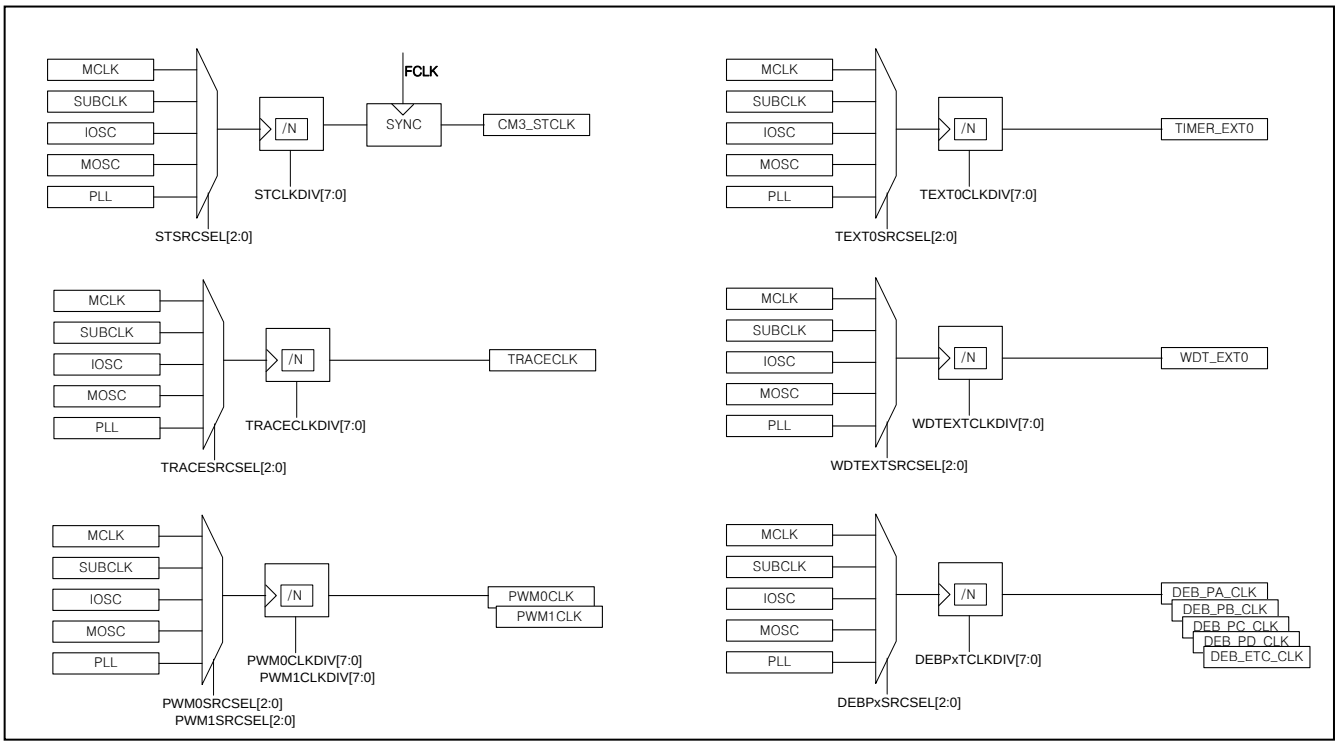


Figure 4.3. Miscellaneous Clock Configuration

PCLK Clock Domain

PCLK is the master clock of all the peripherals. It can be stopped in Power-down mode. Each peripheral clock is generated by the PCER register set.

Clock Configuration

After power up, the default system clock is fed by the RINGOSC (1MHz) clock. RINGOSC is enabled by default during the power up sequence. The other clock sources are enabled by user controls with the RINGOSC system clock.

The MOSC clock can be enabled by the CSCR register. Before enabling the MOSC block, the pin mux configuration should be set for XIN, XOUT function. PC12 and PC13 pins are shared with MOSC's XIN and XOUT function - PCCMR and PCCCR registers should be configured properly. After enabling the MOSC block, you must wait for more than 1 msec to ensure stable operation of crystal oscillation.

The PLL clock can be enabled by the PLLCON register. After enabling the PLL block, you must wait for the PLL lock flag. When the PLL output clock is stable, you can select MCLK for your system requirement. Before changing the system clock, set Flash access wait to the maximum value. After the system clock is changed, set the desired Flash access wait time.

Figure 4.4 shows a flow chart outlining the process to configure the system clock.

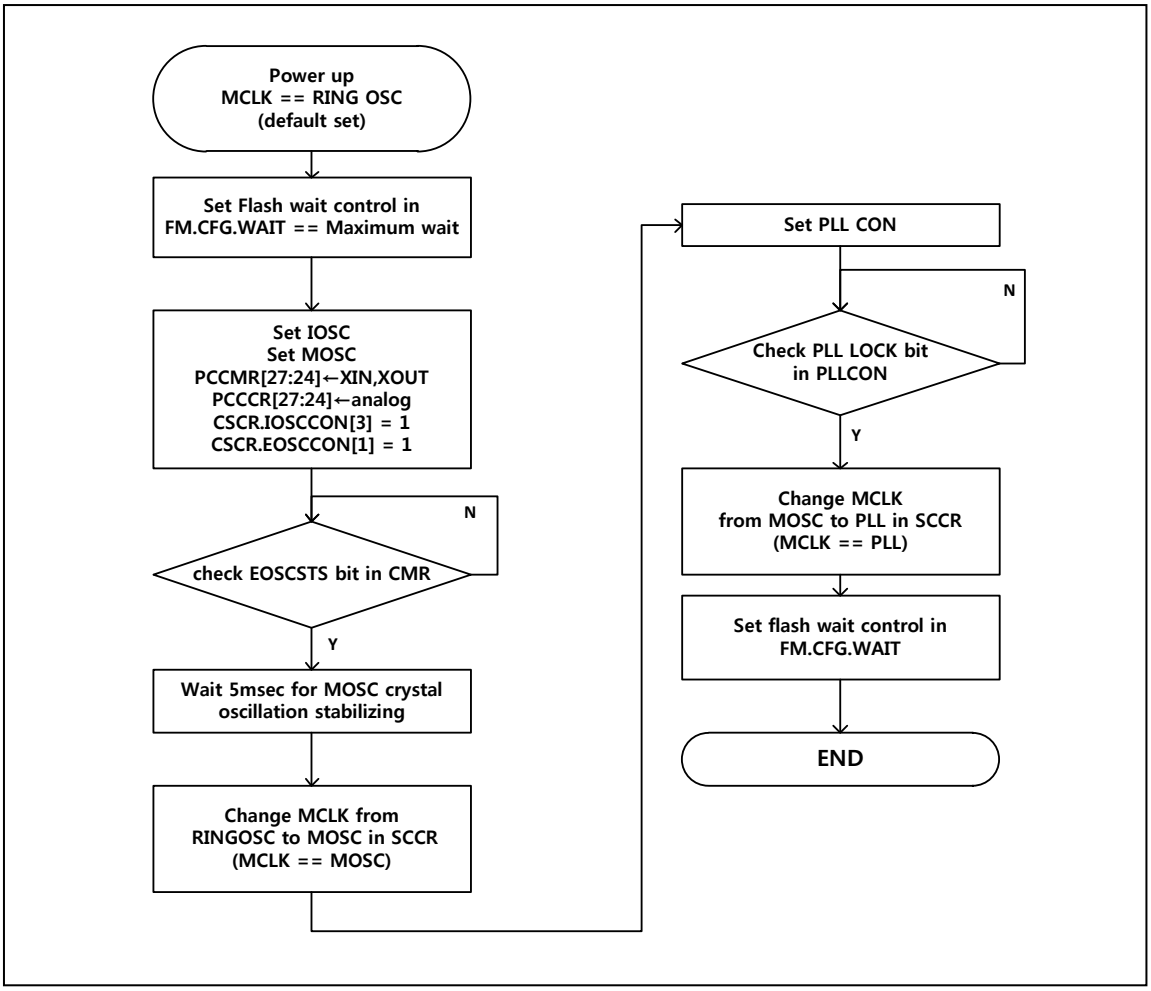


Figure 4.4. Clock Configuration Procedure Flowchart

When you speed up the system clock up to the maximum operating frequency, check the Flash wait control configuration. The CLK3 and CLK4 bit fields in the FMCR register can control the wait time. Flash read access time is one of the limiting factors that can impact performance.

The wait control recommendation is provided in Table 4.2.

Table 4.2. Flash Wait Control Recommendation

FMCR	Flash Access Wait	Available Max System Clock Frequency
CLK3	3 Clock Wait	~48MHz
CLK4	4 Clock Wait	~72MHz

Reset

The Z32F1281 MCU has two system resets:

- Cold reset by POR which is effective during power up or down sequence
- Warm reset which is generated by several reset sources.

The reset event causes the chip to return to its initial state.

Cold reset has only one reset source, which is POR. Warm reset has the following reset sources:

- nRESET pin
- WDT reset
- LVD reset
- MCLK Fail reset
- MOSC Fail reset
- S/W reset
- CPU request reset

Cold Reset

The cold reset is an important feature of the chip when power is up. This characteristic affects the system boot globally. Internal VDC is enabled when VDD power is turned on. The internal VDD level slope is followed by the external VDD power slope. The boot operation is started when the internal PoR trigger level is 1.4V of the internal VDC voltage out level. The RINGOSC clock is enabled and counts 4 msec to stabilize the internal VDC level. During this time, the external VDD voltage level should be higher than the initial LVD level (2.3V). After counting 4 msec, the CPU reset is released and the operation is started.

Figure 4.5 shows the power up sequence and internal reset waveform.

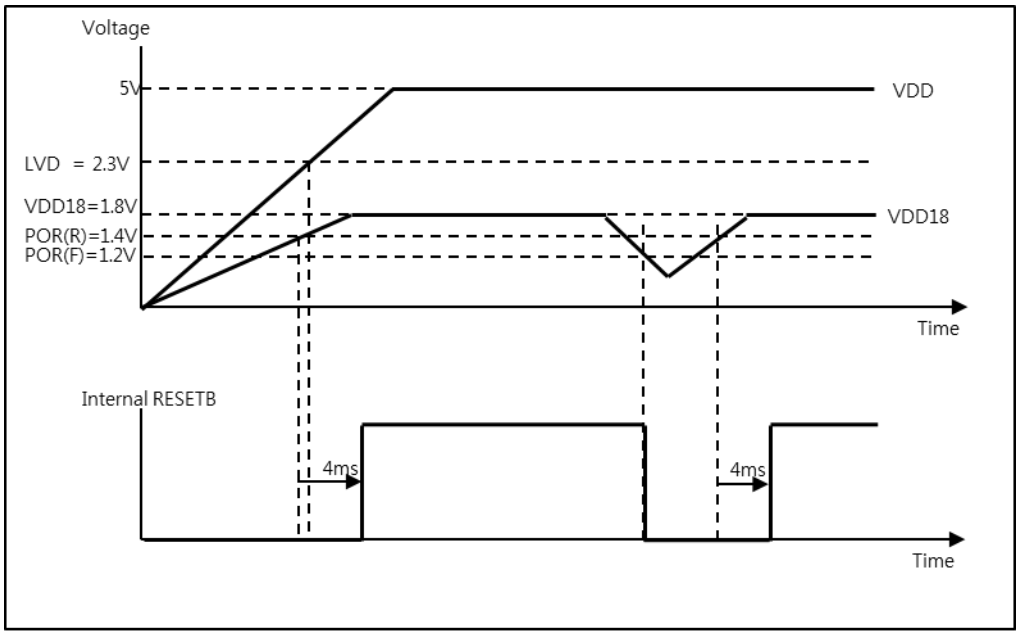


Figure 4.5. Power Up POR Sequence

The RSSR register shows the POR reset status. The last reset comes from POR. RSSR.PORST is set to “1”. After power up, this bit is always “1”. If an abnormal internal voltage drop occurs during normal operation, the system is reset and this bit is also set to “1”.

When a cold reset is applied, the entire chip returns to its initial state.

Warm Reset

The warm reset event has several reset sources and some parts of the chip return to initial state when a warm reset condition occurs.

The warm reset source is controlled by the RSER register and the status appears in the RSSR register. The reset for each peripheral block is controlled by the PRER register. The reset can be masked independently.

The CM3_SYSRESETn signal resets the processor, excluding debug logic in the processor.

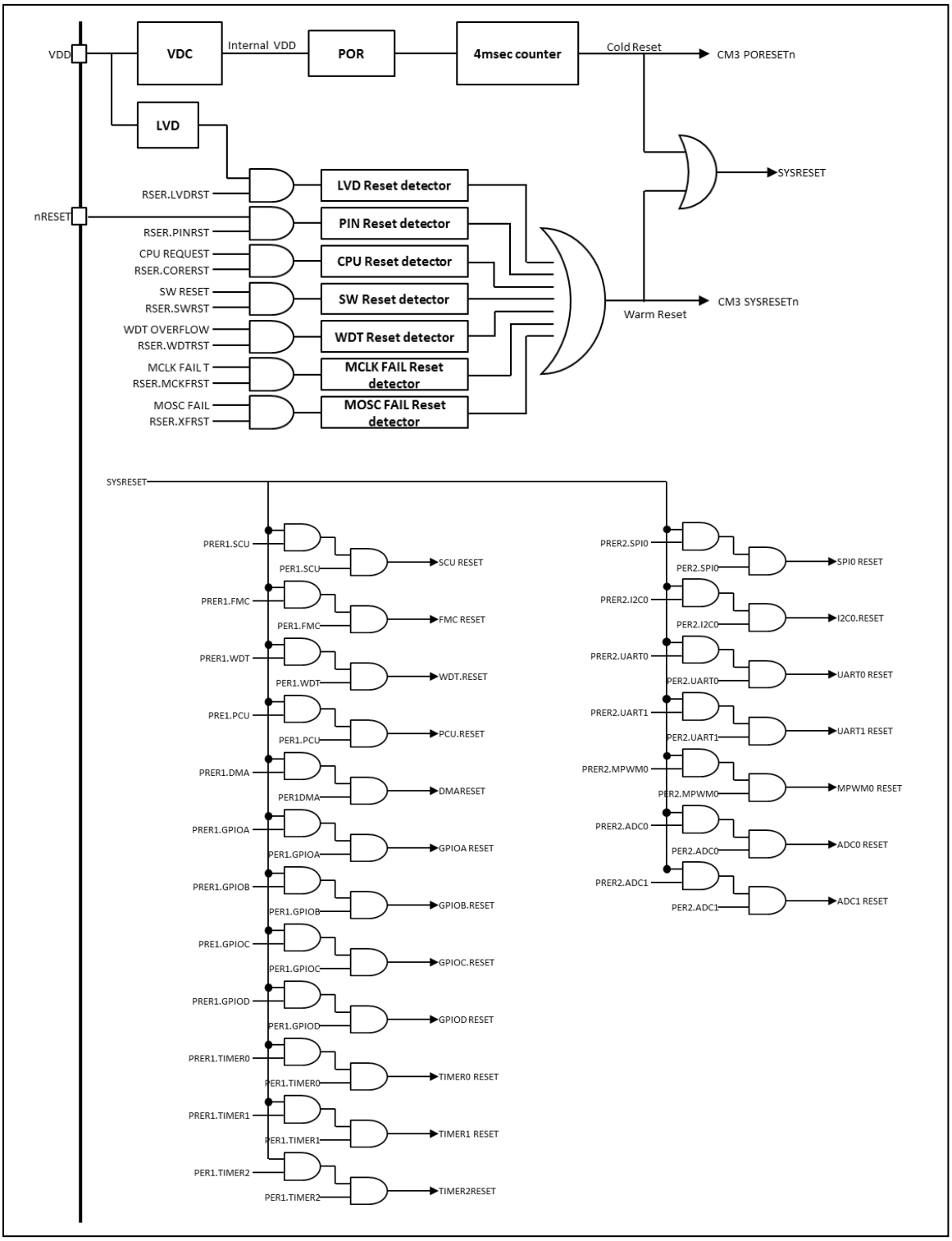


Figure 4.6. Reset Configuration

Operation Mode

Three operational states are available in addition to the Initialization state (INIT). When the reset is asserted (brought low), the Z32F1281 MCU runs at 1 MHz, driven by the ROSC. All the other clocks are disabled and peripheral power and clocks are reset.

The RUN mode is designed to run at maximum performance of the CPU with a high-speed clock system. The IOSC must be enabled in order to enable the MOSC. The SLEEP mode is designed to run in Low Power consumption mode by halting the processor core and any unused peripherals.

Figure 4.7 shows the operation mode transition diagram.

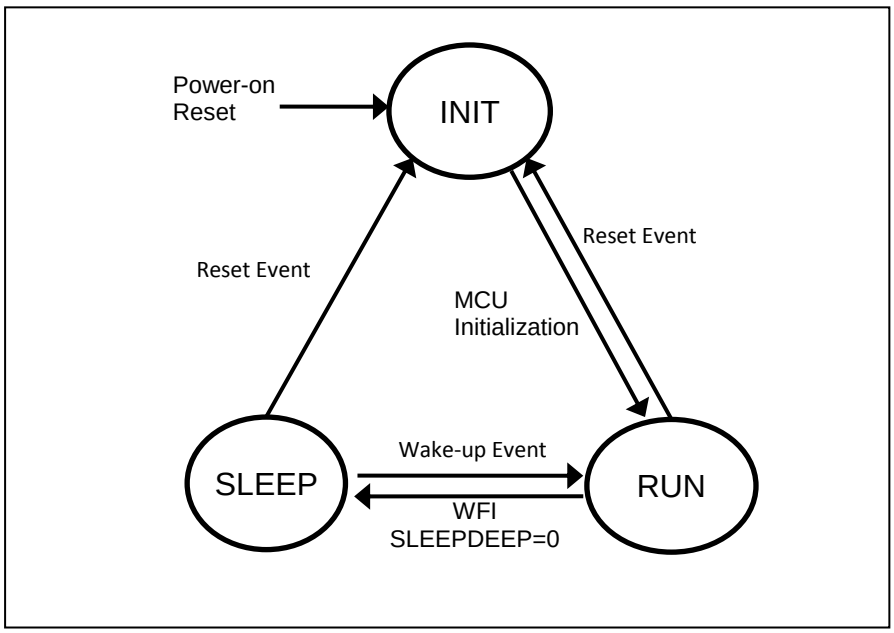


Figure 4.7. Operating Mode

RUN Mode

This mode operates the MCU core and the peripheral hardware by using the high-speed clock. After reset, followed by the INIT state, the system can be configured to enter into RUN mode.

SLEEP Mode

Only the CPU is stopped in this mode. Each peripheral function can be enabled by the function enable and clock enable bit in the PER and PCER registers. To enter sleep mode, configure the system for the desired low power state then use the "wfi" instruction. When exiting sleep mode, the clock will be set to the RING oscillator.

Pin Description

Table 4.3. SCU and PLL Pins

PIN NAME	TYPE	DESCRIPTION
nRESET	I	External Reset Input
XIN/XOUT	OSC	External Crystal Oscillator
STBYO	O	Stand-by Output Signal
CLKO	O	Clock Output Monitoring Signal

Registers

The base address of SCU is `0x4000_0000`. The register map is described in Table 4.4.

Table 4.4. SCU Register Map

Name	Offset	R/W	Description	Reset
CIDR	0x0000	R	CHIP ID Register	AC33_8128
SMR	0x0004	R/W	System Mode Register	0000_0000
SRCR	0x0008	R/W	System Reset Control Register	0000_0000
WUER	0x0010	R/W	Wake up source enable register	0000_0000
WUSR	0x0014	R/W	Wake up source status register	0000_0000
RSER	0x0018	R/W	Reset source enable register	0000_0049
RSSR	0x001C	R/W	Reset source status register	0000_0080*
PRER1	0x0020	R/W	Peripheral reset enable register 1	03FF_1F1F*
PRER2	0x0024	R/W	Peripheral reset enable register 2	00F3_0F33*
PER1	0x0028	R/W	Peripheral enable register 1	0000_000F*
PER2	0x002C	R/W	Peripheral enable register 2	0000_0101*
PCER1	0x0030	R/W	Peripheral clock enable register 1	0000_000F*
PCER2	0x0034	R/W	Peripheral clock enable register 2	0000_0101*
CSCR	0x0040	R/W	Clock Source Control register	0000_0020
SCCR	0x0044	R/W	System Clock Control register	0000_0000
CMR	0x0048	R/W	Clock Monitoring register	0000_0003
NMIR	0x004C	R/W	NMI control register	0000_0000
COR	0x0050	R/W	Clock Output Control register	0000_000F
PLLCON	0x0060	R/W	PLL Control register	0000_1000
VDCCON	0x0064	R/W	VDC Control register	0000_000F
LVDCON	0x0068	R/W	LVD Control register	0000_0001
IOSCTRIM	0x006C	R/W	Internal RC OSC Control Register	0000_0000
OPA0TRIM	0x0070	R/W	OPAM 0 trim register	0000_0000
OPA1TRIM	0x0074	R/W	OPAM 1 trim register	0000_0000
OPA2TRIM	0x0078	R/W	OPAM 2 trim register	0000_0000
OPA3TRIM	0x007C	R/W	OPAM 3 trim register	0000_0000
EOSCR	0x0080	R/W	External Oscillator control register	0000_0000
EMODR	0x0084	R/W	External mode pin read register	0000_000X

DBCLK1	0x009C	R/W	Debounce Clock Control register 1	0000_0000
DBCLK2	0x00A0	R/W	Debounce Clock Control register 2	0000_0000
MCCR1	0x0090	R/W	Misc Clock Control register 1	0404_0001
MCCR2	0x0094	R/W	Misc Clock Control register 2	0000_0000
MCCR3	0x0098	R/W	Misc Clock Control register 3	0000_0001
MCCR4	0x00A4	R/W	Misc Clock Control register 4	0000_0001

CIDR Chip ID Register

The Chip ID Register shows chip identification information. This register is a 32-bit read-only register.

CIDR=0x4000_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHIPID																															
0xAC33_8128																															
R																															

31	CHIP ID	Device ID
0		0xAC33_8128

SMR System Mode Register

The current operating mode is shown in this SCU mode register and the operation mode can be changed by writing the new mode in this register. The previous operating mode will be saved in this register after a reset event. System Mode Register is a 16-bit register.

SMR=0x4000_0004

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										PREVMODE					
0	0	0	0	0	0	0	0	0	0	00		0	0	0	0
										R					

5	PREVMODE	Previous operating mode before current reset event.	
4		00	Previous operating mode was RUN mode
		01	Previous operating mode was SLEEP mode
		10	Previous operating mode was Power-down mode
		11	Previous operating mode was INIT mode

SRCR System Reset Control Register

The System Reset Control Register allows the software to initiate a reset. This register also provides the polarity for the STBYOP pin.

SCR=0x4000_0008

7	6	5	4	3	2	1	0
							SWRST
0	0	0	0	0	0	0	0
							W

1	SWRST	Internal soft reset activation bit
	0	Normal operation
	1	Internal soft reset is applied and auto cleared

WUER Wakeup Source Enable Register

This is the enable wakeup source. The source of chip wakeup should be enabled in each bit field. If the source will be used as the wakeup source, write **1** to its enable bit. If the source will not be used as the wakeup source, write **0** into its enable bit. This register is a 16-bit register.

WUER=-0x4000_0010

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				GPIODWUE	GPIOCWUE	GPIOBWUE	GPIOAWUE							WDTWUE	LVDWUE
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
				RW	RW	RW	RW							RW	RW

12	GPIOEWUE	Enable wakeup source of GPIOE port pin change event
		0 Not used for wakeup source
		1 Enable the wakeup event generation
11	GPIODWUE	Enable wakeup source of GPIOD port pin change event
		0 Not used for wakeup source
		1 Enable the wakeup event generation
10	GPIOCWUE	Enable wakeup source of GPIOC port pin change event
		0 Not used for wakeup source
		1 Enable the wakeup event generation
9	GPIOBWUE	Enable wakeup source of GPIOB port pin change event
		0 Not used for wakeup source
		1 Enable the wakeup event generation
8	GPIOAWUE	Enable wakeup source of GPIOA port pin change event
		0 Not used for wakeup source
		1 Enable the wakeup event generation
1	WDTWUE	Enable wakeup source of watchdog timer event
		0 Not used for wakeup source
		1 Enable the wakeup event generation
0	LVDWUE	Enable wakeup source of LVD event
		0 Not used for wakeup source
		1 Enable the wakeup event generation

WUSR Wakeup Source Status Register

When the system is woken up by any wakeup source, the wakeup source is identified by reading this register. When the bit is set to **1**, the related wakeup source issues the wakeup to the SCU. The bit will be cleared when the event is cleared by the software.

WUSR=0x4000_0014

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				GPIODWU	GPIOCWU	GPIOBWU	GPIOAWU							WDTWU	LVDWU
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
				R	R	R	R							R	R

12	GPIOEWU	Status of wakeup source of GPIOE port pin change event
		0 No wakeup event
		1 Wakeup event was generated
11	GPIODWU	Status of wakeup source of GPIOD port pin change event
		0 No wakeup event
		1 Wakeup event was generated
10	GPIOCWU	Status of wakeup source of GPIOC port pin change event
		0 No wakeup event
		1 Wakeup event was generated
9	GPIOBWU	Status of wakeup source of GPIOB port pin change event
		0 No wakeup event
		1 Wakeup event was generated
8	GPIOAWU	Status of wakeup source of GPIOA port pin change event
		0 No wakeup event
		1 Wakeup event was generated
1	WDTWU	Status of wakeup source of watchdog timer event
		0 No wakeup event
		1 Wakeup event was generated
0	LVDWU	Status of wakeup source of LVD event
		0 No wakeup event
		1 Wakeup event was generated

RSER Reset Source Enable Register

The reset source which will generate the reset event can be selected by the RSER register. Write **1** in the bit field of each reset source to transfer the reset source event to the reset generator. Write **0** in the bit field of each reset source to mask the reset source event, and therefore, not generate the reset event.

RSER=0x4000_0018

7	6	5	4	3	2	1	0
	PINRST	CPURST	SWRST	WDTRST	MCKFRST	XFRST	LVDRST
0	1	0	0	1	0	0	1
	RW	RW	RW	RW	RW	RW	RW

6	PINRST	External pin reset enable bit
		0 Reset from this event is masked
		1 Reset from this event is enabled
5	CPURST	CPU request reset enable bit
		0 Reset from this event is masked
		1 Reset from this event is enabled
4	SWRST	Software reset enable bit
		0 Reset from this event is masked
		1 Reset from this event is enabled
3	WDTRST	Watchdog Timer reset enable bit
		0 Reset from this event is masked
		1 Reset from this event is enabled
2	MCKFRST	MCLK Clock fail reset enable bit
		0 Reset from this event is masked
		1 Reset from this event is enabled
1	XFRST	External OSC Clock fail reset enable bit
		0 Reset from this event is masked
		1 Reset from this event is enabled
0	LVDRST	LVD reset enable bit
		0 Reset from this event is masked
		1 Reset from this event is enabled

RSSR Reset Source Status Register

The RSSR shows the reset source information when a reset event occurs. For a given reset source, **1** indicates that a reset event exists and **0** shows that a reset event does not exist. When a reset source is found, writing **1** to the corresponding bit will clear the reset status. This register is an 8-bit register.

RSSR=0x4000_001C

7	6	5	4	3	2	1	0
PORST	PINRST	CPURST	SWRST	WDTRST	MCKFRST	XFRST	LVDRST
1	0	0	0	0	0	0	0
RC1	RC1	RC1	RC1	RC1	RC1	RC1	RC1

7	PORST	Power on reset status bit
	0	Read : Reset from this event was not exist Write : no effect
	1	Read :Reset from this event was occurred Write : Clear the status
6	PINRST	External pin reset status bit
	0	Read : Reset from this event was not exist Write : no effect
	1	Read :Reset from this event was occurred Write : Clear the status
5	CPURST	CPU request reset status bit
	0	Read : Reset from this event was not exist Write : no effect
	1	Read :Reset from this event was occurred Write : Clear the status
4	SWRST	Software reset status bit
	0	Read : Reset from this event was not exist Write : no effect
	1	Read :Reset from this event was occurred Write : Clear the status
3	WDTRST	Watchdog Timer reset status bit
	0	Read : Reset from this event was not exist Write : no effect
	1	Read :Reset from this event was occurred Write : Clear the status
2	MCLKFRST	MCLK Fail reset status bit
	0	Read : Reset from this event was not exist Write : no effect
	1	Read :Reset from this event was occurred Write : Clear the status
1	XFRST	Clock fail reset status bit
	0	Read : Reset from this event was not exist Write : no effect
	1	Read :Reset from this event was occurred Write : Clear the status
0	LVDRST	LVD reset status bit
	0	Read : Reset from this event was not exist Write : no effect
	1	Read :Reset from this event was occurred Write : Clear the status

PRER1 Peripheral Reset Enable Register 1

The reset of each peripheral by an event reset can be masked by user settings. The PRER1/2 register controls enabling of the event reset. If the corresponding bit is **1**, the peripheral corresponding to this bit accepts the reset event. Otherwise, the peripheral is protected from the reset event and maintains its current operation.

PRER1=0x4000_0020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						TIMER9	TIMER8					TIMER3	TIMER2	TIMER1	TIMER0					GPIOD	GPIOC	GPIOB	GPIOA				DMA	PCU	WDT	FMC	SCU
0	0	0	0	0	0	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	1	1	1	1	1
						RW	RW					RW	RW	RW	RW					RW	RW	RW	RW				RW	RW	RW	RW	RW

25	TIMER9	TIMER9 reset mask
24	TIMER8	TIMER8 reset mask
19	TIMER3	TIMER3 reset mask
18	TIMER2	TIMER2 reset mask
17	TIMER1	TIMER1 reset mask
16	TIMER0	TIMER0 reset mask
11	GPIOD	GPIOE reset mask
10	GPIOC	GPIOE reset mask
9	GPIOB	GPIOE reset mask
8	GPIOA	GPIOA reset mask
4	DMA	DMA reset mask
3	PCU	Port Control Unit reset mask
2	WDT	Watchdog Timer reset mask
1	FMC	Flash memory controller reset mask
0	SCU	System Control Unit reset mask

PRER2 Peripheral Reset Enable Register 2

The reset of each peripheral by an event reset can be masked by user settings. The PRER1/2 register controls enabling of the event reset. If the corresponding bit is **1**, the peripheral corresponding to this bit accepts the reset event. Otherwise, the peripheral is protected from the reset event and maintains its current operation.

PRER2=0x4000_0024

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								AFE	ADC2	ADC1	ADC0			MPWM1	MWPM0					UART3	UART2	UART1	UART0			I2C1	I2C0			SPI1	SPI0
0	0	0	0	0	0	0	0	1	1	1	1	0	0	1	1	0	0	0	0	1	1	1	1	0	0	1	1	0	0	1	1
								RW	RW	RW	RW			RW	RW					RW	RW	RW	RW			RW	RW			RW	RW

23	AFE	AFE reset enable
22	ADC2	ADC2 reset enable
21	ADC1	ADC1 reset enable
20	ADC0	ADC0 reset enable
17	MPWM1	MPWM1 reset enable
16	MPWM0	MPWM0 reset enable
11	UART3	UART3 reset enable
10	UART2	UART2 reset enable
9	UART1	UART1 reset enable
8	UART0	UART0 reset enable
5	I2C1	I²C1 reset enable
4	I2C0	I²C0 reset enable
1	SPI1	SPI1 reset enable
0	SPI0	SPI0 reset enable

PER1 Peripheral Enable Register 1

To use a peripheral unit, it should be activated by writing **1** to the corresponding bit in the PER1/2 register. Prior to activation, the peripheral stays in reset state.

To disable the peripheral unit, write **0** to the corresponding bit in the PER0/1 register, after which the peripheral enters the reset state.

PER1=0x4000_0028

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						TIMER9	TIMER8					TIMER3	TIMER2	TIMER1	TIMER0					GPIOD	GPIOC	GPIOB	GPIOA				DMA				
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
						RW	RW					RW	RW	RW	RW					RW	RW	RW	RW				RW	R	R	R	R

25	TIMER9	TIMER9 function enable
24	TIMER8	TIMER8 function enable
19	TIMER3	TIMER3 function enable
18	TIMER2	TIMER2 function enable
17	TIMER1	TIMER1 function enable
16	TIMER0	TIMER0 function enable
11	GPIOD	GPIOD function enable
10	GPIOC	GPIOC function enable
9	GPIOB	GPIOB function enable
8	GPIOA	GPIOA function enable
4	DMA	DMA function enable
3		Reserved
2		
1		
0		

PER2 Peripheral Enable Register 2

To use a peripheral unit, it should be activated by writing **1** to the corresponding bit in the PER1/2 register. Prior to activation, the peripheral stays in reset state.

To disable the peripheral unit, write **0** to the corresponding bit in the PER0/1 register, after which the peripheral enters the reset state.

PER2=0x4000_002C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								AFE	ADC2	ADC1	ADC0			MPWM1	MWPM0					UART3	UART2	UART1	UART0			I2C1	I2C0			SPI1	SPI0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
								RW	RW	RW	RW			RW	RW					RW	RW	RW	RW			RW	RW			RW	RW

23	AFE	AFE function enable
22	ADC2	ADC2 function enable
21	ADC1	ADC1 function enable
20	ADC0	ADC0 function enable
17	MPWM1	MPWM1 function enable
16	MPWM0	MPWM0 function enable
11	UART3	UART3 function enable
10	UART2	UART2 function enable
9	UART1	UART1 function enable
8	UART0	UART0 function enable
5	I2C1	I²C1 function enable
4	I2C0	I²C0 function enable
1	SPI1	SPI1 function enable
0	SPI0	SPI0 function enable

PCER1 Peripheral Clock Enable Register 1

To use a peripheral unit, its clock should be activated by writing **1** to the corresponding bit in the PCER1/2 register. The peripheral will not operate correctly until its clock is enabled.

To stop the clock of the peripheral unit, write **0** to the corresponding bit in the PCER1/2 register.

PCER1=0x4000_0030

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						TIMER9	TIMER8					TIMER3	TIMER2	TIMER1	TIMER0					GPIOD	GPIOC	GPIOB	GPIOA				DMA				
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
						RW	RW					RW	RW	RW	RW					RW	RW	RW	RW				RW	R	R	R	R

25	TIMER9	TIMER9 clock enable
24	TIMER8	TIMER8 clock enable
19	TIMER3	TIMER3 clock enable
18	TIMER2	TIMER2 clock enable
17	TIMER1	TIMER1 clock enable
16	TIMER0	TIMER0 clock enable
11	GPIOD	GPIOD clock enable
10	GPIOC	GPIOC clock enable
9	GPIOB	GPIOB clock enable
8	GPIOA	GPIOA clock enable
4	DMA	DMA clock enable
3		Reserved
2		
1		
0		

PCER2 Peripheral Clock Enable Register 2

To use a peripheral unit, activate its clock by writing **1** to the corresponding bit in the PCER1/2 register. The peripheral will not operate correctly until its clock is enabled.

To stop the clock of the peripheral unit, write **0** to the corresponding bit in the PCER1/2 register.

PCER2=0x4000_0034

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								AFE	ADC2	ADC1	ADC0			MPWM1	MWPM0					UART3	UART2	UART1	UART0			I2C1	I2C0			SPI1	SPI0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
								RW	RW	RW	RW			RW	RW					RW	RW	RW	RW			RW	RW			RW	RW

23	AFE	AFE clock enable
22	ADC2	ADC2 clock enable
21	ADC1	ADC1 clock enable
20	ADC0	ADC0 clock enable
17	MPWM1	MPWM1 clock enable
16	MPWM0	MPWM0 clock enable
11	UART3	UART3 clock enable
10	UART2	UART2 clock enable
9	UART1	UART1 clock enable
8	UART0	UART0 clock enable
5	I2C1	I²C1 clock enable
4	I2C0	I²C0 clock enable
1	SPI1	SPI1 clock enable
0	SPI0	SPI0 clock enable

CSCR Clock Source Control Register

The Z32F1281 MCU has multiple clock sources to generate internal operating clocks. Each clock source can be controlled by the CSCR register. This register is an 8-bit register.

CSCR=0x4000_0040

7	6	5	4	3	2	1	0
-		RINGOSCCON		IOSCCON		EOSCCON	
00		10		00		00	
R		RW		RW		RW	

5	RINGOSCCON	Internal ring oscillator control
4		0X Stop internal sub oscillator
		10 Enable internal sub oscillator
		11 Enable internal sub oscillator divide by 2
3	IOSCCON	Internal oscillator control
2		0X Stop internal oscillator
		10 Enable internal oscillator
		11 Enable internal oscillator divide by 2
1	EOSCCON	External crystal oscillator control
0		0X Stop External Crystal oscillator
		10 Enable External Crystal oscillator
		11 Enable External Crystal divide by 2

SCCR System Clock Control Register

The Z32F1281 MCU has multiple clock sources to generate internal operating clocks. Each clock source can be controlled by the SCU CSCR register. The MOSC must be running and stable before setting the FINSEL bit.

SCCR=0x4000_0044

7	6	5	4	3	2	1	0
-					FINSEL	MCLKSEL	
0000					0	00	
R					RW	RW	

2	FINSEL	PLL input source FIN select register
		0 IOSC clock is used as FIN clock
		1 MOSC clock is used as FIN clock
1	MCLKSEL	System clock select register
0		0X Internal sub oscillator
		10 PLL bypassed clock
		11 PLL output clock

Note: When changing FINSEL, both internal OSC and external OSC should be alive, otherwise the chip will malfunction.

CMR Clock Monitoring Register

To monitor the internal clock and external oscillator, the MCLKMNT/EOSCMNT bits must be set before the MCLK and EOSC bits are valid. The Clock Monitoring Register is a 16-bit register.

Note: The EOSC bit only checks for the EOSC oscillation, not its stability. When the system detects an MCLKFAIL interrupt, the MCLKREC bit determines if the system dies or will auto-recover using the ROSC. The system usually auto-recovers so that it can continue running.

CMR=0x4000_0048

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCLKREC								MCLKMNT	MCLKIE	MCLKFAIL	MCLKSTS	EOSCMNT	EOSCIE	EOSCFAIL	EOSCSTS
0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	1
RW								RW	RW	RC1	RC1	RW	RW	RC1	RC1

15	MCLKREC	MCLK fail auto recovery
	0	MCLK is changed to RINGOSC by default when MCLKFAIL issued
	1	MCLK auto recovery is disabled
7	MCLKMNT	MCLK monitoring enable
	0	MCLK monitoring disabled
	1	MCLK monitoring enabled
6	MCLKIE	MCLK fail interrupt enable
	0	MCLK fail interrupt disabled
	1	MCLK fail interrupt enabled
5	MCLKFAIL	MCLK fail interrupt
	0	MCLK fail interrupt not occurred
	1	Read : MCLK fail interrupt is pending Write : Clear pending interrupt
4	MCLKSTS	MCLK clock status
	0	No clock is present on MCLK
	1	Clock is present on MCLK
3	EOSCMNT	External oscillator monitoring enable
	0	External oscillator monitoring disabled
	1	External oscillator monitoring enabled
2	EOSCIE	External oscillator fail interrupt enable
	0	External oscillator fail interrupt disabled
	1	External oscillator fail interrupt enabled
1	EOSCFAIL	External oscillator fail interrupt
	0	External oscillator fail interrupt not occurred
	1	Read : External oscillator fail interrupt is pending Write : Clear pending interrupt
0	EOSCSTS	External oscillator status
	0	Not oscillate
	1	External oscillator is working normally

The clock monitoring function cannot cover all malfunction cases. It is just used for the reference. Figure 4.8 shows the operational diagram for the clock monitoring function.

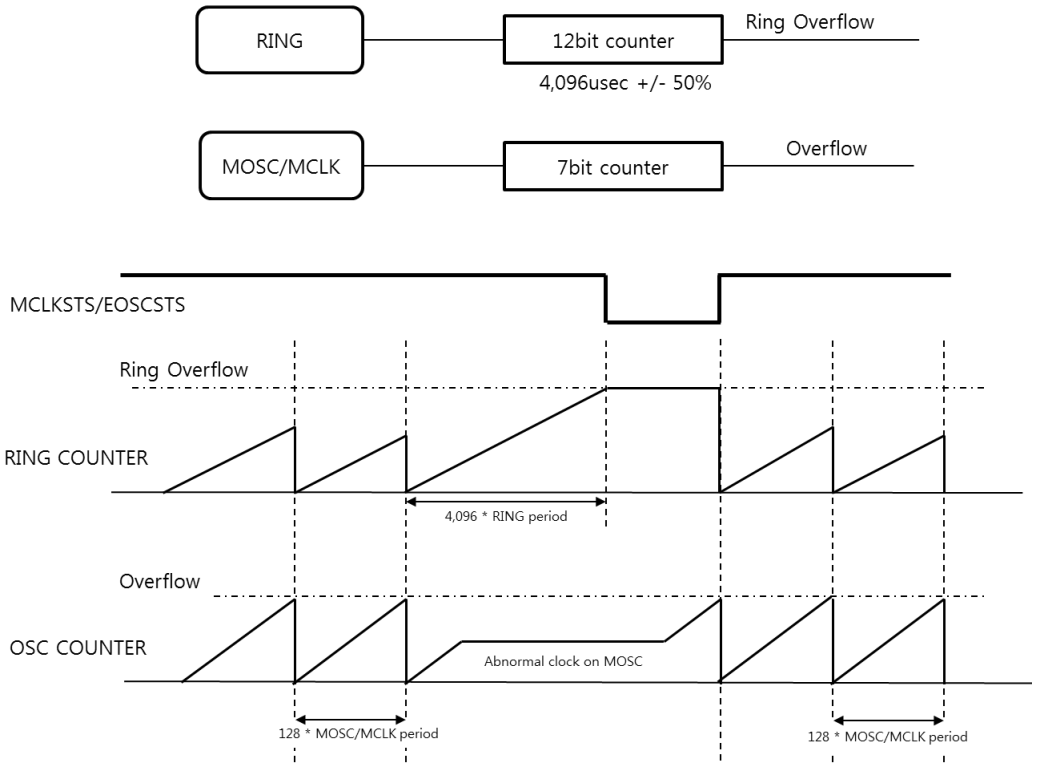


Figure 4.8. Clock Monitoring Function Diagram

NMIR NMI Control Register

Non-Maskable Interrupt pin configuration provides the ability to enable/disable and set the debounce of the NMI pin. It also provides the ability to monitor the interrupt and status of the NMI pin.

NMIR=0x4000_004C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												NMISTAT	NMIFLAG	NMIDBEN	NMIEN
0x00								0	0	0	0	0	0	0	0
RW												RW	RW	RW	RW

3	MMISTAT	NMI Pin status
		0 NMI pin is low status
		1 NMI pin is high status
2	NMIFLAG	NMI interrupt flag
		0 NMI interrupt is not pending
		1 NMI interrupt is pending
1	NMIDBEN	NMI pin debounce enable
		0 NMI pin debounce disable
		1 NMI pin debounce enable
0	NMIEN	NMI Enable Write permission is required by PCU write enable sequence
		0 NMI pin disable
		1 NMI pin enable

COR Clock Output Register

The clock output register controls the enabling/disabling of the clock signal and provides a divider for the clock output. In order to output the clock signal, you must enable the Clock out function pin. For more information, see Chapter 5, Port Control Unit.

COR=0x4000_0050

7	6	5	4	3	2	1	0
-			CLKOEN	CLKODIV			
000			0	1111			
R			RW	RW			

4	CLKOEN	Clock output enable 0 CLKO is disabled and stay "L" output 1 CLKO Is enabled
3 0	CLKODIV	Clock output divider value $\text{CLKO} = \text{MCLK} \quad (\text{CLKODIV} = 0)$ $\text{CLKO} = \frac{MCLK}{2 * (\text{CLKODIV} + 1)} \quad (CLKODIV > 0)$

PLLCON PLL Control Register

Integrated PLL can synthesize the high speed clock for extremely high performance of the CPU from either the internal oscillator (IOSC) or the external oscillator (MOSC). The PLL Control register provides the configuration for the PLL system. By default, the PLL system is in reset mode and disabled. You must negate the reset and enable the PLL to operate (bits 14 and 15 must be set). The Bypass bit must be set to output the PLL clock. The active clock is defined in SCCR bit 2 (FIN).

To calculate the PLL output:

PLL Out = ((Active clock / PREDIV) * FBCTRL) / POSTDIV

For example:

Using MOSC (assuming it is running at 8 MHz and selected):

PREDIV set to 1 (FIN / 2)

FBCTRL set to 0x05 (M=18)

POSTDIV set to 0x00 (N=1)

((8 MHz / 2) * 18) = 72 MHz

PLLCON=0x4000_0060

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PLLRSTB	PLLEN	BYPASS	LOCKSTS				PREDIV	FBCTRL				POSTDIV			
0	0	0	0	0	0	0	0	0000				0000			
RW	RW	RW	R				RW	RW				RW			

15	PLLRSTB	PLL reset	
		0 PLL reset is asserted	
		1 PLL reset is negated	
14	PLLEN	PLL enable	
		0 PLL is disabled	
		1 PLL is enabled	
13	BYPASS	FIN bypass	
		0 FOUT is bypassed as FIN	
		1 FOUT is PLL output	
12	LOCKSTS	LOCK status	
		0 PLL is not locked	
		1 PLL is locked	
8	PREDIV	FIN predivider	
		0 FIN divided by 1	
		1 FIN divided by 2	
7 4	FBCTRL	Feedback control	
		0000 M = 6	1000 M = 32
		0001 M = 8	1001 M = 36
		0010 M = 10	1010 M = 40
		0011 M = 12	1011 M = 64
		0100 M = 16	1100 Not available
		0101 M = 18	1101
		0110 M = 20	1110
		0111 M = 26	1111
3 0	POSTDIV	Post divider control	
		000 N = 1	
		001 N = 2	
		010 N = 3	
		011 N = 4	

100	N = 6
101	N = 8
110	Not available
111	N =16

VDCCON VDC Control Register

The On-chip VDC control register, VDCTRIM, is used for the trim value of VDC output. To modify the VDCTRIM bit, write **1** to VDCTE. The VDCWDLY value can be written when writing **1** to the VDCDE bit simultaneously.

VDCCON=0x4000_0064

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BMRTE					BMRTRIM			VDCTE				VDCTRIM											VDCDE	VDCWDLY							
0	0	0	0	0	100			0	0	0	0	0000				0	0	0	0	0	0	0	0	0x7F							
W					RW							RW											W	RW							

31	BMRTE	Reference BGR trim write enable.
		0 BMRTRIM field is not updated by writing
		1 BMRTRIM filed can be updated by writing
26 24	BMRTRIM	Reference BGR output voltage trim value
23	VDCTE	VDCTRIM value write enable. Write only with VDCTRIM value.
		0 VDCTRIM field is not updated by writing
		1 VDCTRIM filed can be updated by writing
19 16	VDCTRIM	VDC output voltage trim value
8	VDCDE	VDCWDLY value write enable. Write only with VDCWDLY value
		0 Disable writing warm-up delay count value
		1 Enable writing warm-up delay count value
7 0	VDCWDLY	VDC warm-up delay count value. When SCU is woken up from power-down mode, the warm-up delay is inserted to stabilize VDC output. The amount of delay can be defined with this register value 7F : 2msec

LVDCON LVD Control Register

The LVD Control Register is an on-chip brown-out detector control register. There are four voltage levels that can be set for the Low Voltage Detect monitoring and the ability to trim the monitoring voltages. This register is a 32-bit register.

LVDCON=0x4000_0068

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								LVDTE						LVDTRIM		SELEN						LVDSEL										LVDLVL	LVDEN
0	0	0	0	0	0	0	0	0	0	0	0	0	0	00		0	0	0	0	0	0	00		0	0	0	0	0	0	0	1
								W						RW		W						RW								R	RW

23	LVDTE	LVDTRIM value write enable. Write only with LVDTRIM value.
		0 LVDTRIM field is not updated by writing
		1 LVDTRIM filed can be updated by writing
17 16	LVDTRIM	LVD voltage level trim value It can writable when trim enable mode in FMC
15	SELEN	LVDSEL value write enable. Write only with LVDSEL value.
		0 LVDSEL field is not updated by writing
		1 LVDSEL filed can be updated by writing
9 8	LVDSEL	LVD detect level select 00 LVD detect level is 1.8V- 50mV 01 LVD detect level is 2.2V – 50mV 10 LVD detect level is 2.7V -50mV 11 LVD detect level is 4.3V – 50mV
1	LVDLVL	LVD Level
		0 LVD level is not detected
		1 LVD level is detected
0	LVDEN	LVD Function enable
		0 LVD is not enabled
		1 LVD is enabled

IOSCTRIM Internal OSC Trim Register

This is the internal oscillator frequency trim register, which is a 32-bit register.

IOSCTRIM=0x4000_006C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								TSLEN					TSL			LTEN		LT				LTM		UDCEN			UDCH			UDCL	
0	0	0	0	0	0	0	0	0	0	0	0	0	000			0	0	0000				00		0	0	0	00			000	
								W					RW			W		RW				RW		W			RW			RW	

23	TSLEN	TSL trim value write enable. Write only with TSL trim value.
		0 TSL field is not updated by writing

		1 TSL filed can be updated by writing
18 16	TSL[2:0]	TSL trim value
15	LTEN	LTM/LT value write enable. Write only with LTM/LT value
		0 LT field is not updated by writing
		1 LT filed can be updated by writing
13 8	LTM/LT	Internal oscillator LT trim value Not recommended strongly to write into this field
7	UDCEN	UDCH/UDCL value write enable. Write only with UDC value
		0 UDC field is not updated by writing
		1 UDC filed can be updated by writing
4 0	UDCH/UDCL	Internal oscillator UDC trim value Not recommended strongly to write into this field

All trim bits are writable when Trim mode is enabled.

EOSCR External Oscillator Control Register

The External Oscillator control register provides the configuration of the external oscillator connections. The current and amplification types can be modified. The external main crystal oscillator has two characteristics. For noise immunity, the NMOS amp type is recommended and for the low power characteristic, the INV amp type is recommended. This register is a 16-bit register.

EOSCR=0x4000_0080

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISELEN						ISEL		AMPEN							AMPSEL
0	0	0	0	0	0	11		0	0	0	0	0	0	0	0
W						RW		W							RW

15	ISELEN	Write enable of bit field ISEL.
		0 Write access of ISEL field is masked
		1 Write access of ISEL field is accepted
9	ISEL	Select current.
8		00 Minimum current driving option
		01 Low current driving option
		10 High current driving option
		11 Maximum current driving option
7	AMPEN	Write enable of bit field AMPSEL
		0 Write access of AMPSEL field is masked
		1 Write access of AMPSEL field is accepted
0	AMPSEL	Select amplifier type
		0 NMOS type
		1 Inverter type

OPAnTRIM Internal OPAMP n Trim Register

OPA0TRIM Internal OPAMP 0 Trim Register
OPA1TRIM Internal OPAMP 1 Trim Register
OPA2TRIM Internal OPAMP 2 Trim Register
OPA3TRIM Internal OPAMP 3 Trim Register

The Internal OPAMP Trim Register trims the OPAMP.

OPATRIM0=0x4000_0070, OPATRIM1=0x4000_0074
OPATRIM2=0x4000_0078 , OPATRIM3=0x4000_007C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								ABMEN						ABM		GTRIMEN				GTRIMH		GTRIML		ATRIMEN				ATRIM			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	00		0	0	0	0	00		10		0	0	0	0	0000			
								W						RW		W				RW		RW		W				RW			

23	ABMEN	ABM trim value write enable. Write only with ABM trim value.
		0 ABM field is not updated by writing
		1 ABM filed can be updated by writing
18 16	ABM[1:0]	OPAMP BIAS trim value
15	GTRIMEN	GTRIM value write enable. Write only with GTRIM value
		0 GTRIM field is not updated by writing
		1 GTRIM filed can be updated by writing
11 8	GTRIMHL[1:0]/GTRIML[1:0]	OPAMP Gain trim value GAINH[1:0],GAINL[1:0]
7	ATRIMEN	ATRIM value write enable. Write only with ATRIM value
		0 ATRIM field is not updated by writing
		1 ATRIM filed can be updated by writing
3 0	ATRIM[3:0]	OPAMP VIO (Offset) Trimming value

EMODR External Mode Status Register

The External Mode Status Register shows the status of the external mode pins while booting. This register is an 8-bit register.

EMODR=0x4000_0084

7	6	5	4	3	2	1	0
					SCANMD	TEST	BOOT
		0x0			0	0	-
		R			R	R	R

2	SCANMD	SCANMD pin level
		0 SCANMD pin is low
		1 SCANMD pin is high
1	TEST	TEST pin level
		0 TEST pin is low
		1 TEST pin is high
0	BOOT	BOOT pin level
		0 BOOT pin is low
		1 BOOT pin is high

DBCLK1 Debounce Clock Control Register 1

The Debounce Clock Control Register 1 controls the debounce timing configuration for Port A and Port B.

DBCLK1=0x4000_009C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					PBDCSEL			PBDDIV														PADCSEL			PADDIV							
0	0	0	0	0	000			0x01								0	0	0	0	0	000			0x01							
					RW			RW														RW			RW							

26 24	PBDCSEL	Debounce Clock for Port B source select bit 0xx RING OSC 1Mhz 100 MCLK (bus clock) 101 INT OSC 20MHz 110 External Main OSC 111 PLL Clock
23 16	PBDDIV	PORT B Debounce Clock N divider PORT B Debounce clock = Clock source / PBDDIV (If PBDDIV is 0, input clock will be stopped)
10 8	PADCSEL	Debounce Clock for Port A source select bit 0xx RING OSC 1Mhz 100 MCLK (bus clock) 101 INT OSC 20MHz 110 External Main OSC 111 PLL Clock
7 0	PADDIV	PORT A Debounce Clock N divider PORT A Debounce clock = Clock source / PADDIV (If PADDIV is 0, input clock will be stopped)

DBCLK2 Debounce Clock Control Register 2

The Debounce Clock Control Register 2 controls the debounce timing configuration for Port C and Port D.

DBCLK2=0x4000_00A0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					PDDCSEL			PDDDIV														PCDCSEL			PCDDIV							
0	0	0	0	0	000			0x01								0	0	0	0	0	000			0x01							
					RW			RW														RW			RW							

26 24	PDDCSEL	Debounce Clock for PORT D source select bit 0xx RING OSC 1Mhz 100 MCLK (bus clock) 101 INT OSC 20MHz 110 External Main OSC 111 PLL Clock
23 16	PDDDIV	PORT D Debounce Clock N divider PORT D Debounce clock = Clock source / PDDDIV (If PDDDIV is 0, input clock will be stopped)
10 8	PCDCSEL	Debounce Clock for PORT C source select bit 0xx RING OSC 1Mhz 100 MCLK (bus clock) 101 INT OSC 20MHz 110 External Main OSC 111 PLL Clock
7 0	PCDDIV	PORT C Debounce Clock N divider PORT C Debounce clock = Clock source / PCDDIV (If PCDDIV is 0, input clock will be stopped)

MCCR1 Miscellaneous Clock Control Register 1

The Miscellaneous Clock Control Register 1 controls the configuration for both the Trace and the System Tick clocks.

MCCR1=0x4000_0090

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRCPOL					TRCSEL			TRACEDIV													STCSEL			STDIV							
0	0	0	0	0	100			0x04								0	0	0	0	0	000			0x01							
W					RW			RW													RW			RW							

26	TRCSEL	TRACE Clock source select bit	
24		0xx	RING OSC 1Mhz
		100	MCLK (bus clock)
		101	INT OSC 20MHz
		110	External Main OSC
		111	PLL Clock
23	TRACEDIV	TRACE Clock N divider	
16		TRACE Clock = CLK_IN/DIV	
		(If TRACEDIV is 0, input clock will be stopped)	
10	STCSEL	SYSTIC Clock source select bit	
8		0xx	RING OSC 1Mhz
		100	MCLK (bus clock)
		101	INT OSC 20MHz
		110	External Main OSC
		111	PLL Clock
7	STDIV	SYSTIC Clock N divider	
0		Systick input clock = Clock source / STDIV	
		(If STDIV is 0 or 1, input clock will be stopped)	

MCCR2 Miscellaneous Clock Control Register 2

The Miscellaneous Clock Control Register 2 controls the configuration of MPWM0 and MPWM1 clocks.

MCCR2=0x4000_0094

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					PWM1CSEL			PWM1DIV														PWM0CSEL			PWM0DIV							
0	0	0	0	0	000			0x00								0	0	0	0	0	000			0x00							
					RW			RW														RW			RW							

26 24	PWM1CSEL	PWM1 Clock source select bit 0xx RING OSC 1Mhz 100 MCLK (bus clock) 101 INT OSC 20MHz 110 External Main OSC 111 PLL Clock
23 16	PWM1DIV	PWM1 Clock N divider PWM1 input clock = Clock source / PWM1DIV (If PWM1DIV is 0, input clock will be stopped)
10 8	PWM0CSEL	PWM0 Clock source select bit 0xx RING OSC 1Mhz 100 MCLK (bus clock) 101 INT OSC 20MHz 110 External Main OSC 111 PLL Clock
7 0	PWM0DIV	PWM0 Clock N divider PWM0 input clock = Clock source / PWM0DIV (If PWM0DIV is 0, input clock will be stopped)

MCCR3 Miscellaneous Clock Control Register 3

The Miscellaneous Clock Control Register 3 controls the configuration for the Timer EXT0 and WDT clocks.

MCCR3=0x4000_0098

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					TEXT0CSEL			TEXT0DIV														WDTCSEL			WDTDIV							
0	0	0	0	0	000			0x01								0	0	0	0	0	000			0x01							
					RW			RW													RW				RW							

26 24	TEXT0CSEL	TIMER EXT0 Clock source select bit 0xx RING OSC 1Mhz 100 MCLK (bus clock) 101 INT OSC 20MHz 110 External Main OSC 111 PLL Clock
23 16	TEXT0DIV	TEXT0 Clock N divider TEXT0 input clock = Clock source / TEXT0DIV (If TEXT0DIV is 0, input clock will be stopped)
10 8	WDTCSEL	WDT Clock source select bit 0xx RING OSC 1Mhz 100 MCLK (bus clock) 101 INT OSC 20MHz 110 External Main OSC 111 PLL Clock
7 0	WDTDIV	WDT Clock N divider WDT input clock = Clock source / WDTDIV (If WDTDIV is 0, input clock will be stopped)

MCCR4 Miscellaneous Clock Control Register 4

The Miscellaneous Clock Control Register 4 controls the debounce timing configuration for the NMI pin and the clock setting for the ADC peripheral.

0x4000_00A4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					ADCCSEL			ADCCDIV														ETCDSEL			ETCDDIV							
0	0	0	0	0	000			0x00								0	0	0	0	0	000			0x01							
					RW			RW													RW			RW							

26 24	ADCCSEL	ADC clock source select bit 0xx RING OSC 1Mhz 100 MCLK (bus clock) 101 INT OSC 20MHz 110 External Main OSC 111 PLL Clock
23 16	ADCCDIV	ADC Clock N divider ADC clock = Clock source / ADCCDIV (If ADCCDIV is 0, input clock will be stopped)
10 8	ETCDCSEL	Debounce Clock for ETC(NMI) source select bit 0xx RING OSC 1Mhz 100 MCLK (bus clock) 101 INT OSC 20MHz 110 External Main OSC 111 PLL Clock
7 0	ETCDDIV	ETC Debounce Clock N divider ETC clock = Clock source / ETCDDIV (If ETCDDIV is 0, input clock will be stopped)

Functional Description

System Clock Setup Procedure Example for the Internal Clock with PLL

- Configure the FM.CR (after selecting CFG mode through FM.MR) register to the maximum wait
- Enable the internal clock IOSC in the CSCR register.
- Write 0x02 to the SCCR register (system clock control register) to select the IOSC as the PLL source (FIN) with bypassing the PLL output
- In the PLLCON register, Set bits 14,15 to enable PLL, clear bit 13 to bypass PLL output and configure bits 0-8 to the PREDIV/FBCTRL/POSTDIV for desired PLL output. For full speed, the PLLCON register would be set to 0xC110
- Wait for the PLL to be locked by monitoring the LOCK bit (bit 12) in the PLLCON register.
- Set bit 13 of the PLLCON register to enable the PLL output
- Set bit 0 of SCCR to enable the PLL for the system clock
- Set the FM.CR (after selecting CFG mode through FM.MR) register for the appropriate Flash Wait states for the speed selected.

System Clock Setup Procedure Example for the External Clock with PLL

- Enable the Port C peripheral and clock in the SCU PER1 and PCER1 registers
- Unlock the Port Controller using the PORTEN register as defined in PORT CONTROL UNIT (PCU)
- Enable the Alternative function 01b for pins 12 and 13 on PORT C through the PCC_MR register
- Set the Pin type for pins 12 and 13 on PORT C to analog (11b)
- Lock the Port Controller by writing any value to PORTEN register
- Configure the FM.CR (after selecting CFG mode through FM.MR) register to the maximum wait
- If not already enabled, enable Internal oscillator in CSCR
- Set bit 3 in the CMR register to monitor External Oscillator
- Enable External Oscillator in CSCR register
- Wait for bit 0 of the CMR register to be set. Note: if the external oscillator does not start, this bit will never be set.
- Wait for an additional time (more than 1 ms) to allow the oscillator to stabilize
- Write 0x06 to the SCCR register (system clock control register) to select the External Oscillator as the PLL source (FIN)
- Set PLLCON high byte (8-15) to 0xC and low byte (0-7) to the FBCTRL/POSTDIV for desired PLL output.
- Wait until bit 12 of PLLCON is set. Note: if the PLL does not lock, this bit will never be set.
- Set bit 13 to enable the PLL output
- Set bit 0 of SCCR to enable the PLL for the system clock
- Set the FM.CR (after selecting CFG mode through FM.MR) register for the appropriate Flash Wait states for the speed selected.

To Enable Clock Out for monitoring actual clock output

- Enable the Port C peripheral and clock in the SCU PER1 and PCER1 registers
- Unlock the Port Controller using the PORTEN register as defined in PORT CONTROL UNIT (PCU)
- Enable the Alternative function 01b for pin 9 on PORT C through the PCC_MR register
- Set the Pin type for pin 9 on PORT C to output (00b)
- Lock the Port Controller by writing any value to PORTEN register
- Set bit 4 of the COR register (Clock Output Register) to enable the output
- Configure the CLKODIV to the desired output divider

5. Port Control Unit

Overview

The Port Control Unit (PCU) controls the external I/O configuration to:

- Set the multiplex state of each pin (for alternative functions)
- Set external signal type (Analog / Push-Pull output /Open Drain output /Input)
- Set enable/monitor/trigger type for interrupts for each pin
- Set internal pull-up register control for each pin
- Set debounce for each pin

Note: You must enable both the Port Peripheral and the Port Peripheral Clock in PER1/PCER1/ to use the pins of the port.

Figure 5.1 shows a block diagram of the PCU. Figure 5.2 and Figure 5.3 show I/O Port Block diagrams.

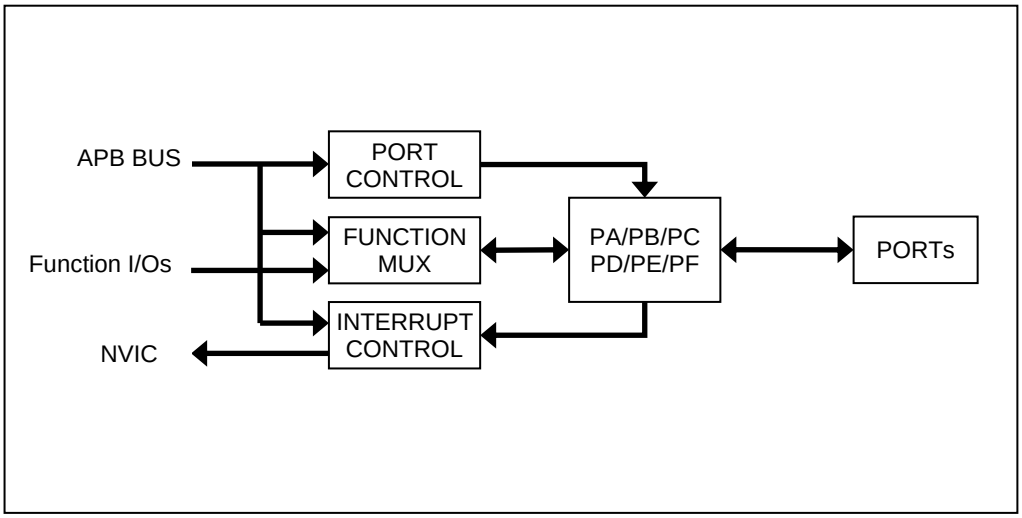


Figure 5.1. PCU Block Diagram

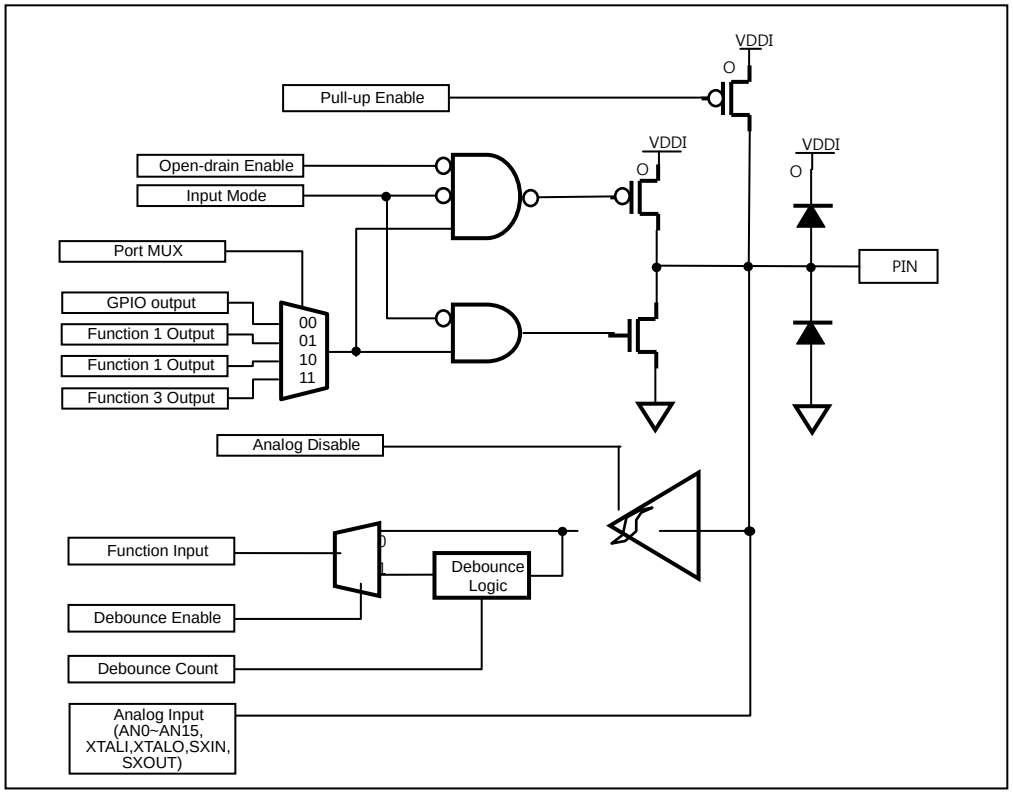


Figure 5.2. I/O Port Block Diagram (ADC and External Oscillator Pins)

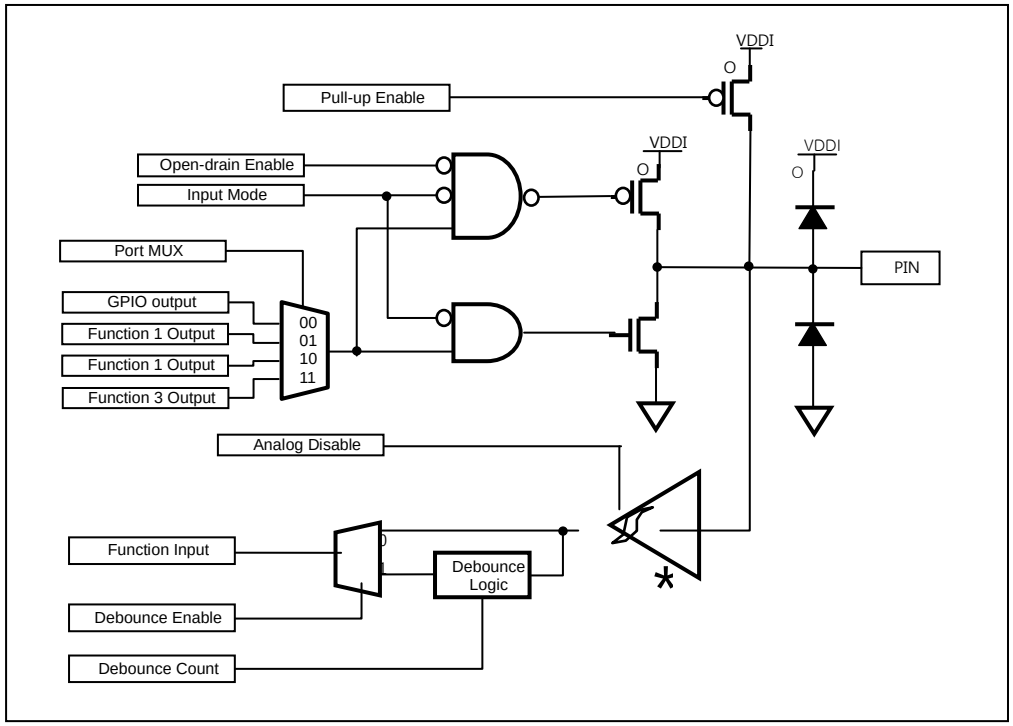


Figure 5.3. I/O Port Block Diagram (General I/O Pins)

Pin Multiplexing

GPIO pins have alternative function pins. Table 5.1 shows the pin multiplexing information.

Table 5.1. GPIO Alternative Function

PORT		FUNCTION			
		00	01	10	11
PA	0	**PA0***			AIN0/COMP0
	1	**PA1***			AIN1/COMP1
	2	**PA2***			AIN2/COMP2
	3	**PA3***			AIN3/COMP3
	4	**PA4***		T0O	AIN4
	5	**PA5***		T1O	AIN5
	6	**PA6***		T2O	AIN6/CREF0
	7	**PA7***	TRACED3	T3O	AIN7/CREF1
	8	**PA8***	TRACECLK	AD0O	AIN8
	9	**PA9***	TRACED0	AD1O	AIN9
	10	**PA10***	TRACED1	AD2O	AIN10
	11	**PA11***	TRACED2		AIN11
	12	**PA12***	SS0	AD2I	AIN12
	13	**PA13***	SCK0		AIN13
	14	**PA14***	MOSI0		AIN14
	15	**PA15***	MISO0		AIN15
PB	0	**PB0***	PWM0H0		
	1	**PB1***	PWM0L0		
	2	**PB2***	PWM0H1		
	3	**PB3***	PWM0L1		
	4	**PB4***	PWM0H2	T9C	
	5	**PB5***	PWM0L2	T9O	
	6	**PB6***	PRTIN0	WDTO(2)	
	7	**PB7***	OVIN0		
	8	**PB8***	PRTIN1	RXD3	
	9	**PB9***	OVIN1	TXD3	
	10	**PB10***	PWM1H0		
	11	**PB11***	PWM1L0		
	12	**PB12***	PWM1H1		
	13	**PB13***	PWM1L1		
	14	**PB14***	PWM1H2		
	15	**PB15***	PWM1L2		

(*) Indicates default pin setting
(2) Indicates secondary port

Table 5.1. GPIO Alternative Function (Continued)

PORT		FUNCTION			
		00	01	10	11
PC	0	PC0	**TCK/SWCLK***		
	1	PC1	**TMS/SWDIO***		
	2	PC2	**TDO/SWO***		
	3	PC3	**TDI***		
	4	PC4	**nTRST***	T0C/PHA(2)	
	5	**PC5***	RXD1	T1C/PHB(2)	
	6	**PC6***	TXD1	T2C/PHZ(2)	
	7	**PC7***	SCL0	T3C	
	8	**PC8***	SDA0		
	9	**PC9***	CLKO	T8O	
	10	PC10	**nRESET***		
	11	**PC11/BOOT***		T8C	
	12	**PC12***	XIN		
	13	**PC13***	XOUT		
	14	**PC14***	RXD0	MISO0(2)	
	15	**PC15***	TXD0	MOSI0(2)	
PD	0	**PD0***	SS1		
	1	**PD1***	SCK1		
	2	**PD2***	MOSI1		
	3	**PD3***	MISO1		
	4	**PD4***	SCL1		
	5	**PD5***	SDA1		
	6	**PD6***	TXD2	AD0I	
	7	**PD7***	RXD2	AD1I	
	8	**PD8***		WDTO	
	9	**PD9***			
	10	**PD10***	AD0SOC	T0C/PHA	
	11	**PD11***	AD0EOC	T1C/PHB	
	12	**PD12***	AD1SOC	T2C/PHZ	
	13	**PD13***	AD1EOC	T3C	
	14	**PD14***	AD2SOC		
	15	**PD15***	AD2EOC		

(*) Indicates default pin setting
(2) Indicates secondary port

Registers

The base address of the PCU block is `0x4000_1000`.

Table 5.2. Base Address of Port

PORT	ADDRESS
PA	0x4000_1000
PB	0x4000_1100
PC	0x4000_1200
PD	0x4000_1300

Table 5.3. PCU Register Map

Register	Offset	R/W	Description
P*n*MR	0x--00	R/W	Port *n* pin mux select register
P*n*CR	0x--04	R/W	Port *n* pin control register
P*n*PCR	0x--08	R/W	Port *n* internal pull-up control register
P*n*DER	0x--0C	R/W	Port *n* debounce control register
P*n*IER	0x--10	R/W	Port *n* interrupt enable register
P*n*ISR	0x--14	R/W	Port *n* interrupt status register
P*n*ICR	0x--18	R/W	Port *n* interrupt control register
PORTEN	0x1FF0	R/W	Port Access enable

PAMR PORT A Pin MUX Register

PAMR is the PA port mode select register. This register and the PERx and PCERx registers must be configured properly before using the port to guarantee its functionality. PERx enables the port and PCERx enables the clock to the port.

PAMR=0x4000_1000

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16	15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
PA15	PA14	PA13	PA12	PA11	PA10	PA9	PA8	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

PORT	SELECTION BIT			
	00	01	10	11
PA0	PA0			AN0_CP0
PA1	PA1			AN1_CP1
PA2	PA2			AN2_CP2
PA3	PA3			AN3_CP3
PA4	PA4		T0O	AN4
PA5	PA5		T1O	AN5
PA6	PA6		T2O	AN6_CREF0
PA7	PA7	TRACED3	T3O	AN7_CREF1
PA8	PA8	TRACECLK	AD0O	AN8
PA9	PA9	TRACED0	AD1O	AN9
PA10	PA10	TRACED1	AD2O	AN10
PA11	PA11	TRACED2		AN11
PA12	PA12	SS0	AD2I	AN12
PA13	PA13	SCK0		AN13
PA14	PA14	MOSI0		AN14
PA15	PA15	MISO0		AN15

PBMR PORT B Pin MUX Register

PBMR is the PB port mode select register. This register and the PERx and PCERx registers must be configured properly before using the port to guarantee its functionality. PERx enables the port and PCERx enables the clock to the port.

PBMR=0x4000_1100

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16	15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
PB15	PB14	PB13	PB12	PB11	PB10	PB9	PB8	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

PORT	SELECTION BIT			
	00	01	10	11
PB0	PB0	MP0UH		
PB1	PB1	MP0UL		
PB2	PB2	MP0VH		
PB3	PB3	MP0VL		
PB4	PB4	MP0WH	T9C	
PB5	PB5	MP0WL	T9O	
PB6	PB6	PRTIN0	WDTO	
PB7	PB7	OVIN0		
PB8	PB8	PRTIN1	RXD3	
PB9	PB9	OVIN1	TXD3	
PB10	PB10	MP1UH		
PB11	PB11	MP1UL		
PB12	PB12	MP1VH		
PB13	PB13	MP1VL		
PB14	PB14	MP1WH		
PB15	PB15	MP1WL		

PCMR PORT C Pin MUX Register

PCMR is the PC port mode select register. This register and the PERx and PCERx registers must be configured properly before using the port to guarantee its functionality. PERx enables the port and PCERx enables the clock to the port.

PCMR=0x4000_1200

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16	15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
PC15	PC14	PC13	PC12	PC11	PC10	PC9	PC8	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
00	00	00	00	00	01	00	00	00	00	00	01	01	01	01	01
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

PORT	SELECTION BIT			
	00	01	10	11
PC0	PC0	TCK_SWCLK		
PC1	PC1	TMS_SWDIO		
PC2	PC2	TDO_SWO		
PC3	PC3	TDI		
PC4	PC4	nTRST	T0C_PHA	
PC5	PC5	RXD1	T1C_PHB	
PC6	PC6	TXD1	T2C_PHZ	
PC7	PC7	SCL0	T3C	
PC8	PC8	SDA0		
PC9	PC9	CLKO	T8O	
PC10	PC10	nRESET		
PC11	PC11(BOOT)		T8C	
PC12	PC12	XIN		
PC13	PC13	XOUT		
PC14	PC14	RXD0	MISO0	
PC15	PC15	TXD0	MOSI0	

PDMR PORT D Pin MUX Register

PDMR is the PD port mode select register. This register and the PERx and PCERx registers must be configured properly before using the port to guarantee its functionality. The PERx enables the port and PCERx enables the clock to the port.

PDMR=0x4000_1300

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16	15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

PORT	SELECTION BIT			
	00	01	10	11
PD0	PD0	SS1		
PD1	PD1	SCK1		
PD2	PD2	MOSI1		
PD3	PD3	MISO1		
PD4	PD4	SCL1		
PD5	PD5	SDA1		
PD6	PD6	TXD2	AD0I	
PD7	PD7	RXD2	AD1I	
PD8	PD8		WDTO	
PD9	PD9			
PD10	PD10	AD0SOC	T0C_PHA	
PD11	PD11	AD0EOC	T1C_PHB	
PD12	PD12	AD1SOC	T2C_PHZ	
PD13	PD13	AD1EOC	T3C	
PD14	PD14	AD2SOC		
PD15	PD15	AD2EOC		

PnCR PORT n Pin Control Register (Except for PCCR)

PnCR is the input or output control of each port pin. Each pin can be configured as an input pin, output pin, or open-drain pin.

PACR=0x4000_1004, PBCR=0x4000_1104, PDCR=0x4000_1304

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16	15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Pn	Port control
00	Push-pull output
01	Open-drain output
10	Input
11	Analog

PCCR PORT C Pin Control Register

This register controls the input or output of each port pin. Each pin can be configured as an input pin, output pin, or open-drain pin.

PCCR=0x4000_1204

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16	15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
11	11	11	11	10	10	11	11	11	11	11	10	10	00	10	10
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Pn	Port control
00	Push-pull output
01	Open-drain output
10	Input
11	Analog

PnPCR PORT n Pull-up Resistor Control Register

Every pin in the port has on-chip pull-up resistors which can be configured by the PnPCR registers.

PAPCR=0x4000_1008, PBPCR=0x4000_1108
PCPCR=0x4000_1208, PDPCR=0x4000_1308

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUE15	PUE14	PUE13	PUE12	PUE11	PUE10	PUE9	PUE8	PUE7	PUE6	PUE5	PUE4	PUE3	PUE2	PUE1	PUE0
0000															
RW															

n	PUEn	Port pull-up control
0	Disable pull-up resistor	
1	Enable pull-up resister	

PnDER PORT n Debounce Enable Register

Every pin in the port has a digital debounce filter which can be configured by the PnDER registers. The Debounce clock can be configured in the DBCLKx registers.

PADER=0x4000_100C, PBDER=0x4000_110C
PCDER=0x4000_120C, PDDER=0x4000_130C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PDE15	PDE14	PDE13	PDE12	PDE11	PDE10	PDE9	PDE8	PDE7	PDE6	PDE5	PDE4	PDE3	PDE2	PDE1	PDE0
0000															
RW															

n	PDEn	Pin debounce enable
		0 Disable debounce filter
		1 Enable debounce filter

PnIER PORT n Interrupt Enable Register

Each individual pin can be an external interrupt source. The edge trigger interrupt and level trigger interrupt are both supported. The interrupt mode can be configured by setting the PnIER registers.

PAIER=0x4000_1010, PBIER=0x4000_1110
PCIER=0x4000_1210, PDIER=0x4000_1310

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16	15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
PIE15	PIE14	PIE13	PIE12	PIE11	PIE10	PIE9	PIE8	PIE7	PIE6	PIE5	PIE4	PIE3	PIE2	PIE1	PIE0
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

PIEn	Pin interrupt enable
	00 Interrupt disabled
	01 Enable interrupt as level trigger mode
	10 Reserved
	11 Enable interrupt as edge trigger mode

PnISR PORT n Interrupt Status Register

When an interrupt is delivered to the CPU, the interrupt status can be detected by reading the PnISR register. The PnISR register will report a source pin of interrupt and a type of interrupt.

PAISR=0x4000_1014, PBISR=0x4000_1114
PCISR=0x4000_1214, PDISR=0x4000_1314

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16	15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
PIS15	PIS14	PIS13	PIS12	PIS11	PIS10	PIS9	PIS8	PIS7	PIS6	PIS5	PIS4	PIS3	PIS2	PIS1	PIS0
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

PISn	Pin interrupt status
00	No interrupt event
01	Low level interrupt or Falling edge interrupt event is present
10	High level interrupt or rising edge interrupt event is present
11	Both of rising and falling edge interrupt event is present in edge trigger interrupt mode. Not available in level trigger interrupt mode

PnICR PORT n Interrupt Control Register

This is the Interrupt mode control register.

PAICR=0x4000_1018, PBICR=0x4000_1118
PCICR=0x4000_1218, PDICR=0x4000_1318

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16	15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
PIC15	PIC14	PIC13	PIC12	PIC11	PIC10	PIC9	PIC8	PIC7	PIC6	PIC5	PIC4	PIC3	PIC2	PIC1	PIC0
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

PICn	Pin interrupt mode
00	Prohibit external interrupt
01	Low level interrupt or Falling edge interrupt mode
10	High level interrupt or rising edge interrupt mode
11	Both of rising and falling edge interrupt mode. Not support for level trigger mode

PORTEN Port Access Enable Register

The Port Access Enable register enables the register writing permission of all PCU registers.

PORTEN=0x4000_1FF0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								PORTEN							
0	0	0	0	0	0	0	0	--							
								WO							

7 0	PORTEN	Writing the sequence of 0x15 and 0x51 in this register enables writing to PCU registers, and writing other values protects all PCU registers from writing.

Functional Description

All the GPIO pins can be configured for different operations – inputs, outputs, and triggered interrupts (both level and edge) through the PDU. The system is also able to disable ports by setting the PER1 and PCER1 registers in the SCU. By default, all pins are disabled (except for UART0/SPI0) so the developer must enable these to operate.
All configuration parameters are protected by the Port Access Enable register. You must write the sequence in order (`0x15`, `0x51`) to the PORTEN register to configure any pin(s). Once the configuration is complete, write any other value to the PORTEN register to lock it.

Note: Do not read in between the sequence; it will prevent the configuration registers from being unlocked.

When the input function of I/O port is used by the Pin Control Register, the output function of I/O port is disabled. The Port Function differs according to the Pin Mux Register. The Input Data Register captures the data present on the I/O pin or debounced input data at every GPIO clock cycle.

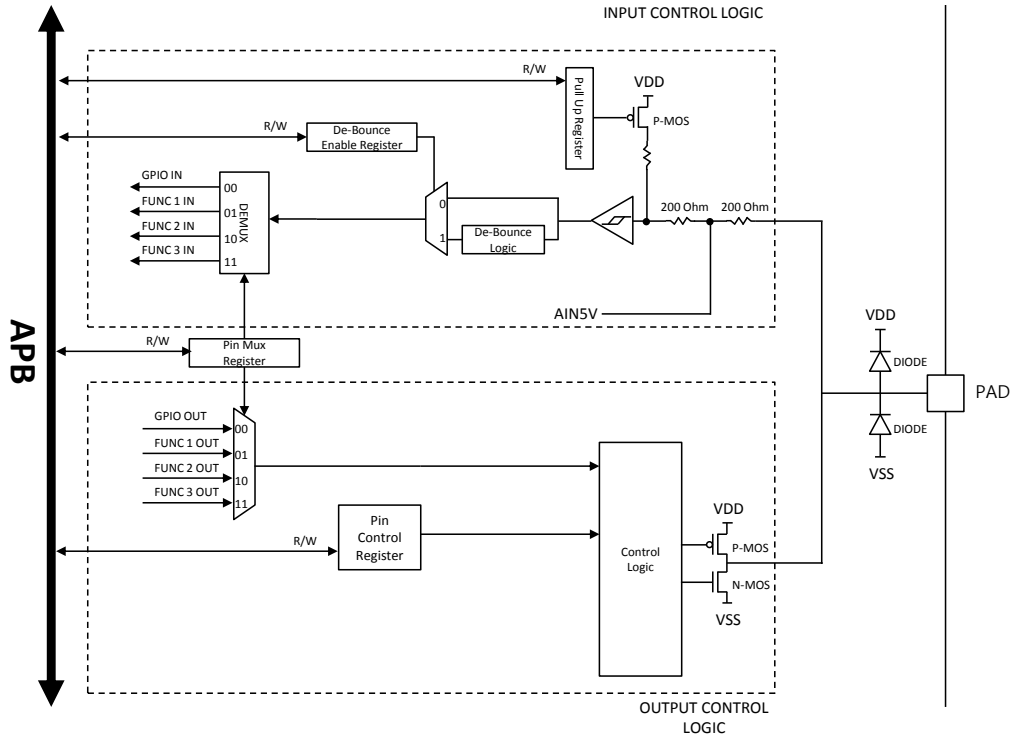


Figure 5.4. Port Diagram

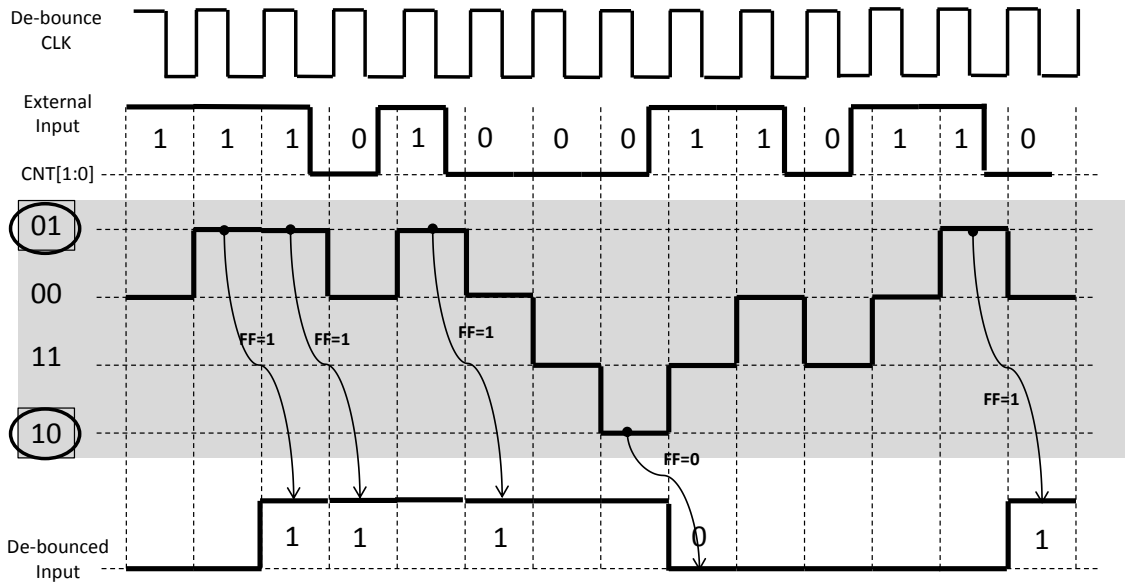


Figure 5.5. Debounce Diagram

When the debounce function of input data is used by the Debounce Enable Register, the external input data is captured by the debounce clock.

- If CNT value is “01”, debounced input data is “1”.
- If CNT Value is “10”, debounced input data is “0”

It is possible to change the Debounce CLK of each port group used by the DBCLK Registers.

6. General Purpose I/O

Overview

Most pins, except the dedicated function pins, can be used as general I/O ports. General input/output ports are controlled by the GPIO block.

- Output signal level (H/L) select
- Input signal level
- Output Set/Clear pin by writing a **1**

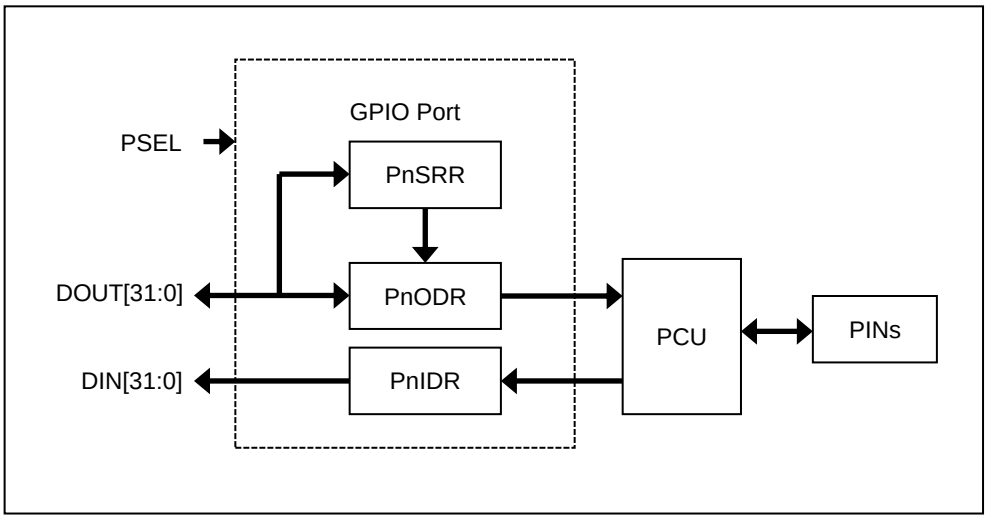


Figure 6.1. Block Diagram

Pin Description

Table 6.1. External Signal

PIN NAME	TYPE	DESCRIPTION
PA	IO	PA0 - PA15
PB	IO	PB0 - PB15
PC	IO	PC0 - PC15
PD	IO	PD0 - PD15

Registers

The base address of GPIO is 0x4000_2000 and register map is described in Table 6.2 and Table 6.3.

Table 6.2. Base Address of Each Port

PORT	Address
PA PORT	0x4000_2000
PB PORT	0x4000_2100
PC PORT	0x4000_2200
PD PORT	0x4000_2300

Table 6.3. GPIO Register Map

Name	Offset	R/W	Description	Reset
PnODR	0x--00	R/W	Port *n* Output data register	0x00000000
PnIDR	0x--04	RO	Port *n* Input data register	0x00000000
PnBSR	0x--08	WO	Port *n* Pin set register	0x00000000
PnBCR	0x—0C	WO	Port *n* Pin clear register	0x00000000

PnODR PORT n Output Data Register

When the pin is set as output and GPIO mode, the pin output level is defined by the PnODR registers.

PAODR=0x4000_2000, PBODR=0x4000_2100
PCODR=0x4000_2200, PDODR=0x4000_2300

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ODR															
0000															
RW															

ODR	Pin output level
	0 Output low level
	1 Output high level

PnIDR PORT n Input Data Register

Each pin level status can be read in the PnIDR register. Even if the pin is in an alternative mode except analog mode, the pin level can be detected in the PnIDR register.

PAIDR=0x4000_2004, PBIDR=0x4000_2104
PCIDR=0x4000_2204, PDIDR=0x4000_2304

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDR															
0000															
RO															

IDR	Pin current level
	0 The pin is low level
	1 The pin is high level

PnBSR PORT n Bit Set Register

PnBSR is a register for controlling each bit of the PnODR register. When you write **1** to a specific bit, the corresponding bit in the PnODR register will be set.

PABSR=0x4000_2008, PBBSR=0x4000_2108
PCBSR=0x4000_2208, PDBSR=0x4000_2308

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BSR															
0000															
WO															

BSR	Pin current level
0	Not effect
1	Set correspondent bit in PnODR register

PnBCR PORT n Bit Clear Register

PnBRR is a register for controlling each bit of the PnODR register. When you write **1** to a specific bit then the correspondent bit in the PnODR register will be clear.

PABCR=0x4000_200C, PBBCR=0x4000_210C
PCBCR=0x4000_220C, PDBCR=0x4000_230C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCR															
0000															
WO															

BCR	Pin current level
0	Not effect
1	Clear correspondent bit in PnODR register

Functional Description

The GPIO registers provide the input/output condition of the GPIO pins. The input data registers give the states of the pins of the ports. The output data register is used to set the port pins. The Set and Clear registers control the pins at the individual level.

When configured as output, the value written to the GPIO Output Data Register is output on the I/O Pin. When setting the Bit Set Register, set the GPIO Output Data Register high. When setting the Bit Clear Register, set the GPIO Output Data Register low.

The Input Data Register captures the data present on the I/O pin or debounced input data at every GPIO clock cycle.

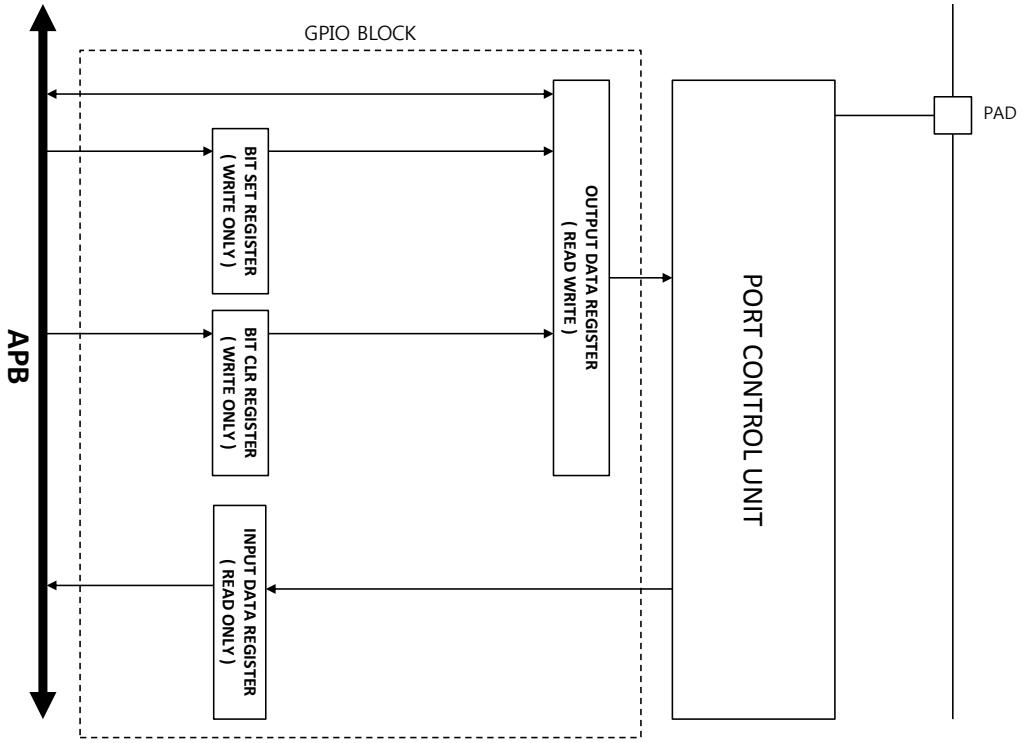


Figure 6.2. GPIO Diagram

7. Flash Memory Controller

Introduction

Flash Memory Controller is an internal Flash memory interface controller with the following features:

- 128 KB Flash code memory
- 32-bit read data bus width
- Code cache block for fast access mode
- 128-byte page size
- Support page erase and macro erase
- 128-byte unit program

Table 7.1. Internal Flash Specification

Item	Description
Size	128KB
Start Address	0x0000_0000
End Address	0x0001_FFFF
Page Size	128-byte
Total Page Count	1,024 pages
PGM Unit	128-byte
Erase Unit	128-byte

Figure 7.1 shows a block diagram of the Flash Memory Controller.

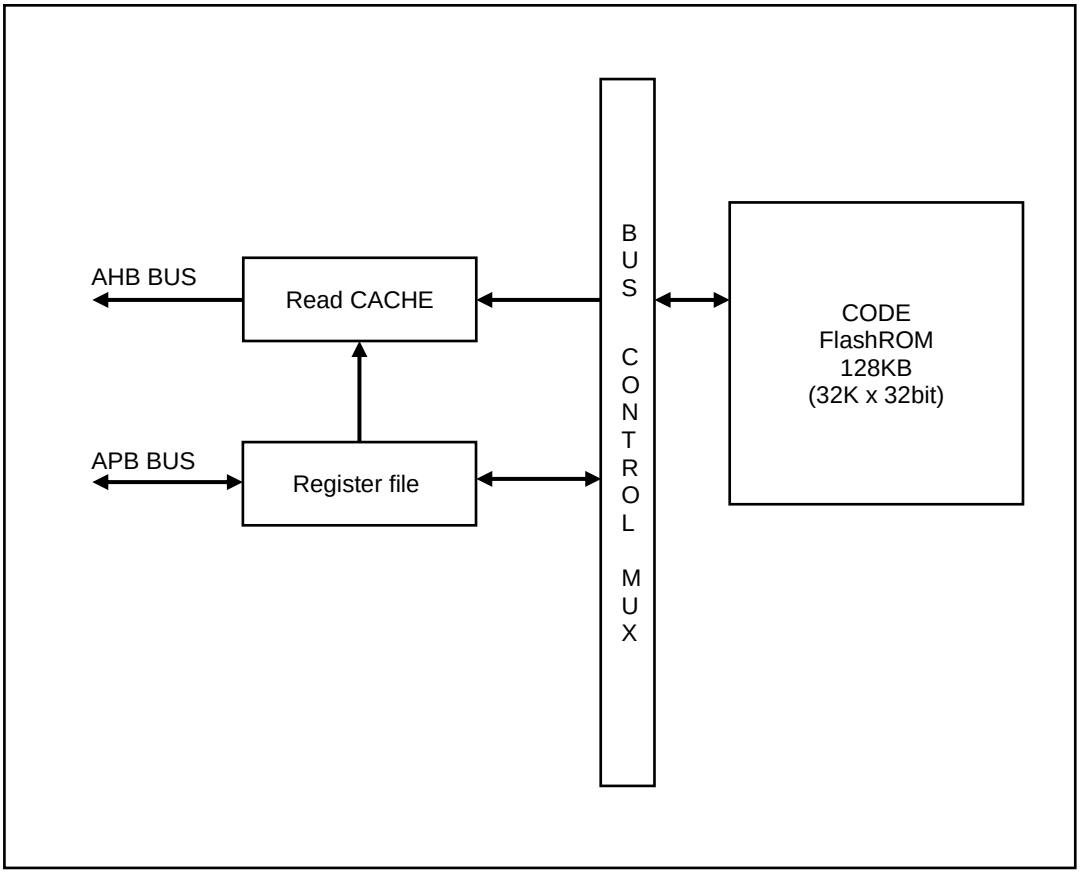


Figure 7.1. Flash Memory Controller Block Diagram

Pin Description

There are no external interface pins for this peripheral.

Registers

The base address of the Flash Memory Controller is shown in Table 7.2.

Table 7.2. Flash Memory Controller Base Address

	Address
Flash Controller	0x4000_0100

Table 7.3 shows the register memory map.

Table 7.3. Flash Memory Controller Register Map

Name	Offset	R/W	Description	Reset
FMMR	0x0004	R/W	Flash Memory Mode Select register	0x01000000
FMCR	0x0008	R/W	Flash Memory Control register	0x82000000
FMAR	0x000C	R/W	Flash Memory Address register	0x00000000
FMDR	0x0010	R/W	Flash Memory Data register	0x00000000
FMTMR	0x0014	R/W	Flash Memory Timer register	0x000000bb
FMDRTY	0x0018	R/W	Flash Memory Dirty bit	
FMTICK	0x001C	RO	Flash Memory Tick Timer	0x00000000
FMCRC	0x0020	RO	Flash Memory Read CRC Value	
BOOTCR	0x0074	R/W	Boot ROM Remap Clear register	0x00000000

FMMR Flash Memory Mode Register

FMMR is the internal Flash Memory Mode Register. This register is a 32-bit register.

FMMR=0x4000_0104

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOOT							IDLE	VERIFY	AMBAEN					TRMEN	TRM							FEMOD	FMOD	ACODE							
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x00							
							R	RW	RW					RW	RW							R	R	RW							

31	BOOT	0	
		1	Boot mode enable status(read only)
24	IDLE	0	
		1	Boot mode enable status(read only)
23	VERIFY	0	
		1	Flash Verify mode enable status(read only)
22	AMBAEN	0	AMBA mode disable
		1	AMBA mode enable (can change wait state and etc)
17	TRMEN	0	
		1	Trim mode entry status(read only)
16	TRM	0	
		1	Trim mode status(read only)
9	FEMOD	0	
		1	Flash mode entry status(read only)
8	FMOD	0	
		1	Flash mode status(read only)
7 0	ACODE	5A → A5	Flash mode
		A5 → 5A	Trim mode
		81 →28	CFG mode (FMCR[31:24])

FMCR Flash Memory Control Register

FMCR is the internal Flash Memory Control Register.

FMCR=0x4000_0108

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HRESPD	TRIM2	TRIM1	TRIM0	PCLK2		CLK4	CLK3	CRCINIT	CRCEN		TIMER			TEST1	TEST0	VPPOUT	EVER	PVER		OTPBE	OTPAE		AE			PMOD	WE	PBLD	PGM	ERS	PBR
1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	RW	RW	RW	RW		RW	RW	RW	RW		RW			RW	RW	R	RW	RW		RW	RW		RW			RW	RW	RW	RW	RW	RW

31	HRESPD		Disable HRESP(error response function) of Data or System bus (HRESP is AMBA AHB signal)
27	PCLK2		Set this bit when PCLK is 1/2 of HCLK (default PCLK = HCLK) It affects state machine of PMODE operation
25	CLK4	0	If CLK4, CLK3 are 00, flash access in 5 cycles
		1	Flash access in 4 cycles
24	CLK3	0	If CLK4, CLK3 are 00, flash access in 5 cycles
		1	Flash access in 3 cycles
23	CRCINIT	0 1	CRC register will be initialized. It should be reset again before read flash to generate CRC16 calculation (Initial value of FMCRC is 0xFFFF)
22	CRCEN	0	CRC16 enable
		1	CRC value will be calculated at every flash read timing
20	TIMER	0	Program/Erase timer enable
		1	(timer can be enable by PGM or ERS bit)
17 16	TEST[1:0]	00	Normal operation
		01	(read) Row voltage mode
		01	(write) ODD Row program
		10	Even Row program
		11	All Row program
15	VPPOUT		Enable charge-pump Vpp output
14	EVER		Set erase verify mode
13	PVER		Set program verify mode
11	OTPBE		OTP area B enable
10	OTPAE		OTP area A enable
8	AE		All erase enable
5	PMODE		PMODE enable(Address path changing)
4	WE		Write enable
3	PBLD		Page buffer load(WE should be set)
2	PGM		Program enable
1	ERS	0	Program mode enable
		1	Erase mode enable
0	PBR		Page buffer reset

FMAR Flash Memory Address Register

FMAR is the internal Flash memory program erase address register.

FMAR=0x4000_010C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FADDR														
0	0x0000														
	RW														

14 0	FADDR	32K words address (one word = 4 bytes)

FMDR Flash Memory Data Register

FMDR is the internal Flash memory program data register.

FMDR=0x4000_0110

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FDATA																															
0x0000_0000																															
RW																															

31 0	FDATA	Flash PGM data (32-bit)

FMTMR Flash Memory Timer Register

The Internal Flash Memory Timer value register (9-bit), Erase/Program timer runs up to {TMR[8:0],0xFF}

FMTMR=0x4000_0114

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							TMR								
0	0	0	0	0	0	0	0x0BB								
							RW								

8 0	TMR	Erase/PGM timer (default, 0xBB) Timer counts up to {TMR[8:0], 0xFF} by 20MHz int. OSC clock

FMDRTY Flash Memory Dirty Bit Register

FMDRTY is the internal Flash memory dirty bit clear register.

FMDRTY=0x4000_0118

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FDRTY																															
-																															
WO																															

31 0	FDRTY	Write any value here, cache line fill flag will be cleared.

FMTICK Flash Memory Tick Timer Register

FMTICK is the internal Flash memory Burst Mode channel selection register.

FMTICK=0x4000_011C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														FTICK																	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x00000															
																RW															

17 0	FTICK	TICK goes to 0x3FFFF from written TICK value while TRM runs by PCLK clock

FMCRC Flash Memory CRC Value Register

The register shows the CRC value resulting from read accesses on internal Flash memory.

FMTICK=0x4000_0120

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																CRC															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	--															
																RO															

15 0	CRC	CRC16 value

BOOTCR Boot ROM Remap Clear Register

The Boot ROM remap clear register is an 8-bit register.

BOOTCR=0x4000_0174

7	6	5	4	3	2	1	0
							BOOTROM
0	0	0	0	0	0	0	1
							R

0	BOOTROM	Boot Mode (only can be written in boot loader mode) This bit is used to clear boot loader mode at end of boot code (when BOOTROM low, external BOOT pin signal is masked)

Functional Description

Flash Memory Controller is an internal Flash memory interface controller which primarily controls the programming of Flash memory and preparing read data to be requested from the bus.

Flash Organization

The 128 Kbytes code Flash memory consists of 1,024 pages which have a uniform 128 bytes page size. The Flash controller allows reading or writing of Flash memory data. This memory is located at `0x0000_0000` address on the system memory map. The system boot address is `0x0000_0000`; therefore, this Flash memory is boot memory. The code data which is programmed in Flash memory will boot up the device after the boot ROM sequence is completed.

Flash Read Operation

The Flash data read operation is requested from the bus. The Flash controller responds to the request. The wait time should be correctly defined because the bus speed is usually faster than Flash data access time.

The normal read operation is not available in Flash mode in the `ACODE.FM.MR` field.

Flash Program Operation

The erase and program access of Flash memory is available only in Flash mode in the ACODE.FM.MR field. Therefore, self-programming is not supported. The Flash program/erase operation should be performed by the execution program on the SRAM memory.

The Flash program operation writes one page to the target address selected by the FM.AR register. At first, users should write the program data into the page buffer. Page buffer write is performed by word write access to the FM.DR register at FM.AR address. After filling the page buffer, users can begin the Flash write operation and should wait for the IDLE bit to be set.

Figure 7.2 shows the page buffer loading operation.

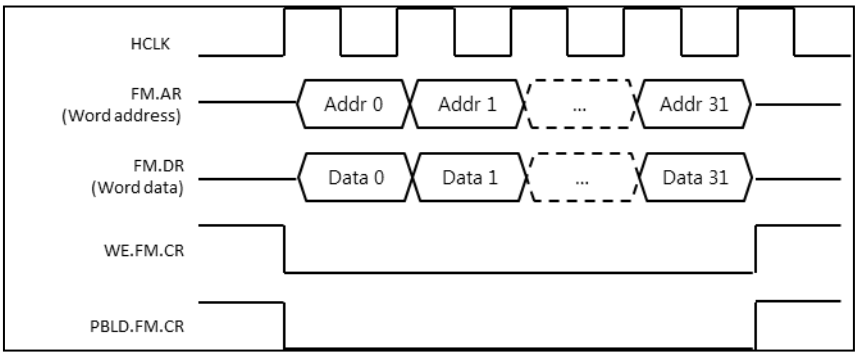


Figure 7.2. Page Buffer Load Timing Diagram

The Flash write of page buffer data is done by the PRGM.FM.CR command. A safe writing operation requires the correct program time. The tPGM program time is defined by the FM.TMR register. When the timer is activated (TIMER.FM.CR bit is set), the IDLE.FM.MR bit is cleared and the Flash controller will start counting the HCLK pulses until the pulse count matches the value in the FM.TMR. When the count is reached, the Flash controller will set the IDLE.FM.MR bit to show the time has elapsed. In this page write operation, the target page address should be written in the FM.AR register.

Figure 7.3 shows the page write operation.

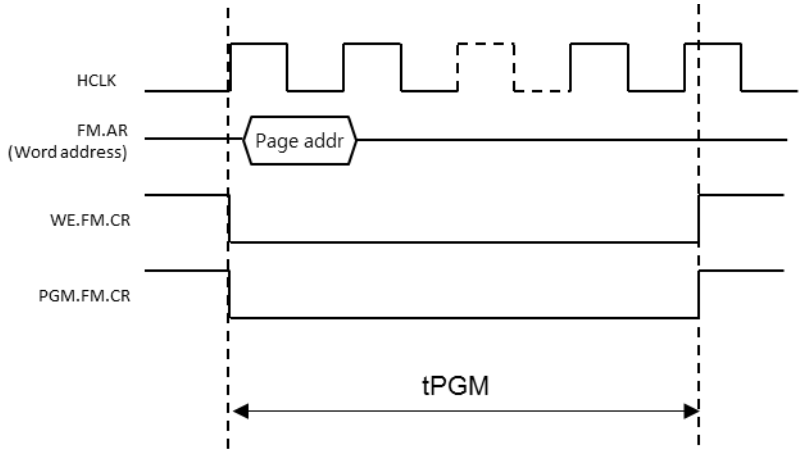


Figure 7.3. Page Write Timing Diagram

Flash Erase Operation

The erase and program access of Flash memory is available only in Flash mode in the ACODE.FM.MR field. Therefore, self-programming is not supported. The Flash program/erase operation should be performed by the execution program on the SRAM memory.

Two types of Flash erase operations are supported – Page erase and Bulk erase. The page erase operation erases one page to the target address selected by the FM.AR register. User starts the Flash write operation and should wait for the IDLE bit to be set. The bulk erase operation erases the entire Flash memory data and the FM.AR address is ignored. The process is the same between page and bulk erase with the exception of the AE.FM.CR bit. When the AE.FM.CR bit is set, the Flash controller will perform a bulk erase.

Figure 7.4 shows the page erase operation.

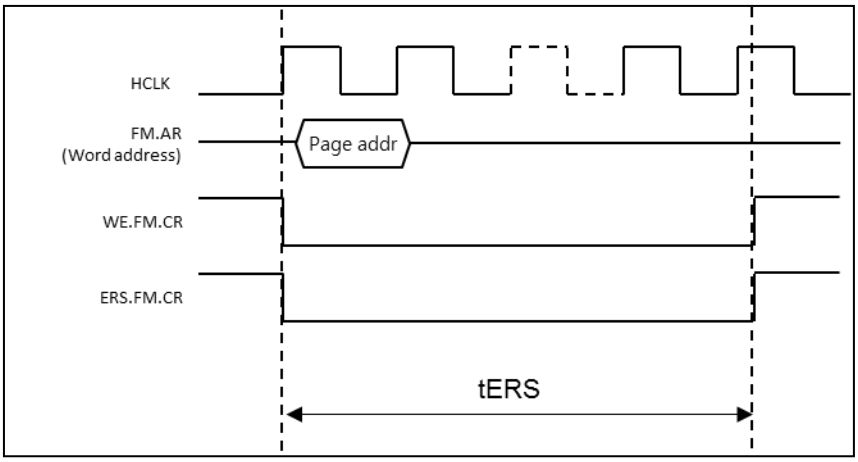


Figure 7.4. Page Erase Timing Diagram

Flash erase is done by the ERS.FM.CR command. A safe writing operation requires the correct program time. The tERS erase time is defined by the FM.TMR register. When the timer is activated (TIMER.FM.CR bit is set), the IDLE.FM.MR bit is cleared and the Flash controller will start counting the HCLK pulses until the pulse count matches the value in the FM.TMR. When the count is reached, the Flash controller will set the IDLE.FM.MR bit to show the time has elapsed.

Figure 7.5 shows the bulk erase operation.

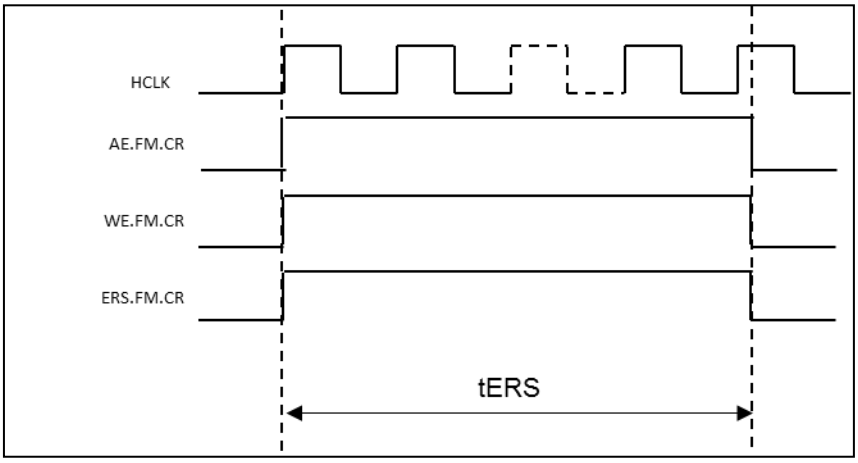


Figure 7.5. Bulk Erase Timing Diagram

The Flash area can be read from directly via the memory address. Writing of Flash memory can be done through Boot mode or in-application programming. The execution for the writing of Flash must occur from the RAM area. The Flash controller cannot read Flash memory (including instructions) once the program bit has been set.
Caution: If the vector table is not placed in RAM, you MUST disable interrupts so as to prevent reading the interrupt service routine in Flash.

To write to Flash memory:

1. Disable the Watch Dog Timer (if enabled).
2. Set the clock to internal oscillator (20 MHz).
3. Write configuration sequence to the MR register to enable Control Register upper bits (24-31).
4. Configure upper bits of the Control Register (HRESPD and Flash access of 4 cycles).
5. Lock the Flash controller by writing `0x00` to the MR register.

6. Write the Flash sequence to the MR register.
7. Clear the DRTY register.
8. Set the PMODE bit in the CR register to set Program Mode.
9. Wait until the IDLE bit in MR register is set.
10. Reset the page buffer (setting and clearing PBR bit in CR register).
11. Set the PBLD bit in the CR register to allow the page buffer to be written to.
12. Write to the page buffer by loading the DR register with each 32-bit word of the page.
13. Clear the PBLD bit in the CR register.
14. Write the address to which the page buffer will be written (in 32-bit words) to the AR register.
15. Set the PGM bit in the CR register.
16. Set the WE bit in the CR register to start writing.
17. Wait until the IDLE bit in MR register is set.
18. Clear the WE bit in the CR register.
19. Clear the PGM bit in the CR register. If there are additional pages to write, repeat the process, starting at step 10.
20. Load 2500 into the TMR register (2.5mS).
21. Set the TIMER bit in the CR register to start the timer.
22. Wait until the IDLE bit in the MR register is set.
23. Clear the Timer bit in the CR register to stop the timer.
24. Clear the PMODE bit in the CR register to take out of Program Mode.
25. Lock the Flash controller by writing 0x00 to the MR Register.
26. Write the configuration sequence to the MR register to enable the Control Register upper bits (24-31).
27. Restore the upper bits of the Control Register.
28. Lock the Flash controller by writing 0x00 to the MR register.
29. Reset the system clock to normal operations.
30. Enable the Watch Dog timer (if desired).

To erase Flash memory:

1. Disable the Watch Dog Timer (if enabled).
2. Set the clock to internal oscillator (20 MHz).
3. Write the configuration sequence to the MR register to enable Control Register upper bits (24-31).
4. Configure upper bits of the Control Register (HRESPD and Flash access of 4 cycles).
5. Lock the Flash controller by writing 0x00 to the MR register.
6. Write the Flash sequence to the MR register.
7. Clear the DRTY register.
8. Set the PMODE bit in the CR register to set Program Mode.
9. Wait until the IDLE bit in the MR register is set.
10. If erasing all the Flash, set the AE bit in the CR register; otherwise, load the page address in the AR register.
11. Set the ERS bit in the CR register.
12. Set the WE bit in the CR register to start the erase operation.
13. Wait until the IDLE bit in the MR register is set.
14. Clear the WE, ERS and AE bits.
15. Load 2500 into the TMR register (2.5mS).
16. Set the TIMER bit in the CR register to start the timer.
17. Wait until the IDLE bit in the MR register is set.
18. Clear the Timer bit in the CR register to stop the timer.

19. Clear the PMODE bit in the CR register to take out of Program Mode.
20. Lock the Flash controller by writing 0x00 to the MR Register.
21. Write the configuration sequence to the MR register to enable the Control Register upper bits (24-31).
22. Restore the upper bits of the Control Register.
23. Lock the Flash controller by writing 0x00 to the MR register.
24. Reset the system clock to normal operations.
25. Enable the Watch Dog timer (if desired).

The CRC16 function allows a CRC check on the Flash bytes to a known value.

To run a CRC16 check on Flash bytes (must be done in memory, since every read, including instructions, would be part of the CRC16 calculations).

1. Disable the Watch Dog Timer (if enabled).
2. Set the CRCINIT bit in the CR register.
3. Clear the CRCINIT bit in the CR register.
4. Read the Flash memory that is to be processed.
5. When completed, the CRC value is located in the CRC register.

8. Internal SRAM

Overview

The Z32F1281 MCU implements zero-wait on the chip's SRAM. The size of SRAM is 12 KB. The SRAM base address is 0x2000_0000.

The SRAM memory area is usually used for data memory and stack memory. Sometimes, the code is dumped into the SRAM memory for fast operation or Flash erase/PGM operation.

This device does not support a memory remap strategy; therefore, a jump and return is required to execute the code in the SRAM memory area.

Figure 8.1 shows a block diagram of the SRAM.

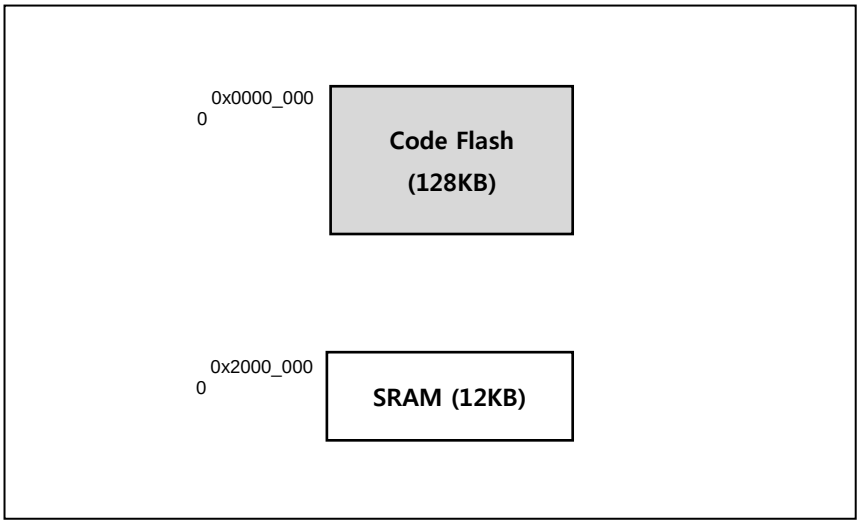


Figure 8.1. SRAM Block Diagram

9. Direct Memory Access Controller

Introduction

The Direct Memory Access (DMA) controller includes the following features:

- 15 channels
- Single transfer only
- Supports 8/16/32-bit data size
- Supports multiple buffers with the same size
- Interrupt condition is transferred through a peripheral interrupt

A block diagram of the DMA controller is shown in Figure 9.1.

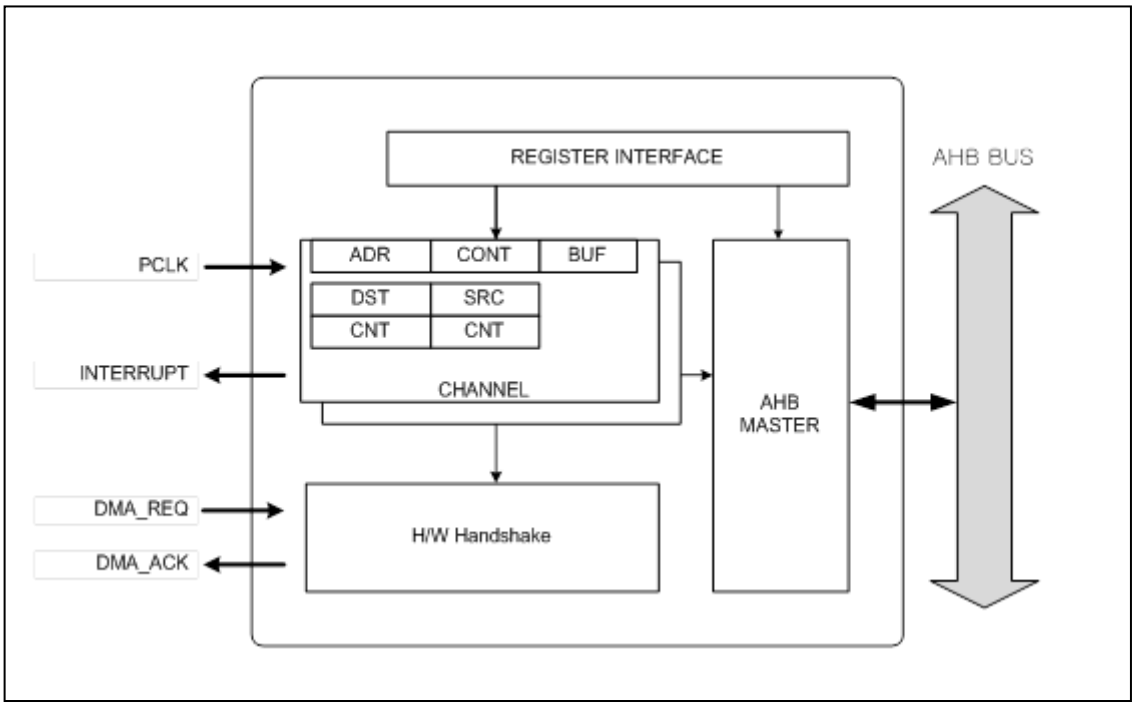


Figure 9.1. DMAC Block Diagram

Pin Description

There are no external interface pins.

Registers

The base address of the DMA controller is shown in Table 9.1.

Table 9.1. DMA Controller Base Address

Ch. No.	BASE ADDRESS	Assigned Peripheral
DMACH0	0x4000_0400	UART0 RX
DMACH1	0x4000_0410	UART0 TX
DMACH2	0x4000_0420	UART1 RX
DMACH3	0x4000_0430	UART1 TX
DMACH4	0x4000_0440	UART2 RX
DMACH5	0x4000_0450	UART2 TX
DMACH6	0x4000_0460	UART3 RX
DMACH7	0x4000_0470	UART3 TX
DMACH8	0x4000_0480	SPI0 RX
DMACH9	0x4000_0490	SPI0 TX
DMACH10	0x4000_04A0	SPI1 RX
DMACH11	0x4000_04B0	SPI1 TX
DMACH12	0x4000_04C0	ADC0
DMACH13	0x4000_04D0	ADC1
DMACH14	0x4000_04E0	ADC2

Table 9.2 shows the register map of the DMA controller.

Table 9.2. DMAC Register Map

Name	Offset	R/W	Description	Reset
DC0CR	0x0000	R/W	DMA Channel 0 Control Register	0x0000_0000
DC0SR	0x0004	R/W	DMA Channel 0 Status Register	0x0000_0000
DC0PAR	0x0008	R	DMA Channel 0 Peripheral Address	UART0_RBR
DC0MAR	0x000C	R/W	DMA Channel 0 Memory Address	0x2000_0000
DC1CR	0x0010	R/W	DMA Channel 1 Control Register	0x0000_0000
DC1SR	0x0014	R/W	DMA Channel 1 Status Register	0x0000_0000
DC1PAR	0x0018	R	DMA Channel 1 Peripheral Address	UART0_THR
DC1MAR	0x001C	R/W	DMA Channel 1 Memory Address	0x2000_0000
DC2CR	0x0020	R/W	DMA Channel 2 Control Register	0x0000_0000
DC2SR	0x0024	R/W	DMA Channel 2 Status Register	0x0000_0000
DC2PAR	0x0028	R	DMA Channel 2 Peripheral Address	UART1_RBR
DC2MAR	0x002C	R/W	DMA Channel 2 Memory Address	0x2000_0000
DC3CR	0x0030	R/W	DMA Channel 3 Control Register	0x0000_0000
DC3SR	0x0034	R/W	DMA Channel 3 Status Register	0x0000_0000
DC3PAR	0x0038	R	DMA Channel 3 Peripheral Address	UART1_THR
DC3MAR	0x003C	R/W	DMA Channel 3 Memory Address	0x2000_0000
DC4CR	0x0040	R/W	DMA Channel 4 Control Register	0x0000_0000
DC4SR	0x0044	R/W	DMA Channel 4 Status Register	0x0000_0000

DC4PAR	0x0048	R	DMA Channel 4 Peripheral Address	UART2_RBR
DC4MAR	0x004C	R/W	DMA Channel 4 Memory Address	0x2000_0000
DC5CR	0x0050	R/W	DMA Channel 5 Control Register	0x0000_0000
DC5SR	0x0054	R/W	DMA Channel 5 Status Register	0x0000_0000
DC5PAR	0x0058	R	DMA Channel 5 Peripheral Address	UART2_THR
DC5MAR	0x005C	R/W	DMA Channel 5 Memory Address	0x2000_0000
DC6CR	0x0060	R/W	DMA Channel 6 Control Register	0x0000_0000
DC6SR	0x0064	R/W	DMA Channel 6 Status Register	0x0000_0000
DC6PAR	0x0068	R	DMA Channel 6 Peripheral Address	UART3_RBR
DC6MAR	0x006C	R/W	DMA Channel 6 Memory Address	0x2000_0000
DC7CR	0x0070	R/W	DMA Channel 7 Control Register	0x0000_0000
DC7SR	0x0074	R/W	DMA Channel 7 Status Register	0x0000_0000
DC7PAR	0x0078	R	DMA Channel 7 Peripheral Address	UART3_THR
DC7MAR	0x007C	R/W	DMA Channel 7 Memory Address	0x2000_0000
DC8CR	0x0080	R/W	DMA Channel 8 Control Register	0x0000_0000
DC8SR	0x0084	R/W	DMA Channel 8 Status Register	0x0000_0000
DC8PAR	0x0088	R	DMA Channel 8 Peripheral Address	SPI0_RDR
DC8MAR	0x008C	R/W	DMA Channel 8 Memory Address	0x2000_0000
DC9CR	0x0090	R/W	DMA Channel 9 Control Register	0x0000_0000
DC9SR	0x0094	R/W	DMA Channel 9 Status Register	0x0000_0000
DC9PAR	0x0098	R	DMA Channel 9 Peripheral Address	SPI0_TDR
DC9MAR	0x009C	R/W	DMA Channel 9 Memory Address	0x2000_0000
DC10CR	0x00A0	R/W	DMA Channel 10 Control Register	0x0000_0000
DC10SR	0x00A4	R/W	DMA Channel 10 Status Register	0x0000_0000
DC10PAR	0x00A8	R	DMA Channel 10 Peripheral Address	SPI1_RDR
DC10MAR	0x00AC	R/W	DMA Channel 10 Memory Address	0x2000_0000
DC11CR	0x00B0	R/W	DMA Channel 11 Control Register	0x0000_0000
DC11SR	0x00B4	R/W	DMA Channel 11 Status Register	0x0000_0000
DC11PAR	0x00B8	R	DMA Channel 11 Peripheral Address	SPI1_TDR
DC11MAR	0x00BC	R/W	DMA Channel 11 Memory Address	0x2000_0000
DC12CR	0x00C0	R/W	DMA Channel 12 Control Register	0x0000_0000
DC12SR	0x00C4	R/W	DMA Channel 12 Status Register	0x0000_0000
DC12PAR	0x00C8	R	DMA Channel 12 Peripheral Address	AD0DDR
DC12MAR	0x00CC	R/W	DMA Channel 12 Memory Address	0x2000_0000
DC13CR	0x00D0	R/W	DMA Channel 13 Control Register	0x0000_0000
DC13SR	0x00D4	R/W	DMA Channel 13 Status Register	0x0000_0000
DC13PAR	0x00D8	R	DMA Channel 13 Peripheral Address	AD1DDR
DC13MAR	0x00DC	R/W	DMA Channel 13 Memory Address	0x2000_0000
DC14CR	0x00E0	R/W	DMA Channel 14 Control Register	0x0000_0000
DC14SR	0x00E4	R/W	DMA Channel 14 Status Register	0x0000_0000
DC14PAR	0x00E8	R	DMA Channel 14 Peripheral Address	AD2DDR
DC14MAR	0x00EC	R/W	DMA Channel 14 Memory Address	0x2000_0000

DC*n*CR DMA Controller Configuration Register

The DMA operation control register is a 32-bit register.

DC0CR=0x4000_0400 , DC1CR=0x4000_0410
DC2CR=0x4000_0420 , DC3CR=0x4000_0430
DC4CR=0x4000_0440 , DC5CR=0x4000_0450
DC6CR=0x4000_0460 , DC7CR=0x4000_0470
DC8CR=0x4000_0480 , DC9CR=0x4000_0490
DC10CR=0x4000_04A0 , DC11CR=0x4000_04B0
DC12CR=0x4000_04C0 , Dc13CR=0x4000_04D0
DC14CR=0x4000_04E0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRANSCNT																												SIZE			
0	0	0	0	0x000												0	0	0	0	0	0	0	0	0	0	0	0	00		0	0
RW																												RW			

31 16	TRANSCNT	Number of DMA transfer remaining Required transfer number should be written before enable DMA transfer.
		0 DMA transfer is done
		N N transfers are remaining
3 2	SIZE	Bus transfer size
		00 DMA transfer is byte size transfer
		01 DMA transfer is half word size transfer
		10 DMA transfer is word size transfer
		11 Reserved

DC*n*SR DMA Controller Status Register

The DMA Controller Status Register is an 8-bit register. This register represents the current status of the DMA controller and enables the DMA function.

DC0SR=0x4000_0404 , DC1SR=0x4000_0414
DC2SR=0x4000_0424 , DC3SR=0x4000_0434
DC4SR=0x4000_0444 , DC5SR=0x4000_0454
DC6SR=0x4000_0464 , DC7SR=0x4000_0474
DC8SR=0x4000_0484 , DC9SR=0x4000_0494
DC10SR=0x4000_04A4 , DC11SR=0x4000_04B4
DC12SR=0x4000_04C4 , Dc13SR=0x4000_04D4
DC14SR=0x4000_04E4

7	6	5	4	3	2	1	0
EOT							DMAEN
1	0	0	0	0	0	0	0
RO							RW

7	EOT	End of transfer	
		0	Data to be transferred exists TRANSCNT shows non zero value
		1	All data is transferred TRANSCNT shows now 0
0	DMAEN	DMA Enable	
		0	DMA is in stop or hold state
		1	DMA is running or enabled

DC*n*PAR DMA Controller Peripheral Address Register

The DMA Controller Peripheral Address register represent the peripheral address.

DC0PAR=0x4000_0408 , DC1PAR=0x4000_0418
DC2PAR=0x4000_0428 , DC3PAR=0x4000_0438
DC4PAR=0x4000_0448 , DC5PAR=0x4000_0458
DC6PAR=0x4000_0468 , DC7PAR=0x4000_0478
DC8PAR=0x4000_0488 , DC9PAR=0x4000_0498
DC10PAR=0x4000_04A8,, DC11PAR=0x4000_04B8
DC12PAR=0x4000_04C8,, Dc13PAR=0x4000_04D8
DC14PAR=0x4000_04E8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PAR																															
DC0PAR=U0RBR, DC1PAR=U0THR DC2PAR=U10RBR, DC3PAR=U10THR DC4PAR=U20RBR, DC5PAR=U20THR DC6PAR=U30RBR, DC7PAR=U30THR DC8PAR=SPI0_RDR, DC9PAR=SPI0_TDR DC10PAR=SPI1_RDR, DC11PAR=SPI1_TDR DC12PAR=AD0DDR, DC13PAR=AD1DDR, DC14PAR=AD2DDR																															
RO																															

31 0	PAR	Target Peripheral address of transmit buffer or receive buffer. Address is fixed address when each transfer is done.

DC*n*MAR DMA Controller Memory Address Register

The DMA Controller Memory Address register represents the memory address.

DC0MAR=0x4000_040C , DC1MAR=0x4000_041C
DC2MAR=0x4000_042C , DC3MAR=0x4000_043C
DC4MAR=0x4000_044C , DC5MAR=0x4000_045C
DC6MAR=0x4000_046C , DC7MAR=0x4000_047C
DC8MAR=0x4000_048C , DC9MAR=0x4000_049C
DC10MAR=0x4000_04AC,, DC11MAR=0x4000_04B8C
DC12MAR=0x4000_04CC,, Dc13MAR=0x4000_04DC
DC14MAR=0x4000_04EC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																MAR															
0x2000																0x0000															
RO																RW															

31 0	MAR	Target memory address of data transfer. Address is automatically incremented according to SIZE bits when each transfer is done.

Functional Description

The DMA controller performs direct memory transfer by sharing the system bus with the CPU core. The system bus is shared by two AHB masters following the round-robin priority strategy. Therefore, the DMA controller can share half of the system bandwidth.

The DMA controller can be triggered only by a peripheral request. When a peripheral requests the transfer to the DMA controller, the related channel is activated and accesses the bus to transfer the requested data from memory to the peripheral data buffer or from the peripheral data buffer to memory space.

The transfer process involves the following steps:

1. User sets the peripheral and memory addresses.
2. User configures the DMA operation mode and transfer count.
3. User enables the DMA channel.
4. The peripheral sends a DMA request.
5. DMA activates the channel that was requested.
6. DMA reads data from the source address and saves it to the internal buffer.
7. DMA writes the buffered data to the destination address.
8. The transfer count number is decreased by 1.
9. When the transfer count becomes 0, the EOT flag is set and a notice is sent to the peripheral to issue the interrupt.
10. DMA does not have an interrupt source; the interrupt-related DMA status can be shown from the assigned peripheral interrupt.

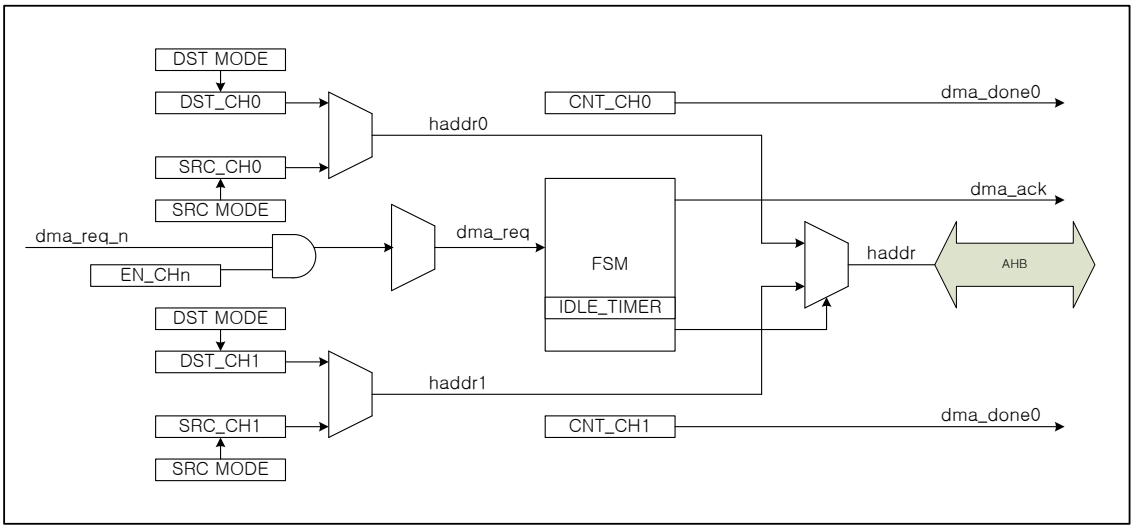


Figure 9.2. DMA Controller Block Diagram

Figure 9.3 shows the functional timing diagram of the DMA controller. The transfer request from the peripheral is pended internally and it invokes source data read transfer on the AHB bus. The read data from the source address is stored in the internal buffer. This data is transferred to the destination address when the AHB bus is available.

The timing diagram for a DMA transfer from peripheral to memory is shown in Figure 9.3. A 4-clock cycle latency exists when accessing the peripheral. If the bus is occupied by a different bus master, the number of bus waiting cycles increase until the bus is available.

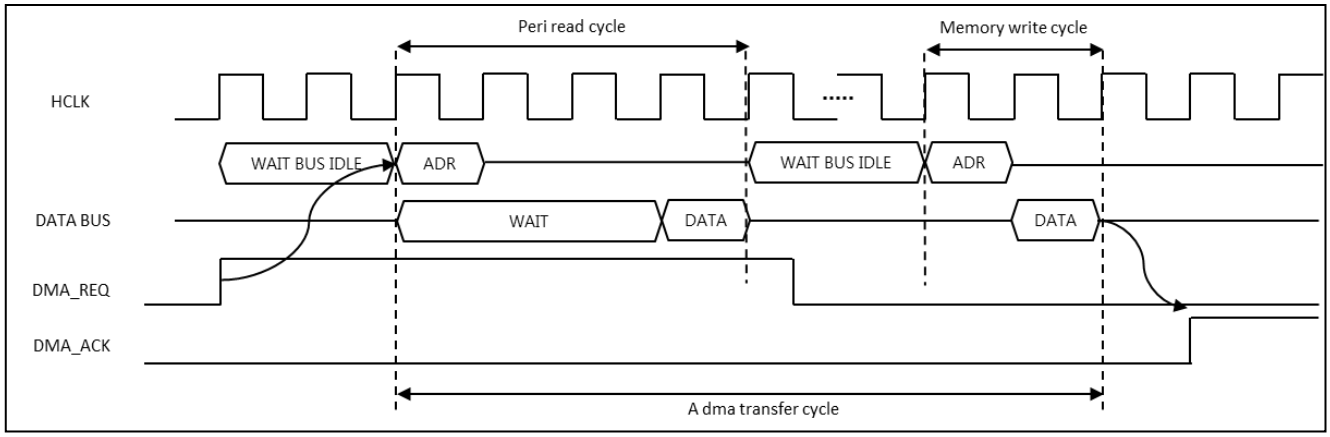


Figure 9.3. DMA Transfer from Peripheral to Memory

The timing diagram for a DMA transfer from memory to peripheral is shown in Figure 9.4. A 4-clock cycle latency exists while accessing the peripheral. If the bus is occupied by a different bus master, the number of bus waiting cycles increase until the bus is available.

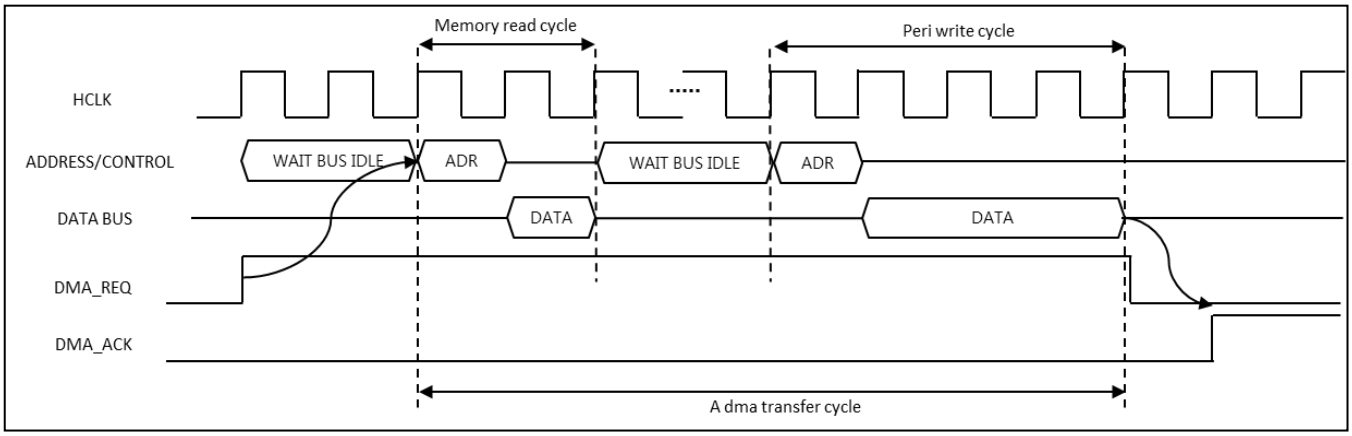


Figure 9.4. DMA Transfer from Memory to Peripheral

Figure 9.5 is an example of N data transfers with the DMA. The DMA transfer is started when DCnSR.DMAEN is set and will be cleared when all transfers are completed.

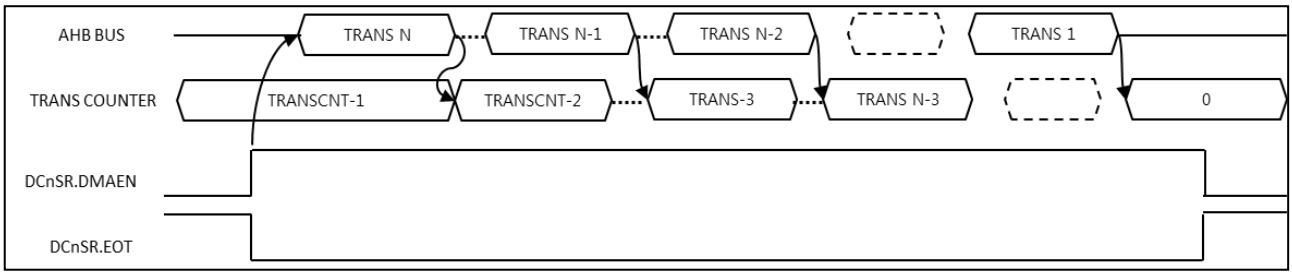


Figure 9.5. N DMA Transfer Example

10. Watch-Dog Timer

Overview

The Watchdog Timer can monitor the system and generate an interrupt or a reset. It has a 32-bit down-counter. The Miscellaneous Clock Control Register 3 provides base clock options with clock dividers to drive the WDT clock. This can be selected in the WDTCON register. To prevent the WDT from firing, reload the LR register with the appropriate value before the WDT times out.

Features include:

- 32-bit down counter (WDTCVR)
- Select reset or periodic interrupt
- Count clock selection
- Dedicated prescaler
- Watchdog overflow output signal

Figure 10.1 shows a block diagram of the Watch-dog Timer.

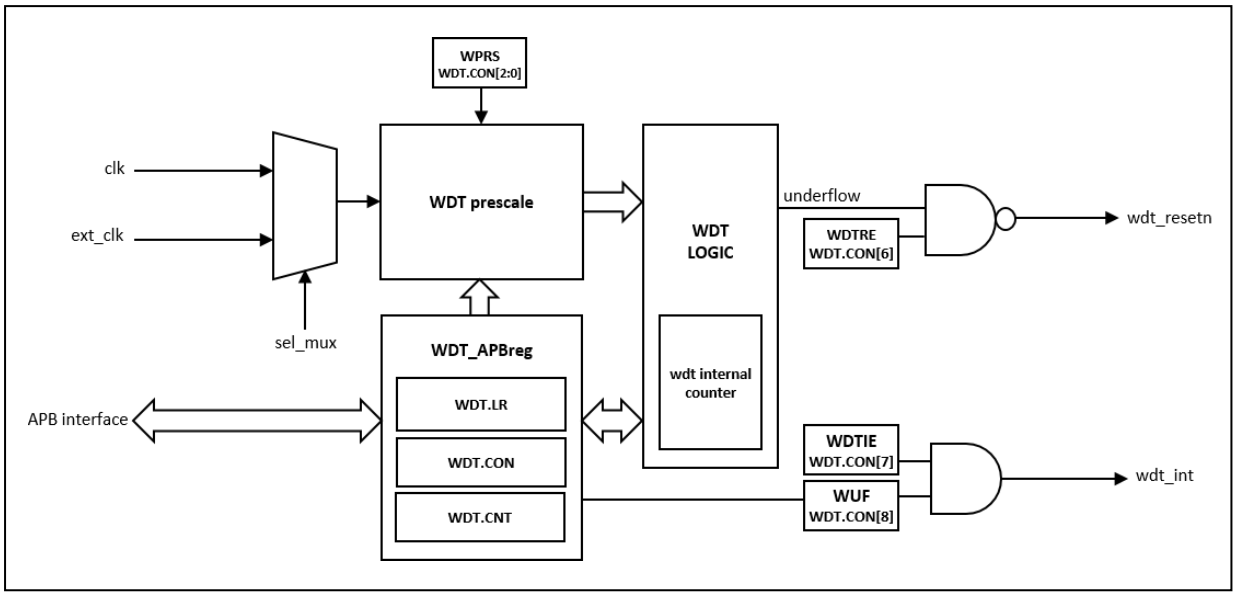


Figure 10.1. WDT Block Diagram

Registers

The base address of the watch-dog timer is 0x4000_0200 and the register map is described in Table 10.1. The initial watch-dog time-out period is set to 2000-miliseconds.

Table 10.1. Watchdog Timer Register Map

Name	Offset	R/W	Description	Reset
WDTLR	0x0000	R/W	WDT Load register	0x00000000
WDTCNT	0x0004	R	WDT Current counter register	0x0000FFFF
WDTCON	0x0008	R/W	WDT Control register	0x0000805C

WDTLR Watchdog Timer Load Register

The WDTLR register is used to update the WDTCVR register. To update the WDTCVR register, the WDTEN bit of WDTCON should be set to **1** and written into the WDTLR register with target value of WDTCVR.

WDTLR=0x4000_0200

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDTLR																															
0x0000_0000																															
RW																															

31 0	WDTLR	Watchdog timer load value register Keeping WDTEN bit as '1', write WDTLR register will update WDTCVR value with written value

WDTCNT Watchdog Timer Current Counter Register

The WDTCNT register represents the current count value of the 32-bit down counter .When the counter value reaches 0, an interrupt or reset will take effect.

WDTLR=0x4000_0204

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDTCNT																															
0x0000_FFFF																															
R																															

31 0	WDTCNT	Watchdog timer current counter register 32-bit down counter will run from the written value.

WDTCON Watchdog Timer Control Register

The WDT module should be configured properly before running. When the target purpose is defined, the WDT can be configured in the WDTCON register.

WDTCON=0x4000_0208

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDBG							WUF	WDTIE	WDTRE		WDTEN	CKSEL	WPRS		
1	0	0	0	0	0	0	0	0	1	0	1	1	100		
RW							R	RW	RW		RW	RW	RW		

15	WDBG	Watchdog operation control in debug mode	
		0	Watchdog counter running when debug mode
		1	Watchdog counter stopped when debug mode
8	WUF	Watchdog timer underflow flag (This bit is cleared when WDTLR is written)	
		0	No underflow
		1	Underflow is pending
7	WDTIE	Watchdog timer counter underflow interrupt enable	
		0	Disable interrupt
		1	Enable interrupt
6	WDTRE	Watchdog timer counter underflow reset enable	
		0	Disable reset
		1	Enable reset
4	WDTEN	Watchdog Counter enable	
		0	Watch dog counter disabled
		1	Watch dog counter enabled
3	CKSEL	WDTCLKIN clock source select	
		0	PCLK
		1	External clock (MCCR3)
2 0	WPRS[2:0]	Counter clock prescaler WDTCLK = WDTCLKIN/WPRS	
		000	WDTCLKIN
		001	WDTCLKIN / 4
		010	WDTCLKIN / 8
		011	WDTCLKIN / 16
		100	WDTCLKIN / 32
		101	WDTCLKIN / 64
		110	WDTCLKIN / 128
		111	WDTCLKIN / 256

Functional Description

The watchdog timer count can be enabled by WDTEN (WDT.CON[4]) to 1. As the watchdog timer is enabled, the down counter will start counting from the Load Value. If WDTRE (WDT.CON[6]) is set to 1, WDT reset will be asserted when the WDT counter value reaches 0 (underflow event) from the WDTLR value. Before the WDT counter goes down to 0, the software can write a certain value to the WDTLR register to reload the WDT counter.

Timing Diagram

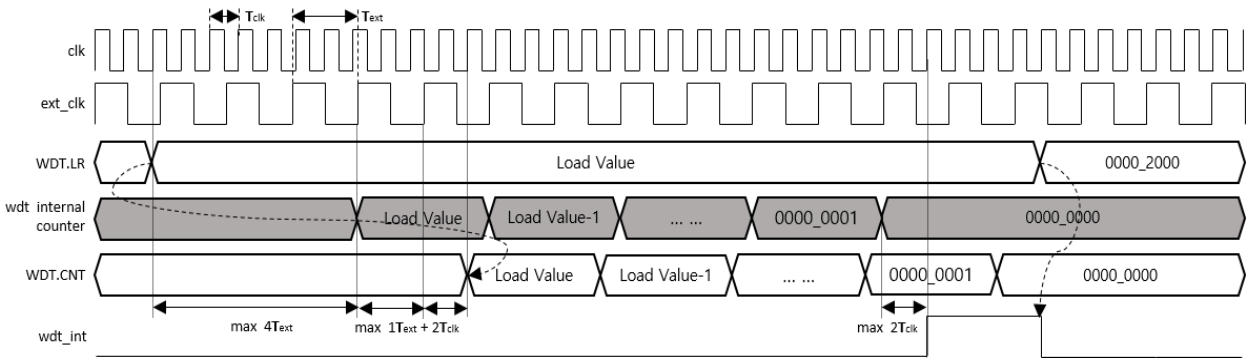


Figure 10.2. Timing Diagram in Interrupt Mode Operation when WDT clock is External Clock

In WDT interrupt mode, after WDT underflow occurs, a certain count value is reloaded to prevent the next WDT interrupt in a short time period. This reloading action is only activated when the watchdog timer counter is set to Interrupt mode (set WDTIE of WDT.CON). It takes up to 5 cycles to go from the Load value to the CNT value. The WDT interrupt signal and CNT value data may be delayed by a maximum of 2 system bus clocks in synchronous logic.

Prescale Table

The WDT includes a 32-bit down counter with programmable pre-scaler to define different time-out intervals.

The clock sources of the watchdog timer can be the peripheral clock (PCLK) or one of 5 external clock sources. The external clock source can be enabled by CKSEL (WDT.CON[3]) set to '1'. The external clock source was chosen in the MCCR3 register of the SCU block.

To make the WDT counter base clock, users can control the 3-bit pre-scaler WPRS [2:0] in the WDT.CON register and the maximum pre-scaled value is "clock source frequency/256". The pre-scaled WDT counter clock frequency values are listed in Table 10.2.

Selectable clock source (40 kHz ~ 16 MHz) and the time out interval when 1 count

Time out period = {(Load Value) * (1/pre-scaled WDT counter clock frequency) + max $5T_{ext}$} + max $4T_{clk}$

*Time out period (time out period from load value to interrupt set '1')

Table 10.2. Pre-scaled WDT Counter Clock Frequency

Clock source	WDTCLKIN	WDTCLKIN /4	WDTCLKIN /8	WDTCLKIN /16	WDTCLKIN /32	WDTCLKIN /64	WDTCLKIN /128	WDTCLKIN /256
Ring OSC	1Mhz	250khz	125khz	62.5khz	31.25khz	15.625khz	7.8125khz	3.90625khz
MCLK	MCLK (BUS CLK)	MCLK/4	MCLK/8	MCLK/16	MCLK/32	MCLK/64	MCLK/128	MCLK/256
IOSC	20Mhz	5Mhz	2.5Mhz	1.25Mhz	625khz	312.5khz	156.25khz	78.125khz
EOSC	XTAL	XTAL/4	XTAL/8	XTAL/16	XTAL/32	XTAL/64	XTAL/128	XTAL/256
PLL	PLL	PLL/4	PLL/8	PLL/16	PLL/32	PLL/64	PLL/128	PLL/256

11. 16-Bit Timer

Overview

The timer block consists of 6 channels of 16-bit general-purpose timers. They can support periodic timer, PWM pulse, one-shot timer, and capture mode.

Features include:

- 16-bit up-counter
- Periodic timer mode
- One-shot timer mode
- PWM pulse mode
- Capture mode
- 10-bit prescaler

Figure 11.1 shows a block diagram of the 16-bit timer.

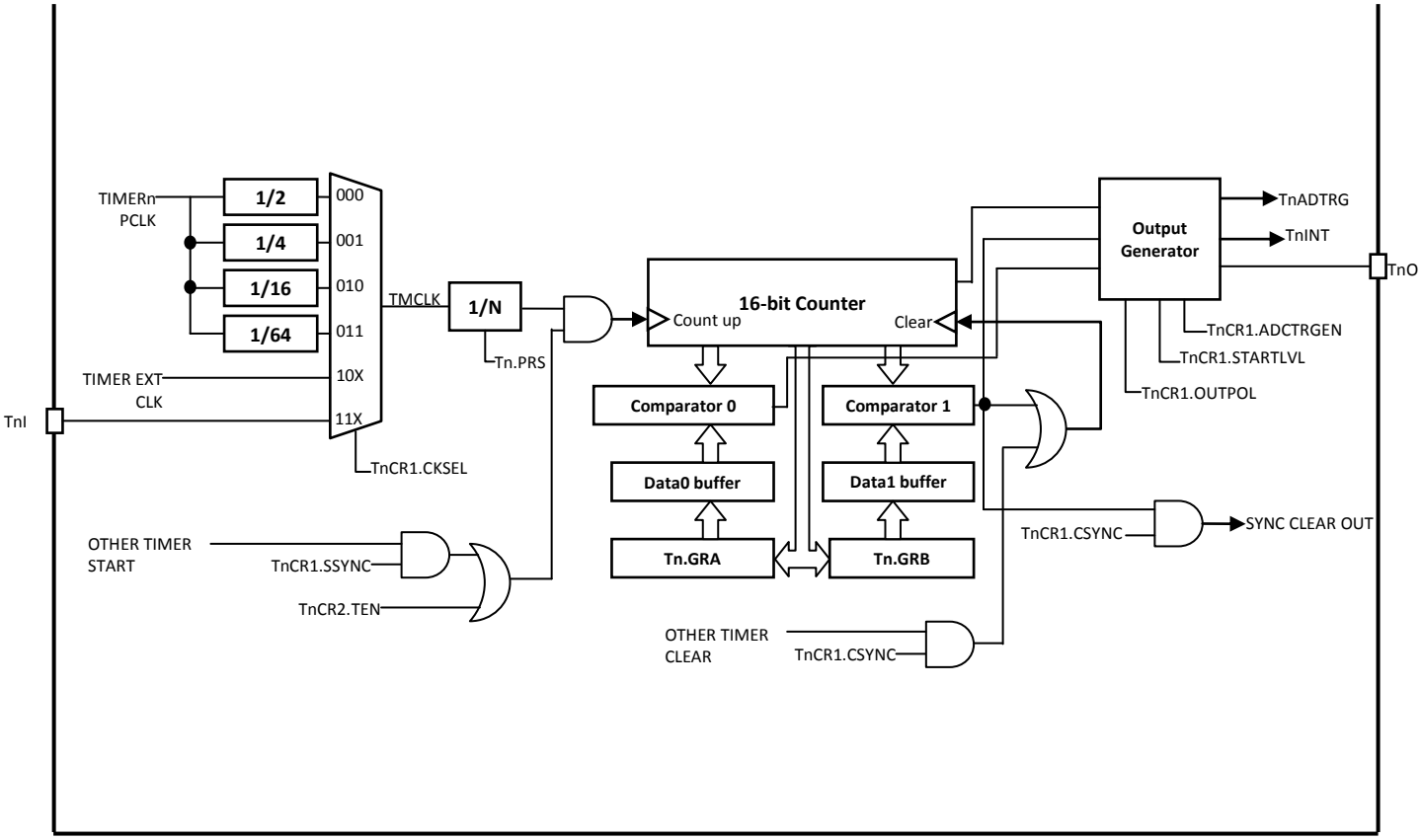


Figure 11.1. 16-bit Timer Block Diagram

Pin Description

Table 11.1. External Pin

PIN NAME	TYPE	DESCRIPTION
TnC	I	External clock / capture input
TnO	O	Timer output

Registers

The base address of the Timer is `0x4000_3000` and the register map is described in Table 11.2 and Table 11.3.

Table 11.2. Base Address of Each Channel

CHANNEL	Address
T0	0x4000_3000
T1	0x4000_3020
T2	0x4000_3040
T3	0x4000_3060
T8	0x4000_3100
T9	0x4000_3120

Table 11.3. Timer Register Map

Name	Offset	R/W	Description	Reset
T*n*CR1	0x--00	R/W	Timer control register 1	0x00000000
T*n*CR2	0x--04	R/W	Timer control register 2	0x00000000
T*n*PRS	0x--08	R/W	Timer prescaler register	0x00000000
T*n*GRA	0x--0C	R/W	Timer general data register A	0x00000000
T*n*GRB	0x--10	R/W	Timer general data register B	0x00000000
T*n*CNT	0x--14	R/W	Timer counter register	0x00000000
T*n*SR	0x--18	R/W	Timer status register	0x00000000
T*n*IER	0x--1C	R/W	Timer interrupt enable register	0x00000000
TGECR	0x0140	R/W	Timer Group Encoder Control Register	0x00000000

TnCR1 Timer n Control Register 1

Timer Control Register 1 is a 16-bit register. The Timer module should be configured properly before running. When the target purpose is defined, the timer can be configured in the TnCR1 register.

T0CR1=0x4000_3000, T1CR1=0x4000_3020
T2CR1=0x4000_3040, T3CR1=0x4000_3060
T8CR1=0x4000_3100, T9CR1=0x4000_3120

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							ADCTRGEN	STARTLV	CKSEL			CLRMOD		MODE	
0	0	0	0	0	0	0	0	0	000			00		00	
							RW	RW	RW			RW		RW	

8	ADCTRGEN	ADC Trigger source enable	
		0	Timer does not trigger ADC
		1	Timer triggers ADC
7	STARTLVL	Interval/PWM/One-shot mode initial output value	
		0	Output starts with ‘L’
		1	Output starts with ‘H’
6 4	CKSEL[2:0]	Counter clock source select	
		000	PCLK/2
		001	PCLK/4
		010	PCLK/16
		011	PCLK/64
		10X	TEXT0 (in MCCR3)
		11X	TnC pin input
3 2	CLRMOD	Clear select when capture mode	
		00	Rising edge clear mode
		01	Falling edge clear mode
		10	Both edge clear mode
		11	None clear mode
1 0	MODE[1:0]	Timer operation mode control	
		00	Normal periodic operation mode
		01	PWM mode
		10	One shot mode
		11	Capture mode

TnCR2 Timer n Control Register 2

Timer Control Register 2 is an 8-bit register.

T0CR2=0x4000_3004, T1CR2=0x4000_3024
T2CR2=0x4000_3044, T3CR2=0x4000_3064
T8CR2=0x4000_3104, T9CR2=0x4000_3124

7	6	5	4	3	2	1	0
						TCLR	TEN
0	0	0	0	0	0	0	0
R	R	R	R	R	R	WO	RW

1	TCLR	Timer Count register clear
	0	No
	1	Initialize timer. If set to '1', count register will be cleared. This is write-only.
0	TEN	Timer enable bit
	0	Disable timer
	1	Enable timer

TnPRS Timer n Prescaler Register

Timer Prescaler Register sets the pre-scale of the input clock for the timer counter.

T0PRS=0x4000_3008, T1PRS=0x4000_3028
T2PRS =0x4000_3048, T3PRS=0x4000_3068
T8PRS=0x4000_3108, T9PRS=0x4000_3128

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						PRS									
0	0	0	0	0	0	000									
						RW									

9	PRS	Pre-scale value of count clock
0		TCLK = CLOCK_IN/(PRS+1)
		(*CLOCK_IN* is a selected timer input clock in TnCR1[CKSEL])

TnGRA Timer n General Register A

Timer General Register A is a 16-bit register.

T0GRA=0x4000_300C, T1GRA=0x4000_302C
T2GRA =0x4000_304C, T3GRA=0x4000_306C
T8GRA=0x4000_310C, T9GRA=0x4000_312C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GRA															
0x0000															
RW															

15 0	GRA	Timer n General Register A Periodic mode - Period value of time internal. - When the counter value is matched with this value, GRA Match interrupt is requested PWM mode - Duty value of PWM Output - When the counter value is matched with this value, GRA Match interrupt is requested One-shot mode - One-shot delay timing before output pulse. - When the counter value is matched with this value, GRA Match interrupt is requested Capture mode - Falling edge of TnC port will capture the count value when rising edge clear mode - Rising edge of TnC port will capture the count value when falling edge clear mode

TnGRB Timer n General Register B

Timer General Register B is a 16-bit register. This register is used for the Timer in PWM modes.

T0GRB=0x4000_3010, T1GRB=0x4000_3030
T2GRB=0x4000_3050, T3GRB=0x4000_3070
T8GRB=0x4000_3110, T9GRB=0x4000_3130

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GRB															
0x0000															
RW															

15 0	GRB		Timer n General Register B
			Periodic mode - Not used. No interrupt generated.
			PWM mode - Time interval value of PWM carrier frequency. - When the counter value is matched with this value, GRB Match interrupt is requested only in PWM and one-shot modes.
			One-shot mode - One-shot pulse output stop timing value. - When the counter value is matched with this value, GRB Match interrupt is requested only in PWM and one-shot modes.
			Capture mode - Rising edge of TnC port will capture the count value when rising edge clear mode - Falling edge of TnC port will capture the count value when falling edge clear mode

TnCNT Timer n Count Register

Timer Count Register is a 16-bit register.

T0CNT=0x4000_3014, T1CNT=0x4000_3034
T2CNT=0x4000_3054, T3CNT=0x4000_3074
T8CNT=0x4000_3114, T9CNT=0x4000_3134

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT															
0x0000															
RW															

15 0	CNT		Timer count value register
		R	Read current timer count value
		W	Set count value

TnSR Timer n Status Register

Timer Status Register is a 16-bit register. This register indicates the current status of the timer module.

T0SR=0x4000_3018, T1SR=0x4000_3038
T2SR=0x4000_3058, T3SR=0x4000_3078
T8SR=0x4000_3118, T9SR=0x4000_3138

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					QDIR	QDIRCH	QRF						MFA	MFB	OVF
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
					RW	RW	RW						RW	RW	RW

10	QDIR	Current Direction
		0 Phase A leading Phase B (clockwise)
		1 Phase B leading Phase A (counterclockwise)
9	QDIRCH	Quadrature direction change
		0 No direction change
		1 Direction is changed. Write '1' to this bit for clear
8	QRF	Quadrature revolution flag
		0 No revolution flag
		1 Revolution flag is detected. Write '1' to this bit for clear
2	MFA	GRA Match flag
		0 Not match with GRA
		1 Match flag with GRA. Write '1' to this bit for clear
1	MFB	GRB Match flag
		0 Not match with GRB
		1 Match flag with GRB. Write '1' to this bit for clear
0	OVF	Counter overflow flag
		0 No overflow event
		1 Counter overflowed. Write '1' to this bit for clear

TnIER Timer n Interrupt Enable Register

The Timer Interrupt Enable Register is a 16-bit register. Each status flag of the timer block can issue the interrupt. To enable the interrupt, write **1** in the corresponding bit in the TnIER register.

T0IER=0x4000_301C, T1IER=0x4000_303C
T2IER=0x4000_305C, T3IER=0x4000_307C
T8IER=0x4000_311C, T9IER=0x4000_313C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						QDIRCHIE	QRIE						MAIE	MBIE	OVIE
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
						RW	RW						RW	RW	RW

9	QDIRCHIE	Quadrature direction change interrupt enable	
		0	Disable direction change interrupt
		1	Enable direction change interrupt
8	QRIE	Quadrature revolution interrupt enable	
		0	Disable revolution flag interrupt
		1	Enable revolution flag interrupt
2	MAIE	GRA Match interrupt enable	
		0	Disable match register A interrupt
		1	Enable match register A interrupt
1	MBIE	GRB Match interrupt enable	
		0	Disable match register B interrupt
		1	Enable match register B interrupt
0	OVIE	Counter overflow interrupt enable	
		0	Disable counter overflow interrupt
		1	Enable counter overflow interrupt

Note: The QMOD in the TGECR register must be set before enabling the Quadrature interrupts.

TGECR Timer Group Encoder Control Register

The Timer Group Encoder Control Register is a 16-bit register. Timer0, Timer1, Timer2, and Timer3 can be used for quadrature encoder interface function.

TGECR=0x4000_3140

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				RDIRCON	PDIRCON	BDIRCON	ADIRCON	QDPHBEG		QDPHAEG		QDPHZEG	QDPHSWAP		QDMOD
0	0	0	0	0	0	0	0	00		00		0	0	0	0
				RW	RW	RW	RW	RW		RW		RW	RW		RW

11	RDIRCON	Revolution counter direction control	
		0	DIR status not affect to the counter
		1	DIR status will change count direction
10	PDIRCON	Position counter direction control	
		0	DIR status not affect to the counter
		1	DIR status will change count direction
9	BDIRCON	Phase B counter direction control	
		0	DIR status not affect to the counter
		1	DIR status will change count direction
8	ADIRCON	Phase A counter direction control	
		0	DIR status not affect to the counter
		1	DIR status will change count direction
7 6	QDPHBEG[1:0]	Quadrature mode phase B count for position count	
		00	Rising edge count
		01	Falling edge count
		1X	Both edge count
5 4	QDPHAEG[1:0]	Quadrature mode phase A count for position count	
		00	Rising edge count
		01	Falling edge count
		1X	Both edge count
3	QDPHZEG	Quadrature mode phase Z count for revolution	
		0	PHZ rising edge count
		1	PHZ falling edge count
2	QDPHSWAP	Quadrature input swap	
		0	No swap
		1	Swap PHA and PHB
0	QDMOD	Quadrature decoder mode	
		0	Normal timer mode
		1	Quadrature decoder count mode Timer0 is phase A counter Timer1 is phase B counter Timer2 is position counter Timer3 is revolution counter

Functional Description

Basic Operation of Timer

In Figure 11.2, TMCLK is a reference clock for operation of the timer. Divide this clock by the prescaler setting to operate the counting clock. The following images show the starting point of the counter and the ending of the period point of the counter in normal periodic mode.

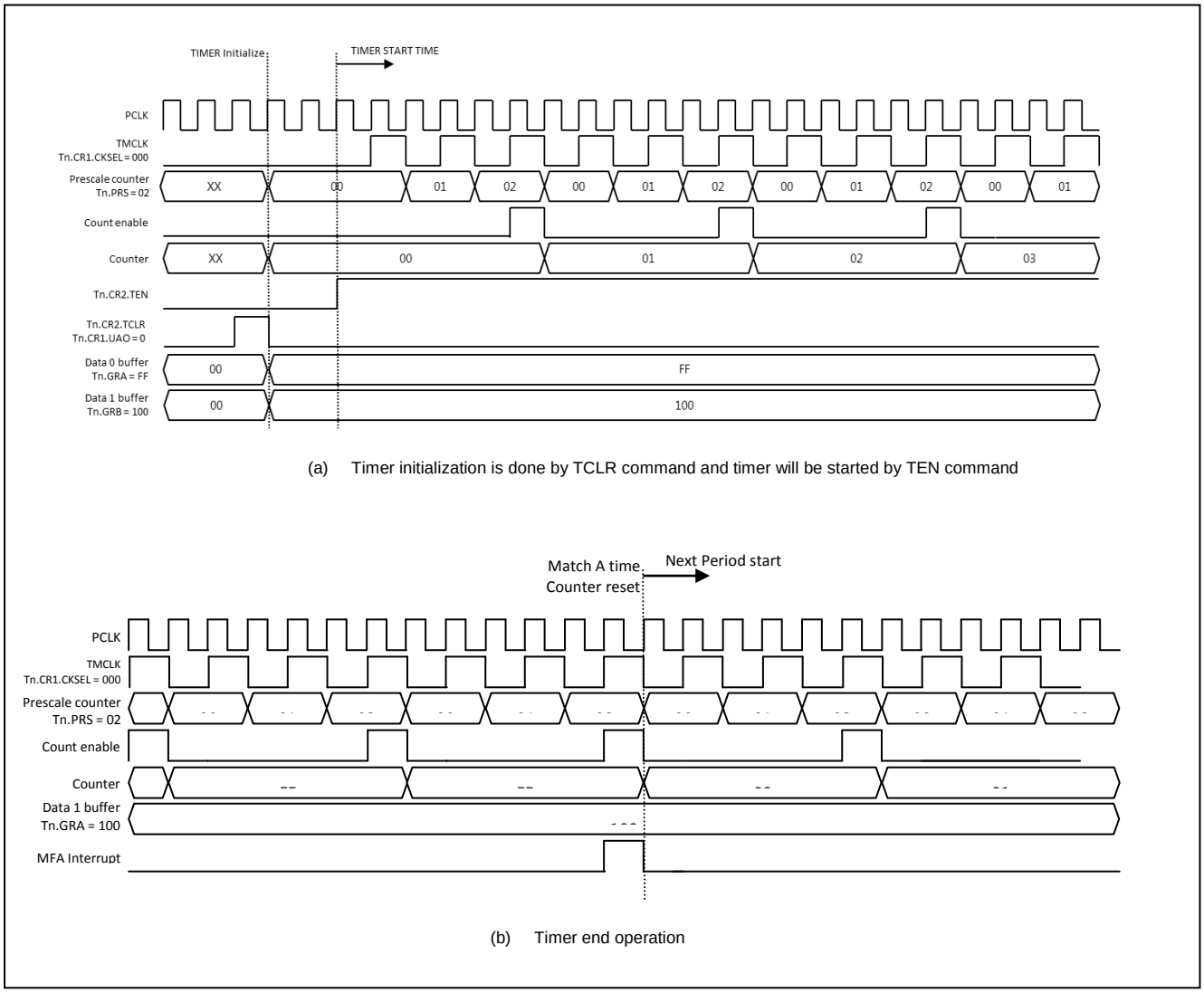


Figure 11.2. Basic Start and Match Operation

The timer count period can be calculated as follows:

The period = TMCLK Period * Tn.GRA value

Match A interrupt time = TMCLK Period * Tn.GRA value

When you change the timer setting and restart the timer with the new setting, Zilog recommends that you write the CR2.TCLR command before the CR2.TEN command.

Normal Periodic Mode

Figure 11.3 shows the timing diagram in normal periodic mode. The Tn.GRA value decides the timer period. The Tn.GRM register value does not matter.

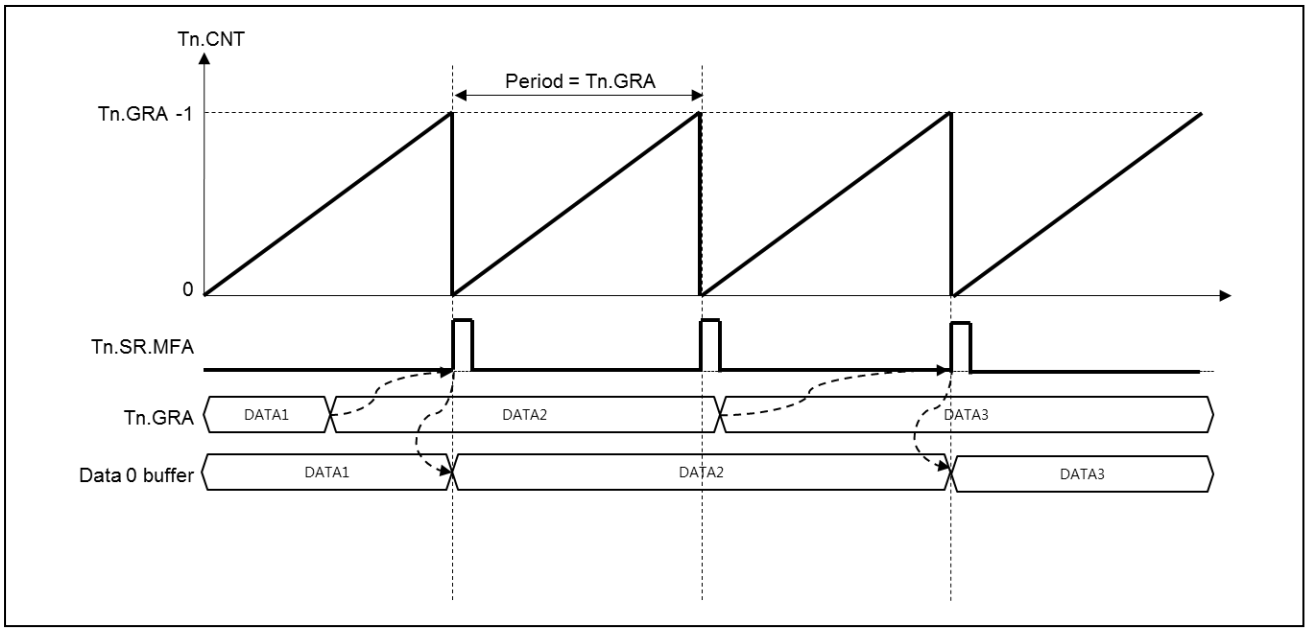


Figure 11.3. Normal Periodic Mode Operation

The timer count period can be calculated as follows:

The period = TMCLK Period * Tn.GRA value

Match A interrupt time = TMCLK Period * Tn.GRA value

If Tn.GRA = 0, the timer cannot be started even if TnCR2.TEN is “1” because the period is “0”.

The values in Tn.GRA and Tn.GRB are loaded into the internal compare data buffer 0 when the loading condition occurs. In this periodic mode, the Tn.CR2.TCLR write operation and the GRA match event will load the compare data buffers.

One Shot Mode

Figure 11.4 shows the timing diagram in one shot mode. The Tn.GRB value decides the one shot period. An additional comparison point is provided with the Tn.GRA register value.

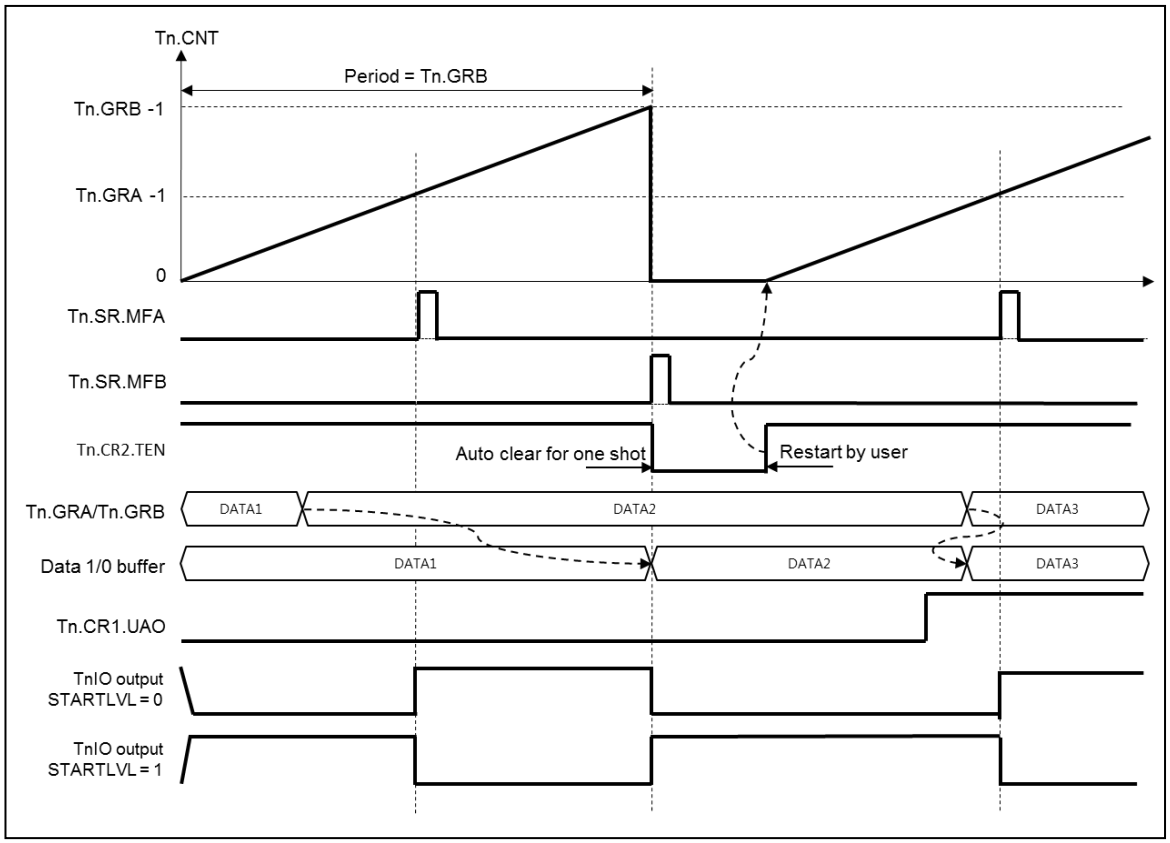


Figure 11.4. One Shot Mode Operation

The one shot count period can be calculated as follows:

The period = TMCLK Period * Tn.GRB value

Match A interrupt time = TMCLK Period * Tn.GRA value

If Tn.GRB = 0, the timer cannot be started even if TnCR2.TEN is “1” because the period is “0”.

The values in Tn.GRA and Tn.GRB are loaded into the internal compare data buffers 0 and 1 when the loading condition occurs. In this mode, the Tn.CR2.TCLR write operation and the GRB match event will load the data buffer.

The TnIO output signal format is the same as PWM mode. The Tn.GRB value defines the output pulse period and the Tn.GRA value defines the pulse width of one shot pulse.

PWM Timer Output Examples

Figure 11.5 shows the timing diagram of PWM output mode. The Tn.GRB value decides the PWM pulse period. An additional comparison point is provided with the Tn.GRA register value which defines the pulse width of PWM output.

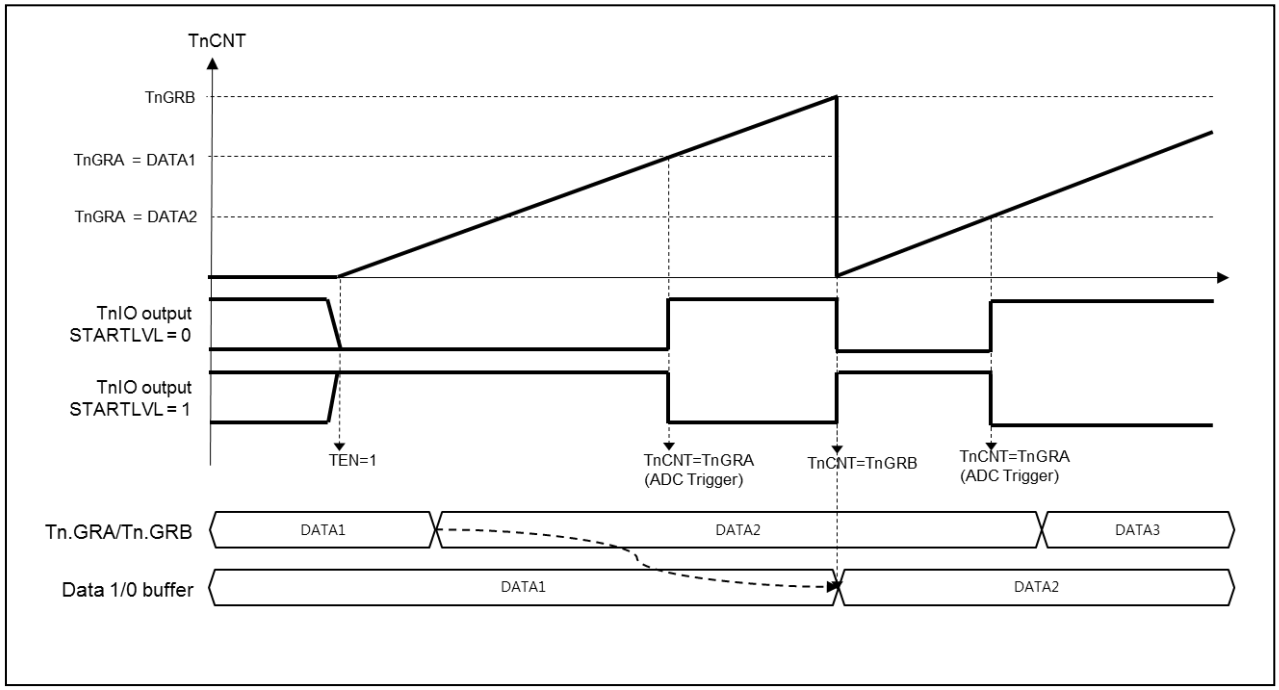


Figure 11.5. PWM Output Operation

The PWM pulse period can be calculated as follows:

The period = TMCLK Period * Tn.GRB value

Match A interrupt time = TMCLK Period * Tn.GRA value

If Tn.GRB = 0, the timer cannot be started even if TnCR2.TEN is “1” because the period is “0”.

The values in Tn.GRA and Tn.GRB are loaded into the internal compare data buffers 0 and 1 when the loading condition occurs. In this mode, the Tn.CR2.TCLR write operation and the GRB match event will load the data buffer.

The TnIO output signal generates a PWM pulse. The Tn.GRB value defines the output pulse period and the Tn.GRA value defines the pulse width of one shot pulse. The active level of PWM pulse can be controlled by the Tn.CR1.STARTLVL bit value.

ADC Trigger generation is available at Match A interrupt time.

Capture Mode

Figure 11.6 shows the timing diagram in capture mode operation. The TnIO input signal is used for the capture pulse. Both rising and falling edges can capture the counter values in each capture condition.

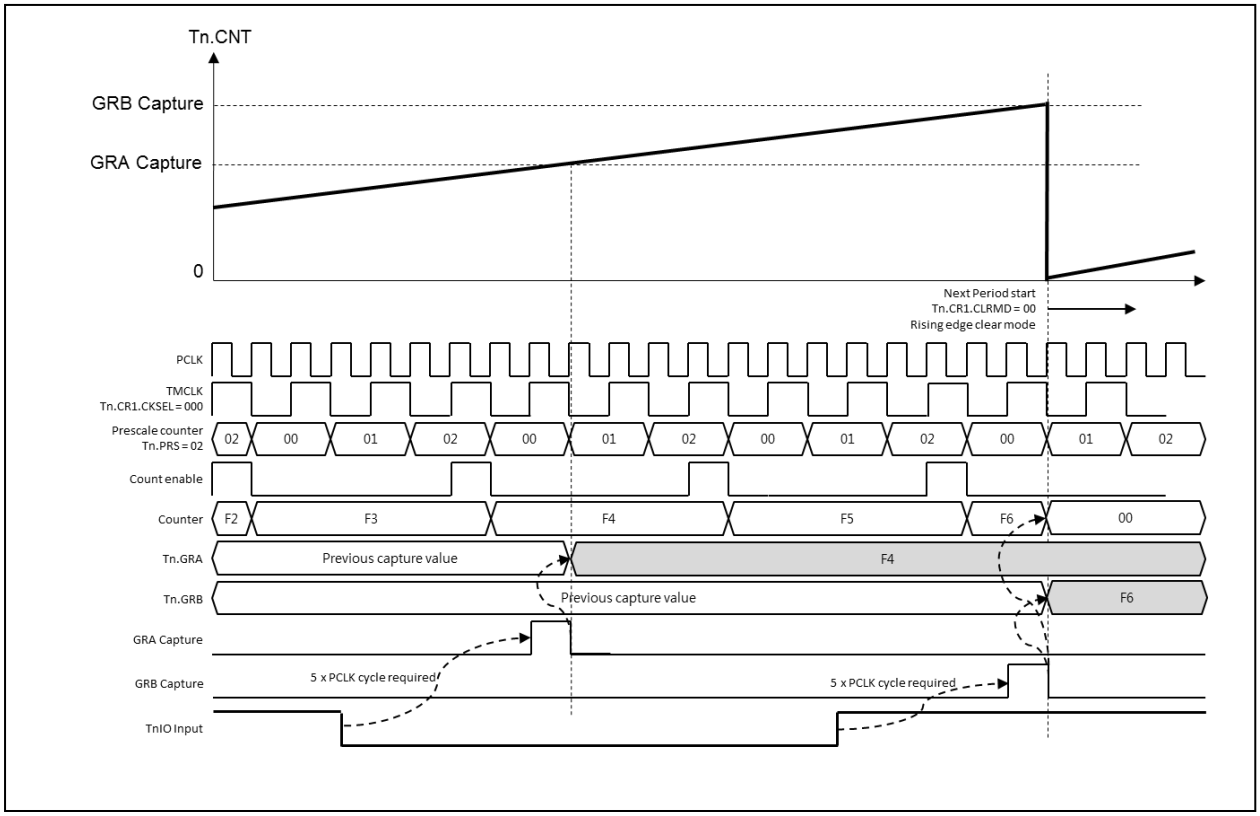


Figure 11.6. Capture Mode Operation

A 5 PCLK clock cycle is required internally. Therefore, the actual capture point is after the 5 PCLK clock cycle from the rising or falling edge of the TnIO input signal.

The internal counter can be cleared in multiple modes. The TnCR1.CLRMD field controls the counter clear mode. Rising edge clear mode, falling edge clear mode, both edge clear mode and none clear mode are supported.

ADC Trigger Function

The timer module can generate ADC start trigger signals. One timer can be one trigger source of the ADC block. Trigger source control is performed by the ADC control register.

Figure 11.7 shows the ADC trigger function.

The conversion rate must be shorter than the timer period; else an overrun situation can occur. ADC acknowledge is not required because the trigger signal is cleared automatically after 3 PCLK clock pulses.

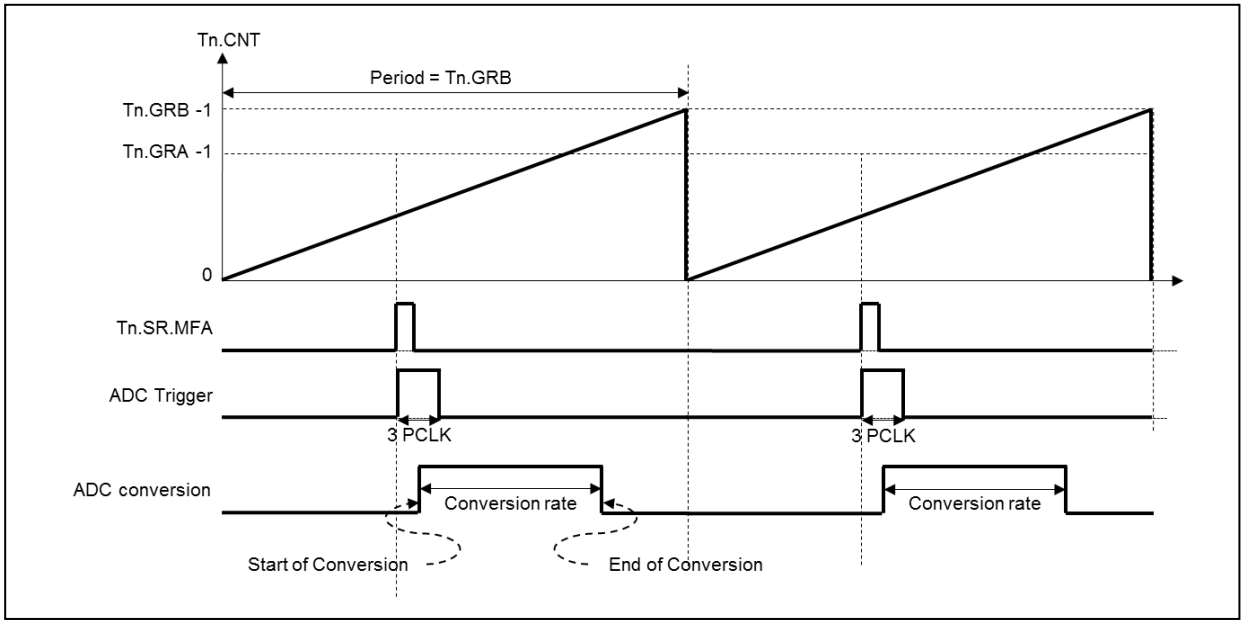


Figure 11.7. ADC Trigger Function Timing Diagram

Setup Example: Using the 16-bit Timer0 for Continuous Mode Operation

1. Enable the Timer0 peripheral by writing the appropriate value to the Peripheral Enable Register (PER1).
2. Enable the Timer0 peripheral clock by writing the appropriate value to the Peripheral Clock Enable Register (PCER).
3. Stop Timer0 before modifying the Timer0 registers by resetting bit0 in the Timer Control Register2 (TnCR2).
4. In Timer Control Register1 (TnCR1), write the appropriate value to enable the Timer0 Normal Period Operation Mode (e.g. 0x0000).
5. Write the appropriate Timer prescalar value to the Timer Prescalar Register (TnPRS).
6. Write the appropriate Timer count match value to the Timer General Register A (TnGRA) register. This timer count match value is compared to the actual count value in the Timer Count Register (TnCNT).
7. Write the appropriate value to Timer Interrupt Enable Register (TnIER) to enable or disable the Timer interrupt.
8. Start the Timer by setting bit0 and bit1; Timer Control Register2 (TnCR2) is enabled and initialized.

Note: Timer General Register A (TnGRA) is used for normal Timer operations. Timer General Register B (TnGRB) is used for Timer PWM modes.

Quadrature Encoder Interface

To use the Quadrature Encoder Interface Mode, Timer 0–Timer 3 are used for the input signals, holding the counter information and issuing the interrupts as necessary. The Timer mode for each of the timers used must be Capture Mode (TnCR1.MODE).

The Quadrature Encoder Interface peripheral receives pulses from the input of Timer 0 (Phase A), Timer 1 (Phase B) and Timer 2 (Phase Z) and processes the information to determine position, direction, and optionally, speed. The input for Timer 3 is not used. The position and revolution counters both use the Timer 2 input (as the Phase Z, or Index input).

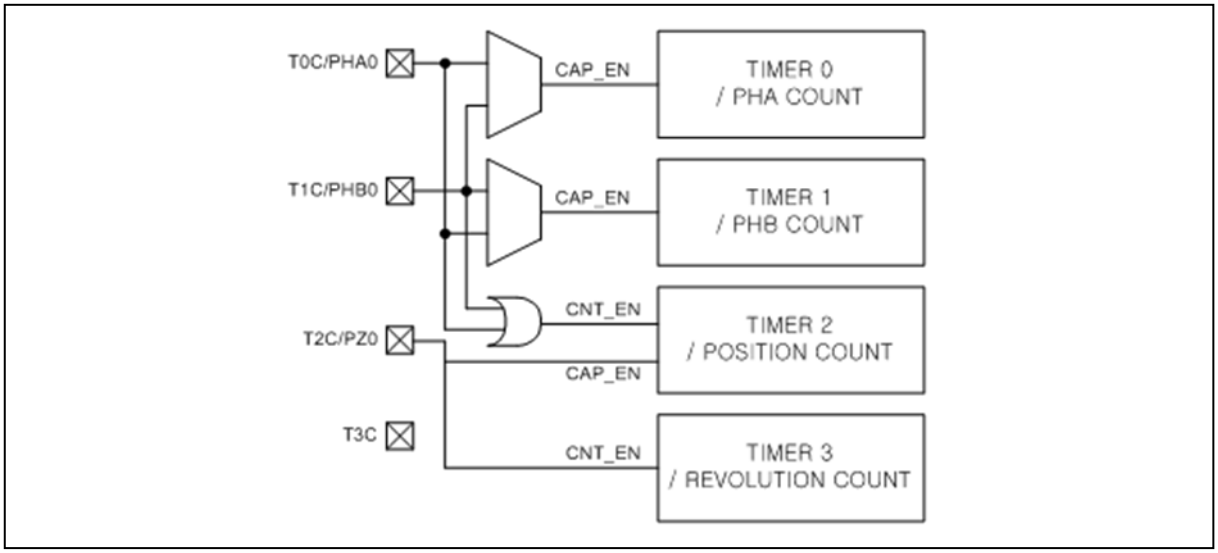


Figure 11.8. Quadrature Encoder Interface Counter Block

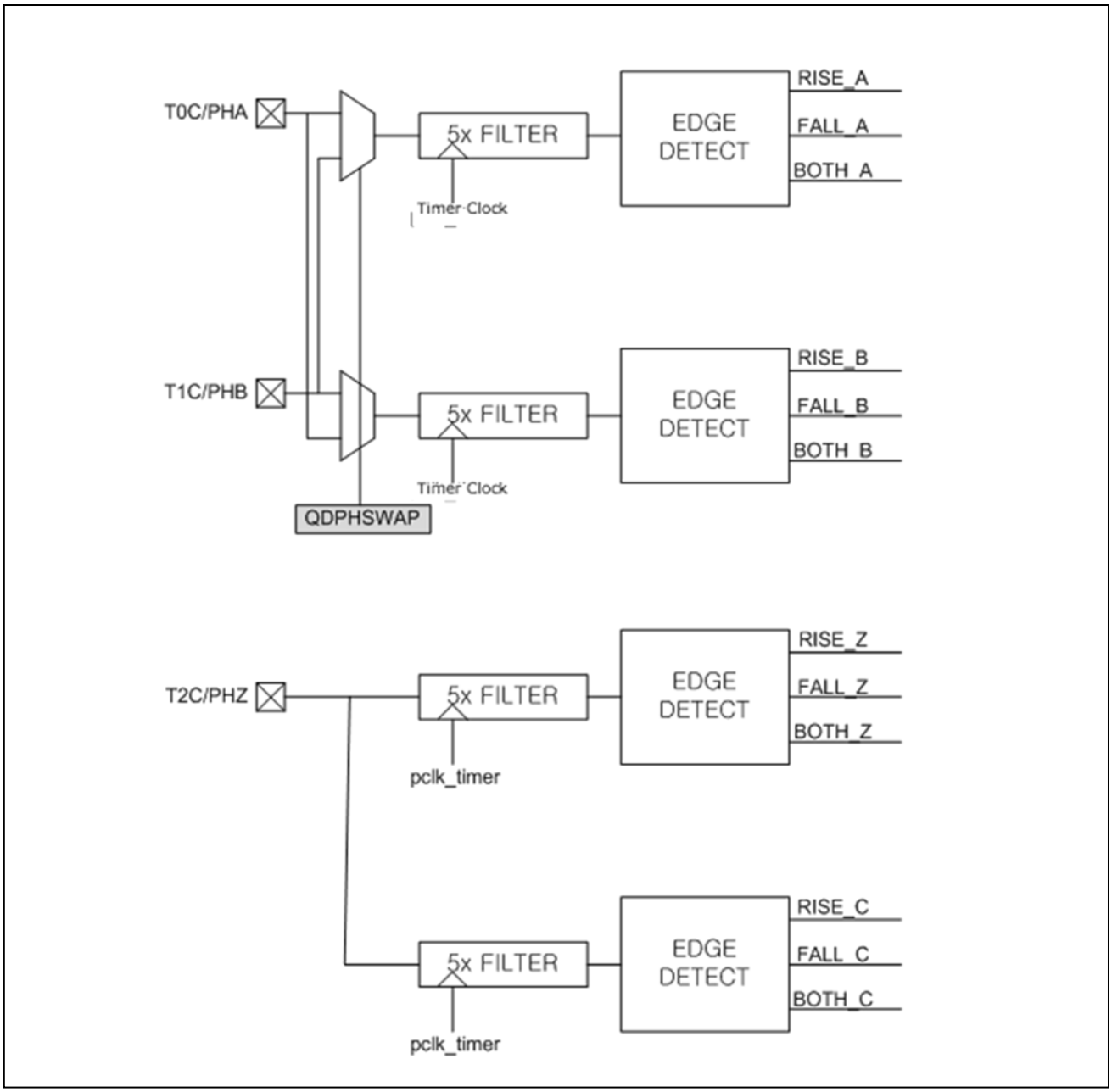


Figure 11.9. Quadrature Encoder Interface Input Block

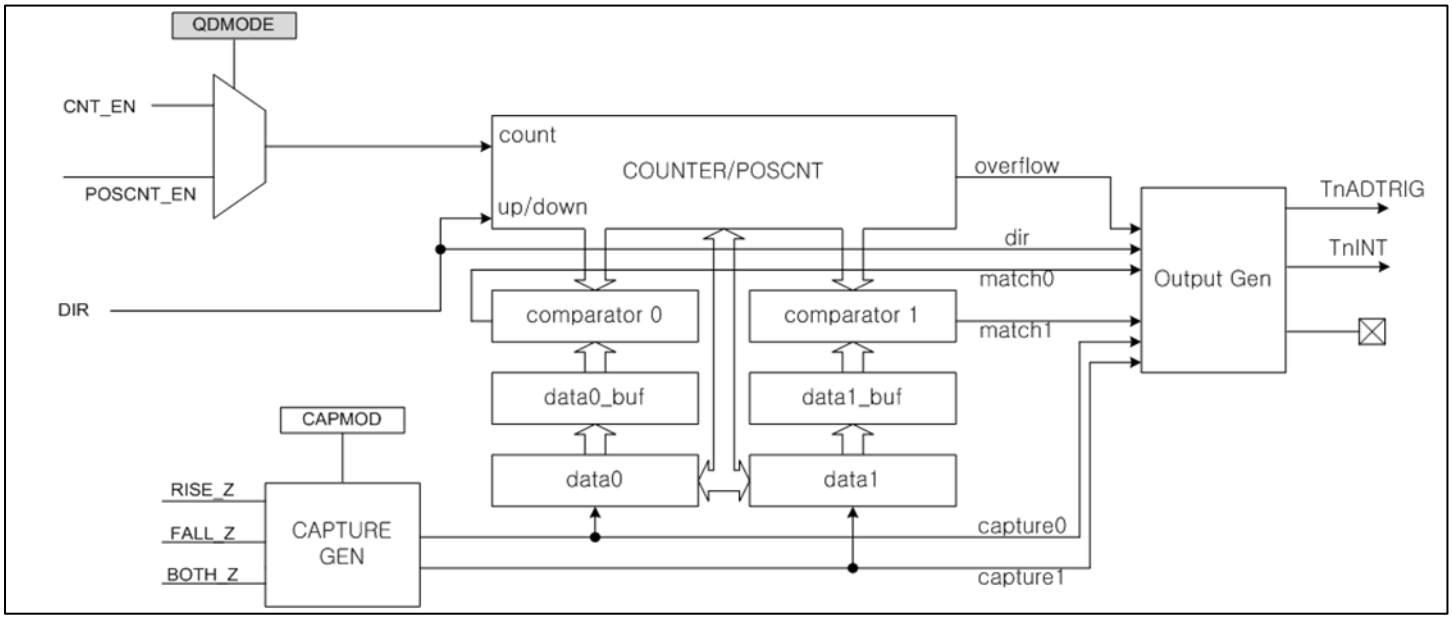


Figure 11.10. Quadrature Encoder Interface Block

The Phase A and Phase B inputs are controlled by the timer clock configuration (TnCR1.CKSEL). The GRA register contains the pulse count from the latest pulse and GRB register contains the previous pulse count. When the MFx interrupt is received, the MFx register has been updated. Depending on how TnCR1.CLRMOD and TnCR1.CKSEL are configured, this value could either be the pulse count or the time between pulses (up to the 16-bit count).

Phase A and Phase B can generate interrupts for QRF (Pulse received), QDIRCH (Direction changed). The QDIR status bit is set or cleared depending on the direction calculated from the last set of pulses from Phase A compared to Phase B.

Timer 2 (Position counter) counts the pulses from Phase A and Phase B. The GRB register contains the number of Phase A + Phase B pulses for each Phase Z input received. The CNT register is the current Phase A + Phase B pulses (giving the position within the Phase Z revolutions).

Timer 3 (Revolution counter) counts the pulses from the Phase Z input. Timer 3 only generates the QRF interrupt on receiving the Phase Z pulse.

To enable the Quadrature Encoder Interface interrupts, TGECR.QDMOD must be set before the desired interrupts in the TnIER register can be set.

12. Universal Asynchronous Receiver/Transmitter

Overview

4-Channel Universal Asynchronous Receiver/Transmitter (UART) modules are provided. Dedicated DMA support exists to transfer data between the memory buffer and the Transmit or Receive buffer of the UART block.

The UART operation status, including error status, can be read from the status register. The prescaler which generates the correct baud rate, exists for each UART channel. The prescaler can divide the UART clock source, PCLK/2, from 1 to 65535. The baud rate is generated by the clock which is internally divided by 16 of the prescaled clock and 8-bit precision clock tuning function.

A programmable interrupt generation function helps control communication via the UART channel.

Features of the UART include:

- Compatible with 16450
- Supports DMA transfer
- Standard asynchronous control bit (start, stop, and parity) configurable
- Programmable 16-bit fractional baud generator
- Programmable serial communication
- 5-, 6-, 7,- or 8- bit data transfer
- Even, odd, or no-parity bit insertion and detection
- 1-, 1.5,- or 2-stop bit-insertion and detection
- 16-bit baud rate generation with 8-bit fraction control
- Hardware inter-frame delay function
- Stop bit error detection
- Detail status register
- Loop-back control

Figure 12.1 shows a block diagram of the UART.

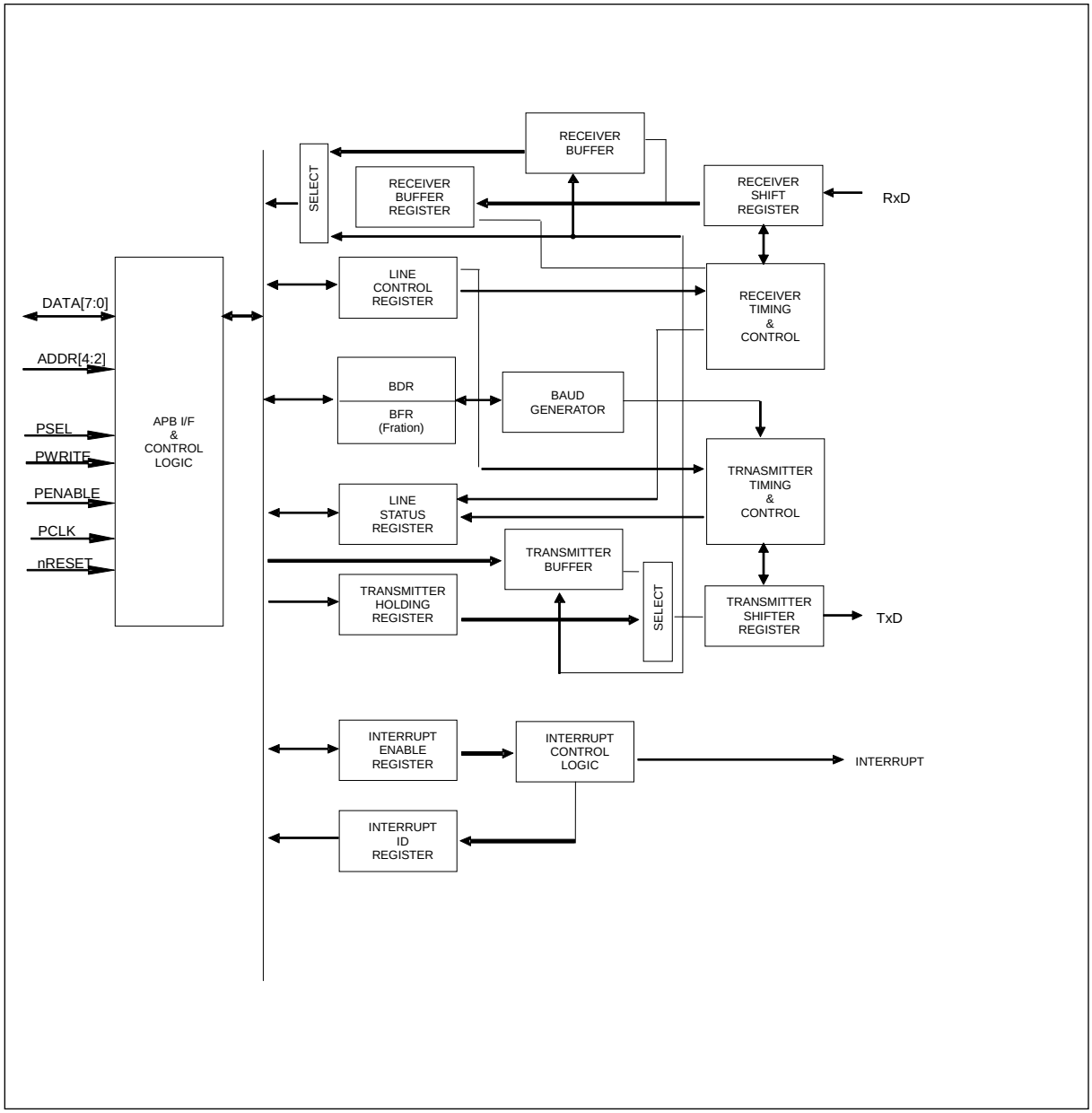


Figure 12.1. UART Block Diagram

Pin Description

Table 12.1. External Signal

PIN NAME	TYPE	DESCRIPTION
TXD0	O	UART Channel 0 transmit output
RXD0	I	UART Channel 0 receive input
TXD1	O	UART Channel 1 transmit output
RXD1	I	UART Channel 1 receive input
TXD2	O	UART Channel 2 transmit output
RXD2	I	UART Channel 2 receive input
TXD3	O	UART Channel 3 transmit output
RXD3	I	UART Channel 3 receive input

Registers

The base address of UART is `0x4000_8000` and the register map is described in Table 12.2 and Table 12.3.

Table 12.2. Base Address of Each Port

UART Channel	Address
UART 0	0x4000_8000
UART 1	0x4000_8100
UART 2	0x4000_8200
UART 3	0x4000_8300

Table 12.3. UART Register Map

Name	Offset	R/W	Description	Reset
UnRBR	0x00	R	Receive data buffer register	0x00
UnTHR	0x00	W	Transmit data hold register	0x00
UnIER	0x04	R/W	Interrupt enable register	0x00
UnIIR	0x08	R	Interrupt ID register	0x01
UnLCR	0x0C	R/W	Line control register	0x00
UnDCR	0x10	R/W	Data Control Register	0x00
UnLSR	0x14	R	Line status register	0x60
UnBDR	0x20	R/W	Baud rate Divisor Latch Register	0x0000
UnBFR	0x24	R/W	Baud rate Fractional Counter Value	0x00
UnIDTR	0x30	R/W	Inter-frame Delay Time Register	0x00

UnRBR Receive Buffer Register

The UART Receive Buffer Register is an 8-bit read-only register.

U0RBR=0x4000_8000, U1RBR=0x4000_8100
U2RBR=0x4000_8200, U3RBR=0x4000_8300

7	6	5	4	3	2	1	0
RBR							
-							
RO							

7 0	RBR	Receive Buffer Register

UnTHR Transmit Data Hold Register

The UART Transmit Data Hold Register is an 8-bit write-only register.

U0THR=0x4000_8000, U1THR=0x4000_8100
U2THR=0x4000_8200, U3THR=0x4000_8300

7	6	5	4	3	2	1	0
THR							
-							
WO							

7 0	THR	Transmit Data Hold Register

UnIER UART Interrupt Enable Register

The UART Interrupt Enable Register is an 8-bit register.

U0IER=0x4000_8004, U1IER=0x4000_8104
U2IER=0x4000_8204, U3IER=0x4000_8304

7	6	5	4	3	2	1	0
-	-	DTXIEN	DRXIEN	-	RLSIE	THREIE	DRIE
0	0	0	0	0	0	0	0
		RW	RW		RW	RW	RW

5	DTXIEN	DMA transmit done interrupt enable
		0 DMA transmit done interrupt is disabled
		1 DMA transmit done interrupt is enabled
4	DRXIEN	DMA receive done interrupt enable
		0 DMA receive done interrupt is disabled
		1 DMA receive done interrupt is enabled
2	RLSIE	Receiver line status interrupt enable
		0 Receive line status interrupt is disabled
		1 Receive line status interrupt is enabled
1	THREIE	Transmit holding register empty interrupt enable
		0 Transmit holding register empty interrupt is disabled
		1 Transmit holding register empty interrupt is enabled
0	DRIE	Data receive interrupt enable
		0 Data receive interrupt is disabled
		1 Data receive interrupt is enabled

UnIIR UART Interrupt ID Register

The UART Interrupt ID Register is an 8-bit register.

U0IIR=0x4000_8008, U1IIR=0x4000_8108
U2IIR=0x4000_8208, U3IIR=0x4000_8308

7	6	5	4	3	2	1	0
				IID			IPEN
0	0	0	0	000			0
				R			R

3	IID	Interrupt source ID
1		See interrupt source ID table
0	IPEN	Interrupt pending bit
		0 Interrupt is pending
		1 No interrupt is pending.

UART supports 3-priority interrupt generation. The interrupt source ID register shows one interrupt source which has the highest priority among pending interrupts. The priority is defined as:

- Receive line status interrupt
- Receive data ready interrupt
- Transmit hold register empty interrupt
- Tx/Rx DMA complete interrupts

Table 12.4. Interrupt ID and Control

Priority	DMA	IID		IPEN	Interrupt sources		
	Bit 3	Bit 2	Bit 1	Bit 0	Interrupt	Interrupt condition	Interrupt clear
-	0	0	0	1	None	-	-
Highest 1	0	1	1	0	Receiver Line Status	Overrun, Parity, Framing or Break Error	Read LSR register
2	0	1	0	0	Receiver Data Available	Receive data is available.	Read receive register or read IIR register
3	0	0	1	0	Transmitter Holding Register Empty	Transmit buffer empty	Write transmit hold register or read IIR register
4	1	1	0	0	Rx DMA done	Rx DMA completed.	Read IIR register
5	1	0	1	0	Tx DMA done	Tx DMA completed.	Read IIR register

UnLCR UART Line Control Register

The UART Line Control Register is an 8-bit register.

U0LCR=0x4000_800C, U1LCR=0x4000_810C
U2LCR=0x4000_820C, U3LCR=0x4000_830C

7	6	5	4	3	2	1	0
	BREAK	STICKP	PARITY	PEN	STOPBIT	DLEN	
0	0	0	0	0	0	0	0
	RW	RW	RW	RW	RW	RW	RW

6	BREAK	When this bit is set, TxD pin will be driven at low state in order to notice the alert to the receiver.
	0	Normal transfer mode
	1	Break transmit mode
5	STICKP	Force parity and it will be effective when PEN bit is set. See Table9.5
	0	Parity stuck is disabled
	1	Parity stuck is enabled
4	PARITY	Parity mode selection bit and stuck parity select bit
	0	Odd parity mode
	1	Even parity mode
3	PEN	Parity bit transfer enable
	0	The parity bit disabled
	1	The parity bit enabled
2	STOPBIT	The number of stop bit followed by data bits.
	0	1 stop bit
	1	1.5 / 2 stop bit In case of 5 bit data case, 1.5 stop bit is added. In case of 6,7 or 8 bit data, 2 stop bit is added
1 0	DLEN	The data length in one transfer word.
	00	5 bit data
	01	6 bit data
	10	7 bit data
	11	8 bit data

The parity bit is generated according to bits 3,4,5 of the UnLCR register. Table 12.5 shows the variation of parity bit generation.

Table 12.5. Variation of Parity Bit Generation

STICKP	PARITY	PEN	Parity
X	X	0	No Parity
0	0	1	Odd Parity
0	1	1	Even Parity
1	0	1	Force parity as “1”
1	1	1	Force parity as “0”

UnDCR UART Data Control Register

The UART Data Control Register is an 8-bit register.

U0DCR=0x4000_8010, U1DCR=0x4000_8110
U2DCR=0x4000_8210, U3DCR=0x4000_8310

7	6	5	4	3	2	1	0
			LBON	RXINV	TXINV		
0	0	0	0	0	0	0	0
			RW	RW	RW		

4	LBON	Local loopback test mode enable	
		0	Normal mode
		1	Local loopback mode (TxD connected to RxD internally)
3	RXINV	Rx Data Inversion Selection	
		0	Normal RxData Input
		1	Inverted RxData Input
2	TXINV	Tx Data Inversion Selection	
		0	Normal TxData Output
		1	Inverted TxData Output

UnLSR UART Line Status Register

The UART Line Status Register is an 8-bit register.

U0LSR=0x4000_8014, U1LSR=0x4000_8114
U2LSR=0x4000_8214, U3LSR=0x4000_8314

7	6	5	4	3	2	1	0
-	TEMT	THRE	BI	FE	PE	OE	DR
0	1	1	0	0	0	0	0
	R	R	R	R	R	R	R

6	TEMT	Transmit empty.
	0	Transmit register has the data is now transferring
	1	Transmit register is empty.
5	THRE	Transmit holding empty.
	0	Transmit holding register is not empty.
	1	Transmit holding register empty
4	BI	Break condition indication bit
	0	Normal status
	1	Break condition is detected
3	FE	Frame Error.
	0	No framing error.
	1	Framing error. The receive character did not have a valid stop bit
2	PE	Parity Error
	0	No parity error
	1	Parity error. The receive character does not have correct parity information.
1	OE	Overrun error
	0	No overrun error
	1	Overrun error. Additional data arrives while the RHR is full
0	DR	Data received
	0	No data in receive holding register.
	1	Data has been received and is saved in the receive holding register

This register provides the status of data transfers between the transmitter and the receiver. Users can get the line status information from this register to handle the next process. Bits 1,2,3,4 will raise the line status interrupt when the RLSIE bit in UnIEN register is set. Other bits can generate its interrupt when its interrupt enable bit in the UnIEN register is set.

UnBDR Baud rate Divisor Latch Register

The UART Baud rate Divisor Latch Register is a 16-bit register.

U0BDR=0x4000_8020, U1BDR=0x4000_8120
U2BDR=0x4000_8220, U3BDR=0x4000_8320

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BDR															
0x0000															
RW															

15 0	BDR	Baud rate Divider latch value

To establish communication with the UART channel, the baud rate should be set properly. The programmable baud rate generator UnBDR provides the 16-bit dividers values. The 16 bit divider register (UnBDR) should be written for the expected baud rate.

The baud rate calculation formula is:

$$\mathrm{BDR} = \frac{PCLK\ /\ 2}{16 \times\ BaudRate}$$

For a speed of 72 MHz PCLK, the divider value and error rate is described in Table 12.6.

Table 12.6. Example of Baud Rate Calculation

PCLK=72 MHz		
Baud rate	**Divider (BDR)**	**Error (%)**
1200	1875	0.00%
2400	937	0.05%
4800	468	0.16%
9600	234	0.16%
19200	117	0.16%
38400	58	1.02%
57600	39	0.16%
115200	19	2.79%

UnBFR Baud Rate Fraction Counter Register

The Baud Rate Fraction Counter Register is an 8-bit register.

U0BFR=0x4000_8024, U1BFR=0x4000_8124
U2BFR=0x4000_8224, U3BFR=0x4000_8324

7	6	5	4	3	2	1	0
BFR							
0x00							
RW							

7 0	BFR	Fractions counter value.	
		0	Fraction counter is disabled
		N	Fraction counter enabled. Fraction compensation mode is operating. Fraction counter is incremented by FCNT.

Table 12.7. Example of Baud Rate Calculation with BFR

PCLK=72 MHz			
Baud rate	**Divider (BDR)**	**FCNT (BFR)**	**Error (%)**
1200	1875	0	0.0%
2400	937	128	0.0%
4800	468	192	0.0%
9600	234	96	0.0%
19200	117	48	0.0%
38400	58	152	0.0%
57600	39	16	0.0%
115200	19	136	0.0%

$$\mathrm{FCNT} = \mathrm{Float} * 256$$

The 8-bit fractional counter counts up by FCNT value every (baud rate)/16 period and whenever the fractional counter overflows, the divisor value increments by 1. Therefore, this period will be compensated. In the next period, the divisor value returns to the original set value.

For example, if 9600 bps,

$$\frac{PCLK\ /\ 2}{16 \times\ BaudRate} = \frac{72000000\ /\ 2}{16 \times\ 9600} = 234.375 \quad Divider = 234, Float = 0.375$$

$$\mathrm{FCNT} = \mathrm{Float} * 256\ = 0.375 * 256 = 96$$

BDR = 234, BFR = 96

UnIDTR Inter-frame Delay Time Register

The UART Inter-frame Time Register is an 8-bit register. A dummy delay can be inserted between 2 continuous transmits.

U0IDTR=0x4000_8030, U1IDTR=0x4000_8130
U2IDTR=0x4000_8230, U3IDTR=0x4000_8330

7	6	5	4	3	2	1	0
-					WAITVAL		
0	0	0	0	0	000		
					RW		

2 0	WAITVAL	Wait time is decided by this value [unit: 1 bit time] $\text{Wait Time} = \frac{WAITVAL}{BAUDRATE}$

Functional Description

The PER2 and PCER2 registers must be configured to enable the UART peripheral and UART peripheral clock. The UART module is compatible with 16450 UART. Additionally, dedicated DMA channels and fractional baud rate compensation logic are provided.

Because there is no internal FIFO block, data transfers are established interactively or by using DMA support. The DMA operation is described here.

2 DMA channels are provided for each UART module – TX transfer and RX transfer. Each channel has a 32-bit memory address register and a 16-bit transfer counter register. Prior to the DMA operation, the DMA Memory Address register and the Transfer Count register should be configured. For the RX operation, the memory address will be the destination memory address and for the TX operation, the memory address will be the source memory address.

The transfer counter register will store the number of transfer data. When a single transfer is done, the counter will be decremented by 1. When the counter reaches zero, the DMA done flag will be delivered to the UART control block. If the interrupt is enabled, this flag will generate the interrupt.

Receiver Sampling Timing

The UARTs operate per the following timing:

If the falling edge is on the receive line, UART judges it as the start bit. From the start timing, the UART oversamples 16 times of 1-bit and detects the bit value at the 7th sample of 16 samples.

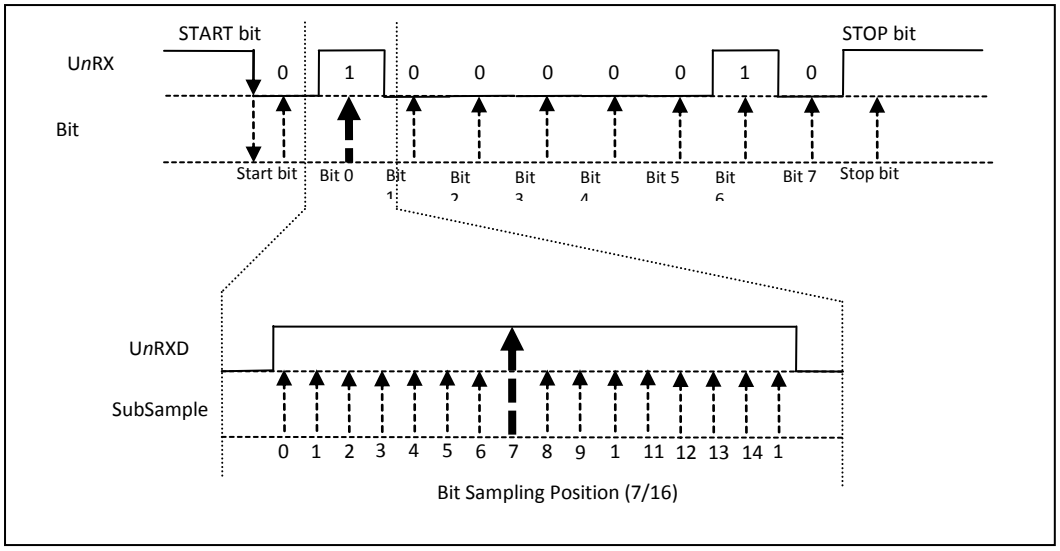


Figure 12.2. Sampling Timing of UART Receiver

Note: Zilog recommends enabling of debounce settings in the PCU block to reinforce the immunity of external glitch noise.

Transmitter

The transmitter's function is to transmit data. The start bit, data bits, optional parity bit, and stop bit are serially shifted, with the least significant bit first. The number of data bits is selected in the DLAN[1:0] field of the Un.LCR register.

The parity bit is set according to the PARITY and PEN bit field of the Un.LCR register. If the parity type is even, it depends on the one bit sum of all data bits. For odd parity, the parity bit is the inverted sum of all data bits.

The number of stop bits is selected in the STOPBIT field of the Un.LCR register.

An example of transmit data format is shown in Figure 12.3.

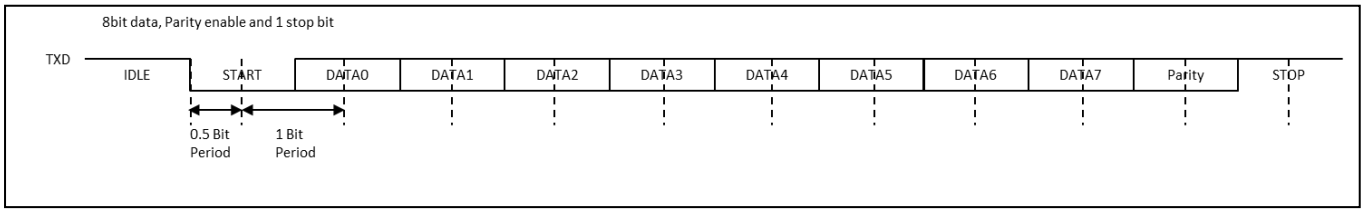


Figure 12.3. Transmit Data Format Example

Inter-frame Delay Transmission

The inter-frame delay function allows the transmitter to insert an idle state on the TXD line between two characters. The width of the idle state is defined in the WAITVAL field of the Un.IDTR register. When this field is set to 0, zero time-delay is generated. Otherwise, the transmitter holds a high level on TXD after each transmitted character during the number of bit periods defined in the WATIVAL field.

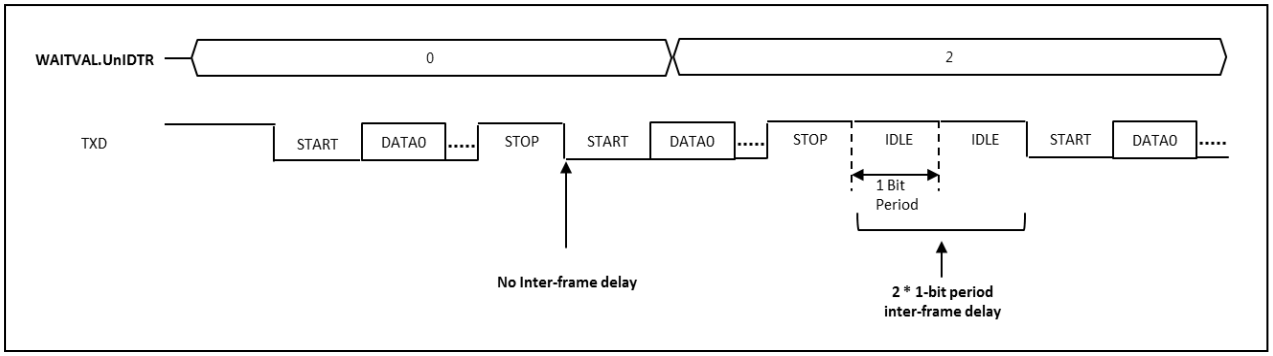


Figure 12.4. Transmit Data Format Example

Transmit Interrupt

The transmit operation generates interrupt flags. When the transmitter holding register is empty, the THRE interrupt flag is set. When the transmitter shifter register is empty, the TXE interrupt flag is set. Users can select the interrupt timing that is best for the application.

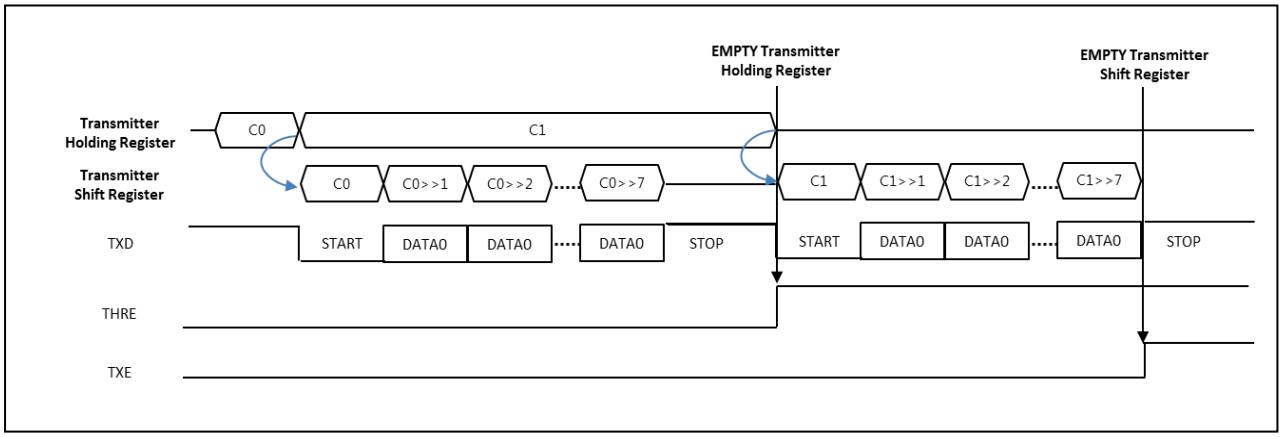


Figure 12.5. Transmit Data Format Example

DMA Transfers

The UART supports the DMA interface function. It is optionally provided depending on the device. The start memory address for transfer data and the length of transfer data are programmed in the registers in the DMA block.

The end of transfer is notified by the related transfer done flag. The transmit with DMA operation invokes the DMA TX done flag DTX.UnIIR and sets the DMA TX done interrupt ID when all the transmit data is written to the transmit holding register. Two transmit data values remain in the UART block registers after the DMA transfer done interrupt.

The Receive with DMA operation invokes the DMA RX done flag RXT.UnIIR and sets the DMA RX done interrupt ID when all the receive data is written to the destination memory. Therefore, the UART RXD signal is already in Idle state when the DMA RX done interrupt is issued.

13. Serial Peripheral Interface

Overview

2-channel serial interface is provided for synchronous serial communication with external peripherals. The SPI block supports master and slave modes. 4 signals are used for SPI communication – SS, SCK, MOSI, and MISO.

Features include:

- Master or slave operation
- Programmable clock polarity and phase
- 8,9,16,17-bit wide transmit/receive register
- 8,9,16,17-bit wide data frame
- Loop-back mode
- Programmable start, burst, and stop delay time
- DMA transfer operation.

Figure 13.1 shows the SPI Block Diagram.

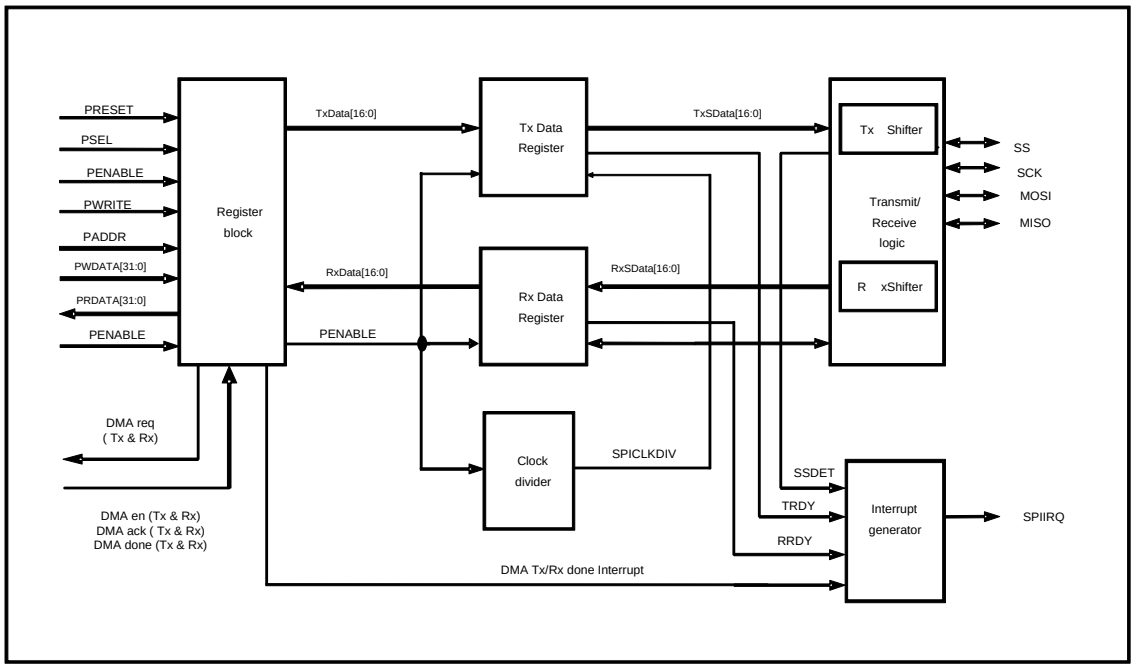


Figure 13.1. SPI Block Diagram

Pin Description

Table 13.1. External Pins

PIN NAME	TYPE	DESCRIPTION
SS0	I/O	SPI0 Slave select (Master output, Slave input)
SCK0	I/O	SPI0 Serial clock (Master output, Slave input)
MOSI0	I/O	SPI0 Serial data (Master output, Slave input)
MISO0	I/O	SPI0 Serial data (Master input, Slave output)
SS1	I/O	SPI1 Slave select (Master output, Slave input)
SCK1	I/O	SPI1 Serial clock (Master output, Slave input)
MOSI1	I/O	SPI1 Serial data (Master output, Slave input)
MISO1	I/O	SPI1 Serial data (Master input, Slave output)

Registers

The base address of SPI is `0x4000_9000` and the register map is described in Table 13.2 and Table 13.3.

Table 13.2. SPI Base Address

Channel	Base address
SPI0	0x4000_9000
SPI1	0x4000_9100

Table 13.3. SPI Register Map

Name	Offset	R/W	Description	Reset
SP*n*TDR	0x--00	W	SPI n Transmit Data Register	-
SP*n*RDR	0x--00	R	SPI n Receive Data Register	0x000000
SP*n*CR	0x--04	R/W	SPI n Control Register	0x001020
SP*n*SR	0x--08	R/W	SPI n Status Register	0x000006
SP*n*BR	0x--0C	R/W	SPI n Baud rate Register	0x0000FF
SP*n*EN	0x--10	R/W	SPI n Enable register	0x000000
SPnLR	0x--14	R/W	SPI n delay Length Register	0x010101

SP*n*CR SPI n Control Register

SPnCR is a 20-bit read/write register and can be set to configure the SPI operation mode.

SP0CR=0x4000_9004, SP1CR=0x4000_9104

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											TXBC	RXBC	TXDIE	RXDIE	SSCIE	TXIE	RXIE	SSMOD	SSOUT	LBE	SSMASK	SSMO	SSPOL			MS	MSBF	CPHA	CPOL	BITSZ	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	00	
											W	W	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW			RW	RW	RW	RW	RW		

20	**TXBC**	Tx buffer clear bit.
		0 No action
		1 Clear Tx buffer
19	**RXBC**	Rx buffer clear bit
		0 No action
		1 Clear Rx buffer
18	**TXDIE**	DMA Tx Done Interrupt Enable bit.
		0 DMA Tx Done Interrupt is disabled.
		1 DMA Tx Done Interrupt is enabled.
17	**RXDIE**	DMA Rx Done Interrupt Enable bit.
		0 DMA Rx Done Interrupt is disabled.
		1 DMA Rx Done Interrupt is enabled.
16	**SSCIE**	SS Edge Change Interrupt Enable bit.
		0 nSS interrupt is disabled.
		1 nSS interrupt is enabled for both edges (L→H, H→L)
15	**TXIE**	Transmit Interrupt Enable bit.
		0 Transmit Interrupt is disabled.
		1 Transmit Interrupt is enabled.
14	**RXIE**	Receive Interrupt Enable bit..
		0 Receive Interrupt is disabled.
		1 Receive Interrupt is enabled.
13	**SSMOD**	SS Auto/Manual output select bit in master mode.
		0 SS output is not set by SSOUT (SPnCR[12]).
		- SS signal is in normal operation mode.
		1 SS output signal is set by SSOUT.
12	**SSOUT**	SS output signal select bit in master mode.
		0 SS output is 'L.'
		1 SS output is 'H'.
11	**LBE**	Loop-back mode select bit in master mode.
		0 Loop-back mode is disabled.
		1 Loop-back mode is enabled.
10	**SSMASK**	SS signal masking bit in slave mode.
		0 SS signal masking is disabled.
		- Receive data when SS signal is active.
		1 SS signal masking is enabled.
		- Receive data at SCLK edges. SS signal is ignored.
9	**SSMO**	SS output signal select bit.
		0 SS output signal is disabled.
		1 SS output signal is enabled.
8	**SSPOL**	SS signal Polarity select bit.
		0 SS signal is Active-Low.
		1 SS signal is Active-High.
7		Reserved

6		
5	MS	Master/Slave select bit.
		0 SPI is in Slave mode.
		1 SPI is in Master mode.
4	MSBF	MSB/LSB Transmit select bit.
		0 LSB is transferred first.
		1 MSB is transferred first.
3	CPHA	SPI Clock Phase bit.
		0 Sampling of data occurs at odd edges (1,3,5,…,15).
		1 Sampling of data occurs at even edges (2,4,6,…,16).
2	CPOL	SPI Clock Polarity bit.
		0 Active-high clocks selected.
		1 Active-low clocks selected.
1	BITSZ	Transmit/Receive Data Bits select bit.
		00 8 bits
		01 9 bits
		10 16 bits
0		11 17 bits

CPOL=0, CPHA=0: data sampling at rising edge, data changing at falling edge
CPOL=0, CPHA=1: data sampling at falling edge, data changing at rising edge
CPOL=1, CPHA=0: data sampling at falling edge, data changing at rising edge
CPOL=1, CPHA=1: data sampling at rising edge, data changing at falling edge

SP*n*SR SPI n Status Register

SPnSR is a 10-bit read/write register. It contains the status of the SPI interface.

SP0SR=0x4000_9008, SP1SR=0x4000_9108

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						TXDMAF	RXDMAF		SSDET	SSON	OVRF	UDRF	TXIDLE	TRDY	RRDY
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
						RC1	RC1		RC1	RC1	RC1	RC1	R	R	R

9	**TXDMAF**	DMA Transmit Operation Complete flag. (DMA to SPI)
		0 DMA Transmit Op is working or is disabled.
		1 DMA Transmit Op is done.
8	**RXDMAF**	DMA Receive Operation Complete flag. (SPI to DMA)
		0 DMA Receive Operation is working or is disabled.
		1 DMA Transmit Op is done.
7		Reserved
6	**SSDET**	The rising or falling edge of SS signal Detect flag.
		0 SS edge is not detected.
		1 SS edge is detected. - The bit is cleared when it is written as “0”.
5	**SSON**	SS signal Status flag.
		0 SS signal is inactive.
		1 SS signal is active.
4	**OVRF**	Receive Overrun Error flag.
		0 Receive Overrun error is not detected.
		1 Receive Overrun error is detected. - This bit is cleared by writing or reading SPnRDR.
3	**UDRF**	Transmit Underrun Error flag.
		0 Transmit Underrun is not occurred.
		1 Transmit Underrun is occurred. - This bit is cleared by writing or reading SPnTDR.
2	**TXIDLE**	Transmit/Receive Operation flag.
		0 SPI is transmitting data
		1 SPI is in IDLE state.
1	**TRDY**	Transmit buffer Empty flag.
		0 Transmit buffer is busy.
		1 Transmit buffer is ready. - This bit is cleared by writing data to SPnTDR.
0	**RRDY**	Receive buffer Ready flag.
		0 Receive buffer has no data.
		1 Receive buffer has data. - This bit is cleared by reading data to SPnRDR.

SP*n*TDR SPI n Transmit Data Register

SPnTDR is a 17-bit read/write register. It contains serial transmit data.

SP0TDR=0x4000_9000, SP1TDR=0x4000_9100

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																TDR															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x00000															
																RW															

16 0	TDR	Transmit Data Register

SP*n*RDR SPI n Receive Data Register

SPnRDR is a 17-bit read/write register. It contains serial receive data.

SP0RDR=0x4000_9000, SP1RDR=0x4000_9100

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																RDR															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x00000															
																RW															

16 0	RDR	Receive Data Register

SP*n*BR SPI n Baud Rate Register

SPnBR is a 16-bit read/write register. Baud rate can be set by writing the register.

SP0BR=0x4000_900C, SP1BR=0x4000_910C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BR															
0x00FF															
RW															

15 0	BR	Baud rate setting bits - Baud Rate = PCLK / (BR + 1). (BR must be bigger than “0”, BR >= 2)

SP*n*EN SPI n Enable Register

SPnEN is an 8-bit read/write register. It contains the SPI enable bit.

SP0EN=0x4000_9010, SP1EN=0x4000_9110

7	6	5	4	3	2	1	0
							ENABLE
0	0	0	0	0	0	0	0
							RW

0	**ENABLE**	SPI Enable bit
		0 SPI is disabled. - SPnSR is initialized by writing “0” to this bit but other registers aren’t initialized.
		1 SPI is enabled. - When this bit is written as “1”, the dummy data of transmit buffer will be shifted. To prevent this, write data to SPTDR before this bit is active.

SP*n*LR SPI n Delay Length Register

SPnLR is a 24-bit read/write register. It contains start, burst, and stop length value.

SP0CR=0x4000_9014, SP1CR=0x4000_9114

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								SPL								BTL								STL							
0	0	0	0	0	0	0	0	0x01								0x01								0x01							
								RW								RW								RW							

23 16	SPL	StoPLength value 0x01 ~ 0xFF : 1 ~ 255 SCLKs. (SPL >= 1)
15 8	BTL	BursTLength value 0x01 ~ 0xFF : 1 ~ 255 SCLKs. (BTL >= 1)
7 0	STL	STart Length value 0x01 ~ 0xFF : 1 ~ 255 SCLKs. (STL >= 1)

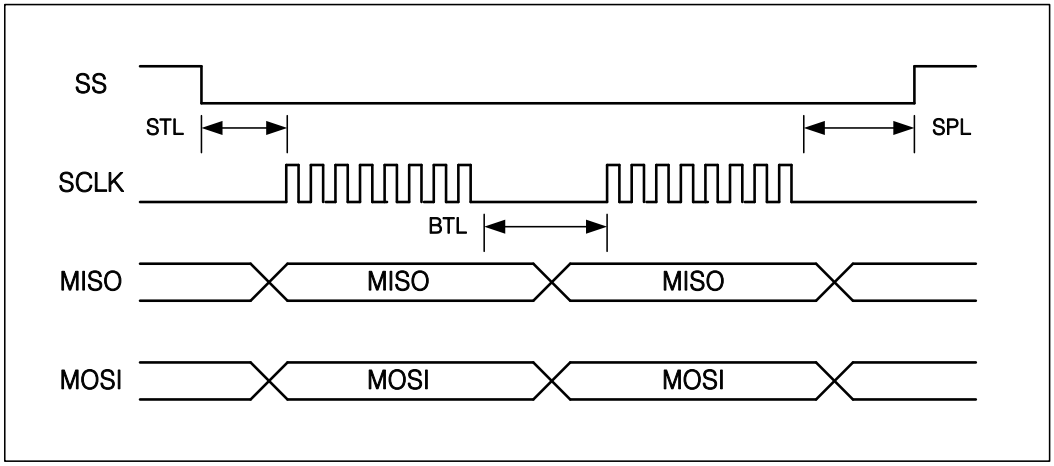


Figure 13.2. SPI waveform (STL, BTL, and SPL)

Functional Description

The SPI Transmit and Receive blocks share the Clock Gen Block; however, they are independent of each other. The Transmit and Receive blocks have double buffers and SPI is available for back-to-back transfer operations.

SPI Timing

The SPI has four modes of operation. These modes essentially control the way data is clocked in or out of an SPI device. The configuration is done by two bits in the SPI Control Register (SPnCR). The clock polarity is specified by the CPOL control bit, which selects an active high or active low clock. The clock phase (CPHA) control bit selects one of two fundamentally different transfer formats. To ensure proper communication

between master and slave, both devices have to run in the same mode. This may require a reconfiguration of the master to match the requirements of different peripheral slaves.

The clock polarity has no significant effect on the transfer format. Switching this bit causes the clock signal to be inverted (active high becomes active low and idle low becomes idle high). However, the settings of the clock phase select one of two different transfer timings, which are described in further detail in the next two chapters. Since the MOSI and MISO lines of the master and the slave are directly connected to each other, the diagrams show the timing of both these devices. The nSS line is the slave select input of the slave. The nSS pin of the master is not shown in the diagrams. It has to be inactivated by a high level on this pin (if configured as an input pin) or by configuring it as an output pin.

The timing of an SPI transfer where CPHA is zero is shown in Figure 13.3 and Figure 13.4. Two wave forms are shown for the SCK signal - one for CPOL equals zero and another for CPOL equals one.

When the SPI is configured as a slave, the transmission starts with the falling edge of the /SS line. This activates the SPI of the slave and the MSB of the byte stored in its data register (SPnTDR) is output on the MISO line. The actual transfer is started by a software write to the SPnTDR of the master. This causes the clock signal to be generated. If the CPHA equals zero, the SCLK signal remains zero for the first half of the first SCLK cycle. This ensures that the data is stable on the input lines of both the master and the slave. The data on the input lines is read with the edge of the SCLK line from its inactive to its active state. The edge of the SCLK line from its active to its inactive state (falling edge if CPOL equals zero and rising edge if CPOL equals one) causes the data to be shifted one bit further so that the next bit is output on the MOSI and MISO lines.

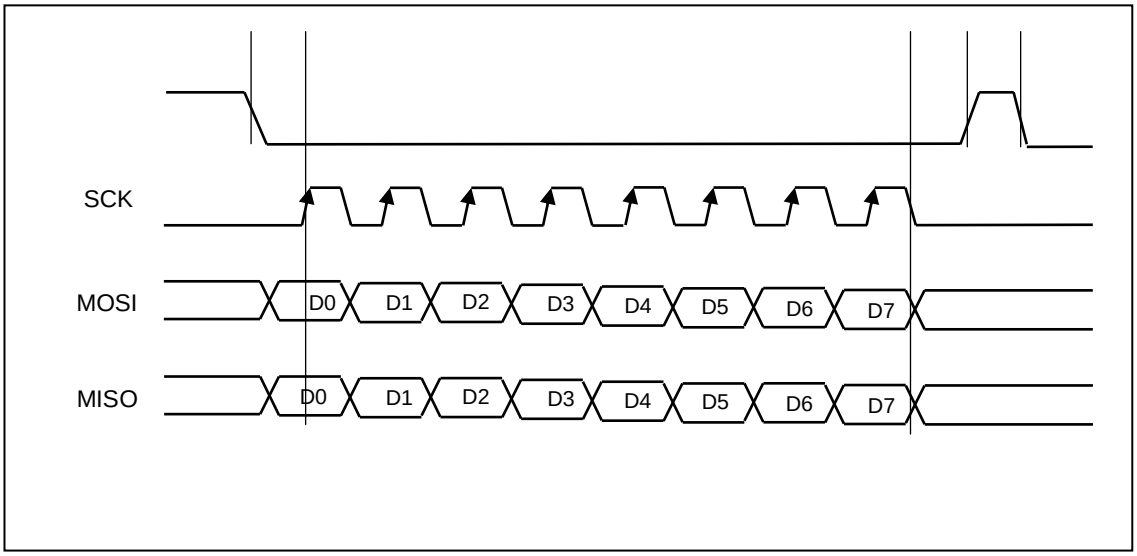


Figure 13.3. SPI Transfer Timing 1/4 (CPHA=0, CPOL=0, MSBF=0)

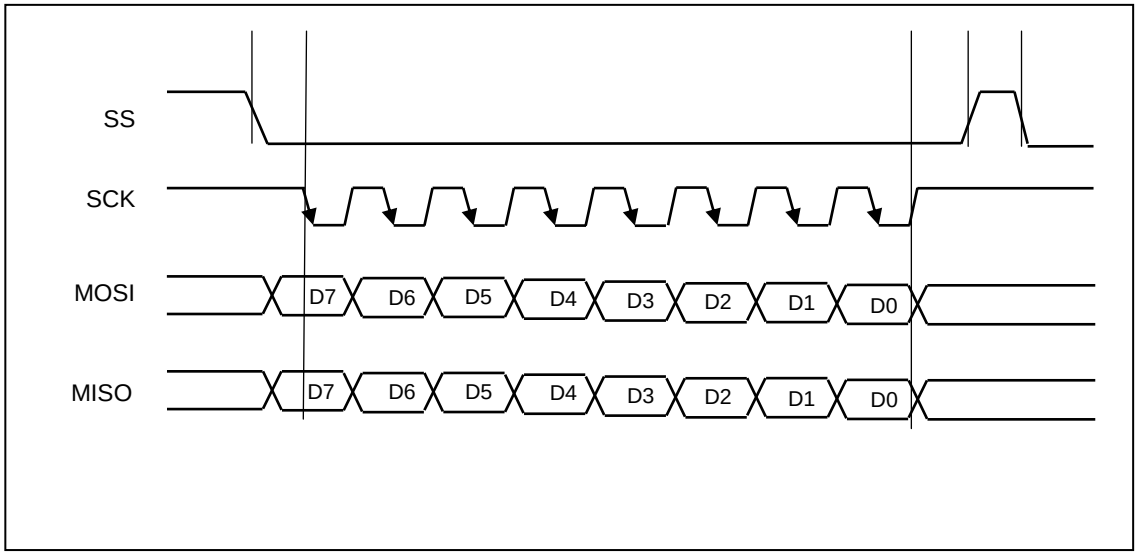


Figure 13.4.SPI Transfer Timing 2/4 (CPHA=0, CPOL=1, MSBF=1)

The timing of an SPI transfer where CPHA is one is shown in Figure 13.5 and Figure 13.6.Two wave forms are shown for the SCLK signal - one for CPOL equals zero and another for CPOL equals one.

As in the previous scenarios, the falling edge of the nSS lines selects and activates the slave. However, in contrast to the previous cases, where CPHA equals zero, the transmission is not started and the MSB is not output by the slave at this stage. The actual transfer is started by a software write to the SPnTDR of the master which causes the clock signal to be generated. The first edge of the SCLK signal from its inactive to its active state (rising edge if CPOL equals zero and falling edge if CPOL equals one) causes both the master and the slave to output the MSB of the byte in the SPnTDR.

As shown in Figure 13.3 and Figure 13.4, there is no delay of half a SCLK-cycle. The SCLK line changes its level immediately at the beginning of the first SCLK-cycle. The data on the input lines is read with the edge of the SCLK line from its active to its inactive state (falling edge if CPOL equals zero and rising edge if CPOL equals one). After eight clock pulses, the transmission is complete.

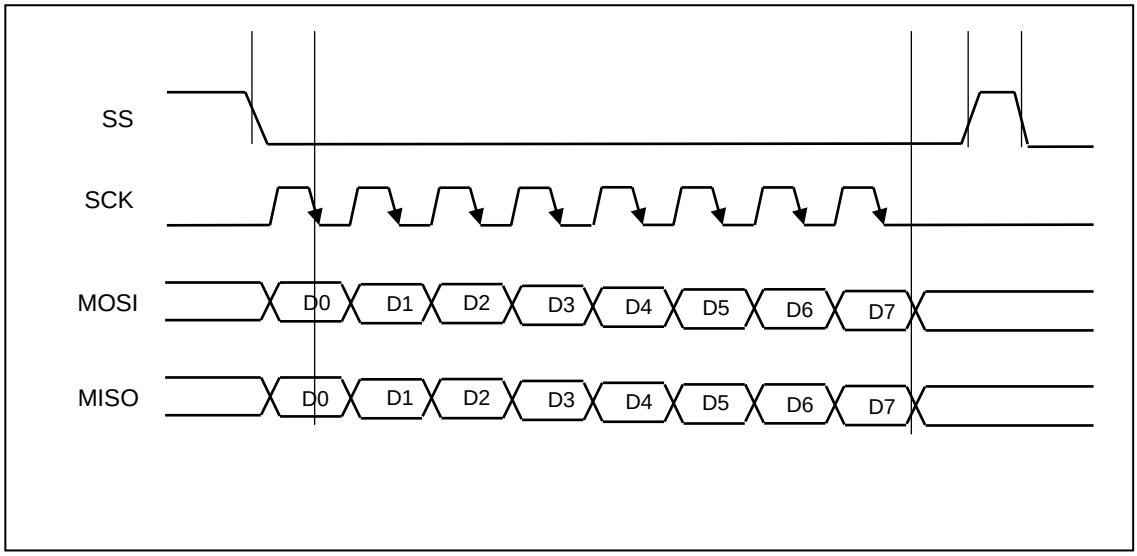


Figure 13.5.SPI Transfer Timing 3/4 (CPHA=1, CPOL=0, MSBF=0)

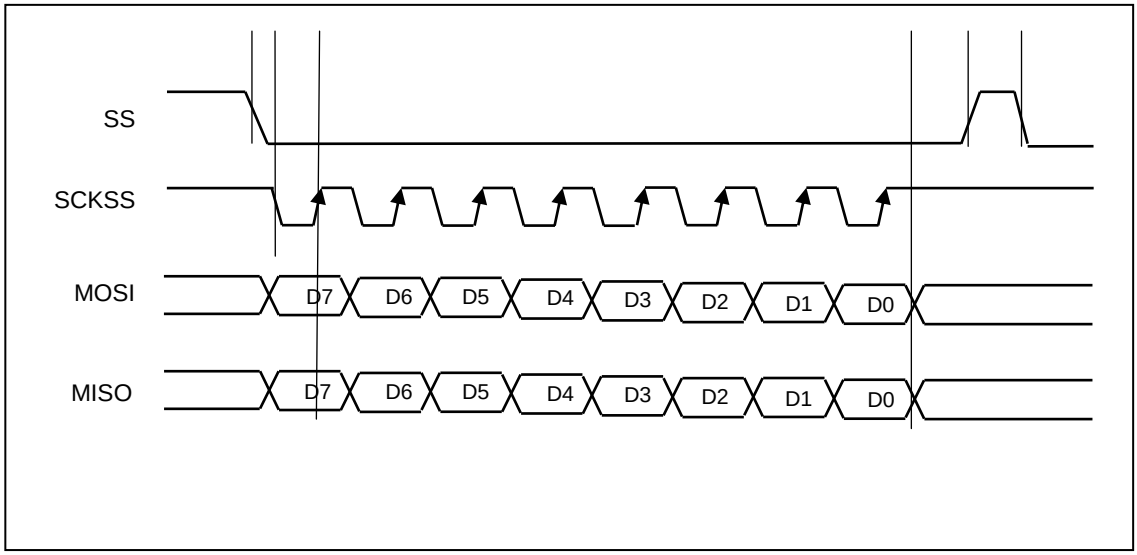


Figure 13.6.SPI Transfer Timing 4/4 (CPHA=1, CPOL=1, MSBF=1)

DMA Handshake

SPI supports the DMA handshaking operation. To operate a DMA handshake, DMA registers should be set first (see Chapter 9, Direct Memory Access Controller). SPI0 has 2 DMA channels – Channel 8 for the receiver and Channel 9 for the transmitter. SPI1 has Channel 10 for the receiver and Channel 11 for the transmitter. Because the transmitter and receiver are independent of each other, SPI can operate the two channels at the same time.

After the DMA channel for receiver is enabled and the receive buffer is filled, SPI sends Rx request to DMA to empty the buffer and waits for an ACK signal from DMA. If the Receive buffer is filled again after the ACK signal, SPI sends an Rx request. If DMA Rx DONE becomes high, RXDMAF (SPnSR[8]) is 1 and an interrupt is serviced when RXDIE (SPnCR[17]) is set.

Similarly, if the transmit buffer is empty after the DMA channel for transmitter is enabled, SPI sends a Tx request to the DMA to fill the buffer and waits for an ACK signal from the DMA. If the transmit buffer is empty again after the ACK signal, SPI sends a Tx request. If DMA Tx DONE becomes high, TXDMAF (SPnSR[9]) is **1** and an interrupt is serviced when TXDIE(SPnCR[18]) is set.

The slave transmitter sends dummy data at the first transfer (8~17 SCLKs) in DMA handshake mode.

Figure 13.7 shows a flowchart of the DMA handshaking process.

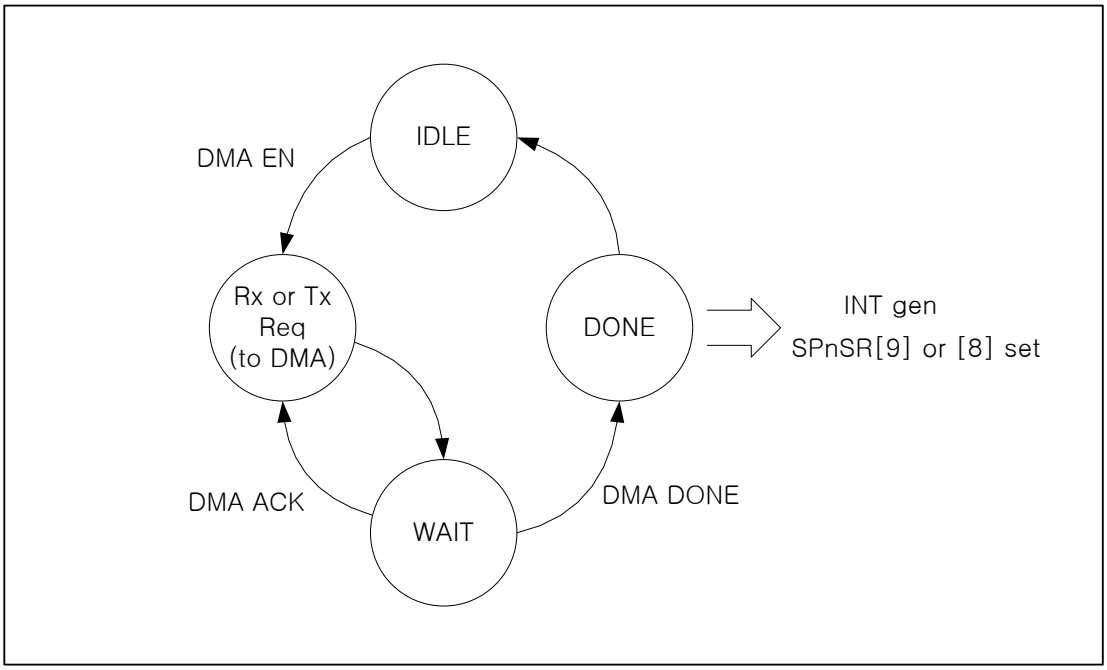


Figure 13.7. DMA Handshake Flowchart

14. I²C Interface

Overview

The Inter-Integrated Circuit (I^2C) bus serves as an interface between the microcontroller and the serial I^2C bus. It provides two wires, a serial bus interface to a large number of popular devices and allows parallel-bus systems to communicate bi-directionally with the I^2C-bus.

Features include:

- Master and slave operation
- Programmable communication speed
- Multi-master bus configuration
- 7-bit addressing mode
- Standard data rate of 100/400 kbps
- STOP signal generation and detection
- START signal generation
- ACK bit generation and detection

Figure 14.1 shows the I^2C block diagram.

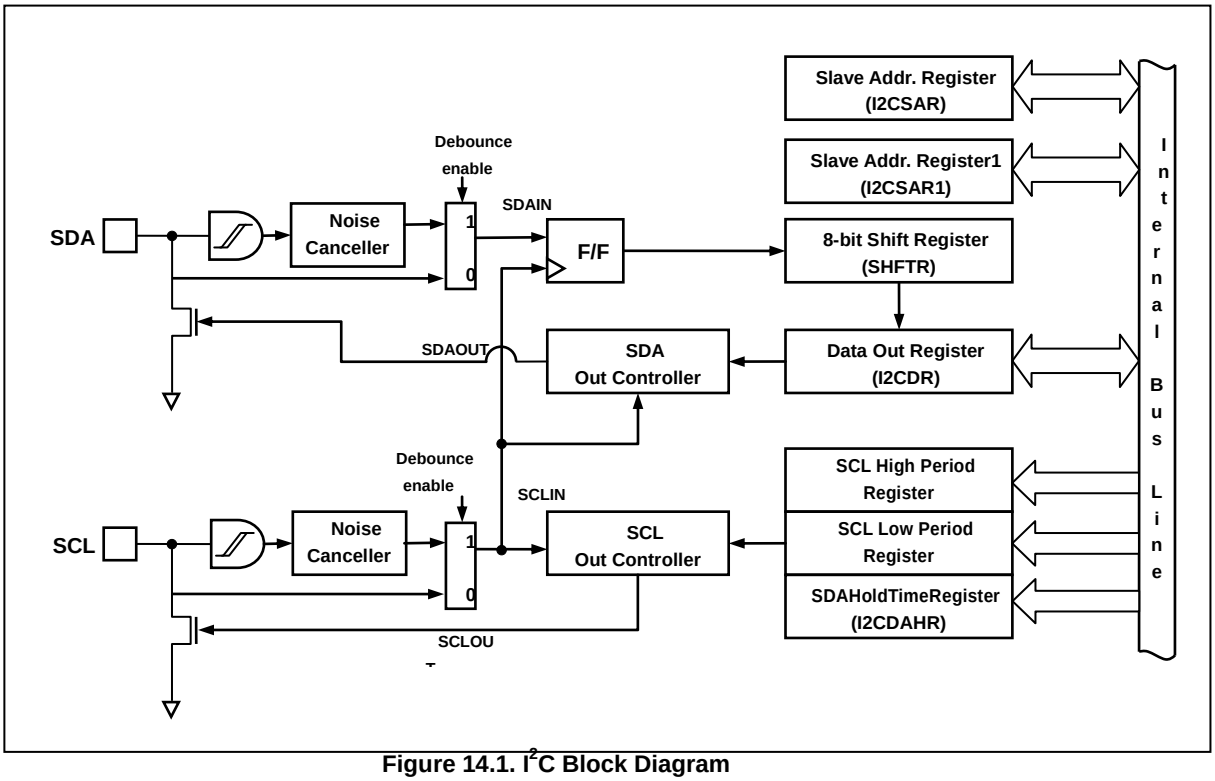


Figure 14.1. I^2C Block Diagram

Pin Description

Table 14.1. I²C Interface External Pins

PIN NAME	TYPE	DESCRIPTION
SCL0	I/O	I²C channel 0 Serial clock bus line (open-drain)
SDA0	I/O	I²C channel 0 Serial data bus line (open-drain)
SCL1	I/O	I²C channel 1 Serial clock bus line (open-drain)
SDA1	I/O	I²C channel 1 Serial data bus line (open-drain)

Registers

The base address of I²C0 is `0x4000_A000` and the base address of I²C1 is `0x4000_A100`. The register map is described in Table 14.2 and Table 14.3.

Table 14.2.I²C Interface Base Address

Channel	Base address
I²C0	0x4000_A000
I²C1	0x4000_A100

Table 14.3.I²C Register Map

Name	Offset	R/W	Description	Reset
IC0DR	0xA000	R/W	I²C0 Data Register	0xFF
IC0SR	0xA008	R, R/W	I²C0 Status Register	0x00
IC0SAR	0xA00C	R/W	I²C0 Slave Address Register	0x00
IC0CR	0xA014	R/W	I²C0 Control Register	0x00
IC0SCLL	0xA018	R/W	I²C0 SCL LOW duration Register	0xFFFF
IC0SCLH	0xA01C	R/W	I²C0 SCL HIGH duration Register	0xFFFF
IC0SDH	0xA020	R/W	I²C0 SDA Hold Register	0x7FFF
IC1DR	0xA100	R/W	I²C1 Data Register	0xFF
IC1SR	0xA108	R, R/W	I²C1 Status Register	0x00
IC1SAR	0xA10C	R/W	I²C1 Slave Address Register	0x00
IC1CR	0xA114	R/W	I²C1 Control Register	0x00
IC1SCLL	0xA118	R/W	I²C1 SCL LOW duration Register	0xFFFF
IC1SCLH	0xA11C	R/W	I²C1 SCL HIGH duration Register	0xFFFF
IC1SDH	0xA120	R/W	I²C1 SDA Hold Register	0x7FFF

ICnDR I²C Data Register

ICnDR is an 8-bit read/write register. It contains a byte of serial data to be transmitted or a byte which has just been received.

IC0DR=0x4000_A000, IC1DR=0x4000_A100,

7	6	5	4	3	2	1	0
DR							
0xFF							
RW							

7 0	ICDR	The most recently received data or data to be transmitted.

ICnSR I²C Status Register

ICnSR is an 8-bit read/write register. It contains the status of the I²C bus interface. Writing to the register clears the status bits.

IC0SR=0x4000_A008, IC1SR=0x4000_A008

7	6	5	4	3	2	1	0
GCALL	TEND	STOP	SSEL	MLOST	BUSY	TMOD	RXACK
0	0	0	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW

7	GCALL	This bit has different meaning depending on whether I²C is master or slave When I²C is a master, this bit represents whether it received AACK(Address ACK) from slave. When I²C is slave, this bit is used to indicate general call.
		0 No AACK is received (master mode)
		1 AACK is received (master mode).
		0 General call is not detected (slave mode)
		1 General call is detected (slave mode)
6	TEND	1 Byte transmission complete flag
		0 The transmission is working or not completed.
		1 The transmission is completed.
5	STOP	STOP flag
		0 STOP is not detected.
		1 STOP is detected.
4	SSEL	Slave flag
		0 Slave is not selected.
		1 Slave is selected.
3	MLOST	Mastership lost flag
		0 Mastership is not lost.
		1 Mastership is lost.
2	BUSY	BUSY flag
		0 I²C bus is in IDLE state.
		1 I²C bus is busy.
1	TMOD	Transmitter/Receiver mode flag
		0 Receiver mode.
		1 Transmitter mode.
0	RXACK	Rx ACK flag
		0 Rx ACK is not received.
		1 Rx ACK is received.

When an I²C interrupt occurs, except for the STOP interrupt, the SCL line is held LOW. To release SCL, write an arbitrary value to ICnSR. When ICnSR is written, the TEND, STOP, SSEL, MLOST, and RXACK bits are cleared.

ICnSAR I²C Slave Address Register

ICnSAR is an 8-bit read/write register. It shows the address in slave mode.

IC0SAR=0x4000_A00C, IC1SAR=0x4000_A10C

7	6	5	4	3	2	1	0
SVAD							GCEN
0x00							0
RW							RW

7 1	SVAD	7-bit Slave Address
0	GCEN	General call enable bit
		0 General call is disabled.
		1 General call is enabled.

ICnCR I²C Control Register

ICnCR is an 8-bit read/write register. This register can be set to configure I²C operation mode and simultaneously allows for I²C transactions to be kicked off.

IC0CR=0x4000_A014, IC1CR=0x4000_A114

7	6	5	4	3	2	1	0
IIF		SOFTRST	INTEN	ACKEN		STOP	START
0	0	0	0	0	0	0	0
RW		RW	RW	RW		RW	RW

7	IIF	Interrupt flag bit
		0 No interrupt is generated or interrupt is cleared
		1 Interrupt is generated
5	SOFTRST	Soft Reset enable bit.
		0 Soft Reset is disabled.
		1 Soft Reset is enabled..
4	INTEN	Interrupt enabled bit.
		0 Interrupt is disabled.
		1 Interrupt is enabled.
3	ACKEN	ACK enable bit in Receiver mode.
		0 ACK is not sent after receiving data.
		1 ACK is sent after receiving data.
1	STOP	Stop enable bit. When this bit is set as "1" in transmitter mode, next transmission will be stopped even though ACK signal has been received.
		0 Stop is disabled.
		1 Stop is enabled. When this bit is set, transmission will be stopped.
0	START	Transmission start bit in master mode.
		0 Waits in slave mode.
		1 Starts transmission in master mode.

ICnSCLL I²C SCL LOW Duration Register

ICnSCLL is a 16-bit read/write register. SCL LOW time can be set by writing this register in master mode.

IC0SDLL=0x4000_A018, IC1SDLL=0x4000_A118

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCLL															
0xFFFF															
RW															

15	SCLL	SCL LOW duration value. SCLL = (PCLK * SCLL[15:0]) + 2*PCLKs
0		Default value is 0xFFFF.

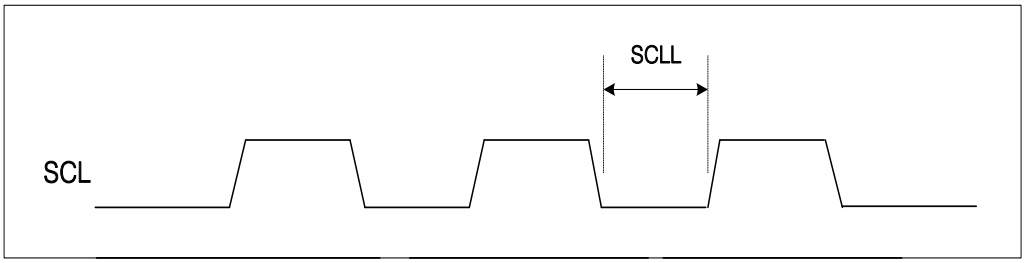


Figure 14.2.SCL LOW Timing

ICnSCLH I²C SCL HIGH duration Register

ICnSCLH is a 16-bit read/write register. SCL HIGH time will be set by writing this register in master mode.

IC0SDLH=0x4000_A01C, IC1SDLH=0x4000_A11C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCLH															
0xFFFF															
RW															

15	SCLH	SCL HIGH duration value. SCLH = (PCLK * SCLH[15:0]) + 3 PCLKs
0		Default value is 0xFFFF.

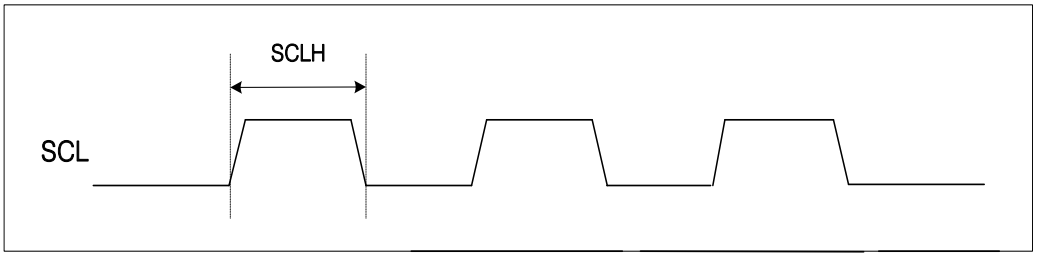


Figure 14.3.SCL HIGH Timing

ICnSDH SDA Hold Register

ICnSDH is a 15-bit read/write register. SDA HOLD time will be set by writing this register in master mode.

IC0SDH=0x4000_A020, IC1SDH=0x4000_A120

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SDH														
0x7FFF															
RW															

14	SDH	SDA HOLD time setting value. SDH = (PCLK * SDH[14:0]) + 4 PCLKs
0		Default value is 0x7FFF.

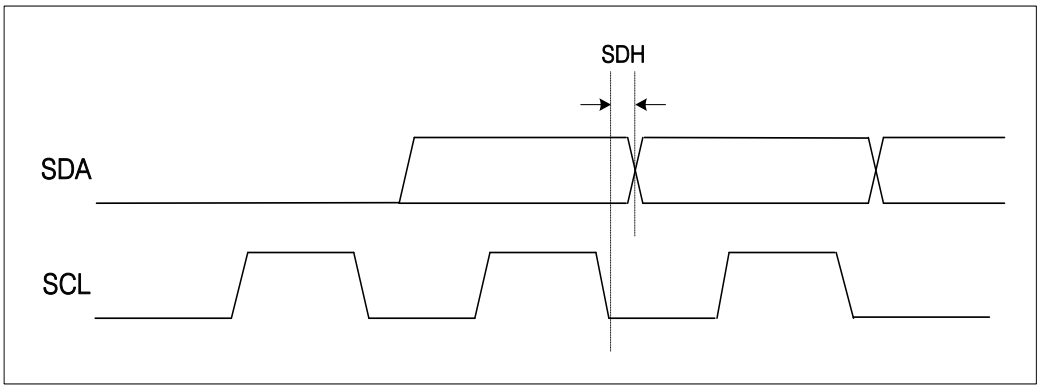


Figure 14.4.SDA HOLD Timing

Functional Description

I²C Bit Transfer

The data on the SDA line must be stable during the “H” period of the clock. The “H” or “L” state of the data line can only change when the clock signal on the SCL line is “L” as shown in Figure 14.5.

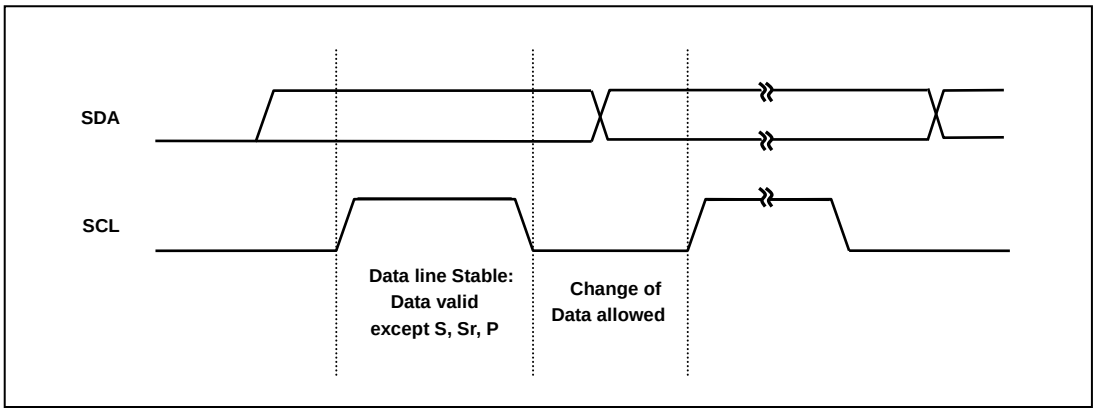


Figure 14.5. I²C Bus Bit Transfer

START/Repeated START/STOP

Within the procedure of the I^2C-bus, unique situations arise which are defined as START(S) and STOP(P) conditions; see Figure 14.6.

An “H” to “L” transition on the SDA line while SCL is “H” is one such unique case. This situation indicates a START condition. An “L” to “H” transition on the SDA line while SCL is “H” defines a STOP condition.

START and STOP conditions are always generated by the master. The bus is considered to be busy after the START condition. The bus is considered to be free again a certain time after the STOP condition.

The bus is busy if a repeated START(Sr) is generated instead of a STOP condition. In this respect, the START(S) and repeated START(Sr) conditions are functionally identical. Therefore, for the remainder of this document, the S symbol will be used as a generic term to represent both the START and repeated START conditions, unless Sr is particularly relevant.

Detection of START and STOP conditions by devices connected to the bus is easy if they incorporate the necessary interfacing hardware. However, microcontrollers with no such interface have to sample the SDA line at least twice per clock period to sense the transition.

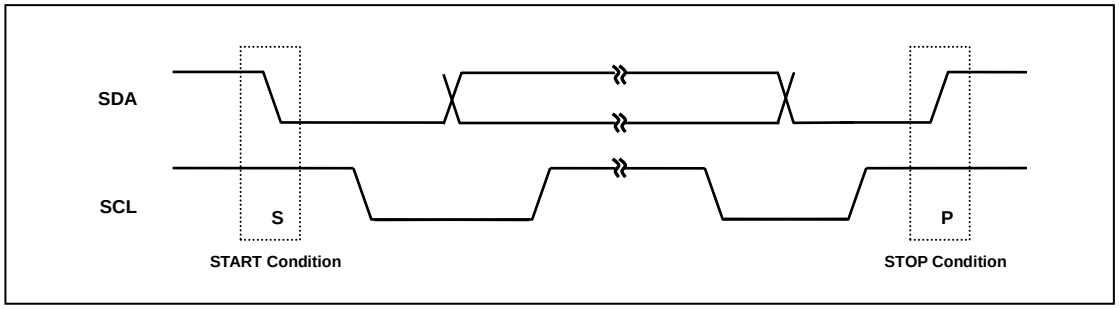


Figure 14.6. START and STOP Condition

Data Transfer

Every byte put on the SDA line must be 8-bit long. The number of bytes that can be transmitted per transfer is unrestricted. Each byte has to be followed by an acknowledge bit. Data is transferred with the most significant bit (MSB) first; see Figure 14.7. If a slave can't receive or transmit another complete byte of data until it has performed some other function, for example servicing an internal interrupt, it can hold the clock line SCL "L" to force the master into a wait state. Data transfer then continues when the slave is ready for another byte of data and releases clock line SCL.

A message which starts with such an address can be terminated by generation of a STOP condition, even during the transmission of a byte. In this case, no acknowledge is generated.

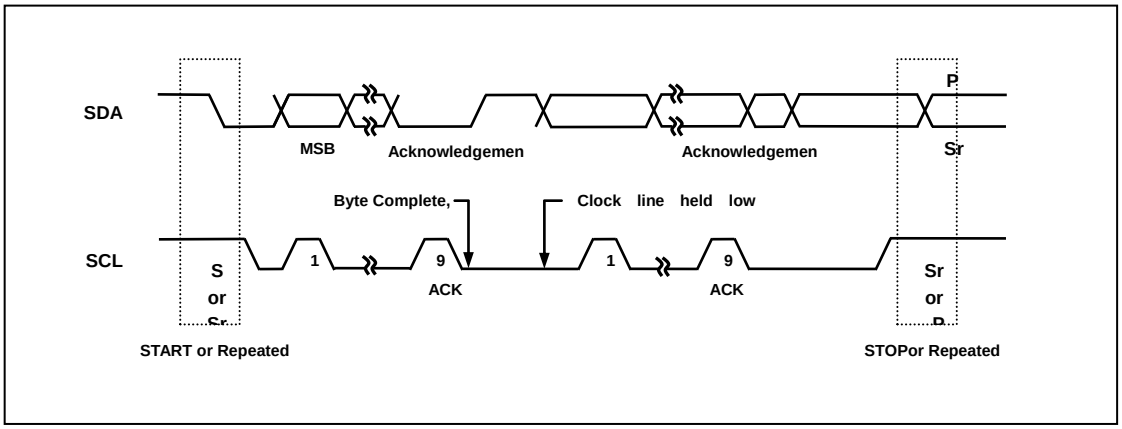


Figure 14.7. I²C Bus Data Transfer

Acknowledge

Data transfer with acknowledge is obligatory. The acknowledge-related clock pulse is generated by the master. The transmitter releases the SDA line (HIGH) during the acknowledge clock pulse.

The receiver must pull down the SDA line during the acknowledge clock pulse so that it remains stable “L” during the “H” period of this clock pulse as shown in Figure 14.8. Set-up and hold times must also be taken into account.

When a slave doesn’t acknowledge the slave address (for example, it is unable to receive or transmit because it is performing a real-time function), the data line must be left “H” by the slave. The master can then generate either a STOP condition to abort the transfer, or a repeated START condition to start a new transfer.

If a slave-receiver does acknowledge the slave address but, sometime later in the transfer cannot receive any more data bytes, the master must again abort the transfer. This is indicated by the slave generating the not-acknowledge on the first byte to follow. The slave leaves the data line “H” and the master generates a STOP or a repeated START condition.

If a master-receiver is involved in a transfer, it must signal the end of data to the slave-transmitter by not generating an acknowledge signal on the last byte that was clocked out of the slave. The slave-transmitter must release the data line to allow the master to generate a STOP or repeated START condition.

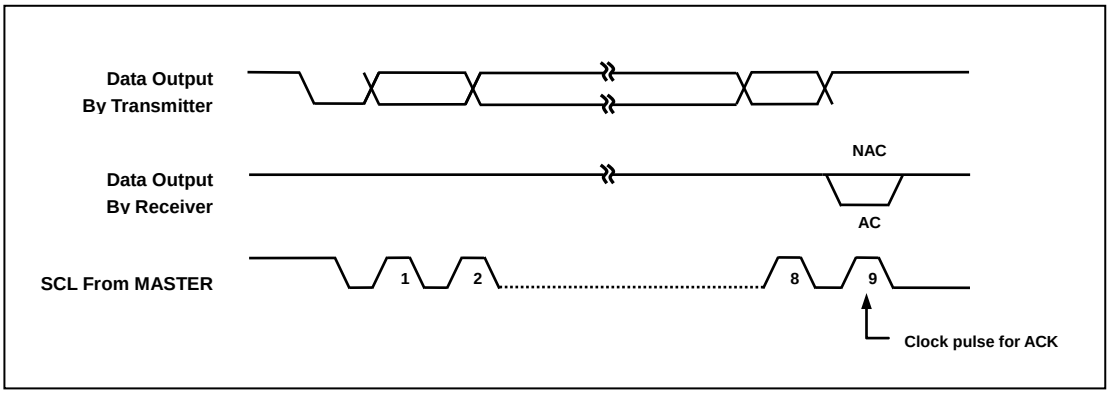


Figure 14.8. I²C Bus Acknowledge

Synchronization

All masters generate their own clock on the SCL line to transfer messages on the I^2C-bus. Data is only valid during the “H” period of the clock. A defined clock is therefore required for the bit-by-bit arbitration procedure to take place.

Clock synchronization is performed using the wires AND connection of I^2C interfaces to the SCL line. This means that an “H” to “L” transition on the SCL line will cause the devices concerned to start counting off their “L” period and, once a device clock has gone to “L” period, it will hold the SCL line in that state until the clock “H” state is reached as shown in Figure 14.9. However, the “L” to “H” transition of this clock may not change the state of the SCL line if another clock is still within its “L” period by the device with the longest “L” period. Devices with shorter “L” periods enter an “H” wait-state during this time.

When all devices concerned have counted off their “L” period, the clock line will be released and go to “H” state. At this point, there will be no difference between the device clocks and the state of the SCL line, and the devices will start counting their “H” periods. The first device to complete its “H” period will again pull the SCL line “L”.

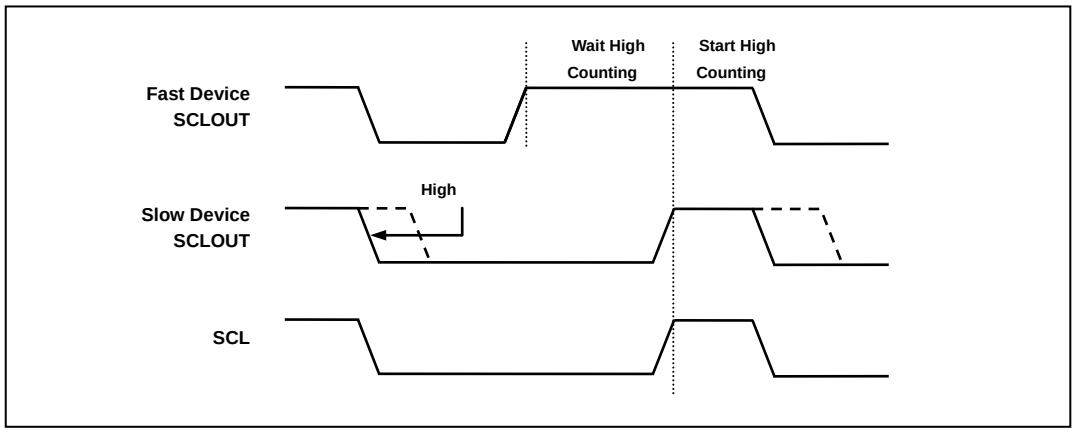


Figure 14.9. Clock Synchronization during the Arbitration Procedure

Arbitration

A master may start a transfer only if the bus is free. Two or more masters may generate a START condition within the minimum hold time of the START condition which results in a defined START condition to the bus.

Arbitration takes place on the SDA line, while the SCL line is at the “H” level, in such a way that the master which transmits “H” level, while another master is transmitting a “L” level, will switch off its DATA output stage because the level on the bus doesn’t correspond to its own level.

Arbitration can continue for many bits. The first stage of arbitration is comparison of the address bits. If the masters are each trying to address the same device, arbitration continues with comparison of the data-bits if they are master-transmitter or acknowledge-bits if they are master-receiver. Because address and data information on the I^2C-bus is determined by the winning master, no information is lost during the arbitration process.

A master that loses the arbitration can generate clock pulses until the end of the byte in which it loses the arbitration.

If a master also incorporates a slave function and it loses arbitration during the addressing stage, it’s possible that the winning master is trying to address it. The losing master must therefore switch over immediately to its slave mode.

Figure 14.10 shows the arbitration procedure for two masters. More masters may be involved, depending on the number of masters connected to the bus. As soon as there is a difference between the internal data level of the master generating Device1 Dataout and the actual level on the SDA line, its data output is switched off, which means that an “H” output level is then connected to the bus. This will not affect the data transfer

initiated by the winning master.

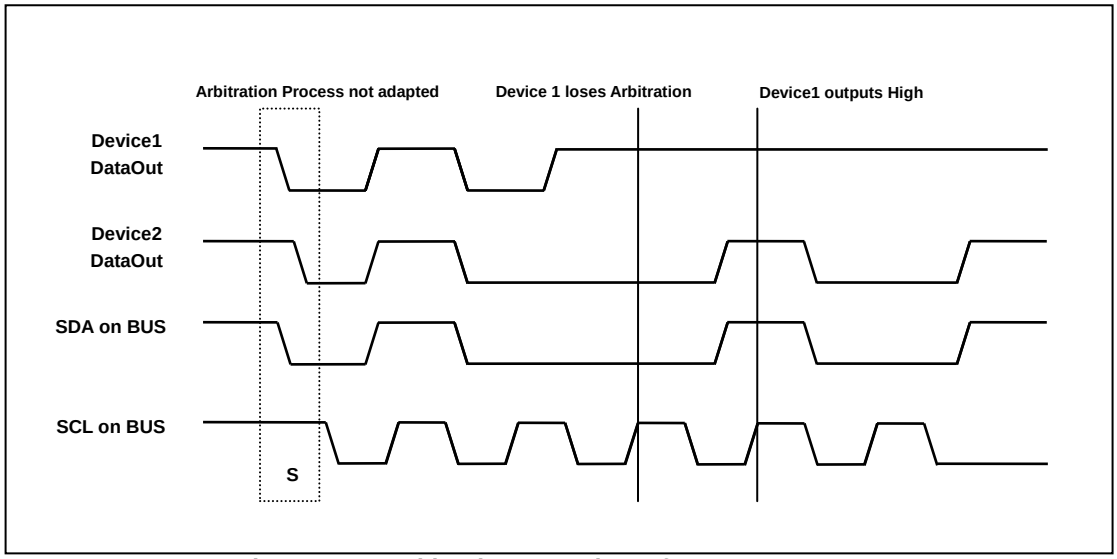


Figure 14.10. Arbitration Procedure of Two Masters

I²C Operation

I²C supports the interrupt operation. Once an interrupt is serviced, the IIF (ICnCR[7]) flag is set. ICnSR shows I²C-bus status information and the SCL line stays “L” before the register is written as a certain value. The status register can be cleared by writing a zero.

Master Transmitter

The master transmitter shows the flow of the transmitter in Master Mode as shown in Figure 14.11.

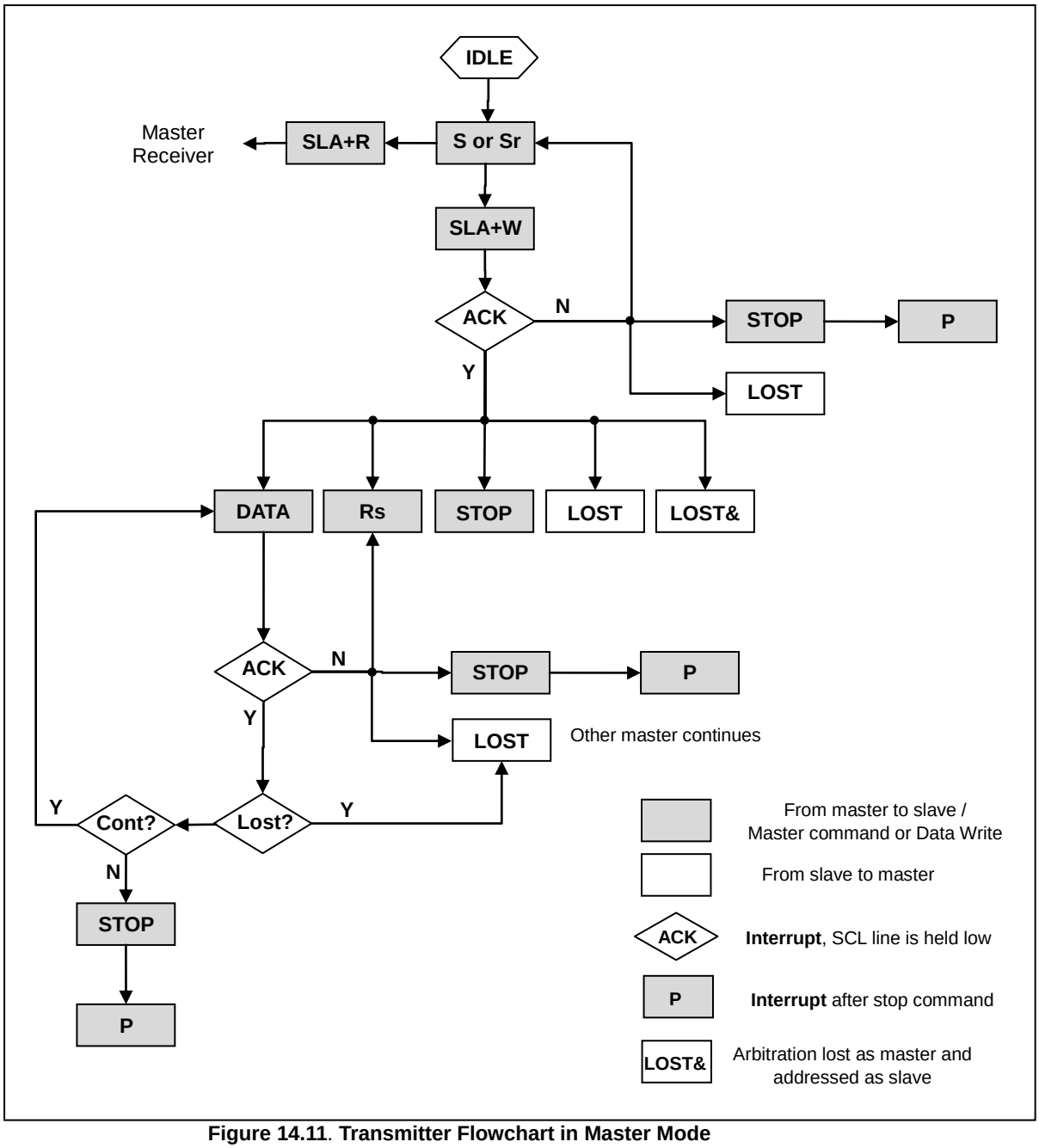


Figure 14.11. Transmitter Flowchart in Master Mode

Master Receiver

The master receiver shows the flow of the receiver in Master Mode as shown in Figure 14.12.

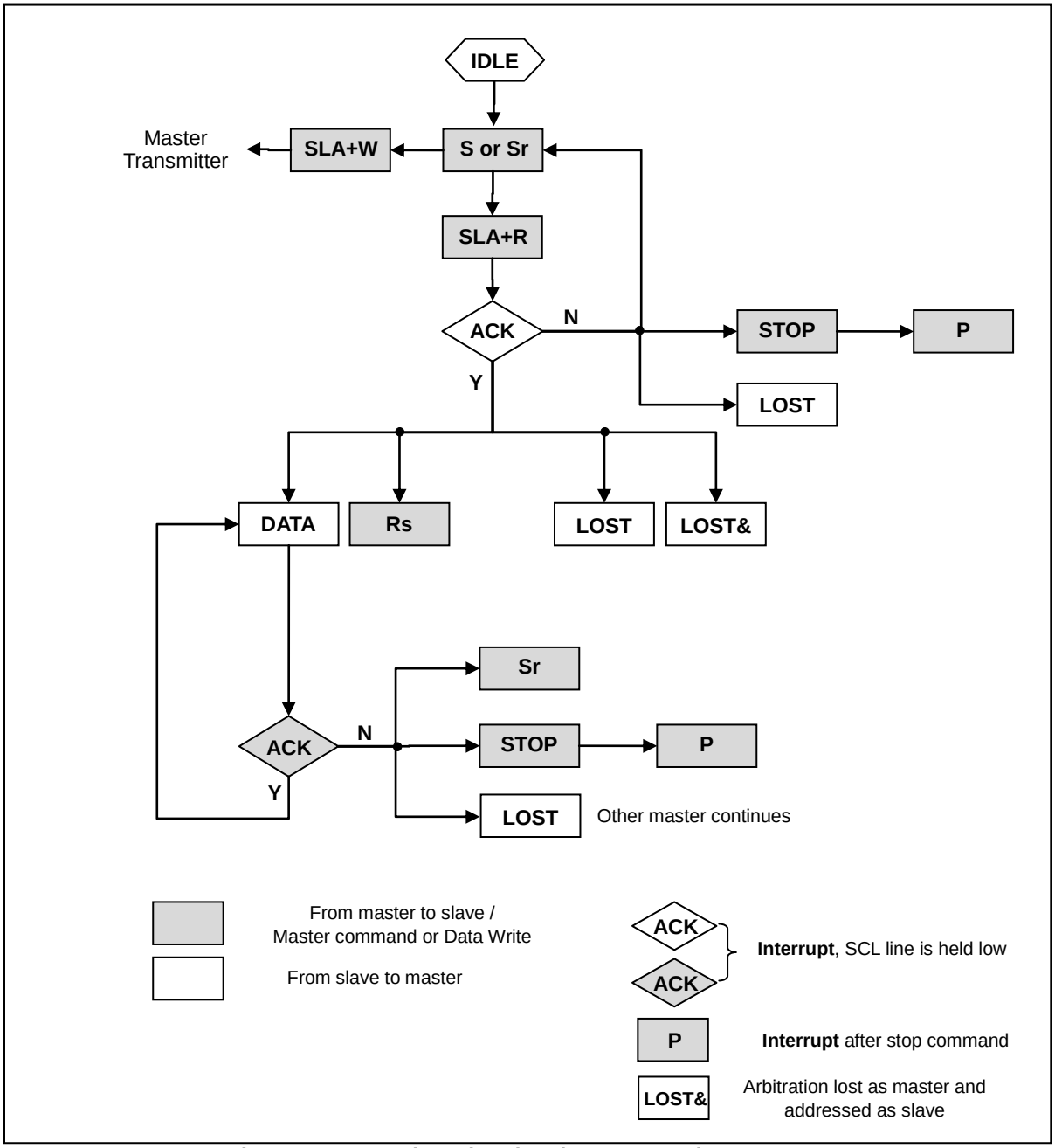


Figure 14.12. Receiver Flowchart in Master Mode

Slave Transmitter

The slave transmitter shows the flow of the transmitter in Slave Mode as shown in Figure 14.13.

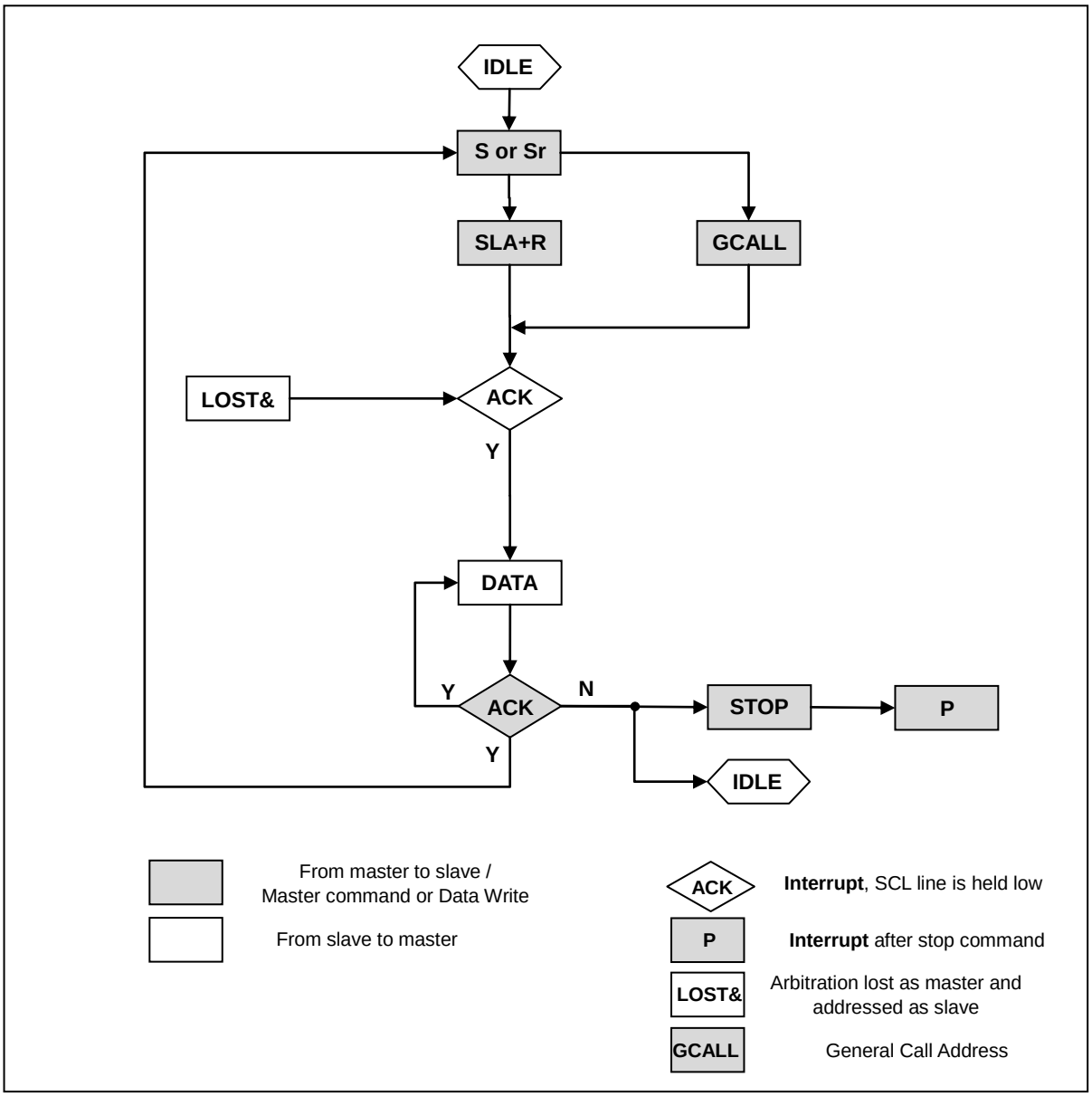


Figure 14.13. Transmitter Flowchart in Slave Mode

Slave Receiver

The slave receiver shows the flow of the receiver in Slave Mode as shown in Figure 14.14.

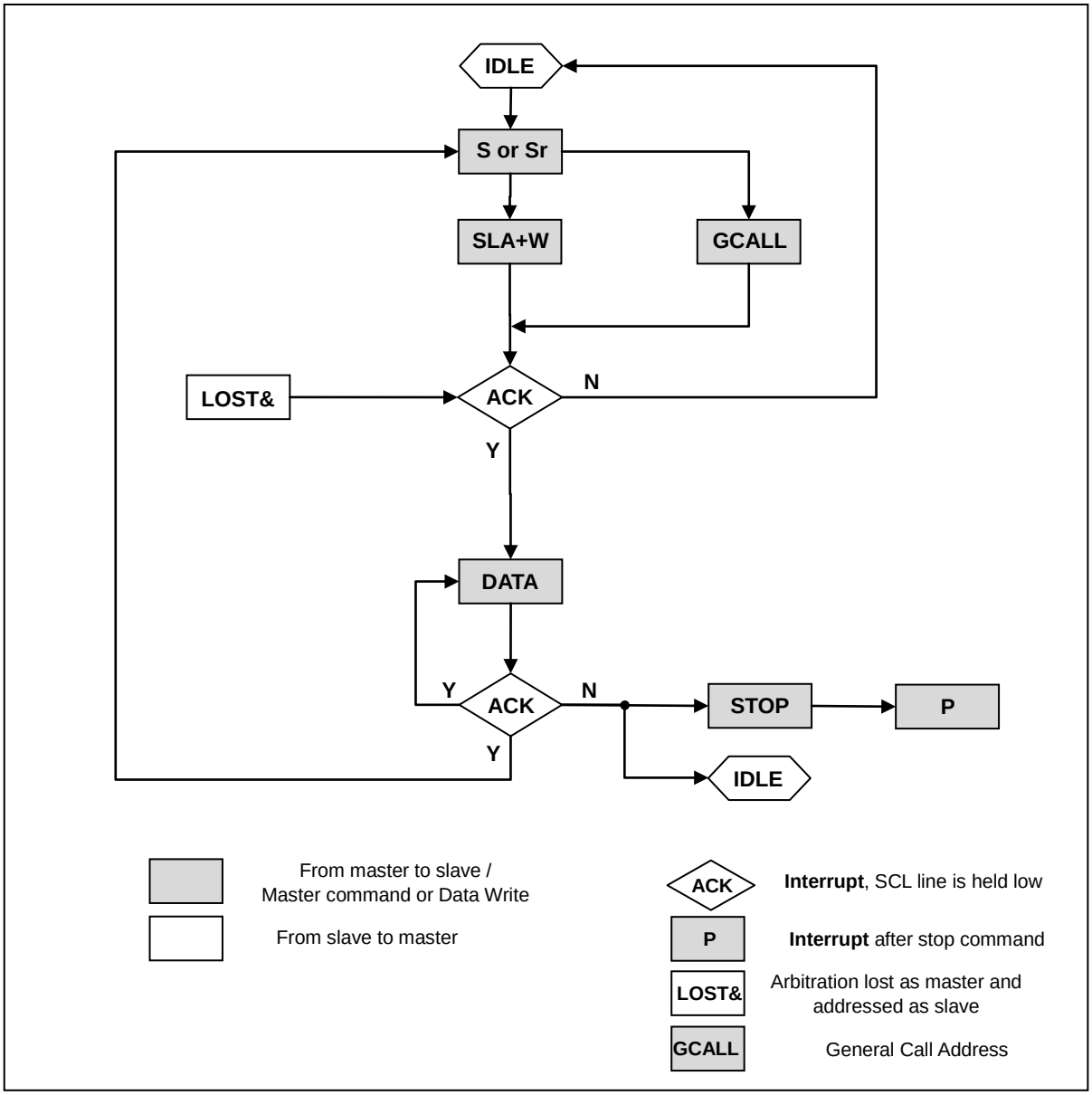


Figure 14.14. Receiver Flowchart in Slave Mode

15. Motor Pulse-Width-Modulator

Introduction

The Motor Pulse Width Modulator (MPWM) is a programmable motor controller. Features include:

- 6-channel output for motor control
- Dead- time zone support
- Protection event and over voltage event handling
- Six ADC trigger outputs
- Interval interrupt mode (period interrupt only)
- Up-down count mode

Figure 15.1 shows the MPWM block diagram.

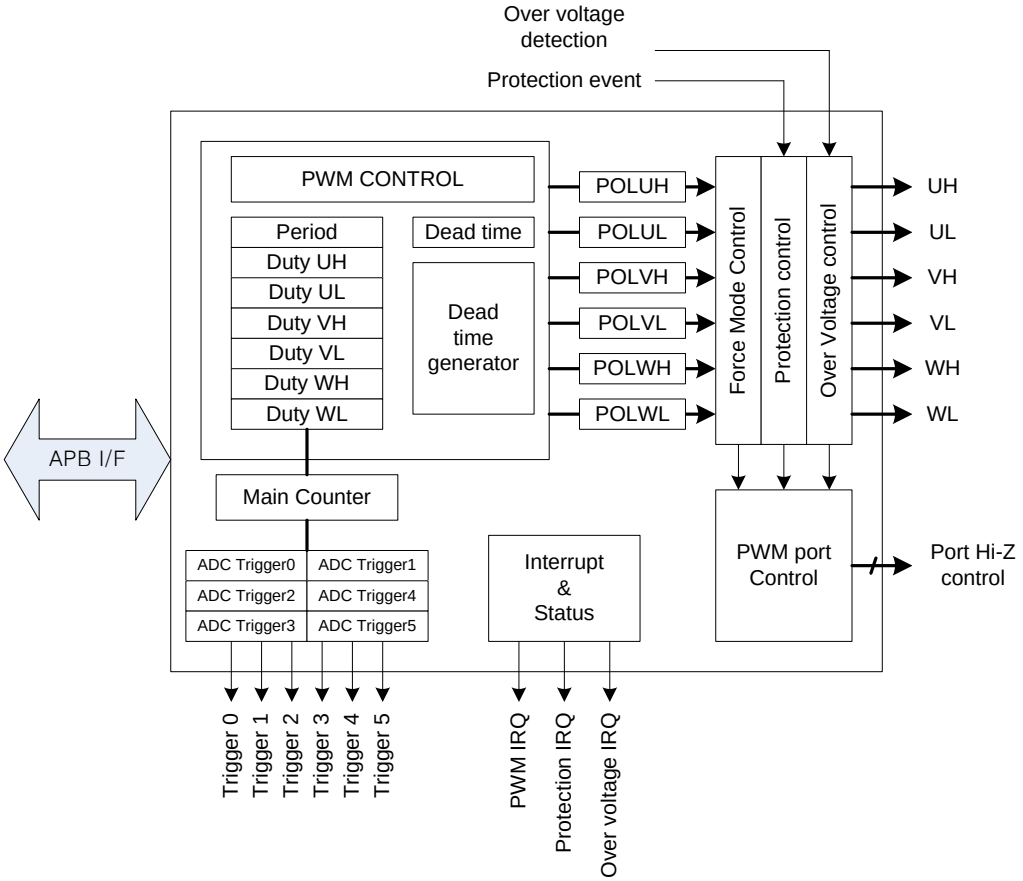


Figure 15.1. MPWM Block Diagram

Pin Description

Table 15.1. External Signals

PIN NAME	TYPE	DESCRIPTION
MP0UH	O	MPWM 0 Phase-U H-side output
MP0UL	O	MPWM 0 Phase-U L-side output
MP0VH	O	MPWM 0 Phase-V H-side output
MP0VL	O	MPWM 0 Phase-V L-side output
MP0WH	O	MPWM 0 Phase-W H-side output
MP0WL	O	MPWM 0 Phase-W L-side output
MP1UH	O	MPWM 1 Phase-U H-side output
MP1UL	O	MPWM 1 Phase-U L-side output
MP1VH	O	MPWM 1 Phase-V H-side output
MP1VL	O	MPWM 1 Phase-V L-side output
MP1WH	O	MPWM 1 Phase-W H-side output
MP1WL	O	MPWM 1 Phase-W L-side output
PRTIN0	I	MPWM 0 Protection Input 0
OVIN0	I	MPWM 0 Over-voltage Input 1
PRTIN1	I	MPWM 1 Protection Input 0
OVIN1	I	MPWM 1 Over-voltage Input 1

Registers

The base address of MPWM is shown in Table 15.2.

Table 15.2. MPWM Base Address

	BASE ADDRESS
MPWM0	0x4000_4000
MPWM1	0x4000_5000

Table 15.3 lists the register memory map.

Table 15.3. MPWM Register Map

Name	Offset	R/W	Description	Reset
MPnMR	0x0000	R/W	PWM Mode register	0x0000_0000
MPnPMR	0x0004	R/W	PWM Port Mode register	0x0000_0000
MPnOCR	0x0008	R/W	PWM Output control	0x0000_0000
MPnPRD	0x000C	R/W	PWM Period register	0x0000_0002
MPnDUH	0x0010	R/W	PWM Duty UH register	0x0000_0001
MPnDVH	0x0014	R/W	PWM Duty VH register	0x0000_0001
MPnDWH	0x0018	R/W	PWM Duty WH register	0x0000_0001
MPnDUL	0x001C	R/W	PWM Duty UL register	0x0000_0001
MPnDVL	0x0020	R/W	PWM Duty VL register	0x0000_0001
MPnDWL	0x0024	R/W	PWM Duty WL register	0x0000_0001
MPnCR1	0x0028	R/W	PWM Control	0x0000_0000
MPnCR2	0x002C	R/W	PWM Start	0x0000_0000
MPnSR	0x0030	R	PWM Status	0x0000_0000
MPnIER	0x0034	R/W	PWM Interrupt Enable	0x0000_0000
MPnCNT	0x0038	R	PWM counter register	0x0000_0001
MPnDTR	0x003C	R/W	PWM dead time control	0x0000_0000
MPnPCR	0x0040	R/W	PWM protection control register	0x0000_0000
MPnPSR	0x0044	R/W	PWM protection status	0x0000_0080
MPnOVCR	0x0048	R/W	PWM over voltage control	0x0000_0000
MPnOVSR	0x004C	R/W	PWM over voltage status	0x0000_0000
MPnATCR	0x0054	R/W	PWM ADC Trigger control	0x0000_0000
MPnATR1	0x0058	R/W	PWM ADC Trigger reg1	0x0000_0000
MPnATR2	0x005C	R/W	PWM ADC Trigger reg2	0x0000_0000
MPnATR3	0x0060	R/W	PWM ADC Trigger reg3	0x0000_0000
MPnATR4	0x0064	R/W	PWM ADC Trigger reg4	0x0000_0000
MPnATR5	0x0068	R/W	PWM ADC Trigger reg5	0x0000_0000
MPnATR6	0x006C	R/W	PWM ADC Trigger reg6	0x0000_0000

MPnMR MPWM Mode Register

The MPWM operation mode register is a 16-bit register.

MP0MR=0x4000_4000, MP1MR=0x4000_5000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOTOR		MCHMOD				UPDATE	UALL	FORCEN		FORCM				PDUP	UPDOWN
0	0	00		0	0	0	0	0	0	00		0	0	0	0
RW		RW				RW	RW	RW		RW				RW	RW

15	**MOTOR**	0	Normal PWM mode
		1	Motor PWM mode In Motor mode initial outputs of H-ch become LOW and outputs of L-ch become High (before PWM START)
13 12	**MCHMOD**	00	Motor control channel mode 2 channels symmetric mode Duty H decides the duty value of H-ch Duty L decides the duty value of L-ch
		01	1 channel asymmetric mode Duty H decides the up-counting duty value of H-ch Duty L decides the down-counting duty value of H-ch L channel become the inversion of H channel
		10	1 channel symmetric mode Duty H decides the duty value of H-ch L channel become the inversion of H channel
		11	Not valid (same with 00)
9	**UPDATE**	0	Update all duty, period register after
		1	Update all duty, period register enable. When UPDATE set, Duty and Period V registers are updated after two PWM clocks It should be cleared before PWM start(set PSTART)
8	**UALL**	0	No effect.
		1	Duty V and Duty W register will be stored with the same value of Duty U value when Duty U is written.
7	**FORCEN**	0	Force mode disable(normal mode)
		1	user can enable and disable each channels by Output control register
5 4	**FORCM**	00	Each channel is “AND”ed with MPnOCR (when port enable is low, output becomes low)
		01	Each channel is “OR”ed with MPnOCR (when port enable is high, output becomes high)
		10	Each channel is “XOR”ed with MPnOCR (when port enable is low, output becomes low)
		11	Each channel is “AND”ed with MPnOCR but when port is disabled, output becomes high-Z
1	**PDUP**	0	Period, duty value updated at every period match (both up count mode and BTB mode)
		1	Period, duty value updated at every period match and bottom(valid in up/down count mode)
0	**UPDOWN**	0	PWM Up count mode
		1	PWM Up and Down count mode Note: See Figure 15.2 for timing and operation.

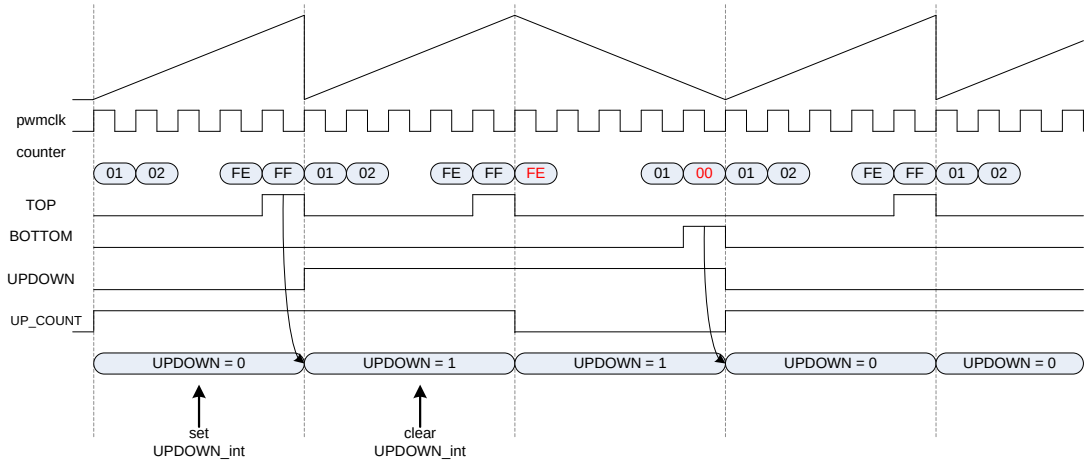


Figure 15.2. MPWM Register Mode

MPnPMR MPWM Port Mode Register

The MPWM Port Mode register is a 16-bit register.

MP0PMR=0x4000_4004, MP1PMR=0x4000_5004

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						PMOD				POLUH	POLUL	POLVH	POLVL	POLWH	POLWL
0	0	0	0	0	0	00		0	0	0	0	0	0	0	0
						RW				RW	RW	RW	RW	RW	RW

9 8	**PMOD**	00	H-ch PWM pulse out, L-ch PWM pulse out
		01	H-ch PWM pulse out , L-ch out High-Z
		10	H-ch out High-Z, L-ch PWM pulse out
		11	H-ch out High-Z, L-ch out High-Z
5 3 1	**POLxH**	0	Normal polarity for UH/VH/WH pins ('H' during duty period in normal mode, 'L' in motor mode. Initial output is 'H')
		1	Inversion polarity for UH/VH/WH pins ('L' during duty period in normal mode, 'H' in motor mode. Initial output is 'L')
4 2 0	**POLxL**	0	Normal polarity for UL/VL/WL pins ('H' during duty period in normal mode, 'L' in motor mode. Initial output is 'L')
		1	Inversion polarity for UL/VL/WL pins ('L' during duty period in normal mode, 'H' in motor mode. Initial output is 'H')

	POL=0		POL=1	
PMODE	**UH**	**UL**	**UH**	**UL**
00	PWMUH	PWMUL	~PWMUH	~PWMUL
01	PWMUH	Hi-Z	~PWMUH	Hi-Z
10	Hi-Z	PWMUL	Hi-Z	~PWMUL
11	Hi-Z	Hi-Z	Hi-Z	Hi-Z

MPnOCR MPWM Output Control Register

The MPWM output control register is an 8-bit register.

MP0OCR=0x4000_4008, MP1OCR=0x4000_5008,

7	6	5	4	3	2	1	0
		UHVAL	ULVAL	VHVAL	VLVAL	WHVAL	WLVAL
0	0	0	0	0	0	0	0
		RW	RW	RW	RW	RW	RW

xHVAL xLVAL	Operator value for each output port in Force Mode (ports output become High/Low or High-Z by FORCM[1:0]) in MPnMR register. Depending on FORCM selection, the output values are calculated with MPnOCR and current MPWM outputs.

MPnPRD MPWM Period Register

The MPWM Period Register is a 16-bit register.

MP0PRD=0x4000400C, MP1PRD=0x40000500C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PERIOD															
0x0002															
RW															

15:0	PERIOD	16-bit PWM period. It should be larger than 0x0010 (if Duty is 0x0000, PWM will not work)

MPnDUH MPWM Duty UH Register

The MPWM U channel duty register is a 16-bit register.

MP0DUH=0x4000_4010, MP1DUH=0x4000_5010

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUH															
0x0001															
RW															

15:0	DUTY UH [15:0]	16-bit PWM Duty for UH output. It should be larger than 0x0001 (if Duty is 0x0000, PWM will not work)

MPnDVH MPWM Duty VH Register

The MPWM V channel duty register is a 16-bit register.

MP0DVH=0x4000_4014, MP1DVH=0x4000_5014

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DVH															
0x0001															
RW															

15:0	DUTY VH	16-bit PWM Duty for VH output. It should be larger than 0x0001 (if Duty is 0x0000, PWM will not work)

MPnDWH MPWM Duty WH Register

The MPWM W channel duty register is a 16-bit register.

MP0DWH=0x4000_4018, MP1DWH=0x4000_5018

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DWH															
0x0001															
RW															

15:0	DUTY WH	16-bit PWM Duty for WH output. It should be larger than 0x0001 (if Duty is 0x0000, PWM will not work)

MPnDUL MPWM Duty UL Register

The MPWM U channel duty register is a 16-bit register.

MP0DUL=0x4000_401C, MP1DUL=0x4000_501C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUL															
0x0001															
RW															

15:0	DUTY UL	16-bit PWM Duty for UL output. It should be larger than 0x0001 (if Duty is 0x0000, PWM will not work)

MPnDVL MPWM Duty VL Register

The MPWM V channel duty register is a 16-bit register.

MP0DVL=0x4000_4020, MP1DVL=0x4000_5020

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DVL															
0x0001															
RW															

15:0	DUTY VL	16-bit PWM Duty for VL output. It should be larger than 0x0001 (if Duty is 0x0000, PWM will not work)

MPnDWL MPWM Duty WL Register

The PWM W channel duty register is a 16-bit register.

MP0DWL=0x4000_4024, MP1DWL=0x4000_5024

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DWL															
0x0001															
RW															

15:0	DUTY WL [15:0]	16-bit PWM Duty for WL output. It should be larger than 0x0001 (if Duty is 0x0000, PWM will not work)

MPnCR1 MPWM Control Register 1

The MPWM Control Register 1 is a 16-bit register.

MP0CR1=0x4000_4028, MP1CR1=0x4000_5028

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTVEN		IRQMD			IRQN			PWMEN							HALT
0	0	0		0	0			0	0	0	0	0	0	0	0
RW		RW			RW			RW							RW

15	INTVEN		IRQ interval mode (IRQ asserts to CPU at every N-th period IRQ)
13 12	IRQMD	0	IRQ at period, duty match (UP)
		1	IRQ at bottom, duty match (DOWN) (only valid in UPDOWN mode)
		2	IRQ at every period, bottom, duty match (UP & DOWN)
10 8	IRQN[2:0]		IRQ interval number (1~8th PRDIRQ)
7	PWMEN		PWM enable
0	HALT		PWM HALT (PWM counter stop but not reset) PWM outputs keep previous state

Each interrupt source can be enabled or disabled by the MPWM interrupt enable register.

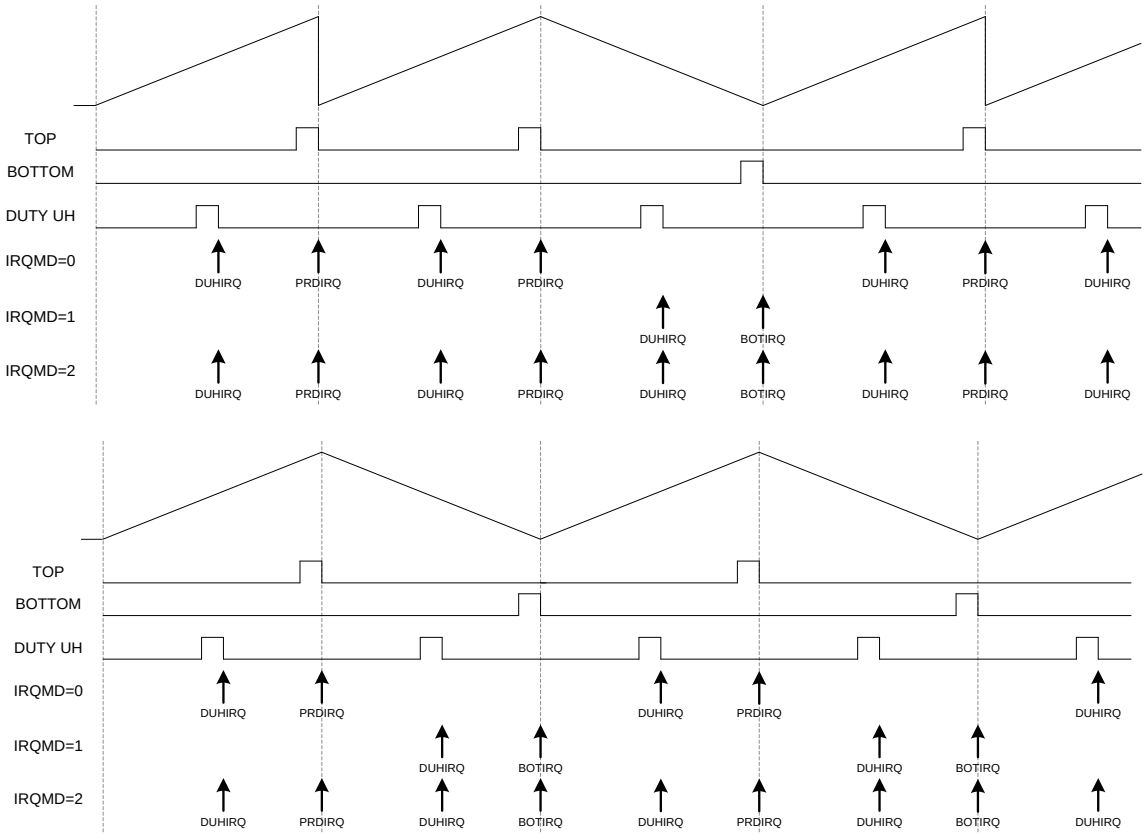


Figure 15.3. PWM-related Interrupt Sources

MPnCR2 MPWM Control Register 2

The MPWM Control Register 2 is an 8-bit register.

MP0CR2=0x4000_402C, MP1CR2=0x4000_502C,

7	6	5	4	3	2	1	0
							PSTART
0	0	0	0	0	0	0	0
							RW

0	PSTART	0	PWM counter stop and clear
		1	PWM counter start (will be re-synced @PWM clock twice) PWMEN should be “1” to start PWM counter

MPnSR MPWM Status Register

The PWM Status Register is a 16-bit register.

MP0SR=0x4000_4030, MP1CR=0x4000_5030

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DOWN	IRQCNT							PRDIRQ	BOTIRQ	DUHIRQ	DULIRQ	DVHIRQ	DVLIRQ	DWHIRQ	DWLIRQ
0	000			0	0	0	0	0	0	0	0	0	0	0	0
RW	RW							RW	RW	RW	RW	RW	RW	RW	RW

15	DOWN	0	PWM Count Up
		1	PWM Count Down (in BTB mode)
14 12	IRQCNT[2:0]		Interrupt count number of period match (Interval PRDIRQ mode)
7	PRDIRQ		PWM period interrupt flag (0: no int / 1: int occurred) Write “1” to clear flag
6	BOTIRQ		PWM bottom interrupt flag (0: no int / 1: int occurred) Write “1” to clear flag
5	DUHIRQ		PWM duty UH interrupt flag (0: no int / 1: int occurred) Write “1” to clear flag
4	DULIRQ		PWM duty UL interrupt flag (0: no int / 1: int occurred) Write “1” to clear flag
3	DVHIRQ		PWM duty VH interrupt flag (0: no int / 1: int occurred) Write “1” to clear flag
2	DVLIRQ		PWM duty VL interrupt flag (0: no int / 1: int occurred) Write “1” to clear flag
1	DWHIRQ		PWM duty UH interrupt flag (0: no int / 1: int occurred) Write “1” to clear flag *This flag will be enabled by DUHIEN bit.
0	DWLIRQ		PWM duty WL interrupt flag (0: no int / 1: int occurred) Write “1” to clear flag

MPnIER MPWM Interrupt Enable Register

The MPWM Interrupt Enable Register is an 8-bit register.

MP0IER=0x4000_4034, MP1IER=0x4000_5034,

7	6	5	4	3	2	1	0
PRDIEN	BOTIEN	DUHIEN	DULIEN	DVHIEN	DVLIEN	DWHIEN	DWLIEN
0	0	0	0	0	0	0	0
RW	RW	RW	RW	RW	RW	RW	RW

7	PRDIEN	PWM period interrupt enable 0: interrupt disable 1: interrupt enable
6	BOTIEN	PWM bottom interrupt enable 0: interrupt disable 1: interrupt enable
5	DUHIEN	PWM U Duty H match interrupt enable 0: interrupt disable 1: interrupt enable
4	DULIEN	PWM U Duty L match interrupt enable 0: interrupt disable 1: interrupt enable
3	DVHIEN	PWM V Duty H match interrupt enable 0: interrupt disable 1: interrupt enable
2	DVLIEN	PWM V Duty L match interrupt enable 0: interrupt disable 1: interrupt enable
1	DWHIEN	PWM W Duty H match interrupt enable 0: interrupt disable 1: interrupt enable
0	DWLIEN	PWM W Duty L match interrupt enable 0: interrupt disable 1: interrupt enable

MPnCNT MPWM Counter Register

The PWM Counter Register is a 16-bit read-only register.

MP0CNT=0x4000_4038, MP1CNT=0x4000_5038

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT															
0x0000															
RW															

	MPnCNT	PWM counter value read (16-bit)

MPnDTR MPWM Dead Time Register

The PWM Dead Time Register is a 16-bit register.

MP0DTR=0x4000_403C, MP1DTR=0x4000_503C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTEN							DTCLK	DT							
0	0	0	0	0	0	0	0	0x00							
RW							RW	RW							

15	DTEN	0	Dead Time disable
		1	Dead Time enable
8	DTCLK	0	Dead time counter uses PWM CLK/4
		1	Dead time counter uses PWM CLK/8
7 0	DT[7:0]	Dead Time value (Dead time setting makes output delay of 'low to high transition' in normal polarity) 0x01 ~0xFF : Dead time	

MPnPCR MPWM Protection Control Register

The PWM Protection Control Register is a 32-bit register.

MP0PCR=0x4000_4040, MP1PCR=0x4000_5040

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								PRTIN	C3IN	C2IN	C1IN	C0IN	AD2IN	AD1IN	AD0IN	PROTDIS		UHPROT	ULPROT	VHPROT	VLPROT	WHPROT	WLPROT	PROTCLR	PTDBC					PTSEL	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	000			0	0	00	
								RW	RW	RW	RW	RW	RW	RW	RW	RW		RW	RW	RW	RW	RW	RW	RW	RW					RW	

23	PRTIN	External PRTIN pin input (Active High)
22	C3IN	Comparator #3 output
21	C2IN	Comparator #2 output
20	C1IN	Comparator #1 output
19	C0IN	Comparator #0 output
18	AD2IN	ADC2 comparator output
17	AD1IN	ADC1 comparator output
16	AD0IN	ADC0 comparator output
15	PROTDIS	Protection mode disable (default 0, protection enable) To set PROTDIS as '1', 0xA5A5 should be written to PROTPAT[31:16]
13	UHPROT	U-phase H-side protection output ('0'=L/'1'=H)
12	ULPROT	U-phase L-side protection output ('0'=L/'1'=H)
11	VHPROT	V-phase H-side protection output ('0'=L/'1'=H)
10	VLPROT	V-phase L-side protection output ('0'=L/'1'=H)
9	WHPROT	W-phase H-side protection output ('0'=L/'1'=H)
8	WLPROT	W-phase L-side protection output ('0'=L/'1'=H)
7	PROTCLR	Protection clear (after protection mode active) To clear PROTCLR bit, 0x39AA should be written to PROTPAT[31:16]
6 4	PTDBC[2:0]	Protection signal debounce 00 – no debounce 1~7 – debounce by (fsystem * PTDBC[2:0])
3 2		reserved
1 0	PTSEL[1:0]	Protection mode select 00 – no output control 01 – no control for UH/VH/WH UL/VL/WL controlled by UL~WLPROT 10 – no control for UL/VL/WL UH/VH/WH controlled by UH~WHPROT 11 – all outputs controlled by UH~WLPROT

MPnPSR MPWM Protection Status Register

The PWM Protection Status Register is a 32-bit register.

MP0PSR=0x4000_4044, MP1PSR=0x4000_5044

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PROTPAT																								PROTEN							PROTIN
0x0000																0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
RW																								RW							RW

31 16	PROTPAT	Lock PROTPAT to set or reset Protection or Over voltage control bit
7	PROTEN	Protection mode enable status
0	PROTIN	Protection input status

MPnOVCR MPWM Over Voltage Control Register

The PWM Over Voltage Control Register is a 32-bit register.

MP0PCR=0x4000_4048, MP1PCR=0x4000_5048

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								OVIN	C3IN	C2IN	C1IN	C0IN	AD2IN	AD1IN	AD0IN	OVEN								OVCLR	OVDBC					OVSEL	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	000			0	0	00	
								RW	RW	RW	RW	RW	RW	RW	RW	RW								RW	RW					RW	

23	OVIN	External OVIN pin input
22	C3IN	Comparator #3 output
21	C2IN	Comparator #2 output
20	C1IN	Comparator #1 output
19	C0IN	Comparator #0 output
18	AD2IN	ADC2 comparator output (AD2CCR[23])
17	AD1IN	ADC1 comparator output (AD1CCR[23])
16	AD0IN	ADC0 comparator output (AD0CCR[23])
15	OVEN	Over voltage protection mode enable (default 0, over voltage protection disable) To set OVEN as '1', 0x7788 should be written to PROTPAT[31:16]
7	OVCLR	OV Protection clear (after OV protection mode active) To clear the OVCLR flag, 0x5596 should be written to PROTPAT[31:16]
6 5 4	OVDBC	Over voltage protection signal debounce 00 – no debounce 1~7 – debounce by (fsystem * PTDBC[2:0])
1 0	OVSEL	Over Voltage Protection mode select 00 – no output control 01 – High output for UH/VH/WH + POL Low output for UL/VL/WL + POL 10 – Low output for UH/VH/WH + POL High output for UL/VL/WL + POL 11 – all outputs controlled by UH~WLPROT

MPnOVSR MPWM Over-Voltage Status Register

The PWM Over Voltage Status Register is an 8-bit read-only register.

MP0OVSR=0x4000_404C, MP1OVCR=0x4000_504C,

7	6	5	4	3	2	1	0
OVSTAT							OVPIN
0	0	0	0	0	0	0	0
R							R

7	OVSTAT	Over voltage protection mode status
0	OVPIN	Over voltage protection input status

MPnATCR MPWM ADC Trigger Control Register

The PWM ADC Trigger Control Register is a 16-bit register.

MP0ATCR=0x4000_4054, MP1ATCR=0x4000_5054

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							ATRGALL	ATRGEN						ATRGM	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	00	
							RW	RW						RW	

8	ATRGALL	ADC Trigger register 0 match event makes all trigger signals	
7	ATRGEN	ADC Trigger mode enable	
1	ATRGM	00	Always ADC Trigger enable when TRGEN is high
0		01	ADC Trigger disable in protection state
		10	ADC Trigger disable in over voltage state
		11	ADC Trigger disable in protection, over voltage state

MPnATRm MPWMn ADC Trigger Counter m Register

MPnATR1 MPWM ADC Trigger Counter 1 Register
MPnATR2 MPWM ADC Trigger Counter 2 Register
MPnATR3 MPWM ADC Trigger Counter 3 Register
MPnATR4 MPWM ADC Trigger Counter 4 Register
MPnATR5 MPWM ADC Trigger Counter 5 Register
MPnATR6 MPWM ADC Trigger Counter 6 Register

The PWM ADC Trigger Counter Register is a 32-bit register.

MP0ATR1=0x4000_4058, MP1ATR1=0x4000_5058
MP0ATR2=0x4000_405C, MP1ATR2=0x4000_505C
MP0ATR3=0x4000_4060, MP1ATR3=0x4000_5060
MP0ATR4=0x4000_4064, MP1ATR4=0x4000_5064
MP0ATR5=0x4000_4068, MP1ATR5=0x4000_5068
MP0ATR6=0x4000_406C, MP1ATR6=0x4000_506C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												ATUDT		ATMOD		ATCNT																
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0x0000															
												RW		RW		RW																

19	ATUDT	0	Trigger register update mode ADC trigger value applied at period match event (at the same time with period and duty registers update)
		1	Trigger register update mode When this bit set, written Trigger register values are sent to trigger compare block after two PWM clocks (through synchronization logic)
17 16	ATMOD	00	ADC trigger Mode register ADC trigger Disable
		01	Trigger out when up count match
		10	Trigger out when down count match
		11	Trigger out when up-down count match
15 0	ATCNT	ADC Trigger counter 0 (it should be less than PWM period)	

Functional Description

The PWMx module allows users to configure the PWM for different types of modulation schemes described in the previous section. The PER2 and PCER2 registers must be configured to enable the PWMx peripheral and the PWMx peripheral clock.

Setting or resetting the MOTOR bit in the MPnMR register allows users to operate the motor in Independent or Complementary PWM modes. For more information about operating modes, refer to the diagrams in the following section.

Normal PWM Mode Timing Diagram Register

Normal PWM Mode

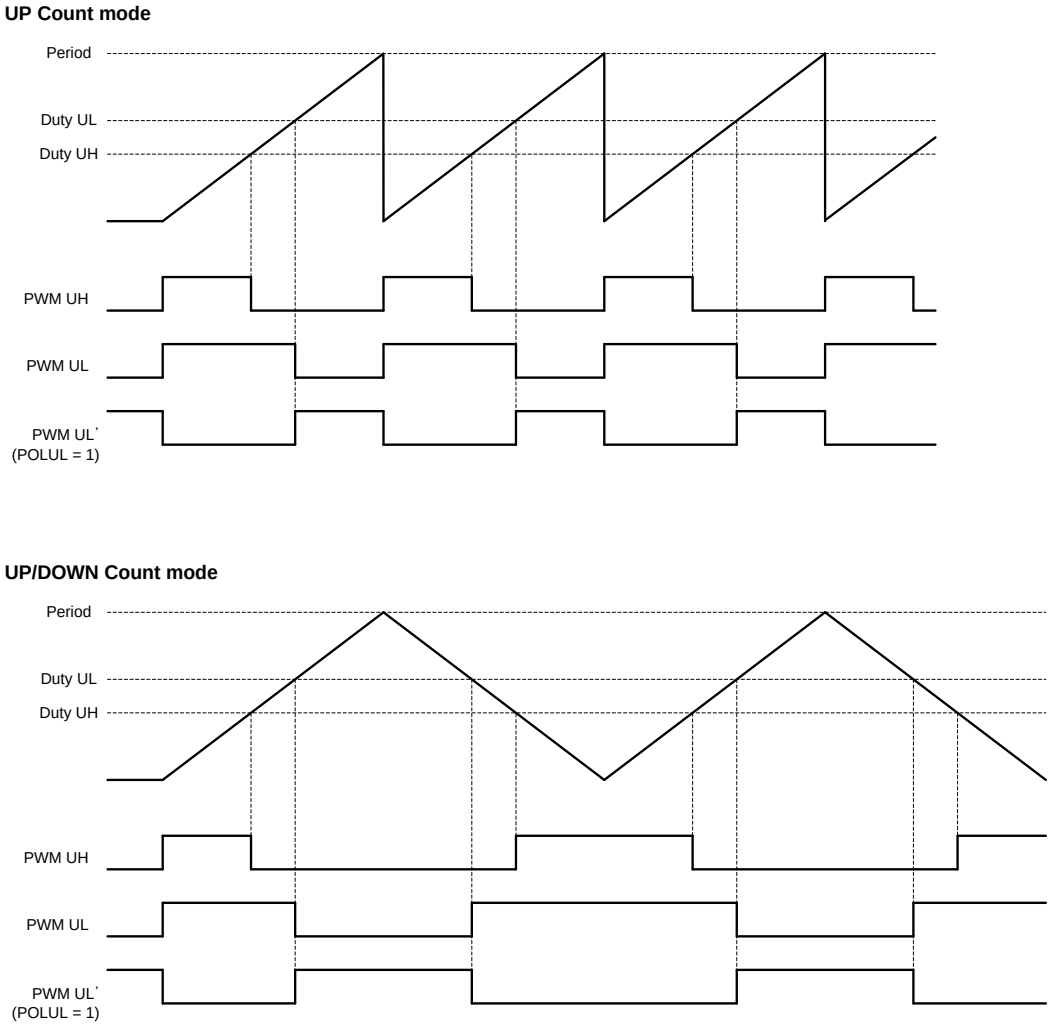


Figure 15.4. Normal PWM Mode

Motor PWM Mode Timing Diagram

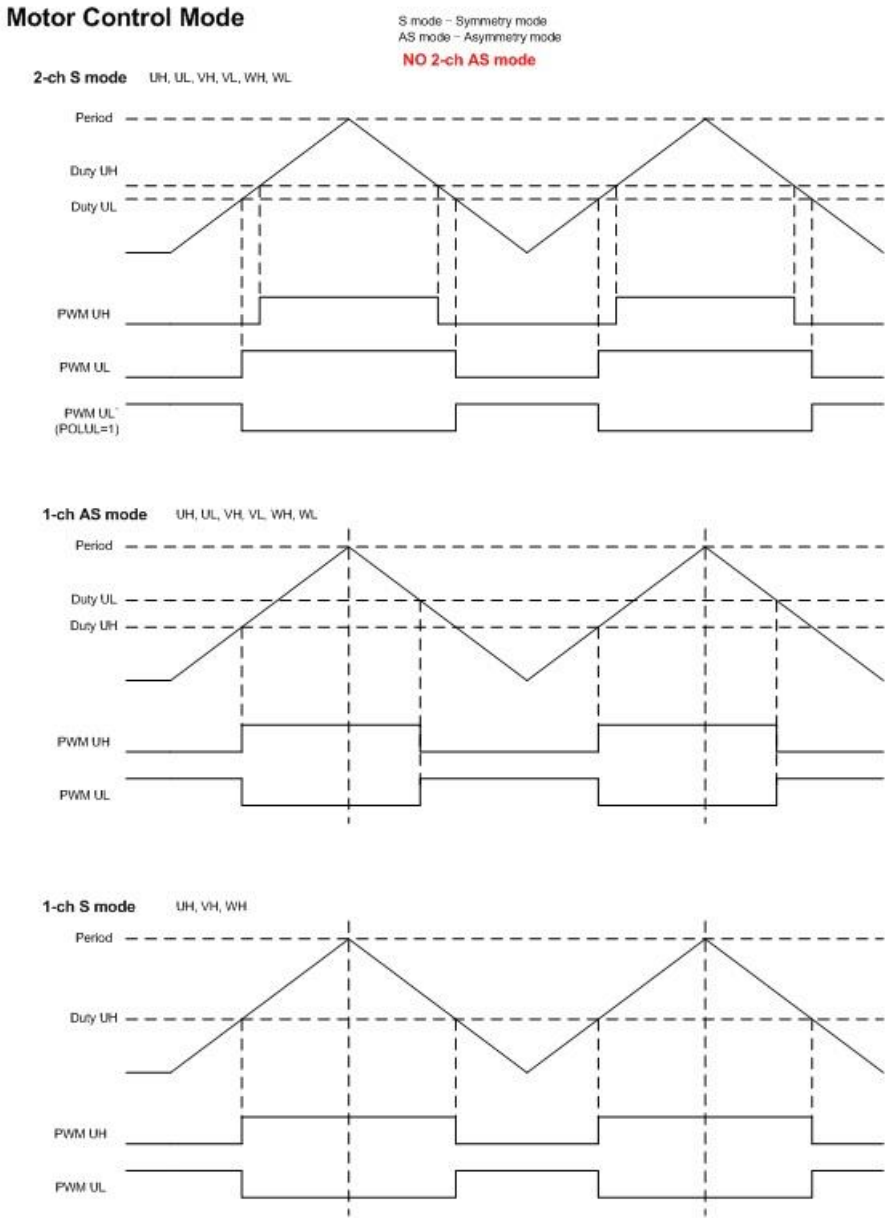


Figure 15.5. Motor PWM Mode Timing Diagram

Motor PWM Mode with Dead Time Zone

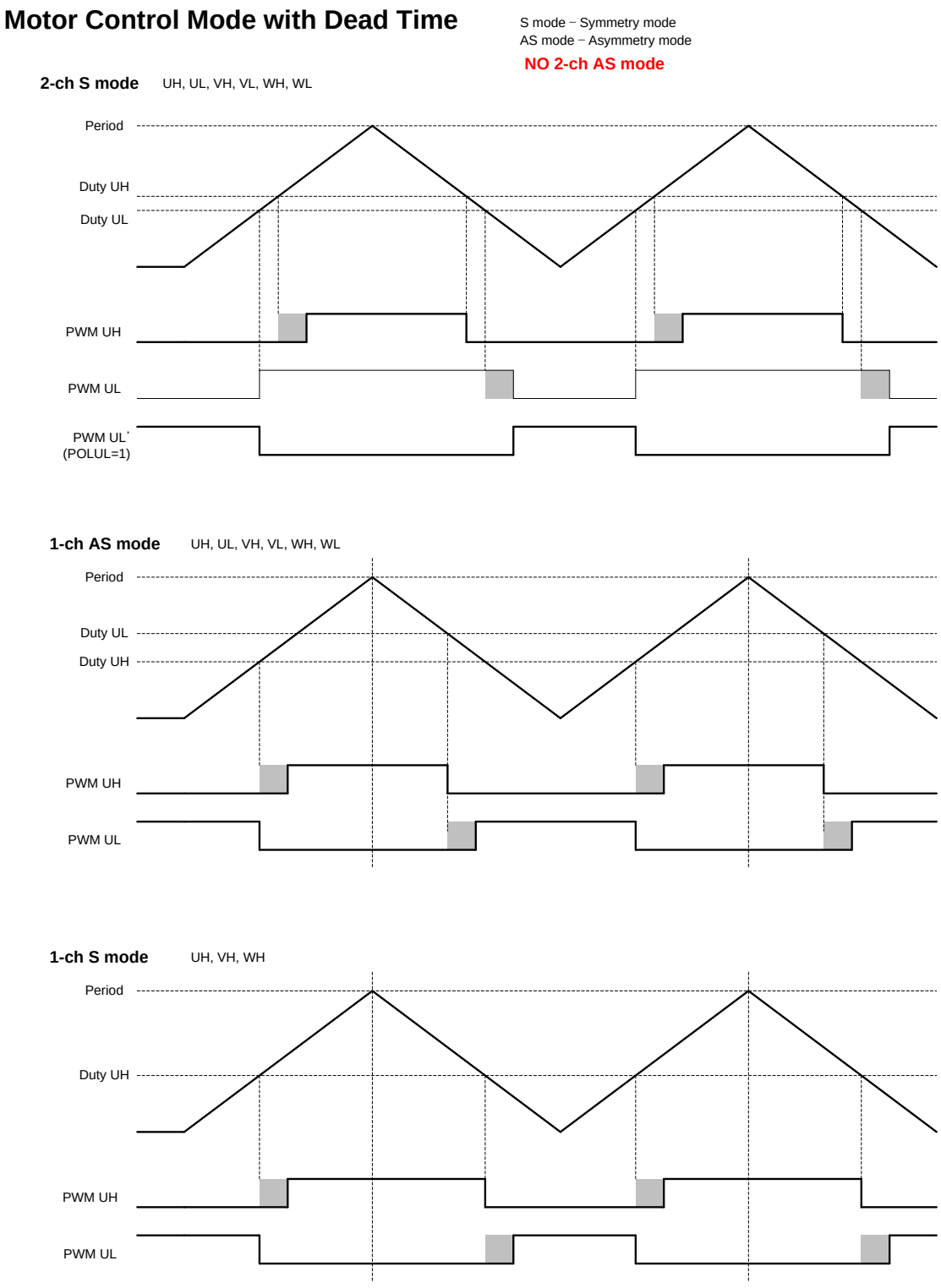


Figure 15.6. Motor PWM Mode with Dead Time Zone

PWM Output Combination Table

PWM mode : PWM out becomes high for duty duration

Motor mode : PWM out becomes low for duty duration

PWM mode		UHOUT	ULOUT	VHOUT	VLOUT	WHOUT	WLOUT
	initial	L	L	L	L	L	L
UPDOWN =0	up count	up@period	up@period	up@period	up@period	up@period	up@period
	up count	down@dutyUH	down@dutyUL	down@dutyVH	down@dutyVL	down@dutyWH	down@dutyWL
UPDOWN =1	up count	down@dutyUH	down@dutyUL	down@dutyVH	down@dutyVL	down@dutyWH	down@dutyWL
	down count	up@dutyUH	up@dutyUL	up@dutyVH	up@dutyVL	up@dutyWH	up@dutyWL
MOTOR mode		UHOUT	ULOUT	VHOUT	VLOUT	WHOUT	WLOUT
2CHS	initial	L	L	L	L	L	L
	up count	up@dutyUH	up@dutyUL	up@dutyVH	up@dutyVL	up@dutyWH	up@dutyWL
	down count	down@dutyUH	down@dutyUL	down@dutyVH	down@dutyVL	down@dutyWH	down@dutyWL
1CHAS	initial	L	~UHOUT	L	~VHOUT	L	~WHOUT
	up count	up@dutyUH	~UHOUT	up@dutyVH	~VHOUT	up@dutyWH	~WHOUT
	down count	down@dutyUL	~UHOUT	down@dutyVL	~VHOUT	down@dutyWL	~WHOUT
1CHS	initial	L	~UHOUT	L	~VHOUT	L	~WHOUT
	up count	up@dutyUH	~UHOUT	up@dutyVH	~VHOUT	up@dutyWH	~WHOUT
	down count	down@dutyUH	~UHOUT	down@dutyVH	~VHOUT	down@dutyWH	~WHOUT
POLARITY control		Polarity UH	Polarity UL	Polarity VH	Polarity VL	Polarity WH	Polarity WL

PMOD	00	UHOUT	ULOUT	VHOUT	VLOUT	WHOUT	WLOUT
	01	UHOUT	hi-Z	VHOUT	hi-Z	WHOUT	hi-Z
	10	hi-Z	ULOUT	hi-Z	VLOUT	hi-Z	WLOUT
priority = 4	11	hi-Z	hi-Z	hi-Z	hi-Z	hi-Z	hi-Z
FORCM	00	UHOUT & UHEN	ULOUT & ULEN	VHOUT & VHEN	VLOUT & VLEN	WHOUT & WHEN	WLOUT & WLEN
	01	UHOUT | UHEN	ULOUT | ULEN	VHOUT | VHEN	VLOUT | VLEN	WHOUT | WHEN	WLOUT | WLEN
	10	UHOUT ^ UHEN	ULOUT ^ ULEN	VHOUT ^ VHEN	VLOUT ^ VLEN	WHOUT ^ WHEN	WLOUT ^ WLEN
	11	UHOUT & UHEN	ULOUT & ULEN	VHOUT & VHEN	VLOUT & VLEN	WHOUT & WHEN	WLOUT & WLEN
priority = 3		if ~UHEN, hi-Z	if ~ULEN, hi-Z	if ~VHEN, hi-Z	if ~VLEN, hi-Z	if ~WHEN, hi-Z	if ~WLEN, hi-Z

PTSEL	00	UHOUT	ULOUT	VHOUT	VLOUT	WHOUT	WLOUT
PROTIN= 1	01	UHOUT	ULPROT	VHOUT	VLPROT	WHOUT	WLPROT
	10	UHPROT	ULOUT	VHPROT	VLOUT	WHPROT	WLOUT
priority = 2	11	UHPROT	ULPROT	VHPROT	VLPROT	WHPROT	WLPROT
OVSEL	00	UHOUT	ULOUT	VHOUT	VLOUT	WHOUT	WLOUT
OVPIN=1	01	high	low	high	low	high	low
	10	low	high	low	high	low	high
priority = 1	11	UHPROT	ULPROT	VHPROT	VLPROT	WHPROT	WLPROT

Figure 15.7. PWM Output Combination Table

16. 12-Bit A/D Converter

Introduction

The ADC block consists of 3 independent ADC units that provide:

- 16 channels of analog inputs
- Single and Continuous conversion mode
- Up to 8 times burst conversion support
- External pin trigger support
- 4 internal trigger sources support (PWMs, timers)
- Adjustable sample & hold time
- ADC clock can be derived from all sources available with configurable dividers

Figure 16.1 shows a block diagram of the 12-bit A/D Converter.

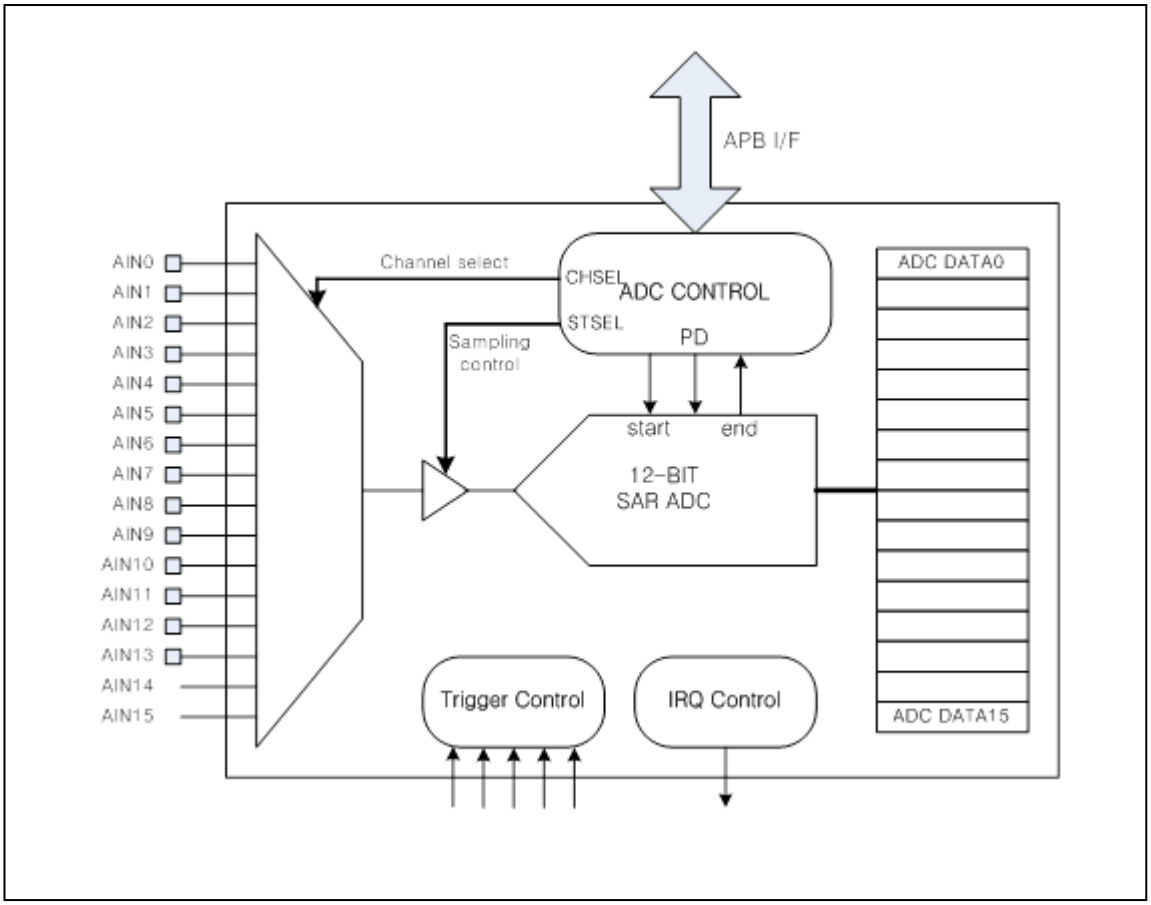


Figure 16.1. 12-bit A/D Converter Block Diagram

Pin Description

Table 16.1. External Signal

PIN NAME	TYPE	DESCRIPTION	Access PIN from ADC Channel		
AVDD	P	Analog Power(3.0V~VDD)	**ADC Channels to Pin Mapping**		
AVSS	P	Analog GND	**ADC0**	**ADC1**	**ADC2**
AN0	A	ADC Input 0	Channel 0	Channel 0	Channel 0
AN1	A	ADC Input 1	Channel 2	Channel 2	Channel 2
AN2	A	ADC Input 2	Channel 4	Channel 4	Channel 4
AN3	A	ADC Input 3	Channel 6	Channel 6	Channel 6
AN4	A	ADC Input 4	Channel 8	Channel 8	Channel 8
AN5	A	ADC Input 5	Channel 9	Channel 9	Channel 9
AN6	A	ADC Input 6	Channel 10	Channel 10	Channel 10
AN7	A	ADC Input 7	Channel 11	--	--
AN8	A	ADC Input 8	Channel 12	--	--
AN9	A	ADC Input 9	Channel 13	--	--
AN10	A	ADC Input 10	--	Channel 11	--
AN11	A	ADC Input 11	--	Channel 12	--
AN12	A	ADC Input 12	--	Channel 13	--
AN13	A	ADC Input 13	--	--	Channel 11
AN14	A	ADC Input 14	--	--	Channel 12
AN15	A	ADC Input 15	--	--	Channel 13

Registers

The base addresses of ADC units are listed in Table 16.2.

Table 16.2. ADC Base Address

	BASE ADDRESS
ADC0	0x4000_B000
ADC1	0x4000_B100
ADC2	0x4000_B200

Table 16.3 lists the register memory map.

Table 16.3. ADC Register Map

Name	Offset	R/W	Description	Reset
ADnMR	0x0000	R/W	ADC Mode register	0x00
ADnCSR	0x0004	R/W	ADC Channel Select register	0x00
ADnCR1	0x0008	R/W	ADC Control register	0x80
ADnTRG0	0x000C	R/W	ADC Trigger 0 channel register	0x00
ADnTRG1	0x0010	R/W	ADC Trigger 1 channel register	0x00
ADnTRG2	0x0014	R/W	ADC Trigger 2 channel register	0x00
ADnBCSR	0x0018	R/W	ADC Burst mode channel select	0x00
ADnCR2	0x0020	R/W	ADC Start	0x00
ADnSR	0x0024	R/W	ADC Status	0x00
ADnIER	0x0028	R/W	ADC Interrupt Enable register	0x00
AD0/1/2 DDR	0x002C	R	ADC0/1/2 DMA Data Register	0x00
ADnCCR	0x0070	R/W	ADC Channel compare register	0x00

ADnMR ADCn Mode Register

ADC Mode Registers are 32-bit registers. This register configures the ADC operation mode.

AD0MR=0x4000_B000, AD1MR=0x4000_B100, AD2MR=0x4000_B200

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
BWAIT	DMAEN DMACH	BWAITEN BSTCNT	ADCEN ADCMOD TRGEN TRGSRC
0x00	0 0 0 0 0 0 0 0	0 0 0 0 0 000	0 0 00 0 000
RW	RW RW	RW RW	RW RW RW RW

31 24	**BWAIT**		Burst wait count value (8-bit) ADC conversion delayed for "BWAIT value" * ADCCLK for next conversion in burst mode		
17	**DMAEN**		DMA enable bit – **should be set to '1'** when ADCEN='1'. When this bit is set to '1' the DDR register will be populated with the ADC reading. When the bit is set to '0' the DDR register is not populated. When DMA function is enabled, DMA request at the end of every conversion (also in burst mode) and interrupt request can only be generated when ADC receives DMA done from DMAC.		
16	**DMACH**		DMA channel option When DMACH is set, Channel information of DMA data will be located at ADDMAR[3:0] for half word size transfer. Channel information is at ADDMAR[19:16] in default.(DMACH is low)		
12	**BWAITEN**		Burst wait Enable In burst mode, wait cycles can be inserted between next channel selection and conversion start		
		0	BWAIT in burst mode disable		
		1	BWAIT in burst mode enable		
10 8	**BSTCNT**		Burst Count This identifies how many burst conversions to do during burst mode.		
		000	No Burst mode(Single)	100	5 burst AD conversion
		001	2 burst AD conversion	101	6 burst AD conversion
		010	3 burst AD conversion	110	7 burst AD conversion
		011	4 burst AD conversion	111	8 burst AD conversion
7	**ADCEN**	0	ADC disable		
		1	ADC enable		
5 4	**ADCMOD**	00	Single conversion mode		
		01	Continuous conversion mode		
		10	Reserved		
		11	Burst Mode		
3	**TRGEN**	0	Trigger sources disable		
		1	Trigger sources enable Trigger sources only support single & burst mode (not support continuous mode)		
2 0	**TRGSRC**	000	External pin Trigger		
		001	Timer 0 Trigger		
		010	Timer 1 Trigger		
		011	Timer 2 Trigger		
		100	MPWM 0 trigger		

101	MPWM 1 trigger
110	Reserved
111	Reserved

If ADCMOD is set for Burst Mode, ADC channels are controlled by BST1CH ~ BST8CH. Burst mode always starts from BST1CH (In 3 burst mode, analog inputs of channels which are assigned at BST1CH/BST2CH /BST3CH are converted sequentially).

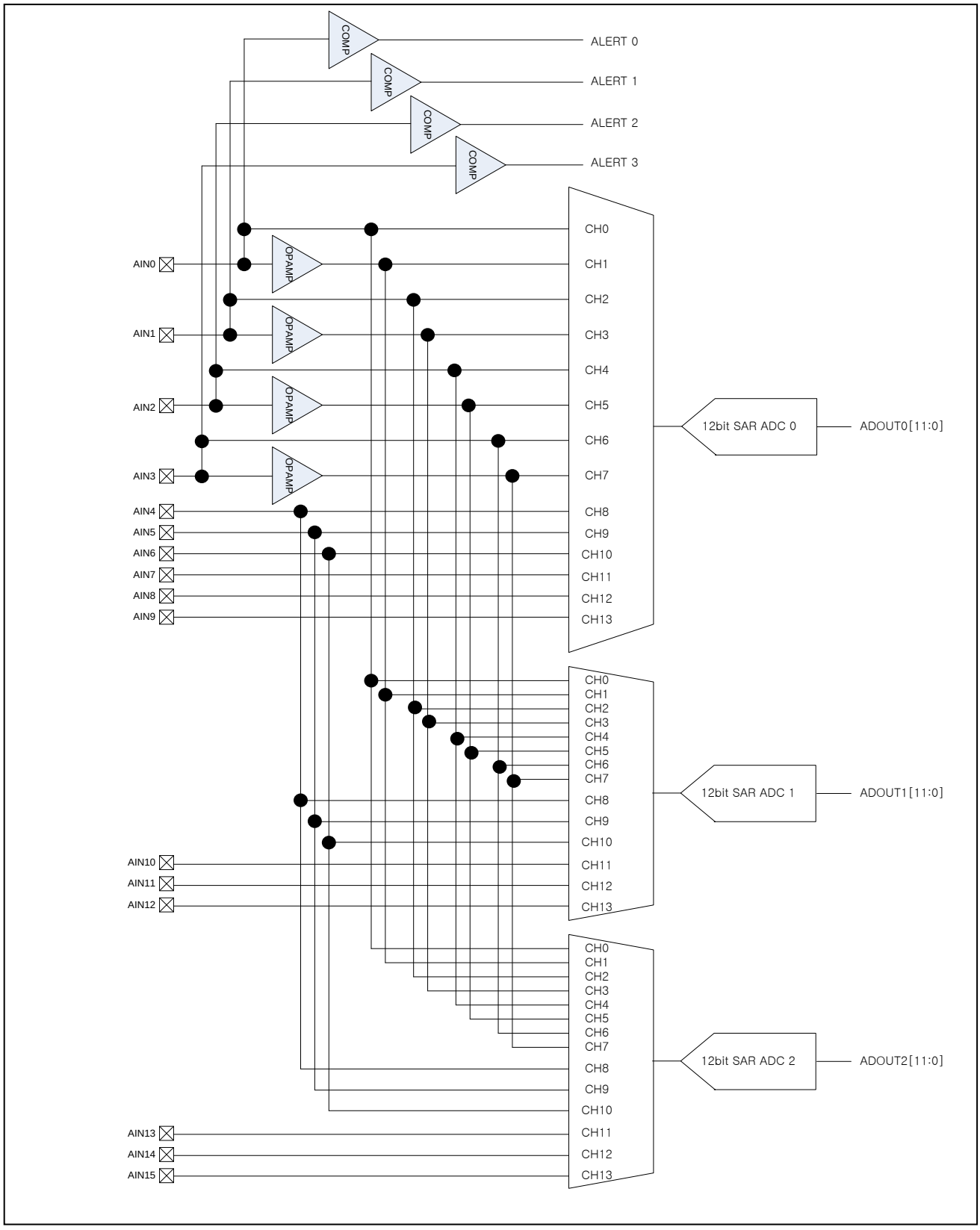


Figure 16.2. Analog Channel Block Diagram

ADnCSR ADCn Channel Select Register

ADC Channel Select Registers are 8-bit registers.

AD0SR=0x4000_B004, AD1SR=0x4000_B104, AD2SR=0x4000_B204,

7	6	5	4	3	2	1	0
				CHSEL			
0	0	0	0	0x0			
				RW			

3	CHSEL	0000	ADC channel 0 selection
0		0001	ADC channel 1 selection
		0010	ADC channel 2 selection
		0011	ADC channel 3 selection
		0100	ADC channel 4 selection
		0101	ADC channel 5 selection
		0110	ADC channel 6 selection
		0111	ADC channel 7 selection
		1000	ADC channel 8 selection
		1001	ADC channel 9 selection
		1010	ADC channel 10 selection
		1011	ADC channel 11 selection
		1100	ADC channel 12 selection
		1101	ADC channel 13 selection
		1110	ADC channel 14 selection
		1111	ADC channel 15 selection

Table16.1. ADC Channel Select

CHSEL	ADC0	ADC1	ADC2	
0000	AIN0	AIN0	AIN0	CH0
0001	AIN0_OPAMP	AIN0_OPAMP	AIN0_OPAMP	CH1
0010	AIN1	AIN1	AIN1	CH2
0011	AIN1_OPAMP	AIN1_OPAMP	AIN1_OPAMP	CH3
0100	AIN2	AIN2	AIN2	CH4
0101	AIN2_OPAMP	AIN2_OPAMP	AIN2_OPAMP	CH5
0110	AIN3	AIN3	AIN3	CH6
0111	AIN3_OPAMP	AIN3_OPAMP	AIN3_OPAMP	CH7
1000	AIN4	AIN4	AIN4	CH8
1001	AIN5	AIN5	AIN5	CH9
1010	AIN6	AIN6	AIN6	CH10
1011	AIN7	AIN10	AIN13	CH11
1100	AIN8	AIN11	AIN14	CH12
1101	AIN9	AIN12	AIN15	CH13
1110	-	-	-	CH14
1111	-	-	-	CH15

ADnCR1 ADCn Control Register 1

ADC Control Registers are 16-bit registers.

AD0CR1=0x4000_B008, AD1CR1=0x4000_B108, AD2CR1=0x4000_B208

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCPDA	CLKDIV							ADCPD	EXTCLK	CLKINVT	STSEL				
0	0x00							1	0	0	0x00				
RW	RW							RW	RW	RW	RW				

15	ADCPDA	ADC R-DAC disable to save power Don't set "1" here(it's optional bit)
14 8	CLKDIV[6:0]	ADC clock divider when EXTCLK is '0'. ADC clock = system clock/CLKDIV CLKDIV=0 : ADC clock=system clock CLKDIV=1 : ADC clock=stop
7	ADCPD	ADC Power Down 0 – ADC normal mode 1 – ADC Power Down mode (no ADC conversion will occur)
6	EXTCLK	ADC Clock Select. Clear this bit if the ADC will use the System clock using the CLKDIV for the divider. Set this bit if the ADC is configured through MCCR4 in the SCU. 0 – Use System clock(CLKDIV enabled to provide divider) 1 – ADC clock is externally configured (MCCR4)
5	CLKINVT	Divided clock inversion(optional bit) 0 – duty ratio of divided clock is larger than 50% 1 – duty ratio of divided clock is less than 50%
4 0	STSEL[4:0]	Sampling Time Selection ADC Sample & Hold circuit sampling time become (2 + STSEL[4:0]) ADCCLK cycles Minimum sampling time is 2 ADCCLK cycle When STSEL[4:0]=11111, sampling channel is always on.

ADnCR2 ADCn Control Register 2

The ADCn Control Register 2 is the ADC start register and is an 8-bit register.

AD0CR2=0x4000_B020, AD1CR=0x4000_B120, AD2CR=0x4000_B220,

7	6	5	4	3	2	1	0
			ASTOP				ASTART
0	0	0	0	0	0	0	0
			W				RW

4	ASTOP	0	No
		1	ADC conversion stop (will be clear next @ADC clock) This will stop the continuous and burst conversions. If ASTOP set after conversion cycle start, present conversion would be completed.
0	ASTART	0	No ADC conversion
		1	ADC conversion start (will be clear next @ADC clock) ADCEN should be "1" to start ADC

AD*n*TRG0 ADC Trigger 0 Channel Register

The ADC Trigger 0 registers are 32-bit registers.

AD0TRG0=0x4000_B00C, AD1TRG0=0x4000_B10C, AD2TRG0=0x4000_B20C

31	30	29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
		TRG0EN	MP0TRG6 MP0TRG5	MP0TRG4 MP0TRG3	MP0TRG2 MP0TRG1
0	0	0x00	0x0 0x0	0x0 0x0	0x0 0x0
		RW	RW RW	RW RW	RW RW

29 24	**TRG0EN**	Bit-5	0 – MP0TRG6 disable 1 – MP0TRG6 enable
		Bit-4	0 – MP0TRG5 disable 1 – MP0TRG5 enable
		Bit-3	0 – MP0TRG4 disable 1 – MP0TRG4 enable
		Bit-2	0 – MP0TRG3 disable 1 – MP0TRG3 enable
		Bit-1	0 – MP0TRG2 disable 1 – MP0TRG2 enable
		Bit-0	0 – MP0TRG1 disable 1 – MP0TRG1 enable
23 20	**MP0TRG6**		ADC trigger channel number for MP0ATR6 trigger (Channel number 14 and 15 are prohibited)
19 16	**MP0TRG5**		ADC trigger channel number for MP0ATR5 trigger (Channel number 14 and 15 are prohibited)
15 12	**MP0TRG4**		ADC trigger channel number for MP0ATR4 trigger (Channel number 14 and 15 are prohibited)
11 8	**MP0TRG3**		ADC trigger channel number for MP0ATR3 trigger (Channel number 14 and 15 are prohibited)
7 4	**MP0TRG2**		ADC trigger channel number for MP0ATR2 trigger (Channel number 14 and 15 are prohibited)
3 0	**MP0TRG1**		ADC trigger channel number for MP0ATR1 trigger (Channel number 14 and 15 are prohibited)

AD*n*TRG1 ADC Trigger 1 Channel Register

ADC Trigger 1 registers are 32-bit registers.

AD0TRG1=0x4000_B010, AD1TRG1=0x4000_B110, AD2TRG1=0x4000_B210

31	30	29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
		TRG1EN	MP1TRG6 / MP1TRG5	MP1TRG4 / MP1TRG3	MP1TRG2 / MP1TRG1
0	0	0x00	0x0 / 0x0	0x0 / 0x0	0x0 / 0x0
		RW	RW / RW	RW / RW	RW / RW

29 24	**TRG1EN**	Bit-5	0 – MP1TRG6 disable 1 – MP1TRG6 enable
		Bit-4	0 – MP1TRG5 disable 1 – MP1TRG5 enable
		Bit-3	0 – MP1TRG4 disable 1 – MP1TRG4 enable
		Bit-2	0 – MP1TRG3 disable 1 – MP1TRG3 enable
		Bit-1	0 – MP1TRG2 disable 1 – MP1TRG2 enable
		Bit-0	0 – MP1TRG1 disable 1 – MP1TRG1 enable
23 20	**MP1TRG6**		ADC trigger channel number by MP1ATR6 trigger (Channel number 14 and 15 are prohibited)
19 16	**MP1TRG5**		ADC trigger channel number by MP1ATR5 trigger (Channel number 14 and 15 are prohibited)
15 12	**MP1TRG4**		ADC trigger channel number by MP1ATR4 trigger (Channel number 14 and 15 are prohibited)
11 8	**MP1TRG3**		ADC trigger channel number by MP1ATR3 trigger (Channel number 14 and 15 are prohibited)
7 4	**MP1TRG2**		ADC trigger channel number by MP1ATR2 trigger (Channel number 14 and 15 are prohibited)
3 0	**MP1TRG1**		ADC trigger channel number by MP1ATR1 trigger (Channel number 14 and 15 are prohibited)

AD*n*TRG2 ADC Trigger 2 Channel Register

ADC Trigger 2 registers are 32-bit registers.

AD0TRG2=0x4000_B014, AD1TRG2=0x4000_B114, AD2TRG2=0x4000_B214

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																				EXTCH				T1CH				T0CH			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x0				0x0				0x0			
																				RW				RW				RW			

11 8	EXTCH	ADC trigger channel number by External Trigger (Channel number 14 and 15 are prohibited)
7 4	T1CH	ADC trigger channel number by TIMER1 trigger (Channel number 14 and 15 are prohibited)
3 0	T0CH	ADC trigger channel number for TIMER0 trigger (Channel number 14 and 15 are prohibited)

AD*n*BCSR ADC Burst Mode Channel Select

The ADC Burst Mode Channel Select Register is a 32-bit register.

AD0BCSR=0x4000_B018, AD1BCSR=0x4000_B118, AD2BCSR=0x4000_B218

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
BST8CH	BST7CH	BST6CH	BST5CH	BST4CH	BST3CH	BST2CH	BST1CH
0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
RW	RW	RW	RW	RW	RW	RW	RW

31 28	BST8CH	8th conversion channel selection in burst mode
27 24	BST7CH	7th conversion channel selection in burst mode
23 20	BST6CH	6th conversion channel selection in burst mode
19 16	BST5CH	5th conversion channel selection in burst mode
15 12	BST4CH	4th conversion channel selection in burst mode
11 8	BST3CH	3rd conversion channel selection in burst mode
7 4	BST2CH	2nd conversion channel selection in burst mode
3 0	BST1CH	1st conversion channel selection in burst mode

ADnSR ADCn Status Register

The ADC Status Register is a 32-bit register.

AD0SR=0x4000_B024, AD1SR=0x4000_B124, AD2SR=0x4000_B224

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		MPWM1TRG								MPWM0TRG						ADCH				TRG	BSTAT			ADEND	ABUSY	DOVRUN	DMAIRQ	TIRQ	BIRQ	CIRQ	SIRQ
0	0	0x00						0	0	0x00						0x0				0	000			0	0	0	0	0	0	0	0
		R								R						R				R	R			R	R	R	R	R	R	R	R

29 24	MPWM1TRG	This is only Test which MPWM triggered the ADC reading
21 16	MPWM0TRG	This is only Test which MPWM1 triggered the ADC reading
15 12	ADCH	ADC channel bits of present operation
11	TRG	Trigger event status TRG bit set @trigger_event and clear @EOC(end of conversion)
10 8	BSTAT	Burst mode operation count status
7	ADEND	ADC conversion end flag (will be reset @next ADC START).
6	ABUSY	ADC conversion busy flag – Conversion in process. Note: this will remain high during burst and continuous modes.
5	DOVRUN	DMA overrun flag (not interrupt) (DMA ACK didn't come until end of next conversion)
4	DMAIRQ	DMA done received (1: DMA transfer is completed) Write "1" to clear flag
3	TIRQ	ADC Trigger interrupt flag (0: no int / 1: int occurred) Write "1" to clear flag
2	BIRQ	ADC Burst interrupt flag (0: no int / 1: int occurred) Write "1" to clear flag
1	CIRQ	ADC Continuous interrupt flag (0: no int / 1: int occurred) Write "1" to clear flag
0	SIRQ	ADC Single interrupt flag (0: no int / 1: int occurred) Write "1" to clear flag. Use this bit to identify when the ADC conversion data is ready after starting a conversion with the ASTART bit.

AD*n*IER Interrupt Enable Register

ADC interrupt enable register. Individual interrupt sources can be enabled by writing a **1**.

AD0IER=0x4000_B028, AD1IER=0x4000_B128, AD2IER=0x4000_B228,

7	6	5	4	3	2	1	0
			DIEN	TIEN	BIEN	CIEN	SIEN
0	0	0	0	0	0	0	0
			RW	RW	RW	RW	RW

4	DIEN	DMA done interrupt enable 0: interrupt disable 1: interrupt enable
3	TIEN	ADC trigger conversion interrupt enable
2	BIEN	ADC burst conversion interrupt enable
1	CIEN	ADC continuous conversion interrupt enable
0	SIEN	ADC single conversion interrupt enable

AD*n*DDR ADC 0/1/2 DMA Data Register

ADC DMA Data Registers are 16-bit registers.

ADC conversion result register for DMA and single conversion (AD data of just completed conversion)

AD0DDR=0x4000B02C, AD1DDR=0x4000B12C, AD2DDR=0x4000B22C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDMAR												ADMACH			
0x000												0x0			
R												R			

15 4	ADDMAR	ADC conversion result data (12-bit)
3 0	ADMACH	ADC data channel indicator

AD*n*CCR ADC Channel Compare Control Register

ADC Channel Compare Control Registers are 32-bit registers.

AD0CCR=0x4000_B070, AD1CCR=0x4000_B170, AD2CCR=0x4000_B270

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								COMPOUT			LTE	CCH				CVAL																
0	0	0	0	0	0	0	0	0	0	0	0	000				0x000												0	0	0	0
								R			RW	RW				RW															

23	COMPOUT	0	If LTE condition is FALSE
		1	If LTE condition is TRUE (MPWM trigger source)
20	LTE	0	Set compare output when AD conversion value is greater than compare value (CVAL)
		1	Set compare output when AD conversion value is less than or equal to compare value(CVAL)
19 16	CCH	Compare channel	
15 4	CVAL	Compare value	

Functional Description

The PER2 and PCER2 registers must be configured to enable the ADC peripheral and the ADC peripheral clock. The ADC block provides the ability to convert an analog signal to a digital value. The ADC compares the input channel with the AVDD voltage and provides a 12-bit value.

Voltage value = (ADC Reading / 4096) * AVDD voltage

The ADC clock can be configured up to 22.5 MHz and be driven from any of the available clocks – System Clock, Ring OSC, Bus Clock, Int OSC, External OSC, or the PLL clock. There is a 6-bit divider available for the system clock (divider must be greater than 1) or the ADC clock can be configured in the MCCR4 register, which provides access to all clocks and the 8-bit divider. The clock is selected in the CR1 register (and optionally configured in the SCU MCCR4 register).

The ADC takes 15 ADC clocks to complete one sample, starting with a single clock, followed by a sample and hold time (minimum of 2 ADC clocks) then 1 clock per bit (12 bits). To increase sample time, configure up to 511 clock sampling time (which would then take 511 + 15 = 526 ADC clocks per sample).

The maximum ADC clock that can be used is calculated as:

ADC Clock = 1.5Msps * (15 clocks per sample + Sample time)

Example (Sampling time = 0):

ADC clock = 1.5Msps * (15 clocks + 0) = 22.5 MHz

In the above example, if the system clock was running at 72 MHz, the divider cannot be less than 4.

The burst feature allows the programmer to retrieve multiple readings (up to 8) with only one start request. The ADC block automatically goes through all 8 and takes readings without intervention. This feature is best utilized with DMA to store the data. It is also possible to wait a specific time (up to 255 ADC clocks) before starting the next conversion.

To use the burst feature, populate Burst channels 0-7 of the reading you wish to take. They can be the same or different channels. Provide the count of burst operations (number of channels that have been assigned) in the burst count, then modify the ADCMOD to specify burst mode. If desired, populate the BWAIT value with the number of clocks you wish to have between readings and set the BWAITEN bit. You can either manually start it by setting the AStart bit or set a trigger to start it.

BURST Mode Example:

To take 3 readings of AN0 and AN2 for an average value, set the Burst channels as:

BST1CH = 0x00 (AN0 is ADC Channel 0)
BST2CH = 0x04 (AN2 is ADC Channel 4)
BST3CH = 0x00
BST4CH = 0x04
BST5CH = 0x00
BST6CH = 0x04
Set the BSTCNT in the MR register to 0x06
Set the ADCMOD in the MR register to 0x03
Set the AStart Bit in the CR2 register to take the readings.

It is possible to trigger the burst reading to take the readings when an event occurs.

General ADC Setup Procedure

1. Allow the modification of the I/O pins to use the ADC inputs needed by writing the unlock sequence as described in Port Control Unit (PCU), no pullups enabled.
2. Enable the ADC peripherals needed in the PER2 register.
3. Enable the ADC peripheral clock in the PCER2 register.
4. Select the alternating function for the ADC inputs (Port n MUX registers).
5. Configure the ADC mode in the ADCnMODE register and enabled the channel ADCn.
6. Configure the ADCnCR1 register and write an appropriate clock divider value.
7. Configure TRG0 to enable or disable ADC trigger sources.
8. Configure the ADC Burst Mode (AD*n*BCSR) register for ADC operation with burst mode described below.
9. Configure the ADnIER ADC interrupt control register.

ADC Single Mode Timing Diagram

ADC conversion is started when ADCn.CR.ASTART is written as '1' in single conversion mode. After ADCnCR.ASTART is set, SOC (start of Conversion) will be activated in 2 ADC clocks; ADCn.SR.SIRQ will be set in 1 ADC clock and 2 PCLKs after the end of conversion.

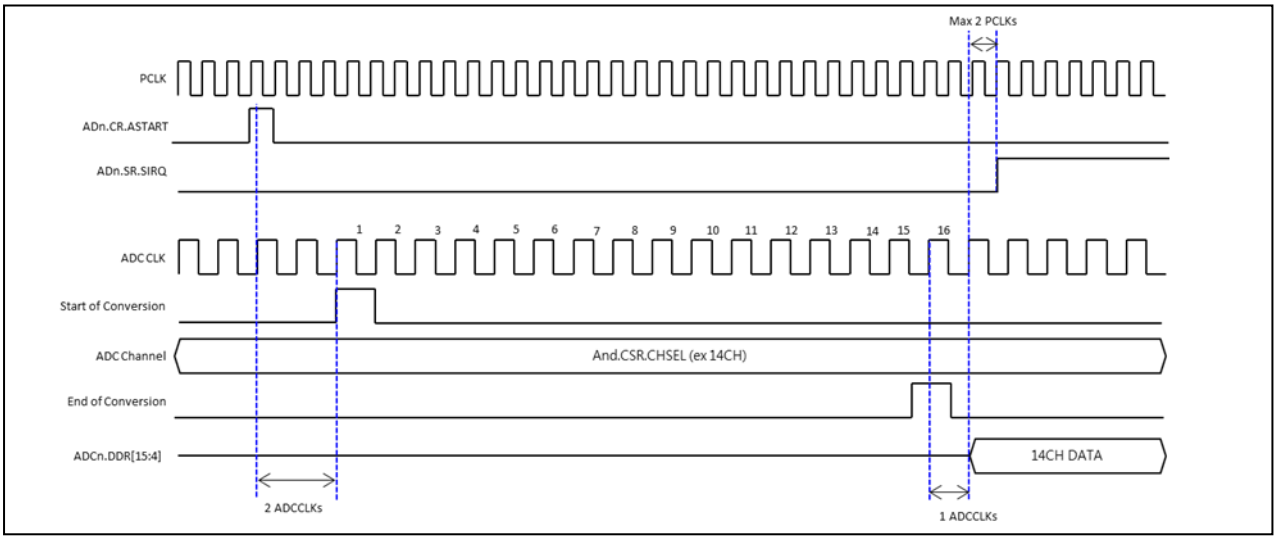


Figure 16.3. ADC Single Mode Timing Diagram

ADC Continuous Mode Timing Diagram

The ADC conversion in burst mode is almost the same as continuous mode. Burst mode has ADnCR.BWAIT. BWAIT in 8-bit register and can delay the time of SOC. Burst wait counter (BWAIT) in ADC clock domain. ADnSR.BIRQ is set as 1 after the last burst operation; see Figure 16.4

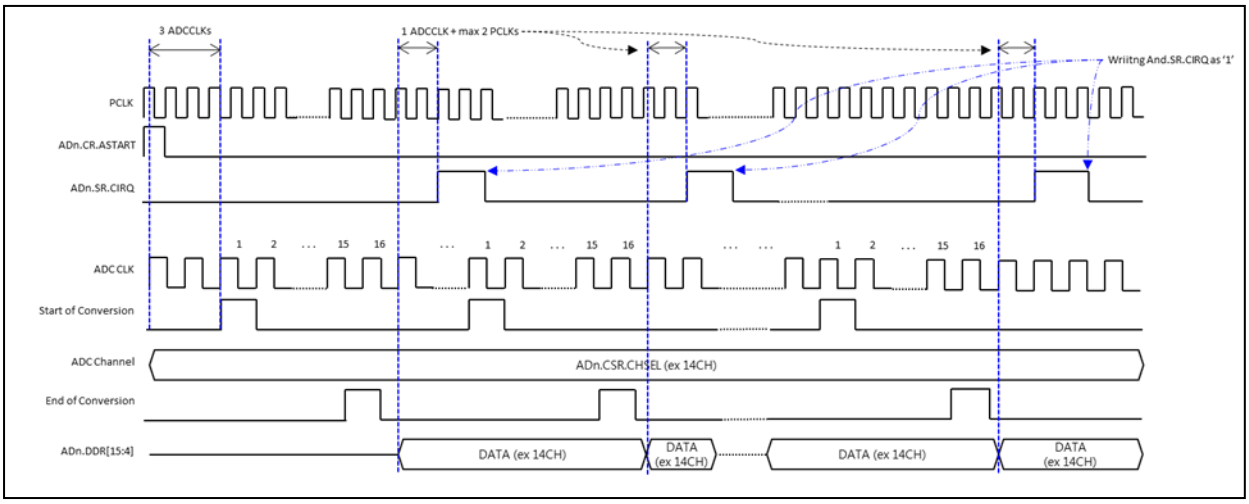


Figure 16.4. ADC Continuous Mode Timing

ADC Burst Mode Timing Diagram

The ADC conversion in burst mode is almost same as in continuous mode. Burst mode has ADnCR.BWAIT. BWAIT in 8-bit register and can delay the time of SOC. Burst wait counter (BWAIT) in the ADC clock domain. ADnSR.BIRQ is set as 1 after the last burst operation; see Figure 16.5

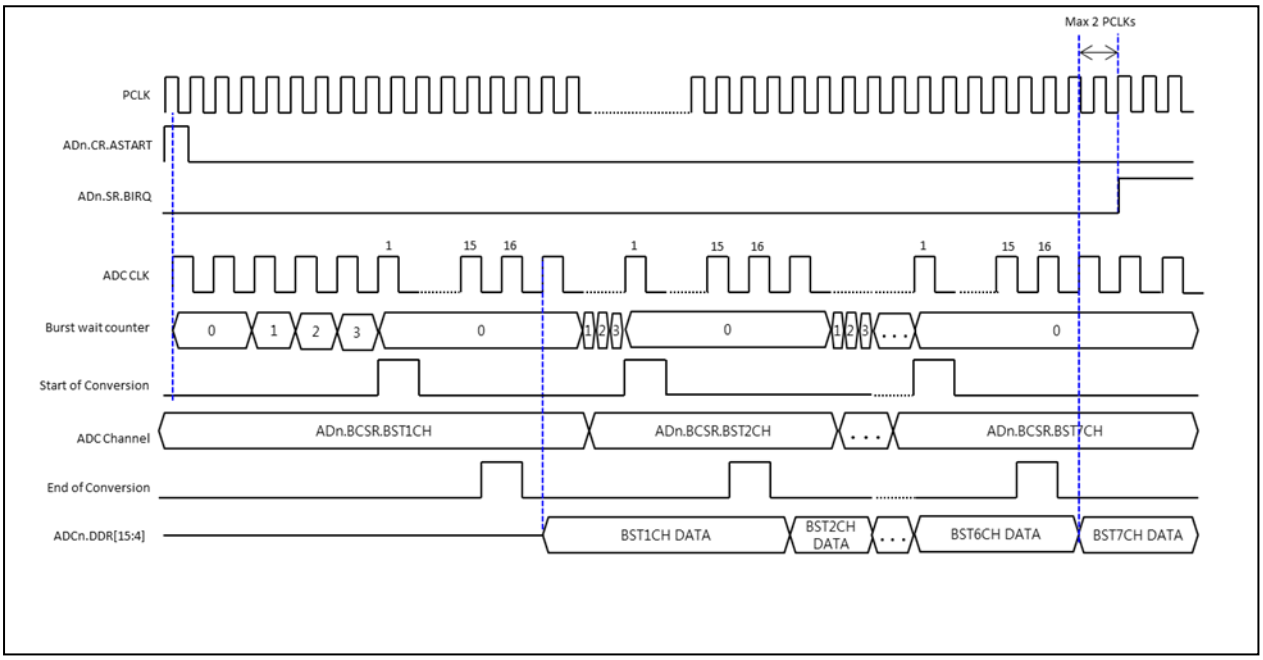


Figure 16.6. ADC Burst Mode Timing (when ADCn.MR.BWAIT = '8'h3' and BSTCNT = '3'b111')

17. Analog Front End

Introduction

Analog Front End (AFE) is the interface controller for OPAMPs and comparators.

Features include:

- 4 OPAMPs
- 4 comparators
- OPAMP output can be connected with ADC or comparator
- Internal BGR reference for comparator
- Comparator output debounce function
- Level and edge interrupt mode support for comparator

Figure 17.1 shows a block diagram of the AFE.

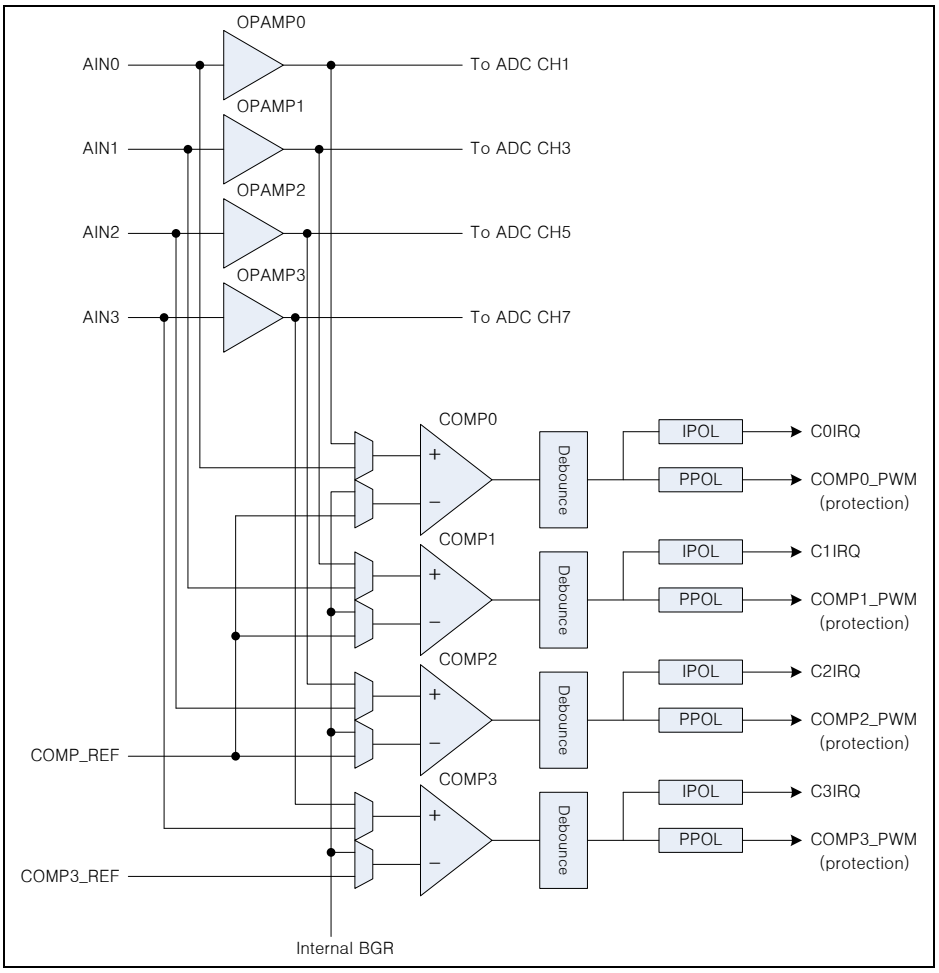


Figure 17.1. Analog Front End Block Diagram

Pin Description

Table 17.1. External Signal

PIN NAME	TYPE	DESCRIPTION
AVDD	P	Analog Power (3.0V~VDD)
AVSS	P	Analog GND
CP0	A	Comparator Input 0
CP1	A	Comparator Input 1
CP2	A	Comparator Input 2
CP3	A	Comparator Input 3
CREF0	A	Comparator Reference Input 0
CREF1	A	Comparator Reference Input 1

Registers

The base address of AFE is listed in Table 17.2.

Table 17.2. AFE Base Address

	BASE ADDRESS
AFE	0x4000_B300

Table 17.3 shows the register memory map.

Table 17.3. AFE Register Map

Name	Offset	R/W	Description	Reset
OPA0CR	0x0000	R/W	AFE OPAMP 0 control register	0x00
OPA1CR	0x0004	R/W	AFE OPAMP 1 control register	0x00
OPA2CR	0x0008	R/W	AFE OPAMP 2 control register	0x00
OPA3CR	0x000C	R/W	AFE OPAMP 3 control register	0x00
CMP0CR	0x0020	R/W	AFE Comparator 0 control register	0x10
CMP1CR	0x0024	R/W	AFE Comparator 1 control register	0x10
CMP2CR	0x0028	R/W	AFE Comparator 2 control register	0x10
CMP3CR	0x002C	R/W	AFE Comparator 3 control register	0x10
CMPDBR	0x0030	R/W	AFE Comparator debounce register	0x00
CMPICR	0x0034	R/W	AFE Comparator interrupt control	0x00
CMPIER	0x0038	R/W	AFE Comparator interrupt enable	0x00
CMPSR	0x003C	R	AFE Comparator status register	0x00

OPAnCR OPAMP 0/1/2/3 Control Registers

Analog-front-end OPAMP 0/1/2/3 Control Registers are 8-bit registers. All four registers (AFEOPA0~AFEOPA3) have the same functions.

OPA0CR=0x4000_B300, OPA1CR =0x4000_B304
OPA2CR =0x4000_B308, OPA3CR =0x4000_B30C

7	6	5	4	3	2	1	0
			OPAEN	GAIN			
0	0	0	0	0x0			
			RW	RW			

4	OPAEN	0	OPAMP n Disable		
		1	OPAMP n Enable		
3	GAIN	0000	Gain = 2.19	1000	Gain = 4.37
0		0001	Gain = 2.33	1001	Gain = 5.0
		0010	Gain = 2.5	1010	Gain = 5.83
		0011	Gain = 2.69	1011	Gain = 7.0
		0100	Gain = 2.92	1100	Gain = 8.74
		0101	Gain = 3.18	1101	Reserved
		0110	Gain = 3.5	1110	Reserved
		0111	Gain = 3.89	1111	Gain = 1.00

CMPnCR Comparator 0/1/2/3 Control Register

Analog-front-end Comparator0/1/2/3 Control Registers are 8-bit registers. All four registers (AFECOMP0~AFECOMP3) have the same functions.

CMP0CR=0x4000_B320,CMP1CR =0x4000_B324
CMP2CR =0x4000_B328, CMP3CR =0x4000_B32C

7	6	5	4	3	2	1	0
			CMPEN			CINSEL	REFSEL
0	0	0	1	0	0	0	0
			RW			RW	RW

4	CMPEN	0	Comparator 0~3 Enable
		1	Comparator 0~3 Disable
1	CINSEL		Comparator input selection
		0	Input from OPAMP 0~3 each
		1	Input from external pin (see pin mux table)
0	REFSEL		Comparator reference selection
		0	Reserved
		1	REF input from external pin (see pin mux table)

When OPAMP is disabled, the OPAMP output is unknown (floating). Therefore, the user should set (write 1) CINSELx to choose the external input when OPAMP is an inactive state.

CMPDBR Comparator Debounce Register

The Analog Front End Comparator Debounce Register is a 32-bit register.

CMPDBR=0x4000_B330

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								DBNCTB								C3DBNC				C2DBNC				C1DBNC				C0DBNC			
0	0	0	0	0	0	0	0	0x00								0x0				0x0				0x0				0x0			
								RW								RW				RW				RW				RW			

23 16	DBNCTB[3:0]	Debounce time base counter System clock/(DBNCTB *2) becomes shift clock of debounce logic When DBNCTB is 0, system clock would be debounce clock.
15 0	CxDBNC[4:0]	Debounce shift Selection When it is 0x0, debounce function is disable Shift number of debounce logic is (CxDBNC + 1) when CxDBNC is more than 1.

CMPICR Comparator Interrupt Control Register

The Analog Front End Comparator Interrupt Control Register is a 16-bit register.

CMPICR=0x4000_B334

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PPOL3	PPOL2	PPOL1	PPOL0	IPOL3	IPOL2	IPOL1	IPOL0	C3IMOD		C2IMOD		C1IMOD		C0IMOD	
-	-	-	-	-	-	-	-	00		00		00		00	
R	R	R	R	R	R	R	R	RW		RW		RW		RW	

15 14 13 12	PPOL3 PPOL2 PPOL1 PPOL0	0	Comparator outs for PWM protection will not be inverted
		1	Comparator outs for PWM protection will be inverted (if debounce is enable, debounced output will be inverted)
11 10 9 8	IPOL3 IPOL2 IPOL1 IPOL0	0	When comparator output is high, IRQ bit is set (CxIMODE = 00)
		1	When comparator output is low, IRQ bit is set (CxIMODE = 00)
3 2 1 0	C3IMODE C2IMODE C1IMODE C0IMODE	00	Comparator interrupt mode IRQ at level output
		01	IRQ at rising edge of comparator output
		10	IRQ at falling edge of comparator output
		11	IRQ at both edge of comparator output

CMPIER Comparator Interrupt Enable Register

The Analog Front End Interrupt Enable Register is an 8-bit register.

CMP0CR=0x4000_B338

7	6	5	4	3	2	1	0
				CMP3IE	CMP2IE	CMP1IE	CMP0IE
0	0	0	0	0	0	0	0
				RW	RW	RW	RW

3	CMP3IE	AFE comparator 3 interrupt enable 0 –interrupt disable 1 –interrupt enable
2	CMP2IE	AFE comparator 2 interrupt enable 0 –interrupt disable 1 –interrupt enable
1	CMP1IE	AFE comparator 1 interrupt enable 0 –interrupt disable 1 –interrupt enable
0	CMP0IE	AFE comparator 0 interrupt enable 0 –interrupt disable 1 –interrupt enable

CMPSR Comparator Status Register

The Analog Front End Status Register is a 16-bit register.

CMPSR=0x4000_B33C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C3RAW	C2RAW	C1RAW	C0RAW	C3OUT	C2OUT	C1OUT	C0OUT					C3IRQ	C2IRQ	C1IRQ	C0IRQ
-	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R					RC1	RC1	RC1	RC1

15 12	C3RAW C2RAW C1RAW C0RAW	AFE comparator raw outputs These values come from comparator output pin (before debouncing)
11 8	C3OUT C2OUT C1OUT C0OUT	AFE comparator output monitor bit These values are debounced outputs.
3 0	C3IRQ C3IRQ C2IRQ C0IRQ	AFE comparator interrupt flag (0: no int / 1: int occurred) Write “1” to clear flag

Functional Description

The PER2 and PCER2 registers must be configured to enable the AFE peripheral and the AFE peripheral clock.

18. Electrical Characteristics

DC Characteristics

Absolute Maximum Ratings

Absolute maximum ratings are limiting values of operating and environmental conditions which should not be exceeded under the worst possible conditions.

Table 18.1. Absolute Maximum Ratings

Parameter		Symbol	min	max	unit
Power Supply (VDD)		VDD	-0.5	+6	V
Analog Power Supply (AVDD)		AVDD	-0.5	+6	V
Input High Voltage			-	VDD+0.5	V
Input Low Voltage			VSS–0.5	-	V
Output Low Current per pin		I_{OL}	-	20	mA
Output Low Current Total	80-pin	$\sum I_{OL}$	-	100	mA
	64-pin	$\sum I_{OH}$	-	80	
Output Low Current per pin		I_{OH}	-	10	mA
Output Low Current Total	80-pin	$\sum I_{OH}$	-	100	mA
	64-pin	$\sum I_{OH}$	-	80	
Power consumption			-	200	mW
Input Main Clock Range			0.4	10	MHz
Operating Frequency			-	72	MHz
Storage Temperature		Tst	-55	+125	℃
Operating Temperature		Top	-40	+85	℃

DC Characteristics

Table 18.2. Recommended Operating Conditions

Parameter	Symbol	Condition	Min	Typ.	Max	unit
Supply Voltage	VDD		3.0		5.5	V
Supply Voltage			3.0	5.0	5.5	V
Operating Frequency	FREQ	MOSC	4		8	MHz
		INTOSC		20		MHz
		PLL	4		80	MHz
Operating Temperature	Top	Top	-40		+85	℃

Table 18.3. DC Electrical Characteristics (VDD = +5V, Ta = 25 ℃)

Parameter	Symbol	Condition	Min	Typ.	Max	unit
Input Low Voltage	V_{IL}	Schmitt input	-	-	0.2VDD	V
Input High Voltage		Schmitt input	0.8VDD	-	-	V
Output Low Voltage	V_{OL}	I_{OL} = 10mA	-	-	VSS+0.5	V
Output High Voltage	V_{OH}	I_{OH} = -3mA	VDD–0.5	-	-	V
Output Low Current	I_{OL}		-	-	10	mA
Output High Current	I_{OH}		-3	-		mA
Input High Leakage	I_{IH}				4	uA
Input Low Leakage	I_{IL}		-4			
Pull-up Resister	R_{PU}	Rmax:VDD=3.0V Rmin:VDD=5V	30	-	70	kΩ

Current Consumption

Table 18.4. Current Consumption in Each Mode (Temperature: +25℃)

Parameter	Symbol	Condition	Min	Typ.	Max	unit
Normal Operation	IDD_{NORMAL}	ROSC=RUN IOSC20=RUN MXOSC=8MHz HCLK=72MHz	-	35	-	mA
Sleep Mode	IDD_{SLEEP}	ROSC=RUN IOSC20=RUN MXOSC=STOP HCLK =RUN	-	3	-	mA

POR Electrical Characteristics

Table 18.5. POR Electrical Characteristics (Temperature: -40 ~ +85℃)

Parameter	Symbol	Condition	Min	Typ.	Max	unit
Operating Voltage	VDD18		1.6	1.8	2.0	V
Operating Current	IDD_{PoR}	Typ. <6uA If always on	-	60	-	nA
POR Set Level	VR_{PoR}	VDD rising (slow)	1.3	1.4	1.55	V
POR Reset Level	VF_{PoR}	VDD falling (slow)	1.1	1.2	1.4	V

LVD Electrical Characteristics

Table 18.6. LVD Electrical Characteristics (Temperature: -40 ~ +85℃)

Parameter	Symbol	Condition	Min	Typ.	Max	unit
Operating Voltage	VDD		1.7		5	V
Operating Current	IDD_{LVD}	Typ. <6uA when always on	-	1	-	mA
LVD Set Level 0	VLVD0	VDD falling (slow)	1.6	1.8	2.0	V
LVD Set Level 1	VLVD1	VDD falling (slow)	2.0	2.2	2.5	V
LVD Set Level 2	VLVD2	VDD falling (slow)	2.5	2.7	3.0	V
LVD Set Level 3	VLVD3	VDD falling (slow)	3.9	4.3	4.6	V

VDC Electrical Characteristics

Table 18.7. VDC Electrical Characteristics (Temperature: -40 ~ +85℃)

Parameter	Symbol	Condition	Min	Typ.	Max	unit
Operating Voltage	VDD_{VDC}		3.0	-	5.5	V
VDC Output Voltage	$VOUT_{VDC}$	@RUN	1.62	1.8	1.98	V
		@STOP	1.4	1.8	2.0	V
Regulation Current	I_{OUT}				100	mA
Drop-out Voltage	$VDROP_{VDC}$	VDDVDC=3.0 V IOUT=100mA	-	-	200	mV
Current Consumption	IDD_{NORM}	@RUN	-	100	150	uA
	IDD_{STOP}	@STOP	-	1	2	uA

External OSC Characteristics

Table 18.8. External OSC Characteristics (Temperature: -40 ~ +85℃)

Parameter	Symbol	Condition	Min	Typ	Max	unit
Operating Voltage	VDD		3.0	-	5.5	V
IDD		@4MHz/5V	-	240		uA
Frequency	$OSCF_{req}$		4	8	10	MHz
Output Voltage	OSC_{VOUT}		1.2	2.4	-	V
Load Capacitance	$LOAD_{CAP}$		5	22	35	pF

Internal RC OSC Characteristics

Table 18.9. Internal RC OSC Characteristics (Temperature: -40 ~ +85℃)

Parameter	Symbol	Condition	Min	Typ	Max	unit
Operating Voltage	VDD		3.0		5.5	V
IDD	I_{OSC}	@20MHz	-	240		uA
Frequency	$IOSCF_{req}$			20		MHz

PLL Electrical Characteristics

Table 18.10. PLL Electrical Characteristics (Temperature: -40 ~ +85℃)

Parameter	Symbol	Condition	Min	Typ.	Max	unit
Operating Voltage	VDD_{PLL}		3.0		5.5	V
Output Frequency	FOUT		4		80	MHz
Operating Current	IDD_{PLL}	@80MHz		1.3		mA
Duty	$FOUT_{DUTY}$		40	-	60	%
P-P Jitter	JITTER	@Lock			500	Ps
VCO	VCO		30		80	MHz
Input Frequency	FIN		4		8	MHz
Locking time	LOCK				1	ms

ADC Electrical Characteristics

Table 18.11. ADC Electrical Characteristics (Temperature: -40 ~ +85℃)

Parameter	Symbol	Condition	Min	Typ.	Max	unit
Operating Voltage	AVDD		3.0	5	5.5	V
Reference Voltage	AVREF		3.0	5	5.5	V
Resolution				12		Bit
Operating Current	IDDA				2.8	mA
Analog Input Range			0		AVDD	V
Conversion Rate				-	1.6	Msps
Operating Frequency	ACLK				25	MHz
DC Accuracy	INL			±2.5		LSB
	DNL			±1.0		LSB
Offset Error				±1.5		LSB
Full Scale Error				±1.5		LSB
SNDR	SNDR			68		dB
THD				-70		dB

OP-Amp Electrical Characteristics

Table 18.12. ADC Electrical Characteristics (Temperature: -40 ~ +85℃)

Parameter	Symbol	Condition	Min	Typ.	Max	unit
Operating Voltage	AVDD		3.0	5	5.5	V
Operating Current	IDDA				2.2	mA
Analog Input Range			0		AVDD-1.4	V
Slew Rate		@ CL = 20pF		15		V/us
Gain Error		Gain=2.19~4.37	-3		+3	%
		Gain=5.0~8.74	-4		+4	%
Common Mode Rejection Ratio			50	70		dB
Power Supply Rejection Ratio			40	70		dB
Gain Bandwidth		@CL=20pF		16		MHz
Open Loop Voltage Gain				100		dB
Open Loop Phase Margin		@CL=20pF		45		°
Closed Loop Phase Margin				70		°
Turn On time				2		us
Gain			2.19		8.74	

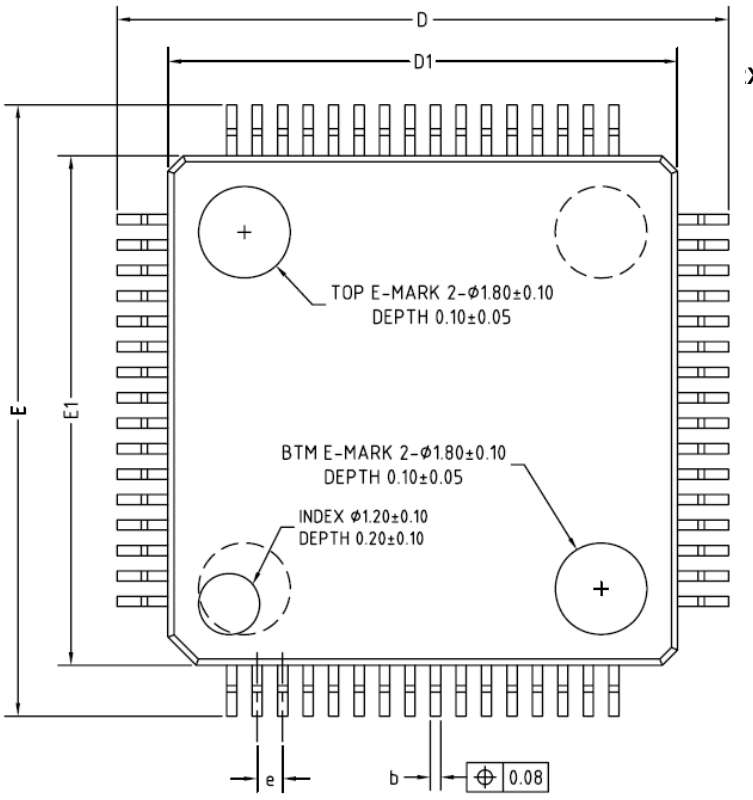
Comparator Electrical Characteristics

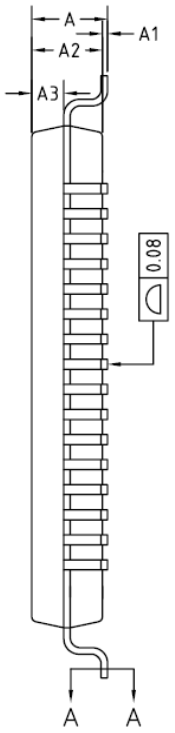
Table 18.13. Comparator Electrical Characteristics (Temperature: -40 ~ +85℃)

Parameter	Symbol	Condition	Min	Typ.	Max	unit
Operating Voltage	AVDD		3.0	5	5.5	V
Analog Input Range	VIN		AVSS		AVDD	V
Reference Input Range	VREF		0.9		AVDD-0.2	V
Input Offset Voltage			-4		+4	%
Response Time					1	us

19. Package

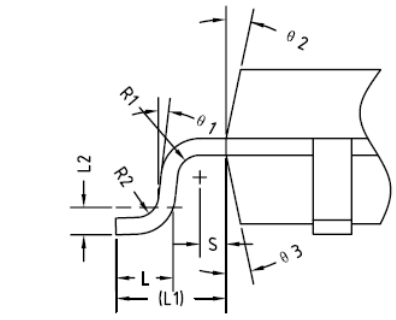
LQFP-80 Package

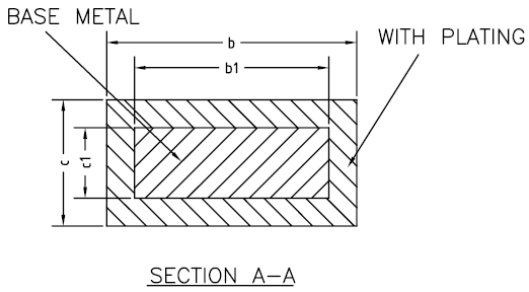




COMMON DIMENSIONS
OF MEASURE=MILLIMETER)

SYMBOL	MIN	NOM	MAX
A	–	–	1.60
A1	0.05	–	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.18	–	0.27
b1	0.17	0.20	0.23
c	0.13	–	0.18
c1	0.12	0.127	0.134
D	13.80	14.00	14.20
D1	11.90	12.00	12.10
E	13.80	14.00	14.20
E1	11.90	12.00	12.10
e	0.40	0.50	0.60
L	0.45	0.60	0.75
L1		1.00REF	
L2		0.25BSC	
R1	0.08	–	–
R2	0.08	–	0.20
θ	0°	3.5°	7°
θ 1	0°	–	–
θ 2	11°	12°	13°
θ 3	11°	12°	13°



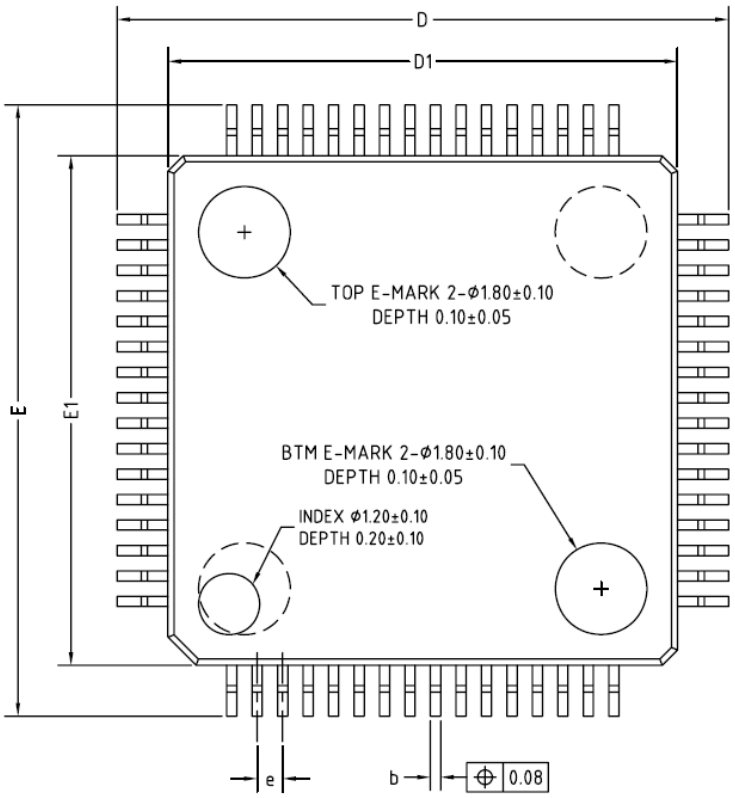


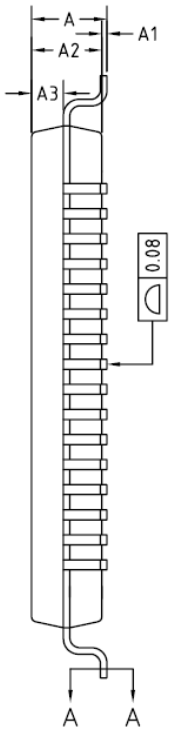
SECTION A–A

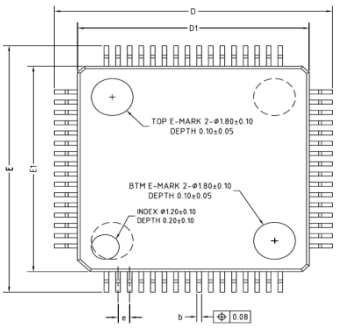
NOTE: ALL DIMENSIONS REFER TO JEDEC STANDARD MS-026 BDD DO NOT INCLUDE MOLD FLASH OR PROTRUSIONS.

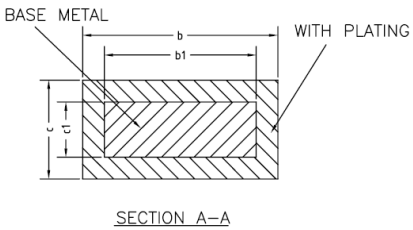
Figure 19.2 Package Dimension (LQFP-80 12X12)

LQFP-64 Package









SECTION A–A

COMMON DIMENSIONS
(UNITS OF MEASURE=MILLIMETER)

SYMBOL	MIN	NOM	MAX
A	–	–	1.60
A1	0.05	–	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.18	–	0.27
b1	0.17	0.20	0.23
c	0.13	–	0.18
c1	0.12	0.127	0.134
D	11.80	12.00	12.20
D1	9.90	10.00	10.10
E	11.80	12.00	12.20
E1	9.90	10.00	10.10
e		0.50BSC	
L	0.45	0.60	0.75
L1		1.00REF	
L2		0.25BSC	
R1	0.08	–	–
R2	0.08	–	0.20
S	0.20	–	–
θ	0°	3.5°	7°
θ 1	0°	–	–
θ 2	11°	12°	13°
θ 3	11°	12°	13°

NOTE:
ALL DIMENSIONS REFER TO JEDEC STANDARD MS-026 BDD DO NOT INCLUDE MOLD FLASH OR PROTRUSIONS.

Figure 19.3. Package Dimension (LQFP-64 10X10)

20. Ordering Information

Table 20.1 identifies the basic features and package styles available for the Z32F1281 MCU.

Table 20.1. Ordering Information for the Z32F128 MCU

Part Number	Flash	SRAM	UART	SPI	I^2C	MPWM	ADC	I/O PORT	PKG
Z32F12811ATS	128KB	12KB	4	2	2	2	3-unit 16 channels	68	LQFP-80
Z32F12811ARS			2	2	1	2		48	LQFP-64

Zilog part numbers consist of a number of components, which are described below using part number Z32F12811ATS as an example.