MB88F333 'Indigo-L'

LSI Product Specifications

27th July, 2011 Edition 1.60

Preface

Objectives and Intended Reader

Thank you very much for your continued special support for Fujitsu Microelectronics semiconductor products.
MB88F333 'Indigo-L' is a LSI product for graphics applications.
This manual describes functions and operations of MB88F333 'Indigo-L' for engineers who design products using MB88F333 'Indigo-L'. Read through this manual before designing an application..

- The contents of this document are subject to change without notice.
  Customers are advised to consult with sales representatives before ordering.

- The information, such as descriptions of function and application circuit examples, in this document are presented solely for the purpose of reference to show examples of operations and uses of FUJITSU SEMICONDUCTOR device; FUJITSU SEMICONDUCTOR does not warrant proper operation of the device with respect to use based on such information. When you develop equipment incorporating the device based on such information, you must assume any responsibility arising out of such use of the information. FUJITSU SEMICONDUCTOR assumes no liability for any damages whatsoever arising out of the use of the information.

- Any information in this document, including descriptions of function and schematic diagrams, shall not be construed as license of the use or exercise of any intellectual property right, such as patent right or copyright, or any other right of FUJITSU SEMICONDUCTOR or any third party or does FUJITSU SEMICONDUCTOR warrant non-infringement of any third-party's intellectual property right or other right by using such information. FUJITSU SEMICONDUCTOR assumes no liability for any infringement of the intellectual property rights or other rights of third parties which would result from the use of information contained herein.

- The products described in this document are designed, developed and manufactured as contemplated for general use, including without limitation, ordinary industrial use, general office use, personal use, and household use, but are not designed, developed and manufactured as contemplated (1) for use accompanying fatal risks or dangers that, unless extremely high safety is secured, could have a serious effect to the public, and could lead directly to death, personal injury, severe physical damage or other loss (i.e., nuclear reaction control in nuclear facility, aircraft flight control, air traffic control, mass transport control, medical life support system, missile launch control in weapon system), or (2) for use requiring extremely high reliability (i.e., submersible repeater and artificial satellite).
  Please note that FUJITSU SEMICONDUCTOR will not be liable against you and/or any third party for any claims or damages arising in connection with above-mentioned uses of the products.

- Any semiconductor devices have an inherent chance of failure. You must protect against injury, damage or loss from such failures by incorporating safety design measures into your facility and equipment such as redundancy, fire protection, and prevention of over-current levels and other abnormal operating conditions.

- Exportation/release of any products described in this document may require necessary procedures in accordance with the regulations of the Foreign Exchange and Foreign Trade Control Law of Japan and/or US export control laws.

- The company names and brand names herein are the trademarks or registered trademarks of their respective owners.

History		
Date	Ver.	Contents
2011/07/27	1.60	TCON: 10.5.3 internal signals removed from the table.
2011/07/22	1.50	Replaced CFGFIFO chapter: 22.4.4.6 FFCfg Bit0 FFEnO0 - FFEnO7 Add the description. 22.5.2 Register Setting FFCfg is added to Register Initial Setting and Transfer Start/Stop.
2011/03/10	1.40	Added GPIO5/& unused pin handling. Corrected typos in register description 'SWReset0 - SWReset7'. Removed all references to SPI Flash, updated Chip Info register values. Added boot sequences with reference to APIX usecase configurations.
2010/11/09	1.30	Numerous corrections, removing obsolete pins, renaming according to current pin list, updated block diagrams (e.g. SMC and EXTIRC). Most important changes are in the following chapters: Preface, Overview, APIX, GPIO, EXTINT, PPG, ADC, Electrical Characteristics.
2010/10/19	1.20	Update for ES2 + CS version of chip. Mainly effects pinning + SMCs/GPIOs. Corrected LSB of CID bitfield in Chip Information register.
2010/07/29	1.10	Corrected typo in section 20.3.8.1 PMM4[1:0]. Modified ADC chapter to reflect reduced number of channels.
2010/07/27	1.00	First version.

Contents

1. Overview

This chapter describes the hardware features, block diagram and function of the MB88F333 Graphics Display Controller (GDC) device.

The MB88F333 is a 180nm embedded flash CMOS device specified for automotive requirements. Its functions are optimized for modularized in-car-applications such as dashboards, HUD systems, CID (Central Information Display) and RSE (Rear Seat Entertainment) systems. The complete graphics processing works line based, so no expensive framebuffer memory is necessary. The device is also well-suited to other application fields.
Additionally a set of 5V capable peripherals such as SMC, PWMs, ADCs is integrated. This allows the realization of very competitive systems.

1.1. Features

MB88F333 has the following features:

- CMOS 180nm flash technology
- Package: LQFP176
- Ambient temperature range: -40…+105℃
- Power-supply voltage: (IO: 5.0V±0.5V, 3.3 ± 0.3V, core: 1.8 ± 0.15V)
- 83.3MHz System Clock
- CPU Interface: Synchronous serial I/F (SPI) or APIX sideband link
- supported sprite color depth 1,2,4,8bpp(indirect) or 16bpp(direct), 24bpp(direct, only internal)
- Up to 16,777,216 colors (internal processing only)
- 4/8bpp Alpha Plane (256-level alpha blending)
- Up to 512 sprites, including 32 special sprites
- 160KB Embedded NOR-flash ROM
- 128KB+8KB embedded SRAM (used by Sprite Engine)
- Automatically animated sprites (Command List Auto-Load, Special Sprite Function)
- Line buffer Method (no need for external VRAM)
- Maximum display resolution supported: 1280x480@60Hz (max. 42MHz pixel clock)
- Up to 18bpp TFT Panel (RSDS or TTL)
- Embedded TCON (Timing controller)
- APIX RX Interface (according to device purchased)
  - Pixel Link and Side Band Link
- Spread Spectrum Clock Modulation (for reduced EMI)
- Signature calculation Function (can be used to reach a required ASIL level – Automotive Safety Integrity level)
- Dither and Gamma Correction Unit (CLUT)
- Data expansion for run-length encoded data
- MCU Peripherals
  - 2 Stepper Motor Controllers
  - Up to 20 PPGs (PWMs, 4 synchronized with pixel clock, 8 synchronized with system clock, 8 synchronized and paired as 4x2) (shared pins)
  - 9 channel ADC (shared pins)
  - I2C
  - 2 channel USART(UART/SPI)
  - Sound Generator Module
  - GPIOs (shared pins)
  - External interrupt input

1.2. Limitations

- Do not execute burst accesses that cross any slave boundary.
- The host interface (HOST_SCK) can be operated at 21MHz or less.
- HOST_SCK must adhere to the clock phase and timing rules shown in "4.4.2.1.Host Interface (clock timing and phase)".
  Normal communication operation is not possible if these rules are not kept to.
- Please input the same voltage as VD5 to the signal of HOST Interface (HOST_DI, HOST_SCK, HOST_XCS). If this is not possible, then please refer to the Level Shifter Application Note on the Indigo website: Application Note (Indigo Host Interface Level Shifter 5V)
  http://www.fujitsu.com/emea/services/microelectronics/gdc/gdcdevices/mb88f332-indigo.html#support
- **Important Note:**
  For for first version E1 (Engineering Samples) of the GDC
  (Part Number MB88F333AAPMC-ESE2) please note that VD5 MUST be connected to 5V.

1.3. Comparison: Indigo / Indigo-L

	Indigo	Indigo-L
General		
Technology	180nm	
Graphic Core	Sprite engine	
Internal flash / RAM	160kB/136kB	
Display output	TCON, RGB/RSDS , dithering, gamma correction, signature unit	
System clock	83.3 MHz, SSCG	
Video input	APIX Rx 1.0	
Temperature Range	-40…+105°C	
Package	QFP 208	QFP 176
Quality Grade	AECQ100	
Modules/Peripherals (*1)		
I^2C	1	
TCON	1	
UART/SPI	2	
Host IF	1	1 (w/o interrupt output)
PWM	16 (4 sync'ed with pixel clock, 12 sync'ed with core clock)	Up to 20 (4 sync'ed with pixel clock, 8 synced & paired as 4 x 2, 8 sync'ed with core clock)
ADC	12	9
SMC	6	2 (see notes)
GPIO	20	15 + up to 8 GPOs
Ext. flash IF	1	0
Interrupt input	8	5
Reload Timer	16 internally (8 of 16 connected to pins)	16 internally (5 of 16 connected to pins)
APIX CFG[0..2]	Possible via external cfg	Internally hardwired to "000" (see notes)

(*1) Table shows maximum number of implemented peripherals. As pins are shared among the peripherals it can not be used the maximum number for all peripherals at the same time.

Notes: Pin Assignment Table 1 contains references to several pins named 'SMC_*' . These can currently be used to drive four stepper motors (although not recommended). The current pin names reflect their intermediate function for ES1 however not for the final version of the chip. Chip version ES2 will have finalized pin assignments for the stepper motor control pins and other PWMs. The final product is not designed to drive four stepper motors. Please refer to Pin Assignment Table 2 for the final assignment.

Please bear in mind that the hardwired configuration equivocates to the 125Mbit mode described in the Hardware Manual. This enables the user to program an initially empty device in the fastest possible mode. Of course it is possible to use any configuration for the APIX interface by programming and using a customized boot sequence in the internal flash.

1.4. Block Diagram

Figure 1-1 shows the block diagram of MB88F333.

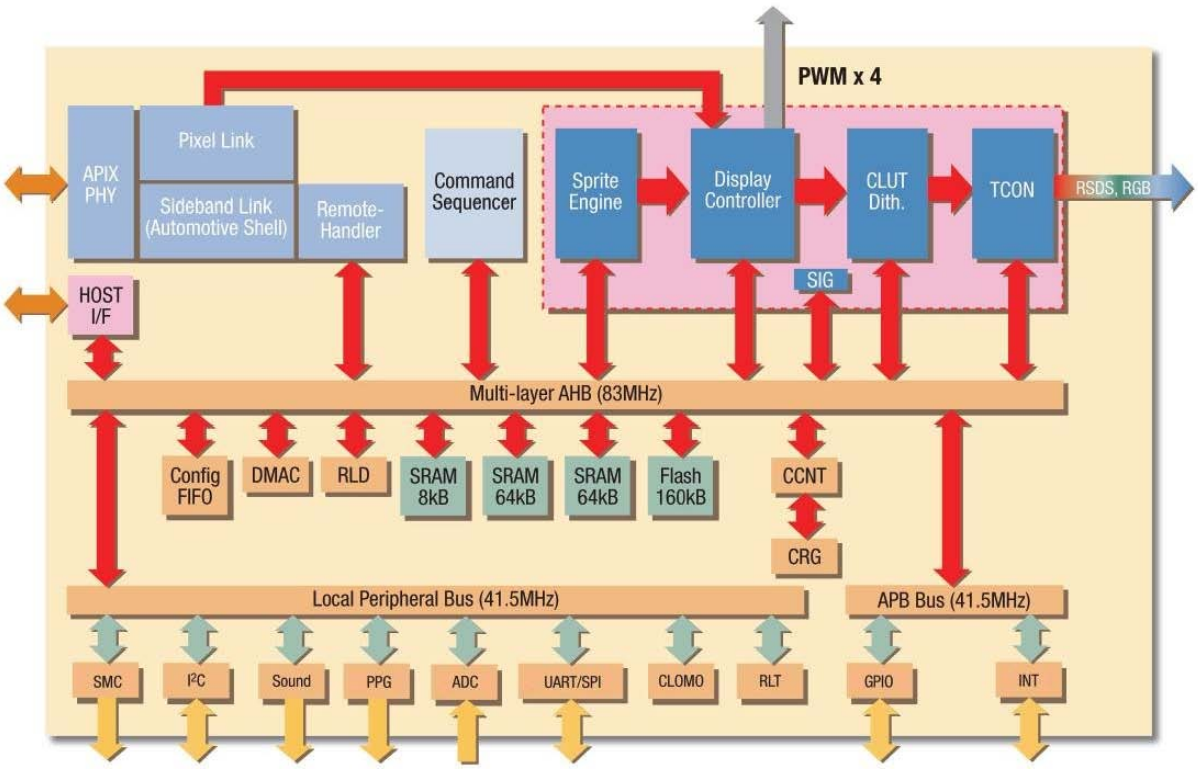


Figure 1-1 Block diagram of MB88F333

AHB bus

The following modules are connected to the AHB bus:

- SRAM: General purpose internal SRAM 64KB × 2 and 8KB × 1
- Embedded Flash Memory: General purpose internal Flash 160KB
- RLD: Run length decompression
- Sprite Engine: GDC register part
- DMAC: General purpose DMA × 2 channels
- Video Display control (VDC): Display control register part
- SIG: Signature generator register part
- TCON: Timing controller register part
- CLUT: Color look up table register part
- Dith: Dithering register part
- Command Sequencer (CMDSEQ)
- Configuration FIFO (Config FIFO)
- Host Interface
- Remote Handler
- CCNT: Chip Control module
- Clock modulator (Clock Modulation/Spread Spectrum)

APB bus

The following modules are connected to the APB bus:

- GPIO × 15 channels (+ up to 8 GPOs)
- External interrupt input

Local Peripheral bus

The following modules are connected to the RBUS bus:

- I^2C controller × 1 channel
- USART (UART/SPI) controller × 2 channels (both are shared)
- SMC × 2 channels
- ADC × 9 channels (3 channels are shared)
- PPG controller (PPG) Up to 20:
  - ✧ 4 sync'ed with pixel clock VPWM_O[3:0],
  - ✧ 8 synced & paired as 4 x 2 SPWM_O[7:0],
  - ✧ 8 sync'ed with core clock PWM_O[11:4]
- Sound generator
- RLT (Reload timer): 16 internally (5 of 16 connected to pins)

1.5. Function list

Function list of MB88F333 is shown below.

Function	Outline
Host IF	• Supports communication to a host CPU with an SPI interface • The length of the SPI interface packets is variable to permit the use of variable length addresses and data accesses • Supports writes/reads to the internal module connected to the AHB (variable, from 1 to 16 bytes) • Conforms to Freescale Semiconductor's advocacy SPI (CPOL=0, CPHA=0) • Corresponds to the speed of general purpose CPUs (set the frequency of SPICLK to 1/2 or less of the HCLK frequency) • Host CPU handshaking communication makes software flow control possible • The MB88F333 can only operate in slave mode whereas the host CPU is the bus master • The packet sizes must be in 8 bit units • No CRC error detection functionality. • No automatic procedure for resends in the case of errors (no ARQ functionality) • Supported burst transfer modes for the AHB are single, incr or incr4 • Correct operation cannot be guaranteed if simultaneous access to the module occurs from the APIX-LINK unit • No interrupt output
APIX interface	• The APIX® PHY provides a high speed serial downstream link and a low speed serial upstream link compliant to the Inova APIX® Industrial Standard. The downstream link transports video or generic data and side-band data. The upstream link transports side-band data only.
Remote Handler	• Interfacing to automotive shell • Unwrapping of messages received by automotive shell and translating them to AHB-master transactions and vice versa • Decoding of command-type field, address translation (address offset) • Evaluation of automotive shell status bits to detect APIX link status • Provide automotive shell configuration and status bits to system bus via AHB-slave interface • Provide Apix PHY configuration and status bits to system bus via AHB-slave interface • Interfacing to AHB-system Bus • AHB-write master with FIFO (depth: 16 messages) • AHB-read master with FIFO (depth: 16 messages) • Arbitration between interrupt read and normal read transactions on AHB-master • Read sequencer for 128 interrupt sources • Building of interrupt messages • Automatic clear of interrupt request flag (programmable) • Locking and unlocking of AHB-write master

Function	Outline

CMD Sequencer	• Processes command lists
	• This module can be started as follows.
	  • Start by power-on reset release
	  • Start by trigger signal
	  • Start by register setting
	• Built in Buffer (32 bit, 16 word)
	• It is possible to forced termination while processing the command list.
	• Detecting the following error ends the processing of the command list.
	  • Mismatch of expected value by COMPREG command
	  • Slave module access error
	• Two kinds of interrupt signals are output.
	  • Interrupt to processing of command list
	  • Interrupt to time-out of WDT

Sprite GDC	• Up to 512 Sprites • 32 Special Sprites for Auto-Animation • Up to 16,777,216 colors • Two 8bpp Color Palette Table (ARGB-8888) • 1bpp, 2bpp, 4bpp, 8bpp indirect color format • ARGB-1555, RGB-565, ARGB-8888 direct color format • Color format of each sprite can be set independently • Alpha-blending (4/8bpp alpha) • Image Reversing Function (horizontal or vertical)
Display controller	• The internal processing format of MB88F333 is RGB888 (however output is dithered to 6-bits) • Display unit internal output RGB24 (24bit/pixel) Digital image output. • Scanning method progressive scan. • Image size Supports common standards, e.g. 320×120 320×160 320×240(QVGA) 400×240(WQVGA) 480×240 500×250 640×160 640×240 640×480(VGA) 800×480(WVGA) 960×160 1280×480@60Hz (max. 42MHz pixel clock) • There are four display layers (LA, LB, background, foreground). • Display layer (Background, Foreground) sets the color using this module. • Display layer LA is one screen display layer from the sprite engine. • Display layer LB is one screen display layer from the external source. • Display Layer (LA, LB, Foreground) has the transparent color. • Display Layer (LA, LB, Foreground) is for the blend function. • Display Layer (LA, LB) can switch overlapping. • Interlace scanning is not possible. • The size of the Display Layer becomes a set image size.

Function	Outline
TCON	• RBM (RSDS Bit Mapping) Conforms to RSDS™ Standard 1.0 (National Semiconductors) Support for single bus (Multidrop bus with single or double end termination) Mapping for 6bit color depth Mapping for 8bit color depth (However RGB output is 6bit color depth) Data and clock outputs can flexible be assigned to the pool of available pins to ease board design • TSIG (Timing Signal Generator) Freely programmable waveforms 12 pulse generators 1 signal sequencer with max.64 signal transitions 12 signal mixers with a programmable function table Inversion control signal for transition minimizing (useful for TTL applications) • IO module Control of Combined TTL/RSDS IO cells Output RSDS clock Output TTL clock 90° phase shift Adjustable differential swing
CLUT	• Single block table with 256 entries and 10 bit accuracy for each color with optional index mode • Parallel programming of the table content • Direct mapping to Configuration Address space • Bypass
Dithering	• Bypass mode • Spatial dithering mode • Temporal dithering mode • Output resolution of 666/565 • Align the output data in the upper or lower part of the byte in order to display the output pixels on an RGB666 monitor
SIG	• Generation of 2 different picture signatures for each color channel summation of color values CRC-32 over color values • Programmable evaluation window position and size • Programmable evaluation window mask • Automatic monitoring using reference signature registers • Interrupt generation • Programmable picture source • Self restoring error counter
Built-in Flash memory	• Flash memory sector architecture: 2x64Kbyte sectors + 4x 8Kbyte • Flash memory program/erase endurance: 10,000 • CPU Mode: Access to the flash memory via the AHB is supported as follows: Write：Sync mode (16bit) Read：Sync mode (32bit) • The access timing can be set at every the cycle of the system clock.(CPU Mode) • Test Mode: Support Flash access from External I/F is as follows. Write：Sync mode (8bit/16bit) , Async mode (8bit/16bit) Read：Sync mode (8bit/16bit) , Async mode (8bit/16bit) • CPU Mode: Access timing can be set with each system clock cycle • Writer Mode: Supports flash access from the external I/F is as follows. Write：Async mode (8bit/16bit) Read：Async mode (8bit/16bit) • The start margin from the falling edge FRSTX to the completion of the reset operation is 20us or more when the flash programmed using the automatic programming algorithm. • FRSTX pulse width is 500ns or more. The start margin from the rising edge of FRSTX to read mode is 200ns or more.
Built-in SRAM	• Embedded general purpose SRAM of 64KB × 2 (32 bit bus) and 8KB × 1 (32bit bus)

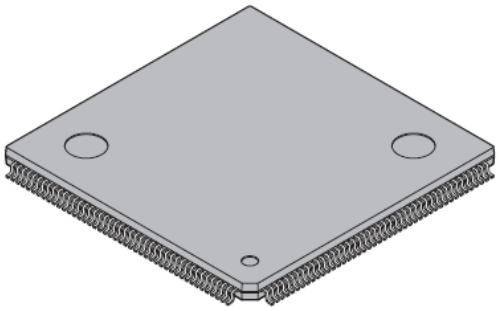
Function	Outline
RLD	• Supports simple run-length compression format (TGA™ similar format) • 1/2/4/8/16/24/32 bit per pixel formats supported • AHB master for data output • FIFO for data input and output, allows burst access of AHB
DMAC	• 2 DMA Channels • Software request (start-up by a register write) • Byte transfer 16 word FIFO shared by all channels Supports INCR, INCR 4/8/16, and WRAP 4/8/16. • Transfer mode Block transfer Burst transfer • 4 bit block register and 16 bit count register are set by programming • Supports 8, 16, and 32 bit transfer widths • Supports increment and fixed addressing to source and destination • Reload count, source address and destination address register • Issues error and completion interrupts • Displays end code of DMA transfer • Hardware support for fixed priority and rotation priority • In fixed priority mode, channel 0 has the highest priority, and channel 1 has the lowest priority
CRG	The ClkSynth module has the following features. • input frequency: 250 MHz • output frequency: 1 MHz to 90 MHz • N divider of range 2…255 results in 253 nominal frequencies which can be selected • P divider of range 0…4 results in 5 different frequencies to be selected between each nominal frequency • subP divider of range 0…255 to specify mean frequency between P values • “Frequency hopping” of 4 different frequencies (in range of delta P and delta subP) for Spread Spectrum Clocking; with delta P and delta subP a subset of 3 additional frequencies can be setup; • period of frequency hopping is configurable The clock control module has the following features. • Determination of the dividing frequency for the APB and the local peripheral bus clocks. • The modulated clock can be output to the AHB and APB bus clocks. • Reset for an internal module is generated from power-on reset. • After PLL clock is steady, reset of an internal module is released. The stability period of PLL is 1ms after releasing power-on reset. • Control of the ON/OFF modes of the clock and GDC internal module resets.
CLOMO	• The clock modulator is implemented for the reduction of electromagnetic interference - EMI, by spreading the spectrum of the clock signal over a wide range of frequencies.

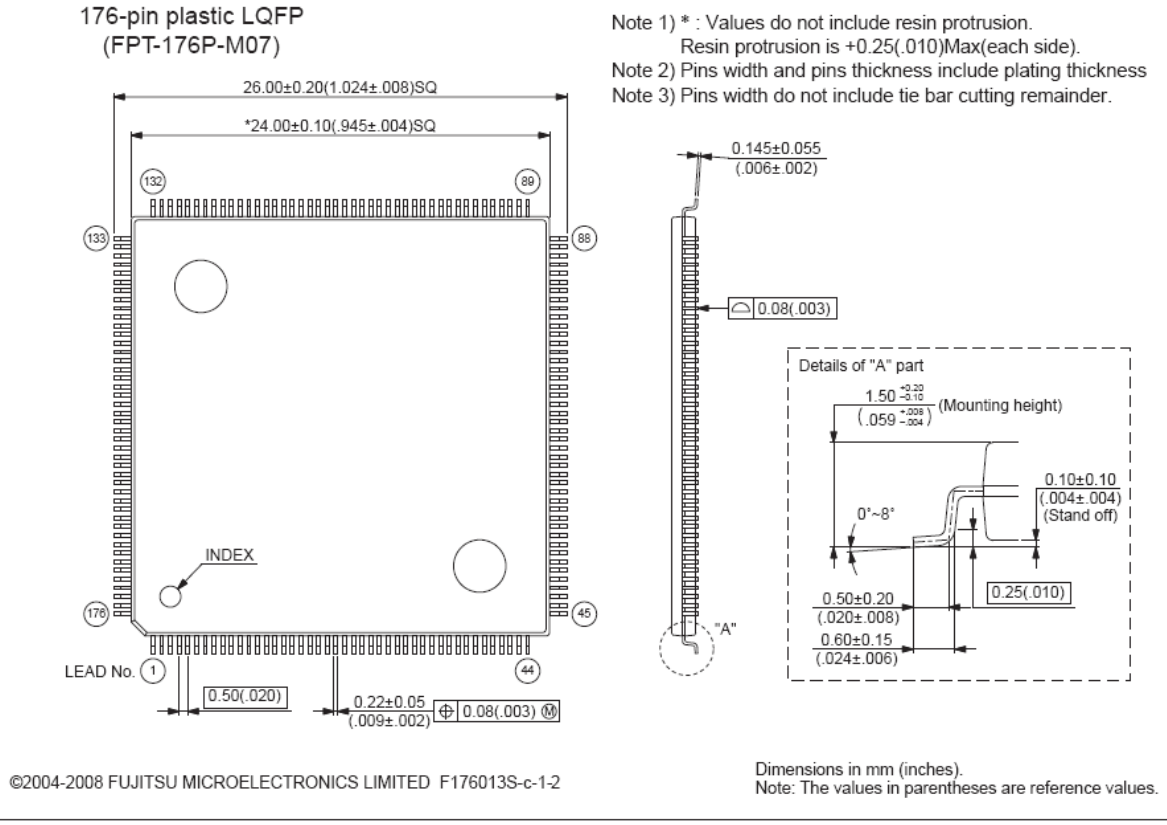
Function	Outline
CCNT	• Chip Information Used to read the revision information, the production year and chip name information of this chip. • Internal interrupt control This module has a function to report internal interrupt information to an external point. • Clock control The clock division register is used to change the dividing clock ratio and to select the modulated clock. Unnecessary clock output is controlled and energy-saving is possible by setting the clock enable register. • Soft reset interface Each module can be initialized by issuing a soft reset to each module in the register. • Data swap control of the byte lane of each module. • Pin switch External pin multiplexing selection. • OSC control • Apix interface termination resistor calibration control
Configuration FIFO	• 8 configurable FIFOs • Usage of one shared memory • Depth of each FIFO is configurable • FIFO upper and lower threshold interrupts • AHB slave interface for FIFO data input, and the Register is write/read. • AHB-master interface for FIFO data output (Only write is supported.) • Trigger input for each FIFO output • Simple local DMA functionality at data output programmable target address diff. addressing modes (including, fixed) • Memory mapped access to 2KByte SRAM • Interrupt is cleared by ISTS (Interrupt Status register) writing, or clear signal from external module (Remote Handler).
SMC	• The stepper motor controller consists of PWM pulse generators, motor drivers and selector logic circuits. The two motor drivers have a high-output driving capability and two motor coils can be connected directly to four pins. The motor rotation is designed to be controlled by a combination of the PWM pulse generators and selector logic circuits. The synchronization mechanism enables the synchronous operation of two PWM pulse generators.
SOUND	• The Sound Generator consists of the Sound Control register, Frequency Data register, Amplitude Data register, Decrement Grade register, Tone Count register, Sound Disable register, PWM pulse generator, Frequency counter, Decrement counter and Tone Pulse counter.

Function	Outline
I^2C	• Master/slave transmitting and receiving functions • Arbitration function • Clock synchronization function • General call addressing support • Transfer direction detection function • Repeated start condition generation and detection function • Bus error detection function • 7 bit addressing as master and slave • 10 bit addressing as master and slave • Possibility to give the interface a seven *and* a ten bit slave address • Acknowledge upon slave address reception can be disabled (master only operation) • Address masking to give interface several slave addresses (in 7 and 10 bit mode) • Up to 400 Kbit/s transfer rate • Possibility to use built-in noise filters for SDA and SCL • Can receive data at 400 KBit/s if the R-Bus-Clock is higher than 6MHz, regardless of prescaler setting • Can generate MCU interrupts on transmission and bus error events • Supports being slowed down (deceleration) by a slave at a bit and byte level
USART	• A general-purpose serial data communication interface for performing synchronous or asynchronous communication with external devices. 16 bytes transmission and reception FIFOs are available for selected channels.
GPIO/GPO	• 15 GPIO + up to 8 x GPO ports • Consists of following three registers Port Data register (PDR) • Data Direction register (DDR)
EXTINT	• Bus slave complied with AMBA (APB). • Four interrupt request channels. • Four selectable request levels: high level, low level, rising edge, and falling edge. Provide a request to the main interrupt controller when returning from the STOP mode.
PPG (PWM)	• Output waveforms • One-shot waveforms • Clamped output • Up to 20: ▪ 4x VPWM synchronized with pixel clock) ▪ 8x SPWM synchronized + paired (4x2) ▪ 8x PWM synchronized
A/D converters	• Conversion time: 3 µs per channel • RC type successive approximation conversion with sample and hold circuit • 10bit or 8 bit resolution • Program section analog input from 9 channels • Single conversion mode: conversion of one selected channel • Scan conversion mode: continuous conversion of multiple channels, programmable for up to 9 channels
RLT	• One-shot operation • Reload operation • 16bit down counter with reload register

1.6. Package dimensions

The package dimensions of the MB88F333 are shown below.

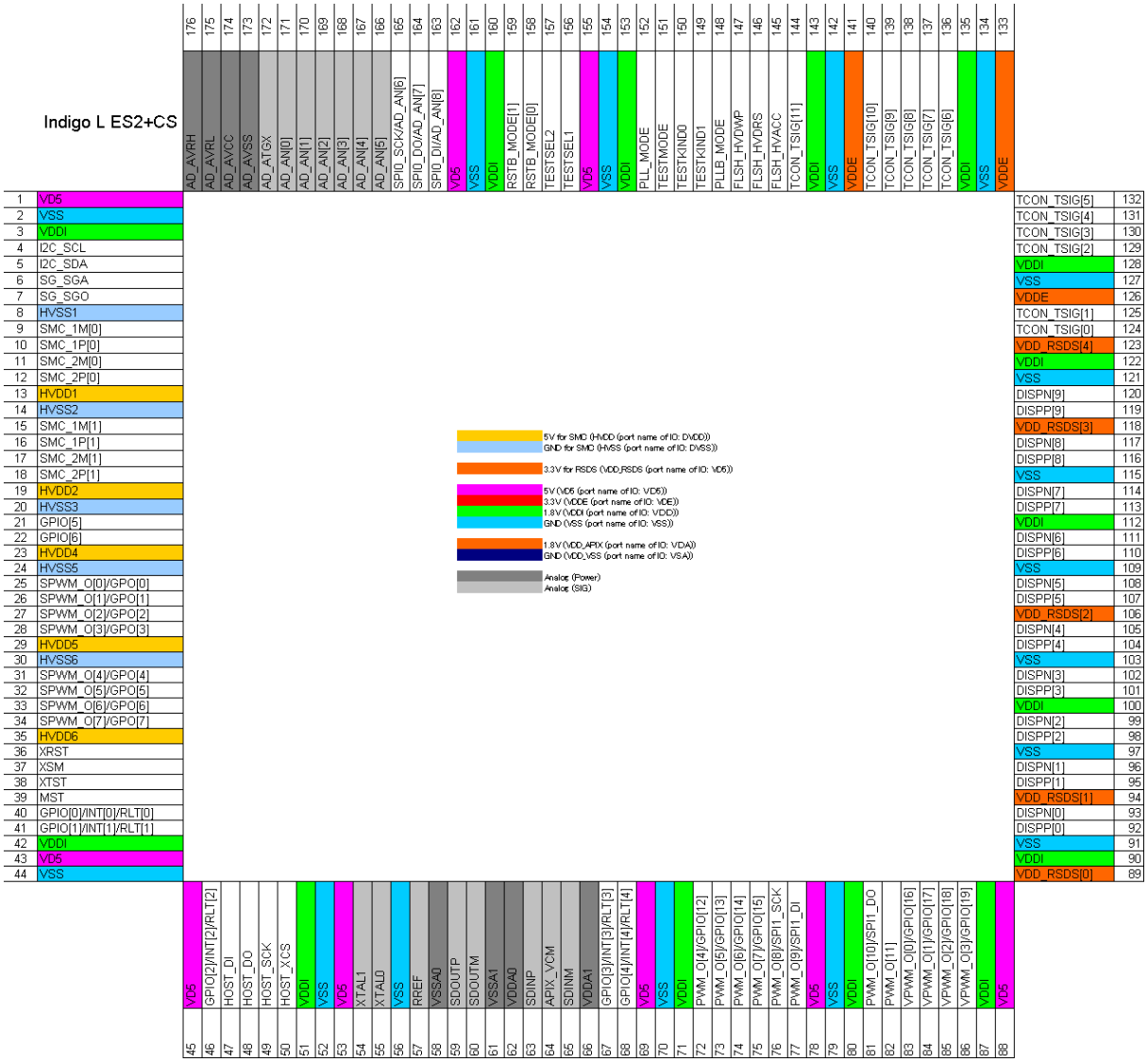
176-pin plastic LQFP		
 (FPT-176P-M07)	Lead pitch	0.50 mm
	Package width × package length	24.0 × 24.0 mm
	Lead shape	Gullwing
	Sealing method	Plastic mold
	Mounting height	1.70 mm MAX
	Code (Reference)	P-LQFP-0176-2424-0.50



Please also refer to:
http://edevice.fujitsu.com/system/mbynavi/package/en/search/

1.7. Pin assignment (version ES2 + CS)

The pin assignment of MB88F333 ES2 + CS is shown below. Please note that the pinning for version ES1 was slightly different.



1.8. Pin assignment table (chip version ES2 + CS)

Changes to chip version ES1 are marked in bold text.

Pin No.	Pin name	Pin No.	Pin name	Pin No.	Pin name	Pin No.	Pin name
1	VD5	45	VD5	89	VDD_RSDS[0]	133	VDDE
2	VSS	46	GPIO[2]/INT[2]/RLT[2]	90	VDDI	134	VSS
3	VDDI	47	HOST_DI	91	VSS	135	VDDI
4	I2C_SCL	48	HOST_DO	92	DISPP[0]	136	TCON_TSIG[6]
5	I2C_SDA	49	HOST_SCK	93	DISPN[0]	137	TCON_TSIG[7]
6	SG_SGA	50	HOST_XCS	94	VDD_RSDS[1]	138	TCON_TSIG[8]
7	SG_SGO	51	VDDI	95	DISPP[1]	139	TCON_TSIG[9]
8	HVSS1	52	VSS	96	DISPN[1]	140	TCON_TSIG[10]
9	SMC_1M[0]	53	VD5	97	VSS	141	VDDE
10	SMC_1P[0]	54	XTAL1	98	DISPP[2]	142	VSS
11	SMC_2M[0]	55	XTAL0	99	DISPN[2]	143	VDDI
12	SMC_2P[0]	56	VSS	100	VDDI	144	TCON_TSIG[11]
13	HVDD1	57	RREF	101	DISPP[3]	145	FLSH_HVACC
14	HVSS2	58	VSSA0	102	DISPN[3]	146	FLSH_HVDRS
15	SMC_1M[1]	59	SDOUTP	103	VSS	147	FLSH_HVDWP
16	SMC_1P[1]	60	SDOUTM	104	DISPP[4]	148	PLLB_MODE
17	SMC_2M[1]	61	VSSA1	105	DISPN[4]	149	TESTKIND1
18	SMC_2P[1]	62	VDDA0	106	VDD_RSDS[2]	150	TESTKIND0
19	HVDD2	63	SDINP	107	DISPP[5]	151	TESTMODE
20	HVSS3	64	APIX_VCM	108	DISPN[5]	152	PLL_MODE
21	**GPIO[5]**	65	SDINM	109	VSS	153	VDDI
22	**GPIO[6]**	66	VDDA1	110	DISPP[6]	154	VSS
23	HVDD4	67	GPIO[3]/INT[3]/RLT[3]	111	DISPN[6]	155	VD5
24	HVSS5	68	GPIO[4]/INT[4]/RLT[4]	112	VDDI	156	TESTSEL1
25	**SPWM_O[0]/GPO[0]**	69	VD5	113	DISPP[7]	157	TESTSEL2
26	**SPWM_O[1]/GPO[1]**	70	VSS	114	DISPN[7]	158	RSTB_MODE[0]
27	**SPWM_O[2]/GPO[2]**	71	VDDI	115	VSS	159	RSTB_MODE[1]
28	**SPWM_O[3]/GPO[3]**	72	PWM_O[4]/GPIO[12]	116	DISPP[8]	160	VDDI
29	HVDD5	73	PWM_O[5]/GPIO[13]	117	DISPN[8]	161	VSS
30	HVSS6	74	PWM_O[6]/GPIO[14]	118	VDD_RSDS[3]	162	VD5
31	**SPWM_O[4]/GPO[4]**	75	PWM_O[7]/GPIO[15]	119	DISPP[9]	163	SPI0_DI/AD_AN[8]
32	**SPWM_O[5]/GPO[5]**	76	PWM_O[8]/SPI1_SCK	120	DISPN[9]	164	SPI0_DO/AD_AN[7]
33	**SPWM_O[6]/GPO[6]**	77	PWM_O[9]/SPI1_DI	121	VSS	165	SPI0_SCK/AD_AN[6]
34	**SPWM_O[7]/GPO[7]**	78	VD5	122	VDDI	166	AD_AN[5]
35	HVDD6	79	VSS	123	VDD_RSDS[4]	167	AD_AN[4]
36	XRST	80	VDDI	124	TCON_TSIG[0]	168	AD_AN[3]
37	XSM	81	PWM_O[10]/SPI1_DO	125	TCON_TSIG[1]	169	AD_AN[2]
38	XTST	82	PWM_O[11]	126	VDDE	170	AD_AN[1]
39	MST	**83**	**VPWM_O[0]/GPIO[16]**	127	VSS	171	AD_AN[0]
40	GPIO[0]/INT[0]/RLT[0]	**84**	**VPWM_O[1]/GPIO[17]**	128	VDDI	172	AD_ATGX
41	GPIO[1]/INT[1]/RLT[1]	**85**	**VPWM_O[2]/GPIO[18]**	129	TCON_TSIG[2]	173	AD_AVSS
42	VDDI	**86**	**VPWM_O[3]/GPIO[19]**	130	TCON_TSIG[3]	174	AD_AVCC
43	VD5	87	VDDI	131	TCON_TSIG[4]	175	AD_AVRL
44	VSS	88	VD5	132	TCON_TSIG[5]	176	AD_AVRH

1.9. Pin Functions

The external pin functions of MB88F333 are described below.

1.9.1. Pin Multiplexing

The MB88F333 deviceuses pin multiplexing technology. A part of the chip's functionality is realized using external pin multiplexing.
There are five functional pin groups, as described below. Each pin's functionality depends on its multiplex group and the selected multiplex mode, making the application of the device and I/O's very flexible.

1. Pin multiplex group #0 (Set register : CCNT-PMM0[1:0])
   - “00” : Function none (the pin is in input state)
   - “01” : VPWM output
   - “1x” : GPIO inout

2. Pin multiplex group #1 (Set register : CCNT-PMM1[1:0])
   - “00” : Function none (the pin is in input state)
   - “01” : PWM output
   - “1x” : Signals related to SPI

3. Pin multiplex group #2 (Set register : CCNT-PMM2[1:0])
   - “00” : Function none (the pin is in input state)
   - “01” : PWM output
   - “10” : GPIO inout
   - “11” : Function none (the pin is in input state)

4. Pin multiplex group #3 (Set register : CCNT-PMM3[1:0])
   - “00” : Function none (the pin is in input state)
   - “01” : GPIO inout
   - “10” : External Interrupt input
   - “11” : External Trigger input for Reload Timer

5. Pin multiplex group #4 (Set register : CCNT-PMM4[1:0])
   - “00” : Function none (the pin is in input state)
   - “01” : Signals related to SPI
   - “1x” : Analog input for ADC

Note :
Please only change the multiplex mode when the function of effected pins are not in use!

1.9.1.1. Pin multiplex group #0

The mode of “Pin multiplex group #0” mode is set by VPMM0 [1:0] register of the CCNT module. This VPWM output (VPWM _O[3:0]) is synchronized with the 'pixel clk' signal.

Pin No.	PMM0 [1:0]					
	00 (Default value)		01		1x	
86	Function none	output	VPWM_O[3]	output	GPIO[19]	inout
85	Function none	output	VPWM_O[2]	output	GPIO[18]	inout
84	Function none	output	VPWM_O[1]	output	GPIO[17]	inout
83	Function none	output	VPWM_O[0]	output	GPIO[16]	inout

1.9.1.2. Pin multiplex group #1

The mode of “Pin multiplex group #1” mode is set by PMM1 [1:0] register of the CCNT module.

Pin No.	PMM1[1:0]					
	00 (Default value)		01		1x	
82	Function none	input	PWM_O[11]	output	Function none	input
81	Function none	input	PWM_O[10]	output	SPI1_DO	output
77	Function none	input	PWM_O[9]	output	SPI1_DI	input
76	Function none	input	PWM_O[8]	output	SPI1_SCK	inout

1.9.1.3. Pin multiplex group #2

The mode of “Pin multiplex group #2” mode is set by PMM2 [1:0] register of the CCNT module.

Pin No.	PMM2[1:0]					
	00 (Default value) / 11		01		10	
75	Function none	input	PWM_O[7]	output	GPIO[15]	inout
74	Function none	input	PWM_O[6]	output	GPIO[14]	inout
73	Function none	input	PWM_O[5]	output	GPIO[13]	inout
72	Function none	input	PWM_O[4]	output	GPIO[12]	inout

1.9.1.4. Pin multiplex group #3

The mode of “Pin multiplex group #3” mode is set by PMM3 [1:0] register of the CCNT module.

Pin No.	PMM3[1:0]							
	00 (Default value)		01		10		11	
68	Function none	input	GPIO[4]	inout	INT[4]	input	RLT[4]	input
67	Function none	input	GPIO[3]	inout	INT[3]	input	RLT[3]	input
46	Function none	input	GPIO[2]	inout	INT[2]	input	RLT[2]	input
41	Function none	input	GPIO[1]	inout	INT[1]	input	RLT[1]	input
40	Function none	input	GPIO[0]	inout	INT[0]	input	RLT[0]	input

1.9.1.5. Pin multiplex group #4

The mode of “Pin multiplex group #4” mode is set by PMM4 [1:0] register of the CCNT module.

Pin No.	PMM4[1:0]					
	00 (Default value)		01		1x	
163	Function none	input	SPI0_DI	input	AD_AN[8]	input
164	Function none	input	SPI0_DO	output	AD_AN[7]	input
165	Function none	input	SPI0_SCK	inout	AD_AN[6]	input

1.9.2. Pin functions

Format

The pin function list is shown in the following format.

Pin name	I/O	Polarity	Analog /Digital	Type	Status of pin after reset	Description

Meaning of item and sign

Pin name

Name of external pin.

I/O

Input/Output signals' distinction based on this LSI.

- I: Pin that can be used as input
- O: Pin that can be used as output
- IO: Pin that can be used as input and output (interactive pin)

Polarity

Active polarity of external pin's input/output signals

- P: "H" active pin (positive logic)
- N: "L" active pin (negative logic)
- PN: "H" and "L" active pins

Analog/Digital

Signal type of external pin

- A: Analog signal
- D: Digital signal

Type

Input/Output circuit type of external pin.

- CLK: Clock
- POD: Pseudo Open Drain
- PU: Pull Up
- PD: Pull Down
- ST: Schmitt Type
- Tri: Tri-state

Pin status after reset

Pin status after external pin reset

- H: "H" level
- L: "L" level
- HiZ: High impedance
- X: "H" level or "L" level
- A: Clock output

Description

Description of the external pin's functionality.

1.9.2.1. Host interface related pins

Pin name	I/O	Polarity	Analog /Digital	Type	Status of pin after reset	Description
HOST_XCS	I	N	D	-	-	SPI Chip select
HOST_DO	O	-	D	-	L	SPI Receive Data
HOST_DI	I	-	D	-	-	SPI Transmit Data
HOST_SCK	I	P	D	CLK	-	SPI Clock

1.9.2.2. APIX I/F related pins

Pin name	I/O	Polarity	Analog /Digital	Type	Status of pin after reset	Description
SDOUTM	O	N	A	-	H	Serial Data Out n channel
SDOUTP	O	P	A	-	L	Serial Data Out p channel
SDINM	I	N	A	-	-	Serial Data In n channel
SDINP	I	P	A	-	-	Serial Data In p channel
APIX_VCM	I	-	A	-	-	Common Mode decoupling * Recommended value 100nF
RREF	I	-	A	-	-	Reference resistor for termination calibration *Recommended value 1kOhm,1% to APIX GND (please also refer to the APIX Layout Recommendations Application Note on the website: http://www.fujitsu.com/emea/services/microelectronics/gdc/gdcdevices/MB88F332-indigo.html#support (please note that the Application Note is for the Indigo GDC, but also applies to Indigo-L)
VDDA0	I	-	A	-	-	supply analog IO 1.8V
VSSA0	I	-	A	-	-	supply analog IO 0V (APIX GND)
VDDA1	I	-	A	-	-	supply analog core 1.8V
VSSA1	I	-	A	-	-	supply analog core 0V (APIX GND)

1.9.2.3. Oscillator I/O I/F related pins

Pin name	I/O	Polarity	Analog /Digital	Type	Status of pin after reset	Description
XTAL0	I	-	A	-	-	For Oscillator I/O
XTAL1	I	-	A	-	-	For Oscillator I/O

1.9.2.4. SMC related pins

Pin name	I/O	Polarity	Analog /Digital	Type	Status of pin after reset	Description
SMC_1P[1,0]	O	-	D	-	L	Stepper Motor Control 1P output
SMC_1M[1,0]	O	-	D	-	L	Stepper Motor Control 1M output
SMC_2P[1,0]	O	-	D	-	L	Stepper Motor Control 2P output
SMC_2M[1,0]	O	-	D	-	L	Stepper Motor Control 2M output

Pin name	I/O	Polarity	Analog /Digital	Type	Status of pin after reset	Description
HVDD1	I	-	A	-	-	High Current Supply
HVDD2	I	-	A	-	-	High Current Supply
HVDD4	I	-	A	-	-	High Current Supply
HVDD5	I	-	A	-	-	High Current Supply
HVDD6	I	-	A	-	-	High Current Supply
HVSS1	I	-	A	-	-	GND
HVSS2	I	-	A	-	-	GND
HVSS3	I	-	A	-	-	GND
HVSS5	I	-	A	-	-	GND
HVSS6	I	-	A	-	-	GND

1.9.2.5. PWM related pins

Pin name	I/O	Polarity	Analog /Digital	Type	Status of pin after reset	Description
PWM_O[15:12]	O	-	D	-	HiZ	PWM output (sync to pixel clk) GPIO[19:16] and PWM[15:12] are multiplexed.
PWM_O[11:8]	O	-	D	-	HiZ	PWM output (sync to internal clk) SPI(SPI_*_1) and PWM[10:8] are multiplexed.
PWM_O[7:4]	O	-	D	-	HiZ	PWM output (sync to internal clk) GPIO[15:12] and PWM[7:4] are multiplexed.
SPWM_O[4]/GPO[4]	O	-	D	-	L	SPWM output pair
SPWM_O[5]/GPO[5]	O	-	D	-	L	SPWM output pair
SPWM_O[6]/GPO[6]	O	-	D	-	L	SPWM output pair
SPWM_O[7]/GPO[7]	O	-	D	-	L	SPWM output pair

1.9.2.6. GPIO related pins

Pin name	I/O	Polarity	Analog /Digital	Type	Status of pin after reset	Explanation
GPIO[19:12, 5:0]	IO	-	D	-	HiZ	General purpose I/O port GPIO[4:0] , INT[4:0] and RLT[4:0] are multiplexed. GPIO[19:16] and VPWM[3:0] and GPIO[15:12] and PWM_O[7:4] are multiplexed.

1.9.2.7. External Interrupt related pins

Pin name	I/O	Polarity	Analog /Digital	Type	Status of pin after reset	Explanation
INT[4:0]	I	-	D	-	HiZ	External Interrupt. GPIO[4:0] , INT[4:0] and RLT[4:0] are multiplexed.

1.9.2.8. Reload timer related pins

Pin name	I/O	Polarity	Analog /Digital	Type	Status of pin after reset	Explanation
RLT[4:0]	I	-	D	-	HiZ	External Trigger for Reload Timer. GPIO[4:0] , INT[4:0] and RLT[4:0] are multiplexed.

1.9.2.9. USART related pins

Pin name	I/O	Polarity	Analog /Digital	Type	Status of pin after reset	Description
SPI0_DO	O	-	D	-	L	USART Receive Data SPI0 DO and AD AN[10] are multiplexed.
SPI0_DI	I	-	D	-	-	USART Transmit Data SPI0 DI and AD AN[11] are multiplexed.
SPI0_SCK	IO	-	D	-	-	USART Clock SPI0 SCK and AD AN[9] are multiplexed.
SPI1_DO	O	-	D	-	L	USART Receive Data SPI1_DO and PWM[10] are multiplexed.
SPI1_DI	I	-	D	-	-	USART Transmit Data SPI1_DI and PWM[9] are multiplexed.
SPI1_SCK	IO	-	D	-	-	USART Clock SPI1_SCK and PWM[8] are multiplexed.

1.9.2.10. ADC related pins

Pin name	I/O	Polarity	Analog /Digital	Type	Status of pin after reset	Explanation
AD_AN[11:9, 5:0]	I	-	A	-	-	ADC input AD_AN[8] and SPI0_DI are multiplexed AD_AN[7] and SPI0_DO are multiplexed AD_AN[6] and SPI0_SCK are multiplexed
AD_ATGX	I	-	D	-	HiZ	Trigger
AD_AVCC	I	-	A	-	-	Analog power Supply
AD_AVRH	I	-	A	-	-	Reference (Voltage H)
AD_AVRL	I	-	A	-	-	Reference (Voltage L)
AD_AVSS	I	-	A	-	-	GND

1.9.2.11. I^2C related pins

Pin name	I/O	Polarity	Analog /Digital	Type	Status of pin after reset	Explanation
I2C_SCL	IO	-	D	POD	HiZ	I^2C Clock
I2C_SDA	IO	-	D	POD	HiZ	I^2C Data

1.9.2.12. SOUND related pins

Pin name	I/O	Polarity	Analog /Digital	Type	Status of pin after reset	Explanation
SG_SGA	O	-	D	-	L	Sound SGA
SG_SGO	O	-	D	-	L	Sound SGO

1.9.2.13. Panel Interface related pins

Pin name	I/O	Polarity	Analog /Digital	Type	Status of pin after reset	Explanation
DISPP[9:0]	O	-	D	-	HiZ	RSDS Digital RGB Output
DISPN[9:0]	O	-	D	-	HiZ	RSDS Digital RGB Output

1.9.2.14. Supply Panel Interface related pins

Pin name	I/O	Polarity	Analog /Digital	Type	Status of pin after reset	Explanation
VDD_RSDS[4:0]	I	-	A	-	-	supply

1.9.2.15. TCON related pins

Pin name	I/O	Polarity	Analog /Digital	Type	Status of pin after reset	Explanation
TCON_TSIG[11:0]	O	-	D	-	HiZ	TCON timing signal

1.9.2.16. System related pins

Pin name	I/O	Polarity	Analog /Digital	Type	Status of pin after reset	Explanation
XRST	I	N	D	ST	-	System Reset
PLL_MODE	I	-	D	ST	-	PLL mode (0: 250MHz / 1: 500MHz)
RSTB_MODE[0]	I	-	D	ST	-	RST bypass mode (0:Normal reset time / 1: Long reset time) See **Note**
RSTB_MODE[1]	I	-	D	ST	-	RST bypass mode (0:Internal / 1: External) See **Note**
TESTMODE	I	-	D	ST	-	Test mode (0:Normal/ 1: Test mode)
TESTKIND0	I	-	D	ST	-	Test mode pin (reserved) (0:Normal/ 1: Test mode) See **Note**
TESTKIND1	I	-	D	ST	-	Test mode pin (reserved) (0:Normal/ 1: Test mode) See **Note**
TESTSEL1	I	-	D	ST	-	Test mode pin (reserved) (0:Normal/ 1: Test mode) See **Note**
TESTSEL2	I	-	D	ST	-	Test mode pin (reserved) (0:Normal/ 1: Test mode) See **Note**
PLLB_MODE	I	-	D	ST	-	PLL bypass mode (0:Normal / 1: Bypass)
XSM	I	-	D	ST	-	Scan mode (0:Scan mode / 1: Normal mode)
XTST	I	-	D	ST	-	Test mode pin (reserved) (0:Normal/ 1: Test mode) See **Note**

Pin name	I/O	Polarity	Analog /Digital	Type	Status of pin after reset	Explanation
MST	I	-	D	ST	-	Test mode pin (reserved) (0:Normal/ 1: Test mode) See **Note**
FLSH_HVACC	I	-	D	-	-	Test mode pin (for Embedded Flash) (0:Normal/ 1: Test mode)
FLSH_HVDRS	I	-	D	-	-	Test mode pin (for Embedded Flash) (0:Normal/ 1: Test mode)
FLSH_HVDWP	I	-	D	-	-	Test mode pin (for Embedded Flash) (0:Normal/ 1: Test mode)

1.9.2.17. Power supply related pins

Pin name	I/O	Polarity	Analog /Digital	Type	Status of pin after reset	Description
VSS	I	-	D	-	-	Ground
VDDE	I	-	D	-	-	External pin power supply (3.3V)
VD5	I	-	D	-	-	External pin power supply (5.0V)
VDDI	I	-	D	-	-	Internal power supply (1.8V)

Note :
The following modes can be selected by the state of the RSTB_MODE[1:0] pins.

1. RSTB_MODE[1:0] : 00
   Reset is handled by embedded reset unit. ResetTime=2^15×XTAL clock.

2. RSTB_MODE[1:0] : 01
   Reset is handled by the embedded reset unit. ResetTime=2^16×XTAL clock.

3. RSTB_MODE[1:0] : 1x
   Reset is handled via the XRST pin.
   Please input XRST ="0"(Reset state) as the initial value when you use this mode.
   Otherwise, there is a possibility that the device will malfunctioning.

   Please note that if you are using the XRST solution, the power sequence can be freely selected and that the minimum wait time for a power-on after a power off (1 ms) also no longer applies.

4. Test mode pins
   Please refer to 'the Unused Pins' section

1.9.3. Unused Pins

Please treat pins as shown below when they are not used.

1.9.3.1. I2C I/F

Pin No.	Pin name	Handling when unused
4	I2C_SCL	Please use a high-resistance pull-up (VD5) or pull-down (VSS).
5	I2C_SDA	Please use a high-resistance pull-up (VD5) or pull-down (VSS).

Note: High-resistance: recommends 4.7k Ohm.

1.9.3.2. Sound I/F

Pin No.	Pin name	Handling when **unused**
6	SG_SGA	This pin can be left floating (open).
7	SG_SGO	This pin can be left floating (open).

1.9.3.3. SMC I/F

Pin No.	Pin name	Handling when **unused**
9	SMC_1M[0]	This pin can be left floating (open).
10	SMC_1P[0]	This pin can be left floating (open).
11	SMC_2M[0]	This pin can be left floating (open).
12	SMC_2P[0]	This pin can be left floating (open).
15	SMC_1M[1]	This pin can be left floating (open).
16	SMC_1P[1]	This pin can be left floating (open).
17	SMC_2M[1]	This pin can be left floating (open).
18	SMC_2P[1]	This pin can be left floating (open).
25	SPWM_O[0]	This pin can be left floating (open).
26	SPWM_O[1]	This pin can be left floating (open).
27	SPWM_O[2]	This pin can be left floating (open).
28	SPWM_O[3]	This pin can be left floating (open).
31	SPWM_O[4]	This pin can be left floating (open).
32	SPWM_O[5]	This pin can be left floating (open).
33	SPWM_O[6]	This pin can be left floating (open).
34	SPWM_O[7]	This pin can be left floating (open).

Note: If the complete SMC/SPWM block is not used, then the HVDDx pins can be connected to GND (see also the 'Electrical Characteristics' chapter).

1.9.3.4. GPIO/INT/RLT I/F

Pin No.	Pin name	Handling when **of unused**
21	GPIO[5]	Please use a high-resistance pull-up (VD5) or pull-down (VSS).
22	GPIO[6]	Please use a high-resistance pull-up (VD5) or pull-down (VSS).
40	GPIO[0]/INT[0]/RLT[0]	Please use a high-resistance pull-up (VD5) or pull-down (VSS).
41	GPIO[1]/INT[1]/RLT[1]	Please use a high-resistance pull-up (VD5) or pull-down (VSS).
46	GPIO[2]/INT[2]/RLT[2]	Please use a high-resistance pull-up (VD5) or pull-down (VSS).
67	GPIO[3]/INT[3]/RLT[3]	Please use a high-resistance pull-up (VD5) or pull-down (VSS).
68	GPIO[4]/INT[4]/RLT[4]	Please use a high-resistance pull-up (VD5) or pull-down (VSS).

Note: High-resistance: recommends 4.7k Ohm.

1.9.3.5. HOST I/F

Pin No.	Pin name	Handling when **unused**
47	HOST_DI	Please use a high-resistance pull-up (VD5) or pull-down (VSS).
48	HOST_DO	This pin can be left floating (open).
49	HOST_SCK	Please use a high-resistance pull-up (VD5) or pull-down (VSS).
50	HOST_XCS	Please use a high-resistance pull-up (VD5) or pull-down (VSS).

Note: High-resistance: recommends 4.7k Ohm.

1.9.3.6. USART I/F

Pin No.	Pin name	Handling when **unused**
163	SPI0_DI/AD_AN[8]	Please use a high-resistance pull-up (VD5) or pull-down (VSS).
164	SPI0_DO/AD_AN[7]	Please use a high-resistance pull-up (VD5) or pull-down (VSS).
165	SPI0_SCK/AD_AN[6]	Please use a high-resistance pull-up (VD5) or pull-down (VSS).

Note: High-resistance: recommends 4.7k Ohm.

1.9.3.7. APIX I/F

The APIX Interface can handle APIX capable chips and APIX uncapable chips alike.

Pin No.	Pin name	Handling when unused
59	SDOUTP	This pin can be left floating (open).
60	SDOUTM	This pin can be left floating (open).
63	SDINP	This pin can be left floating (open).
65	SDINM	This pin can be left floating (open).
57	RREF	This pin can be left floating (open).
64	APIX_VCM	This pin can be left floating (open).

Note: High-resistance: recommended 4.7k Ohm.

1.9.3.8. Oscillator I/O I/F

Pin No.	Pin name	Handling when **unused**

54	XTAL1	This pin can be left floating (open).
55	XTAL0	Please connect this pin with VSS.

1.9.3.9. Display I/F

Pin No.	Pin name	Handling when **unused**
92	DISPP[0]	This pin can be left floating (open).
93	DISPN[0]	This pin can be left floating (open).
95	DISPP[1]	This pin can be left floating (open).
96	DISPN[1]	This pin can be left floating (open).
98	DISPP[2]	This pin can be left floating (open).
99	DISPN[2]	This pin can be left floating (open).
101	DISPP[3]	This pin can be left floating (open).
102	DISPN[3]	This pin can be left floating (open).
104	DISPP[4]	This pin can be left floating (open).
105	DISPN[4]	This pin can be left floating (open).
107	DISPP[5]	This pin can be left floating (open).
108	DISPN[5]	This pin can be left floating (open).
110	DISPP[6]	This pin can be left floating (open).
111	DISPN[6]	This pin can be left floating (open).
113	DISPP[7]	This pin can be left floating (open).
114	DISPN[7]	This pin can be left floating (open).
116	DISPP[8]	This pin can be left floating (open).
117	DISPN[8]	This pin can be left floating (open).
119	DISPP[9]	This pin can be left floating (open).
120	DISPN[9]	This pin can be left floating (open).

1.9.3.10. TCON I/F

Pin No.	Pin name	Handling when **unused**
124	TCON_TSIG[0]	This pin can be left floating (open).
125	TCON_TSIG[1]	This pin can be left floating (open).
129	TCON_TSIG[2]	This pin can be left floating (open).
130	TCON_TSIG[3]	This pin can be left floating (open).
131	TCON_TSIG[4]	This pin can be left floating (open).
132	TCON_TSIG[5]	This pin can be left floating (open).
136	TCON_TSIG[6]	This pin can be left floating (open).
137	TCON_TSIG[7]	This pin can be left floating (open).
138	TCON_TSIG[8]	This pin can be left floating (open).
139	TCON_TSIG[9]	This pin can be left floating (open).
140	TCON_TSIG[10]	This pin can be left floating (open).
144	TCON_TSIG[11]	This pin can be left floating (open).

1.9.3.11. PWM - VPWM I/F

Pin No.	Pin name	Handling when **unused**
72	PWM_O[4]/GPIO[12]	Please use a high-resistance pull-up (VD5) or pull-down (VSS).
73	PWM_O[5]/GPIO[13]	Please use a high-resistance pull-up (VD5) or pull-down (VSS).

74	PWM_O[6]/GPIO[14]	Please use a high-resistance pull-up (VD5) or pull-down (VSS).
75	PWM_O[7]/GPIO[15]	Please use a high-resistance pull-up (VD5) or pull-down (VSS).
76	PWM_O[8]/SPI1_SCK	Please use a high-resistance pull-up (VD5) or pull-down (VSS).
77	PWM_O[9]/SPI1_DI	Please use a high-resistance pull-up (VD5) or pull-down (VSS).
81	PWM_O[10]/SPI1_DO	Please use a high-resistance pull-up (VD5) or pull-down (VSS).
82	PMW_O[11]	Please use a high-resistance pull-up (VD5) or pull-down (VSS).
83	VPWM_O[0]/GPIO[16]	Please use a high-resistance pull-up (VD5) or pull-down (VSS).
84	VPWM_O[1]/GPIO[17]	Please use a high-resistance pull-up (VD5) or pull-down (VSS).
85	VPWM_O[2]/GPIO[18]	Please use a high-resistance pull-up (VD5) or pull-down (VSS).
86	VPWM_O[3]/GPIO[19]	Please use a high-resistance pull-up (VD5) or pull-down (VSS).

Note: High-resistance: recommends 4.7k Ohm.

1.9.3.12. ADC I/F

Pin No.	Pin name	Handling when unused
163	SPI0_DI/AD_AN[8]	Please use a high-resistance pull-up (VD5) or pull-down (VSS).
164	SPI0_DO/AD_AN[7]	Please use a high-resistance pull-up (VD5) or pull-down (VSS).
165	SPI0_SCK/AD_AN[6]	Please use a high-resistance pull-up (VD5) or pull-down (VSS).
166	AD_AN[5]	Please connect this pin with VSS.
167	AD_AN[4]	Please connect this pin with VSS.
168	AD_AN[3]	Please connect this pin with VSS.
169	AD_AN[2]	Please connect this pin with VSS.
170	AD_AN[1]	Please connect this pin with VSS.
171	AD_AN[0]	Please connect this pin with VSS.
172	AD_ATGX	Please connect this pin with VSS.
173	AD_AVSS	Please connect this pin with VSS.
174	AD_AVCC	Please connect this pin with VSS.
175	AD_AVRL	Please connect this pin with VSS.
176	AD_AVRH	Please connect this pin with VSS.

1.9.3.13. System related pins

Pin No.	Pin name	Handling when unused
36	XRST	Please use a high-resistance pull-up (VD5).
37	XSM	Please use a high-resistance pull-up (VD5). (0:Scan mode / 1: Normal mode)
38	XTST	Please use a high-resistance pull-down (VSS). (0:Normal/ 1: Test mode)
39	MST	Please use a high-resistance pull-down (VSS). (0:Normal/ 1: Test mode)
145	FLSH_HVACC	Please use a high-resistance pull-down (VSS). (0:Normal/ 1: Test mode)
146	FLSH_HVDRS	Please use a high-resistance pull-down (VSS). (0:Normal/ 1: Test mode)
147	FLSH_HVDWP	Please use a high-resistance pull-down (VSS). (0:Normal/ 1: Test mode)
148	PLLB	n/a
149	TESTKIND1	Please use a high-resistance pull-down (VSS). (0:Normal/ 1: Test mode)
150	TESTKIND0	Please use a high-resistance pull-down (VSS). (0:Normal/ 1: Test mode)
151	TESTMODE	Please use a high-resistance pull-down (VSS). (0:Normal/ 1: Test mode)
152	PLL_MODE	This pin is a user setting. (0:PLL@250MHz / 1:PLL@500MHz)
156	TESTSEL1	Please use a high-resistance pull-down (VSS). (0:Normal/ 1: Test mode)
157	TESTSEL2	Please use a high-resistance pull-down (VSS). (0:Normal/ 1: Test mode)
158	RSTB_MODE[0]	This pin is a user setting. (0:Normal reset time / 1:Long reset time)
159	RSTB_MODE[1]	This pin is a user setting. (0:Inernal reset / 1:External reset)

Note: High-resistance: recommends 4.7k Ohm.

1.10. Clock Generation

The following diagram gives a top-level, generalized view of the clock generation system within the MB88F333 device.

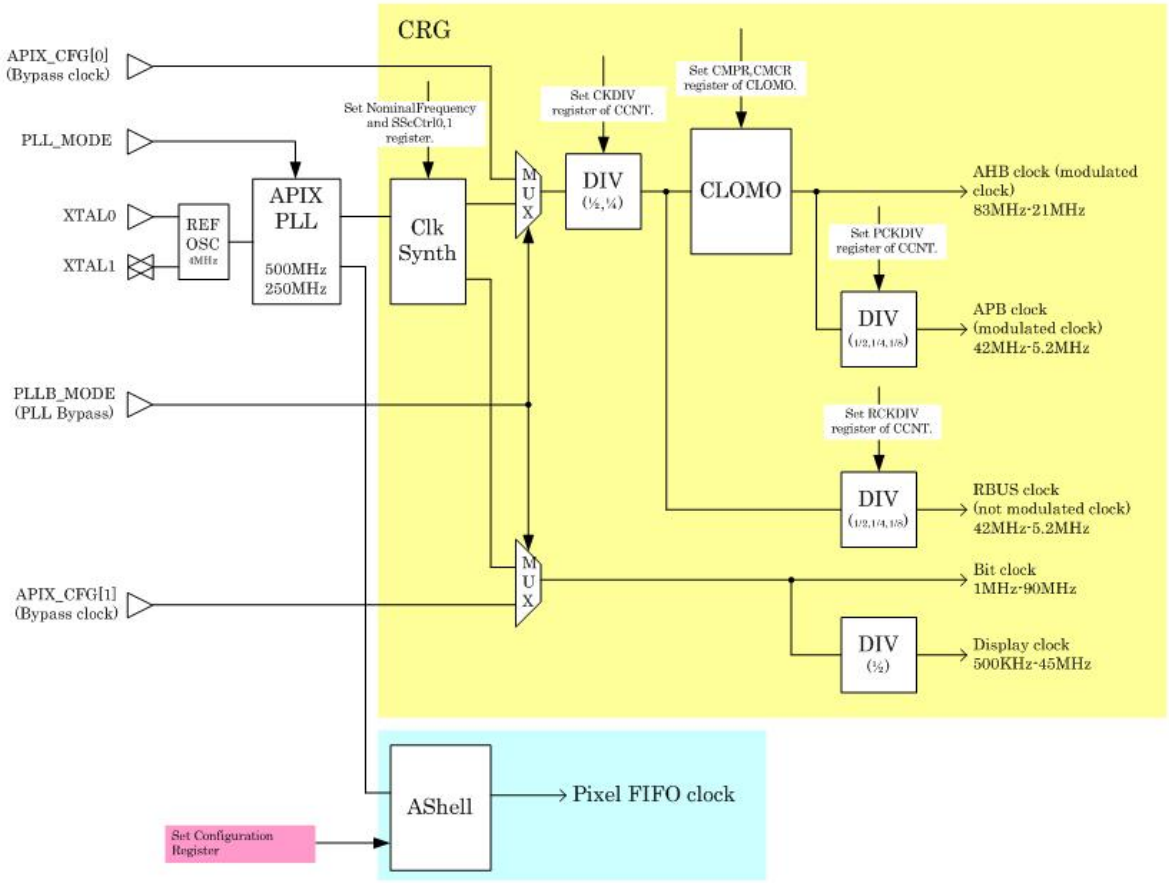


Figure 1-2 MB88F333 Indigo-L Clock Generation

2. System Configuration

Figure 2-1 shows an in-vehicle dashboard system configuration for which the MB88F333 LSI could be used.

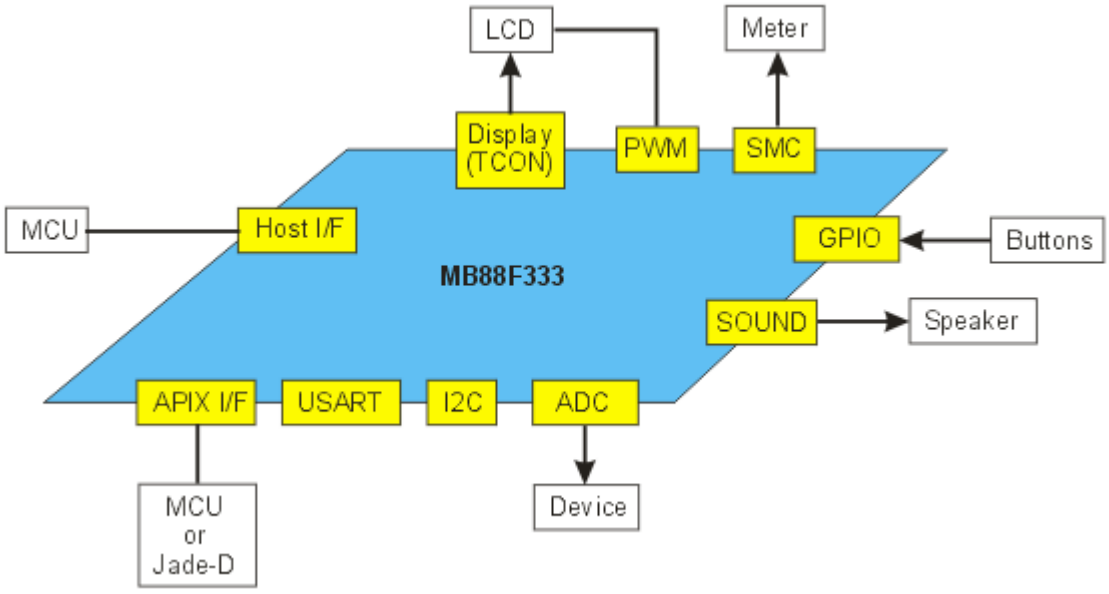


Figure 2-1 MB88F333 system configuration example

2.1. Chip Initialization

It is necessary to program the flash memory of MB88F333 'Indigo-L' using the initialization sequence included with the GDC Studio tool before using the chip for the first time.

In order to handle endianness issues, the ‘Indigo-L’ hardware contains a set of data swappers (which are documented in this specification and configured as part of Chip Control). Note that he initial values of data swappers and address inverters at ‘Indigo-L’ power-up time may not be suitable for every application and may need to be changed. This Hardware Manual is based on a setup that is described below. Access to halfword and byte registers by Host Interface (SPI), Command Sequencer or Configuration FIFO only work as described in this manual if the setup described below is used. The same applies to sub-register access to little-endian modules by the Remote Handler.
The following registers need to be set (in this order!) to set up the system properly:

Register Address Value
CCNT.ClockEnable 0x00010814 0xFFFFFFFF // enable all clocks
CCNT.DataSwapCtr1 0x00010820 0x00000000 // set up swappers
CCNT.DataSwapCtr2 0x00010824 0x00000000
CCNT.DataSwapCtr3 0x00010828 0x00000000
CCNT.DataSwapCtr4 0x0001082C 0x00000505
CCNT.0xE00 0x00010E00 0x5E5F5E76 // set up inverters
CCNT.0xE30 0x00010E30 0x00000000
CCNT.0xE00 0x00010E00 0x00000000

If you are using HwCom, GdcCom, or GI, this is done by HWComSystemSetup(). This function is called by GdcComOpen(), which in turn is called by GiOpen().
If you are not using HwCom, you should set the registers above as part of your reset command sequence stored in flash memory; The SETREG commands would be as follows:

02010000 00010814 FFFFFFFF
02040000 00010820 00000000 00000000 00000000 00000505
02010000 00010E00 5E5F5E76
02010000 00010E30 00000000
02010000 00010E00 00000000

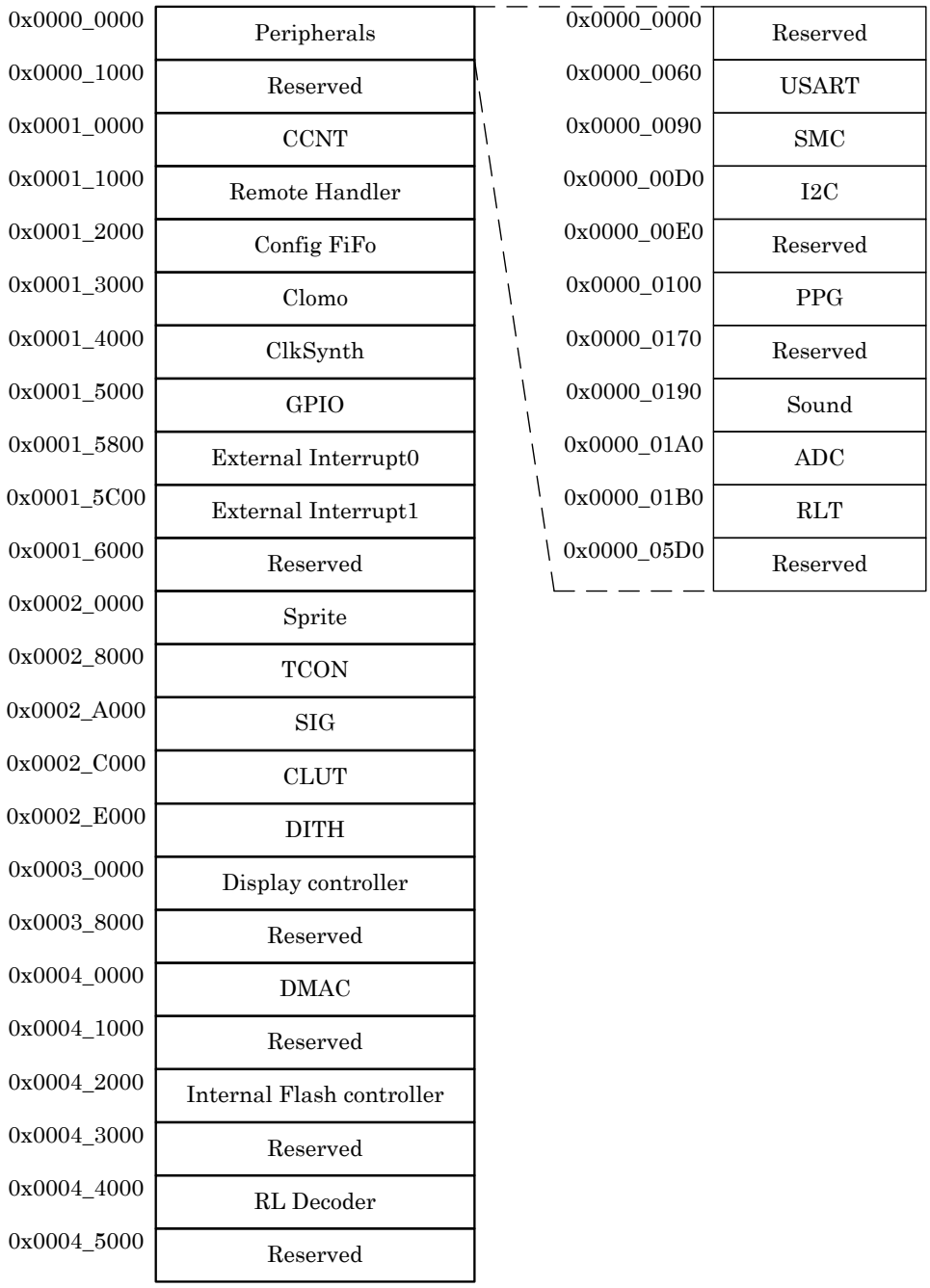
Your installation of Fujitsu GDC Studio should come with an indigo-system-setup.gdcseq register sequence which can be used as a starting point for generating this command sequence.

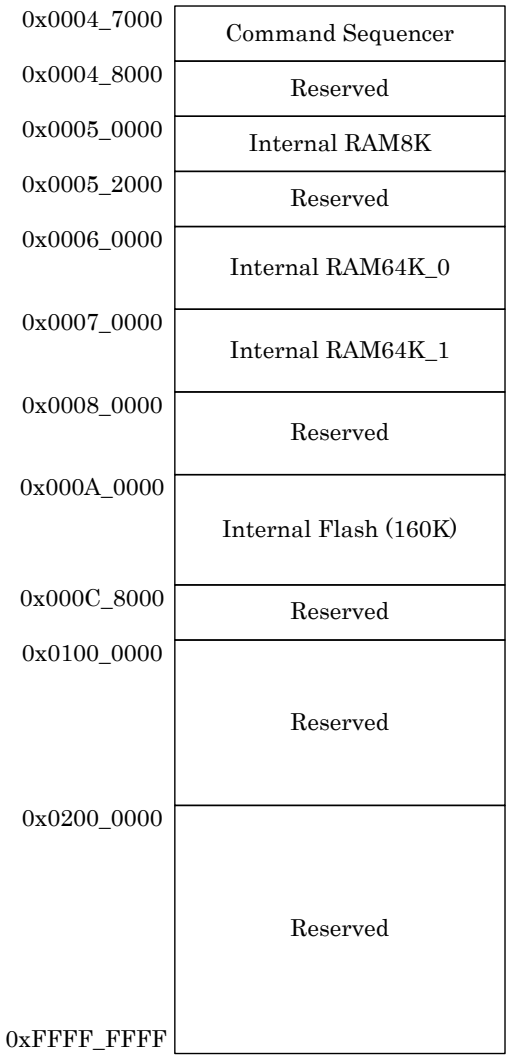
Further details are available in the Application Note (Indigo Byte Order) which is available here: http://www.fujitsu.com/emea/services/microelectronics/gdc/gdcdevices/MB88F332 device-indigo.html#support.

3. Memory map

This chapter shows the memory map and register map of the MB88F333.

3.1. LSI Memory map





Note: Do not use burst accesses that transgress any slave's boundaries.

4. Host Interface

This chapter describes the Host Interface of the MB88F333 LSI.

4.1. Outline

The Host Interface module is an internal module connected to the AHB which is used for communication to an external host CPU (which is connected via the SPI interface). The host CPU can read and write to the internal module. From a host CPU point of view, this module functions as a slave, whereas internally it functions as a master.

4.2. Features

The Host Interface has the features described in the following sections.

4.2.1. Features

Accesses by the host CPU to the internal module can be made using varying address bytes lengths within a range of 1 to 4 bytes. Additionally, the data byte length can be arbitrarily set within a range of 1 to 16 bytes. This means that the received number of bytes can be optimized and forwarding can be done efficiently. These settings can be specified by the CMD byte, allowing a highly flexible solution that abstracts the type of host CPU in use and the access objects.

- Supports communication to a host CPU with an SPI interface
- The length of the SPI interface packets is variable to permit the use of variable length addresses and data accesses
- Supports writes/reads to the internal module connected to the AHB (variable, from 1 to 16 bytes)
- Conforms to Freescale Semiconductor's advocacy SPI (CPOL=0, CPHA=0)
- Corresponds to the speed of general purpose CPUs (set the frequency of SPICLK to 1/2 or less of the HCLK frequency)
- Host CPU handshaking communication makes software flow control possible

4.2.2. Limitations

- The MB88F333 can only operate in slave mode whereby the host CPU is the bus master
- The packet sizes must be in 8 bit units
- No CRC error detection functionality
- No Automatic Repeat reQuest for resends in the case of errors (no ARQ functionality)
- Supported burst transfer modes for the AHB are single, incr or incr4
- Correct operation cannot be guaranteed if simultaneous access to the module occurs from the APIX-LINK unit

4.3. Function

4.3.1. Block Diagram

Figure 4-1 shows a block diagram of the host interface.

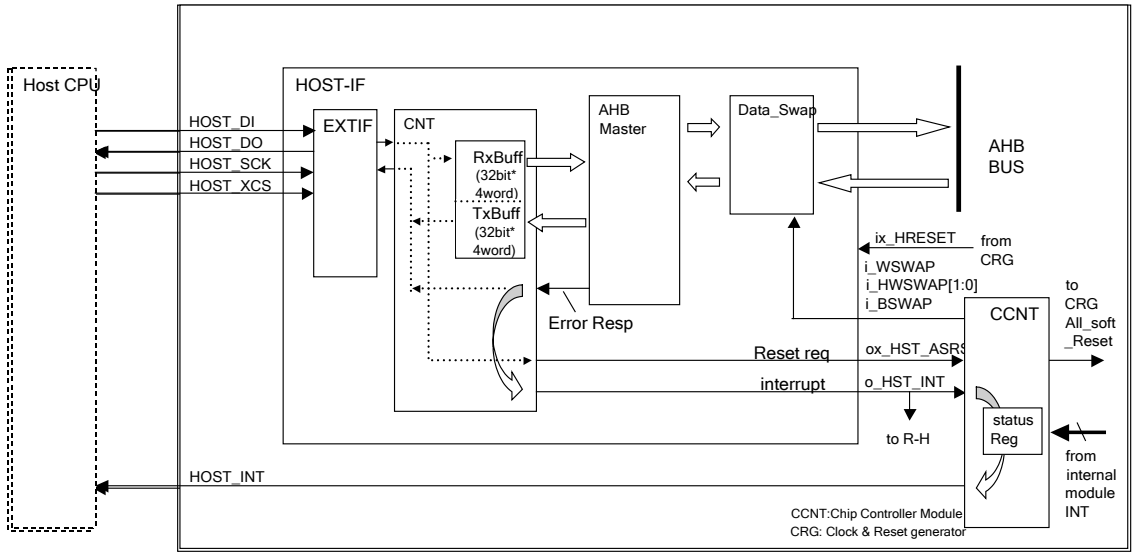


Figure 4-1 Host interface block diagram

4.3.2. SPI Interface

4.3.2.1. Write Access

Accesses from the host CPU to this module can arbitrarily use address byte lengths in a range of 1 to 4 bytes, as set. Also, the data byte length can be arbitrarily set in a range of 1 to 16 bytes.
This module provides a function to notify the host CPU with the result of write processing. It is necessary to send a dummy write CMD after a normal write CMD. The host CPU serial clock is maintained by sending dummy write CMDs. The result of write processing is sent with this clock. The basic format of a write access is shown below.

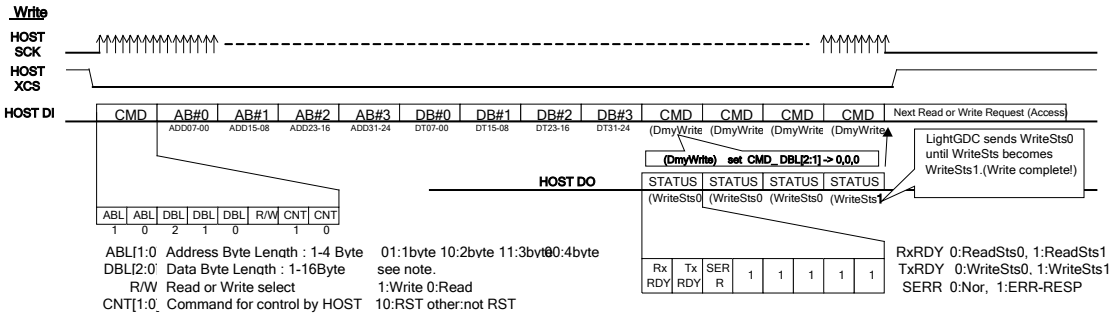


Figure 4-2 Write access

The CMD and STATUS bytes are described as follows:

CMD byte
ABL: Address Byte Length as shown by using a 2 bit code for 1 to 4 bytes
DBL: Data Byte Length as shown by using 3 bit code for 1 to 16 bytes

DBL2	DBL1	DBL0	Data Length
0	***0***	***0***	***Dummy Writes.***
0	0	1	1
0	1	0	2
0	1	1	4
1	0	0	8
1	0	1	12
1	1	0	16
1	1	1	16

AHB HSIZE	AHB HBURST
no access	***no access***
Byte	SINGLE
Half word	SINGLE
1 word	SINGLE
2 word	INCR
3 word	INCR
4 word	INCR4
4 word	INCR4

R/W: Specifies read or write. “1” is a write.

STATUS byte
The write status is shown by the TxRDY bit of the STATUS byte. When write processing is completed and the next transmission is possible, "1" is shown in the TxRDY bit.

The flow of a write action is shown below.

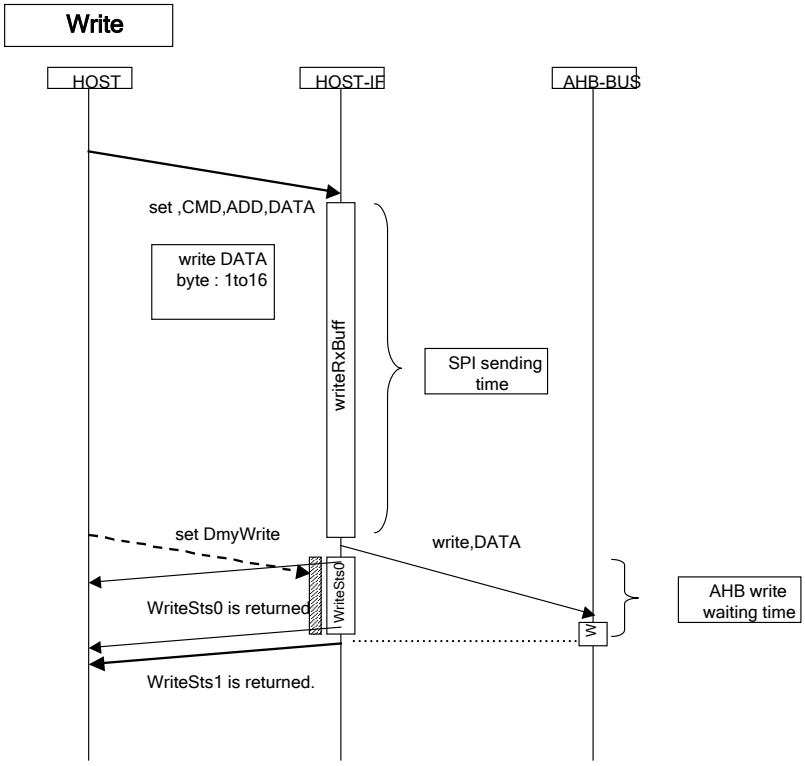


Figure 4-3 Write process flow

4.3.2.2. Read Access

Access from the host CPU to this module can be done using an arbitrarily set address byte length in a range of 1 to 4 bytes. In addition, the data byte length can be arbitrarily set in a range of 1 to 16 bytes.
This module adapts its read access actions accordingly by manipulating the wait time of the AHB bus. The wait and ready states for read accesses can be transmitted via a dummy write CMD which can be used as for a write action too. The basic format of a read access is shown below.

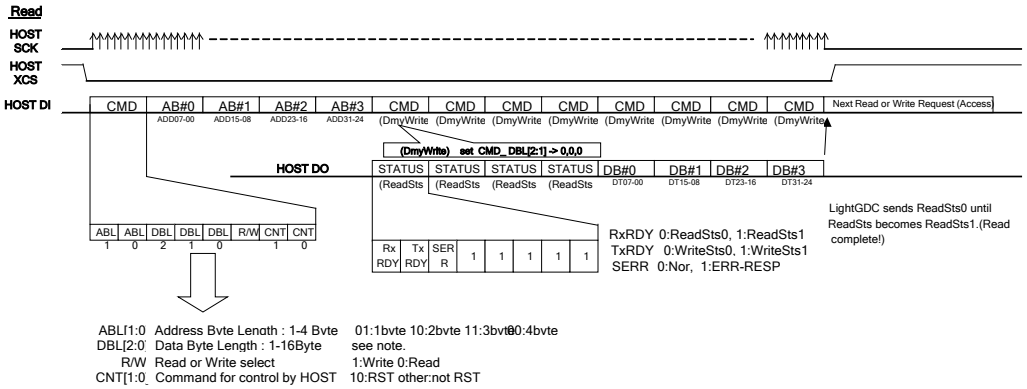


Figure 4-4 Read access

The CMD and STATUS bytes are described as follows:

CMD byte
ABL: Address Byte Length as shown by using a 2 bit code for 1 to 4 bytes
DBL: Data Byte Length as shown by using 3 bit code for 1 to 16 bytes

DBL2	DBL1	DBL0	Data Length
0	***0***	***0***	***Dummy Writes.***
0	0	1	1
0	1	0	2
0	1	1	4
1	0	0	8
1	0	1	12
1	1	0	16
1	1	1	16

AHB HSIZE	AHB HBURST
no access	***no access***
Byte	SINGLE
Half word	SINGLE
1 word	SINGLE
2 word	INCR
3 word	INCR
4 word	INCR4
4 word	INCR4

R/W: Specifies read or write. “0” is a read.

STATUS byte
The read status is shown by the RxRDY bit. When read processing is completed, "1" is shown in the RxRDY bit. The host CPU can retrieve the reading data at the correct time by monitoring the STATUS byte.

The flow of a read action is shown below.

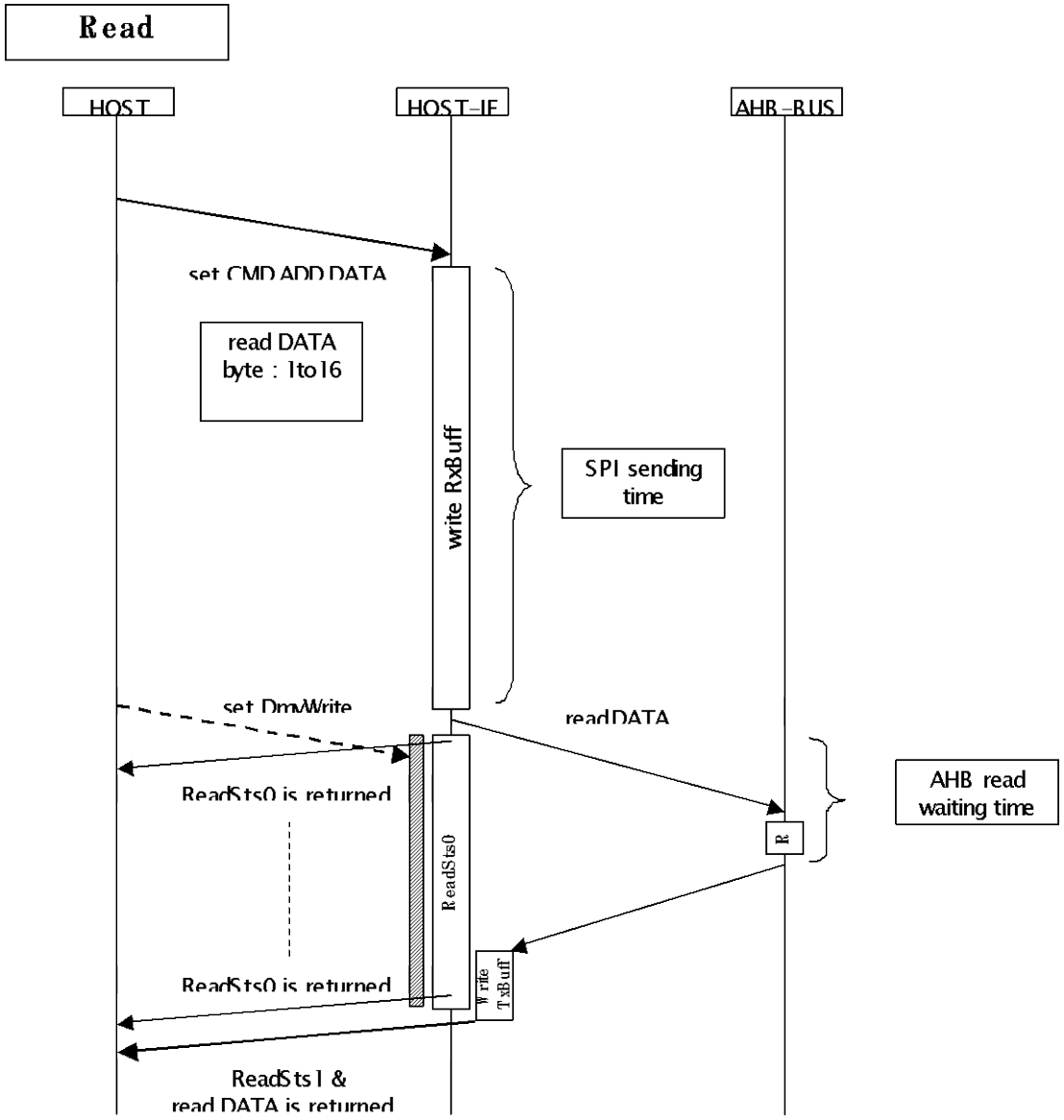


Figure 4-5 Read process flow

4.3.3. Interrupt

4.3.3.1. AHB slave module access error response

An error response from the AHB bus is output to the chip control module, CCNT. In addition, an error response is written to the STATUS byte and the host CPU is immediately notified. The RxRDY bit (or TxRDY bit) is set to '1' at the same time. The HOSTIF module itself does not have a register to maintain this information.

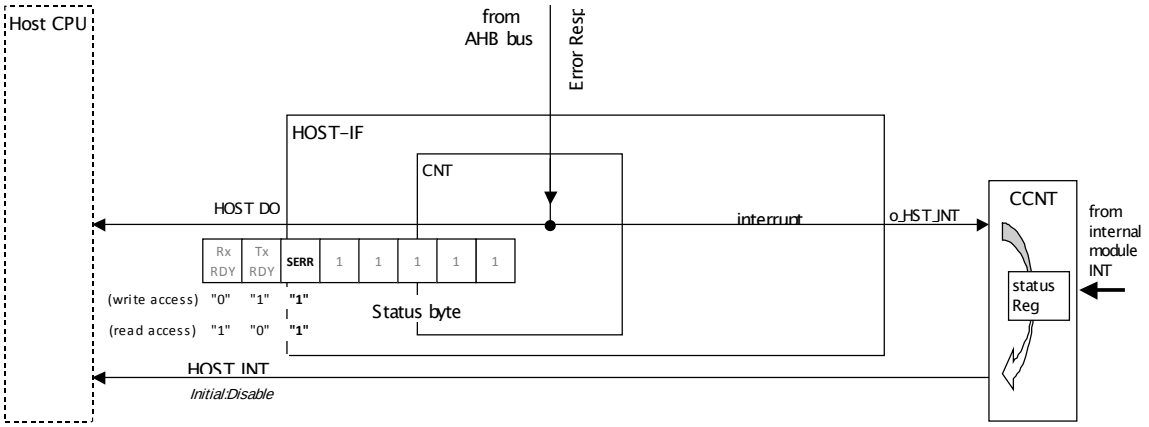


Figure 4-6 Interrupt

When an error response status has been sent to the host CPU, the transaction is completed. If the CCNT interrupt setting is enabled, an interrupt is generated.

4.3.4. Reset Request

A software reset of MB88F333 can be executed on request by the host CPU. If the normal operation of the MB88F333 device is no longer possible due to certain conditions, the host CPU can use the reset request. When a reset is executed, the MB88F333 is rebooted by the CRG unit.

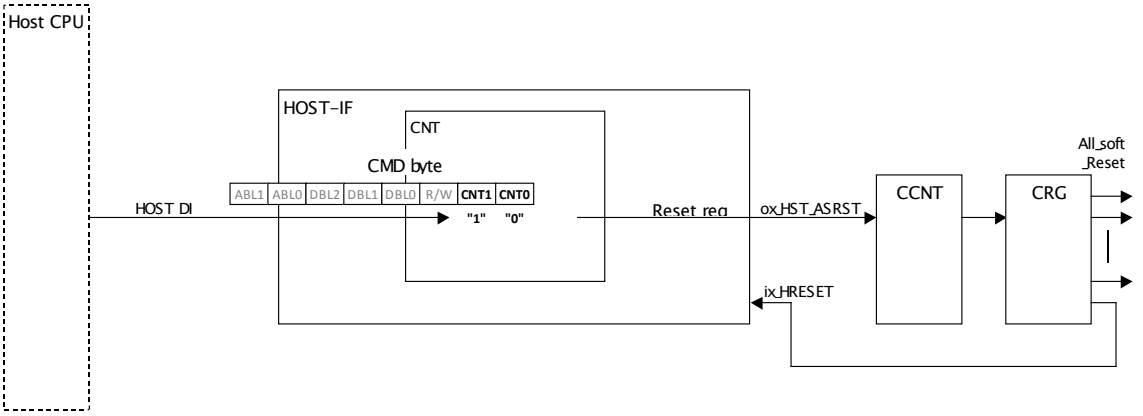


Figure 4-7 Reset request

4.4. External Interfaces

4.4.1. Communication Protocols (Timing Diagrams)

4.4.1.1. SPI protocol stack

The SPI communication protocol stack is shown below.

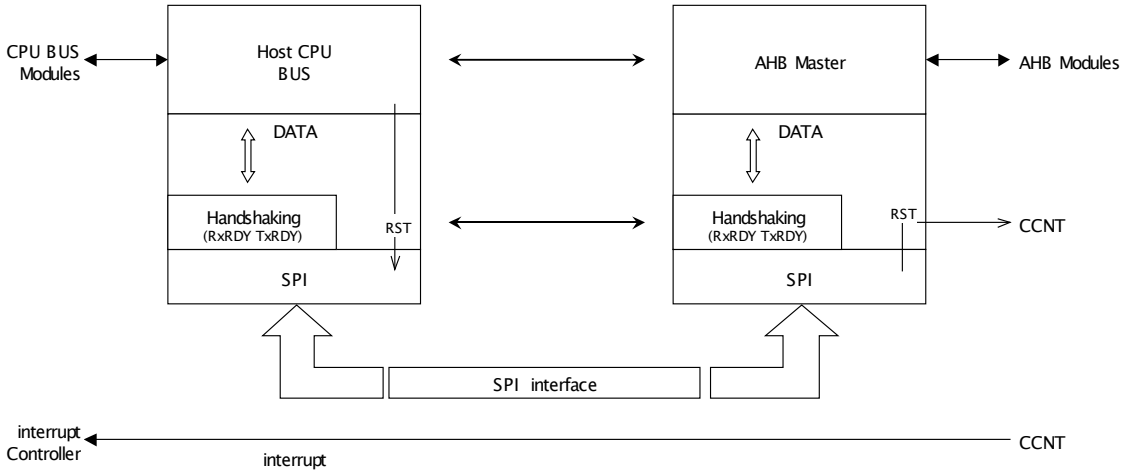


Figure 4-8 SPI communication protocol stack

4.4.2. Data Formats

4.4.2.1. Host Interface (clock timing and phase)

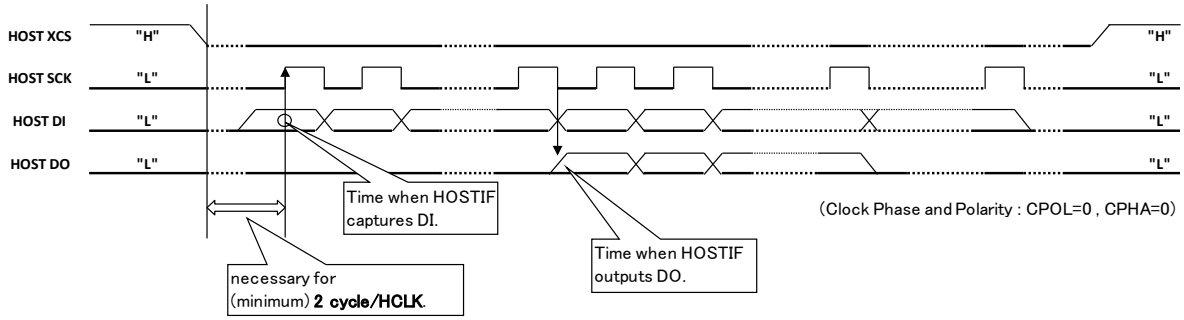


Figure 4-9 Host Interface (clock timing and phase)

4.4.2.2. Reset Frame

The arrangement of the data byte inputs from the host CPU is a specific one. The byte counter of the EXTIF unit will malfunction if the HOSTIF module is initialized while the host CPU is communicating with the HOSTIF module (for example due to an initialization by the MB88F333's watchdog timer (WDT) or by initialization via a RST-CMD). In this case, the arrangement of the data bytes would be mistakenly interpreted. It is therefore necessary to use a reset frame when initializing when the HOSTIF module is communicating.

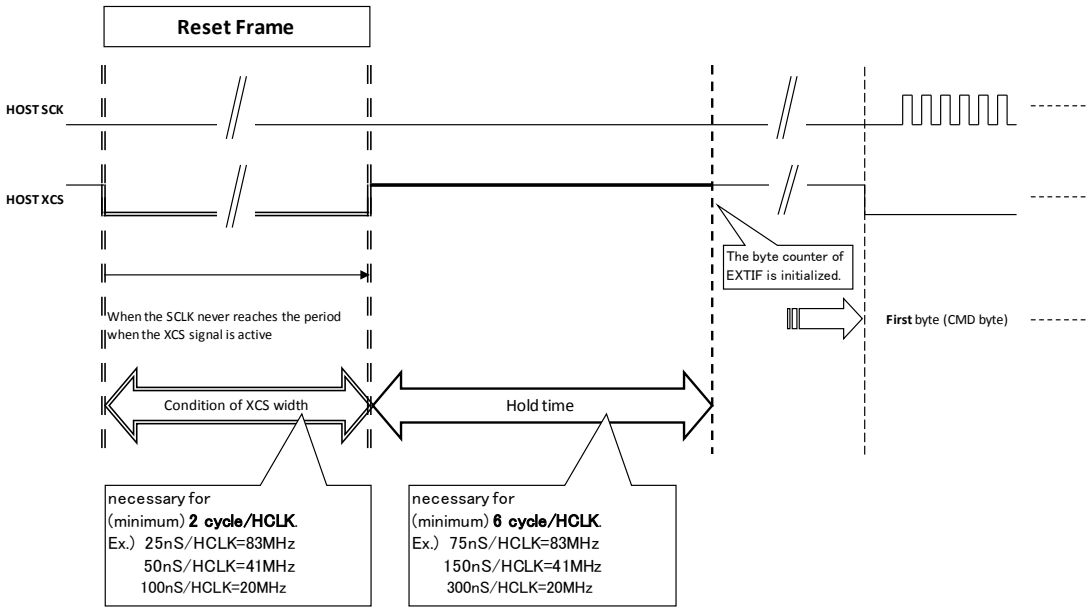


Figure 4-10 Reset Frame

4.4.2.3. Signal input format from the host CPU

The phase relationships of the HOST SCK, HOST XCS, and HOST DI signals is as follows.
The HOSTIF module detects the first '0' of the HOST XCS and stores the data bytes of the specified length. Each byte can be sent using continuous and the non-continuous transmission. The HOSTIF module allows the use of the following three kinds of phase relationships.

4.4.2.3.1. Non-continuous data bytes with non-continuous HOST XCS

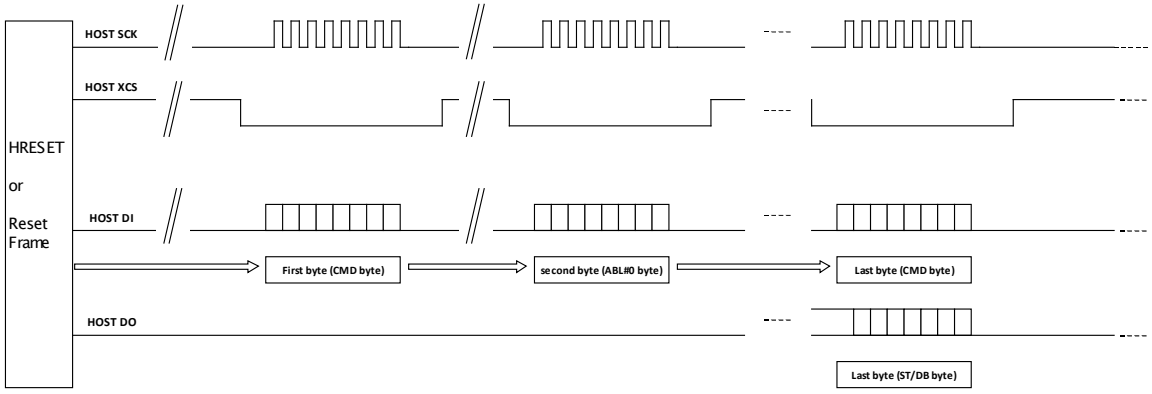


Figure 4-11 Non-continuous data bytes with non-continuous HOST XCS

4.4.2.3.2. Non-continuous data bytes with continuous HOST XCS

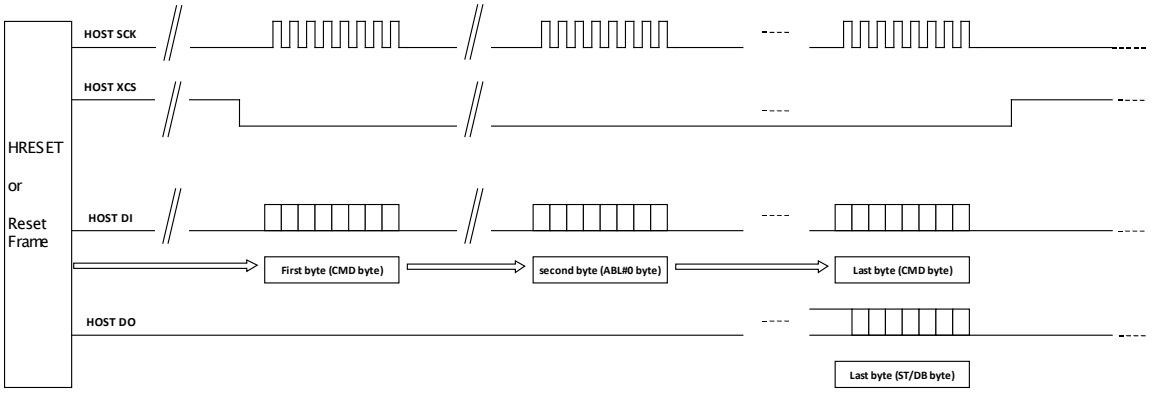


Figure 4-12 Non-continuous data bytes with continuous HOST XCS

4.4.2.3.3. Continuous data bytes with continuous HOST XCS

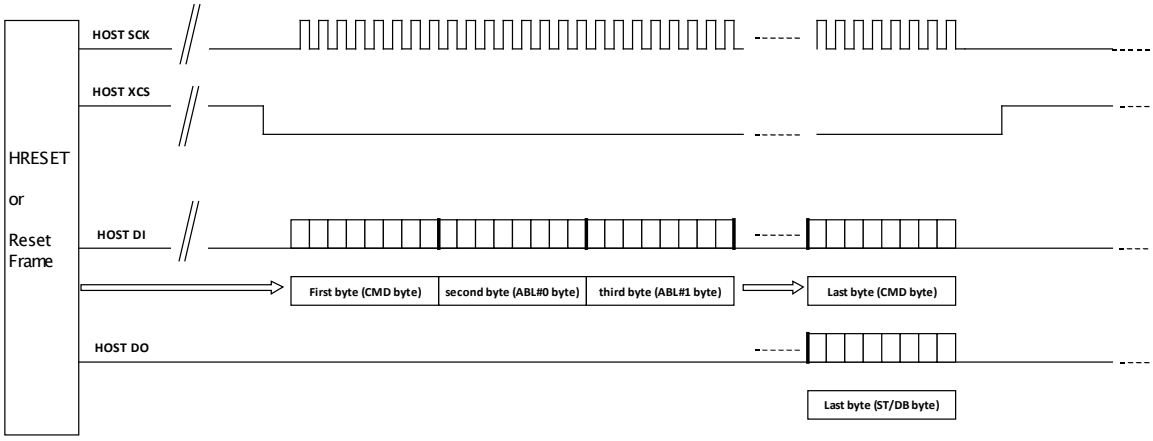


Figure 4-13 Continuous data bytes with continuous HOST XCS

4.5. Application Notes

4.5.1. Processing Flow

4.5.1.1. Begin timing of protocol sequence

The protocol sequence sent to the HOSTIF module must begin with CMDSEQ after booting has completed.

4.5.1.2. Receive operation and the STATUS byte

Normal receive operation is confirmed using five bits in the STATUS byte. If the corresponding five bits are all High, the system is in normal receive operation mode. If, for example, the STATUS byte always contains Low bits or always contains High bits, normal receive operation is not functional.

4.5.1.3. Setting the address

The host CPU can freely select an address byte when accessing the MB88F333. If the address is not set, the previous address is maintained and therefore it is not necessary to repeat the address byte with every access. This implements a very effective forwarding mechanism, an example of which is shown below.

Because address information maintained in the host interface module is initialized when the communication using "Reset Frame" is done, the following forwarding cannot be done.

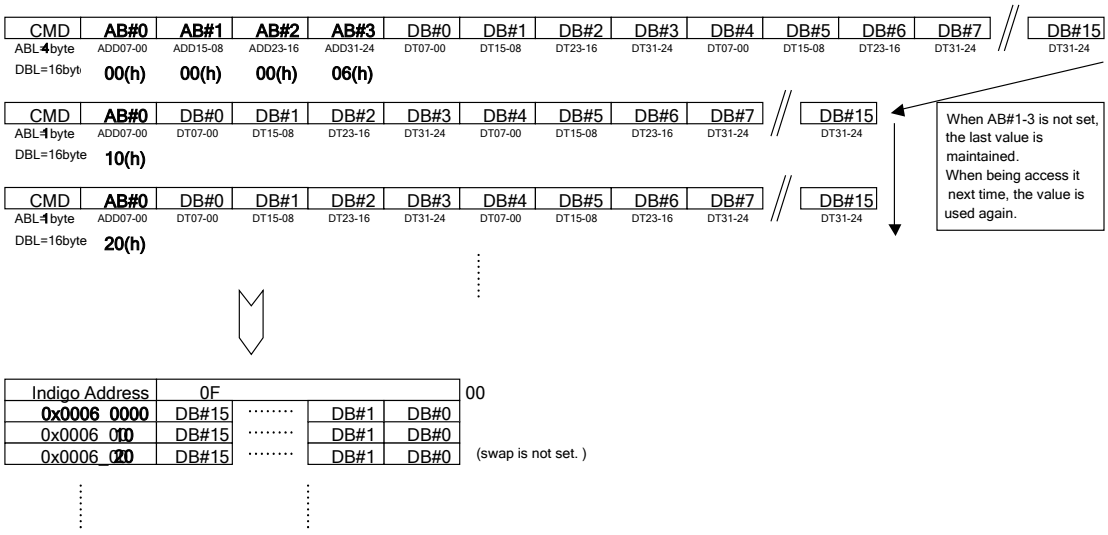


Figure 4-14 Example of setting the address (for write processing)

4.5.1.4. Handling of irregular operating conditions

4.5.1.4.1. XCS abnormality handling

Even if XCS ends prior to the time set in the CMD byte, the MB88F333 will follow the setting of the CMD byte.

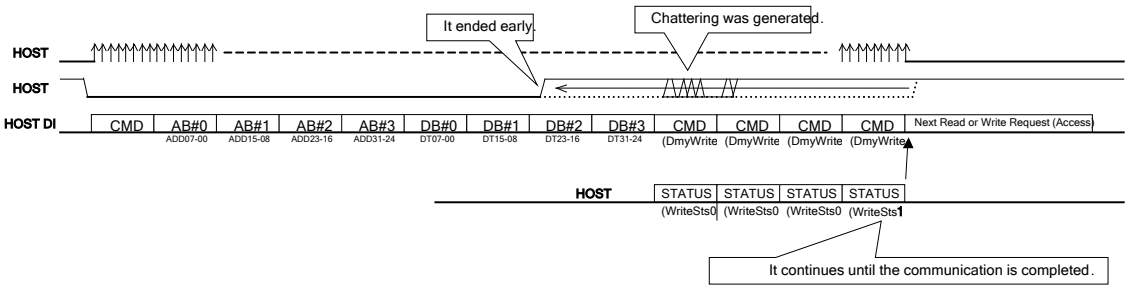


Figure 4-15 XCS abnormality handling

4.5.1.4.2. CLK abnormality handling

The MB88F333 device standard operation uses the internal SPICLK signal. The MB88F333 devices does not have a CRC error correction function. The normal operation of the MB88F333 can not be guaranteed if the CLK signal exceeds specified boundaries.

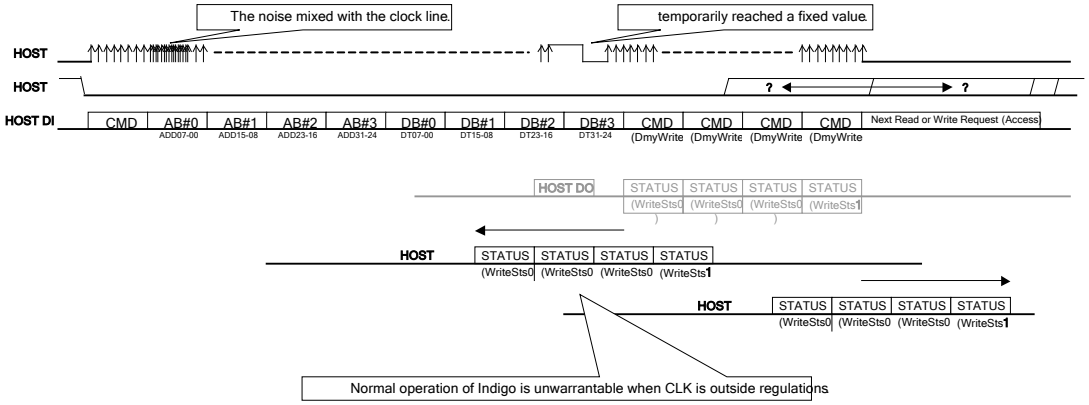


Figure 4-16 CLK abnormality handling

Abnormal CLK signals will lead to a malfunction of the MB88F333 device. Therefore please give careful attention to this issue when designing the clock line.

4.5.1.4.3. The first CMD is a dummy command

When the first CMD is a dummy write, the STATUS byte is immediately returned. The status is different depending on whether a READ or WRITE is returned. At this point in time, the transaction is not issued to the AHB. After the STATUS byte replies, this forwarding transaction is completed.

When the first CMD is a dummy Write (R/W bit=WRITE)

TxRDY is sent back for dummy WRITE CMD. Refer to Figure 0

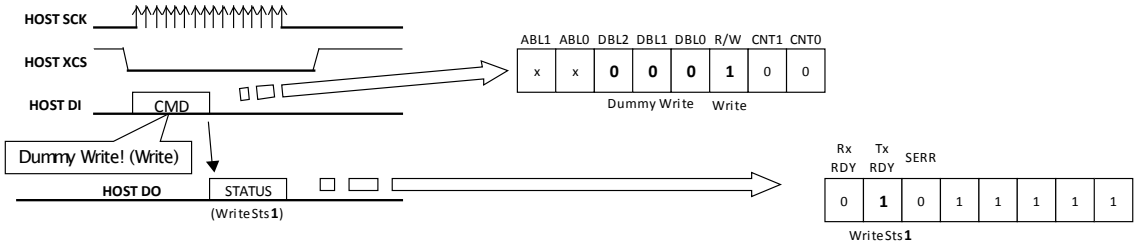


Figure 4-17 Dummy Write (R/W bit=WRITE)

When the first CMD is a dummy Read (R/W bit=Read)

RxRDY is sent back for dummy READ CMD. Refer to Figure 0

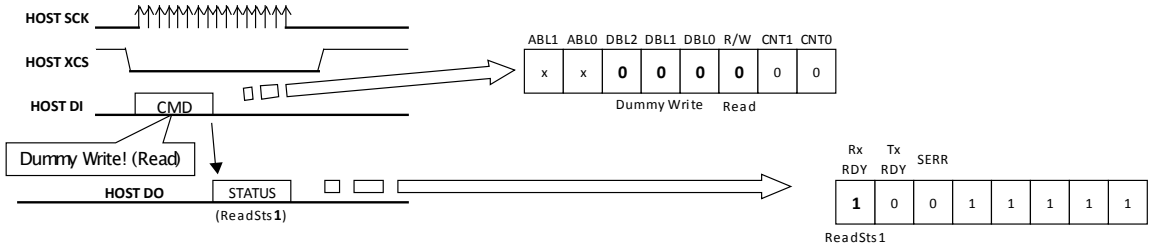


Figure 4-18 Dummy Read (R/W bit= READ)

4.5.1.4.4. The first CMD is a reset request

If the first CMD is a dummy RESET command, the HOSTIF module is reset at once although the response has begun. Please send the reset frame after the reset request.

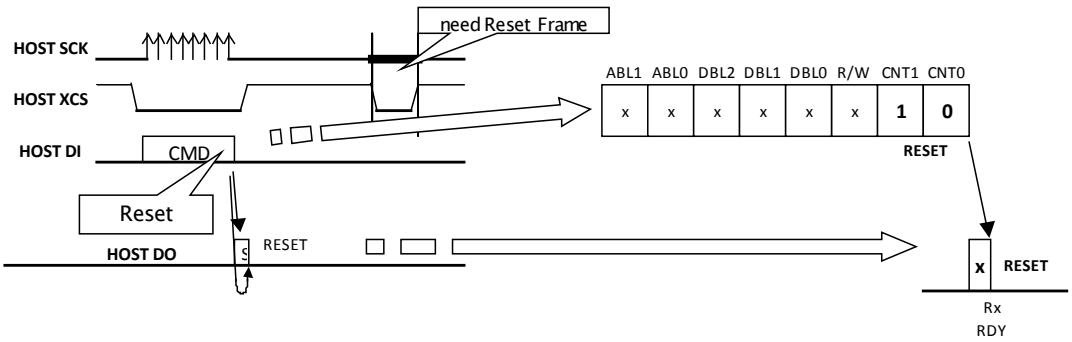


Figure 4-19 Reset Request

4.5.1.4.5. Deadlock Scenario

Situation recognition and recommendations

This section describes a deadlock scenario (at internal points and in the interfaces) of the MB88F333.

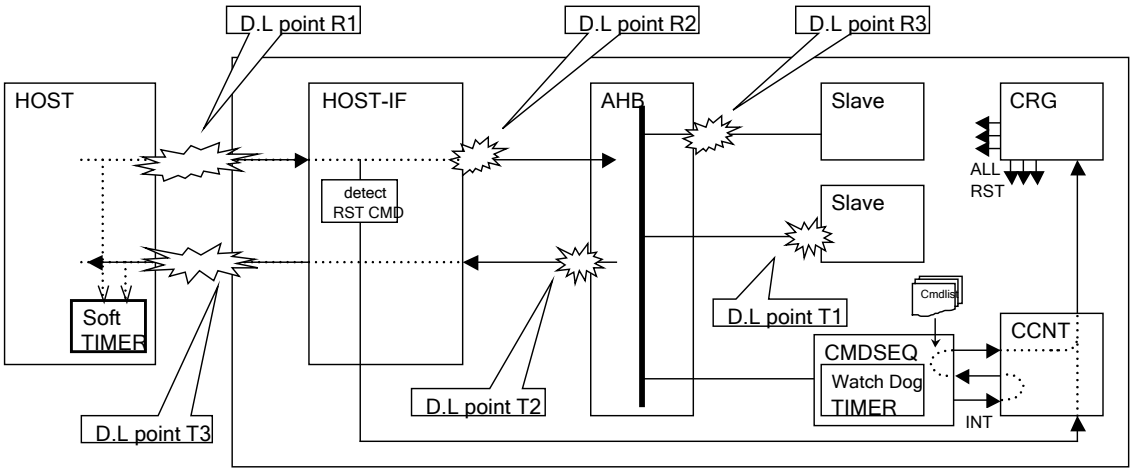


Figure 4-20 Deadlock locations (points)

Deadlock point	Cause of Deadlock	HOST-IF	AHB	Status to HOST	Deadlock detection	
					Soft TIMER	WDT
D.L point R1	in HOST or IF	Idol	Idol	don't send	detect *1	detect
D.L point R2	in HOST-IF	*Deadlock*	Idol or Deadlock	Idol or Deadlock	detect	detect
D.L point R3	in AHB BUS	waiting resp	*Deadlock*	send wait Sts0	detect	detect
D.L point T1	in Slave	waiting resp	waiting resp	send wait Sts0	detect	not detect *2
D.L point T2	in AHB BUS	waiting resp	*Deadlock*	send wait Sts0	detect	not detect *2
D.L point T3	in HOST-IF or IF	*Deadlock*	Idol or Deadlock	*Deadlock*	detect	not detect *2

The MB88F333 device signals the completion of a normal access via the STATUS byte. If a deadlock situation occurs, the device can be automatically recovered using a watchdog (WDT) mechanism. The command list can be used to select whether watchdog reactivation is executed or not.
It is also possible to monitor a MB88F333 deadlock using the timer under the control of the host CPU software. The host CPU can reset the MB88F333 by using a CMD byte. We strongly recommend that you use a host CPU issued reset because the host CPU can not detect a WDT issued reset. The host CPU should therefore detect abnormal operation of the MB88F333 and reactivate it itself.

4.5.1.4.6. Reactivation process

If the status is not returned because the AHB-BUS and the slave device are deadlocked, the host CPU can insert an RST request in the CMD byte during transmission. The figure below shows the deadlock reactivation process.

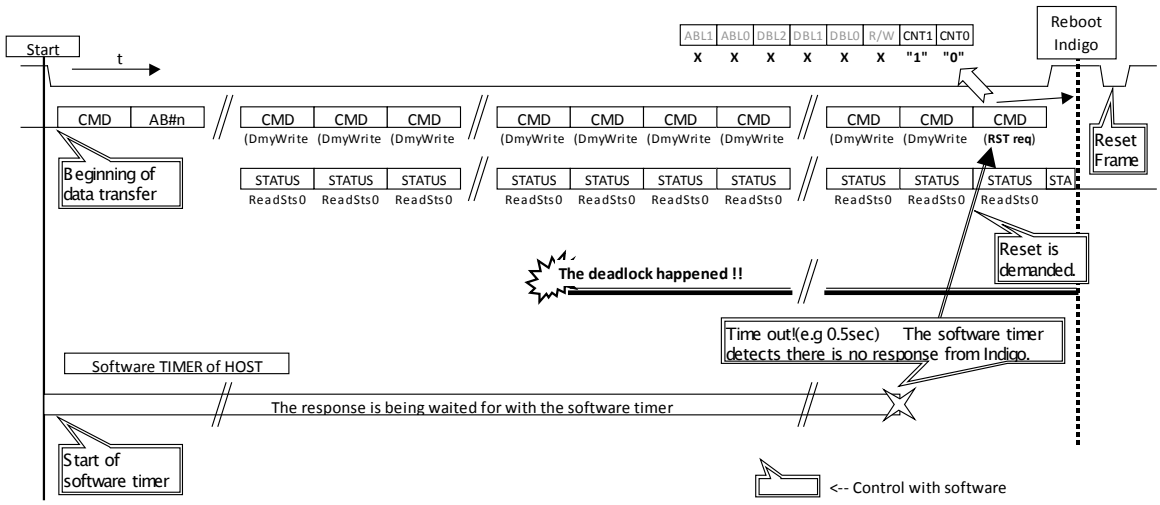


Figure 4-21 Deadlock reactivation process

The response from the HOSTIF module is monitored using the software timer which is managed by the host CPU.
The MB88F333 is reactivated from a deadlock by asserting a RST from the host CPU if there has been no response for a set time (e.g. for 0.5 seconds).

5. APIX Interface

This chapter describes the MB88F333's APIX interface.

5.1. Outline

The APIX interface provides a high speed serial downstream link and a low speed serial upstream link compliant to the Inova APIX Industrial Standard. The downstream link transports video or generic data and sideband data. The upstream link transports sideband data only.

5.2. Features

The APIX interface has following features:

5.2.1. Features List

5.2.1.1. APIX® PHY

The APIX® PHY provides a high speed serial downstream link and a low speed serial upstream link compliant to the Inova APIX® Industrial Standard. The downstream link transports video or generic data and sideband data. The upstream link transports sideband data only.

Details can be found in the document "APIX® Automotive Pixel Link Industrial Standard Rev. 1.0"

The downstream link provides the following bandwidth modes (set via a Command Sequence in Indigo-L):

- Full Bandwidth Mode (1000 Mbit/s)
- Half Bandwidth Mode (500 Mbit/s)
- Low bandwidth Mode 1 (125 Mbit/s)
- Low bandwidth Mode 2 (250 Mbit/s)
- One single video channel

Further features of the APIX® PHY module are

- Establishment and maintenance of serial frame alignment
- Framing/deframing serial frames
- Line coding and DC balancing
- Serialization/deserialization

5.2.1.2. APIX® Ashell

The APIX® Automotive Shell (Ashell®) provides a secured bidirectional communication path for control data. The following key functions are offered.

- Convenient wrapping of APIX® PHY's interface
- Transaction framing and de-framing
- Establishment and maintenance of transaction alignment
- Exchange of transactions utilizing services of APIX® PHY
- Bit error detection
- Bit error management
- Status reporting via remotehandler register interface
- Configuration of APIX® PHY and Ashell® via remotehandler register interface

Details can be found in the document “APIX® Automotive Pixel Link Industrial Standard Rev. 1.0”

5.2.1.3. Indigo-L Restrictions

Indigo-L’s initial APIX configuration is hardwired and corresponds to the Indigo’s default mode (Mode 0, 125 Mbit). The default mode and other APIX modes (Note: these are actually configurations for specific use cases, described later in this chapter) can be set using a command sequence (see also ‘Configuration Interface’ section).
Only a single channel video interface is supported.

5.3. FUNCTION

5.3.1. Block diagram

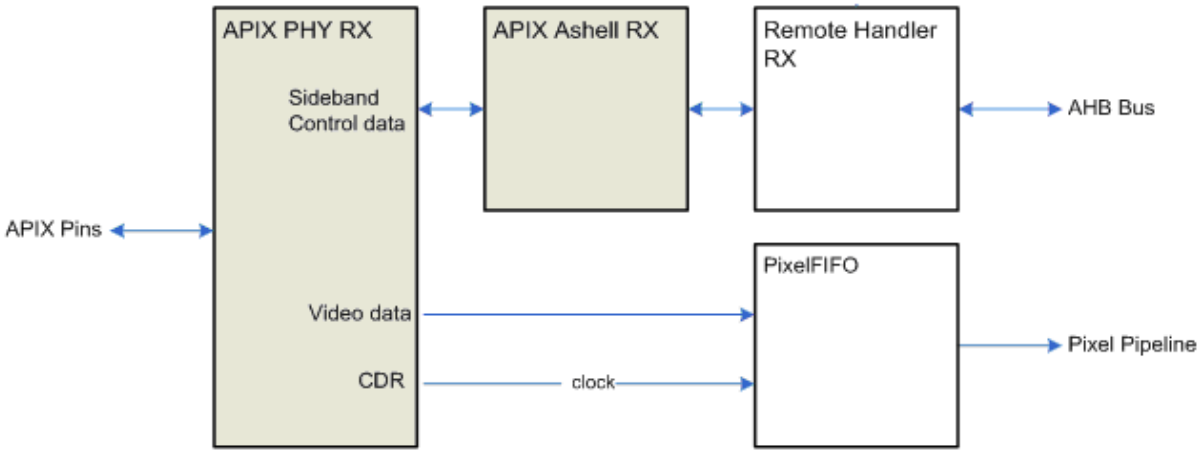


Figure 5-1 Block diagram of APIX® PHY RX and APIX Ashell® RX in the System

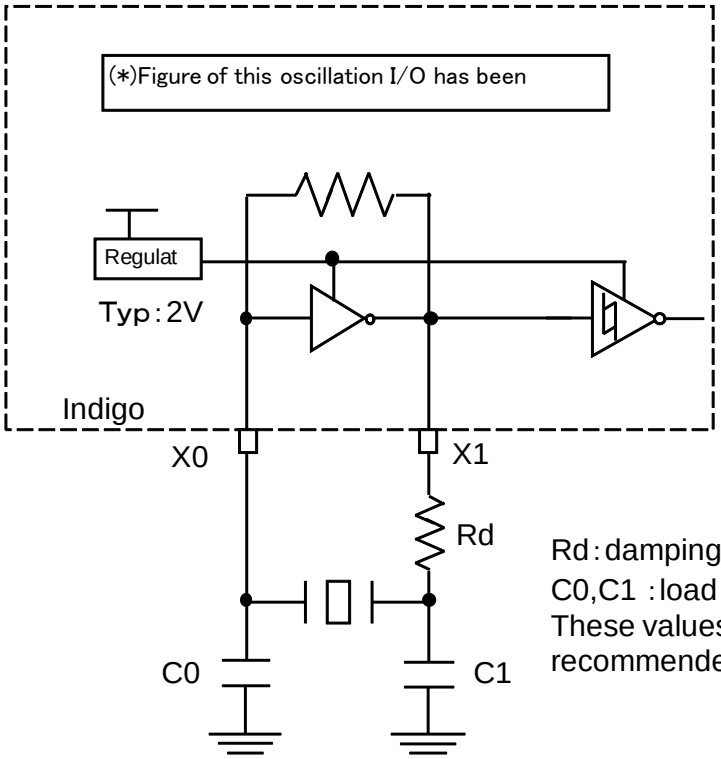
5.4. External Interface

For the external Interface please see chapter “1.7.2.2. APIX I/F related pins”

5.4.1. Xtal Interface

An IO buffer for the oscillator is integrated in Indigo-L.
Please connect the oscillator using XTAL1 and XTAL0.
The IO buffer, incorporates feedback resistance.

A connection example is shown below.



Rd : damping resistor
C0,C1 : load capacitance
These values must use the recommended value of the oscillator

5.5. Configuration Interface

The configuration software interface for APIX® PHY and APIX® Ashell® is available via the PHYConfig[1:0] and ShellConfig[0] registers of the remote handler configuration interface.

These registers can be programmed after chip reset using the Indigo-L Command Sequencer bootup sequence.

Note Reprogramming of the configuration registers during use of the APIX hardware is not possible! The APIX® PHY and APIX® Ashell® units must be set into reset state for reprogramming.

If the modification of analog APIX® parameters is necessary, then this must be taken into consideration when writing a Command Sequencer sequence (recommended: in the boot-up sequence).

5.5.1. APIX® Modes (Use Case Configurations)

In previous documentation releases for MB88F332 ‘Indigo’, specific APIX pins (APIX_cfg) were used to configure the hardware for specific ‘APIX Modes’, which in fact were specific use case configurations, which are described in the following sections (using the terminology ‘modes’). As the APIX_cfg pins have been removed in MB88F333 ‘Indigo-L’, the hardware is hard-wired to (default) mode 0 and all the other modes can only be configured via a command sequence after reset.
The following tables describes the most typical use cases (i.e. excludes e.g. ‘Low bandwidth mode 2 = 250 Mbit/s).

	APIX Mode 0 Low-speed APIX Mode using Indigo Sprite Engine (Hard-wired default)	APIX Mode 1 High-Quality Dual Display Operation	APIX Mode 2 APIX Ashell SW Emulation
APIX PHY:	Low bandwidth (125 Mbit/s)	Half/ or Full bandwidth * (500Mbit/s or 1000Mbit/s)	Half or Full bandwidth * (500Mbit/s or 1000Mbit/s)
Data Link:	Enabled	Enabled (1 bit synchronous sideband link)	Enabled (1 bit synchronous sideband link)
Video Link:	Disabled	Enabled	Enabled
Configuration of RemoteHandler RX registers:	Not required (default)	write APConfig.rx_config = ‘1’ write PHYConfig0 write PHYConfig1 write ShellConfig0 write APConfig.rx_config = ‘0’	write APConfig.rx_config = ‘1’ write PHYConfig0 write PHYConfig1 write ShellConfig0 write APConfig.rx_config = ‘0’

* Mode pin PLL_MODE is used to select between half bandwidth and full bandwidths:
PLL_MODE=0: 500MHz down link rate
PLL_MODE=1: 1000MHz down link rate

5.6. Application Notes

Please use the following link to go to the inova Semiconductors website for application notes that handle the use of the APIX interface:
http://www.inova-semiconductors.de/en/design_application_notes.html

5.6.1. Low-speed APIX Mode using Indigo-L Sprite Engine

This is referred to as ‘Mode 0’. This example uses Fujitsu Semiconductor's MB91F467S MCU, which connects directly to the MB88F333 'Indigo-L' device.

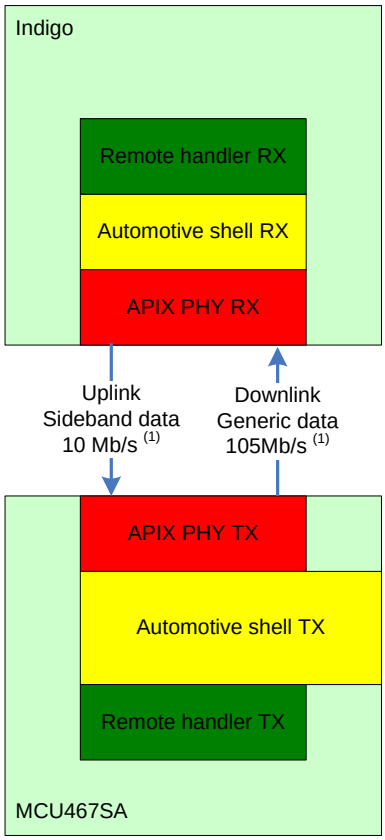


Figure 5-2, APIX® low bandwidth application, (1) values given, are link rate values

This application is mainly characterized, that no video link is provided by the APIX® IPs. The full remaining bandwidth of the link is available for control and status data communication between Indigo-L and an APIX® TX device. This operation mode provides the highest downlink bandwidth for sprite pattern data download and control of real time critical peripherals like e.g. stepper motor drivers or ADCs. The uplink is available like within all other applications for read access to Indigo-L status registers and messaging of Indigo-L interrupt events to a Remote Handler TX device. A sample application is shown with Fujitsu’s MB91F467S MCU. Both, the generic data downlink and the sideband data uplink are operated in synchronous mode.

5.6.2. High-Quality Dual Display Operation

This is referred to as ‘Mode 1’. This example uses Fujitsu Semiconductor's MB91467SA MCU, which connects directly to the MB86R02 'Jade-D' GDC, which in turn controls two MB88F333 'Indigo-L' devices.

Apix RX main characteristics are

- Video Link in full or half bandwidth mode
- 1 bit Sideband up- and downlink in synchronous mode

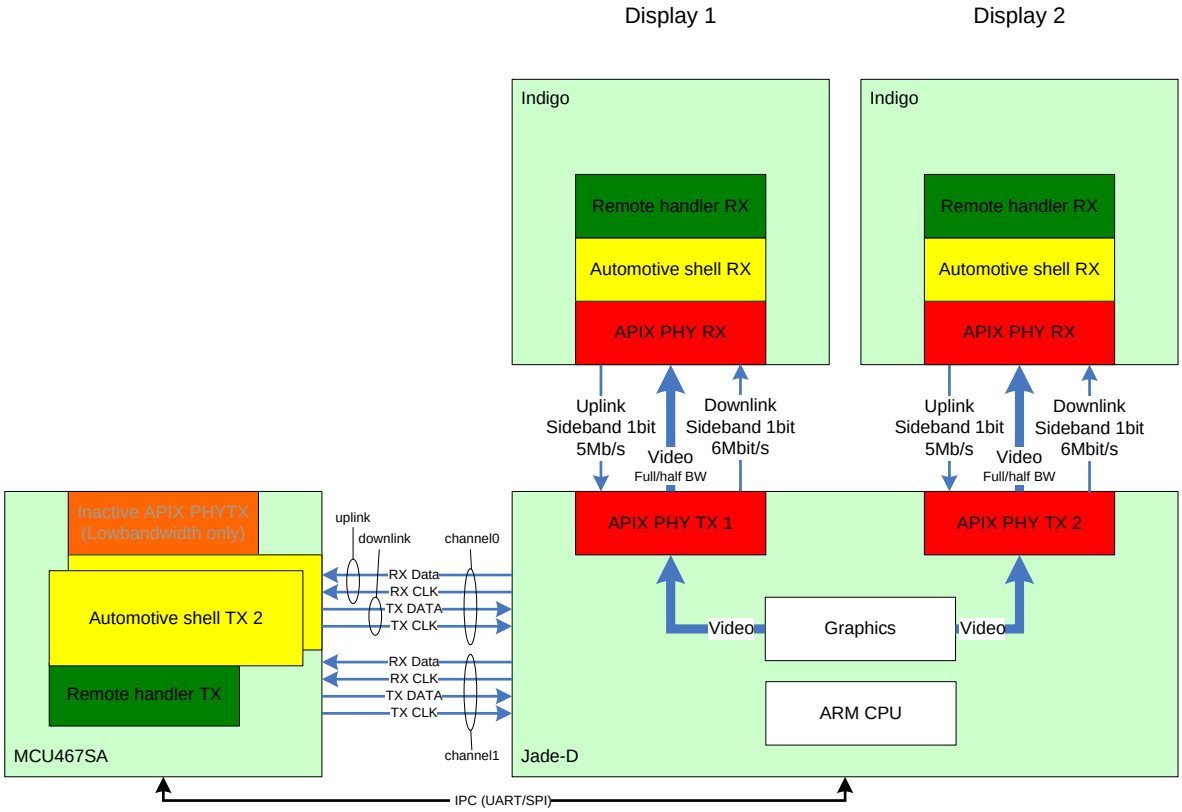


Figure 5-3, High-Quality Dual Display application (variant 1)[1]

[1] All bandwidth numbers given are link rate numbers

5.6.3. APIX® Ashell SW Emulation Use Case

This is referred to as ‘Mode 2’. This example uses a standard MCU, which uses a discrete APIX connection, which connects directly to the MB88F333 'Indigo-L' device.

This application is described in Inova application note document “APIX® SW Emulation Use case, version 0.20”
Main characteristics of such a use case are:

- Usage of existing discrete APIX® TX device
- Usage of two SPI interfaces available at common MCU devices, SPI master for downlink, SPI-slave for uplink
- Reduced Ashell® functionality implemented in SW

APIX® RX main characteristics are:

- Video Link in full or half bandwidth mode
- 2 bit Sideband up- and downlink in asynchronous mode, SPI operation
- Ashell® reduced feature set (error detection ON, error management OFF, SPI protocol via sideband link ON, payload only mode)

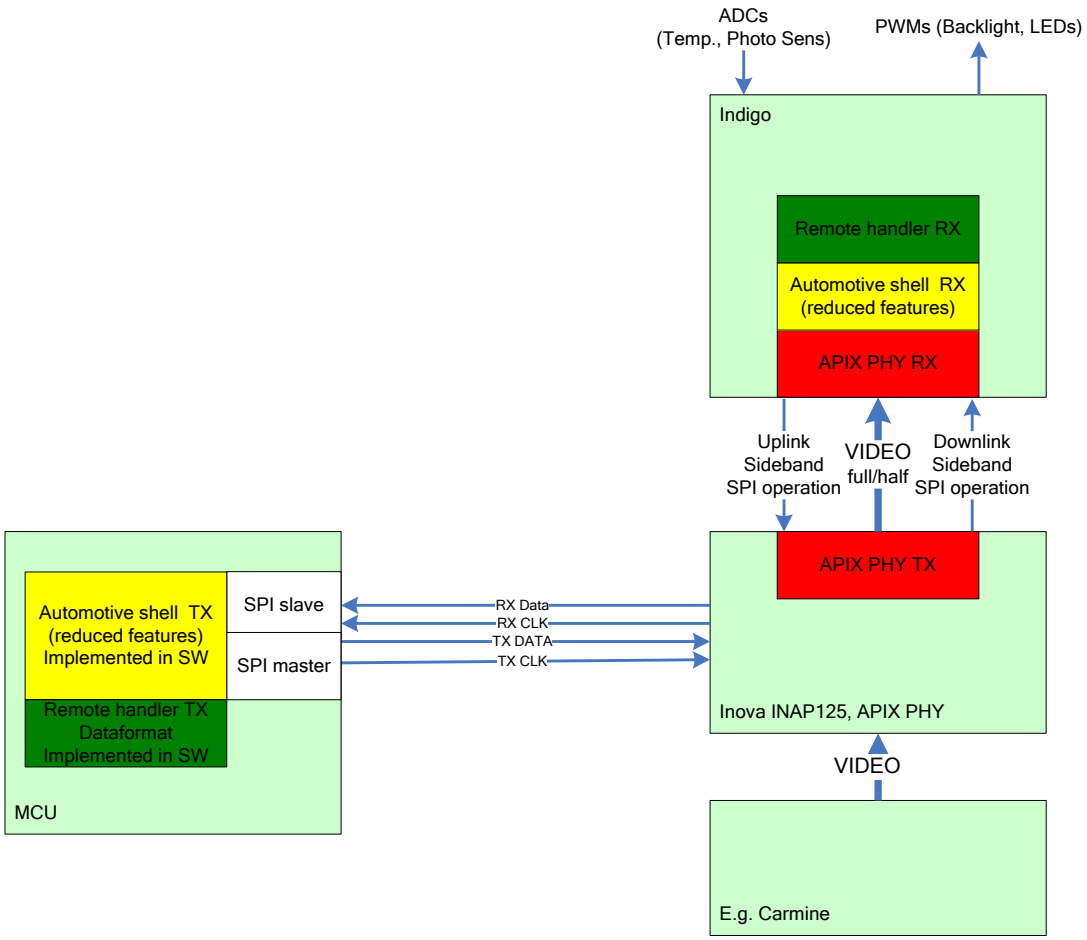


Figure 5-4, APIX compatibility mode for a SW Emulation Use case example

5.7. Control Flow

The general control flow for configuring the APIX PHY and A-Shell units is done via the APIX registers as follows: (this is the recommended order!)

```
write APConfig.rx_config = ‘1’
write PHYConfig0
write PHYConfig1
write ShellConfig0
write APConfig.rx_config = ‘0’
```

Such a configuration sequence can be executed as a command sequence after reset.

6. Remote Handler

This chapter describes the Remote Handler (Rx) of the MB88F333 device.

6.1. Outline

The remote handler unwraps transactions received from external host CPU via the APIX automotive shell to the hardware protocol of the AHB system bus and vice versa. Additionally this module has the task of an intelligent interrupt controller. It builds interrupt messages and sends them as event messages to the TX remote handler. If an application requires to read some addresses from the indigo register space every time a specific indigo interrupt is received the remote handler autonomously collects the data to be read (later) and sends this interrupt message right along with the data. In this way the required communication overhead is reduced.

6.2. Features

The Remote Handler has following features.

6.2.1. Features

- Interface to the automotive shell
- Unwrapping of messages received by automotive shell and translating them to AHB-master transactions and vice versa
- Decoding of command-type field, address translation (address offset)
- Evaluation of automotive shell status bits to detect APIX link status
- Provide automotive shell configuration and status bits to system bus via AHB-slave interface
- Provide APIX PHY configuration and status bits to system bus via AHB-slave interface
- Interfacing to AHB-system Bus
- AHB-write master with FIFO (depth: 16 messages, necessary if bus congestion by accessed slaves is possible)
- AHB-read master with FIFO (depth: 16 messages)
- Arbitration between interrupt read and normal read transactions on AHB-master
- Read sequencer for 128 interrupt sources
- Building of interrupt messages
- Automatic clear of interrupt request flag (programmable)
- Locking and unlocking of AHB-write master

6.2.2. Limitations

None

6.3. Function

6.3.1. Block diagram

Figure 6-1 shows the block diagram of the Remote handler.

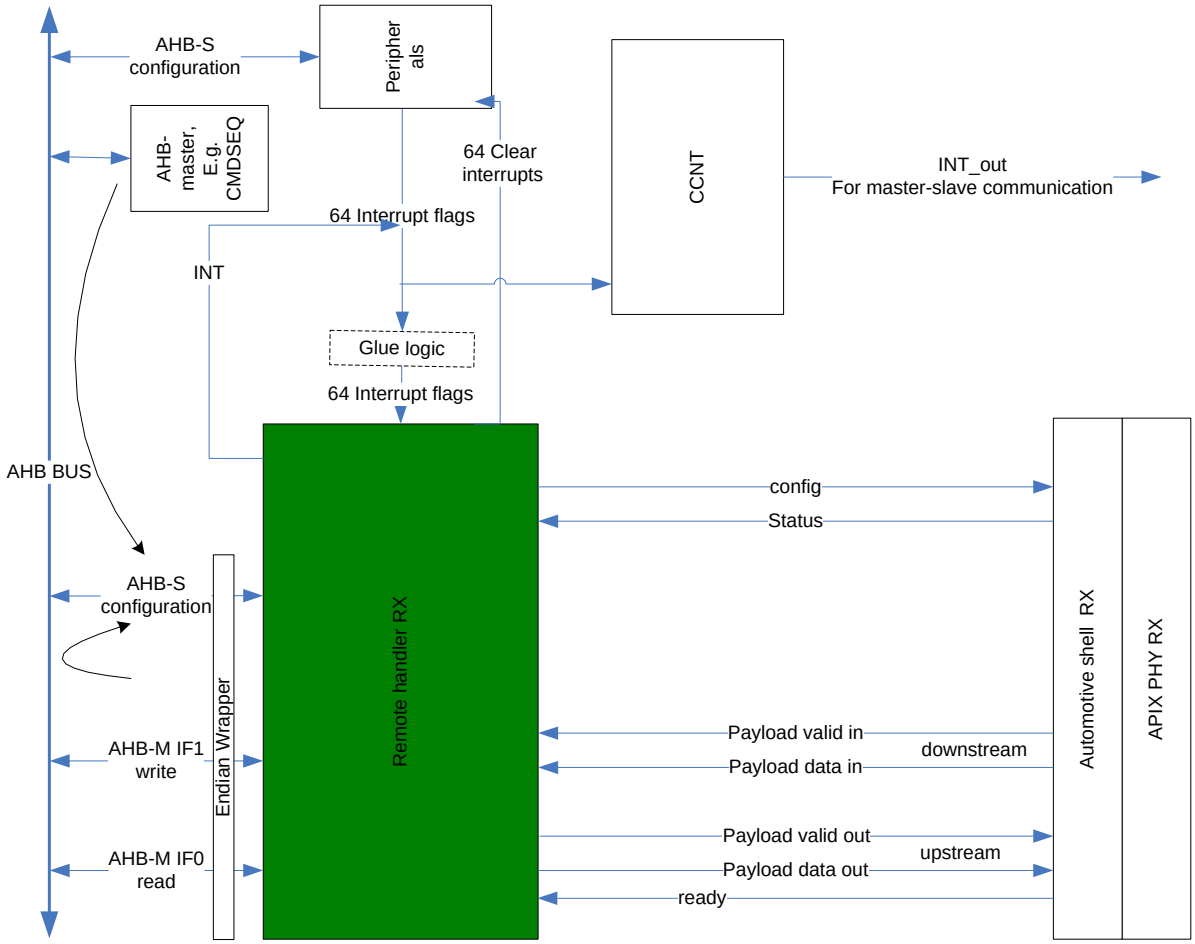


Figure 6-1 Remote handler in the LSI

6.3.2. Interfacing to Automotive shell

One message consists of 56 bits for both directions, upstream and downstream.
The maximum supported downstream message rate is 800K Messages/s.
Automotive shell upstream part supports a minimum of 70K Messages/s for error-free communication.

6.3.3. Interfacing to AHB bus

Two AHB masters are provided, one for read access and one for write access. Inbound data from the host CPU has a big endian byte order. Endianness conversion is not executed within the module. If necessary for system integration, please add the necessary endianess conversion units external to the AHB interfaces.

6.3.4. Message based Interrupts

An event message is sent from remote handler RX to the remote Handler TX event buffer for each interrupt request flag. Interrupt priorisation is handled by an interrupt number for 128 interrupt request flags. The lowest number has the highest priority.

One or zero (NOP) AHB read transactions are executed for each interrupt. The read addresses of these read transactions are defined in the MB88F333 interrupt table (See chapter 6.9). The result is then delivered in the data field of the corresponding event message. If the executed AHB transaction can not be successfully executed, a dedicated message bit indicates the AHB error (for the bit number, please see the frame format definition table). An automatic clear of the interrupt flag can be executed.

For the list of event numbers and corresponding interrupt sources please refer to the MB88F333 interrupt table (See chapter 6.11. Event ID table).

6.3.5. Locked AHB write master

The AHB write master of the module is at reset state in a locked state. Before any write access to the AHB bus are possible, an “unlock pattern” (see chapter 6.4) must be received. During operation it is possible that the AHB write master is locked via a write of an arbitrary AHB master to register AHBMlock of the Remote Handler AHB slave configuration interface. This makes it possible to avoid competing access to resources (for example Sprite Engine). Read access is possible at any time, even if the write master is locked.

6.3.6. Limitations

The application must avoid bus congestion on the RX system bus. Otherwise the transactions FIFOs of the remote handler may overflow. The incoming transactions are then lost or delayed. Control of critical peripherals such as stepper motor controllers is then not possible.
The FIFO threshold interrupt may be used to inform the sending host CPU to stop sending until the bus congestion has disappeared, but since the transmission quality of the apix link communication is variable, the time for sending this interrupt message is unpredictable. Therefore this communication method is not recommended. Possible candidates for causing bus congestion are the run length decompression module or the embedded flash module.

6.4. Data Formats

Downstream Write Transaction
Remote Handler TX (host CPU) → Remote Handler RX (Indigo)

bit	byte	Field	Bits		Value	Description
55			Tsize	1		transaction size: 00b=byte, 01b=halfword(2byte), 10b=word(4byte), 11b=Reserved
54			Tsize	0		
53			OAen			Offset Adress Enable
52		control[3:0]	WRn		1	Write transaction
51			Addr	19		
50			Addr	18		
49			Addr	17		
48	0		Addr	16		
47			Addr	15		
46			Addr	14		
45			Addr	13		
44			Addr	12		
43			Addr	11		
42			Addr	10		
41			Addr	9		
40	1		Addr	8		
39			Addr	7		
38			Addr	6		
37			Addr	5		
36			Addr	4		
35			Addr	3		
34			Addr	2		
33			Addr	1		
32	2	Addr[19:0]	Addr	0		Byte Address
31			Data	31		
30			Data	30		
29			Data	29		
28			Data	28		
27			Data	27		
26			Data	26		
25			Data	25		
24	3		Data	24		
23			Data	23		
22			Data	22		
21			Data	21		
20			Data	20		
19			Data	19		
18			Data	18		
17			Data	17		
16	4		Data	16		

15			Data	15		Data,
						IF Tsize=b00: then
						Data[31:24] is used;
						IF Tsize=b01: then
						Data[31:16] is used;
						IF Tsize=b10: then
						Data[31:0] is used;

14			Data	14		
13			Data	13		
12			Data	12		
11	5	data[31:0]	Data	11		

10			Data	10		
9			Data	9		
8			Data	8		
7			Data	7		
6			Data	6		
5			Data	5		
4			Data	4		
3			Data	3		
2			Data	2		
1			Data	1		
0	6		Data	0		

Downstream Read Transaction
Remote Handler TX (host CPU) → Remote Handler RX (Indigo)

bit	byte	Field	Bits		Value	Description
55			Tsize	1		transaction size: 00b=byte, 01b=halfword(2byte), 10b=word(4byte), 11b=Reserved
54			Tsize	0		
53			OAen			Offset Adress Enable
52		control[3:0]	WRn		0	Read Request transaction
51			Addr	19		
50			Addr	18		
49			Addr	17		
48	0		Addr	16		
47			Addr	15		
46			Addr	14		
45			Addr	13		
44			Addr	12		
43			Addr	11		
42			Addr	10		
41			Addr	9		
40	1		Addr	8		
39			Addr	7		
38			Addr	6		
37			Addr	5		
36			Addr	4		
35			Addr	3		
34			Addr	2		
33			Addr	1		
32	2	Addr[19:0]	Addr	0		Byte Address
31			idx	7		
30			idx	6		
29			idx	5		
28			idx	4		
27			idx	3		
26			idx	2		
25			idx	1		
24	3	idx[7:0]	idx	0		read Index
23	4	Reserved	-			

22			-			
21			-			
20			-			
19			-			
18			-			
17			-			
16			-			
15			-			
14			-			
13			-			
12			-			
11			-			
10			-			
9			-			
8	5		-			
7			-			
6			-			
5			-			
4			-			
3			-			
2			-			
1			-			
0	6		-			unused

Upstream Read Result
Remote Handler RX (Indigo) → Remote Handler TX (host CPU)

bit	byte	Field	Bits		Value	Description
55			Tsize	1		transaction size: 00b=byte, 01b=halfword(2byte), 10b=word(4byte), 11b=Reserved
54			Tsize	0		
53			Oaen		-	Offset Adress Enable
52		control[3:0]	WRn		0	Read Result transaction
51			Err			Chip internal bus Error
50			-			
49			-			
48	0	Reserved	-			Reserved
47			idx	7		
46			idx	6		
45			idx	5		
44			idx	4		
43			idx	3		
42			idx	2		
41			idx	1		
40	1	idx[7:0]	idx	0		Read Index
39			-			
38			-			
37			-			
36			-			
35	2	Reserved	-			Reserved

34			-			
33			-			
32			-			
31			Data	31		
30			Data	30		
29			Data	29		
28			Data	28		
27			Data	27		
26			Data	26		
25			Data	25		
24	3		Data	24		
23			Data	23		
22			Data	22		
21			Data	21		
20			Data	20		
19			Data	19		
18			Data	18		
17			Data	17		
16	4		Data	16		
15			Data	15		
14			Data	14		
13			Data	13		
12			Data	12		
11			Data	11		
10			Data	10		
9			Data	9		
8	5		Data	8		
7			Data	7		
6			Data	6		
5			Data	5		

4			Data	4		Read result data,
						IF Tsize=b00: then
						Data[31:24] is used;
						IF Tsize=b01: then
						Data[31:16] is used;
						IF Tsize=b10: then
						Data[31:0] is used;

3			Data	3		
2			Data	2		
1			Data	1		
0	6	data[31:0]	Data	0		

Upstream Event Message
Remote Handler RX (Indigo) → Remote Handler TX (host CPU)

bit	byte	Field	Bits		Value	Description
55			Tsize	1		
54			Tsize	0	**11**	11b= Event Type
53			Oaen		**-**	Offset Adress Enable
52		control[3:0]	WRn		**-**	Read Request transaction
51			Err			chip internal bus Error
50			-			
49			-			
48	0	Reserved	-			
47			idx	7		
46	1	idx[7:0]	idx	6		Event Index

45			idx	5		
44			idx	4		
43			idx	3		
42			idx	2		
41			idx	1		
40			idx	0		
39			-			
38			-			
37			-			
36			-			
35			-			
34			-			
33			-			
32	2	Reserved	-			Reserved
31			Data	31		
30			Data	30		
29			Data	29		
28			Data	28		
27			Data	27		
26			Data	26		
25			Data	25		
24	3		Data	24		
23			Data	23		
22			Data	22		
21			Data	21		
20			Data	20		
19			Data	19		
18			Data	18		
17			Data	17		
16	4		Data	16		
15			Data	15		
14			Data	14		
13			Data	13		
12			Data	12		
11			Data	11		
10			Data	10		
9			Data	9		
8	5		Data	8		
7			Data	7		Read result data, Data size is defined in interrupt ROM table, IF size=b00(byte): then Data[31:24] is used; IF size=b01(halfword): then Data[31:16] is used; IF size=b10(word): then Data[31:0] is used;
6			Data	6		
5			Data	5		
4			Data	4		
3			Data	3		
2			Data	2		
1			Data	1		
0	6	data[31:0]	Data	0		

Downstream Unlock Transaction
Remote Handler TX (host CPU) → Remote Handler RX (Indigo)

bit	byte	Field	Bits		Value	Description
55			Tsize	1		transaction size: 00b=byte, 01b=halfword(2byte), 10b=word(4byte), 11b=Reserved
54			Tsize	0	**10**	
53			OAen		**0**	Offset Adress Enable
52		control[3:0]	WRn		**1**	Write transaction
51			Addr	19		
50			Addr	18		
49			Addr	17		
48	0		Addr	16		
47			Addr	15		
46			Addr	14		
45			Addr	13		
44			Addr	12		
43			Addr	11		
42			Addr	10		
41			Addr	9		
40	1		Addr	8		
39			Addr	7		
38			Addr	6		
37			Addr	5		
36			Addr	4		
35			Addr	3		
34			Addr	2		
33			Addr	1		
32	2	Addr[19:0]	Addr	0	**11004h**	Byte Address
31			Data	31		
30			Data	30		
29			Data	29		
28			Data	28		
27			Data	27		
26			Data	26		
25			Data	25		
24	3		Data	24		
23			Data	23		
22			Data	22		
21			Data	21		
20			Data	20		
19			Data	19		
18			Data	18		
17			Data	17		
16	4		Data	16		
15			Data	15		
14			Data	14		
13			Data	13		
12			Data	12		
11			Data	11		
10			Data	10		
9			Data	9		
8	5	data[31:0]	Data	8	**7353_CAFEh**	Key value

7			Data	7		
6			Data	6		
5			Data	5		
4			Data	4		
3			Data	3		
2			Data	2		
1			Data	1		
0	6		Data	0		

6.5. Registers

6.5.1. Format of Register Descriptions

The register descriptions in the following sections use the format shown below to describe each bit field of a register.

Register address	Offset																														
Bit number	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
Field name																																
R/W																																
Reset value																																

Meaning of items and sign

Register Address

Register address shows the address (Offset address) of the register.

Bit number

Bit number shows bit position of the register.

Field Name

Field name shows bit name of the register.

R/W

R/W shows the read/write attribute of each bit field:

- R: Read Only
- W: Write Only
- W1C: Write a value of “1” clears the register

Reset value

Reset value indicates the value of each bit field immediately after reset.

- 0: Initial value is "0".
- 1: Initial value is "1".
- X: Undefined.

Unused register fields are marked with a solid grey background.

Bit vectors are unsigned integers, if nothing else specified.

6.5.2. Global Address

For the module base address please refer to the chapter 3 Memory map.

6.5.3. Register summary

Address	Register Name	Description
Base address + 0_H	CommonControl	common device control register
Base address + 4_H	AHBMlock	AHB master lock register
Base address + 8_H	AddressOffset	Address offset (base address)
Base address + C_H	FIFOthresh	threshold value of input message FIFOs
Base address + 10_H	IntClrCfg0	configuration for interrupt 31 downto 0 clearing
Base address + 14_H	IntClrCfg1	configuration for interrupt 63 downto 32 clearing
Base address + 18_H	IntClrCfg2	configuration for interrupt 95 downto 64 clearing
Base address + $1C_H$	IntClrCfg3	configuration for interrupt 127 downto 96 clearing
Base address + 20_H	Mailbox	send a message to the host
Base address + 24_H	IEN	interrupt enable for respective interrupt; see register ISTS for description of interrupt behavior
Base address + 28_H	ISTS	Interrupt status flags, a '1' signifies that the corresponding interrupt condition occurred (even if interrupt is disabled), write '1' clears the flag,
Base address + $2C_H$	AHBWriteErrorAddress	current AHB write address; this value is always updated when IStsWError is not asserted (for debug purpose only)
Base address + 30_H	AHBReadErrorAddress	current AHB read address; this value is always updated when IStsRError is not asserted (for debug purpose only)
Base address + 34_H	StatusFIFOEmpty	Status 'empty' of read and write transaction FIFOs
Base address + $3C_H$	APConfig	Configuration register for A-shell and APIX PHY
Base address + 40_H	PhyConfig0	APIX PHY configuration bytes
Base address + 44_H	PhyConfig1	APIX PHY configuration bytes
Base address + 48_H	ShellConfig0	A-shell configuration bytes
Base address + $4C_H$	ASStatus	A-shell Status register
Base address + 50_H	ASCountRxCRC	number of detected CRC errors in downstream
Base address + 54_H	ASCountSyncLoss	number of detected synchronisation losses of downstream serial channel
Base address + 58_H	ASCountPLLBad	number of detected PLL sync losses in downsteam

6.5.4. Register description

6.5.4.1. CommonControl

Register address	BaseAddress + 0_H																											
Bit number	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
Field name																															MSIenable	FlushFIFO
R/W																															RW	W
Reset value																															1_H	0_H

common device control register	
Bit 1	MSIenable Enable message signalled interrupts (MSI) in Remote Handler: 0b=disable, 1b=enable
Bit 0	FlushFIFO Reset pointers in FIFOs: 0b=no change, 1b=Reset pointers

6.5.4.2. AHBMlock

Register address	BaseAddress + 4_H																															
Bit number	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
Field name																																lock
R/W																																RW
Reset value																																1_H

AHB master lock register	
Bit 0	lock write: 0b=Unlock the AHB write master, 1b=Lock the AHB write master (current AHB transaction is finished, then the write FIFO is held in reset). Note that the external Remote Handler TX can unlock the AHB write master by sending the key pattern (see table 'Downstream Unlock Transaction') read: 0b=AHB write master is unlocked, 1b= AHB write master is locked (not active) Note: AHB read is always possible also during locked AHB write master.

6.5.4.3. Address Offset

Register address	BaseAddress + 8_H																															
Bit number	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
Field name					AddressOffsetRead																AddressOffsetWrite											
R/W					RW																RW											
Reset value					0_H																0_H											

Address offset (base address)	
Bit 27 - 16	AddressOffsetRead Address Offset, MSBs (upper 12 bit) of 32bit AHB read Address; this offset is active when address offset in input message is enabled (OAen=1)
Bit 11 - 0	AddressOffsetWrite Address Offset, MSBs (upper 12 bit) of 32bit AHB write Address; this offset is active when address offset in input message is enabled (OAen=1)

6.5.4.4. FIFOthresh

Register address	BaseAddress + C_H																													
Bit number	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
Field name																					Rthresh								Wthresh			
R/W																					RW								RW			
Reset value																					F_H								F_H			

threshold value of input message FIFOs	
Bit 11 - 8	Rthresh specify the threshold of read message FIFO; when the fill level of the FIFO reaches this value an IRQ is asserted
Bit 3 - 0	Wthresh specify the threshold of write message FIFO; when the fill level of the FIFO reaches this value an IRQ is asserted

6.5.4.5. IntClrCfg0

Register address	BaseAddress + 10_H																													
Bit number	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
Field name	IntClrCfg0																															
R/W	RW																															
Reset value	0_H																															

configuration for interrupt 31 downto 0 clearing	
Bit 31 - 0	IntClrCfg0 IntClrCfg0[n]: automatic clear of interrupt request flag n: 0b=disabled, 1b=enabled
	Attention: Only Interrupts which support automatic clear (see the Event ID table, last column) can be set to “1b”. All others must be handled as Reserved bits, which are not allowed to enable

6.5.4.6. IntClrCfg1

Register address	BaseAddress + 14_H																													
Bit number	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
Field name	IntClrCfg1																															
R/W	RW																															
Reset value	0_H																															

configuration for interrupt 63 downto 32 clearing	
Bit 31 - 0	IntClrCfg1 IntClrCfg1[n]: automatic clear of interrupt request flag n+32: 0b=disabled, 1b=enabled
	Attention: Only Interrupts which support automatic clear (see the Event ID table, last column) can be set to “1b”. All others must be handled as Reserved bits, which are not allowed to enable

6.5.4.7. IntClrCfg2

Register address	BaseAddress + 18_H
Bit number	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
Field name	IntClrCfg2
R/W	RW
Reset value	0_H

configuration for interrupt 95 downto 64 clearing	
Bit 31 - 0	IntClrCfg2 IntClrCfg2[n]: automatic clear of interrupt request flag n+64: 0b=disabled, 1b=enabled
	Attention: Only Interrupts which support automatic clear (see the Event ID table, last column) can be set to “1b”. All others must be handled as Reserved bits, which are not allowed to enable

6.5.4.8. IntClrCfg3

Register address	BaseAddress + $1C_H$
Bit number	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
Field name	IntClrCfg3
R/W	RW
Reset value	0_H

configuration for interrupt 127 downto 96 clearing	
Bit 31 - 0	IntClrCfg3 IntClrCfg3[n]: automatic clear of interrupt request flag n+96: 0b=disabled, 1b=enabled
	Attention: Only Interrupts which support automatic clear (see the Event ID table, last column) can be set to “1b”. All others must be handled as Reserved bits, which are not allowed to enable

6.5.4.9. Mailbox

Register address	BaseAddress + 20_H
Bit number	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
Field name	Data
R/W	RW
Reset value	0_H

send a message to the host	
Bit 31 - 0	Data Writing to this register triggers an interrupt; an upstream event message with this payload will be sent to the host

6.5.4.10. IEN

Register address	BaseAddress + 24_H																													
Bit number	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
Field name																									IEnMail	IEnRError	IEnROverFlow	IEnRUT	IEnWError	IEnWOverFlow	IEnWUT
R/W																									RW	RW	RW	RW	RW	RW	RW
Reset value																									0_H	0_H	0_H	0_H	0_H	0_H	0_H

interrupt enable for respective interrupt

Bit 6	IEnMail Interrupt Enable for interrupt IstsMail : 0b=disabled, 1b=enabled
Bit 5	IEnRError Interrupt Enable for interrupt IstsRError : 0b=disabled, 1b=enabled
Bit 4	IEnROverFlow Interrupt Enable for interrupt IstsROverFlow : 0b=disabled, 1b=enabled
Bit 3	IEnRUT Interrupt Enable for interrupt IStsRUT : 0b=disabled, 1b=enabled
Bit 2	IEnWError Interrupt Enable for interrupt IStsWError : 0b=disabled, 1b=enabled
Bit 1	IEnWOverFlow Interrupt Enable for interrupt IStsWOverFlow : 0b=disabled, 1b=enabled
Bit 0	IEnWUT Interrupt Enable for interrupt IStsWUT : 0b=disabled, 1b=enabled

6.5.4.11. ISTS

Register address	BaseAddress + 28_H																															
Bit number	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
Field name																									IStsMail	IStsRError	IStsROverFlow	IStsRUT	IStsWError	IStsWOverFlow	IStsWUT
R/W																									RW	RW	RW	RW	RW	RW	RW
Reset value																									0_H	0_H	0_H	0_H	0_H	0_H	0_H

Interrupt status flags, a '1' signifies that the corresponding interrupt condition occurred (even if interrupt is disabled), write '1' clears the flag,	
Bit 6	IStsMail Interrupt Status for condition: Write to register Mailbox happened
Bit 5	IStsRError Interrupt Status for condition: AHB transaction not successfull AHB error
Bit 4	IStsROverFlow Interrupt Status for condition: Read transaction FIFO overflow, Transaction received and skipped
Bit 3	IStsRUT Interrupt Status for condition: Read transaction FIFO fill level over UpperThreshold; IStsRUT=0 -> level is reached when FIFO contains 1 message; IStsRUT=15 -> level is reached when FIFO contains 16 message;
Bit 2	IStsWError Interrupt Status for condition: AHB transaction not successfull AHB error
Bit 1	IStsWOverFlow Interrupt Status for condition: Write transaction FIFO overflow, Transaction received and skipped
Bit 0	IStsWUT Interrupt Status for condition: Write transaction FIFO fill level over UpperThreshold; IStsWUT=0 -> level is reached when FIFO contains 1 message; IStsWUT=15 -> level is reached when FIFO contains 16 message;

6.5.4.12. AHBWriteErrorAddress

Register address	BaseAddress + $2C_H$																															
Bit number	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
Field name	ErrWAdr																															
R/W	R																															
Reset value	0_H																															

current AHB write address; this value is always updated when IStsWError is not asserted (for debug purpose only)	
Bit 31 - 0	ErrWAdr

	Address of the last not successfull AHB write transaction

6.5.4.13. AHBReadErrorAddress

Register address	BaseAddress + 30_H																													
Bit number	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
Field name	ErrRAdr																															
R/W	R																															
Reset value	0_H																															

current AHB read address; this value is always updated when IStsRError is not asserted (for debug purpose only)	
Bit 31 - 0	ErrRAdr Address of the last not successfull AHB read transaction

6.5.4.14. StatusFIFOEmpty

Register address	BaseAddress + 34_H																													
Bit number	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
Field name																															WriteFifoEmpty	ReadFifoEmpty
R/W																															R	R
Reset value																															0_H	0_H

Status "empty" of read and write transaction FIFOs	
Bit 1	WriteFifoEmpty Write transaction FIFO is empty 0b=FIFO is not empty, 1b=FIFO is empty
Bit 0	ReadFifoEmpty Read transaction FIFO is empty 0b=FIFO is not empty, 1b=FIFO is empty

6.5.4.15. APConfig

Register address	BaseAddress + $3C_H$																													
Bit number	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
Field name																												rapix_config			ap_configure	ap_restart
R/W																												RW			W	RW
Reset value																												0_H			0_H	0_H

Configuration register for A-shell and APIX PHY	
Bit 2	rapix_config resets APIX PHY and A-Shell, with falling edge the APIX PHY is configured; changes to configuration are allowed only when 'rapix_config' is asserted
Bit 1	ap_configure A-shell and APIX PHY are re-configured (also resulting in reset)
Bit 0	ap_restart A-shell stops operation, clears all state and history information and restarts when deasserted

6.5.4.16. PhyConfig0[1]

Register address	BaseAddress + 40_H																													
Bit number	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
Field name	config_byte_4								config_byte_3								config_byte_2								config_byte_1							
R/W	RW								RW								RW								RW							
Reset value[1]	0_H								0_H								0_H								0_H							

[1] For detailed description and reset values see 1.8

APIX PHY configuration bytes	
Bit 31 - 24	config_byte_4 see chapter 6.8
Bit 23 - 16	config_byte_3 see chapter 6.8
Bit 15 - 8	config_byte_2 see chapter 6.8
Bit 7 - 0	config_byte_1 see chapter 6.8

6.5.4.17. PhyConfig1[1]

Register address	BaseAddress + 44_H																											
Bit number	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
Field name									config_byte_7								config_byte_6								config_byte_5							
R/W									RW								RW								RW							
Reset value[1]									0_H								0_H								0_H							

APIX PHY configuration bytes	
Bit 23 - 16	config_byte_7 see chapter 6.8
Bit 15 - 8	config_byte_6 see chapter 6.8
Bit 7 - 0	config_byte_5 see chapter 6.8

6.5.4.18. ShellConfig0[1]

Register address	BaseAddress + 48_H																											
Bit number	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
Field name	config_byte_shell_4								config_byte_shell_3								config_byte_shell_2								config_byte_shell_1							
R/W	RW								RW								RW								RW							
Reset value[1]	0_H								0_H								0_H								0_H							

A-shell configuration bytes	
Bit 31 - 24	config_byte_shell_4 see chapter 6.8
Bit 23 - 16	config_byte_shell_3 see chapter 6.8
Bit 15 - 8	config_byte_shell_2 see chapter 6.8
Bit 7 - 0	config_byte_shell_1 see chapter 6.8

6.5.4.19. ASStatus

Register address	BaseAddress + $4C_H$																											
Bit number	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
Field name																					inbound_handshake	outbound_handshake	remote_ashell_restarted	protocol_error	rx_px_aligned	crc_timeout	rx_down_ready	ready_for_data	rx_crc_error	operational	rx_pll_good	fatal_error
R/W																					R	R	R	R	R	R	R	R	R	R	R	R
Reset value																					X	X	X	X	X	X	X	X	X	X	X	X

A-shell Status register	
Bit 11	inbound_handshake (none)
Bit 10	outbound_handshake (none)
Bit 9	remote_ashell_restarted (none)
Bit 8	protocol_error detected protocol error
Bit 7	rx_px_aligned indicates that APIX pixel stream path is operational
Bit 6	crc_timeout detected CRC timeout
Bit 5	rx_down_ready downstream serial channel is operational
Bit 4	ready_for_data A-shell is ready to accept outbound transaction
Bit 3	rx_crc_error CRC error of downstream/inbound data
Bit 2	operational A-shell is operational
Bit 1	rx_pll_good Fatal Error
Bit 0	fatal_error Fatal Error

6.5.4.20. ASCountRxCRC

Register address	BaseAddress + 50_H																															
Bit number	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
Field name																									rx_crc_error_cnt							
R/W																									R							
Reset value																									0_H							

number of detected CRC errors in downstream	
Bit 7 - 0	rx_crc_error_cnt number of detected CRC errors in downstream

6.5.4.21. ASCountSyncLoss

Register address	BaseAddress + 54_H																													
Bit number	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
Field name																									sync_loss_cnt							
R/W																									R							
Reset value																									0_H							

number of detected synchronisation losses of downstream serial channel	
Bit 7 - 0	sync_loss_cnt number of detected synchronisation losses of downstream serial channel

6.5.4.22. ASCountPLLBad

Register address	BaseAddress + 58_H																													
Bit number	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
Field name																									rx_pll_bad_cnt							
R/W																									R							
Reset value																									0_H							

number of detected PLL sync losses in downsteam	
Bit 7 - 0	rx_pll_bad_cnt number of detected PLL sync losses in downsteam

6.6. Processing Mode

6.6.1. Processing Flow

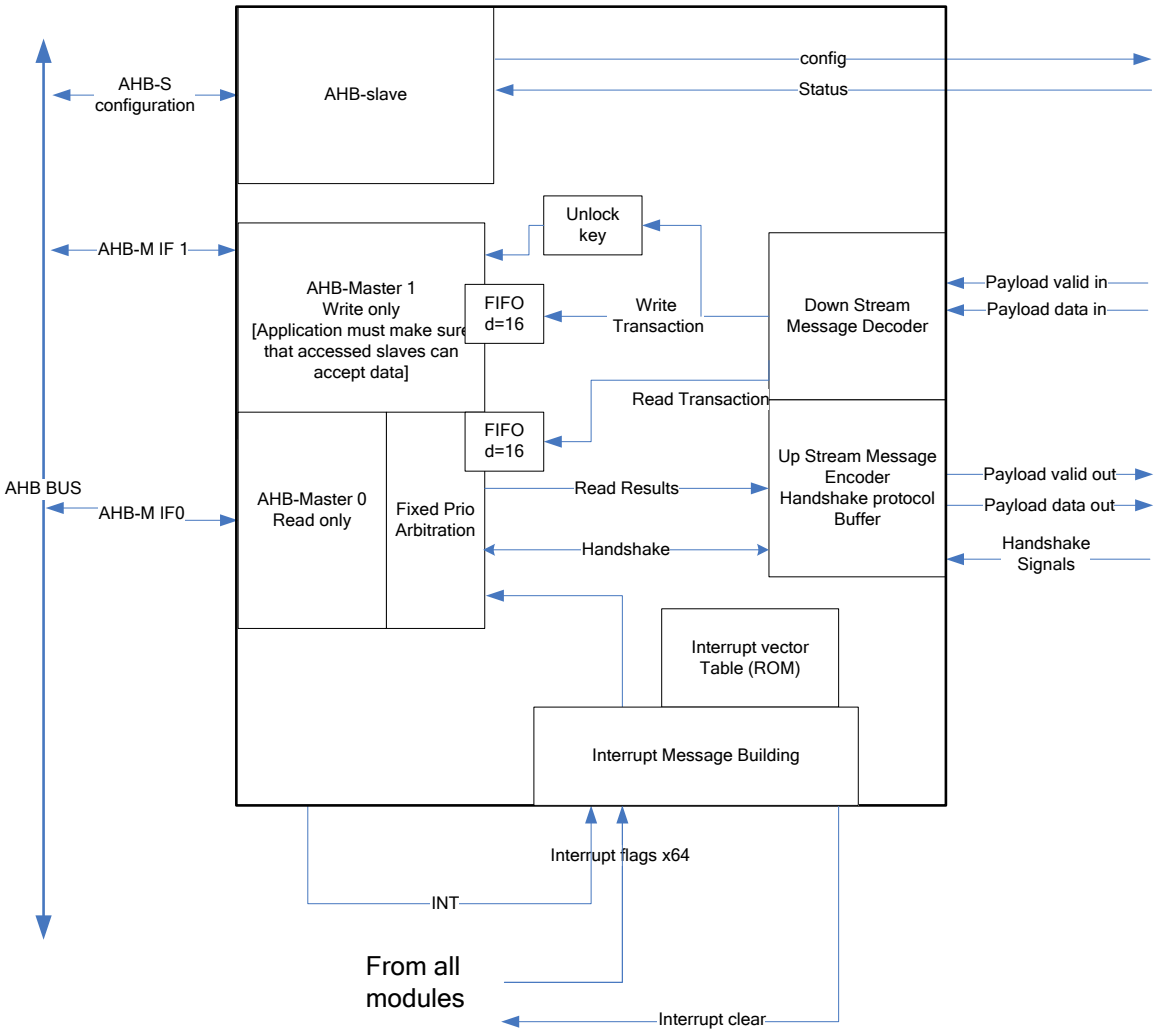


Figure 6-2 Block structure diagram

6.7. Control Flow

- APIX Configuration by boot-routine of command-sequencer from flash or
- (APIX Configuration by host MPU via APIX link possible, but not recommended)

Remote handler TX must perform the following steps after power up of Indigo-L:

1. Poll “connected” bit of TX remote handler (wait until TX and RX Ashell are aligned)
2. Repeat read requests (e.g. to register AHBMlock) until one read result is successfully received
   OR
   Wait until first event message from Indigo (event ID 101 = command sequencer boot-up sequence finished) is received at TX remote handler
3. Send unlock message to enable AHB write access.
4. Configure Remote Handler FIFO thresholds, interrupts
5. Start configuration, control of MB88F333

Remark:
Step 2 is necessary to make sure that Indigo remote handler is released from reset state.

6.8. Reset Values for APIX_shell and APIX PHY config bytes

Mode Number	M0	M1	M2 (not recommended) *1
Use Case	Low-speed APIX Mode using Indigo Sprite Engine	High-Quality Dual Display Operation	APIX Ashell SW Emulation
Description	internal PHYs, low bandwidth e.g. MCU 467S ← → Indigo-L	internal PHYs, full/half bandwidth, 1bit Sideband *2 mcu ←→ jadeD ←→ Indigo-L	internal RX PHY, discrete TX PHY, Asynchronous SB communication. Ashell operates in Compatibility-, Integrity-only-, Payload-only-Mode

Table 6-1 List of use cases mapped to modes

Depending on the selected mode (M0, M1, M2), the reset value of a parameter is listed in the relevant columns M0, M1 or M2:

(*1) For a setting with synchronous transmission which is recommended, please see application note “APIX Ashell SW Emulation Use Case”
(*2) 2bit Sideband operation possible in APIX mode 2. Same reset values are valid as for Mode M1 except config_byte_shell_2 and config_byte_1

config_byte_1					
Bit	M0	M1	M2	Name	Description
7	0	0	1	cfg_up_clk_divider[1]	APIX PHY upstream channel bandwidth setting bandwidth mode of downstream link 1000 Mbit/s 125 MBit/s 500 Mbit/s 00: not applicable 62.50 MBit/s [2] 01: 62.50 MBit/s 31.25 MBit/s 10: 41.67 MBit/s 20.83 MBit/s 11: 31.25 MBit/s not applicable **Note**: upstream bandwidth setting has to match related transmitter device configuration
6	0	1 [3]	0	cfg_up_clk_divider[0]	
5	1	1	1	cfg_upDataSwing[0]	APIX PHY upstream serial output current swing (binary coded, 1 LSB = 0.65mA) 00000: min 0mA(typ) ... 11111: max 26mA(typ) **Note:** please adapt to used cable length...
4	1	1	1	cfg_upDataSwing[1]	
3	1	1	1	cfg_upDataSwing[2]	
2	1	1	1	cfg_upDataSwing[3]	
1	1	1	1	cfg_upDataSwing[4]	
0	1	1	0	cfg_sbup_smode	APIX PHY relation of upstream sideband data to core clock of APIX PHY 0: asynchronous, 1: synchronous

Table 6-2 config_byte_1

[2] If a setup with external TX Ashell is used and the external Ashell works with 62.5MHz core clock, then 62.5MBit/s upstream channel bandwidth is not supported

[3] If APIX_CFG[2:0]=111and PLL_MODE=0, then cfg_up_clk_divider[0]=0 after reset. Please consider also footnote 4.

config_byte_2					
Bit:	M0	M1	M2	Name	Description
7	0	1	1	cfg_pxdata_width[1]	APIX PHY bit width of pixel data 00: reserved (for ES2 10 bits) 01: reserved (for ES2 12bits) 10: 18 bits 11: 24 bits **Note:** width of pixel data setting has to match related transmitter device configuration
6	0	0	0	cfg_pxdata_width[0]	
5	0	1	1	cfg_px_out_ctrl_piggyback[1]	APIX PHY transmission of pixel control signals (px_ctrl[2:0], used for HSYNC, VSYNC, DE) 00: never 01: unused 10: with even pixels only 11: with every pixel **Note:** pixel control signals setting has to match related transmitter device configuration **Note:** to achieve maximum pixel link net bandwidth setting “10” is necessary, see APIX standard
4	0	1	1	cfg_px_out_ctrl_piggyback[0]	
3	1	1	1	Reserved	do not change
2	1	1	1	Reserved	do not change
1	0	0	0	Reserved	do not change
0	0	0	0	Reserved	do not change

Table 6-3 config_byte_2

config_byte_3					
Bit:	M0	M1	M2	Name	Description
7	1	1	1	reserved	do not change
6	0	0	0	reserved	do not change
5	0	0	0	reserved	do not change
4	0	0	0	reserved	
3	0	0	0	reserved	
2	1	1	1	reserved	do not change
1	1	1	1	reserved	do not change
0	0	0	0	reserved	do not change

Table 6-4 config_byte_3

config_byte_4					
Bit:	M0	M1	M2	Name	Description
7	0	0	0	reserved	do not change
6	0	0	0	reserved	do not change
5	0	0	0	reserved	do not change
4	0	0	0	reserved	do not change
3	0	0	0	reserved	do not change
2	0	0	0	reserved	do not change
1	0	0	0	reserved	do not change
0	0	0	0	reserved	do not change

Table 6-5 config_byte_4

config_byte_5					
Bit:	M0	M1	M2	Name:	Description:
7	0	0	0	cfg_crgPmpCtrl[0]	Do not change APIX PHY Charge pump current control bits
6	1	1	1	cfg_crgPmpCtrl[1]	
5	1	1	1	cfg_crgPmpCtrl[2]	
4	0	0	0	cfg_crgPmpCtrl[3]	
3	0	0	0	cfg_pwrDownDwn	APIX PHY (analog): power down downstream path 1: power down 0: power up
2	0	0	0	reserved	do not change
1	0	0	0	cfg_pwrDownUp	APIX PHY (analog): power down upstream path 1: power down 0: power up
0	1	1	1	reserved	do not change

Table 6-6 config_byte_5

config_byte_6					
Bit	M0	M1	M2	Name	Description
7	0	1	1	cfg_downBwMode[1]	APIX PHY selects downstream bandwidth mode 11: 1000 MBit/s (Full Bandwidth Mode) 10: 500 MBit/s (Half Bandwidth Mode) 00: 125 MBit/s (Low Bandwidth Mode 1) 01: not applicable **Note**: downstream bandwidth setting has to match related transmitter device configuration
6	0	0[4] / 1[5]	0 / 1	cfg_downBwMode[0]	
5	1	0	0	cfg_ddown_enable	APIX PHY / AShell: configure downstream data path 0: disable data mode / enable pixel stream mode 1: enable data mode / disable pixel stream mode **note:** for proper operation of 'data mode' the following settings are also mandatory cfg_pxdata_width[1:0] := '00' cfg_px_out_ctrl_piggyback[1:0] := '00'
4	0	0	0	reserved	do not change
3	0	0	0	reserved	do not change
2	0	0	0	reserved	do not change
1	0	0	0	reserved	do not change
0	0	0	0	reserved	do not change

Table 6-7 config_byte_6

[4] Pin PLL_MODE=0 (PLL speed 250MHz)
[5] Pin PLL_MODE=1 (PLL speed 500MHz)

config_byte_7					
Bit	M0	M1	M2	Name	Description
7	1	1	1	reserved	do not change
6	0	0	0	reserved	
5	0	0	0	reserved	
4	1	1	1	reserved	
3	0	0	0	reserved	do not change
2	0	0	0	reserved	do not change
1	1	1	1	reserved	do not change
0	1	1	1	reserved	do not change

Table 6-8 config_byte_7

config_byte_shell1					
Bit	M0	M1	M2	Name	Description
7	0	0	(1)	reserved	Do not change
6	(0)	(0)	0	cfg_sbup_daclk_clength[10]	AShell configures data rate of upstream sideband (see tables below) **Note:** valid if cfg_sbup_daclk[1:0] = "10"
5	(0)	(0)	0	cfg_sbup_daclk_clength[9]	
4	(0)	(0)	0	cfg_sbup_daclk_clength[8]	
3	(0)	(0)	1	cfg_sbup_daclk_clength[7]	
2	(0)	(0)	0	cfg_sbup_daclk_clength[6]	
1	(0)	(0)	0	cfg_sbup_daclk_clength[5]	
0	(0)	(0)	0	cfg_sbup_daclk_clength[4]	

Table 6-9 config_byte_shell1

cfg_downBWMode[1:0]	cfg_up_clk_divider[1:0]	cfg_sbup_daclk_clength[10:0] **supported minimum value**
11	01	14
11	10	20
11	11	26
10, 00	00	8
10, 00	01	14
10, 00	10	20

Table 6-10 Rule for minimum cfg_sbup_daclk_clength parameter

cfg_spi_over_sb	cfg_downBWMode	**C** = cfg_sbup_daclk_clength[10:0] **resulting data rate (Mbit/s)**
0	11	$125 * 10^6 / C$
0	10, 00	$62,5 * 10^6 / C$
1	11	$125 * 10^6 / (2 * C)$
1	10, 00	$62,5 * 10^6 / (2 * C)$

Table 6-11 Formula for resulting uplink datarate

config_byte_shell2					
Bit	M0	M1	M2	Name	Description
7	(1)	(1)	0	cfg_sbup_daclk_clength[3]	AShell configures data rate **Note:** valid if cfg_sbup_daclk[1:0] = "10"
6	(0)	(1)	0	cfg_sbup_daclk_clength[2]	
5	(0)	(1)	0	cfg_sbup_daclk_clength[1]	
4	(0)	(1)	0	cfg_sbup_daclk_clength[0]	
3	1	0[6]	1	cfg_sbup_dwidth	AShell enable sbup ports 1: sbup_data[1:0] 0: sbup_data[0]
2	0	0	1	cfg_sbup_daclk[1]	AShell: generate sbup clock and transmit as sbup_data[1] 11: disable 10: with use of internal counter (asynchronous to core_clk of APIX PHY) 01: reserved 00: disable
1	0	0	0	cfg_sbup_daclk[0]	
0	1	0[7]	1	cfg_sbdown_dwidth	AShell enable sbdown ports 1: sbdown_data[1:0] 0: sbdown_data[0]

Table 6-12 config_byte_shell2

[6] If APIX mode is 2, then cfg_sbup_dwidth = 1
[7] If APIX mode is 2, then cfg_sbdown_dwidth = 1

config_byte_shell3					
Bit	M0	M1	M2	Name	Description
7	0	0	1	cfg_sbdown_daclk	AShell: validate sbdown_data with 1: sbdown_data[1], 0: internal signal (sbdown_valid)
6	0	0	0	Reserved	Do not change
5	0	0	0	Reserved	Do not change
4	0	0	1	cfg_spi_over_sb	AShell: selects between two different sideband transmission modes 0: toggle mode see Figure 6-4 SPI-mode 1: SPI mode see Figure 6-4 SPI-mode
3	1	1	1	cfg_crc_timeout_value [3]	AShell CRC timeout error is generated after N consecutively received and corrupted transitions (CRC mismatch) N = factor1 * factor2 factor1 = cfg_crc_timeout_value [3:2] factor2 = cfg_crc_timeout_value [1:0] factor 1 / factor 2 00: 1 / 00: 2 01: 4 / 01: 4 10: 16 / 10: 6 11: 128 / 11: 10 example: 1001 → N = 64 (16*4) Note: to achieve optimum system behaviour, please adapt to bit fault characteristics of serial link
2	0	0	0	cfg_crc_timeout_value [2]	
1	0	0	0	cfg_crc_timeout_value [1]	
0	1	1	1	cfg_crc_timeout_value [0]	

Table 6-13 config_byte_shell3

config_byte_shell4					
Bit	M0	M1	M2	Name	Description
7	1	1	1	reserved cfg_window_size[3]	AShell do not change defines the window size of the acknowledge protocol (supported size: 1...12)
6	0	0	0	reserved cfg_window_size[2]	
5	1	1	1	reserved cfg_window_size[1]	
4	0	0	0	reserved cfg_window_size[0]	
3	0	0	1	cfg_arq_off	AShell selects "data integrity only" mode
2	0	0	1	cfg_suppress_ita	AShell selects "payload only" mode
1	0	0	0	reserved	
0	0	0	0	reserved	

Table 6-14 config_byte_shell4

6.9. GPIO Interface of Sideband Up- and Downlink

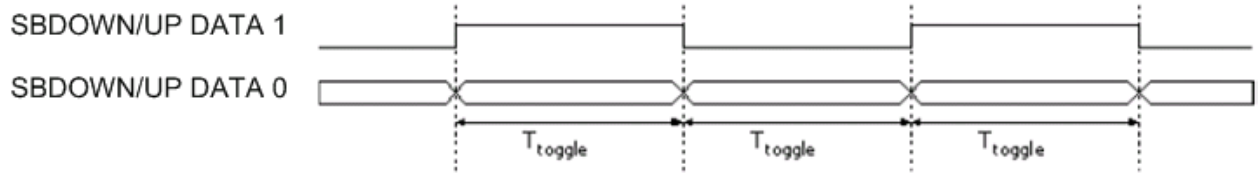


Figure 6-3 Toggle Mode

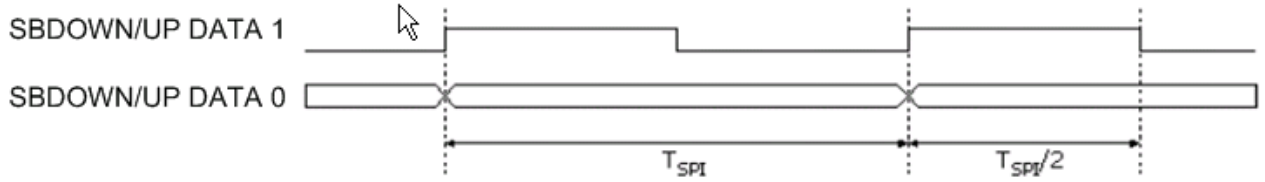


Figure 6-4 SPI-mode

6.10. Configuration of upstream sideband data rate (cfg_sbup_daclk_clength)

Under the configuration cfg_sbup_daclk[1:0] = “10” the RX AShell generates its own controlling bit for outbound data and transmits it together with the outbound data to the RX APIX PHY which is establishing the connection to the TX side (upstream channel).
The cycle length C of the control bit is configurable by config_byte_shell_1[6:0] and config_byte_shell_2[7:4] parameter named cfg_sbup_daclk_clength. The lower limit of cycle length C_{min} is defined by the sampling rate T_{sbup} of the TX APIX PHY (refer to Table 6-15 sampling rate sideband upstream channel), which varies with the downstream bandwidth and the upstream bandwidth and the upper limit C_{max} by the bit width of the counter.

Downstream Bandwidth	Upstream Bandwidth [MBit/s]	$T_{\_sbup}$[ns]
Full bandwidth	62.5	48
	41.67	72
	31.25	96
Half bandwidth	62.5	48
	31.25	96
	20.83	144
Low bandwidth 2	62.5	48
	31.25	96
	20.83	144
Low bandwidth 1	62.5	48
	31.25	96
	20.83	144

Table 6-15 sampling rate sideband upstream channel

The following formula is used to calculate the lower limit of cycle length C_{min}

$$C_{min} = \left\lceil \frac{(2*(T_{sbup} + T^{RX}_{core\_clk\_phy}) + 4*T^{TX}_{core\_clk\_ashell})}{T^{RX}_{core\_clk\_ashell}} \right\rceil \qquad T_{toggle\_min} = \frac{T_{SPI\_min}}{2} = C_{min} * T^{RX}_{core\_clk\_ashell}$$

Equation 1

For Full bandwidth operation $T^{RX}_{core\_clk\_phy} = T^{RX}_{core\_clk\_ashell}$ = 8ns.
For all other bandwidth modes $T^{RX}_{core\_clk\_phy} = T^{RX}_{core\_clk\_ashell}$ = 16ns

Note: If at TX side no external Ashell is used, $T^{TX}_{core\_clk\_ashell}$ is 0ns.

$$C = \left\lceil \frac{T_{toggle}}{T^{RX}_{core\_clk\_ashell}} \right\rceil \qquad C_{max} \le 2047\ (11\ bit\ counter)$$

Figure 6-5 illustrates the correlations between cycle length C and toggle period T_{toggle}.

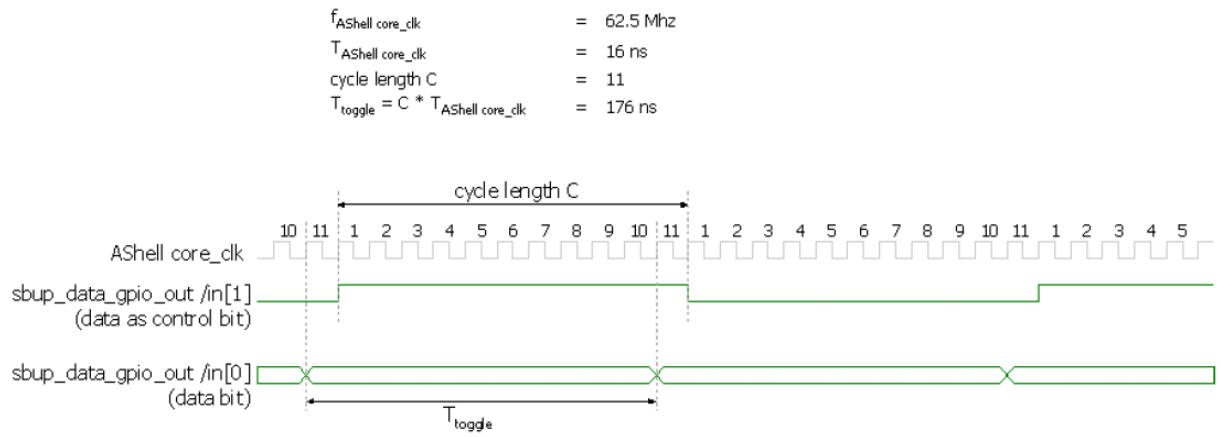


Figure 6-5 timing diagram of toggle mode

Figure 6-6 illustrates the correlations between cycle length C and SPI period T_{SPI}.

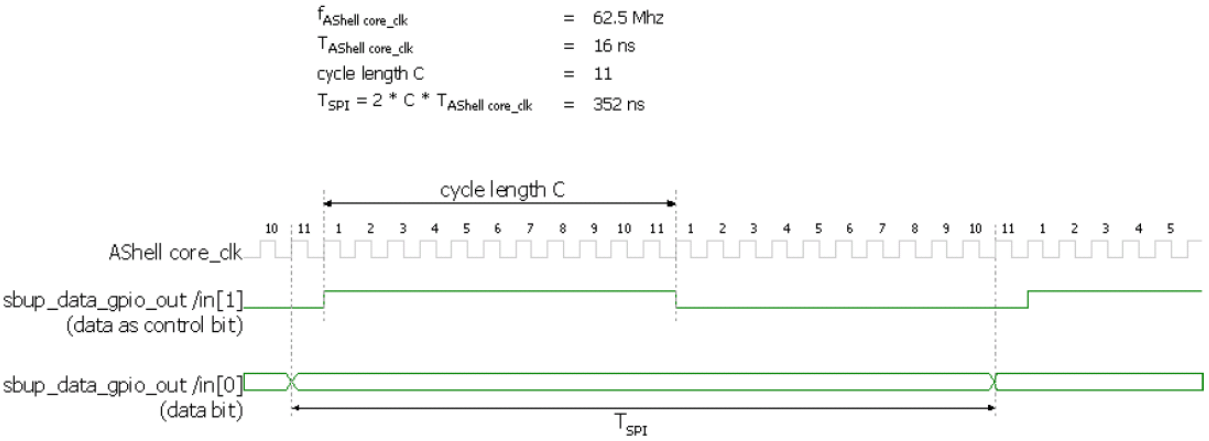


Figure 6-6, timing diagram of SPI mode

6.11. Event ID table

Event ID	Module Name		Interrupt Conditions	Reference Register		Payload for eventmessage			Automatic clear by HW possible
				Base Addr	Address Offset	Base Addr	Address Offset	Register name Description	
0	Reserved								
1	RemoteHandler		write threshold reached	0x11000	0x20		NOP		yes
2	RemoteHandler		read threshold reached	0x11000	0x20		NOP		yes
3	RemoteHandler		write FIFO overflow	0x11000	0x20		NOP		yes
4	RemoteHandler		read FIFO overflow	0x11000	0x20		NOP		yes
5	RemoteHandler		mailbox interrupt	0x11000	0x20	0x11000	0x20	Mailbox	yes
6	RemoteHandler		write bus error	0x11000	0x20	0x11000	0x2C	AHBWriteErrorAddress	yes
7	RemoteHandler		read bus error	0x11000	0x20	0x11000	0x30	AHBReadErrorAddress	yes
8	Config FIFO		Slave Module HRESP=ERROR	0x12000	0x00	0x12000	0x0	FFISTS Register	yes
9	Config FIFO		Underflow channel(i) i=7	0x12000	0x00		NOP		yes
10	Config FIFO		Overflow channel(i) i=7	0x12000	0x00		NOP		yes
11	Config FIFO		Underflow channel(i) i=6	0x12000	0x00		NOP		yes
12	Config FIFO		Overflow channel(i) i=6	0x12000	0x00		NOP		yes
13	Config FIFO		Underflow channel(i) i=5	0x12000	0x00		NOP		yes
14	Config FIFO		Overflow channel(i) i=5	0x12000	0x00		NOP		yes
15	Config FIFO		Underflow channel(i) i=4	0x12000	0x00		NOP		yes
16	Config FIFO		Overflow channel(i) i=4	0x12000	0x00		NOP		yes
17	Config FIFO		Underflow channel(i) i=3	0x12000	0x00		NOP		yes
18	Config FIFO		Overflow channel(i) i=3	0x12000	0x00		NOP		yes
19	Config FIFO		Underflow channel(i) i=2	0x12000	0x00		NOP		yes

20	Config FIFO		Overflow channel(i) i=2	0x12000	0x00		NOP		yes
21	Config FIFO		Underflow channel(i) i=1	0x12000	0x00		NOP		yes
22	Config FIFO		Overflow channel(i) i=1	0x12000	0x00		NOP		yes
23	Config FIFO		Underflow channel(i) i=0	0x12000	0x00		NOP		yes
24	Config FIFO		Overflow channel(i) i=0	0x12000	0x00		NOP		yes
25	Config FIFO		upper threshold channel(i) i=7	0x12000	0x4A0	0x12000	0x49C	FFStatus7 "fill level"	yes
26	Config FIFO		lower threshold channel(i) i=7	0x12000	0x4A8	0x12000	0x49C	FFStatus7 "fill level"	yes
27	Config FIFO		upper threshold channel(i) i=6	0x12000	0x420	0x12000	0x41C	FFStatusi "fill level"	yes
28	Config FIFO		lower threshold channel(i) i=6	0x12000	0x428	0x12000	0x41C	FFStatusi "fill level"	yes
29	Config FIFO		upper threshold channel(i) i=5	0x12000	0x3A0	0x12000	0x39C	FFStatusi "fill level"	yes
30	Config FIFO		lower threshold channel(i) i=5	0x12000	0x3A8	0x12000	0x39C	FFStatusi "fill level"	yes
31	Config FIFO		upper threshold channel(i) i=4	0x12000	0x320	0x12000	0x31C	FFStatusi "fill level"	yes
32	Config FIFO		lower threshold channel(i) i=4	0x12000	0x328	0x12000	0x31C	FFStatusi "fill level"	yes
33	Config FIFO		upper threshold channel(i) i=3	0x12000	0x2A0	0x12000	0x29C	FFStatusi "fill level"	yes
34	Config FIFO		lower threshold channel(i) i=3	0x12000	0x2A8	0x12000	0x29C	FFStatusi "fill level"	yes
35	Config FIFO		upper threshold channel(i) i=2	0x12000	0x220	0x12000	0x21C	FFStatusi "fill level"	yes
36	Config FIFO		lower threshold channel(i) i=2	0x12000	0x228	0x12000	0x21C	FFStatusi "fill level"	yes
37	Config FIFO		upper threshold channel(i) i=1	0x12000	0x1A0	0x12000	0x19C	FFStatus1 "fill level"	yes
38	Config FIFO		lower threshold channel(i) i=1	0x12000	0x1A8	0x12000	0x19C	FFStatus1 "fill level"	yes
39	Config FIFO		upper threshold channel(i) i=0	0x12000	0x120	0x12000	0x11C	FFStatus0 "fill level"	yes
40	Config FIFO		lower threshold channel(i) i=0	0x12000	0x128	0x12000	0x11C	FFStatus0 "fill level"	yes
41	Reserved								
42	RLT(RBUS)	0	TMCSR:UF	0x00000	0x1B6		NOP		yes
43	RLT(RBUS)	1	TMCSR:UF	0x00000	0x1BE		NOP		yes
44	RLT(RBUS)	2	TMCSR:UF	0x00000	0x1C6		NOP		yes
45	RLT(RBUS)	3	TMCSR:UF	0x00000	0x1CE		NOP		yes
46	RLT(RBUS)	4	TMCSR:UF	0x00000	0x1D6		NOP		yes
47	RLT(RBUS)	5	TMCSR:UF	0x00000	0x1DE		NOP		yes

48	RLT(RBUS)	6	TMCSR:UF (for alive message)	0x00000	0x1E6	0x11000	0x4C	RH:ASStatus	yes
49	RLT(RBUS)	7	TMCSR:UF (ADC)	0x00000	0x1EE		NOP		yes
50	RLT(RBUS)	8	TMCSR:UF	0x00000	0x596		NOP		yes
51	RLT(RBUS)	9	TMCSR:UF	0x00000	0x59E		NOP		yes
52	RLT(RBUS)	10	TMCSR:UF	0x00000	0x5A6		NOP		yes
53	RLT(RBUS)	11	TMCSR:UF	0x00000	0x5AE		NOP		yes
54	RLT(RBUS)	12	TMCSR:UF	0x00000	0x5B6		NOP		yes
55	RLT(RBUS)	13	TMCSR:UF	0x00000	0x5BE		NOP		yes
56	RLT(RBUS)	14	TMCSR:UF (alive second read)	0x00000	0x5C6	0x00000	0x8F8	CCNT Dummy Reserve2	yes
57	RLT(RBUS)	15	TMCSR:UF (general purpose)	0x00000	0x5CE		NOP		yes
58	USART4 (RBUS)	4	SSR.TDRE	0x00000	0x64		NOP	-	yes
59	USART5 (RBUS)	5	SSR.TDRE	0x00000	0x64		NOP	-	yes
60	USARRX(RBUS) 2x	4	SSR.TDRE SSR:RDRF SSR:ORE, FRE, PE ESCR:LBD	0x00000	0x64	0x00000	0x64	SSR,RDR,FSR, FCR	yes
61	USARRX(RBUS) 2x	5	SSR.TDRE SSR:RDRF SSR:ORE, FRE, PE ESCR:LBD	0x00000	0x64	0x00000	0x64	SSR,RDR,FSR, FCR	yes
62	PPG(RBUS	0	PCN.IRQF	0x00000	0x116		NOP		yes
63	PPG(RBUS	1	PCN.IRQF	0x00000	0x11E		NOP		yes
64	PPG(RBUS	2	PCN.IRQF	0x00000	0x126		NOP		yes
65	PPG(RBUS	3	PCN.IRQF	0x00000	0x12E		NOP		yes
66	PPG(RBUS	4	PCN.IRQF	0x00000	0x136		NOP		yes
67	PPG(RBUS	5	PCN.IRQF	0x00000	0x13E		NOP		yes

68	PPG(RBUS	6	PCN.IRQF	0x00000	0x146		NOP		yes
69	PPG(RBUS	7	PCN.IRQF	0x00000	0x14E		NOP		yes
70	PPG(RBUS	8	PCN.IRQF	0x00000	0x156		NOP		yes
71	PPG(RBUS	9	PCN.IRQF	0x00000	0x15E		NOP		yes
72	PPG(RBUS	10	PCN.IRQF	0x00000	0x166		NOP		yes
73	PPG(RBUS	11	PCN.IRQF	0x00000	0x16E		NOP		yes
74	I2C(RBUS)	0	IBCR:INT, BER	0x00000	0xD0	0x00000	0xD0		yes
75	SoundPWM(RBUS)		SGCRH:INT	0x00000	0x199		NOP		yes
76	ADC(RBUS)		ADCS1:INT	0x00000	0x1A4	0x00000	0x1A4		yes
77	Reserved								
78	GPIO_COVER		1. External interrupt_0 (input)	0x15800	0x04	0x15800	0x04	status of external requests	
79	GPIO_COVER		1. External interrupt_1 (input)	0x15800	0x04	0x15800	0x04	status of external requests	
80	GPIO_COVER		1. External interrupt_2 (input)	0x15800	0x04	0x15800	0x04	status of external requests	
81	GPIO_COVER		1. External interrupt_3 (input)	0x15800	0x04	0x15800	0x04	status of external requests	
82	GPIO_COVER		1. External interrupt_4 (input)	0x15C00	0x04	0x15C00	0x04	status of external requests	
83	GPIO_COVER		1. External interrupt_5 (input)	0x15C00	0x04	0x15C00	0x04	status of external requests	
84	GPIO_COVER		1. External interrupt_6 (input)	0x15C00	0x04	0x15C00	0x04	status of external requests	
85	GPIO_COVER		1. External interrupt_7 (input)	0x15C00	0x04	0x15C00	0x04	status of external requests	
86	Display Controller		1. Vertical synchronization (1frame end interrupt)	0x30000	-		NOP	-	
87	Display Controller		1. Horizontal synchronization (1line end interrupt)	0x30000	-		NOP	-	
88	Reserved		-		-			-	-
89	Display Controller		1. External VSYNC error	0x30000	-		NOP	-	
90	Display Controller		1. External HSYNC error	0x30000	-		NOP	-	
91	Display Controller		1. Pixel FiFo overflow	0x30000	-		NOP	-	

92	Display Controller		1. Pixel FiFo underflow	0x30000	-		NOP	-	
93	Display Controller		1. VPWM0 pulse width change demand 2. VPWM1 pulse width change demand 3. VPWM2 pulse width change demand 4. VPWM3 pulse width change demand	0x30000	0x4200	0x30000	0x4200		
94	SIG		Shadow took over register	0x2A000	0x5C		NOP	-	yes
95	SIG		Calculation finished	0x2A000	0x5C	0x2A000	0x68	Signature result R	
96	SIG		Calculation finished	0x2A000	0x5C	0x2A000	0x6C	Signature result G	
97	SIG		Calculation finished	0x2A000	0x5C	0x2A000	0x70	Signature result B	yes
98	SIG		error counter reached	0x2A000	0x5C		NOP	-	yes
99	SIG		combined interrupt 1)error counter reached,2) Calculation finished, 3) Shadow took over register	0x2A000	0x5C	0x2A000	0x5C	Interrupt Status Register	
100	RLD		1) Byte count achieved (=finished decoding), 2) AHB Slave Error, 3) input FIFO empty, 4) Input FIFO full	0x44000	0x28	0x44000	0x28	Interrupt Status Register	

101	Command Sequencer		1. End of reset start (End of commandlist prosessing) 2. End of interrupt start (End of commandlist prosessing) 3. End og register start (End of commandlist prosessing) 4. Forced ending of commandlist prosessing 5. Error of ahb slave access 6. Error of comparison 7. End of softwear reset start (End of commandlist prosessing)	0x47000	0x00	0x47000	0x00	Status Register	
102	Reserved		-	-	-	-	-	-	
103	Watchdog		1. Timer expiration	0x47000			NOP		
104	Reserved		-		-			-	
105	Host IF		1. Interrupt by HRESP=ERR	0x10000	0x800	0x10000	0x800	Status Register INT21 (INTerrupt 21)	
106	MEMIF		1. End of Flash Automatic algorithm	0x42000	0x00	0x42000	0x00	Status Register	

107	MEMIF		1. Initiali setting error 2. Write access in Flash Automatic algorithm 3. Ahb slave data interface write access error	0x42000		0x42000	0x00	Status Register	
108	HDMAC		[CH1] 1. Address overflow(Error) 2. Transfer stop request(Error) 3. Source access error(Error) 4. Destination access error(Error) 5. Normal end	0x40000	0x14	0x40000	0x14	DMACB Register (CH1)	
109	HDMAC		[CH2] 1. Address overflow(Error) 2. Transfer stop request(Error) 3. Source access error(Error) 4. Destination access error(Error) 5. Normal end	0x40000	0x24	0x40000	0x24	DMACB Register (CH2)	
110	Reserved								
111	Sprite GDC		1. "EnableBitsChanged"-Interrupt	0x20000	-		NOP	-	

112	Sprite GDC		1. Bus error interrupt	0x20000	-		NOP	-	
113	Sprite GDC		1. Processing error interrupt	0x20000	-		NOP	-	
114	Sprite GDC		1. Line blank interrupt	0x20000	-		NOP	-	
115	Sprite GDC		1. Specified line processing over interrupt	0x20000	-		NOP	-	
116	ClkSynth		clock is/was hanging, restart needed	0x14000	0x10	0x14000	0x10	Status Register	yes
117	Reserved								
118	Reserved								
119	Reserved								
120	Reserved								
121	Reserved								
122	Reserved								
123	Reserved								
124	Reserved								
125	Reserved								
126	Reserved								
127	Reserved								

7. Command Sequencer

This chapter describes the Command Sequencer of the MB88F333 LSI.

7.1. Outline

The Command Sequencer Module (CMDSEQ) interprets the command list in a specified area and processes the sequence. CMDSEQ is started by a reset release, trigger signal and register setting. A watchdog timer (WDT) is built into this module.

7.2. Features

The Command Sequencer has following features.

7.2.1. Features

- The command list can be processed.
- This module can be started as follows.
  - ⇒ Start by power-on reset release
  - ⇒ Start by trigger signal
  - ⇒ Start by register setting
- Built in Buffer (32 bit, 16 word)
- It is possible to forced termination while processing the command list.
- Detecting the following error ends the processing of the command list.
  - ⇒ Disagreement of expected value by COMPREG command
  - ⇒ Slave module access error
- Two kinds of interrupt signals are output.
  - ⇒ Interrupt to processing of command list (o_INT)
  - ⇒ Interrupt to time-out of WDT (o_WINT)

7.2.2. Limitations

- Early burst termination is unsupported.

7.3. Function

7.3.1. Block diagram

Figure 7.3-1 shows a block diagram of the Command Sequencer.

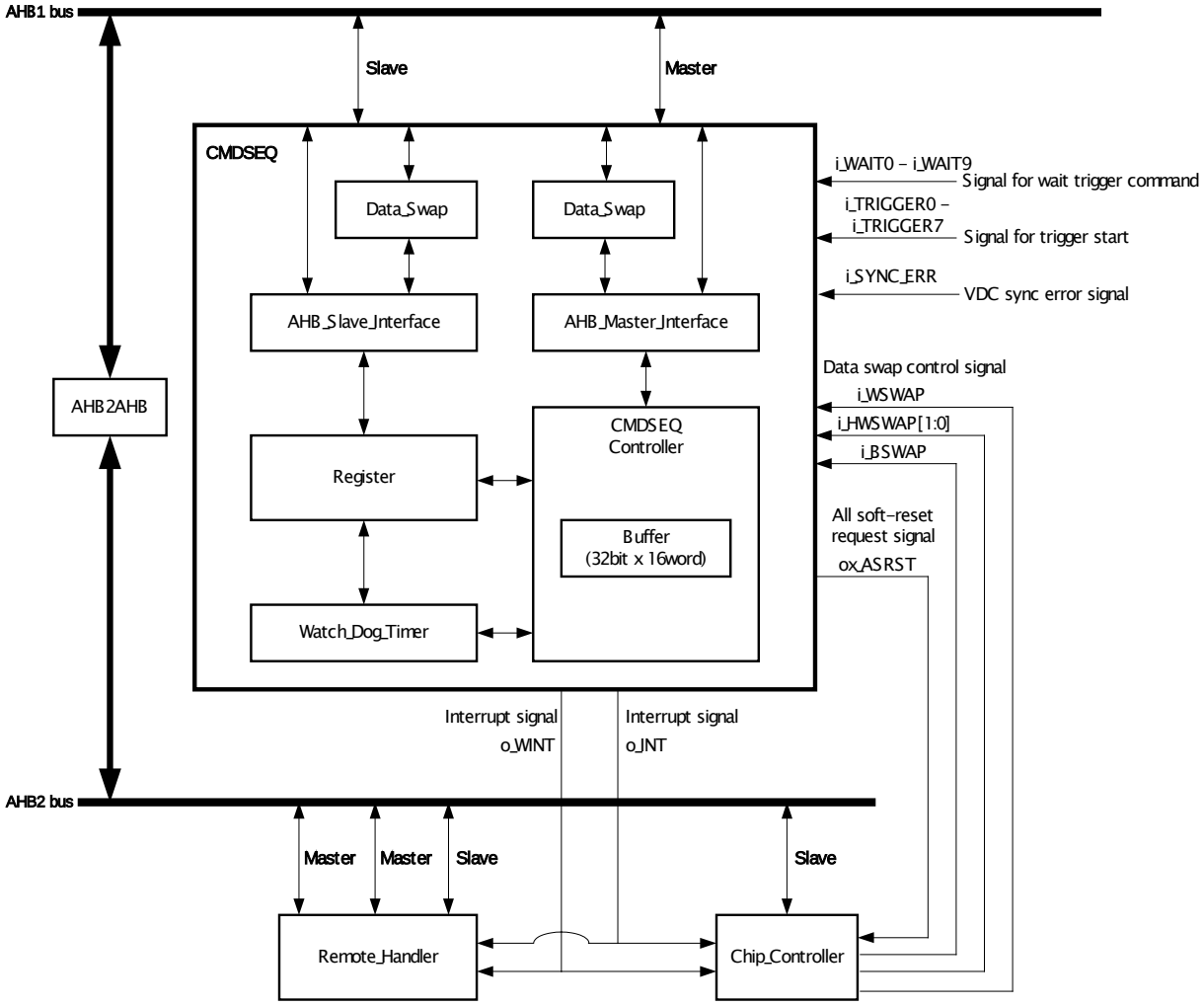


Figure 7.3-1 Block diagram of Command Sequencer

7.3.2. Starting Methods

The method of starting this module has “Reset start”, “Trigger start”, and “Register start”. There is an execution method in each start way.
Execution method in each start way is shown in the following.

- Reset start
  ⇒ Release of power-on reset.
- Trigger start
  ⇒ Interrupt signal by WDT (o_WINT) or interrupt signal by Signature Generator (SIG) (SIG_Diff_interrupt) signal of Video Display Controller (VDC) (i_SYNC_ERR) or trigger signal (i_TRIGGER0 - i_TRIGGER7).
- Register start
  ⇒ It starts by Start register of this module.

7.3.2.1. Reset start

This function processes the command list by releasing the power-on reset. When this function is executed, the data at Embedded Flash Memory base address＋0x00 is read. The value of the read data is an address value where the command list is stored. This module processes the command list at the address sequentially (refer to Figure 7.3-2 Reset start).
When a set address of the Embedded Flash Memory base address + 0x00 is 0xFFFF_FFFF, CMDSEQ ends the processing of the command list and sets “1” in the RESET bit.
If CMDSEQ request is Priority 1 while executing Reset start, processing is cancelled after the transaction execution has ended, and the request of Priority 1 is accepted. Afterwards, other operations are accepted (please refer to 7.3.4 Priority).

After processing completion, this module sets the RESET bit of Status register to “1”. When the RESETE bit of the status enable register is effective, this module outputs the state of the RESET bit to the interrupt signal (o_INT).

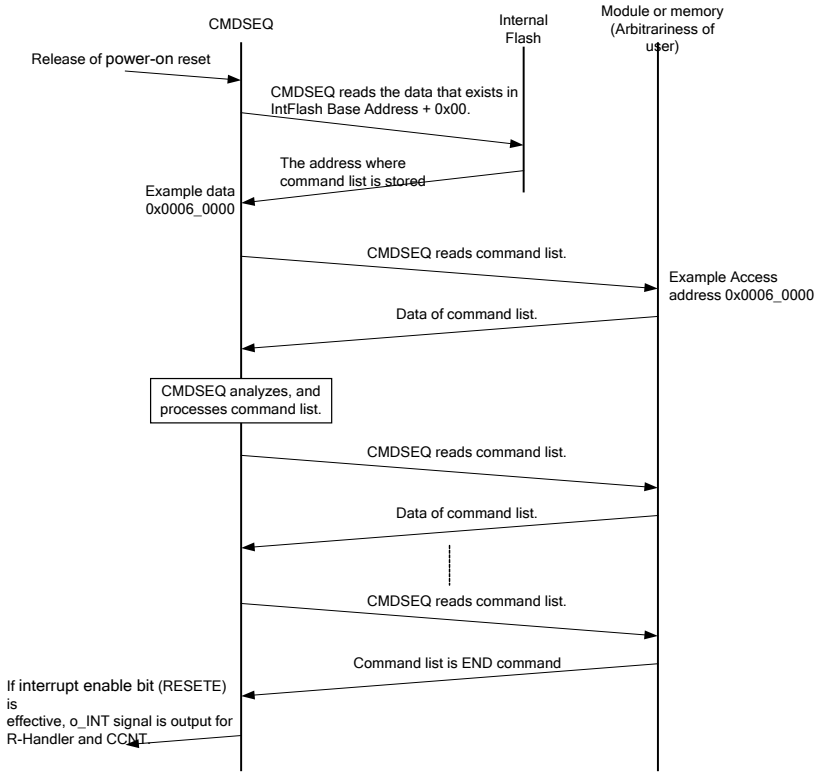


Figure 7.3-2 Reset start

7.3.2.2. Trigger start

When the trigger signal changes to “1”, CMDSEQ executes the command list. There are interrupt signals for WDT (o_WINT), SIG (SIG_Diff_interrupt), VDC (i_SYNC_ERR) and a trigger signal for which a user can arbitrarily set a signal (i_TRIGGER0 – 7). The interrupt signal of WDT and SIG are expressed as Priority1. The interrupt signal of VDC is expressed as Priority2. The trigger signal is expressed as Priority3. Only one Priority3 signal can be selected using the TRIGGER start enable register.

By setting the PRI1EW or PRI1ES bit of the TRIGGER start enable register to "1", the signal of Priority1 becomes effective as a Trigger signal. The trigger start by Priority1 signal is executed after an All Soft Reset release by setting the AWDT bit or the ASIG bit of the All Soft Reset enable register to “1”. If the AWDT or ASIG bit of the All soft reset enable register is “1” and the WDT or SIG interrupt signal is “1”, an All soft reset is executed regardless of the state of this bit.

By setting the PRI2E bit of the TRIGGER start enable register to “1”, only 1 signal becomes effective as a Trigger signal.

Only one signal can choose a Priority3 signal as the Trigger start via the PRI3E bit of the TRIGGER start enable register.

When this function is executed, the data in the Embedded Flash Memory Base Address＋0x04 – 0x28) is read. The value of the read data is an address value where the command list is stored. This module processes the command list at the address sequentially.

When the address in the Embedded Flash Memory Base Address＋0x04-0x28 is 0xFFFF_FFFF, the CMDSEQ module ends command list processing and doesn't write “1” to the TRIG bit (please refer to Figure 7.3-3 Start condition of trigger start).

After processing the command list, starting with Priority1, this module sets the TWDTS bit of the Status register to “1”. When the TWDTSE bit of the Status enable register is effective, this module outputs the state of the TWDTS bit to the interrupt signal (o_INT). After processing the command list starting with Priority2, this module sets the SYNCE bit of the Status register to “1”. When the SYNCEE bit of the Status enable register is effective, this module outputs the state of the SYNCE bit to the interrupt signal (o_INT). After processing the command list starting with Priority3, this module sets the TRIG bit of the Status register to “1”. When the TRIGE bit of the Status enable register is effective, this module outputs the state of the TRIG bit to the interrupt signal (o_INT). Please refer to 7.3.4 Priority.

△ When status bit of register* is effective, CMDSEQ processes the command list.
*It sets by TRIGGER start enable register or All Soft reset enable register.

	Factor of TRIGGER start			
	WDT (Priority1)	SIG (Priority1)	VDC (Priority2)	TRIGGER 0-7 (Priority3)
The command list after All soft reset is released is executed.	△	△		
Command list execution after trigger is generated.	△	△	△	△

Note : When “SIG of Priority1” and “SIG of Priority3” are effective, “SIG of Priority3” is processed, after “SIG of Priority1” is processed.

Figure 7.3-3 Start condition of trigger start

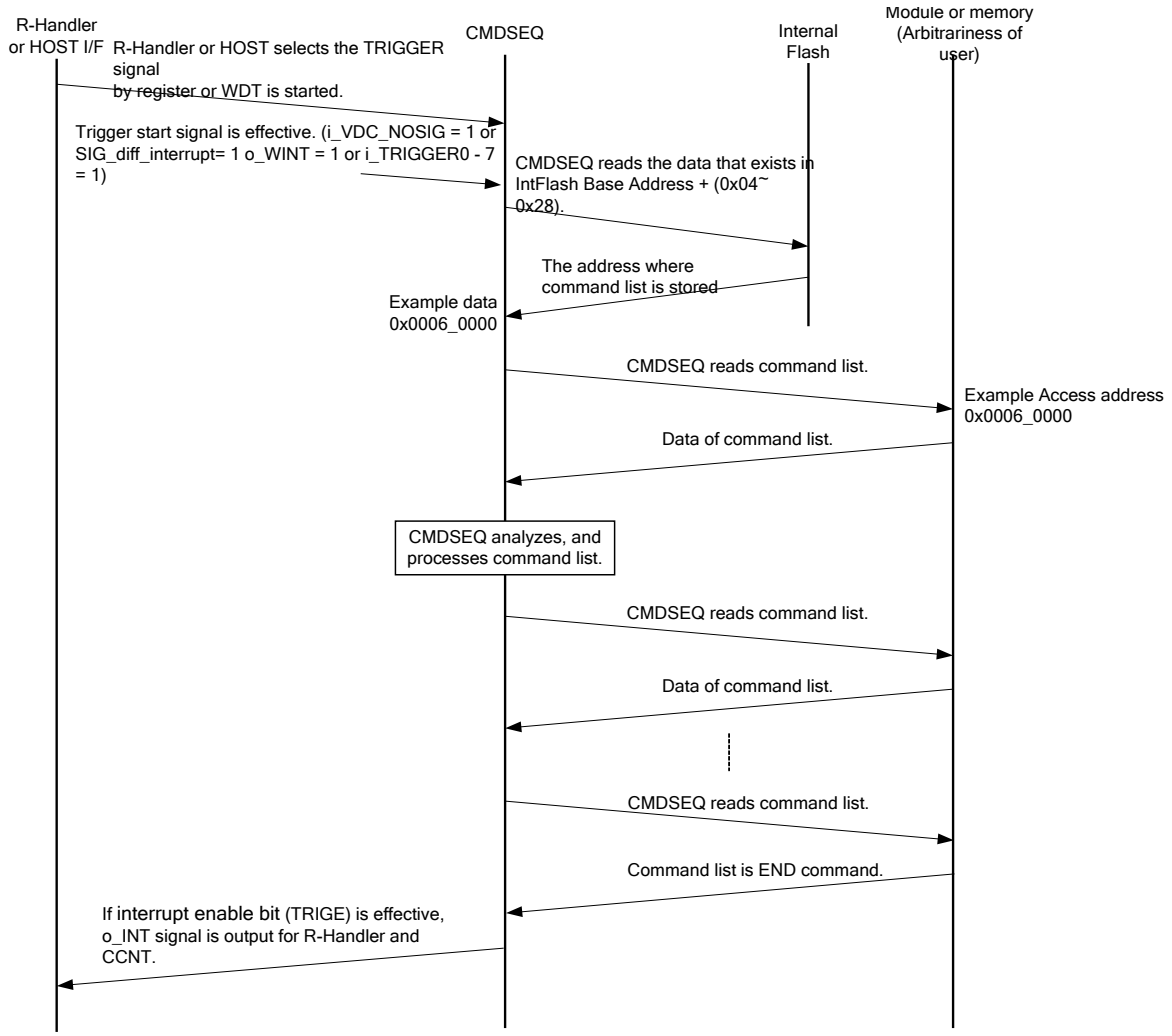


Figure 7.3-4 Trigger start

The trigger signal from this function and access to the address in the Embedded Flash Memory by the trigger signal are described in the following table.

Priority	Interrupt trigger signal	Embedded Flash Memory Access address	Outline of interrupt signal
Priority 1	o_WINT SIG_Diff_interrupt	Base Address+0x04	WDT interrupt signal and SIG interrupt signal are used to Trigger start
Priority 2	i_SYNC_ERR	Base Address + 0x08	"Underflow of Pixel FIFO" or "VSYNC error" from VDC and used to Trigger start.
Priority 3	i_TRIGGER0	Base Address+0x0C	The Signal used to Trigger start.
	i_TRIGGER1	Base Address+0x10	
	i_TRIGGER2	Base Address+0x14	
	i_TRIGGER3	Base Address+0x18	
	i_TRIGGER4	Base Address+0x1C	
	i_TRIGGER5	Base Address+0x20	
	i_TRIGGER6	Base Address+0x24	
	i_TRIGGER7	Base Address+0x28	

Note : The TRIGGER signal uses the following in MB88F333.

- i_TRIGGER0 = SIG_Diff_interrupt (SIG.error_count_reached)
- i_TRIGGER1 = o_INT_HSYNC (HSYNC interrupt from VDC)
- i_TRIGGER2 = o_INT_VSYNC (VSYNC interrupt from VDC)
- i_TRIGGER3 = Line_int (Sprite GDC module specified line processing over)
- i_TRIGGER4 = Line_blank_int (Sprite GDC module line blank)
- i_TRIGGER5 = int_o (RLD interrupt from RL Decoder)
- i_TRIGGER6 = o_DMA_DIRQ0 (HDMAC interrupt from HDMAC0)
- i_TRIGGER7 = o_DMA_DIRQ1 (HDMAC interrupt from HDMAC1)

7.3.2.3. Register start

This function triggers by setting a register and executes a command list.
This module executes the command list at set address by configuring the REGISTER start address register. Please set the Start register to “1” when you execute this function. When starting by this function, it is not possible to start this function again until the command list processing has finished (refer to Figure 7.3-5 Register start).
When processing has completed, this module sets the REG bit of the Status register to “1”. When the REGE bit of the Status enable register is effective, this module outputs the state of the REG bit to the interrupt signal (o_INT). Please refer to 7.3.4 Priority.

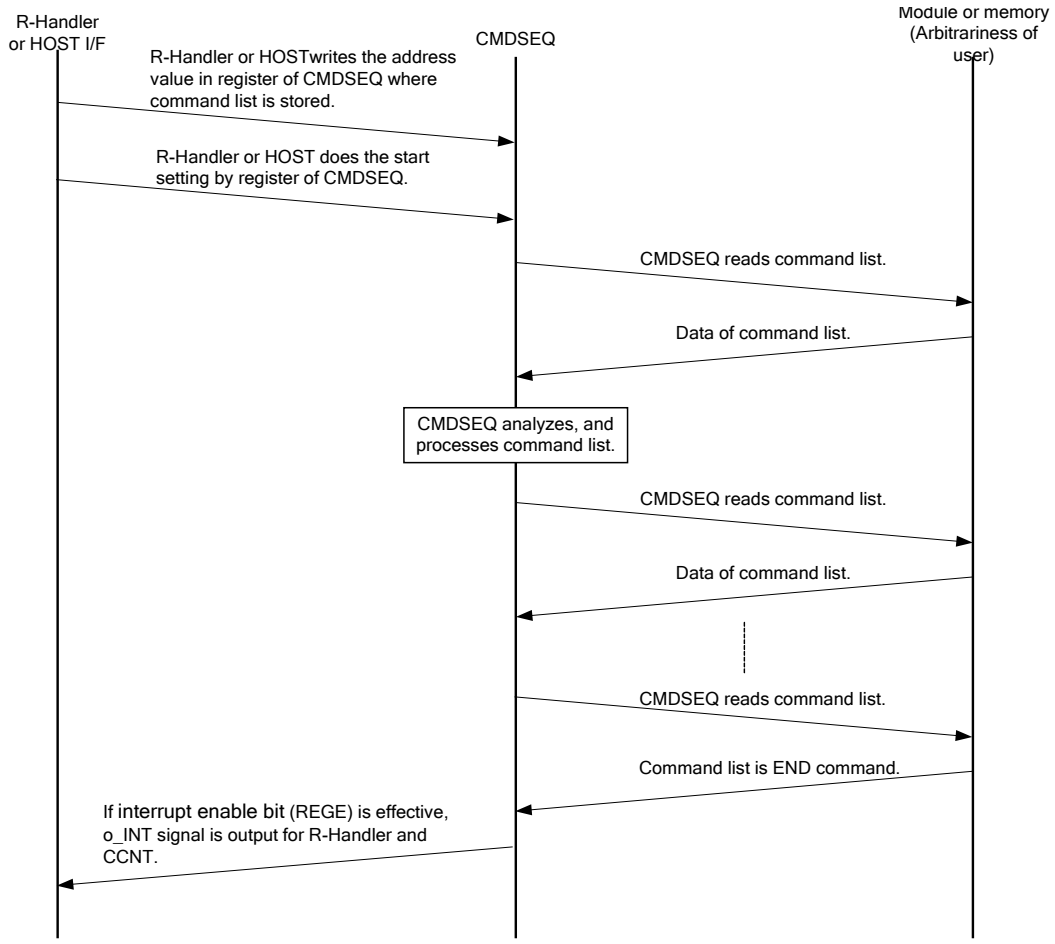


Figure 7.3-5 Register start

7.3.3. Various start end conditions

Conditions under which this module ends operation is described in the following (Start by Reset start, Trigger start or Register start).

- The END command was detected while processing the command list.
- There was an error reply while accessing the Slave module.
- The forced termination function became effective.
- There was a mismatch of the expected value while executing the COMPREG command.

7.3.4. Priority

Priority levels exist for Reset start, Trigger start and Register start. The priority is as follows when they are simultaneously started.
Trigger start (By Priority1 signal) > Reset start > Trigger start (By Priority2 signal) > Trigger start (By Priority 3 signal) > Register start

When various errors (forced termination, the error reply from the slave module and the expected value error) occurred operating, all the activate requests are canceled.

The following processing table for various start requests during operation

☆ Processing is interrupted and the demand is accepted.
△ After operated, the demand is accepted.
□ The demand is disregarded.
▽ The demand doesn't happen

Operation demand \ It's operating	RESET START	TRIGGER START(Pri1)	TRIGGER START(Pri2)	TRIGGER START(Pri3)	REGISTER START
RESET START	▽	▽	▽	▽	▽
TRIGGER START(Pri1)	☆	□	☆	☆	☆
TRIGGER START(Pri2)	△	△	□	☆	☆
TRIGGER START(Pri3)	△	△	△	□	☆
REGISTER START	△	△	△	△	▽

Following Figure 7.3-6 State machine in each start, higher priority Triggers cancel operation of Lower priority command list execution.

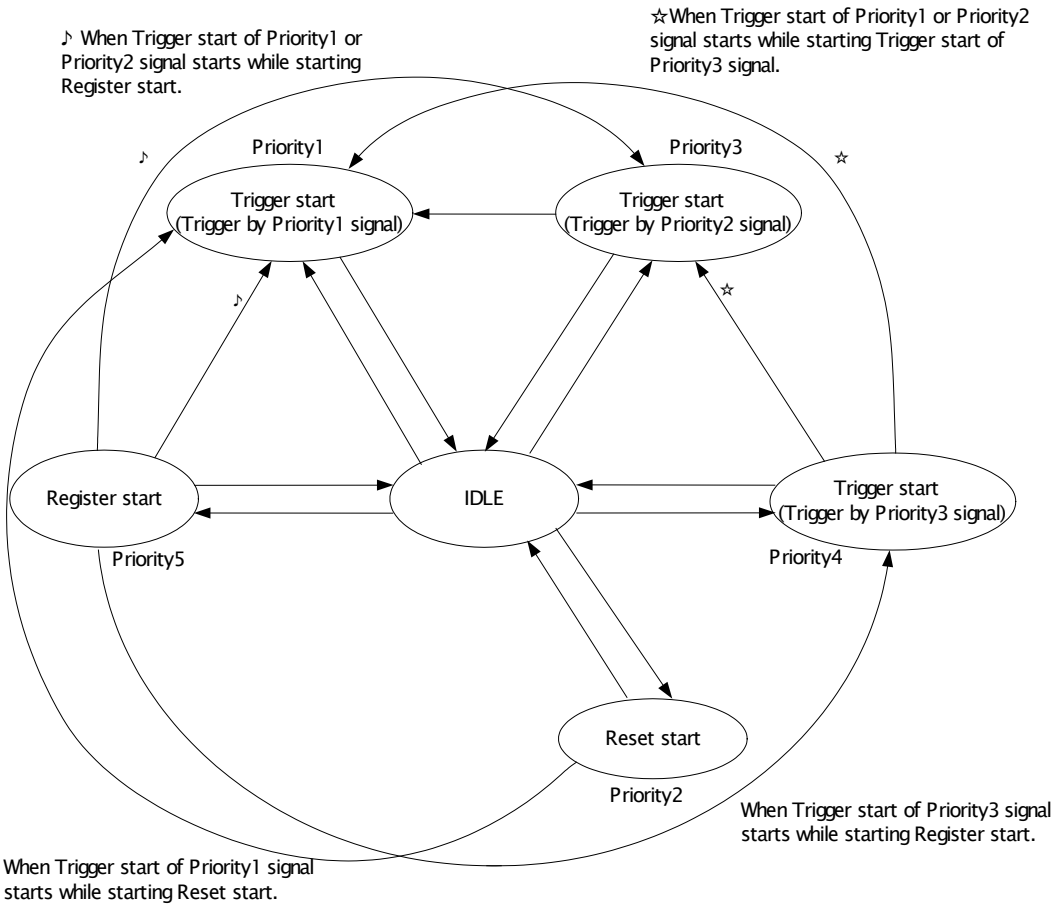


Figure 7.3-6 State machine in each start

7.3.5. Command list

A set of different parameter settings and pattern data is called a command list. It is possible to use a command list to achieve a wait operation for various event signals and trigger signals.
A command list packet consists of a header, which describes the content of the packet and the packet itself.

Header part
Parameter parts

The following table shows the packet code lists. The CMDSEQ module identifies the type of command list using the packet code included in the header section.

Packet Code	Packet Name	Comment
0000_0001(bin)	WAIT Trigger	WAIT by trigger
0000_0010(bin)	SETREG	Set Register
0000_0011(bin)	OSETREG	Offset Address Set Register
0000_0100(bin)	COMPREG	Comparison Register
FFFF_FFFF (hex)	END	Command End

7.3.5.1. WAIT Trigger

A WAIT trigger is used to halt CMDSEQ processing for a set number of counts. The signal that acts as a WAIT trigger and the number of counts to be waited for are specified in the header section.
Please refer to Figure 7.3-7 for the process flow of a WAIT trigger used by this hardware module.

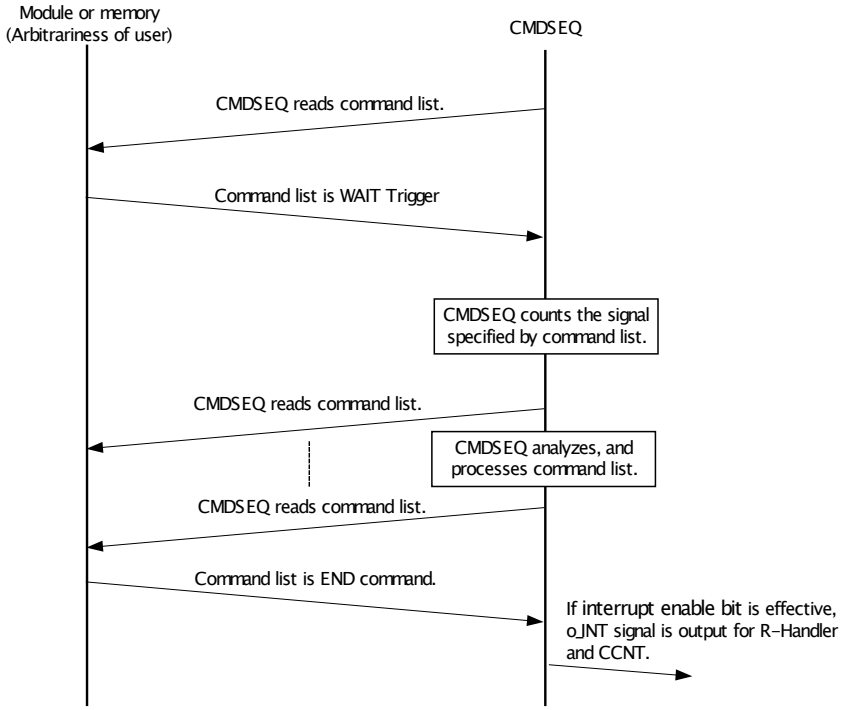


Figure 7.3-7 Transaction flow of WAIT Trigger

A format and a set value of command list of WAIT Trigger are as follows.

31 – 24	23 – 16	15 – 4	3 – 0
WAIT=0x01	Count	Reserved	Trigger

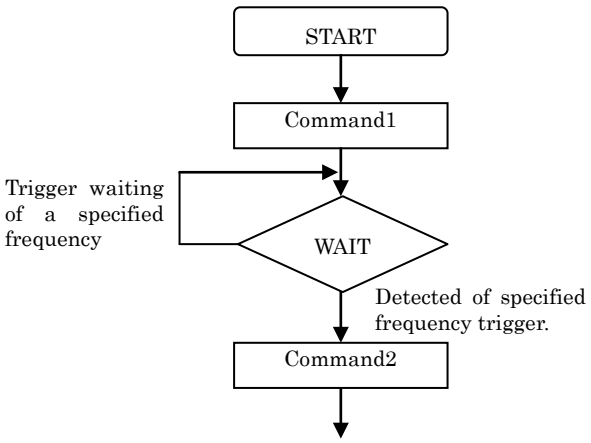
Trigger [3:0] Selection of synchronizing display trigger signal
- 0000: i_WAIT0
- 0001: i_WAIT1
- 0010: i_WAIT2
- ⋮
- 1001: i_WAIT9

Count [23:16] Trigger signal count value.
- 0: CMDSEQ waits until 256 times WAIT signal becomes effective.
- 1: CMDSEQ waits until 1 time WAIT signal becomes effective.
- 2 CMDSEQ waits until 2 times WAIT signal becomes effective.
- ⋮
- FF: CMDSEQ waits until 255 times WAIT signal becomes effective.

Note : The WAIT Trigger uses the following in MB88F333. Please do not set the values other than Trigger = 0x0 – 0x9. Nothing is connected with WAIT10 – WAIT15, so please do not set Trigger = 0xA – 0xF.

- i_WAIT0 = HCLK (AHB bus clock)
- i_WAIT1 = SIG.error_count_reached
- i_WAIT2 = o_INT_HSYNC (HSYNC interrupt from VDC)
- i_WAIT3 = o_INT_VSYNC (VSYNC interrupt from VDC)
- i_WAIT4 = Line_int (Sprite GDC module specified line processing over)
- i_WAIT5 = Line_blank_int (Sprite GDC module line blank)
- i_WAIT6 = int_o (RLD interrupt from RL Decoder)
- i_WAIT7 = o_DMA_DIRQ0 (Interrupt of HDMAC0)
- i_WAIT8 = o_DMA_DIRQ1 (Interrupt of HDMAC1)
- i_WAIT9 = o_INT_MEMIF (Interrupt of Built-in Flash memory)

Following example of flow chart by WAIT Trigger.



7.3.5.2. SETREG

SETREG is used to write data to a register and to memory allocated in the memory map. Data can be continuously written in blocks using a count value. In this mode, the address increases by 04hTransmission size is only word access and 2 bits of low ranks of an address do not see. By controlling Inside BUFFER, it transmits automatically according to the number of transmission in the burst size (a maximum of 16 bursts) of SINGLE, INCR4, INCR8 and INCR16, and unfixed length (INCR).

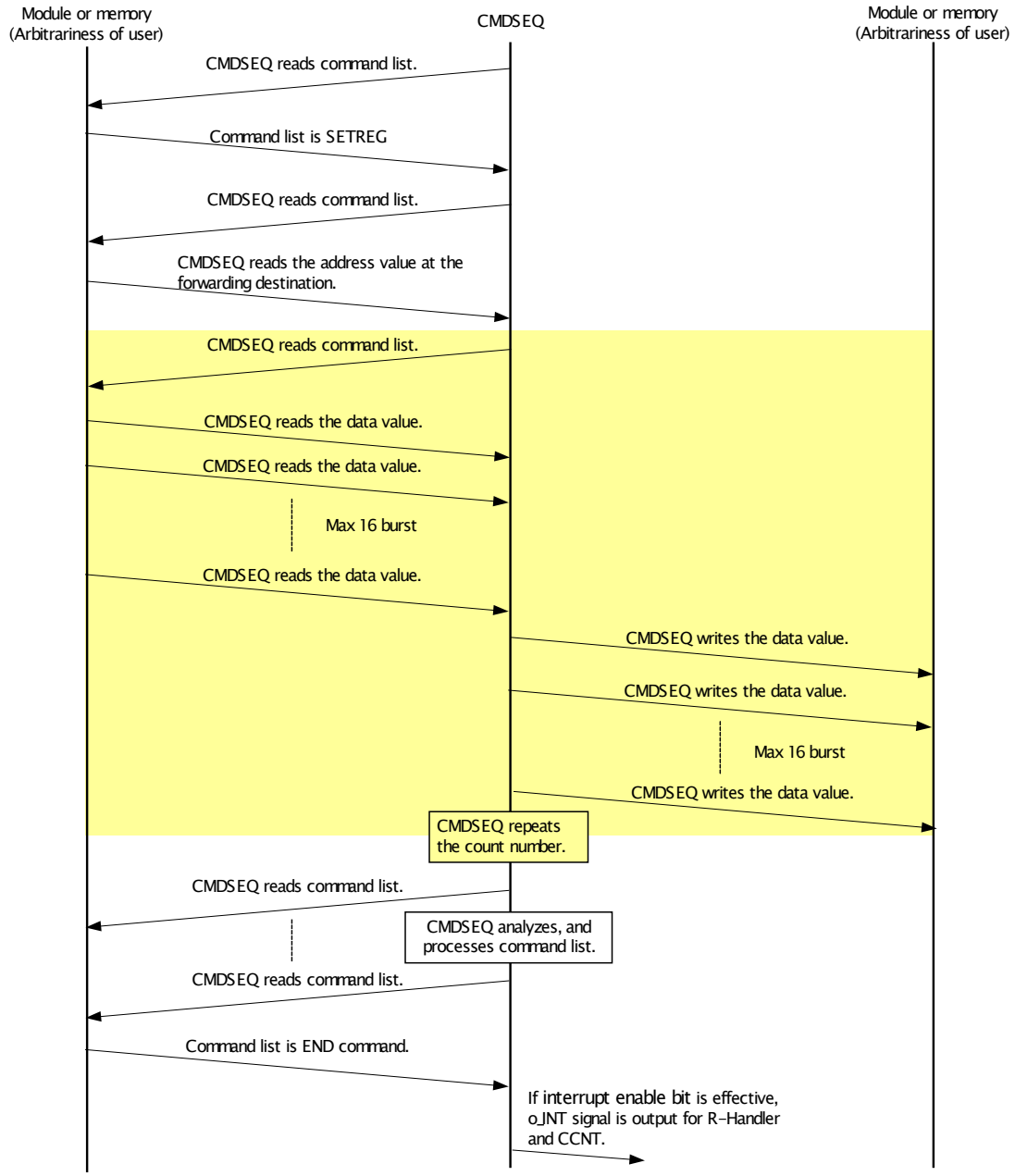


Figure 7.3-8 Transaction flow of SETREG

A format and a set value of command list of SETREG are as follows.

31 24	23 16	15 2	1	0
SETREG=0x02	Count	Reserved		
Address				Reserved
(Data0)				
(Data1)				
…				
(Data255)				

Count [23:16]	The number set up to a register or a memory 0: Data is set 256 times. 1: Data is set 1 times. 2: Data is set 2 times. ⋮ FF: Data is set 255 times.
Address	The first address of register or memory that begins data set
Data	Set data in register or memory. It is possible to set it by 256 data or less.

It is an example of formatting SETREG command as follows.

SETREG Count = 5	Address where data is set	Set data
Address = 0x1000_0000	Address	Data
Data0 = 0x1	0x1000_0000	0x0000_0001
Data1 = 0x2	0x1000_0004	0x0000_0002
Data2 = 0x3	0x1000_0008	0x0000_0003
Data3 = 0x4	0x1000_000c	0x0000_0004
Data4 = 0x5	0x1000_0010	0x0000_0005

7.3.5.3. OSETREG

OSETREG writes data to a register and memory allocated in the memory map. The transfer destination address is: Base Address + Offset Address. Data can be written to an arbitrary address. Data [31:24] bits are written with a 0 and the 2 subordinate position bits of the address is visible for a 24-bit transfer. Also, a single subordinate position bit of an address is not seen during a 16-bit transfer (please refer to the format of the CSETREG command list).
The address used to write data is Base Address + Offset Address. Please ensure that the Base Address and Offset Address are transmitted together and in this order when creating a command list.
Burst sizes of SINGLE, INCR4, INCR8, INCR16 and of an undefined length (INCR) (16 bursts or less) can be automatically used in accordance with the number of forwarding being forwarded for the data READ. In the case of a write, it becomes correspondence of only SINGLE transaction.
Please refer to Figure 7.3-9 for the processing flow of OSETREG by the CMDSEQ module.

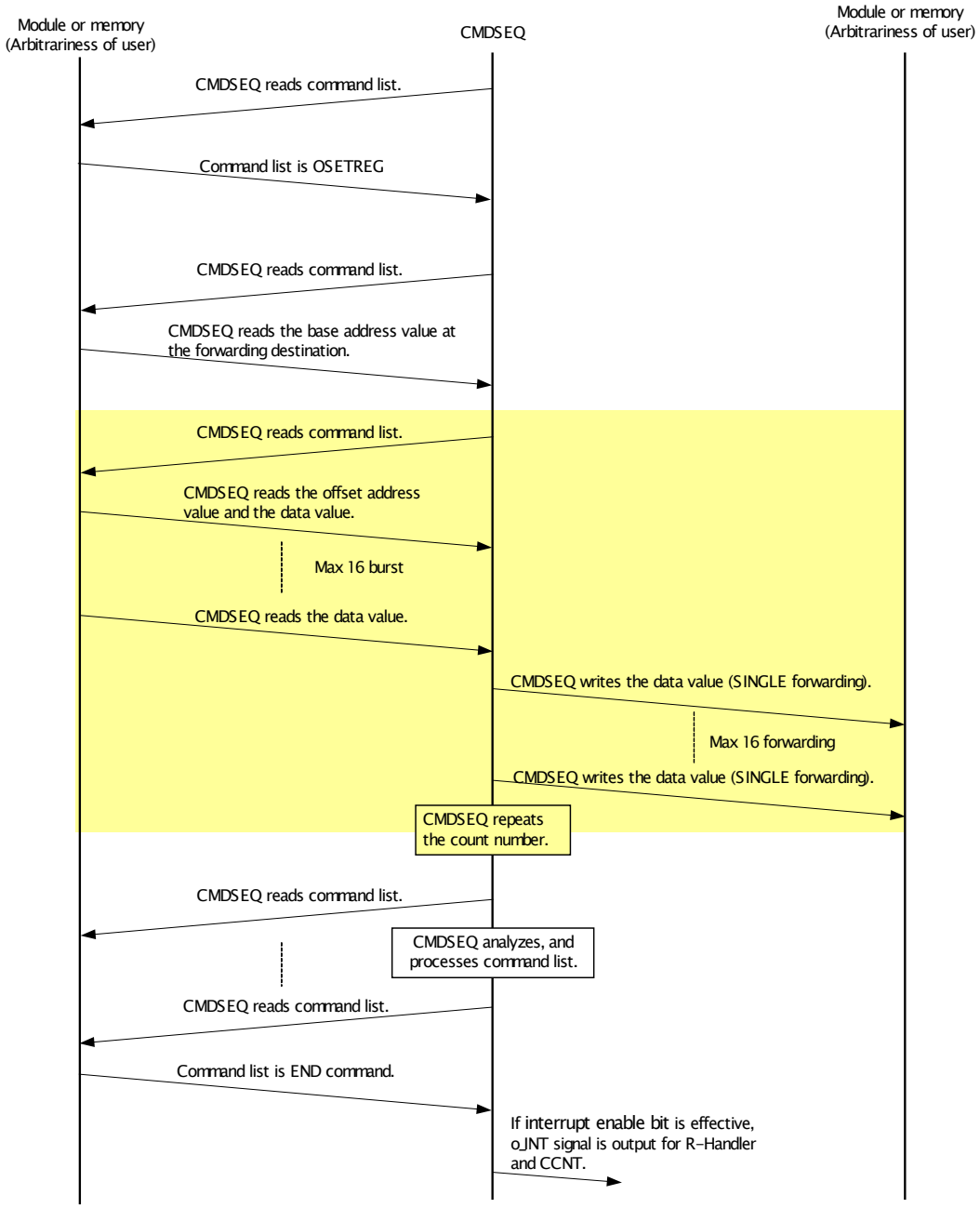


Figure 7.3-9 Transaction flow of OSETREG

A format and a set value of command list of OSETREG are as follows.

Transfer format of 8 bits

31 24	23 16	15 8	7 2	1 0
OSETREG=0x03	Count [23:16]	Reserved		Size=00
Base Address [31:0]				
Offset Address0			(Data0)	
Offset Address1			(Data1)	
:				
Offset Address255			(Data255)	

Note: The transmission place address of 8-bit transfer is as follows.

Base Address[31:0]+ Offset Address[31:8]

Transfer format of 16 bits

31 24	23 17	16	15 8	7 2	1 0
OSETREG=0x03	Count [23:16]		Reserved		Size=01
Base Address [31:1]					Reserved
Offset Address0		Reserved	(Data0)		
Offset Address1		Reserved	(Data1)		
:					
Offset Address255		Reserved	(Data255)		

Note: The transmission place address of 16-bit transfer is as follows.

(Base Address [31:1], 1’b0) + (Offset Address [31:17], 1’b0)

Transfer format of 24 bits

31 26	25 24	23 16	15 8	7 2	1 0
OSETREG=0x03		Count [23:16]	Reserved		Size=10
Base Address [31:2]					Reserved
Offset Address0	Reserved	(Data0)			
Offset Address1	Reserved	(Data1)			
:					
Offset Address255	Reserved	(Data255)			

Note: The transmission place address of 24-bit transfer is as follows.

(Base Address [31:2], 2’b00) + (Offset Address [31:26], 2’b00)

Note: (A,B) expresses bit connection.

Example: (Base Address [31:1], 1’b0) Base Address [31:1] = A 1’b0 = B

(Base Address [31:1], 1’b0) = (A,B)

Size [1:0]	Select transfer size. 00: 8 bit transfer (Only byte transfer is supported. 01: 16 bit transfer (Only Half Word transfer is supported. 10: 24 bit transfer (Only Word transfer is supported. 11: 24 bit transfer (Only Word transfer is supported.
Count [23:16]	The number set up to a register or a memory 0: Data is set 256 times. 1: Data is set 1 times. 2: Data is set 2 times. ⋮ FF: Data is set 255 times.
Base Address	Base address of register or memory that sets data.
Offset Address [7:0]	Offset address of register or memory that sets data.
Data	Set data in register or memory. . Whenever 24 bit transfer, 0 is set to 31:24 bits of data. Data up to 256 can be set.

A format and a set value of command list of OSETREG are as follows.

OSETREG Count = 7 Size = 0		Address where data is set	Set data
Base Address = 0x1000_0000		Address	Data
Offset Address0 = 0x000001	Data0 = 0x1	0x1000_0001	0x01
Offset Address1 = 0x000003	Data1 = 0x2	0x1000_0003	0x02
Offset Address2 = 0x020000	Data2 = 0x3	0x1002_0002	0x03
Offset Address3 = 0x240001	Data3 = 0x4	0x1024_0001	0x04
Offset Address4 = 0x002500	Data4 = 0x5	0x1000_2500	0x05
Offset Address5 = 0x00c000	Data5 = 0x6	0x10c0_0000	0x06
Offset Address6 = 0xc40000	Data6 = 0x7	0x1040_0000	0x07

7.3.5.4. COMPREG

The register allocated in the memory map is read and expected values are compared with the values transferred via the command list. Data can be continuously transferred according to the count value. In this case, the address increases by 04h. Transfer size is only word access. 2 bits of low ranks of an address do not see. By controlling Inside BUFFER, it transmits automatically according to the number of transmission in the burst size (a maximum of 16 bursts) of SINGLE, INCR4, INCR8 and INCR16, and unfixed length (INCR).

If the comparison value is not what is expected (error), this module sets the CMPE bit of the Status register to “1” and the CMDSEQ module terminates processing. In addition, the error address is stored in the COMPREG error address hold register.

If the CMPEE bit of the Status enable register becomes effective, the CMDSEQ module outputs the state of the CMPE bit to the interrupt signal (o_INT) and the address where the error occurred is held in the COMPREG error address hold register. The address held is the address, where the error first occurred.

Please refer to Figure 7.3-10 for the process flow of COMPREG by this module.

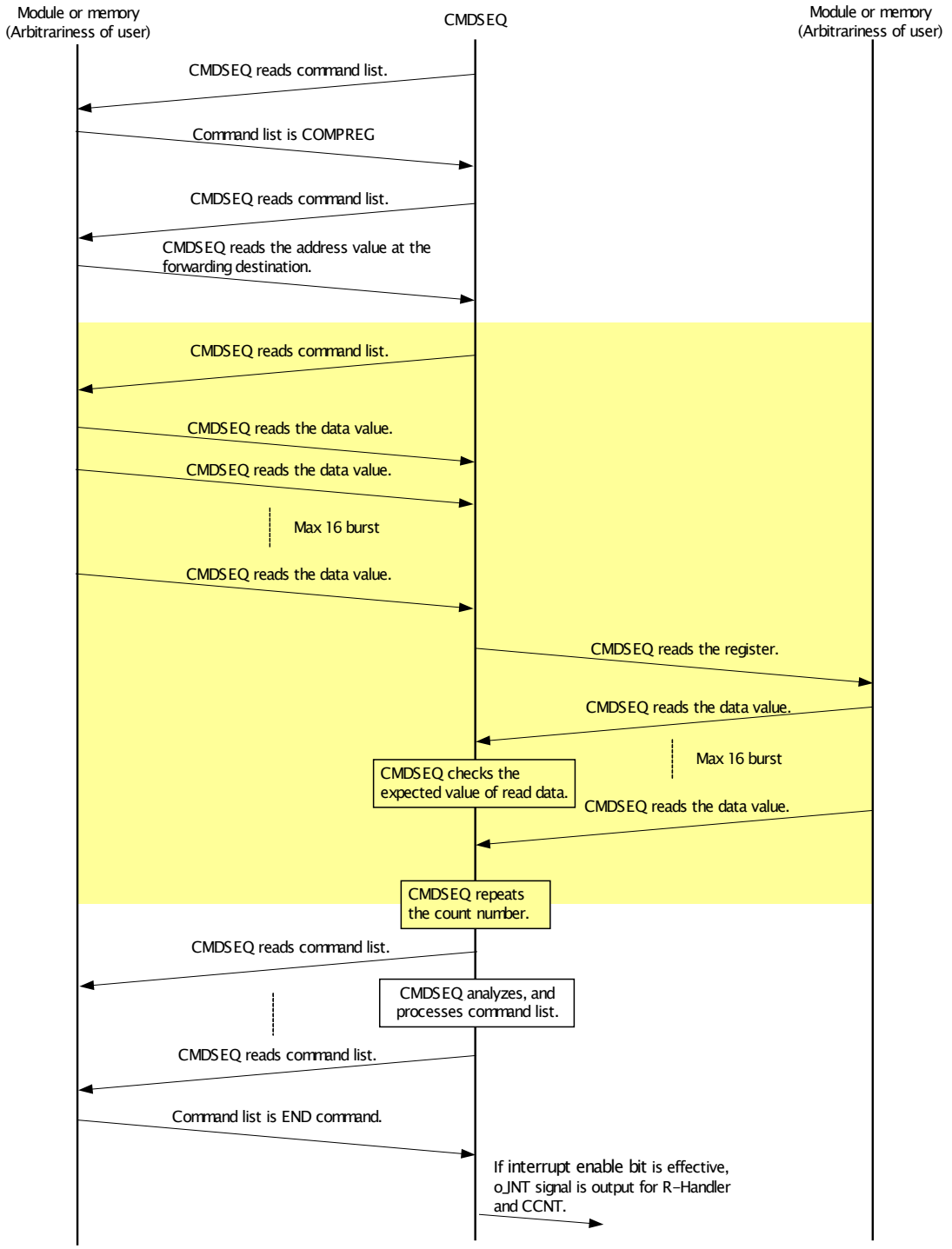


Figure 7.3-10 Transaction flow of COMPREG

A format and a set value of command list of COMPREG are as follows.

31 24	23 16	15 2	1	0
COMPREG=0x04	Count	Reserved		
Address				Reserved
(Data0)				
(Data1)				
…				
(Data255)				

Count [23:16]	The data values are compared. 0: Data is comparison 256 times. 1: Data is comparison 1 times. 2: Data is comparison 2 times. ⋮ FF: Data is comparison 255 times.
Address	The first address of register or memory that begins expected value comparison.
Data	Expected value data. It is possible to set it by 256 data or less.

A format and a set value of command list of COMPREG are as follows.

COMPREG Count = 5	Address where data is set	Set data
Address = 0x1000_0000	Address	Data
Data0 = 0x1	0x1000_0000	0x0000_0001
Data1 = 0x2	0x1000_0004	0x0000_0002
Data2 = 0x3	0x1000_0008	0x0000_0003
Data3 = 0x4	0x1000_000c	0x0000_0004
Data4 = 0x5	0x1000_0010	0x0000_0005

7.3.5.5. END

The command list processing is ended.
Please refer to Figure 7.3-11 for the processing flow of END by this module.

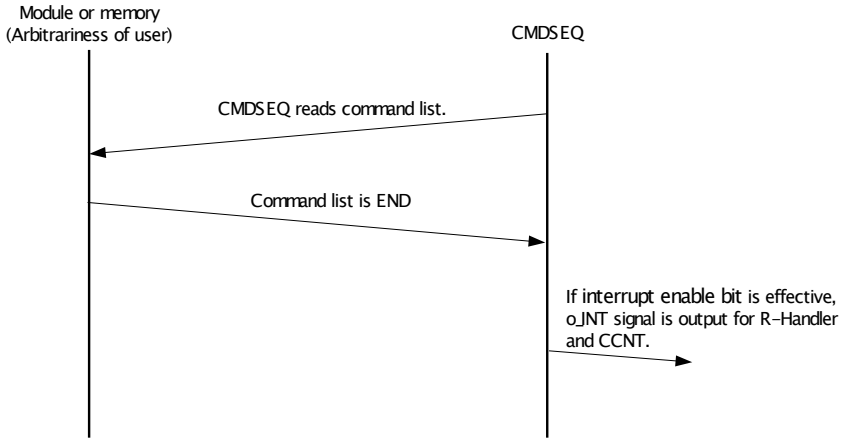
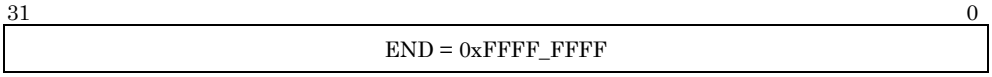
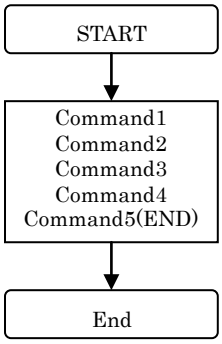


Figure 7.3-11 Transaction flow of END

A format and a set value of command list of END are as follows.



Following example of flow chart by END.



7.3.6. Slave module access error response

This module sets the SERR bit of the Status enable register to “1” if an error reply from the slave module is issued. The error address is stored in the Slave access error address hold register.
In addition, if the SERRE bit of the Status enable register is effective, this module outputs the state of the SERR bit to the interrupt signal (o_INT).
This module terminates processing if an error reply from the slave module is issued.

7.3.7. Forced termination

This module can execute a forced termination of command list processing. When this function becomes effective, this module discontinues transfer activities after executing the command list being processed. The FCEND bit of the forced termination register is set to “1” and the forced termination function is then effective.
When processing is completed, this module sets the FCTERM bit of the Status register to “1”. When the FTERME bit of the Status enable register is effective, this module outputs the state of the FTERM bit to the interrupt signal (o_INT).

7.3.8. Transfer status

The following can be confirmed by Transfer status register1-3.

- Start method that this module is executing now (Reset start, Trigger start (by o_WINT or i_SYNC_ERR or i_TRIGGER0-7), register start).
- Header section of command list.
- The first address or offset address set next to the header section of the command list.

7.3.9. All Soft Reset

By setting the AWDT bit of the All Soft Reset enable register to “1”, this module performs an 'All Soft Reset' in the MB88F333 device when the watchdog timer (WDT) issues a time-out. This module processes a command list after the All Soft Reset has been released. In addition, an All Soft Reset is performed in the MB88F333 device with the interrupt signal (SIG_Diff_interrupt) from the SIG unit by setting the ASIG bit of the All Soft Reset enable register to “1”. This module processes a command list after the All Soft Reset has been released. This module sets "1" as the TWDTS bit of Status register, after processing a command list by this starting.
Refer to the clause of Chip Controller Module for the details of All Soft Reset.

7.3.10. Embedded flash access address

The access address of the embedded (i.e. internal) flash used by the CMDSEQ module is described below. Please write an address value that is definitely followed by a command list (in the subsequent addresses). Access by this module to the embedded flash is done using Words only and the subordinate 2 bits of the access address are disregarded. If the data value written to the subsequent address is 0xFFFF_FFFF, this module ends processing (Exclude Reset start).

Embedded Flash Memory access address	Start factor
Base Address＋0x00	Reset start
Base Address＋0x04	Trigger start (For o_WDT or SIG_Diff_interrupt. Priority1 signal).
Base Address＋0x08	Trigger start (For i_SYNC_ERR Priority 2 signal).
Base Address＋0x0C	Trigger start (For priority3 signal).
Base Address＋0x10	
Base Address＋0x14	
Base Address＋0x18	
Base Address＋0x1C	
Base Address＋0x20	
Base Address＋0x24	
Base Address＋0x28	

7.3.11. Watch Dog Timer (WDT)

The watchdog timer is a unit that is used to monitor whether the system is operating normally or not. The watchdog timer function is started if the WCLEA bit of the Watchdog Timer clear register is set to “1”. After the WDT has been started, the CMDSEQ unit counts downwards using the WCOUNT bitfield of the watchdog timer count register. The WDT bit of the Status register is also set to “1”. If the WCOUNT bitfield reaches “0” and the WDTE bit of the Status enable register is enabled, this module outputs an interrupt signal (o_WINT) of the information on WDT bit. If you write a “1” to the WCLEA bit before the WCOUNT bitfield of Watch Dog Timer disable register reaches “0”, the WCOUNT bitfield value is reset.
The watchdog timer function is disabled if the WDIS bit of the watchdog timer disable register is set and effective.
The WDT timer can be set in a range from 0.01s to 3.2s.

7.3.12. Interrupt

This module has 2-interrupt signal of o_INT and o_WINT.

7.3.12.1. CMDSEQ Interrupt Factor

The interrupt factor (source) of o_INT outputs the logical or of the following: Reset start, Trigger start, Register start, forced termination, error reply from slave module and an expected value error. In the event of a forced termination or an error reply from the slave module or an expected value error, each status bit for these factors (forced termination, the error reply from the slave module and the expected value error) changes to “1” even if they are generated at the same time.
Interrupts concerned are shown below:

☆ Status bit become effective.
□ Status bit doesn't become effective.
▽ Doesn't happen

		Various starts				
		RESET START	TRIGGER START (By Pri1)	TRIGGER START (By Pri2)	TRIGGER START (By Pri3)	REGISTER START
Execute	END command	☆	☆	☆	☆	☆
	Forced termination	☆	☆	☆	☆	☆
	Slave module access error	☆	☆	☆	☆	☆
	COMPREG Error	☆	☆	☆	☆	☆
	INTFlash data all"F"	☆	□	□	□	▽

7.3.12.2. CMDSEQ Interrupt Clear

o_INT that is interrupt signal can be “0” by setting “1“ to each bit (RESET, TWDTS, SYNCE, TRIG, REG, FTERM, SERR, CMPE) of Status register.

7.3.12.3. Watch Dog Timer Interrupt Factor

Interrupt factor of o_WINT is as follows for Status register.

- When the watchdog timer count register reaches “0”.

7.3.12.4. Watch Dog Timer Interrupt Clear

The watchdog timer o_WINT interrupt signal can be cleared to “0” by setting the WDT bit of the Status register to ”0”. If the WDTE bit of the Status enable register is invalid, then only the WDT bit will be cleared.

7.4. Registers

7.4.1. Format of Register Descriptions

- Endian

  Only Little Endian access corresponds to the register of this module.

- Address

  “Address” shows the address of the register. (Base address + Offset address)

- Bit

  “Bit” shows the bit number of the register.

- Name

  “Name” shows the bit field name of the register.

- R/W

  “R/W” shows the attribute of Read/Write in each Bit field.
  R0: The reading value is always "0".
  R1: The reading value is always "1".
  W0: The writing value is always "0".When “1” is written, it is disregarded.
  W1: The writing value is always "1". When “0” is written, it is disregarded.
  R: Read
  W: Write

  Note : If a value is written to registers/bitfields that list R0, R1 and R in the following descriptions, then this value will not be changed in those registers/bitfields.

- Initial value

  “Initial value” is an initial value when reset is released.
  0: It becomes "0".
  1: It becomes "1".
  X: It becomes irregular.

7.4.2. Global Address

For the module base address please refer to the chapter 3 Memory Map.

7.4.3. Register summary

Address	Register Name	Description
Base+0x0000	Status	CMDSEQ status register
Base+0x0004	StatusEnable	Enable/Disable of CMDSEQ interrupt signal
Base+0x0008	Start	Start of CMDSEQ
Base+0x000C	IDLEStatus	Status confirmation of IDLE.
Base+0x0010	REGISTERStartAdd	Addressing of register start setting
Base+0x0014	ForcedTermination	Forced termination
Base+0x0018	TRIGGERStartEnable	Enable/Disable TRIGGER start signal
Base+0x001C	TransferStatus1	Execution status of various starts
Base+0x0020	TransferStatus2	Value of header part
Base+0x0024	TransferStatus3	Value of the first address or base address
Base+0x0028	COMPREGErrorAddressHold	The address of comparison error is held.
Base+0x002C	SlaveAccessErrorAddressHold	The address of slave access error address is held
Base+0x0030	WDTClear	WDT clear
Base+0x0034	WDTCount	WDT timer count
Base+0x0038	WDTDisable	WDT disable
Base+0x003C	WDTCountCheck	Check of WDT timer count value
Base+0x0040	AllSoftResetEnable	Processing setting of All Soft reset by WDT or SIG error
Base+0x0044	Reserved	-
:	:	:
Base+0x0FFC	Reserved	-

7.4.4. Register Description

7.4.4.1. Status register

Address	Base Address + 0x0000															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															WDT
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R/W1
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved								CMPE	SERR	FTERM	REG	TRIG	SYNCE	TWDTS	RESET
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register shows status of the CMDSEQ module.
The following items are output factors of the o_INT signal, and their logical or is expressed by the Status register.
This module outputs the CMDSEQ interrupt signal (o_INT). Its logical OR operation is expressed by the Status register (RESET, WDTE, SYNCE, TRIG, REG, FTERM, SERR, CMPE, SRSET).
This module outputs the WDT interrupt signal (o_WINT). Its logical OR is expressed by the Status register (WDT).

Bit0 RESET (RESET start interrupt)
This bit displays "1" if the data value 0xFFFF_FFFF written to the embedded flash Base Address＋0x00 was executed, the command was processed, forced termination was performed, an expected value error occurred, or a slave access error occurred during a reset start.
If the RESETE bit of the Status enable register is set to 1, this bit is output as the interrupt signal (o_INT).
This bit is cleared by writing “1”.
0:
1: End of Reset start

Bit1 TWDTS (interrupt of Trigger start by WDT or SIG)
This bit displays "1" if the command was processed, forced termination was performed, an expected value error occurred, or a slave access error occurred during a Trigger start by the WDT or SIG unit.
If the WDTE bit of the Status enable register is set to 1, this bit is output as the interrupt signal (o_INT).
This bit is cleared by writing “1”.
0:
1: End of Trigger start by WDT/SIG

Bit2 SYNCE (interrupt of trigger start by VDC Sync Error)
This bit displays "1" if the command was processed, forced termination was performed, an expected value error occurred, or a slave access error occurred a during Trigger start by VDC No sync error, it displays "1."
If the SYNCEE bit of the Status enable register is set to 1, this bit is output as the interrupt signal (o_INT).
This bit is cleared by writing “1”.
0:
1: End of Trigger start by VDC sync error

Bit3 TRIG (interrupt of trigger start by various TRIGgers)
This bit displays "1" if the command was processed, forced termination was performed, an expected value error occurred, or a slave access error occurred during the various Trigger starts.
If the TRIGE bit of the Status enable register is set to 1, this bit is output as the interrupt signal (o_INT).
This bit is cleared by writing “1”.
0:
1: End of Trigger start by various Trigger

Bit4 REG (REGister start interrupt)
This bit displays "1" if the command was processed, forced termination was performed, an expected value error occurs, or a slave access error occurred during a Register start.
If the REGE bit of the Status enable register is set to 1, this bit is output as the interrupt signal (o_INT).
This bit is cleared by writing “1”.
0:
1: End of register start

Bit5 FTERM (Forced TERMination interrupt)
When a forced termination is completed, this bit is set to "1".
If the FTERME bit of the Status enable register is set to 1, this bit is output as the interrupt signal (o_INT).
This bit is cleared by writing “1”.
0:
1: Forced termination was executed.

Bit6 SERR (AHB module Slave access ERRor interrupt)

If an AHB slave error reply is executed, this bit is set to "1".
If the SERRE bit of the Status enable register is set to 1, this bit is output as the interrupt signal (o_INT).
This bit is cleared by writing “1”.
0:
1: Slave error reply

Bit7 CMPE (CoMParison Error interrupt)
When the COMPREG command is completed, this bit is set to "1".
If the CMPEE bit of Status enable register is set to 1, this bit is output as the interrupt signal (o_INT).
This bit is cleared by writing “1”.
0:
1: Mismatch of expected value comparison

Bit16 WDT (Watchdog Timer interrupt)
If the WDT timer issues a timeout, this bit is set to "1".
If the WDTE bit of the Status enable register is set to 1, this bit is output as the interrupt signal (o_WINT).
This bit is cleared by writing “1”.
An All Soft Reset does not clear this bit.
0:
1: Timeout

7.4.4.2. StatusEnable register

Address	Base Address + 0x0004															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															WDTE
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved								CMPEE	SERRE	FTERME	REGE	TRIGE	SYNCEE	WDTSE	RESETE
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1

This register controls information on Status register.

Bit0 RESETE(RESET start interrupt Enable)
Writing “1” to this bit validates RESET of Status register interrupt signal (o_INT).
This bit doesn't output the interrupt signal by writing "0".
0: Information on RESET is invalid.
1: Information on RESET is valid.

Bit1 TWDTSE(interrupt Enable of Trigger start by WDT or Sig)
Writing “1” to this bit validates TWDTS of Status register interrupt signal (o_INT).
This bit doesn't output the interrupt signal by writing "0".
0: Information on WDTE is invalid.
1: Information on WDTE is valid.

Bit2 SYNCEE(interrupt Enable of trigger start by vdc SYNC Error)
Writing “1” to this bit validates SYNCE of Status register interrupt signal (o_INT).
This bit doesn't output the interrupt signal by writing "0".
0: Information on SYNCE is invalid
1: Information on SYNCE is valid.

Bit3 TRIGE(interrupt Enable of trigger start by various TRIGger)
Writing “1” to this bit validates TRIG of Status register interrupt signal (o_INT).
This bit doesn't output the interrupt signal by writing "0".
0: Information on TRIG is invalid.
1: Information on TRIGT is valid.

Bit4 REGE(REGister start interrupt Enable)
Writing “1” to this bit validates REG of Status register interrupt signal (o_INT).
This bit doesn't output the interrupt signal by writing "0".
0: Information on REG is invalid.
1: Information on REG is valid.

Bit5 FTERME(Forced TERMination interrupt Enable)
Writing “1” to this bit validates FTERM of Status register interrupt signal (o_INT).
This bit doesn't output the interrupt signal by writing "0".
0: Information on FTERM is invalid.
1: Information on FTERM is valid.

Bit6 SERRE(ahb module Slave access ERRor interrupt Enable)
Writing “1” to this bit validates SERR of Status register interrupt signal (o_INT).
This bit doesn't output the interrupt signal by writing "0".
0: Information on SERR is invalid.
1: Information on SERR is valid.

Bit7 CMPEE(CoMParison Error interrupt Enable)
Writing “1” to this bit validates CMPE of Status register interrupt signal (o_INT).
This bit doesn't output the interrupt signal by writing "0".
0: Information on CMPE is invalid.
1: Information on CMPE is valid.

Bit16 WDTE(Watch Dog Timer interrupt Enable)
Writing “1” to this bit validates WDT of Status register interrupt signal (o_INT).
This bit doesn't output the interrupt signal by writing "0".
0: Information on WDT is invalid.
1: Information on WDT is valid.

7.4.4.3. Start register

Address	Base Address + 0x0008															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved															START
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R/W1
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is to issue a Register start.

Bit0 START (register START)

If "1" is written to this bit, this module start processing a command list by a Register start.

If this bit is set to “1”, it cannot be set or cleared until the register start is completed.

The following conditions cause this bit to automatically become “0”:

While executing the command list by a Register start, the END command was executed.

Forced termination became effective.

Trigger start (By Priority1) became effective or Register start was forcibly terminated.

Please set the REGADR bit of the REGISTER start address register before setting this bit to “1”.

0: Processing by Register start is unexecuted or has ended.

1: Processing by Register start is scheduled for execution or is executing.

7.4.4.4. IDLEStatus register

Address	Base Address + 0x000C															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved															IDLE
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register indicates when this module is in IDLE mode.

Bit0 IDLE (IDLE state)

When the CMDSEQ’s master is in an IDLE state, this bit becomes 0.

This bit is read-only.

0: IDLE state

1: Active state

7.4.4.5. REGISTERStartAdd register

Address	Base Address + 0x0010															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	REGADR															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	REGADR														Reserved	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Sets the address to execute for a Register start.

Bit31-2 REGADR (REGister start ADdRess)

Use this bitfield to specify the address for the Register start beginning point.

This bit cannot be rewritten while the START bit of the Start register is set to "1".

This bit is not cleared after a register start.

7.4.4.6. ForcedTermination register

Address	Base Address + 0x0014															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved															FCTERM
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R/W1
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register executes a forced termination.

Bit0 FCTERM (ForCed TERMination)

If this bit is changed to“1”, the CMDSEQ module discontinues transfer after executing the current transaction.

After the forced termination has been executed, this bit is automatically set to "0".

0:

1: Forced termination

7.4.4.7. TRIGGERStartEnable register

Address	Base Address + 0x0018															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved												PRI3E			
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved													PRI2E	PRI1ES	PRI1EW
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register selects Priority1, Priority2 and Priority3 signals for a Trigger start.

Bit0 PRI1E (PRIority1 Enable)
If this bit is changed to "1", CMDSEQ does a Trigger start if a Priority1 signal (o_WDT) becomes effective. This bit can not be rewritten until processing has ended for a Priority1 initiated trigger start (depends on whether WDT or SIG executed it).
If the AWDT bit of the All Soft Reset enable register is 1 and o_WDT is 1, an All Soft Reset is executed regardless of the state of this bit.
0:
1: Priority1 signal trigger start (o_WDT) is effective.

Bit1 PRI1ES (PRIority1 start Enable by SIG)
If this bit is changed to "1", CMDSEQ does a Trigger start if a Priority1 signal (SIG_Diff_interrupt) becomes effective. This bit can not be rewritten until processing has ended for a Priority1 initiated trigger start (depends on whether WDT or SIG executed it)
If the ASIG bit of the All Soft Reset enable register is 1 and SIG_Diff_interrupt is 1, an All Soft Reset is executed regardless of the state of this bit.
0:
1: Priority1 signal trigger start (SIG_Diff_interrupt) is effective.

Bit2 PRI2E (PRIority2 Enable)
If this bit is changed to "1", CMDSEQ does a Trigger start if a Priority2 signal (i_SYNC_ERR) becomes effective. This bit is cleared by writing 0.
0:
1: Priority2 signal Trigger start (i_SYNC_ERR) is effective

Bit19-16 PRI3E(PRIority3 Enable)
This bit selects a Priority3 signal from Trigger0 - Trigger7.
Please do not set a value of 0x9 or more, correct operation of the module can not be guaranteed if this is done.
This bit is cleared by writing 0
0: Trigger0 – Trigger7 signal are invalid.
1: Trigger0 is set to Priority3 signal.
2: Trigger1 is set to Priority3 signal.
3: Trigger2 is set to Priority3 signal.
4: Trigger3 is set to Priority3 signal.
5: Trigger4 is set to Priority3 signal.
6: Trigger5 is set to Priority3 signal.
7: Trigger6 is set to Priority3 signal.
8: Trigger7 is set to Priority3 signal.

7.4.4.8. TransferStatus1 register

Address	Base Address + 0x001C															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
R/W	R	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved											REGS	TRIGS	SYNCES	WDTS	RESETS
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R	R	R
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is shown which start of various starts is executing it.

Bit0 RESETS(RESET Status)
This bit can confirm the Reset start.
As for this bit, only the lead is possible.
0:
1: Reset start is starting.

Bit1 WDTS(Status 0f WDT by trigger start)
This bit can confirm the Trigger start by WDT.
As for this bit, only the read is possible.
0:
1: Trigger start is starting by WDT.

Bit2 SYNCES(Status of vdc SYNC Error by trigger start)
This bit can confirm the Trigger start by VDC sync error.
As for this bit, only the read is possible.
0:
1: Trigger start is starting by VDC sync error.

Bit3 TRIGS(TRIgger Status)
This bit can confirm the Trigger start by various Triggers.
As for this bit, only the read is possible.
0:
1: Trigger start is starting by various Triggers.

Bit4 REGS(Register Status)
This bit can confirm the Register start.
As for this bit, only the read is possible.
0:
1: Register start is starting.

7.4.4.9. TransferStatus2 register

Address	Base Address + 0x0020															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CMDLIST															
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CMDLIST															
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is used to confirm that the header part of a command list is being executed.

Bit31-0 CMDLIST (CoMmanD list header part value)
These bits confirm that the header part of a command list is being executed.
This bit is read-only.

7.4.4.10. TransferStatus3 register

Address	Base Address + 0x0024															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CMDADDR															
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CMDADDR															
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is used to confirm that the header address or base address next to the header part is being executed.

Bit31-0 CMDADDR (CoMmanD list ADDRess)
This bit monitors the status of the value of a set address of the command list being executed now.
This bit is read-only.

7.4.4.11. COMPREGErrorAddressHold register

Address	Base Address + 0x0028															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CMPEADDR															
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CMPEADDR															
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register holds the first address of an expected value error determined by the COMPREG command.

Bit31-0 CMPEADDR (CoMParison Error ADDRess)
These bits holds first address of the expected value error determined by the COMPREG command.
The error address held is that of the *first* error only.
This bit is read-only.

7.4.4.12. SlaveAccessErrorAddressHold register

Address	Base Address + 0x002C															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SEADDR															
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SEADDR															
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register holds the first address of an error reply from a slave module.

Bit31-0 SEADDR (Slave access Error ADDRess)
This bit holds the first address of an error reply from a slave module.
The error address held is that of the *first* error only.
This bit is read-only.

7.4.4.13. WDTClear register

Address	**Base Address** + 0x0030															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved															WCLEA
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R/W1
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register clears the watchdog timer (WDT)

Bit0 WCLEA (Watchdog timer CLEAr)
Writing a “1” to this bit starts or restarts the WDT (clear).
If this bit set to “1” again, the WDT counter is set back to the WDT count value.
While the WDT counter is at least “1”, this bit stays set.
When the WDT counter becomes “0”, this bit is cleared.
When the WDIS bit of the WDT disable register is set to “1”, this bit is cleared.
0: WDT not running
1: WDT running

7.4.4.14. WDTCount register

Address	Base Address + 0x0034															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved								WCOUNT							
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

This register is used to start the WDT.

Bit7-0 WCOUNT (Watch dog timer COUNTer)
This bitfield sets the time count value for the WDT output interrupt signal.
When the WCLEA bit of the WDT clear register is set to “1”, the countdown value of the WDT counter is set on the positive edge of HCLK. (AHB Bus Clock)
While the WCLEA of the WDT clear register is “1”, this bitfield cannot be rewritten
When the WDIS bit of the WDT disable register is set to “1”, this bitfield is reset to 0xFF.

The count time is determined by 1 cycle time × the set count value of this function. The count number can be set in the range 0x10_0000 … 0xFF0_0000. The 20 subordinate position bits of the number of counts are masked. Please set the 8 high order bits.
The WDT timer can be set from 0.01s to 3.2s.
0x0: Reserved
0x1: HCLK83MHz -> about 0.01s
:
0xF: HCLK83MHz-> about 0.18s
:
0xF0: HCLK83MHz-> about 3.0s
:
0xFF: HCLK83MHz -> about 3.2s

7.4.4.15. WDTDisable register

Address	Base Address + 0x0038															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved															WDIS
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0/W1
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register disables the WDT.

Bit0 WDIS (Watchdog timer DISable)
When "1" is written to this bit, the WCLEA bit and the WCOUNT bitfield are reset.
After the WDT has been disabled, this bit is cleared.

Even if this bit is set to “1” when the WDT times out, the WDT interrupt bit changes to “1”.
0: WDT enabled
1: WDT disable

7.4.4.16. WDTCountCheck register

Address	Base Address + 0x003C															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved				WDTCHECK											
R/W	R0	R0	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	WDTCHECK															
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is used to check the count value of the WDT.

Bit27-0 WDTCHECK (Watch Dog Timer count CHECK)
This bit monitors the status of the WDT count value.
This bitfield is read-only.

7.4.4.17. AllSoftResetEnable register

Address	Base Address + 0x0040															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved														TSIGA	TWDTA
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R	R
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved														ASIG	AWDT
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register can initiate an All Soft Reset start using the WDT or a SIG error interrupt (SIG_Diff_interrupt).

Bit0 AWDT (All Soft Reset enable by WDT)
When this bit set to 1 and the WDT times out, an All Soft Reset is issued (ox_ASRST=0).
This bit is not cleared by the All Soft Reset (ix_ASRST=0).
0:
1: If the WDT times out an All Soft Reset is issued.

Bit1 ASIG (All Soft Reset enable by SIG error)
When this bit set to 1 and the SIG error (SIG_Diff_interrupt) times out an All Soft Reset is issued (ox_ASRST=0).
This bit is not cleared by the All Soft Reset (ix_ASRST=0).
0:
1: On a SIG error, an All Soft Reset is issued.

Bit16 TWDTA (Trigger start by WDT and All Soft Reset)
When the AWDT bit is 1 and the WDT times out, this bit is changed to 1.
After releasing the All Soft Reset and when this bit is 1, a trigger start by WDT is executed.
The executed trigger start operates in a similar way to a trigger start by Priority1 (CMDSEQ access to embedded Flash Base Address+0x04).
This bit is set to "0" when a trigger start by a WDT time out has completed after the All Soft Reset release.
The end status of a trigger start by the WDT after an All Soft Reset release is signalled in the TWDTS bit of the Status register.
This bit is not cleared by an All Soft Reset (ix_ASRST=0).
0:
1: Since WDT became a timeout, trigger start is processing.

Bit17 TSIGA (Trigger start by SIG error and All Soft Reset)
When the ASIG bit is 1 and a SIG error interrupt (SIG_Diff_interrupt) occurs, this bit changes to 1.
After releasing the All Soft Reset and when this bit is 1, a trigger start by the SIG error interrupt (SIG_Diff_interrupt) is executed.
The executed trigger start operates in a similar way to a trigger start by Priority1 (CMDSEQ access to embedded Flash Base Address+0x04).
This bit is set to "0" when a trigger start by a SIG error interrupt (SIG_Diff_interrupt) is completed after an All Soft Reset release.
The end status of trigger starting by SIG error interrupt (SIG_Diff_interrupt) after All soft reset release is reflected in the TWDTS bit of Status register.
This bit is not cleared by All soft reset (ix_ASRST=0).
0:
1: Since SIG error, trigger start is processing.

7.5. Application notes

7.5.1. Reset start

Figure 7.5-1 Reset start application note is shown in the following.

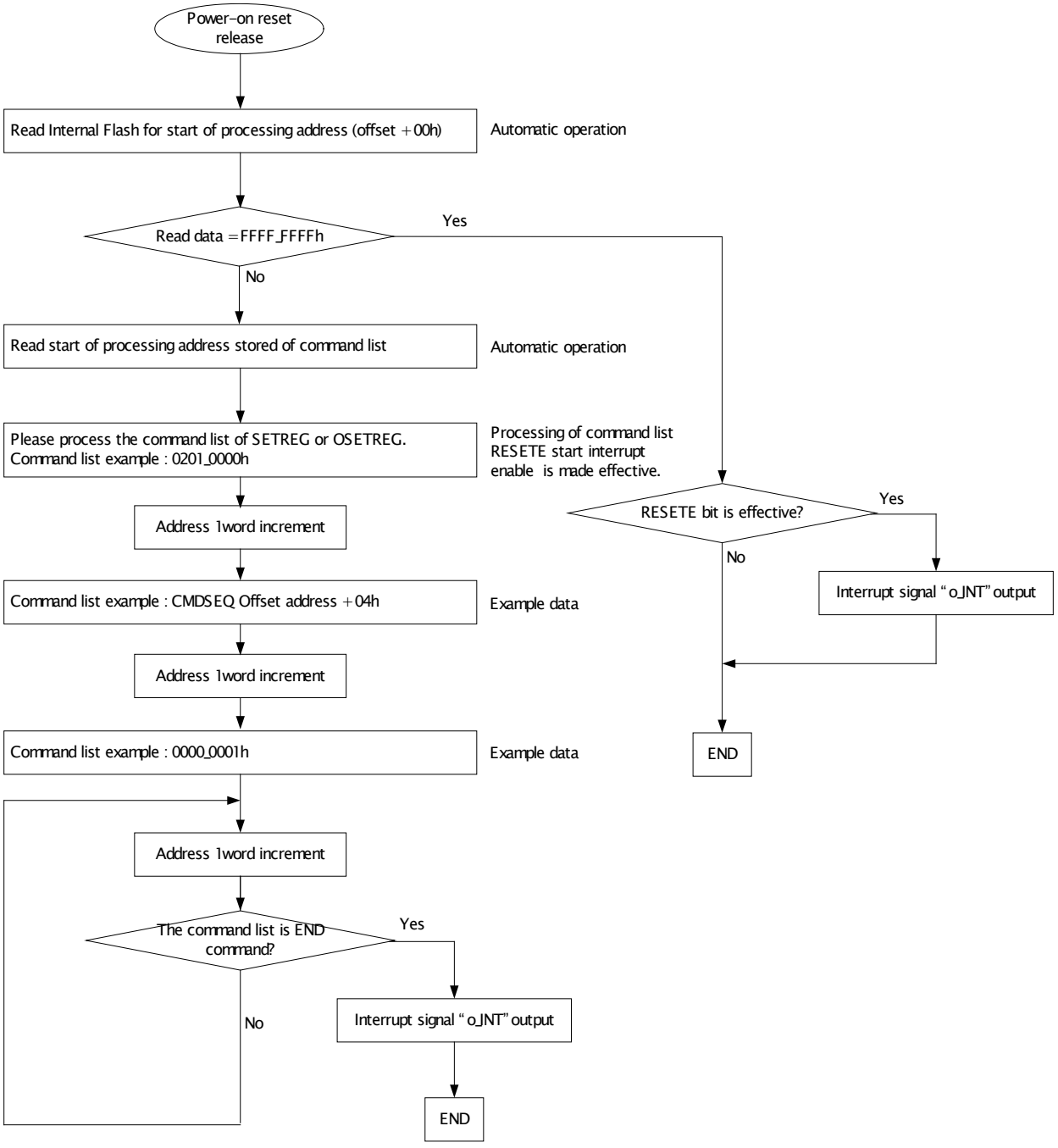


Figure 7.5-1 Reset start application note

7.5.2. Trigger start

Figure 7.5-2 Trigger start application note is shown in the following.

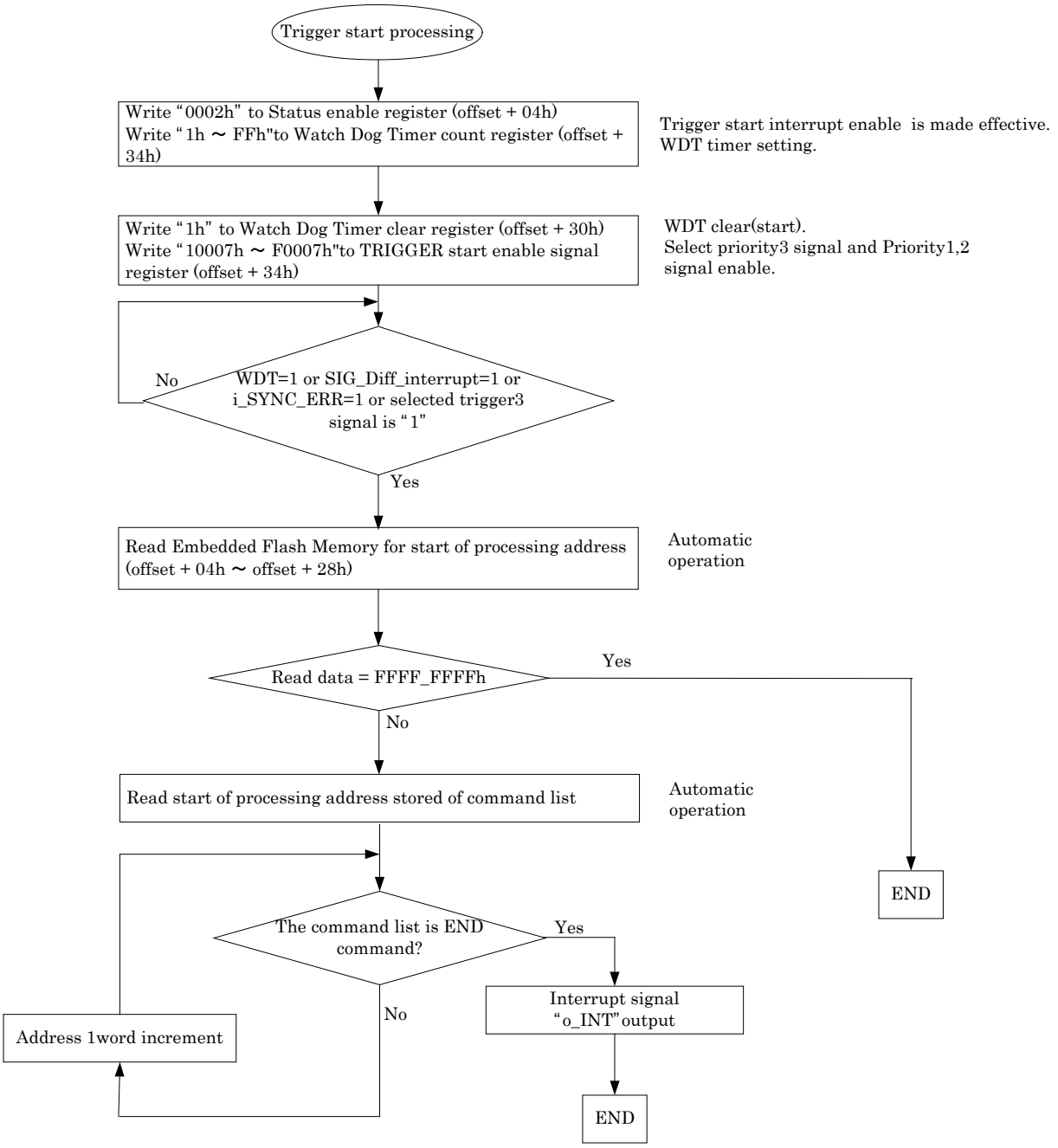


Figure 7.5-2 Trigger start application note

7.5.3. Register start

Figure 7.5-3 Register start application note is shown in the following.

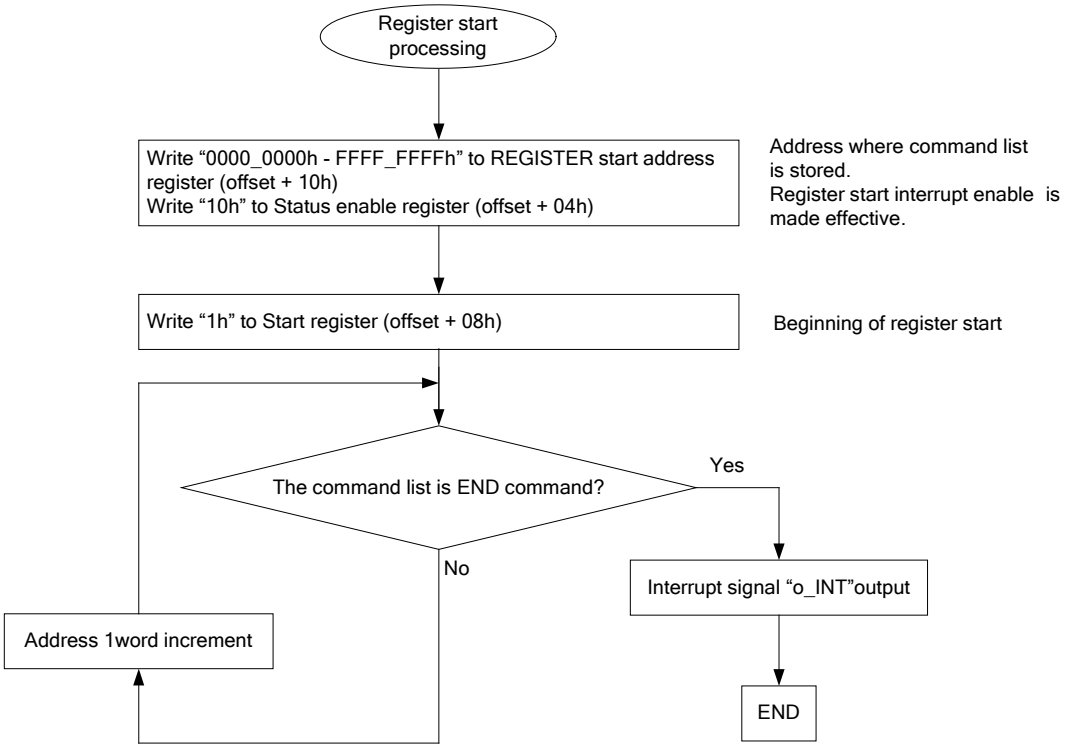


Figure 7.5-3 Register start application note

7.5.4. WDT processing

Figure 7.5-4 WDT processing application note is shown in the following.

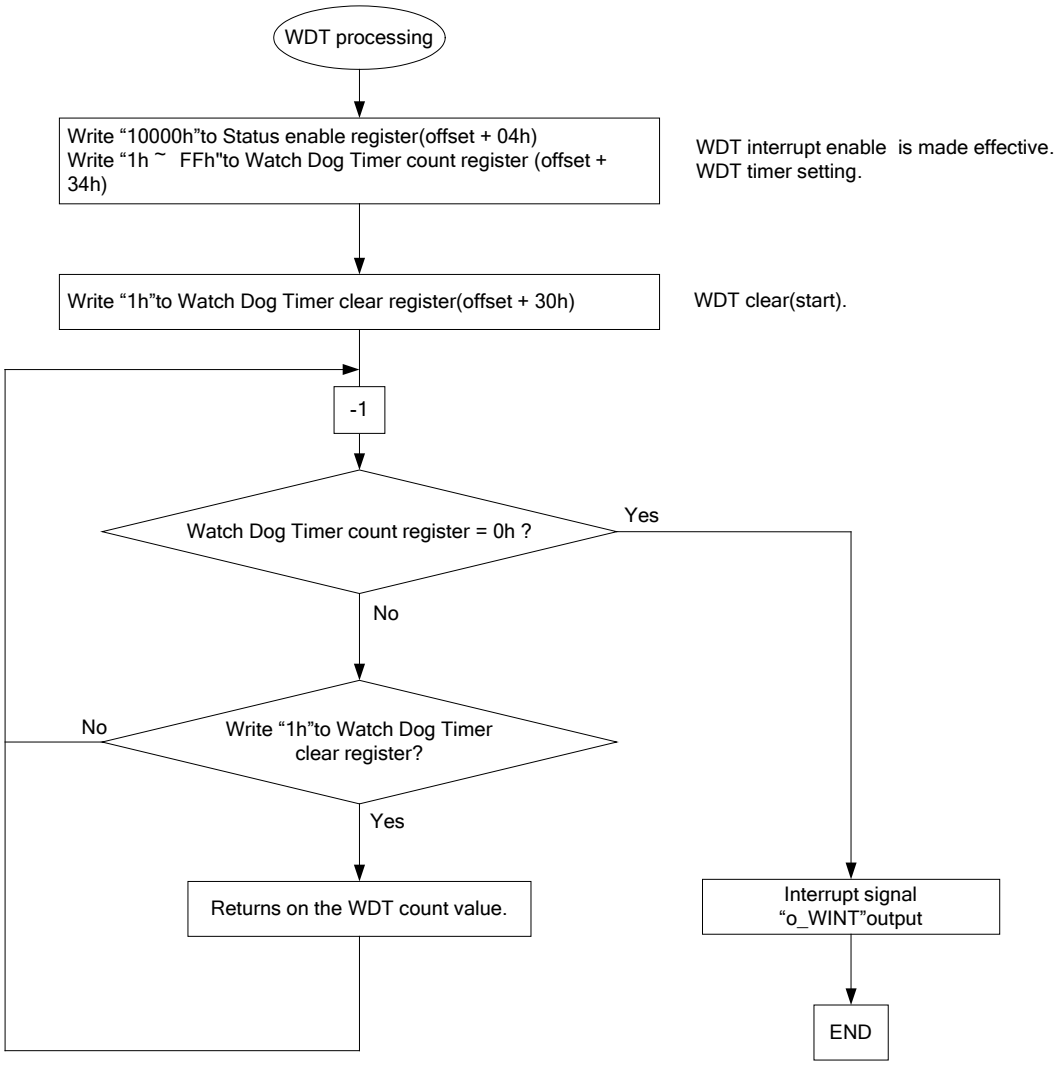


Figure 7.5-4 WDT processing application note

8. Sprite Engine (SPE)

This chapter describes the MB88F333 Sprite Engine (SPE).

8.1. Outline

The Sprite Engine is used to create images with the MB88F333. Versatile image processing methods can be used according to various register configurations set by software.

8.2. Features

The SPE has the following features.

8.2.1. Features

- Up to 512 Sprites
- 32 Special Sprites for Auto-Animation
- Up to 16777216 colors
- Two 8bpp Color Palette Table (ARGB-8888)
- 1bpp, 2bpp, 4bpp, 8bpp indirect color format
- ARGB-1555, RGB-565, ARGB-8888 direct color format
- Color format of each sprite can be set independently
- Alpha-blending (4/8bpp alpha)
- Image Reversing Function (horizontal or vertical)

8.2.2. Limitations

Note that the blink function can not be simultaneously used with the move function and image-switching functions for a special sprite.

8.3. Function

8.3.1. Block diagram

Figure 8-1shows SPE's block diagram.

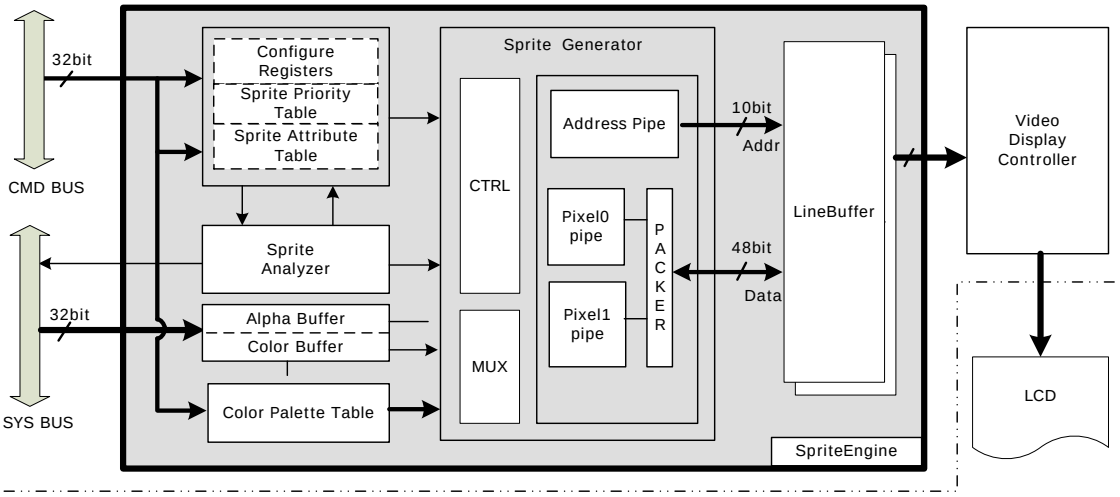


Figure 8-1 Block diagram of Sprite Engine

8.3.2. Processing Overview

The Sprite Engine is the core image processing unit in MB88F333. It has shifting, reversing, alpha-blending and other functions. All these functions can be controlled by the configuration of the corresponding registers by software.

8.3.2.1. Sprite Generation Flow

The Sprite Engine is a scanline rendering hardware implementation. With each scanline, the SPE analyzes the **SAT** (Sprite Attribute Table) and other registers to find the sprite with the highest priority having pixels positioned on the current scanline, reads its pattern data including alpha value, does image processing and then writes the final color data result into the LineBuffer.

8.3.2.2. Frame Generation Flow

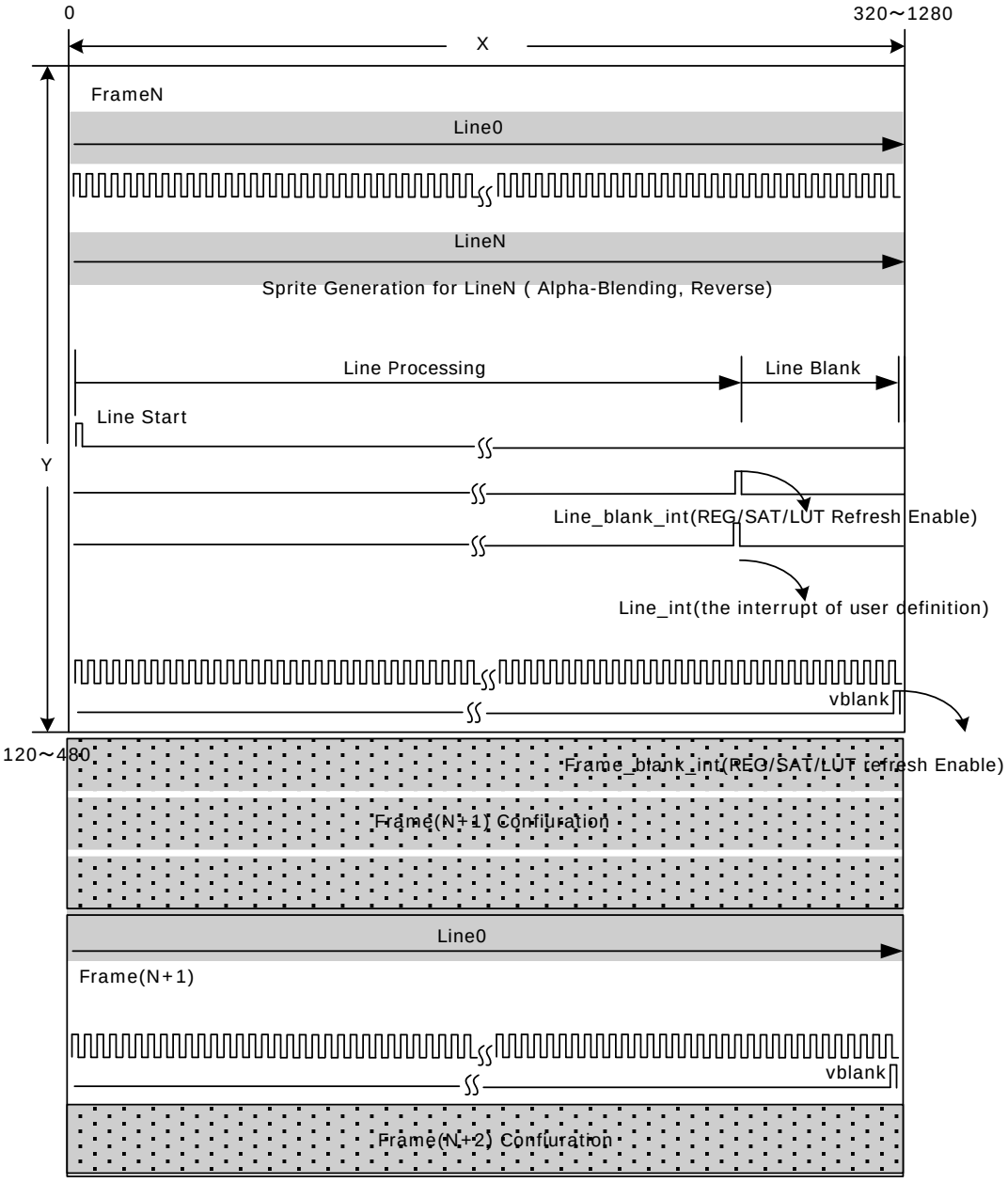


Figure 8-2 Processing Overview for a frame

Figure 8-2 shows two interrupt flags, “Line_blank_int” and “Line_int”. The “Line_blank_int” stands for the beginning of the line blank time (Line processing end). It can be enabled by setting the register ***LBKE(SPECTL)*** to “1”. And the “Line_int” is line blank time of a user defined line. It is controlled by the register ***ILNE(SPECTL)*** and a line number should be also specified by setting the register ***SPEILN***.

The **SAT (Sprite Attribute Table)** and other registers should be refreshed at vertical blank time or at the specified line blank time. However, some shadow registers can be configured at any time.

8.3.3. Sprite

A sprite is the source image display on a screen. It can be any pattern which is stored in the pattern memory and is processed according to its attributes, such as reversing…etc. The location of a sprite is defined by the top left-hand corner of the sprite pattern. The sprite can be moved pixel by pixel by setting the SAT (Sprite Attribute Table) accordingly.

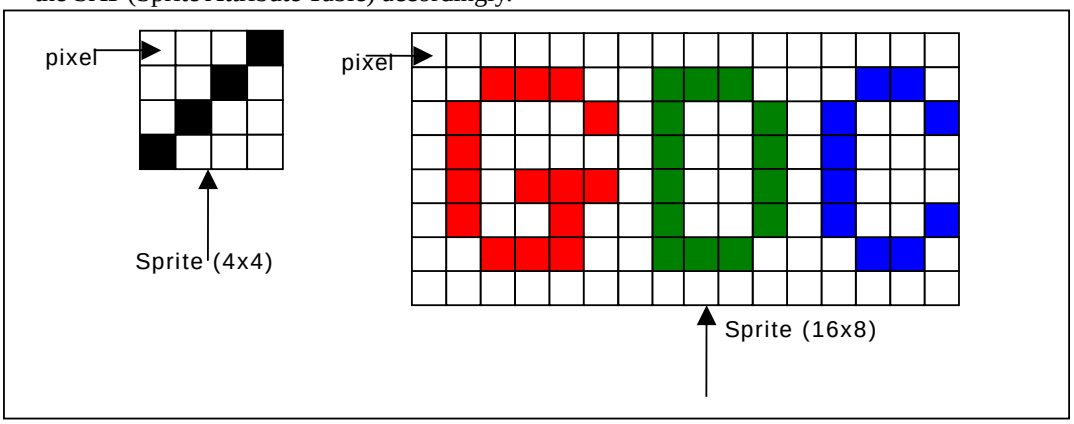


Figure 8-3 Sprite Definition

For the MB88F333, the size of a sprite is from 4x4pixels to 512x512pixels, and either X direction or Y direction can be set independently (X in 4 pixels steps, Y by 1 pixel). Figure8-3 illustrates a 4x4 pixels sprite, and a 16x8 pixels sprite.

8.3.4. Sprite Number

The serial number of every sprite is fixed in MB88F333, from 0 to 511, and they are consequently named **SPR0, SPR1 … SPR511**.

8.3.5. Sprite Priority

There are a maximum of 512 sprites in the sprite layer. The location of sprite can be defined anywhere freely, so they may overlap each other. There are two modes: fixing mode and SPT (Sprite Priority Table) mode, to decide the priority of the 512 sprites. The mode used depend on the setting of the register ***PRI (SPECTL)***

8.3.5.1. Fixing Mode

In fixing mode, the priority of a sprite is determined by its own serial number, i.e. from 0 to 511. By default, the priority of each sprite is fixed. The higher the serial number, the higher the priority. The higher the priority, the closer it is positioned with respect to a viewer. Of course, each sprite layer has a higher priority than the background layer.

Background < SPR0 < SPR1 < … < SPR31 < … < SPR511

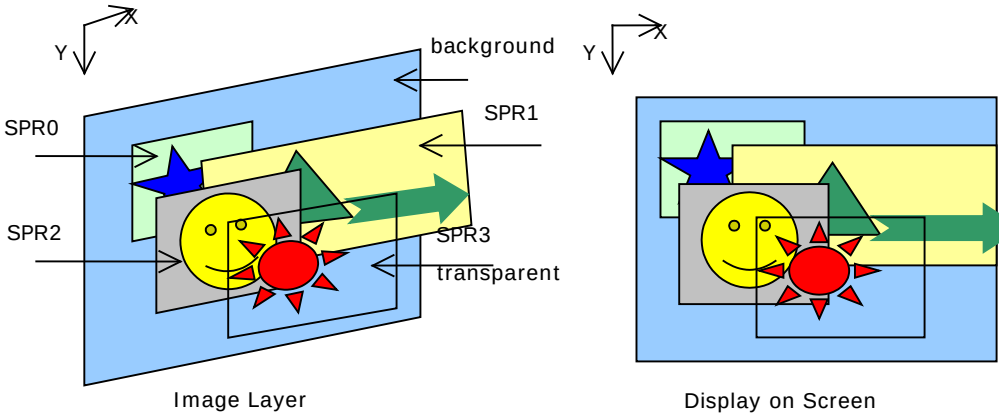


Figure 8-4 Sprite Priority (fixing mode)

Figure 8-4 illustrates 4 sprites displayed on screen simultaneously. If they overlapped each other, the visibility is determined by their priority. Except for pixels whose color value equals the transparent color (and when the transparent function is enabled), the higher priority pixel will be visible.

8.3.5.2. SPT Mode

In SPT mode, the priority of the 512 sprites is not decided by the sprite's number, but its setting in the SPT. The SPT is a 512x9bit register table for storing sprite numbers. The SPE determine the priority according to the address that the sprite has. The lower the address, the lower the priority. If using this mode, the SPT must be initialized before SPE processing.

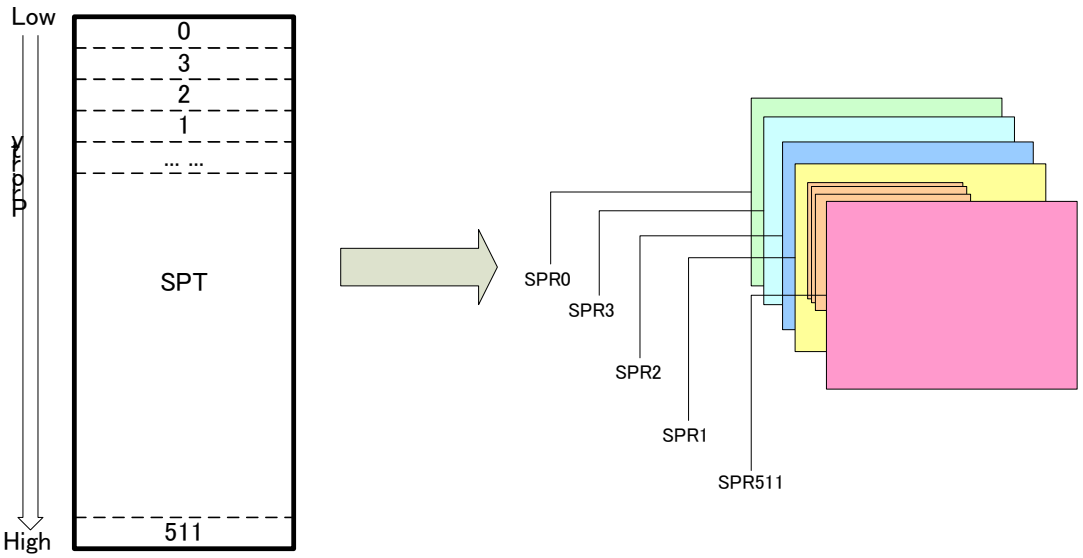


Figure 8-5 Sprite Priority (SPT mode)

Figure 8-5 illustrates the 512 sprites priorities when in SPT mode. The numbers written into the SPT are from a low address to a high address i.e. 0, 3, 2, 1, … , 511, then the sprites display order on the screen is SPR0, SPR3, SPR2, SPR1, … , SPR511 from the back to the front.

8.3.6. Sprite Display Area

All sprites can be imagined as above a display area, and the coordinates of a sprite are relative to the coordinates of the display area. But any parts of a sprite which are beyond the display area will not be displayed on the screen. The display area is configured by the user, and can be defined as 320~1280 pixels at X direction and 120~600 pixels at Y direction. Therefore, up to WVGA formats can be supported by MB88F333.

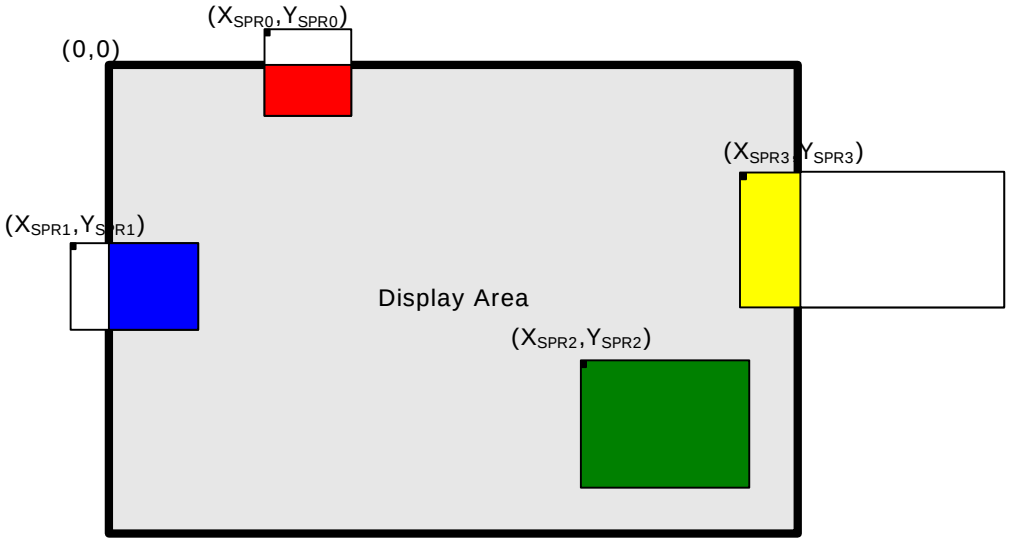


Figure 8-6 Sprite display area

The display coordinates of **SPRn** are derived from the **DX, DY** in the ***SAT*** register ***SPESnCR2*** (n=0~511). Either the **DX** and **DY** is a 12bit signed register, which can be defined from -2048 to +2047.

8.3.7. Pattern Data Format

Pattern data are the source image data of sprites. They are stored in memory with different color data formats and arranged line-by-line in order to improve the memory reading efficiency for the LineBuffer method.

Additionally, when the alpha table is enabled ***ATE***(SAT), the alpha table exists in the pattern memory ***ATS***(SAT), provided the color format of the pattern is not ARGB-8888 or ARGB-1555. It is located beneath the pattern data, as an alpha table which holds the alpha data of all pixels in the pattern. As for the pattern data, alpha data is also arranged line-by-line in the memory.

The start address of the pattern data and the alpha table should be a 32bit address.

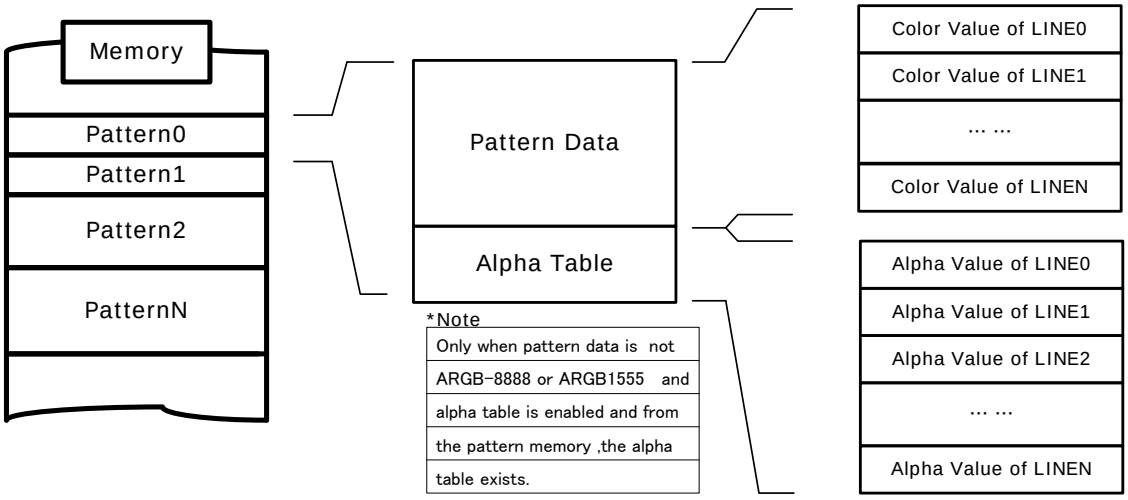


Figure 8-7 Structure of the Pattern Memory

The structure of line data including the color and the alpha is “little-endian”. This means that, a pixel with smaller X coordinates in the same line, should have a smaller address and bit number in memory. Also, the color data or the alpha data of all pixels must be successive with no gaps.

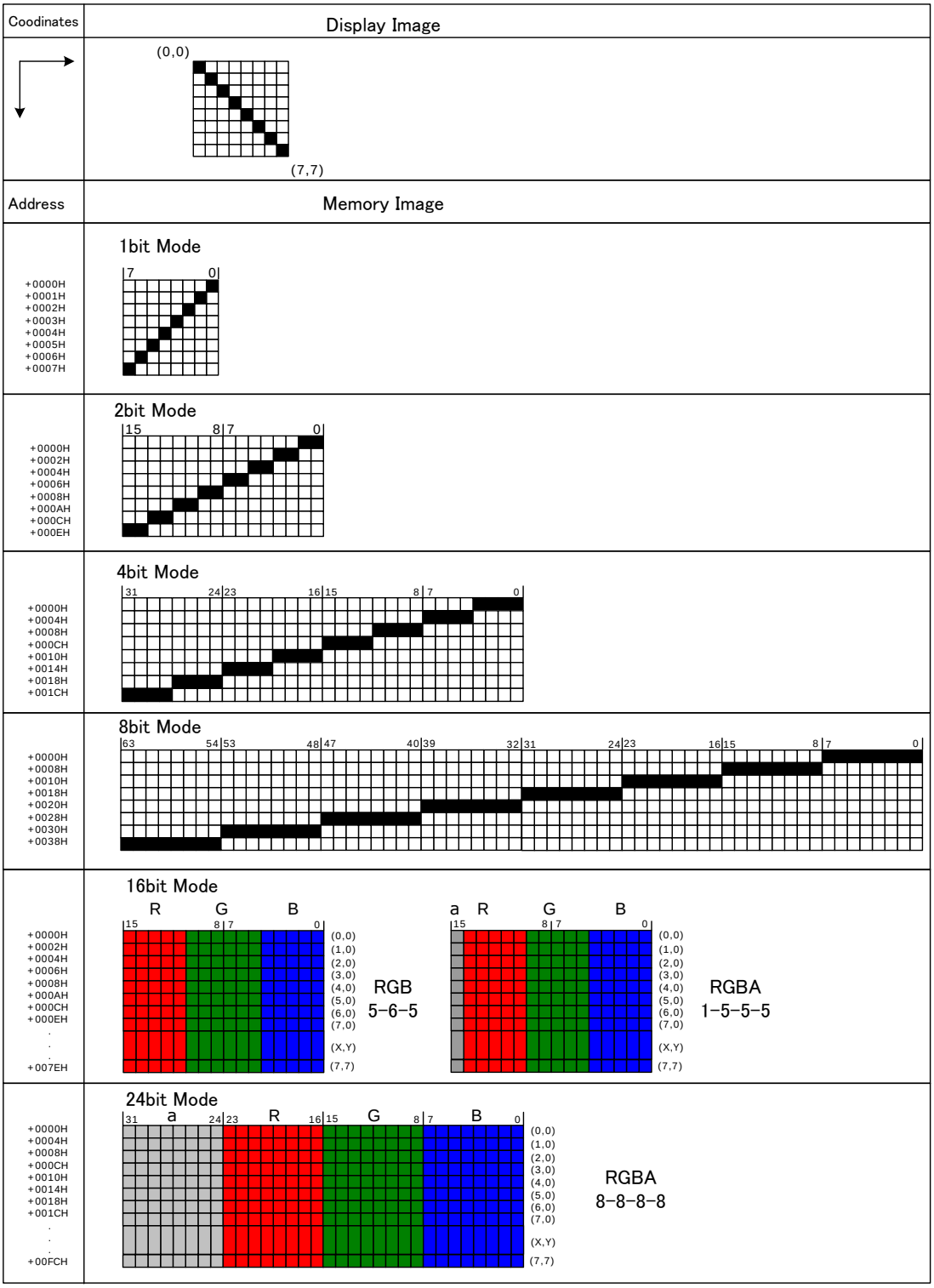


Figure 8-8 Pattern Data Format

Figure 8-8 illustrates an 8x8pixels sprite's pattern data in memory with different color data formats.
The alpha table has two modes, 4bpp and 8bpp, as determined by the corresponding register in SAT. The structure of the alpha table is the same as the color value when in 4bpp, 8bpp color format.

8.3.8. Transparent Color Function

Transparent color can be configured by the user to decide which color in the Color Palette table is transparent. A pixel of a sprite with the transparent color will not be displayed on the screen. Indirect color mode and direct color modes have different setting registers, ***SPETP0*** and ***SPETP1***.

For indirect color formats, the register ***SPETP0*** defines the transparent color of all four color formats, 1bpp, 2bpp, 4bpp and 8bpp.

The MB88F333 supports three direct color formats, RGB(565),ARGB(1555),ARGB(8888). For each color format, the MB88F333 decides the transparent color as shown below:

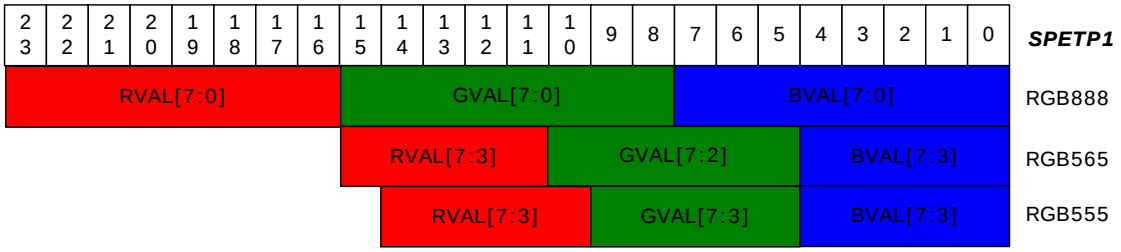


Figure 8-9 Transparent color format of direct color mode

8.3.9. Background Color Function

The background mode is configured by setting the register ***BGM* (*SPECTL*).** The background color will be displayed at a pixel on the screen where a sprite does not exists or the color is transparent at the location, in ordinary background mode (**BGM=”0”**). The background color is a single color, 24bpp (RGB-888), which can be configured by the user using the setting in the corresponding register ***SPEBG*.**

The SPE also supports another background mode for optimized overlapping with other layers in the display controller.

When the SPE works in **BGM = ”1”** mode, all transparent pixels will be treated as a fixed color data “0x000001” and the real color data “0x000001” will be converted to “0x000000”.

8.3.10. Color Palette Table

The SPE has a 2048 Byte RAM (512w x 32bit) for the **Color Palette Table**, which is named as ***LUT_RAM***. Every 32bit word of the RAM represents a single color in the format ARGB-8888. It can be made up of 256 palettes when in 1bit-mode, 128 palettes when in 2bit-mode, 32palettes when in 4bit-mode, or 2 palettes when in 8bit-mode.

A value in the **Color Palette Table** can be configured directly by the external CPU via the CMD bus or it can be transferred from external memory by the DMAC.

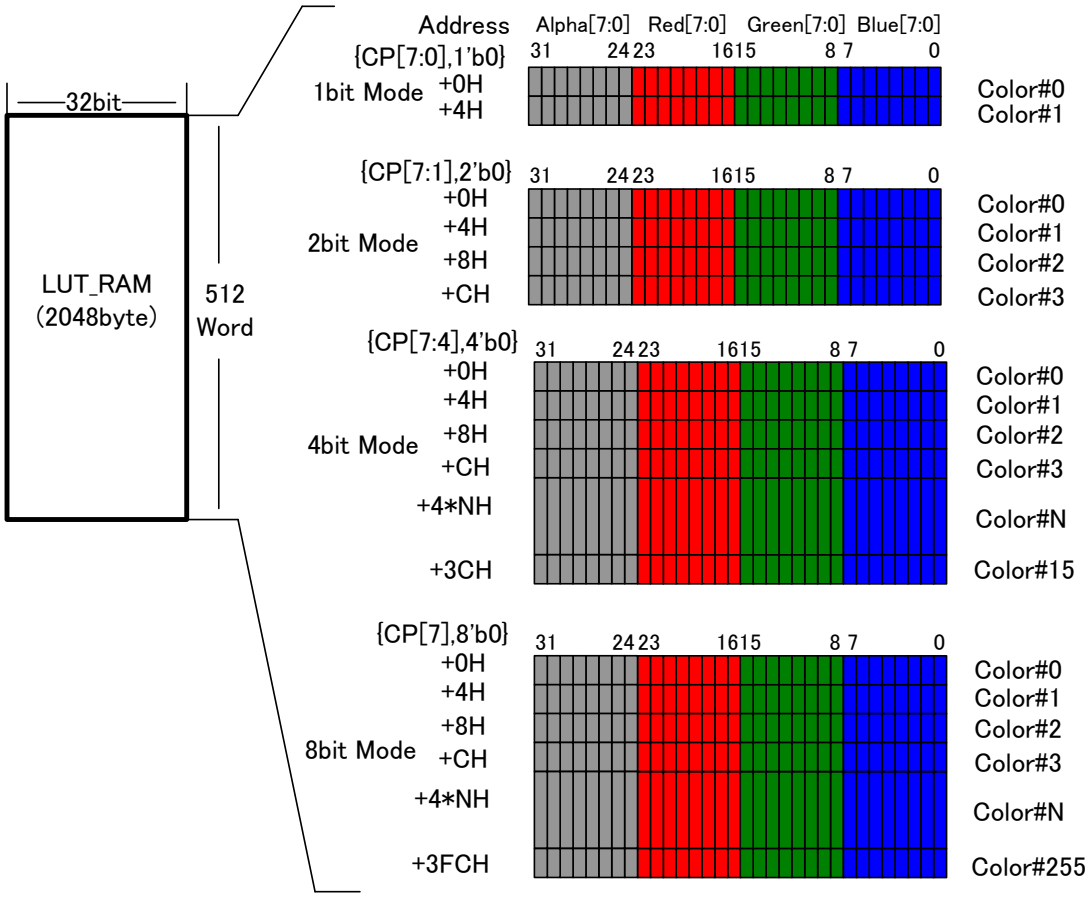


Figure 8-10 Color Palette Table

Which color palette a sprite uses is determined by the **CP[7:0]** field in the SAT register ***SPECnCR1*. (n=0~511)**

8.3.11. Reverse Function

SPE supports a vertical and horizontal reverse function. Whether or not a sprite is reversed when displayed on screen, is decided by its corresponding register ***HR*** and ***VR*** in the sprite attribute table. The horizontal reverse and the vertical reverse functions can be simultaneously enabled.

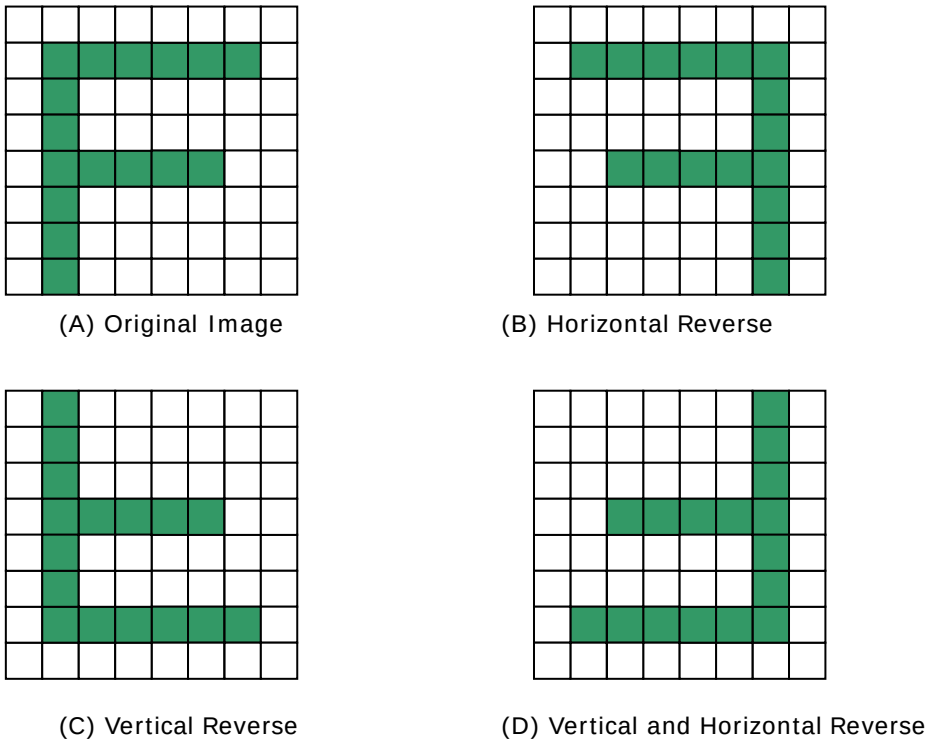


Figure 8-11 Reverse Function

8.3.12. Alpha blend

The SPE (Sprite Engine) supports an alpha-blending function.
When writing the image data of a drawing sprite into the LineBuffer, provided the corresponded register is enabled and depending on the setting of alpha-blending mode, alpha-blending will be done with the image data of sprites already drawn or with just a single color (RGB888). The alpha value data are obtained from the alpha table below the pattern data in the pattern memory or from the alpha channel in LUT_RAM when in pixel alpha-blending mode or from the sprite attribute table when in sprite alpha-blending mode.

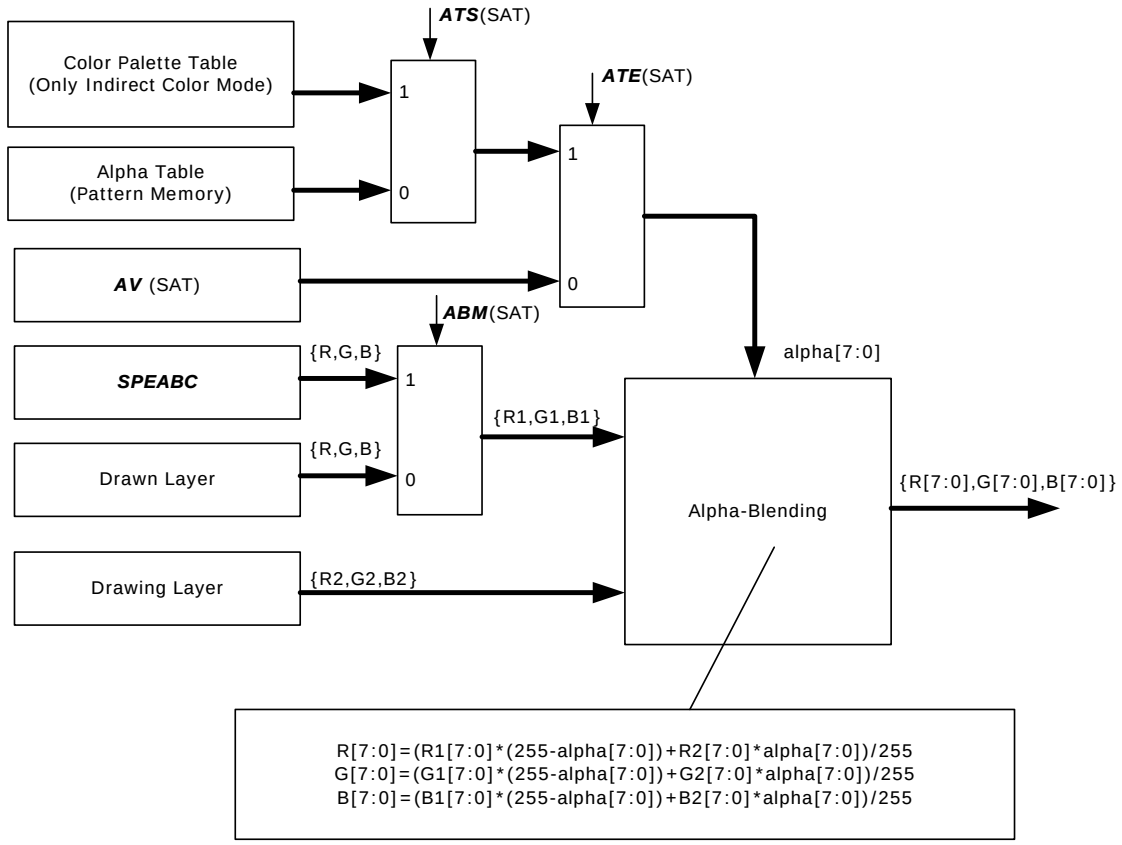


Figure 8-12 Alpha-blending Function

According to the calculation formula, when “alpha[7:0]” is 0x00, the blending result is that only the lower layer (the single color or the drawn layer) is remained. That is, the drawing layer is transparent. In opposite, if the alpha value is 0xff, the drawing layer is not transparent.

8.3.13. Special Sprites

There are 32 special sprites in the MB88F333. Any of all the 512 sprites can be defined as a special one by setting the register ***SPESSN0~SPESSN31***. These sprites can be used to realize many 2D animation functions automatically, such as blinking, auto-movement, and image-switching… All these functions are controlled by the registers ***SPESS(0~31)CR3, SPESS(0~31)CR4***,and ***SPESS(0~31)CR5***. Note that the blink function can not be simultaneously used with the move function and image-switching functions for a special sprite. However the auto-movement function and the image-switching function can be executed simultaneously.

8.3.13.1. Blink Function

The blink function of a special sprite is implemented by controlling the time that the sprite is displayed or not displayed. The visible period is set by ***VCNT0(SPESSnCR3)***, and the invisible period is set by ***VCNT1(SPESSnCR3)***; To enable/disable this function use the bit ***BLINK(SPESSnCR3)***. The number of times that the blink function will be executed is set by ***CNT0(SPESSnCR3)***. Before using this function for a special sprite, an initial state (visible or invisible), should be configured by setting ***SDET***.

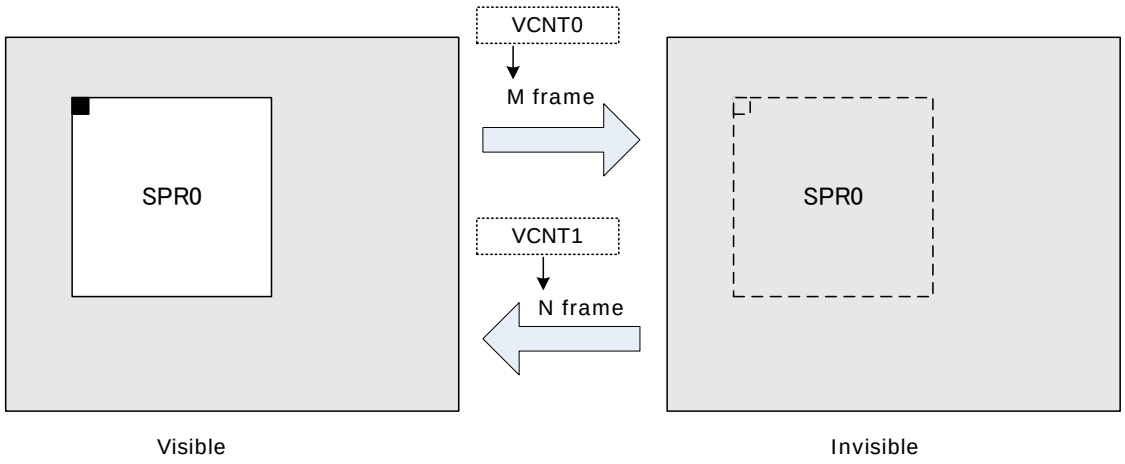


Figure 8-13 Blink Function

8.3.13.2. Auto-movement Function

Special sprites support many move methods. By setting the corresponding register ***SPESS(0~31)CR4***, it is possible to implement not only simple movements such as left-right and up-down movements, but also complex movements.

The interval time of every movement is set via register ***VCNT0(SPESSnCR3)***.

The auto-movement mode and the movement of all pixels at one time can be set individually in the X and Y directions. The moving pixels in the X direction are set using register ***MX(SPESSnCR4)*** and the moving pixels in the Y direction are set using register ***MY(SPESSnCR4)***. Both can be set to 1 ~ 256 pixels. Auto-movement mode in the X direction is set by register **XMV*(SPESSnCR4)*** and auto-movement mode in the Y direction is set by register **YMV*(SPESSnCR4)***.

When in auto-movement mode, the setting of the register ***CNT1(SPESSnCR3)*** means the times that a sprite makes a sequence of moves. However, the actual number of times that a sprite needs to move is different according to the setting of the auto-movement mode.

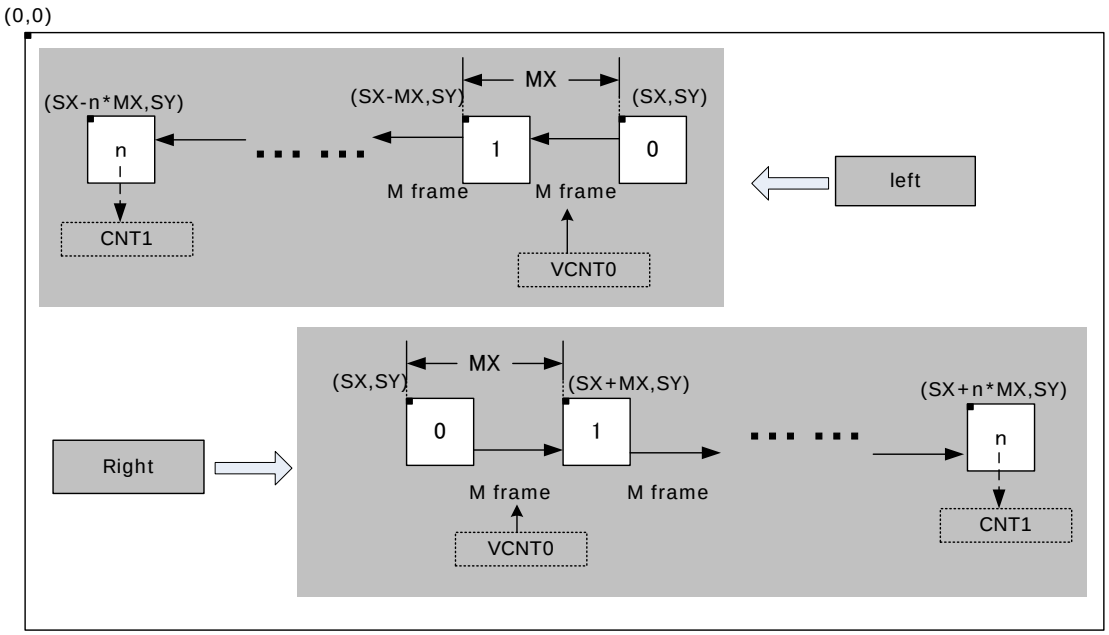


Figure 8-14 Basic movement in X direction

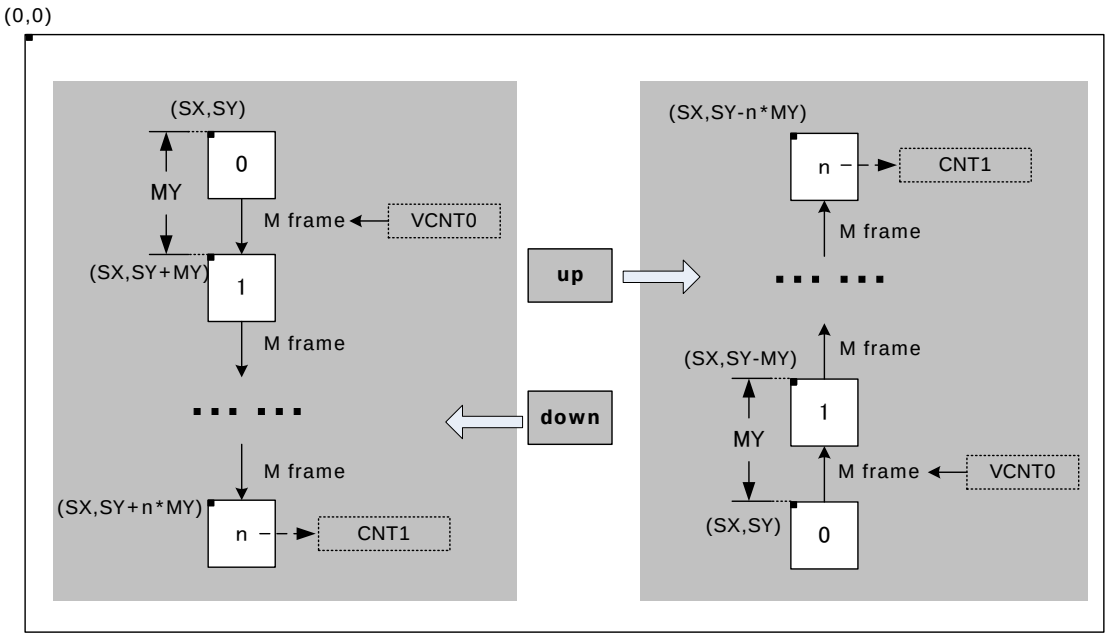


Figure 8-15 Basic movement in Y direction

Figure 8-14 and Figure 8-15 illustrates a basic move method in X and Y direction of the special sprite. In this mode, the sprite will move n (n=***CNT1***+1) times in a specific direction (an auto-movement unit).

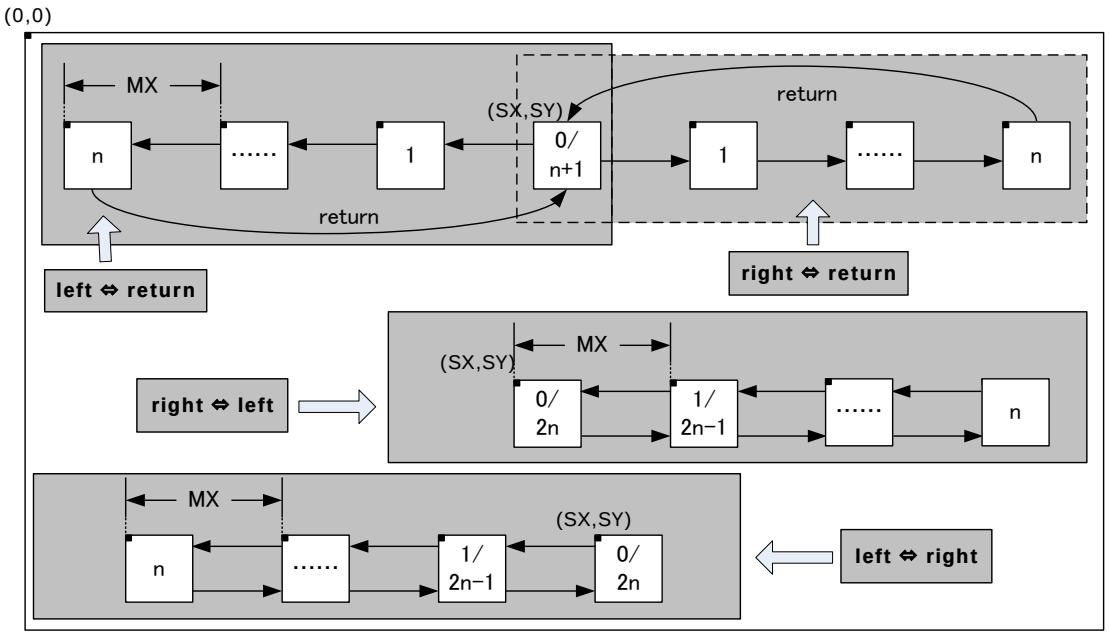


Figure 8-16 Specific movement in X direction

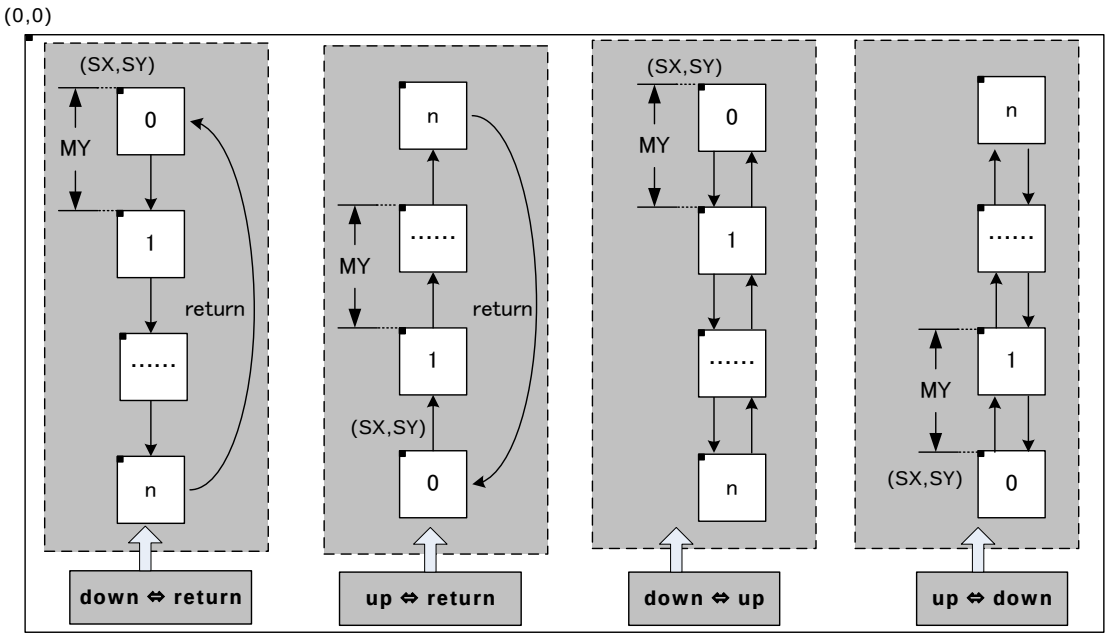


Figure 8-17 Specific movement in Y direction

Figure 8-16 and Figure 8-17 illustrate two specific movement methods of a special sprite in the X direction and Y direction. Both methods move the sprite in a direction n times (n=***CNT1***+1) and return it to the original location. The difference between the two methods is the number of movements when the sprite returns to the original location, which is moving either once or moving **n** times. For this reason they are named **STR** (Single Time Return) and **NTR** (N Times Return). Also, the auto-movement units of the two methods is different, (**n+1)** times and **2n** times.

Because the auto-movement methods can configured individually in the X direction and Y direction, combined auto-movement methods could be supported. The figures below show some examples. For every combination, the auto-movement unit is different.

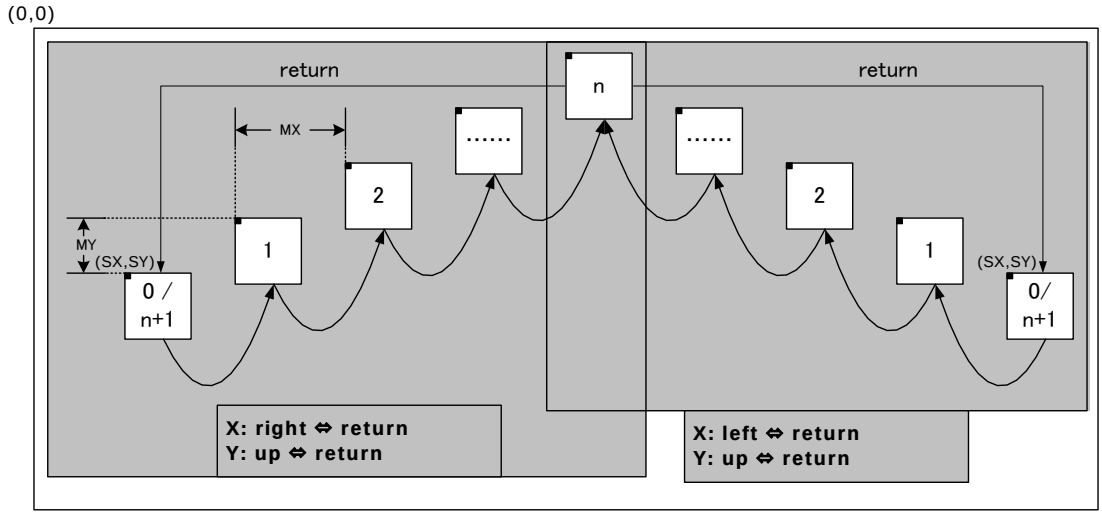


Figure 8-18 Combined movement at in the X direction with Y direction (1)

Figure 8-18 illustrates a movement method, X direction **STR** with Y direction **STR**, the auto-movement unit is **n+1** times.

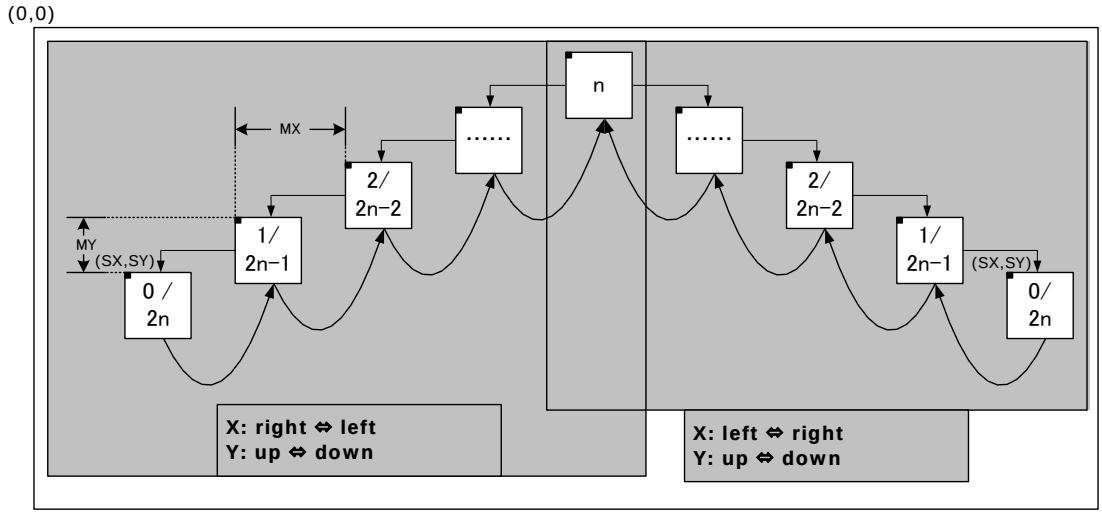


Figure 8-19 Combined movement in the X direction with Y direction (2)

Figure 8-19 illustrates a movement method, X direction **NTR** with Y direction **NTR**, the auto-movement unit is **2n** times.

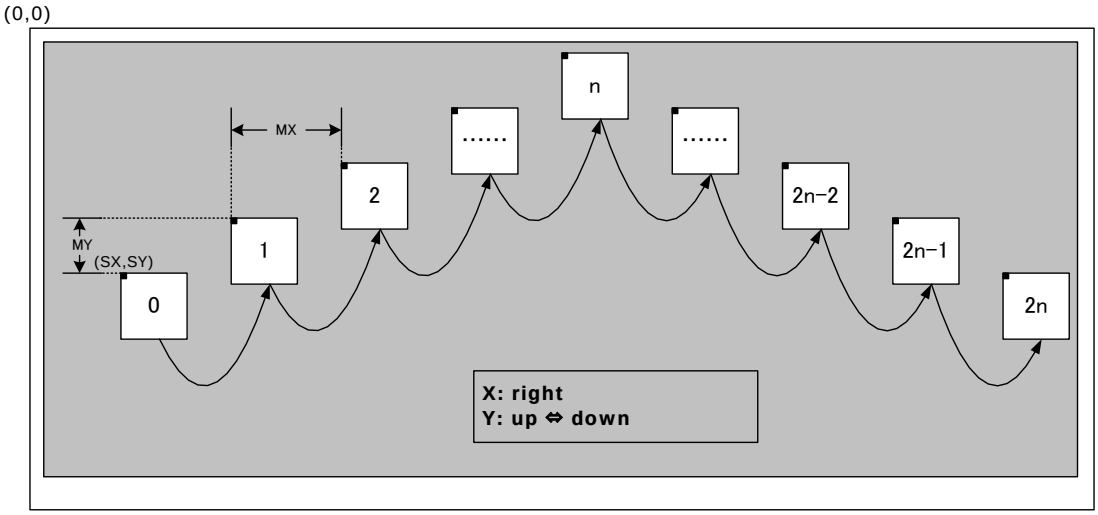


Figure 8-20 Combined movement in X direction with Y direction (3)

Figure 8-20 illustrates a movement method, X direction **single direction movement** with Y direction **NTR**, the auto-movement unit is **2n** times.

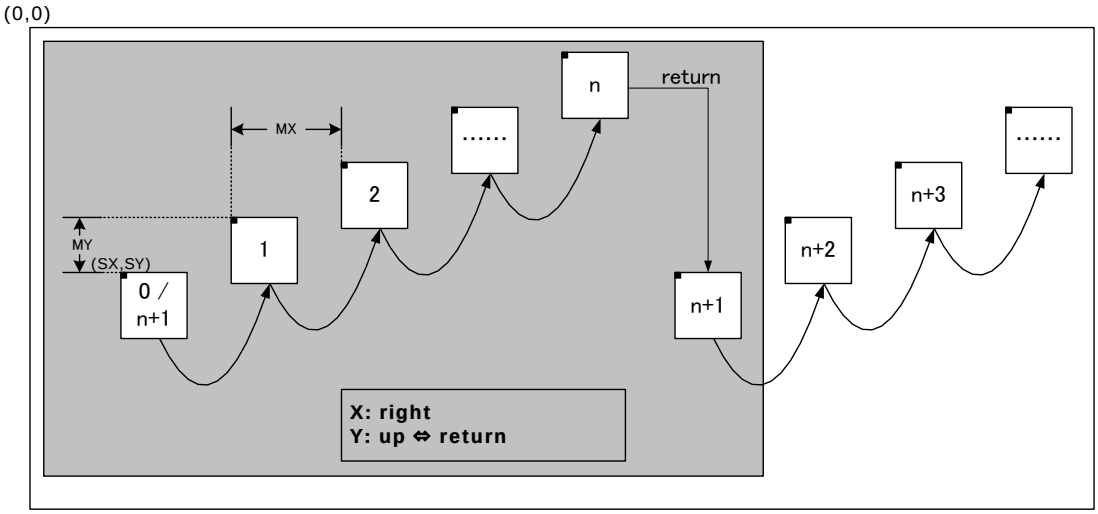


Figure 8-21 Combined movement in X direction with Y direction (4)

Figure 8-21 illustrates a movement method, X direction **single direction movement** with Y direction **STR**, the auto-movement unit is **n+1** times.

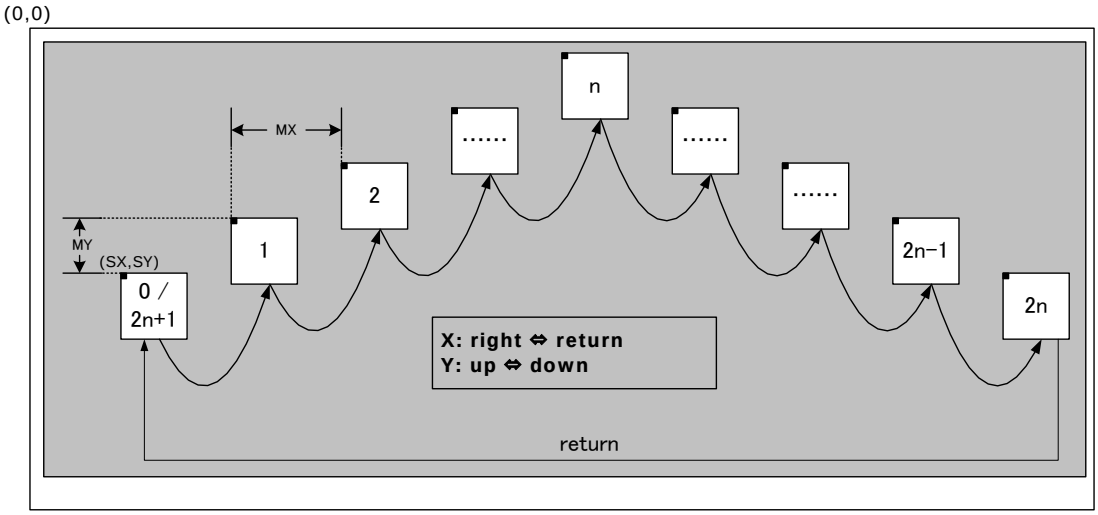


Figure 8-22 Combined movement in X direction with Y direction (5)

Figure 8-22 illustrates a movement method, X direction **STR** with Y direction **NTR (**a special combined movement method). The auto-movement unit is **2n+1** times. Notice that the locations of the sprite position **2n+1** and **2n** have the same Y coordinates.

8.3.13.3. Image-Switching Function

The special sprites may automatically load the next pattern data address from an address table in the pattern memory by itself when ***PAAL*** is set to "1". The start address of the address table used by a special sprite is configured by the register ***ADTA*****(SPESSnCR5)**. The interval time of the image-switching is also set by register ***VCNT0(SPESSnCR3)***. The series image numbers that a special sprite use are set by the register ***CNT1(SPESSnCR3)*** and the function execution count is set by the register ***CNT0(SPESSnCR3)***.
Because the first image of the sprite will be load by the special sprite itself automatically before processing the frame 0, the ***PA(SPESnCR0)*** of the sprite don't need to be initiated by users.
Note that when the image-switching function is used with the auto-movement function for a special sprite, the series image numbers will be equal to the auto-movement unit. That is, it depends on the auto-movement method setting.
In the pattern address table, one word represents a 32bit address of an image pattern in pattern memory. Because the ***PA(SPESnCR0)*** *i*s a 27bit register, only the lower 27bit of a word data in the pattern address table is effect.

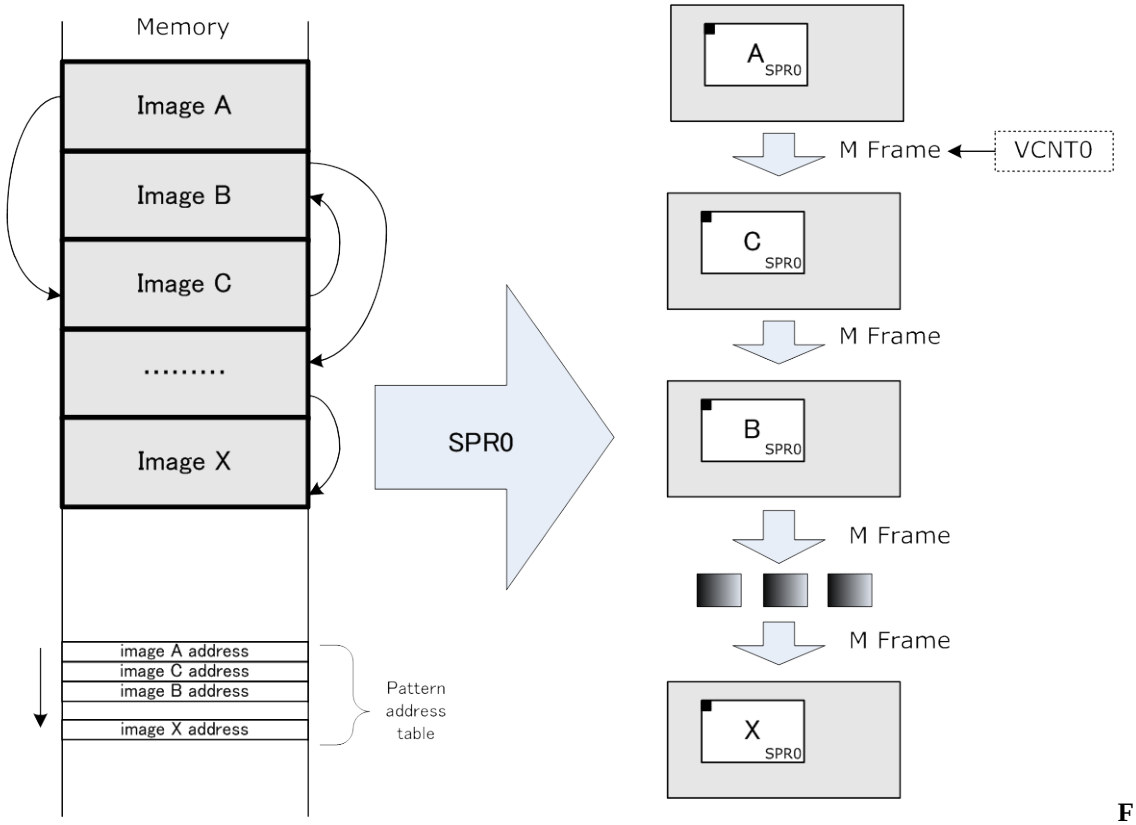


Figure 8-23 Image-Switching Function

8.4. Registers

8.4.1. Format of Register Descriptions

8.4.2. Global Address

For the module base address please refer to the chapter 3 Memory map.

8.4.3. Register summary

The base address of all registers in Sprite Engine is listed below:

Name	Base Address	Description
SPE_BASE	0x00020000	The base address of SPE registers
REG_BASE	SPE_BASE + 0x0000	Function registers base address
SPT_BASE	SPE_BASE + 0x1000	SPT base address
SAT_BASE	SPE_BASE + 0x2000	SAT base address
LUT_BASE	SPE_BASE + 0x6000	LUT base address

Table 8-1 Register Mapping Address

The lists of all registers in Sprite Engine are shown in Table 8-2.

Add(offset)	Bit	Name	Access	Initial Value	Description
Function Register (REG_BASE)					
0x000	23:0	SPEBG	R/W(S)	0x000000	Background color definition register
0x004	14:0	SPETP0	R/W(S)	0x0000	Transparent color definition register0(indirect)
0x008	23:0	SPETP1	R/W(S)	0x000000	Transparent color definition register1(direct)
0x00c	23:0	SPEABC	R/W(S)	0x000000	Alpha blend single color definition register
0x010	11:0	SPECTL	R/W	0x000	Sprite Engine control register
0x014	1:0	SPEST	R	0x0	Status of the sprite engine register
0x018	11:0	SPEILN	R/W	0x000	Interrupt line (0~4095) specify register
0x01C	24:0	SPESDN	R/W(S)	0x00000000	Sprite number range for display
0x020	24:0	SPEDETC	R/W(S)	0x00000000	Sprite display enable table clear register
0x024	10:0	SPEDPAR	R/W	0x000	Display area setting register
0x028	-	-	-	-	Reserved
0x02C	2:0	SPERSC	R/W	0x0	Resource FIFO enable register
0x030-0x0FC	-	-	-	-	Reserved
SDET (REG_BASE)					
0x100	31:0	SPESDE0	R/W(S)	0x00000000	Sprite display enable(SPR0~SPR31)
0x104	31:0	SPESDE1	R/W(S)	0x00000000	Sprite display enable(SPR32~SPR63)
...	...	...	...	...	...
0x13C	31:0	SPESDE15	R/W(S)	0x00000000	Sprite display enable(SPR480~SPR511)
0x140-0x1FC	-	-	-	-	Reserved
SSCR (Special Sprite Configure Register) (REG_BASE)					
0x200	31:0	SPESS0CR3	R/W	0x00000000	Special Sprite0 control register3
0x204	31:0	SPESS0CR4	R/W	0x00000000	Special Sprite0 control register4

0x208	31:0	SPESS0CR5	R/W	0x00000000	Special Sprite0 control register5
….	…	…	…	…	…
0x374	31:0	SPESS31CR3	R/W	0x00000000	Special Sprite31 control register3
0x378	31:0	SPESS31CR4	R/W	0x00000000	Special Sprite31 control register4
0x37C	31:0	SPESS31CR5	R/W	0x00000000	Special Sprite31 control register5
0x380-0x3FC	-	-	-	-	Reserved
0x400	8:0	SPESSN0	R/W	0x000	Special sprite specify register0
0x404	8:0	SPESSN1	R/W	0x000	Special sprite specify register1
…	…	…	…	…	…
0x47C	8:0	SPESSN31	R/W	0x000	Special sprite specify register31
SPT (SPT_BASE)					
0x000	8:0	SPESPRI0	R/W	…	Priority 0 set register
0x004	8:0	SPESPRI1	R/W	…	Priority 1 set register
0x008	8:0	SPESPRI2	R/W	…	Priority 2 set register
…	…	…	…	…	…
0x7FC	8:0	SPESPRI511	R/W	…	Priority 511 set register
SAT (SAT_BASE)					
0x0000	31:0	SPES0CR0	R/W	…	Sprite0 control register0
0x0004	31:0	SPES0CR1	R/W	…	Sprite0 control register1
0x0008	31:0	SPES0CR2	R/W	…	Sprite0 control register2
….	….	….	….	….	….
0x0BF0	31:0	SPES254CR2	R/W	…	Sprite254 control register2
0x0BF4	31:0	SPES255CR0	R/W	…	Sprite255 control register0
0x0BF8	31:0	SPES255CR1	R/W	…	Sprite255 control register1
0x0BFC	31:0	SPES255CR2	R/W	…	Sprite255 control register2
…	…	…	…	…	…
0x17F4	31:0	SPES511CR0	R/W	…	Sprite511 control register0
0x17F8	31:0	SPES511CR1	R/W	…	Sprite511 control register1
0x17FC	31:0	SPES511CR2	R/W	…	Sprite511 control register2
LUT (LUT_BASE)					
0x000	31:0	SPELUTS0	R/W	…	LUT Setting register0
0x004	31:0	SPELUTS1	R/W	…	LUT Setting register1
0x008	31:0	SPELUTS2	R/W	…	LUT Setting register2
…	…	…	…	…	…
0x7FC	31:0	SPELUTS511	R/W	…	LUT Setting register511

Table 8-2 Summary of Sprite Engine Registers

*Note:

1: (S) indicates that the register is a shadow register. The shadow register will be loaded with the next VSYNC event.

2: All reserved or unused address locations should not be accessed.

3: Only the SAT supports the byte or the half word accessing.

8.4.4. Register description

8.4.4.1. Background Color Register, SPEBG

Address: 0x000; Initial value: 0x00000000;

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
23	22	21	20	19	18	17	16
RVAL[7]	**RVAL[6]**	**RVAL[5]**	**RVAL[4]**	**RVAL[3]**	**RVAL[2]**	**RVAL[1]**	**RVAL[0]**
15	14	13	12	11	10	9	8
GVAL[7]	**GVAL[6]**	**GVAL[5]**	**GVAL[4]**	**GVAL[3]**	**GVAL[2]**	**GVAL[1]**	**GVAL[0]**
7	6	5	4	3	2	1	0
BVAL[7]	**BVAL[6]**	**BVAL[5]**	**BVAL[4]**	**BVAL[3]**	**BVAL[2]**	**BVAL[1]**	**BVAL[0]**

Bits	Name	Function
31:24	-------	Reserved
23:16	RVAL	Red color definition of background
15:8	GVAL	Green color definition of background
7:0	BVAL	Blue color definition of back ground

*note

1) This register is active only when the SPECTL.BGM is “0”.

2) This register is a “shadow register”.

8.4.4.2. Transparent Color Register0 (indirect color mode), SPETP0

Address: 0x004; Initial value: 0x00000000;

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8
0	**TP1B**	**TP2B[2]**	**TP2B[1]**	**TP4B[3]**	**TP4B[2]**	**TP4B[1]**	**TP4B[0]**
7	6	5	4	3	2	1	0
TP8B[7]	**TP8B[6]**	**TP8B[5]**	**TP8B[4]**	**TP8B[3]**	**TP8B[2]**	**TP8B[1]**	**TP8B[0]**

Bits	Name	Function
31:15	-------	Reserved
14	TP1B	Transparent color code definition when in 1bit color mode
13:12	TP2B	Transparent color code definition when in 2bit color mode
11:8	TP4B	Transparent color code definition when in 4bit color mode
7:0	TP8B	Transparent color code definition when in 8bit color mode

*note

1) This register is a “shadow register”.

8.4.4.3. Transparent Color Register1 (direct color mode), SPETP1

Address: 0x008; Initial value: 0x00000000;

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
23	22	21	20	19	18	17	16
RVAL[7]	**RVAL[6]**	**RVAL[5]**	**RVAL[4]**	**RVAL[3]**	**RVAL[2]**	**RVAL[1]**	**RVAL[0]**
15	14	13	12	11	10	9	8
GVAL[7]	**GVAL[6]**	**GVAL[5]**	**GVAL[4]**	**GVAL[3]**	**GVAL[2]**	**GVAL[1]**	**GVAL[0]**
7	6	5	4	3	2	1	0
BVAL[7]	**BVAL[6]**	**BVAL[5]**	**BVAL[4]**	**BVAL[3]**	**BVAL[2]**	**BVAL[1]**	**BVAL[0]**

Bits	Name	Function
31:24	-------	Reserved
23:16	RVAL	Red color definition of transparent color when in direct color mode
15:8	GVAL	Green color definition of transparent color when in direct color mode
7:0	BVAL	Blue color definition of transparent color when in direct color mode

*note

1) This register is a “shadow register”.

8.4.4.4. Alpha-Blending Color Register, SPEABC

Address: 0x00C; Initial value: 0x00000000;

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
23	22	21	20	19	18	17	16
RVAL[7]	**RVAL[6]**	**RVAL[5]**	**RVAL[4]**	**RVAL[3]**	**RVAL[2]**	**RVAL[1]**	**RVAL[0]**
15	14	13	12	11	10	9	8
GVAL[7]	**GVAL[6]**	**GVAL[5]**	**GVAL[4]**	**GVAL[3]**	**GVAL[2]**	**GVAL[1]**	**GVAL[0]**
7	6	5	4	3	2	1	0
BVAL[7]	**BVAL[6]**	**BVAL[5]**	**BVAL[4]**	**BVAL[3]**	**BVAL[2]**	**BVAL[1]**	**BVAL[0]**

Bits	Name	Function
31:24	-------	Reserved
23:16	RVAL	Red color definition of alpha blending with single color
15:8	GVAL	Green color definition of alpha blending with single color
7:0	BVAL	Blue color definition of alpha blending with single color

*note

1) This register is a “shadow register”.

2) This register is used when alpha-blending mode is set to the mode blending with a single color.

8.4.4.5. Control Register, SPECTL

Address: 0x0010; Initial value: 0x00000000;

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8
0	0	0	**EBCE**	**BERE**	**PERE**	**ILNE**	**LBKE**
7	6	5	4	3	2	1	0
0	0	**SRL**	**SSPE**	**BGM**	**SRE**	**PRI**	**SEN**

Bits	Name	Function
31:13	-------	Reserved
12	EBCE	"Enable Bits Changed" Interrupt enable: 1: Enable. 0: Not enable.
11	BERE	Bus error interrupt flag enable: 1: Enable. 0: Not enable.
10	PERE	Processing Error interrupt flag enable: 1: Enable. 0: Not enable.
9	ILNE	Line interrupt enable: 1: Enable. 0: Not enable.
8	LBKE	Line blank interrupt enable: 1: Enable. 0: Not enable.
7:6	-------	Reserved
5	SRL	Shadow registers loading active: 1: Load the shadow registers at the next VSYNC. 0: Do not load the shadow registers.
4	SSPE	Special sprite function Enable: 1: Enable. 0: Not enable.
3	BGM	Background mode select: 1: Fixed value 0x000001. 0: the data set in the register SPEBG.
2	SRE	Shadow registers updates enable: 1: Enable. 0: Not enable.
1	PRI	Sprite display priority mode select: 1: SPT mode. 0: Fixing mode. Fixed by sprite number. SPR0 < SPR1 < … < SPR511
0	SEN	Sprite Engine Processing Enable: 1: Enable. 0: Not enable.

*note:

1) ***SEN*** should be set when the initial configuration of other function registers is completed and the pattern data is ready.

2) If ***SRE*** is set to “1”, all the “shadow registers” will only be updated at the next VSYNC after the ***SRL*** is set to “1.

3) "Enable Bits Changed" Interrupt is a timing flag that the registers ***SPEDE0-SPEDE15*** are updated by

the “shadow registers” themselves

8.4.4.6. Status Register, SPEST

SPEST is a read-only register.

Address: 0x0014; Initial value: 0x00000000;

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

Bits	Name	Function
31:2	-------	Reserved
1	SSUP	The sprite attribute of special sprite are updating by SPE itself.
0	BSY	Sprite Engine busy flag: 1 = SPE is busy. 0 = SPE is idle

8.4.4.7. Interrupt Line Register, SPEILN

Address: 0x0018; Initial value: 0x00000000;

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

Bits	Name	Function
31:12	-------	Reserved
11:0	ILIN	Interrupt Line Number (0 ~ 4095)

8.4.4.8. Sprite Display Number, SPESDN

Address: 0x001C; Initial value: 0x00000000;

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

Bits	Name	Function
31:25	------	Reserved
24:16	ESN	SPECTL.PRI=1: End address of SPT. (32bit word address, 0~511) SPECTL.PRI=0: End Sprite Number to Display. (from **SPR0~SPR511**)
15:9	------	Reserved
8:0	BSN	SPECTL.PRI=1: Begin address of SPT. (32bit word address, 0~511) SPECTL.PRI=0: Begin Sprite Number to Display. (from **SPR0~SPR511**)

*note

1) This register has a different function when the register ***SPECTL*** is set.
2) ***ESN*** must be set to a higher value than ***BSN***.
3) This register is a “shadow register”.

8.4.4.9. Sprite Display Enable Table Clear Register, SPEDETC

Address: 0x020; Initial value: 0x00000000;

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	**ESN[8]**
23	22	21	20	19	18	17	16
ESN[7]	**ESN[6]**	**ESN[5]**	**ESN[4]**	**ESN[3]**	**ESN[2]**	**ESN[1]**	**ESN [0]**
15	14	13	12	11	10	9	8
GO	**DEV**	0	0	0	0	0	**BSN[8]**
7	6	5	4	3	2	1	0
BSN[7]	**BSN[6]**	**BSN[5]**	**BSN[4]**	**BSN[3]**	**BSN[2]**	**BSN[1]**	**BSN [0]**

Bits	Name	Function
31:25	------	Reserved
24:16	ESN	End Sprite Number (from **SPR0~SPR511**)
15	GO	Clear start trigger. This bit will be cleared by SPE itself.
14	DEV	Display Enable Value 0: Do not display 1: Display
13:9	------	Reserved
8:0	BSN	Begin Sprite Number (from **SPR0~SPR511**)

*note:

This register is a “shadow register”

8.4.4.10. Display Area Register, SPEDPAR

Address: 0x024; Initial value: 0x00000000;

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

Bits	Name	Function
31:26	-------	Reserved
10:0	DPWD	Display Width (1 ~ 80) x 16 pixels. DPWD[3:0] are fixed to 0;

8.4.4.11. Resource FIFO Enable Register, SPERSC

Address: 0x02C; Initial value: 0x00000000;

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

Bits	Name	Function
31:3	-------	Reserved
2	SPTE	Resource FIFO for configure SPT: 1: Enable. 0: Not enable.
1	SATE	Resource FIFO for configure SAT: 1: Enable. 0: Not enable.
0	LUTE	Resource FIFO for configure LUT: 1: Enable. 0: Not enable.

*Notes:

1) The Resource FIFO temporarily keeps the SPT, SAT and LUT configuration information when SPE is drawing a frame, if the correlated register of each field is enabled.

2) When out of the SPE processing time, the corresponding register field will be configured through the Resource FIFO.

3) Notice that when in the SPE processing time, only a total 512 registers setting information can be kept into the Resource FIFO.

4) The FIFO is 512 words in size and is flushed with the VSYNC signal

8.4.4.12. SPT (Sprite Priority Table)

The display priority of up to 512 sprites can be programmed in the SPT. SPT is a 512x9 register table for setting 512-level display priorities. And this table is effective only when priority mode is SPT mode. In SPT, lower address means lower priority. So that the sprite number which the sprite is upper than other sprites should be written into the higher address in SPT. And through setting the register ***BSN*** and ***ESN*** (***SPESDN***), series sprites switching function can be realized easily.

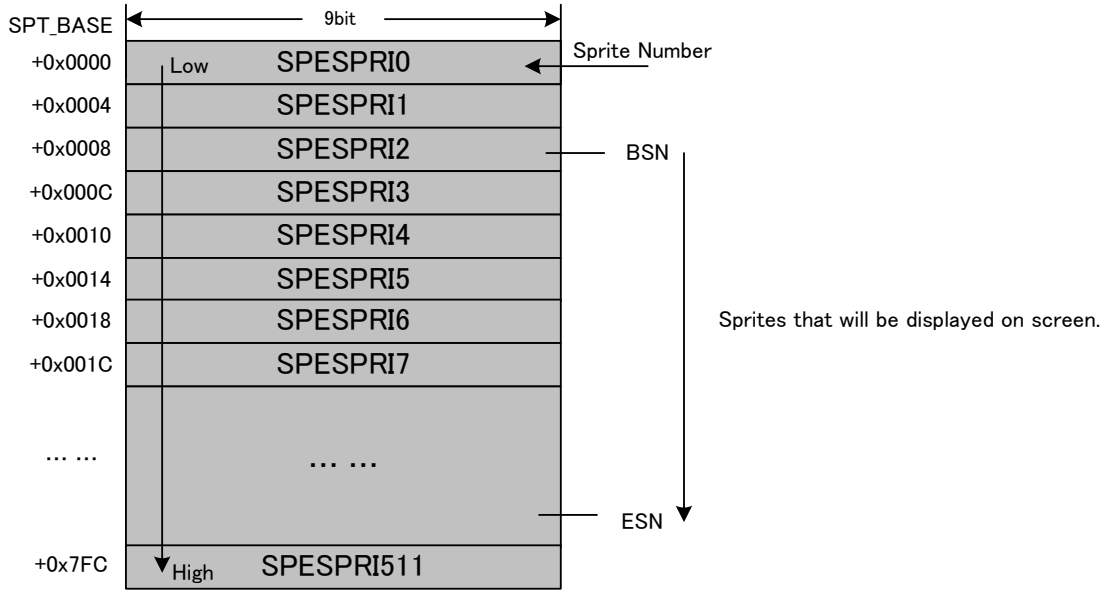


Figure 8-24 SPT Structure

8.4.4.12.1. Sprite Priority Register(0~511), SPESPRI(0~511)

Offset Address: 0x(n<<2),n=0~511; Initial value: none;

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

Bits	Name	Function
31:9	------	Reserved
8:0	SPN	Sprite number. (**0~511**)

8.4.4.13. SAT (Sprite Attribute Table)

Each sprite has its attribute register group. Therefore, there are 512 groups of sprite attribute registers to make up the sprite attribute table in the MB88F333.

Name	Width	Attribute
DY	12	The Y coordinate in display area
DX	12	The X coordinate in display area
PA	27	Address of a Pattern that the sprite uses (32bit word address) A bitshift of 2 bits is required, i.e. 0x60000 should be 0x18000 in the register.
SH	9	Y size of the source pattern
SW	7	X size of the source pattern
VR	1	Vertical reverse enable
HR	1	Horizontal reverse enable
CF	3	Color format select
CP	8	Color palette select
TE	1	Transparent enable
ABM	1	Alpha blend mode select
ATE	1	Alpha table enable
ATS	1	Alpha table select
AF	1	Alpha data format (only for alpha table data in the pattern memory)
AV	8	Alpha blend Value
DE	1	Sprite Display Enable

Table 8-3 Basic Sprite Attribute Table

Name	Width	Attribute
MY	8	Move pixels at Y direction
MX	8	Move pixels at X direction
YMV	3	Move method at Y direction
XMV	3	Move method at X direction
PAAL	1	Pattern address auto-load function enable
BLINK	1	Blink function enable
AMV	1	Auto-movement function enable
REPEAT	1	Action repeat
VCNT0	8	VSYNC counter0
VCNT1/CNT1	8	VSYNC counter1 or CNT1
CNT0	8	Animation function execute times

Table 8-4 Special Sprite Attribute Table

8.4.4.13.1. SDET (Sprite Display Enable Table)

Offset Address: 0x100 + 0x(n<<2), n=0~15; Initial value: 0x00000000;

Addr	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
+0x00	SPESDE0	DE[31: 0]
+0x04	SPESDE1	DE[63: 32]
+0x08	SPESDE2	DE[95: 64]
+0x0C	SPESDE3	DE[127: 96]
+0x10	SPESDE4	DE]159:128]
+0x14	SPESDE5	DE[191:160]
+0x18	SPESDE6	DE[223:192]
+0x1C	SPESDE7	DE[255:224]
+0x20	SPESDE8	DE[287:256]
+0x24	SPESDE9	DE[319:288]
+0x28	SPESDE10	DE[351:320]
+0x2C	SPESDE11	DE[383:352]
+0x30	SPESDE12	DE[415:384]
+0x34	SPESDE13	DE[447:416]
+0x38	SPESDE14	DE[479:448]
+0x3C	SPESDE15	DE[511:480]

*note:

1) These registers are “shadow register”.

2) There is a high speed clear function in SPE for configuring this table quickly by setting the register ***SPEDETC***.

8.4.4.13.2. Sprite Configuration Register0, SPES(0~511)CR0

Offset Address: 0x(n*3<<2),n=0~511; Initial value: none;

31	30	29	28	27	26	25	24
CF[2]	**CF[1]**	**CF[0]**	**AF**	**TE**	**PA[26]**	**PA[25]**	**PA[24]**
23	22	21	20	19	18	17	16
PA[23]	**PA[22]**	**PA[21]**	**PA[20]**	**PA[19]**	**PA[18]**	**PA[17]**	**PA[16]**
15	14	13	12	11	10	9	8
PA[15]	**PA[14]**	**PA[13]**	**PA[12]**	**PA[11]**	**PA[10]**	**PA[9]**	**PA[8]**
7	6	5	4	3	2	1	0
PA[7]	**PA[6]**	**PA[5]**	**PA[4]**	**PA[3]**	**PA[2]**	**PA[1]**	**PA[0]**

Bits	Name	Function
31:29	CF	Color Format 000 : 1bit 001 : 2bit 010 : 4bit 011 : 8bit 100 : 16bit(RGB565) 101 : 16bit(ARGB1555) 110 : 32bit(ARGB8888) 111 : Single Color Sprite, 24bit(RGB888)
28	AF	Alpha Data Format (only for alpha table in the pattern memory) 0: 4bit 1: 8bit
27	TE	Transparent Enable: 0: Disable. 1: Enable. The pixel with defined transparent color will not display
26:0	PA	Pattern Data Address (32bit word address) except the single color sprite format. And the ***PA[23:0]*** will be used as 24bit color when the sprite is a single color sprite.

8.4.4.13.3. Sprite Configuration Register1, SPES(0~511)CR1

Offset Address: 0x((n*3<<2)+4),n=0~511; Initial value: none;

31	30	29	28	27	26	25	24
CP[7]	**CP[6]**	**CP[5]**	**CP[4]**	**CP[3]**	**CP[2]**	**CP[1]**	**CP[0]**
23	22	21	20	19	18	17	16
AV[7]	**AV[6]**	**AV[5]**	**AV[4]**	**AV[3]**	**AV[2]**	**AV[1]**	**AV[0]**
15	14	13	12	11	10	9	8
SW[6]	**SW[5]**	**SW[4]**	**SW[3]**	**SW[2]**	**SW[1]**	**SW[0]**	**SH[8]**
7	6	5	4	3	2	1	0
SH[7]	**SH[6]**	**SH[5]**	**SH[4]**	**SH[3]**	**SH[2]**	**SH[1]**	**SH[0]**

Bits	Name	Function
31:24	CP	Color Palette Select (address of LUT indirect color mode)
23:16	AV	Alpha Value of the sprite
15:9	SW	Sprite Width (1~128)x4 pixels: **(SW + 1) x 4**
8:0	SH	Sprite Height (1~512) pixels: **(SH + 1)**

8.4.4.13.4. Sprite Configuration Register2, SPES(0~511)CR2

Offset Address: 0x((n*3<<2)+8),n=0~511; Initial value: none;

31	30	29	28	27	26	25	24
VR	0	0	0	**DY[11]**	**DY[10]**	**DY[9]**	**DY[8]**
23	22	21	20	19	18	17	16
DY[7]	**DY[6]**	**DY[5]**	**DY[4]**	**DY[3]**	**DY[2]**	**DY[1]**	**DY[0]**
15	14	13	12	11	10	9	8
HR	**ATE**	**ABM**	**ATS**	**DX[11]**	**DX[10]**	**DX[9]**	**DX[8]**
7	6	5	4	3	2	1	0
DX[7]	**DX[6]**	**DX[5]**	**DX[4]**	**DX[3]**	**DX[2]**	**DX[1]**	**DX[0]**

Bits	Name	Function
31	VR	Vertical Reverse
30:28	------	Reserved
27:16	DY	Display Y Coordinate (-2048~2047)
15	HR	Horizontal Reverse
14	ATE	Alpha Table Enable 0: Disable; Sprite alpha-blending mode 1: Enable; Pixel alpha-blending mode
13	ABM	Alpha Blend Mode 0 : With the Pre-layer already drawn 1 : With Single Color ***(SPEABC)***
12	ATS	Alpha Table Select 0: memory 1: LUT (only indirect color mode)
11:0	DX	Display X Coordinate (-2048~2047)

8.4.4.13.5. Sprite Configuration Register3, SPESS(0~31)CR3 (for Special Sprites)

Address: 0x200 + 0x(n*3<<2),n=0~31; Initial value: 0x00000000;

31	30	29	28	27	26	25	24
VCNT1[7]/ CNT1[7]	**VCNT1[6]/ CNT1[7]**	**VCNT1[5]/ CNT1[7]**	**VCNT1[4]/ CNT1[7]**	**VCNT1[3]/ CNT1[7]**	**VCNT1[2]/ CNT1[7]**	**VCNT1[1]/ CNT1[7]**	**VCNT1[0]/ CNT1[7]**
23	22	21	20	19	18	17	16
VCNT0[7]	**VCNT0[6]**	**VCNT0[5]**	**VCNT0[4]**	**VCNT0[3]**	**VCNT0[2]**	**VCNT0[1]**	**VCNT0[0]**
15	14	13	12	11	10	9	8
PAAL	**BLINK**	**AMV**	**REPEAT**	0	0	0	0
7	6	5	4	3	2	1	0
CNT0[7]	**CNT0[6]**	**CNT0[5]**	**CNT0[4]**	**CNT0[3]**	**CNT0[2]**	**CNT0[1]**	**CNT0[0]**

Bits	Name	Function
31:24	VCNT1/CNT1	These bits define different parameters for different animation modes: VSYNC1: VSYNC Counter1 in blink mode (1~256); CNT1: Defines the movement times in move function mode or defines the image switch times in image-switching mode (1~256);
23:16	VCNT0	VSYNC Counter0(1~256);
15	PAAL	Pattern address auto-load function enable; 1: Enable. 0: Not enable.
14	BLINK	Blink function enable; 1: Enable. 0: Not enable.
13	AMV	Auto-movement function enable; 1: Enable. 0: Not enable.
12	REPEAT	Auto-animation repeat always, yet do not refers the setting of ***CNT0***;
11:8	------	Reserved
7:0	CNT0	The times that auto-animation function will be executed (1~256) **TIMES = CNT0 + 1;**

8.4.4.13.6. Sprite Configuration Register4, SPESS(0~31)CR4 (for Special Sprites)

Address: 0x200 + 0x((n*3<<2)+4),n=0~31; Initial value: 0x00000000;

31	30	29	28	27	26	25	24
YMV[2]	**YMV[1]**	**YMV[0]**	0	0	0	0	0
23	22	21	20	19	18	17	16
MY[7]	**MY[6]**	**MY[5]**	**MY[4]**	**MY[3]**	**MY[2]**	**MY[1]**	**MY[0]**
15	14	13	12	11	10	9	8
XMV[2]	**XMV[1]**	**XMV[0]**	0	0	0	0	0
7	6	5	4	3	2	1	0
MX[7]	**MX[6]**	**MX[5]**	**MX[4]**	**MX[3]**	**MX[2]**	**MX[1]**	**MX[0]**

Bits	Name	Function
31:29	YMV	The movement method at Y direction 000 : keep still at Y direction 001 : up 010 : down 011 : reserved 100 : up → down → up 101 : up → return → up 110 : down → up → down 111 : down → return → down
28:24	------	Reserved
23:16	MY	The pixels that the special sprite will move at Y direction(1~256pixels)
15:13	XMV	The movement method at X direction 000 : keep still at X direction 001 : left 010 : right 011 : reserved 100 : left → right → left 101 : left → return → left 110 : right → left → right 111 : right → return → right
12:8	------	Reserved
7:0	MX	The pixels that the special sprite will move at X direction(1~256pixels)

*note: However the auto-movement mode that a special is set, do not let the sprite cross the sprite display area, both X(-2048~2047) and Y(-2048-2047).

8.4.4.13.7. Sprite Configuration Register5, SPESS(0~31)CR5 (for Special Sprites)

Address: 0x200 + 0x((n*3<<2)+8),n=0~31; Initial value: 0x00000000;

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

Bits	Name	Function
31:27	------	Reserved
26:0	ADTA	Pattern data address table base address. (32bit word address)

8.4.4.13.8. Special Sprite Number Specify Register, SPESSN(0~31) (for Special Sprites)

Address: 0x400 + 0x(n<<2),n=0~31; Initial value: 0x00000000;

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

Bits	Name	Function
31:9	------	Reserved
8:0	SPN	Sprite number. (**0~511**) Specify which sprite is the special sprite N(0~31).

*note:

1) These registers can not be set to the same value.
2) There registers must be set before using the special sprite.

8.4.4.14. LUT (Look-up Table)

Use the LUT registers for writing and reading alpha/color data from LUT. The LUT registers are named as SPELUTS(0~511) and mapped to address 0x000 to 0x07FC.

8.4.4.14.1. LUT Setting Register(0~511), SPELUTS(0~511)

Offset Address: 0x(n<<2),n=0~511; Initial value: none;

31	30	29	28	27	26	25	24
AVAL[7]	**AVAL[6]**	**AVAL[5]**	**AVAL[4]**	**AVAL[3]**	**AVAL[2]**	**AVAL[1]**	**AVAL[0]**
23	22	21	20	19	18	17	16
RVAL[7]	**RVAL[6]**	**RVAL[5]**	**RVAL[4]**	**RVAL[3]**	**RVAL[2]**	**RVAL[1]**	**RVAL[0]**
15	14	13	12	11	10	9	8
GVAL[7]	**GVAL[6]**	**GVAL[5]**	**GVAL[4]**	**GVAL[3]**	**GVAL[2]**	**GVAL[1]**	**GVAL[0]**
7	6	5	4	3	2	1	0
BVAL[7]	**BVAL[6]**	**BVAL[5]**	**BVAL[4]**	**BVAL[3]**	**BVAL[2]**	**BVAL[1]**	**BVAL[0]**

Bits	Name	Function
31:24	AVAL	Alpha value definition
23:16	RVAL	Red color definition
15:8	GVAL	Green color definition
7:0	BVAL	Blue color definition

9. Video Display Controller

This chapter describes the Video Display Controller of the MB88F333 device.

9.1. Outline

The Display Controller Module (VDC) outputs RGB digital data to the display.
RGB digital data is output from the sprite engine.
There are four display layers (LA, LB, background, foreground).

9.2. Features

The Display Controller has the features described in the following section.

9.2.1. Features

- Display output
  RGB (24bit/pixel) digital image output.
- Scanning method
  progressive scan.
- Image size
  Possible image size s are: (examples)
  320×120
  320×160
  320×240 (QVGA)
  400×240 (WQVGA)
  480×240
  500×250
  640×160
  640×240
  640×480 (VGA)
  800×480 (WVGA)
  960×160
  1280×480@60Hz (max. 42MHz pixel clock)
- There are four display layers (LA, LB, background, foreground).
- This module can set a display layer background and foreground color.
- Display layer LA is a screen display layer from the sprite engine.
- Display layer LB is a screen display layer from an external source.
- Display Layers LA, LB and foreground can use a transparency color.
- Display Layers LA, LB and foreground can use a blend function.
- Display Layers LA and LB can switch overlapping.

9.2.2. Limitations

- Interlace scanning is not possible.
- The size of the display layer determines the set image size.

9.2.3. Performance Requirements

- The Sprite Engine (line buffer) must prepare the display data of the next raster before the Video Display Controller (VDC) finishes the processing of a raster
- After five cycles, the VDC outputs the RGB data of one pixel received from the Sprite Engine (line buffer).
- The maximum Bus CLK value is 96MHz
- The maximum Pixel CLK value is 42MHz. The 42 MHz limit is due to the APIX and TCON units of Indigo-L.
- For further information about the Sprite Engine performance, please refer to the corresponding application note on the GDC website (http://www.fujitsu.com/emea/services/microelectronics/gdc/gdcdevices/MB88F333-indigo.html#support)

9.3. Function

9.3.1. Block diagram

Figure 9-1 shows Display Controller's block diagram.

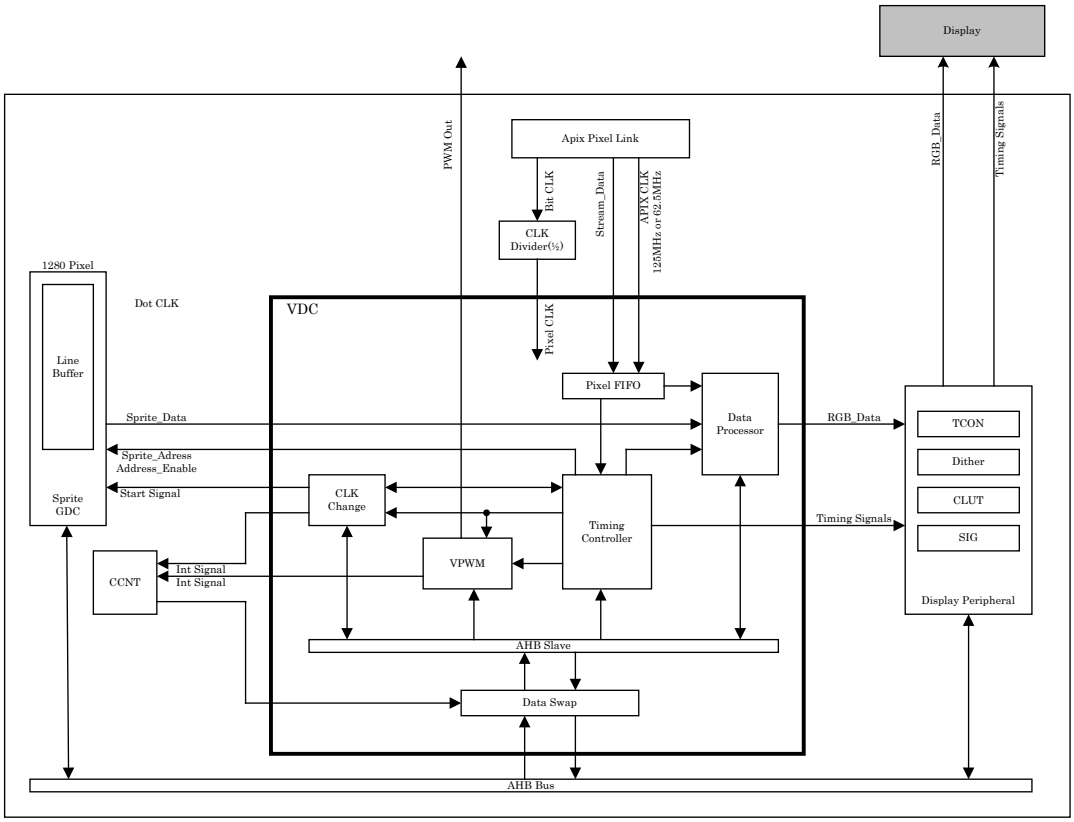


Figure 9-1 Display Controller block diagram

9.3.1.1. Overview of each block

- AHB Slave
  An interface with the AHB. It controls the register and operates with the AHB clock.
- Pixel FIFO
  Please refer to Pixel FIFO Module.
- Timing Controller
  Controls the timing of the control signal and operates with the Pixel clock.
- Data Processor
  Generates RGB data and operates with the Pixel clock.
- CLK Change
  Changes from Pixel CLK to AHB CLK and from AHB CLK to Pixel CLK.
- VPWM
  Generates the LCD backlight adjustment signal and operates with the Pixel clock.
  PWM[15:12] are available..
- Data Swap
  Please refer to Data Swap Control (21.3.7).

9.3.2. Display Parameters

A display area is defined using the parameters shown below. Each parameter is set as a register value.

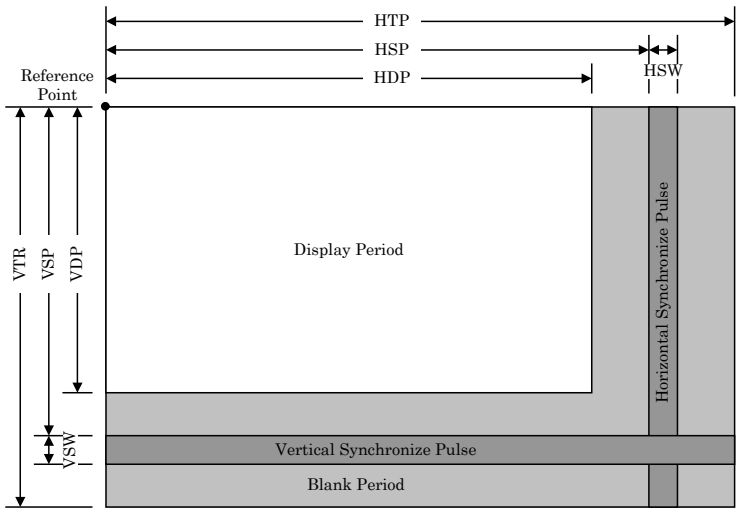


Figure 9-2 Display Parameters

HTP	Horizontal Total Pixels	Horizontal Display Period + Horizontal Blank Period
HSP	Horizontal Synchronize pulse Position	Period from Reference Point to Horizontal Synchronize Pulse
HSW	Horizontal Synchronize pulse Width	
HDP	Horizontal Display Period	
VTR	Vertical Total Raster	Vertical Display Period + Vertical Blank Period
VSP	Vertical Synchronize pulse Position	Period from Reference Point to Vertical Synchronize Pulse
VSW	Vertical Synchronize pulse Width	
VDP	Vertical Display Period	

NOTE: The display parameter value is determined from the register setting value **+1.**

The values set must be in accordance with the following relationships. If this rule is disregarded, the correct operation of the system can not be guaranteed.

$0 < HDP < HSP < HSP + HSW < HTP$, $HSP+6 < HTP$

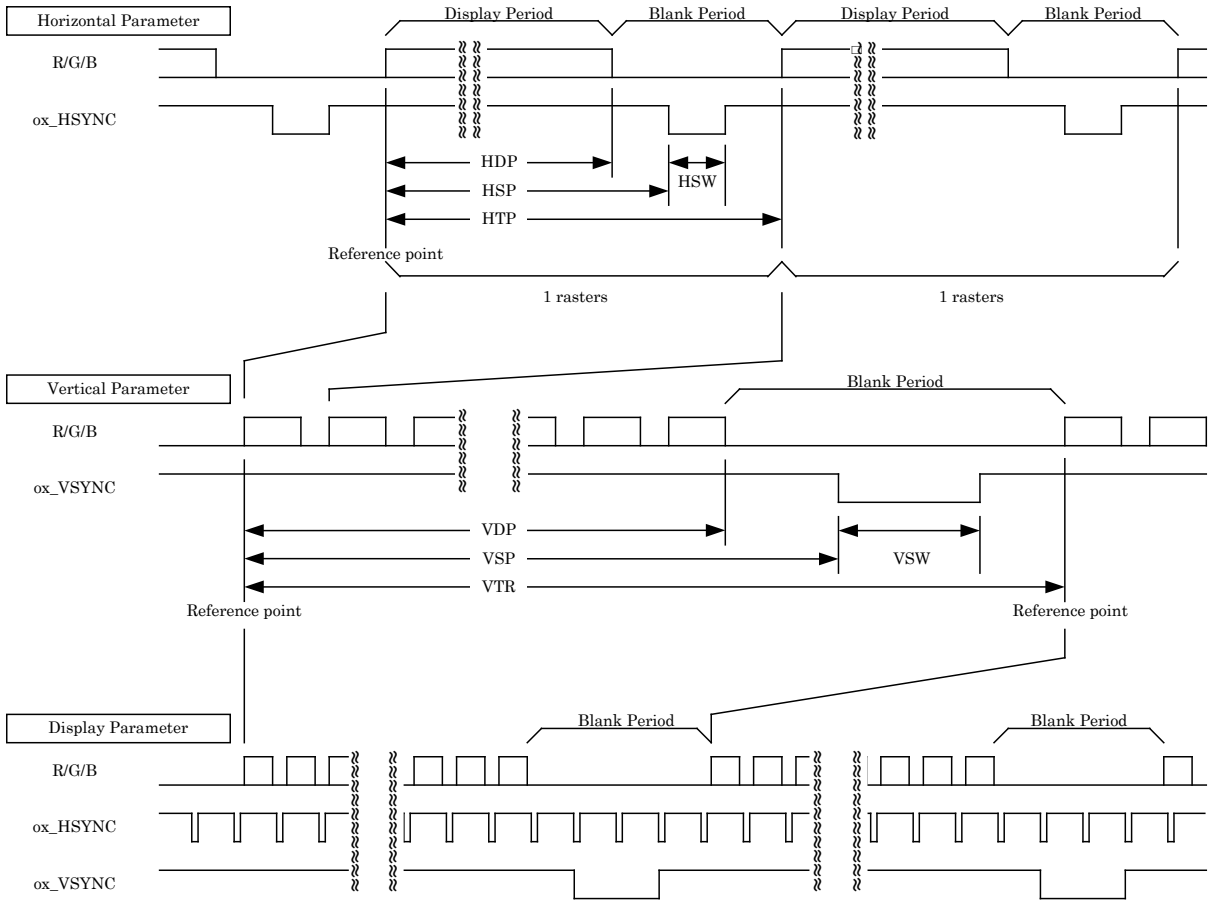
$0 < VDP < VSP < VSP + VSW < VTR$

Please follow the following limitations when you use only the external synchronization.

$0 < HDP < HSP < HSP + HSW < HTP$

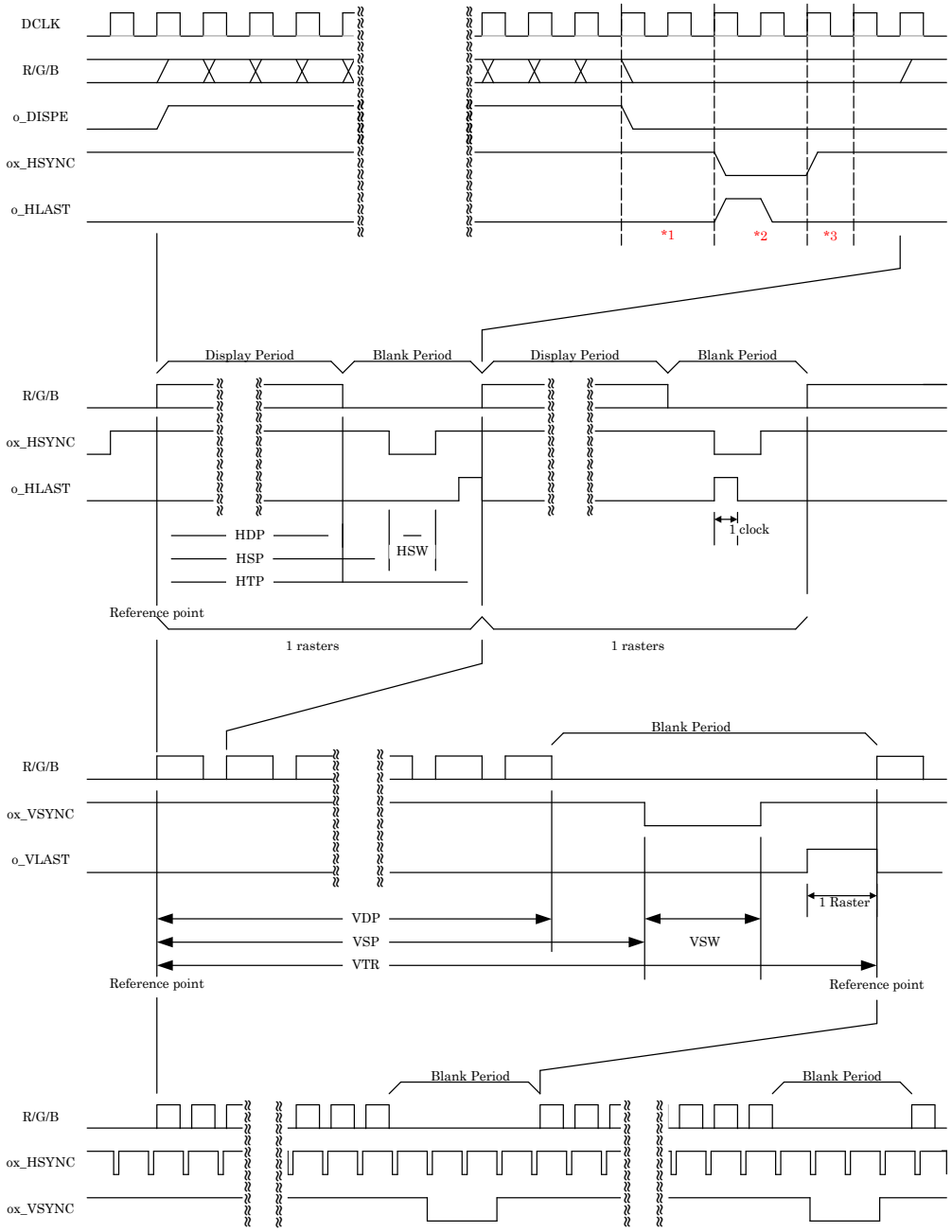
$0 < VDP < VSP < VSP + VSW < VTR$

9.3.2.1. Display Parameter details



R/G/B	RGB data displayed on the display. Uses o_RDATA, o_GDATA, and o_BDATA.
ox_HSYNC	Horizontal Synchronize Pulse. Signal to indicate the horizontal timing of the display output. Scanning lines that start from the left to the right are returned leftmost.
ox_VSYNC	Vertical Synchronize Pulse. It is a signal to show vertical timing to the display.

9.3.3. Display one frame composition



*1 : Cycle of the clock frequency changes by a set value of HSP.
*2 : Cycle of the clock frequency changes by a set value of HSW.
*3 : Cycle of the clock frequency changes by a set value of HTP.

9.3.4. Supported Displays

The following table lists the typical resolutions of displays that can be used and the frequency of their synchronization signals. The pixel clock frequency is a clock derived from the external.
Resolution Setting (9.4.3 Register list)

Table 9-1 Resolution and Display Frequency

Resolution	Pixel frequency	Horizontal total pixel count	Horizontal frequency	Vertical total raster count	Vertical frequency
320×120	3.14MHz	400	7.85 kHz	131	59.9Hz
320×160	4.19MHz	400	10.48 kHz	175	59.9Hz
320×240	6.31MHz	400	15.78 kHz	263	60.0Hz
400×240	7.91MHz	500	15.82 kHz	263	60.1Hz
640×160	8.45MHz	800	10.56 kHz	175	60.4Hz
480×240	9.47MHz	600	15.78 kHz	263	60.0Hz
500×250	10.3MHz	630	16.40 kHz	273	60.0Hz
960×160	12.5MHz	1200	10.41 kHz	175	59.5Hz
640×240	12.5MHz	800	15.62 kHz	263	59.4Hz
640×480	25.0MHz	800	31.25 kHz	525	59.5Hz
800×480	31.3MHz	1000	31.30 kHz	525	59.6Hz
1280×480	41.6MHz	1300	32.00 kHz	525	60.9Hz

Note : This table is one example.

9.3.5. Internal synchronization

This module generates timing signals and displays sprites when external synchronous mode is not used . The timing depends on the setting of the display parameter values register.

9.3.5.1. Horizontal Display Parameter State Machine

The horizontal display state machine is shown below.

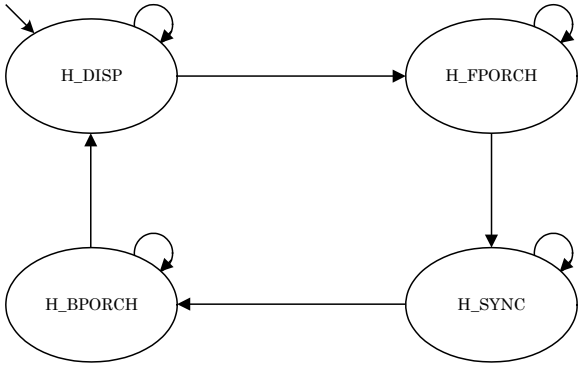


Figure 9.3-3 Horizontal Display Parameter State Machine

State	Next State	Transition condition
H_DISP (Initial)	H_FPROCH	Horizontal pixel counter reaches HDP.
H_FPROCH	H_SYNC	Horizontal pixel counter reaches HSP.
H_SYNC	H_BPROCH	Horizontal synchronization pulse counter reaches HSW.
H_BPROCH	H_DISP	Horizontal pixel counter reaches HTP.

The unit is controlled by a horizontal pixel counter and a horizontal synchronization pulse counter.
The horizontal pixel counter operates constantly.
When the counter value reaches HTP, it is reset.

The horizontal synchronization pulse counter operates in the H_SYNC state.
When the counter value reaches HSW, it is reset.

When external synchronization is used, the corresponding state machine is applied.
Please refer to 9.3.6 External Synchronization for further information about external synchronization.

9.3.5.2. Vertical Display Parameter State Machine

The vertical display state machine is shown below.

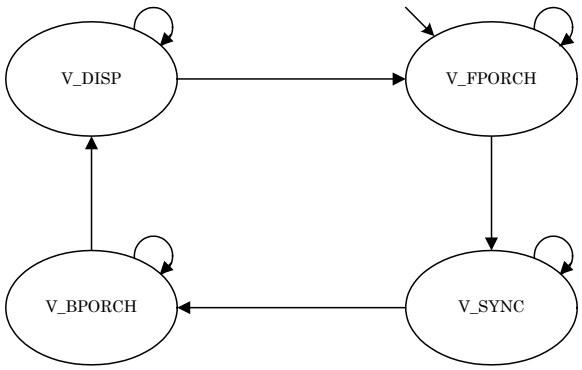


Figure 9.3-4 Vertical Display Parameter State Machine

State	Next State	Transition condition
V_DISP	V_FPROCH	Vertical pixel counter reaches VDP.
V_FPROCH (Initial)	V_SYNC	Vertical pixel counter reaches VSP.
V_SYNC	V_BPROCH	Vertical synchronization pulse counter reaches VSW.
V_BPROCH	V_DISP	Vertical pixel counter reaches VTR.

The unit is controlled by a vertical pixel counter and a vertical synchronization pulse counter.
The vertical pixel counter operates constantly.
When the counter value reaches VTR, it is reset.

The vertical synchronization pulse counter operates in the V_SYNC state.
When the counter value reaches VSW, it is reset.

When external synchronization is used, the corresponding state machine is applied.
Please refer to 9.3.6 External Synchronization for further information about external synchronization.

9.3.6. External Synchronization

Display output can be performed in synchronization with externally generated horizontal/vertical synchronization signals when the VDC is operating in external synchronization mode.
External synchronization can be monitored using the status register. (9.4.4.8 Status)

9.3.6.1. External Synchronization Detection

External synchronization mode is selected using a register. Output display then uses a forward alignment guard time and a backward alignment guard time for vertical synchronization and synchronizes with the external video signal. The frequency of the forward alignment guard time and backward alignment guard time can be set using registers (9.4.4.2 DCMx) (9.4.4.20 AGT).

Note : When external synchronization display mode is used, one frame may display incorrectly while internal and external units align.

9.3.6.2. External Synchronization 'Undetection'

External Synchronization 'undetection' describes the following conditions:

- When an external vertical synchronizing signal cannot be detected by the forward alignment guard time. Interrupt signal output (o_EXT_VS).
- When the pixel FIFO enters an underflow state.
  Interrupt signal output (o_PFUF).
- When either of the conditions described above is generated, a synchronization error (o_SYNC_ERR) is output.

The VDC switches to internal automatic operation if the external synchronization 'undetection' condition occurs. The VDC monitors for external synchronzation signals again.
Note : o_SYNC_ERR is not an interrupt signal. It is a control signal to the Command Sequencer unit.

9.3.6.3. External Synchronization State Machine

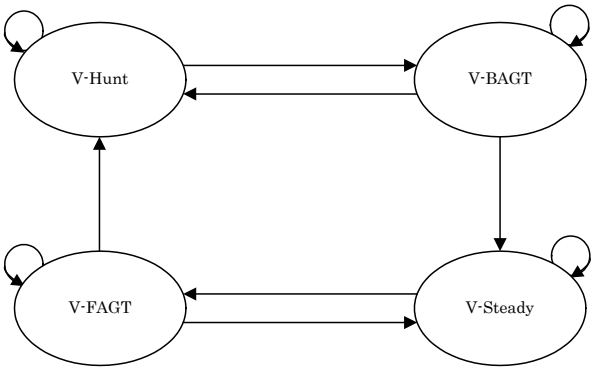


Figure 9.3-5 External Synchronization State Machine

- V-Hunt (Vertical Hunting)
  External vertical synchronization signal detection
- VBAGT (Backward Alignment Guard Time)
  External vertical synchronization signal backward alignment guard time
- V-Steady (Vertical Synchronization Steady)
  External synchronization Steady state
- V-FAGT (Vertical Forward Alignment Guard Time)
  External vertical synchronization signal forward alignment guard time

State	Next State	Transition condition
V-Hunt (Initial)	V-BAGT	External Vertical Synchronization Signal Detection
V-BAGT	V-Steady	Backward Alignment Guard Time Detection
	V-Hunt	No Backward Alignment Guard Time Detection
V-Steady	V-FAGT	No External Vertical Synchronization Signal Detection
V-FAGT	V-Steady	Forward Alignment Guard Time Detection
	V-Hunt	No Forward Alignment Guard Time Detection

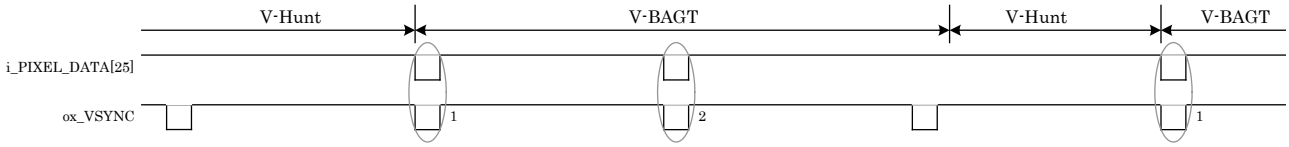
9.3.6.4. External Synchronization Alignment Guard Time

- Backward Alignment Guard Time

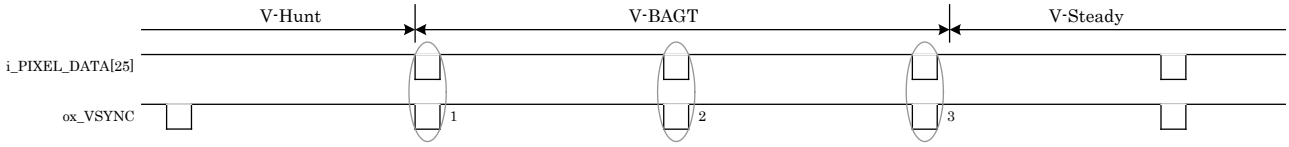
Backward Alignment Guard Time is required confirm whether external synchronization is used in a number of consecutive frames. The object of the Backward Alignment Guard Time is avoid the mistaken use of a momentary external vertical synchronization signal for external synchronization mode.

The operation of the Backward Alignment Guard Time is shown below.
The number of Backward Alignment Guard Time steps is assumed to be three steps.

When external synchronization is not used



When external synchronization is used



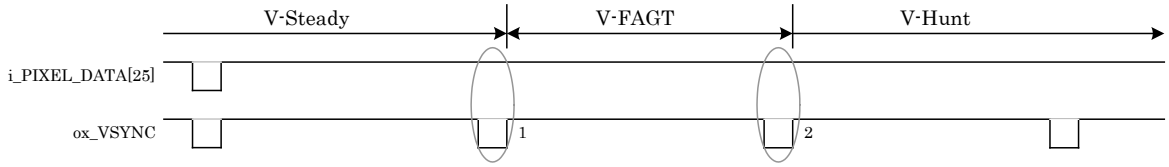
i_PIXEL_DATA[25] : External Vertical Synchronize Pulse input
ox_VSYNC : Vertical Synchronize Pulse output

- Forward Alignment Guard Time

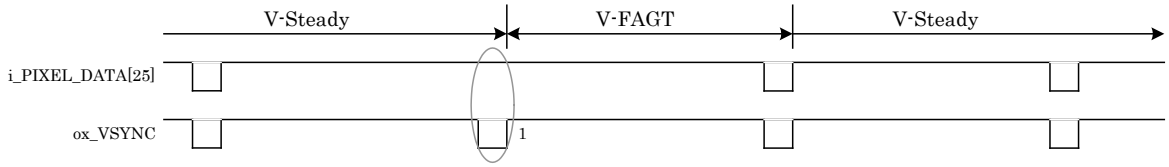
Forward Alignment Guard Time is required to determine whether synchronization should not be used in a number of consecutive frames. The object is to avoid the mistaken use of a momentary external vertical synchronization signal for external synchronization mode and to prevent an external synchronization 'undetection' condition.

The operation of Forward Alignment Guard Time is shown below.
The number of Forward Alignment Guard Time steps is assumed to be two steps.

External Synchronization Undetection



When an external synchronous error is only one frame long



i_PIXEL_DATA[25] : External Vertical Synchronize Pulse input
ox_VSYNC : Vertical Synchronize Pulse output

9.3.6.5. External Horizontal Display Parameter State Machine

The horizontal display state machine is shown below.

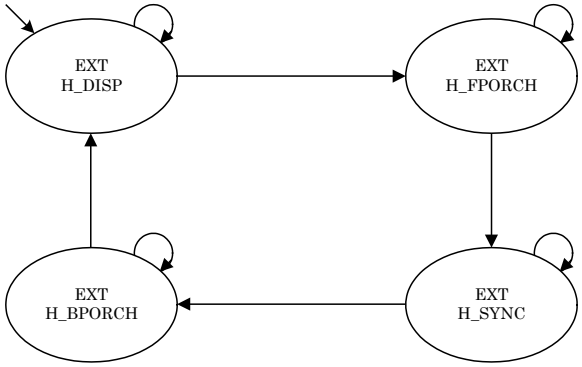


Figure 9.3-6 External Horizontal State Machine

State	Next State	Transition condition
EXT H_DISP (Initial)	EXT H_FPORCH	Horizontal pixel counter for external reaches HDP.
EXT H_FPORCH	EXT H_SYNC	Horizontal synchronize pulse position counter for external reaches 255.
		External horizontal synchronizing signal falling edge detection
EXT H_SYNC	EXT H_BPORCH	Horizontal synchronization pulse counter for external reaches HSW.
EXT H_BPORCH	EXT H_DISP	Horizontal pixel counter for external reaches HTP.

The unit is controlled by a horizontal pixel (external) counter, a horizontal synchronize pulse position (external) counter and a horizontal synchronization pulse counter (external).

The external horizontal pixel counter stops operating in the EXT H_FPORCH state.
When changing from EXT H_FPORCH state to EXT H_SYNC state, The external horizontal pixel counter loads HSP.
The external horizontal pixel counter start countup in the EXT H_SYNC state again.
When the counter value reaches HTP, it is reset.

The horizontal synchronization pulse position (external) counter works in the EXT H_FPORCH state.
If an external horizontal synchronizing signal is detected or the counter value reaches 255, this counter is cleared.

The horizontal synchronization pulse counter (external) works in the EXT H_SYNC state.
If the counter value reaches HSW, it is reset.

9.3.6.6. External Vertical Display Parameter State Machine

The vertical display state machine is shown below.

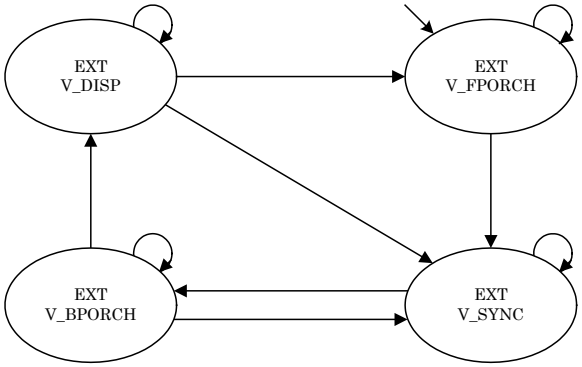


Figure 9.3-7 External Vertical State Machine

State	Next State	Transition condition
EXT V_DISP	EXT V_FPORCH	Vertical pixel counter for external reaches VDP.
	EXT V_SYNC	The external synchronous state machine changes from V-Hunt to V-BAGT.
EXT V_FPORCH (Initial)	EXT V_SYNC	Vertical synchronize pulse position counter for external reaches 255.
		External vertical synchronizing signal falling edge detection
EXT V_SYNC	EXT V_BPORCH	Vertical synchronization pulse for external counter reaches VSW.
EXT V_BPORCH	EXT V_DISP	Vertical pixel counter for external reaches VTR.
	EXT V_SYNC	The external synchronous state machine changes from V-Hunt to V-BAGT.

The unit is controlled by an external vertical pixel counter, external vertical synchronize pulse position counter and an external vertical synchronization pulse counter.

The external vertical pixel counter stops operation in the EXT V_FPORCH state.
When changing from EXT V_FPORCH state to EXT V_SYNC state, The external vertical pixel counter loads VSP.
The external vertical pixel counter start countup in the EXT V_SYNC state again.
When the counter value reaches VTR, it is cleared.

The external vertical synchronization pulse position counter operates in the EXT V_FPORCH state.
When an external vertical synchronization signal is detected, or the counter value reaches 255, this counter is cleared.

The external vertical synchronization pulse counter operates in the EXT V_SYNC state.
When the counter value reaches VSW, it is cleared.

9.3.6.7. External Synchronization Display Parameters

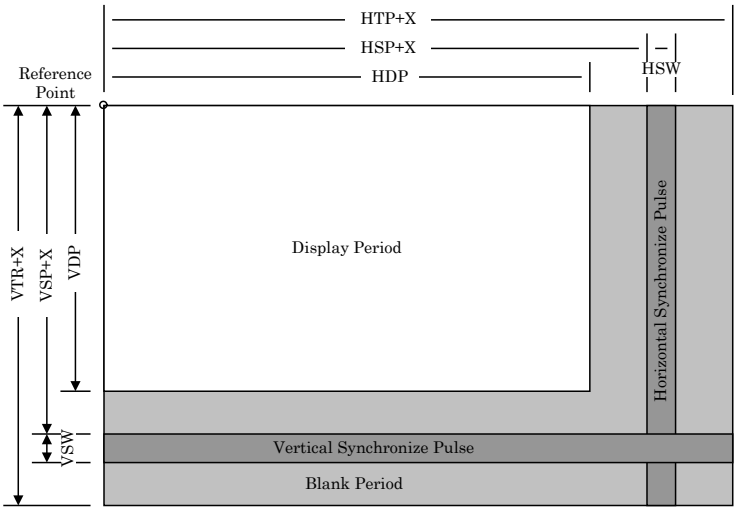


Figure 9.3-8 External Synchronization Display Parameters

The external horizontal pixel counter stops operation at horizontal state H_FPORCH (9.3.6.5 External Horizontal Display Parameter State Machine). Then it aligns to the external synchronous idle signal. For this reason, HTP could be different from the HTP register setting value.
The HSP period changes according to whether external synchronization was detected.

The external vertical pixel counter stops operation at vertical state V_FPORCH (9.3.6.6 External Vertical Display Parameter State Machine). Then it aligns to the external synchronous idle signal. For this reason, VTR could be different from the VTR register setting value.
The VSP period changes according to whether external synchronization was detected.

9.3.6.8. External RGB Data Evaluation

The VDC has the following output signals: o_RDATA, o_GDATA, and o_BDATA.
The o_GV signal is used to determine whether these signals are external RGB data or internal RGB data.

- o_GV outputs '0' during external RGB data output.
- o_GV outputs '1' during sprite RGB data output or blending RGB data output.
- o_GV outputs ‘0’ during the blank period.

9.3.7. Display Layers

The display layer structure is shown below:

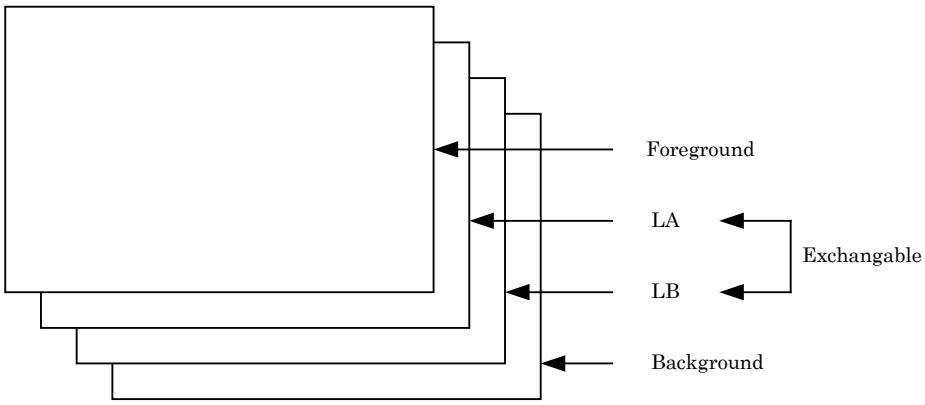


Figure 9-9 Display Layer structure

Foreground	The foreground is a top priority display layer. The foreground displays only one color. The color is set by this module.(9.4.4.17 DFGC)
LA	LA is a display layer used to display sprite data.
LB	LB is a display layer used to display external stream data.
Background	The background is a display layer of lowest priority at the bottom of the stack. The background displays only one color. This should not be a transparent color. The color is set by this module.(9.4.4.18 DBGC)

Image data for the four layers (LA, LB, Foreground, Background) is processed as follows.

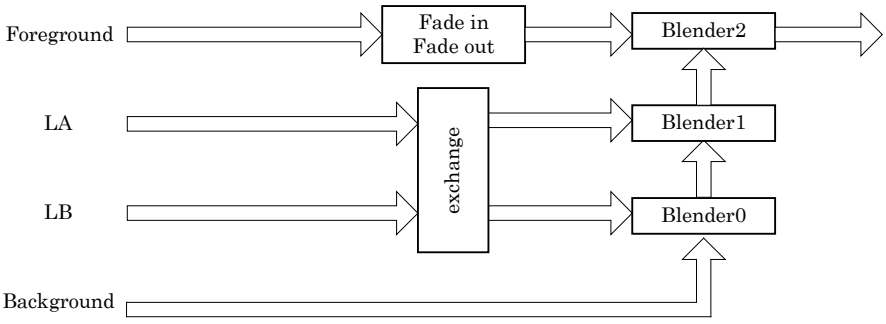


Figure 9-10 Display Layer processing

Exchange : The exchange unit swaps an upper image and a lower image.
(9.4.4.11 DLS)

Blender0 : Blender0 does transparency processing with Background by setting the register of an upper layer. It doesn't do blending.

Blender1 : Blender1 executes blending and the transparency processing by setting the register of an upper layer. When the lower layer is background data, blending can be selected. (9.4.4.18 DBGC)

Blender2 : Blender2 does blending by setting the register of a Foreground. It doesn't process transparency.

9.3.7.1. Replacement Mode (Exchange)

In this mode, the stacking sequence of layers can be changed arbitrarily.
This can be configured by a register (9.4.4.11 DLS).

9.3.7.2. Blend Mode (Blender x)

The unit does blending using a blend factor defined for each layer, or it stacks layers in accordance with the transparent color definition.
These can be set by the register. (9.4.3 Register summary)

In blend mode, the following calculations are made using the defined blend ratio 'r'.
$D_{view} = D_{front} \times r + D_{back} \times (1 - r)$
r : Blend ratio (the blending ratio can be set via the blend register)
D_{view} : View Layer
D_{front} : Upper Layer
D_{back} : Lower Layer

9.3.7.3. Transparent Mode (Blender x)

A transparent color is defined for each layer. If a front layer uses a transparent color, the back layer is displayed (i.e. it is a view layer). If a front layer does not use a transparent color, it is displayed.
This can be set by a register. (9.4.3 Register summary)
$D_{view} = D_{front}$ (if a front layer is not in a transparent color)
$= D_{back}$ (if a front layer is in a transparent color)

9.3.7.4. Fade in / Fade out Mode

Fade in / fade out mode can do an auto-adjustment of the blend ratio of the foreground color. The blend ratio is changed for the period of a vertical blank. The change value is constant. The change value can be set by the register.(9.4.4.19 FADEC)

$R_{fore} = R_{fore} + n$
$= R_{fore} - n$

R_{fore} : Foreground Blend Ratio
n : change value

If the blend ratio is added and an overflow occurs, the blend ratio becomes ‘1’.
If the blend ratio is subtracted and an underflow occurs, the blend ratio becomes ‘0’.

When the end condition is reached, '1' is written in the FEND bit of FADEC Register.
The end condition is defined as follows:
1, When the internal counter reaches the VCNT value set.
2, When R_{fore} overflows
3, When R_{fore} underflows
VCNT is a register setting value (9.4.4.19 FADEC).

9.3.8. Display Method

Display output is handled as a progressive scan.
An example of the screen handling is shown below using VGA.

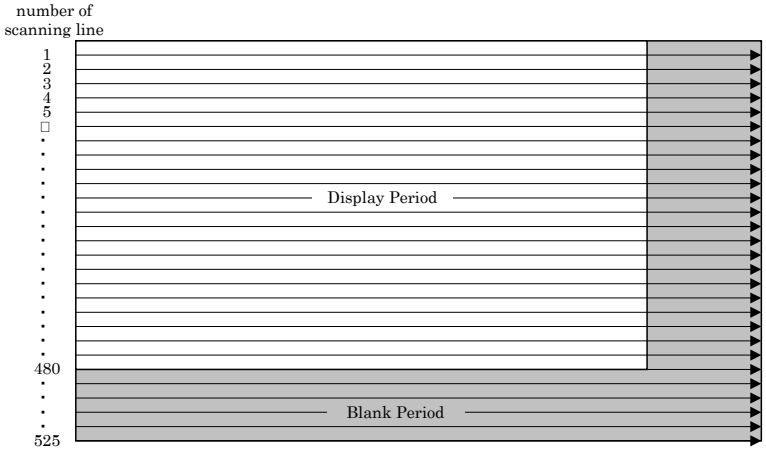


Figure 9-11 Progressive scan

A progressive scan executes sequential horizontal scanning from the reference point to the ending point.
One scanning line is one raster.

9.3.9. Interrupts

The VDC has seven interrupt signals:
o_EXT_VS, o_EXT_HS, o_PFOF, o_PFUF, o_INT_VSYNC, o_INT_HSYNC, o_INT_VPWM

9.3.9.1. External Vertical Synchronization Error Interrupt

If the vertical synchronization can not be detected in an external synchronization mode, o_EXT_VS is generated.
Reasons for non-detection of external vertical synchronization are as follows:

- Incorrect vertical display parameter settings
- Malfunctional external synchronization signal

Control of the o_EXT_VS output signal is achieved by the EVIE bit of 0 Interrupt_Enable.

9.3.9.2. External Horizontal Synchronization Error Interrupt

If the horizontal synchronization can not be detected in an external synchronization mode, o_EXT_HS is generated.
Reasons for non-detection of external horizontal synchronization are as follows:

- Incorrect horizontal display parameter settings
- Malfunctional external synchronization signal

Control of the o_EXT_HS output signal is achived by the EHIE bit of 0 Interrupt_Enable.

9.3.9.3. Pixel FIFO Underflow Interrupt

If the pixel FIFO underflows, o_PFUF is generated.
Control of the o_PFUF output signal is achived by the PFUIE bit of 0 Interrupt_Enable.
Please refer to the Pixel FIFO Module.

9.3.9.4. Pixel FIFO Overflow Interrupt

If the pixel FIFO underflows, o_PFOF is generated.
Control of the o_PFUF output signal is achived by the PFUIE bit of 0 Interrupt_Enable.

Please refer to the Pixel FIFO Module.

9.3.9.5. One Frame End Interrupt

IWhen the display output of a single frame ends, o_INT_VSYNC is generated. The output timing is as follows. The output control of o_INT_VSYNC can be done via the VSIE bit of 0 Interrupt_Enable.

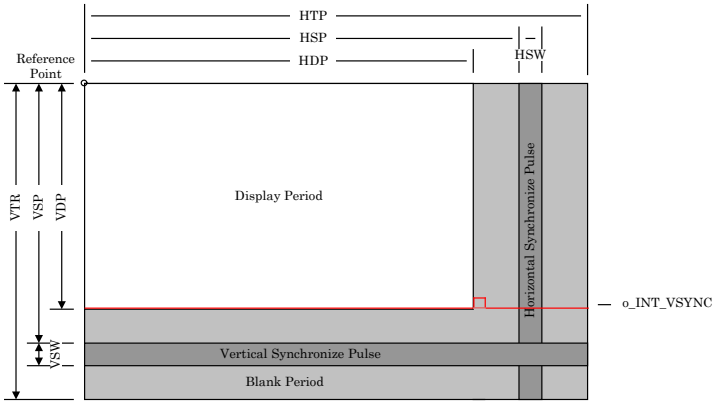


Figure 9.3-12 Output timing of o_INT_VSYNC

o_INT_VSYNC is generated in synchronization with the pixel clock. It then outputs it and then transfers to the AHB clock. Therefore, the output timing is somewhat different from Figure 9.3-12.

9.3.9.6. One Line End Interrupt

When the display output of a single line ends, o_INT_HSYNC is generated.
The output control of o_INT_HSYNC can be done via the HSIE bit of 0 Interrupt_Enable.

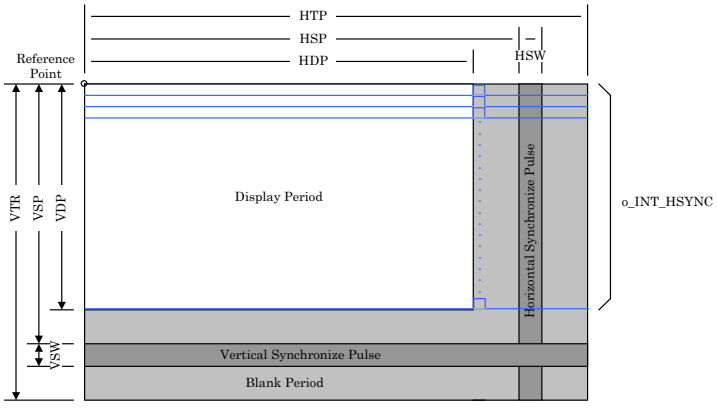


Figure 9.3-13 Output timing of o_INT_HSYNC

o_INT_HSYNC is generated in sync with the pixel clock. It then outputs it and then transfers to the AHB clock. Therefore, the output timing is somewhat different from Figure 9.3-13.

9.3.9.7. VPWM Interrupt

When the internal count value of VPWMx reaches the VPWMxE bit of the VPWMxSE register, o_VPWM_INT is generated.
o_VPWM_INT is a logical OR of the interrupt signal of the four VPWMs built into the VDC.
The VPxIE bit of 0 Interrupt_Enable controls the output control of o_VPWM_INT.
The Status register (9.4.4.8 Status) is used to check which PWM outputs the interrupt.

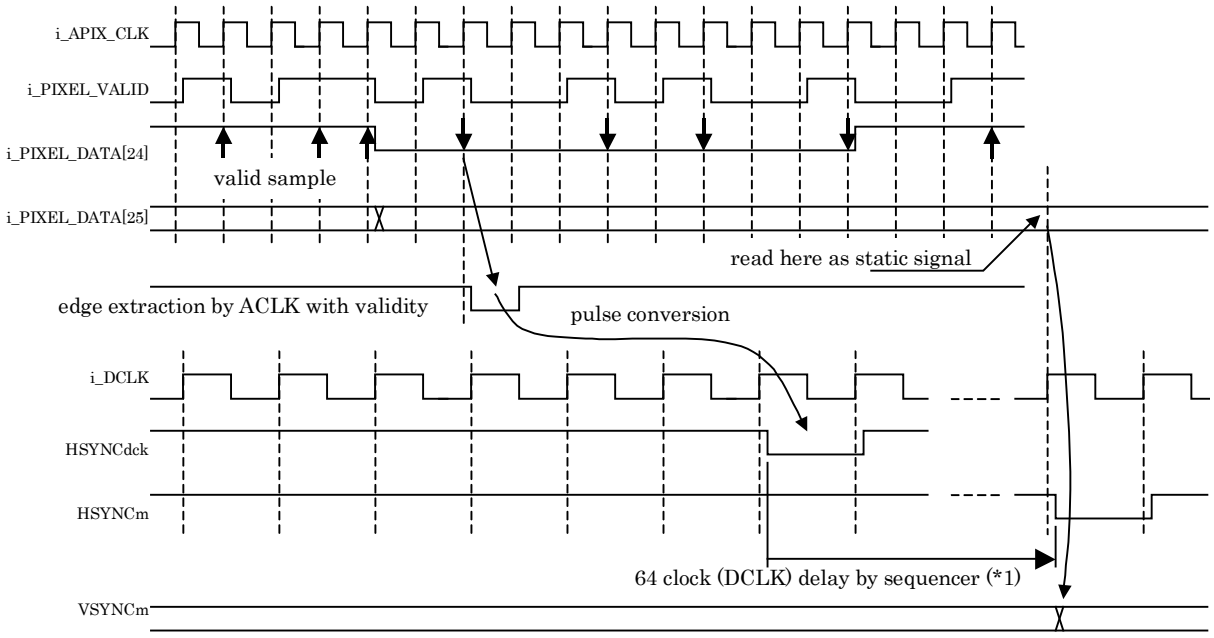
9.3.10. Pixel FIFO

The pixel FIFO is a Stream Data Buffer used to maintain a constant data output speed.
External data is supplied in a stable flow to the Timing Controller (9.3.1 Block diagram).

9.3.10.1. Sync Control

The sync control unit has two purposes:

- Resampling with the pixel clock.
- To give HSYNC a 64-clock delay to start the FIFO with 64 words of data each raster.
  64 words is half of the FIFO capacity in order to absorb timing lag and lead.

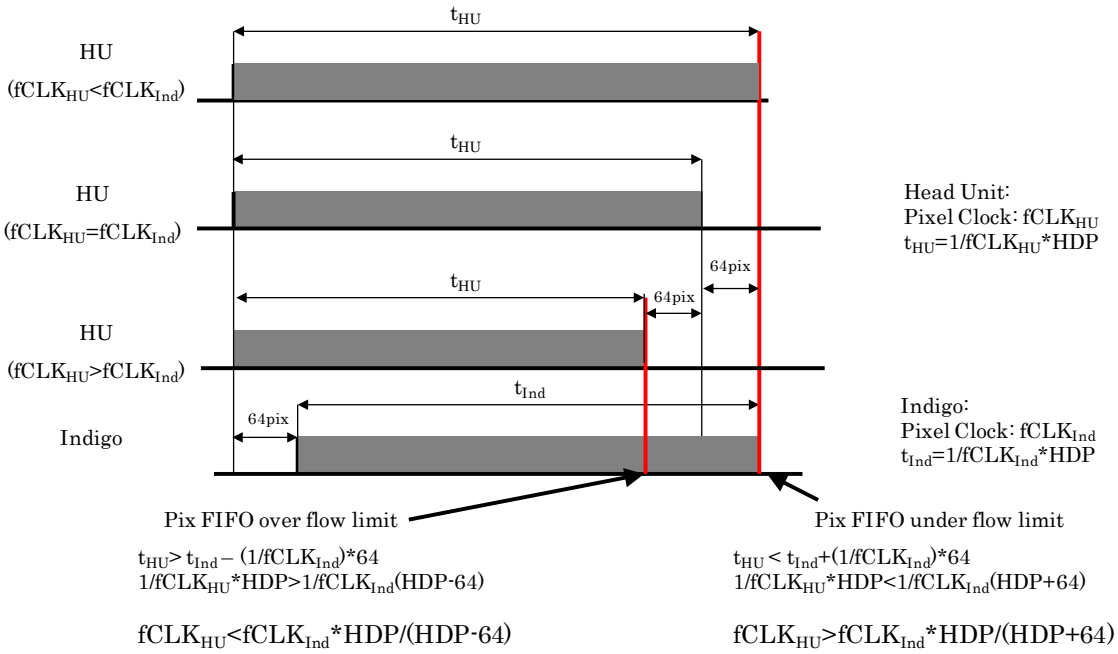


Note : This delay value is programmable by Hdelay input but 64 is a typical value.

i_APIX_CLK	APIX Clock (125Mhz or 62.5MHz)
i_PIXEL_VALID	i_PIXEL_DATA Enable signal
i_PIXEL_DATA[24]	External Horizontal Synchronization signal
i_PIXEL_DATA[25]	External Vertical Synchronization signal
i_DCLK	Pixel Clock
HSYNCdck	External Horizontal Synchronization signal after resampling with Pixel Clock Pixel
HSYNCm	External Horizontal Synchronization for the supply to Timing Controller in VDC.
VSYNCm	External Vertical Synchronization for the supply to Timing Controller in VDC.

9.3.10.2. External Head Unit limited

There is a limitation concerning Indigo-L's pixel clock timing and it's relationship to that of the external head unit. Please make sure that the following requirements are met for single line
timing done externally in the Indigo-L head unit:



The overflow zone entails a potential risk, as does the underflow zone.
Please implement timings with safe margins.

9.3.10.3. FIFO Control

The FIFO is controlled with following strategy:

- External RGB data (i_PIXEL_DATA[23:0]) is written
  if i_PIXEL_VALID=1 and DiplayEnable (i_PIXEL_DATA[26])=1
- It is initialized by External Vertical Synchronization (i_PIXEL_DATA[25]).
- It is not initialized by External Horizontal Synchronization (i_PIXEL_DATA[24]).
- External RGB data (i_PIXEL_DATA[23:0]) is read if o_SP_REN=1.

9.3.10.4. Interrupt

If an error condition occurs in the read or write timing of the FIFO, the following interrupts are generated:

- If a FIFO underflow occurs, o_PFUF is generated.
- If a FIFO overflow occurs, o_PFOF is generated.

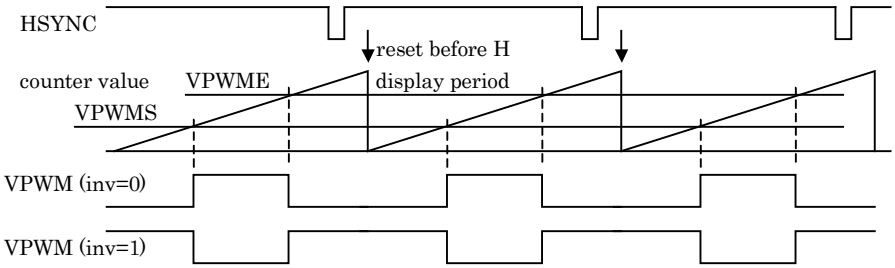
Output control is handled by 9.4.4.9 Interrupt_Enable.

9.3.11. Video sync PWM

This generates the LCD backlight adjustment signal.
PWM signals that synchronizes with HSYNC, VSYNC and the pixel clock are generated and output.
These can be set by a register (9.4.4.21 ~ 9.4.4.24 VPWMxM)
Three typical examples are shown below.

9.3.11.1. HSYNC Reset

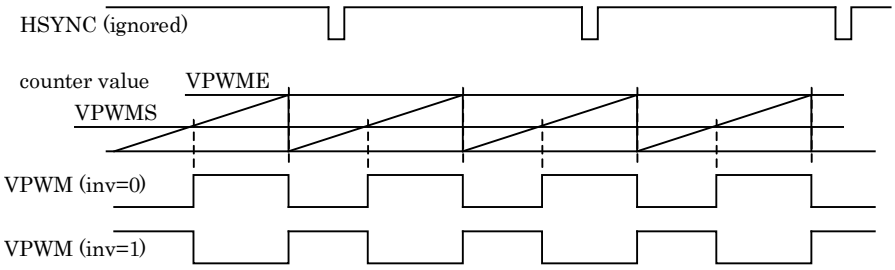
ENDR=0, VSYR=1, HSYB=0 (9.4.4.21 ~ 9.4.4.24 VPWMxM)
VPWM counter value is reset before the horizontal display period.
The VPWM counter is touched each pixel clock.



VPWMS, VPWME B are a register setting value (9.4.4.25 ~ 9.4.4.28 VPWMxSE).

9.3.11.2. END Count Value Reset

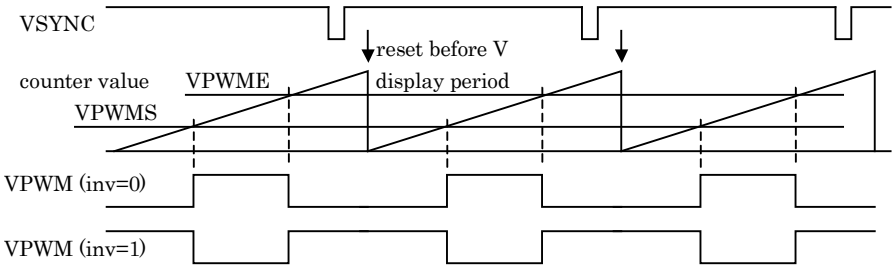
ENDR=1, VSYR=0, HSYB=0 (9.4.4.21 ~ 9.4.4.24 VPWMxM)
VPWM counter value is reset when count value is equal or greater than VPWME.
VPWM counter is touched each pixel clock.



VPWMS,VPWME are a register setting value. (9.4.4.25 ~ 9.4.4.28 VPWMxSE)

9.3.11.3. VSYNC Reset

ENDR=0, VSYR=1, HSYB=1 (9.4.4.21 ~ 9.4.4.24 VPWMxM)
VPWM counter value is reset before Vertical display period.
VPWM counter is touched each HSYNC.



VPWMS,VPWME are a register setting value. (9.4.4.25 ~ 9.4.4.28 VPWMxSE)

9.3.11.4. Interrupt

VPWM outputs an interrupt signal when the internal count value reaches VPWME. (o_VPWM_INT)
There are four PWMs in the VDC.
The status register (9.4.4.8 Status) is used to determine which PWM is outputting the interrupt.

9.4. Registers

9.4.1. Format of Register Descriptions

- Endian

  Only Little Endian access is allowed to the registers of this module.

- Address

  “Address” shows the address of the register. (Base address + Offset address)

- Bit

  “Bit” shows the bit number of the register.

- Name

  “Name” shows the bit field name of the register.

- R/W

  “R/W” shows the attribute of Read/Write in each Bit field.
  R0: The Read value is always "0".
  R1: The Read value is always "1".
  W0: The Write value is always "0". When “1” is written, it is disregarded.
  W1: The Write value is always "1". When “0” is written, it is disregarded.
  R: Read
  W: Write

  Note : If a value is written to registers/bitfields that list R0, R1 and R in the following descriptions, then this value will not be changed in those registers/bitfields.

- Initial value

  “Initial value” is an initial value when reset is released.
  0: It becomes "0".
  1: It becomes "1".
  X: It becomes irregular.

9.4.2. Global Address

For the module base address please refer to the chapter 3 Memory map.

9.4.3. Register summary

Address	Register Name	Description
Base+0x0000	DCM0	Display Control Mode
Base+0x0004	HTP	Horizontal Total Pixels
Base+0x0008	HDP	Horizontal Display Period
Base+0x000C	HSP_HSW_VSW	Horizontal Synchronize pulse Position Horizontal Synchronize pulse Width Vertical Synchronize pulse Width
Base+0x0010	VTR	Vertical Total Raster
Base+0x0014	VDP_VSP	Vertical Display Period Vertical Synchronize pulse Position
Base+0x0018 | Base+0x00FC	Reserved	-
Base+0x0100	DCM1	Display Control Mode
Base+0x0104 | Base+0x3FFC	Reserved	
Base+0x4000	DCM0	Display Control Mode
Base+0x4004	HTP	Horizontal Total Pixels
Base+0x4008	HDP	Horizontal Display Period
Base+0x400C	HSP_HSW_VSW	Horizontal Synchronize pulse Position Horizontal Synchronize pulse Width Vertical Synchronize pulse Width
Base+0x4010	VTR	Vertical Total Raster
Base+0x4014	VDP_VSP	Vertical Display Period Vertical Synchronize pulse Position
Base+0x4018 | Base+0x40FC	Reserved	-
Base+0x4100	DCM1	Display Control Mode
Base+0x4104 | Base+0x41FC	Reserved	-
Base+0x4200	Status	Status
Base+0x4204	Interrupt_Enable	Interrupt_Enable
Base+0x4208	Reserve	-
Base+0x420C	IDLE	IDLE State
Base+0x4210	DLS	Layer Select
Base+0x4214 | Base+0x421C	Reserved	-
Base+0x4220	FGBLD	Foreground Layer - Setting of blend
Base+0x4224	LABLD	LA Display Layer - Setting of blend
Base+0x4228	LBBLD	LB Display Layer - Setting of blend
Base+0x422C | Base+0x423C	Reserved	-
Base+0x4240	LAETC	LA Display Layer - Setting of transparency
Base+0x4244	LBETC	LB Display Layer - Setting of transparency
Base+0x4248 | Base+0x425C	Reserved	-
Base+0x4260	DFGC	Setting of color value of Foreground Layer
Base+0x4264 | Base+0x427C	Reserved	-
Base+0x4280	DBGC	Setting of color value of Background Layer
Base+0x4284 | Base+0x428C	Reserved	-
Base+0x4290	FADEC	Setting of Fade in / Fade out
Base+0x4294	AGT	Setting of Forward Alignment Guard Time and Backward Alignment Guard Time
Base+0x4294 | Base+0x429C	Reserved	
Base+0x4300	VPWM0M	Setting of VPWM0 Mode
Base+0x4304	VPWM1M	Setting of VPWM1 Mode

Base+0x4308	VPWM2M	Setting of VPWM2 Mode
Base+0x430C	VPWM3M	Setting of VPWM3 Mode
Base+0x4310 | Base+0x431C	Reserved	
Base+0x4320	VPWM0SE	Setting of VPWM0 Start/End
Base+0x4324	VPWM1SE	Setting of VPWM1 Start/End
Base+0x4328	VPWM2SE	Setting of VPWM2 Start/End
Base+0x432C	VPWM3SE	Setting of VPWM3 Start/End
Base+0x4230 | Base+0x423C	Reserved	
Base+0x4340	VPWM0C	VPWM0 Count Read
Base+0x4344	VPWM1C	VPWM1 Count Read
Base+0x4348	VPWM2C	VPWM2 Count Read
Base+0x434C	VPWM3C	VPWM3 Count Read
Base+0x4250 | Base+0x425C	Reserved	
Base+0x4360	PFC	Pixel FIFO Delay
Base+0x4364	PFD	Pixel FIFO Depth
Base+0x4350 | Base+0x7FFC	Reserved	

Note: The initial value of the display parameters is VGA.

9.4.4. Register description

9.4.4.1. DCM0 (Display Control Mode 0 Register)

Address	**Base Address** + 0x0000 **Base Address** + 0x4000															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DEN	Reserved												LBE	LAE	FGE
R/W	R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved												SF	ESY	Reserved	
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R/W	R/W	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.4.4.2. DCM1 (Display Control Mode 1 Register)

Address	**Base Address** + 0x0100 **Base Address** + 0x4100															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DEN	Reserved												LBE	LAE	FGE
R/W	R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved												SF	ESY	Reserved	
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R/W	R/W	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

These registers set a display control mode. The DCM0 and DCM1 registers are a single register, available at different addresses.

Bit2 ESY (External SYnchronize)
Sets external synchronous mode.
0 Disables external synchronization.
1 Enables external synchronization.

Bit3 SF (Synchronize signal Format)
Sets the format of the synchronization signal (VSYNC, HSYNC).
0 Negative logic
1 Positive logic

Bit16 FGE (ForeGround Enable)
The Foreground Layer display is made effective.
0 Performs no Foreground display.
1 Performs Foreground display.

Bit17 LAE (LA-layer Enable)
The LA Layer display is made effective.
0 Performs no LA layer display.
1 Performs LA layer display.

Bit18 LBE (LB-layer Enable)
The LB Layer display is made effective.
0 Performs no LB layer display.
1 Performs LB layer display.

Bit31 DEN (Display ENable)
Enables display.
0 Performs no output of the display signal.
1 Performs output of the display signal.

Note: Background Layer is always being displayed. Bacground Layer doesn't set the display or non-display.
Note: The Display output when DEN disables is black (ALL0).
Note: If '0' is written after '1' has been written to DEN, the set register value is preserved.

9.4.4.3. HTP (Horizontal Total Pixels Register)

Address	**Base Address** + 0x0004 **Base Address** + 0x4004															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved				HTP											
R/W	R0	R0	R0	R0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	1	1	0	0	0	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved															
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit27-16 HTP (Horizontal Total Pixels)
Specifies the horizontal total pixel count. "Set value +1" is the horizontal total pixel count.

Note: When HTP is set to 31 pixels or less, HTP is set to31 pixel. Because a minimum resolution is 320x120, the minimum value of HTP is suitable value that is smaller than it.
Note: The set values must meet the following size relationships:
0 < HDP < HSP < HSP+HSW < **HTP** , HSP+6 < **HTP**
When only the external synchronization is used, The set values must meet the following size relationships:
0 < HDP < HSP < HSP + HSW < **HTP**

9.4.4.4. HDP (Horizontal Display Period Register)

Address	**Base Address** + 0x0008 **Base Address** + 0x4008															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved				HDP											
R/W	R0	R0	R0	R0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	1	0	0	1	1	1	1	1	1	1

Bit11-0 HDP (Horizontal Display Period)
Specifies the horizontal display period in units of pixel clocks. "Set value +1" is the pixel count for the display period.

Note: When HDP is set to 15 or less, HDP is set to 15. Because a minimum resolution is 320x120, the minimum value of HDP is suitable value that is smaller than it.
Note: The set values must meet the following magnitude relations:
0 < **HDP** < HSP < HSP+HSW < HTP , HSP+6 < HTP
When only the external synchronization is used, The set values must meet the following size relationships:
0 < **HDP** < HSP < HSP + HSW < HTP

9.4.4.5. HSP_HSW_VSW (Hor. Synchronize pulse Position + Width, Vert. Synchronize pulse Width Register)

Address	**Base Address** + 0x000C **Base Address** + 0x400C															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved		VSW						HSW							
R/W	R0	R0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	1	0	1	0	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved				HSP											
R/W	R0	R0	R0	R0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	1	0	1	0	0	0	1	1	1	1

Bit11-0 HSP (Horizontal Synchronize pulse Position)
Specifies the pulse position of the horizontal synchronization signal in units of pixel clocks. When the clock count from the start of the display period (an offset (15 clocks) is included) reaches "set value + 1", the horizontal synchronization signal is asserted.

Bit23-16 HSW (Horizontal Synchronize pulse Width)
Specifies the pulse width of the horizontal synchronization signal in units of pixel clocks. "Set value + 1" is the clock count of the pulse width.

Bit29-24 VSW (Vertical Synchronize pulse Width)
Specifies the pulse width of the vertical synchronization signal in units of rasters. If internal synchronisation is active, then "Set value + 1" is the raster count of the pulse width. If external synchronisation is active, then "Set value" is the raster count of the pulse width.

Note: When HSP is set to 19 or less, HSP is set to 19. Because a minimum resolution is 320x120, the minimum value of HSP is suitable value that is smaller than it.
Note: When HSW is set to 3 or less, HSW is set to 3. Because a minimum resolution is 320x120, the minimum value of HSW is suitable value that is smaller than it.
Note: When VSW is set to 1 or less, VSW is set to 1. Because a minimum resolution is 320x120, the minimum value of VSW is suitable value that is smaller than it.
Note: The set values must meet the following size relationships:
0 < HDP < **HSP** < **HSP**+**HSW** < HTP , **HSP**+6 < HTP
0 < VDP < VSP < VSP+**VSW** < VTR
When only the external synchronization is used, The set values must meet the following size relationships:
0 < HDP < HSP < **HSP** + HSW < HTP
0 < VDP < VSP < VSP+**VSW** < VTR

9.4.4.6. VTR (Vertical Total Raster Register)

Address	**Base Address** + 0x0010 **Base Address** + 0x4010															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved				VTR											
R/W	R0	R0	R0	R0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved															
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit27-16 VTR (Vertical Total Rasters)
Specifies the vertical total raster count. "Set value + 1" is the vertical total raster count.

Note: When VTR is set to 31 or less, VTR is set to 31. Because a minimum resolution is 320x120, the minimum value of VTR is suitable value that is smaller than it.
Note: The set values must meet the following magnitude relations:
0 < VDP < VSP < VSP+VSW < **VTR**

9.4.4.7. VDP_VSP (Display_Vertical Display Period_Vertical Synchronize pulse Position Register)

Address	**Base Address** + 0x0014 **Base Address** + 0x4014															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved				VDP											
R/W	R0	R0	R0	R0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	1	1	1	0	1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved				VSP											
R/W	R0	R0	R0	R0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	1	1	1	1	0	1	0	0	1

Bit11-0 VSP (Vertical Synchronize pulse Position)
Specifies the pulse position of the vertical synchronization signal in units of rasters. The vertical synchronizing pulse is asserted at the (set value + 1)-th raster relative to the display starting raster.

Bit27-16 VDP (Vertical Display Period)
Specifies the vertical display period in units of rasters. Set value + 1 is the display raster count.

Note: When VSP is set to 19 or less, VSP is set to 19. Because a minimum resolution is 320x120, the minimum value of VSP is suitable value that is smaller than it.
Note: When VDP is set to 15 or less, VDP is set to 15. Because a minimum resolution is 320x120, the minimum value of VDP is suitable value that is smaller than it.
Note: The set values must meet the following magnitude relations:
0 < **VDP** < **VSP** < **VSP**+VSW < VTR

9.4.4.8. Status (Status Register)

Address	**Base Address** + 0x4200															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved												VP3I	VP2I	VP1I	VP0I
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R/W1	R/W1	R/W1	R/W1
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit0 VP0I (VPWM0 Interrupt)
Shows the status of VPWM0. VP0I shows "1" while requesting a pulse width change.
Please set VP0IE bit of Interrupt Enable register to "1" to enable this bit.
Write “1” to clear this bit.
0 There is no pulse width change request.
1 There is a pulse width change request.

Bit1 VP1I (VPWM1 Interrupt)
Shows the status of VPWM 1. VP1I shows "1" while requesting the pulse width change.
Please set VP1IE bit of Interrupt Enable register to "1" to enable this bit.
Write “1” to clear this bit.
0 There is no pulse width change request.
1 There is a pulse width change request.

Bit2 VP2I (VPWM2 Interrupt)
Shows the status of VPWM 2. VP2I shows "1" while requesting the pulse width change.
Please set VP2IE bit of Interrupt Enable register to "1" to enable this bit.
Write “1” to clear this bit.
0 There is no pulse width change request.
1 There is a pulse width change request.

Bit3 VP3I (VPWM3 Interrupt)
Shows the status of VPWM 3. VP3I shows "1" while requesting the pulse width change.
Please set VP3IE bit of Interrupt Enable register to "1" to enable this bit.
Write “1” to clear this bit.
0 There is no pulse width change request.
1 There is a pulse width change request.

9.4.4.9. Interrupt_Enable (Interrupt_Enable Register)

Address	**Base Address** + 0x4204															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved						HSIE	VSIE	PFOIE	PFUIE	EHIE	EVIE	VP3IE	VP2IE	VP1IE	VP0IE
R/W	R0	R0	R0	R0	R0	R0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit0 VP0IE (VPWM0 Interrupt Enable)
The status of the VP0I bit of the Status register is output to o_INT_VPWM by writing "1" to this bit.
0: VP0I bit status is not output.
1: VP0I bit status is output.

Bit1 VP1IE (VPWM1 Interrupt Enable)
The status of the VP1I bit of the Status register is output to o_INT_VPWM by writing "1" to this bit.
0: VP1I bit status is not output.
1: VP1I bit status is output.

Bit2 VP2IE (VPWM2 Interrupt Enable)
The status of the VP2I bit of the Status register is output to o_INT_VPWM by writing "1" to this bit.
0: VP2I bit status is not output.
1: VP2I bit status is output.

Bit3 VP3IE (VPWM3 Interrupt Enable)
The status of the VP3I bit of the Status register is output to o_INT_VPWM by writing "1" to this bit.
0: VP3I bit status is not output.
1: VP3I bit status is output.

Bit4 EVIE (External Vsync Interrupt Enable)
o_EXT_VS is output by writing '1' to this bit.
0: o_EXT_VS is not output.
1: o_EXT_VS is output.

Bit5 EHIE (External Hsync Interrupt Enable)
o_EXT_HS is output by writing '1' to this bit.
0: o_EXT_HS is not output.
1: o_EXT_HS is output.

Bit6 PFUIE (Pixel Fifo Underflow Interrupt Enable)
o_PFUF is output by writing '1' to this bit.
0: o_PFUF is not output.
1: o_PFUF is output.

Bit7 PFOIE (Pixel Fifo Overflow Interrupt Enable)
o_PFOF is output by writing '1' to this bit.
0: o_PFOF is not output.
1: o_PFOF is output.

Bit8 VSIE (VSYNC Interrupt Enable)
o_INT_VSYNC is output by writing '1' to this bit.
0: o_INT_VSYNC is not output.
1: o_INT_VSYNC is output.

Bit9 HSIE (HSYNC Interrupt Enable)
o_INT_HSYNC is output by writing '1' to this bit.
0: o_INT_HSYNC is not output.
1: o_INT_HSYNC is output.

9.4.4.10. IDLE (IDLE Register)

Address	**Base Address** + 0x4208															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved															IDLE
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit0 IDLE (IDLE)

Indicates when the VDC is in an IDLE state. It is the logical OR of the following register values.

DEN of DCMx Register
V0EN of VPWM0M Register
V1EN of VPWM1M Register
V2EN of VPWM2M Register
V3EN of VPWM3M Register

0 IDLE
1 BUSY

9.4.4.11. DLS (Display Layer Select Register)

Address	**Base Address** + 0x4210															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved															DLS
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit0 DLS (Display Layer Select)

Selects the top layer from LA and LB, which are switchable.

0 LA layer is on top
1 LB layer is on top

9.4.4.12. FGBLD (ForeGround BLenD Register)

Address	**Base Address** + 0x4220															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															FGBE
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved						FGBS	FGBI	FGBR							
R/W	R0	R0	R0	R0	R0	R0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit7-0 FGBR (ForeGround Blend Ratio)
Sets the blend ratio. Basically, the blend ratio is the set value/256.
If FEN of the FADEC register is set to '1', the new blend ratio is held in this bitfield

Bit8 FGBI (ForeGround Blend Increment)
Selects whether or not 1/256 is added when the blend ratio is not 0.
0 Does not add 1/256.
1 Adds 1/256.

Bit9 FGBS (ForeGround Blend Select)
Selects a blend operation formula.
0 Upper image × blend ratio + lower image × (1 – blend ratio)
1 Upper image × (1 – blend ratio) + lower image × blend ratio

Bit16 FGBE (ForeGround Blend Enable)
Enables foreground blending
0 Disable foreground blending.
1 Enable foreground blending.

Note: This register is a blend of the foreground layer and the lower image.

9.4.4.13. LABLD (LA BLenD Register)

Address	**Base Address** + 0x4224															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															LABE
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved						LABS	LABI	LABR							
R/W	R0	R0	R0	R0	R0	R0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit7-0 LABR (LA-layer Blend Ratio)
Sets the blend ratio. The blend ratio is basically the set value/256.

Bit8 LABI (LA-layer Blend Increment)
Selects whether or not 1/256 is added when the blend ratio is not 0.
0 Does not add 1/256.
1 Adds 1/256.

Bit9 LABS (LA-layer Blend Select)
Selects a blend operation formula.
0 Upper image × blend ratio + lower image × (1 – blend ratio)
1 Upper image × (1 – blend ratio) + lower image × blend ratio

Bit16 LABE (LA-layer Blend Enable)
Enables LA-layer blend
0 Disable LA-layer blending.
1 Enable LA-layer blending.

Note: This register is a blend of the LA Layer and the lower image.

9.4.4.14. LBBLD (LB BLenD Register)

Address	**Base Address** + 0x4228															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															LBBE
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved						LBBS	LBBI	LBBR							
R/W	R0	R0	R0	R0	R0	R0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit7-0 LBBR(LB-layer Blend Ratio)
Sets the blend ratio. The blend ratio is basically the set value/256.

Bit8 LBBI (LB-layer Blend Increment)
Selects whether or not 1/256 is added when the blend ratio is not 0.
0 Does not add 1/256.
1 Adds 1/256.

Bit9 LBBS(LB-layer Blend Select)
Selects a blend operation formula.
0 Upper image × blend ratio + lower image × (1 – blend ratio)
1 Upper image × (1 – blend ratio) + lower image × blend ratio

Bit16 LBBE (LB-layer Blend Enable)
Enables LB-layer blend
0 Disable LB-layer blending.
1 Enable LB-layer blending.

Note: This register is a blend of the LB Layer and the lower image.

9.4.4.15. LAETC (LA-layer Extend Transparency Control Register)

Address	**Base Address** + 0x4240															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	LAEZT	Reserved							LATEC							
R/W	R/W	R0	R0	R0	R0	R0	R0	R0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	LATEC															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit23-0 LAETC (LA-layer Extend Transparent Color)
Sets the color value (code) displayed as transparent color for LA layer.
LATEC[23:16] Red Data
LATEC[15:8] Green Data
LATEC[7:0] Blue Data

Bit31 LAEZT (LA-layer Extend Zero Transparency)
Sets the handling of a color value (code 0, black) for LA layer.
0 Does not handle code 0 as transparent.
1 Handles code 0 as transparent.

Note: When the color value is set to LAETC, and "1" is set to LAEZT, the color value and color 0 (black) two colors are treated as a transparent color.
Note: When the color value is set to LAETC, and "0" is set to LAEZT, only the color value of LAETC is treated as a transparent color.
Note: When all values "0" (The color value is black) are set to LAETC, and "0" is set to LAEZT, the transparent color is not set.

9.4.4.16. LBETC (LB-layer Extend Transparency Control Register)

Address	**Base Address** + 0x4244															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	LBEZT	Reserved							LBTEC							
R/W	R/W	R0	R0	R0	R0	R0	R0	R0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	LBTEC															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit23-0 LBETC (LB-layer Extend Transparent Color)
Sets the color value (code) displayed as transparent color for LB layer.
LBTEC[23:16] Red Data
LBTEC[15:8] Green Data
LBTEC[7:0] Blue Data

Bit31 LBEZT (LB-layer Extend Zero Transparency)
Sets the handling of a color value (code 0, black) for LB layer.
0 Does not handle code 0 as transparent.
1 Handles code 0 as transparent.

Note: When the color value is set to LBETC, and "1" is set to LBEZT, the color value and color 0 (black) two colors are treated as a transparent color.
Note: When the color value is set to LBETC, and "0" is set to LBEZT, only the color value of LAETC is treated as a transparent color.
Note: When all values "0" (The color value is a black) are set to LBETC, and "0" is set to LBEZT, the transparent color is not set.

9.4.4.17. DFGC (Display_Fore Ground Color Register)

Address	**Base Address** + 0x4260															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved								FGC							
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FGC															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit23-0 FGC (ForeGround Color)
Sets the color value displayed as the front color for the foreground layer.
FGC[23:16] Red Data
FGC[15:8] Green Data
FGC[7:0] Blue Data

9.4.4.18. DBGC (Display_Back Ground Color Register)

Address	**Base Address** + 0x4280															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BBS	Reserved							BGC							
R/W	R/W	R0	R0	R0	R0	R0	R0	R0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	BGC															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit23-0 BGC (BackGround Color)
Sets the color value displayed as the back color for background layer.
BGC[23:16] Red Data
BGC[15:8] Green Data
BGC[7:0] Blue Data

Bit31 BBS (Background color Blend Select)
Blend processing with the background color in Blender1 or it selects it.(9.3.7)
0 The blend is not processed.
1 The blend is processed.

9.4.4.19. FADEC (FADE in / FADE out Control Register)

Address	**Base Address** + 0x4290															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	FEN	Reserved														
R/W	R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	VCNT								STEP							
R/W	R0	R0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit7-0 STEP (blend ratio STEP up value)
The value added to the blend ratio of Foreground is set (FGBR) in two's complement format.
01111111 127
01111110 126
| |
00000001 1
00000000 0
11111111 -1
| |
10000001 -127
10000000 -128

Bit15-8 VCNT (Vertical Synchronize pulse CouNT)
The frequency of the vertical synchronizing signal is set. (frequency in which STEP is processed)
00000000 0
00000001 1
00000010 2
| |
111111111 255

Bit31 FEN (Fade ENable)
FADEC Enable
When 'FADE in/FADE out' mode has ended, '0' is shown. The end conditions are as follows.
· When the internal counter reaches VCNT setting value.
· If the blend ratio of Foreground overflows.
· If the blend ratio of Foreground underflows.
Please write '1' to this bit when you want to execute the FADE in/FADE out mode again.
0 Disable FADEC operation.
1 Enable FADEC operation.

9.4.4.20. AGT (Alignment Guard Time Register)

Address	**Base Address** + 0x4294															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved				V-BAGT				Reserved				V-FAGT			
R/W	R0	R0	R0	R0	R/W	R/W	R/W	R/W	R0	R0	R0	R0	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0

Bit3-0 V-FAGT (Vertical Forward Alignment Guard Time)
Forward Alignment Guard Time of Vertical is set.
0000 0
| |
1111 15

Bit11-8 V-BAGT (Vertical Backward Alignment Guard Time)
Backward Alignment Guard Time of Vertical is set.
0000 0
| |
1111 15

Note: If V-BAGT is set to 0 , V-BAGT is set to 1.
Note: There is no Horizontal Forward Alignment Guard Time.
There is no Horizontal Backward Alignment Guard Time.

9.4.4.21. VPWMOM (Video sync PWM0 Mode Register)

Address	**Base Address** + 0x4300															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	V0EN	Reserved														
R/W	R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved											V0INV	V0VSYR	V0HSYR	V0ENDR	V0HSYB
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit0 V0HSYB (VPWM0 HSYNC time base)
Selects a standard signal for counting upwards
0 counter runs every Pixel clock.
1 counter runs every HSYNC.

Bit1 V0ENDR (VPWM0 END count Reset)
Resets the END counter.
0 ignore VPWM0E register value.
1 reset counter and output if counter value gets equal to or greater than VPWM0E value.

Bit2 V0HSYR (VPWM0 HSYnc Reset)
Resets counter and output based on HSYNC.
0 ignore HSYNC.
1 reset counter and output if HSYNC occurs.

Bit3 V0VSYR (VPWM0 VSYnc Reset)
Resets counter and output based on VSYNC.
0 ignore VSYNC.
1 reset counter and output if VSYNC occurs.

Bit4 V0ENV (VPWM0 INVert output)
Sets the format of the output.
0 VPWM0 output is L initially and becomes H if it is activated.
1 VPWM0 output is H initially and becomes L if it is activated.

Bit31 V0EN (VPWM0 ENable)
VPWM0 Enable
0 Disable VPWM0 operation.
1 Enable VPWM0 operation.

9.4.4.22. VPWM1M (Video sync PWM1 Mode Register)

Address	**Base Address** + 0x4304															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	V1EN	Reserved														
R/W	R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved											V1INV	V1VSYR	V1HSYR	V1ENDR	V1HSYB
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit0 V1HSYB (VPWM1 HSYNC time base)
Selects a standard signal for counting upwards
0 counter runs every Pixel clock.
1 counter runs every HSYNC.

Bit1 V1ENDR (VPWM1 END count Reset)
Resets the END counter.
0 ignore VPWM1E register value.
1 reset counter and output if counter value gets equal to or greater than VPWM1E value.

Bit2 V1HSYR (VPWM1 HSYnc Reset)
Resets counter and output based on HSYNC.
0 ignore HSYNC.
1 reset counter and output if HSYNC occurs.

Bit3 V1VSYR (VPWM1 VSYnc Reset)
Resets counter and output based on VSYNC.
0 ignore VSYNC.
1 reset counter and output if VSYNC occurs.

Bit4 V1ENV (VPWM1 INVert output)
Sets the format of the output.
0 VPWM1 output is L initially and gets H if it is activated.
1 VPWM1 output is H initially and gets L if it is activated.

Bit31 V1EN (VPWM1 ENable)
VPWM1 Enable
0 Disable VPWM1 operation.
1 Enable VPWM1 operation.

9.4.4.23. VPWM2M (Video sync PWM2 Mode Register)

Address	**Base Address** + 0x4308															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	V2EN	Reserved														
R/W	R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved											V2INV	V2VSYR	V2HSYR	V2ENDR	V2HSYB
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit0 V2HSYB (VPWM2 HSYNC time base)
Selects a standard signal for counting upwards
0 counter runs every Pixel clock.
1 counter runs every HSYNC.

Bit1 V2ENDR (VPWM2 END count Reset)
Resets the END counter.
0 ignore VPWM2E register value.
1 reset counter and output if counter value gets equal to or greater than VPWM2E value.

Bit2 V2HSYR (VPWM2 HSYnc Reset)
Resets counter and output based on HSYNC.
0 ignore HSYNC.
1 reset counter and output if HSYNC occurs.

Bit3 V2VSYR (VPWM2 VSYnc Reset)
Resets counter and output based on VSYNC.
0 ignore VSYNC.
1 reset counter and outputif VSYNC occurs.

Bit4 V2ENV (VPWM2 INVert output)
Sets the format of the output.
0 VPWM2 output is L initially and gets H if it is activated.
1 VPWM2 output is H initially and gets L if it is activated.

Bit31 V2EN (VPWM2 ENable)
VPWM2 Enable
0 Disable VPWM2 operation.
1 Enable VPWM2 operation.

9.4.4.24. VPWM3M (Video sync PWM3 Mode Register)

Address	**Base Address** + 0x430C															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	V3EN	Reserved														
R/W	R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved											V3INV	V3VSYR	V3HSYR	V3ENDR	V3HSYB
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit0 V3HSYB (VPWM3 HSYNC time base)
Selects a standard signal for counting upwards
0 counter runs every Pixel clock.
1 counter runs every HSYNC.

Bit1 V3ENDR (VPWM3 END count Reset)
Resets the END counter.
0 ignore VPWM3E register value.
1 reset counter and output if counter value gets equal to or greater than VPWM3E value.

Bit2 V3HSYR (VPWM3 HSYnc Reset)
Resets counter and output based on HSYNC.
0 ignore HSYNC.
1 reset counter and output if HSYNC occurs.

Bit3 V3VSYR (VPWM3 VSYnc Reset)
Resets counter and output based on VSYNC.
0 ignore VSYNC.
1 reset counter and outputif VSYNC occurs.

Bit4 V3ENV (VPWM3 INVert output)
Sets the format of the output.
0 VPWM3 output is L initially and gets H if it is activated.
1 VPWM3 output is H initially and gets L if it is activated.

Bit31 V3EN (VPWM3 ENable)
VPWM3 Enable
0 Disable VPWM3 operation.
1 Enable VPWM3 operation.

9.4.4.25. VPWM0SE (Video sync PWM0 Start End Register)

Address	**Base Address** + 0x4320															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved				VPWM0E											
R/W	R0	R0	R0	R0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved				VPWM0S											
R/W	R0	R0	R0	R0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit0-11 VPWM0S (VPWM0 Start)
VPWM0S defines the VPWM0 counter value at which VPWM0 output is activated.

Bit16-27 VPWM0E (VPWM0 End)
VPWM0E defines the VPWM0 counter value at which VPWM0 output is de-activated or the VPWM0 counter is reset.

9.4.4.26. VPWM1SE (Video sync PWM1 Start End Register)

Address	**Base Address** + 0x4324															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved				VPWM1E											
R/W	R0	R0	R0	R0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved				VPWM1S											
R/W	R0	R0	R0	R0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit0-11 VPWM1S (VPWM1 Start)
VPWM1S defines the VPWM1 counter value at which VPWM1 output is activated.

Bit16-27 VPWM1E (VPWM1 End)
VPWM1E defines the VPWM1 counter value at which VPWM1 output is de-activated or the VPWM1 counter is reset.

9.4.4.27. VPWM2SE (Video sync PWM2 Start End Register)

Address	**Base Address** + 0x4328															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved				VPWM2E											
R/W	R0	R0	R0	R0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved				VPWM2S											
R/W	R0	R0	R0	R0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit0-11 VPWM2S (VPWM2 Start)
VPWM2S defines the VPWM2 counter value at which VPWM2 output is activated.

Bit16-27 VPWM2E (VPWM2 End)
VPWM2E defines the VPWM2 counter value at which VPWM2 output is de-activated or the VPWM2 counter is reset.

9.4.4.28. VPWM3SE (Video sync PWM3 Start End Register)

Address	**Base Address** + 0x432C															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved				VPWM3E											
R/W	R0	R0	R0	R0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved				VPWM3S											
R/W	R0	R0	R0	R0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit0-11 VPWM3S (VPWM3 Start)
VPWM3S defines the VPWM3 counter value at which VPWM3 output is activated.

Bit16-27 VPWM3E (VPWM3 End)
VPWM3E defines the VPWM3 counter value at which VPWM3 output is de-activated or the VPWM3 counter is reset.

9.4.4.29. VPWM0C (Video sync PWM0 Count Register)

Address	**Base Address** + 0x4340															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	V0HOLD	Reserved														
R/W	R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved				VPWM0C											
R/W	R0	R0	R0	R0	R	R	R	R	R	R	R	R	R	R	R	R
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit0-11 VPWM0C (VPWM0 Count)
VPWM0 counter value can be read here. To get precise value from different clock domain, counter value should be held in intermediate register by HOLD flag before read operation.

Bit31 V0HOLD (VPWM0 HOLD)
Counter value maintenance setting
0 counter value is not held.
1 counter value is held at intermediate register to read.

9.4.4.30. VPWM1C (Video sync PWM1 Count Register)

Address	**Base Address** + 0x4344															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	V0HOLD	Reserved														
R/W	R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved				VPWM1C											
R/W	R0	R0	R0	R0	R	R	R	R	R	R	R	R	R	R	R	R
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit0-11 VPWM1C (VPWM1 Count)
VPWM1 counter value can be read here. To get precise value from different clock domain, counter value should be held in intermediate register by HOLD flag before read operation.

Bit31 V0HOLD (VPWM1 HOLD)
Counter value maintenance setting
0 counter value is not held.
1 counter value is held at intermediate register to read.

9.4.4.31. VPWM2C (Video sync PWM2 Count Register)

Address	**Base Address** + 0x4348															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	V0HOLD	Reserved														
R/W	R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved				VPWM2C											
R/W	R0	R0	R0	R0	R	R	R	R	R	R	R	R	R	R	R	R
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit0-11 VPWM2C (VPWM2 Count)
VPWM2 counter value can be read here. To get precise value from different clock domain, counter value should be held in intermediate register by HOLD flag before read operation.

Bit31 V0HOLD (VPWM2 HOLD)
Counter value maintenance setting
0 counter value is not held.
1 counter value is held at intermediate register to read.

9.4.4.32. VPWM3C (Video sync PWM3 Count Register)

Address	**Base Address** + 0x434C															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	V0HOLD	Reserved														
R/W	R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved				VPWM3C											
R/W	R0	R0	R0	R0	R	R	R	R	R	R	R	R	R	R	R	R
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit0-11 VPWM3C (VPWM3 Count)
VPWM3 counter value can be read here. To get precise value from different clock domain, counter value should be held in intermediate register by HOLD flag before read operation.

Bit31 V0HOLD (VPWM3 HOLD)
Counter value maintenance setting
0 counter value is not held.
1 counter value is held at intermediate register to read.

9.4.4.33. PFC (Pixel FIFO Control Register)

Address	**Base Address** + 0x4360															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved									HDELAY						
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit0-6 HDELAY (Hsync DELAY)
HDELAY is a delay added to the external horizontal sync signal in order to stabilize the pixel FIFO.

0000000	512
0000001	4
0000010	8
|	|
11111111	508

9.4.4.34. PFD (Pixel FIFO Depth Register)

Address	**Base Address** + 0x4364															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved								PDMIN							
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R	R	R	R	R	R	R	R
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved								PDMAX							
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R	R	R	R	R	R	R	R
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit0-7 PDMAX (Pixel FIFO Depth MAX)
The maximum FIFO fill level of last video frame can be read.
PDMAX is always 0 or positive. (values > 128 mean overflow)

Bit16-23 PDMIN (Pixel FIFO Depth MINimum)
The minimum FIFO fill level of last video frame can be read in two's complement format.
PDMIN is always 0 or negative.(range 0..-128, negative values mean underflow)

9.5. Application notes

9.5.1. Processing flow

It is necessary to set the display parameters register, before DCM1 Register is enabled (DEN).

An example of VGA display is shown below.

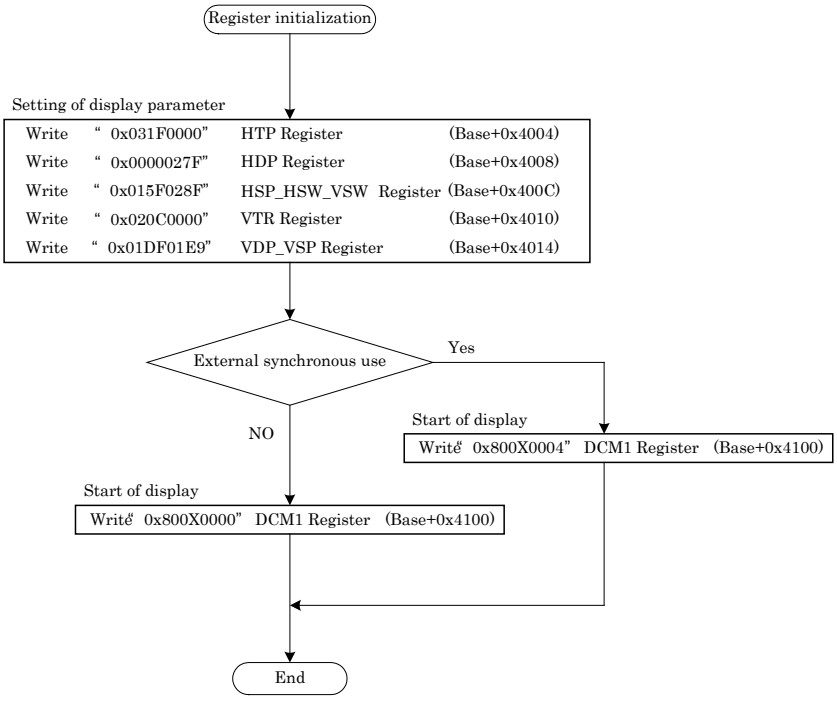


Figure 9-14 Initial Setting

9.5.2. Overlay Pattern

Overlay pattern and setting method are as follows.
The following example is for the LA display layer.

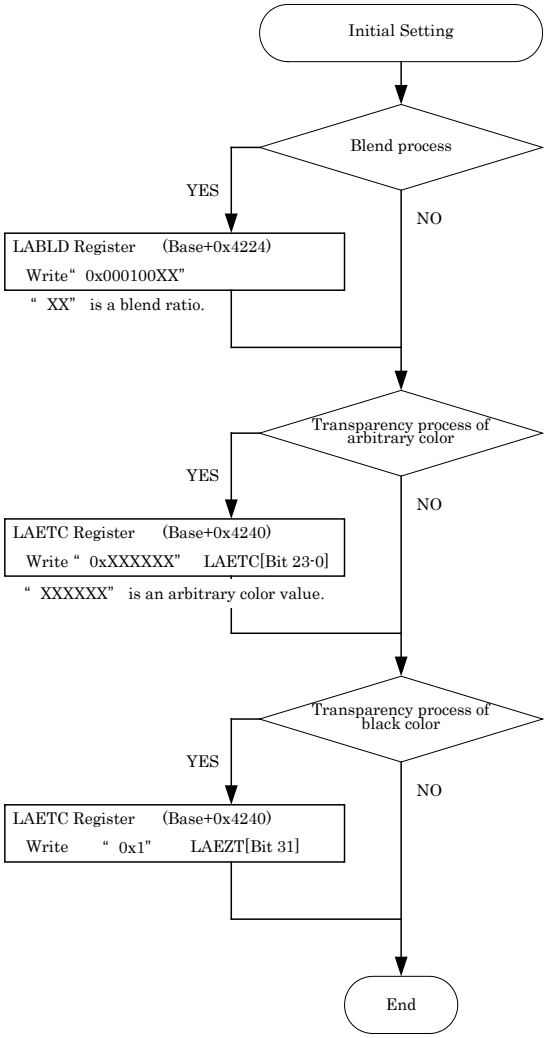


Figure 9-15 Overlay Pattern

9.5.3. Fade in / Fade out flow

Please make the initial settings as follows:
Please write '1' to the FGE bit of the DCMx register.
Please write '1' to the FGBE bit of the FGBLD register.

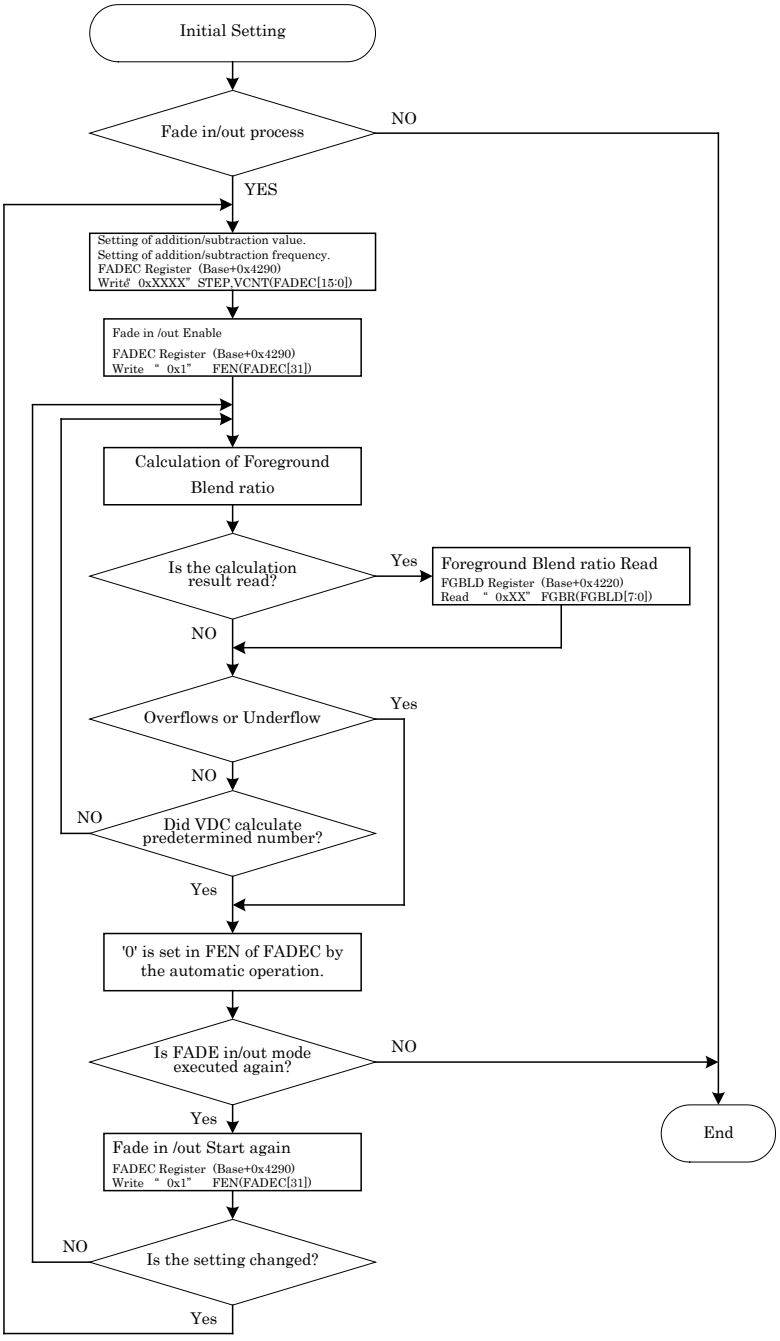


Figure 9-16 Fade in /Fade out flow

9.5.4. Display Parameter Settings

Examples of display parameter settings are shown below, using decimal numbers.

Horizontal Display Parameter

Resolution	HTP	HSP	HSW	HDP
320	399	327	47	319
400	499	409	59	399
480	599	491	71	479
500	624	511	75	499
640	799	655	95	639
800	999	819	119	799
960	1199	983	143	959
1280	1299	1285	7	1279

Vertical Display Parameter

Resolution	VTR	VSP	VSW	VDP
120	139	123	2	119
160	174	162	2	159
240	262	244	4	239
250	272	254	4	249
480	524	489	8	479

10. Timing Controller (TCON)

This chapter describes the Timing Controller of the MB88F333.

10.1. Outline

The Timing Controller module TCON generates the control signals and data signals for direct interfacing to the column and row drivers of a display panel. The freely programmable waveform of the generated timing control signals makes it possible to emulate almost every common timing controller IC (TCON IC) used in display panels. The RGB data is transmitted as single-ended TTL signals or as low voltage differential swing signals conforming to the RSDS™ standard (**R**educed **S**wing **D**ifferential **S**ignal).
The module consists of three submodules; a Timing Signal Generator (TSIG) module, an RSDS™ bit mapping module (RBM) and an IO module for the control of special RSDS™ or TTL capable IO-cells.

10.2. Features

The Timing Controller has the following features.

10.2.1. Features

- **RBM (RSDS Bit Mapping)**
  - Conforms to RSDS™ Standard 1.0 (National Semiconductors)
  - Support for single bus (Multidrop bus with single or double end termination)
  - Mapping for 6 bit color depth
  - Mapping for 8 bit color depth (However RGB output is 6bit color depth)
  - Data and clock outputs can flexible be assigned to the pool of available pins to ease board design

  References:
  RSDS™ “Intra-Panel” Interface Specification, Revision 1.0, May 2003, (National Semiconductor Corporation©)

- **TSIG (Timing Signal Generator)**
  - Freely programmable waveforms
  - 12 pulse generators
  - 1 signal sequencer with max. 64 signal transitions
  - 12 signal mixers with a programmable function table
  - Inversion control signal for transition minimizing (useful for TTL applications)

- **IO module**
  - Control of combined TTL / RSDS IO cells
  - Output RSDS clock
    Output TTL clock
    90° phase shift
  - Adjustable differential swing

10.3. Block diagram

Figure 10-1 shows Timing Controller's block diagram.

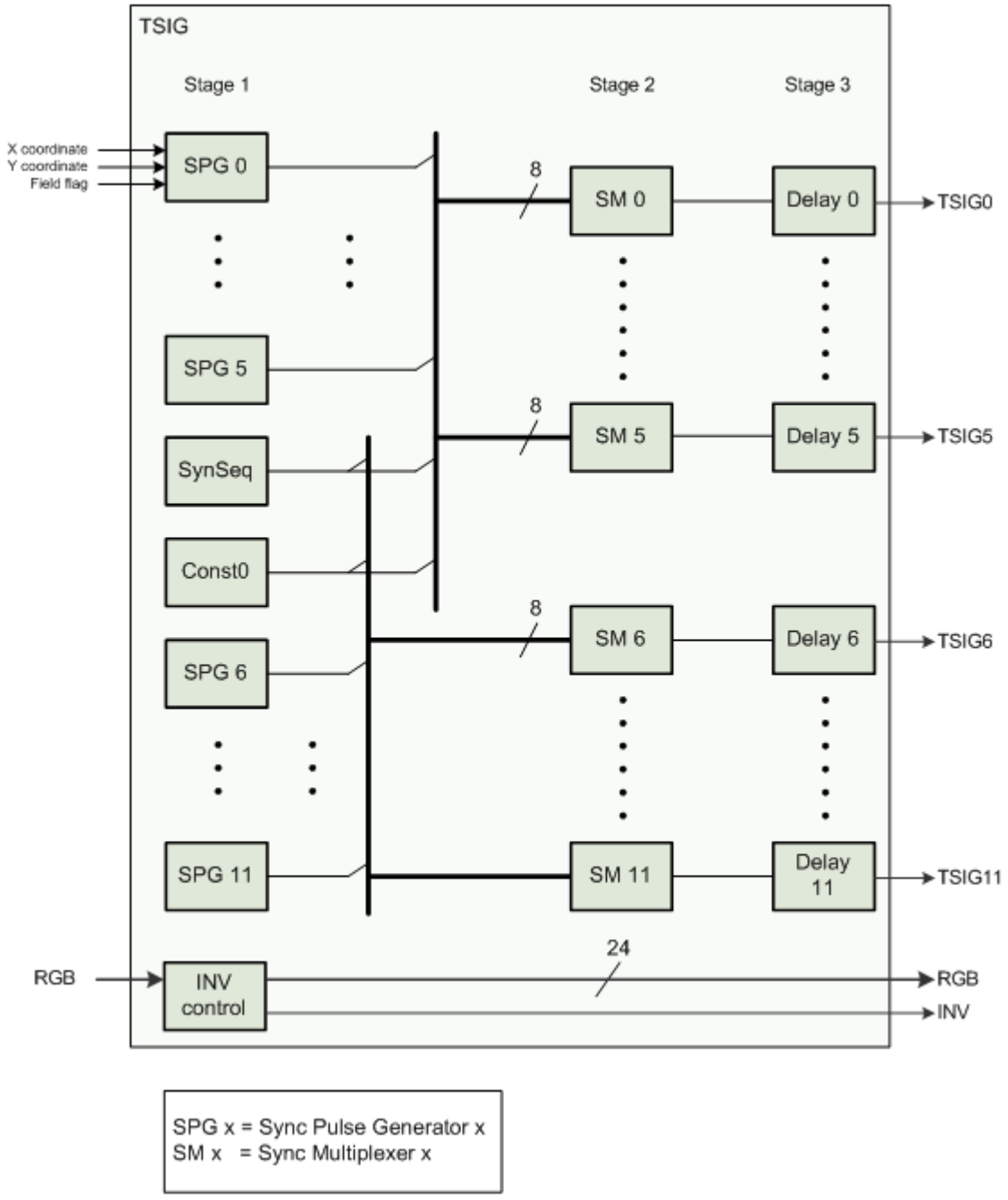


Figure 10-1 Timing Controller module

10.4. Registers

10.4.1. Format of Register Descriptions

The register descriptions in the following sections use the format shown below to describe each bit field of a register.

Register address	Offset																														
Bit number	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
Field name																																
R/W																																
Reset value																																

Meaning of items and sign

Register Address

Register address shows the address (Offset address) of the register.

Bit number

Bit number shows bit position of the register.

Field Name

Field name shows bit name of the register.

R/W

R/W shows the read/write attribute of each bit field:

- R: Read Only
- W: Write Only
- W1C: Writing a value of "1" clears the register.

Reset value

Reset value indicates the value of each bit field immediately after reset.

- 0: Initial value is "0".
- 1: Initial value is "1".
- X: Undefined.

Unused register fields are marked with a solid grey background.

Bit vectors are unsigned integers, if nothing else specified.

10.4.2. Global Address

For the module base address please refer to the chapter 3 Memory map.

10.4.3. Register summary

Address	Register Name	Description
Base address + 0_H : Base address + FF_H	DIR_SSqCnts[1]	Sequencer position definitions, only 32 bit word access is supported
Base address + 400_H	DIR_SWreset	TCON Software Reset - for deactivation conditions see description chapter 10.5.2.2!
Base address + 404_H	DIR_SPG0PosOn	Sync pulse generator 0, "switch-on" position
Base address + 408_H	DIR_SPG0MaskOn	
Base address + $40C_H$	DIR_SPG0PosOff	Sync pulse generator 0, "switch-off" position
Base address + 410_H	DIR_SPG0MaskOff	
Base address + 414_H	DIR_SPG1PosOn	Sync pulse generator 1, "switch-on" position
Base address + 418_H	DIR_SPG1MaskOn	
Base address + $41C_H$	DIR_SPG1PosOff	Sync pulse generator 1, "switch-off" position
Base address + 420_H	DIR_SPG1MaskOff	

[1] Only 32bit word access is supported, no byte or halfword access allowed to this register

Base address + 424_H	DIR_SPG2PosOn	Sync pulse generator 2, "switch-on" position
Base address + 428_H	DIR_SPG2MaskOn	
Base address + $42C_H$	DIR_SPG2PosOff	Sync pulse generator 2, "switch-off" position
Base address + 430_H	DIR_SPG2MaskOff	
Base address + 434_H	DIR_SPG3PosOn	Sync pulse generator 3, "switch-on" position
Base address + 438_H	DIR_SPG3MaskOn	
Base address + $43C_H$	DIR_SPG3PosOff	Sync pulse generator 3, "switch-off" position
Base address + 440_H	DIR_SPG3MaskOff	
Base address + 444_H	DIR_SPG4PosOn	Sync pulse generator 4, "switch-on" position
Base address + 448_H	DIR_SPG4MaskOn	
Base address + $44C_H$	DIR_SPG4PosOff	Sync pulse generator 4, "switch-off" position
Base address + 450_H	DIR_SPG4MaskOff	
Base address + 454_H	DIR_SPG5PosOn	Sync pulse generator 5, "switch-on" position
Base address + 458_H	DIR_SPG5MaskOn	
Base address + $45C_H$	DIR_SPG5PosOff	Sync pulse generator 5, "switch-off" position
Base address + 460_H	DIR_SPG5MaskOff	
Base address + 464_H	DIR_SPG6PosOn	Sync pulse generator 6, "switch-on" position
Base address + 468_H	DIR_SPG6MaskOn	
Base address + $46C_H$	DIR_SPG6PosOff	Sync pulse generator 6, "switch-off" position
Base address + 470_H	DIR_SPG6MaskOff	
Base address + 474_H	DIR_SPG7PosOn	Sync pulse generator 7, "switch-on" position
Base address + 478_H	DIR_SPG7MaskOn	
Base address + $47C_H$	DIR_SPG7PosOff	Sync pulse generator 7, "switch-off" position
Base address + 480_H	DIR_SPG7MaskOff	
Base address + 484_H	DIR_SPG8PosOn	Sync pulse generator 8, "switch-on" position
Base address + 488_H	DIR_SPG8MaskOn	
Base address + $48C_H$	DIR_SPG8PosOff	Sync pulse generator 8, "switch-off" position
Base address + 490_H	DIR_SPG8MaskOff	
Base address + 494_H	DIR_SPG9PosOn	Sync pulse generator 9, "switch-on" position
Base address + 498_H	DIR_SPG9MaskOn	
Base address + $49C_H$	DIR_SPG9PosOff	Sync pulse generator 9, "switch-off" position
Base address + $4A0_H$	DIR_SPG9MaskOff	
Base address + $4A4_H$	DIR_SPG10PosOn	Sync pulse generator 10, "switch-on" position
Base address + $4A8_H$	DIR_SPG10MaskOn	
Base address + $4AC_H$	DIR_SPG10PosOff	Sync pulse generator 10, "switch-off" position
Base address + $4B0_H$	DIR_SPG10MaskOff	
Base address + $4B4_H$	DIR_SPG11PosOn	Sync pulse generator 11, "switch-on" position
Base address + $4B8_H$	DIR_SPG11MaskOn	
Base address + $4BC_H$	DIR_SPG11PosOff	Sync pulse generator 11, "switch-off" position
Base address + $4C0_H$	DIR_SPG11MaskOff	
Base address + $4C4_H$	DIR_SSqCycle	
Base address + $4C8_H$	DIR_SMx0Sigs	Sync mixer 0 signal select
Base address + $4CC_H$	DIR_SMx0FctTable	mixer output = function table [a] a = $s_4*2^4+s_3*2^3+s_2*2^2+s_1*2^1+s_0*2^0$
Base address + $4D0_H$	DIR_SMx1Sigs	Sync mixer 1 signal select
Base address + $4D4_H$	DIR_SMx1FctTable	mixer output = function table [a] a = $s_4*2^4+s_3*2^3+s_2*2^2+s_1*2^1+s_0*2^0$
Base address + $4D8_H$	DIR_SMx2Sigs	Sync mixer 2 signal select
Base address + $4DC_H$	DIR_SMx2FctTable	mixer output = function table [a] a = $s_4*2^4+s_3*2^3+s_2*2^2+s_1*2^1+s_0*2^0$
Base address + $4E0_H$	DIR_SMx3Sigs	Sync mixer 3 signal select
Base address + $4E4_H$	DIR_SMx3FctTable	mixer output = function table [a] a = $s_4*2^4+s_3*2^3+s_2*2^2+s_1*2^1+s_0*2^0$
Base address + $4E8_H$	DIR_SMx4Sigs	Sync mixer 4 signal select
Base address + $4EC_H$	DIR_SMx4FctTable	mixer output = function table [a] a = $s_4*2^4+s_3*2^3+s_2*2^2+s_1*2^1+s_0*2^0$
Base address + $4F0_H$	DIR_SMx5Sigs	Sync mixer 5 signal select
Base address + $4F4_H$	DIR_SMx5FctTable	mixer output = function table [a] a = $s_4*2^4+s_3*2^3+s_2*2^2+s_1*2^1+s_0*2^0$
Base address + $4F8_H$	DIR_SMx6Sigs	Sync mixer 6 signal select
Base address + $4FC_H$	DIR_SMx6FctTable	mixer output = function table [a] a = $s_4*2^4+s_3*2^3+s_2*2^2+s_1*2^1+s_0*2^0$
Base address + 500_H	DIR_SMx7Sigs	Sync mixer 7 signal select
Base address + 504_H	DIR_SMx7FctTable	mixer output = function table [a] a = $s_4*2^4+s_3*2^3+s_2*2^2+s_1*2^1+s_0*2^0$
Base address + 508_H	DIR_SMx8Sigs	Sync mixer 8 signal select
Base address + $50C_H$	DIR_SMx8FctTable	mixer output = function table [a] a = $s_4*2^4+s_3*2^3+s_2*2^2+s_1*2^1+s_0*2^0$
Base address + 510_H	DIR_SMx9Sigs	Sync mixer 9 signal select
Base address + 514_H	DIR_SMx9FctTable	mixer output = function table [a] a = $s_4*2^4+s_3*2^3+s_2*2^2+s_1*2^1+s_0*2^0$
Base address + 518_H	DIR_SMx10Sigs	Sync mixer 10 signal select
Base address + $51C_H$	DIR_SMx10FctTable	mixer output = function table [a] a = $s_4*2^4+s_3*2^3+s_2*2^2+s_1*2^1+s_0*2^0$
Base address + 520_H	DIR_SMx11Sigs	Sync mixer 11 signal select
Base address + 524_H	DIR_SMx11FctTable	mixer output = function table [a] a = $s_4*2^4+s_3*2^3+s_2*2^2+s_1*2^1+s_0*2^0$
Base address + 528_H	DIR_SSwitch	Sync switch
Base address + $52C_H$	DIR_RBM_CTRL	RSDS Bit Map Control
Base address + 530_H	DIR_IO_CTRL	IO-module Control
Base address + 534_H	DIR_PIN0_CTRL	IO Module Pad 0 Control

Base address + 538_H	DIR_PIN1_CTRL	IO Module Pad 1 Control
Base address + $53C_H$	DIR_PIN2_CTRL	IO Module Pad 2 Control
Base address + 540_H	DIR_PIN3_CTRL	IO Module Pad 3 Control
Base address + 544_H	DIR_PIN4_CTRL	IO Module Pad 4 Control
Base address + 548_H	DIR_PIN5_CTRL	IO Module Pad 5 Control
Base address + $54C_H$	DIR_PIN6_CTRL	IO Module Pad 6 Control
Base address + 550_H	DIR_PIN7_CTRL	IO Module Pad 7 Control
Base address + 554_H	DIR_PIN8_CTRL	IO Module Pad 8 Control
Base address + 558_H	DIR_PIN9_CTRL	IO Module Pad 9 Control

10.4.4. Register Description

10.4.4.1. DIR_SSqCnts[0...63]

Register address	BaseAddress + 0H : BaseAddress + FFH																															
Bit number	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
Field name	SSQCNTS_OUT	SSQCNTS_SEQX															reservedseq	SSQCNTS_SEQY														
R/W	RW	RW															RW	RW														
Reset value	X	X															X	X														

Sequencer position definitions, only 32 bit word access is supported

Bit 31 SSQCNTS_OUT
output value, when position is reached

Bit 30 - 16 SSQCNTS_SEQX
X scan position

Bit 15 reservedseq
reserved (set to 0)

Bit 14 - 0 SSQCNTS_SEQY
Y scan position

10.4.4.2. DIR_SWreset

Register address	BaseAddress + 400H																															
Bit number	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
Field name																																SWReset
R/W																																RW
Reset value																																1H

TCON Software Reset - for deactivation conditions see description chapter 10.5.2.2!

Bit 0 SWReset
Software reset: write 0b=no effect, 1b=activate Reset, SW reset is deasserted by internal logic (see description chapter 10.5.2.2.), read: 0b: reset not active 1b: reset active (that means no last pixel of video frame was input to TCON since last activation of sw-reset)

10.4.4.3. DIR_SPG0PosOn

Register address	BaseAddress + 404H																														
Bit number	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
Field name	SPGPSON_TOGGLE0	SPGPSON_X0															reservedon0	SPGPSON_Y0														
R/W	RW	RW															RW	RW														
Reset value	0H	0H															0H	0H														

Sync pulse generator 0, 'Switch on' position
Bit 31 SPGPSON_TOGGLE0 Toggle enable: 0b=disable, 1b=enable
Bit 30 - 16 SPGPSON_X0 X scan position
Bit 15 reservedon0 reserved (set to 0)
Bit 14 - 0 SPGPSON_Y0 Y scan position

10.4.4.4. DIR_SPG0MaskOn

Register address	BaseAddress + 408H																														
Bit number	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
Field name		SPGMKON0																														
R/W		RW																														
Reset value		0H																														

Bit 30 - 0 SPGMKON0
Mask bits: 0b=include bit in position matching, 1b= do not include this bit in position matching

10.4.4.5. DIR_SPG0PosOff

Register address	BaseAddress + 40CH																														
Bit number	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
Field name	SPGPSOFF_TOGGLE0	SPGPSOFF_X0															reservedoff0	SPGPSOFF_Y0														
R/W	RW	RW															RW	RW														
Reset value	0H	0H															0H	0H														

Sync pulse generator 0, 'Switch off' position
Bit 31 SPGPSOFF_TOGGLE0 Toggle enable: 0b=disable, 1b=enable
Bit 30 - 16 SPGPSOFF_X0 X scan position
Bit 15 reservedoff0 reserved (set to 0)
Bit 14 - 0 SPGPSOFF_Y0 Y scan position

10.4.4.6. DIR_SPG0MaskOff

Register address	BaseAddress + 410H																														
Bit number	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
Field name		SPGMKOFF0																														
R/W		RW																														

Reset value		0_H

Bit 30 - 0 SPGMKOFF0
Mask bits: 0b=include bit in position matching, 1b= do not include this bit in position matching

10.4.4.7. DIR_SPG1PosOn

Register address	BaseAddress + 414_H																														
Bit number	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
Field name	SPGPSON_TOGGLE1	SPGPSON_X1															reservedon1	SPGPSON_Y1														
R/W	RW	RW															RW	RW														
Reset value	0_H	0_H															0_H	0_H														

Sync pulse generator 1, 'Switch on' position
Bit 31 SPGPSON_TOGGLE1 Toggle enable: 0b=disable, 1b=enable
Bit 30 - 16 SPGPSON_X1 X scan position
Bit 15 reservedon1 reserved (set to 0)
Bit 14 - 0 SPGPSON_Y1 Y scan position

10.4.4.8. DIR_SPG1MaskOn

Register address	BaseAddress + 418_H																														
Bit number	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
Field name		SPGMKON1																														
R/W		RW																														
Reset value		0_H																														

Bit 30 - 0 SPGMKON1
mask bits (1= do not include this bit into position matching)

10.4.4.9. DIR_SPG1PosOff

Register address	BaseAddress + $41C_H$																														
Bit number	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
Field name	SPGPSOFF_TOGGLE1	SPGPSOFF_X1															reservedoff1	SPGPSOFF_Y1														
R/W	RW	RW															RW	RW														
Reset value	0_H	0_H															0_H	0_H														

Sync pulse generator 1, 'Switch off' position
Bit 31 SPGPSOFF_TOGGLE1 Toggle enable: 0b=disable, 1b=enable
Bit 30 - 16 SPGPSOFF_X1 X scan position
Bit 15 reservedoff1 reserved (set to 0)
Bit 14 - 0 SPGPSOFF_Y1 Y scan position

10.4.4.10. DIR_SPG1MaskOff

Register address	BaseAddress + 420_H																														
Bit number	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
Field name		SPGMKOFF1																														

R/W		RW
Reset value		0_H

Bit 30 - 0 SPGMKOFF1
Mask bits: 0b=include bit in position matching, 1b= do not include this bit in position matching

10.4.4.11. DIR_SPG2PosOn

Register address	BaseAddress + 424_H			
Bit number	31	30 - 16	15	14 - 0
Field name	SPGPSON_TOGGLE2	SPGPSON_X2	reservedon2	SPGPSON_Y2
R/W	RW	RW	RW	RW
Reset value	0_H	0_H	0_H	0_H

Sync pulse generator 2, 'Switch on' position
Bit 31 SPGPSON_TOGGLE2 Toggle enable: 0b=disable, 1b=enable
Bit 30 - 16 SPGPSON_X2 X scan position
Bit 15 reservedon2 reserved (set to 0)
Bit 14 - 0 SPGPSON_Y2 Y scan position

10.4.4.12. DIR_SPG2MaskOn

Register address	BaseAddress + 428_H																														
Bit number	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
Field name		SPGMKON2																														
R/W		RW																														
Reset value		0_H																														

Bit 30 - 0 SPGMKON2
Mask bits: 0b=include bit in position matching, 1b= do not include this bit in position matching

10.4.4.13. DIR_SPG2PosOff

Register address	BaseAddress + $42C_H$			
Bit number	31	30 - 16	15	14 - 0
Field name	SPGPSOFF_TOGGLE2	SPGPSOFF_X2	reservedoff2	SPGPSOFF_Y2
R/W	RW	RW	RW	RW
Reset value	0_H	0_H	0_H	0_H

Sync pulse generator 2, 'Switch off' position
Bit 31 SPGPSOFF_TOGGLE2 Toggle enable: 0b=disable, 1b=enable
Bit 30 - 16 SPGPSOFF_X2 X scan position
Bit 15 reservedoff2 reserved (set to 0)
Bit 14 - 0 SPGPSOFF_Y2 Y scan position

10.4.4.14. DIR_SPG2MaskOff

Register address	BaseAddress + 430_H																														
Bit number	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
Field name		SPGMKOFF2																														

R/W		RW
Reset value		0_H

Bit 30 - 0 SPGMKOFF2
Mask bits: 0b=include bit in position matching, 1b= do not include this bit in position matching

10.4.4.15. DIR_SPG3PosOn

Register address	BaseAddress + 434_H			
Bit number	31	30 - 16	15	14 - 0
Field name	SPGPSON_TOGGLE3	SPGPSON_X3	reservedon3	SPGPSON_Y3
R/W	RW	RW	RW	RW
Reset value	0_H	0_H	0_H	0_H

Sync pulse generator 3, 'Switch on' position
Bit 31 SPGPSON_TOGGLE3 Toggle enable: 0b=disable, 1b=enable
Bit 30 - 16 SPGPSON_X3 X scan position
Bit 15 reservedon3 reserved (set to 0)
Bit 14 - 0 SPGPSON_Y3 Y scan position

10.4.4.16. DIR_SPG3MaskOn

Register address	BaseAddress + 438_H																													
Bit number	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
Field name		SPGMKON3																														
R/W		RW																														
Reset value		0_H																														

Bit 30 - 0 SPGMKON3
Mask bits: 0b=include bit in position matching, 1b= do not include this bit in position matching

10.4.4.17. DIR_SPG3PosOff

Register address	BaseAddress + $43C_H$			
Bit number	31	30 - 16	15	14 - 0
Field name	SPGPSOFF_TOGGLE3	SPGPSOFF_X3	reservedoff3	SPGPSOFF_Y3
R/W	RW	RW	RW	RW
Reset value	0_H	0_H	0_H	0_H

Sync pulse generator 3, 'Switch off' position
Bit 31 SPGPSOFF_TOGGLE3 Toggle enable: 0b=disable, 1b=enable
Bit 30 - 16 SPGPSOFF_X3 X scan position
Bit 15 reservedoff3 reserved (set to 0)
Bit 14 - 0 SPGPSOFF_Y3 Y scan position

10.4.4.18. DIR_SPG3MaskOff

Register address	BaseAddress + 440_H																													
Bit number	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
Field name		SPGMKOFF3																														

R/W		RW
Reset value		0_H

Bit 30 - 0 SPGMKOFF3
Mask bits: 0b=include bit in position matching, 1b= do not include this bit in position matching

10.4.4.19. DIR_SPG4PosOn

Register address	BaseAddress + 444_H			
Bit number	31	30 - 16	15	14 - 0
Field name	SPGPSON_TOGGLE4	SPGPSON_X4	reservedon4	SPGPSON_Y4
R/W	RW	RW	RW	RW
Reset value	0_H	0_H	0_H	0_H

Sync pulse generator 4, 'Switch on' position
Bit 31 SPGPSON_TOGGLE4 Toggle enable: 0b=disable, 1b=enable
Bit 30 - 16 SPGPSON_X4 X scan position
Bit 15 reservedon4 reserved (set to 0)
Bit 14 - 0 SPGPSON_Y4 Y scan position

10.4.4.20. DIR_SPG4MaskOn

Register address	BaseAddress + 448_H																														
Bit number	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
Field name		SPGMKON4																														
R/W		RW																														
Reset value		0_H																														

Bit 30 - 0 SPGMKON4
Mask bits: 0b=include bit in position matching, 1b= do not include this bit in position matching

10.4.4.21. DIR_SPG4PosOff

Register address	BaseAddress + $44C_H$			
Bit number	31	30 - 16	15	14 - 0
Field name	SPGPSOFF_TOGGLE4	SPGPSOFF_X4	reservedoff4	SPGPSOFF_Y4
R/W	RW	RW	RW	RW
Reset value	0_H	0_H	0_H	0_H

Sync pulse generator 4, 'Switch off' position
Bit 31 SPGPSOFF_TOGGLE4 Toggle enable: 0b=disable, 1b=enable
Bit 30 - 16 SPGPSOFF_X4 X scan position
Bit 15 reservedoff4 reserved (set to 0)
Bit 14 - 0 SPGPSOFF_Y4 Y scan position

10.4.4.22. DIR_SPG4MaskOff

Register address	BaseAddress + 450_H																														
Bit number	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
Field name		SPGMKOFF4																														

R/W		RW
Reset value		0_H

Bit 30 - 0 SPGMKOFF4
Mask bits: 0b=include bit in position matching, 1b= do not include this bit in position matching

10.4.4.23. DIR_SPG5PosOn

Register address	BaseAddress + 454_H			
Bit number	31	30 - 16	15	14 - 0
Field name	SPGPSON_TOGGLE5	SPGPSON_X5	reservedon5	SPGPSON_Y5
R/W	RW	RW	RW	RW
Reset value	0_H	0_H	0_H	0_H

Sync pulse generator 5, 'Switch on' position
Bit 31 SPGPSON_TOGGLE5 Toggle enable: 0b=disable, 1b=enable
Bit 30 - 16 SPGPSON_X5 X scan position
Bit 15 reservedon5 reserved (set to 0)
Bit 14 - 0 SPGPSON_Y5 Y scan position

10.4.4.24. DIR_SPG5MaskOn

Register address	BaseAddress + 458_H																													
Bit number	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
Field name		SPGMKON5																														
R/W		RW																														
Reset value		0_H																														

Bit 30 - 0 SPGMKON5
Mask bits: 0b=include bit in position matching, 1b= do not include this bit in position matching

10.4.4.25. DIR_SPG5PosOff

Register address	BaseAddress + $45C_H$			
Bit number	31	30 - 16	15	14 - 0
Field name	SPGPSOFF_TOGGLE5	SPGPSOFF_X5	reservedoff5	SPGPSOFF_Y5
R/W	RW	RW	RW	RW
Reset value	0_H	0_H	0_H	0_H

Sync pulse generator 5, 'Switch off' position
Bit 31 SPGPSOFF_TOGGLE5 Toggle enable: 0b=disable, 1b=enable
Bit 30 - 16 SPGPSOFF_X5 X scan position
Bit 15 reservedoff5 reserved (set to 0)
Bit 14 - 0 SPGPSOFF_Y5 Y scan position

10.4.4.26. DIR_SPG5MaskOff

Register address	BaseAddress + 460_H																													
Bit number	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
Field name		SPGMKOFF5																														

R/W		RW
Reset value		0_H

Bit 30 - 0 SPGMKOFF5
Mask bits: 0b=include bit in position matching, 1b= do not include this bit in position matching

10.4.4.27. DIR_SPG6PosOn

Register address	BaseAddress + 464_H			
Bit number	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
Field name	SPGPSON_TOGGLE6	SPGPSON_X6	reservedon6	SPGPSON_Y6
R/W	RW	RW	RW	RW
Reset value	0_H	0_H	0_H	0_H

Sync pulse generator 6, 'Switch on' position
Bit 31 SPGPSON_TOGGLE6 Toggle enable: 0b=disable, 1b=enable
Bit 30 - 16 SPGPSON_X6 X scan position
Bit 15 reservedon6 reserved (set to 0)
Bit 14 - 0 SPGPSON_Y6 Y scan position

10.4.4.28. DIR_SPG6MaskOn

Register address	BaseAddress + 468_H																													
Bit number	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
Field name		SPGMKON6																														
R/W		RW																														
Reset value		0_H																														

Bit 30 - 0 SPGMKON6
Mask bits: 0b=include bit in position matching, 1b= do not include this bit in position matching

10.4.4.29. DIR_SPG6PosOff

Register address	BaseAddress + $46C_H$			
Bit number	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
Field name	SPGPSOFF_TOGGLE6	SPGPSOFF_X6	reservedoff6	SPGPSOFF_Y6
R/W	RW	RW	RW	RW
Reset value	0_H	0_H	0_H	0_H

Sync pulse generator 6, 'Switch off' position
Bit 31 SPGPSOFF_TOGGLE6 Toggle enable: 0b=disable, 1b=enable
Bit 30 - 16 SPGPSOFF_X6 X scan position
Bit 15 reservedoff6 reserved (set to 0)
Bit 14 - 0 SPGPSOFF_Y6 Y scan position

10.4.4.30. DIR_SPG6MaskOff

Register address	BaseAddress + 470_H																													
Bit number	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
Field name		SPGMKOFF6																														

R/W		RW
Reset value		0_H

Bit 30 - 0 SPGMKOFF6
Mask bits: 0b=include bit in position matching, 1b= do not include this bit in position matching

10.4.4.31. DIR_SPG7PosOn

Register address	BaseAddress + 474_H																															
Bit number	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
Field name	SPGPSON_TOGGLE7	SPGPSON_X7															reservedon7	SPGPSON_Y7														
R/W	RW	RW															RW	RW														
Reset value	0_H	0_H															0_H	0_H														

Sync pulse generator 7, 'Switch on' position
Bit 31 SPGPSON_TOGGLE7 Toggle enable: 0b=disable, 1b=enable
Bit 30 - 16 SPGPSON_X7 X scan position
Bit 15 reservedon7 reserved (set to 0)
Bit 14 - 0 SPGPSON_Y7 Y scan position

10.4.4.32. DIR_SPG7MaskOn

Register address	BaseAddress + 478_H																															
Bit number	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
Field name		SPGMKON7																														
R/W		RW																														
Reset value		0_H																														

Bit 30 - 0 SPGMKON7
Mask bits: 0b=include bit in position matching, 1b= do not include this bit in position matching

10.4.4.33. DIR_SPG7PosOff

Register address	BaseAddress + $47C_H$																															
Bit number	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
Field name	SPGPSOFF_TOGGLE7	SPGPSOFF_X7															reservedoff7	SPGPSOFF_Y7														
R/W	RW	RW															RW	RW														
Reset value	0_H	0_H															0_H	0_H														

Sync pulse generator 7, 'Switch off' position
Bit 31 SPGPSOFF_TOGGLE7 Toggle enable: 0b=disable, 1b=enable
Bit 30 - 16 SPGPSOFF_X7 X scan position
Bit 15 reservedoff7 reserved (set to 0)
Bit 14 - 0 SPGPSOFF_Y7 Y scan position

10.4.4.34. DIR_SPG7MaskOff

Register address	BaseAddress + 480_H																															
Bit number	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
Field name		SPGMKOFF7																														

R/W		RW
Reset value		0_H

Bit 30 - 0 SPGMKOFF7
Mask bits: 0b=include bit in position matching, 1b= do not include this bit in position matching

10.4.4.35. DIR_SPG8PosOn

Register address	BaseAddress + 484_H			
Bit number	31	30 - 16	15	14 - 0
Field name	SPGPSON_TOGGLE8	SPGPSON_X8	reservedon8	SPGPSON_Y8
R/W	RW	RW	RW	RW
Reset value	0_H	0_H	0_H	0_H

Sync pulse generator 8, 'Switch on' position
Bit 31 SPGPSON_TOGGLE8 Toggle enable: 0b=disable, 1b=enable
Bit 30 - 16 SPGPSON_X8 X scan position
Bit 15 reservedon8 reserved (set to 0)
Bit 14 - 0 SPGPSON_Y8 Y scan position

10.4.4.36. DIR_SPG8MaskOn

Register address	BaseAddress + 488_H																													
Bit number	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
Field name		SPGMKON8																														
R/W		RW																														
Reset value		0_H																														

Bit 30 - 0 SPGMKON8
Mask bits: 0b=include bit in position matching, 1b= do not include this bit in position matching

10.4.4.37. DIR_SPG8PosOff

Register address	BaseAddress + $48C_H$			
Bit number	31	30 - 16	15	14 - 0
Field name	SPGPSOFF_TOGGLE8	SPGPSOFF_X8	reservedoff8	SPGPSOFF_Y8
R/W	RW	RW	RW	RW
Reset value	0_H	0_H	0_H	0_H

Sync pulse generator 8, 'Switch off' position
Bit 31 SPGPSOFF_TOGGLE8 Toggle enable: 0b=disable, 1b=enable
Bit 30 - 16 SPGPSOFF_X8 X scan position
Bit 15 reservedoff8 reserved (set to 0)
Bit 14 - 0 SPGPSOFF_Y8 Y scan position

10.4.4.38. DIR_SPG8MaskOff

Register address	BaseAddress + 490_H																													
Bit number	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
Field name		SPGMKOFF8																														

R/W		RW
Reset value		0_H

Bit 30 - 0 SPGMKOFF8
Mask bits: 0b=include bit in position matching, 1b= do not include this bit in position matching

10.4.4.39. DIR_SPG9PosOn

Register address	BaseAddress + 494_H			
Bit number	31	30 - 16	15	14 - 0
Field name	SPGPSON_TOGGLE9	SPGPSON_X9	reservedon9	SPGPSON_Y9
R/W	RW	RW	RW	RW
Reset value	0_H	0_H	0_H	0_H

Sync pulse generator 9, 'Switch on' position
Bit 31 SPGPSON_TOGGLE9 Toggle enable: 0b=disable, 1b=enable
Bit 30 - 16 SPGPSON_X9 X scan position
Bit 15 reservedon9 reserved (set to 0)
Bit 14 - 0 SPGPSON_Y9 Y scan position

10.4.4.40. DIR_SPG9MaskOn

Register address	BaseAddress + 498_H																														
Bit number	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
Field name		SPGMKON9																														
R/W		RW																														
Reset value		0_H																														

Bit 30 - 0 SPGMKON9
Mask bits: 0b=include bit in position matching, 1b= do not include this bit in position matching

10.4.4.41. DIR_SPG9PosOff

Register address	BaseAddress + $49C_H$			
Bit number	31	30 - 16	15	14 - 0
Field name	SPGPSOFF_TOGGLE9	SPGPSOFF_X9	reservedoff9	SPGPSOFF_Y9
R/W	RW	RW	RW	RW
Reset value	0_H	0_H	0_H	0_H

Sync pulse generator 9, 'Switch off' position
Bit 31 SPGPSOFF_TOGGLE9 Toggle enable: 0b=disable, 1b=enable
Bit 30 - 16 SPGPSOFF_X9 X scan position
Bit 15 reservedoff9 reserved (set to 0)
Bit 14 - 0 SPGPSOFF_Y9 Y scan position

10.4.4.42. DIR_SPG9MaskOff

Register address	BaseAddress + $4A0_H$																														
Bit number	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
Field name		SPGMKOFF9																														

R/W		RW
Reset value		0_H

Bit 30 - 0 SPGMKOFF9
Mask bits: 0b=include bit in position matching, 1b= do not include this bit in position matching

10.4.4.43. DIR_SPG10PosOn

Register address	BaseAddress + $4A4_H$																													
Bit number	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
Field name	SPGPSON_TOGGLE10	SPGPSON_X10															reservedon10	SPGPSON_Y10														
R/W	RW	RW															RW	RW														
Reset value	0_H	0_H															0_H	0_H														

Sync pulse generator 10, 'Switch on' position
Bit 31 SPGPSON_TOGGLE10 toggle enable: 0b=disable, 1b=enable
Bit 30 - 16 SPGPSON_X10 X scan position
Bit 15 reservedon10 reserved (set to 0)
Bit 14 - 0 SPGPSON_Y10 Y scan position

10.4.4.44. DIR_SPG10MaskOn

Register address	BaseAddress + $4A8_H$																													
Bit number	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
Field name		SPGMKON10																														
R/W		RW																														
Reset value		0_H																														

Bit 30 - 0 SPGMKON10
Mask bits: 0b=include bit in position matching, 1b= do not include this bit in position matching

10.4.4.45. DIR_SPG10PosOff

Register address	BaseAddress + $4AC_H$																													
Bit number	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
Field name	SPGPSOFF_TOGGLE10	SPGPSOFF_X10															reservedoff10	SPGPSOFF_Y10														
R/W	RW	RW															RW	RW														
Reset value	0_H	0_H															0_H	0_H														

Sync pulse generator 10, 'Switch off' position
Bit 31 SPGPSOFF_TOGGLE10 toggle enable: 0b=disable, 1b=enable
Bit 30 - 16 SPGPSOFF_X10 X scan position
Bit 15 reservedoff10 reserved (set to 0)
Bit 14 - 0 SPGPSOFF_Y10 Y scan position

10.4.4.46. DIR_SPG10MaskOff

Register address	BaseAddress + $4B0_H$																													
Bit number	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
Field name		SPGMKOFF10																														

R/W		RW
Reset value		0_H

Bit 30 - 0 SPGMKOFF10
Mask bits: 0b=include bit in position matching, 1b= do not include this bit in position matching

10.4.4.47. DIR_SPG11PosOn

Register address	BaseAddress + $4B4_H$			
Bit number	31	30 - 16	15	14 - 0
Field name	SPGPSON_TOGGLE11	SPGPSON_X11	reservedon11	SPGPSON_Y11
R/W	RW	RW	RW	RW
Reset value	0_H	0_H	0_H	0_H

Sync pulse generator 11, 'Switch on' position
Bit 31 SPGPSON_TOGGLE11 toggle enable: 0b=disable, 1b=enable
Bit 30 - 16 SPGPSON_X11 X scan position
Bit 15 reservedon11 reserved (set to 0)
Bit 14 - 0 SPGPSON_Y11 Y scan position

10.4.4.48. DIR_SPG11MaskOn

Register address	BaseAddress + $4B8_H$																													
Bit number	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
Field name		SPGMKON11																														
R/W		RW																														
Reset value		0_H																														

Bit 30 - 0 SPGMKON11
Mask bits: 0b=include bit in position matching, 1b= do not include this bit in position matching

10.4.4.49. DIR_SPG11PosOff

Register address	BaseAddress + $4BC_H$			
Bit number	31	30 - 16	15	14 - 0
Field name	SPGPSOFF_TOGGLE11	SPGPSOFF_X11	reservedoff11	SPGPSOFF_Y11
R/W	RW	RW	RW	RW
Reset value	0_H	0_H	0_H	0_H

Sync pulse generator 11, 'Switch off' position
Bit 31 SPGPSOFF_TOGGLE11 toggle enable: 0b=disable, 1b=enable
Bit 30 - 16 SPGPSOFF_X11 X scan position
Bit 15 reservedoff11 reserved (set to 0)
Bit 14 - 0 SPGPSOFF_Y11 Y scan position

10.4.4.50. DIR_SPG11MaskOff

Register address	BaseAddress + $4C0_H$																													
Bit number	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
Field name		SPGMKOFF11																														

R/W		RW
Reset value		0_H

Bit 30 - 0 SPGMKOFF11
Mask bits: 0b=include bit in position matching, 1b= do not include this bit in position matching

10.4.4.51. IR_SSqCycle

Register address	BaseAddress + $4C4_H$																													
Bit number	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
Field name																									SSQCYCLE							
R/W																									RW							
Reset value																									0_H							

Bit 5 - 0 SSQCYCLE
Sequencer cycle length (number-1) of sequencer cycles

10.4.4.52. DIR_SMx0Sigs

Register address	BaseAddress + $4C8_H$																													
Bit number	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
Field name																	SMX0SIGS_S4				SMX0SIGS_S3			SMX0SIGS_S2			SMX0SIGS_S1			SMX0SIGS_S0		
R/W																	RW				RW			RW			RW			RW		
Reset value																	0_H				0_H			0_H			0_H			0_H		

Sync mixer 0 signal selection
Bit 14 - 12 SMX0SIGS_S4
select 4 000b=const zero,001b=sync sequencer output, 010b...111b sync pulse generator output
Bit 11 - 9 SMX0SIGS_S3
select 3
Bit 8 - 6 SMX0SIGS_S2
select 2
Bit 5 - 3 SMX0SIGS_S1
select 1
Bit 2 - 0 SMX0SIGS_S0
select 0

10.4.4.53. DIR_SMx0FctTable

Register address	BaseAddress + $4CC_H$																													
Bit number	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
Field name	SMXFCT0																															
R/W	RW																															
Reset value	0_H																															

Sync mixer output = function table [a] $a = s_4*2^4+s_3*2^3+s_2*2^2+s_1*2^1+s_0*2^0$
Bit 31 - 0 SMXFCT0
Sync mixer 0 function table

10.4.4.54. DIR_SMx1Sigs

Register address	BaseAddress + 4D0H																															
Bit number	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
Field name																	SMX1SIGS_S4				SMX1SIGS_S3			SMX1SIGS_S2			SMX1SIGS_S1			SMX1SIGS_S0		
R/W																	RW				RW			RW			RW			RW		
Reset value																	0H				0H			0H			0H			0H		

Sync mixer 1 signal selection
Bit 14 - 12 SMX1SIGS_S4
select 4 000b=const zero,001b=sync sequencer output, 010b...111b sync pulse generator output
Bit 11 - 9 SMX1SIGS_S3
select 3
Bit 8 - 6 SMX1SIGS_S2
select 2
Bit 5 - 3 SMX1SIGS_S1
select 1
Bit 2 - 0 SMX1SIGS_S0
select 0

10.4.4.55. DIR_SMx1FctTable

Register address	BaseAddress + 4D4H																															
Bit number	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
Field name	SMXFCT1																															
R/W	RW																															
Reset value	0H																															

Sync mixer output = function table [a] $a = s_4*2^4+s_3*2^3+s_2*2^2+s_1*2^1+s_0*2^0$
Bit 31 - 0 SMXFCT1
Sync mixer 0 function table

10.4.4.56. DIR_SMx2Sigs

Register address	BaseAddress + 4D8H																															
Bit number	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
Field name																	SMX2SIGS_S4				SMX2SIGS_S3			SMX2SIGS_S2			SMX2SIGS_S1			SMX2SIGS_S0		
R/W																	RW				RW			RW			RW			RW		
Reset value																	0H				0H			0H			0H			0H		

Sync mixer 2 signal selection
Bit 14 - 12 SMX2SIGS_S4
select 4 000b=const zero,001b=sync sequencer output, 010b...111b sync pulse generator output
Bit 11 - 9 SMX2SIGS_S3
select 3
Bit 8 - 6 SMX2SIGS_S2
select 2
Bit 5 - 3 SMX2SIGS_S1
select 1
Bit 2 - 0 SMX2SIGS_S0
select 0

10.4.4.57. DIR_SMx2FctTable

Register address	BaseAddress + $4DC_H$																													
Bit number	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
Field name	SMXFCT2																															
R/W	RW																															
Reset value	0_H																															

Sync mixer output = function table [a] $a = s_4*2^4+s_3*2^3+s_2*2^2+s_1*2^1+s_0*2^0$
Bit 31 - 0 SMXFCT2
Sync mixer 0 function table

10.4.4.58. DIR_SMx3Sigs

Register address	BaseAddress + $4E0_H$																													
Bit number	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
Field name																	SMX3SIGS_S4				SMX3SIGS_S3			SMX3SIGS_S2			SMX3SIGS_S1			SMX3SIGS_S0		
R/W																	RW				RW			RW			RW			RW		
Reset value																	0_H				0_H			0_H			0_H			0_H		

Sync mixer 3 signal selection
Bit 14 - 12 SMX3SIGS_S4
select 4 000b=const zero,001b=sync sequencer output, 010b...111b sync pulse generator output
Bit 11 - 9 SMX3SIGS_S3
select 3
Bit 8 - 6 SMX3SIGS_S2
select 2
Bit 5 - 3 SMX3SIGS_S1
select 1
Bit 2 - 0 SMX3SIGS_S0
select 0

10.4.4.59. DIR_SMx3FctTable

Register address	BaseAddress + $4E4_H$																													
Bit number	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
Field name	SMXFCT3																															
R/W	RW																															
Reset value	0_H																															

Sync mixer output = function table [a] $a = s_4*2^4+s_3*2^3+s_2*2^2+s_1*2^1+s_0*2^0$
Bit 31 - 0 SMXFCT3
Sync mixer 0 function table

10.4.4.60. DIR_SMx4Sigs

Register address	BaseAddress + 4E8H																													
Bit number	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
Field name																	SMX4SIGS_S4				SMX4SIGS_S3			SMX4SIGS_S2			SMX4SIGS_S1			SMX4SIGS_S0		
R/W																	RW				RW			RW			RW			RW		
Reset value																	0H				0H			0H			0H			0H		

Sync mixer 4 signal selection
Bit 14 - 12 SMX4SIGS_S4
select 4 000b=const zero,001b=sync sequencer output, 010b...111b sync pulse generator output
Bit 11 - 9 SMX4SIGS_S3
select 3
Bit 8 - 6 SMX4SIGS_S2
select 2
Bit 5 - 3 SMX4SIGS_S1
select 1
Bit 2 - 0 SMX4SIGS_S0
select 0

10.4.4.61. DIR_SMx4FctTable

Register address	BaseAddress + 4ECH																													
Bit number	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
Field name	SMXFCT4																													
R/W	RW																													
Reset value	0H																													

Sync mixer output = function table [a] $a = s_4*2^4+s_3*2^3+s_2*2^2+s_1*2^1+s_0*2^0$
Bit 31 - 0 SMXFCT4
Sync mixer 0 function table

10.4.4.62. DIR_SMx5Sigs

Register address	BaseAddress + 4F0H																													
Bit number	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
Field name																	SMX5SIGS_S4				SMX5SIGS_S3			SMX5SIGS_S2			SMX5SIGS_S1			SMX5SIGS_S0		
R/W																	RW				RW			RW			RW			RW		
Reset value																	0H				0H			0H			0H			0H		

Sync mixer 5 signal selection
Bit 14 - 12 SMX5SIGS_S4
select 4 000b=const zero,001b=sync sequencer output, 010b...111b sync pulse generator output
Bit 11 - 9 SMX5SIGS_S3
select 3
Bit 8 - 6 SMX5SIGS_S2
select 2
Bit 5 - 3 SMX5SIGS_S1
select 1
Bit 2 - 0 SMX5SIGS_S0
select 0

10.4.4.63. DIR_SMx5FctTable

Register address	BaseAddress + $4F4_H$																													
Bit number	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
Field name	SMXFCT5																															
R/W	RW																															
Reset value	0_H																															

Sync mixer output = function table [a] $a = s_4*2^4+s_3*2^3+s_2*2^2+s_1*2^1+s_0*2^0$
Bit 31 - 0 SMXFCT5
Sync mixer 0 function table

10.4.4.64. DIR_SMx6Sigs

Register address	BaseAddress + $4F8_H$																													
Bit number	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
Field name																	SMX6SIGS_S4				SMX6SIGS_S3			SMX6SIGS_S2			SMX6SIGS_S1			SMX6SIGS_S0		
R/W																	RW				RW			RW			RW			RW		
Reset value																	0_H				0_H			0_H			0_H			0_H		

Sync mixer 6 signal selection
Bit 14 - 12 SMX6SIGS_S4
select 4 000b=const zero,001b=sync sequencer output, 010b...111b sync pulse generator output
Bit 11 - 9 SMX6SIGS_S3
select 3
Bit 8 - 6 SMX6SIGS_S2
select 2
Bit 5 - 3 SMX6SIGS_S1
select 1
Bit 2 - 0 SMX6SIGS_S0
select 0

10.4.4.65. DIR_SMx6FctTable

Register address	BaseAddress + $4FC_H$																													
Bit number	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
Field name	SMXFCT6																															
R/W	RW																															
Reset value	0_H																															

Sync mixer output = function table [a] $a = s_4*2^4+s_3*2^3+s_2*2^2+s_1*2^1+s_0*2^0$
Bit 31 - 0 SMXFCT6
Sync mixer 0 function table

10.4.4.66. DIR_SMx7Sigs

Register address	BaseAddress + 500_H																															
Bit number	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
Field name																		SMX7SIGS_S4			SMX7SIGS_S3			SMX7SIGS_S2			SMX7SIGS_S1			SMX7SIGS_S0		
R/W																		RW			RW			RW			RW			RW		
Reset value																		0_H			0_H			0_H			0_H			0_H		

Sync mixer 7 signal selection
Bit 14 - 12 SMX7SIGS_S4
select 4 000b=const zero,001b=sync sequencer output, 010b...111b sync pulse generator output
Bit 11 - 9 SMX7SIGS_S3
select 3
Bit 8 - 6 SMX7SIGS_S2
select 2
Bit 5 - 3 SMX7SIGS_S1
select 1
Bit 2 - 0 SMX7SIGS_S0
select 0

10.4.4.67. DIR_SMx7FctTable

Register address	BaseAddress + 504_H																															
Bit number	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
Field name	SMXFCT7																															
R/W	RW																															
Reset value	0_H																															

Sync mixer output = function table [a] $a = s_4*2^4+s_3*2^3+s_2*2^2+s_1*2^1+s_0*2^0$
Bit 31 - 0 SMXFCT7
Sync mixer 0 function table

10.4.4.68. DIR_SMx8Sigs

Register address	BaseAddress + 508_H																															
Bit number	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
Field name																		SMX8SIGS_S4			SMX8SIGS_S3			SMX8SIGS_S2			SMX8SIGS_S1			SMX8SIGS_S0		
R/W																		RW			RW			RW			RW			RW		
Reset value																		0_H			0_H			0_H			0_H			0_H		

Sync mixer 8 signal selection
Bit 14 - 12 SMX8SIGS_S4
select 4 000b=const zero,001b=sync sequencer output, 010b...111b sync pulse generator output
Bit 11 - 9 SMX8SIGS_S3
select 3
Bit 8 - 6 SMX8SIGS_S2
select 2
Bit 5 - 3 SMX8SIGS_S1
select 1
Bit 2 - 0 SMX8SIGS_S0
select 0

10.4.4.69. DIR_SMx8FctTable

Register address	BaseAddress + $50C_H$																													
Bit number	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
Field name	SMXFCT8																															
R/W	RW																															
Reset value	0_H																															

Sync mixer output = function table [a] $a = s_4*2^4+s_3*2^3+s_2*2^2+s_1*2^1+s_0*2^0$
Bit 31 - 0 SMXFCT8
Sync mixer 0 function table

10.4.4.70. DIR_SMx9Sigs

Register address	BaseAddress + 510_H																													
Bit number	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
Field name																	SMX9SIGS_S4				SMX9SIGS_S3			SMX9SIGS_S2			SMX9SIGS_S1			SMX9SIGS_S0		
R/W																	RW				RW			RW			RW			RW		
Reset value																	0_H				0_H			0_H			0_H			0_H		

Sync mixer 9 signal selection
Bit 14 - 12 SMX9SIGS_S4
select 4 000b=const zero,001b=sync sequencer output, 010b...111b sync pulse generator output
Bit 11 - 9 SMX9SIGS_S3
select 3
Bit 8 - 6 SMX9SIGS_S2
select 2
Bit 5 - 3 SMX9SIGS_S1
select 1
Bit 2 - 0 SMX9SIGS_S0
select 0

10.4.4.71. DIR_SMx9FctTable

Register address	BaseAddress + 514_H																													
Bit number	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
Field name	SMXFCT9																															
R/W	RW																															
Reset value	0_H																															

Sync mixer output = function table [a] $a = s_4*2^4+s_3*2^3+s_2*2^2+s_1*2^1+s_0*2^0$
Bit 31 - 0 SMXFCT9
Sync mixer 0 function table

10.4.4.72. DIR_SMx10Sigs

Register address	BaseAddress + 518_H																													
Bit number	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
Field name																	SMX10SIGS_S4				SMX10SIGS_S3			SMX10SIGS_S2			SMX10SIGS_S1			SMX10SIGS_S0		
R/W																	RW				RW			RW			RW			RW		
Reset value																	0_H				0_H			0_H			0_H			0_H		

Sync mixer 10 signal selection
Bit 14 - 12 SMX10SIGS_S4
select 4 000b=const zero,001b=sync sequencer output, 010b...111b sync pulse generator output
Bit 11 - 9 SMX10SIGS_S3
select 3
Bit 8 - 6 SMX10SIGS_S2
select 2
Bit 5 - 3 SMX10SIGS_S1
select 1
Bit 2 - 0 SMX10SIGS_S0
select 0

10.4.4.73. DIR_SMx10FctTable

Register address	BaseAddress + $51C_H$																													
Bit number	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
Field name	SMXFCT10																															
R/W	RW																															
Reset value	0_H																															

Sync mixer output = function table [a] $a = s_4*2^4+s_3*2^3+s_2*2^2+s_1*2^1+s_0*2^0$
Bit 31 - 0 SMXFCT10
Sync mixer 0 function table

10.4.4.74. DIR_SMx11Sigs

Register address	BaseAddress + 520_H																													
Bit number	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
Field name																	SMX11SIGS_S4				SMX11SIGS_S3			SMX11SIGS_S2			SMX11SIGS_S1			SMX11SIGS_S0		
R/W																	RW				RW			RW			RW			RW		
Reset value																	0_H				0_H			0_H			0_H			0_H		

Sync mixer 11 signal selection
Bit 14 - 12 SMX11SIGS_S4
select 4 000b=const zero,001b=sync sequencer output, 010b...111b sync pulse generator output
Bit 11 - 9 SMX11SIGS_S3
select 3
Bit 8 - 6 SMX11SIGS_S2
select 2
Bit 5 - 3 SMX11SIGS_S1
select 1
Bit 2 - 0 SMX11SIGS_S0
select 0

10.4.4.75. DIR_SMx11FctTable

Register address	BaseAddress + 524_H																													
Bit number	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
Field name	SMXFCT11																															
R/W	RW																															
Reset value	$FFFFFFFF_H$																															

Sync mixer output = function table [a] $a = s_4*2^4+s_3*2^3+s_2*2^2+s_1*2^1+s_0*2^0$

Bit 31 - 0 SMXFCT11
Sync mixer 0 function table

10.4.4.76. DIR_SSwitch

Register address	BaseAddress + 528H																													
Bit number	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
Field name																			InvCtrEn	SSWITCH											
R/W																			RW	RW											
Reset value																			0H	0H											

Sync switch

Bit 13 InvCtrEn
Enable for inversion control: 0b=disabled, 1b=enabled

Bit 12 - 0 SSWITCH
Delay selection for all TSIG outputs including. inversion control (bit 12) (0=none, 1=0.5 cycle delay of pixel clock)

10.4.4.77. DIR_RBM_CTRL

Register address	BaseAddress + $52C_H$																													
Bit number	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
Field name																					ColOrder				BitOrder		swapoddevenbit	BitPerCol	IfcType		Bypass
R/W																					RW				RW		RW	RW	RW		RW
Reset value																					0_H				0_H		0_H	0_H	0_H		1_H

RSDS Bitmap Control

Bit 10 - 8 ColOrder
Color Component Ordering: 000b=RGB, 001b=BRG 010b=GBR 011b=RBG 100b=GRB 101b=BGR 110b=reserved 111b=reserved

Bit 5 BitOrder
Bit Order Inversion: 0b=normal order (MSB 7 downto 0), 1b=inverted order (0 upto 7 MSB)

Bit 4 swapoddevenbit
ES1: Reserved, ES2: This field has only effect for ES2 and later: swap odd and even bits, 0b=no change, 1b=bit 6 and 7, 4 and 5, 2 and 3, 0 and 1 are swapped, This is needed for RSDS channel order inversion

Bit 3 BitPerCol
Bits per Colour: 0b=6bits (2 LSBs are set to '0'), 1b=8bits

Bit 2 - 1 IfcType
Interface protocol type: 00b=TTL, 01b=RSDS, 10b,11b=reserved

Bit 0 Bypass
Bypass module: 0b=bypass disable, 1b=bypass enable

10.4.4.78. DIR_IO_CTRL

Register address	BaseAddress + 530_H																													
Bit number	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
Field name																											BiasPc				DClevel
R/W																											RW				RW	
Reset value																											0_H				0_H	

IO Module Control
Bit 4 - 2 BiasPc
reserved, this bitfield has no effect (programmable bias current)
Bit 0 DClevel
reserved, this bitfield has no effect (DC level selection:, 0b=1.2V, 1b=1.3V)

10.4.4.79. DIR_PIN0_CTRL

Register address	BaseAddress + 534_H																													
Bit number	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
Field name												NChanSel0		ChanSel0		PC0	Susp0	NDelay0	Delay0					InOut0		NPolarity0	Polarity0	Mode0			Boost0	
R/W												RW		RW		RW	RW	RW	RW					RW		RW	RW	RW			RW	
Reset value												0_H		0_H		1_H	1_H	0_H	0_H					0_H		0_H	0_H	0_H			0_H	

IO Module Pad 0 Control
Bit 20 - 19 NChanSel0
Channel selection for N-Pin of Pad i=0 TTL: 00b=channel(i*2+1), 01b=channel(i*2), 10b=clk, 11b=const0 (TTL mode only)
Bit 18 - 17 ChanSel0
Channel selection for Pad i=0: for RSDS: 00b=channel i, 01b=reserved, 10b=clk, 11b=const0, for TTL: 00b=channel i*2, 01b=INV (from inversion control function), 10b=clk, 11b=const0
Bit 16 PC0
Pad 0 RSDS Biason, enable internal bias generator 0b=off, 1b= on (this bit has effect only for RSDS mode, switch it off only for deep power down mode!
Bit 15 Susp0
Pad 0 Normal Power Down: 0b=enable RSDS output buffer, 1b=disable RSDS output buffer (high impedance, for this set 'Mode' = differential) (this bit has only effect for RSDS mode!)
Bit 14 NDelay0
N-pin Padcell 0 delay: 0b=no delay, 1b= half bit clock cycle delay (TTL-mode only)
Bit 13 Delay0
Pad 0 delay: 0b=no delay, 1b= half bit clock cycle delay
Bit 7 InOut0
reserved, this bit has no effect
Bit 6 NPolarity0
N-pin of Padcell 0 drive polarity: 0b=normal, 1b=inverted
Bit 5 Polarity0
Pad 0 drive polarity: 0b=normal, 1b=inverted
Bit 4 Mode0
Pad 0 drive mode: 0b=differential, 1b=TTL
Bit 1 - 0 Boost0
Boost factor for drive current: x0b=min, x1b=max (only boosti[0] has effect)

10.4.4.80. DIR_PIN1_CTRL

Register address	BaseAddress + 538H																													
Bit number	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
Field name												NChanSel1		ChanSel1		PC1	Susp1	NDelay1	Delay1					InOut1		NPolarity1	Polarity1	Mode1				Boost1
R/W												RW		RW		RW	RW	RW	RW					RW		RW	RW	RW				RW
Reset value												0H		0H		1H	1H	0H	0H					0H		0H	0H	0H				0H

IO Module Pad 1 Control

Bit 20 - 19 NChanSel1
Channel selection for N-Pin of Pad i=1 TTL: 00b=channel(i*2+1), 01b=channel(i*2), 10b=clk, 11b=const0 (TTL mode only)

Bit 18 - 17 ChanSel1
Channel selection for Pad i=1: for RSDS: 00b=channel i, 01b=channel(i-1), 10b=clk, 11b=const0, for TTL : 00b=channel i*2, 01b=channel i*2-1, 10b=clk, 11b=const0

Bit 16 PC1
Pad 1 RSDS Biason, enable internal bias generator 0b=off, 1b= on (this bit has effect only for RSDS mode, switch it off only for deep power down mode!)

Bit 15 Susp1
Pad 1 Normal Power Down: 0b=enable RSDS output buffer, 1b=disable RSDS output buffer (high impedance, for this set 'Mode' = differential) (this bit has only effect for RSDS mode!)

Bit 14 NDelay1
N-pin Padcell 1 delay: 0b=no delay, 1b= half bitclock cycle delay (TTL-mode only)

Bit 13 Delay1
Pad 1 delay: 0b=no delay, 1b= half bit clock cycle delay

Bit 7 InOut1
reserved, this bit has no effect

Bit 6 NPolarity1
N-pin of Padcell 1 drive polarity: 0=normal, 1b=inverted

Bit 5 Polarity1
Pad 1 drive polarity: 0=normal, 1b=inverted

Bit 4 Mode1
Pad 1 drive mode: 0b=differential, 1b=TTL

Bit 1 - 0 Boost1
Boost factor for drive current: x0b=min, x1b=max (only boosti[0] has effect)

10.4.4.81. DIR_PIN2_CTRL

Register address	BaseAddress + $53C_H$																															
Bit number	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
Field name												NChanSel2		ChanSel2		PC2	Susp2	NDelay2	Delay2					InOut2	NPolarity2	Polarity2	Mode2				Boost2
R/W												RW		RW		RW	RW	RW	RW					RW	RW	RW	RW				RW
Reset value												0_H		0_H		1_H	1_H	0_H	0_H					0_H	0_H	0_H	0_H				0_H

IO Module Pad 2 Control

Bit 20 - 19 NChanSel2
Channel selection for N-Pin of Pad i=2 TTL: 00b=channel(i*2+1), 01b=channel(i*2), 10b=clk, 11b=const0 (TTL mode only)

Bit 18 - 17 ChanSel2
Channel selection for Pad i=2 for RSDS: 00b=channel i, 01b=channel(i-1), 10b=clk, 11b=const0, for TTL: 00b=channel i*2, 01b=channel i*2-1, 10b=clk, 1b=const0

Bit 16 PC2
Pad 2 RSDS Biason, enable internal bias generator 0b=off, 1b= on (this bit has effect only for RSDS mode, switch it off only for deep power down mode!)

Bit 15 Susp2
Pad 2 Normal Power Down: 0b=enable RSDS output buffer, 1b=disable RSDS output buffer (high impedance, for this set 'Mode' = differential) (this bit has only effect for RSDS mode!)

Bit 14 NDelay2
N-pin Padcell 2 delay: 0b=no delay, 1b= half bit clock cycle delay (TTL-mode only)

Bit 13 Delay2
Pad 2 delay, 0b=no delay: 1b= half bit clock cycle delay

Bit 7 InOut2
reserved, this bit has no effect

Bit 6 NPolarity2
N-pin of Padcell 2 drive polarity: 0=normal, 1b=inverted

Bit 5 Polarity2
Pad 2 drive polarity: 0=normal, 1b=inverted

Bit 4 Mode2
Pad 2 drive mode: 0b=differential, 1b=TTL

Bit 1 - 0 Boost2
Boost factor for drive current: x0b=min, x1b=max (only boosti[0] has effect)

10.4.4.82. DIR_PIN3_CTRL

Register address	BaseAddress + 540H																															
Bit number	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
Field name												NChanSel3		ChanSel3		PC3	Susp3	NDelay3	Delay3					InOut3	NPolarity3		Polarity3	Mode3				Boost3
R/W												RW		RW		RW	RW	RW	RW					RW	RW		RW	RW				RW
Reset value												0H		0H		1H	1H	0H	0H					0H	0H		0H	0H				0H

IO Module Pad 3 Control

Bit 20 - 19 NChanSel3
Channel selection for N-Pin of Pad i=3 TTL: 00b=channel(i*2+1), 01b=channel(i*2), 10b=clk, 11b=const0 (TTL mode only)

Bit 18 - 17 ChanSel3
Channel selection for Pad i=3 for RSDS: 00b=channel i, 01b=channel(i-1), 10b=clk, 11b=const0, for TTL: 00b=channel i*2, 01b=channel i*2-1, 10b=clk, 11b=const0

Bit 16 PC3
Pad 3 RSDS Biason, enable internal bias generator 0b=off, 1b= on (this bit has effect only for RSDS mode, switch it off only for deep power down mode!)

Bit 15 Susp3
Pad 3 Normal Power Down: 0b=enable RSDS output buffer, 1b=disable RSDS output buffer (high impedance, for this set 'Mode' = differential) (this bit has only effect for RSDS mode!)

Bit 14 NDelay3
N-pin Padcell 3 delay: 0b=no delay, 1b= half bit clock cycle delay (TTL-mode only)

Bit 13 Delay3
Pad 3 delay: 0b=no delay, 1b= half bit clock cycle delay

Bit 7 InOut3
reserved, this bit has no effect

Bit 6 NPolarity3
N-pin of Padcell 3 drive polarity: 0=normal, 1b=inverted

Bit 5 Polarity3
Pad 3 drive polarity: 0=normal, 1b=inverted

Bit 4 Mode3
Pad 3 drive mode: 0b=differential, 1b=TTL

Bit 1 - 0 Boost3
Boost factor for drive current: x0b=min, x1b=max (only boosti[0] has effect)

10.4.4.83. DIR_PIN4_CTRL

Register address	BaseAddress + 544H																															
Bit number	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
Field name												NChanSel4		ChanSel4		PC4	Susp4	NDelay4	Delay4					InOut4	NPolarity4		Polarity4	Mode4				Boost4
R/W												RW		RW		RW	RW	RW	RW					RW	RW		RW	RW				RW
Reset value												0H		0H		1H	1H	0H	0H					0H	0H		0H	0H				0H

IO Module Pad 4 Control

Bit 20 - 19 NChanSel4
Channel selection for N-Pin of Pad i=4 TTL: 00b=channel(i*2+1), 01b=channel(i*2), 10b=clk, 11b=const0 (TTL mode only)

Bit 18 - 17 ChanSel4
Channel selection for Pad i=4 for RSDS: 00b=channel i, 01b=channel(i-1), 10b=clk, 11b=const0, for TTL: 00b=channel i*2, 01b=channel i*2-1, 10b=clk, 11b=const0

Bit 16 PC4
Pad 4 RSDS Biason, enable internal bias generator 0b=off, 1b= on (this bit has effect only for RSDS mode, switch it off only for deep power down mode!)

Bit 15 Susp4
Pad 4 Normal Power Down: 0b=enable RSDS output buffer, 1b=disable RSDS output buffer (high impedance, for this set 'Mode' = differential) (this bit has only effect for RSDS mode!)

Bit 14 NDelay4
N-pin Padcell 4 delay: 0b=no delay, 1b= half bit clock cycle delay (TTL-mode only)

Bit 13 Delay4
Pad 4 delay: 0b=no delay, 1b= half bit clock cycle delay

Bit 7 InOut4
reserved, this bit has no effect

Bit 6 NPolarity4
N-pin of Padcell 4 drive polarity: 0=normal, 1b=inverted

Bit 5 Polarity4
Pad 4 drive polarity: 0=normal, 1b=inverted

Bit 4 Mode4
Pad 4 drive mode: 0b=differential, 1b=TTL

Bit 1 - 0 Boost4
Boost factor for drive current: x0b=min, x1b=max (only boosti[0] has effect)

10.4.4.84. DIR_PIN5_CTRL

Register address	BaseAddress + 548H																															
Bit number	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
Field name												NChanSel5		ChanSel5		PC5	Susp5	NDelay5	Delay5					InOut5	NPolarity5	Polarity5	Mode5				Boost5	
R/W												RW		RW		RW	RW	RW	RW					RW	RW	RW	RW				RW	
Reset value												0H		0H		1H	1H	0H	0H					0H	0H	0H	0H				0H	

IO Module Pad 5 Control

Bit 20 - 19 NChanSel5
Channel selection for N-Pin of Pad i=5 TTL: 00b=channel(i*2+1), 01b=channel(i*2), 10b=clk, 11b=const0 (TTL mode only)

Bit 18 - 17 ChanSel5
Channel selection for Pad i=5 for RSDS: 00b=channel i, 01b=channel(i-1), 10b=clk, 11b=const0, for TTL: 00b=channel i*2, 01b=channel i*2-1, 10b=clk, 11b=const0

Bit 16 PC5
Pad 5 RSDS Biason, enable internal bias generator 0b=off, 1b= on (this bit has effect only for RSDS mode, switch it off only for deep power down mode!)

Bit 15 Susp5
Pad 5 Normal Power Down: 0b=enable RSDS output buffer, 1b=disable RSDS output buffer (high impedance, for this set 'Mode' = differential) (this bit has only effect for RSDS mode!)

Bit 14 NDelay5
N-pin Padcell 5 delay: 0b=no delay, 1b= half bit clock cycle delay (TTL-mode only)

Bit 13 Delay5
Pad 5 delay: 0b=no delay, 1b= half bit clock cycle delay

Bit 7 InOut5
reserved, this bit has no effect

Bit 6 NPolarity5
N-pin of Padcell 5 drive polarity: 0=normal, 1b=inverted

Bit 5 Polarity5
Pad 5 drive polarity: 0=normal, 1b=inverted

Bit 4 Mode5
Pad 5 drive mode: 0b=differential, 1b=TTL

Bit 1 - 0 Boost5
Boost factor for drive current: x0b=min, x1b=max (only boosti[0] has effect)

10.4.4.85. DIR_PIN6_CTRL

Register address	BaseAddress + 54C$_H$																													
Bit number	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
Field name												NChanSel6		ChanSel6		PC6	Susp6	NDelay6	Delay6					InOut6	NPolarity6	Polarity6	Mode6				Boost6
R/W												RW		RW		RW	RW	RW	RW					RW	RW	RW	RW				RW
Reset value												0$_H$		0$_H$		1$_H$	1$_H$	0$_H$	0$_H$					0$_H$	0$_H$	0$_H$	0$_H$				0$_H$

IO Module Pad 6 Control

Bit 20 - 19 NChanSel6
Channel selection for N-Pin of Pad i=6 TTL : 00b=channel(i*2+1), 01b=channel(i*2), 10b=clk, 11b=const0 (TTL mode only)

Bit 18 - 17 ChanSel6
Channel selection for Pad i=6 for RSDS: 00b=channel i, 01b=channel(i-1), 10b=clk, 11b=const0, for TTL: 00b=channel i*2, 01b=channel i*2-1, 10b=clk, 11b=const0

Bit 16 PC6
Pad 6 RSDS Biason, enable internal bias generator 0b=off, 1b= on (this bit has effect only for RSDS mode, switch it off only for deep power down mode!)

Bit 15 Susp6
Pad 6 Normal Power Down: 0b=enable RSDS output buffer, 1b=disable RSDS output buffer (high impedance, for this set 'Mode' = differential) (this bit has only effect for RSDS mode!)

Bit 14 NDelay6
N-pin Padcell 6 delay: 0b=no delay, 1b= half bit clock cycle delay (TTL-mode only)

Bit 13 Delay6
Pad 6 delay, 0b=no delay: 1b= half bit clock cycle delay

Bit 7 InOut6
reserved, this bit has no effect

Bit 6 NPolarity6
N-pin of Padcell 6 drive polarity: 0=normal, 1b=inverted

Bit 5 Polarity6
Pad 6 drive polarity: 0=normal, 1b=inverted

Bit 4 Mode6
Pad 6 drive mode: 0b=differential, 1b=TTL

Bit 1 - 0 Boost6
Boost factor for drive current: x0b=min, x1b=max (only boosti[0] has effect)

10.4.4.86. DIR_PIN7_CTRL

Register address	BaseAddress + 550H																													
Bit number	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
Field name												NChanSel7		ChanSel7		PC7	Susp7	NDelay7	Delay7					InOut7	NPolarity7	Polarity7	Mode7			Boost7	
R/W												RW		RW		RW	RW	RW	RW					RW	RW	RW	RW			RW	
Reset value												0H		0H		1H	1H	0H	0H					0H	0H	0H	0H			0H	

IO Module Pad 7 Control

Bit 20 - 19 NChanSel7
Channel selection for N-Pin of Pad i=7 TTL: 00b=channel(i*2+1), 01b=channel(i*2), 10b=clk, 11b=const0 (TTL mode only)

Bit 18 - 17 ChanSel7
Channel selection for Pad i=7 for RSDS: 00b=channel i, 01b=channel(i-1), 10b=clk, 11b=const0, for TTL: 00b=channel i*2, 01b=channel i*2-1, 10b=clk, 11b=const0

Bit 16 PC7
Pad 7 RSDS Biason, enable internal bias generator 0b=off, 1b= on (this bit has effect only for RSDS mode, switch it off only for deep power down mode!)

Bit 15 Susp7
Pad 7 Normal Power Down: 0b=enable RSDS output buffer, 1b=disable RSDS output buffer (high impedance, for this set 'Mode' = differential) (this bit has only effect for RSDS mode!)

Bit 14 NDelay7
N-pin Padcell 7 delay: 0b=no delay, 1b= half bit clock cycle delay (TTL-mode only)

Bit 13 Delay7
Pad 7 delay, 0b=no delay: 1b= half bit clock cycle delay

Bit 7 InOut7
reserved, this bit has no effect

Bit 6 NPolarity7
N-pin of Padcell 7 drive polarity: 0=normal, 1b=inverted

Bit 5 Polarity7
Pad 7 drive polarity: 0=normal, 1b=inverted

Bit 4 Mode7
Pad 7 drive mode: 0b=differential, 1b=TTL

Bit 1 - 0 Boost7
Boost factor for drive current: x0b=min, x1b=max (only boosti[0] has effect)

10.4.4.87. DIR_PIN8_CTRL

Register address	BaseAddress + 554H																													
Bit number	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
Field name												NChanSel8		ChanSel8		PC8	Susp8	NDelay8	Delay8					InOut8	NPolarity8	Polarity8	Mode8				Boost8
R/W												RW		RW		RW	RW	RW	RW					RW	RW	RW	RW				RW
Reset value												0H		0H		1H	1H	0H	0H					0H	0H	0H	0H				0H

IO Module Pad 8 Control

Bit 20 - 19 NChanSel8
Channel selection for N-Pin of Pad i=8 TTL: 00b=channel(i*2+1), 01b=channel(i*2), 10b=clk, 11b=const0 (TTL mode only)

Bit 18 - 17 ChanSel8
Channel selection for Pad i=8 for RSDS: 00b=channel i, 01b=channel(i-1), 10b=clk, 11b=const0, for TTL: 00b=channel i*2, 01b=channel i*2-1, 10b=clk, 11b=const0

Bit 16 PC8
Pad 8 RSDS Biason, enable internal bias generator 0b=off, 1b= on (this bit has effect only for RSDS mode, switch it off only for deep power down mode!)

Bit 15 Susp8
Pad 8 Normal Power Down: 0b=enable RSDS output buffer, 1b=disable RSDS output buffer (high impedance, for this set 'Mode' = differential) (this bit has only effect for RSDS mode!)

Bit 14 NDelay8
N-pin Padcell 8 delay: 0b=no delay, 1b= half bit clock cycle delay (TTL-mode only)

Bit 13 Delay8
Pad 8 delay: 0b=no delay: 1b= half bit clock cycle delay

Bit 7 InOut8
reserved, this bit has no effect

Bit 6 NPolarity8
N-pin of Padcell 8 drive polarity: 0=normal, 1b=inverted

Bit 5 Polarity8
Pad 8 drive polarity: 0=normal, 1b=inverted

Bit 4 Mode8
Pad 8 drive mode: 0b=differential, 1b=TTL

Bit 1 - 0 Boost8
Boost factor for drive current: x0b=min, x1b=max (only boosti[0] has effect)

10.4.4.88. DIR_PIN9_CTRL

Register address	BaseAddress + 558H																															
Bit number	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
Field name												NChanSel9		ChanSel9		PC9	Susp9	NDelay9	Delay9					InOut9		NPolarity9		Polarity9	Mode9			Boost9
R/W												RW		RW		RW	RW	RW	RW					RW		RW		RW	RW			RW
Reset value												0H		0H		1H	1H	0H	0H					0H		0H		0H	0H			0H

IO Module Pad 9 Control

Bit 20 - 19 NChanSel9
Channel selection for N-Pin of Pad i=9 TTL: 00b=channel(i*2+1)(reserved for 6bit/color!), 01b=channel(i*2)(reserved for 6bit/color!), 10b=clk, 11b=const0 (TTL mode only)

Bit 18 - 17 ChanSel9
Channel selection for Pad i=9 for RSDS: 00b=channel i(reserved for 6bit/color!), 01b=channel(i-1), 10b=clk, 11b=const0, for TTL : 00b=channel i*2(reserved for 6bit/color!), 01b=channel i*2-1, 10b=clk, 11b=const0

Bit 16 PC9
Pad 9 RSDS Biason, enable internal bias generator 0b=off, 1b= on (this bit has effect only for RSDS mode, switch it off only for deep power down mode!)

Bit 15 Susp9
Pad 9 Normal Power Down: 0b=enable RSDS output buffer, 1b=disable RSDS output buffer (high impedance, for this set 'Mode' = differential) (this bit has only effect for RSDS mode!)

Bit 14 NDelay9
N-pin Padcell 9 delay: 0b=no delay, 1b= half bit clock cycle delay (TTL-mode only)

Bit 13 Delay9
Pad 9 delay: 0b=no delay, 1b= half bit clock cycle delay

Bit 7 InOut9
reserved, this bit has no effect

Bit 6 NPolarity9
N-pin of Padcell 9 drive polarity: 0=normal, 1b=inverted

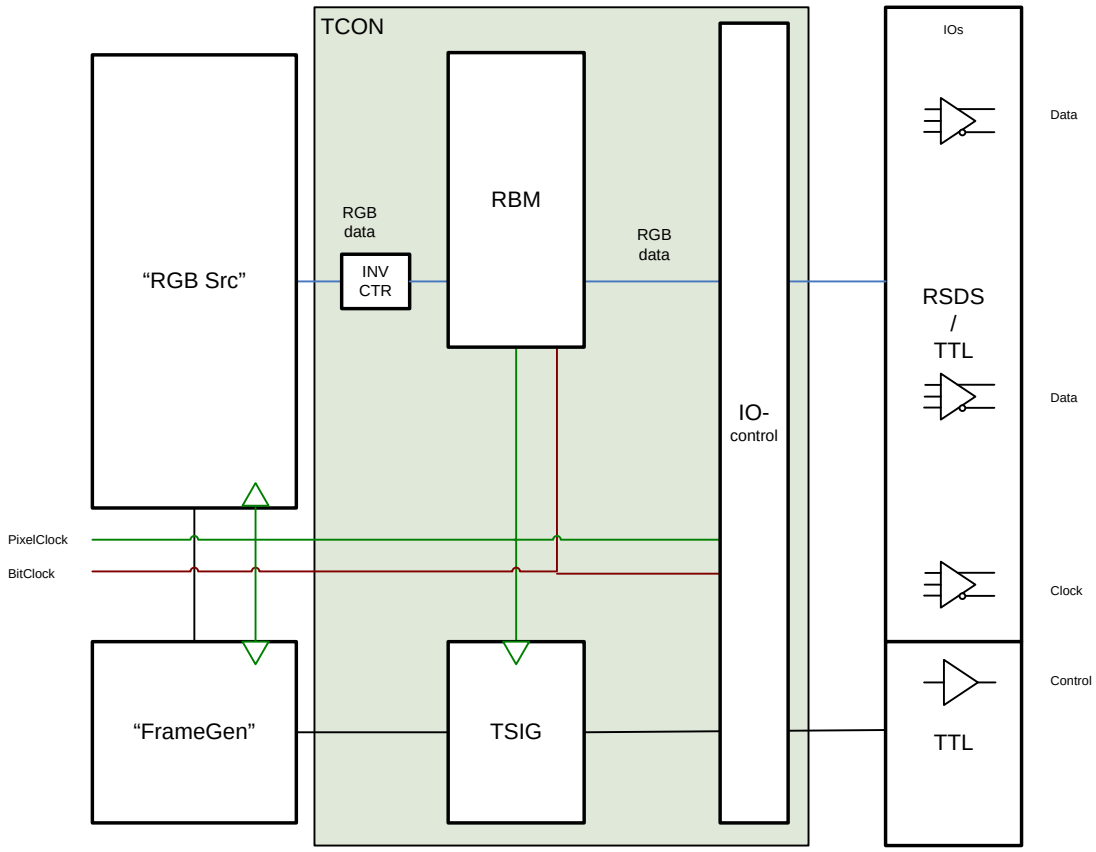
Bit 5 Polarity9
Pad 9 drive polarity: 0=normal, 1b=inverted

Bit 4 Mode9
Pad 9 drive mode, 0b=differential: 1b=TTL

Bit 1 - 0 Boost9
Boost factor for drive current: x0b=min, x1b=max (only boosti[0] has effect)

10.5. Processing Mode

10.5.1. Processing Flow



10.5.2. Processing Algorithm

10.5.2.1. Operation Modes

The TCON module is either active or in bypass mode (Register RBM_DIR_CTRL.bypass). In bypass mode, the RGB data from the RGB source (e.g. the Indigo-L video display controller) is transmitted unchanged through the RBM submodule. Additional 3 timing signals (HSYNC, VSYNC, DE) from the frame generator (submodule timing controller of Indigo-L video display controller) are bypassed to the TSIG output signals TSIG[0:2]. The RGB data and the 3 timing signals have the same latency.

10.5.2.2. SW Reset

The software reset is invoked by writing to its register. The software reset synchronizes all internal states. After power on reset the TCON module remains in this "sw reset active" state.
It is deasserted by internal logic synchronous to internal video synchronization signals, that means last pixel of video frame (inclusive blanking). After configuring TCON it is therefore necessary to setup the video frame (HTP, VTP, HDP, VDP) in module DISP to provide a valid video frame to TCON module.

Otherwise no RGB and display clock data is output.
Configuration registers are not reseted by SW reset, only internal states of TCON.

10.5.2.3. RSDS Bitmap module (RBM)

10.5.2.3.1. Block Diagram

The following block diagram shows the functional design of the RBM module.

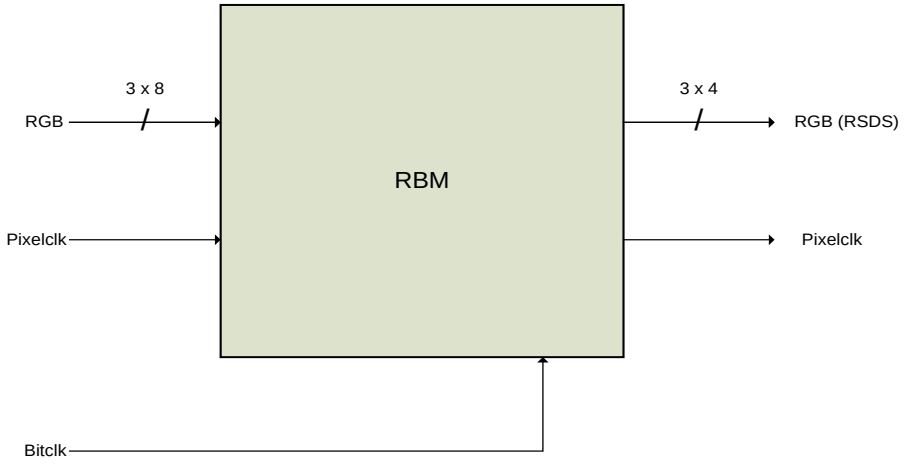


Figure 10-2 Block diagram of RBM

10.5.2.3.2. Bit Mapping

RSDS 8bpc	Rising	Falling
Ch0	R0	R1
Ch1	R2	R3
Ch2	R4	R5
Ch3	R6	R7
Ch4	G0	G1
Ch5	G2	G3
Ch6	G4	G5
Ch7	G6	G7
Ch8	B0	B1
Ch9	B2	B3
Ch10	B4	B5
Ch11	B6	B7

Table 10-1 Bitmapping RSDS 8bpc (for internal processing only! Output is still 6 bits per pixel)

RSDS 6bpc	Rising	Falling
Ch0	R2	R3
Ch1	R4	R5
Ch2	R6	R7
Ch3	G2	G3
Ch4	G4	G5
Ch5	G6	G7
Ch6	B2	B3
Ch7	B4	B5
Ch8	B6	B7
Ch9	0	0
Ch10	0	0
Ch11	0	0

Table 10-2 Bitmapping RSDS 6bpc

TTL 8bpc	Rising	Falling
Ch0	R0	R0
Ch1	R1	R1
Ch2	R2	R2
Ch3	R3	R3
Ch4	R4	R4
Ch5	R5	R5
Ch6	R6	R6
Ch7	R7	R7
Ch8	G0	G0
Ch9	G1	G1
Ch10	G2	G2
Ch11	G3	G3
Ch12	G4	G4
Ch13	G5	G5
Ch14	G6	G6
Ch15	G7	G7
Ch16	B0	B0
Ch17	B1	B1
Ch18	B2	B2
Ch19	B3	B3
Ch20	B4	B4
Ch21	B5	B5
Ch22	B6	B6
Ch23	B7	B7

Table 10-3 Bitmapping TTL 8bpc (for internal processing only! Output is still 6 bits per pixel)

TTL 6bpc	Rising	Falling
Ch0	R2	R2
Ch1	R3	R3
Ch2	R4	R4
Ch3	R5	R5
Ch4	R6	R6
Ch5	R7	R7
Ch6	G2	G2
Ch7	G3	G3
Ch8	G4	G4
Ch9	G5	G5
Ch10	G6	G6
Ch11	G7	G7
Ch12	B2	B2
Ch13	B3	B3
Ch14	B4	B4
Ch15	B5	B5
Ch16	B6	B6
Ch17	B7	B7
Ch18	0	0
Ch19	0	0
Ch20	0	0
Ch21	0	0
Ch22	0	0
Ch23	0	0

Table 10-4 Bitmapping TTL 6bpc

10.5.2.3.3. Bitmapping Appnote

To achieve a RSDS channel order inversion per color the following setup is needed:

DIR_RBM_CTRL.BitOrder = 1
DIR_RBM_CTRL. Swapoddevenbit = 1
DIR_RBM_CTRL.IfcType = 01

RSDS 8bpc	Rising	Falling
Ch0	R6	R7
Ch1	R4	R5
Ch2	R2	R3
Ch3	R0	R1
Ch4	G6	G7
Ch5	G4	G5
Ch6	G2	G3
Ch7	G0	G1
Ch8	B6	B7
Ch9	B4	B5
Ch10	B2	B3
Ch11	B0	B1

Table 10-5 Bitmapping RSDS 8bpc inverted RSDS channel order

RSDS 6bpc	Rising	Falling
Ch0	R6	R7
Ch1	R4	R5
Ch2	R2	R3
Ch3	G6	G7
Ch4	G4	G5
Ch5	G2	G3
Ch6	B6	B7
Ch7	B4	B5
Ch8	B2	B3
Ch9	0	0
Ch10	0	0
Ch11	0	0

Table 10-6 Bitmapping RSDS 6bpc inverted RSDS channel order

10.5.2.4. TSIG Module

10.5.2.4.1. Block Diagram

The following block diagram shows the functional design of the TSIG module (note the stages).

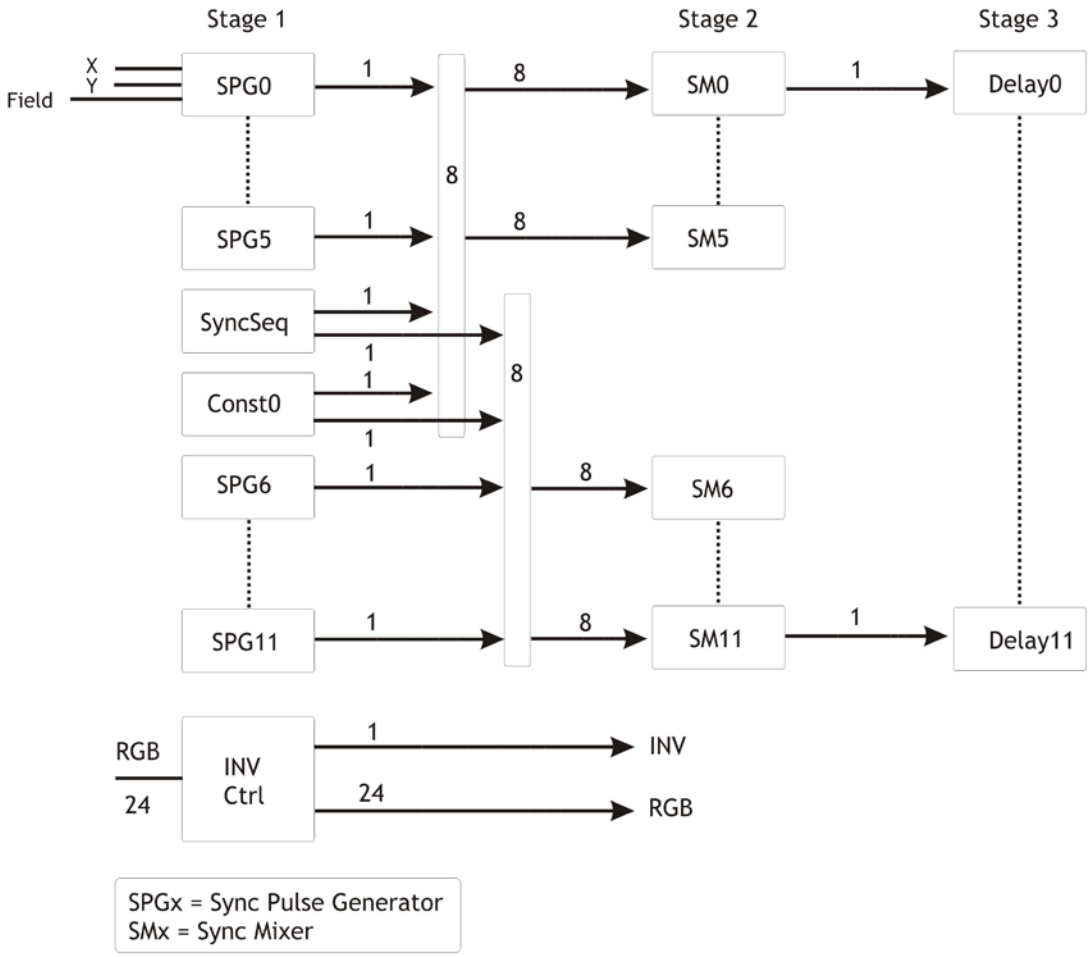


Figure 10-3 Block diagram of TSIG

10.5.2.4.2. Overview

Sync signals are generated using a three stage approach in order to achieve maximum flexibility. In the first stage, signals are generated which carry positional timing information. Two methods are used to create these signals. The second stage combines them to form more complex waveforms. The third stage is used to create a programmable delay of half a pixel clock cycle.

10.5.2.4.3. Position Matching

One way to form the first stage signals is to use simple position matching to trigger an RS flip-flop or a toggle flip-flop. This is done using an array of twelve identical Sync Pulse Generators (SPG's). The following diagram shows the working principle. For progressive-only systems like Indigo-L the bit „F" (odd-even frame flag) must always be set to '0'.

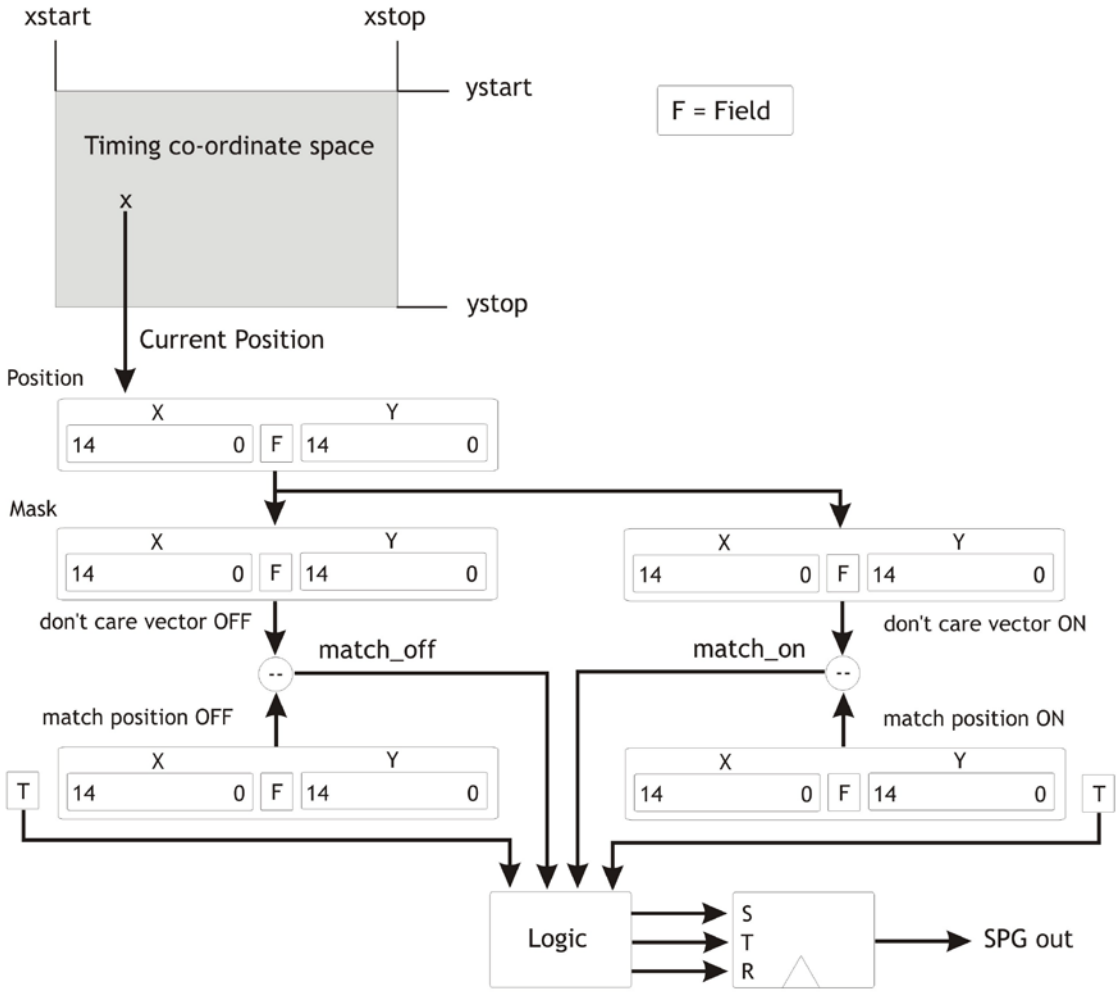


Figure 10-4 Matching position with sync pulse generators

TOGGLE_MODE = OFF:
The output of a sync pulse generator is set or reset if the current position equals the respective programmable position in all bits for which its don't-care-vector (which is also programmable) contains zeros. The Off matching is dominant, i.e. when both On and Off positions are matched at the same time, the output of the sync pulse generator is reset.

TOGGLE_MODE = ON:
The output of a sync pulse generator toggles if the current position equals the respective programmable position in all bits for which its don't-care-vector (which is also programmable) contains zeros. Toggle mode allows e.g. frame wise toggling signals. Set/Reset overrides toggle, and if both positions match and toggle, they cancel each other out.

10.5.2.4.4. Sequence Matching

A more sophisticated and powerful approach to creating first-stage signals is the use of a sequencer RAM to match a whole sequence of positions. The following diagram shows the principle of operation. A sync sequencer (SyncSeq) follows an arbitrary sequence of timing positions and generates an appropriate output signal. The length of the sequence as well as the contents of the RAM, consisting of the position and the assigned output value are programmable.

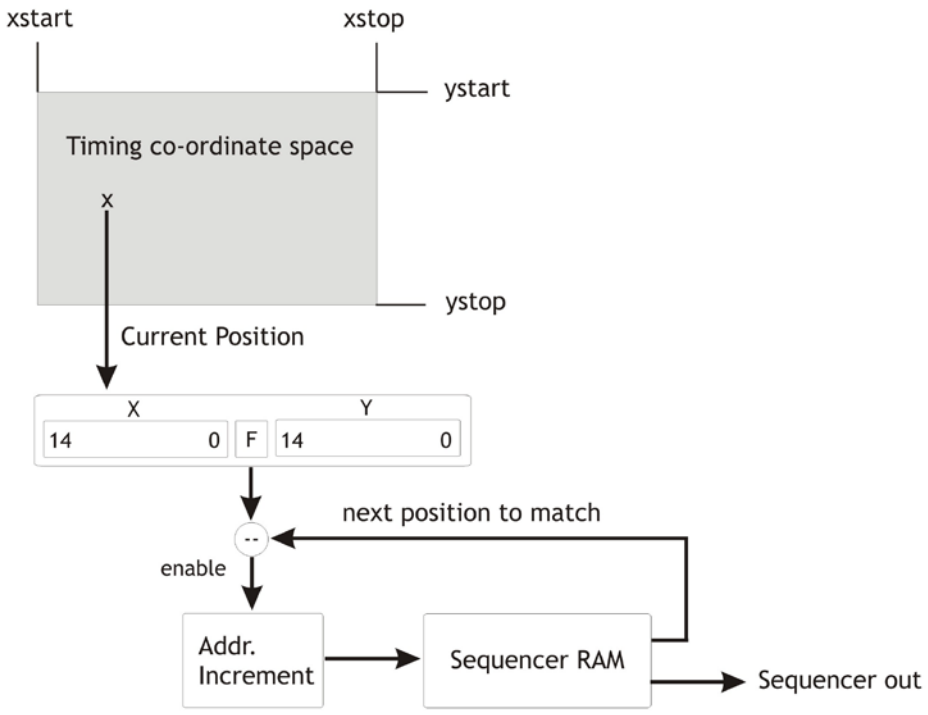


Figure 10-5 Matching whole sequences with the Sync Sequencer

Operation is as follows. To start, the address counter is reset to zero and the RAM outputs the first position that matches and the output value for this position. If the comparator signals match, the RAM address is incremented, the preset output value (bit 31) is propagated and the RAM then outputs the next position to match.

This match/address increment cycle continues until the programmed sequence length is reached. If the last position is matched, the address counter is reset to zero again and the cycle starts again. It is thus possible to generate arbitrarily complex waveforms with up to 64 edges (which is the maximum sequence length).

10.5.2.4.5. Co-ordinate space

The coordinate space for position matching of the sync pulse generators and the sync sequencer is shown at Figure10-6. The coordinate reference point (0,0) is determined by the VDC (Video Display Controller) module horizontal sync signal falling edge and the first active pixel line. For definition of all parameters please refer to chapter 9.3.2 “Display Parameters”.

For Indigo-L the allowed ranges are therefore:
$0 \leq X \leq HTP$ (horizontal total pixels)
$0 \leq Y \leq VTR$ (vertical total raster)

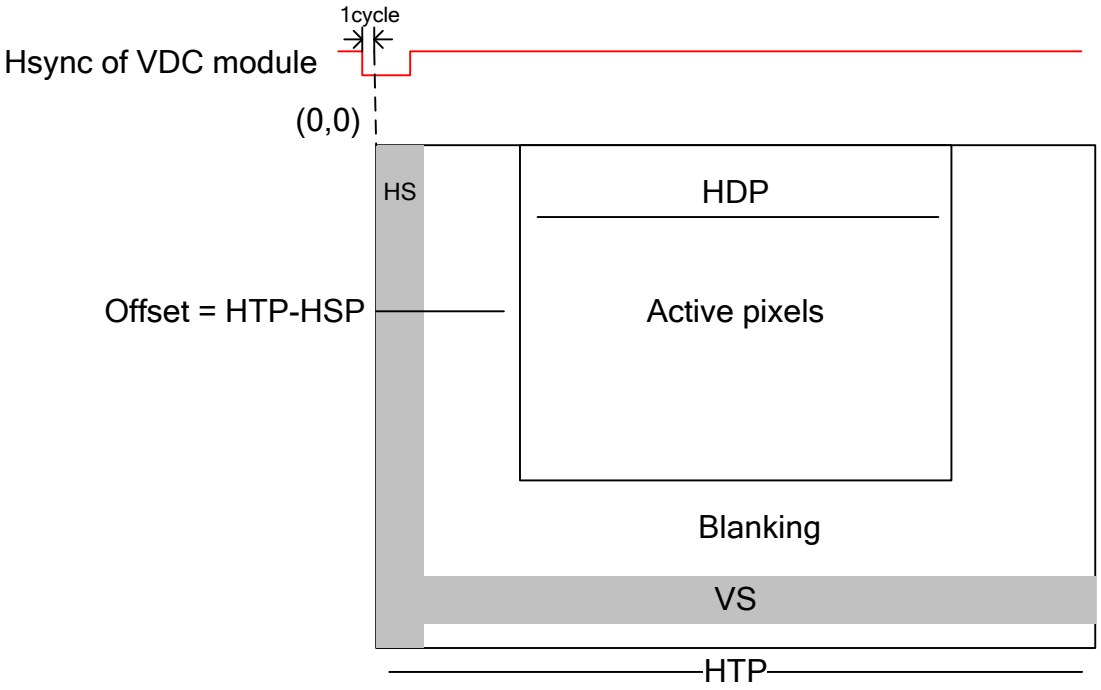


Figure 10-6 Coordinate System

Note: If VDC module is operated in external synchronization mode, the number of HTP pixels may vary. HS pulse of VDC is synchronized to external Sync signal.
Note: The coordinate reference point(0,0) is after one cycle(pixel clock) of the falling edge of Hsync of VDC module.

10.5.2.4.6. Combining First Stage Sync Signals

As shown above, there are twelve sync pulse generator outputs and one sync sequencer output. To obtain more complex waveforms, these signals can be combined in a second stage. Here, an array of twelve sync mixers (SMx) is used to calculate Boolean functions of first-stage signals. Each sync mixer can form any Boolean function on up to five inputs. The basic structure of one such mixer is depicted in the following diagram.

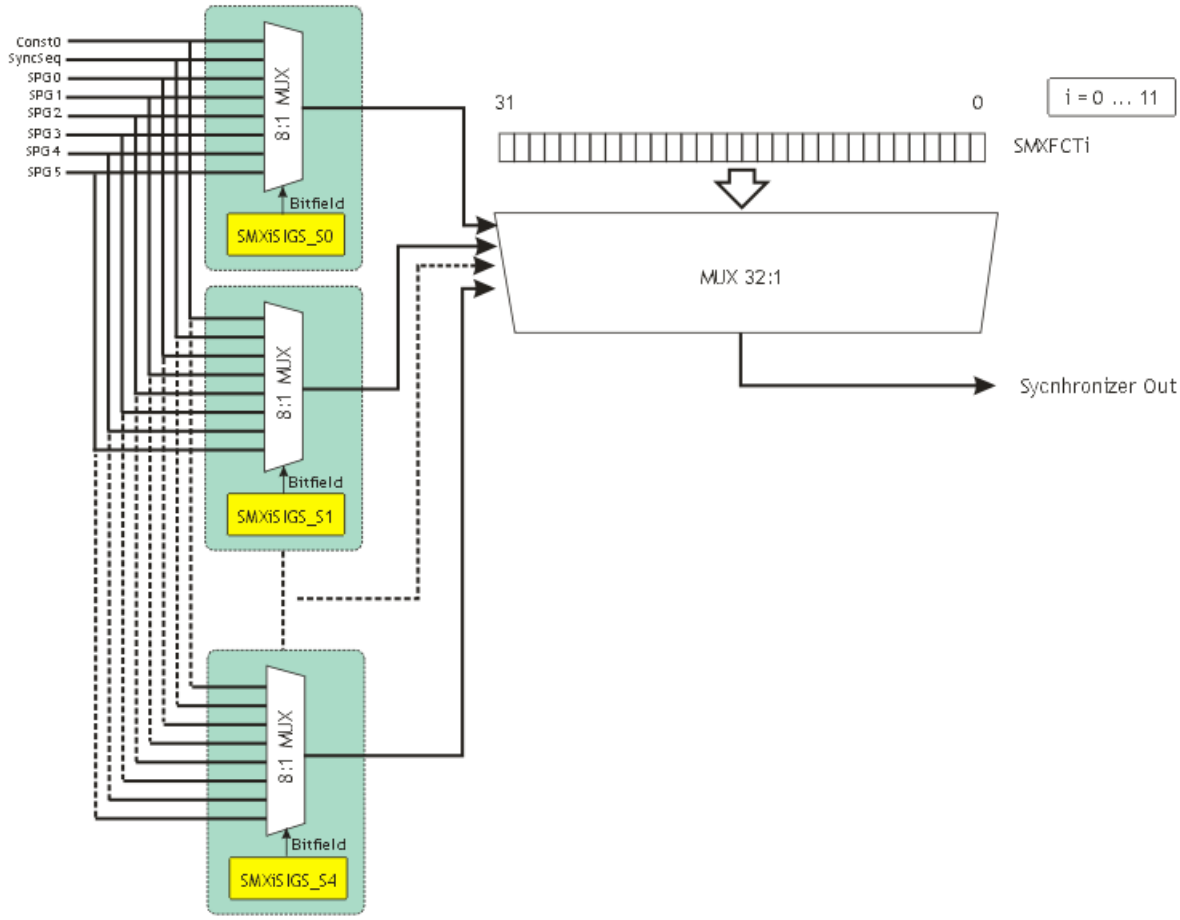


Figure 10-7 Basic structure of a Sync Mixer

Basic structure of a Sync Mixer: Each of the five address lines of the 32 to 1 multiplexer can be individually selected from any of the first-stage signals. The output is the result of a table look-up. The register FctTable contains the truth table of the Boolean function calculated.

The concept of the sync mixers needs some explanation. In a first step the signals to be combined are selected. These are referred to then as S0…S4 and form the address for the function table. This function table is used to look up the result of the Boolean operation the five selected signals shall be subject to.

An example may help understand the topic. Assuming the outputs of three Sync Pulse Generators shall form a combined signal with the function , one would proceed as follows.
At first, the Sync Mixer signals S0…S4 are assigned the Sync Pulse Generator outputs or constant zero by programming the respective multiplexers. The next step is to build the function's truth table, as shown below. As the intended function has only three inputs, only eight entries need be specified.

Selected First-stage Signals					Desired Output
S4 = 0	S3 = 0	S2 = SPG1	S1 = SPG0	S0 = SyncSeq	SMx = f(S0….S4)
0	0	0	0	0	0
0	0	0	0	1	0
0	0	0	1	0	0
0	0	0	1	1	1
0	0	1	0	0	0
0	0	1	0	1	0
0	0	1	1	0	0
0	0	1	1	1	0
Combinations [S4…S0] = 10000…11111 can never occur since S4 and S3 are selected constantly zero					need not be specified

Table 10-7 Function table for the Sync Mixer example

It is recommended that S4…S0 are listed in order of binary number representation. This makes it possible to use the function result row directly as register contents for the Sync Mixer function table, i.e. the last row is interpreted as binary 32 bit number with the LSB in the first row and the MSB in the last. For the example this would be [xxxx xxxx xxxx xxxx xxxx xxxx 0000 1000] binary, with x's denoting arbitrarily set or reset bits, since these will never be read out of the function table.

10.5.2.4.7. Sync Signal Delay Adjustment

Before the outputs of the twelve Sync Mixers are connected to actual GDC pins, they are fed through a programmable delay stage. This allows the signals either to be left untouched or delayed for half a pixel clock cycle. This delay can be set for each of the twelve Sync Mixer output signals individually with the Sync Switch register.

10.5.2.5. Inversion Signal Generation

The purpose of the inversion signal INV is the minimization of total signal edge transitions on the RGB data bus. Especially for TTL RGB signals this brings benefits for EMI.
The inversion signal is transmitted as accompanying signal to the output RGB data signals.

The input data of time (n - 1) is compared to the data at time n. If more than the half of the active RGB bits have a transition from low to high or vice versa, then INV toggles between HIGH and LOW. When INV is output as HIGH, all the bits of the current pixel are inverted.

10.5.3. Bypass-Mode

10.5.3.1. Pin mapping “Bypass Mode Data”

Pin name	Internal signal name	MSIO cell	color
DISPP[0]	msio_data_a0[0]	0	R2
DISPN[0]	msio_data_a1[0]		R3
DISPP[1]	msio_data_a0[1]	1	R4
DISPN[1]	msio_data_a1[1]		R5
DISPP[2]	msio_data_a0[2]	2	R6
DISPN[2]	msio_data_a1[2]		R7
DISPP[3]	msio_data_a0[3]	3	G2
DISPN[3]	msio_data_a1[3]		G3
DISPP[4]	msio_data_a0[4]	4	G4
DISPN[4]	msio_data_a1[4]		G5
DISPP[5]	msio_data_a0[5]	5	G6
DISPN[5]	msio_data_a1[5]		G7
DISPP[6]	msio_data_a0[6]	6	B2
DISPN[6]	msio_data_a1[6]		B3
DISPP[7]	msio_data_a0[7]	7	B4
DISPN[7]	msio_data_a1[7]		B5
DISPP[8]	msio_data_a0[8]	8	B6
DISPN[8]	msio_data_a1[8]		B7
DISPP[9]	msio_data_a0[9]	9	clk
DISPN[9]	msio_data_a1[9]		clk

10.5.3.2. Pin mapping “Bypass Mode Control Signals”

Pin name	signal name	Description
TSIG[0]	video_h_i	Horizontal Sync from VDC module
TSIG[1]	video_v_i	Vertical Sync from VDC module
TSIG[2]	video_en_i	Data Enable signal from VDC module
TSIG[3]	video_gp_i	Reserved
TSIG[4]	video_h_i	Horizontal Sync from VDC module
TSIG[5]	video_v_i	Vertical Sync from VDC module
TSIG[6]	video_en_i	Data Enable signal from VDC module
TSIG[7]	video_gp_i	Reserved
TSIG[8]	video_h_i	Horizontal Sync from VDC module
TSIG[9]	video_v_i	Vertical Sync from VDC module
TSIG[10]	video_en_i	Data Enable signal from VDC module
TSIG[11]	video_gp_i	Reserved
TSIG[12]	“0”	Constant level output

10.5.4. Limitations

- Several configuration registers only have an effect when TTL mode is enabled. Please see chapter

10.4.4. These registers are marked “TTL-mode only”.

- Reprogramming the configuration registers during active display can have undefined sideeffects.
- Correct timing signal generation is only possible for uniform and unbroken LastColumn and LastRow input signals.
- To addressrange 0h … 0FFh (embedded memory) only word access is supported. Byte or halfword access is not allowed to this address range. All the other addresses support byte, halfword, word access.

10.6. Application Note

10.6.1. Channel to pin mapping.

10.6.2. Pin mapping RSDS

In RSDS mode each IO-cell can be used for clock distribution.
The table below shows possible positions for clock output of a 24 bit RGB panel Interface. For 18bit interfaces only cell0 to 9 is available. Programming is done by register DIR_PIN[i]_CTRL.(N)channel_sel[i].
For channel to color bit assignment see Table 10-1 or Table 10-2 (18bit interface).

Table 10-8

		clock position (RSDS)												
cell	pin	0	1	2	3	4	5	6	7	8	9	10	11	12
0	d	clk	ch0	ch0	ch0	ch0	ch0	ch0	ch0	ch0	ch0	ch0	ch0	ch0
1	d	ch0	clk	ch1	ch1	ch1	ch1	ch1	ch1	ch1	ch1	ch1	ch1	ch1
2	d	ch1	ch1	clk	ch2	ch2	ch2	ch2	ch2	ch2	ch2	ch2	ch2	ch2
3	d	ch2	ch2	ch2	clk	ch3	ch3	ch3	ch3	ch3	ch3	ch3	ch3	ch3
4	d	ch3	ch3	ch3	ch3	clk	ch4	ch4	ch4	ch4	ch4	ch4	ch4	ch4
5	d	ch4	ch4	ch4	ch4	ch4	clk	ch5	ch5	ch5	ch5	ch5	ch5	ch5
6	d	ch5	ch5	ch5	ch5	ch5	ch5	clk	ch6	ch6	ch6	ch6	ch6	ch6
7	d	ch6	ch6	ch6	ch6	ch6	ch6	ch6	clk	ch7	ch7	ch7	ch7	ch7
8	d	ch7	ch7	ch7	ch7	ch7	ch7	ch7	ch7	clk	ch8	ch8	ch8	ch8
9	d	ch8	ch8	ch8	ch8	ch8	ch8	ch8	ch8	ch8	clk	ch9	ch9	ch9
10	d	ch9	ch9	ch9	ch9	ch9	ch9	ch9	ch9	ch9	ch9	clk	ch10	ch10
11	d	ch10	ch10	ch10	ch10	ch10	ch10	ch10	ch10	ch10	ch10	ch10	clk	ch11
12	d	ch11	ch11	ch11	ch11	ch11	ch11	ch11	ch11	ch11	ch11	ch11	ch11	clk

10.6.3. Pin mapping TTL

In single-ended TTL mode, each of the 24 output pins can be used for clock distribution.
Table 10-8 visualizes possible rationable settings for a 24 bit output data plus clock to single-ended I/O cells, which can be programmed by registers DIR_PIN[i]_CTRL.(N)channel_sel[i].

Table 10-9 visualizes possible rationable settings for a 18 bit output data plus clock to single-ended I/O cells, which can be programmed by registers DIR_PIN[i]_CTRL.(N)channel_sel[i].

Pads can be configured to contribute two clock sources (this allows at the board to combine these two clock signals to achieve a higher drive strength if necessary, see Figure 10-8) or to contribute one clock source and one inversion control signal “INV” on different pins shown in the table below.

Remark: In case of integration within systems with only 10 pad cells (e.g. Indigo-L IC) color channel 18 is provided but not needed. So it makes sense to use cell 9 pin a1 for clk..

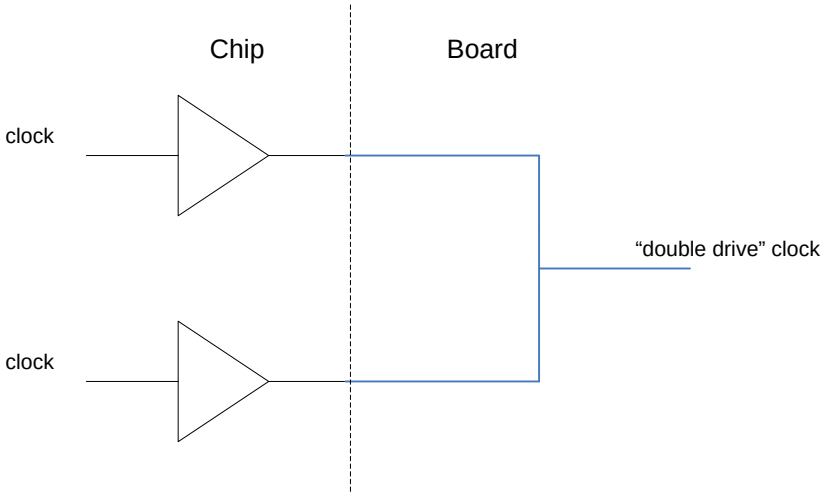


Figure 10-8

Table 10-9

		clock position (TTL)																								
cell	pin	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
	a0	Clk, invl	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	a1	0	clk	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	a0	1	1	clk	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
1	a1	2	2	2	clk	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
	a0	3	3	3	3	clk	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
2	a1	4	4	4	4	4	clk	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
	a0	5	5	5	5	5	5	clk	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
3	a1	6	6	6	6	6	6	6	clk	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
	a0	7	7	7	7	7	7	7	7	clk	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
4	a1	8	8	8	8	8	8	8	8	8	clk	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
	a0	9	9	9	9	9	9	9	9	9	9	clk	10	10	10	10	10	10	10	10	10	10	10	10	10	10
5	a1	10	10	10	10	10	10	10	10	10	10	10	clk	11	11	11	11	11	11	11	11	11	11	11	11	11
	a0	11	11	11	11	11	11	11	11	11	11	11	11	clk	12	12	12	12	12	12	12	12	12	12	12	12
6	a1	12	12	12	12	12	12	12	12	12	12	12	12	12	clk	13	13	13	13	13	13	13	13	13	13	13
	a0	13	13	13	13	13	13	13	13	13	13	13	13	13	13	clk	14	14	14	14	14	14	14	14	14	14
7	a1	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	clk	15	15	15	15	15	15	15	15	15
	a0	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	clk	16	16	16	16	16	16	16	16
8	a1	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	clk	17	17	17	17	17	17	17
	a0	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	clk	18	18	18	18	18	18
9	a1	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	clk	19	19	19	19	19
	a0	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	19	clk	20	20	20	20
10	a1	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	clk	21	21	21
	a0	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	21	clk	22	22
11	a1	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	22	clk	23
	a0	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	23	clk, inv
12	a1	clk, inv	clk, inv	clk, inv	clk, inv	clk, inv	clk, inv	clk, inv	clk, inv	clk, inv	clk, inv	clk, inv	clk, inv	clk, inv	clk, inv	clk, inv	clk, inv	clk, inv	clk, inv	clk, inv	clk, inv	clk, inv	clk, inv	clk, inv	clk, inv	clk, inv

clk, inv

Means either clock or INV (inversion control signal) is output

Table 10-10

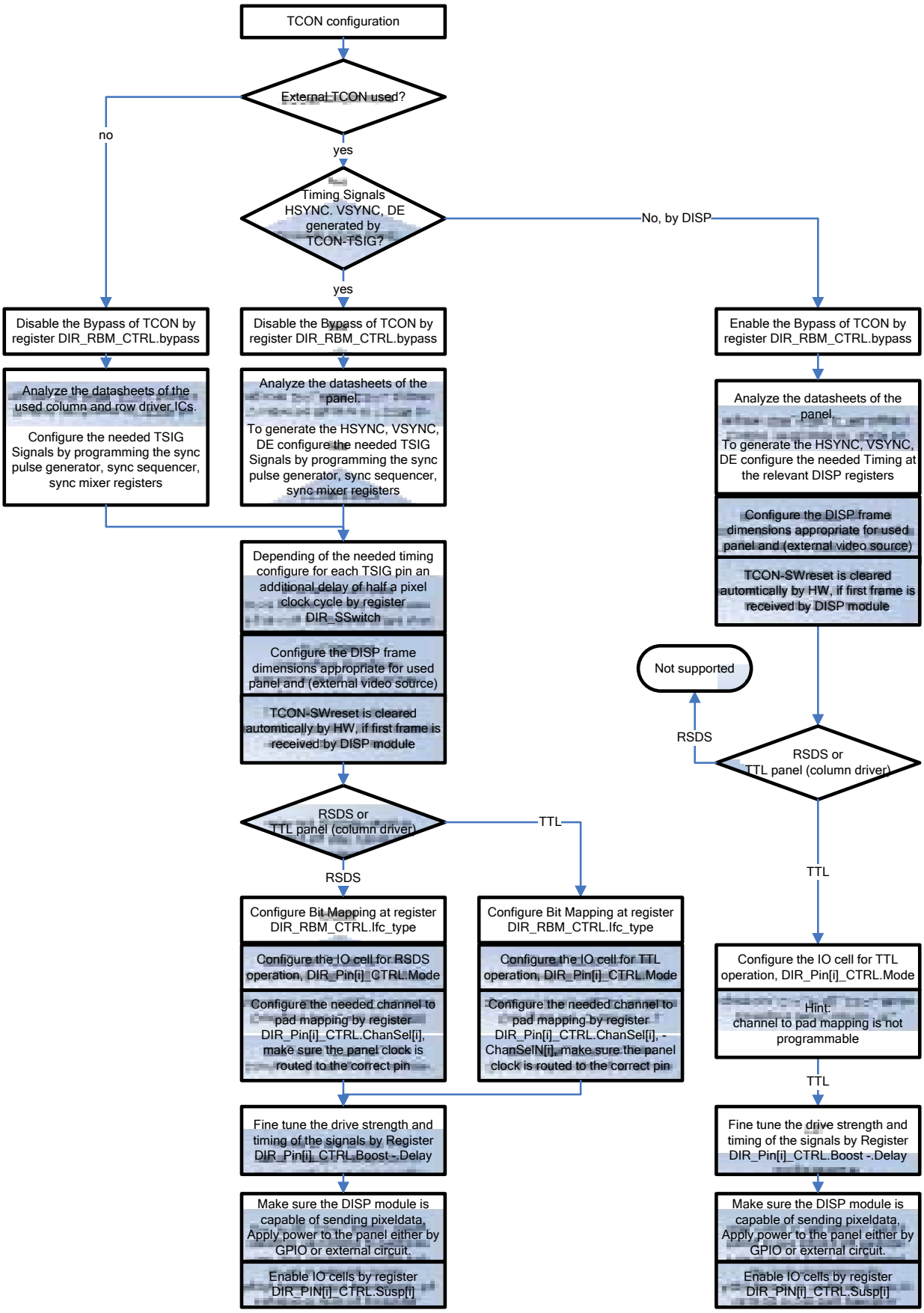
		clock position (TTL)																				
cell	pinn	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	18&19	19
0	a0	clk	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	inv
	a1	0	clk	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
1	a0	1	1	clk	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1
	a1	2	2	2	clk	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2
2	a0	3	3	3	3	clk	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	3
	a1	4	4	4	4	4	clk	5	5	5	5	5	5	5	5	5	5	5	5	5	5	4
3	a0	5	5	5	5	5	5	clk	6	6	6	6	6	6	6	6	6	6	6	6	6	5
	a1	6	6	6	6	6	6	6	clk	7	7	7	7	7	7	7	7	7	7	7	7	6
4	a0	7	7	7	7	7	7	7	7	clk	8	8	8	8	8	8	8	8	8	8	8	7
	a1	8	8	8	8	8	8	8	8	8	clk	9	9	9	9	9	9	9	9	9	9	8
5	a0	9	9	9	9	9	9	9	9	9	9	clk	10	10	10	10	10	10	10	10	10	9
	a1	10	10	10	10	10	10	10	10	10	10	10	clk	11	11	11	11	11	11	11	11	10
6	a0	11	11	11	11	11	11	11	11	11	11	11	11	clk	12	12	12	12	12	12	12	11
	a1	12	12	12	12	12	12	12	12	12	12	12	12	12	clk	13	13	13	13	13	13	12
7	a0	13	13	13	13	13	13	13	13	13	13	13	13	13	13	clk	14	14	14	14	14	13
	a1	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	clk	15	15	15	15	14
8	a0	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	clk	16	16	16	15
	a1	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	clk	17	17	16
9	a0	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	clk	clk	17
	a1	Const0	Const0	Const0	Const0	Const0	Const0	Const0	Const0	Const0	Const0	Const0	Const0	Const0	Const0	Const0	Const0	Const0	Const0	Const0	clk	clk

10.7. Control Flow

10.7.1. Example Control Flow

The following diagram shows the decision flow and configuration steps for a panel driver routine in principle. Of course a lot of panel timing relevant parameters are configured at the DISP module. This is not covered in this section.

Some decisions must already be taken when designing the board, selecting the used panel, driver ICs, etc. The diagram does not show the detailed flow and exact order of configuration steps for one application.


TCON configuration
External TCON used?
no
yes
Timing Signals HSYNC. VSYNC, DE generated by TCON-TSIG?
No, by DISP
yes
Disable the Bypass of TCON by register DIR_RBM_CTRL.bypass
Disable the Bypass of TCON by register DIR_RBM_CTRL.bypass
Enable the Bypass of TCON by register DIR_RBM_CTRL.bypass
Analyze the datasheets of the used column and row driver ICs.
Configure the needed TSIG Signals by programming the sync pulse generator, sync sequencer, sync mixer registers
Analyze the datasheets of the panel.
To generate the HSYNC, VSYNC, DE configure the needed TSIG Signals by programming the sync pulse generator, sync sequencer, sync mixer registers
Analyze the datasheets of the panel.
To generate the HSYNC, VSYNC, DE configure the needed Timing at the relevant DISP registers
Configure the DISP frame dimensions appropriate for used panel and (external video source)
TCON-SWreset is cleared automtically by HW, if first frame is received by DISP module
Depending of the needed timing configure for each TSIG pin an additional delay of half a pixel clock cycle by register DIR_SSwitch
Configure the DISP frame dimensions appropriate for used panel and (external video source)
TCON-SWreset is cleared automtically by HW, if first frame is received by DISP module
Not supported
RSDS
RSDS or TTL panel (column driver)
RSDS or TTL panel (column driver)
TTL
RSDS
TTL
Configure Bit Mapping at register DIR_RBM_CTRL.Ifc_type
Configure the IO cell for RSDS operation, DIR_Pin[i]_CTRL.Mode
Configure the needed channel to pad mapping by register DIR_Pin[i]_CTRL.ChanSel[i], make sure the panel clock is routed to the correct pin
Configure Bit Mapping at register DIR_RBM_CTRL.Ifc_type
Configure the IO cell for TTL operation, DIR_Pin[i]_CTRL.Mode
Configure the needed channel to pad mapping by register DIR_Pin[i]_CTRL.ChanSel[i], -ChanSelN[i], make sure the panel clock is routed to the correct pin
Configure the IO cell for TTL operation, DIR_Pin[i]_CTRL.Mode
Hint: channel to pad mapping is not programmable
TTL
Fine tune the drive strength and timing of the signals by Register DIR_Pin[i]_CTRL.Boost -.Delay
Fine tune the drive strength and timing of the signals by Register DIR_Pin[i]_CTRL.Boost -.Delay
Make sure the DISP module is capable of sending pixeldata, Apply power to the panel either by GPIO or external circuit.
Enable IO cells by register DIR_PIN[i]_CTRL.Susp[i]
Make sure the DISP module is capable of sending pixeldata, Apply power to the panel either by GPIO or external circuit.
Enable IO cells by register DIR_PIN[i]_CTRL.Susp[i]

11. Color Look Up Table (CLUT)

This chapter describes the MB88F333's Color Look Up Table (CLUT).

11.1. Outline

A Color LUT (CLUT) is used either to compensate the non-linearity of color transmission or to adapt to the individual characteristics of a display panel by converting a logical color to a physical color that can be displayed on a monitor. A logical color (y) would – in the perfect world – map to an identical physical output color (x) on a display. However, due to physical transmission tolerances, this relationship is not completely linear. To correct deviations in the linearity, the output colors in the CLUT are programmed with a corrective offset factor (k), whereby k is dependant on the panel characteristics. This yields the CLUT function *y=x power k.*

From a hardware point of view, the CLUT is simply a block of fast RAM with 256 entries, each of which is 10 bits wide for each RGB component and can be programmed by software.
The CLUT operates in one of three modes:

- Bypass mode: The CLUT is disabled i.e. bypassed
- Direct mode: The 256 entries are directly mapped to the **ColourIndex** register address space
- Index mode: The incoming 8 red color channel bits are used as an index value to address a specific color in the CLUT's palette (max. 256 colors are therefore addressable).

The CLUT is a part of the display output interface and must be initialized by the application software during the initialization phase because there are no default values for its contents.

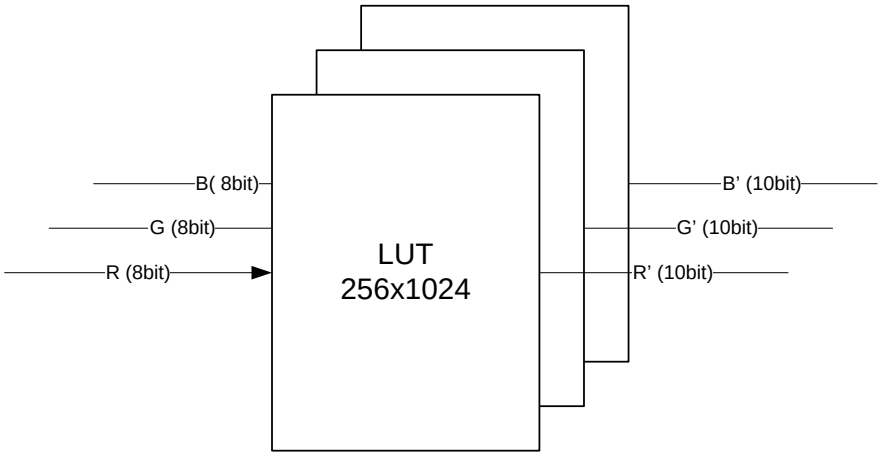


Figure 11-1 CLUT Input/Output

11.2. Features

The Color Look Up Table unit has the following features:

- Single block table with 256 entries and 10 bit accuracy for each color with optional index mode
- Parallel programming of the table content
- Direct mapping to Configuration Address space
- Indirect mode
- Bypass functionality

11.3. Block diagram

Figure 11-2shows a block diagram of the CLUT unit.

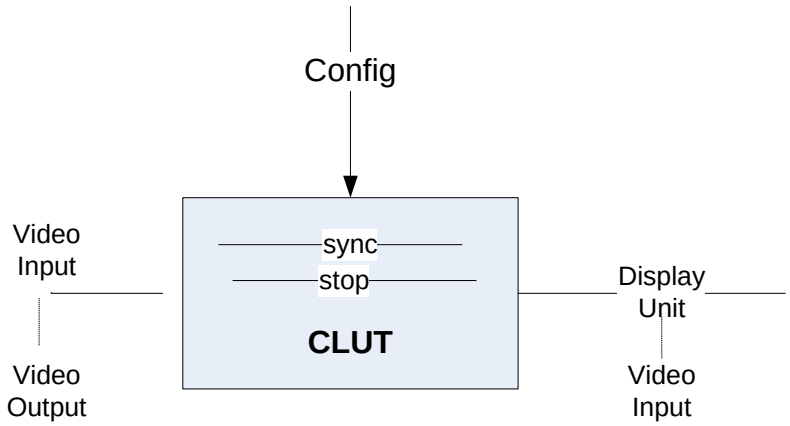
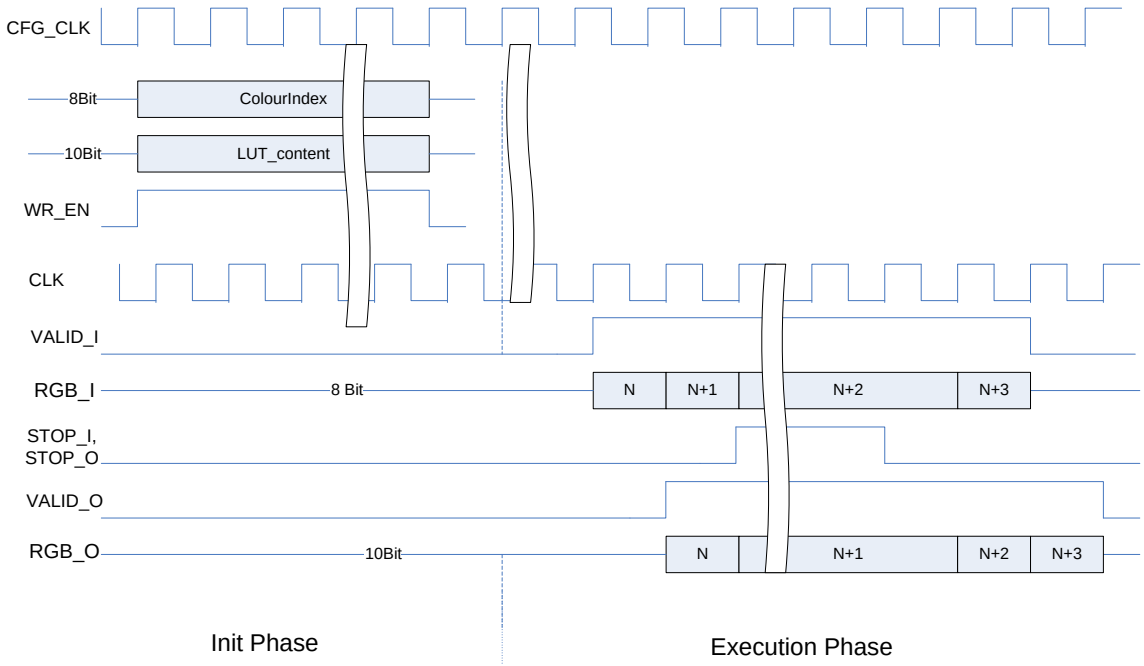


Figure 11-2 CLUT block diagram

11.3.1. Timing Diagram



11.4. Registers

11.4.1. Format of Register Descriptions

The register descriptions in the following sections use the format shown below to describe each bit field of a register.

Register address	Offset																														
Bit number	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
Field name																																
R/W																																
Reset value																																

Meaning of items and sign

Register Address

Register address shows the address (Offset address) of the register.

Bit number

Bit number shows bit number of the register.

Field Name

Field name shows bit name of the register.

R/W

R/W shows the read/write attribute of each bit field:

- R: Read Only
- RW: Read and Write
- W1C: Writing a value of "1" clears the register.

Reset value

Reset value indicates the value of each bit field immediately after reset.

- 0: Initial value is "0".
- 1: Initial value is "1".
- X: Undefined.

Unused register fields are marked with a solid grey background.

Bit vectors are unsigned integers, if nothing else specified.

11.4.2. Global Address

For the module base address please refer to the chapter 3 Memory map.

11.4.3. Register summary

Address	Register Name	Description
Base address + 0_H : Base address + $3FF_H$	ColourIndex	Lookup table for colour indexing
Base address + 400_H	CLUTControl	Colour Lookup table Control

11.4.4. Register Description

11.4.4.1. ColourIndex[0…255]

Register address	BaseAddress + 0_H : BaseAddress + $3FF_H$																													
Bit number	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
Field name			BlueColourComponentValue										GreenColourComponentValue										RedColourComponentValue									
R/W			RW										RW										RW									
Reset value			X										X										X									

Lookuptable for colour indexing. Restriction: only 32bit word access is supported
Bit 29 - 20 BlueColourComponentValue
Bit 19 - 10 GreenColourComponentValue
Bit 9 - 0 RedColourComponentValue

11.4.4.2. CLUTControl

Register address	BaseAddress + 400_H																													
Bit number	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
Field name																							clut_index						clut_bypass		
R/W																							RW						RW		
Reset value																							0_H						1_H		

Colour Lookuptable Control
Bit 8 clut_index
Sets the index mode, 0b=disabled, 1b=enabled
Bit 0 clut_bypass
Bypass for Colour Lookuptable, 0b=bypass disable, 1b=bypass enable

11.5. Limitations

The CLUT uses a single block of embedded RAM. Changes made to this area are effective immediately (i.e. there is no shadow RAM to buffer modifications), even if the video frame is active. Modifications to the CLUT content should therefore only be made during the vertical blanking period (or by turning off the display during the reconfiguration).

Note also that likewise, the internal bypass functionality of the CLUT module is effective immediately when the corresponding register is written (i.e. there is no synchronization with the vertical blanking period).

Word access only is supported for the address range 0h … 3FFh (embedded memory). Byte or halfword access is not allowed to this address range.

11.6. Initialization procedure

- Program the values for all 3 channels; red, green and blue in the CLUT
- Optional: enable index mode
- Enable the CLUT by setting clut_bypass to 0. If clut_bypass=1, the video input will be bypassed to the output.

We strongly recommend that you enable the dither unit when the CLUT is enabled in order to avoid artifacts which can be caused by quantization when the k power function operates in the area close to zero.

12. Dither Unit (DITH)

This chapter describes the Dithering Unit of the MB88F333.

12.1. Outline

A dither unit is required in a display processing pipe line to display images on a graphic device which has less color levels than those contained in the original picture data and in order to obtain a better visual result. To achieve this, the dither unit modifies pixels in such a way that the average color level of these pixels is used.

12.2. Features

The Dither Unit has following features.

12.2.1. Features

- Bypass mode

In Bypass mode, the input data will be passed to the output, whereby the 2 LSB of the input data will be dropped (removed).

- Spatial dithering mode

In spatial dithering mode, the intensities of neighboring pixels are modified so that their combined intensities average out to the desired value. Using a 4x4 matrix (see Table 12-1), the incoming pixel value will be filtered depending on the location of the pixel in a frame. The pixel location is generated according to the horizontal and vertical synchronization signals.

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

Table 12-1 4x4 Ordered dither matrix

- Temporal dithering mode

In temporal dithering mode, the temporal variation (time division) of the pixels can be achieved by generating a random vector that is used for addressing the dither matrix.

- Output resolution of 666 / 565 (see Table 12-2)
- Align the output data to the upper or lower part of the byte in order to display the output pixels on an RGB888 monitor (see Table 12-2)

dither_Align	dither_format	Output format 666/565							
		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	666	0	0	RGB 666					
1	666	RGB 666						0	0
0	565	0	0	0	R5				
		0	0	G6					
		0	0	0	B5				
1	565	R5					0	0	0
		G6						0	0
		B5					0	0	0

Table 12-2 Output format as a function of dither_align and dither_format

12.2.2. Limitations

None.

12.3. Block diagram

Figure 12-1shows the block diagram of the dither unit.

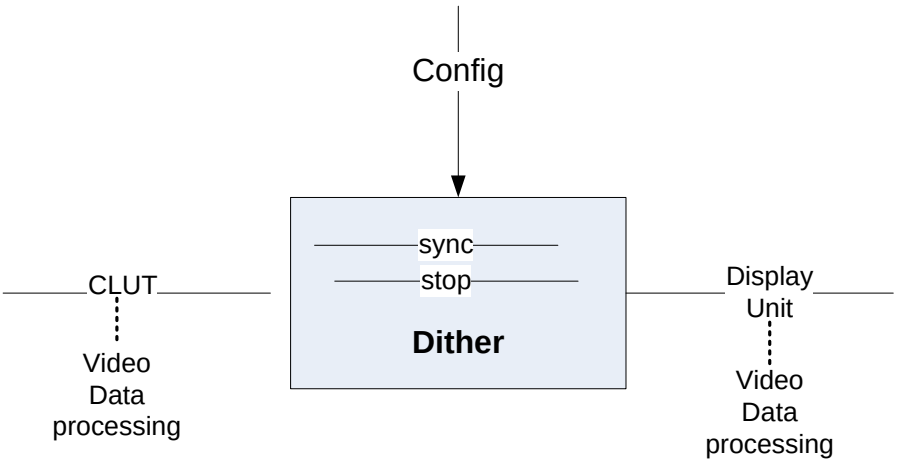


Figure 12-1 Dither module

12.4. Registers

12.4.1. Format of Register Descriptions

The register descriptions in the following sections use the format shown below to describe each bit field of a register.

Register address	Offset																														
Bit number	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
Field name																																
R/W																																
Reset value																																

Meaning of items and sign

Register Address

Register address shows the address (Offset address) of the register.

Bit number

Bit number shows bit number of the register.

Field Name

Field name shows bit name of the register.

R/W

R/W shows the read/write attribute of each bit field:

- R: Read
- W: Write
- W1C: Writing a value of "1" clears the register.

Reset value

Reset value indicates the value of each bit field immediately after reset.

- 0: Initial value is "0".
- 1: Initial value is "1".
- X: Undefined.

Unused register fields are marked with a solid grey background.

Bit vectors are unsigned integers, if nothing else specified.

12.4.2. Global Address

For the module base address please refer to the chapter 3 Memory map.

12.4.3. Register summary

Address	Register Name	Description
Base address + 0_H	DitherControl	Dither Unit Control

12.4.4. Register Description

12.4.4.1. DitherControl

Register address	BaseAddress + 0_H																													
Bit number	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
Field name																								dither_align	dither_reserved_2		dither_format		dither_reserved_1		dither_mode	dither_bypass
R/W																								RW	RWS		RW		RWS		RW	RW
Reset value																								0_H	X		0_H		X		0_H	1_H

Dither Unit Control
Bit 8 dither_align
Dithering Align, 0b=right aligned, 1b=left aligned
Bit 7 - 6 dither_reserved_2
Bit 5 - 4 dither_format
Dithering Format, 00b=reserved, 01b=reserved, 10b=666, 11b=565
Bit 3 - 2 dither_reserved_1
Bit 1 dither_mode
Dithering Mode Register, 0b=temporal, 1b=spatial
Bit 0 dither_bypass
Bypass for Dither Unit, 0b=bypass disable, 1b=bypass enable

Note:
There is no shadow register for the synchronization of configuration parameters during the vertical blanking period. To avoid visible artifacts during reconfiguration, we recommend you to find a trigger point to modify the configuration register during the vertical blanking by software.

12.5. Initialization procedure

Setup e.g. for 8 bit panel
• Select the dither_mode
• Select dither_format = 0x00
• Enable dithering by setting dither_bypass = 0
In case of the output resolution less than 8 bit, it's strongly recommended to enable the Dither Unit to obtain a better visual display.

13. Signature Generator (SIG)

This chapter describes the Signature Generator of the MB88F333.

13.1. Outline

The Signature Generator unit (SIG) calculates different types of checksums for input data. Application is the generation of a checksum (signature) for pixel stream data for a user-defined evaluation window (whose size and position can be programmed).

The system micro controller can use such signatures e.g. to determine whether the displayed image is exactly identical (or almost identical) to the original image data submitted. This is necessary for critical safety displays and helps to fulfil the requirements of safety standards (e.g. Automotive Safety Integrity Level ASIL).

13.2. Features

The Signature Generator has following features.

13.2.1. Features

- Generation of 2 different picture signatures for each color channel
  - summation of color values
  - CRC-32 for color values
- Programmable evaluation window position and size
- Programmable evaluation window mask
- Automatic monitoring using reference signature registers
- Interrupt generation
- Programmable picture source
- Self restoring error counter

13.2.2. Limitations

None

13.3. Function

13.3.1. Block diagram

Figure 13-1 shows SIG's block diagram.

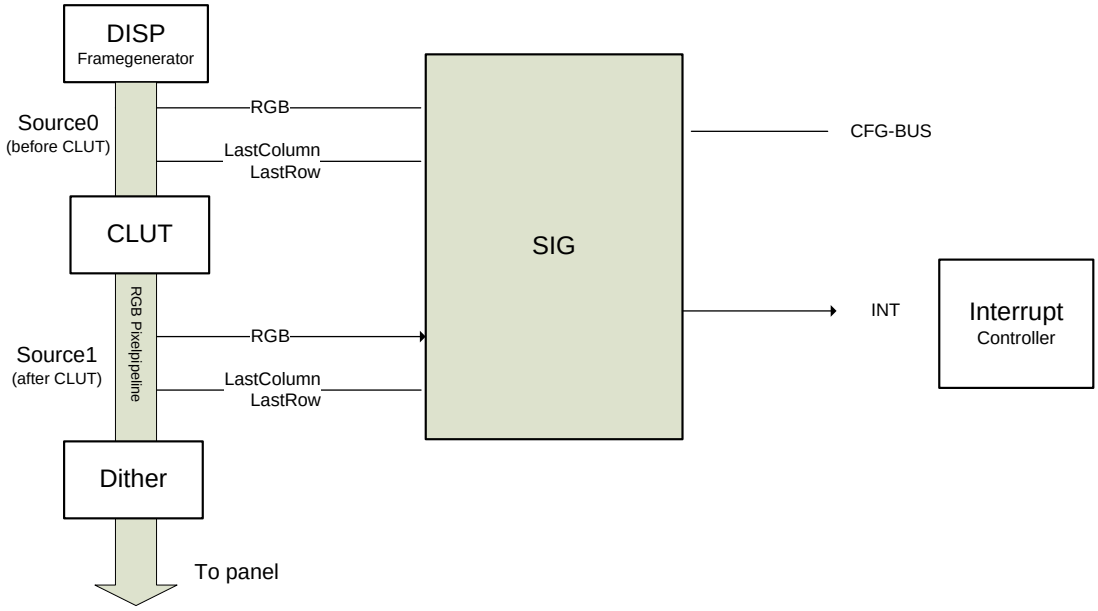


Figure 13-1 Position of Block in whole LSI

13.3.2. Signature A: CRC-32 Signature

For each color channel in the evaluation window area a CRC-32 checksum is generated. The same CRC32 algorithm as in the Ethernet Standard (CRC-32-IEEE 802.3) except the last step e is applied. That means at the end the resulting bit sequence is not complemented. See C++-code below.

The default polynome is:
$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$
Start value:
FFFF_FFFF

```
long crc32::crc32_input ( short data ) {

  int i ;
  long temp, polynome ;
  bool lsb ;

  polynome = 0x04C11DB7 ;

  temp = current_value ;

  for ( i = 7 ; i >= 0 ; i-- ) {
    lsb = data & ( 1 << i ) ;
    if ( ( temp & 0x80000000 ) != lsb ) {
      temp = ( temp << 1 ) ^ polynome ;
    } else {
      temp = temp << 1 ;
    }
  }

  current_value = temp ;

  return temp ;

}

crc32::crc32() {
  current_value = 0xFFFFFFFF ;
}

crc32::~crc32() {}

long crc32::get_value() {
  return current_value ;
}

void crc32::reinit() {
  current_value = 0xFFFFFFFF ;
}
```

13.3.3. Signature B: Summation Signature

The sum of pixel color values for each color channel (R, G, B) in the evaluation window area is calculated.

13.3.4. Programmable Evaluation Window (Position and Size)

The position and size (upper left and lower right coordinates) of the evaluation window are programmable. Window coordinates follow the shown coordinate system of Figure 13-2 The coordinate reference point is determined by the VDC (Video Display Controller) module horizontal sync signal falling edge and the first active pixel line. For definition of all parameters please refer to chapter 9.3.2 “Display Parameters”.

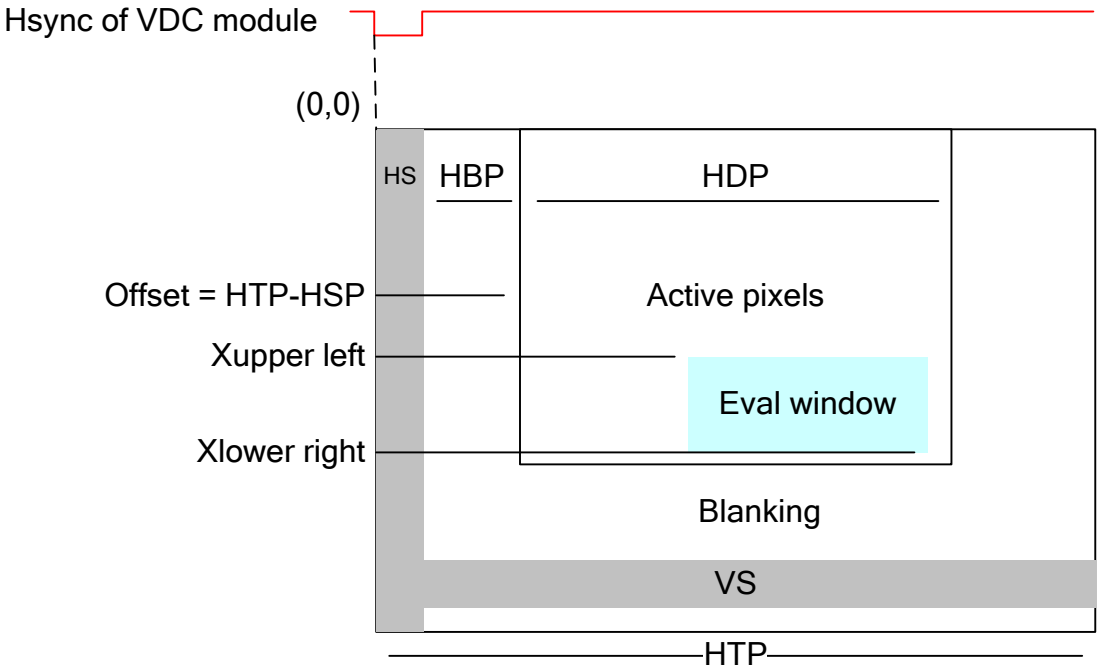


Figure 13-2 Coordinate System

13.3.5. Programmable Evaluation Window Mask

A programmable mask window allows the exclusion of incoming pixels for the signature calculation. Coordinate system of Figure 13-2 applies.

13.3.6. Automatic Monitoring and Interrupt

A set of reference signature registers allows the monitoring of the calculated signatures. An interrupt can be generated on the detection of a difference between the calculated signature and the reference value. For signature B the difference can be threshold filtered to limit the interrupt load for the microcontroller.

13.3.7. Self Restoring Error Counter

A counter is incremented if one of the active signature results differs from the corresponding reference values. If a programmable error counter threshold is reached an interrupt may be generated. The same counter is reset to zero if a programmable number of consecutive video frames with correct signature values is received.

13.3.8. Interrupts For Control Flow

An interrupt can be generated for both the start and end of a signature calculation. The start interrupt (CfgCop) indicates that the configuration parameters (e.g. window coordinates) have been copied from the shadow registers and are now active for the current signature calculation. This allows that next configuration parameters can then be loaded into the (shadow) registers without disturbing the current calculation. The end interrupt (ResVal) indicates a signature calculation has completed and the result data can be read from the result shadow registers. These interrupts help to control signature calculations for every incoming frame with different evaluation window coordinates.

13.3.9. Programmable Input Picture Source

Input data can be selected from four different sources in the processing pipeline, either before or after the CLUT.

13.3.10. Limitations

- The maximum resolution for picture sources and windows is 4096 x 4096 pixels
- The evaluation window position must be completely inside the picture source frame
- The source selection must be configured before the evaluation window coordinates and may not be changed during operation.
- There is no support for interlaced sources. Also for de-interlaced sources, no special processing is supported, that means that reference values must be calculated by SW considering the right de-interlace algorithm for such material.

13.4. Registers

13.4.1. Format of Register Descriptions

The register descriptions in the following sections use the format shown below to describe each bit field of a register.

Register address	Offset																														
Bit number	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
Field name																																
R/W																																
Reset value																																

Meaning of items and sign

Register Address

Register address shows the address (Offset address) of the register.

Bit number

Bit number shows bit position of the register.

Field Name

Field name shows bit name of each bit field.

R/W

R/W shows the read/write attribute of each bit field:

- R: Read Only
- W: Write Only
- W1C: Writing a value of "1" clears the register.

Reset value

Reset value indicates the value of each bit field immediately after reset.

- 0: Initial value is "0".
- 1: Initial value is "1".
- X: Undefined.

Unused register fields are marked with a solid grey background.

Bit vectors are unsigned integers, if nothing else specified.

13.4.2. Global Address

For the module base address please refer to the chapter 3 Memory map.

13.4.3. Register summary

Address	Register Name	Description
Base address + 0_H	SigSWreset	SIG-module SW reset
Base address + 4_H	SigCtrl	SIG-module general config register
Base address + 8_H	MaskHorizontalUpperLeft	Mask coordinates
Base address + C_H	MaskHorizontalLowerRight	Mask coordinates
Base address + 10_H	MaskVerticalUpperLeft	Mask coordinates
Base address + 14_H	MaskVerticalLowerRight	Mask coordinates
Base address + 18_H	HorizontalUpperLeftW0	Evaluation Window HorizontalUpperLeft
Base address + $1C_H$	HorizontalLowerRightW0	Evaluation Window HorizontalLowerRight
Base address + 20_H	VerticalUpperLeftW0	Evaluation Window VerticalUpperLeft
Base address + 24_H	VerticalLowerRightW0	Evaluation Window VerticalLowerRight
Base address + 28_H	SignAReferenceRW0	Signature A Reference value channel R
Base address + $2C_H$	SignAReferenceGW0	Signature A Reference value channel G
Base address + 30_H	SignAReferenceBW0	Signature A Reference value channel B
Base address + 34_H	SignBReferenceRW0	Signature B Reference value channel R
Base address + 38_H	SignBReferenceGW0	Signature B Reference value channel G
Base address + $3C_H$	SignBReferenceBW0	Signature B Reference value channel B
Base address + 40_H	ThrBRW0	Threshold Signature B
Base address + 44_H	ThrBGW0	Threshold Signature B
Base address + 48_H	ThrBBW0	Threshold Signature B
Base address + $4C_H$	ErrorThreshold	Error Counter Threshhold
Base address + 50_H	CtrlCfgW0	Control/Configuration register for evaluation window
Base address + 54_H	TriggerW0	Trigger register
Base address + 58_H	IENW0	Interrupt Enable Register
Base address + $5C_H$	InterruptStatusW0	Interrupt status register
Base address + 60_H	StatusW0	status register
Base address + 64_H	Signature_error	The amount of video frames with signature errors
Base address + 68_H	SignatureARW0	Signature A Result channel R
Base address + $6C_H$	SignatureAGW0	Signature A Result channel G
Base address + 70_H	SignatureABW0	Signature A Result channel B
Base address + 74_H	SignatureBRW0	Signature B Result channel R
Base address + 78_H	SignatureBGW0	Signature B Result channel G
Base address + $7C_H$	SignatureBBW0	Signature B Result channel B

13.4.4. Register Description

13.4.4.1. SigSWreset

Register address	BaseAddress + 0_H																													
Bit number	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
Field name																																SWRes
R/W																																RW
Reset value																																0_H

SIG-module SW reset
Bit 0 SWRes
SW reset

13.4.4.2. SigCtrl

Register address	BaseAddress + 4_H																													
Bit number	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
Field name															SrcSel								Vmask_mode						Hmask_mode		
R/W															RW								RW						RW			
Reset value															0_H								0_H						0_H			

SIG-module general config register
Bit 17 - 16 SrcSel
Source Select 00b=source 0 (before CLUT), 01b=source 1(after CLUT), 10b=reserved, 11b=reserved
Bit 9 - 8 Vmask_mode
00b=nomask, 01b=mask inside vertical coordinates, 10b=mask outside vertical coordinates, 11b=reserved
Bit 1 - 0 Hmask_mode
00b=nomask, 01b=mask inside horizontal coordinates, 10b=mask outside horizontal coordinates, 11b=reserved

13.4.4.3. MaskHorizontalUpperLeft

Register address	BaseAddress + 8_H																													
Bit number	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
Field name																					MaskHorizontalUpperLeft											
R/W																					RW											
Reset value																					0_H											

Mask coordinates
Bit 11 - 0 MaskHorizontalUpperLeft
Mask Horizontal Upper Left

13.4.4.4. MaskHorizontalLowerRight

Register address	BaseAddress + C_H																													
Bit number	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
Field name																					MaskHorizontalLowerRight											
R/W																					RW											
Reset value																					0_H											

Mask coordinates
Bit 11 - 0 MaskHorizontalLowerRight
Mask Horizontal Lower right

13.4.4.5. MaskVerticalUpperLeft

Register address	BaseAddress + 10_H																													
Bit number	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
Field name																					MaskVerticalUpperLeft											
R/W																					RW											
Reset value																					0_H											

Mask coordinates
Bit 11 - 0 MaskVerticalUpperLeft
Mask Vertical Upper Left

13.4.4.6. MaskVerticalLowerRight

Register address	BaseAddress + 14_H																													
Bit number	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
Field name																					MaskVerticalLowerRight											
R/W																					RW											
Reset value																					0_H											

Mask coordinates
Bit 11 - 0 MaskVerticalLowerRight
Mask Vertical Lower Right

13.4.4.7. HorizontalUpperLeftW0

Register address	BaseAddress + 18_H																													
Bit number	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
Field name																					HorizontalUpperLeftW0											
R/W																					RW											
Reset value																					0_H											

Evaluation Window HorizontalUpperLeft
Bit 11 - 0 HorizontalUpperLeftW0
Evaluation Window HorizontalUpperLeft

13.4.4.8. HorizontalLowerRightW0

Register address	BaseAddress + $1C_H$																													
Bit number	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
Field name																					HorizontalLowerRightW0											
R/W																					RW											
Reset value																					0_H											

Evaluation Window HorizontalLowerRight
Bit 11 - 0 HorizontalLowerRightW0
Evaluation Window HorizontalLowerRight

13.4.4.9. VerticalUpperLeftW0

Register address	BaseAddress + 20_H																													
Bit number	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
Field name																					VerticalUpperLeftW0											
R/W																					RW											
Reset value																					0_H											

Evaluation Window VerticalUpperLeft
Bit 11 - 0 VerticalUpperLeftW0
Evaluation Window VerticalUpperLeft

13.4.4.10. VerticalLowerRightW0

Register address	BaseAddress + 24_H																													
Bit number	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
Field name																					VerticalLowerRightW0											
R/W																					RW											
Reset value																					0_H											

Evaluation Window VerticalLowerRight
Bit 11 - 0 VerticalLowerRightW0
Evaluation Window VerticalLowerRight

13.4.4.11. SignAReferenceRW0

Register address	BaseAddress + 28_H																													
Bit number	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
Field name	SignAReferenceRW0																															
R/W	RW																															
Reset value	0_H																															

Signature A (CRC) Reference value channel R
Bit 31 - 0 SignAReferenceRW0
Signature A (CRC) Reference value channel R

13.4.4.12. SignAReferenceGW0

Register address	BaseAddress + $2C_H$																													
Bit number	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
Field name	SignAReferenceGW0																															
R/W	RW																															
Reset value	0_H																															

Signature A (CRC) Reference value channel G
Bit 31 - 0 SignAReferenceGW0
Signature A (CRC) Reference value channel G

13.4.4.13. SignAReferenceBW0

Register address	BaseAddress + 30_H																													
Bit number	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
Field name	SignAReferenceBW0																															
R/W	RW																															
Reset value	0_H																															

Signature A (CRC) Reference value channel B
Bit 31 - 0 SignAReferenceBW0
Signature A (CRC) Reference value channel B

13.4.4.14. SignBReferenceRW0

Register address	BaseAddress + 34_H																													
Bit number	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
Field name	SignBReferenceRW0																															
R/W	RW																															
Reset value	0_H																															

Signature B (Summation) Reference value channel R
Bit 31 - 0 SignBReferenceRW0
Signature B (Summation) Reference value channel R

13.4.4.15. SignBReferenceGW0

Register address	BaseAddress + 38_H																													
Bit number	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
Field name	SignBReferenceGW0																															
R/W	RW																															
Reset value	0_H																															

Signature B (Summation) Reference value channel G
Bit 31 - 0 SignBReferenceGW0
Signature B (Summation) Reference value channel G

13.4.4.16. SignBReferenceBW0

Register address	BaseAddress + $3C_H$																													
Bit number	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
Field name	SignBReferenceBW0																															
R/W	RW																															
Reset value	0_H																															

Signature B (Summation) Reference value channel B
Bit 31 - 0 SignBReferenceBW0
Signature B (Summation) Reference value channel B

13.4.4.17. ThrBRW0

Register address	BaseAddress + 40_H																													
Bit number	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
Field name	ThrBRW0																															
R/W	RW																															
Reset value	0_H																															

Threshold Signature B (Summation)
Bit 31 - 0 ThrBRW0
Threshold Signature B (Summation) for channels R

13.4.4.18. ThrBGW0

Register address	BaseAddress + 44_H																													
Bit number	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
Field name	ThrBGW0																															
R/W	RW																															
Reset value	0_H																															

Threshold Signature B (Summation)
Bit 31 - 0 ThrBGW0
Threshold Signature B (Summation) for channel G

13.4.4.19. ThrBBW0

Register address	BaseAddress + 48_H																													
Bit number	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
Field name	ThrBBW0																															
R/W	RW																															
Reset value	0_H																															

Threshold Signature B (Summation)
Bit 31 - 0 ThrBBW0
Threshold Signature B (Summation) for channel B

13.4.4.20. ErrorThreshold

Register address	BaseAddress + $4C_H$																													
Bit number	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
Field name									ErrThresReset																ErrThres							
R/W									RW																RW							
Reset value									8_H																1_H							

Error Counter Threshold
Bit 23 - 16 ErrThresReset
number of consecutive error free video frames which cause resetting of error_count.
0h= no reset, 1h= 1, …FFh=255
Bit 7 - 0 ErrThres
threshold of error counter, 0h=256, 1h=1, ...,FFh=255 If error_counter >= "ErrThres" it generates interrupt

13.4.4.21. CtrlCfgW0

Register address	BaseAddress + 50_H																													
Bit number	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
Field name																EnCoordW0								EnSignB						EnSignA
R/W																RW								RW						RW
Reset value																0_H								0_H						0_H

Control/Configuration register for evaluation window
Bit 16 EnCoordW0
enable coordinates for window 0
Bit 8 EnSignB
Enable for Signature calculation B
Bit 0 EnSignA
Enable for Signature calculation A

13.4.4.22. TriggerW0

Register address	BaseAddress + 54_H																													
Bit number	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
Field name																							TrigMode								Trigger
R/W																							RW								W
Reset value																							0_H								0_H

Trigger register
Bit 9 - 8 TrigMode
00b=start one generation, cancel cyclic generation., 01b=start cyclic generations, 10b= reserved ,11b=reserved
Bit 0 Trigger
generate trigger for signature generation, see TrigMode for used trigger mode

13.4.4.23. IENW0

Register address	BaseAddress + 58_H																													
Bit number	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
Field name																										IEnResVal				IEnCfgCop		IEnDiff
R/W																										RW				RW		RW
Reset value																										0_H				0_H		0_H

Control/Configuration register for evaluation window
Bit 2 IEnResVal
Interrupt enable (for condition see the relevant status field)
Bit 1 IEnCfgCop
Interrupt enable (for condition see the relevant status field)
Bit 0 IEnDiff
Interrupt enable for difference interrupt

13.4.4.24. InterruptStatusW0

Register address	BaseAddress + $5C_H$																															
Bit number	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
Field name																											IStsResVal			IStsCfgCop		IStsDiff
R/W																											RW			RW		RW
Reset value																											0_H			0_H		0_H

Interrupt status register

Bit 2 IStsResVal
Interrupt status flags, a '1' signifies that the corresponding interrupt condition occurred (even if interrupt is disabled), write '1' clears the flag, Condition: Result register is valid

Bit 1 IStsCfgCop
Interrupt status flags, a '1' signifies that the corresponding interrupt condition occurred (even if interrupt is disabled), write '1' clears the flag, Condition: Configuration Registers copied to shadow registers

Bit 0 IStsDiff
Interrupt status flags, a '1' signifies that the corresponding interrupt condition occurred (even if interrupt is disabled), write '1' clears the flag, Condition: one of the active signature results differs from the corresponding reference values

13.4.4.25. StatusW0

Register address	BaseAddress + 60_H																															
Bit number	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
Field name														Diff_B_B	Diff_B_G	Diff_B_R						Diff_A_B	Diff_A_G	Diff_A_R	Reserved2				Active	Pending		
R/W														R	R	R						R	R	R	RWS				R	R		
Reset value														0_H	0_H	0_H						0_H	0_H	0_H	0_H				0_H	0_H		

status register

Bit 18 Diff_B_B
Signature B (Summation): Comparison of valid B result vs reference value 0b= equal, 1b=different

Bit 17 Diff_B_G
Signature B (Summation): Comparison of valid G result vs reference value 0b= equal, 1b=different

Bit 16 Diff_B_R
Signature B (Summation): Comparison of valid R result vs reference value 0b= equal, 1b=different

Bit 10 Diff_A_B
Signature A (CRC): Comparison of valid B result vs reference value 0b= equal, 1b=different

Bit 9 Diff_A_G
Signature A (CRC): Comparison of valid G result vs reference value 0b= equal, 1b=different

Bit 8 Diff_A_R
Signature A (CRC): Comparison of valid R result vs reference value 0b= equal, 1b=different

Bit 7 - 5 Reserved2

Bit 1 Active
Generation Task is active

Bit 0 Pending
Generation Task is pending

13.4.4.26. Signature_error

Register address	BaseAddress + 64_H																															
Bit number	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
Field name																					Sig_error_count											
R/W																					R											
Reset value																					0_H											

The amount of video frames with signature errors

Bit 11 - 0 Sig_error_count
The amount video frames with Signature errors. Every Trigger (see TriggerW0) will reset Signature_error to 0

13.4.4.27. SignatureARW0

Register address	BaseAddress + 68_H																													
Bit number	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
Field name	SignatureARW0																															
R/W	R																															
Reset value	0_H																															

Signature A (CRC) Result channel R
Bit 31 - 0 SignatureARW0
Signature A (CRC) Result channel R

13.4.4.28. SignatureAGW0

Register address	BaseAddress + $6C_H$																													
Bit number	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
Field name	SignatureAGW0																															
R/W	R																															
Reset value	0_H																															

Signature A (CRC) Result channel G
Bit 31 - 0 SignatureAGW0
Signature A (CRC) Result channel G

13.4.4.29. SignatureABW0

Register address	BaseAddress + 70_H																													
Bit number	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
Field name	SignatureABW0																															
R/W	R																															
Reset value	0_H																															

Signature A(CRC) Result channel B
Bit 31 - 0 SignatureABW0
Signature A (CRC) Result channel B

13.4.4.30. SignatureBRW0

Register address	BaseAddress + 74_H																													
Bit number	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
Field name	SignatureBRW0																															
R/W	R																															
Reset value	0_H																															

Signature B(Summation) Result channel R
Bit 31 - 0 SignatureBRW0
Signature B (Summation) Result channel R

13.4.4.31. SignatureBGW0

Register address	BaseAddress + 78_H																													
Bit number	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
Field name	SignatureBGW0																															
R/W	R																															
Reset value	0_H																															

Signature B(Summation) Result channel G
Bit 31 - 0 SignatureBGW0
Signature B (Summation) Result channel G

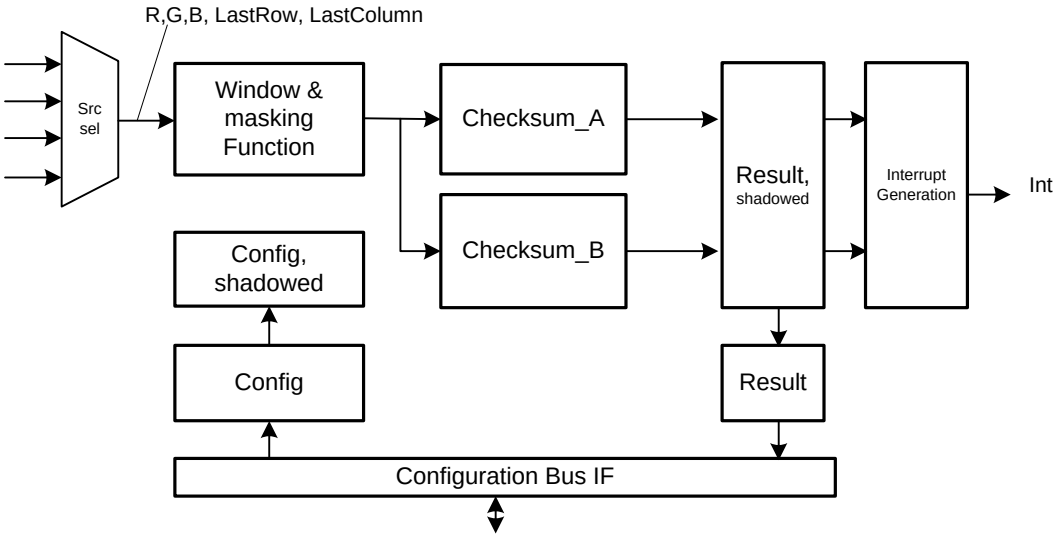
13.4.4.32. SignatureBBW0

Register address	BaseAddress + $7C_H$																													
Bit number	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
Field name	SignatureBBW0																															
R/W	R																															
Reset value	0_H																															

Signature B(Summation) Result channel B
Bit 31 - 0 SignatureBBW0
Signature B (Summation) Result channel B

13.5. Processing Mode

13.5.1. Processing Flow



13.5.2. Processing Algorithm

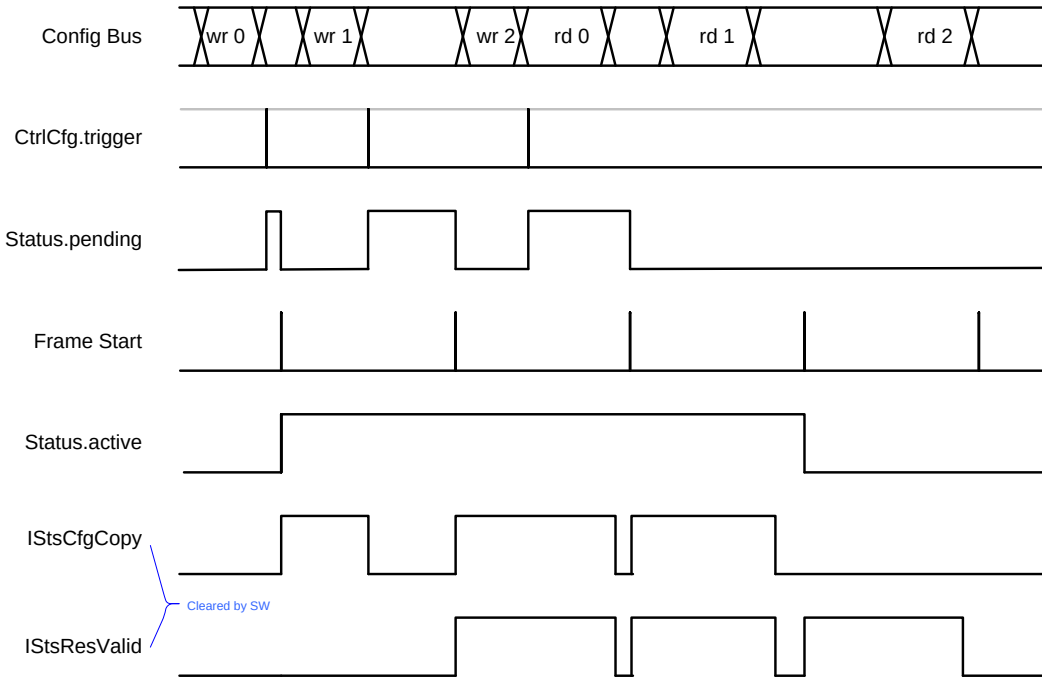
Please see chapter 13-2.
Checksum generations are possible for each incoming pixel frame. When a generation is triggered, after each incoming pixel frame, a set of signature checksum results is valid.

If continuous generation is enabled (Register TriggerW0.TrigMode=01, for each incoming pixel frame both checksums (CRC and summation) are calculated. With each frame end (VSYNC) the internal result values (Result,shadowed-block) are transferred to the external readable result registers. This register stays constant for the next frame.

If interrupts and RH events are enabled, for each CRC result value an RH event message (ID 95-97) is generated.

13.6. Control Flow

13.6.1. Example Control Flow



13.6.2. Signature Generation with every incoming frame

```
General Configuration phase: (most registers are not shadowed)
    Enable mask mode
    Write mask window coordinates
    Enable Signatures types
    Enable interrupts

Configuration phase for calculation 0
    Write Window 0 coordinates
    Set Triggermode to one single generation
    Trigger one generation by writing ‘1’ to the trigger field

Wait on interrupt or poll IStsCfgCop, IStsResVal

On IStsCfgCop
Configuration phase for calculation n
    Write Window n coordinates
    Setup single TriggerMode
    Trigger one generation by writing to the trigger field
```

```
On IStsResVal
    Read result registers Signature A (B)
    Process results
```

13.6.3. Cyclic Signature Generation with every incoming frame

```
CYCLIC monitoring of one window:

General Configuration phase: (most registers are not shadowed)
    Enable mask mode
    Write mask window coordinates
    Enable Signatures types
    Enable interrupts

Configuration phase for calculation 0
    Write Window 0 coordinates
    Setup cyclic TriggerMode
    Trigger cyclic generation by writing ‘1’ to the trigger field

Wait for interrupt or poll IStsResVal

On IStsResVal
    Read result registers Signature A, (B)
    Process results
```

13.6.4. Cyclic Signature Generation with every incoming frame, limiting read accesses

```
Cyclic for one window, with interrupt monitoring

General Configuration phase: (not shadowed registers)
    Enable mask mode
    Write mask window coordinates
    Enable Signatures types
    Enable interrupts

Configuration phase for calculation 0
    Write Window 0 coordinates
    Write reference values for Signature A
    (Write reference values for Signature B, Set threshold for B)
    Setup cyclic TriggerMode
    Trigger cyclic generation by writing ‘1’ to the trigger field

Wait on interrupt or poll IStsDiff

On IStsDiff
    Read result registers Signature A, (B)
    Process results
```

```
Before display content of evaluation window changes:
Cancel cyclic trigger and reprogram reference values
```

14. Embedded Flash Memory

This chapter describes in the embedded flash memory of the MB88F333.

14.1. Outline

The MB88F333 has the 160Kbytes of flash memory.
The Memory Interface Module controls the interface of the AHB busses to memory.
External access to the chip is possible via a mode setting.

14.2. Features

The embedded flash and its interface have the following features.

14.2.1. Features

- Flash memory sector architecture: 2 x 64Kbyte sectors + 4 x 8Kbyte
- Flash memory program/erase endurance: 10,000
- CPU Mode: Access to the flash memory via the AHB is supported as follows:
  Write：Sync mode (16bit)
  Read：Sync mode (32bit)
- CPU Mode: Access timing can be set with each system clock cycle
- Writer Mode: Supports flash access from the external I/F as follows:
  Write：Async mode (8bit/16bit)
  Read：Async mode (8bit/16bit)

14.2.2. Limitations

- The start margin from the falling edge FRSTX to the completion of the reset operation is 20us or more when the flash programmed using the automatic programming algorithm.
- FRSTX pulse width is 500ns or more.
- The start margin from the rising edge of FRSTX to read mode is 200ns or more.

14.3. Function

14.3.1. Block diagram

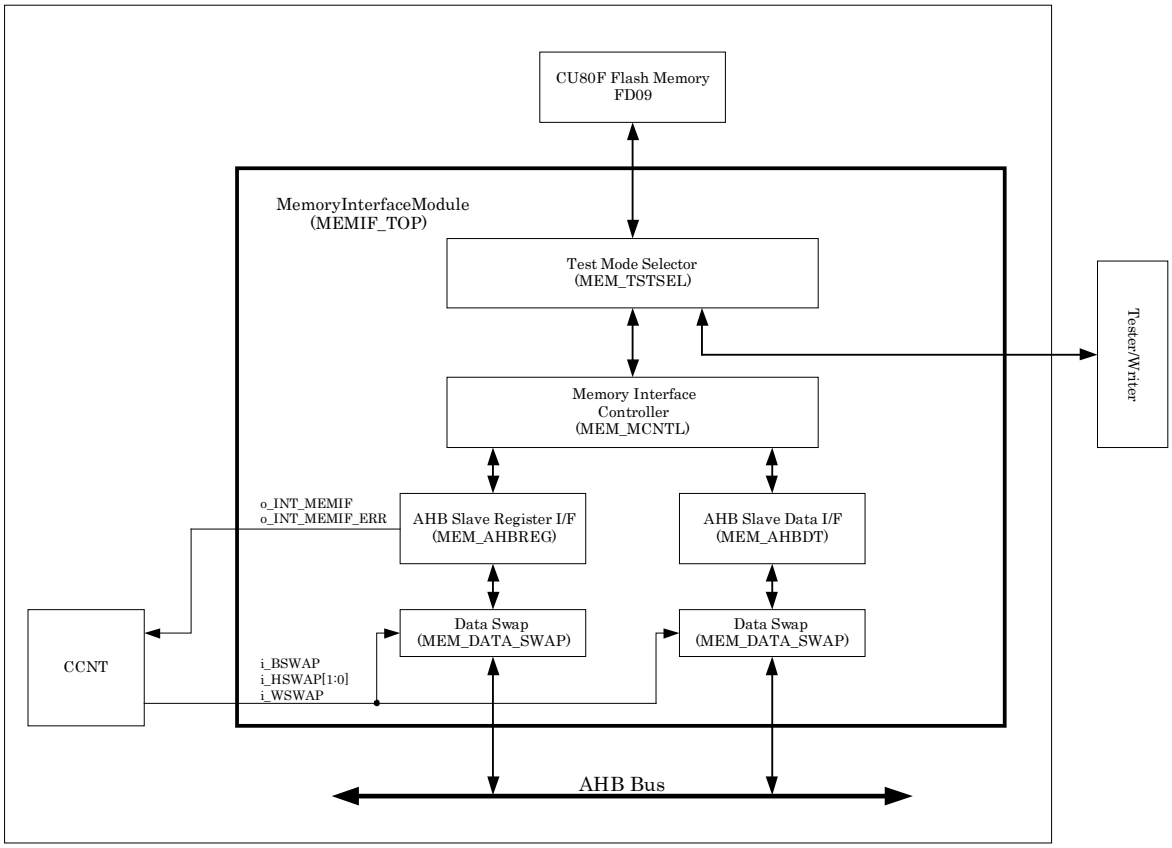


Figure 15.3-1 Memory Interface Module block chart

14.3.2. Sector Architecture of 160Kbyte Flash

The 160KB memory is organized into two 64Kbyte sector blocks with each sector, having an internal data width of 8bit x 4 / 16bit x 2 / 32bit blocks and the 32KByte sector is organized with four 8Kbyte blocks. The address is set in the ascending order as depicted by the SA number, SA0 - SA5.

14.3.3. Mode Select

This module selects an operation mode using the test mode signal (FLSH_HVACC & FLSH_HVDRS& FLSH_HVDWP) and a test register enable signal (GPIO[2]).
The mode selection method is as follows.

	test mode signal	test register enable	Source	Destination
CPU Mode	0	Don't care	AHB	Flash memory
Writer Mode	1	1	Writer (external chip)	Flash memory

14.3.3.1. CPU Mode

This module can execute read and write accesses from an AHB bus to flash memory by setting CPU mode
This mode is to be able to set the access timing by each clock cycle of the system.
Accesses supported are as follows:

- 16 bit sync write
- 32bit sync read (o_MRD32=1)

Note : Not supported: async mode (o_MSYNC=0) , byte access (ox_MBYTE=0).

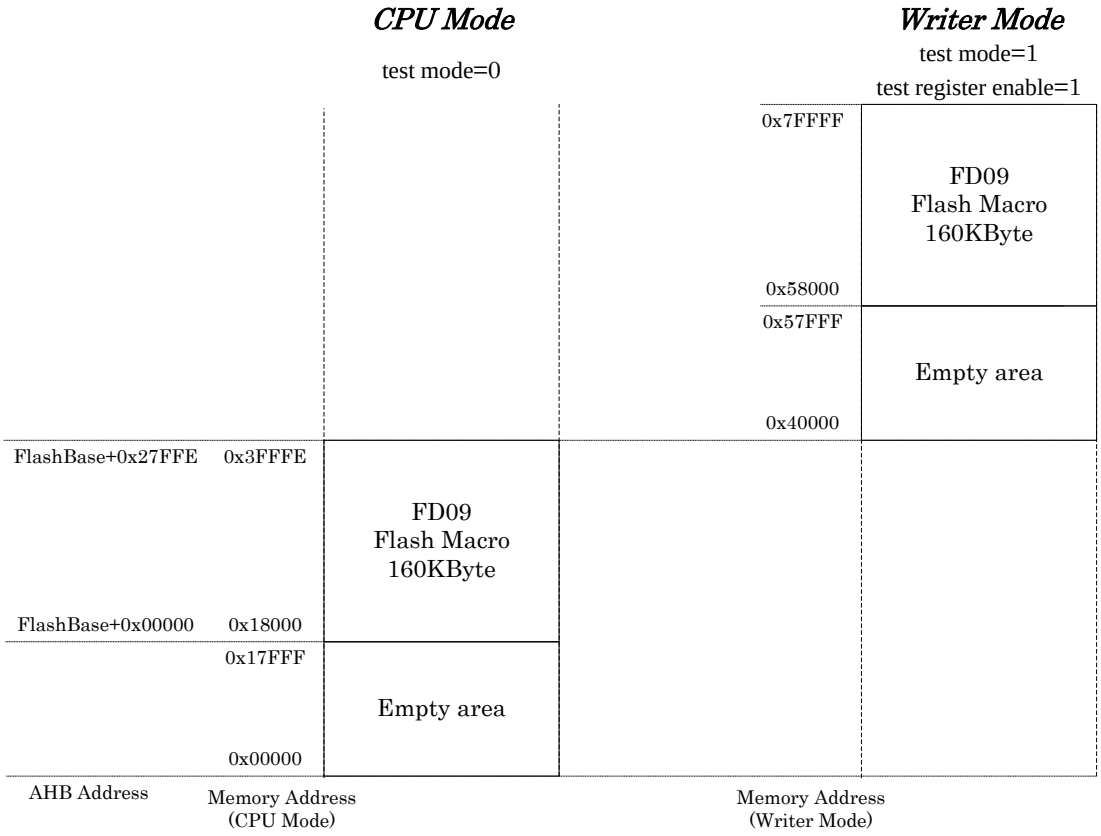
14.3.3.2. Writer Mode

This module supports read and write accesses to the embedded flash memory from a parallel writer by enabling Writer mode).
Accesses supported are as follows:

- 8/16 bit async read
- 8/16 bit async write

14.3.3.3. Memory Area

During access to flash memory, the access domain is different and depends on the active mode: CPU mode or Writer mode.
This module adds an offset to an address in the case of a CPU mode. (+0x18000)
A master can access the flash memory from "FlashBaseAddress+0x00000" over an AHB bus.
If reads are executed from a flash domain (a memory space) during writer mode, this module will output "1" for all read data.



14.3.4. Access Timing

14.3.4.1. CPU Mode

This module can set the memory access timing using the Memory Timing register (**MTIMING0 - MTIMING2**).
An example of the setting range of the register for access timing and an access wave pattern are shown below.
Please refer to the register details for the recommended setting points.

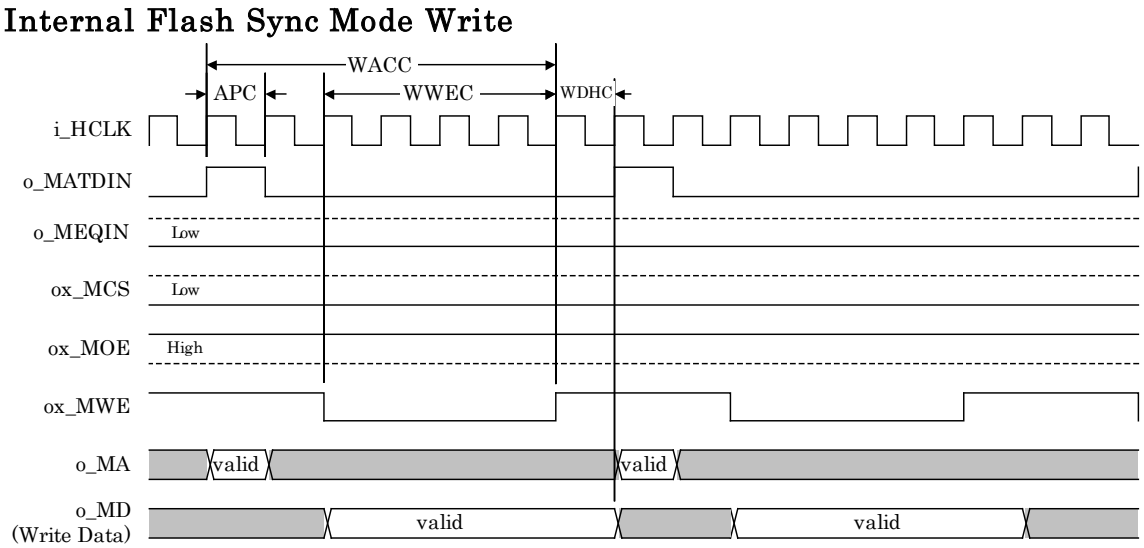


Figure 15.3-2 Flash Write Access

Register	Description	min	max
WACC	These bits set the number of cycles necessary for write access.	4T	35T
WWEC	These bits set the number of cycles for a write access to be effective.	2T	17T
APC	These bits set the number of cycles for an address valid signal (o_ATDIN) to be effective.	1T	16T
WDHC	These bits set the number of cycles for the write data hold time.	1T	16T

Note : The values set must meet the following size relationships. If these are not kept to, an interrupt will be issued.
WACC ＞ APC + WWEC

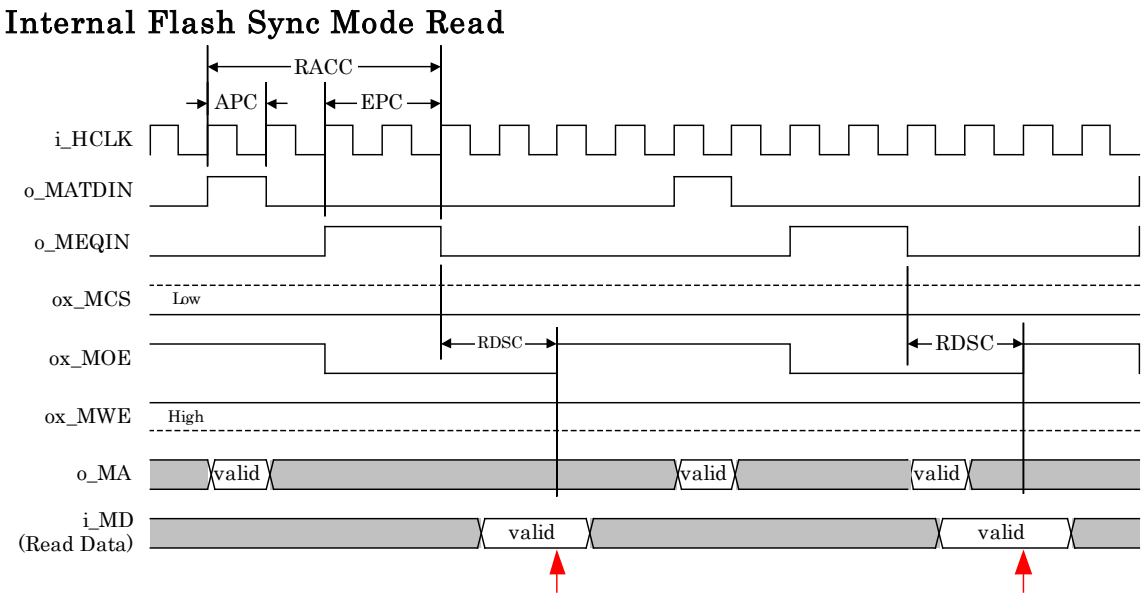


Figure 15.3-3 Flash Read Access

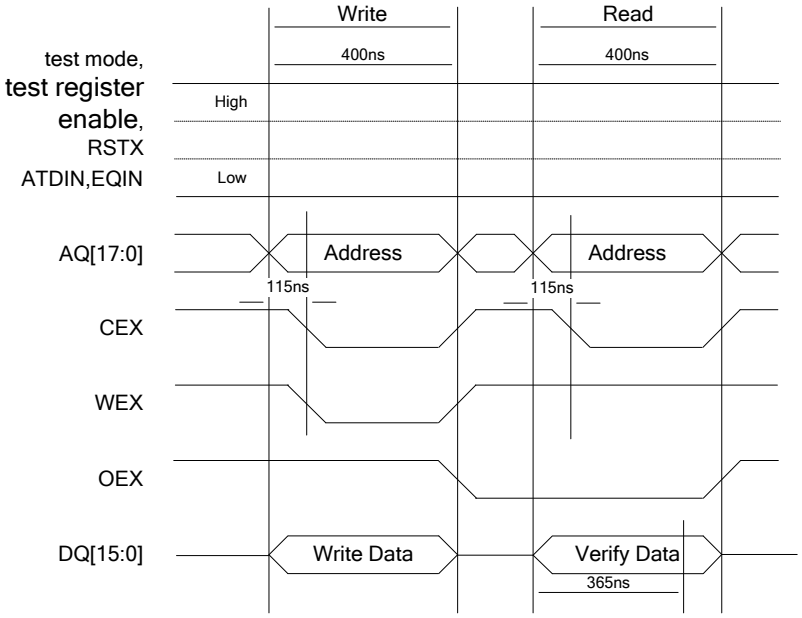
Register	Description	min	max
RACC	These bits set the number of cycles necessary for read access.	3T	18T
APC	These bits set the number of cycles for an address valid signal (o_ATDIN) to be effective.	1T	16T
EPC	These bits set the number of cycles for Data latch signal (o_EQIN) to be effective	1T	16T
RDSC	These bits set the number of cycles from Data latch signal (o_EQIN) falling edge to Read data valid.	1T	16T

Note : The values set must meet the following size relationships. If these are not kept to, an interrupt will be issued.

RACC ＞ APC + EPC

14.3.4.2. Writer Mode

For access timing in Writer Mode, please refer to the following.
Please refer to Chapter 15.5.2 for the connection with Flash Writer.



14.3.5. Reset in Flash algorithm

If a reset was initiated during the automatic programming algorithm state, access to the flash memory is not possible until the flash reset has been completed.
If an AHB master attempts to execute an access to flash memory before the reset has been completed, this module will keep HREADY Low until the reset has been completed and therefore stalls the access.

14.3.6. Reset release

If an access is executed by an AHB master within 200ns of a reset release, this module will keep HREADY Low for 200ns, stalling the access for this time.
The access begin margin from the rising edge of the FRSTX signal to read mode is more than 200ns.

14.3.7. Flash Internal Status

This module can read the status of the flash automatic algorithm via the Information register.

14.3.8. Interrupt

14.3.8.1. Interrupt Conditions

This module can output a normal interrupt (o_INT_MEMIF) or an abnormal interrupt (o_INT_MEMIF_ERR).

Normal interrupt output o_INT_MEMIF.

- The automatic programming algorithm of the flash memory was completed (Status Register bit8)

Abnormal interrupt output o_INT_MEMIF_ERR does a logical OR of the following factors and outputs accordingly:

- Violation of the timing parameter settings (Status Register bit0)
- Write access is not supported in the AHB Slave Data Interface (Status Register bit2)

It is necessary for a host CPU to check all the interrupt conditions up on the detection of an interrupt, o_INT_MEMIF, o_INT_MEMIF_ERR is controlled by Status Enable Register (STATUSEN).

14.3.8.2. Interrupt clear

Interrupts are cleared by writing a 1 to a corresponding bit that represents each interrupt factor in the Status register (STATUS).

14.4. Registers

14.4.1. Format of Register Descriptions

- Endian

  The registers of this module support Little Endian access only.

- Address

  “Address” shows the address of the register. (RegBase address + Offset address)

- Bit

  “Bit” shows the bit number of the register.

- Name

  “Name” shows the bit field name of the register.

- R/W

  “R/W” shows the attribute of Read/Write in each Bit field.
  R0: The reading value is always "0".
  R1: The reading value is always "1".
  W0: The writing value is always "0".When “1” is written, it is disregarded.
  W1: The writing value is always "1". When “0” is written, it is disregarded.
  R: Read
  W: Write

  Note : If a value is written to registers/bitfields that list R0, R1 and R in the following descriptions, then this value will not be changed in those registers/bitfields.

- Initial value

  “Initial value” is an initial value when reset is released.
  0: It becomes "0".
  1: It becomes "1".
  X: It becomes irregular.

14.4.2. Global Address

For the module base address please refer to the chapter 3 Memory map.

14.4.3. Register summary

Address	Register Name	Description
RegBase +0x000	STATUS	Status Register
RegBase +0x004	STATUSEN	Status Enable Register
RegBase +0x008	Reserved	-
RegBase +0x00C	IDLE	Idle Register
RegBase +0x010	INFO	Information Register
RegBase +0x014	MTIMING0	Memory Timing Register0
RegBase +0x018	MTIMING1	Memory Timing Register1
RegBase +0x01C	Reserved	-
~	~	~
RegBase +0xFFC	Reserved	-

14.4.4. Register Description

Only the access from an AHB bus is a possible register.

14.4.4.1. STATUS (Status Register)

Address	**RegBase Address** + 0x000															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved							ENDFA	Reserved					WACER	Reserved	TIMER
R/W	R0	R0	R0	R0	R0	R0	R0	R/W1	R0	R0	R0	R0	R0	R/W1	R0	R/W1
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit0 TIMER (TIMing parameter setting ERror interrupt)
This bit changes to 1 if parameter settings are violated.
Interrupt information is cleared by writing in 1 to this bit.
0: No interrupt
1: Interrupt

Bit2 WACER (AHB slave data interface Write ACcess ERror interrupt)
This bit changes to 1 if write access is not supported by the AHB Slave Data Interface.
This module simultaneously outputs ERROR to HRESP of the AHB Slave Data Interface.
Interrupt information is cleared by writing in 1 to this bit.
0: No interrupt
1: Interrupt

Bit8 ENDFA (END of Flash Automatic programming algorithm interrupt)
This bit changes to 1, when automatic programming algorithm of the Flash memory is completed.
Interrupt information is cleared by writing in 1 to this bit.
0: No interrupt
1: Interrupt

14.4.4.2. STATUSEN (Status Enable Register)

Address	**RegBase Address** + 0x004															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved							EFAEN	Reserved					WEREN	RESV	TEREN
R/W	R0	R0	R0	R0	R0	R0	R0	R/W	R0	R0	R0	R0	R0	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit0 TEREN (TIMER Enable)
This bit controls the output to o_INT_MEMIF_ERR from the TIMER bit of the Status Register.
0: Performs no output to o_INT_MEMIF_ERR
1: Performs output to o_INT_MEMIF_ERR

Bit1 RESV (Reserved Register)

Bit2 WEREN (WACER Enable)
This bit controls the output to o_INT_MEMIF_ERR from the WACER bit of the Status Register.
0: Performs no output to o_INT_MEMIF_ERR
1: Performs output to o_INT_MEMIF_ERR

Bit8 EFAEN (ENDFA Enable)
This bit controls the output to o_INT_MEMIF from the ENDFA bit of the Status Register.
0: Performs no output to o_INT_MEMIF
1: Performs output to o_INT_MEMIF

14.4.4.3. IDLE (Idle Register)

Address	**RegBase Address** + 0x00C															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved															IDLE
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit0 IDLE (IDLE status)
This bit shows the inside state of the MEMIF module.
0: IDLE state
1: Active state

14.4.4.4. INFO (Information Register)

Address	**RegBase Address** + 0x010															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved															RDY
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit0 RDY (RDY status)
This bit shows the state of the RDY terminal of the flash memory.

0: READY — Automatic programming algorithm stop state. Accesses to the flash are accepted.

1: BUSY — Automatic programming algorithm state. Only read accesses to the flash are accepted. (Output is verify / status data.)

14.4.4.5. MTIMING0 (Memory Timing Register0)

Address	**RegBase Address** + 0x014															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved											WACC				
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved				WDHC				Reserved				WWEC			
R/W	R0	R0	R0	R0	R/W	R/W	R/W	R/W	R0	R0	R0	R0	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit3-0 WWEC (Write Enable Cycle)
These bits set the number of cycles for write enable to become effective.
0: 2 cycles
| |
15: 17 cycles

Bit11-8 WDHC (Write Data Hold Cycle)
These bits set the number of cycles for the write data hold time.
0: 1 cycles
| |
15: 16 cycles

Bit20-16 WACC (Write ACcess Cycle)
These bits set the number of cycles necessary for write access.
0: 4 cycles
| |
31: 35 cycles

Note : If you change the setting, please confirm that a stop condition exists using the IDLE register.
Note : The set values must fulfill the following size relationships:
WACC > APC + WWEC

14.4.4.6. MTIMING1 (Memory Timing Register1)

Address	**RegBase Address** + 0x018															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved				RACC				Reserved				RDSC			
R/W	R0	R0	R0	R0	R/W	R/W	R/W	R/W	R0	R0	R0	R0	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved				EPC				Reserved				APC			
R/W	R0	R0	R0	R0	R/W	R/W	R/W	R/W	R0	R0	R0	R0	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit3-0 APC (Atdin Pulse Cycle)
These bits set the number of cycles for the address valid signal (o_ATDIN) to become effective.
0: 1 cycle
| |
15: 16 cycles

Bit11-8 EPC (Eqin Pulse Cycle)
These bits set the number of cycles for the Data latch signal (o_EQIN) to become effective.
0: 1 cycle
| |
15: 16 cycles

Bit19-16 RDSC (Read Data Setup Cycle)
These bits set the number of cycles from Data latch signal (o_EQIN) falling edge to Read data valid.
0: 1 cycle
| |
15: 16 cycles

Bit27-24 RACC (Read ACcess Cycle)
These bits set the number of cycles necessary for read access.
0: 3 cycles
| |
15: 18 cycles
Note : If you change the setting, please confirm that a stop condition exists using the IDLE register.
Note : The set values must fulfill the following size relationships:
RACC > APC + EPC

Recommended Memory Timing register settings are shown below.

Table 15-1 Recommended access timing settings

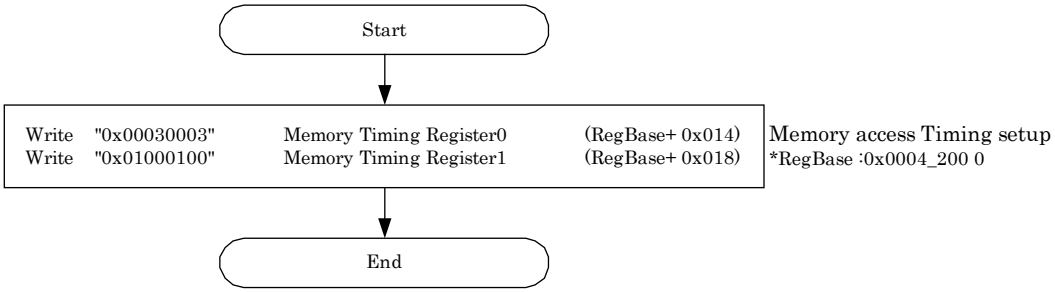
Register	83.33MHz (12.0ns) initial	41.66MHz (24.0ns)	20.83MHz (48.0ns)
APC	0: 1cycle	0: 1cycle	0: 1cycle
EPC	1: 2cycle	0: 1cycle	0: 1cycle
RACC	1: 4cycle	0: 3cycle	0: 3cycle
RDSC	0: 1cycle	0: 1cycle	0: 1cycle
WEC	3: 5cycle	1: 3cycle	0: 2cycle
WACC	3: 7cycle	1: 5cycle	0: 4cycle
WDHC	0: 1cycle	0: 1cycle	0: 1cycle

14.5. Application notes

14.5.1. Processing flow

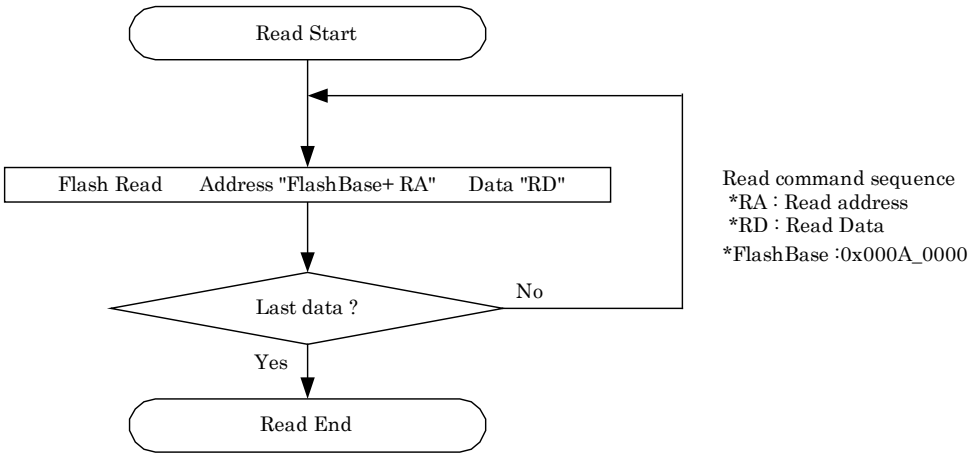
14.5.1.1. Register initialization flow

An initial setting example of a register is as follows.
After power-on reset release, please set register before attempting access to the memory device.
Note：Confirm stop condition via the IDLE register before changing settings



14.5.1.2. Flash Read flow

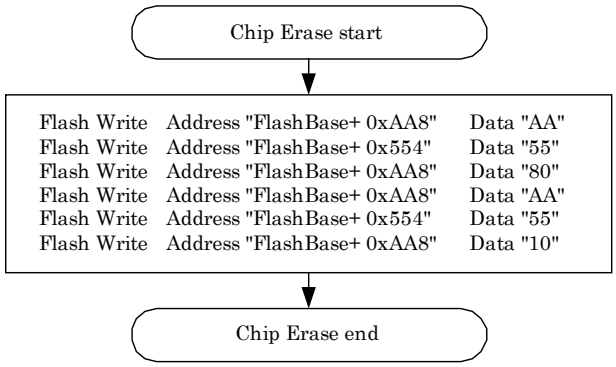
The flash READ flow is as follows.



14.5.1.3. Flash Erase flow

The flash ERASE flow is as follows.
Please execute an ERASE before programming the flash.

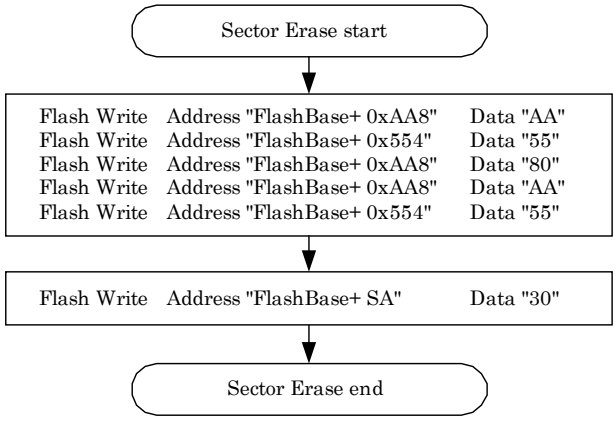
14.5.1.3.1. Chip Erase flow



Chip Erase command sequence

*FlashBase :0x000A_0000

14.5.1.3.2. Sector Erase flow



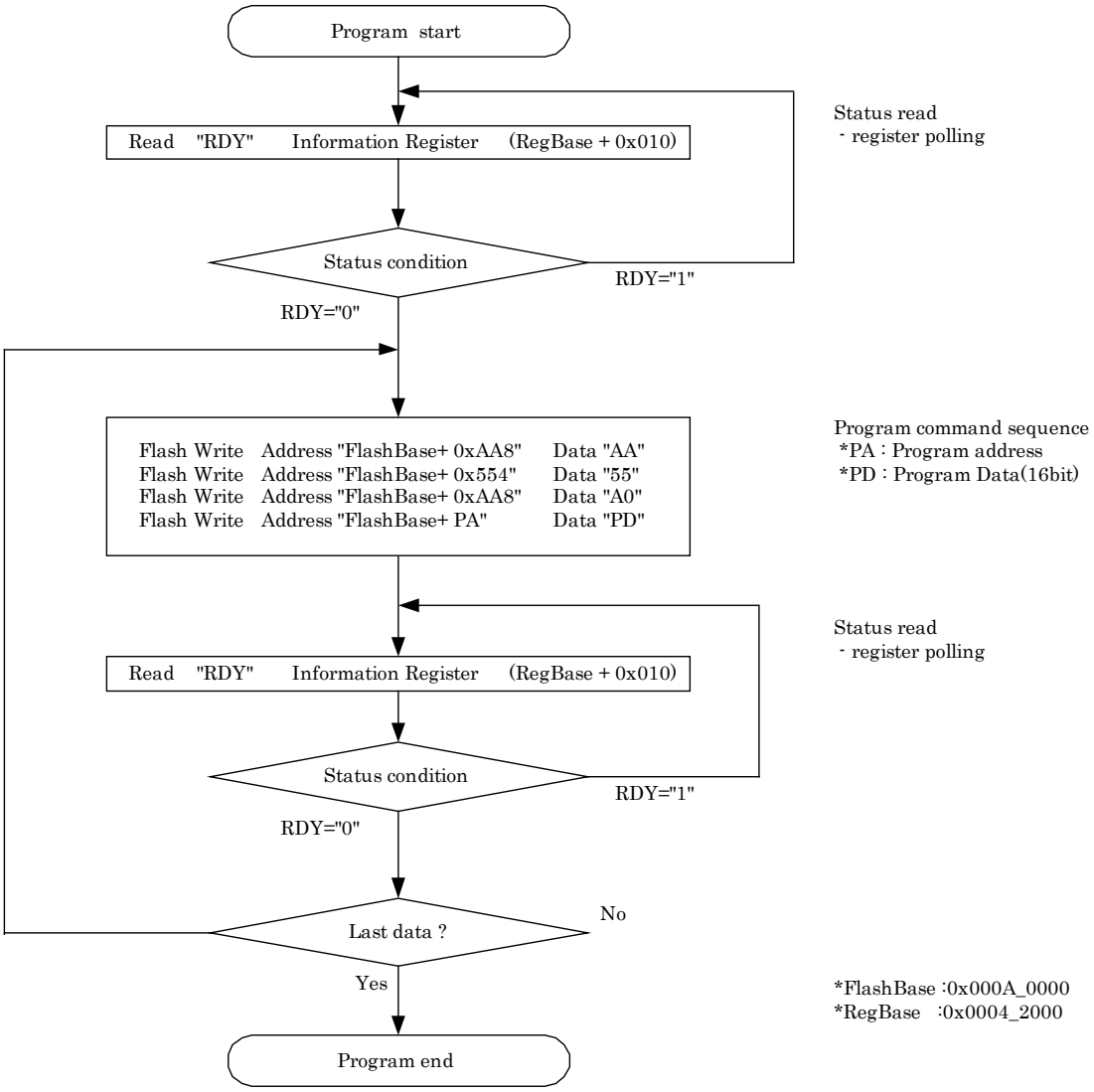
Sector Erase command sequence

*SA :Any address in the sector

*FlashBase :0x000A_0000

14.5.1.4. Flash Programming flow

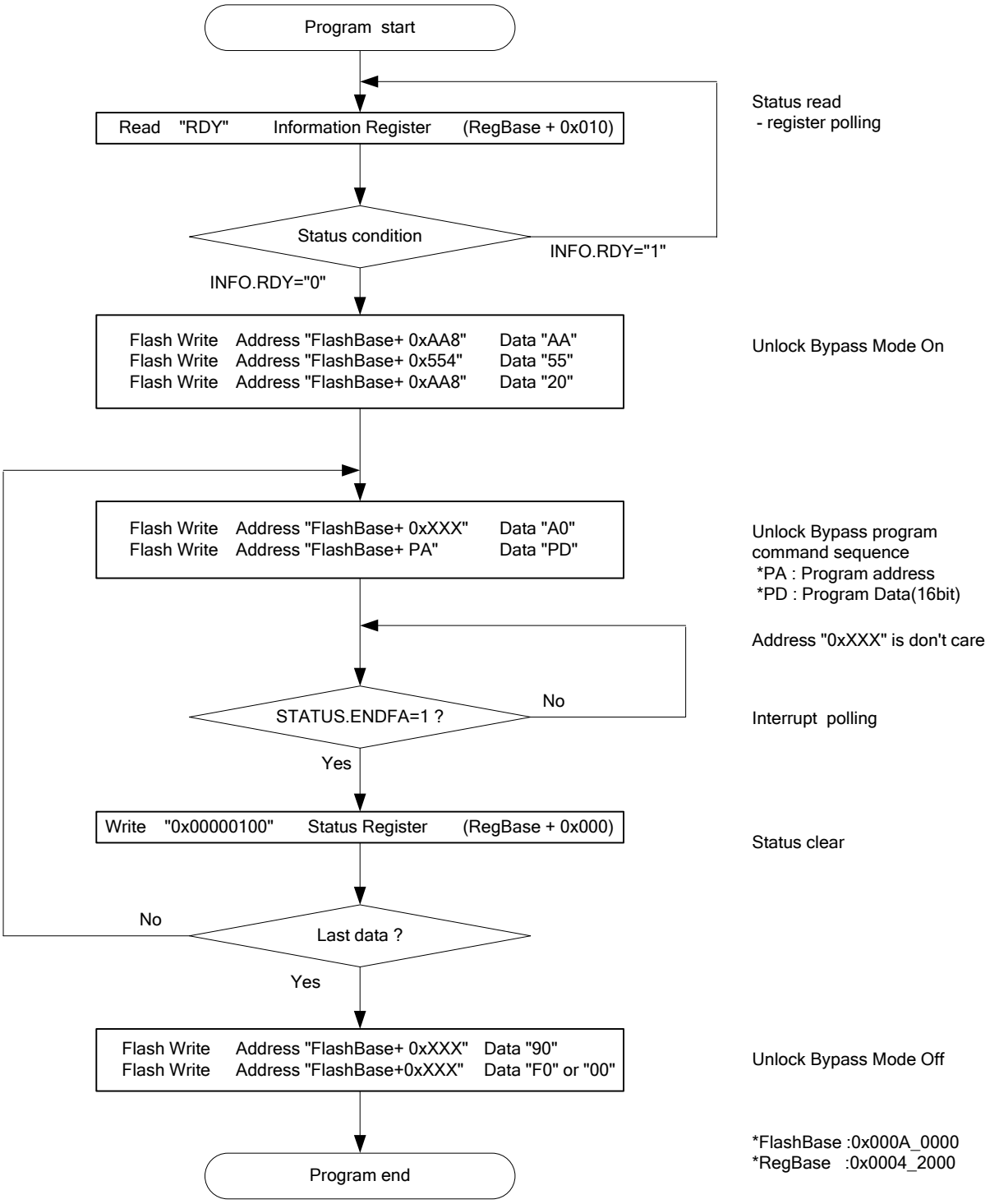
This module keeps the internal bus locked until the automatic programming algorithm has been completed. Application SW can prevent internal bus lock by polling the RDY bit of the Information register or by use of "automatic programming algorithm completion interrupt".
A flow using the status confirmation of the RDY bit is shown as follows.



Note: Data cannot be overwrited in the same address.
Please write after executing Erace.

14.5.1.5. Flash Unlock Bypass Program flow

The Unlock Bypass Program can shorten the programming sequence.
It is not possible to read from the flash in Unlock Bypass Mode.
A flow using the automatic programming algorithm completion interrupt is shown as follows.



Note: Data cannot be overwrited in the same address.
Please write after executing Erace.

14.5.2. Connection example of Flash writer mode

Please refer to the following application notes "Application Note (Flash Memory Parallel Writer Specification)"
URL: http://www.fujitsu.com/emea/services/microelectronics/gdc/gdcdevices/mb88f332-indigo.html
This specification is written for Indigo, but also applies to Indigo-L.

14.6. AC characteristics

14.6.1. Synchronous Read

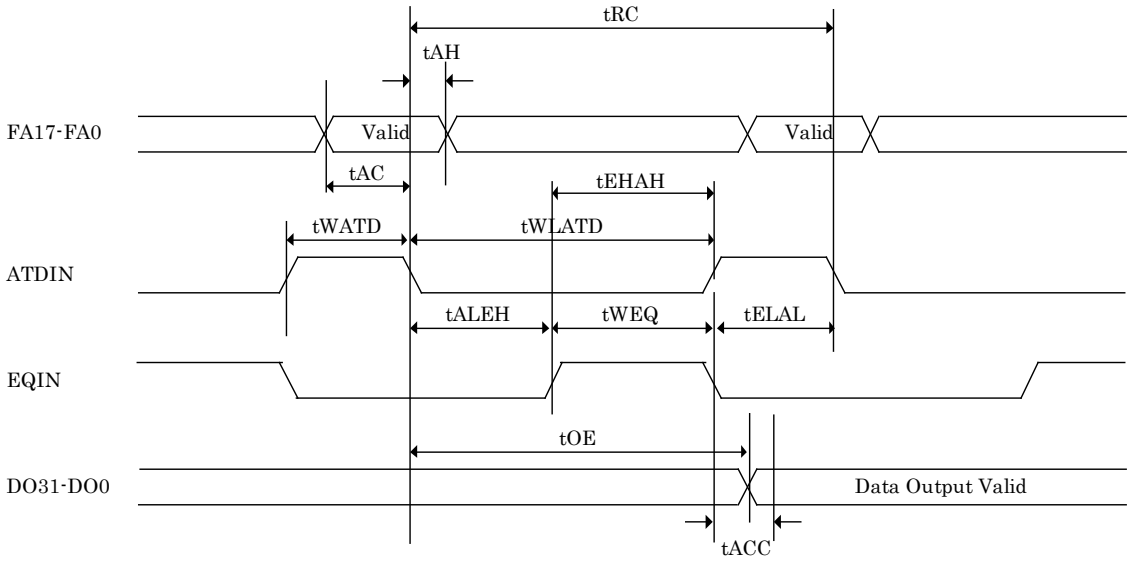


Figure 15.6-1 Synchronous Read

Parameter	Symbol	Flash AC (Unit: ns)	83.33MHz (12.0ns) initial	41.66MHz (24.0ns)	20.83MHz (48.0ns)
Read cycle time	tRC	39.5	72.0	120.0	240.0
Address setup time	tAC	5	12.0	24.0	48.0
Address hold time	tAH	1.4	60.0	96.0	192.0
ATDIN pulse width (HIGH width)	tWATD	9.5	12.0	24.0	48.0
EQIN pulse width (HIGH width)	tWEQ	19.5	24.0	24.0	48.0
ATDIN pulse width (LOW width)	tWLATD	19.5	60.0	96.0	192.0
EQIN rise edge from ATDIN fall edge	tALEH	9.5	12.0	24.0	48.0
ATDIN rise edge from EQIN rise edge	tEHAH	9.5	48.0	72.0	144.0
ATDIN fall edge from EQIN fall edge	tELAL	9.5	36.0	72.0	144.0
Output data valid from EQIN fall edge	tACC	4.2(max)	-	-	-
Output data hold time (From ATDIN fall edge)	tOE	29	48	72.0	144.0

14.6.2. Synchronous Write

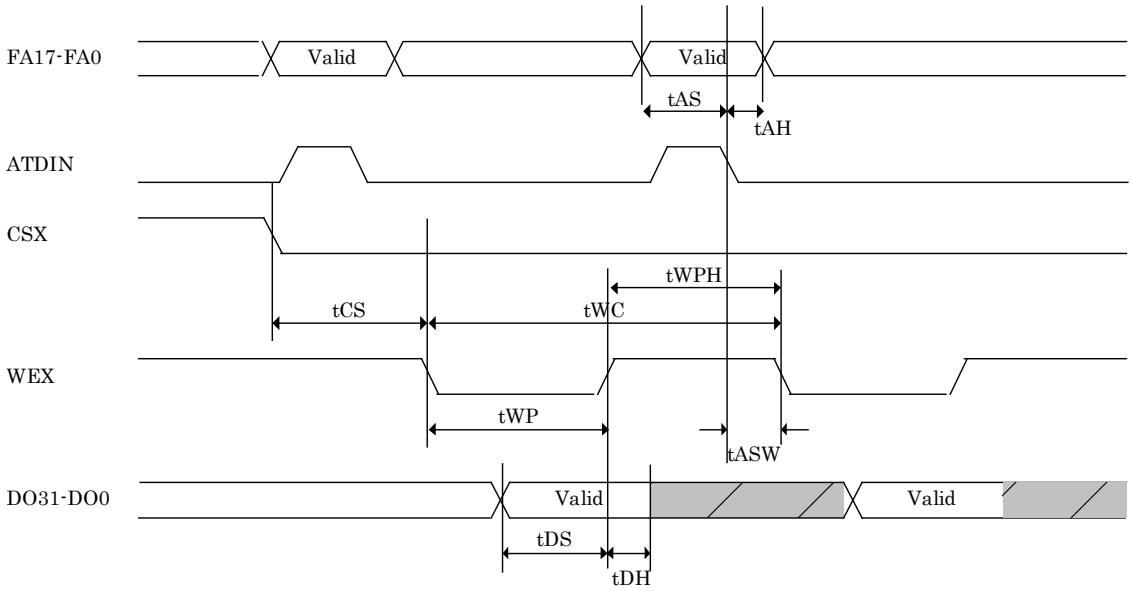


Figure 15.6-2 Synchronous Write

Parameter	Symbol	Flash AC (Unit: ns)	83.33MHz (12.0ns) initial	41.66MHz (24.0ns)	20.83MHz (48.0ns)
Write cycle time	tWC	80	120.0	192.0	336.0
Address setup time	tAS	5	12.0	24.0	48.0
Address hold time	tAH	1.4	108.0	144.0	288.0
Data setup time	tDS	45	72.0	96.0	144.0
Data hold time	tDH	0	48.0	96.0	192.0
WEX fall edge from ATDIN fall edge	tASW	5	12.0	24.0	48.0
Output enable hold time OEX fall edge from WEX rise edge	tOE	10	60.0	120.0	240.0
CEX setup time	tCS	0	-	-	-
WEX pulse width (LOW width)	tWP	50	60	72.0	96.0
WEX pulse width (HIGH width)	tWPH	30	60	120.0	240.0
Latency Between Read and Write Operations	tSR/W	14.4	36.0	72.0	144.0

14.6.3. Programming/Erase characteristics

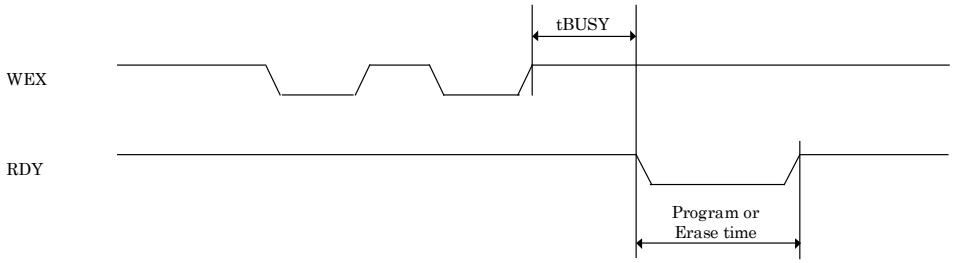


Figure 15.6-3 Program/Erase characteristics

Parameter	Symbol	Min	Typ	Max	Unit
Programming/Erase valid to RDY delay	tBUSY	90	-	-	ns
Sector erase time	-	-	0.9	3.6	sec
Word programming time	-	-	23	370	us

15. Embedded SRAM

This chapter describes in the embedded SRAM of the MB88F333.

15.1. Outline

This MB88F333 has 128KB of SRAM used to store sprite data and 8KB of SRAM used to store command lists (display lists).

15.2. Features

The internal RAM has the following features.

15.2.1. Features

- Operation as bus slave of AMBA (AHB)
- The three embedded SRAMs are simultaneously accessible from different 3 AHB masters

15.2.2. Limitations

15.3. Function

15.3.1. Block diagram

Figure 16-1shows the internal SRAM's block diagram.

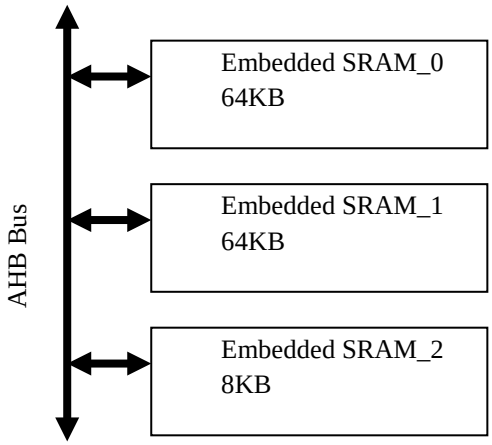


Figure 16-1 Block diagram of Internal SRAM

16. Run-Length Decompression (RLD)

This chapter describes the Run-Length Decompression unit of the MB88F333.

16.1. Outline

The Run-Length Decompression unit (RLD) allows unpacking of run length compressed data without processor interaction by pure HW means. For simple graphic content such as pictograms, company image (logo) run-length encoding brings a substantial benefit for bandwidth saving at the used external bus system especially in the setup phase. Precondition for processor load relieving is, that the input data complies with the data format and organization needed in the target location for uncompressed data (e.g. video RAM, sprite pattern RAM), so that no further reformatting is necessary.

The RLD unit consists of a decompression module, an AHB slave, used for configuration data and compressed data input, and an AHB master for decompressed data output. This allows the decompressing of data delivered by arbitrary AHB modules of the system. Also the target location may be an arbitrary AHB slave of the system.

16.2. Features

The RLD has following features.

- Support of a simple run length compressing format (TGA™-like format)
- 1/2/4/8/16/24/32 bit per pixel formats supported
- AHB master for data output
- FIFO for data input and output, allows burst access of AHB

16.2.1. Limitations

None

16.2.2. References

Truevision™ TGA™ FILE FORMAT SPECIFICATION, Truevision Inc.Version 2.0

16.2.3. Integration and Application Hints

16.2.3.1. Use of the MB88F333's RLD unit

Compressed data delivered via the APIX Sideband link or SPI link is transferred by the corresponding AHB master of these link interfaces to the AHB slave of the RLD unit. Target location is the Pattern SRAM.

RLD can not be used to decompress data on the fly for delivering sprite pattern data for the sprite engine.

16.3. Function

16.3.1. Block diagram

Figure 17-1 shows a block diagram of the RLD unit.

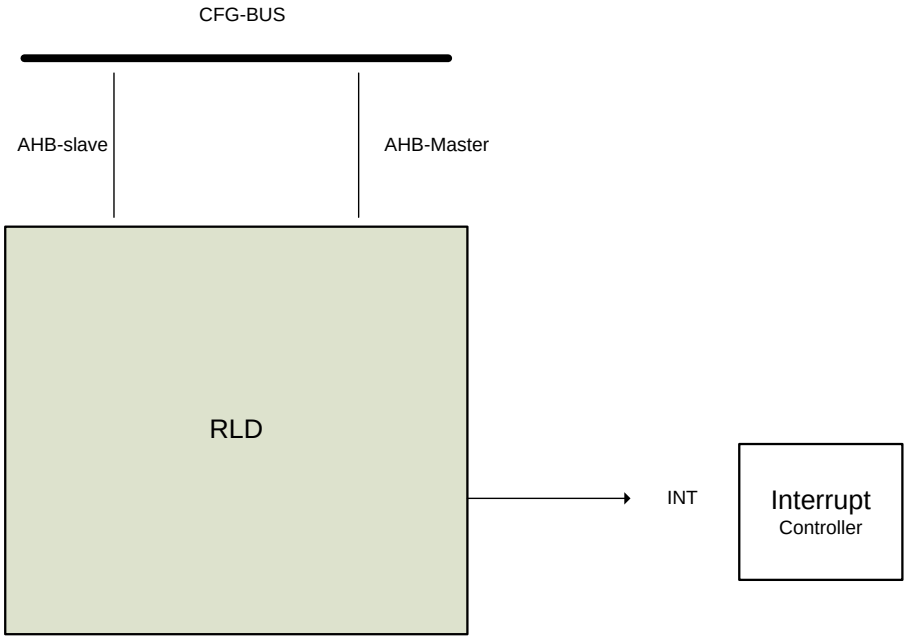


Figure 17-1 Position of RLD

16.4. Data Formats

16.4.1. Input Data Format

The following table describes the supported data formats.

Format [bit per pixel]	**Mode**	**Command Byte** <MSB LSB>	**Color Bytes** <MSB LSB>
1bpp	Compressed	<1NNN NNNN>	Color bit <C> (bit-aligned
	uncompressed	<0NNN NNNN>	(NNN_NNNN +1) bits with color data <C…C> (bit-aligned)
2bpp	Compressed	<1NNN NNNN>	Color bit <CC> (bit-aligned)
	uncompressed	<0NNN NNNN>	(NNN_NNNN+1) x2 bits with color data <CC…CC> (bit-aligned)
4bpp	Compressed	<1NNN NNNN>	Color bits<CCCC> (bit- aligned)
	uncompressed	<0NNN NNNN>	(NNN_NNNN+1) x4 bits with color data <CCCC…CCCC> (bit-aligned)
8bpp	Compressed	<1NNN NNNN>	1 color byte <CCCC CCCC>
	uncompressed	<0NNN NNNN>	(NNN_NNNN +1) bytes with color data <CCCC CCCC>
16bpp[1]	Compressed	<1NNN NNNN>	2 bytes color data (<CCCC CCCC>,<CCCC CCCC>)
	uncompressed	<0NNN NNNN>	(NNN_NNNN +1)*2 bytes with color data (<CCCC CCCC>,<CCCC CCCC>)
24bpp	Compressed	<1NNN NNNN>	3 bytes color data (<CCCC CCCC>,<CCCC CCCC>, <CCCC CCCC>)
	uncompressed	<0NNN NNNN>	(NNN_NNNN +1)*3 bytes with color data (<CCCC CCCC>,<CCCC CCCC>, <CCCC CCCC>)
32bpp	Compressed	<1NNN NNNN>	4 bytes color data (<CCCC CCCC>,<CCCC CCCC>, <CCCC CCCC>,<CCCC CCCC>)
	uncompressed	<0NNN NNNN>	(NNN_NNNN +1)*4 bytes with color data (<CCCC CCCC>,<CCCC CCCC>, <CCCC CCCC>,<CCCC CCCC>)

For all modes applies:
Compressed data “<1NNN NNNN> <pixel>” is decompressed to (<NNN NNNN> + 1) pixels.

[1] May be used for e.g. RGB555 or RGB565

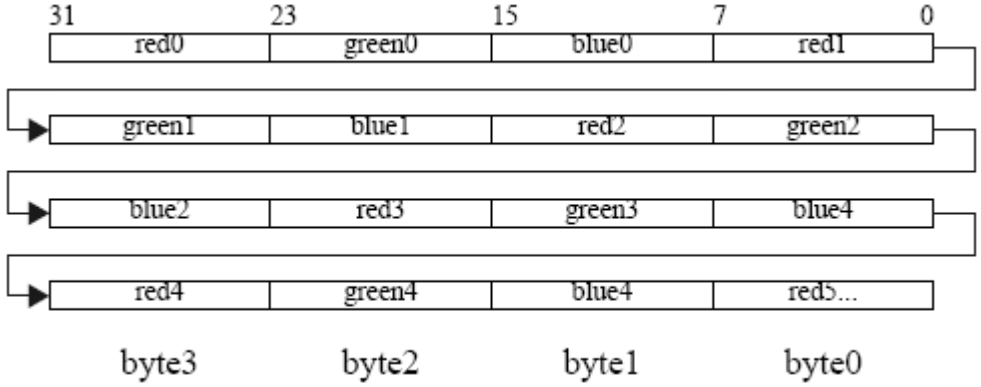


Figure 17-2 Example for uncompressed data input RGB888

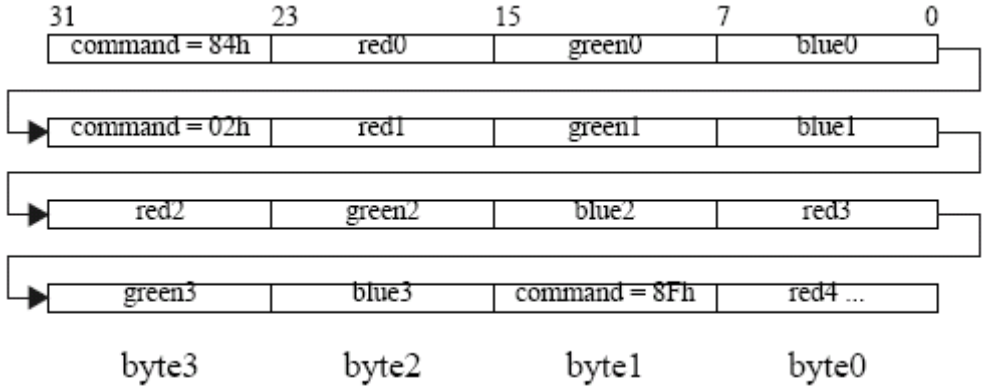


Figure 17-3 Example for compressed data input RGB888

16.4.2. Output Data Format

Output data format depends on the selected BPP (bit per pixel) format. Further bit/word alignment, memory stride calculation is supported in HW. The output data is organized Little Endian.

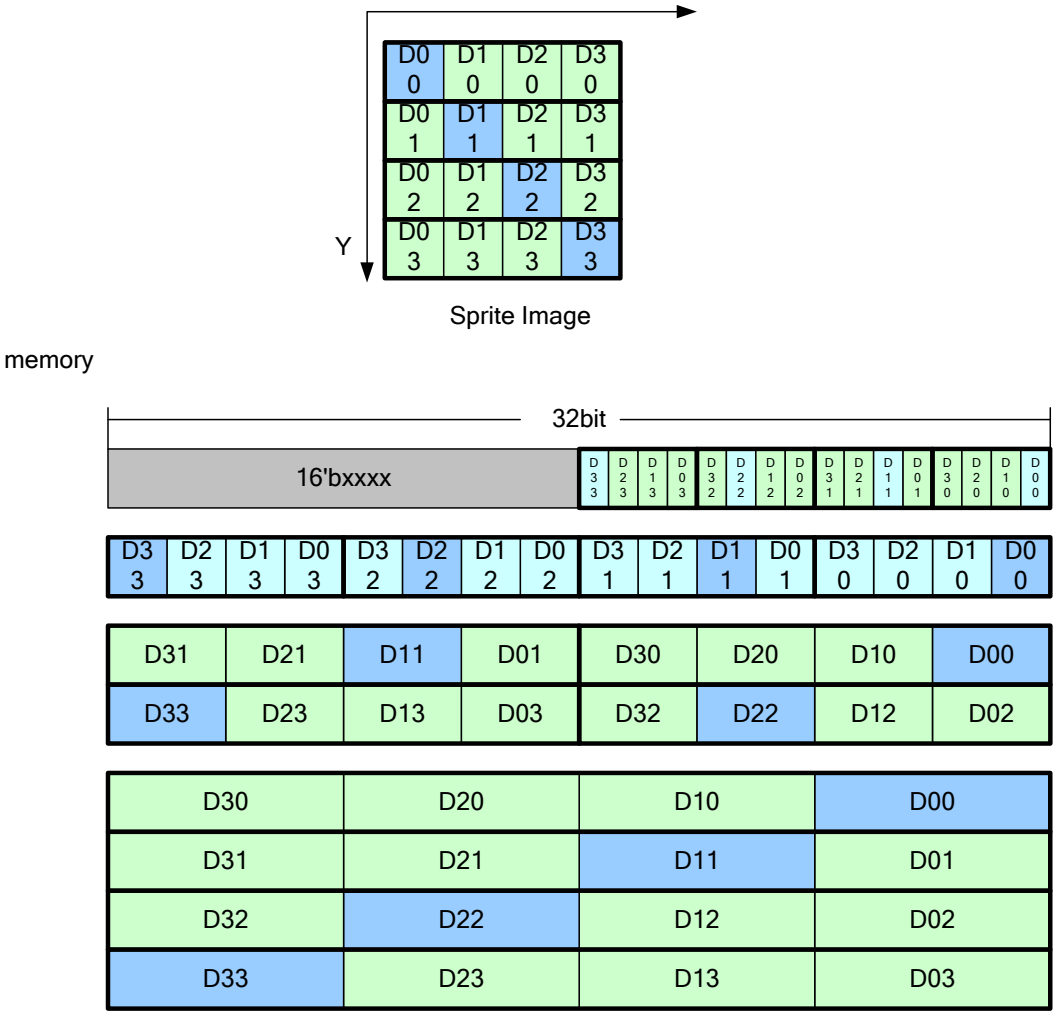


Figure 17-4 Memory organization for Sprite Engine

16.4.2.1. Memory Stride

Please see next figure for explanation of the stride calculation in HW.

16.4.3. Notes

HINT:
8bit/pixel can be used for blend factor layer or indirect color mode (RGB666 color palette)

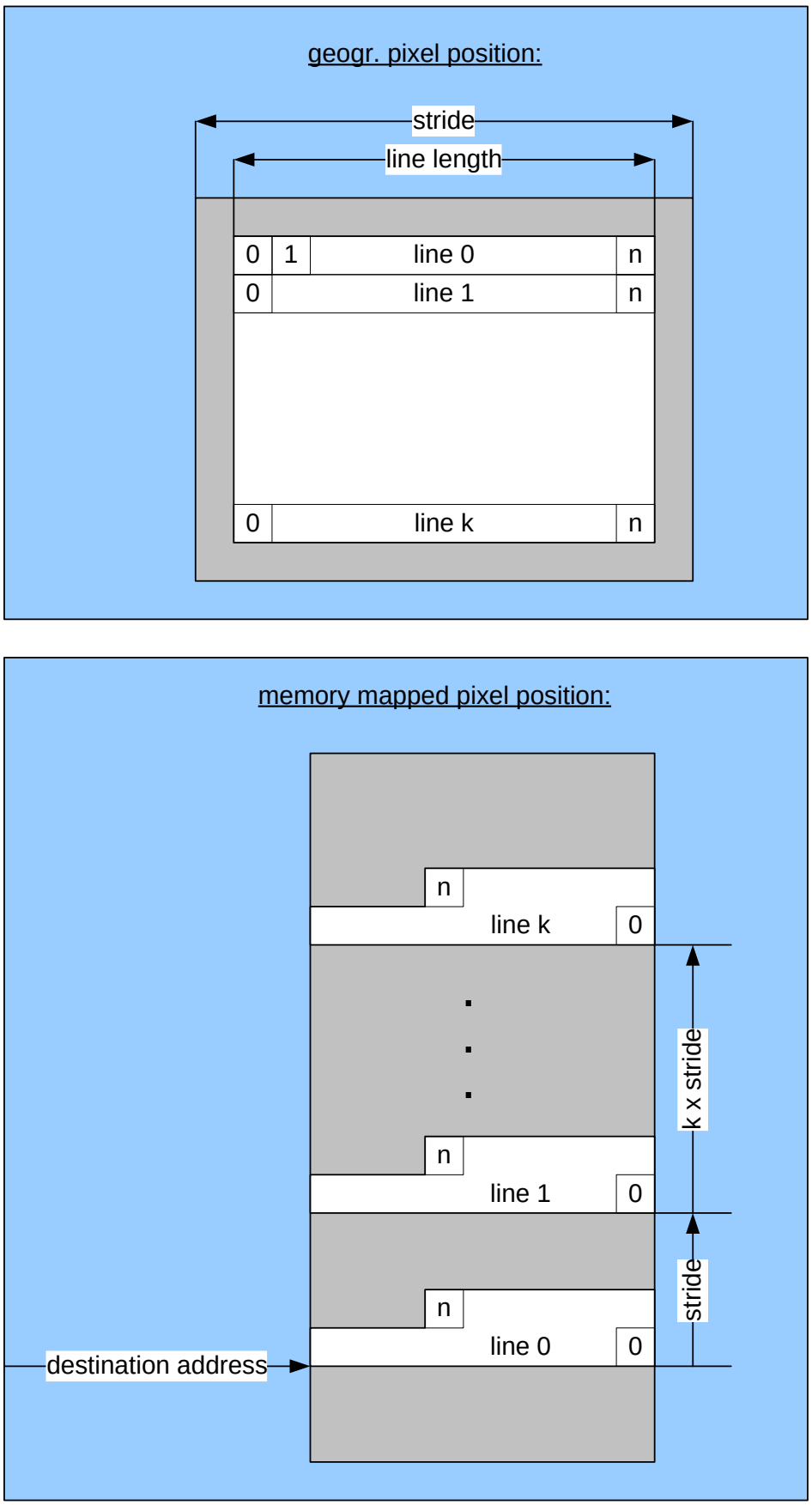


Figure 17-5 Memory Stride

16.5. Registers

16.5.1. Format of Register Descriptions

The register descriptions in the following sections use the format shown below to describe each bit field of a register.

Register address	Offset																													
Bit number	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
Field name																																
R/W																																
Reset value																																

Meaning of items and sign

Register Address

Register address shows the address (Offset address) of the register.

Bit number

Bit number shows bit position of the register.

Field Name

Field name shows bit name of the register.

R/W

R/W shows the read/write attribute of each bit field:

- R: Read Only
- W: Write Only
- W1C: Write a value of “1” clears the register

Reset value

Reset value indicates the value of each bit field immediately after reset.

- 0: Initial value is "0".
- 1: Initial value is "1".
- X: Undefined.

Unused register fields are marked with a solid grey background.

Bit vectors are unsigned integers, if nothing else specified.

16.5.2. Global Address

For the module base address please refer to the chapter 3 Memory map.

16.5.3. Register summary

Address	Register Name	Description
Base address + 0_H	SWReset	SW reset
Base address + 4_H	RldCfg	general configuration register
Base address + 8_H	StrideCfg0	Stride general configuration register
Base address + C_H	StrideCfg1	Line / Stride Length
Base address + 10_H	BYTECNT	Target number of decompressed bytes
Base address + 14_H	OFIFO	Output FIFO Control
Base address + 18_H	DestAddress	Local AHB-master transfer Destination address
Base address + $1C_H$	AHBMCtrl	Local AHB-master transfer Configuration/Control
Base address + 20_H	RLDCtrl	General Control
Base address + 24_H	IEN	Interrupt Enable register
Base address + 28_H	ISTS	Interrupt status flags, a '1' signifies that the corresponding interrupt condition occurred (even if interrupt is disabled), write '1' clears the flag,
Base address + $2C_H$	Status	Status register
Base address + 30_H	SAHBData	AHB Slave Input Data
Base address + 34_H	TransferCount	Local AHB-master transfer count
Base address + 38_H	CurAddress	Local AHB-master transfer Current address

16.5.4. Register Description

16.5.4.1. SWReset

Register address	BaseAddress + 0_H																													
Bit number	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
Field name																																SWReset
R/W																																RW
Reset value																																0_H

SW reset
Bit 0 SWReset
sw reset (flush all FIFOs)

16.5.4.2. RldCfg

Register address	BaseAddress + 4_H																													
Bit number	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
Field name																								Align						BPP		
R/W																								RW						RW		
Reset value																								0_H						0_H		

general configuration register
Bit 8 Align Mode
output data format 0b=bit aligned output 1b=word (32bit) aligned output
Bit 2 - 0 BPP
Bit per pixel, 000b=1, 001b=2, 010b=4, 011b=8, 100b=16, 101b=24, 110b=32 others=reserved

16.5.4.3. StrideCfg0

Register address	BaseAddress + 8_H																													
Bit number	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
Field name																																StrideEn
R/W																																RW
Reset value																																0_H

Stride general configuration register
Bit 0 StrideEn
Enable for output data stride aligned, 0b=disabled (no observation of LineLength and Stride, needed for 4x4 1bpp sprites), 1b=enabled

16.5.4.4. StrideCfg1

Register address	BaseAddress + C_H																													
Bit number	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
Field name	Stride																		LineLength													
R/W	RW																		RW													
Reset value	0_H																		0_H													

Line / Stride Length
Bit 31 - 16 Stride
Stride: number of byte -1
Bit 13 - 0 LineLength
number of bytes per line - 1

16.5.4.5. BYTECNT

Register address	BaseAddress + 10_H																													
Bit number	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
Field name	ByteCnt																															
R/W	RW																															
Reset value	0_H																															

Target number of decompressed bytes
Bit 31 - 0 ByteCnt
Target number of decompressed bytes -1 (granularity of decompressed size is byte, be aware for 1bpp, 2bpp and 4bpp format)

16.5.4.6. OFIFO

Register address	BaseAddress + 14_H																													
Bit number	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
Field name																																WriteThreshold
R/W																																RW
Reset value																																0_H

Output FIFO Control
Bit 3 - 0 WriteThreshold
number of words-1 after which a write burst is initialized

16.5.4.7. DestAddress

Register address	BaseAddress + 18_H																													
Bit number	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
Field name	AHBMDA																												reserved			
R/W	RW																												RW			
Reset value	0_H																												0_H			

Local AHB-master transfer Destination address (byte address)
Bit 31 - 2 AHBMDA
Destination address to start AHB-master transfer (word address)
Bit 1 - 0 reserved
Do not change

16.5.4.8. AHBMCtrl

Register address	BaseAddress + $1C_H$																													
Bit number	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
Field name								Reserved0												AHBMTransferWidth							AHBMFixedDest			
R/W								RWS												RW							RW			
Reset value								0_H												0_H							0_H			

Local AHB-master transfer Configuration/Control
Bit 23 - 16 Reserved0
Bit 9 - 8 AHBMTransferWidth
00b=byte, 01b=halfword, 10b=word, 11b=reserved
Bit 0 AHBMFixedDest
0b=destination address is incremented, 1b=destination address is fixed

16.5.4.9. RLDCtrl

Register address	BaseAddress + 20_H																													
Bit number	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
Field name																												AcceptData				
R/W																												RW				
Reset value																												0_H				

General Control
Bit AcceptData
0 Enable the receipt of compressed data, reset by the hardware after completion, 0b=received data is discarded, 1b=received data will be processed

16.5.4.10. IEN

Register	BaseAddress + 24_H

address																															
Bit number	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
Field name																									IEnIFfull		IEnIFempty			IEnError	IEnComplete	
R/W																									RW		RW			RW	RW	
Reset value																									0_H		0_H			0_H	0_H	

Interrupt Enable register
Bit 3 IEnIFfull
Interrupt enable
Bit 2 IEnIFempty
Interrupt enable
Bit 1 IEnError
Interrupt enable
Bit 0 IEnComplete
Interrupt enable

16.5.4.11. ISTS

Register address	BaseAddress + 28_H																															
Bit number	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
Field name																									ISTsIFfull		ISTsIFempty				IStsError	IStsComplete
R/W																									RW		RW				RW	RW
Reset value																									0_H		0_H				0_H	0_H

Interrupt status flags, a '1' signifies that the corresponding interrupt condition occurred (even if interrupt is disabled), write '1' clears the flag,
Bit 3 IStsIFfull
Interrupt Status for condition Input FIFO full
Bit 2 IStsIFempty
Interrupt Status for condition Input FIFO empty
Bit 1 IStsError
Interrupt Status for condition AHB Destination access error (info from AHB HRESP)
Bit 0 IStsComplete
Interrupt Status for Condition RLD complete (Target Byte Count reached)

16.5.4.12. Status

Register address	BaseAddress + $2C_H$

Bit number	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
Field name																									Reserved				IFIFOempty		OFIFOfull	IFIFOfull	Busy
R/W																									RWS				R		R	R	R
Reset value																									0				1		0	0	0

Status register
Bit 4 Reserved
Bit 3 IFIFOempty
Input FIFO currently empty
Bit 2 OFIFOfull
Output FIFO currently full
Bit 1 IFIFOfull
Input FIFO currently full
Bit 0 Busy
RLD busy

16.5.4.13. SAHBData

Register address	BaseAddress + 30_H
Bit number	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
Field name	InData
R/W	RW
Reset value	0_H

AHB Slave Input Data
Bit 31 - 0 InData
RLD input data (Data written at this address is latched into the RLD IFIFO)
Please access the SAHBData register after setting various registers.

16.5.4.14. TransferCount

Register address	BaseAddress + 34_H
Bit number	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
Field name	AHBMTransferCount
R/W	R
Reset value	0_H

Local AHB-master transfer count
Bit 31 - 0 AHBMTransferCount
status count of remaining bytes to transfer during current transaction (decrementing counter)

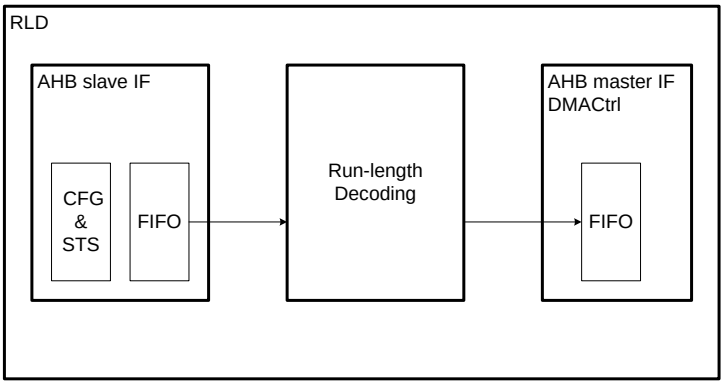
16.5.4.15. CurAddress

Register address	BaseAddress + 38_H
Bit number	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
Field name	AHBMCA
R/W	R
Reset value	X

Local AHB-master transfer Current address
Bit 31 - auto AHBMCA
Current Destination address

16.6. Processing Mode

16.6.1. Processing Flow



16.6.2. Processing Algorithm

16.6.2.1. Processing Modes

RLD supports only two processing modes – ‘in operation’ mode and ‘disabled’ mode.

16.7. Control Flow

16.7.1. Example Control Flow

1) Microcontroller decodes TGA header info:
- BitPerPixel
- Source size
- picture dimensions --> number of decompressed bytes

Configuration of RLD:
2) reset OFIFO, IFIFO
3) set bpp-format
4) set target number of bytes
5) Configure AHB master IF
- e.g. Dest. Address in Video Memory

6) Enable RLD

Deliver compressed data to RLD AHB slave:
Case A: compressed data source is AHB slave:
1) configure DMA transfer:
- set source (e.g. flash)
- set DMA destination: RLD

2) start DMA transfer
3) interrupt after completion

Case B: compressed data source is AHB master
1) Write actively data to RLD AHB slave
2) interrupt after completion

17. DMA Controller (DMAC)

This chapter describes the DMA Controller of the MB88F333.

17.1. Outline

The DMA Controller has 2 channels.

17.2. Features

The DMAC has the following features.

17.2.1. Features

- 2 DMA Channels
- Software request (start-up by a register write)
- Beat transfer
  16 word FIFO shared by all channels
  Supports INCR, INCR 4/8/16, and WRAP 4/8/16.
- Transfer mode
  - Block transfer
  - Burst transfer
- 4 bit block register and 16 bit count register are set by programming
- Supports 8, 16, and 32 bit transfer widths
- Supports increment and fixed addressing to source and destination
- Reload count, source address and destination address register
- Issues error and completion interrupts
- Displays end code of DMA transfer
- Hardware support for fixed priority and rotation priority
  In fixed priority mode, channel 0 has the highest priority, and channel 1 has the lowest priority

17.2.2. Limitations

None.

17.3. Function

17.3.1. Block diagram

Figure 18-1 shows a block diagram of the DMA controller.

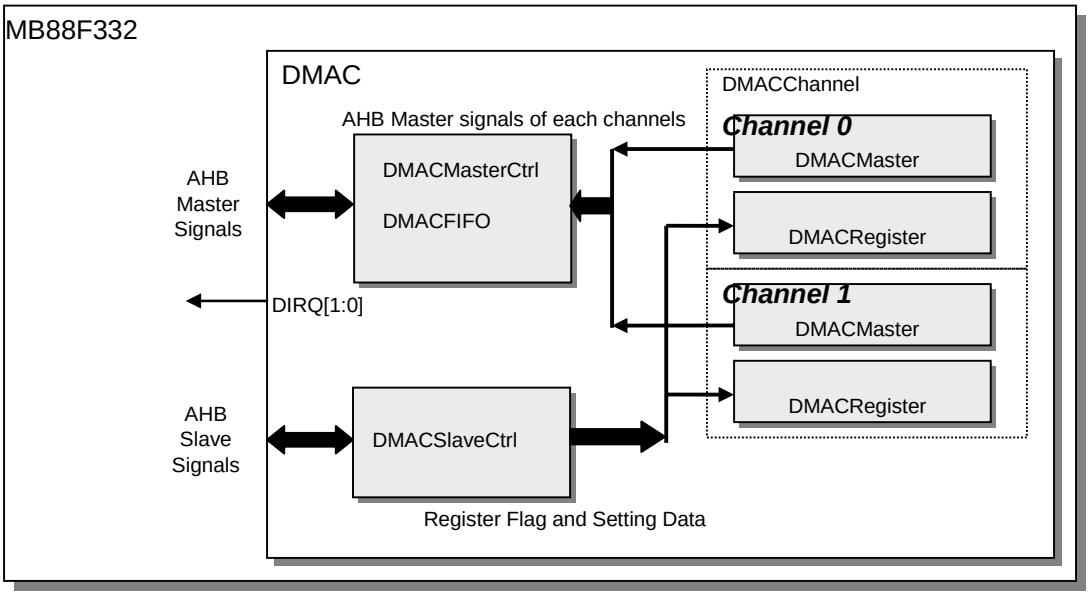


Figure 18-1 Block diagram of DMA controller

Function of each block

Table 18-1 lists the functions of each block contained in this module.

Table 18-1 Function of each block

Block	Function
DMAC	Top module
DMACMasterCtrl	Priority controller and valid channel selector for AHB Master transaction.
DMACSlaveCtrl	DMAC AHB slave I/F controller and valid channel selector for AHB Slave transaction.
DMACChannel	DMAC 1 channel module DMAC has 2 modules
DMACMaster	DMAC AHB master main controller
DMACRegister	DMAC DMA configuration register controller
DMACFIFO	DMAC 16 word FIFO

17.4. Register

This section describes DMAC register.

17.4.1. Format of Register Descriptions

The following format is used for the description of each register bit in "18.4.4.1 DMA configuration register (DMACR)" through to "18.4.4.5 DMAC destination address register (DMACDAx)".

Address	Base address + Offset															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name																
R/W																
Initial value																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name																
R/W																
Initial value																

Meaning of items and signs

Address

Address (base address + offset address) of the register

Bit

Bit number of the register

Name

Bit field name of the register

R/W

Attribution of read/write of each bit field

- R0:Read value is always "0"
- R1: Read value is always "1"
- W0: Write value is always "0", and write access of "1" is ignored
- W1: Write value is always "1", and write access of "0" is ignored
- R: Read
- W: Write

Initial value

Each bit field’s value after reset

- 0: Value is "0"
- 1: Value is "1"
- X: Value is undefined

17.4.2. Global Address

For the module base address please refer to the chapter 3 Memory map.

17.4.3. Register summary

DMAC control related registers are shown below.

Module	Address	Register	Function
DMAC common	Base + 00(h)	DMACR	DMAC configuration register
	Base + 04(h) Base + 08(h)	Reserved	
DMAC ch0	Base + 10(h)	DMACA0	DMAC0 configuration A register
	Base + 14(h)	DMACB0	DMAC0 configuration B register
	Base + 18(h)	DMACSA0	DMAC0 source address register
	Base + 1C(h)	DMACDA0	DMAC0 destination address register
DMAC ch1	Base + 20(h)	DMACA1	DMAC1 configuration A register
	Base + 24(h)	DMACB1	DMAC1 configuration B register
	Base + 28(h)	DMACSA1	DMAC1 source address register
	Base + 2C(h)	DMACDA1	DMAC1 destination address register

Notice for register settings

Note the following when setting DMAC registers:

- DMACR, DMACA, DMACB, DMACSA, and DMACDA registers are accessible in byte, half-word, and word size.
- Do not set the DMAC register address in the DMACSA and DMACDA registers.
- Do not change the registers channel settings during DMA transfer except the DE/DH bits of DMACR and the EB/PB bits of DMACA.

17.4.4. Register Description

17.4.4.1. DMA configuration register (DMACR)

Address	Base + 00(h)															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DE	DS	-	PR	DH[3:0]				(Reserved)							
R/W	R/W	R/W0	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	(Reserved)															
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit field No.	Bit field Name	Description
31	DE (DMA Enable)	This bit is used to control all DMA channel's transfer. 0: All DMA channels are disabled and don't perform DMA transfer until this bit is set to 1. If this bit is set to 0 during a DMA transfer, the channel which is in the middle of transfer stops the DMA at the transfer gap. 1: The start of the DMA transfer depends on the setting of each channel. **[Transfer gap]** The transfer gap depends on the transfer mode. • Block transfer: It gaps at BC=0 (After the transfer of BC unit is completed). • Burst transfer: There is no transfer gap. The transfer gap means that DMAC de-asserts bus request (HBUSREQ) to the Arbiter for about 4 clocks in the middle of a DMA transfer. The DE bit is useful to reconfigure the configuration registers of all channels during a DMA transfer.
30	DS (DMA Stop)	This bit indicates that the DMA transfer of all channels has been halted. 0: Disable/halt setting is cleared. (Initial value) 1: The DMA transfer of all channels are halted by disable/halt setting. This bit is set to 1 when one of the following conditions is satisfied during DMA transfer: • DMACR/DE bit is set to 0. (All channels are disabled.) • DMACR/DH bits are set to the value other than 4'h0. (All channels are halted.) This bit is useful to confirm that DMAC stops the transfer by all channels' disable/halt setting. It is no use to write 0 or 1 to this bit except for register's initialization (set 0) because this bit is for only confirming whether the DMA transfer of all channels has been halted.
29	(Reserved)	Reserved bits. Write access is ignored. Read value of this bit is always "0".
28	PR (Priority Rotation)	This bit controls DMA channels priority. 0: "Fixed" Priority order: Ch0 > Ch1 1: "Rotation" Priority order is rotated The channel is switched during the DMA transfer gap. Concerning the transfer gap, please refer to the description of the DE bit.

<table>
<tr><th colspan="2">Bit field</th><th rowspan="2">Description</th></tr>
<tr><th>No.</th><th>Name</th></tr>
<tr><td>27-24</td><td>DH[3:0]
(DMA Halt)</td><td>These bits control the halt state of all DMA channels.
If these bits are set to a value other than 4’b0000, all DMA channels are stopped and don’t perform DMA transfer until these bits are set to 4’b0000.
If these bits are set to a value other than 4’b0000 during DMA transfer, the channel which is in the middle of transfer stops DMA at the transfer gap.
Concerning the transfer gap, please refer to the description of the DE bit.

These bits are useful to halt a DMA transfer without needing to reconfigure the configuration registers of all channels.

<table><tr><td>0000</td><td>Initial value</td></tr><tr><td>Other than 0000</td><td>All channels are halted.</td></tr></table></td></tr>
<tr><td>23-0</td><td>(Reserved)</td><td>Reserved bits. Write access is ignored.
Read value of this bit is always "0".</td></tr>
</table>

17.4.4.2. DMA configuration A register (DMACAx)

Address	ch0:Base+10 (h) ch1:Base+20 (h)															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	EB0 EB1	PB0 PB1	ST0 ST1	(Reserved)					BT0[3:0] BT1[3:0]				BC0[3:0] BC1[3:0]			
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TC0[15:0] TC1[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit field		Description
No.	Name	
31	ch0:EB0 ch1:EB1 (Enable Bit)	This bit is used to control its DMA channel's transfer. If this bit is set to 1, the channel waits for the trigger to start a DMA transfer. (prior to this, DMACR/DE bit needs to be set to 1 already.) DMAC sets this bit to 0 after a DMA transfer is completed. If this bit is set to 0, this channel is disabled and does not perform a DMA transfer until this bit is set to 1. When this bit is set to 0 during a DMA transfer, the DMA stops at the transfer gap. And that is regarded as the enforced stop. About the transfer gap, please refer to the description of DMACR/DE bit. This bit is useful to reconfigure each configuration registers of its channel during DMA transfer. 0: This channel is disabled (initial value) 1: This channel is enabled
30	ch0:PB0 ch1:PB1 (Pause Bit)	This bit is used to control a pause of its DMA channel's transfer. If this bit is set to 1, this channel stops the transfer, and does not perform a DMA transfer until this bit is cleared. If this bit is set to 1 during a DMA transfer, the DMA stops at the transfer gap. About the transfer gap, please refer to the description of DMACR/DE bit. If this bit is set to 1 before receiving the transfer request and getting the bus, DMAC enters the pause state immediately. In this case, DMAC does not hold its transfer request during a pause. If this bit is set to 0 during the pause of DMA transfer, the pause condition is cleared and DMAC waits for a new transfer request. This bit is useful to halt a DMA transfer without reconfiguration of each configuration registers of its channel. 0: Initial value 1: This channel is halted
29	ch0:ST0 ch1:ST1 (Software Trigger)	This bit is used to generate a software trigger. If this bit is set to 1, DMA transfer is started because a software request has been received. DMAC sets this bit to 0 after DMA transfer by this bit has been completed. If this bit is set to 0 during a DMA transfer by this software request, the DMA transfer is stopped at the transfer gap. 0: Initial value 1: Software request
28-24	(Reserved)	Reserved bits

<table>
<tr><th colspan="2">Bit field</th><th rowspan="2">Description</th></tr>
<tr><th>No.</th><th>Name</th></tr>
<tr><td>23-20</td><td>ch0:BT0[3:0]
ch1:BT1[3:0]
(Beat Type)</td><td>These bits are used to select beat transfer on the AHB.

If these bits are set to NORMAL or SINGLE, single source access and single destination access are performed alternately.
If these bits are set to INCR* or WRAP*, continuous source access and continuous destination access are performed alternately.
DMAC has a 64-Byte FIFO which is shared among all channels. This FIFO is used for INCR* or WRAP* DMA transfer.
About INCR*, and WRAP*, please refer to AMBA specification (v2.0).
When INCR (undefined length burst) is set, burst length is specified at BC bits.

<table>
<tr><th>BT[3:0]</th><th>Function</th></tr>
<tr><td>0(h)</td><td>Normal (same as Single) (Initial value)</td></tr>
<tr><td>1(h)-7(h)</td><td>Invalid</td></tr>
<tr><td>8(h)</td><td>Single (same as Normal)</td></tr>
<tr><td>9(h)</td><td>INCR</td></tr>
<tr><td>A(h)</td><td>WRAP4</td></tr>
<tr><td>B(h)</td><td>INCR4</td></tr>
<tr><td>C(h)</td><td>WRAP8</td></tr>
<tr><td>D(h)</td><td>INCR8</td></tr>
<tr><td>E(h)</td><td>WRAP16</td></tr>
<tr><td>F(h)</td><td>INCR16</td></tr>
</table>

Fixed length burst (INCR*, WRAP*) and Undefined length burst (INCR) are valid when DMACB/MS is set to Block transfer or Burst transfer.</td></tr>
<tr><td>19-16</td><td>ch0:BC0[3:0]
ch1:BC1[3:0]
(Block Count)</td><td>These bits are used to specify the block count for the block/burst transfer.
The max block count is 16(0xF).
These bits are valid when Beat Typer (BT) is NORMAL or SINGLE or INCR. When the other beat type (fixed length burst and wrap) is set, these bits are ignored.
These bits can be read during DMA transfer. Usually when single source access and single destination access is completed successfully, BC bits are decremented by 1.

[Note]
These bits can be set when the Beat Type bit (BT) is INCR. But in the INCR DMA transfer, the read data of BC is always 4’h0 after DMA transfer starts.
Therefore monitoring of BC during INCR DMA transfer is no use.

DMAC sets these bits to 4’b0000 after DMA transfer is completed normally.

<table>
<tr><th>BC[3:0]</th><th>Function</th></tr>
<tr><td>x(h)</td><td>Block count (initial value: 4’b0000)</td></tr>
</table></td></tr>
<tr><td>15-0</td><td>ch0:TC0[15:0]
ch0:TC1[15:0]
(Transfer Count)</td><td>These bits are used to specify the transfer count for the block/burst transfer.
The max transfer count is 65536(0xFFFF).
These bits are valid when BT is set to all beat type.

These bits can be read during a DMA transfer. Usually when a DMA transfer of BC=0 is completed successfully, TC bits are decremented by 1 in the NORMAL mode or SINGLE mode (BT = NORMAL or SINGLE).
In the other Beat Transfer mode (INCR, INCR*, WRAP*), TC bits are decremented by 1 after once sequential source / destination accesses are completed. (e.g. in the case of INCR4, TC bits are decremented by 1 after sequential 4 source accesses and sequential 4 destination accesses are completed.)

DMAC sets these bits to 16’h0000 after a DMA transfer is completed normally.

<table>
<tr><th>TC[3:0]</th><th>Function</th></tr>
<tr><td>x(h)</td><td>Number of transfer (initial value: 16’h0000)</td></tr>
</table></td></tr>
</table>

17.4.4.3. DMA configuration B register (DMACBx)

Address	ch0:Base+14 (h) ch1:Base+24 (h)															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TT0[1:0] TT1[1:0]		MS0[1:0] MS1[1:0]		TW0[1:0] TW1[1:0]		FS0 FS1	FD0 FD1	RC0 RC1	RS0 RS1	RD0 RD1	EI0 EI1	CI0 CI1	SS0[2:0] SS1[2:0]		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W0	R/W0	R/W0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	(Reserved)				(Reserved)				(Reserved)							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit field		Description
No.	Name	
31-30	ch0:TT0[1:0] ch1:TT1[1:0] (Transfer Type)	These bits are used to specify the transfer type. Now DMAC supports 2 cycle transfer mode only. **TT[1:0]** / **Function** 0(h) / 2 cycle transfer (initial value) Other than 0(h) / Reserved
29-28	ch0:MS0[1:0] ch1:MS1[1:0] (Mode Select)	These bits are used to select the transfer mode. **MS[1:0]** / **Function** 0(h) / Block transfer mode (initial value) 1(h) / Burst transfer mode 2(h) / Reserved 3(h) / Reserved
27-26	ch0:TW0[1:0] ch1:TW1[1:0] (Transfer Width)	These bits are used to specify the transfer data width. HSIZE of DMAC issues this value on AHB. **TW[1:0]** / **Function** 0(h) / Byte (initial value) 1(h) / Half-word 2(h) / Word 3(h) / Reserved
25	ch0:FS0 ch1:FS1 (Fixed Source)	This bit is used to determine the source address. If the source address needs to increase after each transfer, this bit must be set to 0. **FS** / **Function** 0(h) / Source address is incremented (initial value) 1(h) / Source address is fixed
24	ch0:FD0 ch1:FD1 (Fixed Destination)	This bit is used to determine the destination address. If the destination address needs to increase after each transfer, this bit must be set to 0. **FD** / **Function** 0(h) / Destination address is incremented (initial value) 1(h) / The destination address is fixed
23	ch0:RC0 ch1:RC1 (Reload Count)	This bit is used to control the reload function of the block count (DMACA/BC bits) and transfer count (DMACA/TC bits). When this bit is set to 1, DMACA/BC and DMACA/TC is set to the initial setting value after a DMA transfer is completed. **RC** / **Function** 0(h) / Reload function for number of transfer is disabled (initial value) 1(h) / Reload function for number of transfer is enabled

<table>
<tr><th colspan="2">Bit field</th><th rowspan="2">Description</th></tr>
<tr><th>No.</th><th>Name</th></tr>
<tr><td>22</td><td>ch0:RS0
ch1:RS1
(Reload Source)</td><td>This bit is used to control the reload function of the source address (DMACSA).
"1" is set to this bit: DMACSA is set to the initial setting value after a DMA transfer is completed.
"0" is set to this bit: DMAC sets the next source address to DMACSA after a DMA transfer is completed.

<table><tr><th>RS</th><th>Function</th></tr><tr><td>0(h)</td><td>The reload function of the source address is disabled (initial value)</td></tr><tr><td>1(h)</td><td>The reload function of the source address is enabled</td></tr></table></td></tr>
<tr><td>21</td><td>ch0:RD0
ch1:RD1
(Reload Destination)</td><td>This bit is used to control the reload function of the destination address (DMACDA).
"1" is set to this bit: DMACDA is set to the initial setting value after a DMA transfer is completed.
"0" is set to this bit: DMAC sets the next destination address to DMACDA after a DMA transfer is completed.

<table><tr><th>RD</th><th>Function</th></tr><tr><td>0(h)</td><td>The reload function of the destination address is disabled (initial value)</td></tr><tr><td>1(h)</td><td>The reload function of the destination address is enabled</td></tr></table></td></tr>
<tr><td>20</td><td>ch0:EI0
ch1:EI1
(Error Interrupt)</td><td>This bit is used to control the issue of an interrupt (DIRQ) caused by an Error.
If this bit is set to 1, an Error interrupt is issued by the following transfer error.
• Address overflow
• Transfer stop request by DSTP, IDSTP, or disable the transfer with EB or DE bit
• Source access error
• Destination access error

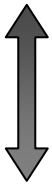
<table><tr><th>EI</th><th>Function</th></tr><tr><td>0(h)</td><td>Error interrupt issue is disabled (initial value)</td></tr><tr><td>1(h)</td><td>Error interrupt issue is enabled</td></tr></table></td></tr>
<tr><td>19</td><td>ch0:CI0
ch1:CI1
(Completion Interrupt)</td><td>This bit is used to control the issue of an interrupt (DIRQ) caused by a transfer completion.
If this bit is set to 1, a completion interrupt is issued after the DMA transfer has been normally completed.

<table><tr><th>CI</th><th>Function</th></tr><tr><td>0(h)</td><td>Completion interrupt issue is disabled (initial value)</td></tr><tr><td>1(h)</td><td>Completion interrupt issue is enabled</td></tr></table></td></tr>
</table>

<table>
<tr><th colspan="2">Bit field</th><th rowspan="2">Description</th></tr>
<tr><th>No.</th><th>Name</th></tr>
<tr><td>18-16</td><td>ch0:SS0[2:0]
ch1:SS1[2:0]
(Stop Status)</td><td>These bits are used to indicate the end code of DMA transfer.
The end code is shown as below.
These bits are also used to clear the interrupt (DIRQ).
If an interrupt has been issued with error or normal end, writing 3’b000 to these bits clears this interrupt.

<table>
<tr><th>SS</th><th>Function</th><th>Status type</th></tr>
<tr><td>0(h)</td><td>Initial value</td><td>None</td></tr>
<tr><td>1(h)</td><td>Address overflow</td><td>Error</td></tr>
<tr><td>2(h)</td><td>Transfer stop request</td><td>Error</td></tr>
<tr><td>3(h)</td><td>Source access error</td><td>Error</td></tr>
<tr><td>4(h)</td><td>Destination access error</td><td>Error</td></tr>
<tr><td>5(h)</td><td>Normal end</td><td>End</td></tr>
<tr><td>6(h)</td><td>Reserved</td><td></td></tr>
<tr><td>7(h)</td><td>DMA pause</td><td>None</td></tr>
</table>

If various errors occur at same time, the end code is displayed according to the following priority.

High priority

Reset
Clear by writing 3'b000
Address overflow
Stop request
Source access error
Destination access error
Low priority</td></tr>
<tr><td>15-12</td><td>(Reserved)</td><td>Reserved bits</td></tr>
<tr><td>11-8</td><td>(Reserved)</td><td>Reserved bits</td></tr>
<tr><td>7-0</td><td>(Reserved)</td><td>Reserved bits.
Write access is ignored. Read value of this bit is always "0".</td></tr>
</table>

17.4.4.4. DMAC source address register (DMACSAx)

Address	ch0:Base+18 (h) ch1:Base+28 (h)															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DMACSA0[31:16] DMACSA1[31:16]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DMACSA0[15:0] DMACSA1[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit field		Description
No.	Name	
31-0	ch0:DMACSA0[31:0] ch1:DMACSA1[31:0] (DMAC Source Address)	These bits are used to specify the source address to start a DMA transfer. These bits can be read during a DMA transfer. When the fixed address function (DMACB/FS) is disabled, these bits are incremented according to the transfer width (DMACB/TB) after a source access is performed successfully. When a DMA transfer is completed, DMAC sets the next source address to these bits. [Note] Do not set DMAC register address to DMACSA.

DMACSA	Function
x(h)	Source address to start DMA transfer (initial value: 32'h00000000)

17.4.4.5. DMAC destination address register (DMACDAx)

Address	ch0:Base+1C (h) ch1:Base+2C (h)															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DMACDA0[31:16] DMACDA1[31:16]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DMACDA0[15:0] DMACDA1[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit field		Description
No.	Name	
31-0	ch0:DMACDA0[31:0] ch1:DMACDA1[31:0] (DMAC Destination Address)	These bits are used to specify the destination address to start a DMA transfer. These bits can be read during a DMA transfer. If the fixed address function (DMACB/FD) is disabled, these bits are incremented according to the transfer width (DMACB/TB) after a destination access has been performed successfully. When a DMA transfer is completed, DMAC sets the next destination address to these bits. [Note] Do not set DMAC register address to DMACDA.

DMACDA	Function
x(h)	Destination address to start DMA transfer (initial value: 32'h00000000)

17.5. Operation

This section describes the operation of DMAC.

17.5.1. Transfer mode

DMAC has 3 different transfer modes. The transfer mode is determined using DMACB/MS[1:0].

17.5.1.1. Block transfer

17.5.1.1.1. Operation

In the block transfer mode, the DMA transfer that is specified to the block count (DMACA/BC) is performed with one transfer request. When the transfer count (DMACA/TC) is set to a value other than 0, TC is decremented by 1 after a DMA transfer of BC has been completed. So the DMA transfer is finished after the last transfer (BC is 4’h0 and TC is 16’h0000) has been performed.

17.5.1.1.2. Transfer gap

In block transfer mode, DMAC negates the bus request to the arbiter once after the transfer of BC has been completed. This operation thereofore prevents DMAC from taking over the bus.
The transfer gap is useful to reflect the register setting (e.g. disable/pause setting) to DMAC during DMA transfer.

17.5.1.1.3. Transfer request

Set "1" to DMACA/ST and set 5'b00000 to DMACA/IS

17.5.1.1.4. Timing chart

Figure 18-2 shows the timing chart of a Block transfer.

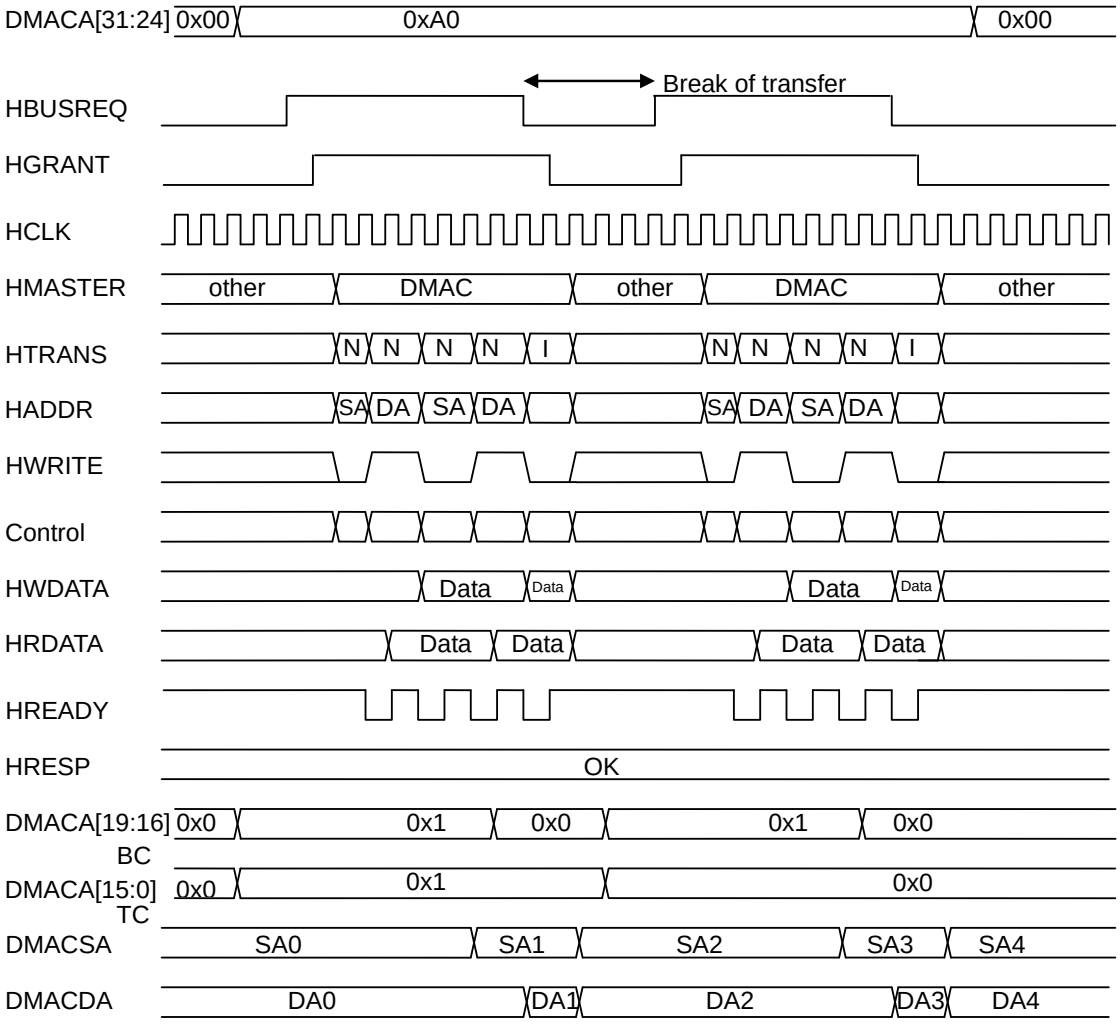


Figure 18-2 A Block transfer (for BC=0x1, TC=0x1)

17.5.1.2. Burst transfer

17.5.1.2.1. Operation

In the burst transfer mode, the DMA transfer that multiplies the block count by the transfer count (DMACA/BC x DMACA/TC) is performed with one transfer request. When the transfer count (DMACA/TC) is set to a value other than 0, TC is decremented by 1 after DMA transfer of BC has been completed. So the DMA transfer is finished after the last transfer (BC is 4’h0 and TC is 16’h0000) has been performed.

17.5.1.2.2. Transfer gap

In burst transfer mode, DMAC negates the bus request to the arbiter after the DMA transfer has been completed. Therefore the transfer gap is not generated in the burst transfer. The change of the register setting (e.g. disable/pause setting) during a DMA transfer is reflected after the DMA transfer has been completed.

17.5.1.2.3. Transfer request

Set "1" to DMACA/ST and set 5'b00000 to DMACA/IS

17.5.1.2.4. Timing chart

Figure 18-3 shows the timing chart of the Burst transfer.

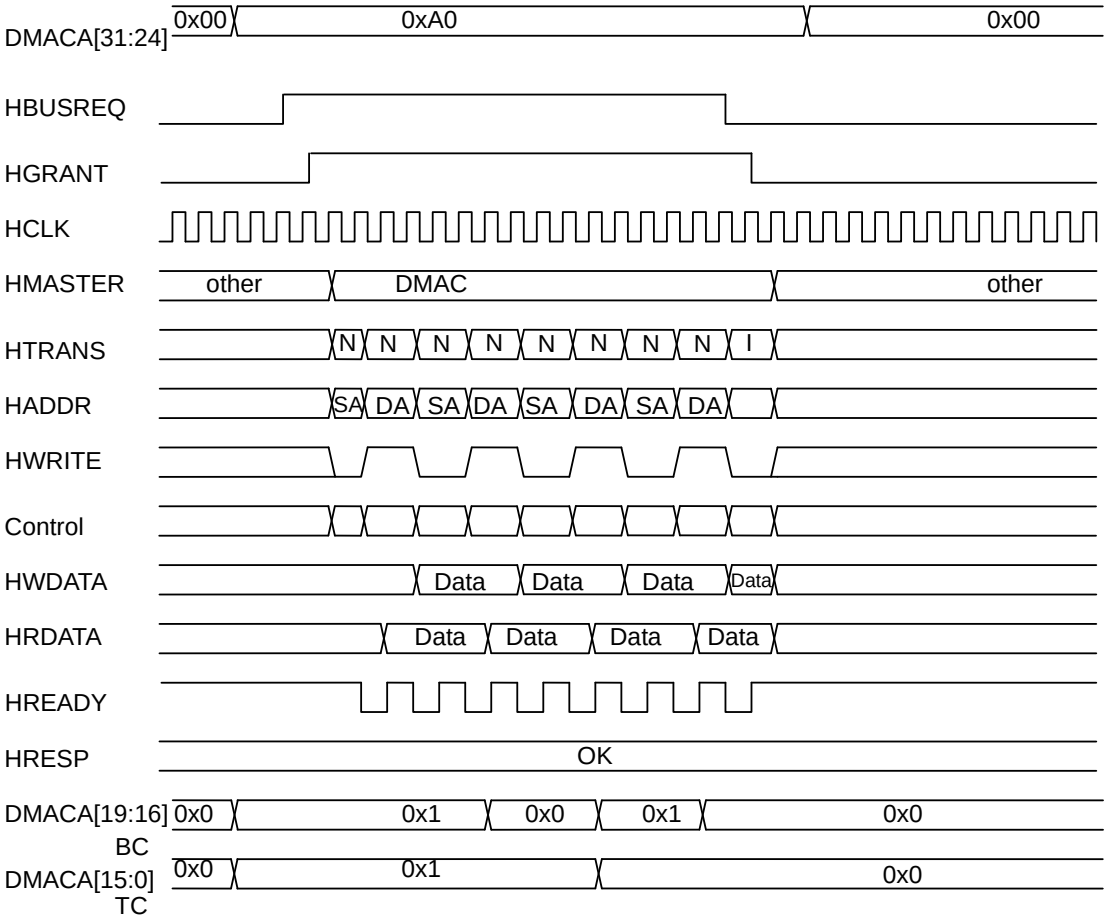


Figure 18-3 The Burst transfer (In case of BC=0x1, TC=0x1)

17.5.2. Beat transfer

DMAC supports beat transfers. The term 'beat transfer' means the incrementing/wrapping burst described in the AMBA specification.

DMAC has a 64-byte FIFO which is shared by all channels, so it is possible to perform sequential source access and destination access.

The type of the beat transfer is set to DMACA/BT bits.

The relationship between DMACA/BT and HBURST of AHB is shown as below.

Table 18-2 DMACA/BT and HBURST

DMACA/BT	Beat transfer type	HBURST	DMACA/MS (mode select)	
			Block	Burst
4'b0000	Normal	Single	OK	OK
4'b1000	Single	Single	OK	OK
4'b1001	INCR	INCR	OK	OK
4'b1010	WRAP4	WRAP4	OK	OK
4'b1011	INCR4	INCR4	OK	OK
4'b1100	WRAP8	WRAP8	OK	OK
4'b1101	INCR8	INCR8	OK	OK
4'b1110	WRAP16	WRAP16	OK	OK
4'b1111	INCR16	INCR16	OK	OK

17.5.2.1. Normal and Single transfer

NORMAL transfer and SINGLE transfer are the same transfer method. Single source access and single destination access are performed alternately like Figure 18-2.

17.5.2.2. INCREMENTING and Wrapping transfer

When incrementing beat transfer (INCR, INCR4, INCR8 and INCR16) or wrapping beat transfer (WRAP4,WRAP8,WRAP16) is set to DMACA/BT, sequential source access and destination access are performed by using the 64-byte DMAC FIFO.

In case of INCR4 (DMACA/BT= 4'b1011), DMAC performs 4 sequential source accesses. The data from the source is stored in the FIFO of DMAC and then the data is driven to the destination sequentially.

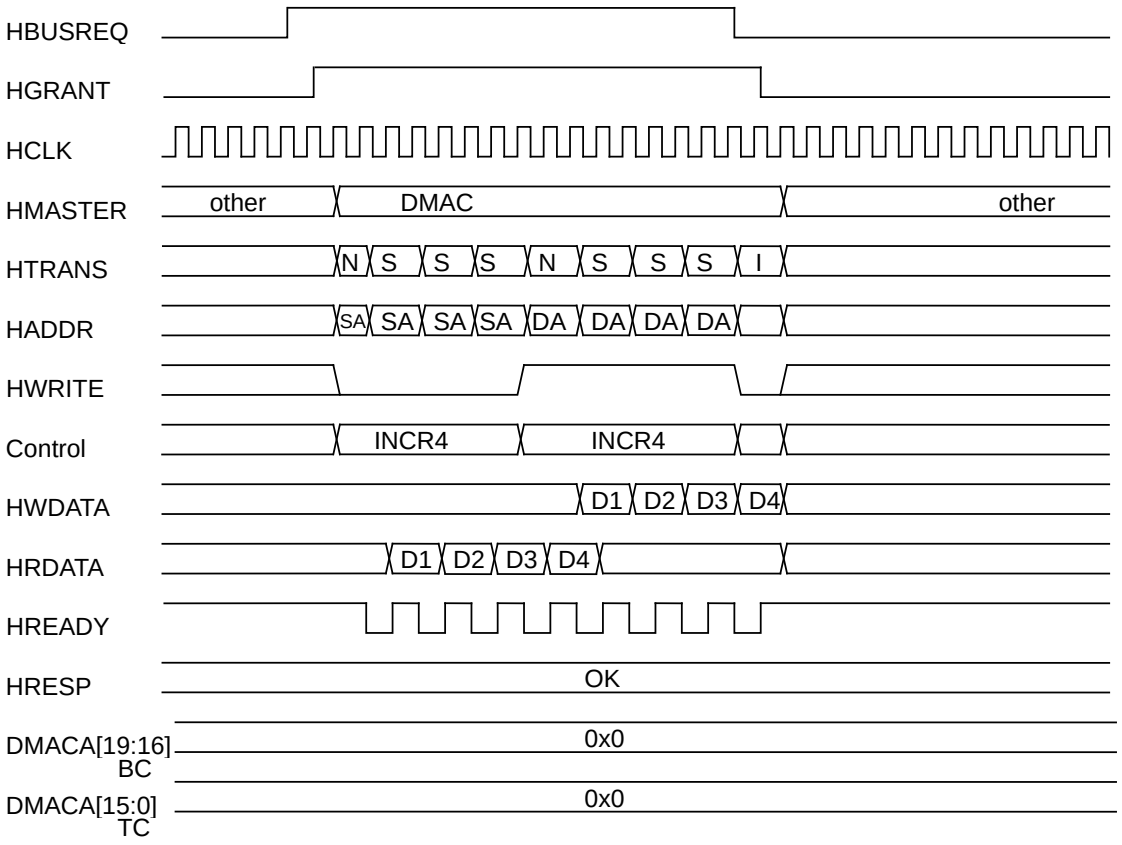


Figure 18-4 Incrementing/Wrapping beat transfer (e.g. INCR4 block transfer)

Note: When Incrementing beat transfer is executed three times, the transition of the address is as follows.

ex) DMACA/BT= 4'b1011 , DMACA/TC= 16'h0002 ,
DMACB/TW = 2'b10 , DMACB/FS = 1'b0 , DMACB/FD = 1'b0 ,
DMACSA = 32'h00000004 , DMACSA = 32'h10000004
source address
00000004 -> 00000008 -> 0000000C -> 00000010 ->
00000014 -> 00000018 -> 0000001C -> 00000020 ->
00000024 -> 00000028 -> 0000002C -> 00000030
destination address
10000004 -> 10000008 -> 1000000C -> 10000010 ->
10000014 -> 10000018 -> 1000001C -> 10000020 ->
10000024 -> 10000028 -> 1000002C -> 10000030

Note: When Wrapping beat transfer is executed three times, the transition of the address is as follows.

ex) DMACA/BT= 4'b1010 , DMACA/TC= 16'h0002 ,
DMACB/TW = 2'b10 , DMACB/FS = 1'b0 , DMACB/FD = 1'b0 ,
DMACSA = 32'h00000004 , DMACSA = 32'h10000004
source address
00000004 -> 00000008 -> 0000000C -> 00000000 ->
00000004 -> 00000008 -> 0000000C -> 00000000 ->
00000004 -> 00000008 -> 0000000C -> 00000000
destination address
10000004 -> 10000008 -> 1000000C -> 10000000 ->
10000004 -> 10000008 -> 1000000C -> 10000000 ->
10000004 -> 10000008 -> 1000000C -> 10000000

17.5.3. Channel priority control

DMAC can control each channel's priority with the DMACR/PR bit.

17.5.3.1. Fixed priority

When Fixed Priority is set to DMACR/PR bit, the priority order is fixed, and the lowest number channel can be granted the bus. The priority controller of DMAC switches the channel during the active channel's transfer gap.
Therefore, if all channels are active at the same time, the lowest numbered channel (channel #0) is selected by the priority controller and it can start the transfer.
Also, when the active channel (channel #0) loses the bus temporarily because of its transfer gap, the second lowest numbered channel (channel #1) is granted the bus. When the second lowest number channel (#1) loses the bus because of the transfer gap, the lowest numbered channel (#0) gets the bus again.
Thus the lowest numbered channel and the second lowest numbered channel can get the bus preferentially in fixed priority mode.

Figure 18-5 shows the valid channel in the fixed priority mode.

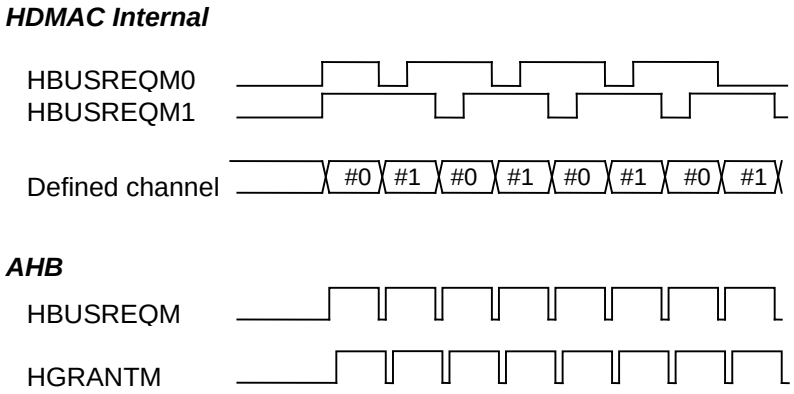


Figure 18-5 valid channel in the fixed priority

17.5.3.2. Rotate priority

When Rotated Priority is set in the DMACR/PR bit, the priority is rotated. At first, the lowest numbered channel can be granted the bus. The priority controller of DMAC switches the channel at the active channel's transfer gap.
Therefore, if all channels are active at same time, the lowest numbered channel (channel #0) is selected by the priority controller and can start the transfer.
Also, when the active channel (channel #0) loses the bus temporarily because of its transfer gap, the second lowest numbered channel (channel #1) is granted the bus. When the second lowest numbered channel (#1) loses the bus because of the transfer gap, the third lowest numbered channel (#2) gets the bus.
Thus all channels can get the bus by rotation in the rotated priority mode.

Figure 18-6 shows the valid channel in the rotated priority mode.

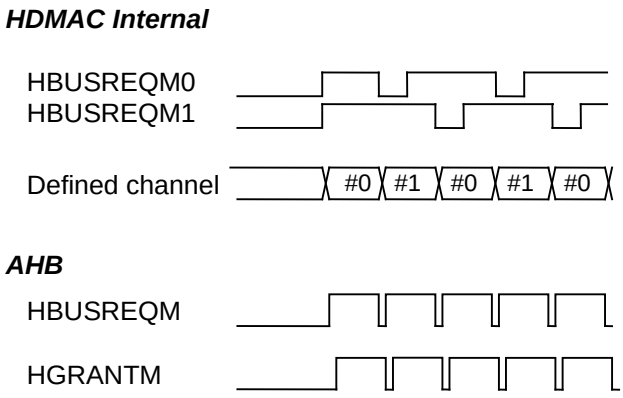


Figure 18-6 valid channel in the rotated priority

17.5.4. Error

DMAC supports error responses from the AHB slave.

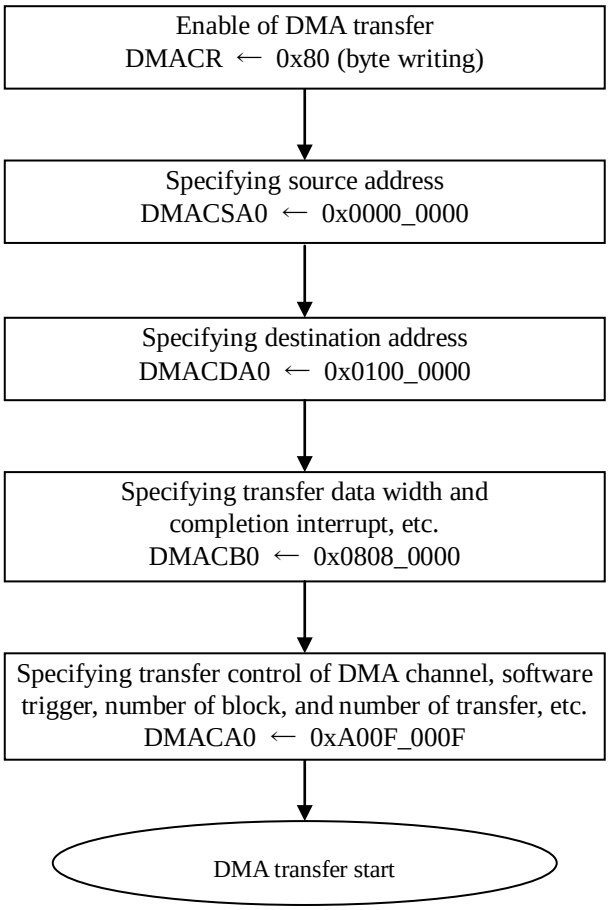
17.5.4.1. Error

When DMAC receives an ERROR response from an AHB slave during DMA transfer, DMAC negates the bus request and stops the transfer soon even if the DMA transfer is not completed.
In this case, the block/transfer count register and source/destination address register are not updated.

17.6. Example of DMAC setting

17.6.1. DMA start in Single channel

Example of block transfer by software request



For a software request, the DMACA register should be configured at the end. Moreover, DMACR register should be set by byte writing.

17.6.2. DMA start in all channels

All channels are able to start simultaneously by setting the DMACR register after setting the registers of all channels. In this case, the priority controller of DMAC receives a request of all channels at the same time, then transfer starts by selecting the channel related to the DMACR/PR setting.

18. Clock and Reset Generator

This chapter describes the Clock and Reset Generator of the MB88F333.

18.1. Outline

The Clock and Reset Generator (CRG) of MB88F333 has a Clock Synthesizer part (ClkSynth) and a Clock and reset control part.
The ClkSynth module generates the BITCLOCK for the display and 83MHz clock for the system.

The Clock Synthesizer module consists of a "N.P calc" module and a "Npclk Divider" logic block module. In the "Npclk Divider" module the received input clock e.g. from PLL is processed and two clocks are generated.
The setup for configurable BITCLOCK parameters is done in the "N.P calc" module.

The Clock and Reset Control module is responsible for clock generation and the control of the GDCs internal modules. The clock for the AHB, APB and local peripheral busses is generated from the clock from the CLOMO module. The display clock is generated with a clock from the Clock Synth module. In addition, the reset signal for each module is generated based on the power-on reset signal.

18.2. Features

18.2.1. Features

The ClkSynth module has the following features.

- input frequency: 250 MHz
- output frequency: 1 MHz to 90 MHz
- N divider of range 2…255 results in 253 nominal frequencies which can be selected
- P divider of range 0…4 results in 5 different frequencies to be selected between each nominal frequency
- subP divider of range 0…255 to specify mean frequency between P values
- “Frequency hopping” of 4 different frequencies (in range of delta P and delta subP) for Spread Spectrum Clocking; with delta P and delta subP a subset of 3 additional frequencies can be setup;
- period of frequency hopping is configurable

The clock control module has the following features.

- Determination of the dividing frequency for the APB and the local peripheral bus clocks.
- The modulated clock can be output to the AHB and APB bus clocks.
- Reset for an internal module is generated from power-on reset.
- After PLL clock is steady, reset of an internal module is released.The stability period of PLL is 1ms after releasing power-on reset.
- Control of the ON/OFF modes of the clock and GDC internal module resets.

18.2.2. Limitations

- Please set up the CLOMO module separately if you intend to use a modulated clock.
- The clock and reset of some modules cannot be controlled.

18.3. Function

18.3.1. Block diagrams

18.3.1.1. ClkSynth

The Clock Synthesizer module consists of “N.P calc” module and logic block of “Npclk Divider” module.

In “Npclk Divider” module the received input clock e.g. from PLL is processed and two clocks are generated.
The setup for configurable BITCLOCK parameters is done in “N.P calc” module.

Following figure shows the composition of the two modules.

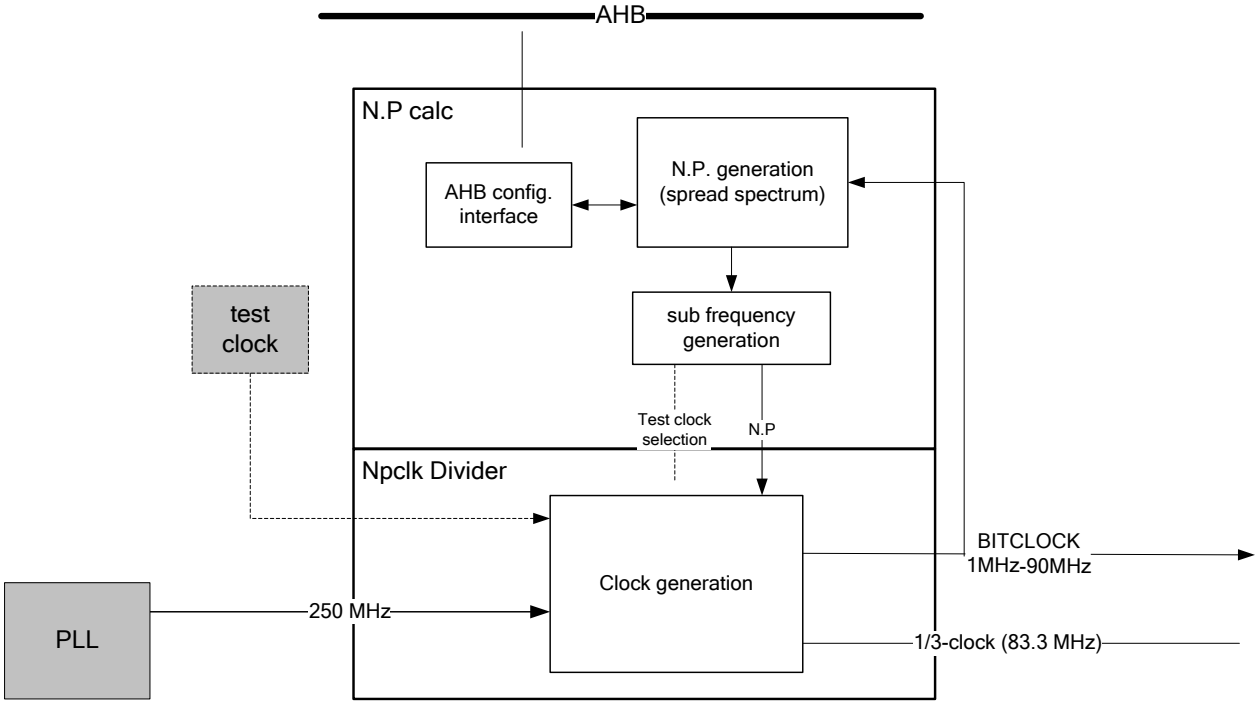


Figure 19-1 Block diagram of ClkSynth module

18.3.2. ClkSynth

18.3.2.1. N.P calculation (N.P calc)

18.3.2.1.1. Sub-frequency generation

To specify frequencies between a sub-frequency generator switches between two P values in an configurable ratio given by value subP. E.g. if subP is set to 1 then in a period of 256 clock cycles P is 255 times set to P-value P0 and 1 time to P-value P0+1.
That way the resulting „mean“ frequency can be specified more precise by the 8-bit subP value.

18.3.2.1.2. Processing Algorithm

To obtain N,P,subP:

T = 250MHz / BITCLOCK_frequency

N = int(T)

P = int(5* (T-N))

subP = int(256* (5*(T-N) – P) +0.5)

Use of “subP” <>0 or of spread spectrum causes jitter in Pixclk. Therefore if spread spectrum is not required, it is recommended to keep subP = 0.

The following frequencies can be generated with subP=0:

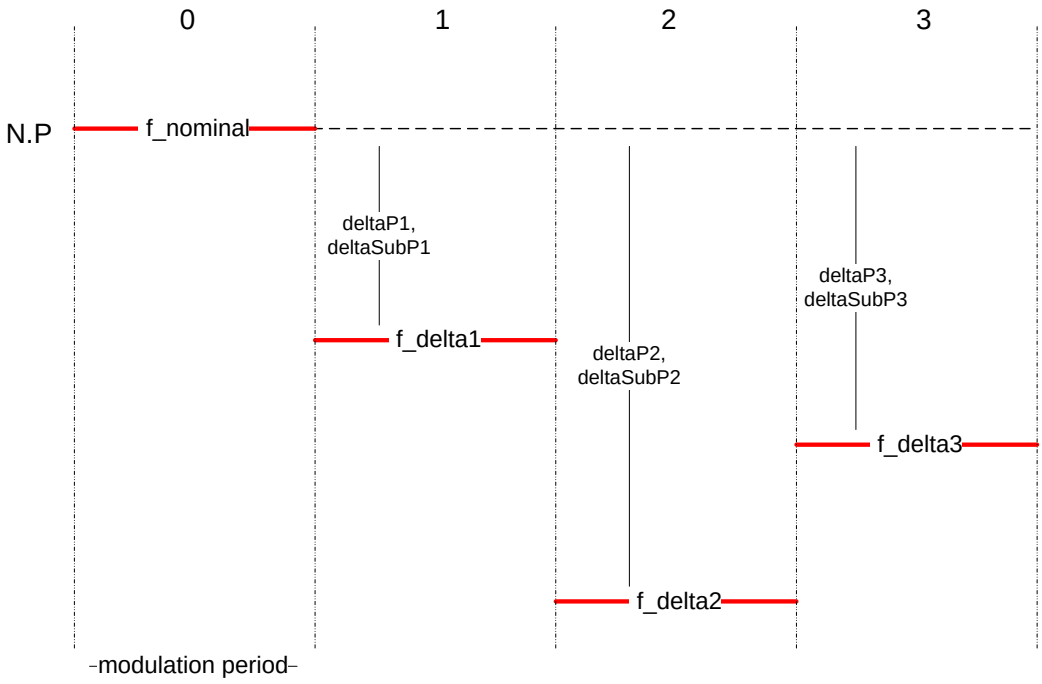
N.P	BITCLOCK_frequency
2.0	125 MHz (max. theoretical)
2.1	113.64 (theor.)
2.2	104.17 (theor.)
2.3	96.15
2.4	89.29
3.0	83.33
3.1	78.13
3.2	73.53
3.3	69.44
3.4	65.79
4.0	62.5
4.1	59.52
4.2	56.82
4.3	54.35
4.4	52.08
5.0	50
Etc	

Any frequency up to 90MHz can be generated to with a tolerance of 3.6% in CS86 technology.

18.3.2.2. Spread Spectrum Clocking

To spread the frequency spectrum for EMI aspects three additional delta values for P and subP can be set up. The delta values are added automatically to the nominal offset value of N.P and subP.
The modulation period between the different frequencies is configurable.

The following figure shows an example of the four different frequencies. If delta values are all set to '0' then spread spectrum clocking is deactivated.



A possible overflow is accounted for in addition and the N and P values will be incremented in this case.

18.3.3. Clock and Reset Control

18.3.3.1. Clock Generation

This module outputs the following clocks to internal modules:

- AHB bus clock
- APB bus clock
- Local Peripheral bus clock
- Display (Pixel) clock

The list shown below lists clock connections to internal modules.

Table 19-1 Clock connection list

Clock	Module	Port name	Control	AHB Bus
AHB bus clock	CMD Sequencer	o_HCLK_CMDS	i_CKEN[0]	AHB1
	Sprite GDC	o_HCLK_SGDC	i_CKEN[1]	AHB1
	Display Controller TCON,CLUT,DITH,SIG	o_HCLK_DISP	i_CKEN[2]	AHB1
	DMAC	o_HCLK_DMAC	i_CKEN[4]	AHB1
	RLD	o_HCLK_RLD	i_CKEN[5]	AHB1
	Serial Flash I/F	o_HCLK_SFIF	i_CKEN[6]	AHB1
	Flash MEM I/F	o_HCLK_FMIF	i_CKEN[7]	AHB1
	INTMEM 8K	o_HCLK_MIF8K	i_CKEN[8]	AHB1
	INTMEM 64K0	o_HCLK_MIF64K0	i_CKEN[9]	AHB1
	INTMEM 64K1	o_HCLK_MIF64K1	i_CKEN[10]	AHB1
	ConfigFIFO	o_HCLK_CNFG	i_CKEN[11]	AHB2
	AHB Bus Matrix1,2 & AHB2AHB & CCNT & Host I/F & APIX I/F & R-Handler	o_HCLK_BM	none	AHB2
	TIC	o_HCLK_TIC	i_TIC_MODE	AHB2
	Clk Synth	o_SSCLK	none	-
Modulated clock	CLOMO	o_HCLK_CLM	none	-
local peripheral bus clock	RESOURCE	o_RCLK	i_CKEN[12]	AHB2
	CLOMO	o_RCLK_CLM	none	AHB2
APB bus clock	GPIO & ExtINT	o_PCLK	i_CKEN[13]	AHB2
Display clock	Sprite GDC	o_DCLK_SGDC	i_CKEN[1]	AHB1
	Display Controller TCON,CLUT,DITH,SIG	o_DCLK_DISP	i_CKEN[2]	AHB1

18.3.3.2. AHB bus clock generation

The clock signal for modules connected to the AHB bus is generated from the 83 MHz clock and the modulated clock. Please set up the CLOMO module if you intend to use the modulated clock.
The setting of the ratio of dividing frequency and the control of the clock can be set to Register of CCNT.

18.3.3.2.1. Clock division

The frequency division ratio can be set via the setting of the dividing frequency (i_HCKDIV [1:0]) for the AHB bus.
An example of a clock division frequency for the system clock = 83.33 MHz from ClkSynth is shown below.

i_HCKDIV	Ratio of dividing frequency	AHB bus clock
00 (Initial value)	1	83.33MHz
01	2	41.66MHz
10	4	20.83MHz
11	1	83.33MHz

18.3.3.2.2. Clock output control

The output clock of each module is controlled according to the clock control signal (i_CLKEN).
Please confirm the control signal of each module using Table 19-1. If the clock control signal corresponding to each clock output is "H", then the clock is output. The clock is stopped if the control signal is “L”. If the clock has been stopped, "L" is output.

i_CLKEN[n]=”H”	The clock is output.
i_CLKEN[n]=”L”	The clock is stopped.

(n shows bit number.)

Note : The initial state of the clock control to the Embedded Flash Memory memory interface module and the CMD Sequencer module is 'clock output enabled'. The initial state of other modules is 'clock output disabled'.

18.3.3.3. APB bus clock generation

The APB bus clock is generated from the AHB bus clock. The dividing frequency ratio setting and the control of the clock can be set in the CCNT register.
If the modulated clock is selected by the AHB bus clock, the modulated clock is output to the APB bus clock.

18.3.3.3.1. Clock division

The ratio of the dividing frequency can be set by the setting of dividing frequency (i_PCKDIV [1:0]) for the APB bus.
An example of setting the clock dividing frequency when the clock for the AHB bus is 83.33MHz is shown below.

i_PCKDIV	Ratio of dividing frequency	APB bus clock
00 (Initial value)	2	41.66MHz
01	4	20.83MHz
10	8	10.42MHz
11	1	83.33MHz

Note：i_PCKDIV="11" is a test setting.
Modules connected to the APB bus cannot be operated at 83.33MHz. Please do not set i_PCKDIV =”11" for normal use.

18.3.3.3.2. Clock control

The clock output to the APB bus module is controlled by the clock control signal (i_CLKEN[13]).
Please confirm the control signal of each module using Table 19-1. If the clock control signal corresponding to a clock output is "H", the clock is output. The clock stops when the control signal is “L”. When the clock has been stopped, "L" is output.
The clock is stopped by default.

i_CLKEN[13]=”H”	Clock is output
i_CLKEN[13]=”L”	Clock is stopped

18.3.3.4. Peripheral bus clock

An 83MHz clock input from the Clk Synth module generates the clock to modules connected to the local peripheral bus.
The setting of the ratio of dividing frequency and the control of the clock can be set to Register of CCNT.
The modulated clock is not output.

18.3.3.4.1. Clock dividing frequency

The ratio of the dividing frequency can be set by the setting the dividing frequency (i_RCKDIV[1:0]).
Examples of the setting the clock dividing frequency when the clock for the AHB bus is 83.33MHz are shown below.

i_RCLKDIV	Ratio	Peripheral bus clock
00 (Initial value	2	41.66MHz
01	4	20.83MHz
10	8	10.42MHz
11	1	83.33MHz

Note：i_RCKDIV="11" is a test setting.
Modules connected to the local bus cannot be operated at 83.33MHz. Please do not set i_RCKDIV =”11" for normal use.

18.3.3.4.2. Clock control

The clock output to the local peripheral bus module is controlled by the clock control signal (i_CLKEN[12]).
Please confirm the control signal of each module with Table 19-1. If the clock control signal corresponding to each clock output is "H", the clock is output. The clock stops when the control signal is “L”. If the clock has been stopped, "L" is output.
The clock is stopped by default.

i_CLKEN[n]=”H”	The clock is output.
i_CLKEN[n]=”L”	The clock is stopped.

(n shows bit number.)

18.3.3.5. Display clock generation

BitClock (i_BCLK) input from the Clk Synth module generates a clock signal (Pixel Clock) for the display.
The display clock is half cycle of BitClk.

18.3.3.5.1. Clock control

The display clock is controlled by the clock control signal (i_CLKEN [2:1]).
The control setting of the clock can be set to ClockEnable Register of CCNT.
Please confirm the control signal of each module with Table 19-1. If the clock control signal corresponding to each clock output is "H", the clock is output. The clock stops when the control signal is “L”. If the clock has been stopped, "L" is output.
The clock is stopped by default.

i_CLKEN[n]=”H”	The clock is output.
i_CLKEN[n]=”L”	The clock stops.

(n shows bit Number.)

i_CLKEN [2:1] is effective to the Sprite GDC module and the Display Control module.
The AHB clock and the display clock are controlled at the same time.

18.3.3.6. Generation of reset

A reset can be generated for any module based on power-on reset signal (ix_RESET) and after waiting for a stability period of 1ms of the PLL clock.
Reset is permitted at power-on reset signal ="L"
All resets are permitted for i_RSTEN[31]="0"
The module list shown below lists the reset connection destinations.

Table 19-2 Reset connection list

System	Module	Port name	Control
AHB bus	CMD Sequencer	ox_HRST_CMDS	None
	R-Handler	ox_HRST_APIX	i_RSTEN[1] &[31]
	Host I/F	ox_HRST_HOST	i_RSTEN[2] &[31]
	Sprite GDC	ox_HRST_SGDC	i_RSTEN[3] &[31]
	Display Controller	ox_HRST_DISP	i_RSTEN[4] &[31]
	TCON,CLUT,DITH,SIG	ox_HRST_DPERI	i_RSTEN[5] &[31]
	RLD	ox_HRST_RLD	i_RSTEN[6] &[31]
	DMAC	ox_HRST_DMAC	i_RSTEN[7] &[31]
	Serial Flash I/F	ox_HRST_SFIF	i_RSTEN[8] &[31]
	Flash MEM I/F	ox_HRST_FMIF	i_RSTEN[9] &[31]
	INTMEM 8K	ox_HRST_MIF8K	i_RSTEN[10] &[31]
	INTMEM 64K0	ox_HRST_MIF64K0	i_RSTEN[11] &[31]
	INTMEM 64K1	ox_HRST_MIF64K1	i_RSTEN[12] &[31]
	ConfigFIFO	ox_HRST_CNFG	i_RSTEN[13] &[31]
	AHB Bus Matrix1	ox_HRST_BM1	i_RSTEN[16] &[31]
	AHB Bus Matrix2 & TIC & AHB2AHB & Default Slave	ox_ALL_RST	i_RSTEN[31]
	CMD Sequencer	ox_ARST_CMDS	i_RSTEN[31]
local peripheral bus	RESOURCE	ox_RRST	i_RSTEN[14] &[31]
APB bus	GPIO	ox_PRST	i_RSTEN[15] &[31]

18.3.3.7. PLL clock stability wait period

After releasing a power-on reset, the PLL clock continues to cyclically reset internal modules (with a period of 1ms) until it has stabilized.
It continues to reset the circuit of the dividing frequency in the CRG module etc.
After the PLL clock stability period has expired, the reset of internal modules is released.
The hardware that counts the PLL clock stability wait period operates using an external reference clock (i_CLK:4MHz).

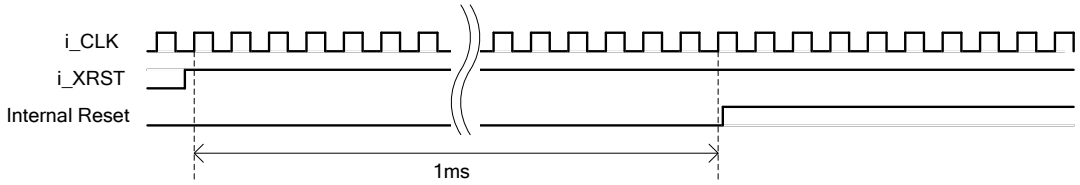


Figure 19-2 PLL stable waiting period timing diagram

18.3.3.8. Reset Control

18.3.3.8.1. Module Reset Control

The reset control of a module is handled according to the reset control signal (ix_RSTEN).
The control setting of reset can be set to SoftReset Register of CCNT.
Please check the control signal that controls each module using Table 19-2.
A reset is permitted by modifying the reset control signal corresponding to each reset output signal "L" ("L" output). Reset is released by making the reset control signal "H" ("H" output).

ix_RSTEN[n]="H"	Reset release
ix_RSTEN[n]="L"	Reset

(n shows bit Number.)

18.3.3.8.2. ALL Reset

Reset is permitted to all modules connected to the CRG according to the reset control signal [31].
Please confirm the modules connected with CRG by Table 19-2.

The following modules do not output a reset from the CRG. Reset from the power-on reset module is connected directly.

- APIX PHY
- CLOMO
- Clock Synth
- CCNT

18.4. Registers

18.4.1. Format of Register Descriptions

- Endian
  Only Little Endian access corresponds to the register of this module.

- Address
  “Address” shows the address of the register. (Base address + Offset address)

- Bit
  “Bit” shows the bit number of the register.

- Name
  “Name” shows the bit field name of the register.

- R/W
  “R/W” shows the attribute of Read/Write in each Bit field.
  R0: The reading value is always "0".
  R1: The reading value is always "1".
  W0: The writing value is always "0".When “1” is written, it is disregarded.
  W1: The writing value is always "1". When “0” is written, it is disregarded.
  R: Read
  W: Write

Note : If a value is written to registers/bitfields that list R0, R1 and R in the following descriptions, then this value will not be changed in those registers/bitfields.

- Initial value
  “Initial value” is an initial value when reset is released.
  0: It becomes "0".
  1: It becomes "1".
  X: It becomes irregular.

18.4.2. Global Address

For the module base address please refer to the chapter 3 Memory map.
Only ClkSynth has the register.

18.4.3. Register summary

Address	Register Name	Description
Base address + 0_H	GeneralCtrl	general clock control register
Base address + 4_H	NominalFrequency	upper frequency of synthesized clock
Base address + 8_H	SscCtrl0	Control register 0 for SSC (spread spectrum clocking)
Base address + C_H	SscCtrl1	Control register 1 for SSC (spread spectrum clocking)
Base address + 10_H	Status	Status register of module
Base address + 14_H	TestClockSourceSelection	select test clock source

18.4.4. Register Description

18.4.4.1. GeneralCtrl

Register address	BaseAddress + 0_H																															
Bit number	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
Field name																														IRQRestartEn	SSCen	SWReset
R/W																														RW	RW	RW
Reset value																														0_H	1_H	1_H

general clock control register

Bit 2 IRQRestartEn
enable IRQ output on register field ISTSRestart

Bit 1 SSCen
enable spread spectrum clocking/frequency hopping; 1=enabled, 0=disabled

Bit 0 SWReset
software reset; change configuration only during reset is asserted; 1=active, 0=inactive

18.4.4.2. NominalFrequency

Register address	BaseAddress + 4_H																															
Bit number	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
Field name									Ndiv													Pdiv			Psub							
R/W									RW													RW			RW							
Reset value									14_H													4_H			$2A_H$							

upper frequency of synthesized clock

Bit 23 - 16 Ndiv
divider value N of N.P; N = 2...255

Bit 10 - 8 Pdiv
divider value P of N.P; P = 0...4

Bit 7 - 0 Psub
sub P value to set frequencies between N.P

18.4.4.3. SscCtrl0

Register address	BaseAddress + 8_H																															
Bit number	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
Field name						DeltaPdiv1			DeltaPsub1								HoppingPeriod															
R/W						RW			RW								RW															
Reset value						0_H			0_H								0_H															

Control register 0 for SSC (spread spectrum clocking)

Bit 26 - 24 DeltaPdiv1
value is added to value of P; DeltaPdiv1 = 0...4

Bit 23 - 16 DeltaPsub1
value is added to value of Psub; DeltaPsub1 = 0...255

Bit 15 - 0 HoppingPeriod
value specifies the Period after how many clock cycles frequency changes; 0 : change after every clock cycle

18.4.4.4. SscCtrl1

Register address	BaseAddress + C_H																													
Bit number	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
Field name											DeltaPdiv3			DeltaPsub3								DeltaPdiv2			DeltaPsub2					
R/W											RW			RW								RW			RW					
Reset value											0_H			0_H								0_H			0_H					

Control register 1 for SSC (spread spectrum clocking)
Bit 21 - 19 DeltaPdiv3
value is added to value of P; DeltaPdiv3 = 0...4
Bit 18 - 11 DeltaPsub3
value is added to value of Psub; DeltaPsub3 = 0...255
Bit 10 - 8 DeltaPdiv2
value is added to value of P; DeltaPdiv2 = 0...4
Bit 7 - 0 DeltaPsub2
value is added to value of Psub; DeltaPsub2 = 0...255

18.4.4.5. Status

Register address	BaseAddress + 10_H																													
Bit number	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
Field name																												ISTSRestart				NPerror
R/W																												RW				R
Reset value																												0_H				0_H

Status register of module
Bit 1 ISTSRestart
clock is/was hanging; HLB 'npclk_divider' wants to be restarted
Bit 0 NPerror
N.P value is not valid for N.P. clock divider module

18.4.4.6. TestClockSourceSelection

Register address	BaseAddress + 14_H																													
Bit number	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
Field name																												BypassTestClock				SsClkEn
R/W																												RW				RW
Reset value																												0_H				0_H

select test clock source
Bit 1 BypassTestClock
0: generated clock is used; 1: bypass test/ssclk to output
Bit 0 SsClkEn
0: generated clock is used; 1: enable input ssclk to output

18.5. Application Note

18.5.1. Flow

18.5.1.1. Clock setting of CRG

A clock setting flow example is shown below.

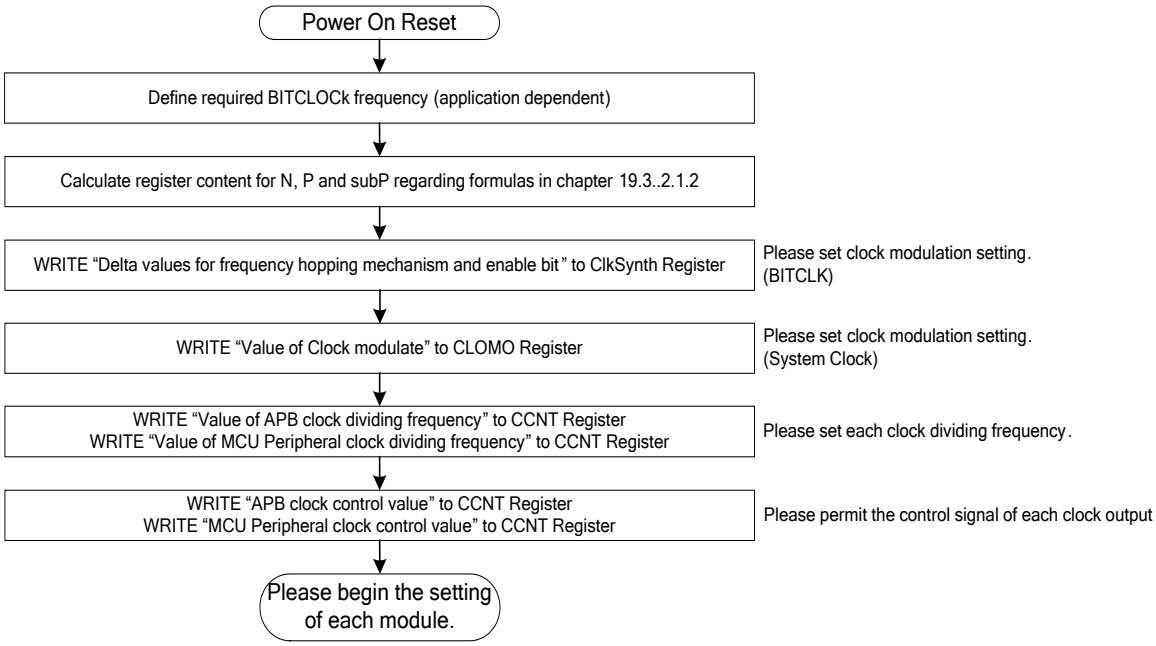


Figure 19-3 Clock setting flow

19. Clock Modulator

This chapter describes the MB88F333 Clock Modulator.

19.1. Outline

The clock modulator is implemented for the reduction of electromagnetic interference - EMI, by spreading the spectrum of the clock signal over a wide range of frequencies.
The module is fed with an unmodulated reference clock with frequency F0, provided by the System clock(AHB bus clock). This reference clock is frequency modulated, controlled by a random signal.
The mean frequency of the modulated clock is equal to the reference clock frequency F0.

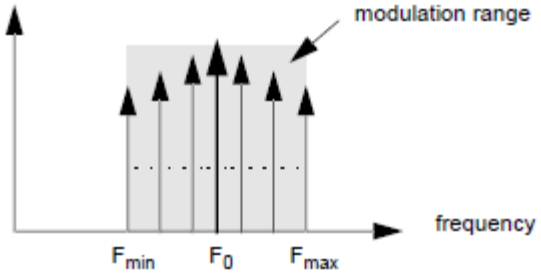


Figure 20-1 Frequency spectrum of the modulated clock (fundamentals only)

19.1.1. Modulation degree and frequency resolution in frequency modulation mode

The maximum and minimum frequencies (F_{max} and F_{min}) of the modulated clock are defined by the modulation degree parameter. Furthermore, the resolution of the modulation range is selectable in 7 steps from low (1) to high (7). Higher resolution implies a finer granularity of discrete frequencies in the spectrum of the modulated clock but less possible modulation degrees.
In general the highest possible frequency resolution combined with the highest possible modulation degree results in the highest EMI reduction. However, in some situations lower modulation degrees may result in better EMI behavior. Please refer to the table of possible settings in Table 20-4 Clock Modulator settings.

19.2. Clock Modulator Registers

This section lists the clock modulator registers and describes the function of each register in detail.

19.2.1. Clock modulator registers

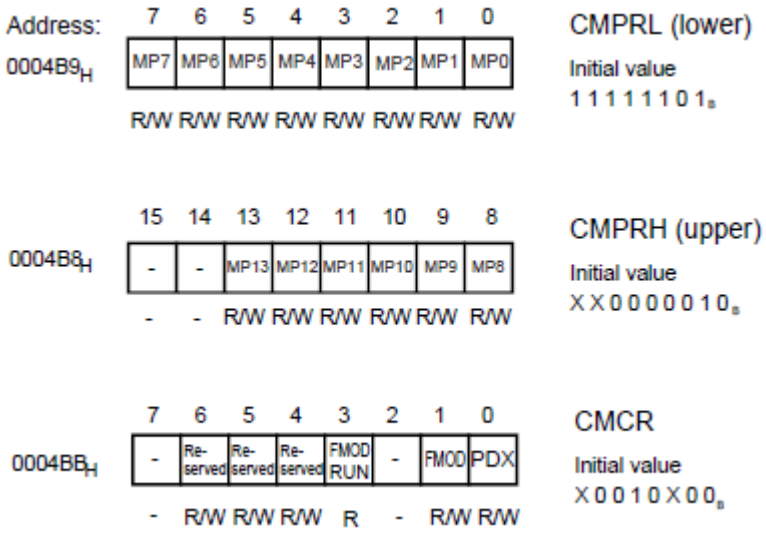


Figure 20-2 Clock modulator registers

19.2.2. Clock Modulator Control Register (CMCR)

The Control Register (CMCR) has the following functions:

- Set the modulator to power down mode
- Modulator enable/disable in frequency modulation mode
- Indicates the status of the modulator

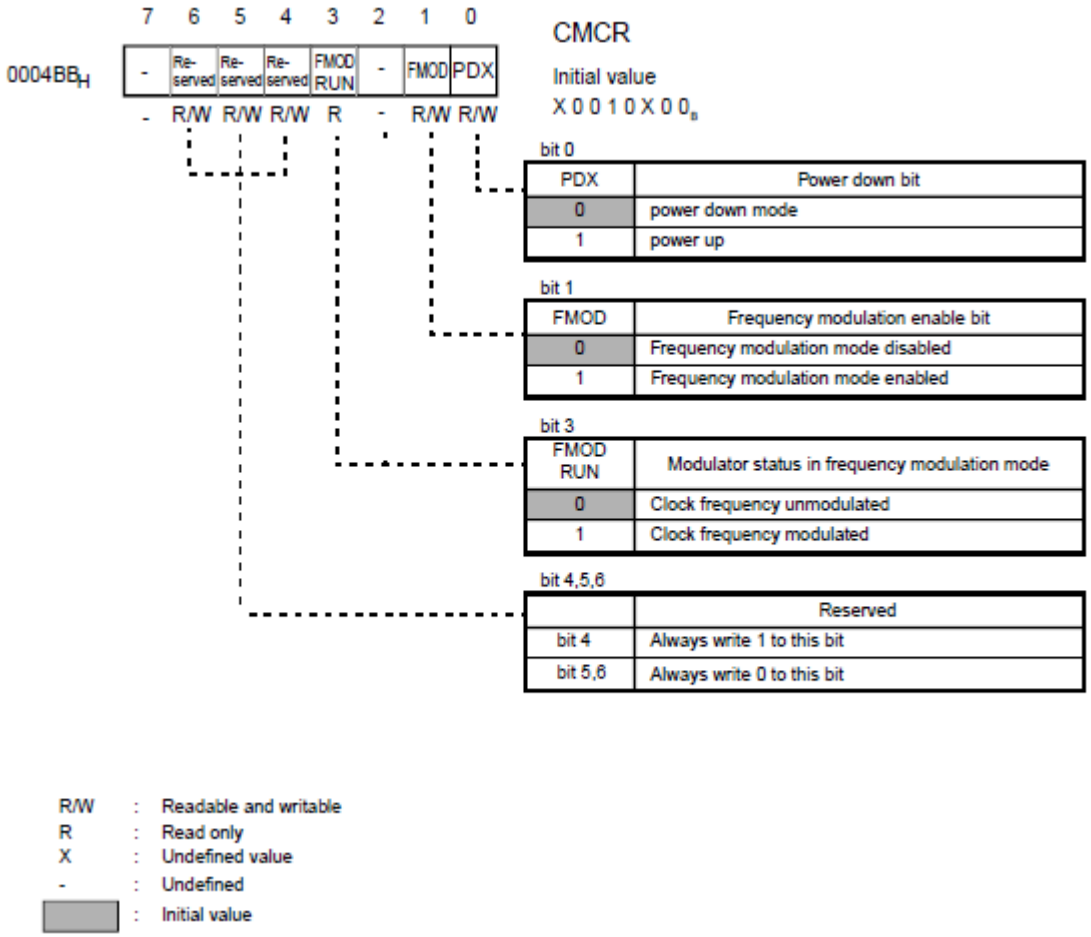


Figure 20-3 Configuration of the clock modulator control register (CMCR)

The bits FMODRUN, FMOD, PDX control or indicate the status of the frequency modulation mode. Frequency modulation mode needs additional configuration via the CMPR register.

19.2.3. Clock modulator control register contents

Table 20-1 Function of each bit of the clock modulator control register

Bit name		Function
bit7	undefined	It is undefined. An initial value is irregular..
bit 6 to 5	Reserved	Always write 0 to this bit.
bit 4	Reserved	Always write 1 to this bit.
bit 3	FMOD RUN: Modulator status in frequency modulation mode bit	"0": MB88F333 is running with unmodulated clock "1": MB88F333 is running with frequency modulated clock FMODRUN indicates the status of the modulator output clock in frequency modulation mode (FMOD=1). If the output clock is frequency modulated, MODRUN is set to 1, otherwise MODRUN is set to 0. After enabling the frequency modulation mode by setting FMOD RUNto 1, the modulator is calibrated. During this time, the clock is unmodulated. Therefore it takes several us before the output clock switches to modulated clock and the FMODRUN bit is set to 1. During normal operation, after calibration is finished, the clock is not switched to unmodulated clock mode. Due to the synchronization of the FMODRUN signal and the synchronized switching to unmodulated clock, it takes less than 9 x T0 (input clock period) before FMODRUN changes to 0 and the clock switches to unmodulated clock mode after the modulator is disabled. The FMODRUN bit is read only. Writing to FMODRUN has no effect. Before changing the parameter register CMPR, the modulator must be disabled -> FMOD=0 and FMODRUN=0.
bit 2	Undefined	It is undefined. An initial value is irregular..
bit 1	FMOD: Frequency modulation enable bit	"0": Frequency modulation disabled. "1": Frequency modulation enabled. To enable the modulator in frequency modulation mode, FMOD must be set to 1. Before the modulator can be enabled, the PLL must deliver a stable reference clock (PLL lock time must be elapsed). Each PLL output frequency offers a set of possible modulation parameters. The selected setting (CMPR register) and the PLL frequency must match. Please refer to the CMPR register description. Whenever the PLL output frequency is changed or the PLL is switched off e.g. in power down modes, the modulator must be disabled before -> FMOD=0 and FMODRUN=0. Before the modulator can be enabled, it must be switched from power down to active mode by setting PDX to 1. And the startup time of 6us must be awaited. Please refer to the application note for a description of the recommended startup sequence. Before the modulator can be enabled in frequency modulation mode, a proper setting must be selected via the parameter register CMPR. After enabling the frequency modulation mode by setting FMOD to 1, the modulator is calibrated. During this time, the clock is unmodulated. Therefore the output clock does not switch immediately to modulated clock. The status of the clock (frequency modulated / unmodulated) is indicated by the FMODRUN status bit. Please refer to the FMODRUN bit description. Due to the synchronization of the FMOD signal and the synchronized switching to unmodulated clock, it takes less than 9 x T0 (input clock period) before the clock switches to unmodulated clock after the modulator is disabled. The modulator can be disabled at any time. Before changing the parameter register CMPR, the modulator must be disabled -> FMOD=0 and FMODRUN=0.
bit 0	PDX: Power down bit	"0": Power down mode "1": Power up PDX is the power down signal for the modulator. Before the frequency modulation mode can be enabled, this bit must be set to 1 and the startup time of 6us must be waited. Please refer to the application note for a description of the recommended startup sequence. Before switching to power down mode (PDX=0), the modulator must be disabled -> FMOD=0 and FMODRUN=0.

The table below summarizes the modulator states:

Table 20-2 States of the modulator

		FMOD	PDX	FMODRUN (read only)
modulator disabled		0	0	0
modulator power on, waiting modulator startup time (> 6 us)		0	1	0
modulator enabled in frequency modulation mode, modulator is calibrating, modulation not active		1	1	0
modulator is running in frequency modulation mode modulation is active		1	1	1
	others settings are not allowed			

19.2.4. Clock Modulation Parameter Register (CMPR)

The Modulation Parameter Register (CMPR) determines the modulation degree in frequency modulation mode.

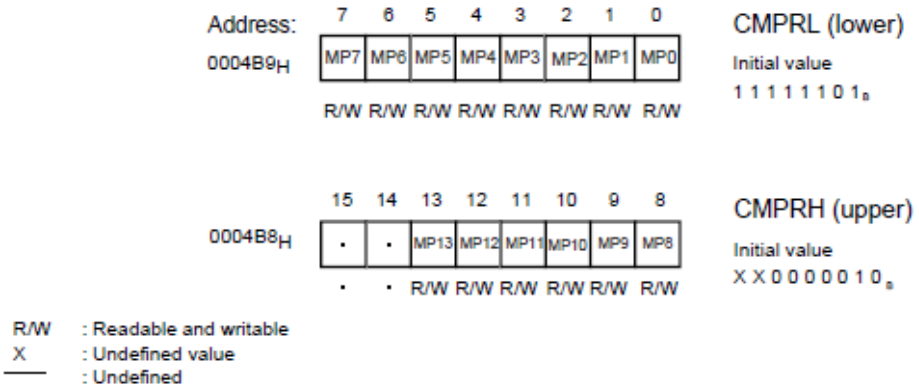


Figure 20-4 Modulation parameter register

- The modulation parameter determines the degree of modulation and the maximal and minimal occurring frequencies in the modulated clock. Please refer to the application note (see below) for a description of how to select the optimal setting.
- Each set of possible modulation parameters refers to a particular PLL frequency. The PLL frequency and the selected parameter must match. Please refer to the following table (Table 20-3) of possible settings.
- The modulation parameter only egffects the frequency modulation mode.

Note:
The modulation parameter may only be changed when the modulator is disabled and the RUN flag is 0 (FMOD=0, FMODRUN=0).

19.2.5. Modulation parameter register contents

Table 20-3 Function of each bit of the modulation parameter register (CMPR)

Bit name		Function
bit 15, 14	Undefined	It is undefined. An initial value is irregular..
bit 13 to 0	MP13 to 0: Modulation Parameter bits	Depending on the PLL frequency, the following modulation parameter settings are possible. The corresponding CMPR register value has been described to Table 20-4.

F0: Frequency of unmodulated input clock (system clock)

resolution: Resolution of frequencies in the modulated clock. low (1) to high (7)
Fmin: Minimal frequency of the frequency modulated clock
Fmax: Maximal frequency of the frequency modulated clock
CMPR: Register setting of the CMPR register

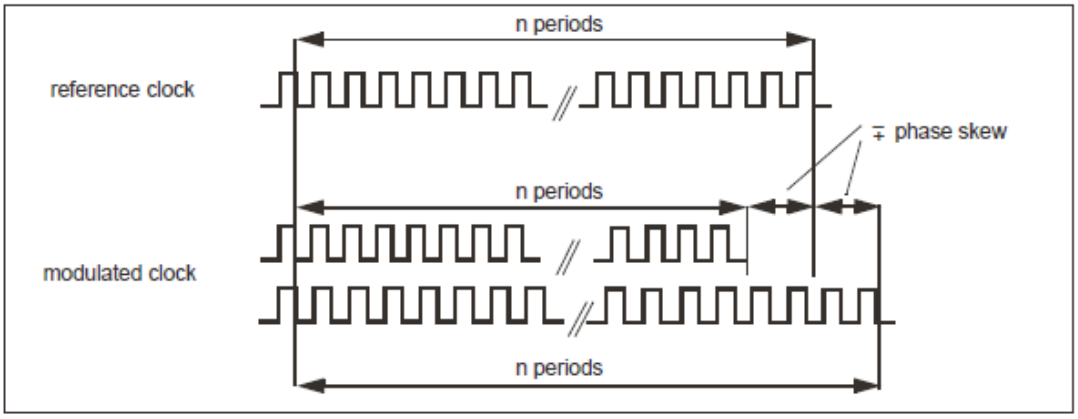


Figure 20-5 Skew of modulated clock vs. unmodulated clock

Table 20-4 shows the recommended setting for several MB88F333 clocks and modulation parameters:

19.3. Application Note

This section covers the following topics:

- Startup/stop sequence for frequency modulation mode.
- Modulation parameter for frequency modulation mode.

19.3.1. Recommended startup sequence for frequency modulation mode

Start

1. Switch modulator from power down to power up mode PDX=1
2. Set CMPR register to a proper setting
3. Enable frequency modulation mode FMOD=1
   After the calibration is finished, the clock switches from unmodulated to modulated clock and the FMODRUN flag changes to 1
   ... running...

stop

4. Disable modulator FMOD=0
5. Wait until FMODRUN changes to 0
6. Switch to power down mode PDX=0

Note:
Do not enable the modulator before the PLL lock time has elapsed. Do not disable the PLL while the modulator is running.

19.3.2. Modulation parameters for frequency modulation mode

It is not possible to recommend a particular modulation parameter setting to achieve a specific reduction in EMI. The best setting depends much on the actual application, the whole system and the requirements.
In order to find the optimal modulation parameter setting in frequency modulation mode, the following approach is recommended.

Note: For full sprite and peripheral performance only one setting is supported.

		Full Performance	Reduced Performance
1.	Define the required PLL frequency based on performance needs. ClockDiv register of CCNT is set.	e.g. 83.3MHz	e.g. 41.7MHz
2.	Determine the maximal allowed clock frequency of the MB88F333	e.g. 93.0 MHz	e.g. 93.0 MHz
3.	Choose the setting with the highest resolution and the highest modulation degree, whose maximal frequency is below the maximal allowed clock frequency of the MB88F333.	e.g. resolution:1, degree:1, CMPR=0x026F (Fmax= 93.0MHz)	e.g. resolution:3, degree:2, CMPR=0x046E (Fmax= 48.8MHz)
4.	Perform EMI measurements		
5.	If the EMI measurements do not fulfill the requirements, you may either reduce the modulation degree at the same frequency resolution (this may improve the reduction in the upper frequency band > 100MHz, but decrease the reduction of the fundamental < 100MHz)	Not possible	e.g. resolution:2, degree:2, CMPR=0x04AC (Fmax= 57.13MHz)
	or increase the modulation degree at a lower frequency resolution (this may improve the reduction of the fundamental < 100MHz, but worsen the reduction in the upper frequency band > 100MHz)	Not possible	or e.g. resolution:2, degree:1, CMPR=0x02AE (Fmax= 48.77MHz)
6.	Repeat item 3) with the new setting and continue until the best setting is identified		

19.3.3. Recommended settings

The following table shows all possible settings of the clock modulator for two system clock frequencies.

Table 20-4 Clock Modulator settings

F0 AHB Bus Clock frequency	clock modulator setting					Remark
	resolution	modulation degree	F_{max}	F_{min}	CMPR	
83.33MHz	1	1	93.02MHz	75.47MHz	0x026F	recommended
41.66MHz	1	1	45.45MHz	38.46MHz	0x026F	reduced sprite performance
41.66MHz	2	1	48.77MHz	36.36MHz	0x02AE	reduced sprite performance
41.66MHz	3	1	52.63MHz	34.48MHz	0x02ED	reduced sprite performance
41.66MHz	4	1	57.13MHz	32.78MHz	0x032C	reduced sprite performance
41.66MHz	1	2	48.77MHz	40.81MHz	0x046E	reduced sprite performance
41.66MHz	2	2	57.13MHz	32.78MHz	0x04AC	reduced sprite performance
41.66MHz	1	3	52.62MHz	34.48MHz	0x066D	reduced sprite performance
20.83MHz	1	1	22.47MHz	19.41MHz	0x026F	reduced sprite performance
20.83MHz	2	1	24.09MHz	18.35MHz	0x02AE	reduced sprite performance
20.83MHz	3	1	25.97MHz	17.39MHz	0x02ED	reduced sprite performance
20.83MHz	4	1	28.16MHz	16.53MHz	0x032C	reduced sprite performance
20.83MHz	5	1	30.76MHz	15.75MHz	0x036B	reduced sprite performance
20.83MHz	6	1	33.89MHz	15.04MHz	0x03AA	reduced sprite performance
20.83MHz	7	1	37.73MHz	14.39MHz	0x03E9	reduced sprite performance
20.83MHz	1	2	24.09MHz	18.35MHz	0x046E	reduced sprite performance
20.83MHz	2	2	28.16MHz	16.53MHz	0x4AC	reduced sprite performance
20.83MHz	3	2	33.89MHz	15.04MHz	0x04EA	reduced sprite performance
20.83MHz	1	3	25.97MHz	17.39MHz	0x066D	reduced sprite performance
20.83MHz	2	3	33.89MHz	15.04MHz	0x06AA	reduced sprite performance
20.83MHz	1	4	28.16MHz	16.53MHz	0x086C	reduced sprite performance

The following table shows all possible settings of the clock modulator for three system clock frequencies.

20. Chip Control (CCNT)

This chapter describes the Chip Control of the MB88F333.

20.1. Outline

The Chip Control Module (referred to CCNT in this document) delivers chip information and executes module control and external pin multiplexing.

20.2. Features

The CCNT module has the following features.

20.2.1. Features

- Chip Information
  Used to read the revision information, the production year and chip name information of this chip.
- Internal interrupt control
  This module has a function to report internal interrupt information to an external point.
- Clock control
  The clock division register is used to change the dividing clock ratio and to select the modulated clock. Unnecessary clock output is controlled and energy-saving is possible by setting the clock enable register.
- Soft reset interface
  Each module can be initialized by issuing a soft reset to each module in the register.
- Data swap
  control of the byte lane of each module.
- Pin switch
  External pin multiplexing selection.
- OSC control
- APIX interface termination resistor calibration control

20.3. Function

20.3.1. Block diagram

Figure 21-1 shows the CCNT's block diagram.

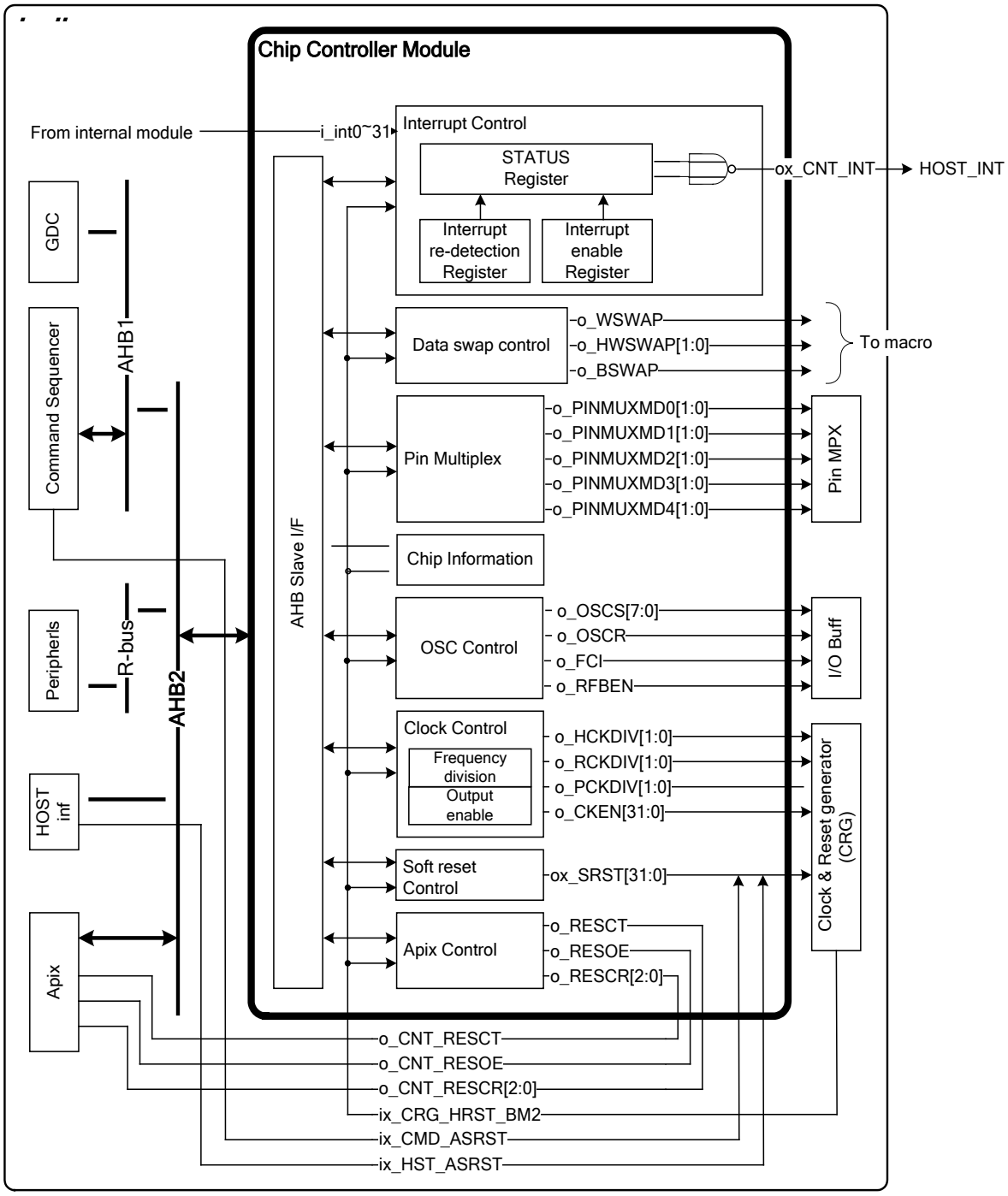


Figure 21-1 Block diagram of CCNT

20.3.2. Interrupt control

Interrupt control stores the interrupt signal from a module in the Status register and outputs an external interrupt signal (ox_INT).

20.3.3. Interrupt detection

The rising edge of an internal interrupt signal is detected.
Detected interrupt information is stored in the Status register.

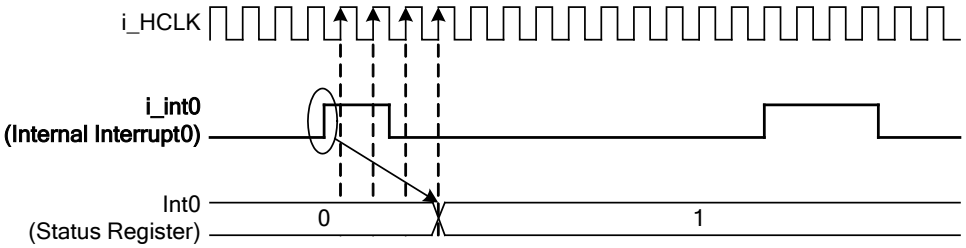


Figure 21-2 Timing diagram of interrupt detection

20.3.3.1. Interrupt clear

Interrupt information is cleared by writing 1 to the corresponding bit of the Status register.
There are two methods of interrupt detection after the Status register has been cleared, as described below. The following detection handling methods can be set in the Interrupt re-detection register.
(Refer to 21.4.4.2 and 21.4.4.3)

1) After the status register has been cleared, an interrupt can be detected again (Interrupt re-detection=0)

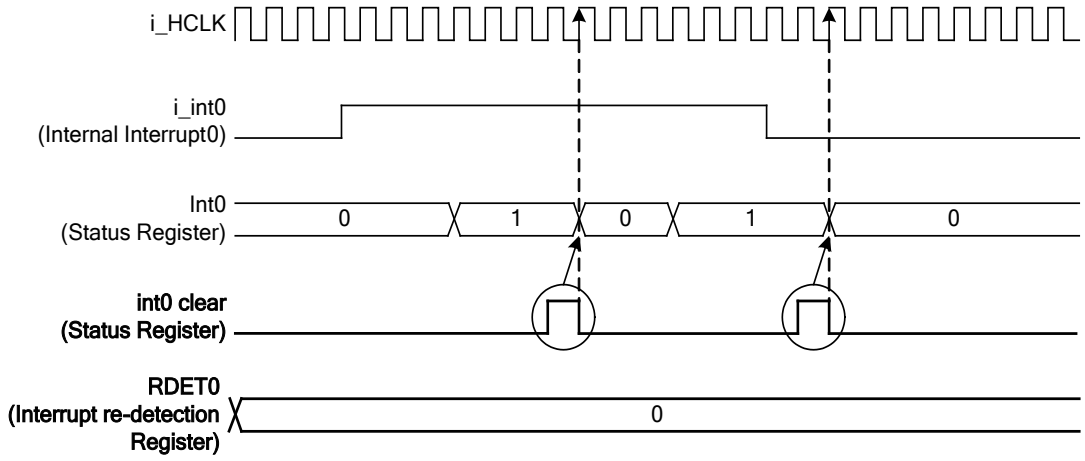


Figure 21-3 Timing diagram of interrupt re-detection (Interrupt re-detection=0)

2) After the status register has been cleared, an interrupt is not detected (Interrupt re-detection=1)

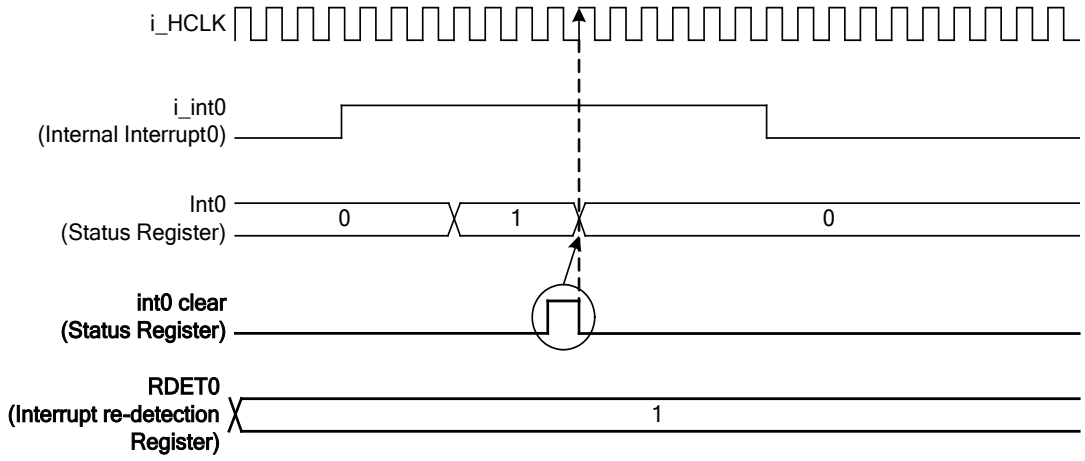


Figure 21-4 Timing diagram of interrupt no re-detection (Interrupt re-detection=1)

20.3.3.2. External output signal

The interrupt output signal to external targets is ox_INT.

This ox_INT signal is a logical sum of output information in the Status register.

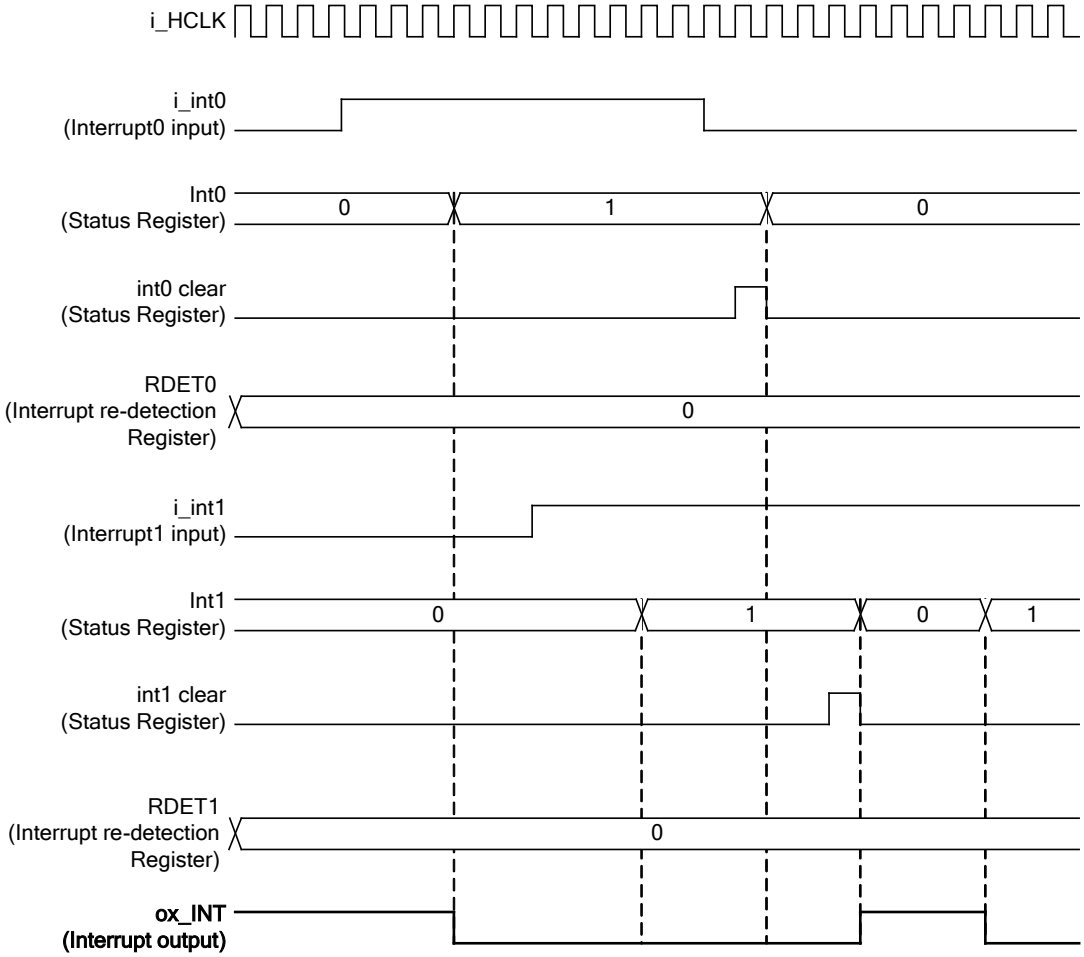


Figure 21-5 Timing diagram of external output signal

20.3.3.3. External interrupt output control

Interrupt signals (Status register) which are disabled via the Interrupt enable register are not output in the external interrupt signal (ox_INT). (Refer to 21.4.4.4)

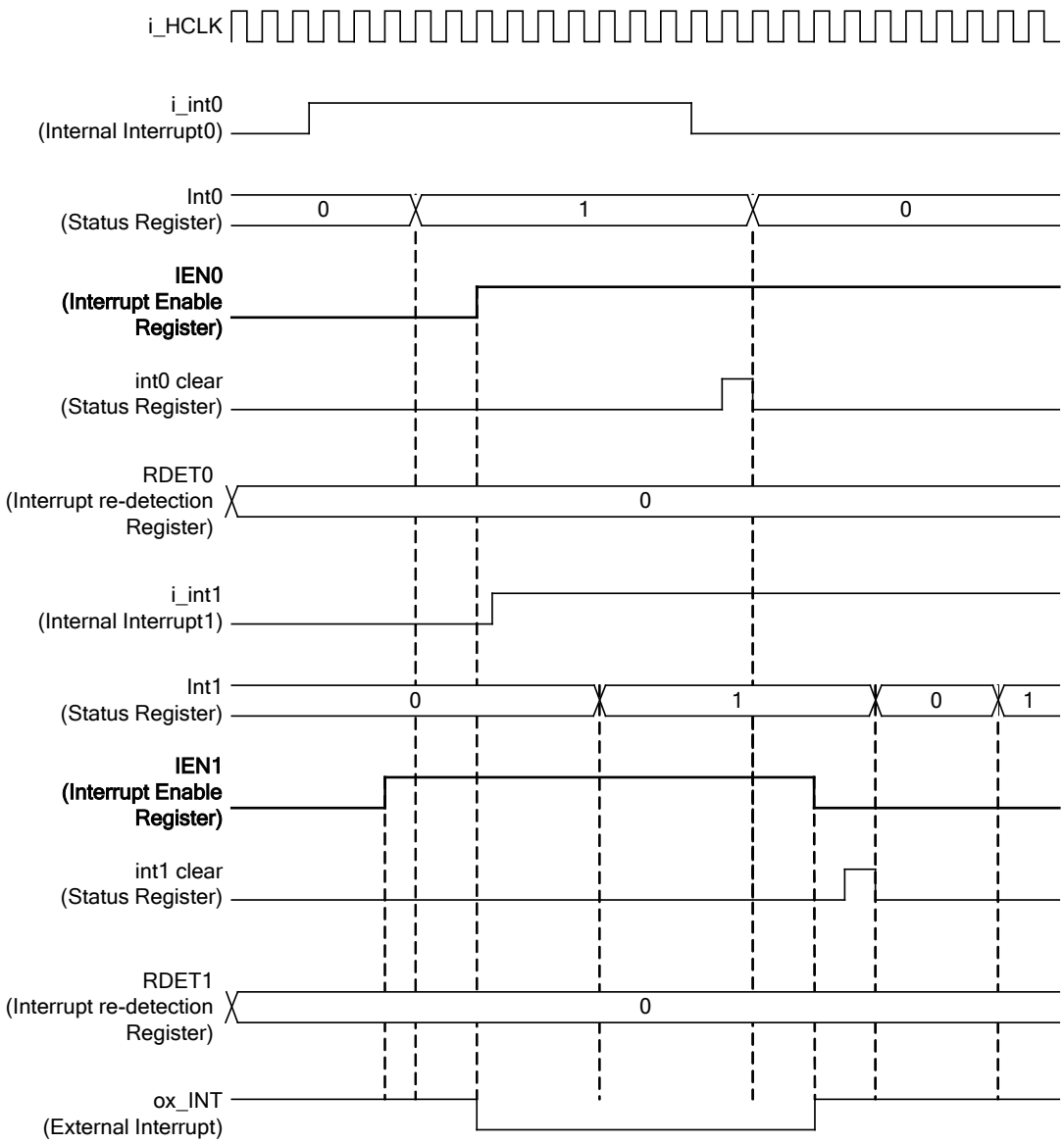


Figure 21-6 Timing diagram of external interrupt output control

20.3.4. Clock control

Clock control controls the clock output to the modules.
Unnecessary clock output can be controlled for energy-saving purposes.

20.3.4.1. Clock output enable

Clock output enable outputs the control information set to the Clock Enable Register, the Clock & Reset Generator module (following CRG) and the AHB bus. (Refer to 21.4.4.6)
Note: The clock to the AHB BUS is enabled in order to avoid a bus lock-up.

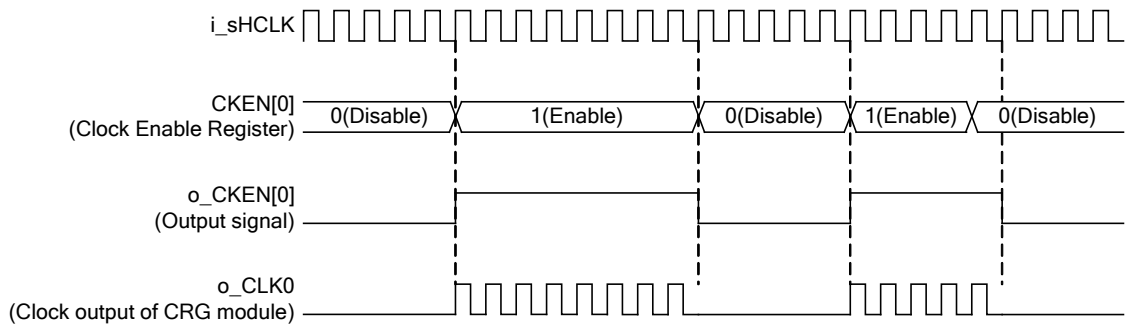


Figure 21-7 Clock enable and clock output

20.3.4.1.1. Target module of Clock output enable

Clock control target modules are listed below:

Signal name	Target module
o_CKEN[0]	Command Sequencer
o_CKEN[1]	Sprite GDC
o_CKEN[2]	Display Controller, Display Peripherals
o_CKEN[3]	Reserved
o_CKEN[4]	DMAC
o_CKEN[5]	RL Decompress
o_CKEN[6]	Reserved
o_CKEN[7]	Memory I/F
o_CKEN[8]	Internal RAM1 I/F
o_CKEN[9]	Internal RAM2 I/F
o_CKEN[10]	Internal RAM3 I/F
o_CKEN[11]	Config FiFo
o_CKEN[12]	R-BUS Peripherals
o_CKEN[13]	APB Peripherals
o_CKEN[31:14]	Reserved

20.3.4.2. Frequency division unit

The frequency division unit outputs the ratio of the clock dividing frequency set in the Clock Division register to the CRG module.

20.3.4.2.1. AHB BUS clock

The frequency division ratio of the AHB BUS clock can be set as follows.

HCKDIV[1:0]	Dividing ratio	AHB BUS Clock
00 (initial)	1	83.33MHz
01	2	41.66MHz
10	4	20.83MHz
11	1	83.33MHz

20.3.4.2.2. APB BUS clock

The frequency division ratio of the APB BUS clock can be set as follows. The base clock of the dividing frequency is the AHB BUS clock.

RCKDIV[1:0]	Dividing ratio	PCLK (APB BUS Clock)
00 (initial)	2	PCLK = HCLK/2
01	4	PCLK = HCLK/4
10	8	PCLK = HCLK/8
11	1	PCLK = HCLK *1

Note*1 : The APB BUS clock limit is 41.66MHz.
Therefore, please do not write 2'b11 to PCKDIV if you set 2'b00 or 2'b11 in HCKDIV.

20.3.4.2.3. Peripheral BUS clock

The frequency division ratio of the Peripheral BUS clock can be set as follows. The base clock of the dividing frequency is the AHB BUS clock.

RCKDIV[1:0]	Divide ratio	RCLK (Peripheral BUS Clock)
00 (initial)	2	RCLK = HCLK/2
01	4	RCLK = HCLK/4
10	8	RCLK = HCLK/8
11	1	RCLK = HCLK *2

Note*2 : The Peripheral clock limit is 41.66MHz.
Therefore, please do not write 2'b11 to RCKDIV if you set 2'b00 or 2'b11 in HCKDIV.

20.3.4.3. Limitations

Please confirm that the target module is in an IDLE state before writing to its Clock Control. If you change its clock control setting if it is not in IDLE, it is possible that the GDC will lock up.

20.3.5. Soft reset control

The Soft reset control unit can execute a reset via the Soft Reset register setting for the target module.
In addition, all modules (excluded CRG, Clksynth, and CLOMO) can be initialized by the ix_ASRST signal from the Host I/F module.

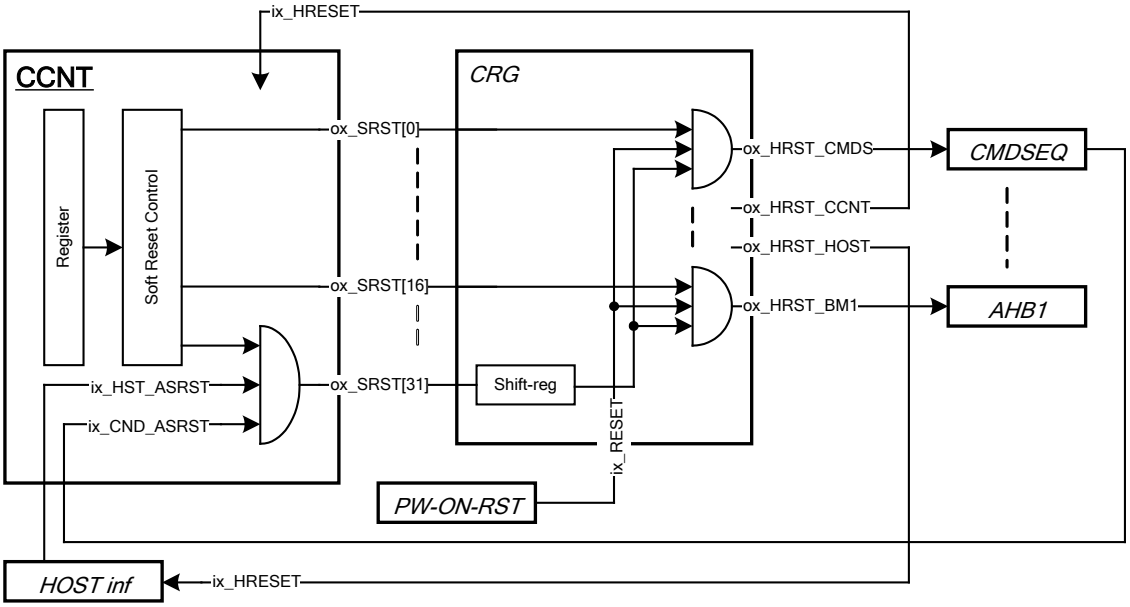


Figure 21-8 Soft reset control image

20.3.5.1. Soft Reset target modules

The targets of the Soft Reset unit are as follows:

Singnal name	Target module
ox_SRST[0]	Command Sequencer
ox_SRST[1]	Remote Handler
ox_SRST[2]	Host I/F
ox_SRST[3]	Sprite GDC
ox_SRST[4]	Display Controller
ox_SRST[5]	Display Peripherals (SIG, CLUT, Dithering, TCON)
ox_SRST[6]	RL Decompress
ox_SRST[7]	DMAC
ox_SRST[8]	Reserved
ox_SRST[9]	Memory I/F
ox_SRST[10]	Internal RAM1 I/F
ox_SRST[11]	Internal RAM2 I/F
ox_SRST[12]	Internal RAM3 I/F
ox_SRST[13]	Config FiFo
ox_SRST[14]	R-BUS Peripherals
ox_SRST[15]	APB Peripherals
ox_SRST[16]	AHB1 BUS MATRIX
ox_SRST[30:17]	Reserved
ox_SRST[31]	All modules

20.3.5.2. Limitations

- There is a possibility that the BUS is locked up if a soft reset is executed when the target module is accessing BUS.
- The following modules are not effected by the 'All modules' target (ox_SRST[31]):
  CRG, CLOCKSYNTH, CLOMO, CCNT, APIX PHY

20.3.6. Chip Information

The Chip Information unit provides the information about the version of this GDC, the Chip ID and the production year.

20.3.6.1. Version

The version information of this GDC (refer to 21.4.4.1).

20.3.6.2. Chip ID

The Chip ID information of this GDC (refer to 21.4.4.1).

20.3.6.3. Year

The production year information of this GDC (refer to 21.4.4.1).

20.3.7. Data swap control

The Data swap control unit sets the data swap characteristics of bytes during accesss to Words and Hwords. Individual settings can be used for all the modules in the GDC. In addition, Data swap settings permit flexible transitions between Words and Hwords as it is possible to do to individual byte accesses.

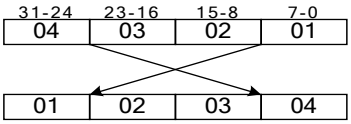
20.3.7.1. Word access

Data Swap during Word access

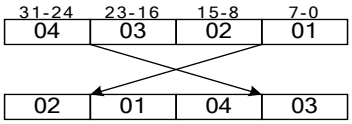
Data swap during Word access can use a Hword swap and a Byte swap in Hword units.

R-BUS and clomo only

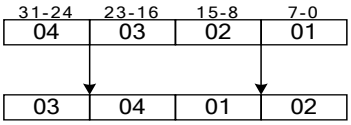
1) Hword swap ON, Byte swap ON



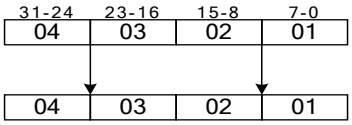
2) Hword swap ON, Byte swap OFF



3) Hword swap OFF, Byte swap ON

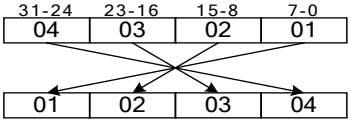


4) Hword swap OFF, Byte swap OFF

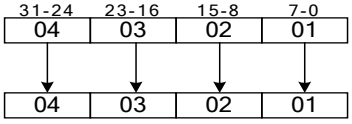


Other modules

1) Byte swap ON



2) Byte swap OFF

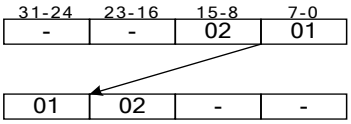


20.3.7.2. Hword access

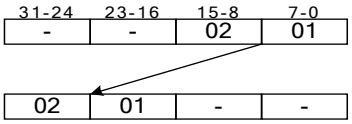
Data Swap during Hword access

Data swap during Hword access use Hword swap and Byte swap in Hwords units.

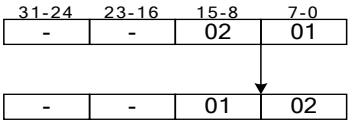
1) Hword swap ON, Byte swap ON



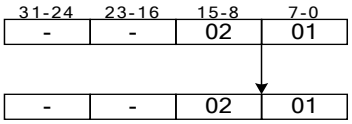
2) Hword swap ON, Byte swap OFF



3) Hword swap OFF, Byte swap ON



4) Hword swap OFF, Byte swap OFF

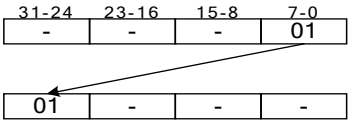


20.3.7.3. Byte access

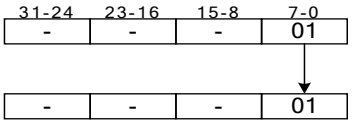
During Byte access, Data swap functionality determines byte swapping.

Byte access

1) Byte swap ON



2) Byte swap OFF



20.3.7.4. Target modules of Data swap

The target modules of a Data swap are as follows.

Data swap No	Target module
0	Reserved
1	Remote Handler (Master)
2	Remote Handler (Slave)
3	Host I/F
4	Sprite GDC
5	Display Controller
6	Display Peripherals (SIG, CLUT, Dithering, TCON)
7	RL Decompress
8	DMAC
9	Reserved
10	Reserved
11	Internal RAM1 I/F
12	Internal RAM2 I/F
13	Internal RAM3 I/F
14	Config FiFo (Master)
15	Config FiFo (Slave)
16	APB Peripherals
17	Reserved
18	Clocksynth
19	Reserved
20	CCNT
21	Command Sequencer
22	Memory I/F
23	Reserved
24	R-BUS Peripherals
25	Clomo

20.3.7.5. Limitations

Please determine the data swap functionality when the CMDSEQ (Command List) is initiated. Setting at any other time means that correct data swap functionality is not guaranteed.

20.3.8. Pin Multiplex

Switches the functionality of specific external pins of the GDC.

20.3.8.1. Pin switch

An external pin's functionality can be switched via the PIN MUX Register. The external switch table is shown below:

Register name	Bit	Function (Pin exchange)
PMM0[1:0]	2'b00	I/O buffer is input state
	2'b01	PWM select (VPWM)
	2'b1X	GPIO[19:16] select
PMM1[1:0]	2'b00	I/O buffer is input state
	2'b01	PWM select
	2'b1X	SPI1 select
PMM2[1:0]	2'b00	I/O buffer is input state
	2'b01	PWM select
	2'b1X	GPIO[15:8] select
PMM3[1:0]	2'b00	I/O buffer is input state
	2'b01	GPIO[7:0] select
	2'b1X	INT[7:0] select
PMM4[1:0]	2'b00	I/O buffer is input state
	2'b01	SPI select
	2'b1X	AD_AN[8] and SPI0_DI are multiplexed AD_AN[7] and SPI0_DO are multiplexed AD_AN[6] and SPI0_SCK are multiplexed

20.3.8.2. Limitations

Please determine the data swap functionality when the CMDSEQ (Command List) is initiated. Setting at any other time means that correct data swap functionality is not guaranteed.

20.4. Registers

20.4.1. Format of Register Descriptions

- Endian

  Only Little Endian access corresponds to the register of this module.

- Address

  “Address” shows the address of the register. (Base address + Offset address)

- Bit

  “Bit” shows the bit number of the register.

- Name

  “Name” shows the bit field name of the register.

- R/W

  “R/W” shows the attribute of Read/Write in each Bit field.
  R0: The Read value is always "0".
  R1: The Read value is always “1”.
  W0: The Write value is always "0", When “1” is written, it is disregarded.
  W1: The Write value is always “1”, When “0” is written, it is disregarded.
  R: Read
  W: Write

  Note : If a value is written to registers/bitfields that list R0, R1 and R in the following descriptions, then this value will not be changed in those registers/bitfields.

- Initial value

  “Initial value” is an initial value when reset is released.
  0: Initial value is "0".
  1: Initial value is "1".
  X: Undetermined

20.4.2. Global Address

For the module base address please refer to the chapter 3 Memory map.

20.4.3. Register summary

Address	Register Name	Description
Base+0x000	ChipInfo	Chip revision
Base+0x800	IntStatus	Status of interrupt
Base+0x804	InterruptReDetect	Interrupt re-detection control
Base+0x808	InterruptEnable	Output control of external interrupt
Base+0x810	ClockDiv	Clock HCLK, RCLK, PCLK division
Base+0x814	ClockEnable	Low power consumption control
Base+0x820	DataSwapCtr1	Hword/Byte swap control
Base+0x824	DataSwapCtr2	Hword/Byte swap control
Base+0x828	DataSwapCtr3	Hword/Byte swap control
Base+0x82C	DataSwapCtr4	Hword/Byte swap control
Base+0x830	SoftReset	Module reset control
Base+0x840	PinMUX	PIN select control
Base+0x850	OSC	CSC control
Base+0x860	Reserved	
Base+0x864	Reserved	

20.4.4. Register Description

20.4.4.1. ChipInfo Register

The Chip Information Register is used to check the version, production year and chip ID of this GDC.

Address	**Base Address** + 0x0000															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	YEAR[15:0]															
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CID[7:0]								VER[7:0]							
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit7-0 VER (VERsion)
Version information
Version is 0x00 (initial version)

Bit15-8 CID (Chip ID)
Chip ID information.
Chip ID is 0x01

Bit31-16 YEAR (YEAR)
Production year information
Production year is 2010

20.4.4.2. IntStatus Register

Status Register shows interrupt information. This register detects the rising edge of an interrupt signal from each module and stores this change. This register is cleared by writing “1”. This register is a logical sum of interrupt information and outputs to the external interrupt (ox_INT).

Address	**Base Address** + 0x800															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	INT31	INT30	INT29	INT28	INT27	INT26	INT25	INT24	INT23	INT22	INT21	INT20	INT19	INT18	INT17	INT16
R/W	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	INT15	INT14	INT13	INT12	INT11	INT10	INT9	INT8	INT7	INT6	INT5	INT4	INT3	INT2	INT1	INT0
R/W	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit0 INT0 (INTerrupt 0)
CMDSEQ module Interrupt
Clear by writing “1”.
0: No interrupt
1: Interrupt

Bit1 INT1 (INTerrupt 1)
CMDSEQ module watch dog timer error
Clear by writing “1”.
0: No interrupt
1: Interrupt

Bit2 INT2 (INTerrupt 2)
Reserved bit.
0: No interrupt
1: Interrupt

Bit3 INT3 (INTerrupt 3)
Display controller module VSYNC
Clear by writing “1”.
0: No interrupt
1: Interrupt

Bit4 INT4 (INTerrupt 4)
Display controller module HSYNC
Clear by writing “1”.
0: No interrupt
1: Interrupt

Bit5 INT5 (INTerrupt 5)
Display controller module external Vsync error
Clear by writing “1”.
0: No interrupt
1: Interrupt

Bit6 INT6 (INTerrupt 6)
Display controller module external Hsync error
Clear by writing “1”.
0: No interrupt
1: Interrupt

Bit7 INT7 (INTerrupt 7)
Display controller module pixel FIFO overflow
Clear by writing “1”.
0: No interrupt
1: Interrupt

Bit8 INT8 (INTerrupt 8)
Display controller module pixel FIFO underflow
Clear by writing "1".
0: No interrupt
1: Interrupt

Bit9 INT9 (INTerrupt 9)
Display controller module PWM demand
Clear by writing "1".
0: No interrupt
1: Interrupt

Bit10 INT10 (INTerrupt 10)
Sprite GDC module enable bits change
Clear by writing "1".
0: No interrupt
1: Interrupt

Bit11 INT11 (INTerrupt 11)
Sprite GDC module bus error
Clear by writing "1".
0: No interrupt
1: Interrupt

Bit12 INT12 (INTerrupt 12)
Sprite GDC module processing error
Clear by writing "1".
0: No interrupt
1: Interrupt

Bit13 INT13 (INTerrupt 13)
Sprite GDC module line blank
Clear by writing "1".
0: No interrupt
1: Interrupt

Bit14 INT14 (INTerrupt 14)
Sprite GDC module specified line processing over
Clear by writing "1".
0: No interrupt
1: Interrupt

Bit15 INT15 (INTerrupt 15)
SIG module Interrupt
Clear by writing "1".
0: No interrupt
1: Interrupt

Bit16 INT16 (INTerrupt 16)
MEM INF module Interrupt
Clear by writing "1".
0: No interrupt
1: Interrupt

Bit17 INT17 (INTerrupt 17)
MEM INF module Error
Clear by writing "1".
0: No interrupt
1: Interrupt

Bit18 INT18 (INTerrupt 18)
Reserved

Bit19 INT19 (INTerrupt 19)
RLD module Interrupt
Clear by writing “1”.
0: No interrupt
1: Interrupt

Bit20 INT20 (INTerrupt 20)
Config FiFo module Interrupt
Clear by writing “1”.
0: No interrupt
1: Interrupt

Bit21 INT21 (INTerrupt 21)
Host INF Hresp error
0: No interrupt
1: Interrupt

Bit22 INT22 (INTerrupt 22)
GPIO0(External Interrupt)
Clear by writing “1”.
0: No interrupt
1: Interrupt

Bit23 INT23 (INTerrupt 23)
GPIO1(External Interrupt)
Clear by writing “1”.
0: No interrupt
1: Interrupt

Bit24 INT24 (INTerrupt 24)
GPIO2(External Interrupt)
Clear by writing “1”.
0: No interrupt
1: Interrupt

Bit25 INT25 (INTerrupt 25)
GPIO3(External Interrupt)
Clear by writing “1”.
0: No interrupt
1: Interrupt

Bit26 INT26 (INTerrupt 26)
GPIO4(External Interrupt)
Clear by writing “1”.
0: No interrupt
1: Interrupt

Bit27 INT27 (INTerrupt 27)
Reserved

Bit28 INT28 (INTerrupt 28)
Reserved

Bit29 INT29 (INTerrupt 29)
Reserved

Bit30 INT30 (INTerrupt 30)
HDMAC module Ch0 interrupt
Clear by writing “1”.
0: No interrupt
1: Interrupt

Bit31 INT31 (INTerrupt 31)
HDMAC module Ch1 interrupt
Clear by writing “1”.
0: No interrupt
1: Interrupt

20.4.4.3. InterruptReDetect Register

The Interrupt re-detection register selects the interrupt detection mode after the Status information in Status Register has been cleared.

Address	**Base Address** +0x804															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RDET31	RDET30	RDET29	RDET28	RDET27	RDET26	RDET25	RDET24	RDET23	RDET22	RDET21	RDET20	RDET19	RDET18	RDET17	RDET16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RDET15	RDET14	RDET13	RDET12	RDET11	RDET10	RDET9	RDET8	RDET7	RDET6	RDET5	RDET4	RDET3	RDET2	RDET1	RDET0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit0 RDET0 (interrupt Re-DETection 0)
This bit sets the interrupt re-detection of Status Register bit0.
0: Re-detection
1: No re-detection

Bit1 RDET1 (interrupt Re-DETection 1)
This bit sets the interrupt re-detection of Status Register bit1.
0: Re-detection
1: No re-detection

Bit2 RDET2 (interrupt Re-DETection 2)
Reserved bit.
0: Re-detection
1: No re-detection

Bit3 RDET3 (interrupt Re-DETection 3)
This bit sets the interrupt re-detection of Status Register bit3.
0: Re-detection
1: No re-detection

Bit4 RDET4 (interrupt Re-DETection 4)
This bit sets the interrupt re-detection of Status Register bit4.
0: Re-detection
1: No re-detection

Bit5 RDET5 (interrupt Re-DETection 5)
This bit sets the interrupt re-detection of Status Register bit5.
0: Re-detection
1: No re-detection

Bit6 RDET6 (interrupt Re-DETection 6)
This bit sets the interrupt re-detection of Status Register bit6.
0: Re-detection
1: No re-detection

Bit7 RDET7 (interrupt Re-DETection 7)
This bit sets the interrupt re-detection of Status Register bit7.
0: Re-detection
1: No re-detection

Bit8 RDET8 (interrupt Re-DETection 8)
This bit sets the interrupt re-detection of Status Register bit8.
0: Re-detection
1: No re-detection

Bit9 RDET9 (interrupt Re-DETection 9)
This bit sets the interrupt re-detection of Status Register bit9.
0: Re-detection
1: No re-detection

Bit10 RDET10 (interrupt Re-DETection 10)
This bit sets the interrupt re-detection of Status Register bit10.
0: Re-detection
1: No re-detection

Bit11 RDET11 (interrupt Re-DETection 11)
This bit sets the interrupt re-detection of Status Register bit11.
0: Re-detection
1: No re-detection

Bit12 RDET12 (interrupt Re-DETection 12)
This bit sets the interrupt re-detection of Status Register bit12.
0: Re-detection
1: No re-detection

Bit13 RDET13 (interrupt Re-DETection 13)
This bit sets the interrupt re-detection of Status Register bit13.
0: Re-detection
1: No re-detection

Bit14 RDET14 (interrupt Re-DETection 14)
This bit sets the interrupt re-detection of Status Register bit14.
0: Re-detection
1: No re-detection

Bit15 RDET15 (interrupt Re-DETection 15)
This bit sets the interrupt re-detection of Status Register bit15.
0: Re-detection
1: No re-detection

Bit16 RDET16 (interrupt Re-DETection 16)
This bit sets the interrupt re-detection of Status Register bit16.
0: Re-detection
1: No re-detection

Bit17 RDET17 (interrupt Re-DETection 17)
This bit sets the interrupt re-detection of Status Register bit17.
0: Re-detection
1: No re-detection

Bit18 RDET18 (interrupt Re-DETection 18)
Reserved

Bit19 RDET19 (interrupt Re-DETection 19)
This bit sets the interrupt re-detection of Status Register bit19.
0: Re-detection
1: No re-detection

Bit20 RDET20 (interrupt Re-DETection 20)
This bit sets the interrupt re-detection of Status Register bit20.
0: Re-detection
1: No re-detection

Bit21 RDET21 (interrupt Re-DETection 21)
This bit sets the interrupt re-detection of Status Register bit21.
0: Re-detection
1: No re-detection

Bit22 RDET22 (interrupt Re-DETection 22)
This bit sets the interrupt re-detection of Status Register bit22.
0: Re-detection
1: No re-detection

Bit23 RDET23 (interrupt Re-DETection 23)
This bit sets the interrupt re-detection of Status Register bit23.
0: Re-detection
1: No re-detection

Bit24 RDET24 (interrupt Re-DETection 24)
This bit sets the interrupt re-detection of Status Register bit24.
0: Re-detection
1: No re-detection

Bit25 RDET25 (interrupt Re-DETection 25)
This bit sets the interrupt re-detection of Status Register bit25.
0: Re-detection
1: No re-detection

Bit26 RDET26 (interrupt Re-DETection 26)
This bit sets the interrupt re-detection of Status Register bit26.
0: Re-detection
1: No re-detection

Bit27 RDET27 (interrupt Re-DETection 27)
This bit sets the interrupt re-detection of Status Register bit27.
0: Re-detection
1: No re-detection

Bit28 RDET28 (interrupt Re-DETection 28)
This bit sets the interrupt re-detection of Status Register bit28.
0: Re-detection
1: No re-detection

Bit29 RDET29 (interrupt Re-DETection 29)
This bit sets the interrupt re-detection of Status Register bit29.
0: Re-detection
1: No re-detection

Bit30 RDET30 (interrupt Re-DETection 30)
This bit sets the interrupt re-detection of Status Register bit30.
0: Re-detection
1: No re-detection

Bit31 RDET31 (interrupt Re-DETection 31)
This bit sets the interrupt re-detection of Status Register bit31.
0: Re-detection
1: No re-detection

20.4.4.4. InterruptEnable Register

The Interrupt Enable Register controls the output to the external ox_INT of the Status Register information.

Address	**Base Address** +0x808															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IEN31	IEN30	IEN29	IEN28	IEN27	IEN26	IEN25	IEN24	IEN23	IEN22	IEN21	IEN20	IEN19	IEN18	IEN17	IEN16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IEN15	IEN14	IEN13	IEN12	IEN11	IEN10	IEN9	IEN8	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit0 IEN0 (Interrupt ENable 0)
This bit controls whether to output Interrupt0 information on Status Register to ox_INT.
0: Disable Not output to ox_INT
1: Enable output to ox_INT

Bit1 IEN1 (Interrupt ENable 1)
This bit controls whether to output Interrupt1 information on Status Register to ox_INT.
0: Disable Not output to ox_INT
1: Enable output to ox_INT

Bit2 IEN2 (Interrupt ENable 2)
Reserved bit.
0: Disable Not output to ox_INT
1: Enable output to ox_INT

Bit3 IEN3 (Interrupt ENable 3)
This bit controls whether to output Interrupt3 information on Status Register to ox_INT.
0: Disable Not output to ox_INT
1: Enable output to ox_INT

Bit4 IEN4 (Interrupt ENable 4)
This bit controls whether to output Interrupt4 information on Status Register to ox_INT.
0: Disable Not output to ox_INT
1: Enable output to ox_INT

Bit5 IEN5 (Interrupt ENable 5)
This bit controls whether to output Interrupt5 information on Status Register to ox_INT.
0: Disable Not output to ox_INT
1: Enable output to ox_INT

Bit6 IEN6 (Interrupt ENable 6)
This bit controls whether to output Interrupt6 information on Status Register to ox_INT.
0: Disable Not output to ox_INT
1: Enable output to ox_INT

Bit7 IEN7 (Interrupt ENable 7)
This bit controls whether to output Interrupt7 information on Status Register to ox_INT.
0: Disable Not output to ox_INT
1: Enable output to ox_INT

Bit8 IEN8 (Interrupt ENable 8)
This bit controls whether to output Interrupt8 information on Status Register to ox_INT.
0: Disable Not output to ox_INT
1: Enable output to ox_INT

Bit9 IEN9 (Interrupt ENable 9)
This bit controls whether to output Interrupt9 information on Status Register to ox_INT.
0: Disable Not output to ox_INT
1: Enable output to ox_INT

Bit10 IEN10 (Interrupt ENable 10)
This bit controls whether to output Interrupt10 information on Status Register to ox_INT.
0: Disable Not output to ox_INT
1: Enable output to ox_INT

Bit11 IEN11 (Interrupt ENable 11)
This bit controls whether to output Interrupt11 information on Status Register to ox_INT.
0: Disable Not output to ox_INT
1: Enable output to ox_INT

Bit12 IEN12 (Interrupt ENable 12)
This bit controls whether to output Interrupt12 information on Status Register to ox_INT.
0: Disable Not output to ox_INT
1: Enable output to ox_INT

Bit13 IEN13 (Interrupt ENable 13)
This bit controls whether to output Interrupt13 information on Status Register to ox_INT.
0: Disable Not output to ox_INT
1: Enable output to ox_INT

Bit14 IEN14 (Interrupt ENable 14)
This bit controls whether to output Interrupt14 information on Status Register to ox_INT.
0: Disable Not output to ox_INT
1: Enable output to ox_INT

Bit15 IEN15 (Interrupt ENable 15)
This bit controls whether to output Interrupt15 information on Status Register to ox_INT.
0: Disable Not output to ox_INT
1: Enable output to ox_INT

Bit16 IEN16 (Interrupt ENable 16)
This bit controls whether to output Interrupt16 information on Status Register to ox_INT.
0: Disable Not output to ox_INT
1: Enable output to ox_INT

Bit17 IEN17 (Interrupt ENable 17)
This bit controls whether to output Interrupt17 information on Status Register to ox_INT.
0: Disable Not output to ox_INT
1: Enable output to ox_INT

Bit18 Reserved

Bit19 IEN19 (Interrupt ENable 19)
This bit controls whether to output Interrupt19 information on Status Register to ox_INT.
0: Disable Not output to ox_INT
1: Enable output to ox_INT

Bit20 IEN20 (Interrupt ENable 20)
This bit controls whether to output Interrupt20 information on Status Register to ox_INT.
0: Disable Not output to ox_INT
1: Enable output to ox_INT

Bit21 IEN21 (Interrupt ENable 21)
This bit controls whether to output Interrupt21 information on Status Register to ox_INT.
0: Disable Not output to ox_INT
1: Enable output to ox_INT

Bit22 IEN22 (Interrupt ENable 22)
This bit controls whether to output Interrupt22 information on Status Register to ox_INT.
0: Disable Not output to ox_INT
1: Enable output to ox_INT

Bit23 IEN23 (Interrupt ENable 23)
This bit controls whether to output Interrupt23 information on Status Register to ox_INT.
0: Disable Not output to ox_INT
1: Enable output to ox_INT

Bit24 IEN24 (Interrupt ENable 24)
This bit controls whether to output Interrupt24 information on Status Register to ox_INT.
0: Disable Not output to ox_INT
1: Enable output to ox_INT

Bit25 IEN25 (Interrupt ENable 25)
This bit controls whether to output Interrupt25 information on Status Register to ox_INT.
0: Disable Not output to ox_INT
1: Enable output to ox_INT

Bit26 IEN26 (Interrupt ENable 26)
This bit controls whether to output Interrupt26 information on Status Register to ox_INT.
0: Disable Not output to ox_INT
1: Enable output to ox_INT

Bit27 IEN27 (Interrupt ENable 27)
This bit controls whether to output Interrupt27 information on Status Register to ox_INT.
0: Disable Not output to ox_INT
1: Enable output to ox_INT

Bit28 IEN28 (Interrupt ENable 28)
This bit controls whether to output Interrupt28 information on Status Register to ox_INT.
0: Disable Not output to ox_INT
1: Enable output to ox_INT

Bit29 IEN29 (Interrupt ENable 29)
This bit controls whether to output Interrupt29 information on Status Register to ox_INT.
0: Disable Not output to ox_INT
1: Enable output to ox_INT

Bit30 IEN30 (Interrupt ENable 30)
This bit controls whether to output Interrupt30 information on Status Register to ox_INT.
0: Disable Not output to ox_INT
1: Enable output to ox_INT

Bit31 IEN31 (Interrupt ENable 31)
This bit controls whether to output Interrupt31 information on Status Register to ox_INT.
0: Disable Not output to ox_INT
1: Enable output to ox_INT

20.4.4.5. ClockDiv Register

ClokeDiv (Clock Division) register sets the division ratio for the frequency of each CLK.

Address	**Base Address** + 0x810															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved														HCKDIV[1:0]	
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	RW	RW
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved						RCKDIV[1:0]		Reserved						PCKDIV[1:0]	
R/W	R0	R0	R0	R0	R0	R0	R/W	R/W	R0	R0	R0	R0	R0	R0	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit1-0 PCKDIV[1:0] (PClK DIVision [1:0])
Set the division ratio of PCLK (APB bus clock)
00: 2 PCLK = HCLK/2
01: 4 PCLK = HCLK/4
10: 8 PCLK = HCLK/8
11: 1 PCLK = HCLK *1

Bit9-8 RCKDIV[1:0] (RClK DIVision [1:0])
Set the division ratio of RCLK (local bus clock)
00: 2 RCLK = HCLK/2
01: 4 RCLK = HCLK/4
10: 8 RCLK = HCLK/8
11: 1 RCLK = HCLK *2

Bit17-16 HCKDIV[1:0] (HClK DIVision [1:0])
Set the division ratio of HCLK (AHB bus clock)
00: 1 83.33MHz
01: 2 41.66MHz
10: 4 20.83MHz
11: 1 83.33MHz

Note*1 : The limit of the APB BUS clock is 41.66MHz.
Therefore, please do not write 2'b11 to PCKDIV if you have set 2'b00 or 2'b11 in HCKDIV.

Note*2 : The limit of the Peripheral BUS clock is 41.66MHz.
Therefore, please do not write 2'b11 to RCKDIV if you set 2'b00 or 2'b11 in HCKDIV.

20.4.4.6. ClockEnable Register

Clock Enable Register controls the clock output for each module.

Address	**Base Address** + 0x814															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CKEN[31:16]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CKEN[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1

Bit0 CKEN[0] (ClocK ENable [0])
Clock0 (**Command Sequencer module**)
0: Disable Clock0 Stop
1: Enable Clock0 Output

Bit1 CKEN[1] (ClocK ENable [1])
Clock1 (**Sprite GDC module**)
0: Disable Clock1 Stop
1: Enable Clock1 Output

Bit2 CKEN[2] (ClocK ENable [2])
Clock2 (**Display Controller & Display Peripherals module**)
0: Disable Clock2 Stop
1: Enable Clock2 Output

Bit3 CKEN[3] (ClocK ENable [3])
Clock3 (**Reserved**)
0: Disable Clock3 Stop
1: Enable Clock3 Output

Bit4 CKEN[4] (ClocK ENable [4])
Clock4 (**DMAC module**)
0: Disable Clock4 Stop
1: Enable Clock4 Output

Bit5 CKEN[5] (ClocK ENable [5])
Clock5 (**RL Decompress module**)
0: Disable Clock5 Stop
1: Enable Clock5 Output

Bit6 Reserved

Bit7 CKEN[7] (ClocK ENable [7])
Clock7 (**MEM I/F module**)
0: Disable Clock7 Stop
1: Enable Clock7 Output

Bit8 CKEN[8] (ClocK ENable [8])
Clock8 (**INT RAM1 I/F module**)
0: Disable Clock8 Stop
1: Enable Clock8 Output

Bit9 CKEN[9] (ClocK ENable [9])
Clock9 (**INT RAM2 I/F module**)
0: Disable Clock9 Stop
1: Enable Clock9 Output

Bit10 CKEN[10] (ClocK ENable [10])
Clock10 (**INT RAM3 I/F module**)
0: Disable Clock10 Stop
1: Enable Clock10 Output

Bit11 CKEN[11] (ClocK ENable [11])
Clock11 (**Config FiFo module**)
0: Disable Clock11 Stop
1: Enable Clock11 Output

Bit12 CKEN[12] (ClocK ENable [12])
Clock12 (**R-BUS Peripherals**)
0: Disable Clock12 Stop
1: Enable Clock12 Output

Bit13 CKEN[13] (ClocK ENable [13])
Clock13 (**APB Peripherals**)
0: Disable Clock13 Stop
1: Enable Clock13 Output

Bit31:14 CKEN[31:14] (ClocK ENable [31:14])
Clock31-14 (**Reserved**)
0: Disable Clock31-14 Stop
1: Enable Clock31-14 Output

20.4.4.7. DataSwapCtr1 Register

The Data swap control 1 register controls the data swap functionality of each module.

Address	**Base Address** + 0x820															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RLD				Display Peripheral				Display				Sprite GDC			
	WSWP7	HSWP7[1:0]		BSWP7	WSWP6	HSWP6[1:0]		BSWP6	WSWP5	HSWP5[1:0]		BSWP5	WSWP4	HSWP4[1:0]		BSWP4
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Host I/F				Remote Handler (Slave)				Remote Handler (Master)				Dswap0			
	WSWP3	HSWP3[1:0]		BSWP3	WSWP2	HSWP2[1:0]		BSWP2	WSWP1	HSWP1[1:0]		BSWP1	WSWP0	HSWP0[1:0]		BSWP0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0

Bit0 BSWP 0(Byte SWaP0) Dswap0 (Reserved)
Dswap0 (Reserved) control.
0: Byte swap OFF
1: Byte swap ON

Bit2-1 HSWP0[1:0] (Hword SWaP0 [1:0]) Dswap0 (Reserved)
Dswap0 (Reserved) control.
00: Byte swap OFF , Hword swap OFF
01: Byte swap ON , Hword swap OFF
10: Byte swap OFF , Hword swap ON
11: Byte swap ON , Hword swap ON

Bit3 WSWP0 (Word SWaP 0) Dswap0 (Reserved)
Dswap0 (Reserved) control.
0: Byte swap OFF
1: Byte swap ON

Bit4 BSWP1 (Byte SWaP1) Remote Handler (Master) module
Remote Handler (Master) module control.
0: Byte swap OFF
1: Byte swap ON

Bit6-5 HSWP1[1:0] (Hword SWaP1 [1:0]) Remote Handler (Master) module
Remote Handler (Master) module control.
00: Byte swap OFF , Hword swap OFF
01: Byte swap ON , Hword swap OFF
10: Byte swap OFF , Hword swap ON
11: Byte swap ON , Hword swap ON

Bit7 WSWP1 (Word SWaP1) Remote Handler (Master) module
Remote Handler (Master) module control.
0: Byte swap OFF
1: Byte swap ON

Bit8 BSWP2 (Byte SWaP2) Remote Handler (Slave) module
Remote Handler (Slave) module control.
0: Byte swap OFF
1: Byte swap ON

Bit10-9 HSWP2[1:0] (Hword SWaP2 [1:0]) Remote Handler (Slave) module
Remote Handler (Slave) module control.
00: Byte swap OFF , Hword swap OFF
01: Byte swap ON , Hword swap OFF
10: Byte swap OFF , Hword swap ON
11: Byte swap ON , Hword swap ON

Bit11 WSWP2 (Word SWaP2) Remote Handler (Slave) module
Remote Handler (Slave) module control.
0: Byte swap OFF
1: Byte swap ON

Bit12 BSWP3 (Byte SWaP3) Host I/F module
Host I/F module control
0: Byte swap OFF
1: Byte swap ON

Bit14-13 HSWP3[1:0] (Hword SWaP3 [1:0]) Host I/F module
Host I/F module control
00: Byte swap OFF , Hword swap OFF
01: Byte swap ON , Hword swap OFF
10: Byte swap OFF , Hword swap ON
11: Byte swap ON , Hword swap ON

Bit15 WSWP3 (Word SWaP3) Host I/F module
Host I/F module control
0: Byte swap OFF
1: Byte swap ON

Bit16 BSWP4 (Byte SWaP4) Sprite GDC module
Sprite GDC module control
0: Byte swap OFF
1: Byte swap ON

Bit18-17 HSWP4[1:0] (Hword SWaP4 [1:0]) Sprite GDC module
Sprite GDC module control
00: Byte swap OFF , Hword swap OFF
01: Byte swap ON , Hword swap OFF
10: Byte swap OFF , Hword swap ON
11: Byte swap ON , Hword swap ON

Bit19 WSWP4 (Word SWaP4) C Sprite GDC module
Sprite GDC module control
0: Byte swap OFF
1: Byte swap ON

Bit20 BSWP5 (Byte SWaP5) Display Controller module
Display Controller module control
0: Byte swap OFF
1: Byte swap ON

Bit22-21 HSWP5[1:0] (Hword SWaP5 [1:0]) Display Controller module
Display Controller module control
00: Byte swap OFF , Hword swap OFF
01: Byte swap ON , Hword swap OFF
10: Byte swap OFF , Hword swap ON
11: Byte swap ON , Hword swap ON

Bit23 WSWP5 (Word SWaP5) Display Controller module
Display Controller module control
0: Byte swap OFF
1: Byte swap ON

Bit24 BSWP6 (Byte SWaP6) Display Peripherals
Display Peripherals control
0: Byte swap OFF
1: Byte swap ON

Bit26-25 HSWP6[1:0] (Hword SWaP6 [1:0]) Display Peripherals
Display Peripherals control
00: Byte swap OFF , Hword swap OFF
01: Byte swap ON , Hword swap OFF
10: Byte swap OFF , Hword swap ON
11: Byte swap ON , Hword swap ON

Bit27 WSWP6 (Word SWaP6) Display Peripherals
Display Peripherals control
0: Byte swap OFF
1: Byte swap ON

Bit28 BSWP7 (Byte SWaP7) RL Decompress module
RL Decompress module control
0: Byte swap OFF
1: Byte swap ON

Bit30-29 HSWP7[1:0] (Hword SWaP7 [1:0]) RL Decompress module
RL Decompress module control
00: Byte swap OFF , Hword swap OFF
01: Byte swap ON , Hword swap OFF
10: Byte swap OFF , Hword swap ON
11: Byte swap ON , Hword swap ON

Bit31 WSWP7 (Word SWaP7) RL Decompress module
RL Decompress module control
0: Byte swap OFF
1: Byte swap ON

20.4.4.8. DataSwapCtr2 Register

The Data swap control 2 register controls the data swap functionality of each module.

Address	**Base Address** + 0x824															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Config FiFo (Slave)				Config FiFo (Master)				INT RAM 3				INT RAM 2			
	WSWP15	HSWP15[1:0]		BSWP15	WSWP14	HSWP14[1:0]		BSWP14	WSWP13	HSWP13[1:0]		BSWP13	WSWP12	HSWP12[1:0]		BSWP12
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	INTRAM I.F 1				Dswap1				SPI Flash I/F				DMAC			
	WSWP11	HSWP11[1:0]		BSWP11	WSWP10	HSWP10[1:0]		BSWP10	WSWP9	HSWP9[1:0]		BSWP9	WSWP8	HSWP8[1:0]		BSWP8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit0 BSWP8 (Byte SWaP8) DMAC module
DMAC module control
0: Byte swap OFF
1: Byte swap ON

Bit2-1 HSWP8[1:0] (Hword SWaP8 [1:0]) DMAC module
DMAC module control
00: Byte swap OFF , Hword swap OFF
01: Byte swap ON , Hword swap OFF
10: Byte swap OFF , Hword swap ON
11: Byte swap ON , Hword swap ON

Bit3 WSWP8 (Word SWaP8) DMAC module
DMAC module control
0: Byte swap OFF
1: Byte swap ON

Bit4 BSWP9 (Byte SWaP9) SPI Flash I/F module
SPI Flash I/F module control
0: Byte swap OFF
1: Byte swap ON

Bit6-5 HSWP9[1:0] (Hword SWaP9 [1:0]) SPI Flash I/F module
SPI Flash I/F module control
00: Byte swap OFF , Hword swap OFF
01: Byte swap ON , Hword swap OFF
10: Byte swap OFF , Hword swap ON
11: Byte swap ON , Hword swap ON

Bit7 WSWP9 (Word SWaP9) SPI Flash I/F module
SPI Flash I/F module control
0: Byte swap OFF
1: Byte swap ON

Bit8 BSWP10 (Byte SWaP10) Dswap1 (Reserved)
Dswap1 (Reserved) control
0: Byte swap OFF
1: Byte swap ON

Bit10-9 HSWP10[1:0] (Hword SWaP10 [1:0]) Dswap1 (Reserved)
Dswap1 (Reserved) control
00: Byte swap OFF , Hword swap OFF
01: Byte swap ON , Hword swap OFF
10: Byte swap OFF , Hword swap ON
11: Byte swap ON , Hword swap ON

Bit11 WSWP10 (Word SWaP10) Dswap1 (Reserved)
Dswap1 (Reserved) control
0: Byte swap OFF
1: Byte swap ON

Bit12 BSWP11 (Byte SWaP11) Internal RAM1 I/F module
Internal RAM1 I/F module control
0: Byte swap OFF
1: Byte swap ON

Bit14-13 HSWP11[1:0] (Hword SWaP11 [1:0]) Internal RAM1 I/F module
Internal RAM1 I/F module control
00: Byte swap OFF , Hword swap OFF
01: Byte swap ON , Hword swap OFF
10: Byte swap OFF , Hword swap ON
11: Byte swap ON , Hword swap ON

Bit15 WSWP11 (Word SWaP11) Internal RAM1 I/F module
Internal RAM1 I/F module control
0: Byte swap OFF
1: Byte swap ON

Bit16 BSWP12 (Byte SWaP12) Internal RAM2 I/F module
Internal RAM2 I/F module control
0: Byte swap OFF
1: Byte swap ON

Bit18-17 HSWP12[1:0] (Hword SWaP12 [1:0]) Iternal RAM2 I/F module
Internal RAM2 I/F module control
00: Byte swap OFF , Hword swap OFF
01: Byte swap ON , Hword swap OFF
10: Byte swap OFF , Hword swap ON
11: Byte swap ON , Hword swap ON

Bit19 WSWP12 (Word SWaP12) Internal RAM2 I/F module
Internal RAM2 I/F module control
0: Byte swap OFF
1: Byte swap ON

Bit20 BSWP13 (Byte SWaP13) Internal RAM3 I/F module
Internal RAM3 I/F module control
0: Byte swap OFF
1: Byte swap ON

Bit22-21 HSWP13[1:0] (Hword SWaP13 [1:0]) Internal RAM3 I/F module
Internal RAM3 I/F module control
00: Byte swap OFF , Hword swap OFF
01: Byte swap ON , Hword swap OFF
10: Byte swap OFF , Hword swap ON
11: Byte swap ON , Hword swap ON

Bit23 WSWP13 (Word SWaP13) Internal RAM3 I/F module
Internal RAM3 I/F module control
0: Byte swap OFF
1: Byte swap ON

Bit24 BSWP14 (Byte SWaP14) Config FiFo (Master) module
Config FiFo (Master) module control
0: Byte swap OFF
1: Byte swap ON

Bit26-25 HSWP14[1:0] (Hword SWaP14 [1:0]) Config FiFo (Master) module
Config FiFo (Master) module control
00: Byte swap OFF , Hword swap OFF
01: Byte swap ON , Hword swap OFF
10: Byte swap OFF , Hword swap ON
11: Byte swap ON , Hword swap ON

Bit27 WSWP14 (Word SWaP14) Config FiFo (Master) module
Config FiFo (Master) module control
0: Byte swap OFF
1: Byte swap ON

Bit28 BSWP15 (Byte SWaP15) Config FiFo (Slave)
Config FiFo (Slave) control
0: Byte swap OFF
1: Byte swap ON

Bit30-29 HSWP15[1:0] (Hword SWaP15 [1:0]) Config FiFo (Slave)
Config FiFl (Slave) control
00: Byte swap OFF , Hword swap OFF
01: Byte swap ON , Hword swap OFF
10: Byte swap OFF , Hword swap ON
11: Byte swap ON , Hword swap ON

Bit31 WSWP15 (Word SWaP15) Config FiFo (Slave)
Config FiFo (Slave) control
0: Byte swap OFF
1: Byte swap ON

20.4.4.9. DataSwapCtr3 Register

The Data swap control 3 register controls the data swap functionality of each module.

Address	**Base Address** + 0x828															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Dswap4				Memory I/F				Command Sequencer				CCNT			
	WSWP23	HSWP23[1:0]		BSWP23	WSWP22	HSWP22[1:0]		BSWP22	WSWP21	HSWP21[1:0]		BSWP21	WSWP20	HSWP20[1:0]		BSWP20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Dswap3				Clock synth				Dswap2				APB Peripherals			
	WSWP19	HSWP19[1:0]		BSWP19	WSWP18	HSWP18[1:0]		BSWP18	WSWP17	HSWP17[1:0]		BSWP17	WSWP16	HSWP16[1:0]		BSWP16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit0 BSWP16 (Byte SWaP16) APB Peripherals
APB Peripherals control
0: Byte swap OFF
1: Byte swap ON

Bit2-1 HSWP16[1:0] (Hword SWaP16 [1:0]) APB Peripherals
APB Peripherals control
00: Byte swap OFF , Hword swap OFF
01: Byte swap ON , Hword swap OFF
10: Byte swap OFF , Hword swap ON
11: Byte swap ON , Hword swap ON

Bit3 WSWP16 (Word SWaP16) APB Peripherals
APB Peripherals control
0: Byte swap OFF
1: Byte swap ON

Bit4 BSWP17 (Byte SWaP17) Dswap2 (Reserved)
Dswap2(Reserved) control
0: Byte swap OFF
1: Byte swap ON

Bit6-5 HSWP17[1:0] (Hword SWaP17 [1:0]) Dswap2 (Reserved)
Dswap2 (Reserved) control
00: Byte swap OFF , Hword swap OFF
01: Byte swap ON , Hword swap OFF
10: Byte swap OFF , Hword swap ON
11: Byte swap ON , Hword swap ON

Bit7 WSWP17 (Word SWaP17) Dswap2 (Reserved)
Dswap2 (Reserved) control
0: Byte swap OFF
1: Byte swap ON

Bit8 BSWP18 (Byte SWaP18) Clocksynth module
Clocksynth module control
0: Byte swap OFF
1: Byte swap ON

Bit10-9 HSWP18[1:0] (Hword SWaP18 [1:0]) Clocksynth module
Clocksynth module control
00: Byte swap OFF , Hword swap OFF
01: Byte swap ON , Hword swap OFF
10: Byte swap OFF , Hword swap ON
11: Byte swap ON , Hword swap ON

Bit11 WSWP18 (Word SWaP18) Clocksynth module
Clocksynth module control
0: Byte swap OFF
1: Byte swap ON

Bit12 BSWP19 (Byte SWaP19) Dswap3 (Reserved)
Dswap3 (Reserved) control
0: Byte swap OFF
1: Byte swap ON

Bit14-13- HSWP19[1:0] (Hword SWaP19 [1:0]) Dswap3 (Reserved)
Dswap3 (Reserved) control
00: Byte swap OFF , Hword swap OFF
01: Byte swap ON , Hword swap OFF
10: Byte swap OFF , Hword swap ON
11: Byte swap ON , Hword swap ON

Bit15 WSWP19 (Word SWaP19) Dswap3 (Reserved)
Dswap3 (Reserved) control
0: Byte swap OFF
1: Byte swap ON

Bit16 BSWP20 (Byte SWaP20) CCNT module
CCNT module control
0: Byte swap OFF
1: Byte swap ON

Bit18-17 HSWP20[1:0] (Hword SWaP20 [1:0]) CCNT module
CCNT module control
00: Byte swap OFF , Hword swap OFF
01: Byte swap ON , Hword swap OFF
10: Byte swap OFF , Hword swap ON
11: Byte swap ON , Hword swap ON

Bit19 WSWP20 (Word SWaP20) CCNT module
CCNT module control
0: Byte swap OFF
1: Byte swap ON

Bit20 BSWP21 (Byte SWaP21) Command Sequencer module
Dswap1 (Reserved) control
0: Byte swap OFF
1: Byte swap ON

Bit22-21- HSWP21[1:0] (Hword SWaP21 [1:0]) Command Sequencer module
Dswap1 (Reserved) control
00: Byte swap OFF , Hword swap OFF
01: Byte swap ON , Hword swap OFF
10: Byte swap OFF , Hword swap ON
11: Byte swap ON , Hword swap ON

Bit23 WSWP21 (Word SWaP21) Command Sequencer module
Dswap1 (Reserved) control
0: Byte swap OFF
1: Byte swap ON

Bit24 BSWP22 (Byte SWaP22) Memory I/F module
TIC module control
0: Byte swap OFF
1: Byte swap ON

Bit26-25 HSWP22[1:0] (Hword SWaP22 [1:0]) Memory I/F module
TIC module control
00: Byte swap OFF , Hword swap OFF
01: Byte swap ON , Hword swap OFF
10: Byte swap OFF , Hword swap ON
11: Byte swap ON , Hword swap ON

Bit27 WSWP22 (Word SWaP22) Memory I/F module
TIC module control
0: Byte swap OFF
1: Byte swap ON

Bit28 BSWP23 (Byte SWaP23) Dswap4 (Reserved)
Dswap4 (Reserved) control
0: Byte swap OFF
1: Byte swap ON

Bit30-29 HSWP23[1:0] (Hword SWaP23 [1:0]) Dswap4 (Reserved)
Dswap4 (Reserved) control
00: Byte swap OFF , Hword swap OFF
01: Byte swap ON , Hword swap OFF
10: Byte swap OFF , Hword swap ON
11: Byte swap ON , Hword swap ON

Bit31 WSWP23 (Word SWaP23) Dswap4 (Reserved)
Dswap4 (Reserved) control
0: Byte swap OFF
1: Byte swap ON

20.4.4.10. DataSwapCtr4 Register

The Data swap control 4 register controls the data swap functionality of each module.

Address	**Base Address** + 0x82C															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved			Clomo					Reserved			R-BUS Peripherals				
				WSWP25[1:0]		HSWP25[1:0]		BSWP25				WSWP24[1:0]		HSWP24[1:0]		BSWP24
R/W	R0	R0	R0	R/W	R/W	R/W	R/W	R/W	R0	R0	R0	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit0 BSWP24 (Byte SWaP24) R-BUS Peripherals
R-BUS Peripherals control
0: Byte swap OFF
1: Byte swap ON

Bit2-1 HSWP24[1:0] (Hword SWaP24 [1:0]) R-BUS Peripherals
R-BUS Peripherals control
00: Byte swap OFF , Hword swap OFF
01: Byte swap ON , Hword swap OFF
10: Byte swap OFF , Hword swap ON
11: Byte swap ON , Hword swap ON

Bit4-3 WSWP24 (Word SWaP16) R-BUS Peripherals
R-BUS Peripherals control
00: Byte swap OFF , Hword swap OFF
01: Byte swap ON , Hword swap OFF
10: Byte swap OFF , Hword swap ON
11: Byte swap ON , Hword swap ON

Bit8 BSWP25 (Byte SWaP25) Clomo module
Clomo module control
0: Byte swap OFF
1: Byte swap ON

Bit10-9 HSWP25[1:0] (Hword SWaP25 [1:0]) Clomo module
Clomo module control
00: Byte swap OFF , Hword swap OFF
01: Byte swap ON , Hword swap OFF
10: Byte swap OFF , Hword swap ON
11: Byte swap ON , Hword swap ON

Bit12-11 WSWP25 (Word SWaP25) Clomo module
Clomo module control
00: Byte swap OFF , Hword swap OFF
01: Byte swap ON , Hword swap OFF
10: Byte swap OFF , Hword swap ON
11: Byte swap ON , Hword swap ON

20.4.4.11. SoftReset Register

The Soft Reset Register can be used to reset each module.

Address	**Base Address** + 0x830															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SRST[31:16]															
R/W	R0/W1	R0/W1	R0/W1	R0/W1	R0/W1	R0/W1	R0/W1	R0/W1	R0/W1	R0/W1	R0/W1	R0/W1	R0/W1	R0/W1	R0/W1	R0/W1
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SRST[15:0]															
R/W	R0/W1	R0/W1	R0/W1	R0/W1	R0/W1	R0/W1	R0/W1	R0/W1	R0/W1	R0/W1	R0/W1	R0/W1	R0/W1	R0/W1	R0/W1	R0/W1
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit0 SRST[0] (Soft ReSeT [0])
Writing "1" ox_SRST[0] (CMDSEQ module)
Reset output 1 cycles (HCLK).
1: Soft reset ox_SRST[0] (CMDSEQ module) reset output.

Bit1 SRST[1] (Soft ReSeT [1])
Writing "1" ox_SRST[1] (Remote Handler module)
Reset output 1 cycles (HCLK).
1: Soft reset ox_SRST[1] (Remote Handler module) reset output.

Bit2 SRST[2] (Soft ReSeT [2])
Writing "1" ox_SRST[2] (Host I/F module)
Reset output 1 cycles (HCLK).
1: Soft reset ox_SRST[2] (Host I/F module) reset output.

Bit3 SRST[3] (Soft ReSeT [3])
Writing "1" ox_SRST[3] (Sprite GDC module)
Reset output 1 cycles (HCLK).
1: Soft reset ox_SRST[3] (Sprite GDC module) reset output.

Bit4 SRST[4] (Soft ReSeT [4])
Writing "1" ox_SRST[4] (Display Controller module)
Reset output 1 cycles (HCLK).
1: Soft reset ox_SRST[4] (Display Controller module) reset output.

Bit5 SRST[5] (Soft ReSeT [5])
Writing "1" ox_SRST[5] (Display Peripherals)
Reset output 1 cycles (HCLK).
1: Soft reset ox_SRST[5] (Display Peripherals) reset output.

Bit6 SRST[6] (Soft ReSeT [6])
Writing "1" ox_SRST[6] (RLD module)
Reset output 1 cycles (HCLK).
1: Soft reset ox_SRST[6] (RLD module) reset output.

Bit7 SRST[7] (Soft ReSeT [7])
Writing "1" ox_SRST[7] (HDMAC module)
Reset output 1 cycles (HCLK).
1: Soft reset ox_SRST[7] (HDMAC module) reset output.

Bit8 SRST[8] (Soft ReSeT [8])
Writing "1" ox_SRST[8] (RLDSPI Flash I/F module)
Reset output 1 cycles (HCLK).
1: Soft reset ox_SRST[8] (SPI Flash I/F module) reset output.

Bit9 SRST[9] (Soft ReSeT [9])
Writing "1" ox_SRST[9] (MEM I/F module)
Reset output 1 cycles (HCLK).
1: Soft reset ox_SRST[9] (MEM I/F module) reset output.

Bit10 SRST[10] (Soft ReSeT [10])
Writing "1" ox_SRST[10] (INT RAM1 I/F module)
Reset output 1 cycles (HCLK).
1: Soft reset ox_SRST[10] (INT RAM1 I/F module) reset output.

Bit11 SRST[11] (Soft ReSeT [11])

Writing "1" ox_SRST[11] (INT RAM2 I/F module)
Reset output 1 cycles (HCLK).
1: Soft reset ox_SRST[11] (INT RAM2 I/F module) reset output.

Bit12 SRST[12] (Soft ReSeT [12])
Writing "1" ox_SRST[12] (INT RAM3 I/F module)
Reset output 1 cycles (HCLK).
1: Soft reset ox_SRST[12] (INT RAM3 I/F module) reset output.

Bit13 SRST[13] (Soft ReSeT [13])
Writing "1" ox_SRST[13] (Config FiFo module)
Reset output 1 cycles (HCLK).
1: Soft reset ox_SRST[13] (Config FiFo module) reset output.

Bit14 SRST[14] (Soft ReSeT [14])
Writing "1" ox_SRST[14] (R-BUS Peripherals)
Reset output 1 cycles (HCLK).
1: Soft reset ox_SRST[14] (R-BUS Peripherals) reset output.

Bit15 SRST[15] (Soft ReSeT [15])
Writing "1" ox_SRST[15] (APB Peripherals)
Reset output 1 cycles (HCLK).
1: Soft reset ox_SRST[15] (APB Peripherals) reset output.

Bit16 SRST[16] (Soft ReSeT [16])
Writing "1" ox_SRST[16] (AHB-1 BUS MATRIX)
Reset output 1 cycles (HCLK).
1: Soft reset ox_SRST[16] (AHB-1 BUS MATRIX) reset output.

Bit30-17 SRST[30:17] (Soft ReSeT [30:17])
Writing "1" ox_SRST[30:17] (Reserved)
Reset output 1 cycles (HCLK).
1: Soft reset ox_SRST[30:17] (Reserved) reset output.

Bit31 SRST[31] (Soft ReSeT [31])
Writing "1" ox_SRST[31] (All module)
Reset output 1 cycles (HCLK).
1: Soft reset ox_SRST[31] (All module) reset output.

20.4.4.12. PinMUX Register

The Pin MUX register is used to switch the functionality of multiplexed pins.

Address	**Base Address** + 0x840															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved						PMM4		PMM3		PMM2		PMM1		PMM0	
R/W	R0	R0	R0	R0	R0	R0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit1-0 PMM0[1:0] (Pin Mux Mode 0 [1:0]) - **Pin multiplex group #0**
Selects VPWM[3:0] or GPIO[19:16] functionality
00: Pin is in input state
01: VPWM[3:0] output
1X: GPIO[19:16] input/output

Bit3-2 PMM1[1:0] (Pin Mux Mode 1 [1:0]) - **Pin multiplex group #1**
Select PWM[11:8] or SPI1 functionality
00: Pin is in input state
01: PWM[11:8] output
1X: SPI1 input/output

Bit5-4 PMM2[1:0] (Pin Mux Mode 2 [1:0]) - **Pin multiplex group #2**
Selects PWM_O[7:4] or GPIO[15:12] functionality
00: Pin is in input state
01: PWM[7:4] output
10: GPIO[15:12] input/output
11: Pin is in input state

Bit7-6 PMM3[1:0] (Pin Mux Mode 3 [1:0]) - **Pin multiplex group #3**
Selects GPIO[4:0] or INT[4:0] or RLT[4:0] functionality
00: Pin is in input state
01: GPIO[4:0] input/output
10: INT[4:0] input
11: RLT[4:0] input

Bit9-8 PMM4[1:0] (Pin Mux Mode 4 [1:0]) - **Pin multiplex group #4**
Selects SPI0 or AD_AN[8:6] functionality
00: Pin is in input state
01: SPI0 input/output
1X: AD_AN[8:6] input

20.4.4.13. OSC Register

The OSC register controls the clock and power of the oscillator.

Address	**Base Address** + 0x850															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved					RFBEN	FCI	OSCR	OSCS[7:0]							
R/W	R0	R0	R0	R0	R0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	1

Bit7-0 OSCS[7:0] (OSCS [7:0])
OSCS[7:0] allows the power to be adjusted.
FF: Full Power
7F: Half Power
3F: Quarter Power etc

Bit8 OSCR (OSCR)
Oscillator Power Adjustment Enable.
OSCR=1 allows the power of the oscillator to be adjusted using the OSCS[7:0] control register.
0: Permanent full power
1: Enable power adjustment

Bit9 FCI (Fast Clock Input enable)
X0 can be used as an input to force in High speed clock for test.
0: Disable
1: Enable

Bit10 RFBEN (RFBEN)
Internal feedback Resistor Enable
0: Enabled
1: Disabled

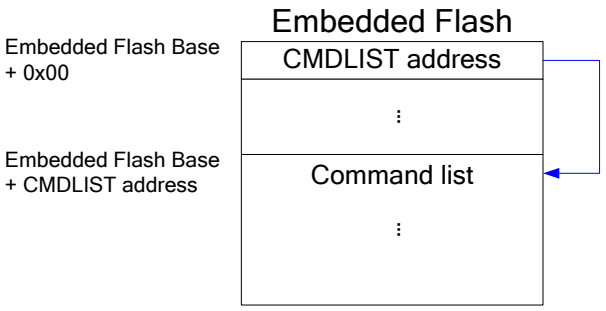
20.5. Application Note

20.5.1. Clock Control

20.5.1.1. Initial setting

The clock control is initialized according to the following procedures.

(1) Please prepare aand save a command list for the clock control of SETREG or OSETREG in the Embedded Flash Memory. The first address of the destination of the stored command list is written to the Embedded Flash Memory base address + 0x00. Then write the command list to the first address in this storage destination.



Please create the contents of the command list in the following order.

- Please set the clock dividing frequency via the Clock Division Register.
- Please enable the clock of the module used via the Clock Enable register. Please set Enable because the initial values of Clock Enable Register are all 'Disable'.
- Please include END at the end of the command list.

(2) After releasing Power On Reset, the CMDSEQ runs the boot process. The command list stored in the Embedded Flash Memory is then executed.

(3) The clock control is executed by CMDSEQ for CCNT.

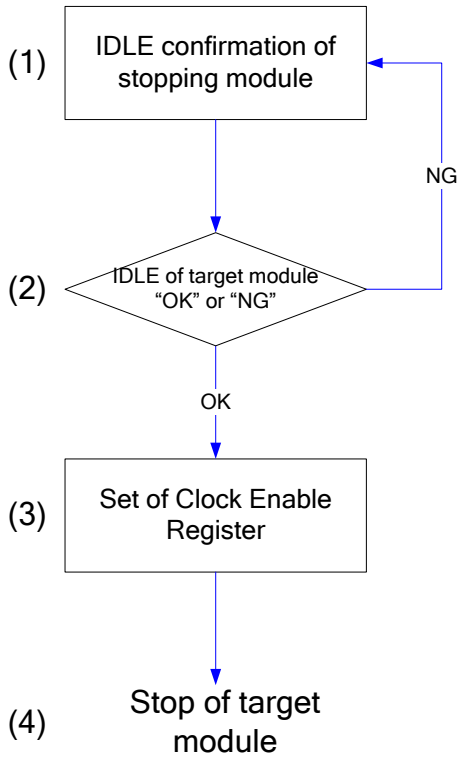
(4) The information set is output by CMDSEQ from CCNT to CRG.

(5) Clock control is executed by CRG for each module.

20.5.1.2. Stopping the clock during operation

Stopping a clock during operation can be done according to the following procedure:

(1) Please confirm that the state of a target module whose clock is to be stopped is 'IDLE'.
(2) Please proceed to (3) if the target module is in IDLE, otherwise confirm IDEL by (1) again.
(3) Please use the Clock Enable register setting to stop the clock of the target module.
(4) The clock supplied to the target module is stopped.



20.5.1.3. Soft Reset control

20.5.1.3.1. Soft Reset setting

The procedure of a soft reset is described below.

20.5.1.3.2. Module reset

(1) Please confirm the state of IDLE of the target module. (Please refer to the chapter of each module for the confirm method of IDLE.)
(2) Please set the target module using bits 0-30 of the Soft Reset register.
(3) Reset is issued to the target module.

20.5.1.3.3. All modules reset

There are three modes for am 'All modules' reset.

CCNT is directly controlled externally

(1) Please set the All modules reset (bit31) of the Soft Reset register.

Reset is issued to all modules (Clocksynth and clomo, CRG, CCNT, APIX PHY are excluded). Internal Flash Base

+ 0x04

(2)

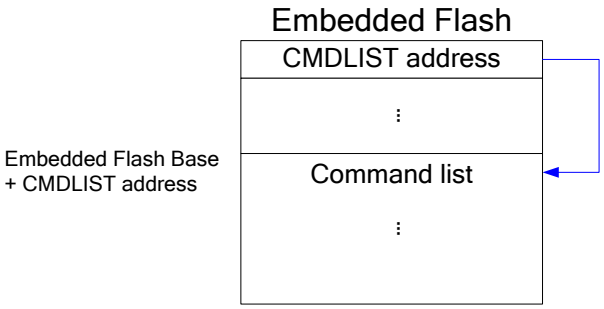
CCNT is controlled by Watch Dog Timer (WDT)

(1) Command List as shown below.

Example of an All modules soft reset by SETREG

31 24	23 16	15 8	7 0
SETREG (0x02)	COUNT (0x01)	Reserved	
Address (CCNT Base Address + 0x830)			
Data0 (0x8000_0000)			
END (0xFFFFFFFF)			

(2) Please write the command list to the Embedded Flash Memory. The first address of the destination of the stored command list is written to flash base address + 0x04. Then, please write the command list starting at he first address of this storage destination.

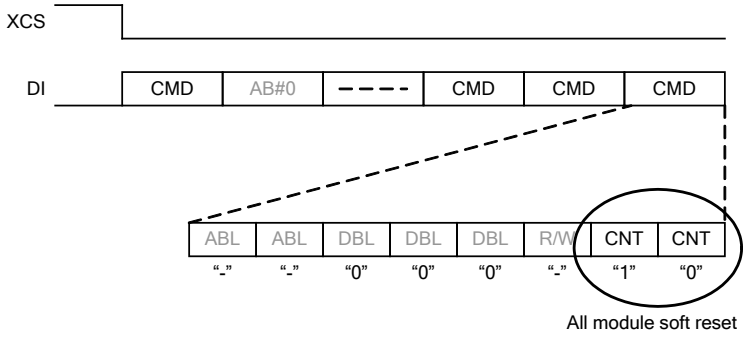


(3) WDT is set and started by the WDT bitfield of the CMDSEQ register.
(4) WDT timeout is monitored by CMDSEQ.

(5) After a WDT timeout has been detected by CMDSEQ, the command list is executed.
(6) The Soft Reset (bit31) of CCNT is executed by CMDSEQ.
(7) An All modules soft reset (ox_SRST31) is output by CCNT to CRG.
(8) Reset is output from CRG to all modules (excluding CRG and TIC).

It is controlled by the SPI command on the HOST interface

(1) Please execute an All modules soft reset via the external HOST connected to the HOST interface by an SPI command (CNT bit of the CMD header).

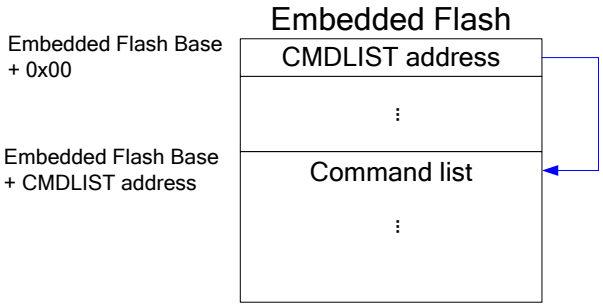


(2) Ox_ASRST is output from HOST inf to CCNT.
(3) All module soft reset (ox_SRST 31) is output from CCNT to CRG.
(4) Reset is output from CRG to all modules (CRG excluded).

20.5.1.4. Data swap control

Data swap control functionality is handled according to the following procedure.

(1) Please save a command list for Data swap in the Embedded Flash Memory with SETREG or OSETREG. The first destination address to save the command list is flash base address + 0x00. Then, please write the command list starting at the first address of this storage destination.



Please add END at the end of the command list.

(2) After releasing Power On Reset, CMDSEQ runs the boot process. The command list saved in the Embedded Flash Memory is then executed.

(3) The Data swap control is executed by CMDSEQ for CCNT.

(4) The Data swap control is executed by CCNT for each module.

Note:

In order to handle endianness issues, the 'Indigo-L' hardware contains a set of data swappers, and a few address inverters (which are not documented).

The initial setup of data swappers and address inverters at 'Indigo-L' power-up does not match all requirements for operation. Access to halfword and byte registers by Host Interface (SPI), Command Sequencer or Configuration FIFO does not work as expected, not does sub-register access to little-endian modules by the Remote Handler.

The following registers need to be set, in this order, to set up the system properly:

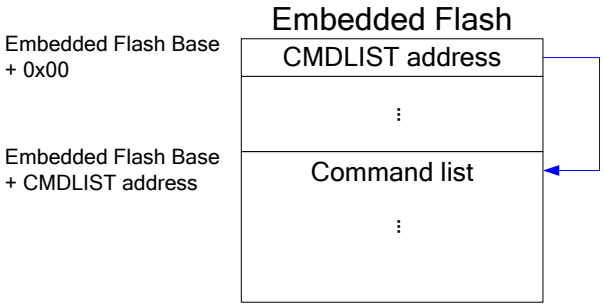
Register Address Value
CCNT.ClockEnable 0x00010814 0xFFFFFFFF // enable all clocks
CCNT.DataSwapCtr1 0x00010820 0x00000000 // set up swappers
CCNT.DataSwapCtr2 0x00010824 0x00000000
CCNT.DataSwapCtr3 0x00010828 0x00000000
CCNT.DataSwapCtr4 0x0001082C 0x00000505
CCNT.0xE00 0x00010E00 0x5E5F5E76 // set up inverters
CCNT.0xE30 0x00010E30 0x00000000
CCNT.0xE00 0x00010E00 0x00000000

For details, please see the " INDIGO BYTE ORDER APPLICATION NOTE".

20.5.1.5. Pin Multiplex control

The Pin Multiplex control is handled according to the following procedure.

(1) Please save a command list for pin multiplex control in the Embedded Flash Memory with SETREG or OSETREG. The first destination address to save the command list is flash base address + 0x00. Then, please write the command list starting at the first address of this storage destination.



Please add END at the end of the command list.

(2) After releasing Power On Reset, CMDSEQ runs the boot process. The command list saved in the Embedded Flash Memory is then executed.

(3) The Pin Multiplex control is executed by CMDSEQ for CCNT.

(4) The Pin Multiplex control is executed by CCNT.

20.5.1.6. Chip initialization Example

An example of chip initialization is shown below.

(1) A command list is created as shown below.

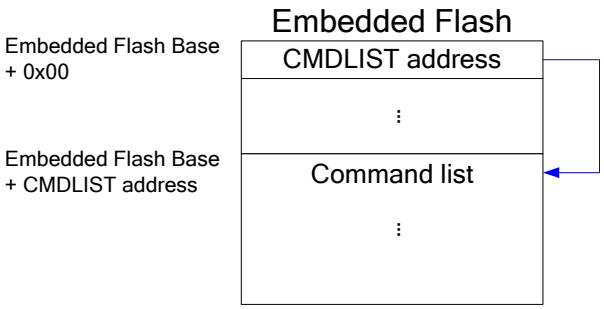
Example: Pin Multiplex, Data Swap, Interrupt Enable, and Clockl Control are done via OSETREG.

31 24	23 16	15 8	7 0
OSETREG (0x03h)	COUNT (0x06h)	Reserved	Size (0x01h)
Base Address (CCNT Base Address)			Reserved
Offset Addr0 (0x840h)	Reserved	Data0 (0x0299h)	
Offset Addr1 (0x820h)	Reserved	Data1 (0x0080h)	
Offset Addr2 (0x826h)	Reserved	Data2 (0x8000h)	
Offset Addr3 (0x808h)	Reserved	Data3 (0xFFFFh)	
Offset Addr3 (0x80Ah)	Reserved	Data4 (0x7FFFh)	
Offset Addr5 (0x814h)	Reserved	Data5 (0x3FFFh)	
END (0xFFFFFFFFh)			

Please note the following points when you do chip initialization:

- Please describe the Clock enable control at the end of the command list.

(2) Please save the command list in the Embedded Flash Memory. The first destination address to save the command list is flash base address + 0x00. Then, please write the command list starting at the first address of this storage destination.



(5) After releasing Power On Reset, CMDSEQ runs the boot process. The command list saved in the Embedded Flash Memory is then executed.

(6) The chip inizialzation is executed by CMDSEQ for CCNT.

(7) The chip inizialzation is executed by CCNT.

21. Configuration FIFO

This chapter describes the configuration FIFO of the MB88F333.

21.1. Outline

The configuration FIFO can be used to decouple a MCU command-stream from a peripheral command stream. This is e.g. necessary to allow isochronous reconfiguration of special peripherals like stepper motor controllers if the command stream from MCU is not “jitter-free”, that means communication is disturbed and requires repeated transmissions.

For each of the 8 FIFOs a trigger input is provided. Each trigger signal starts a transfer from the respective FIFO to the programmed destination address. If multiple FIFOs are triggered at the same time. Each trigger request is serviced by a fixed priority scheme. Lower FIFO numbers have higher priority and are serviced first.

21.2. Features

The Configuration FIFO has following features.

21.2.1. Features

- 8 configurable FIFOs
- Usage of one shared memory
- Depth of each FIFO is configurable
- FIFO upper and lower threshold interrupts
- AHB slave interface for FIFO data input, and the register is write/read.
- AHB-master interface for FIFO data output (only write is supported.)
- Trigger input for each FIFO output
- Simple local DMA functionality at data output
  programmable target address
  different addressing modes (including fixed)
- Memory mapped access to 2KByte SRAM
- Interrupt is cleared by ISTS (Interrupt Status register) writing, or clear signal from external module (Remote Handler).

21.2.2. Limitations

None

21.3. Function

21.3.1. Block diagram

Figure 21-1 shows Configuration FIFO's block diagram.

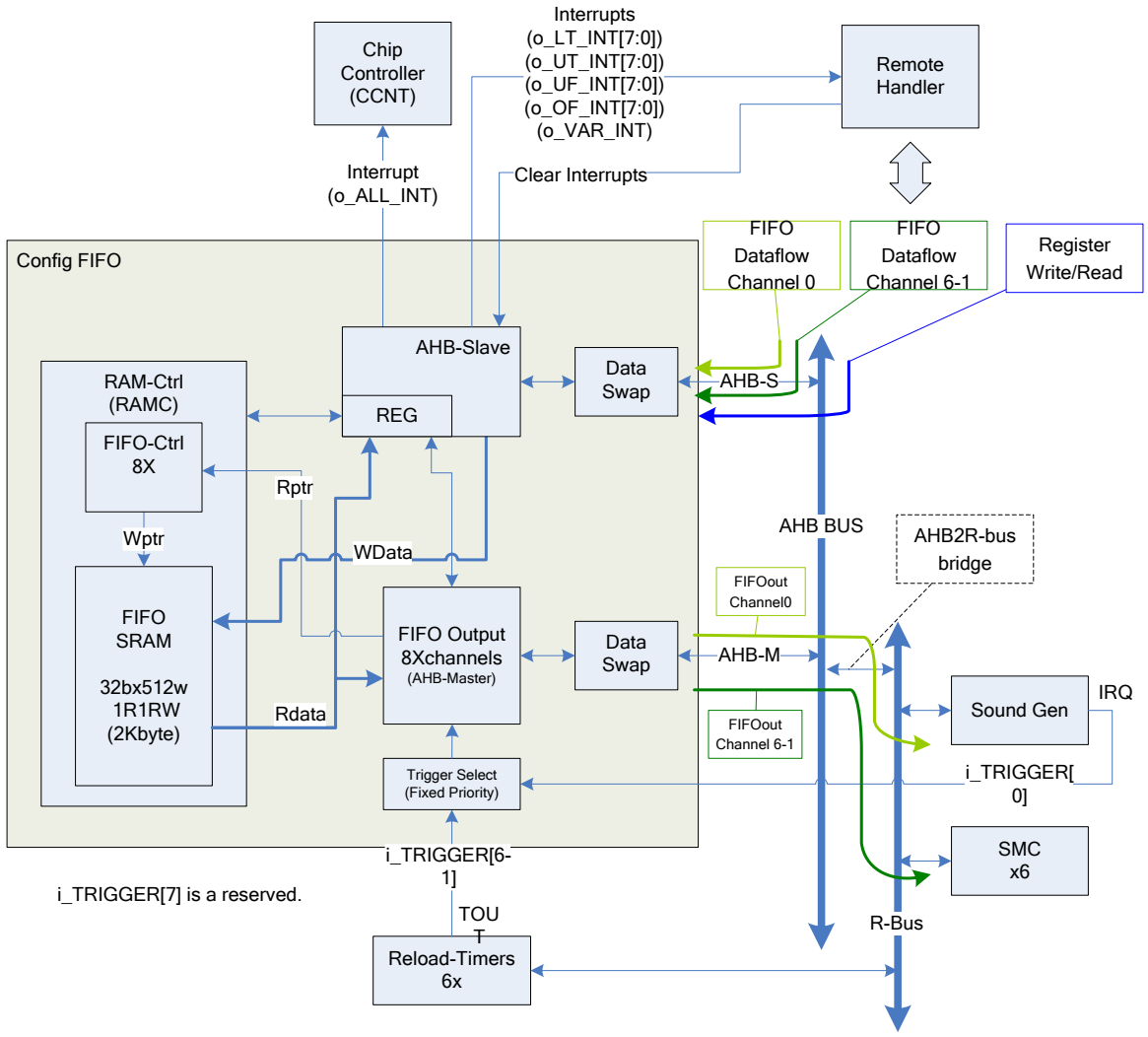


Figure 21-1 Configuration FIFO block

21.3.2. AHB Slave Interface

The AHB slave interface is used to write and read registers and to input FIFO data.
Registers exist for 8 channels.
If the FIFO is full, write data transfers are not executed. However, HREADY is asserted.
It is necessary to monitor this situation via an interrupt signal.

21.3.3. AHB Master Interface

Data is output from a FIFO when this is requested from a channel via a trigger request.
In the case of FFEmptyMode=0, and if the FIFO is emptied, the FFEnO bit of a FFCfg register will be automatically set to 0. If a FIFO is emptied, it is not possible to write a 1 to the FFEnO bit. The application software must first write data to the FIFO and must then to set FFEnO back to 1.

If the number of transfers (TransferNumber of a TransferCfg register) is over the FILL Level (FillLevel of a FFStatus register), an AHB master will not transfer, even if a trigger request signal is initiated.
In the following example, if the left side of the equation is too large, an AHB master does not transfer.
(((TransferNumber-1)+1) << transferwidth) > FILL Level
TransferNumber=0 is 64 transfers.

In the case of FFEmptyMode=0, if a trigger request is received and the FILL Level is the same as the number transfers set, then FFEnO is automatically set to 0 after the transfer has completed.

Burst mode for the AHB Master transfer is automatically set depending on the number of transfers.
- In the case of TransferINCR=0 of TransferCfg
o_mHBURST is always transferred using SINGLE regardless of the number of transfers.

- In the case of TransferINCR=1 of TransferCfg
When TransferNumber is 1, o_mHBURST becomes SINGLE.
When TransferNumber is 4, o_mHBURST becomes INCR4.
When TransferNumber is 8, o_mHBURST becomes INCR8.
When TransferNumber is 16, o_mHBURST becomes INCR16.

As for other values, o_mHBURST becomes INCR (Indefinite length burst).
When a transfer address exceeds 1KB, transfer is not executed using INCR4 and INCR8 and INCR16. It is transferred using INCR.

21.3.4. FIFO Memory

The FIFO memory is used by the AHB slave interface for FIFO data input. If 2 data words (8 bytes) from the FFDataInL and FFDataInU registers are received, these are written to the FIFO. Writes to the FIFO are in 2 data word units. The internal FIFO memory is only updated if both registers are written to (else the FIFO memory remains unchanged).
This mode is referred to as the “temporary latch feature” and is very effective if the FFTempMode bit of a FFCfg register is 1.
In the case of FFTempMode=0 when 1 data block (byte, hword, word) is received, this is written to the FIFO. A receiving address writes data to the address held in FFDataInL. In addition, in the case of FFTempMode=0, the application software needs to check the setting of TransferSize in the corresponding TransferCfg register, and it is necessary to control flow so that the write and read sizes are the same.
In both cases, if the FIFO is full, receive data is not written to the FIFO.

In the case of channel 0, it is necessary to write the application software in such a way that it can write to the following addresses:

FFTempMode=0

Input Size is BYTE	140h receive -> FIFO Write! -> 140h receive -> FIFO Write! -> … (Data bit 7 – 0 is used)
Input Size is HWORD	140h receive -> FIFO Write! -> 140h receive -> FIFO Write! -> … (Data bit 15 - 0 is used)
Input Size is WORD	140h receive -> FIFO Write! -> 140h receive -> FIFO Write! -> … (Data bit 31 - 0 is used)

FFTempMode=1 (temporary latch feature mode)

Input Size is BYTE	140h -> 141h -> 142h -> … 146h -> 147h -> FIFO Write! -> 140h …
Input Size is HWORD	140h -> 142h -> 144h -> 146h -> FIFO Write! -> 140h …
Input Size is WORD	140h -> 144h -> FIFO Write! -> 140h -> 144h -> FIFO Write! -> 140h …

AHB-master interface for FIFO data output
The shared memory (fixed 2KB) is used by up to 8 channels (programmable)
The FIFO area of each channel can be arbitrarily changed.
The area is set with LowerBoundAdr and UpperBoundAdr register of FFB.
An area is specified using a double word address.
Do not overlap any channel boundary areas in the settings.
If areas overlap, FIFO operation may malfunction.
The maximum area for the channels is 2KB.
If used for one channel, the other channels can not be used.
A wraparound address is not permitted (e.g. UpperBoundAdr=0x010, LowerBoundAdr=0xf0).

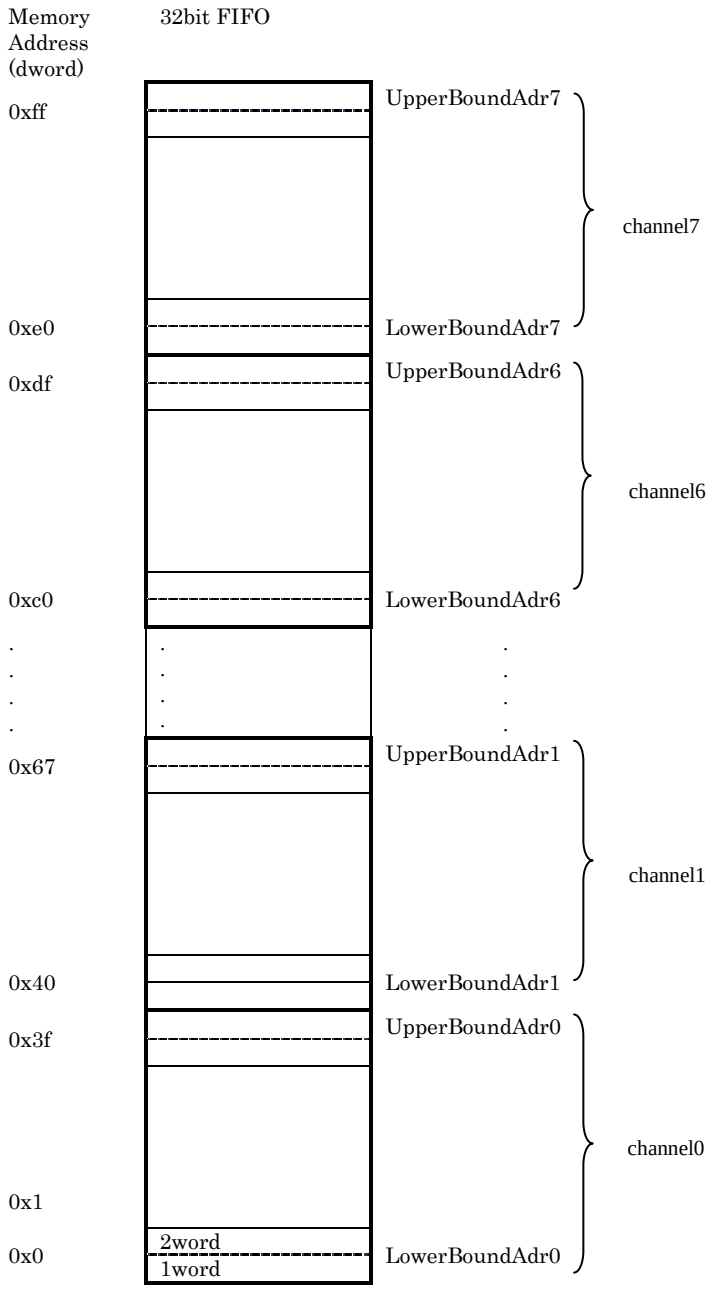


Figure 21-2 FIFO Address setting example

21.3.5. Trigger Request

A trigger signal uses positive edge detection. If a trigger signal is a toggle output, the timer preset value is halved and it is necessary to use it as a positive edge output.
Each trigger request is serviced by a fixed priority scheme.
The signal connected with Ch0 (i_TRIGGER[0]) has higher priority by default (initial value).
The signal connected with Ch7 (i_TRIGGER[7]) has lowest priority.
If two or more trigger requests occur simultaneously, the channel with the highest priority will be processed.
A trigger's priority can be set and changed via the CHPriority register.
The value of a CHPriority register becomes a channel of the highest priority.
CHPriority Register= 010b : High Priority Ch2 – Ch3 – Ch4 – Ch5 – Ch6 – Ch7 – Ch0 – Ch1 Low Priority.

Notes

i_Trigger[0] is connected to IRQ of SoundGen.
ConfigFIFO detects a trigger by the rising edge of i_Trigger.
However, the IRQ of SoundGen continues to stay at a H level until a 0 is written to the INT register.
In this case, ConfigFIFO cannot detect discontinuation again.
It is necessary to write 0 to the INT register of SoundGen.
Possible ways to write 0 in an INT register are shown below:

- INT register is written in 0 by SW.
- An automatic HW clear is used by Remote Handler.
- A command which writes 0 in an INT register is written in the FIFO data of ConfigFIFO.

21.3.6. Interrupt

The interrupt mechanism has the following output signals:

The various interrupt signal output (o_VAR_INT) active high by the following factors:

- AHB master received HRESP error.

The Lower Threshold level interrupt signal output (o_LT_INT[7:0]) active high by the following factors:

- FIFO (Ch7-0) fill level lower threshold below.

The Upper Threshold level interrupt signal output (o_UT_INT[7:0]) active high by the following factors:

- FIFO (Ch7-0) fill level upper threshold over.

The Underflow interrupt signal output (o_UF_INT[7:0]) active high by the following factors:

- FIFO (Ch7-0) Underflow (empty).

The Overflow interrupt signal output (o_OF_INT[7:0]) active high by the following factors:

- FIFO (Ch7-0) Overflow (full).

The above items are interrupts to the Remote Handler.
The following items are interrupts to CCNT.

The collective output of all the interrupt signals combined (o_ALL_INT).

- Each Channel (Ch0-7) HRESP error + Lower Threshold + Upper Threshold + Underflow + Overflow.

If the interrupt signal goes (active) high, it signals this using the interrupt status register (FFISTS,FFISTS_LT7-0,FFISTS_UT7-0).
The interrupt signal is cleared by writing 1 to the interrupt status register (FFISTS,FFISTS_LT7-0,FFISTS_UT7-0) or by receiving a clear signal from an external module (automatic clear).

An interrupt factor is detected using the edge of an i_TRIGGER signal (except o_VAR_INT).

Example case (o_VAR_INT):
- AHB master of ConfigFIFO received HRESP=Error., 1 interrupt (o_VAR_INT) is generated.

Example case (o_LT_INT):
- When the state is "fill level < lower threshold", interrupt (o_LT_INT) is NOT generated.
- i_TRIGGER signal High asserted, 1 interrupt (o_LT_INT) is generated.

Interrupt Enable register (FFIEN,FFIEN_LT7-0,FFIEN_UT7-0) controls the interrupt output signal.
The interrupt status register can not be controlled by the interrupt enable register.
If the interrupt factor has acted, the value read from the interrupt status register (FFITST,FFISTS_LT7-0,FFISTS_UT7-0) will be 1, even if it is disabled in the interrupt enable register (FFIEN,FFIEN_LT7-0,FFIEN_UT7-0).

21.3.7. Data Swap

Please refer to 21.3.7.

21.4. Register

21.4.1. Format of Register Description

The register descriptions in the following sections use the format shown below to describe each bit field of a register.

Register address	Offset																														
Bit number	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
Field name																																
R/W																																
Reset value																																

Meaning of items and sign

Register address

Register address shows the address (Offset address) of the register.

Bit number

Bit number shows bit position of the register.

Field name

Field name shows bit name of the register.

R/W

R/W shows the read/write attribute of each bit field:

- R: Read
- W: Write
- R0: The read value is always "0".
- R1: The read value is always "1".
- W0: The write value is always "0".When “1” is written, it is disregarded.
- W1: The write value is always "1". When “0” is written, it is disregarded.

Reset value

Reset value indicates the value of each bit field immediately after reset.

- 0: Initial value is "0".
- 1: Initial value is "1".
- X: Undefined.

Unused register fields are marked with a solid grey background.
Bit vectors are unsigned integers, if nothing else is specified.

21.4.2. Global Address

For the module base address please refer to the chapter 3 Memory map.

21.4.3. Register summary

Address	Register Name	Channel	Description
Base address + 000H	FFISTS	Common	Interrupt status flags, a '1' signifies that the corresponding interrupt condition occurred (even if interrupt is disabled), write '1' clears the flag,
Base address + 004H	FFIEN		Interrupt Enable register
			Reserved
Base address + 010H	CFIFO_IDLE		A summary of the state of all the channels is shown.
Base address + 014H	CHPriority		Priority of Trigger Request is changed.
			Reserved
Base address + 100H	SWReset0	Channel0	SW reset
Base address + 104H	FFCfg0		FIFO Configuration
Base address + 108H	FFB0		FIFO Boundary Address
Base address + 10CH	FFT0		FIFO Threshold Level
Base address + 110H	DestAddress0		Local AHB-master transfer Destination address
Base address + 114H	AdrCfg0		Address generation Configuration
Base address + 118H	TransferCfg0		Local AHB-master transfer Configuration
Base address + 11CH	FFStatus0		Status register
Base address + 120H	FFISTS_LT0		Interrupt Lower Threshold level status flags, a '1' signifies that the corresponding interrupt condition occurred (even if interrupt is disabled), write '1' clears the flag,
Base address + 124H	FFIEN_LT0		Interrupt Lower Threshold level Enable register
Base address + 128H	FFISTS_UT0		Interrupt Upper Threshold level status flags, a '1' signifies that the corresponding interrupt condition occurred (even if interrupt is disabled), write '1' clears the flag,
Base address + 12CH	FFIEN_UT0		Interrupt Upper Threshold level Enable register
			Reserved
Base address + 140H	FFDataInL0		FIFO Data Lower In
Base address + 144H	FFDataInU0		FIFO Data Upper In
			Reserved
Base address + 180H	SWReset1	Channel1	SW reset
Base address + 184H	FFCfg1		FIFO Configuration
Base address + 188H	FFB1		FIFO Boundary Address
Base address + 18CH	FFT1		FIFO Threshold Level
Base address + 190H	DestAddress1		Local AHB-master transfer Destination address
Base address + 194H	AdrCfg1		Address generation Configuration
Base address + 198H	TransferCfg1		Local AHB-master transfer Configuration
Base address + 19CH	FFStatus1		Status register
Base address + 1A0H	FFISTS_LT1		Interrupt Lower Threshold level status flags, a '1' signifies that the corresponding interrupt condition occurred (even if interrupt is disabled), write '1' clears the flag,
Base address + 1A4H	FFIEN_LT1		Interrupt Lower Threshold level Enable register
Base address + 1A8H	FFISTS_UT1		Interrupt Upper Threshold level status flags, a '1' signifies that the corresponding interrupt condition occurred (even if interrupt is disabled), write '1' clears the flag,
Base address + 1ACH	FFIEN_UT1		Interrupt Upper Threshold level Enable register
			Reserved
Base address + 1C0H	FFDataInL1		FIFO Data Lower In
Base address + 1C4H	FFDataInU1		FIFO Data Upper In
			Reserved
Base address + 200H	SWReset2	Channel2	SW reset
Base address + 204H	FFCfg2		FIFO Configuration
Base address + 208H	FFB2		FIFO Boundary Address
Base address + 20CH	FFT2		FIFO Threshold Level
Base address + 210H	DestAddress2		Local AHB-master transfer Destination address
Base address + 214H	AdrCfg2		Address generation Configuration
Base address + 218H	TransferCfg2		Local AHB-master transfer Configuration
Base address + 21CH	FFStatus2		Status register
Base address + 220H	FFISTS_LT2		Interrupt Lower Threshold level status flags, a '1' signifies that the corresponding interrupt condition occurred (even if interrupt is disabled), write '1' clears the flag,
Base address + 224H	FFIEN_LT2		Interrupt Lower Threshold level Enable register
Base address + 228H	FFISTS_UT2		Interrupt Upper Threshold level status flags, a '1' signifies that the corresponding interrupt condition occurred (even if interrupt is disabled), write '1' clears the flag,
Base address + 22CH	FFIEN_UT2		Interrupt Upper Threshold level Enable register
			Reserved
Base address + 240H	FFDataInL2		FIFO Data Lower In

Address	Register Name	Channel	Description
Base address + 244H	FFDataInU2		FIFO Data Upper In
			Reserved
Base address + 280H	SWReset3	Channel3	SW reset
Base address + 284H	FFCfg3		FIFO Configuration
Base address + 288H	FFB3		FIFO Boundary Address
Base address + 28CH	FFT3		FIFO Threshold Level
Base address + 290H	DestAddress3		Local AHB-master transfer Destination address
Base address + 294H	AdrCfg3		Address generation Configuration
Base address + 298H	TransferCfg3		Local AHB-master transfer Configuration
Base address + 29CH	FFStatus3		Status register
Base address + 2A0H	FFISTS_LT3		Interrupt Lower Threshold level status flags, a '1' signifies that the corresponding interrupt condition occurred (even if interrupt is disabled), write '1' clears the flag,
Base address + 2A4H	FFIEN_LT3		Interrupt Lower Threshold level Enable register
Base address + 2A8H	FFISTS_UT3		Interrupt Upper Threshold level status flags, a '1' signifies that the corresponding interrupt condition occurred (even if interrupt is disabled), write '1' clears the flag,
Base address + 2ACH	FFIEN_UT3		Interrupt Upper Threshold level Enable register
			Reserved
Base address + 2C0H	FFDataInL3		FIFO Data Lower In
Base address + 2C4H	FFDataInU3		FIFO Data Upper In
			Reserved
Base address + 300H	SWReset4	Channel4	SW reset
Base address + 304H	FFCfg4		FIFO Configuration
Base address + 308H	FFB4		FIFO Boundary Address
Base address + 30CH	FFT4		FIFO Threshold Level
Base address + 310H	DestAddress4		Local AHB-master transfer Destination address
Base address + 314H	AdrCfg4		Address generation Configuration
Base address + 318H	TransferCfg4		Local AHB-master transfer Configuration
Base address + 31CH	FFStatus4		Status register
Base address + 320H	FFISTS_LT4		Interrupt Lower Threshold level status flags, a '1' signifies that the corresponding interrupt condition occurred (even if interrupt is disabled), write '1' clears the flag,
Base address + 324H	FFIEN_LT4		Interrupt Lower Threshold level Enable register
Base address + 328H	FFISTS_UT4		Interrupt Upper Threshold level status flags, a '1' signifies that the corresponding interrupt condition occurred (even if interrupt is disabled), write '1' clears the flag,
Base address + 32CH	FFIEN_UT4		Interrupt Upper Threshold level Enable register
			Reserved
Base address + 340H	FFDataInL4		FIFO Data Lower In
Base address + 344H	FFDataInU4		FIFO Data Upper In
			Reserved
Base address + 380H	SWReset5	Channel5	SW reset
Base address + 384H	FFCfg5		FIFO Configuration
Base address + 388H	FFB5		FIFO Boundary Address
Base address + 38CH	FFT5		FIFO Threshold Level
Base address + 390H	DestAddress5		Local AHB-master transfer Destination address
Base address + 394H	AdrCfg5		Address generation Configuration
Base address + 398H	TransferCfg5		Local AHB-master transfer Configuration
Base address + 39CH	FFStatus5		Status register
Base address + 3A0H	FFISTS_LT5		Interrupt Lower Threshold level status flags, a '1' signifies that the corresponding interrupt condition occurred (even if interrupt is disabled), write '1' clears the flag,
Base address + 3A4H	FFIEN_LT5		Interrupt Lower Threshold level Enable register
Base address + 3A8H	FFISTS_UT5		Interrupt Upper Threshold level status flags, a '1' signifies that the corresponding interrupt condition occurred (even if interrupt is disabled), write '1' clears the flag,
Base address + 3ACH	FFIEN_UT5		Interrupt Upper Threshold level Enable register
			Reserved
Base address + 3C0H	FFDataInL5		FIFO Data Lower In
Base address + 3C4H	FFDataInU5		FIFO Data Upper In
			Reserved
Base address + 400H	SWReset6	Channel6	SW reset
Base address + 404H	FFCfg6		FIFO Configuration
Base address + 408H	FFB6		FIFO Boundary Address
Base address + 40CH	FFT6		FIFO Threshold Level
Base address + 410H	DestAddress6		Local AHB-master transfer Destination address
Base address + 414H	AdrCfg6		Address generation Configuration

Address	Register Name	Channel	Description
Base address + 418_H	TransferCfg6		Local AHB-master transfer Configuration
Base address + $41C_H$	FFStatus6		Status register
Base address + 420_H	FFISTS_LT6		Interrupt Lower Threshold level status flags, a '1' signifies that the corresponding interrupt condition occurred (even if interrupt is disabled), write '1' clears the flag,
Base address + 424_H	FFIEN_LT6		Interrupt Lower Threshold level Enable register
Base address + 428_H	FFISTS_UT6		Interrupt Upper Threshold level status flags, a '1' signifies that the corresponding interrupt condition occurred (even if interrupt is disabled), write '1' clears the flag,
Base address + $42C_H$	FFIEN_UT6		Interrupt Upper Threshold level Enable register
			Reserved
Base address + 440_H	FFDataInL6		FIFO Data Lower In
Base address + 444_H	FFDataInU6		FIFO Data Upper In
			Reserved
Base address + 480_H	SWReset7	Channel7	SW reset
Base address + 484_H	FFCfg7		FIFO Configuration
Base address + 488_H	FFB7		FIFO Boundary Address
Base address + $48C_H$	FFT7		FIFO Threshold Level
Base address + 490_H	DestAddress7		Local AHB-master transfer Destination address
Base address + 494_H	AdrCfg7		Address generation Configuration
Base address + 498_H	TransferCfg7		Local AHB-master transfer Configuration
Base address + $49C_H$	FFStatus7		Status register
Base address + $4A0_H$	FFISTS_LT7		Interrupt Lower Threshold level status flags, a '1' signifies that the corresponding interrupt condition occurred (even if interrupt is disabled), write '1' clears the flag,
Base address + $4A4_H$	FFIEN_LT7		Interrupt Lower Threshold level Enable register
Base address + $4A8_H$	FFISTS_UT7		Interrupt Upper Threshold level status flags, a '1' signifies that the corresponding interrupt condition occurred (even if interrupt is disabled), write '1' clears the flag,
Base address + $4AC_H$	FFIEN_UT7		Interrupt Upper Threshold level Enable register
			Reserved
Base address + $4C0_H$	FFDataInL7		FIFO Data Lower In
Base address + $4C4_H$	FFDataInU7		FIFO Data Upper In
			Reserved
Base address + 800_H : Base address + FFF_H	MemoryData		memory mapped access to SRAM

21.4.4. Register Description

21.4.4.1. FFISTS

Register address	BaseAddress + 000H																															
Bit number	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
Field name	IStsOverFlow7	IStsUnderFlow7	Reserved	SLV_ERR7	IStsOverFlow6	IStsUnderFlow6	Reserved	SLV_ERR6	IStsOverFlow5	IStsUnderFlow5	Reserved	SLV_ERR5	IStsOverFlow4	IStsUnderFlow4	Reserved	SLV_ERR4	IStsOverFlow3	IStsUnderFlow3	Reserved	SLV_ERR3	IStsOverFlow2	IStsUnderFlow2	Reserved	SLV_ERR2	IStsOverFlow1	IStsUnderFlow1	Reserved	SLV_ERR1	IStsOverFlow0	IStsUnderFlow0	Reserved	SLV_ERR0
R/W	RW1	RW1	R0/W	RW1	RW1	RW1	R0/W	RW1	RW1	RW1	R0/W	RW1	RW1	RW1	R0/W	RW1	RW1	RW1	R0/W	RW1	RW1	RW1	R0/W	RW1	RW1	RW1	R0/W	RW1	RW1	RW1	R0/W	RW1
Reset value	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	0	0	0	0

Interrupt status flags, a '1' signifies that the corresponding interrupt condition occurred (even if the interrupt is disabled), write '1' clears the flag.
Even if the interrupt factor is canceled (e.g. not empty), or if an automatic clear signal (from Remote Handler) is set to 1, this bit will be cleared automatically.
When the rising edge and StatusClear of an interrupt factor occur simultaneously, Status Register gives priority to an interrupt factor. This register is initialized by SWReset of each channel.

Bit 31 IStsOverFlow7 (Ch7)
Interrupt Status for condition: FIFO overflow, FIFO input written during FIFO full, no data is overwritten, data is skipped, the application must avoid this situation!
If the interrupt factor has occurred, this register is set to 1 by the rising edge of i_TRIGGER[7].
This bit will be cleared, if "1" is written in this register or the auto clear signal from Remote Handler performs.
0: No interrupt
1: Interrupt

Bit 30 IStsUnderFlow7 (Ch7)
Interrupt Status for condition: FIFO underflow, FIFO output triggered during FIFO empty.
When the interrupt factor has occurred, this register is set to 1 by the rising edge of i_TRIGGER[7].
This bit will be cleared, if "1" is written in this register or the auto clear signal from Remote Handler performs.

Bit 29 Reserved

Bit 28 SLV_ERR6 (Ch7)
Interrupt Status for condition: AHB Slave module signal HRESP is set to ERROR.
This bit will be cleared, if "1" is written in this register or the auto clear signal from Remote Handler performs.

Bit 27 IStsOverFlow6 (Ch6)
Interrupt Status for condition: FIFO overflow, FIFO input written during FIFO full, no data is overwritten, data is skipped, Application must avoid this situation!
When the interrupt factor has occurred, this register is set to 1 by the rising edge of i_TRIGGER[6].
This bit will be cleared, if "1" is written in this register or the auto clear signal from Remote Handler performs.

Bit 26 IStsUnderFlow6 (Ch6)
Interrupt Status for condition: FIFO underflow, FIFO output triggered during FIFO empty.
When the interrupt factor has occurred, this register is set to 1 by the rising edge of i_TRIGGER[6].
This bit will be cleared, if "1" is written in this register or the auto clear signal from Remote Handler performs.

Bit 25 Reserved

Bit 24 SLV_ERR6 (Ch6)
Interrupt Status for condition: AHB Slave module signal HRESP is set to ERROR.
This bit will be cleared, if "1" is written in this register or the auto clear signal from Remote Handler performs.

Bit 23 IStsOverFlow5 (Ch5)
Interrupt Status for condition: FIFO overflow, FIFO input written during FIFO full, no data is overwritten, data is skipped, Application must avoid this situation!
When the interrupt factor has occurred, this register is set to 1 by the rising edge of i_TRIGGER[5].
This bit will be cleared, if "1" is written in this register or the auto clear signal from Remote Handler performs.

Bit 22 IStsUnderFlow5 (Ch5)
Interrupt Status for condition: FIFO underflow, FIFO output triggered during FIFO empty.
When the interrupt factor has occurred, this register is set to 1 by the rising edge of i_TRIGGER[5].
This bit will be cleared, if "1" is written in this register or the auto clear signal from Remote Handler performs.

Bit 21 Reserved

Bit 20 SLV_ERR5 (Ch5)
Interrupt Status for condition: AHB Slave module signal HRESP is set to ERROR.
This bit will be cleared, if "1" is written in this register or the auto clear signal from Remote Handler performs.

Bit 19 IStsOverFlow4 (Ch4)
Interrupt Status for condition: FIFO overflow, FIFO input written during FIFO full, no data is overwritten, data is skipped, Application must avoid this situation!
When the interrupt factor has occurred, this register is set to 1 by the rising edge of i_TRIGGER[4].
This bit will be cleared, if "1" is written in this register or the auto clear signal from Remote Handler performs.

Bit 18 IStsUnderFlow4 (Ch4)
Interrupt Status for condition: FIFO underflow, FIFO output triggered during FIFO empty.
When the interrupt factor has occurred, this register is set to 1 by the rising edge of i_TRIGGER[4].
This bit will be cleared, if "1" is written in this register or the auto clear signal from Remote Handler performs.

Bit 17 Reserved

Bit 16 SLV_ERR4 (Ch4)
Interrupt Status for condition: AHB Slave module signal HRESP is set to ERROR.
This bit will be cleared, if "1" is written in this register or the auto clear signal from Remote Handler performs.

Bit 15 IStsOverFlow3 (Ch3)
Interrupt Status for condition: FIFO overflow, FIFO input written during FIFO full, no data is overwritten, data is skipped, Application must avoid this situation!
When the interrupt factor has occurred, this register is set to 1 by the rising edge of i_TRIGGER[3].
This bit will be cleared, if "1" is written in this register or the auto clear signal from Remote Handler performs.

Bit 14 IStsUnderFlow3 (Ch3)
Interrupt Status for condition: FIFO underflow, FIFO output triggered during FIFO empty.
When the interrupt factor has occurred, this register is set to 1 by the rising edge of i_TRIGGER[3].
This bit will be cleared, if "1" is written in this register or the auto clear signal from Remote Handler performs.

Bit 13 Reserved

Bit 12 SLV_ERR3 (Ch3)
Interrupt Status for condition: AHB Slave module signal HRESP is set to ERROR.
This bit will be cleared, if "1" is written in this register or the auto clear signal from Remote Handler performs.

Bit 11 IStsOverFlow2 (Ch2)
Interrupt Status for condition: FIFO overflow, FIFO input written during FIFO full, no data is overwritten, data is skipped, Application must avoid this situation!
When the interrupt factor has occurred, this register is set to 1 by the rising edge of i_TRIGGER[2].
This bit will be cleared, if "1" is written in this register or the auto clear signal from Remote Handler performs.

Bit 10 IStsUnderFlow2 (Ch2)
Interrupt Status for condition: FIFO underflow, FIFO output triggered during FIFO empty.
When the interrupt factor has occurred, this register is set to 1 by the rising edge of i_TRIGGER[2].
This bit will be cleared, if "1" is written in this register or the auto clear signal from Remote Handler performs.

Bit 9 Reserved

Bit 8 SLV_ERR2 (Ch2)
Interrupt Status for condition: AHB Slave module signal HRESP is set to ERROR.
This bit will be cleared, if "1" is written in this register or the auto clear signal from Remote Handler performs.

Bit 7 IStsOverFlow1 (Ch1)
Interrupt Status for condition: FIFO overflow, FIFO input written during FIFO full, no data is overwritten, data is skipped, Application must avoid this situation!
When the interrupt factor has occurred, this register is set to 1 by the rising edge of i_TRIGGER[1].
This bit will be cleared, if "1" is written in this register or the auto clear signal from Remote Handler performs.

Bit 6 IStsUnderFlow1 (Ch1)
Interrupt Status for condition: FIFO underflow, FIFO output triggered during FIFO empty.
When the interrupt factor has occurred, this register is set to 1 by the rising edge of i_TRIGGER[1].
This bit will be cleared, if "1" is written in this register or the auto clear signal from Remote Handler performs.

Bit 5 Reserved

Bit 4 SLV_ERR1 (Ch1)
Interrupt Status for condition: AHB Slave module signal HRESP is set to ERROR.
This bit will be cleared, if "1" is written in this register or the auto clear signal from Remote Handler performs.

Bit 3 IStsOverFlow0 (Ch0)
Interrupt Status for condition: FIFO overflow, FIFO input written during FIFO full, no data is overwritten, data is skipped, Application must avoid this situation!
When the interrupt factor has occurred, this register is set to 1 by the rising edge of i_TRIGGER[0].
This bit will be cleared, if "1" is written in this register or the auto clear signal from Remote Handler performs.

Bit 2 IStsUnderFlow0 (Ch0)
Interrupt Status for condition: FIFO underflow, FIFO output triggered during FIFO empty.
When the interrupt factor has occurred, this register is set to 1 by the rising edge of i_TRIGGER[0].
This bit will be cleared, if "1" is written in this register or the auto clear signal from Remote Handler performs.

Bit 1 Reserved

Bit 0 SLV_ERR0 (Ch0)
Interrupt Status for condition: AHB Slave module signal HRESP is set to ERROR.
This bit will be cleared, if "1" is written in this register or the auto clear signal from Remote Handler performs.

21.4.4.2. FFIEN

Register address	BaseAddress + 004H																															
Bit number	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
Field name	IEnOverFlow7	IEnUnderFlow7	Reserved	IESLV_ERR7	IEnOverFlow6	IEnUnderFlow6	Reserved	IESLV_ERR6	IEnOverFlow5	IEnUnderFlow5	Reserved	IESLV_ERR5	IEnOverFlow4	IEnUnderFlow4	Reserved	IESLV_ERR4	IEnOverFlow3	IEnUnderFlow3	Reserved	IESLV_ERR3	IEnOverFlow2	IEnUnderFlow2	Reserved	IESLV_ERR2	IEnOverFlow1	IEnUnderFlow1	Reserved	IESLV_ERR1	IEnOverFlow0	IEnUnderFlow0	Reserved	IESLV_ERR0
R/W	RW	RW	R0/W	RW	RW	RW	R0/W	RW	RW	RW	R0/W	RW	RW	RW	R0/W	RW	RW	RW	R0/W	RW	RW	RW	R0/W	RW	RW	RW	R0/W	RW	RW	RW	R0/W	RW
Reset value	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	0	0	0	0

Interrupt Enable register. '1' is enable.

Bit 31 IEnOverFlow7 (Ch7)
Interrupt enable
This bit controls the output to o_OF_INT[7] in a IStsOverFlow7 bit of the Status Register.
0: o_OF_INT[7] is Disable
1: o_OF_INT[7] is Enable

Bit 30 IEnUnderFlow7 (Ch7)
Interrupt enable
This bit controls the output to o_UF_INT[7] in a IStsUnderFlow7 bit of the Status Register.
0: o_UF_INT[7] is Disable
1: o_UF_INT[7] is Enable

Bit 29 reserved

Bit 28 IESLV_ERR7 (Ch7)
Interrupt enable
This bit controls the output to o_VAR_INT in a SLV_ERR7 bit of the Status Register.
0: o_VAR_INT is Disable
1: o_VAR_INT is Enable

Bit 27 IEnOverFlow6 (Ch6)
Interrupt enable
This bit controls the output to o_OF_INT[6] in a IStsOverFlow6 bit of the Status Register.
0: o_OF_INT[6] is Disable
1: o_OF_INT[6] is Enable

Bit 26 IEnUnderFlow6 (Ch6)
Interrupt enable
This bit controls the output to o_UF_INT[6] in a IStsUnderFlow6 bit of the Status Register.
0: o_UF_INT[6] is Disable
1: o_UF_INT[6] is Enable

Bit 25 reserved

Bit 24 IESLV_ERR6 (Ch6)
Interrupt enable
This bit controls the output to o_VAR_INT in a SLV_ERR6 bit of the Status Register.
0: o_VAR_INT is Disable
1: o_VAR_INT is Enable

Bit 23 IEnOverFlow5 (Ch5)
Interrupt enable
This bit controls the output to o_OF_INT[5] in a IStsOverFlow5 bit of the Status Register.
0: o_OF_INT[5] is Disable
1: o_OF_INT[5] is Enable

Bit 22 IEnUnderFlow5 (Ch5)
Interrupt enable
This bit controls the output to o_UF_INT[5] in a IStsUnderFlow5 bit of the Status Register.
0: o_UF_INT[5] is Disable
1: o_UF_INT[5] is Enable

Bit 21 reserved

Bit 20 IESLV_ERR5 (Ch5)
Interrupt enable
This bit controls the output to o_VAR_INT in a SLV_ERR5 bit of the Status Register.
0: o_VAR_INT is Disable
1: o_VAR_INT is Enable

Bit 19 IEnOverFlow4 (Ch4)
Interrupt enable
This bit controls the output to o_OF_INT[4] in a IStsOverFlow4 bit of the Status Register.
0: o_OF_INT[4] is Disable
1: o_OF_INT[4] is Enable

Bit 18 IEnUnderFlow4 (Ch4)
Interrupt enable
This bit controls the output to o_UF_INT[4] in a IStsUnderFlow4 bit of the Status Register.
0: o_UF_INT[4] is Disable
1: o_UF_INT[4] is Enable

Bit 17 reserved

Bit 16 IESLV_ERR5 (Ch5)
Interrupt enable
This bit controls the output to o_VAR_INT in a SLV_ERR4 bit of the Status Register.
0: o_VAR_INT is Disable
1: o_VAR_INT is Enable

Bit 15 IEnOverFlow3 (Ch3)
Interrupt enable
This bit controls the output to o_OF_INT[3] in a IStsOverFlow3 bit of the Status Register.
0: o_OF_INT[3] is Disable
1: o_OF_INT[3] is Enable

Bit 14 IEnUnderFlow3 (Ch3)
Interrupt enable
This bit controls the output to o_UF_INT[3] in a IStsUnderFlow3 bit of the Status Register.
0: o_UF_INT[3] is Disable
1: o_UF_INT[3] is Enable

Bit 13 reserved

Bit 12 IESLV_ERR3 (Ch3)
Interrupt enable
This bit controls the output to o_VAR_INT in a SLV_ERR3 bit of the Status Register.
0: o_VAR_INT is Disable
1: o_VAR_INT is Enable

Bit 11 IEnOverFlow2 (Ch2)
Interrupt enable
This bit controls the output to o_OF_INT[2] in a IStsOverFlow2 bit of the Status Register.
0: o_OF_INT[2] is Disable
1: o_OF_INT[2] is Enable

Bit 10 IEnUnderFlow2 (Ch2)
Interrupt enable
This bit controls the output to o_UF_INT[2] in a IStsUnderFlow2 bit of the Status Register.
0: o_UF_INT[2] is Disable
1: o_UF_INT[2] is Enable

Bit 9 reserved

Bit 8 IESLV_ERR2 (Ch2)
Interrupt enable
This bit controls the output to o_VAR_INT in a SLV_ERR2 bit of the Status Register.
0: o_VAR_INT is Disable
1: o_VAR_INT is Enable

Bit 7 IEnOverFlow1 (Ch1)
Interrupt enable
This bit controls the output to o_OF_INT[1] in a IStsOverFlow1 bit of the Status Register.
0: o_OF_INT[1] is Disable
1: o_OF_INT[1] is Enable

Bit 6 IEnUnderFlow1 (Ch1)
Interrupt enable
This bit controls the output to o_UF_INT[1] in a IStsUnderFlow1 bit of the Status Register.
0: o_UF_INT[1] is Disable
1: o_UF_INT[1] is Enable

Bit 5 reserved

Bit 4 IESLV_ERR1 (Ch1)
Interrupt enable
This bit controls the output to o_VAR_INT in a SLV_ERR1 bit of the Status Register.
0: o_VAR_INT is Disable
1: o_VAR_INT is Enable

Bit 3 IEnOverFlow0 (Ch0)
Interrupt enable
This bit controls the output to o_OF_INT[0] in a IStsOverFlow0 bit of the Status Register.
0: o_OF_INT[0] is Disable
1: o_OF_INT[0] is Enable

Bit 2 IEnUnderFlow0 (Ch0)
Interrupt enable
This bit controls the output to o_UF_INT[0] in a IStsUnderFlow0 bit of the Status Register.
0: o_UF_INT[0] is Disable
1: o_UF_INT[0] is Enable
Bit 1 reserved
Bit 0 IESLV_ERR0 (Ch0)
Interrupt enable
This bit controls the output to o_VAR_INT in a SLV_ERR0 bit of the Status Register.
0: o_VAR_INT is Disable
1: o_VAR_INT is Enable

21.4.4.3. CFG_IDLE

Register address	BaseAddress + 010_H																															
Bit number	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
Field name	Reserved																															CFG_IDLE
R/W	R0/W																															R
Reset value	0																															0

A summary of the state of all the channels is shown.
Bit 0 CFG_IDLE
0: All the channels of ConfigFIFO are in an idle state.
1: Some channels are in an activity state.

21.4.4.4. CHPriority

Register address	BaseAddress + 014_H																															
Bit number	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
Field name	Reserved																														Priority	
R/W	R0/W																														R/W	
Reset value	0																														0	

Changes the priority of a trigger request. Set up by an initial setting before a trigger request comes. If a value is changed during operation, correct operation can not be guaranteed.
Bit 2 - 0 Priority
000: Priority High Ch0 – Ch1 – Ch2 – Ch3 – Ch4 – Ch5 – Ch6 – Ch7 Priority Low
001 Priority High Ch1 – Ch2 – Ch3 – Ch4 – Ch5 – Ch6 – Ch7 – Ch0 Priority Low
010 Priority High Ch2 – Ch3 – Ch4 – Ch5 – Ch6 – Ch7 – Ch0 – Ch1 Priority Low
011 Priority High Ch3 – Ch4 – Ch5 – Ch6 – Ch7 – Ch0 – Ch1 – Ch2 Priority Low
100 Priority High Ch4 – Ch5 – Ch6 – Ch7 – Ch0 – Ch1 – Ch2 – Ch3 Priority Low
101 Priority High Ch5 – Ch6 – Ch7 – Ch0 – Ch1 – Ch2 – Ch3 – Ch4 Priority Low
110 Priority High Ch6 – Ch7 – Ch0 – Ch1 – Ch2 – Ch3 – Ch4 – Ch5 Priority Low
111: Priority High Ch7 – Ch0 – Ch1 – Ch2 – Ch3 – Ch4 – Ch5 – Ch6 Priority Low

21.4.4.5. SWReset0 - SWReset7

Register address	Channel0 : BaseAddress + 100_H Channel1 : BaseAddress + 180_H Channel2 : BaseAddress + 200_H Channel3 : BaseAddress + 280_H Channel4 : BaseAddress + 300_H Channel5 : BaseAddress + 380_H Channel6 : BaseAddress + 400_H Channel7 : BaseAddress + 480_H																															
Bit number	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
Field name	Reserved																															SWReset0 SWReset1 SWReset2 SWReset3 SWReset4 SWReset5 SWReset6 SWReset7
R/W	R0/W																															RW
Reset value	0																															0_H

SW reset

Bit 0 SWReset0 - SWReset7

sw reset (flush FIFO, resets DMA target address counter)

Reset by writing a "1" to SWReset. When "0" is written, the reset is released.

Not every register is initialized by SWReset (FFISTS,FFISTS_LT,FFISTS_UT exceptions). When SWReset is "1", Trigger request is not permitted.

After checking that the state of the FFStates register is IDLE, it is necessary to reset. Because, it is for EBT (Early Burst Termination) to occur, if it resets during BURST transfer.

21.4.4.6. FFCfg0 - FFCfg7

Register address	Channel0 : BaseAddress + 104_H Channel1 : BaseAddress + 184_H Channel2 : BaseAddress + 204_H Channel3 : BaseAddress + 284_H Channel4 : BaseAddress + 304_H Channel5 : BaseAddress + 384_H Channel6 : BaseAddress + 404_H Channel7 : BaseAddress + 484_H			
Bit number	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
Field name	Reserved	FFTempMode0 FFTempMode1 FFTempMode2 FFTempMode3 FFTempMode4 FFTempMode5 FFTempMode6 FFTempMode7	FFEmptyMode0 FFEmptyMode1 FFEmptyMode2 FFEmptyMode3 FFEmptyMode4 FFEmptyMode5 FFEmptyMode6 FFEmptyMode7	FFEnO0 FFEnO1 FFEnO2 FFEnO3 FFEnO4 FFEnO5 FFEnO6 FFEnO7
R/W	R0/W	RW	RW	RW
Reset value	0	0_H	0_H	0_H

Configures the FIFO functionality.

Bit 2 FFTempMode0 - FFTempMode7
Selects FIFO write mode.
0 Temporary latch feature mode is disabled.
1 Temporary latch feature mode is enabled.

Bit 1 FFEmptyMode0 - FFEmptyMode7
This register specifies handling when the FIFO is empty.
0 If FIFO is emptied, FFEnO will be automatically set as 0.
1 Even if FIFO becomes empty, FFEnO is not automatically set to 0.

Bit 0 FFEnO0 - FFEnO7
Enable for FIFO Output.
0 Disable FIFO output, waits to complete the present transfer. The next request is not received after completion.
1 Trigger change to FIFO active state FIFO output is activated.
In the case of FFEmptyMode0=0, if FIFO is emptied, it will be automatically set to 0. If the FIFO is empty and present value or previous value of FFEmptyMode0 = 0, it can not be set to 1.

21.4.4.7. FFB0 - FFB7

Register address	Channel0 : BaseAddress + 108_H Channel1 : BaseAddress + 188_H Channel2 : BaseAddress + 208_H Channel3 : BaseAddress + 288_H Channel4 : BaseAddress + 308_H Channel5 : BaseAddress + 388_H Channel6 : BaseAddress + 408_H Channel7 : BaseAddress + 488_H																													
Bit number	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
Field name	Reserved								UpperBoundAdr0 UpperBoundAdr1 UpperBoundAdr2 UpperBoundAdr3 UpperBoundAdr4 UpperBoundAdr5 UpperBoundAdr6 UpperBoundAdr7								Reserved								LowerBoundAdr0 LowerBoundAdr1 LowerBoundAdr2 LowerBoundAdr3 LowerBoundAdr4 LowerBoundAdr5 LowerBoundAdr6 LowerBoundAdr7							
R/W	R0/W								RW								R0/W								RW							
Reset value	0								0_H								0								0_H							

This address sets the boundary address of the common (sharing) FIFO.
Do not overlap the channels boundary area settings. If the area overlaps, operation may malfunction.
An example of a setting is shown in Figure 21-2 FIFO Address setting example
When using this channel, it sets up so that it may be set to "UpperBoundAdr >= LowerBoundAdr."

Bit 23 - 16 UpperBoundAdr0 - UpperBoundAdr7
Memory address for Upper Boundary of FIFO (Double Word Address)

Bit 7 - 0 LowerBoundAdr0 - LowerBoundAdr7
Memory address for Lower Boundary of FIFO (Double Word Address)

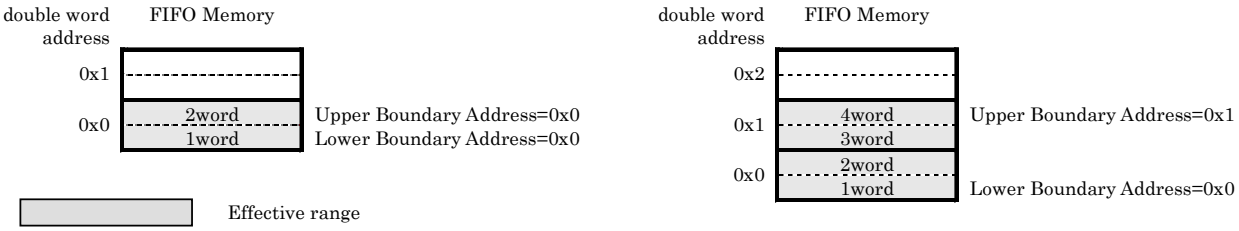


Figure 21-3 FIFO Address setting example.

21.4.4.8. FFT0 - FFT7

Register address	Channel0 : BaseAddress + $10C_H$ Channel1 : BaseAddress + $18C_H$ Channel2 : BaseAddress + $20C_H$ Channel3 : BaseAddress + $28C_H$ Channel4 : BaseAddress + $30C_H$ Channel5 : BaseAddress + $38C_H$ Channel6 : BaseAddress + $40C_H$ Channel7 : BaseAddress + $48C_H$																													
Bit number	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
Field name	Reserved					UpperThres0 UpperThres1 UpperThres2 UpperThres3 UpperThres4 UpperThres5 UpperThres6 UpperThres7											Reserved					LowerThres0 LowerThres1 LowerThres2 LowerThres3 LowerThres4 LowerThres5 LowerThres6 LowerThres7											
R/W	R0/W					RW											R0/W					RW									
Reset value	0					0_H											0					0_H									

Level threshold is set to the address in Boundary Area.

Bit 26 - 16 UpperThres0 - UpperThres7
The upper fill level threshold, an interrupt is generated if the fill level exceeds this threshold
This register sets the number of data bytes.
If the fill data count exceeds the set value, it is used as a warning.
(Fill data > Upper threshold level)
ex) UpperThres0=1fh
If the fill data is 32 bytes or more, it is used as a warning (Figure 21-4 Upper threshold warning)

Bit 10 - 0 LowerThres0 - LowerThres7
Lower fill level threshold, an interrupt is generated if the fill level drops below the threshold level.
This register sets the number of data bytes.
If the fill data count drops below the set value, it is used as a warning.
(Fill data < lower threshold level)
ex) LowerThres0=4h
If the data of the remainder falls below 4 bytes, it is used as a warning (Figure 22-5 Lower threshold warning)

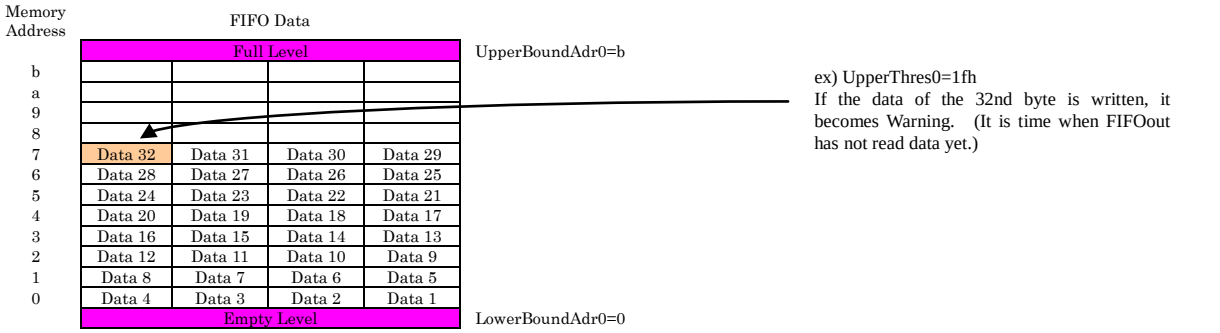


Figure 21-4 Upper threshold warning

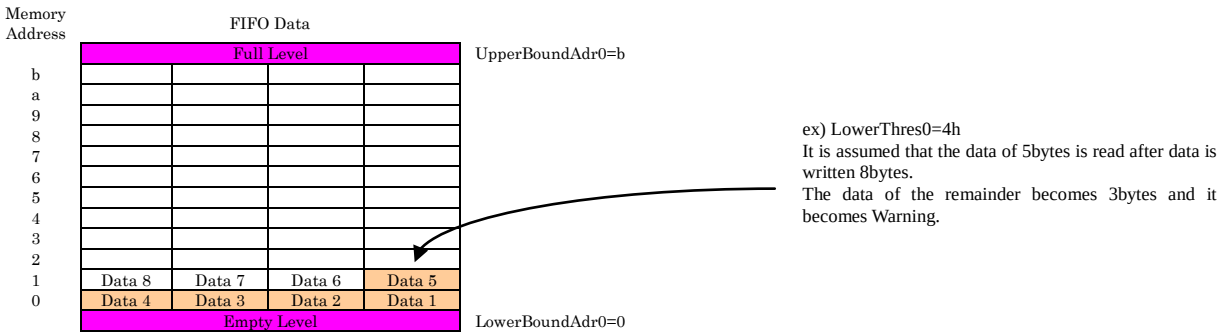


Figure 21-5 Lower threshold warning

The FIFO does not shift data, even if data is read from FIFOout.
When 1 byte of data is read from FIFOout, the Empty and Full and Level is carried out +1,

21.4.4.9. DestAddress0 - DestAddress7

Register address	Channel0 : BaseAddress + 110_H Channel1 : BaseAddress + 190_H Channel2 : BaseAddress + 210_H Channel3 : BaseAddress + 290_H Channel4 : BaseAddress + 310_H Channel5 : BaseAddress + 390_H Channel6 : BaseAddress + 410_H Channel7 : BaseAddress + 490_H																													
Bit number	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
Field name	AHBMDA0 AHBMDA1 AHBMDA2 AHBMDA3 AHBMDA4 AHBMDA5 AHBMDA6 AHBMDA7																													
R/W	RW																													
Reset value	0_H																													

Local AHB-master transfer Destination address
Bit 31 - 0 AHBMDA0 - AHBMDA7
Destination address to start AHB-master transfer. A value does not change during transfer.

21.4.4.10. AdrCfg0 - AdrCfg7

Register address		Channel0 : BaseAddress + 114_H Channel1 : BaseAddress + 194_H Channel2 : BaseAddress + 214_H Channel3 : BaseAddress + 294_H Channel4 : BaseAddress + 314_H Channel5 : BaseAddress + 394_H Channel6 : BaseAddress + 414_H Channel7 : BaseAddress + 494_H																															
Bit number	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
Field name	Reserved																														AdrMode10 AdrMode11 AdrMode12 AdrMode13 AdrMode14 AdrMode15 AdrMode16 AdrMode17	AdrMode00 AdrMode01 AdrMode02 AdrMode03 AdrMode04 AdrMode05 AdrMode06 AdrMode07
R/W	R0/W																														RW	RW
Reset value	0																														0_H	0_H

Address generation Configuration

Bit 1 AdrMode10 - AdrMode17

- 0 Destination address is reset with each trigger to AHBMDA,
- 1 Destination address is not reset (It continues from the last destination address.), only with SWRESET. A value does not change during transfer.

Bit 0 AdrMode00 - AdrMode07

- 0 Destination address is incremented (TransferWidth is byte=+1, hword=+2, word=+4).
- 1 Destination address is fixed. A value does not change during transfer.

21.4.4.11. TransferCfg0 - TransferCfg7

Register address	Channel0 : BaseAddress + 118_H Channel1 : BaseAddress + 198_H Channel2 : BaseAddress + 218_H Channel3 : BaseAddress + 298_H Channel4 : BaseAddress + 318_H Channel5 : BaseAddress + 398_H Channel6 : BaseAddress + 418_H Channel7 : BaseAddress + 498_H

Bit number	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
Field name	Reserved											TransferINCR0 TransferINCR1 TransferINCR2 TransferINCR3 TransferINCR4 TransferINCR5 TransferINCR6 TransferINCR7	Reserved		TransferWidth0 TransferWidth1 TransferWidth2 TransferWidth3 TransferWidth4 TransferWidth5 TransferWidth6 TransferWidth7		Reserved										TransferNumber0 TransferNumber1 TransferNumber2 TransferNumber3 TransferNumber4 TransferNumber5 TransferNumber6 TransferNumber7					
R/W	R0/W											RW	R0/W		RW		R0/W										RW					
Reset value	0											0_H	0		0_H		0										0_H					

Local AHB master transfer Configuration

Bit 20 TransferINCR0 - TransferINCR7

0 INCR transfer is disabled
INCR (Indefinite length burst),INCR4,INCR8,INCR16 is not supported.
All the transfer is set to SINGLE.

1 INCR transfer is enabled
INCR (Indefinite length burst) and SINGLE,INCR4,INCR,INCR16 are supported.
When TransferNumber is 1 or AdrCfg0=1 (Address is Fixed mode), o_mHBURST becomes SINGLE.
When TransferNumber is 4, o_mHBURST becomes INCR4.
When TransferNumber is 8, o_mHBURST becomes INCR8.
When TransferNumber is 16, o_mHBURST becomes INCR16.
As for other values, o_mHBURST becomes INCR.
When a transfer address exceeds 1KB, transfer is not carried out by INCR4, and 8 and 16. It transmits by INCR.

Bit 17 - 16 TransferWidth0 - ransferWidth7

HSIZE set. A value does not change during transfer.
00b byte
01b halfword
10b word
11b reserved (word)

Bit 5 - 0 TransferNumber0 - ransferNumber7

Number of AHB-master Transfers for each Trigger beat.
A value does not change during transfer.
"0_H"=64 transfers
TransferWidth=00b (byte), TransferNumber=00000001b -> 1 transfer by the byte size.
TransferWidth=01b (hword), TransferNumber=00000001b -> 1 transfer by the hword size.
TransferWidth=10b (word), TransferNumber=00000001b -> 1 transfer by the word size.

When the number of transfer (TransferNumber of a TransferCfg register) is over FILL Level (FillLevel of a FFStatus register), even if a trigger request signal comes, an AHB master does not transfer.
When the following example has the large left side, an AHB master does not transfer.
(((TransferNumber-1)+1) << transferwidth) > FillLevel

21.4.4.12. FFStatus0 - FFStatus7

Register address	Channel0 : BaseAddress + 11CH Channel1 : BaseAddress + 19CH Channel2 : BaseAddress + 21CH Channel3 : BaseAddress + 29CH Channel4 : BaseAddress + 31CH Channel5 : BaseAddress + 39CH Channel6 : BaseAddress + 41CH Channel7 : BaseAddress + 49CH								
Bit number	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
Field name	Reserved	FillLevel0 FillLevel1 FillLevel2 FillLevel3 FillLevel4 FillLevel5 FillLevel6 FillLevel7	UBLV0 UBLV1 UBLV2 UBLV3 UBLV4 UBLV5 UBLV6 UBLV7	LBLV0 LBLV1 LBLV2 LBLV3 LBLV4 LBLV5 LBLV6 LBLV7	state0 state1 state2 state3 state4 state5 state6 state7	Reserved	full0 full1 full2 full3 full4 full5 full6 full7	empty0 empty1 empty2 empty3 empty4 empty5 empty6 empty7
R/W	R0/W	R	R	R	R	R0/W	R	R
Reset value	0	0_H	0_H	0_H	0	0	0	1

Status register

Bit 27 - 16 FillLevel0 - FillLevel7
FIFO fill level (0-2048byte).

Bit 15 - 12 UBLV0 (temporary latch Byte Lane Valid0) - UBLV7 (temporary latch Byte Lane Valid7)
It can be checked which byte lane of Upper Temporary Latch Register has been updated.
The byte lane which wrote in FFDataInU is set to 1.
If all the bits of LBLV and UBLV are set to 1, it will write in FIFO. And this bit is cleared by 0.
If this register becomes FIFO full, it will be cleared by 0.

Input Size	Input Address	UBLV
i_sHSIZE=00b (byte)	i_sHADDR[1:0]=00b	4'b0001
i_sHSIZE=00b (byte)	i_sHADDR[1:0]=01b	4'b0010
i_sHSIZE=00b (byte)	i_sHADDR[1:0]=10b	4'b0100
i_sHSIZE =00b (byte)	i_sHADDR[1:0]=11b	4'b1000
i_sHSIZE =01b (hword)	i_sHADDR[1:0]=0xb	4'b0011
i_sHSIZE =01b (hword)	i_sHADDR[1:0]=1xb	4'b1100
i_sHSIZE =1xb (word)	i_sHADDR[1:0]=xxb	4'b1111

Bit 11 - 8 LBLV0 (temporary latch Byte Lane Valid0) - LBLV7 (temporary latch Byte Lane Valid7)
It can be checked which byte lane of Lower Temporary Latch Register has been updated.
If all the bits of LBLV and UBLV are set to 1, it will write in FIFO. And this bit is cleared by 0.
If this register becomes FIFO full, it will be cleared by 0.

Input Size	Input Address	LBLV
i_sHSIZE=00b (byte)	i_sHADDR[1:0]=00b	4'b0001
i_sHSIZE=00b (byte)	i_sHADDR[1:0]=01b	4'b0010
i_sHSIZE=00b (byte)	i_sHADDR[1:0]=10b	4'b0100
i_sHSIZE =00b (byte)	i_sHADDR[1:0]=11b	4'b1000
i_sHSIZE =01b (hword)	i_sHADDR[1:0]=0xb	4'b0011
i_sHSIZE =01b (hword)	i_sHADDR[1:0]=1xb	4'b1100
i_sHSIZE =1xb (word)	i_sHADDR[1:0]=xxb	4'b1111

Bit 7 state0 - state7
The IDLE state of each channel is shown.
0b=idle, 1b=active

Bit 1 full0 - full7
FIFO full
0b=not full, 1b=full

Bit 0 empty0 - empty7
FIFO empty
0b=not empty, 1b=empty

21.4.4.13. FFISTS_LT0 - FFISTS_LT7

Register address	Channel0 : BaseAddress + 120_H Channel1 : BaseAddress + $1A0_H$ Channel2 : BaseAddress + 220_H Channel3 : BaseAddress + $2A0_H$ Channel4 : BaseAddress + 320_H Channel5 : BaseAddress + $3A0_H$ Channel6 : BaseAddress + 420_H Channel7 : BaseAddress + $4A0_H$																															
Bit number	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
Field name	Reserved																															IStsLT0 IStsLT1 IStsLT2 IStsLT3 IStsLT4 IStsLT5 IStsLT6 IStsLT7
R/W	R0/W																															RW1
Reset value	0																															0

Interrupt status flags, a '1' signifies that the corresponding interrupt condition occurred (even if interrupt is disabled), write '1' clears the flag.
Even if the interrupt factor is canceled (e.g. not empty), Or if an automatic clear signal (from Remote Handler) is set to 1, this bit will be cleared automatically.
When the rising edge and StatusClear of an interrupt factor happen simultaneously, Status Register gives priority to an interrupt factor.
This register is initialized by SWReset.

Bit 0 IStsLT0 - IStsLT7
Interrupt Status for condition: FIFO fill level below LowerThres.
When the interrupt factor has occurred, this register is set to 1 by the rising edge of i_TRIGGER[0-7].
This bit will be cleared, if "1" is written in this register or the auto clear signal from Remote Handler performs.
0: No interrupt
1: Interrupt

21.4.4.14. FFIEN_LT0 - FFIEN_LT7

Register address	Channel0 : BaseAddress + 124_H Channel1 : BaseAddress + $1A4_H$ Channel2 : BaseAddress + 224_H Channel3 : BaseAddress + $2A4_H$ Channel4 : BaseAddress + 324_H Channel5 : BaseAddress + $3A4_H$ Channel6 : BaseAddress + 424_H Channel7 : BaseAddress + $4A4_H$																															
Bit number	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
Field name	Reserved																															IEnLT0 IEnLT1 IEnLT2 IEnLT3 IEnLT4 IEnLT5 IEnLT6 IEnLT7
R/W	R0/W																															RW
Reset value	0																															0

Interrupt Enable register. '1' is enable.

Bit 0 IEnLT0 - IEnLT7
Interrupt enable
This bit controls the output to o_LT_INT[0] - o_LT_INT[7] in a IStsLT0 - IStsLT7 bit of the Status Register.
0: o_LT_INT[0] - o_LT_INT[7] is Disable
1: o_LT_INT[0] - o_LT_INT[7] is Enable

21.4.4.15. FFISTS_UT0 - FFISTS_UT7

Register address	Channel0 : BaseAddress + 128_H Channel1 : BaseAddress + $1A8_H$ Channel2 : BaseAddress + 228_H Channel3 : BaseAddress + $2A8_H$ Channel4 : BaseAddress + 328_H Channel5 : BaseAddress + $3A8_H$ Channel6 : BaseAddress + 428_H Channel7 : BaseAddress + $4A8_H$																															
Bit number	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
Field name	Reserved																															IStsUT0 IStsUT1 IStsUT2 IStsUT3 IStsUT4 IStsUT5 IStsUT6 IStsUT7
R/W	R0/W																															RW1
Reset value	0																															0

Interrupt status flags, a '1' signifies that the corresponding interrupt condition occurred (even if interrupt is disabled), write '1' clears the flag.
Even if the interrupt factor is canceled (e.g. not empty), Or if an automatic clear signal (from Remote Handler) is set to 1, this bit will be cleared automatically.
When the rising edge and StatusClear of an interrupt factor happen simultaneously, Status Register gives priority to an interrupt factor.
This register is initialized by SWReset.

Bit 0 IStsUT0 - IStsUT7
Interrupt Status for condition: FIFO fill level over UpperThres.
When the interrupt factor has occurred, this register is set to 1 by the rising edge of i_TRIGGER[0-7].
This bit will be cleared, if "1" is written in this register or the auto clear signal from Remote Handler performs.
0: No interrupt
1: Interrupt

21.4.4.16. FFIEN_UT0 - FFIEN_UT7

Register address	Channel0 : BaseAddress + $12C_H$ Channel1 : BaseAddress + $1AC_H$ Channel2 : BaseAddress + $22C_H$ Channel3 : BaseAddress + $2AC_H$ Channel4 : BaseAddress + $32C_H$ Channel5 : BaseAddress + $3AC_H$ Channel6 : BaseAddress + $42C_H$ Channel7 : BaseAddress + $4AC_H$																															
Bit number	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
Field name	Reserved																															IEnUT0 IEnUT1 IEnUT2 IEnUT3 IEnUT4 IEnUT5 IEnUT6 IEnUT7
R/W	R0/W																															RW
Reset value	0																															0

Interrupt Enable register. '1' is enable.

Bit 0 IEnUT0 - IEnUT7
Interrupt enable
This bit controls the output to o_UT_INT[0] - o_UT_INT[7] in a IStsUT0 - IStsUT7 bit of the Status Register.
0: o_UT_INT[0] - o_UT_INT[7] is Disable
1: o_UT_INT[0] - o_UT_INT[7] is Enable

21.4.4.17. FFDataInL0 - FFDataInL7

Register address	Channel0 : BaseAddress + 140_H Channel1 : BaseAddress + $1C0_H$ Channel2 : BaseAddress + 240_H Channel3 : BaseAddress + $2C0_H$ Channel4 : BaseAddress + 340_H Channel5 : BaseAddress + $3C0_H$ Channel6 : BaseAddress + 440_H Channel7 : BaseAddress + $4C0_H$																													
Bit number	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
Field name	DataInL0 DataInL1 DataInL2 DataInL3 DataInL4 DataInL5 DataInL6 DataInL7																															
R/W	R0W																															
Reset value	0_H																															

FIFO Data In Lower
Bit 31 - 0 DataInL0 - DataInL7
Lower Data Input for FIFO channel 0 , read delivers always 0

If 2 data words (8byte) of FFDataInL and FFDataInU register is receives, it will write in FIFO. Only if both registers are written the internal FIFO memory is updated, otherwise the internal FIFO memory remains unchanged.
This Mode is called “temporary latch feature” and it becomes effective in the case of FFTempEn=1 of a FFCfg register.

In the case of FFTempEn=0, Whenever it receives 1 data (byte,hword,word), it writes in FIFO. A receiving address writes data in the address of FFDataInL. Moreover, in the case of FFTmpEn=0, application needs to check a setup of TransferSize of a TransferCfg register, and it is necessary to control so that the size of writing and read-out becomes the same.
Even if both the modes receive data in the case of FIFO full, it does not write in FIFO.

(The example of transfer of Channel0)
- FFTempEN0=0
  Input Size is BYTE : 140h -> FIFO Write! -> 140h -> FIFO Write! -> … (Taking Data bit 7 - 0)
  Input Size is HWORD : 140h -> FIFO Write! -> 140h -> FIFO Write! -> … (Taking Data bit 15 - 0)
  Input Size is WORD : 140h -> FIFO Write! -> 140h -> FIFO Write! -> … (Taking Data bit 31 - 0)

- FFTempEN0=1 (temporary latch feature mode)
  Input Size is Byte : 140h -> 141h -> 142h -> … 146h -> 147h -> FIFO Write! -> 140h …
  Input Size is HWORD: 140h -> 142h -> 144h -> 146h -> FIFO Write! -> 140h …
  Input Size is WORD : 140h -> 144h -> FIFO Write! -> 140h -> 144h -> FIFO Write! -> 140h …

The writing to FFDataInL is once held to a Lower Temporary Latch register.
The Temporary Latch register of Upper and Lower will be written in FIFO, if all the byte lanes (8bytes) are updated.

It can be checked by LBLV and UBLV of FFStatus Register which byte lane has been updated.
However, Temporary Latch Byte Lane Valid register (LBLV) is not updated when it writes in to FFDataIn at the time of FIFO full.
Before writing to FIFO, it will be overwritten if data is transmitted to the same address.

21.4.4.18. FFDataInU0 - FFDataInU7

Register address	Channel0 : BaseAddress + 144_H Channel1 : BaseAddress + $1C4_H$ Channel2 : BaseAddress + 244_H Channel3 : BaseAddress + $2C4_H$ Channel4 : BaseAddress + 344_H Channel5 : BaseAddress + $3C4_H$ Channel6 : BaseAddress + 444_H Channel7 : BaseAddress + $4C4_H$																															
Bit number	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
Field name	DataInU0 DataInU1 DataInU2 DataInU3 DataInU4 DataInU5 DataInU6 DataInU7																															
R/W	R0W																															
Reset value	0_H																															

FIFO Data In Upper
Bit 31 - 0 DataInU0 - DataInU7
Upper Data Input for FIFO channel 0 , read delivers always 0

In the case of FFTempEn=1 of a FFCfg register, this register is used.
In the case of FFTempEn=0 of a FFCfg register, even if it receives to this register, it does not write in FIFO.

It can be checked by LBLV and UBLV of FFStatus Register which byte lane has been updated.
However, Temporary Latch Byte Lane Valid register (UBLV) is not updated when it writes in to FFDataIn at the time of FIFO full.
Before writing to FIFO, it will be overwritten if data is transmitted to the same address.

In the case of FFTempEn=1 of a FFCfg register, The writing to FIFO is as follows.

LowerBoundAddr <<1 +3h				
LowerBoundAddr <<1 +2h				
LowerBoundAddr <<1 +1h	FFDataInU[31:24]	FFDataInU[23:16]	FFDataInU[15:8]	FFDataInU[7:0]
LowerBoundAddr <<1 +0h	FFDataInL[31:24]	FFDataInL[23:16]	FFDataInL[15:8]	FFDataInL[7:0]

21.4.4.19. Memory Data [0…511]

Register address	BaseAddress + 800_H : BaseAddress + FFF_H																															
Bit number	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
Field name	Data																															
R/W	R																															
Reset value	X																															

memory mapped access to SRAM
Bit FIFO Data
31 - 0 This register is read only.

21.5. Processing Mode

21.5.1. Processing FIFO Flow

FIFO is empty after reset. Transfer more FIFOin data than Lower Threshold Level.

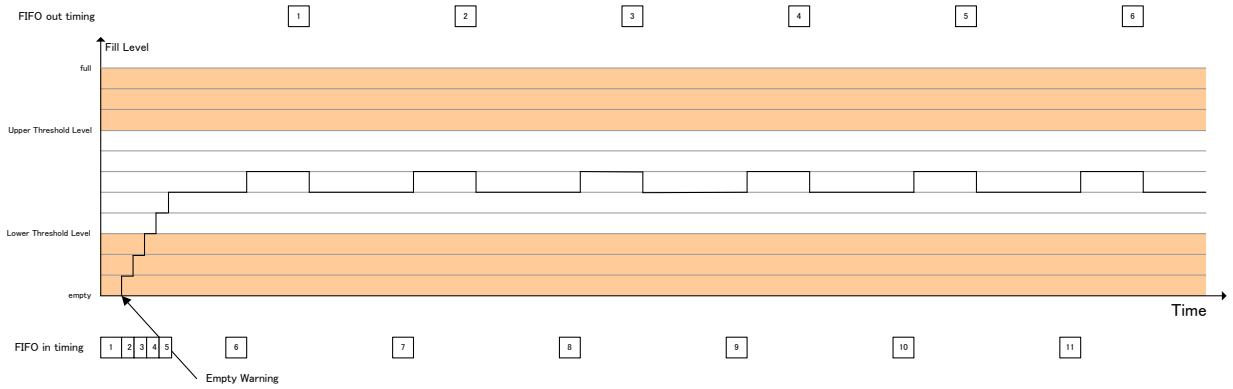


Figure 21-6 FIFO Flow (Empty Warning)

If the upper threshold warning level is exceeded, FIFOinput needs to postpone sending at least as much step data, as the warning buffer can hold.

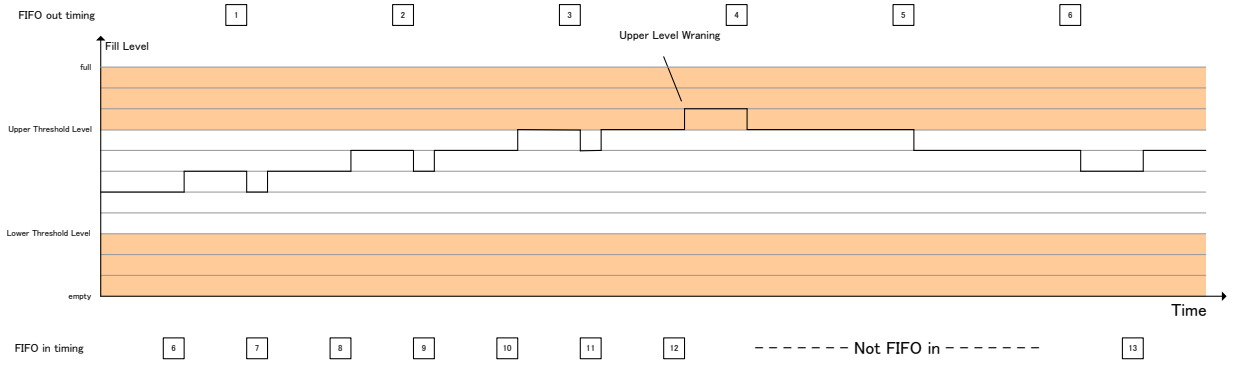


Figure 21-7 FIFO Flow (Upper Threshold Warning)

When Lower Threshold Level Warning occurs, transfer more FIFOin data than Lower Threshold Level.

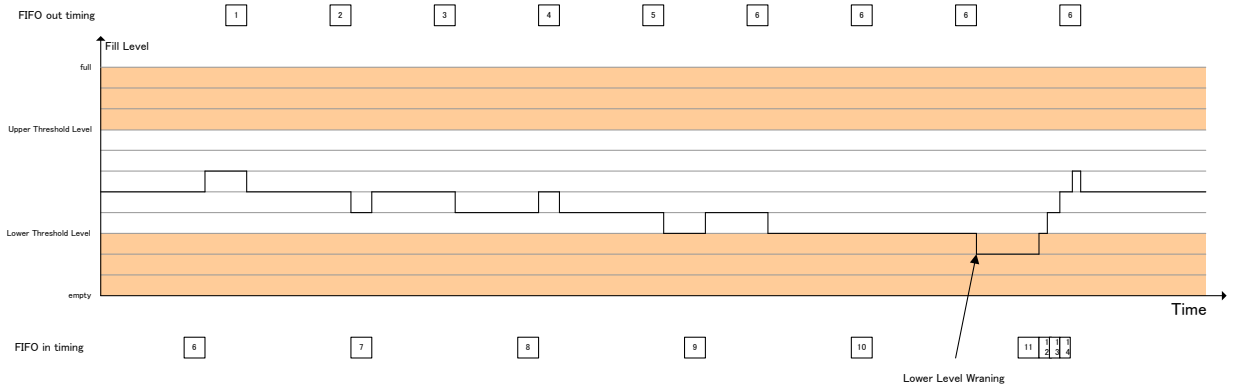


Figure 21-8 FIFO Flow (Lower Threshold Warning)

21.5.2. Register Setting

The procedure of a register set is shown below.
When setting up a register, please pay attention to the respective notes.
After setting up the register of the channel to be used, the FFCfg.FFEnO bit is set to Enable (“1”).
Operation cannot be guaranteed if the value of a register is changed during a transfer.
To change a set value during operation, it is necessary to set the SWReset Register to Clear ("1") or FFEnO bit to Disable (“0”).

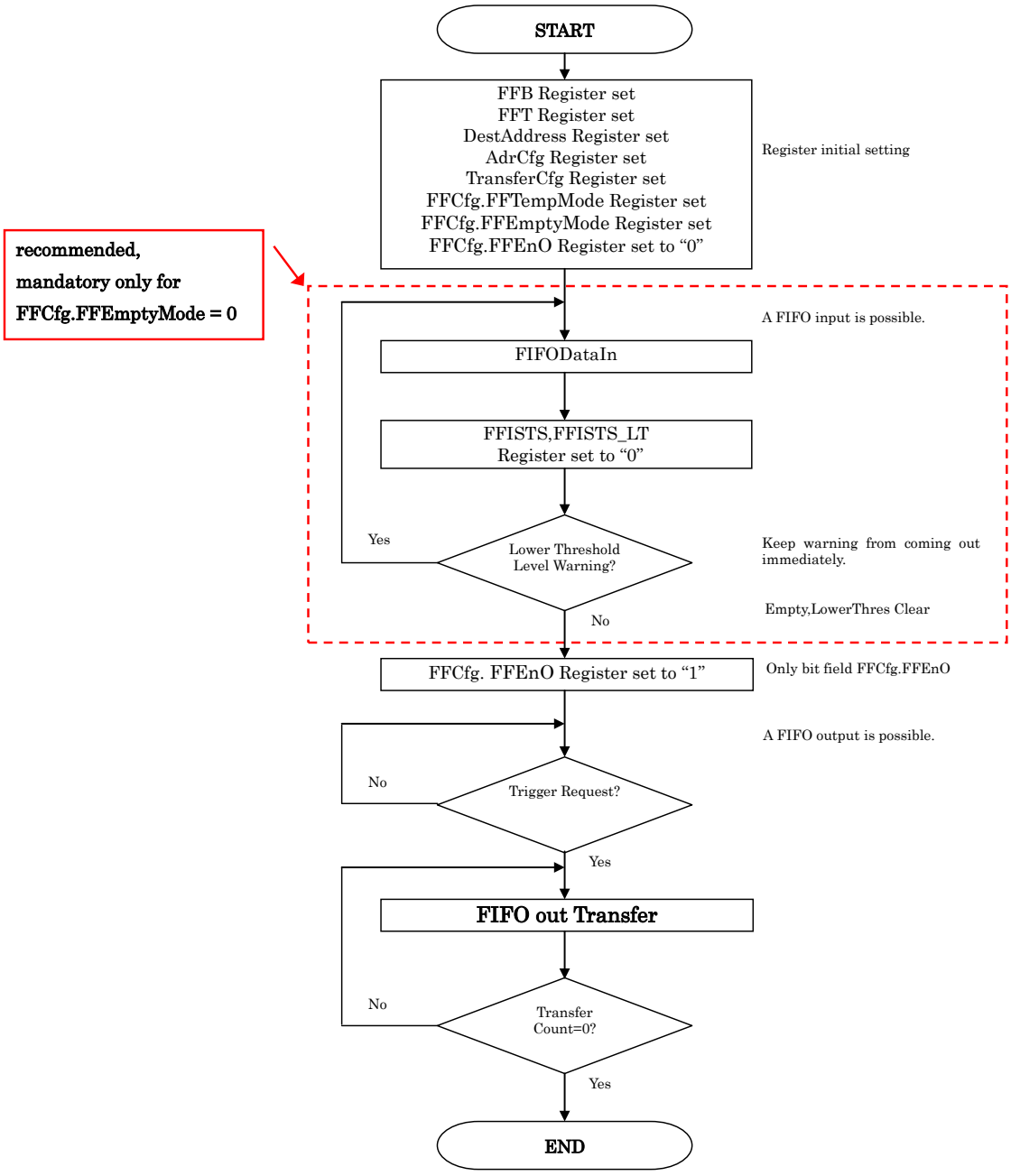


Figure 21-9 Register Initial Setting Flow

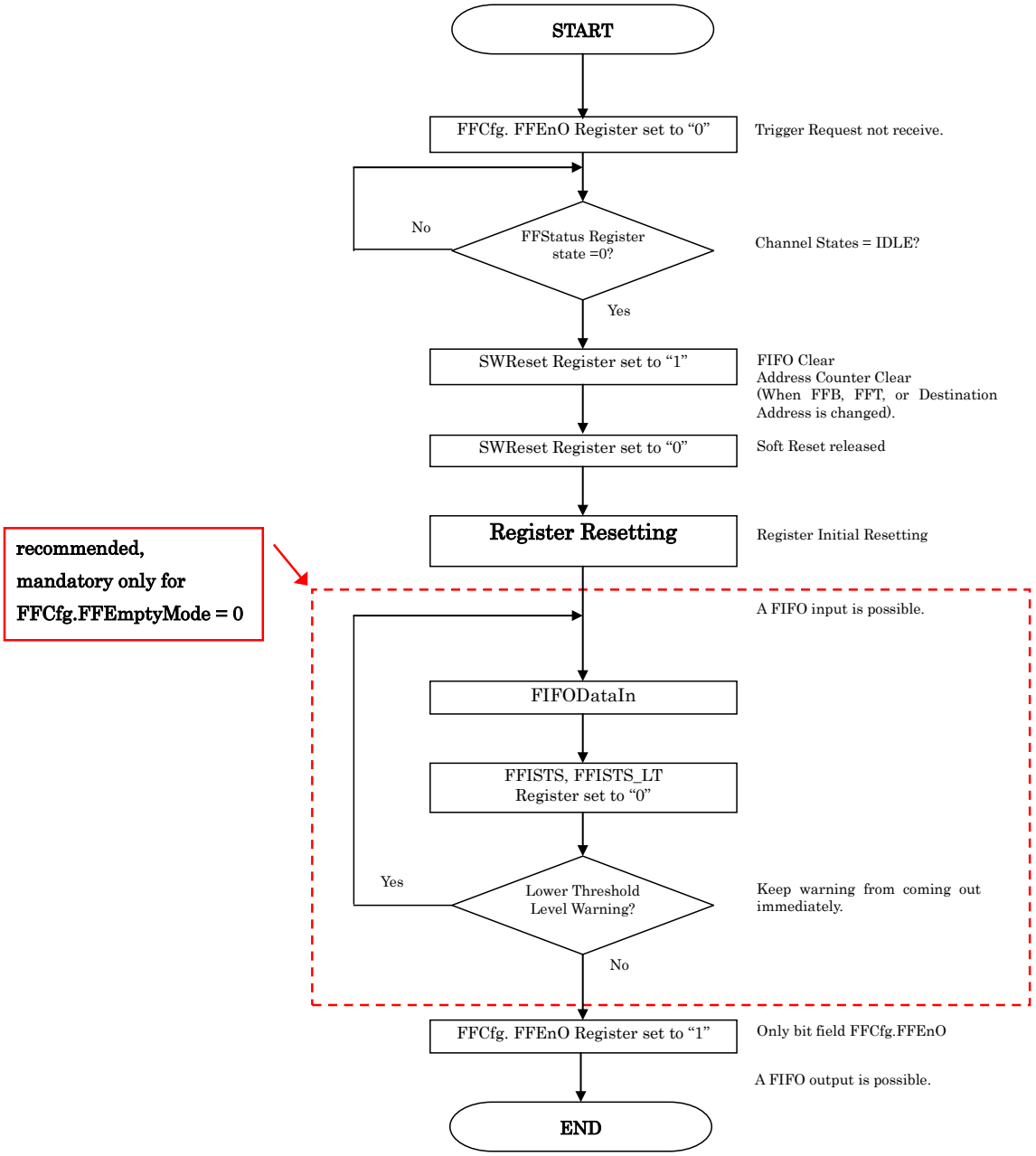


Figure 21-10 Register Resetting Flow

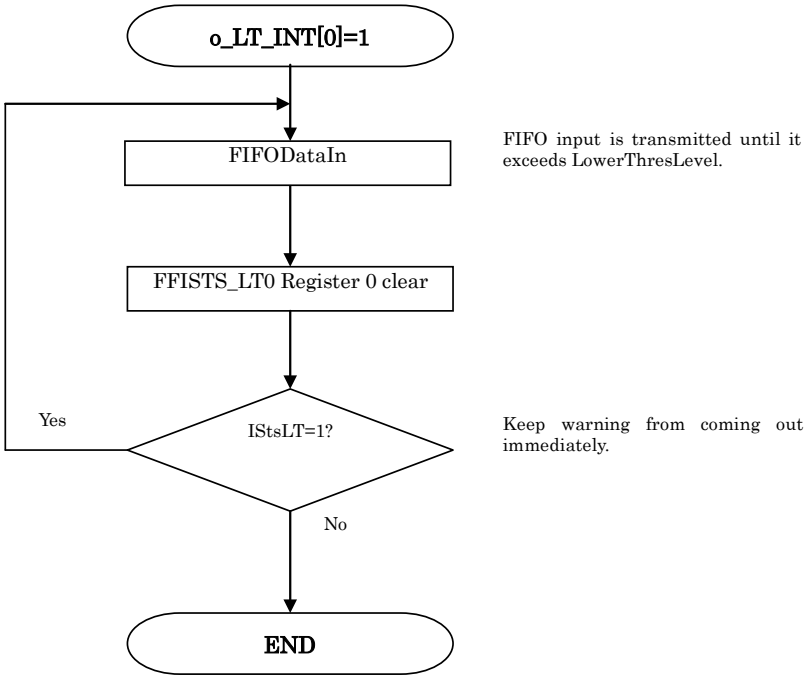


Figure 21-11 FIFO Lower Threshold Warning Flow

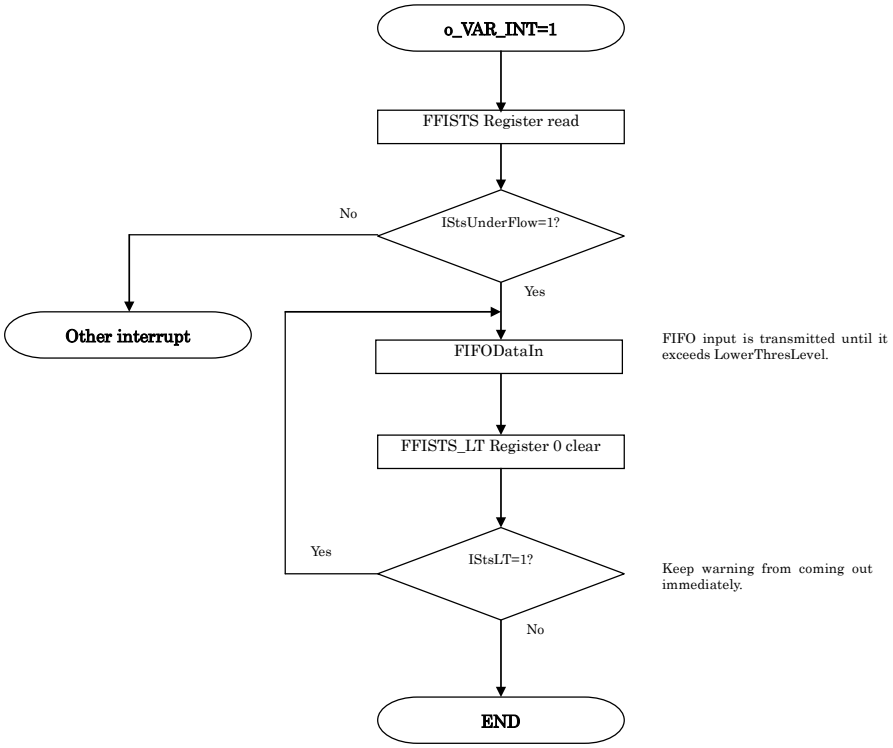


Figure 21-12 FIFO Empty Flow

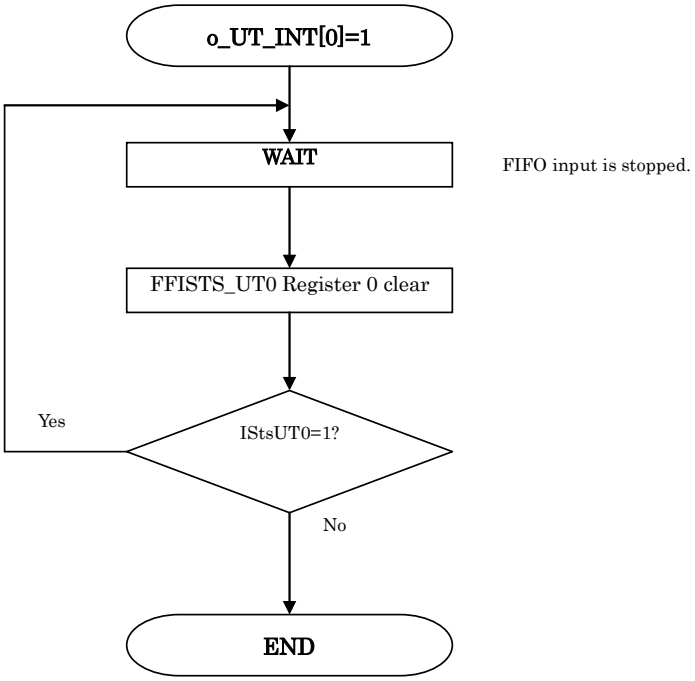


Figure 21-13 FIFO Upper Threshols Warinig Flow

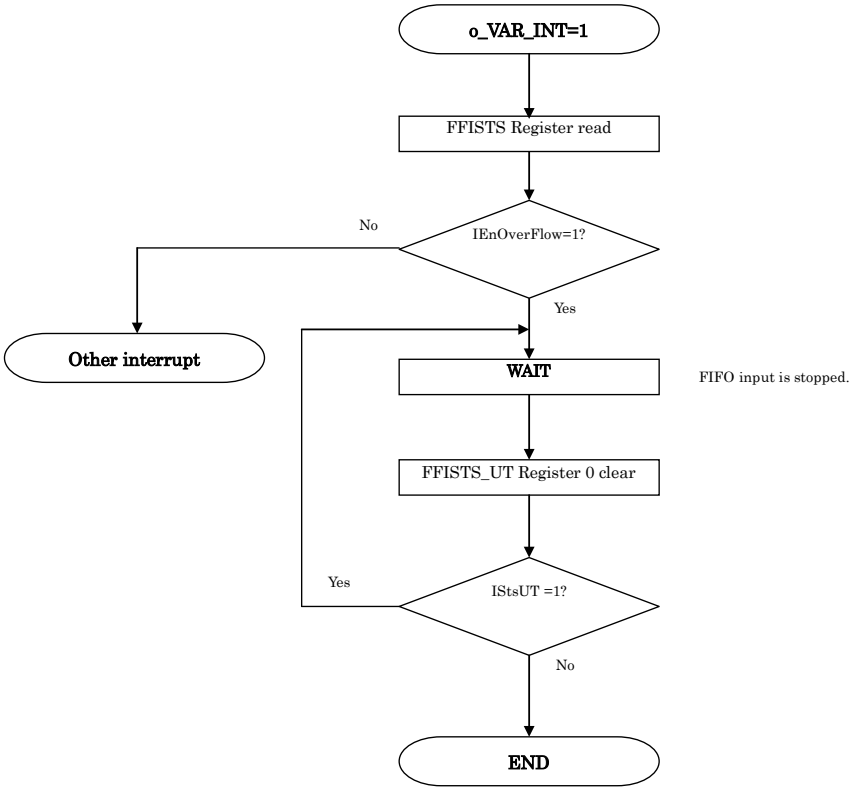


Figure 21-14 FIFO Full Flow

22. Stepper Motor Controller

This chapter describes the Stepper Motor Controller of the MB88F333.

22.1. Outline

The stepper motor controller consists of PWM pulse generators, motor drivers and selector logic circuits. The two motor drivers have a high-output driving capability and two motor coils can be connected directly to four pins. The motor rotation is designed to be controlled by a combination of the PWM pulse generators and selector logic circuits. The synchronization mechanism enables the synchronous operation of two PWM pulse generators.

22.2. Block Diagram of the Stepper Motor Controller

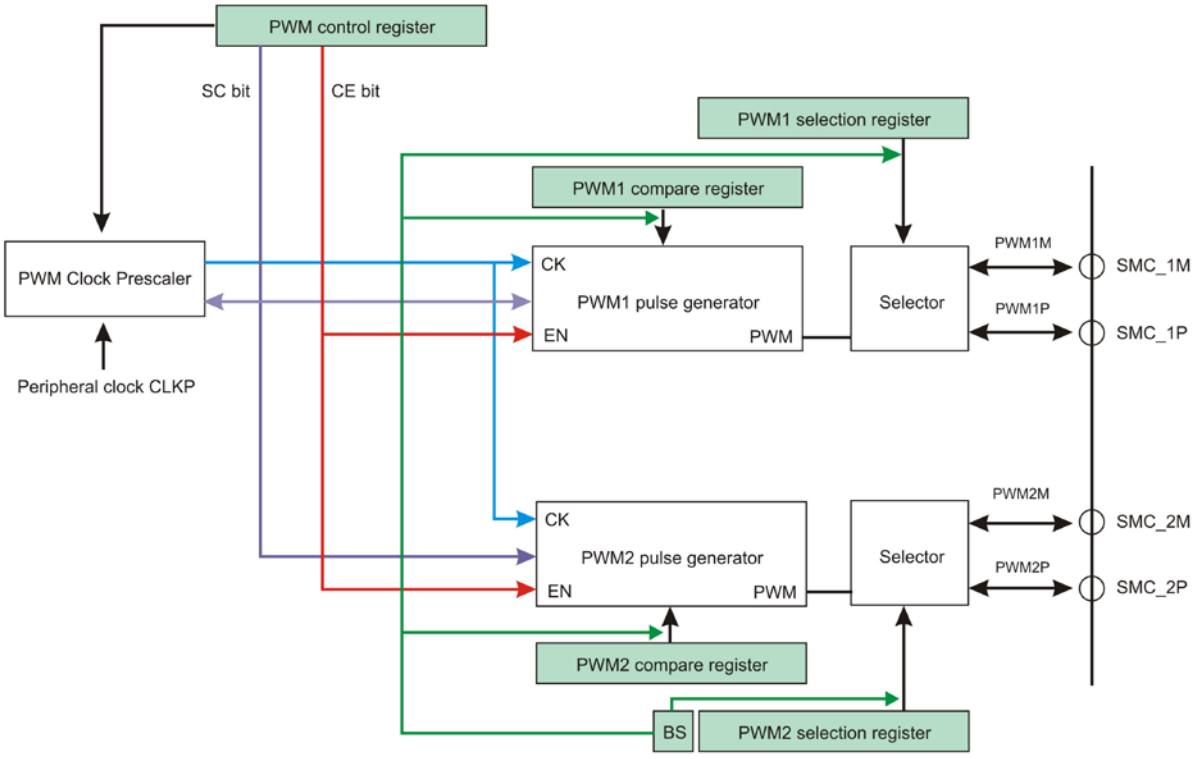


Figure 22-1 Stepper Motor Controller

22.3. Registers

There are five types of registers for the stepper motor controller:
PWM Control register
PWM1 Compare register
PWM2 Compare register
PWM1 Selection register
PWM2 Selection register

22.3.1. Register summary

PWM control register (PWC0, PWC1, PWC4*, PWC5*)
Addresses: 0x0C1, 0x0C3, 0x0C9*, 0x0CB*
* Only for use as General Purpose outputs (GPOs) and as SPWMs.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	P2	P1	P0	CE	SC	Reserved	Reserved
-	R/W	R/W	R/W	R/W	R/W	-	-
-	0	0	0	0	0	-	-

PWM1 compare register (PWC10, PWC11, PWC14, PWC15)
Addresses: 0x092, 0x09A, 0x0B2, 0x0BA, 0x093, 0x09B, 0x0B3, 0x0BB

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	D9	D8
-	-	-	-	-	-	R/W	R/W
-	-	-	-	-	-	X	X

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
X	X	X	X	X	X	X	X

PWM2 compare register (PWC20, PWC21, PWC24, PWC25)
Addresses: 0x090, 0x098, 0x0B0, 0x0B8, 0x091, 0x099, 0x0B1, 0x0B9

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	D9	D8
-	-	-	-	-	-	R/W	R/W
-	-	-	-	-	-	X	X

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
X	X	X	X	X	X	X	X

PWM1 selection register (PWS10, PWS11, PWS14, PWS15)
Addresses: 0x097, 0x09F, 0x0B7, 0x0BF

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	Reserved	P2	P1	P0	M2	M1	M0
-	-	R/W	R/W	R/W	R/W	R/W	R/W
-	-	0	0	0	0	0	0

PWM2 selection register (PWS20, PWS21, PWS24, PWS25)
Addresses: 0x096, 0x09E, 0x0B6, 0x0BE

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reserved	BS	P2	P1	P0	M2	M1	M0
-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
-	0	0	0	0	0	0	0

Figure 22-2 Stepper Motor Controller Register overview

22.3.2. PWM Control Register

The PWM control register starts/stops the stepper motor controller, performs interrupt control and sets external output pins, etc., for the stepper motor controller.

22.3.2.1. PWM Control Register

PWM control register (PWC0, PWC1, PWC4*, PWC5*)
Addresses: 0x0C1, 0x0C3, 0x0C9*, 0x0CB*
* Only for use as General Purpose outputs (GPOs) and as SPWMs.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	P2	P1	P0	CE	SC	Reserved	Reserved
-	R/W	R/W	R/W	R/W	R/W	-	-
-	0	0	0	0	0	-	-

[bit 7] Reserved bit
Always set the reserved bit to "0".
[bit 6 to 4] P2, P1, P0: Operating clock selection bits (bits to select the operating clock)
The P2, P1 and P0 bits specify the clock input signal for the PWM pulse generator.

P2	P1	P0	Clock input	PWM cycle (at Fcp = 41.66MHz)		PWM cycle (at Fcp = 20.83MHz)		PWM cycle (at Fcp = 10.42MHz)	
				SC=0	SC=1	SC=0	SC=1	SC=0	SC=1
0	0	0	FCP	6.1 us	24.6 us	12.3 us	49.2 us	24.3 us	97.3 us
0	0	1	FCP/4	24.6 us	98.3 us	49.2 us	196.6 us	97.3 us	389.1 us
0	1	0	FCP/5	30.7 us	122.9 us	61.4 us	245.8 us	121.6 us	486.4 us
0	1	1	FCP/6	36.9 us	147.5 us	73.7 us	294.9 us	145.9 us	583.7 us
1	0	0	FCP/8	49.2 us	196.6 us	98.3 us	393.2 us	194.6 us	778.2 us
1	0	1	FCP/10	61.4 us	245.8 us	122.9 us	491.5 us	243.2 us	972.8 us
1	1	0	FCP/12	73.7 us	294.9 us	147.5 us	589.8 us	291.8 us	1.2 ms
1	1	1	FCP/16	98.3 us	393.2 us	196.6 us	786.4 us	389.1 us	1.6 ms

FCP: Peripheral clock CLKP

Note:
After operation clock selection, set 1 in CE.

[bit 3] CE: Count Enable bit

The CE bit enables the operation of the PWM pulse generator. When "1" is set to the CE bit, the PWM pulse generator starts operation. The PWM2 pulse generator starts after the PWM1 pulse generator starts, in order to reduce the switching noise generated by the output driver.

If the CE bit is cleared to 0 during the operation of the PWM pulse generator, the generator's stop mode is initiated.

[bit 2] SC: 8/10 bits switching bit

If "1" is set to the SC bit, the PWM pulse generator operates in 10 bit mode. If "0" is set to the SC bit, the PWM pulse generator operates in 8 bit mode.

[bit 1 to 0] Reserved bits

Always set the reserved bits to "00".

22.3.3. PWM1&2 Compare Registers

The value of the two 8(10) bits comparison registers of PWM1&2 determine the width of the PWM pulse. A stored "00_H (000_H)" value indicates that the PWM duty cycle is 0%, and a stored "FF_H" ("$3FF_H$") value indicates that the PWM duty cycle is 99.6% (99.9%).

22.3.3.1. PWM1&2 Comparison Registers

The PWM1&2 comparison registers can be accessed at any time, but the changed value is reflected in the pulse width at the end of the current PWM cycle after "1" is set to the BS bit of the PWM2 selection register.
When "0" is set to the SC bit of the PWM control register, and PWM performs 8-bit operation, the D9 and D8 bits hold an undefined value.
Be sure to perform half-word access to the PWM1&2 comparison registers

PWM1 compare register (PWC10, PWC11, PWC14, PWC15)
Addresses: 0x092, 0x09A, 0x0B2, 0x0BA, 0x093, 0x09B, 0x0B3, 0x0BB

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	D9	D8
-	-	-	-	-	-	R/W	R/W
-	-	-	-	-	-	X	X

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
X	X	X	X	X	X	X	X

PWM2 compare register (PWC20, PWC21, PWC24, PWC25)
Addresses: 0x090, 0x098, 0x0B0, 0x0B8, 0x091, 0x099, 0x0B1, 0x0B9

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	D9	D8
-	-	-	-	-	-	R/W	R/W
-	-	-	-	-	-	X	X

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
X	X	X	X	X	X	X	X

[bit 15 to 10] Reserved bits
Always set reserved bits to "0".
[bits 9 to 0] D9 to D0: Compare data
These bits are used to set the PWM pulse width.

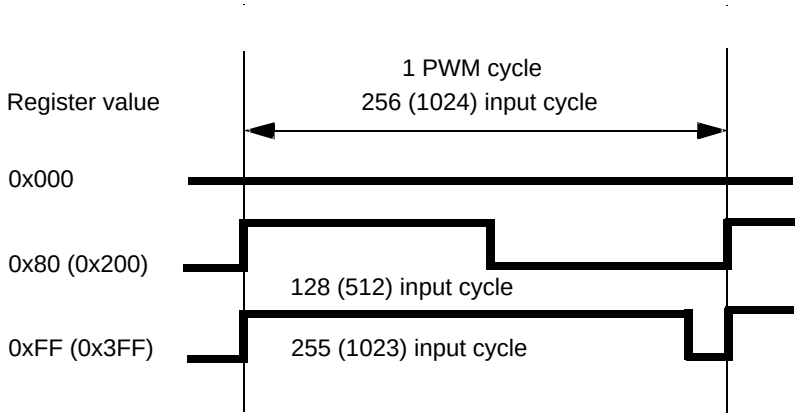


Figure 22-3 Relationship between the Comparison Register Setting Value and PWM Pulse Width

22.3.4. PWM1&2 Selection Registers

The PWM1&2 selection registers determine whether to set the output of the external pin of the stepper motor controller to "0", "1", PWM pulse or high impedance.

22.3.4.1. PWM1&2 Selection Registers

PWM1 selection register (PWS10, PWS11, PWS14, PWS15)
Addresses: 0x097, 0x09F, 0x0B7, 0x0BF

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	Reserved	P2	P1	P0	M2	M1	M0
-	-	R/W	R/W	R/W	R/W	R/W	R/W
-	-	0	0	0	0	0	0

PWM2 selection register (PWS20, PWS21, PWS24, PWS25)
Addresses: 0x096, 0x09E, 0x0B6, 0x0BE

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reserved	BS	P2	P1	P0	M2	M1	M0
-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
-	0	0	0	0	0	0	0

[bit 15] Reserved bit
Always set the reserved bit to "0".
[bit 14] BS: Rewrite bit
The BS bit makes the setting for the PWM output match another setting. Until the BS bit is set, changes made to the two comparison registers and two selection registers are not reflected in the output signal.
When "1" is set to the BS bit, the PWM pulse generator and the selector load the values of the registers at the end of the current PWM cycle. The BS bit is cleared to “0” automatically at the beginning of the next PWM cycle.
If "1" is set to the BS bit and it is being cleared automatically by software simultaneously, the BS bit remains set to “1” (no change is made to the BS bit) and the automatic clearing is cancelled.
If "0" is set to the BS bit and it is being cleared automatically by software simultaneously, the BS bit is cleared to “0”, however the PWM pulse generator and the selector do not load the values of the registers at

the end of the current PWM cycle.

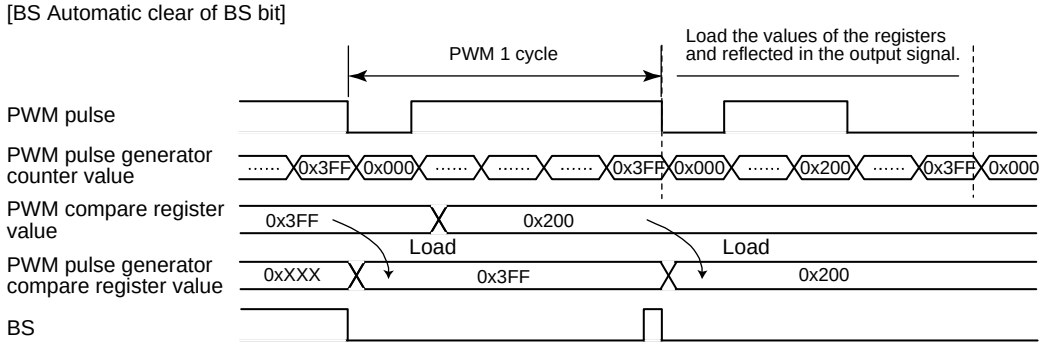
Note:

If the BS bit is “1” when executing a read-modify-write type instruction, the BS bit is "1" read as “1” and "1" is written to the BS bit again.

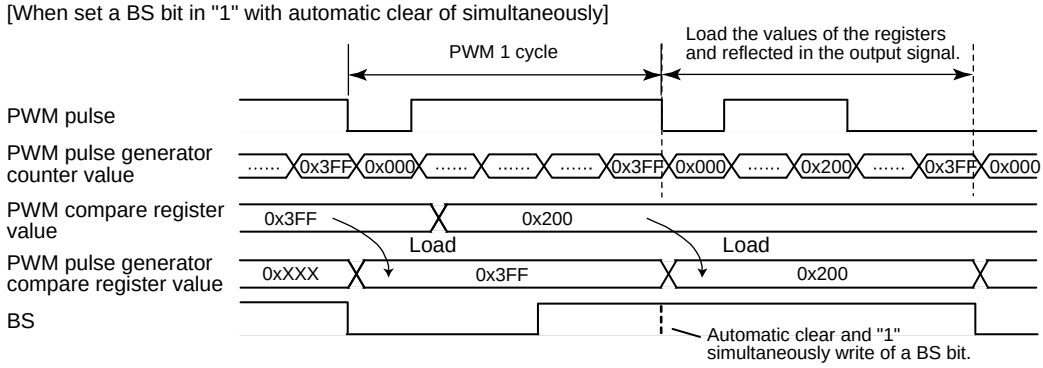
If the BS bit was cleared automatically by the start of a PWM cycle inbetween a read and write, "1" is set after BS bit has been cleared again.

Therefore, the values of the registers are loaded into the PWM pulse generator and selector when the BS bit is not set to "1" at the end of the next PWM cycle.

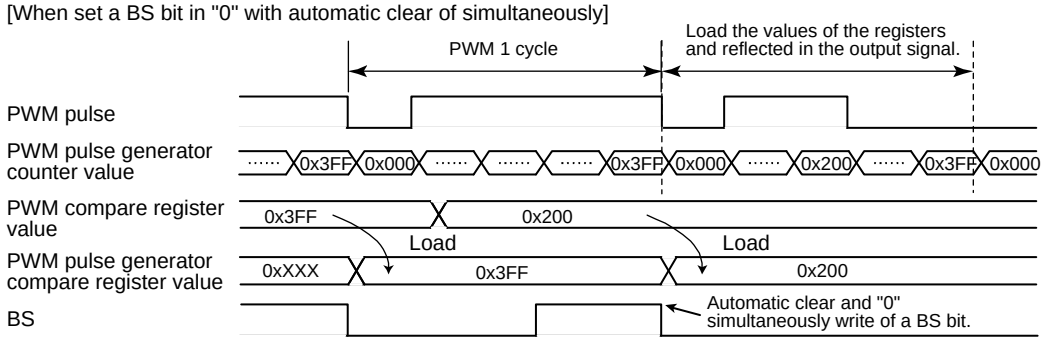
[BS Automatic clear of BS bit]



[When set a BS bit in "1" with automatic clear of simultaneously]



[When set a BS bit in "0" with automatic clear of simultaneously]



[When set a BS bit in "0" before the PWM cycle end]

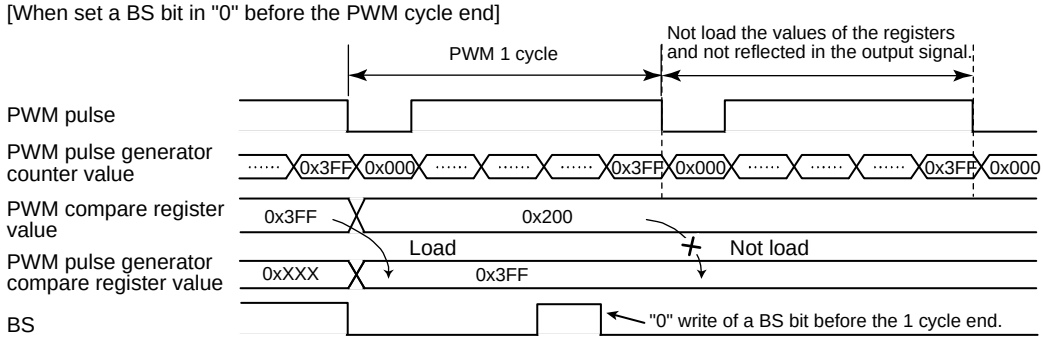


Figure 22-4 load timing of PWM compare register value

[bit 13 to 11] P2 to P0: Output select bits

These bits are used to select the output signal for SMC2P.

[bit 10 to 8] M2 to M0: Output select bits

These bits are used to select the output signal for SMC2M.

[bit 7 to 6] Reserved bit

Always set the reserved bit to "0".

[bit 5 to 3] P2 to P0: Output select bits

These bits are used to select the output signal for SMC1P.

[bit 2 to 0] M2 to M0: Output select bits

These bits are used to select the output signal for SMC1M.

The table below shows the relationship between the output levels and the select bits.

P2	P1	P0	PWMmPn	M2	M1	M0	PWMmMn
0	0	0	L	0	0	0	L
0	0	1	H	0	0	1	H
0	1	X	PWM Pulse	0	1	X	PWM Pulse
1	X	X	High impedance	1	X	X	High impedance

m = 1 to 2 (motor coils)

n = 0 to 2 (stepper motor channels)

22.4. Operation

The operation of the stepper motor controller is explained.

22.4.1. Setting the Operation of the Stepper Motor Controller

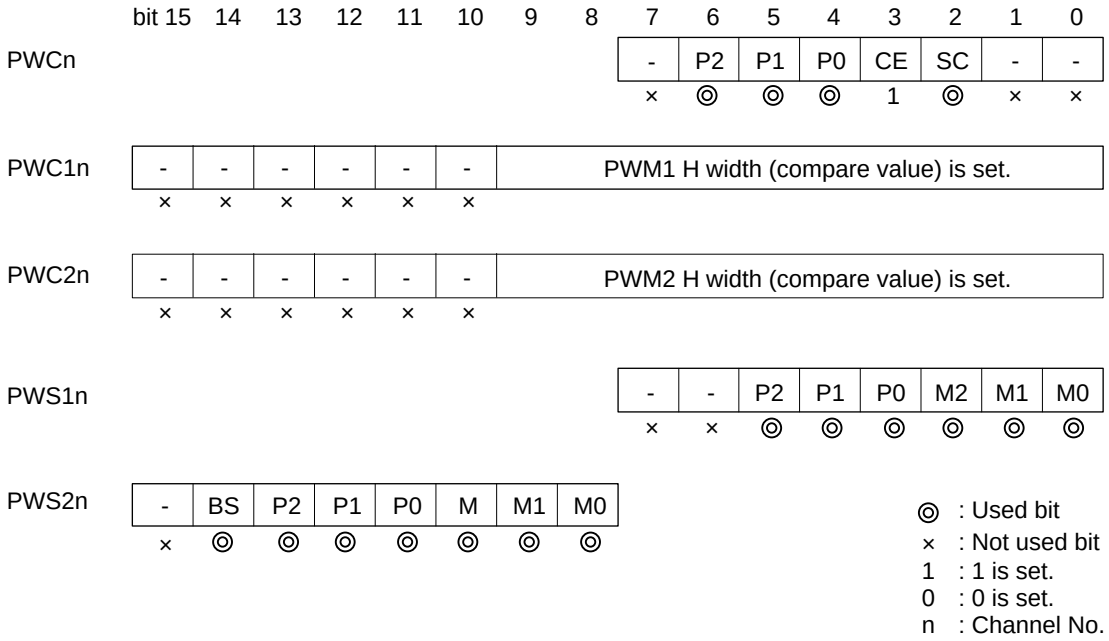


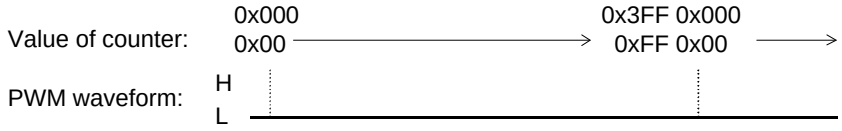
Figure 22-5 Setting of Stepper motor controller

22.4.2. Operation of PWM-pulse generator

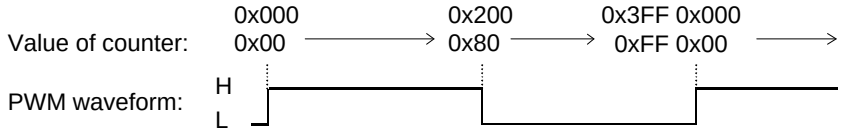
When the counter is started (PWC: CE = 1), the counter starts incrementing from 00H on the selected count clock rising. The PWM output pulse wave remains "H" until the value of the counter matches the value set to PWM compare register, and then changes to and remains "L" until the value of the counter overflows (FF_H --> 00_H).

"Examples of PWM1&2 Waveform Output" shows the PWM waveform generated by the PWM generator

When the value of compare register is "0x00"/"0x000" (duty ratio is 0%):



When the value of compare register is "0x80"/"0x200" (duty ratio is 50%):



When the value of compare register is "0xFF"/"0x3FF" (duty ratio is 99.6%/99.9%):

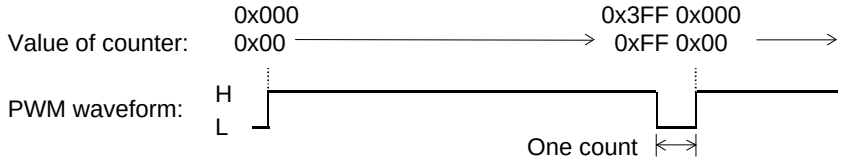


Figure 22-6 Examples of PWM1&2 Waveform Output

22.4.3. Selection of motor drive signals

Motor drive signals that are output to each pin related to the stepper motor controller can be selected among four types of signals for each pin by setting the PWM selection register.
"Selection of Motor Drive Signals and Setting of PWM Selection Registers 1&2" lists the selection of the motor drive signals and the settings of PWM selection registers 1&2.
When these registers are set and "1" is written to the BS bit of the PWM selection register 2, the setting of these registers is enabled at the end of the current PWM cycle. The BS bit is to be cleared automatically to 0 at the beginning of the next PWM cycle. When "1" is written to the BS bit, and the BS bit is cleared to 0 simultaneously at the beginning of the next PWM cycle, "1" is written to the BS bit and clearing of the BS bit is cancelled.

Table 22-1 Selection of Motor Drive Signals and Setting of PWM Selection Registers 1&2

P2, P1, P0 Bits	PWM2P Output PWM1P Output	M2, M1, M0 Bits	PWM1M Output PWM2M Output
000B	L	000B	L
001B	H	001B	H
01XB	PWM Pulse	01XB	PWM Pulse
1XXB	High impedance	1XXB	High impedance

22.5. Caution

The caution when using the stepper motor controller are described below.

22.5.1. Caution when Changing PWM Setting

The PWM comparison registers 1&2 (PWC1, PWC2) and the PWM selection registers 1&2 (PWS1, PWS2) can be accessed at any time. However, to change setting of the "H" width of PWM or to change the PWM output, "1" must be written to the BS bit of the PWM2 selection register after (or and at the same

time) a setting is written to those registers (the PWM comparison registers 1&2 and the PWM selection registers 1&2).
When "1" is set to the BS bit, the new setting is enabled at the end of the current PWM cycle and the BS bit is cleared automatically.
Also, when "1" is written to the BS bit and the BS bit is reset at the end of the PWM cycle simultaneously, "1" is written to the BS and the reset of the BS bit is cancelled.

23. SOUND Generator

This chapter describes the SOUND Generator of the MB88F333.

23.1. Outline

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

The Sound Generator consists of the Sound Control register, Frequency Data register, Amplitude Data register, Decrement Grade register, Tone Count register, Sound Disable register, PWM pulse generator, Frequency counter, Decrement counter and Tone Pulse counter.

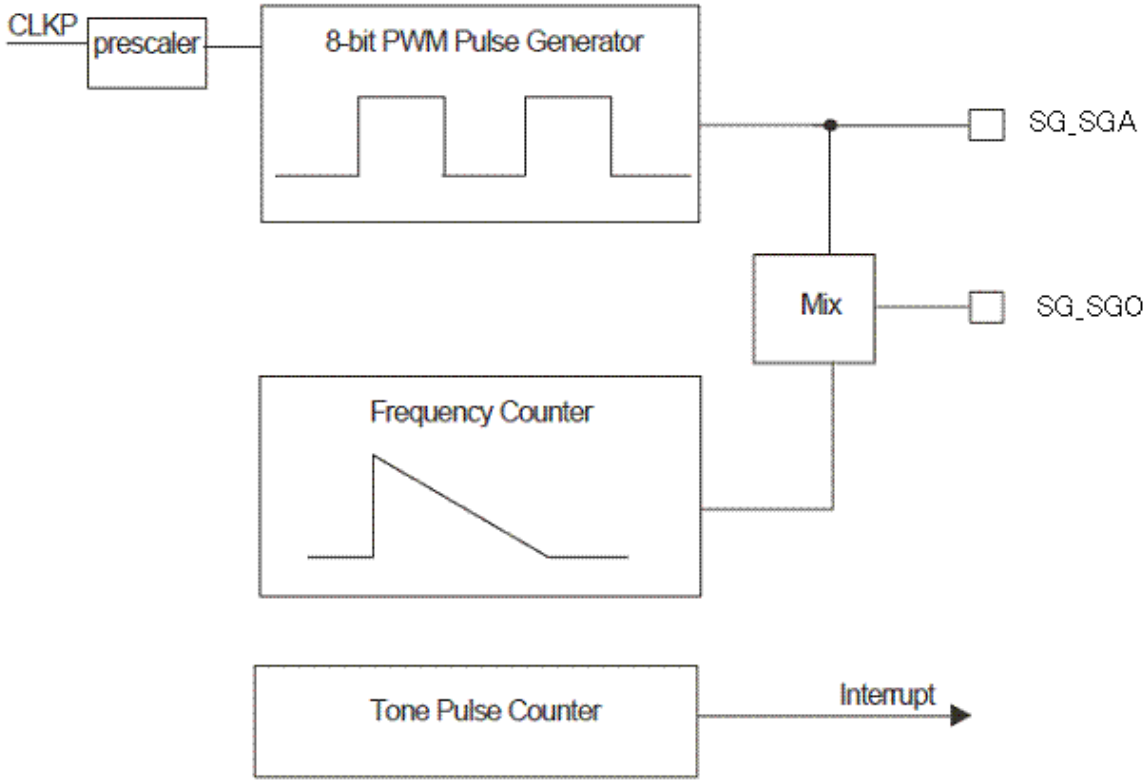


Figure 24-1 Block diagram of the sound generator

23.2. Detailed Structure of the Sound generator

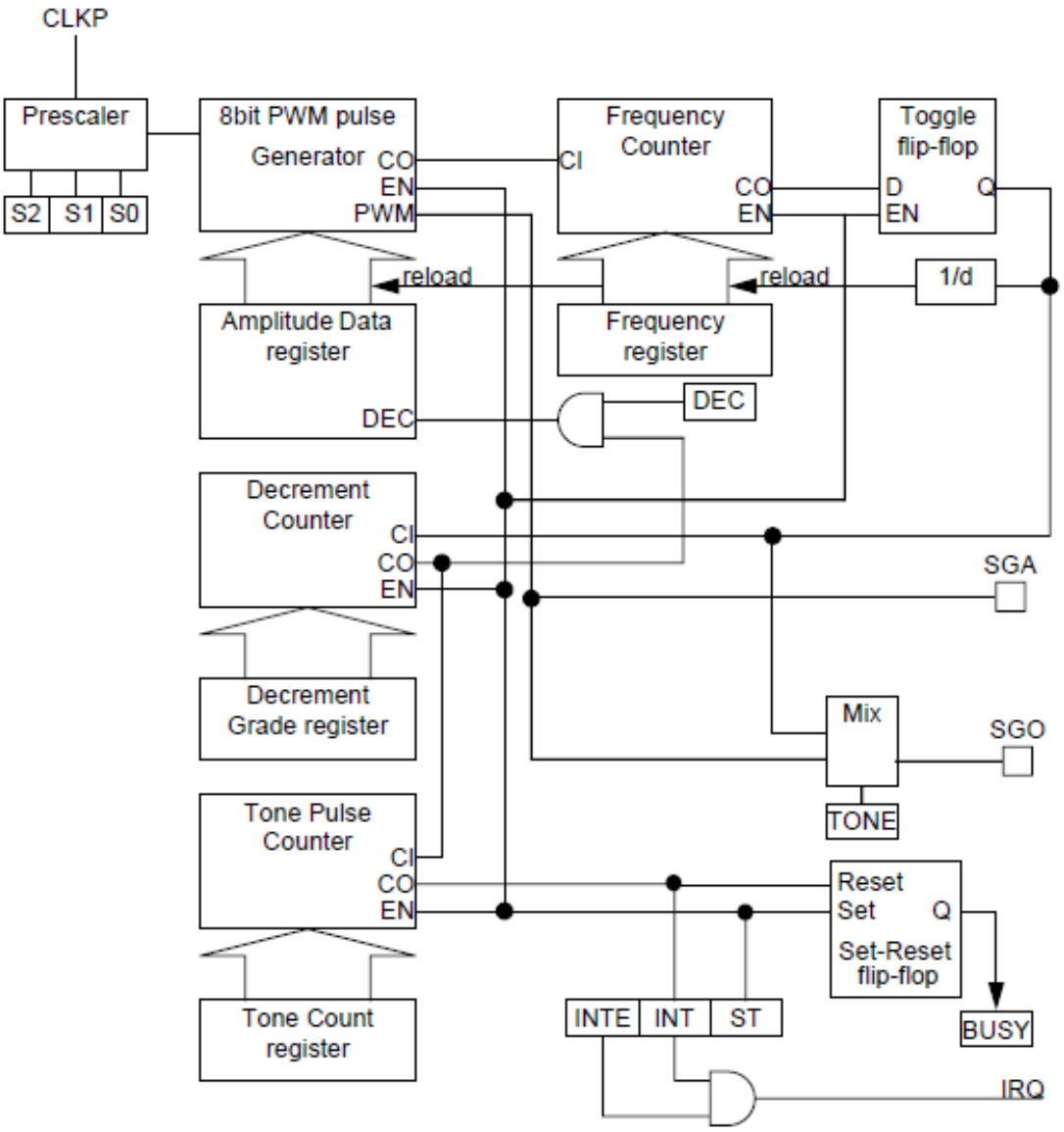


Figure 24-2 Detailed Structure of the sound generator

23.3. Registers

	Register address		Base Address + 0x198							
Bit number	15	14	13	12	11	10	9	8		
Bit field name	D15	D14	D13	D12	D11	D10	D9	D8		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		
	Initial value		X	X	X	X	X	X	X	X
	Register address		Base Address + 199h							
	Bit number		7	6	5	4	3	2	1	0
Bit field name	D7	D6	D5	D4	D3	D2	D1	D0		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		
Initial value	X	X	X	X	X	X	X	X		

Table 24-1 Frequency Data register (SGFR)

	Register address		Base Address + 0x19A					
Bit number	15	14	13	12	11	10	9	8
Bit field name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

Table 24-2 Amplitude Data register (SGAR)

	Register address		Base Address + 0x19C					
Bit number	15	14	13	12	11	10	9	8
Bit field name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	X	X	X	X	X	X	X	X

Table 24-3 Tone Count Register (SGTR)

	Register address		Base Address + 0x19D					
Bit number	7	6	5	4	3	2	1	0
Bit field name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	X	X	X	X	X	X	X	X

Table 24-4 Decrement Grade Register (SGDR)

	Register address		Base Address + 0x19E					
Bit number	15	14	13	12	11	10	9	8
Bit field name	TST	S2	S1	S0	-	-	BUSY	DEC
R/W	R/W	R/W	R/W	R/W	-	-	R	R/W
Initial value	0	0	0	0	-	-	0	0

Table 24-5 Sound Control Register (SGCRH)

	Register address		Base Address + 0x19F					
Bit number	7	6	5	4	3	2	1	0
Bit field name	-	-	TONE	-	-	INTE	INT	ST
R/W	-	-	R/W	-	-	R/W	R/W	R/W
Initial value	-	-	0	-	-	0	0	0

Table 24-6 Sound Control Register (SGCRL)

23.3.1. Register Description

23.3.1.1. Frequency Data Register (SGFR)

Register address		Base Address + 0x198						
Bit number		15	14	13	12	11	10	9	8
Bit field name		D15	D14	D13	D12	D11	D10	D9	D8
R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value		X	X	X	X	X	X	X	X
Register address		Base Address + 0x199							
Bit number		7	6	5	4	3	2	1	0
Bit field name		D7	D6	D5	D4	D3	D2	D1	D0
R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value		X	X	X	X	X	X	X	X

The Frequency Data register stores the reload value for the Frequency counter. The stored value represents the frequency of the sound (or the tone signal from the toggle flip-flop). The register value is reloaded into the counter at every transition of the toggle signal.
Figure24-3 shows the relationship between the register value and the tone signal.

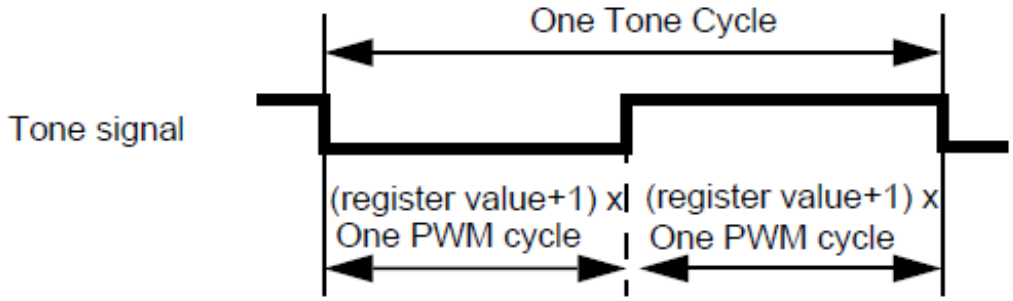


Figure 24-3 The relationship between the register value and the tone signal

It should be noted that modifications of the register value during operation may alter the duty cycle of 50% depending on the timing of the modification.

23.3.1.2. Amplitude Data Register (SGAR)

Register address	Base Address + 0x19A							
Bit number	15	14	13	12	11	10	9	8
Bit field name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

Table 24-7 Amplitude Data register (SGAR)

The Amplitude Data register stores the reload value for the PWM pulse generator. The register value represents the amplitude of the sound. The register value is reloaded into the PWM pulse generator at the end of every tone cycle.
When the DEC bit is "1" and the Decrement counter reaches its reload value, this register value is decremented by 1(one). When the register value reaches "00", further decrements are not performed, however the sound generator continues its operation until the ST bit is cleared.

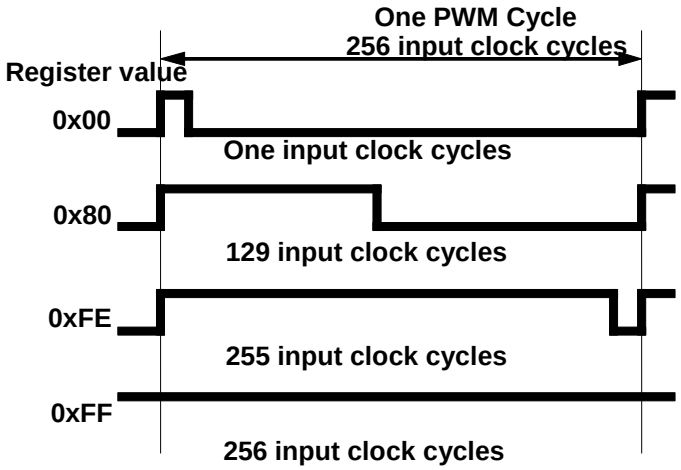


Figure 24-4 Relationship between the register value and the PWM pulse

When the register value is set to "FF", the PWM signal is always "1".

23.3.1.3. Tone Count Register (SGTR)

Register address	Base Address + 0x19C							
Bit number	15	14	13	12	11	10	9	8
Bit field name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	X	X	X	X	X	X	X	X

Table 24-8 Tone Count Register (SGTR)

The Tone Count register stores the reload value for the Tone Pulse counter. The Tone Pulse counter accumulates the number of tone pulses (or number of decrement operations) and when it reaches the reload value it sets the INT bit. They are intended to reduce the frequency of interrupts.

The count input of the Tone Pulse counter is connected to the carry-out signal from the Decrement counter. When the Tone count register is set to "00", the Tone Pulse counter sets the INT bit every carry-out from the Decrement counter. Thus the number of accumulated tone pulses is:

((Decrement Grade register) +1) x ((Tone Count register) +1)

i.e. when the both registers are set to "00", the INT bit is set every tone cycle.

23.3.1.4. Decrement Grade Register (SGDR)

Register address	Base Address + 0x19D							
Bit number	7	6	5	4	3	2	1	0
Bit field name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	X	X	X	X	X	X	X	X

Table 24-9 Decrement Grade Register (SGDR)

The Decrement Grade register stores the reload value for the Decrement counter. It is prepared to automatically decrement the stored value in the Amplitude Data register.

When the DEC bit is "1" and the Decrement counter counts the number of tone pulses up to the reload value, the stored value in the Amplitude Data register is decremented by 1 (one) at the end of the tone cycle.

This operation realizes automatic decrement of the sound with less CPU interventions.

It should be noted that the number of the tone pulses specified by this register equals to "register value +1".

When the Decrement Grade register is set to "00", the decrement operation is performed every tone cycle.

23.3.1.5. Sound Control Register (SGCR)

Register address	Base Address + 0x19E							
Bit number	15	14	13	12	11	10	9	8
Bit field name	TST	S2	S1	S0	-	-	BUSY	DEC
R/W	R/W	R/W	R/W	R/W	-	-	R	R/W
Initial value	0	0	0	0	-	-	0	0

Table 24-10 Sound Control Register (SGCRH)

Register address	Base Address + 0x19F							
Bit number	7	6	5	4	3	2	1	0
Bit field name	-	-	TONE	-	-	INTE	INT	ST
R/W	-	-	R/W	-	-	R/W	R/W	R/W
Initial value	-	-	0	-	-	0	0	0

Table 24-11 Sound Control Register (SGCRL)

[bit 15] TST: Test bit

This bit is available for device testing. In a user application it should be set to "0".

[bits14 to12] S2 to S0: Operation clock select bits

These bits specify the clock input signal for the Sound Generator.

S2	S1	S0	Clock input
0	0	0	CLK
0	0	1	1/2 CLK
0	1	0	1/4 CLK
0	1	1	1/8 CLK
1	0	0	1/16 CLK

[bit 9] BUSY: Busy bit

This bit indicates whether the Sound Generator is in operation. This bit is set to "1" upon the ST bit is set to "1". It is reset to "0" when the ST bit is reset to "0" and the operation is completed at the end of one tone cycle. Any write instruction performed on this bit has no effect.

[bit 8] DEC: Auto-decrement enable bit

The DEC bit is prepared for an automatic decrement of the sound in conjunction with the Decrement Grade register. If this bit is set to "1", the stored value in the Amplitude Data register is decremented by 1(one), every time when the Decrement counter counts the number of tone pulses from the toggle flip-flop specified by the Decrement Grade register.

[bit 5] TONE: Tone output bit

When this bit is set to "1", the SGO signal becomes a simple square-waveform (tone pulses) created from the toggle flip-flop. Otherwise it is the mixed (AND logic) signal of the tone and PWM pulses.

[bit 2] INTE: Interrupt enable bit

This bit enables the interrupt signal of the Sound Generator. When this bit is "1" and the INT bit is set to "1", the Sound Generator signals an interrupt.

[bit 1] INT: Interrupt bit

This bit is set to "1" when the Tone Pulse counter counts the number of the tone pulses specified by the Tone Count register and Decrement Grade register. This bit is reset by writing a "0" to it. Writing "1" has no effect and Read-Modify-Write instructions always result in reading "1".

[bit 0] ST: Start bit

This bit is for starting the operation of the Sound Generator. While this bit is "1", the Sound Generator performs its operation.

When this bit is reset to "0", the Sound Generator stops its operation at the end of the current tone cycle. The BUSY bit indicates whether the Sound Generator is fully stopped.

24. I²C Controller

This chapter describes the I²C Controller of the MB88F333.

24.1. Outline

The I²C interface is a serial I/O port which supports the Inter-Integrated Circuit bus and operates as a master/slave device on it.

24.2. Features

The I²C interface has following features.

- Master/slave transmitting and receiving functions
- Arbitration function
- Clock synchronization function
- General call addressing support
- Transfer direction detection function
- Repeated start condition generation and detection function
- Bus error detection function
- 7 bit addressing as master and slave
- 10 bit addressing as master and slave
- Possibility to give the interface a seven *and* a ten bit slave address
- Acknowledge upon slave address reception can be disabled (master only operation)
- Address masking to give interface several slave addresses (in 7 and 10 bit mode)
- Up to 400 Kbit/s transfer rate
- Possibility to use built-in noise filters for SDA and SCL
- Can receive data at 400 Kbit/s if the R-Bus-Clock is higher than 6MHz, regardless of prescaler setting
- Can generate MCU interrupts on transmission and bus error events
- Supports being slowed down (deceleration) by a slave at a bit and byte level

The I²C interface does not support SCL clock stretching at a bit level as it can receive the full 400 Kbit/s data rate if the R-Bus-Clock (CLKP) is higher than 6MHz regardless of the prescaler setting. However, clock stretching at a byte level is performed because SCL is pulled low during an interrupt (INT='1' in the IBCR2 register).

24.3. Block Diagram

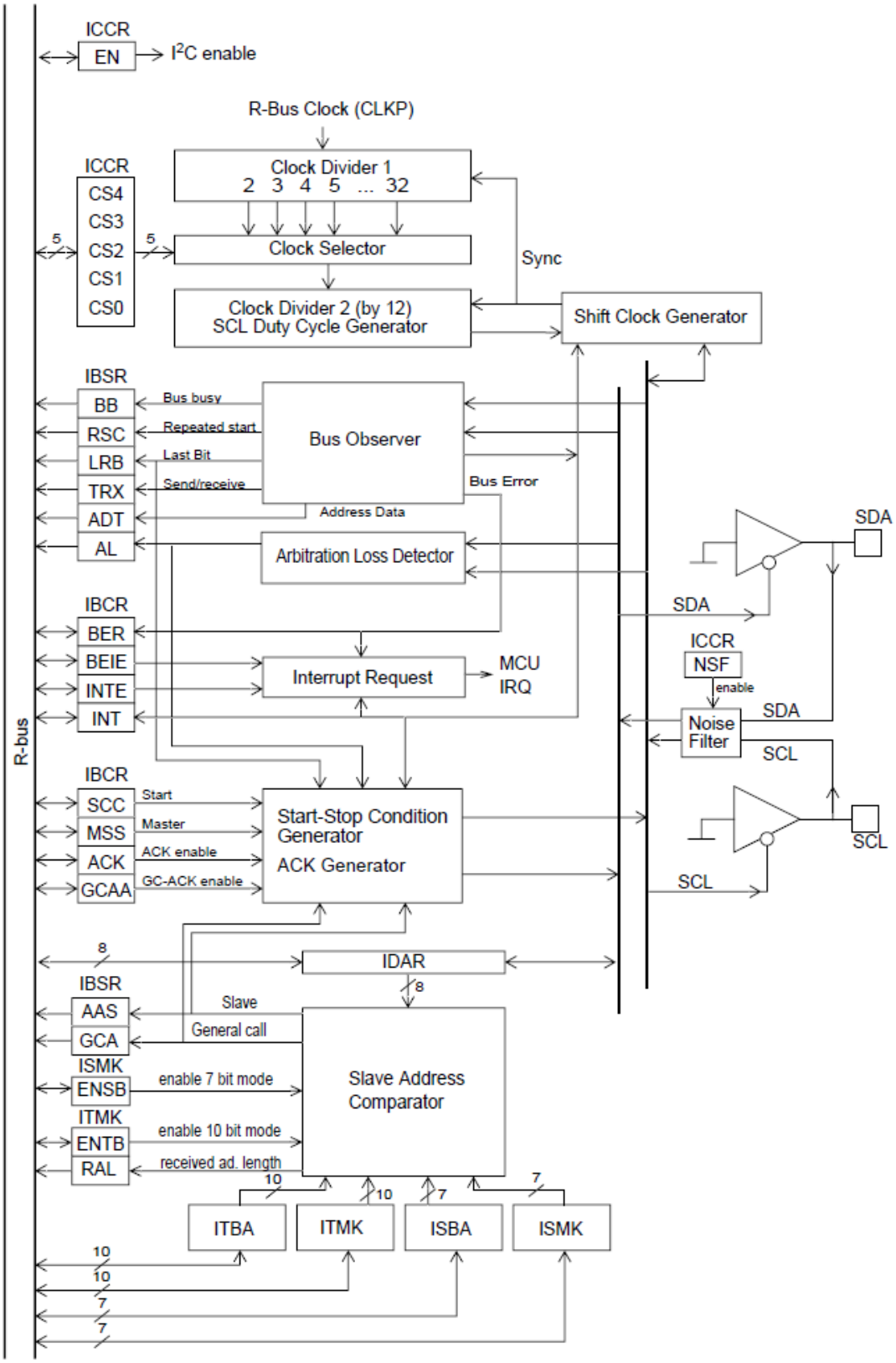


Figure 25-1 Block diagram of I²C Controller

24.4. I²C Interface Registers

This section describes the function of the I²C interface registers in detail.

Bus control register	15	14	13	12	11	10	9	8	⇐ Bit no.
Address : 0000D0H	BER	BEIE	SCC	MSS	ACK	GCAA	INTE	INT	IBCR0
Read/write ⇒	(R/W)	(R/W)	(W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

Figure 25-2 Bus Control Register (IBCR0)

Bus status register	7	6	5	4	3	2	1	0	⇐ Bit no.
Address : 0000D1H	BB	RSC	AL	LRB	TRX	AAS	GCA	ADT	IBSR0
Read/write ⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

Figure 25-3 Bus Status Register (IBSR0)

Ten Bit Address high byte	15	14	13	12	11	10	9	8	⇐ Bit no.
Address : 0000D2H	---	---	---	---	---	---	TA9	TA8	ITBAH0
Read/write ⇒	(-)	(-)	(-)	(-)	(-)	(-)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

Ten Bit Address low byte	7	6	5	4	3	2	1	0	⇐ Bit no.
Address : 0000D3H	TA7	TA6	TA5	TA4	TA3	TA2	TA1	TA0	ITBAL0
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

Figure 25-4 Ten Bit slave Address register (ITBA0)

Ten Bit Address Mask high byte	15	14	13	12	11	10	9	8	⇐ Bit no.
Address : 0000D4H	ENTB	RAL	---	---	---	---	TM9	TM8	ITMKH0
Read/write ⇒	(R/W)	(R)	(-)	(-)	(-)	(-)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(1)	(1)	(1)	(1)	(1)	(1)	

Ten Bit Address Mask low byte	7	6	5	4	3	2	1	0	⇐ Bit no.
Address : 0000D5H	TM7	TM6	TM5	TM4	TM3	TM2	TM1	TM0	ITMKL0
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(1)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	

Figure 25-5 Ten bit slave address MasK register (ITMK0)

Seven Bit Address Mask register	15	14	13	12	11	10	9	8	⇐ Bit no.
Address : 0000D6H	ENSB	SM6	SM5	SM4	SM3	SM2	SM1	SM0	ISMK0
Read/write ⇒	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(1)	(1)	(1)	(1)	(1)	(1)	(1)	

Figure 25-6 Seven bit slave address MasK register (ISMK0)

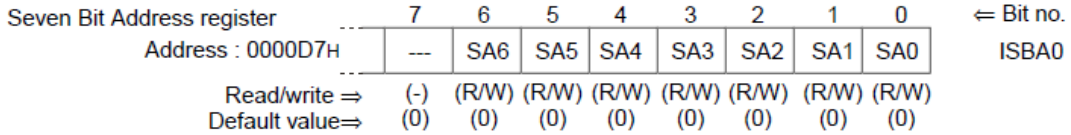


Figure 25-7 Seven Bit slave Address register (ISBA0)

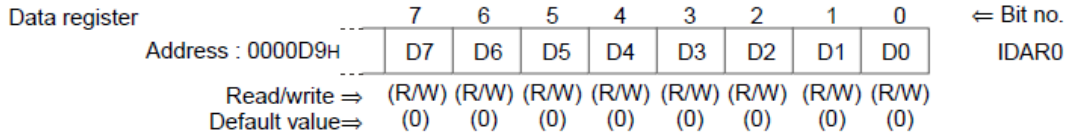


Figure 25-8 Data Register (IDAR0)

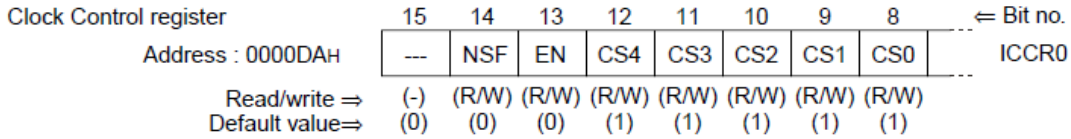


Figure 25-9 Clock control register (ICCR0)

24.4.1. Bus Control Register (IBCR0)

The bus control register (IBCR0) has the following functions:

- Interrupt enabling flags
- Interrupt generation flag
- Bus error detection flag
- Repeated start condition generation
- Master / slave mode selection
- General call acknowledge generation enabling
- Data byte acknowledge generation enabling

Write access to this register should only occur during INT='1' or if a transfer is to be started. The user should not write to this register during an ongoing transfer as changes to the ACK or GCAA bits could result in bus errors. All the bits in this register (except the BER and the BEIE bits) are cleared if the interface is not enabled (EN='0' in ICCR0).

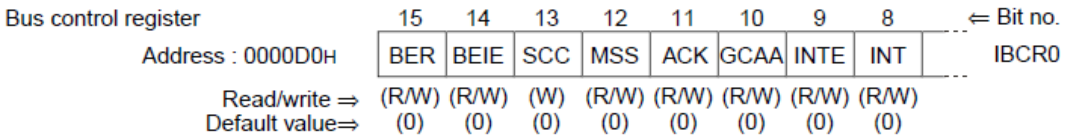


Figure 25-10 Bus Control Register (IBCR0)

[bit 15] BER (Bus ERror)

This bit is the bus error interrupt flag. It is set by hardware and cleared by the user application. It always reads '1' in a Read-Modify-Write access.

(Write access)

0	Clear bus error interrupt flag.
1	No effect.

(Read access)

0	No bus error detected.
1	One of the error conditions described below detected.

When this bit is set, the EN bit in the ICCR0 register is cleared, the I^2C interface enters pause status, data transfer is interrupted and all the bits in the IBSR0 and the IBCR0 registers except BER, BEIE and INT are cleared. *The BER bit must be cleared before the interface can be re-enabled.*

This bit is set to '1' if:

- start or stop conditions are detected at wrong points in time: during an address data transfer or during the transfer of the bits two to nine (acknowledge bit)
- a ten bit address header with read access is received before a ten bit write access
- a stop condition is detected while the interface is in master mode

The detection of the first two of the above conditions is enabled after the reception of the first stop condition to prevent false bus error reports if the interface is being enabled during an ongoing transfer.

[bit 14] BEIE (Bus Error Interrupt Enable)

This bit enables the bus error interrupt. It can only be changed by the user application.

0	Bus error interrupt disabled.
1	Bus error interrupt enabled.

Setting this bit to '1' enables MCU interrupt generation when the BER bit is set to '1'.

[bit 13] SCC (Start Condition Continue)

This bit is used to generate a repeated start condition. It is write only - it always reads '0'.

0	No effect.
1	Generate repeated start condition during master transfer.

A repeated start condition is generated if a '1' is written to this bit during an interrupt in master mode (MSS='1' and INT='1') and the INT bit is cleared automatically.

[bit 12] MSS (Master Slave Select)

This is the master/slave mode selection bit. It can only be set by the user, but it can be cleared by the user and the hardware.

0	Go to slave mode.
1	Go to master mode, generate start condition and send address data byte in IDAR0 register.

It is cleared if an arbitration loss event occurs during master sending.

If a '0' is written to it during a master interrupt (MSS='1' and INT='1'), the INT bit is cleared automatically, a stop condition will be generated and the data transfer ends. Note that the MSS bit is reset immediately, the generation of the stop condition can be checked by polling the BB bit in the IBSR0 register.

If a '1' is written to it while the bus is idle (MSS='0' and BB='0'), a start condition is generated and the contents of the IDAR0 register (which should be address data) is sent.

If a '1' is written to the MSS bit while the bus is in use (BB='1' and TRX='0' in IBSR0; MSS='0' in IBCR0), the interface waits until the bus is free and then starts sending.

If the interface is addressed as slave with write access (data *reception*) in the meantime, it will start sending after the transfer ended and the bus is free again. If the interface is *sending* data as slave in the meantime (AAS='1' and TRX='1' in IBSR0), it will not start sending data if the bus is free again. It is important to check whether the interface was addressed as slave (AAS='1' in IBSR0) and sent the data byte successfully (MSS='1' in IBCR0) or failed to send the data byte (AL='1' in IBSR0) at the next interrupt!

[bit 11] ACK (ACKnowledge)

This is the acknowledge generation on *data* byte reception enable bit. It can only be changed by the user.

0	The interface will not acknowledge on data byte reception.
1	The interface will acknowledge on data byte reception.

This bit is not valid when receiving *address* bytes in slave mode - if the interface detects its 7 or 10 bit slave address, it will acknowledge if the corresponding enable bit (ENTB in ITMK0 or ENSB in ISMK0) is set.

Write access to this bit should occur during an interrupt (INT='1') or if the bus is idle (BB='0' in the IBSR0 register) write access to this bit is only possible if the interface is enabled (EN='1' in ICCR0) and if there is no bus error (BER='0' in IBCR0).

[bit 10] GCAA (General Call Address Acknowledge)

This bit enables acknowledge generation when a general call address is received. It can only be changed by the user.

0	The interface will not acknowledge on general call address byte reception.
1	The interface will acknowledge on general call address byte reception.

Write access to this bit should occur during an interrupt (INT='1') or if the bus is idle (BB='0' in IBSR0 register) write access to this bit is only possible if the interface is enabled (EN='1' in ICCR0) and if there is no bus error (BER='0' in IBCR0).

[bit 9] INTE (INTerrupt Enable)

This bit enables the MCU interrupt generation. It can only be changed by the user.

0	Interrupt disabled.
1	Interrupt enabled.

Setting this bit to '1' enables MCU interrupt generation when the INT bit is set to '1' (by the hardware).

[bit 8]: INT (INTerrupt)

This bit is the transfer end interrupt request flag. It is changed by the hardware and can be cleared by the user. It always reads '1' in a Read-Modify-Write access.

(Write access)

0	Clear transfer end interrupt request flag.
1	No effect.

(Read access)

0	Transfer not ended or not involved in current transfer or bus is idle.
1	Set at the end of a 1-byte data transfer or reception including the acknowledge bit under the following conditions: • Device is bus master. • Device is addressed as slave. • General call address received. • Arbitration loss occurred. Set at the end of an address data reception (after first byte if seven bit address received, after second byte if ten bit address received) including the acknowledge bit if the device is addressed as slave.

While this bit is '1' the SCL line will hold to an 'L' level signal. Writing '0' to this bit clears the setting, releases the SCL line, and executes the transfer of the next byte or a repeated start or stop condition is generated. Additionally, this bit is cleared if a '1' is written to the SCC bit or the MSS bit is being cleared.

SCC, MSS And INT Bit Competition

Simultaneously writing to the SCC, MSS and INT bits causes a conflict of events: to transfer the next byte, to generate a repeated start condition or to generate a stop condition. In these cases the order of priority is as follows:

Next byte transfer and stop condition generation.

When '0' is written to the INT bit and '0' is written to the MSS bit, the MSS bit takes priority and a stop condition is generated.

Next byte transfer and start condition generation.

When '0' is written to the INT bit and '1' is written to the SCC bit, the SCC bit takes priority. A repeated start condition is generated and the content of the IDAR0 register is sent.

Repeated start condition generation and stop condition generation.

When '1' is written to the SCC bit and '0' to the MSS bit, the MSS bit clearing takes priority. A stop condition is generated and the interface enters slave mode.

24.4.2. Bus Status Register (IBSR0)

The bus status register (IBSR0) has the following functions:

- Bus busy detection
- Repeated start condition detection
- Arbitration loss detection
- Acknowledge detection
- Data transfer direction indication
- Addressing as slave detection
- General call address detection
- Address data transfer detection

This register is read-only, all bits are controlled by the hardware. All bits are cleared if the interface is not enabled (EN='0' in ICCR0).

Bus status register	7	6	5	4	3	2	1	0	⇐ Bit no.
Address : 0000D1H	BB	RSC	AL	LRB	TRX	AAS	GCA	ADT	IBSR0
Read/write ⇒	(R)	(R)	(R)	(R)	(R)	(R)	(R)	(R)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

Figure 25-11 Bus Status Register (IBSR0)

[bit 7] BB (Bus Busy)

This bit indicates the status of the I^2C bus.

0	Stop condition detected (bus idle).
1	Start condition detected (bus in use).

This bit is set to '1' if a start condition is detected. It is reset upon a stop condition.

[bit 6] RSC (Repeated Start Condition)

This bit indicates detection of a repeated start condition.

0	Repeated start condition not detected.
1	Bus in use, repeated start condition detected.

This bit is cleared at the end of an address data transfer (ADT='0') or detection of a stop condition.

[bit 5] AL (Arbitration Loss)

This bit indicates an arbitration loss.

0	No arbitration loss detected.
1	Arbitration loss occurred during master sending.

This bit is cleared by writing '0' to the INT bit or by writing '1' to the MSS bit in the IBCR0 register.

An arbitration loss occurs if:

- the data sent does not match the data read on the SDA line at the rising SCL edge
- a repeated start condition is generated by another master in the first bit of a data byte
- the interface could not generate a start or stop condition because another slave pulled the SCL line low before

[bit 4] LRB (Last Received Bit)

This bit is used to store the acknowledge message from the receiving side at the transmitter side.

0	Receiver acknowledged.
1	Receiver did not acknowledge.

It is changed by the hardware upon reception of bit 9 (acknowledge bit) and is also cleared by a start or stop condition.

[bit 3] TRX (Transferring data)

This bit indicates data sending operation during data transfer.

0	Not transmitting data.
1	Transmitting data.

It is set to ‘1’:

- if a start condition was generated in master mode at the end of a first byte transfer and read access as slave or sending data as master

It is set to ‘0’ if:

- the bus is idle (BB=‘0’ in IBCR0)
- an arbitration loss occurred
- a ‘1’ is written to the SCC bit during master interrupt (MSS=‘1’ and INT=‘1’)
- the MSS bit is cleared during master interrupt (MSS=‘1’ and INT=‘1’)
- the interface is in slave mode and the last transferred byte was not acknowledged
- the interface is in slave mode and it is receiving data
- the interface is in master mode and is reading data from a slave

[bit 2] AAS (Addressed As Slave)

This bit indicates detection of a slave addressing.

0	Not addressed as slave.
1	Addressed as slave.

This bit is cleared by a (repeated-) start or stop condition. It is set if the interface detects its seven and/or ten bit slave address.

[bit 1] GCA (General Call Address)

This bit indicates detection of a general call address (0x00).

0	General call address not received as slave.
1	General call address received as slave.

This bit is cleared by a (repeated-) start or stop condition.

[bit 0] ADT (Address Data Transfer)

This bit indicates the detection of an address data transfer.

0	Incoming data is not address data (or bus is not in use).
1	Incoming data is address data.

This bit is set to ‘1’ by a start condition. It is cleared after the second byte if a ten bit slave address header with write access is detected, else it is cleared after the first byte.

"After" the first/second byte means:

- a ‘0’ is written to the MSS bit during a master interrupt (MSS=‘1’ and INT=‘1’ in IBCR0)
- a ‘1’ is written to the SCC bit during a master interrupt (MSS=‘1’ and INT=‘1’ in IBCR0)
- the INT bit is being cleared
- the beginning of every byte transfer if the interface is not involved in the current transfer as master or slave

24.4.3. Ten Bit Slave Address Register (ITBA0)

This register (ITBAH0 / ITBAL0) designates the ten bit slave address.
Write access to this register is only possible if the interface is disabled (EN='0' in ICCR0).

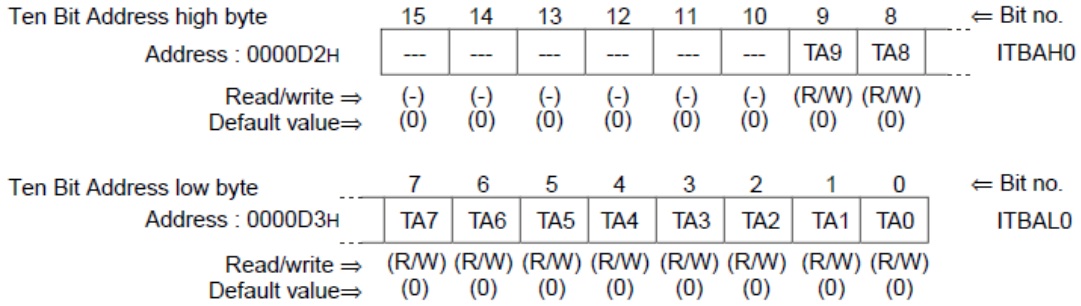


Figure 25-12 Ten Bit Slave Address Register (ITBA0)

[bit 15] - [bit 10] Not used.
These bits always read '0'.

[bit 9] - [bit 0] TBA - Ten Bit slave Address (TA9-TA0)
When address data is received in slave mode, it is compared with the ITBA0 register if the ten bit address is enabled (ENTB='1' in the ITMK0 register). An acknowledge is sent to the master after reception of a ten bit address header with write access[1]. Then, the second incoming byte is compared to the ITBA0 register. If a match is detected, an acknowledge signal is sent to the master device and the AAS bit is set.
Additionally, the interface acknowledges upon the the reception of a ten bit header with read access[2] after a *repeated* start condition.
All bits of the slave address may be masked using the ITMK0 register. The received ten bit slave address is written back to the ITBA0 register, and is only valid while the AAS bit in the IBSR0 register is '1'.

24.4.4. Ten Bit Address Mask Register (ITMK0)

This register contains the ten bit slave address mask and the ten bit slave address enable bit.

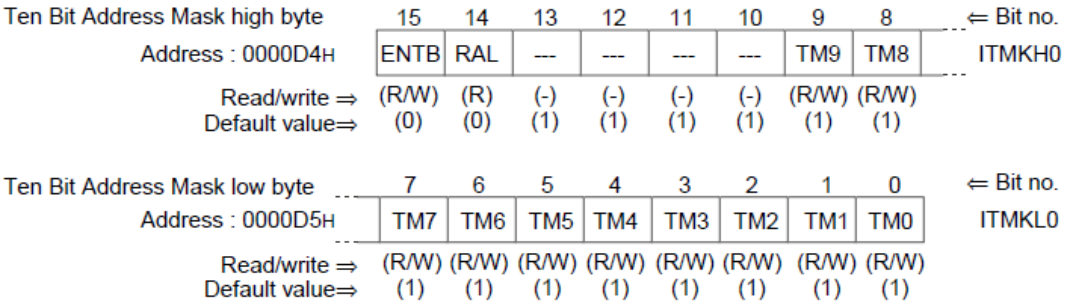


Figure 25-13 Ten Bit Address Mask Register (ITMK0)

[bit 15] ENTB - EnaBle Ten Bit slave address
This bit enables the ten bit slave address (and the acknowleding upon its reception). Write access to this bit is only possible if the interface is disabled (EN='0' in ICCR0).

0	Ten bit slave address disabled.
1	Ten bit slave address enabled.

1 Note: a ten bit header (write access) consists of the following bit sequence: 11110, TA9, TA8, 0.
2 Note: a ten bit header (read access) consists of the following bit sequence: 11110, TA9, TA8, 1.

[bit 14] RAL - Received slave Address Length

This bit indicates whether the interface was addressed as a seven or ten bit slave. It is read-only.

0	Addressed as seven bit slave.
1	Addressed as ten bit slave.

This bit can be used to determine whether the interface was addressed as a seven or ten bit slave if both slave addresses are enabled (ENTB='1' and ENSB='1'). Its contents is only valid if the AAS bit in the IBSR0 register is '1'. This bit is also reset if the interface is disabled (EN='0' in ICCR0).

[bit 13] - [bit 10] Not used.

These bits always read '1'.

[bit 9] - [bit 0] TMK - Ten bit slave address MasK (TM9-TM0).

This register is used to mask the ten bit slave address of the interface. Write access to these bits is only possible if the interface is disabled (EN='0' in ICCR0).

0	Bit is not used in slave address comparision.
1	Bit is used in slave address comparision.

This can be used to make the interface acknowledge on multiple ten bit slave addresses. Only the bits set to '1' in this register are used in the ten bit slave address comparison. The received slave address is written back to the ITBA0 register and thus may be determined by reading the ITBA0 register if the AAS bit in the IBSR0 register is '1'.

Note: If the address mask is changed after the interface had been enabled, the slave address should also be set again since it could have been overwritten by a previously received slave address.

24.4.5. Seven Bit Slave Address Register (ISBA0)

This register designates a seven bit slave address.

Write access to this register is only possible if the interface is disabled (EN='0' in ICCR0).

Seven Bit Address register	7	6	5	4	3	2	1	0	⇐ Bit no.
Address : 0000D7H	---	SA6	SA5	SA4	SA3	SA2	SA1	SA0	ISBA0
Read/write ⇒	(-)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	

Figure 25-14 Seven Bit Slave Address Register (ISBA0)

[bit 7] Not used.

This bit always reads '0'.

[bit 6] - [bit 0] Seven Bit slave Address (SA6-SA0)

When address data is received in slave mode, it is compared to the ISBA0 register if seven bit address mode is enabled (ENSB='1' in the ISMK0 register). If a match is detected, an acknowledge signal is sent to the master device and the AAS bit is set.

All the bits of the slave address may be masked using the ISMK0 register. The received seven bit slave address is written back to the ISBA0 register, it is only valid while the AAS bit in the IBSR0 register is '1'.

The interface does not compare the contents of this register to the incoming data if a ten bit header or a general call is received.

24.4.6. Seven Bit Slave Address Mask Register (ISMK0)

This register contains the seven bit slave address mask and the seven bit mode enable bit. Write access to this register is only possible if the interface is disabled (EN='0' in ICCR0).

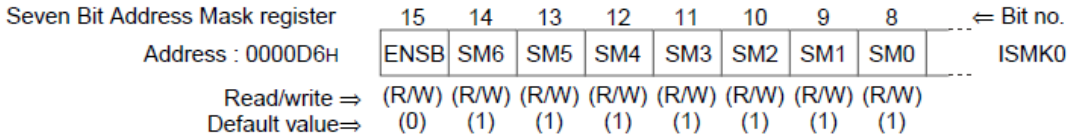


Figure 25-15 Seven Bit Slave Address Mask Register (ISMK0)

[bit 15] ENSB - EnaBle Seven Bit slave address

This bit enables the seven bit slave address (and the acknowleding upon its reception).

0	Seven bit slave address disabled.
1	Seven bit slave address enabled.

[bit 14] - [bit 8] SMK - Seven bit slave address MasK (SM6-SM0)

This register is used to mask the seven bit slave address of the interface.

0	Bit is not used in slave address comparision.
1	Bit is used in slave address comparision.

This can be used to make the interface acknowledge on multiple seven bit slave addresses. Only the bits set to '1' in this register are used in the seven bit slave address comparison. The received slave address is written back to the ISBA0 register and thus may be determined by reading the ISBA0 register if the AAS bit in the IBSR0 register is '1'.

Note: If the address mask is changed after the interface had been enabled, the slave address should also be set again since it could have been overwritten by a previously received slave address.

24.4.7. Data Register (IDAR0)

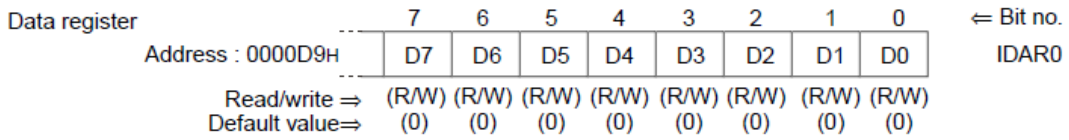


Figure 25-16 Data Register (IDAR0)

[bit 15] - [bit 8] Not used.

These bits always read '0'.

[bit 7] - [bit 0] Data bits (D7-D0)

The data register is used in serial data transfer, and transfers data MSB-first. This register is double buffered on the write side, so that when the bus is in use (BB='1'), write data can be loaded to the register for serial transfer. The data byte is loaded into the internal transfer register if the INT bit in the IBCR0 register is being cleared or the bus is idle (BB='0' in IBSR0). For read access, the internal register is read directly, therefore *received data values in this register are only valid if INT='1' in the IBCR2 register*.

24.4.8. Clock Control Register (ICCR0)

The clock control register (ICCR0) has the following functions:

- Enable IO pad noise filters
- Enable I²C interface operation
- Setting the serial clock frequency

Clock Control register	15	14	13	12	11	10	9	8	⇐ Bit no.
Address : 0000DAH	---	NSF	EN	CS4	CS3	CS2	CS1	CS0	ICCR0
Read/write ⇒	(-)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	(R/W)	
Default value⇒	(0)	(0)	(0)	(1)	(1)	(1)	(1)	(1)	

Figure 25-17 Clock Control Register (ICCR0)

[bit 15] Not used.
This bit always reads '0'.

[bit 14] IO pad NoiSe Filter enable.
This bit enables the noise filters built into the SDA and SCL IO pads.
The noise filter will suppress single spikes with a pulse width of 0 ns (minimum) and between 1 and 1.5 cycles of R-bus (maximum). The maximum depends on the phase relationship between I²C signals (SDA, SCL) and R-bus clock.
It should be set to '1' if the interface is transmitting or receiving at data rates above 100 Kbit/s.

[bit 13] EN (ENable)
This bit enables the I²C interface operation. It can only be set by the application but may be cleared by the application and the hardware.

0	Interface disabled.
1	Interface enabled.

When this bit is set to '0', all the bits in the IBSR0 register and IBCR0 register (except the BER and BEIE bits) are cleared, the module is disabled and the I²C lines are left open. It is cleared by the hardware if a bus error occurs (BER='1' in IBCR0).
Warning: The interface *immediately* stops transmitting or receiving if is it is disabled. This could leave the I²C bus in an undesired state!

[bit 12] - [bit 8] CS4-0 (Clock preScaler)
These bits select the serial bit rate. They can only be changed if the interface is disabled (EN='0') or the EN bit is being cleared in the same write access.
It is determined by the following formula:

$$\text{Bitrate} = \frac{\phi}{n*12 + 18}$$ **Noise filter disabled**
n>0; ϕ : R-Bus clock CLKP (set by DIVR0 register)

$$\text{Bitrate} = \frac{\phi}{n*12 + 19\ (+1)}$$ **Noise filter enabled**
n>0; ϕ : R-Bus clock CLKP (set by DIVR0 register)
(+1): Unaccurancy caused by noise filter operation

(Note) Because of the noise filter (depending on the relationship between an external signal and the internal clock, the filter will cause different delays) the divider in the second formula can vary between (12n + 19) and (12n + 20).

24.4.8.1. Prescaler settings:

Table 25-1 I²C Prescaler Settings

n	CS4	CS3	CS2	CS1	CS0
1	0	0	0	0	1
2	0	0	0	1	0
3	0	0	0	1	1
...					
31	1	1	1	1	1

Do not use prescaler setting n=0, it violates SDA/SCL timings!
The table below shows SCL frequency measurement results for the most common R-bus clock settings and the recommended related pre-scaler settings for 100 Kbit/s and 400 Kbit/s operation.

R-Bus Clock (CLKP)	100 kBit (Noise filter disabled)		400 kBit (Noise filter enabled)	
[MHz]	n	Bitrate [kBit]	n	Bitrate [kBit]
41.66			8	359.2
20.83	16	99.2	3	372.0
10.42	8	91.4		

It should be noted that the measured values have been determined by examining the last 8 cycles of a transfer. This was done because the first cycle of all address or data transfers is longer than the other cycles. To be more precise: In the case of an address transfer, this first cycle is 3 prescaler periods longer than the other cycles, in the case of a data transfer, it is 4 prescaler periods longer (see figure below).

24.4.8.2. SCL Waveforms

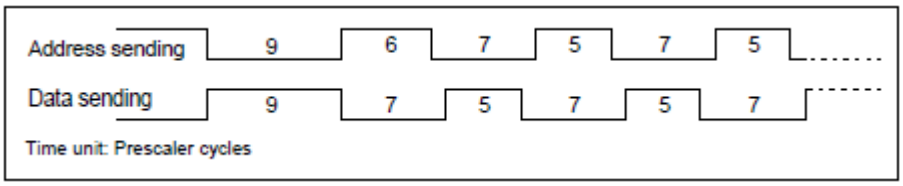


Figure 25-18 SCL Waveforms

Figure 25-18 shows the SCL waveform for sending address and data bits. The timings given in the figure are prescaler periods (e.g. '9' means 9 times the prescaler count based on the R-Bus clock). The timings in the figure are only valid if no other device on the I²C bus influences the SCL timing.

24.5. I²C Interface Operation

The I²C bus executes communication using two bi-directional bus lines, the serial data line (SDA) and serial clock line (SCL). The I²C interface has two open-drain I/O pins (SDA/SCL) corresponding to these lines, enabling wired logic applications.

- **Start Conditions**

When the bus is free (BB='0' in IBSR0, MSS='0' in IBCR2), writing '1' to the MSS bit places the I²C interface in master mode and generates a start condition.
If a '1' is written to it while the bus is idle (MSS='0' and BB='0'), a start condition is generated and the contents of the IDAR0 register (which should be address data) is sent.
Repeated start conditions can be generated by writing '1' to the SCC bit when in bus master mode and interrupt status (MSS='1' and INT='1' in IBCR0).

If a '1' is written to the MSS bit while the bus is in use (BB='1' and TRX='0' in IBSR0; MSS='0'and INT='0'in IBCR0), the interface waits until the bus is free and then starts sending.
If the interface is addressed as a slave with write access (data reception) in the meantime, it will start sending after the transfer has ended and the bus is free again. If the interface is sending data as a slave in the meantime, it will not start sending data if the bus is free again. It is important to check whether the interface was addressed as slave (MSS='0' in IBCR0 and AAS='1' in IBSR0), sent the data byte successfully (MSS='1' in IBCR0) or failed to send the data byte (AL='1' in IBSR0) at the next interrupt!

Writing '1' to the MSS bit or SCC bit in any other situation has no significance.

- **Stop Conditions**

Writing '0' to the MSS bit in master mode (MSS='1' and INT='1' in IBCR0) generates a stop condition and places the device in slave mode. Writing '0' to the MSS bit in any other situation has no significance.
After clearing the MSS bit, the interface tries to generate a stop condition which might fail if another master pulls the SCL line low before the stop condition has been generated. *This will generate an interrupt after the next byte has been transferred!*

- **Slave Address Detection**

In slave mode, after a start condition has been generated, the BB is set to '1' and data sent from the master device is received into the IDAR0 register.
After the reception of eight bits, the contents of the IDAR2 register are compared to the ISBA register using the bit mask stored in ISMK0 if the ENSB bit in the ISMK0 register is '1'. If a match results, the AAS bit is set to '1' and an acknowledge signal is sent to the master. Then bit 0 of the received data (bit 0 of the IDAR0 register) is inverted and stored in the TRX bit.
If the ENTB bit in the ITMK0 register is '1' and a ten bit address header (11110, TA1, TA0, 0_B) is detected, the interface sends an acknowledge signal to the master and stores the inverted last data bit in the TRX register. No interrupt is generated. Then, the next transferred byte is compared (using the bit mask stored in ITMK0) to the lower byte of the ITBA0 register. If a match is found, an acknowledge signal is sent to the master, the AAS bit is set and an interrupt is generated.
If the interface was addressed as a slave and detects a repeated start condition, the AAS bit is set after reception of the ten bit address header (11110, TA1, TA0, 1_B) and an interrupt is generated.
Since there are separate registers for the ten and seven bit address and their bitmasks, it is possible to make the interface acknowledge on both addresses by setting the ENSB (in ISMK0) and ENTB (in ITMK0) bits.
The received slave address length (seven or ten bit) may be determined by reading the RAL bit in the ITMK0 register (this bit is valid if the AAS bit is set only).
It is also possible to give the interface no slave address by setting both bits to '0' if it is only used as a master.
All slave address bits may be masked with their corresponding mask register (ITMK0 or ISMK0).

- **Slave Address Masking**

Only the bits set to ‘1’ in the mask registers (ITMK0 / ISMK0) are used for address comparison, all other bits are ignored. The received slave address can be read from the ITBA0 (if ten bit address received, RAL=‘1’) or ISBA0 (if seven bit address received, RAL=‘0’) register if the AAS bit in the IBSR0 register is ‘1’.
If the bitmasks are cleared, the interface can be used as a bus monitor since it will always be addressed as slave. Note that this is not a real bus monitor because it *acknowledges* upon any slave address reception, even if there is no other slave listening.

- **Addressing Slaves**

In master mode, after a start condition is generated the BB and TRX bits are set to ‘1’ and the contents of the IDAR0 register is sent in MSB first order. After address data is sent and an acknowledge signal was received from the slave device, bit 0 of the sent data (bit 0 of the IDAR0 register after sending) is inverted and stored in the TRX bit. Acknowledgement by the slave may be checked using the LRB bit in the IBSR0 register. This procedure also applies to a repeated start condition.
In order to address a ten bit slave for write access, two bytes have to be sent. The first one is the ten bit address header which consists of the bit sequence ‘1 1 1 1 0 A9 A8 0’, it is followed by the second byte containing the lower eight bits of the ten bit slave address (A7 - A0).
A ten bit slave is accessed for reading by sending the above byte sequence and generating a repeated start condition (SCC bit in IBCR0) followed by a ten bit address header with read access (1 1 1 1 0 A9 A8 1).
Summary of the address data bytes:
7 bit slave, write access: Start condition - A6 A5 A4 A3 A2 A1 A0 0.
7 bit slave, read access: Start condition - A6 A5 A4 A3 A2 A1 A0 1.
10 bit slave, write access: Start condition - 1 1 1 1 0 A9 A8 0 - A7 A6 A5 A4 A3 A2 A1 A0.
10 bit slave, read access: Start condition - 1 1 1 1 0 A9 A8 1 - A7 A6 A5 A4 A3 A2 A1 A0 - repeated start - 1 1 1 1 0 A9 A8 1.

- **Arbitration**

During sending in master mode, if another master device is sending data at the same time, arbitration is performed. If a device is sending the data value ‘1’ and the data on the SDA line has an ‘L’ level value, the device is considered to have lost arbitration, and the AL bit is set to ‘1.’ Also, the AL bit is set to ‘1’ if a start condition is detected at the first bit of a data byte but the interface did not want to generate one or the generation of a start or stop condition failed by some reason.
Arbitration loss detection clears both the MSS and TRX bit and immediately places the device in slave mode so it is able to acknowledge if its own slave address is being sent.

- **Acknowledgement**

Acknowledge bits are sent from the receiver to the transmitter. The ACK bit in the IBCR0 register can be used to select whether to send an acknowledgment when data bytes are received.
When data is send in slave mode (read access from another master), if no acknowledgement is received from the master, the TRX bit is set to ‘0’ and the device goes to receiving mode. This enables the master to generate a stop condition as soon as the slave has released the SCL line.
In master mode, acknowledgement by the slave may be checked by reading the LRB bit in the IBSR0 register.

24.6. Programming Flow Charts

- **Example Of Slave Addressing And Sending Data**

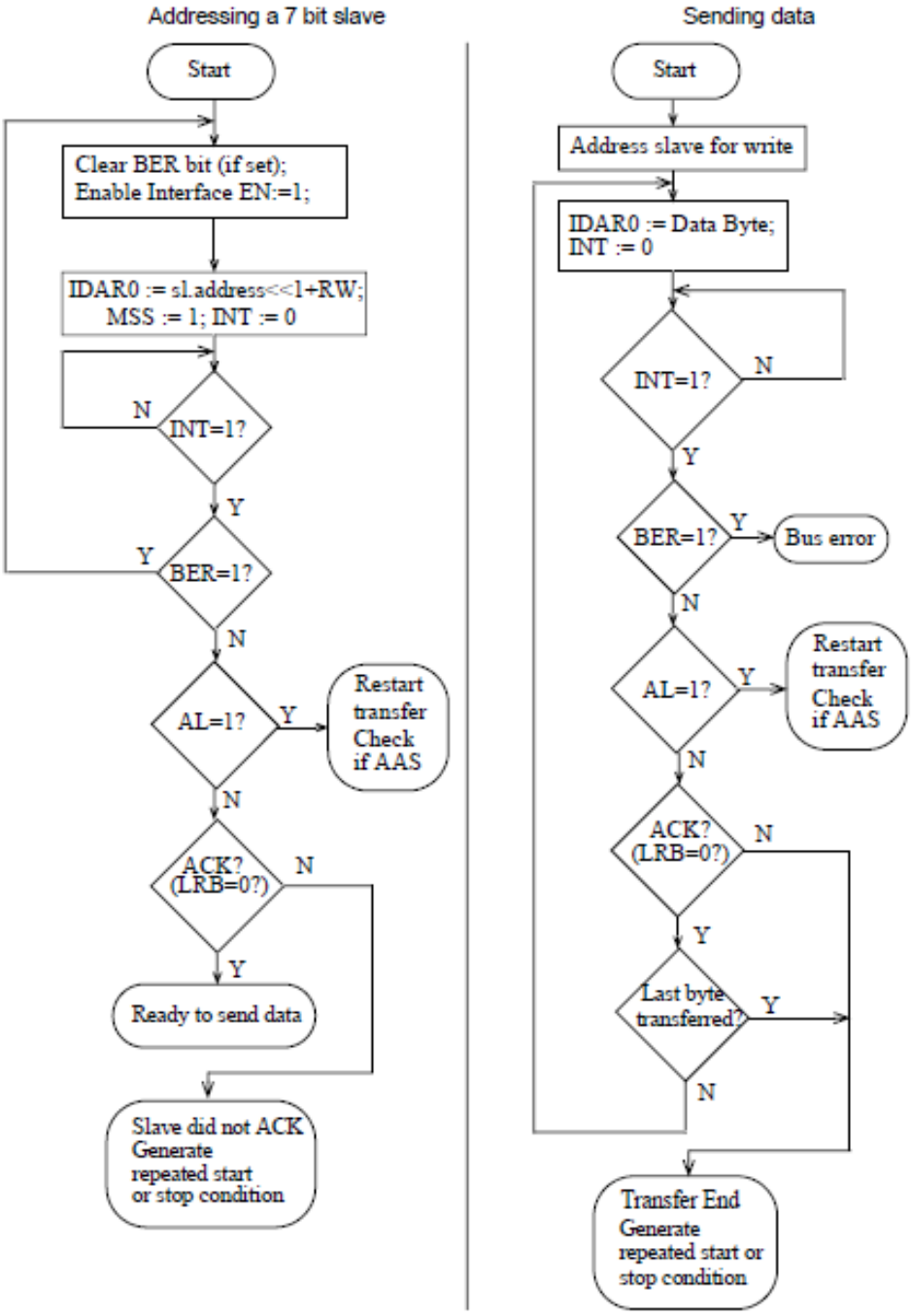


Figure 25-19 Example of Slave Addressing and Sending Data

- **Example Of Receiving Data**

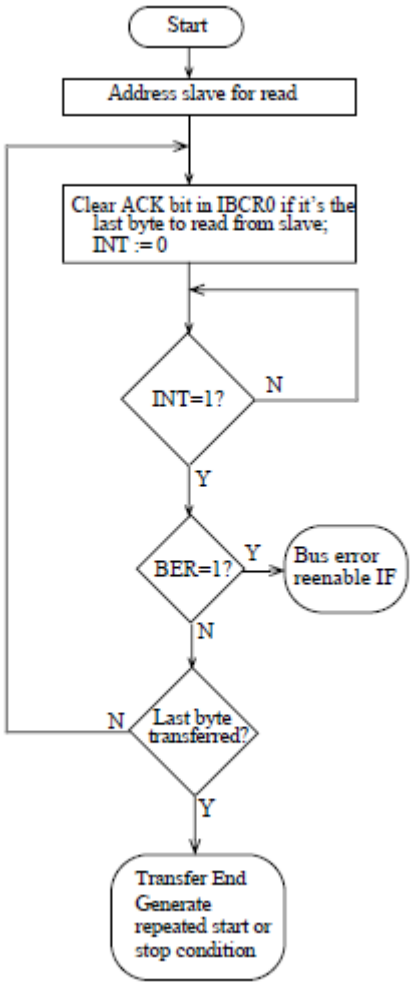


Figure 25-20 Example of Receiving Data

- **Example Of An Interrupt Handler**

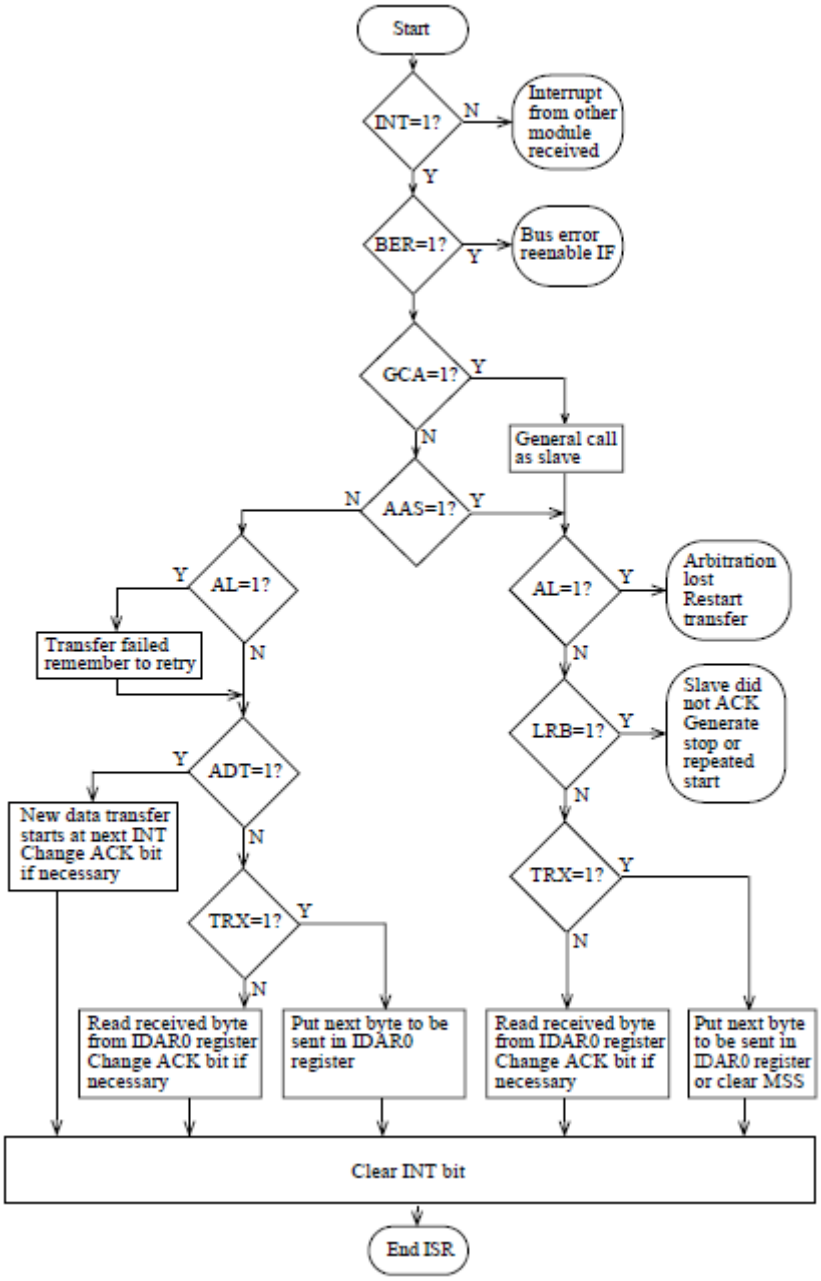


Figure 25-21 Example of an Interrupt Handler

25. USART (LIN/FIFO)

This chapter describes the USART interface of the MB88F333.

25.1. Overview

This chapter explains the function and operation of the USART. The USART with LIN (Local Interconnect Network) - unit is a general-purpose serial data communication interface for performing synchronous or asynchronous communication with external devices. 16 bytes transmission and reception FIFOs are available for selected channels.
The USART provides bidirectional communication function (normal mode), master-slave communication function (multiprocessor mode in master/slave systems), and special features for LIN-bus systems (working both as master or as slave device).

(Note) This chapter only lists the registers and addresses of USART0.

25.1.1. USART Functions

The USART is a general-purpose, serial data communication interface for transmitting serial data to and receiving data from another CPU or peripheral device. It has the functions listed in Table 26-1**.**

Table 26-1 functions

Item	Function
Data buffer	Full-duplex
Serial Input	5 times oversampling in asynchronous mode
Transfer mode	- Clock synchronous (start-stop synchronization and start-stop-bit-option) - Clock asynchronous (using start-, stop-bits)
Baud rate	- A dedicated baud rate generator is provided, which consists of a 15-bit-reload counter - An external clock can be input and also be adjusted by the reload counter
Data length	- 7 bits (not in synchronous or LIN mode) - 8 bits
Signal mode	Non-return to zero (NRZ) and return to zero (RZ)
Start bit timing	Clock synchronization to the falling edge of the start bit in asynchronous mode
Reception error Detection	- Framing error - Overrun error - Parity error
Interrupt request	- Reception interrupt (reception complete, reception error detect, Bus-Idle, LIN-Synch-break detect) - Transmission interrupt (transmission complete)
Master-slave communication function (multiprocessor mode)	One-to-n communication (one master to n slaves) (This function is supported both for master and slave system).
Synchronous mode	Function as Master- or Slave-USART
Transceiving pins	Direct access possible
LIN bus options	- Operation as master device - Operation as slave device - Generation of LIN-Sync-break - Detection of LIN-Sync-break - Detection of start/stop edges in LIN-Sync-field connected to ICU 0 and 2
Synchronous serial clock	The synchronous serial clock can be output continuously on the SPIx_SCK pin for synchronous communication with start & stop bits

Clock delay option	Special synchronous Clock Mode for delaying clock (useful for SPI-compliance)
16 word deep FIFO	FIFO can be activated with receive progammable trigger level

25.1.2. USART operation modes

The USART operates in four different modes, which are determined by the MD0- and the MD1-bit of the serial mode register (SMR). Modes 0 and 2 are used for bidirectional serial communication, mode 1 for master/slave communication and mode 3 for LIN master/slave communication.

Table 26-2 operation modes

Operation mode		Data length		Synchronization of mode	Length of stop bit	data bit direction*
		parity disabled	parity enabled			
0	normal mode	7 or 8		asynchronous	1 or 2	L/M
1	multiprocessor	7 or 8 + 1**	-	asynchronous	1 or 2	L/M
2	normal mode	8		synchronous	0, 1 or 2	L/M
3	LIN mode	8	-	asynchronous	1	L

* means the data bit transfer format: LSB or MSB first.

** "+1" means the indicator bit of the address/data selection in the multiprocessor mode, instead of parity.

Mode 1 operation is supported both for master or slave operation in a master-slave connection system. In Mode 3 the function is locked to 8N1-Format, LSB first.

If the mode is changed, the device cuts off all possible transmission/reception and then awaits new action.

The MD1and MD0 bits of the Serial Mode Register (SMR) determine the operation mode as shown in the following table:

Table 26-3 Mode Bit Setting

MD1	MD0	Mode	Description
0	0	0	Asynchronous (normal mode)
0	1	1	Asynchronous (multiprocessor mode)
1	0	2	Synchronous (normal mode)
1	1	3	Asynchronous (LIN mode)

25.2. USART Configuration

25.2.1. USART consists of the following blocks:

- Reload Counter
- Reception Control Circuit
- Reception Shift Register
- Reception Data Register
- Transmission Control Circuit
- Transmission Shift Register
- Transmission Data Register
- Error Detection Circuit
- Oversampling Unit
- Interrupt Generation Circuit
- LIN Break Generation
- LIN Break and Synch Field Detection
- Bus Idle Detection Circuit
- Serial Mode Register (SMR)
- Serial Control Register (SCR)
- Serial Status Register (SSR)
- Extended Com. Contr. Reg. (ECCR)
- Extended Status/Contr. Reg. (ESCR)
- FIFO Control Register (FCR)
- FIFO Status Register (FSR)

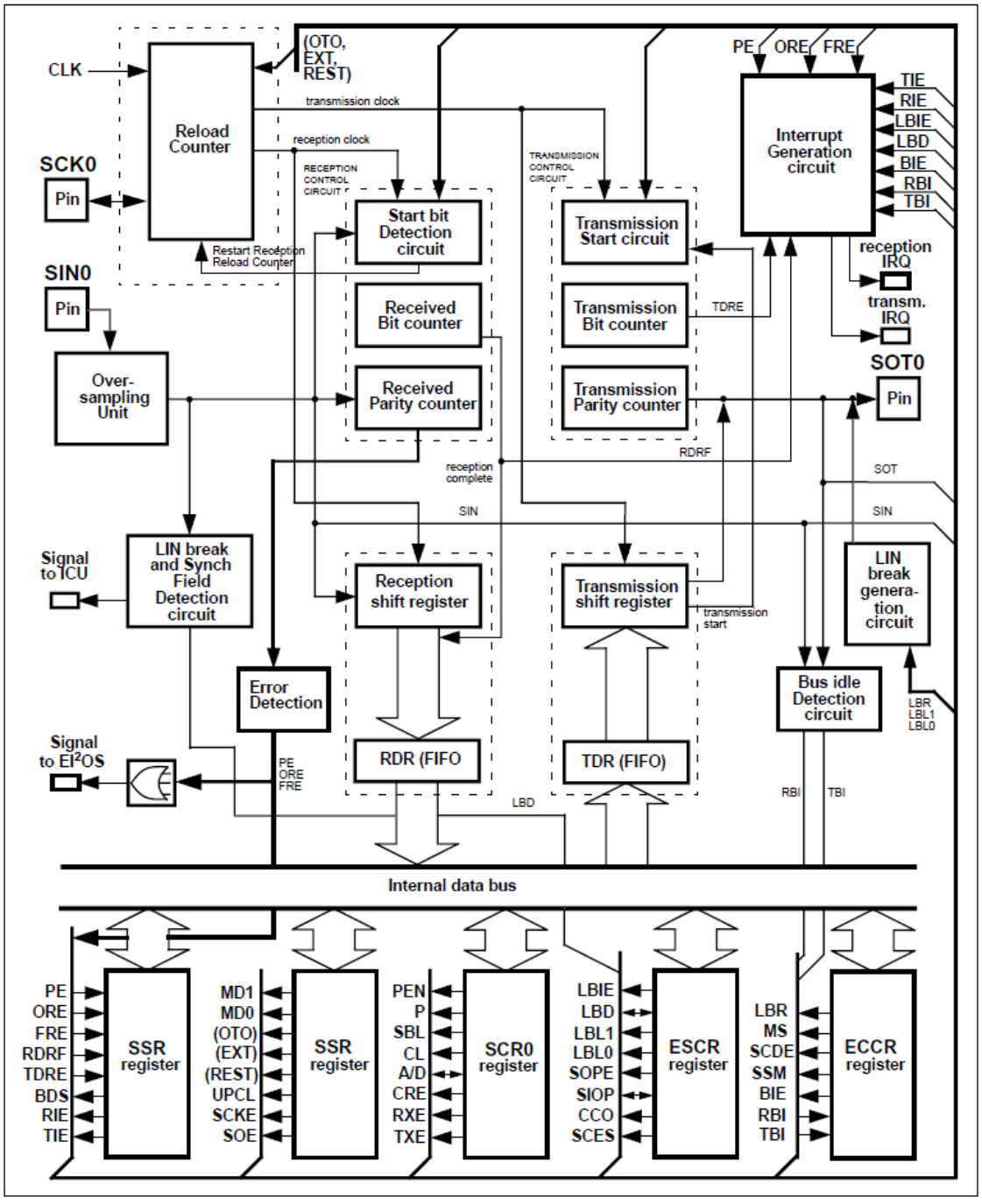


Figure 26-1 Block Diagram

25.2.2. Explanation of the different blocks

- Reload Counter

The reload counter functions as the dedicated baud rate generator. It can select external input clock or internal clock for the transmitting and receiving clocks. The reload counter has a 15 bit register for the reload value. The actual count of the transmission reload counter can be read via the BGR0/1 registers.

- Reception Control Circuit

The reception control circuit consists of a received bit counter, start bit detection circuit and received parity counter. The received bit counter counts reception data bits. When reception of one data item for the specified data length is complete, the received bit counter sets the Reception data register full flag. When the FIFO is enabled, the flag is set if the trigger level is reached. The start bit detection circuit detects start bits from the serial input signal and sends a signal to the reload counter to synchronize it to the falling edge of these start bits. The reception parity counter calculates the parity of the reception data.

- Reception Shift Register

The reception shift register fetches reception data input from the SPIx_DI pin, shifting the data bit by bit. When reception is complete, the reception shift register transfers receive data to the RDR register.

- Reception Data Register

This register retains reception data. Serial input data is converted and stored in this register. If the FIFO is enabled up to 16 receptions can be saved, the trigger level is programmable.

- Transmission Control Circuit

The transmission control circuit consists of a transmission bit counter, transmission start circuit, and transmission parity counter. The transmission bit counter counts transmission data bits. When the transmission of one data item of the specified data length is complete, the transmission bit counter sets the Transmission data register empty flag. The transmission start circuit starts transmission when data is written to TDR. The transmission parity counter generates a parity bit for data to be transmitted if parity is enabled.

- Transmission Shift Register

The transmission shift register transfers data written to the TDR register to itself and outputs the data to the SPIx_DO pin, shifting the data bit by bit.

- Transmission Data Register

This register sets transmission data. Data written to this register is converted to serial data and output.
If the FIFO is enabled, up to 16 transmissions can be saved and continuously transmitted

- Error Detection Circuit

The error detection circuit checks if there was any error during the last reception. If an error has occurred it sets the corresponding error flags.

- Oversampling Unit

The oversampling unit oversamples the incoming data at the SPIx_DI pin for five times. It is switched off in synchronous operation mode.

- Interrupt Generation Circuit

The interrupt generation circuit administers all cases of generating a reception or transmission interrupt. If a corresponding enable flag is set and an interrupt case occurs the interrupt will be generated immediately.

- LIN Break and Synchronization Field Detection Circuit

The LIN break and LIN synchronization field detection circuit detects a LIN break, if a LIN master node is sending a message header. If a LIN break is detected a special flag bit (LBD) is generated. The first and the fifth falling edge of the synchronization field is recognized by this circuit by generating an internal signal for the Input Capture Unit to measure the actual serial clock time of the transmitting master node.

- LIN Break Generation Circuit

The LIN break generation circuit generates a LIN break of a determined length.

- Bus Idle Detection circuit

The bus idle detection circuit recognizes if neither reception nor transmission is going on. In this case the circuit generates a special flag bit (RBI and TBI).

- Serial Mode Register

This register performs the following operations:

- Selecting the USART operation mode
- Selecting a clock input source
- Selecting if an external clock is connected “one-to-one” or connected to the reload counter
- Resetting the USART (preserving the settings of the registers)
- Specifying whether to enable serial data output to the corresponding pin
- Specifying whether to enable clock output to the corresponding pin

- Serial Control Register

This register performs the following operations:

- Specifying whether to provide parity bits
- Selecting parity bits
- Specifying a stop bit length
- Specifying a data length
- Selecting a frame data format in mode 1
- Clearing the error flags
- Specifying whether to enable transmission
- Specifying whether to enable reception
- Serial Status Register

This register checks the transmission and reception status and error status, and enables and disables transmission and reception interrupt requests.

- Extended Status/Control Register

This register provides several LIN functions, direct access to the SPIx_DI and SPIx_DO pin and setting for the USART synchronous clock mode.

- Extended Communication Control Register

The extended communication control register provides bus idle recognition interrupt settings, synchronous clock settings, and the LIN break generation.

- FIFO Control Register

With the FCR register the TX/RX FIFOs can be enabled, the RX interrupt trigger level can be set and the FIFO status register mode can be set.

- FIFO Status Register

With the FSR register shows the number of valid RX/TX data in the FIFO buffers.

25.3. USART Registers

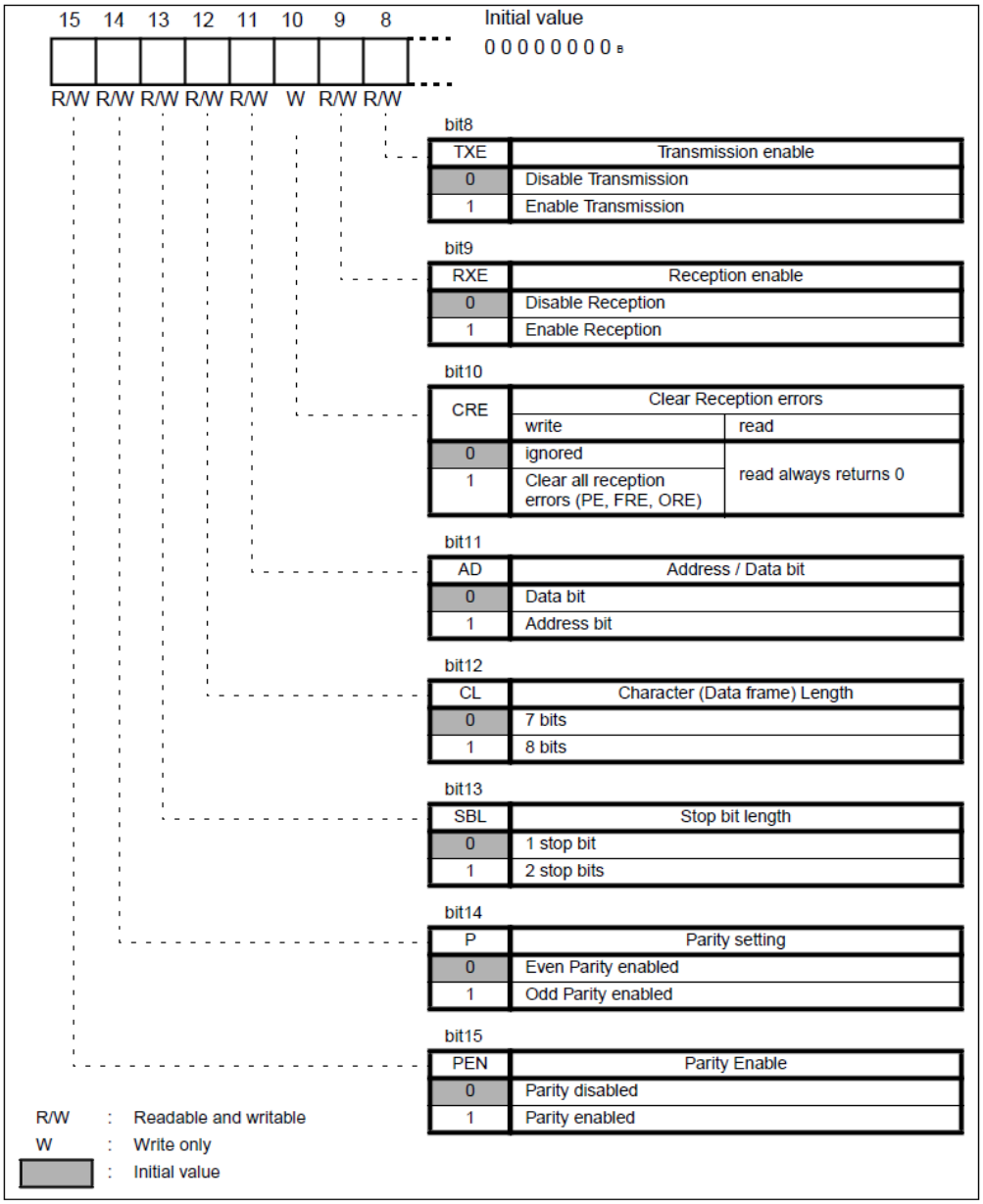
The following table defines the USART0 registers:

Table 26-4 Registers

Address	bit 15 bit 8	bit 7 bit 0
060H, 061H	SCR0 (Serial Control Register)	SMR0 (Serial Mode Register)
062H, 063H	ESCR0 (Extended Status/Control Reg.)	ECCR0 (Extended Comm. Contr. Reg.)
064H, 065H	SSR0 (Serial Status Register)	RDR0/TDR0 (Rx, Tx Data Register)
066H, 067H	FSR0 (FIFO status register)	FCR 0 (FIFO control register)
068H, 069H	SCR1 (Serial Control Register)	SMR1 (Serial Mode Register)
06aH, 06bH	ESCR1 (Extended Status/Control Reg.)	ECCR1 (Extended Comm. Contr. Reg.)
06cH, 06dH	SSR1 (Serial Status Register)	RDR1/TDR1 (Rx, Tx Data Register)
06eH, 06fH	FSR1 (FIFO status register)	FCR 1 (FIFO control register)
088H, 089H	BGR00 (Baud Rate Generator Reg. 0-0)	BGR01 (Baud Rate Generator Reg.0-1)
08aH,08bH	BGR10 (Baud Rate Generator Reg. 1-0)	BGR11 (Baud Rate Generator Reg.1-1)

25.3.1. Serial Control Register (SCRx)

This register specifies parity bits, selects the stop bit and data lengths, selects a frame data format in mode 1, clears the reception error flag and specifies whether to enable transmission and reception.

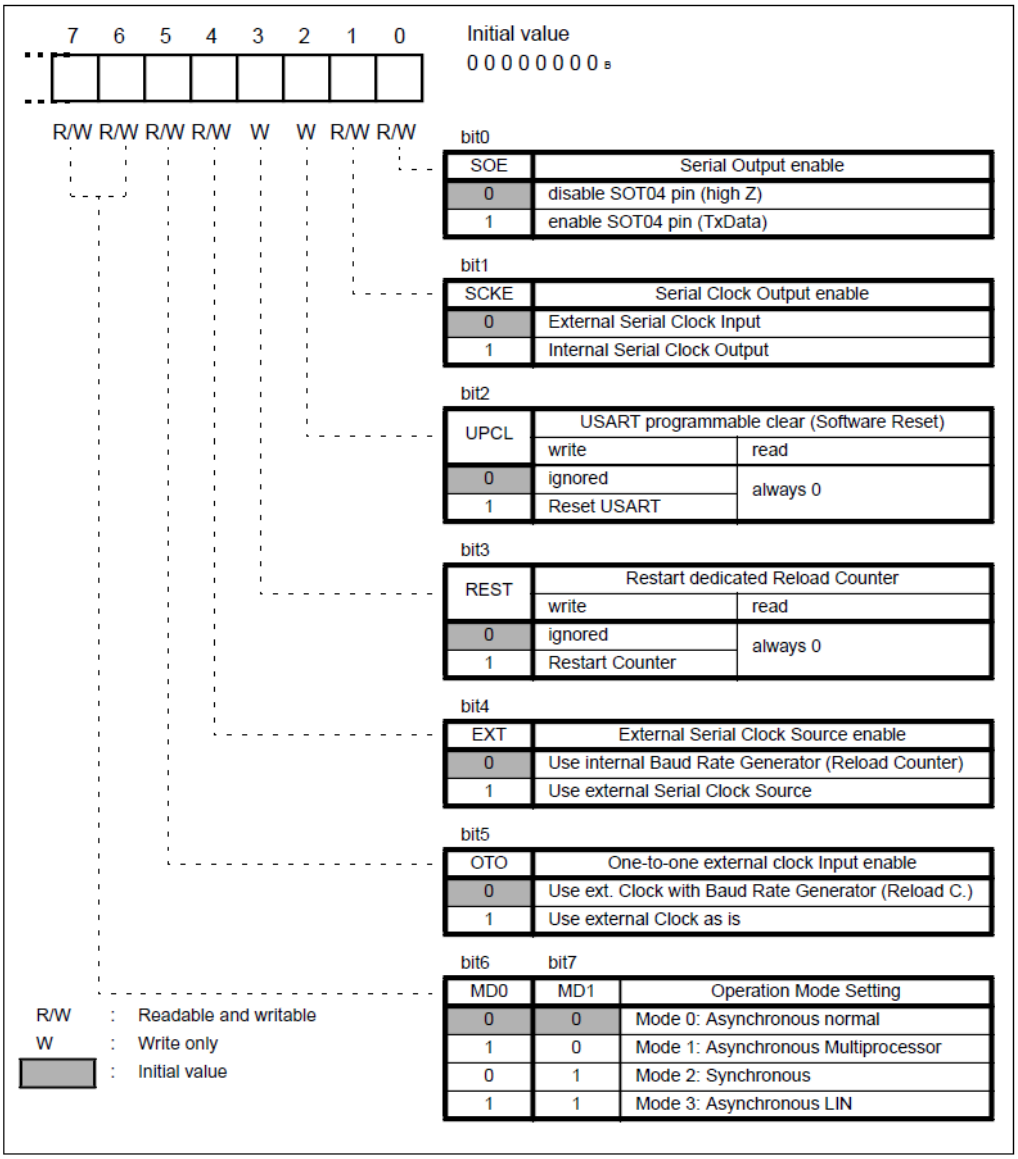


Bit name		Function
bit15	PEN: Parity enable bit	This bit selects whether to add a parity bit during transmission in serial asynchronous mode or detect it during reception. Parity is only provided in mode 0 and in mode 2 if SSM of the ECCR is selected. This bit is fixed to 0 (no parity) in mode 3 (LIN).
bit14	P: Parity selection bit	When parity is provided and enabled this bit selects even (0) or odd (1) parity
bit13	SBL: Stop bit length selection bit	This bit selects the length of the stop bit of an asynchronous data frame or a synchronous frame if SSM of the ECCR is selected. This bit is fixed to 0 (1 stop bit) in mode 3 (LIN).
bit12	CL: Data length selection bit	This bit specifies the length of transmission or reception data. This bit is fixed to 1 (8 bits) in mode 2 and 3.

<table>
<tr><td>bit11</td><td>AD: Address/Data selection bit *</td><td>This bit specifies the data format in multiprocessor mode 1. Writing to this bit determines an address or data frame to be sent next, reading from it returns the last received kind of frame. “1” indicates an address frame, “0” indicates a usual data frame.
Note:
During a RMW-Read cycle the AD bit returns the value to be sent instead of the last received AD bit.
see table below*
<table><tr><th>Cycle</th><th>Action</th></tr><tr><td>Write</td><td>Write data to be sent to AD-Bit</td></tr><tr><td>Normal Read</td><td>Read received AD-Bit</td></tr><tr><td>RMW-Read</td><td>Read data to be sent from AD-Bit</td></tr></table></td></tr>
<tr><td>bit10</td><td>CRE: Clear reception error flags bit</td><td>This bit clears the FRE, ORE, and PE flag of the Serial Status Register (SSR). This bit also clears a possible reception interrupt caused by errors.
Writing a 1 to it clears the error flag.
Writing a 0 has no effect.
Reading from it always returns 0.</td></tr>
<tr><td>bit9</td><td>RXE: Reception enable bit</td><td>This bit enables USART reception.
If this bit is set to 0, USART disables the reception of data frames. The LIN break detection in mode 0 or 3 remains unaffected.</td></tr>
<tr><td>bit8</td><td>TXE: Transmission enable bit</td><td>This bit enables USART transmission. If this bit is set to 0, USART disables the transmission of data frames.</td></tr>
</table>

25.3.2. Serial Mode Register (SMRx)

This register selects an operation mode and baud rate clock and specifies whether to enable output of serial data and clocks to the corresponding pin.

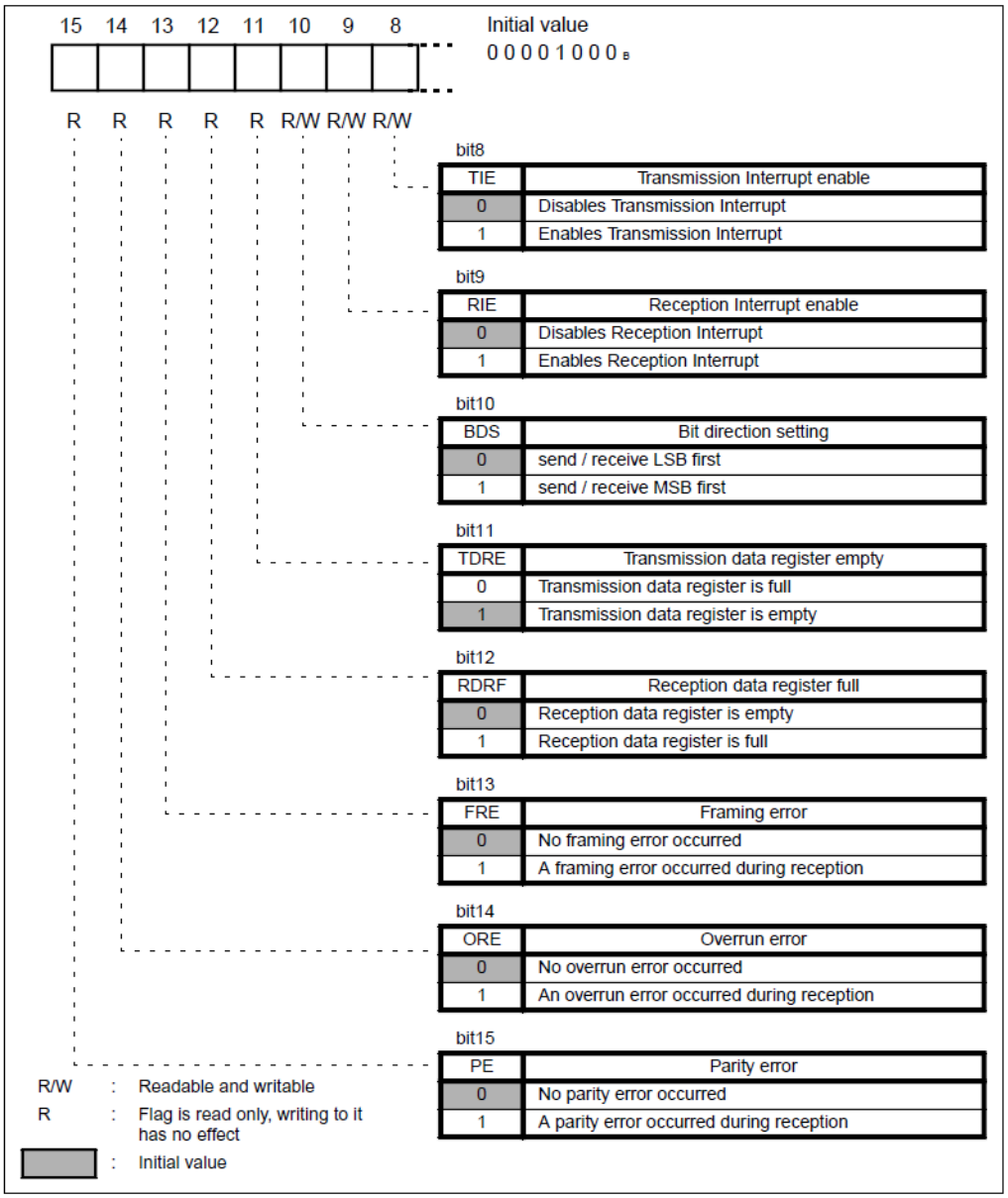


Bit name		Function
bit7 bit6	MD1 and MD0: Operation mode selection bits	These two bits sets the USART operation mode.
bit5	OTO: One-to-one external clock selection bit	This bit sets an external clock directly to the USART's serial clock. This function is used for synchronous slave mode operation
bit4	EXT: External clock selection bit	This bit selects internal or external clock source for the reload counter
bit3	REST: Restart of transmission reload counter bit	If a 1 is written to this bit the reload counter is restarted. Writing 0 to it has no effect. Reading from this bit always returns 0.
bit2	UPCL:USART programmable clear bit (Software reset)	Writing a 1 to this bit resets USART immediately. The register settings are preserved. Possible reception or transmission will cut off. All error flags are cleared and the Reception Data Register (RDR) contains 00h. Writing 0 to this bit has no effect. Reading from it always returns 0.
bit1	SCKE: Serial clock output enable	This bit controls the serial clock input-output ports. When this bit is 0, the SPIx_SCK pin operates as serial clock input pin. When this bit is 1, the SPIx_SCK pin operates as serial clock output pin. <Caution>

		When using the SPIx_SCK pin as serial clock input (SCKE=0) pin, set the pin as input port. Also, select external clock (EXT = 1) using the external clock selection bit.
bit0	SOE: Serial data output enable bit	This bit enables or disables the output of serial data. When this bit is 0, the SPIx_DO pin outputs the default mark level. When this bit is 1, the SPIx_DO pin outputs the transmission data.

25.3.3. Serial Status Register (SSRx)

This register checks the transmission status, reception status and error status, and enables and disables the transmission and reception interrupts.

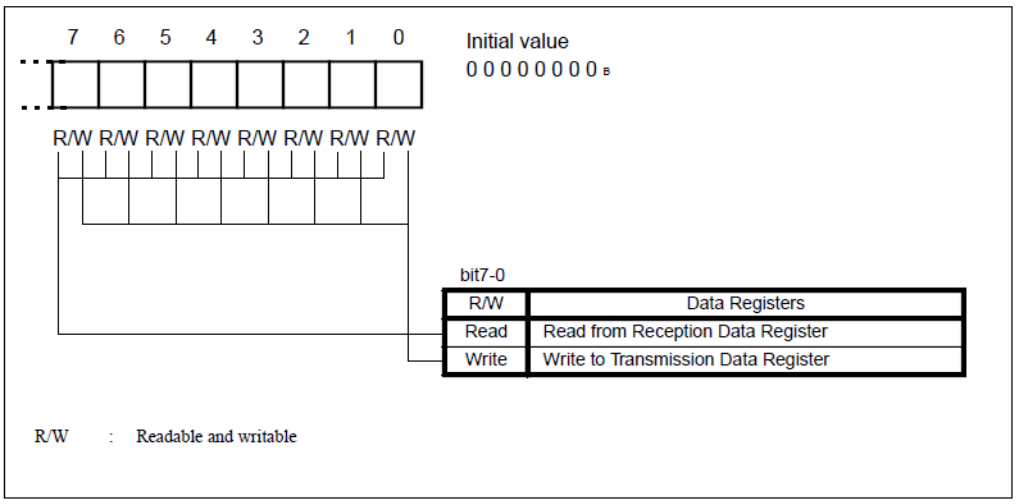


Bit name		Function
bit15	PE: Parity error flag bit	This bit is set to 1 when a parity error occurs during reception and is cleared when 0 is written to the CRE bit of the serial control register (SCR). A reception interrupt request is output when this bit and the RIE bit are 1. Data in the reception data register (RDR) is invalid when this flag is set.
bit14	ORE: Overrun error flag bit	This bit is set to 1 when an overrun error occurs during reception and is cleared when 0 is written to the CRE bit of the serial control register (SCR). A reception interrupt request is output when this bit and the RIE bit are 1. Data in the reception data register (RDR) is invalid when this flag is set.
bit13	FRE: Framing error flag bit	This bit is set to 1 when a framing error occurs during reception and is cleared when 0 is written to the CRE bit of the serial control register 1 (SCR). A reception interrupt request is output when this bit and the RIE bit are 1. Data in the reception data register (RDR) is invalid when this flag is set.
bit12	RDRF: Receive data full flag bit	This flag indicates the status of the reception data register (RDR). This bit is set to 1 when reception data is loaded into RDR and can only be cleared to 0 when the reception data register (RDR) is read. A reception interrupt request is output when this bit and the RIE bit are 1.

bit11	TDRE: Transmission data empty flag bit	This flag indicates the status of the transmission data register (TDR). This bit is cleared to 0 when transmission data is written to TDR and is set to 1 when data is loaded into the transmission shift register and transmission starts. A transmission interrupt request is generated if this bit and the RIE bit are 1. <Caution> This bit is set to 1 (TDR empty) as its initial value.
bit10	BDS: Transfer direction selection bit	This bit selects whether to transfer serial data from the least significant bit (LSB first, BDS=0) or the most significant bit (MSB first, BDS=1). <Caution> The high-order and low-order sides of serial data are interchanged with each other during reading from or writing to the serial data register. If this bit is set to another value after the data is written to the RDR register, the data becomes invalid. This bit is fixed to 0 in mode 3 (LIN)
bit9	RIE: Reception interrupt request enable bit	This bit enables or disables input of a request for transmission interrupt to the CPU. A reception interrupt request is output when this bit and the reception data flag bit (RDRF) are 1 or this bit and one or more error flag bits (PE, ORE, and FRE) are 1.
bit8	TIE: Transmission interrupt request enable bit	This bit enables or disables output of a request for transmission interrupt to the CPU. A transmission interrupt request is output when this bit and the TDRE bit are 1.

25.3.4. Reception and Transmission Data Register (RDRx / TDRx)

The reception data register (RDR) holds the received data. The transmission data register (TDR) holds the transmission data. Both RDR and TDR registers are located at the same address.
(Note):TDR is a write-only register and RDR is a read-only register. These registers are located in the same address, so the read value is different from the write value. Therefore, instructions that perform a read-modify-write (RMW) operation, such as the INC/DEC instruction, cannot be used.



25.3.4.1. Reception:

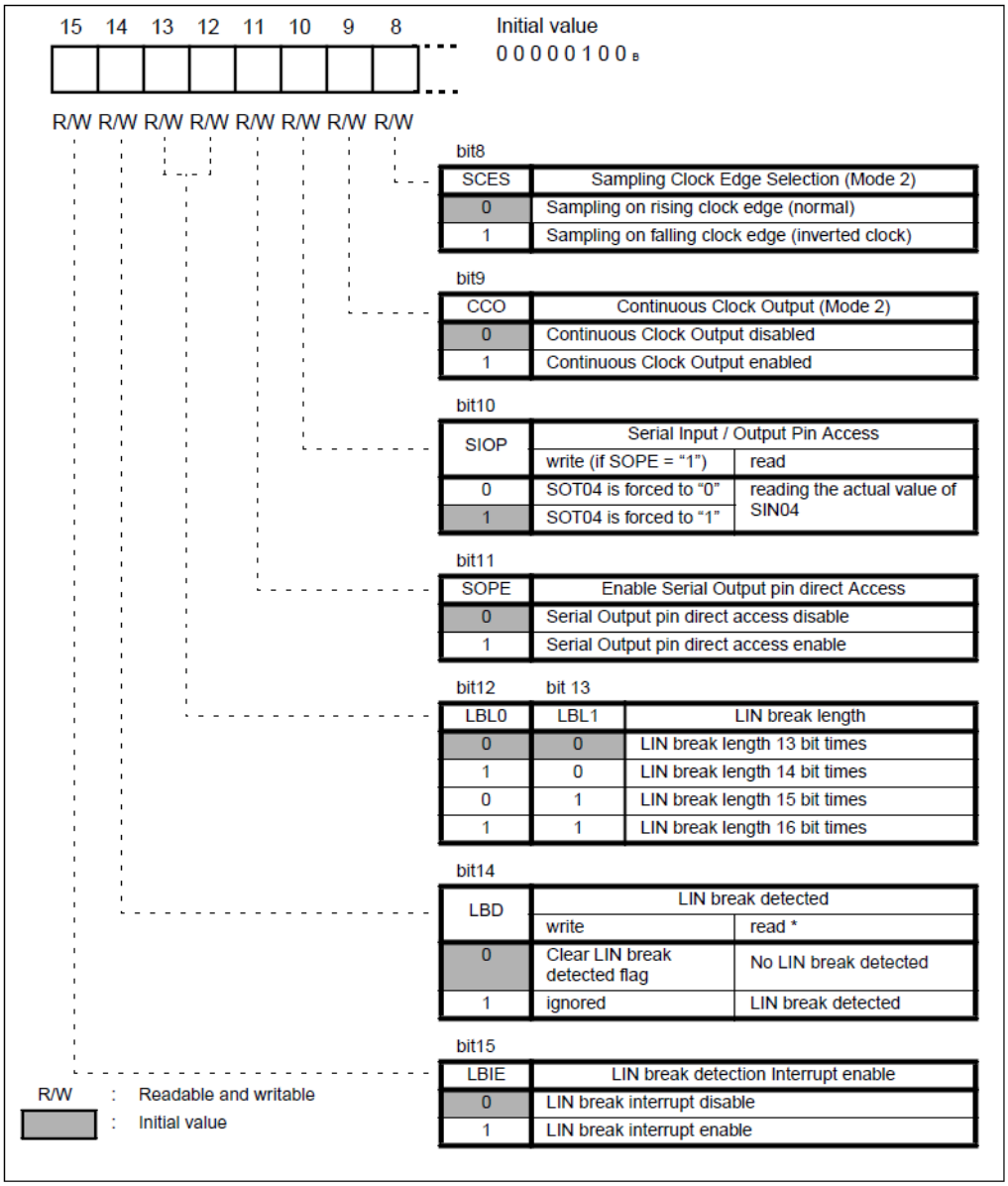
RDR is the register that contains reception data. The serial data signal transmitted to the SPIx_DI pin is converted in the shift register and stored there. When the data length is 7 bits, the uppermost bit (D7) contains 0. When reception is complete, the data is stored in this register and the reception data full flag bit (SSR: RDRF) is set to 1. If a reception interrupt request is enabled at this point, a reception interrupt occurs.
Read RDR when the RDRF bit of the status register (SSR) is 1. The RDRF bit is cleared automatically to 0 when RDR is read. Also the reception interrupt is cleared if it is enabled and no error has occurred.
Data in RD0 is invalid when a reception error occurs (SSR: PE, ORE, or FRE = 1).

25.3.4.2. Transmission:

When data to be transmitted is written to the transmission data register in transmission enable state, it is transferred to the transmission shift register, then converted to serial data, and transmitted from the serial data output terminal (SPIx_DO pin). If the data length is 7 bits, the uppermost bit (D7) is not sent.
When transmission data is written to this register, the transmission data empty flag bit (SSR: TDRE) is cleared to 0. When transfer to the transmission shift register is complete, the bit is set to 1. When the TDRE bit is 1, the next part of transmission data can be written. If output transmission interrupt requests have been enabled, a transmission interrupt is generated. Write the next part of transmission data when a transmission interrupt is generated or the TDRE bit is 1.

25.3.5. Extended Status/Control Register (ESCRx)

This register provides several LIN functions, direct access to the SPIx_DI and SPIx_DO pin and setting for synchronous clock mode.



bit8

SCES	Sampling Clock Edge Selection (Mode 2)
0	Sampling on rising clock edge (normal)
1	Sampling on falling clock edge (inverted clock)

bit9

CCO	Continuous Clock Output (Mode 2)
0	Continuous Clock Output disabled
1	Continuous Clock Output enabled

bit10

SIOP	Serial Input / Output Pin Access	
	write (if SOPE = "1")	read
0	SOT04 is forced to "0"	reading the actual value of SIN04
1	SOT04 is forced to "1"	

bit11

SOPE	Enable Serial Output pin direct Access
0	Serial Output pin direct access disable
1	Serial Output pin direct access enable

bit12 bit 13

LBL0	LBL1	LIN break length
0	0	LIN break length 13 bit times
1	0	LIN break length 14 bit times
0	1	LIN break length 15 bit times
1	1	LIN break length 16 bit times

bit14

LBD	LIN break detected	
	write	read *
0	Clear LIN break detected flag	No LIN break detected
1	ignored	LIN break detected

bit15

LBIE	LIN break detection Interrupt enable
0	LIN break interrupt disable
1	LIN break interrupt enable

Bit name		Function
bit15	LBIE: LIN break detection interrupt enable bit	This bit enables a reception interrupt, if a LIN break was detected.
bit14	LBD: LIN break detected flag	This bit goes 1 if a LIN break was detected. Writing a 0 to it clears this bit and the corresponding interrupt, if it is enabled. Note: RMW instructions always return "1". In this case, the value "1" does not indicate a LIN-Break.
bit13 bit12	LBL1/0: LIN break length selection	These two bits determine how many serial bit times the LIN break is generated by USART. Receiving a LIN break is always fixed to 13 bit times.
bit11	SOPE: Serial Output pin direct access enable*	Setting this bit to 1 enables the direct write to the SPIx_DO pin, if SOE = 1 (SMR).*
bit10	SIOP: Serial Input/Output Pin direct access*	Normal read instructions always return the actual value of the SPIx_DI pin. Writing to it sets the bit value to the SPIx_DO pin, if SOPE = 1. During a Read-Modify-Write instruction the bit returns the SPIx_DO value in the read cycle.*
bit9	CCO: Continuos Clock Output enable bit	This bit enables a continuos serial clock at the SPIx_SCK pin if USART operates in master mode 2 (synchronous) and the SPIx_SCK pin is configured as a clock output.
bit8	SCES: Serial clock edge selection bit	This bit inverts the internal serial clock in mode 2 (synchronous) and the output clock signal, if USART operates in master mode 2 (synchronous) and the SPIx_SCK pin is configured as a clock output. In slave mode 2 the sampling time turns from rising edge to falling edge.

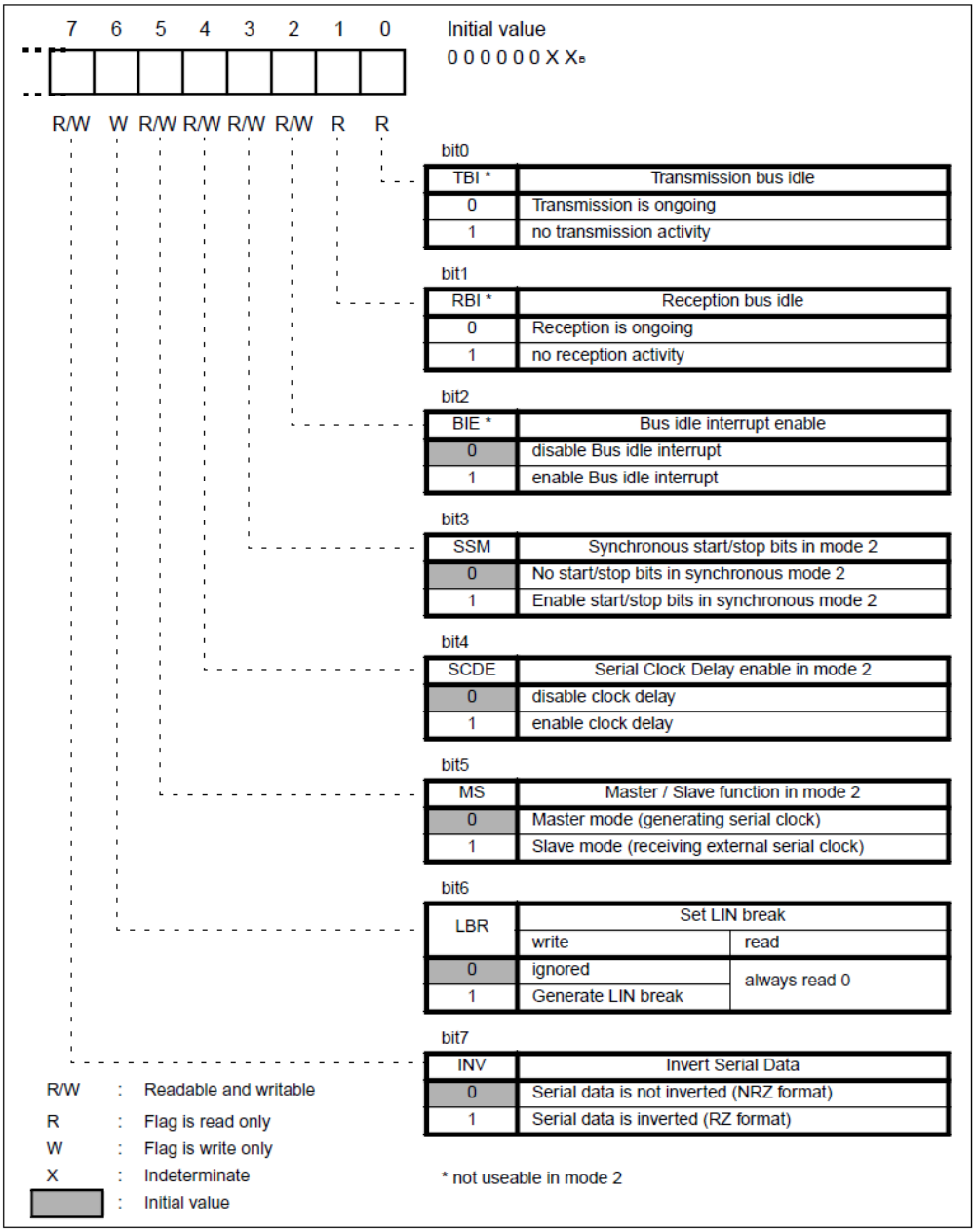
* see Table 26-5 for SOPE and SIOP interaction

Table 26-5 * Description of the interaction of SOPE and SIOP:

SOPE	SIOP	Writing to SIOP	Reading from SIOP
0	R/W	has no effect on the SPIx_DO pin but holds the written value.	returns current value of SPIx_DI
1	R/W	write "0" or "1" to SPIx_DO	returns current value of SPIx_DI
1	RMW		returns current value of SPIx_DO and writes it back

25.3.6. Extended Communication Control Register (ECCRx)

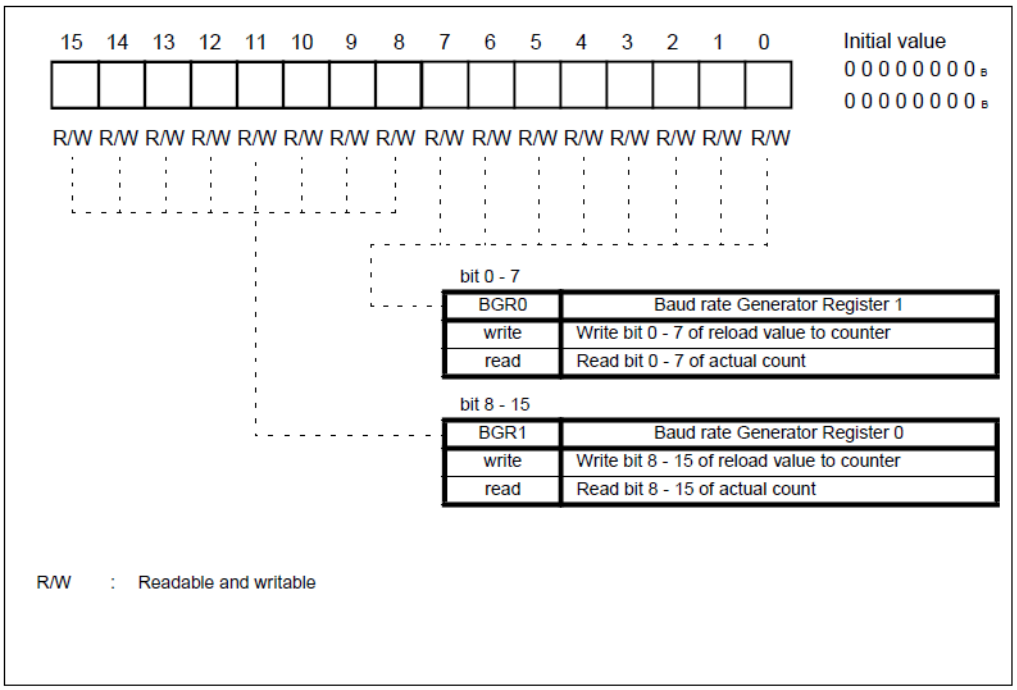
The extended communication control register provides bus idle recognition, interrupt settings, synchronous clock settings, and the LIN break generation.



Bit name		Function
bit7	INV: Invert serial data	This bit inverts the serial data at SPIx_DI and SPIx_DO pin. SPIx_SCK is not affected (see ESCR: SCES). Writing "0": The serial data format is NRZ (default) Writing "1": The serial data is inverted (RZ format) RMW instructions do not affect this bit.
bit6	LBR: Set LIN break bit	Writing a 1 to this bit generates a LIN break of the length selected by the LBL0/1 bits of the ESCR, if operation mode 0 or 3 is selected.
bit5	MS: Master/Slave mode selection bit	This bit selects master or slave mode of USART in synchronous mode 2. If master is selected USART generates the synchronous clock by itself. If slave mode is selected USART receives external serial clock. <Caution> If slave mode is selected, the clock source must be external and set to “One-to-One” (SMR: SCKE = 0, EXT = 1, OTO = 1).
bit4	SCDE: Serial clock delay enable bit	If this bit is set, the serial output clock is delayed by 1 CLKP cycle (or half of its period in SPI-compliance). This only applies, if USART operates in master mode 2.
bit3	SSM: Start/Stop bit mode enable	This bit adds start and stop bits to the synchronous data format in operation mode 2. It is ignored in mode 0, 1, and 3.
bit2	BIE: Bus idle interrupt enable	This bit enables a reception interrupt, if there is neither reception nor transmission ongoing (RBI = 1, TBI = 1). Note: Do not use BIE in mode 2.
bit1	RBI: Reception bus idle flag bit	This bit is “1” if there is no reception activity on the SPIx_DI pin. Note: Do not use this flag in mode 2.
bit0	TBI: Transmission bus idle flag bit	This bit is “1” if there is no transmission activity on the SPIx_DO pin. Note: Do not use this flag in mode 2.

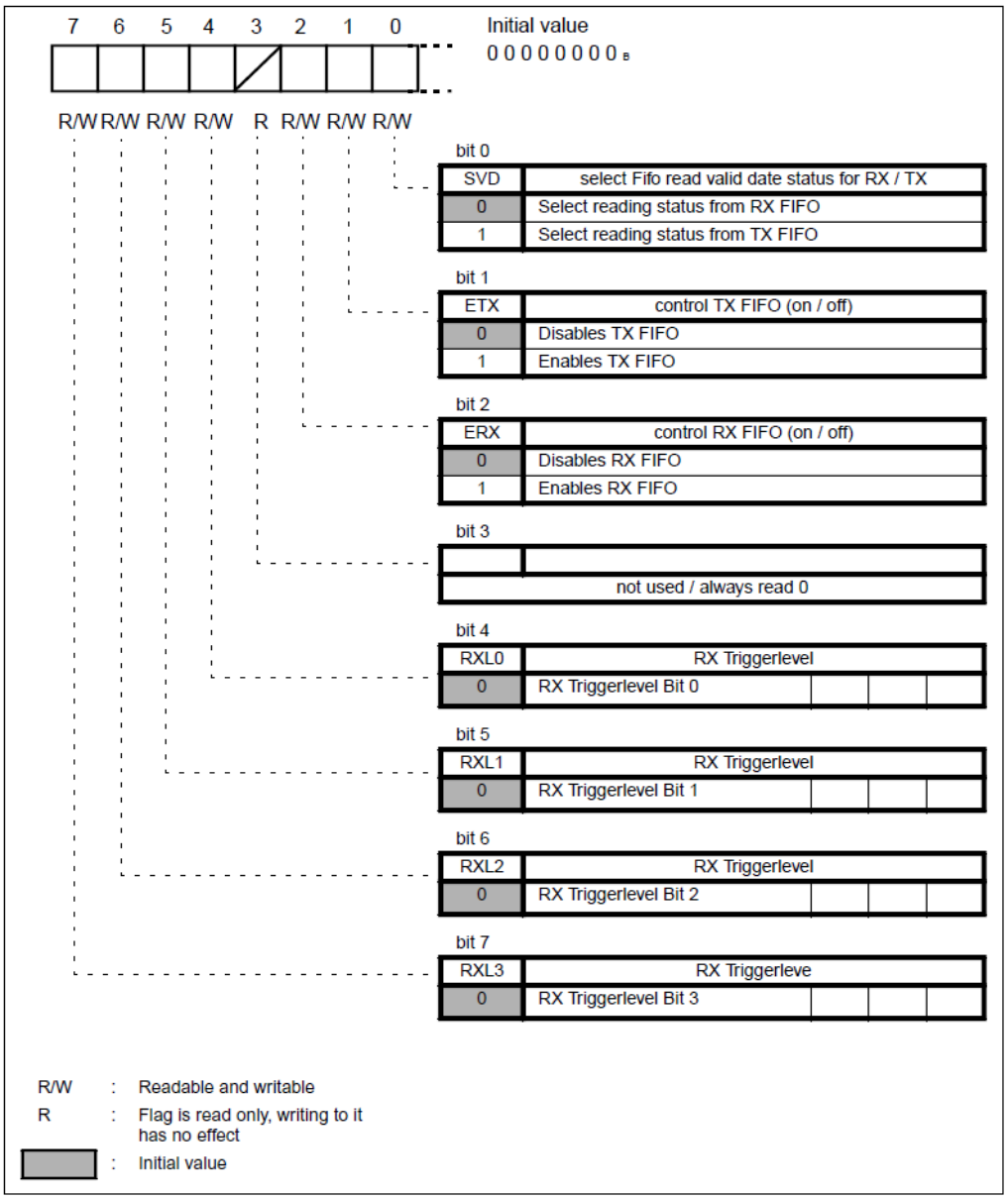
25.3.7. Baud Rate / Reload Counter Register 0 and 1 (BGRx0 / x1)

The baud rate / reload counter registers set the division ratio for the serial clock. Also the actual count of the transmission reload counter can be read.



The Baud Rate / Reload Counter Registers determine the division ratio for the serial clock. Both registers can be read or written via byte or word access.

25.3.8. FIFO Control Register (FCRx)

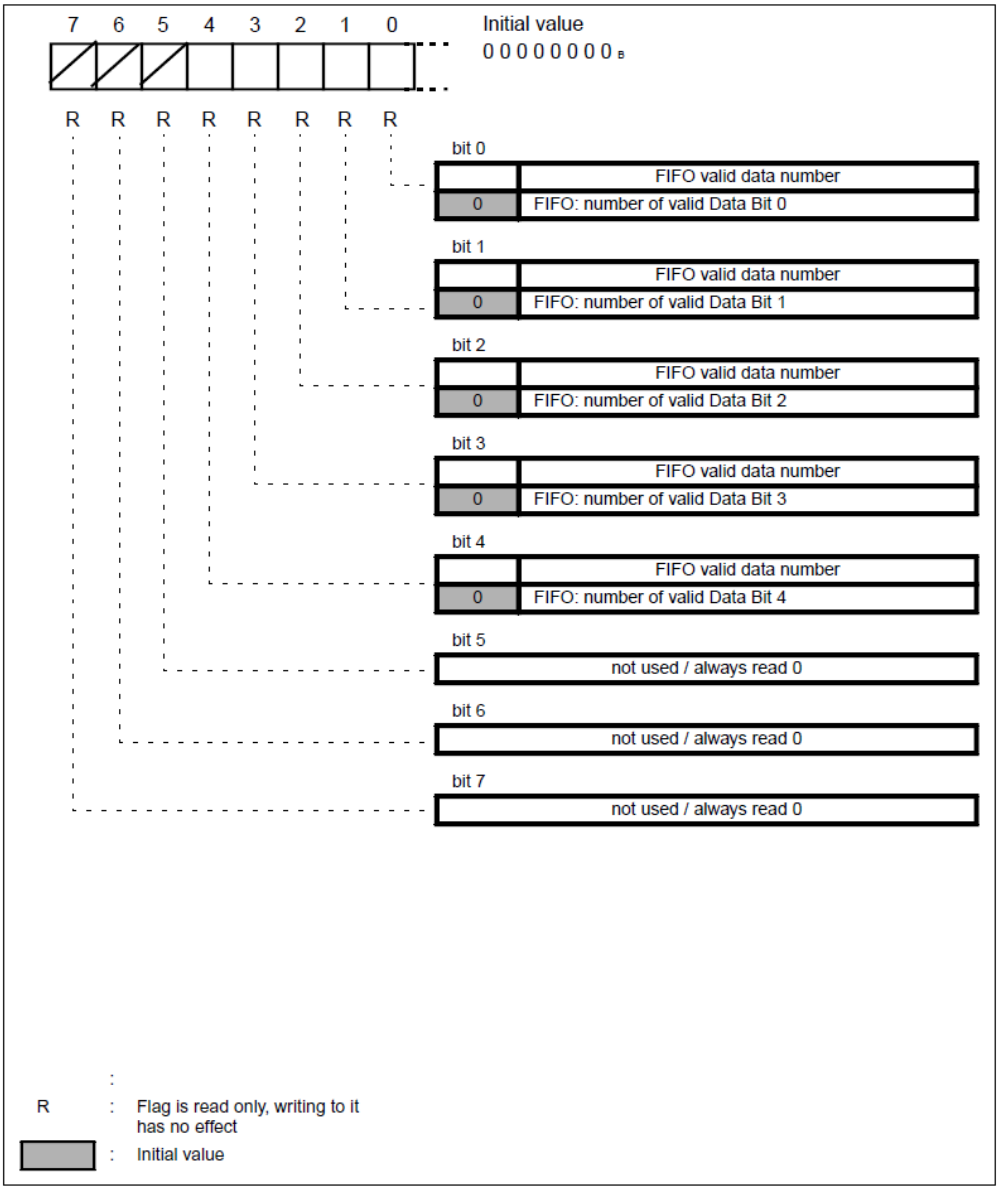


(Note):Writing the FCR results in a reset of the FIFO buffer. E.g. a reconfiguration of the trigger levels during a transmission or reception resets the FIFO buffer.

Bit name		Function
bit 0	SVD: select Valid Data Fifo read	If this bit is set to 0 the fifo status register shows the number of valid data from the RX fifo If this bit is set to 1 the fifo status register shows the number of valid data from the TX fifo
bit 1	ETX: enable TX fifo	If this bit is set to 0 the TX fifo is disabled / fifo data is cleared If this bit is set to 1 the TX fifo is enabled.
bit 2	ERX: enable RX fifo	If this bit is set to 0 the RX fifo is disabled / fifo data is cleared If this bit is set to 1 the RX fifo is enabled.
bit 3		reserved
bit 4	RXL0: RX Triggerlevel bit 0	Set the triggerlevel of RX Interrupt
bit 5	RXL1: RX Triggerlevel bit 1	Set the triggerlevel of RX Interrupt
bit 6	RXL2: RX Triggerlevel bit 2	Set the triggerlevel of RX Interrupt
bit 7	RXL3: RX Triggerlevel bit 3	Set the triggerlevel of RX Interrupt

(Note):The RX trigger level sets the reception FIFO level where the reception interrupt is activated. E.g. if the trigger level is at its default value of RXL[3:0]=0000, the interrupt is activated if one reception is stored in the FIFO. If the trigger level is set to RXL[3:0]=1111, the interrupt is activated if 16 receptions are stored in the FIFO. In general: a reception interrupt is triggered if FSR[4:0] > FCR[7:4].

25.3.9. FIFO Status Register (FSRx)



(Note): The FSR[4:0] FIFO valid data bits indicates the number of stored receptions (SVD=0) or pending transmissions (SVD=1) in the FIFO buffer.

Bit name		Function
bit 0	FIFO: number of valid Data	shows the number of valid FIFO - Data for RX and TX Fifo, depending on selection bit.
bit 1	FIFO: number of valid Data	shows the number of valid FIFO - Data for RX and TX Fifo, depending on selection bit.
bit 2	FIFO: number of valid Data	shows the number of valid FIFO - Data for RX and TX Fifo, depending on selection bit..
bit 3	FIFO: number of valid Data	shows the number of valid FIFO - Data for RX and TX Fifo, depending on selection bit..
bit 4	FIFO: number of valid Data	shows the number of valid FIFO - Data for RX and TX Fifo, depending on selection bit.

25.4. USART Interrupts

The USART uses both reception and transmission interrupts. An interrupt request can be generated for either of the following causes:

- Receive data is set in the Reception Data Register (RDR), or a reception error occurs.
- Transmission data is transferred from the Transmission Data Register (TDR) to the transmission shift register.
- A LIN break is detected
- No bus activity (neither reception nor transmission)

USART Interrupts

Reception/ transmission/ ICU	Interrupt request flag bit	Flag Register	Operation mode				Interrupt cause	Interrupt cause enable bit	How to clear the Interrupt Request
			0	1	2	3			
Reception	RDRF	SSR	x	x	x	x	receive data is written to RDR (FIFO level reached)	SSR : RIE	Receive data is read
	ORE	SSR	x	x	x	x	Overrun error		"1" is written to clear rec. error bit (SCR: CRE)
	FRE	SSR	x	x	*	x	Framing error		
	PE	SSR	x		*		Parity error		
	LBD	ESCR	x			x	LIN synch break detected	ESCR0 : LBIE	"0" is written to ESCR0 : LBD
	TBI & RBI	ESCR	x	x		x	no bus activity	ECCR : BIE	Receive data / Send data
Transmission	TDRE	SSR	x	x	x	x	Empty transmission register	SS0R : TIE	Transfer data is written
Input Capture Unit	ICP	IPCP	x			x	1st falling edge of LIN synch field	IPCP : ICE	disable ICE temporary
	ICP	IPCP	x			x	5th falling edge of LIN synch field	IPCP : ICE	disable ICE

Table 26-6 Interrupt control bits and interrupt causes of USART

x : Used

* : Only available if ECCR/SSM = 1

Reception Interrupt

If one of the following events occur in reception mode, the corresponding flag bit of the Serial Status Register (SSR) is set to "1":

- - Data reception is complete, i. e. the received data was transferred from the serial input shift register to the Reception Data Register (RDR) and data can be read: **RDRF** (if FIFO is enabled, trigger level is reached)
- - Overrun error, i. e. RDRF = 1 and RDR was not read by the CPU: **ORE**
- - Framing error, i. e. a stop bit was expected, but a "0"-bit was received: **FRE**
- - Parity error, i. e. a wrong parity bit was detected: **PE**

If at least one of these flag bits above go "1" and the reception interrupt is enabled (SSR: RIE = 1), a reception interrupt request is generated.

If the Reception Data Register (RDR) is read, the RDRF flag is automatically cleared to "0". Note that this is the *only* way to reset the RDRF flag. The error flags are cleared to "0", if a "1" is written to the Clear Reception Error (CRE) flag bit of the Serial Control Register (SCR). The RDR contains only valid data if the RDRF flag is "1" and no error bits are set.

Note that the CRE flag is "write only" and by writing a "1" to it, it is internally held to "1" for one CPU clock cycle.

Transmission Interrupt

If transmission data is transferred from the Transmission Data Register (TDR) to the transfer shift register (this happens, if the shift register (or FIFO) is empty and transmission data exists), the Transmission Data Register Empty flag bit (TDRE) of the Serial Status Register (SSR) is set to "1". In this case an interrupt request is generated, if the Transmission Interrupt Enable (TIE) bit of the SSR was set to "1" before.

Note, that the initial value of TDRE (after hardware or software reset) is "1". So an interrupt is generated immediately then, if the TIE flag is set to "1". Also note, that the *only* way to reset the TDRE flag is writing data to the Transmission Data Register (TDR).

LIN Synchronization Break Interrupt

This paragraph is only relevant, if operates in mode 0 or 3 as a LIN slave.

If the bus (serial input) goes "0" (dominant) for more than 11 bit times, the LIN Break Detected (LBD) flag bit of the Extended Status/Control Register (ESCR) is set to "1". Note, that in this case after 9 bit times the reception error flags are set to "1", therefore the RIE flag has to be set to "0" or the RXE flag has to be set to "0", if only a LIN synch break detect is desired. In the other case a reception error interrupt would be generated first, and the interrupt handler routine has then to wait for LBD = 1.

The interrupt and the LBD flag are cleared after writing a "1" to the LBD flag. This makes sure, that the CPU has detected the LIN synch break, because of the following procedure of adjusting the serial clock to the LIN master.

LIN Synchronization Field Edge Detection Interrupts

This paragraph is only relevant, if USART operates in mode 0 or 3 as a LIN slave. After a LIN break detection the next falling edge of the reception bus is indicated by. Simultaneously an internal signal connected to the ICU is set to "1". This signal is reset to "0" after the fifth falling edge of the LIN Synchronization Field. In both cases the ICU4 generates an interrupt, if "both edge detection" and the ICU interrupt are enabled. The difference of the ICU4 counter values is the serial clock multiplied by 8. Dividing it by 8 results in the new detected and calculated baud rate for the dedicated reload counter. This value - 1 has then to be written to the Baud Rate Generator Registers (BGR1/0).There is no need to restart the reload counter, because it is automatically reset if a falling edge of a start bit is detected.

Bus Idle Interrupt

If there is no reception activity on the SPIx_DI pin, the RBI flag bit of the ECCR goes "1". The TBI flag bit respectively goes "1", when no data is transmitted. If the Bus Idle Interrupt Enable bit (BIE) of the ECCR is set and **both** bus idle flag bits (TBI **and** RBI) are "1", an interrupt is generated.

(Note): The TBI flag goes also "0" if there is no bus activity, but a "0" is written to the SIOP bit, if SOPE is "1".

(Note): TBI nd RBI cannot be used in mode 2 (synchronous communication).

Figure 26-2 illustrates how the bus idle interrupt is generated

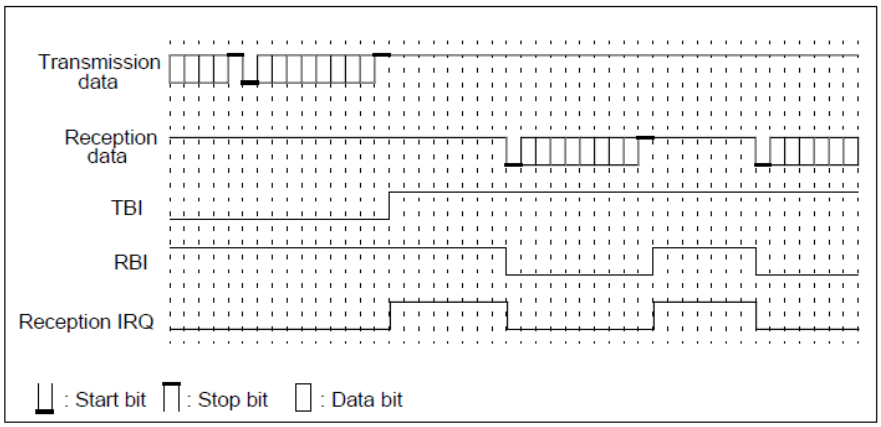


Figure 26-2 Bus idle interrupt generation

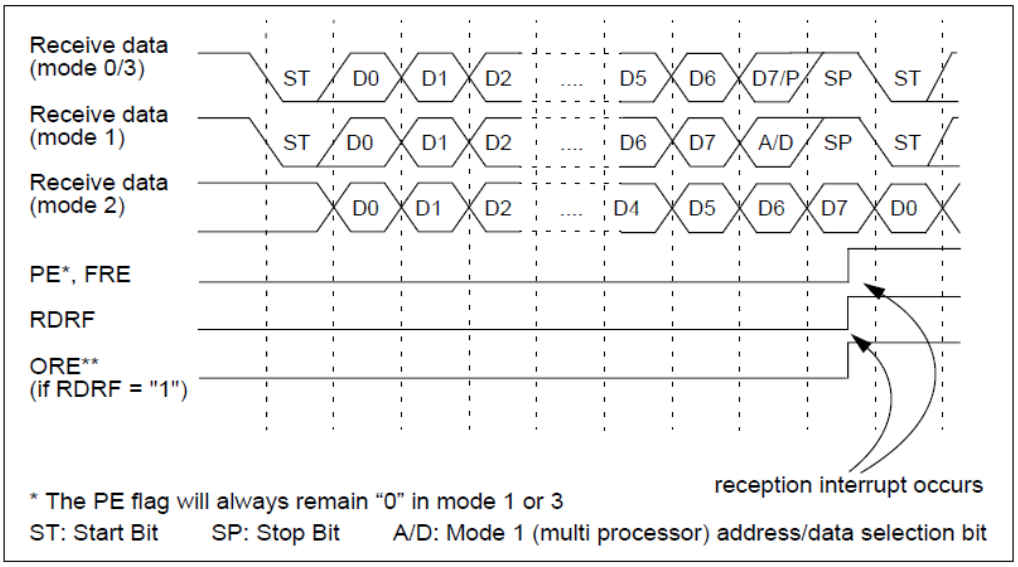
25.4.1. Reception Interrupt Generation and Flag Set Timing

The following are the reception interrupt causes: Completion of reception (SSR: RDRF) and occurrence of a reception error (SSR: PE, ORE, or FRE).

Reception Interrupt Generation and Flag Set Timing

Generally a reception interrupt is generated if the received data is complete (RDRF = 1) and the Reception Interrupt Enable (RIE) flag bit of the Serial Status Register (SSR) was set to "1". This interrupt is generated if the first stop bit is detected in mode 0, 1, 2 (if SSM = 1), 3, or the last data bit was read in mode 2 (if SSM = 0).

(Note): If a reception error has occurred, the Reception Data Register (RDR) contains invalid data in each mode.



**ORE only occurs, if the reception data is not read by the CPU (RDRF = 1) and another data frame is read.

Figure 26-3 Reception operation and flag set timing

(Note)The example in figure 21-3 does not show all possible reception options for mode 0 and 3. Here it is: "7p1" and "8N1" (p = "E" [even] or "O" [odd]), all in NRZ data format (ECCR: INV = 0).

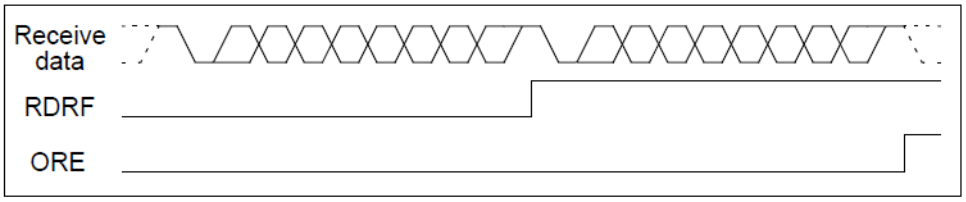


Figure 26-4 ORE set timing

25.4.2. Transmission Interrupt Generation and Flag Set Timing

A transmission interrupt is generated when the next data to be sent is ready to be written to the transmission data register (TDR).

Transmission Interrupt Generation and Flag Set Timing

A transmission interrupt is generated, when the next data to be send is ready to be written to the Transmission Data Register (TDR), i.e. the TDR is empty, and the transmission interrupt is enabled by setting the Transmission Interrupt Enable (TIE) bit of the Serial Status Register (SSR) to "1".

The Transmission Data Register Empty (TDRE) flag bit of the SSR indicates an empty TDR. Because the TDRE bit is "read only", it only can be cleared by writing data into TDR.

The following figure demonstrates the transmission operation and flag set timing for the four modes of USART.

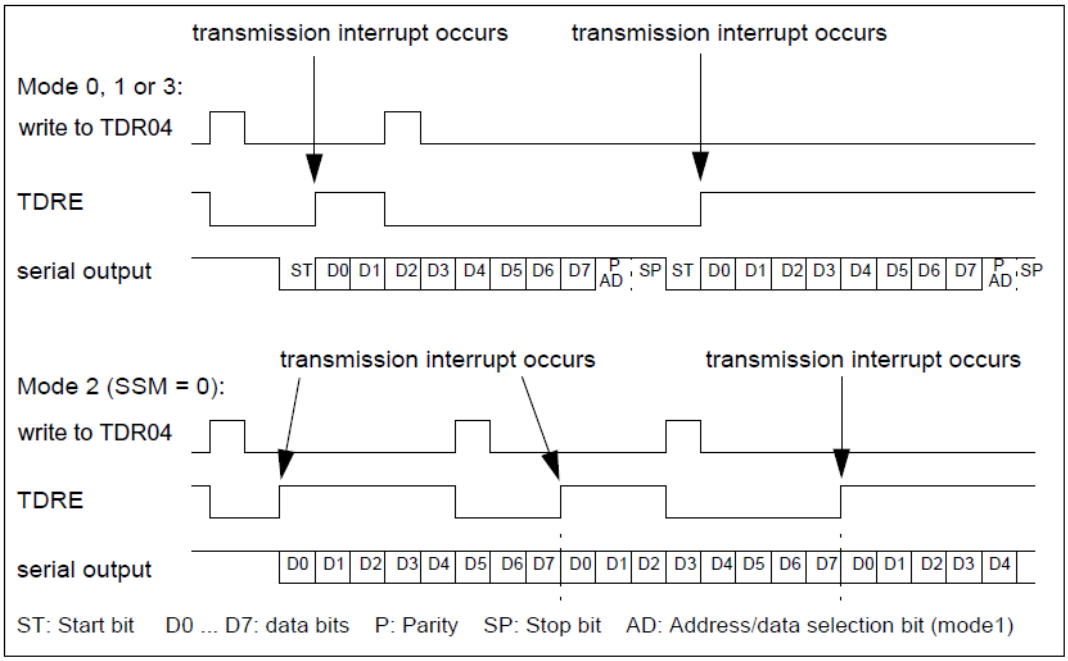


Figure 26-5 Transmission operation and flag set timing

(Note)The example in figure 21-5 does not show all possible transmission options for mode 0. Here it is: "8p1" (p = "E" [even] or "O" [odd]), ECCR: INV = 0. Parity is not provided in mode 3 or 2, if SSM = 0.

Transmission Interrupt Request Generation Timing

If the TDRE flag is set to 1 when a transmission interrupt is enabled (SSR: TIE=1) a transmission interrupt request is generated.

A transmission completion interrupt is generated immediately after the transmission interrupt is enabled (TIE=1) because the TDRE bit is set to 1 as its initial value. TDRE is a read-only bit that can be cleared only by writing new data to the output data register (TDR). Carefully specify the transmission interrupt enable timing.

25.5. USART Baud Rates

One of the following can be selected for the serial clock source:

- Dedicated baud rate generator (Reload Counter)
- External clock as it is (clock input to the SPIx_SCK pin)
- External clock connected to the baud rate generator (Reload Counter)

Baud Rate Selection

The baud rate selection circuit is designed as shown below. One of the following three types of baud rates can be selected:

- Baud Rates Determined Using the Dedicated Baud Rate Generator (Reload Counter)

USART has two independent internal reload counters for transmission and reception serial clock. The baud rate can be selected via the 15-bit reload value determined by the Baud Rate Generator Register 0 and 1 (BGR0/1).

The reload counter divides the peripheral clock by the value set in the Baud Rate Generator Register 0 and 1.

- Baud Rates determined using external clock (one-to-one mode)

The clock input from USART clock pulse input pins (SPIx_SCK) is used as it is (synchronous). Any baud rate less than the peripheral clock divided by 4 and is divisible can be set externally

- Baud Rates determined using the dedicated baud rate generator with external clock

An external clock source can also be connected internally to the reload counter. In this mode it is used instead of the internal peripheral clock. This was designed to use quartz oscillators with special frequencies and having the possibility to divide them.

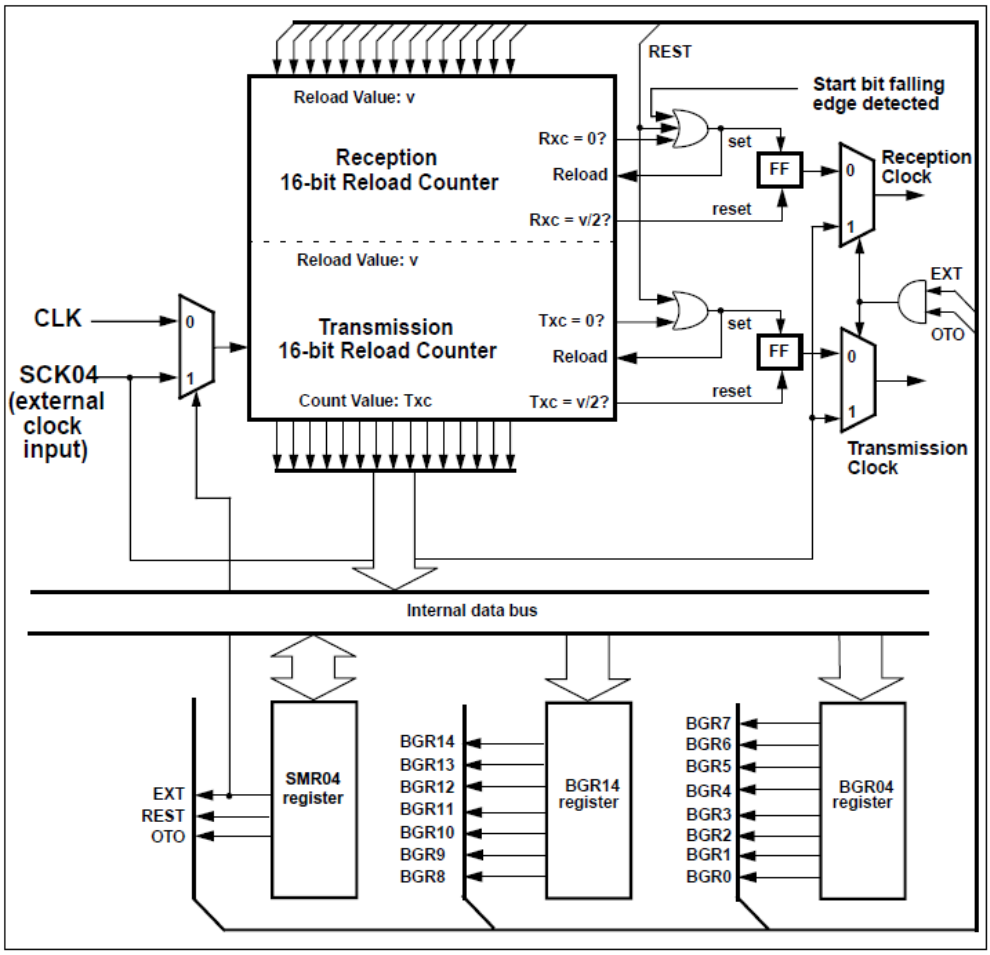


Figure 26-6 Baud rate selection circuit (reload counter)

25.5.1. Setting the Baud Rate

This section describes how the baud rates are set and the resulting serial clock frequency is calculated.

Calculating the baud rate

The both 15-bit Reload Counters are programmed by the Baud Rate Generator Registers 1 and 0 (BGR1, 0). The following calculation formula should be used to set the wanted baud rate:
Reload Value:

$$v = [F / b] - 1 \quad ,$$

where F is the resource clock (CLKP), b the baud rate and [ ] gaussian brackets (mathematical rounding function).

Example of Calculation

If the CPU clock is 41.66MHz and the desired baud rate is 19200 baud then the reload value v is:

$v = [41.66 * 10^6 / 19200] - 1 =$ **2169**

The exact baud rate can then be recalculated: $b_{exact} = F / (v + 1)$, here it is: $41.66 * 10^6 / 2170 = 19198.16$

(Note) Setting the reload value to 0 stops the reload counter.

(Note) The minimum recommended division ratio is 4 (i.e. reload value is 3) due to RX oversampling filter in asynchronous communication modes (mode 0,1 and 3).

Suggested Division Ratios for different machine speeds and baud rates

The following settings are suggested for different clock speeds and baud rates:

Table 26-7 Suggested Baud Rates and reload values at different machine speeds.

Baud rate	10.42MHz		20.83MHz		41.66MHz	
	value	% dev.	value	% dev.	value	% dev.
250000	-	-	-	-	166	0.22
230400	-	-	-	-	180	0.1
153600	-	-	-	-	270	-0.08
125000	-	-	166	0.22	332	-0.08
115200	-	-	180	0.1	361	0.1
76800	-	-	270	-0.08	541	-0.08
57600	180	0.05	361	0.1	722	-0.04
38400	270	-0.13	541	-0.08	1084	0.01
28800	361	0.05	722	-0.04	1446	0.03
19200	542	0.05	1084	0.01	2169	0.01
10417	999	-0.03	1999	0.02	3998	-0.01
9600	1084	-0.04	2169	0.01	4339	0.01
7200	1446	-0.02	2882	<0.01	5785	<0.01
4800	2170	0.01	4339	0.01	8678	<0.01
2400	4341	0.01	8678	<0.01	17357	<0.01
1200	8682	<0.01	17357	<0.01	-	-
600	17366	<0.01	-	-	-	-

Counting Example

Assume the reload value is 832. The Figure 26-7 demonstrates the behavior of the both Reload Counters:

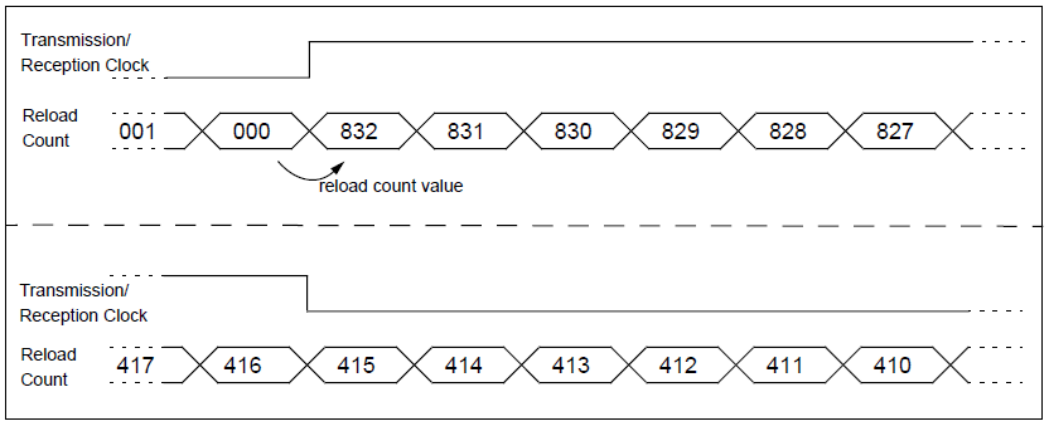


Figure 26-7 Counting example of the reload counters

(Note) The falling edge of the Serial Clock Signal always occurs after | (v + 1) / 2 |.

25.5.2. Restarting the Reload Counter

The Reload Counter can be restarted because of the following reasons:

Transmission and Reception Reload Counter:

- Global Reset
- Programmable clear (SMR: UPCL bit)
- User programmable restart (SMR: REST bit)

Reception Reload Counter:

- Start bit falling edge detection in asynchronous mode

Programmable Restart

If the REST bit of the Serial Mode Register (SMR) is set by the user, both Reload Counters are restarted at the next clock cycle. This feature is intended to use the Transmission Reload Counter as a small timer.

The following figure illustrates a possible usage of this feature (assume that the reload value is 100.)

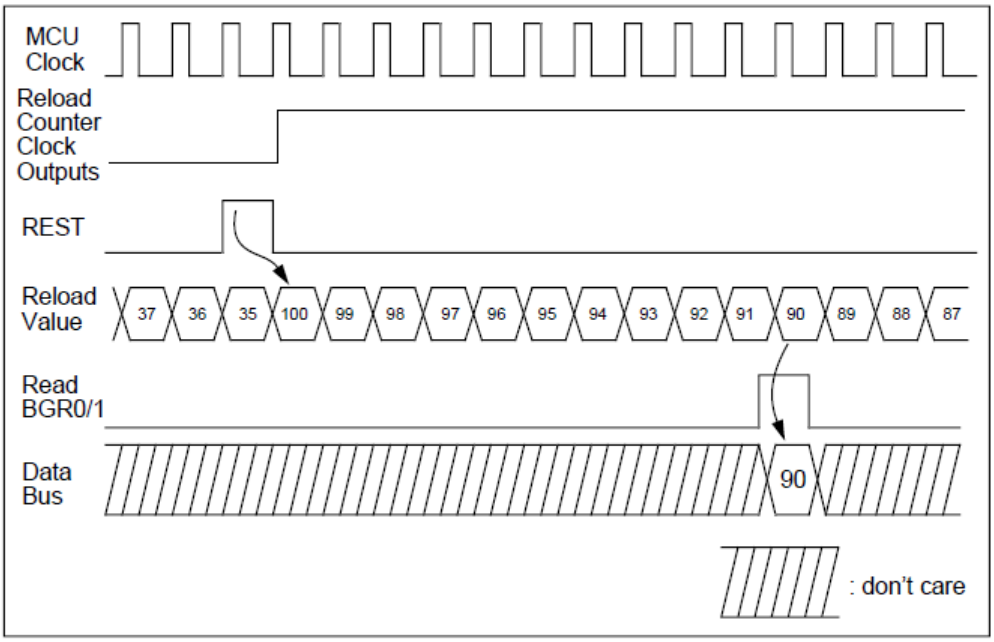


Figure 26-8 Reload Counter Restart example

In this example the number of clock cycles (cyc) after REST is then:

***cyc* = *v* - *c* + 1 = 100 - 90 + 1 = 11**,

where *v* is the reload value and *c* is the read counter value.

(Note) If is reset by setting SMR:UPCL, the Reload Counters will restart too.

Automatic Restart

In asynchronous UART mode if a falling edge of a start bit is detected, the Reception Reload Counter is restarted. This is intended to synchronize the serial input shifter to the incoming serial data stream.

25.6. USART Operation

USART operates in operation mode 0 for normal bidirectional serial communication, in mode 2 and 3 in bidirectional communication as master or slave, and in mode 1 as master or slave in multiprocessor communication.

Operation of USART

- Operation modes

There are four USART operation modes: modes 0 to 3. As listed in table 26-8, an operation mode can be selected according to the inter-CPU connection method and data transfer mode.

Table 26-8 operation mode

Operation mode		Data length		Synchroniza-tion of mode	Length of stop bit	data bit direction*
		parity disabled	parity enabled			
0	normal mode	7 or 8		asynchronous	1 or 2	L/M
1	multiprocessor	7 or 8 + 1**	-	asynchronous	1 or 2	L/M
2	normal mode	8		synchronous	0, 1 or 2	L/M
3	LIN mode	8	-	asynchronous	1	L

* means the data bit transfer format: LSB or MSB first

** "+1" means the indicator bit of the address/data selection in the multiprocessor mode, instead of parity.

(Note) Mode 1 operation is supported both for master or slave operation of USART in a master-slave connection system. In Mode 3 the USART function is locked to 8N1-Format, LSB first.

If the mode is changed, USART cuts off all possible transmission or reception and awaits then new action.

Inter-CPU Connection Method

External Clock One-to-one connection (normal mode) and master-slave connection (multiprocessor mode) can be selected. For either connection method, the data length, whether to enable parity, and the synchronization method must be common to all CPUs. Select an operation mode as follows:

- In the one-to-one connection method, operation mode 0 or 2 must be used in the two CPUs. Select operation mode 0 for asynchronous transfer mode and operation mode 2 for synchronous transfer mode.

Note, that one CPU has to set to the master and one to the slave in synchronous mode 2.

- Select operation mode 1 for the master-slave connection method and use it either for the master or slave system.

Synchronization Methods

In asynchronous operation USART reception clock is automatically synchronized to the falling edge of a received start bit.

In synchronous mode the synchronization is performed either by the clock signal of the master device or by USART itself if operating as master.

Signal Mode

USART can treat data in non-return to zero (NRZ) and return to zero (RZ) format. For this option the ECCR: INV bit is provided.

Operation Enable Bit

USART controls both transmission and reception using the operation enable bit for transmission (SCR: TXE) and reception (SCR: RXE). If each of the operations is disabled, stop it as follows:

- If reception operation is disabled during reception (data is input to the reception shift register), finish frame reception and read the received data of the reception data register (RDR). Then stop the reception operation.
- If the transmission operation is disabled during transmission (data is output from the transmission shift register), wait until there is no data in the transmission data register (TDR) before stopping the transmission operation.

25.6.1. Operation in Asynchronous Mode (Op. Modes 0 and 1)

When USART is used in operation mode 0 (normal mode) or operation mode 1 (multiprocessor mode), the asynchronous transfer mode is selected.

Transfer data format

Generally each data transfer in the asynchronous mode operation begins with the start bit (low-level on bus) and ends with at least one stop bit (high-level). The direction of the bit stream (LSB first or MSB first) is determined by the BDS-Bit of the Serial Status Register (SSR). The parity bit (if enabled) is always placed between the last data bit and the (first) stop bit.

In operation mode 0 the length of the data frame can be 7 or 8 bits, with or without parity, and 1 or 2 stop bits.

In operation mode 1 the length of the data frame can be 7 or 8 bits with a following address-/data-selection bit instead of a parity bit. 1 or 2 stop bits can be selected.

The calculation formula for the bit length of a transfer frame is:

Length = 1 + d + p + s

(d = number of data bits [7 or 8], p = parity [0 or 1], s = number of stop bits [1 or 2]

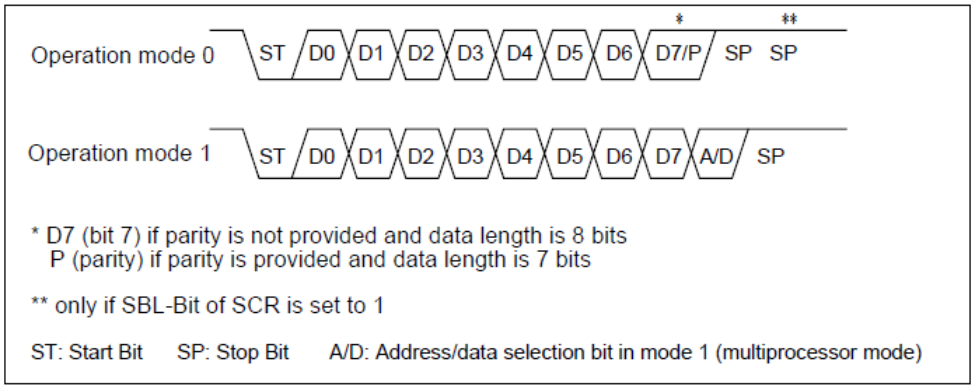


Figure 26-9 Transfer data format (operation modes 0 and 1)

(Note) If BDS-Bit of the Serial Status Register (SSR) is set to "1" (MSB first), the bit stream processes as: D7, D6, ... , D1, D0, (P).

During Reception both stop bits are detected, if selected. But the Reception data register full (RDRF) flag will go "1" at the first stop bit. The bus idle flag (RBI of ECCR) goes "1" after the second stop bit if no further start bit is detected. (The second stop bit belongs to "bus activity", although it is just mark level.)

Transmission Operation

If the Transmission Data Register Empty (TDRE) flag bit of the Serial Status Register (SSR) is "1", transmission data is allowed to be written to the Transmission Data Register (TDR). When data is written, the TDRE flag goes "0". If the transmission operation is enabled by the TXE-Bit ("1") of the Serial Control Register (SCR), the data is written next to the transmission shift register and the transmission starts at the next clock cycle of the serial clock, beginning with the start bit. Thereby the TDRE flag goes "1", so that new data can be written to the TDR.

If transmission interrupt is enabled (TIE = 1), the interrupt is generated by the TDRE flag. Note, that the initial value of the TDRE flag is "1", so that in this case if TIE is set to "1" an interrupt will occur immediately.

Reception Operation

Reception operation is performed every time it is enabled by the Reception Enable (RXE) flag bit of the SCR. If a start bit is detected, a data frame is received according to the format specified by the SCR. By occurring errors, the corresponding error flags are set (PE, ORE, FRE). However after the reception of the data frame the data is transferred from the serial shift register to the Reception Data Register (RDR) and the Receive Data Register Full (RDRF) flag bit of the SSR is set. The data then has to be read by the CPU.

By doing so, the RDRF flag is cleared. If reception interrupt is enabled (RIE = 1), the interrupt is simply generated by the RDRF.

Note: Only when the RDRF flag bit is set and no errors have occurred does the Reception Data Register (RDR) contain valid data.

Stop Bit, Error Detection, and Parity

For transmission, 1 or 2 stop bits can be selected. During reception, if selected, both stop bits are checked, to set the reception bus idle (RBI) flag of ECCR correctly after the second stop bit.

In mode 0 parity, overrun, and framing errors can be detected.

In mode 1, overrun and framing errors can be detected. Parity is not provided.

By setting the Parity Enable (PEN) bit of the Serial Control Register (SCR) the USART provides parity calculation (during transmission) and parity detection and check (during reception) in mode 0 (and mode 2 if the SSM bit of ECCR is set).

Even parity is set, if the P bit of SCR is cleared, odd parity if the flag bit is set. In mode 1, overrun and framing errors can be detected. Parity is not provided.

Signal mode NRZ and RZ

To set USART to the NRZ data format, set the ECCR:INV bit to 0 (initial value).

RZ data format is set, if the ECCR:INV bit was set to 1.

25.6.2. Operation in Synchronous Mode (Operation Mode 2)

The clock synchronous transfer method is used for USART operation mode 2 (normal mode).

Transfer data format (standard synchronous)

In the synchronous mode, 8-bit data is transferred with no start or stop bits if the SSM bit of the Extended Communication Control Register (ECCR) is 0. A special clock signal belongs to the data format in mode 2. The figure below illustrates the data format during a transmission in the synchronous operation mode

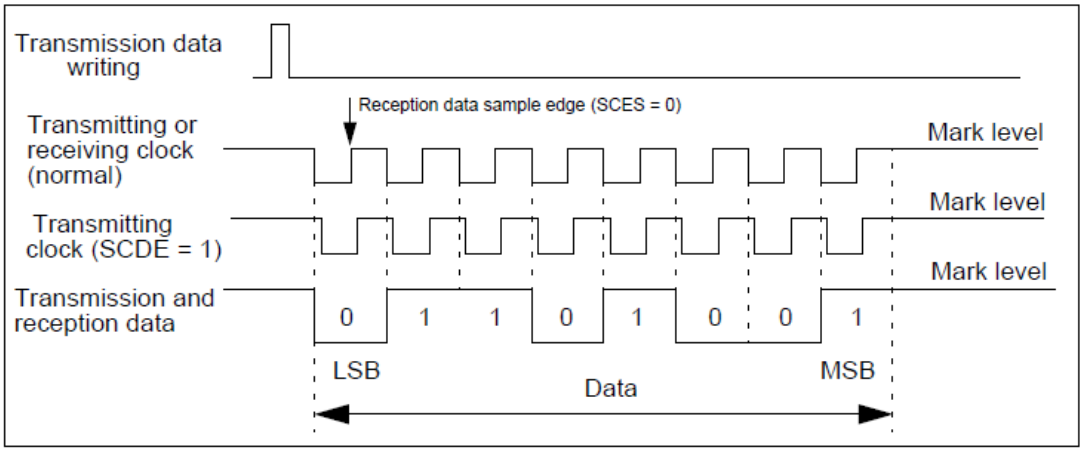


Figure 26-10 Transfer data format (operation mode 2).

Transfer data format

In the synchronous mode, 8-bit data is transferred with no start or stop bits if the SSM bit of the Extended Communication Control Register (ECCR) is 0. A special clock signal belongs to the data format in mode 2. The figure below illustrates the data format during a transmission in the synchronous operation mode.

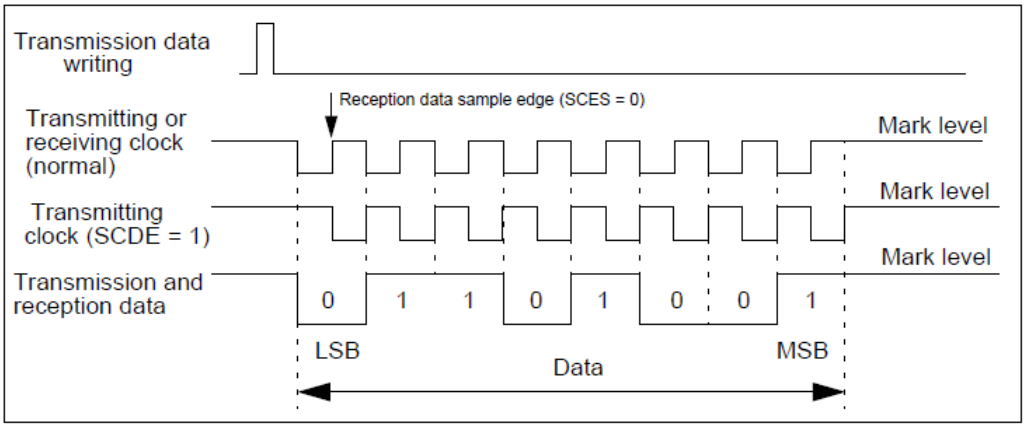


Figure 26-11 SPI Transfer data format (operation mode 2)

Clock inversion and start/stop bits in mode 2

If the SCES bit of the Extended Status/Control Register (ESCR) is set the serial clock is inverted. Therefore in slave mode USART samples the data bits at the falling edge of the received serial clock. Note, that in master mode if SCES is set the clock signal's mark level is "0". If the SSM bit of the Extended Communication Control Register (ECCR) is set the data format gets additional start and stop bits like in asynchronous mode

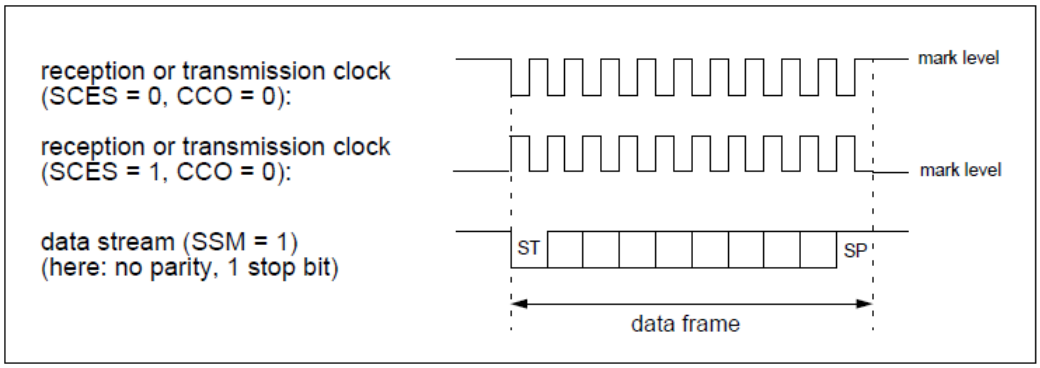


Figure 26-12 Transfer data format with clock inversion

Clock Supply

In clock synchronous (normal) mode (I/O extended serial), the number of the transmission and reception bits has to be equal to the number of clock cycles. Note, that if start/stop bits communication is enabled, the number of clock cycles has to match with the quantity for the additional start and stop bit(s).

If the internal clock (dedicated reload counter) is selected, the data receiving synchronous clock is generated automatically if data is transmitted.

If external clock is selected, be sure, that the transmission side of the Transmission Data Register contains data and then clock cycles for each bit to sent have to be generated and supplied from outside. The mark level ("H") must be retained before transmission starts and after it is complete if SCES is “0”.

Setting the SCDE bit of ECCR delays the transmitting clock signal by 1 CLKP cycle (or half a clock period in SPI). This will make sure, that the transmission data is valid and stable at any falling clock edge. (Necessary, if the receiving device samples the data at falling clock edge). This function is disabled when CCO is enabled.

If the Serial Clock Edge Select (SCES) bit of the ESCR is set, the USARTs clock is inverted and thus samples the reception data at the falling clock edge. In this case, the sending device must make sure that the serial data is valid at the falling serial clock edge.

When both the SCES and the SCDE bit are set, data is stable at the rising clock edge, as in the case of SCES = SCDE = 0. However, the marker value for idle state is inverted (low).

If the CCO bit of the Extended Status/Control Register (ESCR) is set, the serial clock on the SPIx_SCK pin in master mode is continuously clocked out. It is strongly recommended to use start and stop bits in this mode to signalize the receiver, when a data frame begins and when it stops. Figure 26-13 illustrates this.

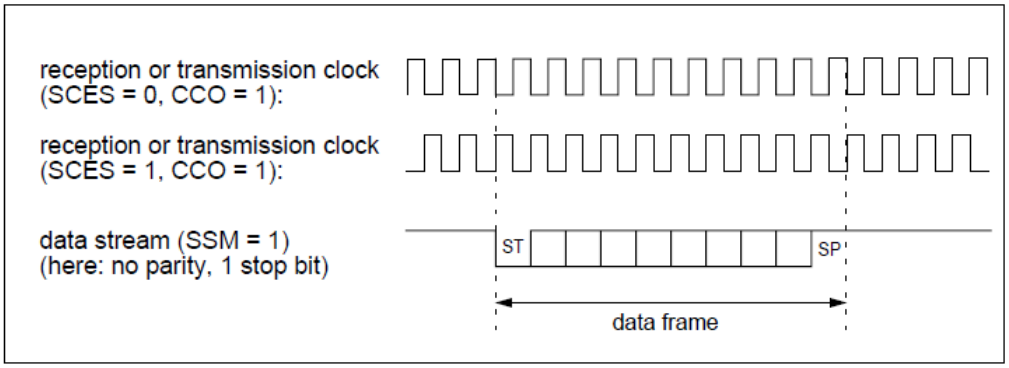


Figure 26-13 Continuous clock output in mode 2

Data signal mode

NRZ data format is selected, if ECCR: INV = 0, otherwise the signal mode for the serial data input and output pin is RZ.

Error Detection

If no Start/Stop bits are selected (ECCR: SSM = 0) only overrun errors are detected.

Communication

For initialization of the synchronous slave mode, the following settings have to be done:

- Baud Rate Generator Registers (BGR0/1):

Set the desired reload value for the dedicated Baud Rate Reload Counter

- Serial Control Register (SCR):
- RXE, TXE: set both of these flags to “0“
- PEN: no parity provided - Value: don’t care
- P, SBL, A/D: no parity, no stop bit(s), no Address/Data selection - Value: don’t care
- CL: automatically fixed to 8-bit data - Value: don’t care
- CRE: "1" (the error flag is cleared for initialization, possible transmission or reception will cut off)
- Serial Mode Control Register (SMR):
- MD1, MD0: "10b" (Mode 2)
- SCKE: “1” for dedicated Baud Rate Reload Counter
  “0” for external clock input
- SOE: “1” for transmission and reception
  “0” for reception only¥
- Serial Status Register (SSR):
- BDS: "0" for LSB first, “1" for MSB first
- RIE: “1" if interrupts are used; "0" if not
- TIE: "1" if interrupts are used; "0" if not
- Extended Communication Control Register (ECCR):
- SSM: "0" if no start/stop bits are desired (normal)
  "1" for adding start/stop bits (special)
- MS: "0" for master mode (USART generates the serial clock);
  "1" for slave mode (USART receives serial clock from the master device)
- Serial Control Register (SCR):
- RXE, TXE: set one or both of these control bits to "1" to begin communication.

To start the communication, write data into the Transmission Data Register (TDR).

To receive data, disable the Serial Output Enable (SOE) bit of the SMR and write dummy data to TDR.

(Note) Setting continuous clock and start-/stop-bit mode, duplex transfer is possible like in asynchronous modes.

25.6.3. Operation with LIN Function (Operation Mode 3)

USART can be used either for LIN-Master devices or LIN-Slave devices. For this LIN function a special mode (3) is provided. Setting the USART to mode 3, configure the data format to 8N1-LSB-first format.

USART as LIN master

In LIN master mode, the master determines the baud rate of the whole sub bus. Therefore, slave devices have to synchronize to the master and the desired baud rate remains fixed in master operation after initialization.

Writing a "1" into the LBR bit of the Extended Status/Communication Register (ECCR) generates a 13 - 16 bit times low-level on the SPIx_DO pin, which is the LIN synchronization break and the start of a LIN message. Thereby the TDRE flag of the Serial Status Register (SSR) goes "0" and is reset to "1" after the break, and generates a transmission interrupt for the CPU (if TIE of SSR is "1").

The length of the Synchronization break to be sent can be determined by the LBL1/0 bits of the ESCR as follows:

Table 26-9 LIN break length

LBL1	LBL0	Length of Break
0	0	13 Bit times
0	1	14 Bit times
1	0	15 Bit times
1	1	16 Bit times

The Synch Field can be sent as a simple 0x55-Byte after the LIN break. To prevent a transmission interrupt, the 0x55 can be written to the TDR just after writing the "1" to the LBR bit, although the TDRE flag is "0". The internal transmission shifter waits until the LIN break has finished and shifts the TDR value out afterwards. In this case no interrupt is generated after the LIN break and before the start bit.

USART as LIN slave

In LIN slave mode USART has to synchronize to the master's baud rate. If Reception is disabled (RXE = 0) but LIN break Interrupt is enabled (LBIE = 1) USART will generate a reception interrupt, if a synchronization break of the LIN master is detected, and indicates it with the LBD flag of the ESCR. Writing a "0" to this bit clears the interrupt. The next step is to analyze the baud rate of the LIN master. The first falling edge of the Synch Field is detected by USART. The USART signals it then to the Input Capture Unit (ICU1/5) via a rising edge of an internal connection. The fifth falling edge resets the ICU signal. Therefore the ICU has to be configured for the LIN input capture and its interrupts have to be enabled (ICS). The values of th ICU counter register after the first Interrupt (a) and after the second interrupt (b) yield the BGR value:

without timer overflow: BGR value = (b - a) / 8 ,

with timer overflow: BGR value = (max - b + a) / 8 ,

where max is the timer maximum value at which the overflow occurs.

The figure 26-14 shows a typical start of a LIN message frame and the behavior of the USART.

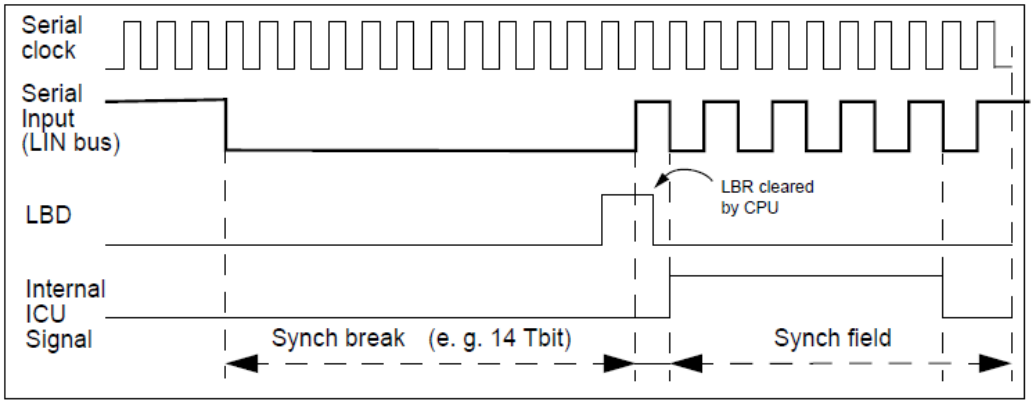


Figure 26-14 USART behavior as slave in LIN mode

LIN bus timing

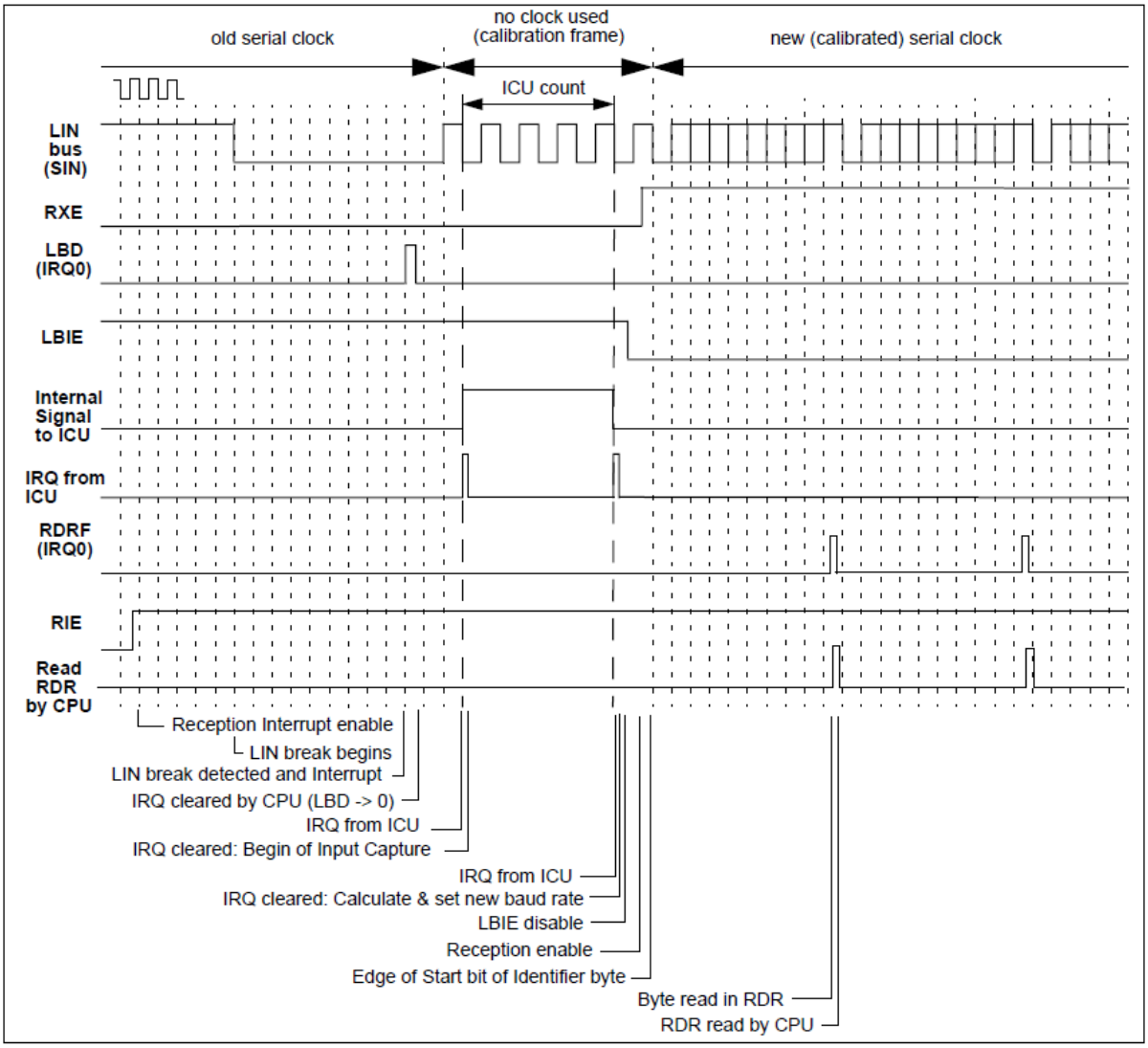


Figure 26-15 LIN bus timing and USART signals

25.6.4. Direct Access to Serial Pins

USART allows the user to access directly the transmission pin (SPIx_DO) or the reception pin (SPIx_DI).

USART Direct Pin Access

The USART provides the ability for the user to directly access the serial input or output pin. The software can always monitor the incoming serial data by reading the SIOP bit of the ESCR. If setting the Serial Output Pin direct access Enable (SOPE) bit of the ESCR the software can force the SPIx_DO pin to a desired value. Note, that this access is only possible, if the transmission shift register is empty (i. e. no transmission activity).

In LIN mode this function can be used for reading back the own transmission and is used for error handling if something is physically wrong with the single-wire LIN-bus.

(Note) Write the desired value to the SIOP pin **before** enabling the output pin access, to prevent undesired peaks. The peaks can occur because SIOP holds the last written value.

During a Read-Modify-Write operation the SIOP bit returns the actual value of the SPIx_DO pin in the read cycle instead the value of SPIx_DI during a normal read instruction.

25.6.5. Bidirectional Communication Function (Normal Mode)

In operation mode 0 or 2, normal serial bidirectional communication is available. Select operation mode 0 for asynchronous communication and operation mode 2 for synchronous communication.

Bidirectional Communication Function

The settings shown in figure 26-16 are required to operate in normal mode (operation mode 0 or 2).

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR04,SMR04	PEN	P	SBL	CL	AD	CRE	RXE	TXE	MD1	MD0	OTO	EXT	REST	UPCL	SCKE	SOE
Mode 0	◎	◎	◎	◎	x	0	◎	◎	0	0	x	0	0	0	◎	◎
Mode 2	□	□	□	x	x	0	◎	◎	1	0	◎	◎	0	0	◎	◎
SSR04, TDR04/RDR04	PE	ORE	FRE	RDRF	TDRE	BDS	RIE	TIE	Set transmission data (during writing) Retain reception data (during reading)							
Mode 0	◎	◎	◎	◎	◎	◎	◎	◎								
Mode 2	□	◎	□	◎	◎	◎	◎	◎								
ESCR04,ECCR04	LBIE	LBD	LBL1	LBL0	SOPE	SIOP	CCO	SECS	-	LBR	MS	SCDE	SSM	BIE	RBI	TBI
Mode 0	x	x	x	x	x	x	x	x		x	x	x	x	x	x	x
Mode 2	x	x	x	x	x	x	◎	◎		x	◎	◎	◎	x	x	x

◎ Bit is used
x Bit is not used
0 / 1 Set bit to 0 / 1
□ Bit is used if SSM = 1 (Synchronous start-/stop-bit mode)

Figure 26-16 Settings for USART operation mode 0 and 2

Inter-CPU Connection

As shown in figure 26-17, interconnect two CPUs in USART mode 2

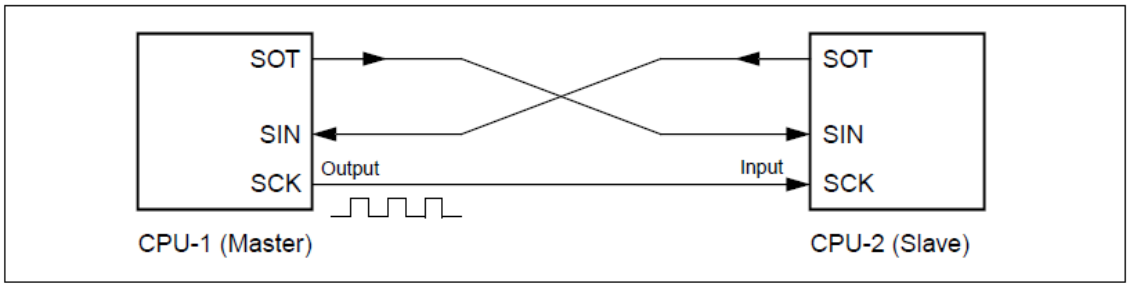


Figure 26-17 Connection example of USART mode 2 bidirectional communication

25.6.6. Master-Slave Communication Function (Multiprocessor Mode)

USART communication with multiple CPUs connected in master-slave mode is available for both master or slave systems.

Master-slave Communication Function

The settings shown in Figure 26-18 are required to operate USART in multiprocessor mode (operation mode 1).

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR04,SMR04	PEN	P	SBL	CL	AD	CRE	RXE	TXE	MD1	MD0	OTO	EXT	REST	UPCL	SCKE	SOE
Mode 1	x	x	◎	◎	◎	0	◎	◎	0	1	x	0	0	0	1	◎
SSR04, TDR04/RDR04	PE	ORE	FRE	RDRF	TDRE	BDS	RIE	TIE	Set transmission data (during writing) Retain reception data (during reading)							
Mode 1	x	◎	◎	◎	◎	◎	◎	◎								

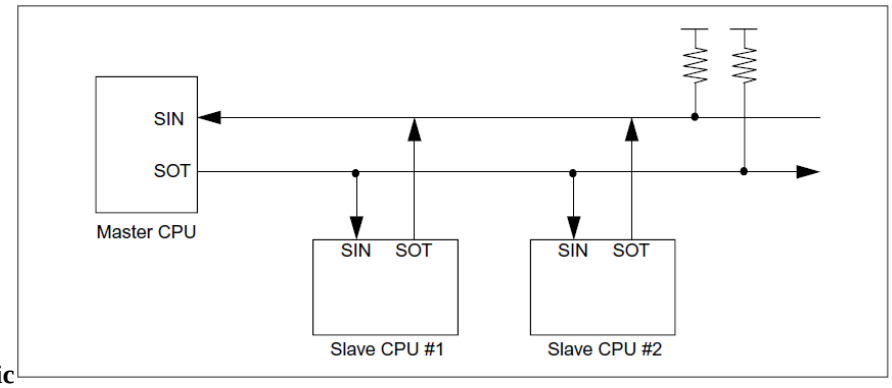
◎ Bit is used
x Bit is not used
0 / 1 Set bit to 0 / 1

Figure 26-18 Settings for USART operation mode 1

Inter-CPU Connection

As shown in figure 26-19, a communication system consists of one master CPU and multiple slave CPUs connected to two communication lines. USART can be used for the master or slave CPU.

Figure 26-19 Connection example of USART master-slave communication



Function Selection

Select the operation mode and data transfer mode for master-slave communication as shown in table 26-10.

Table 26-10 Selection of the master-slave communication function

	Operation mode		Data	Parity	Synchronization method	Stop bit	Bit direction
	Master CPU	Slave CPU					
Address transmission and reception	Mode 1 (send AD-bit)	Mode 1 (receive AD-bit)	AD="1" + 7- or 8-bit address	None	Asyn-chronous	1 or 2 bits	LSB or MSB first
Data transmission and reception			AD="0" + 7- or 8-bit data				

Communication Procedure

When the master CPU transmits address data, communication starts. The A/D bit in the address data is set to 1, and the communication destination slave CPU is selected. Each slave CPU checks the address data using a program. When the address data indicates the address assigned to a slave CPU, the slave CPU communicates with the master CPU (ordinary data). Figure 26-20 shows a flowchart of master-slave communication (multiprocessor mode)

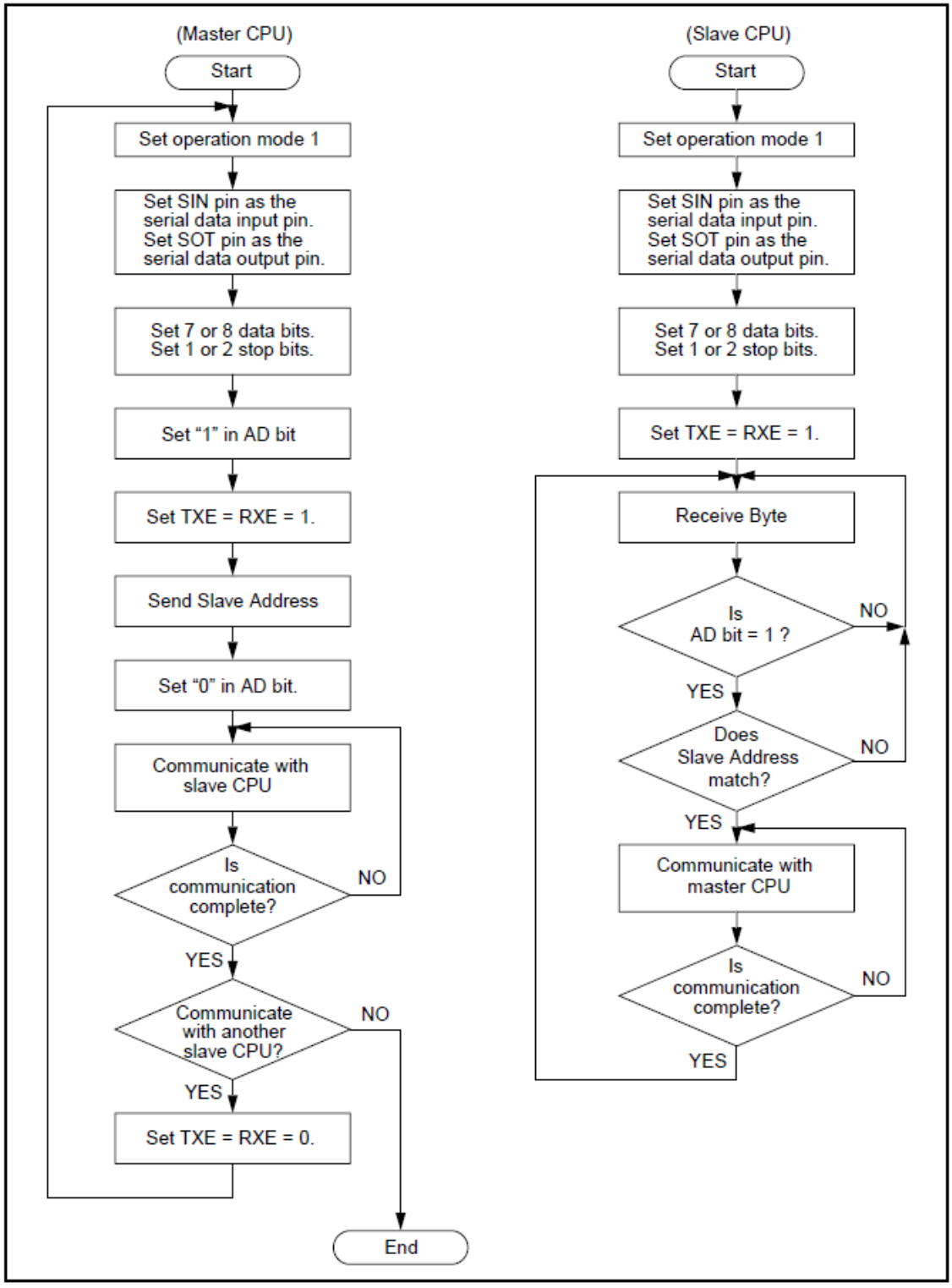


Figure 26-20 Master-slave communication flowchart

25.6.7. LIN Communication Function

USART communication with LIN devices is available for both LIN master or LIN slave systems.

LIN-Master-Slave Communication Function

The settings shown in the figure below are required to operate USART in LIN communication mode (operation mode 3).

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR04,SMR04	PEN	P	SBL	CL	AD	CRE	RXE	TXE	MD1	MD0	OTO	EXT	REST	UPCL	SCKE	SOE
Mode 3	x	x	+	+	x	0	◎	◎	1	1	x	0	0	0	1	◎
SSR04, TDR04/RDR04	PE	ORE	FRE	RDRF	TDRE	BDS	RIE	TIE	Set transmission data (during writing) Retain reception data (during reading)							
Mode 3	x	◎	◎	◎	◎	+	◎	◎								
ESCR04,ECCR 04	LBIE	LBD	LBL1	LBL0	SOPE	SIOP	CCO	SECS	-	LBR	MS	SCDE	SSM	BIE	RBI	TBI
Mode 3	◎	◎	◎	◎	x	x	x	x		◎	x	x	x	◎	x	x

◎ Bit is used
x Bit is not used
0 / 1 Set bit to 0 / 1
+ Bit is automatically set to the correct value

Figure 26-21 Settings for USART

LIN device connection

As shown in the Figure below, a communication system of one LIN-Master device and a LIN-Slave device. can operate both as LIN-Master or LIN-Slave.

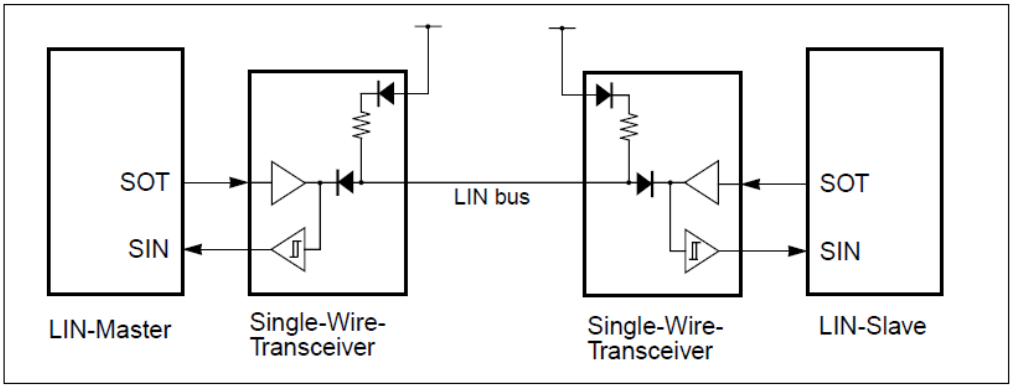


Figure 26-22 Connection example of a small LIN-Bus system

25.6.8. Sample Flowcharts for in LIN Communication (Operation Mode 3)

This section contains sample flowcharts for USART in LIN communication.

USART as master device

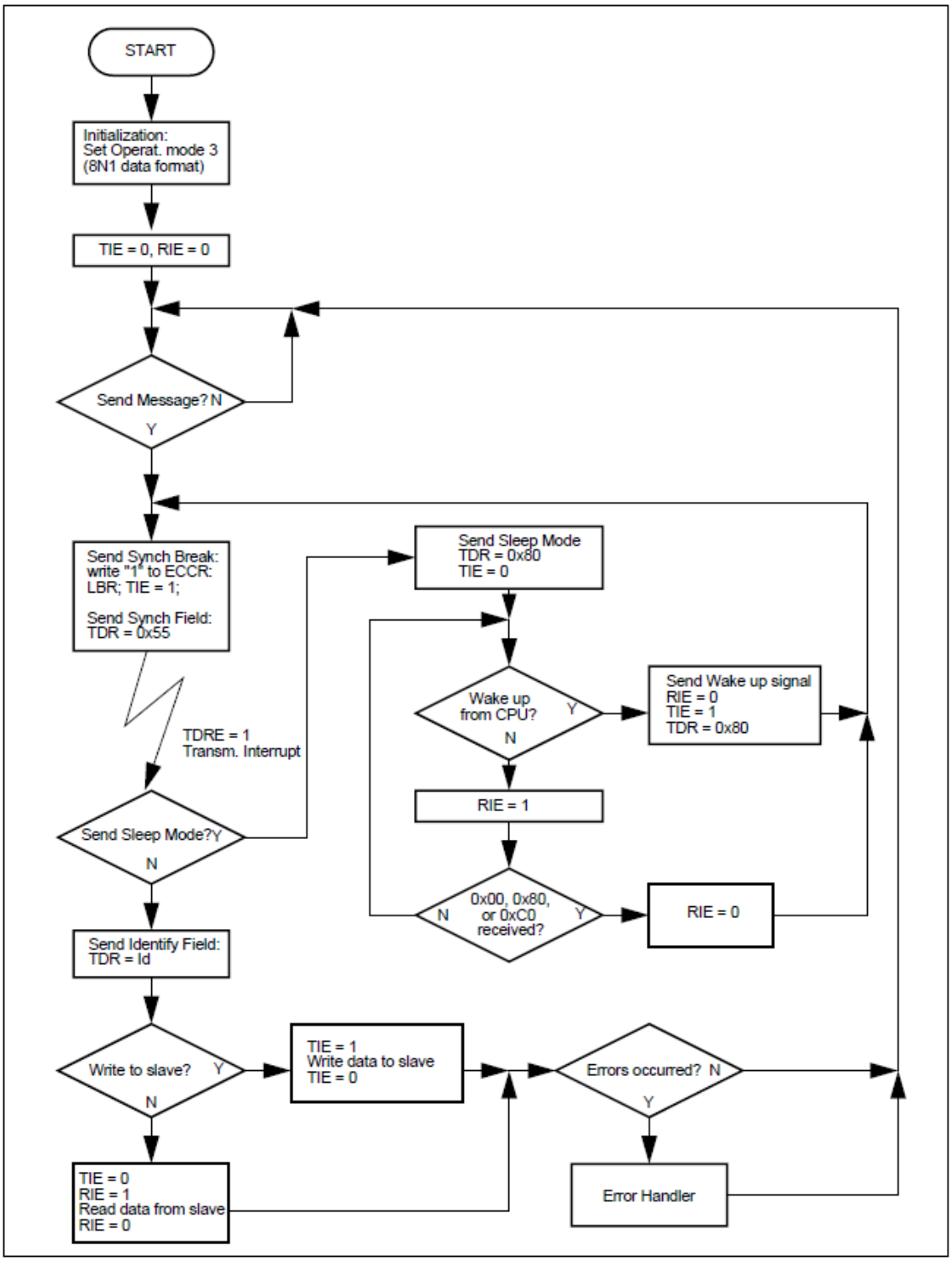


Figure 26-23 USART LIN master flow chart

USART as slave device

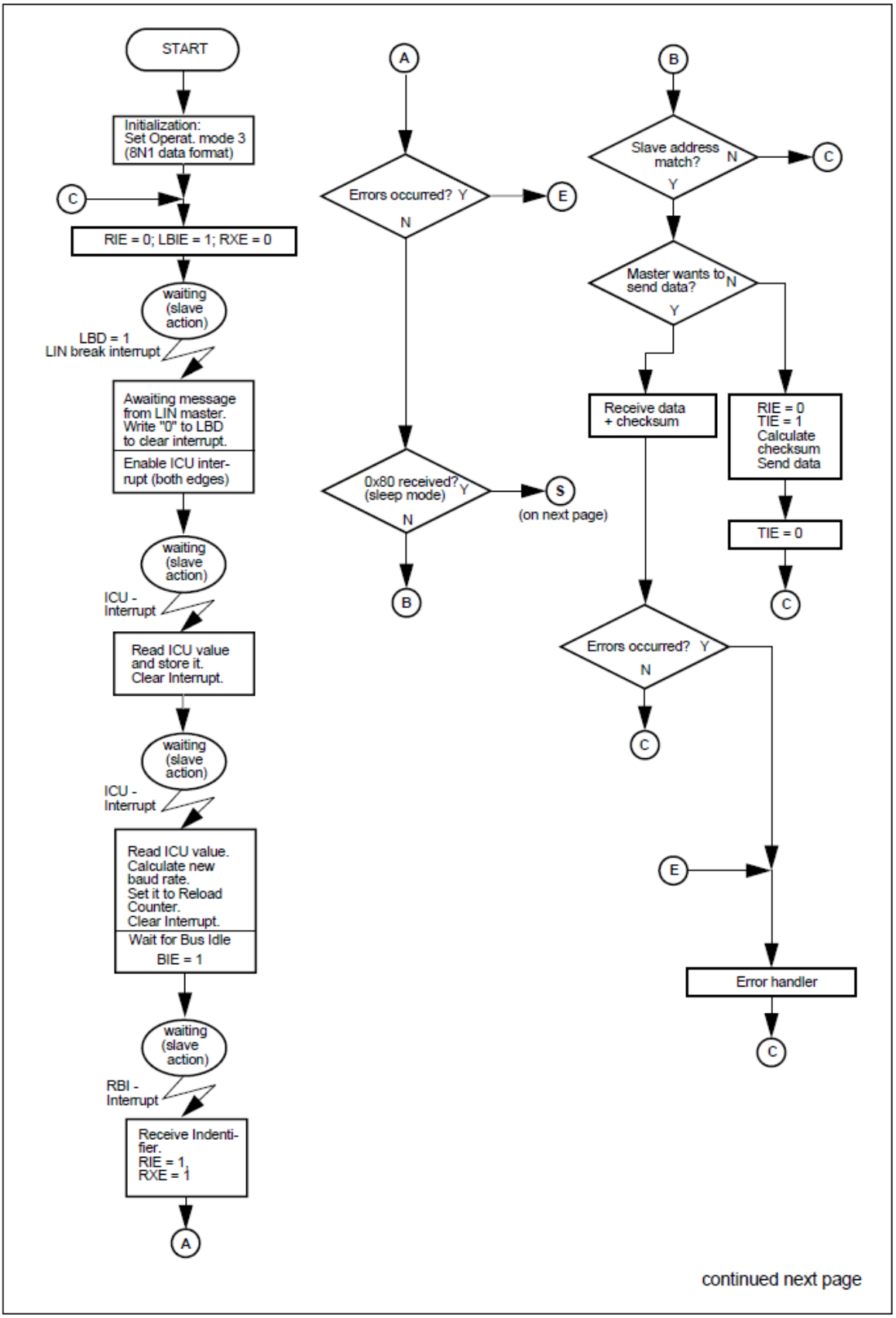


Figure 26-24 USART LIN slave flow chart (part1)

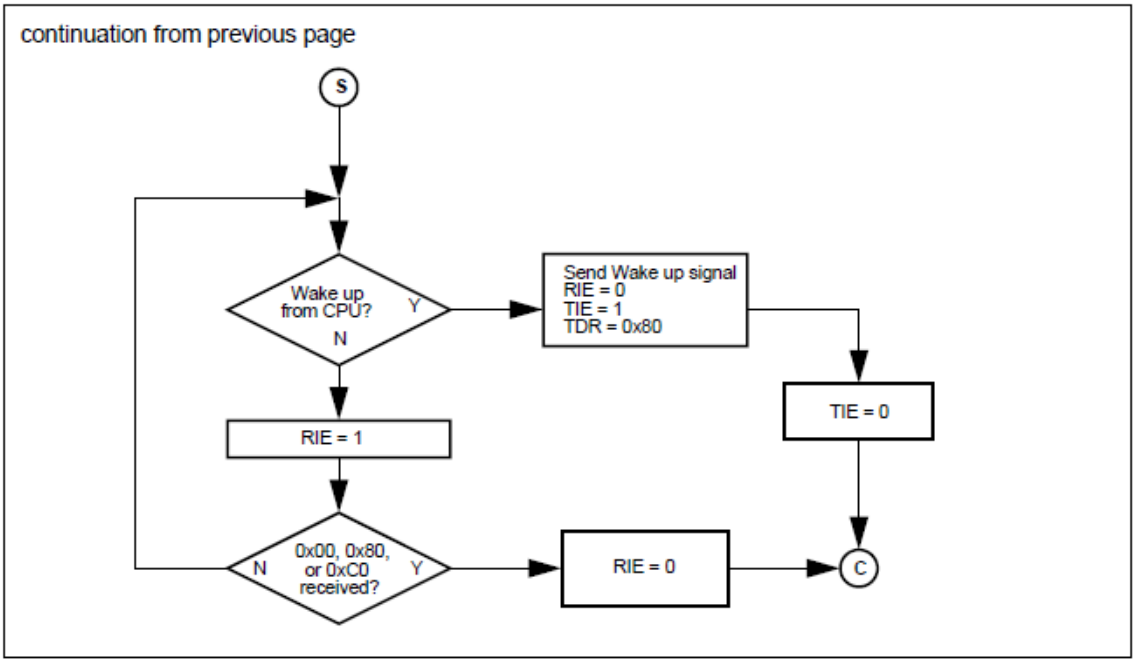


Figure 26-25 USART LIN slave flow chart (part 2)

25.7. Notes on using USART

Notes on using USART are given below.

Enabling Operations

In USART, the control register (SCR) has TXE (transmission) and RXE (reception) operation enable bits. Both, transmission and reception operations, must be enabled before the transfer starts because they have been disabled as the default value (initial value).

In single wire bus systems like ISO 9141 (LIN bus system) because of the mono directional communication it is highly recommended to enable only one of these two bits at the same time. Because of the automatic reception the sent data by USART would be received by USART too.

Cancelling Transfers

Transfers can be cancelled by clearing their operation enable bits TXE / RXE. If RXE is cleared during an ongoing reception, then the USART state machines must be reset (set UPCL=1). If a transmission is ongoing at the same time, it will be cancelled too! In this case, wait until TDRE=1 and TBI=1 before setting UPCL.

Software reset of UART

Perform the software reset (SMR: UPCL=1), when the TXE bit of the SCR register is "0".

Clearing reception errors

Please set SCR:CRE in synchronous slave mode only, if SCR:RXE = 0.

Communication Mode Setting

Set the communication mode while the system is not operating. If the mode is changed during transmission or reception, the transmission or reception is stopped and possible data will be get lost.

Transmission Interrupt Enabling Timing

The default (initial value) of the transmission data empty flag bit (SSR: TDRE) is “1” (no transmission data and transmission data write enable state). A transmission interrupt request is generated as soon as the transmission interrupt request is enabled (SSR: TIE=1). Be sure to set the TIE flag to “1” after setting the transmission data to avoid an immediate interrupt.

Using LIN operation mode 3

The LIN features are also available in mode 0 (transmitting, receiving break), but using mode 3 sets the USART data format automatically to LIN format (8N1, LSB first). So, break features are appliable for bus protocols other than LIN in mode 0. Note, that the transmission time of the break is variable, but the detection is specified to a minimum of 11 serial bit times.

Changing Operation Settings

It is recommended to disable the communication (RXE = 0, TXE = 0), if the UART setting or mode is changed or UART is initialized.

It is strongly recommended to reset USART after changing operation settings.

Particularly in synchronous mode 2 if (for example) start-/stop-bits are added to or removed from the data format.

<Caution>

If settings in the Serial Mode Register (SMR) are desired, it is not useful to set the UPCL bit at the same time to reset USART. The correct operation settings are *not* guaranteed in this case. Thus it is recommended to set the bits of the SMR and *then* to set them again plus the UPCL bit.

LIN slave settings

To initiate USART for LIN slave make sure to set the baudrate before receiving the first LIN synchronization break. This is needed to detect safely the minimum of 11 bit times of a LIN synch break.

Software compatibility

Although USART is similar to older Fujitsu-UARTs it is **not** software compatible to them. The

programming models may be the same, but the structure of the registers differ. Furthermore the setting of the baud rate is now determined by a reload value instead of selecting a preset value.

Bus Idle Function

The Bus Idle Function cannot be used in synchronous mode 2.

Baud Rate Detection Using the Input Capture Units

The USARTs provide the signal LSYN that can be connected to the ICU so that LSYN's pulse length can be measured to derive the baud rate. The connection of the LSYN signals to the ICUs is controlled by the Port 14 function register PFR and EPFR.

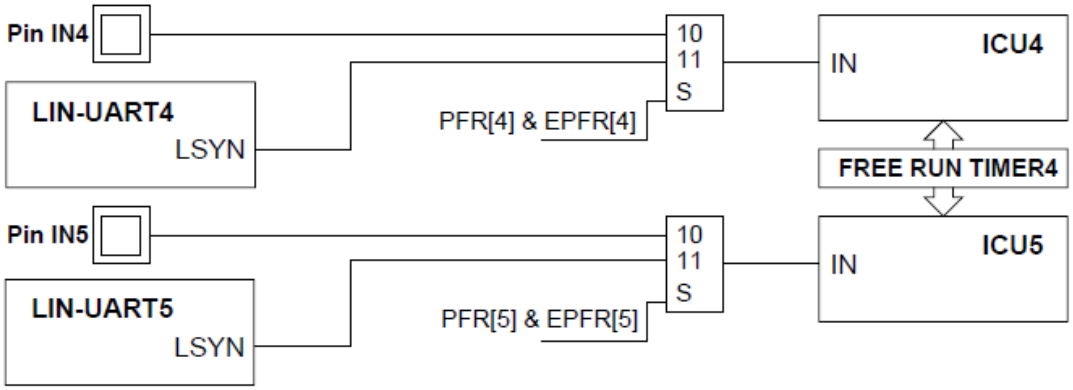


Figure 26-26 Baud Rate Detection Using the Input Capture Units

If the PFR bit equals '1' and the EPFR bits equals '0', the ICU is connected to its corresponding input pin IN.

If the PFR bit equals '1' and the EPFR bits equals '1', the USARTs are connected to the ICU.

The user has to take into account that:

- ICU4 and ICU5 share one free running timer (prescaler).

Effects of reception errors and CRE bit

CRE resets reception state machine and next falling edge at SPIx_DI starts reception of new byte. Therefore either set CRE bit immediately (within half bit time) after receiving errors to prevent data stream desynchronization or wait an application dependent time after receiving errors and set CRE, when SPIx_DI is idle.

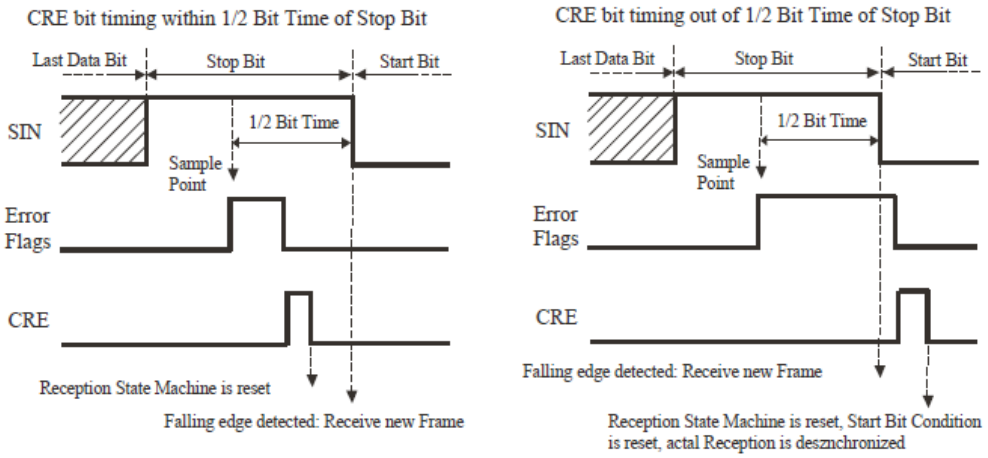


Figure 26-27 Timing of the CRE bit

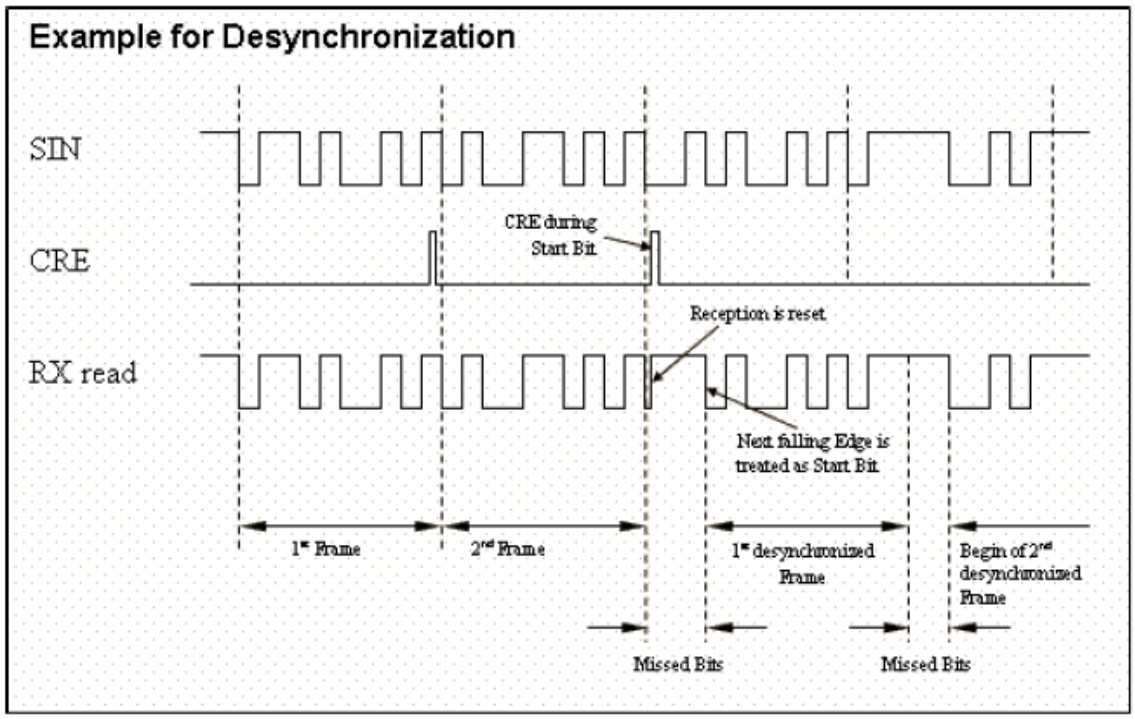


Figure 26-28 Data Stream Synchronization

Please note, that in case a framing error occurred (stop bit: SPIx_DI = "0") and next start bit (SPIx_DI = "0") follows immediately, this start bit is recognized regardless of no falling edge before. This is used to remain UART synchronized to the data stream and to determine bus always dominant errors (“Fig. 26-28” upper figure) by producing next framing errors, if a recessive stop bit is expected. If this behaviour is not wanted, please disable the reception temporarily (RXE = 1 -> 0 -> 1) after a framing error. In this case, reception goes on at next falling edge on SPIx_DI. (“Fig.26-29“ lower figure).

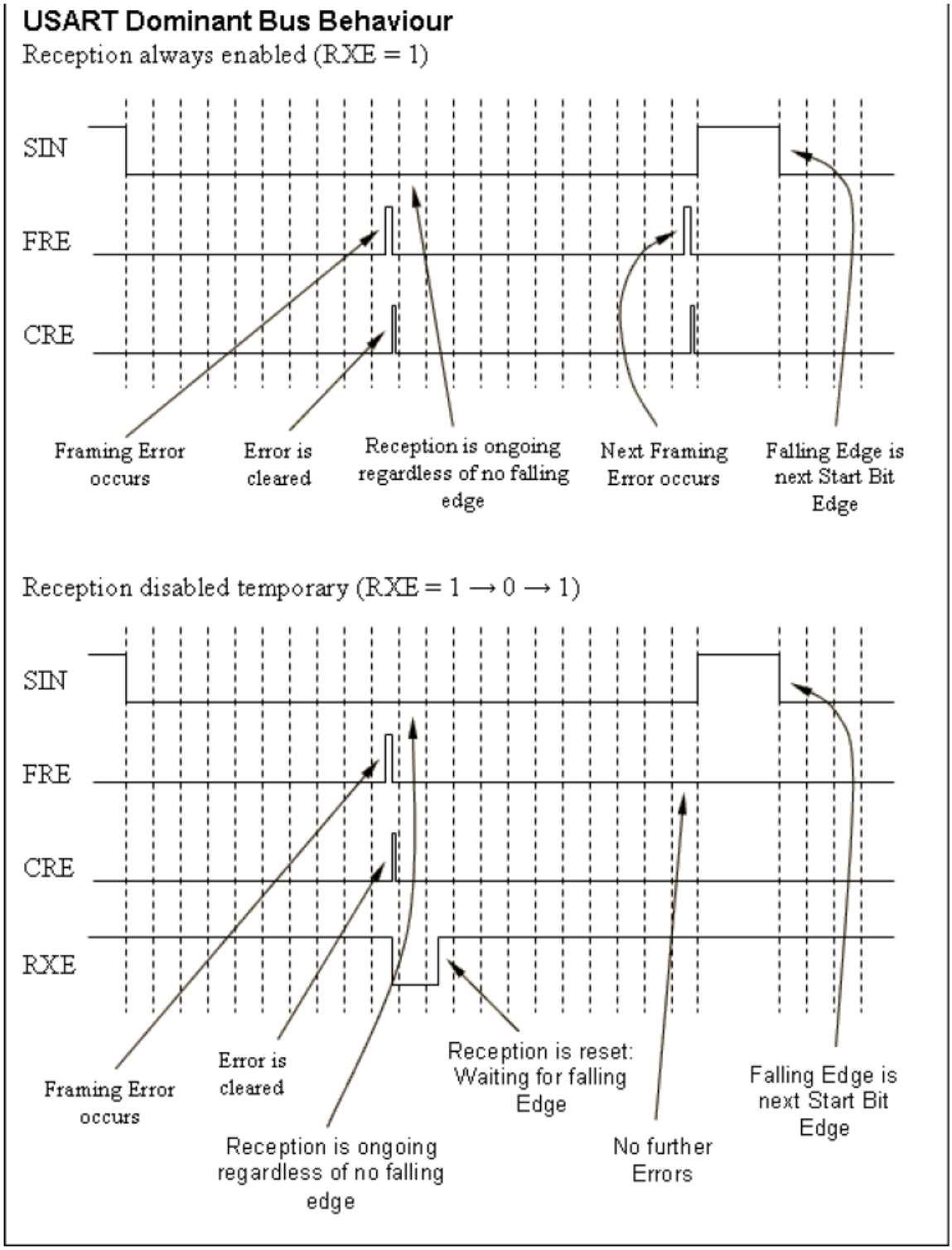


Figure 26-29 USART Dominant Bus Behaviour

26. GPIO Interface

This chapter describes the GPIO interface of the MB88F333.

26.1. Outline

The MB88F333 device has a GPIO port (maximum of 15 bits) of which is shared with other peripheral ports. Direction control and data read/write modes are selected via GPIO control registers.

26.2. Features

The GPIO module has following features:

- 15 bit GPIO ports
- Consists of the following registers :
  (1) Two port Data registers (PDR)
  (2) One Data Direction register (DDR)

26.3. Block Diagram

Figure 26-1 shows a block diagram of the GPIO module basic block. The GPIO module has 15 such basic blocks.

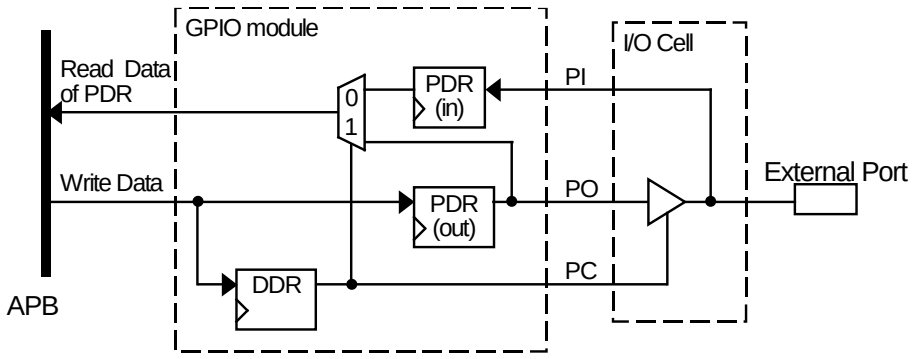


Figure 26-1 GPIO interface basic block block diagram

26.4. Registers

26.4.1. Format of Register Descriptions

The descriptions in the configuration tables are as follows.

1. Register address : Address of each register
2. Bit number : Bit number of each register
3. Bit field name : Name of bit field included in each register
4. R/W : R/W attribute of each bit field
   Symbols with the following meanings:
   R: Read Only
   W: Write Only
   R/W: Readable and Writeable
5. Initial value : Initial value indicates the value of each bit field immediately after a reset.
   "—" means that the initial value is not defined.

26.4.2. Global Address

Please refer to chapter 3 'Memory Map' for the module base address.

26.4.3. Register Summary

Table 26-1 lists the GPIO module registers.

Table 26-1 GPIO Interface Register List

Address	Register name	Description
Base address + 0x00	PDR0	Port Data Register 0
Base address + 0x04	PDR1	Port Data Register 1
Base address + 0x08	PDR2	Port Data Register 2
Base address + 0x10	DDR0	Data Direction Register 0
Base address + 0x14	DDR1	Data Direction Register 1
Base address + 0x18	DDR2	Data Direction Register 2

26.4.4. Register Description

26.4.4.1. PDR0 (Port Data Register 0): GPIO[6:0]

Register address	Base Address + 0x00															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field name	Reserved															
R/W	—															
Initial value	—															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved									PDR0						
R/W	—									R/W						
Initial value	—									-						

Bit 31 : 7	Reserved
	Reserved bits. Read data is not defined. Write access is ignored.
Bit 6 : 0	PDR0 (Port Data Register)
	PDR0 registers are the data register of GPIO ports. The input data and the output data are transferred through this register. PDR0 registers correspond to the bit6 ~ bit0 of GPIO ports respectively. The direction of the ports is defined by the corresponding bits of the DDR0 register. The initial value of these bits is not defined.

26.4.4.2. PDR1 (Port Data Register 1): GPIO[15:12]

Register address	Base Address + 0x04															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field name	Reserved															
R/W	—															
Initial value	—															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								PDR1				Reserved			
R/W	—								R/W				-			
Initial value	—								-				-			

Bit 31 : 8	Reserved
	Reserved bits. Read data is not defined. Write access is ignored.
Bit 7 : 0	PDR1 (Port Data Register)
	PDR1 registers are 8bit data register of GPIO ports. The input data and the output data are transferred through this register. PDR1 registers correspond to the bit15 ~ bit12 of GPIO ports respectively. The direction of the ports is defined by the corresponding bits of the DDR1 register. The initial value of these bits is not defined.

26.4.4.3. PDR2 (Port Data Register 2): GPIO[19:16]

Register address	Base Address + 0x08															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field name	Reserved															
R/W	—															
Initial value	—															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved												PDR2			
R/W	—												R/W			
Initial value	—												—			

Bit 31 : 8	Reserved
	Reserved bits. Read data is not defined. Write access is ignored.
Bit 7 : 0	PDR2 (Port Data Register)
	PDR2 registers are 4bit data register of GPIO ports. The input data and the output data are transferred through this register. PDR2 registers correspond to the bit19 ~ bit16 of GPIO ports respectively. The direction of the ports is defined by the corresponding bits of the DDR2 register. The initial value of these bits is not defined.

26.4.4.4. DDR0 (Data Direction Register 0)

Register address	Base Address + 0x10															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field name	Reserved															
R/W	—															
Initial value	—															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								DDR0							
R/W	—								R/W							
Initial value	—								-							

Bit 31 : 8	Reserved
	Reserved bits. Read data is not defined. Write access is ignored.
Bit 7 : 0	DDR0 (Data Direction Register)
	DDR0 registers are 8bits direction control register of GPIO ports. DDR0 = 0 : the port is an input port. DDR0 = 1 : the port is an output port. DDR0 registers correspond to bit7~ bit0 of the GPIO ports respectively.

26.4.4.5. DDR1 (Data Direction Register 1)

Register address	Base Address + 0x14															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field name	Reserved															
R/W	—															
Initial value	—															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								DDR1				Reserved			
R/W	—								R/W				-			
Initial value	—								0				-			

Bit 31 : 8	Reserved
	Reserved bits. Read data is not defined. Write access is ignored.
Bit 7 : 0	DDR1 (Data Direction Register)
	DDR1 registers are 8bits direction control register of GPIO ports. DDR1 = 0 : the port is an input port. DDR1 = 1 : the port is an output port. DDR1 registers correspond to the bit815~ bit12 of the GPIO ports respectively.

26.4.4.6. DDR2 (Data Direction Register 2)

Register address	Base Address + 0x18															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field name	Reserved															
R/W	—															
Initial value	—															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved												DDR2			
R/W	—												R/W			
Initial value	—												0			

Bit 31 : 8	Reserved
	Reserved bits. Read data is not defined. Write access is ignored.
Bit 7 : 0	DDR2 (Data Direction Register)
	DDR2 registers are 4bits direction control register of GPIO ports. DDR2 = 0 : the port is an input port. DDR2 = 1 : the port is an output port. DDR2 registers correspond to bit19 ~ bit16 of the GPIO ports respectively.

26.5. Description of Operation

26.5.1. Direction Control

The direction of a GPIO port is determined using the DDR register. The direction of each GPIO (15 bits) can be changed using the DDR register. The initial port direction after a reset (initial value of DDR register) is “0” (i.e. all act as input ports).

If the port direction is changed, the designer must pay careful attention to avoid a bus conflict.

26.5.2. Data Transmission

If the GPIO port is used as an input port (DDR=0), the input data arriving via the Port Input line (PI) is

stored in the PDR (in) with every rising edge of the PCLK signal, as shown in Figure 26-1. A user can see the input data by reading the PDR register. Write access to the PDR register is effective in this case and PDR (out) is changed by write access. Data in PDR (out) is not readable when the GPIO port is used as an input (DDR=0).
When the GPIO port is used as output port (DDR=1), the value of PDR is output to the Port Output line (PO). In this case the read data of PDR is the same as the Port Output line (PO).

Table 26-2 summarizes the operation of GPIO ports in each mode.

Table 26-2 Summary of GPIO operation

Mode	Direction	DDR	PDR	
			Read data	Write data
GPIO	Input	0	Same value as PI	Write data is stored in PDR, but readable.
	Output	1	Same value as PO	Write data is stored in PDR and transferred to the PO.

27. External Interrupt

This chapter describes the External Interrupt unit of the MB88F333.

27.1. Outline

The external interrupt controller handles interrupt requests from external ports or interrupt sources which do not keep asserting requests until their status registers are cleared.

27.2. Features

The External Interrupt has the following functions.

- Bus slave compliant to AMBA (APB).
- Five interrupt request channels.
- Four selectable request levels: high level, low level, rising edge, and falling edge.
- Provide a request to the main interrupt controller when returning from the STOP mode.

27.3. Block diagram

Figure 28-1 shows the block diagram of the external interrupt controller

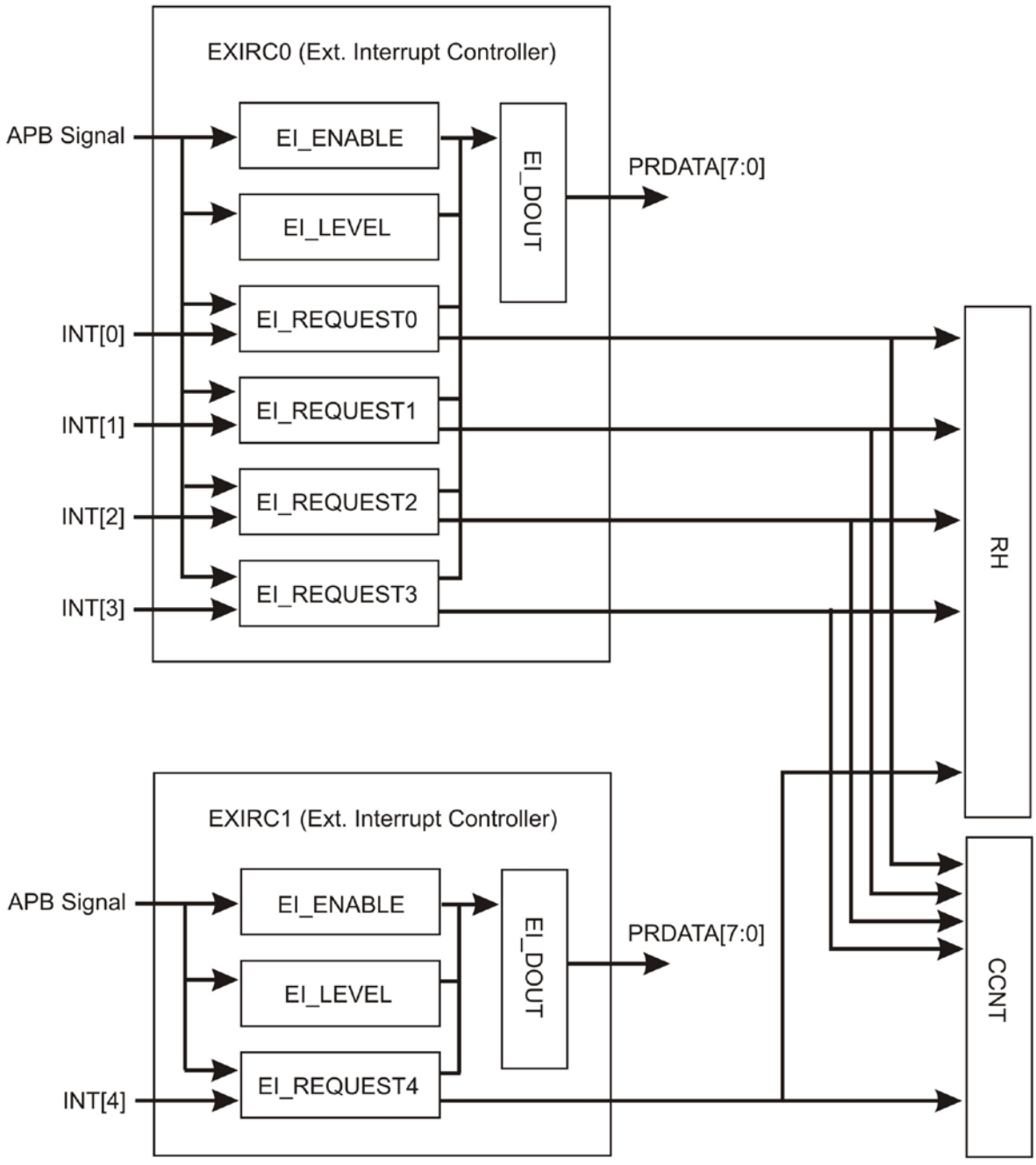


Figure 28-1 Block Diagram of External Interrupt Controller

Table 28-1 lists functions of each block contained in this module.

Table 28-1 Function of Each Block

Block name	Function
EI_ENABLE	Enable to provide an external interrupt request to the main interrupt controller.
EI_LEVEL	Set the request level: high level, low level, rising edge, or falling edge.
EI_REQUEST	Synchronize and hold interrupt requests.
EI_DOUT	Output register value as PRDATA.

27.4. Registers

27.4.1. Format of Register Descriptions

- Endian
  Only Little Endian access corresponds to the register of this module.

- Address
  “Address” shows the address of the register. (Base address + Offset address)

- Bit
  “Bit” shows the bit number of the register.

- Name
  “Name” shows the bit field name of the register.

- R/W
  “R/W” shows the attribute of Read/Write in each Bit field.
  R0: The Read value is always "0".
  R1: The Read value is always “1”.
  W0: The Write value is always "0", When “1” is written, it is disregarded.
  W1: The Write value is always “1”, When “0” is written, it is disregarded.
  R: Read
  W: Write

Note : If a value is written to registers/bitfields that list R0, R1 and R in the following descriptions, then this value will not be changed in those registers/bitfields.

- Initial value
  “Initial value” is an initial value when reset is released.
  0: Initial value is "0".
  1: Initial value is "1".
  X: Undetermined

27.4.2. Global Address

For the module base address please refer to chapter 3, Memory map.

27.4.3. Register summary

Table 28-2 lists the registers contained in the external interrupt controller module.

Table 28-2 List of Registers Contained in External Interrupt Controller Module

Address	Register name
00h	EIENB(External Interrupt Enable)
04h	EIREQ(External Interrupt Request)
08h	EILVL(External Interrupt Level)

27.4.4. Register Description

27.4.4.1. EIENB (External Interrupt Enable)

Address	**Base Address** + 0x00															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved											Resvd	ENB3	ENB2	ENB1	ENB0
R/W	R0	R/W	R/W	R/W	R/W	R0	R0	R0	R0	R0	R0	R0	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit3:0	ENB3 to 0 (Interrupt Enable of Channel 3 to 0)	
	If ENB bit is 0, any external interrupt request is ignored.	
	0	Interrupt request channel is not enabled ←Initial value
	1	Interrupt request channel is enabled
	The initial value of this bit is 0.	

27.4.4.2. EIREQ (External Interrupt Request)

Address	**Base Address** + 0x04															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved											Resvd	REQ3	REQ2	REQ1	REQ0
R/W	R0	R/W	R/W	R/W	R/W	R0	R0	R0	R0	R0	R0	R0	R/W0	R/W0	R/W0	R/W0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit3:0	REQ3 to 0 (Interrupt Request of Channel 3 to 0)	
	If an external interrupt request is accepted, REQ bit is set in 1.	
	0	No interrupt request ←Initial value
	1	An interrupt request is accepted.
	The initial value of this bit is 0.	

27.4.4.3. EIREQ (External Interrupt Request)

Address	**Base Address** + 0x08															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Reserved						Res	Res	LVL3[1]	LVL3[0]	LVL2[1]	LVL2[0]	LVL1[1]	LVL1[0]	LVL0[1]	LVL0[0]
R/W	R0	R/W	R/W	R/W	R/W	R0	R0	R0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1

Bit9:0	LVL9[1:0] to LVL0[1:0] (Request Level of Channel 3 to 0)
	There are four selectable request levels of EXTIRQIN for each channel. 00 Low level 01 High level ←Initial value 10 Rising edge 11 Falling edge The initial value of these bits is 01.

27.5. Description of Operation

This section describes the operation of the external interrupt controller module.

27.5.1. External Interrupt

When a request is input to the corresponding channel after setting the EIENB and EILVL registers, the external interrupt controller issues an interrupt request signal to the main interrupt controller.

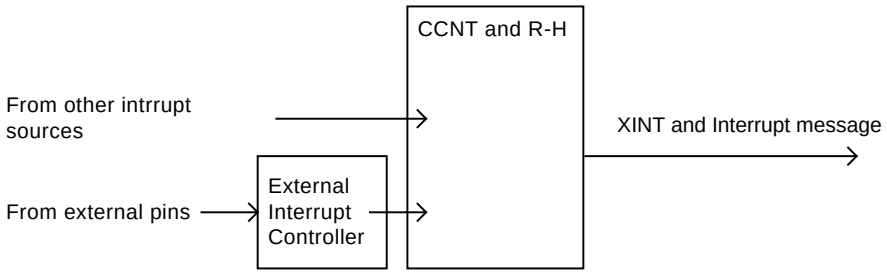


Figure 28-2 Integration of Interrupt Conteroller and External Interrupt Controller

27.5.2. Operating Procedure for External Interrupt

To set the register within the external interrupt, proceed as follows:

(1) Disable the corresponding bit of the EIENB register.
(2) Set the corresponding bit of the EILVL register.
(3) Clear the corresponding bit of the EIREQ register.
(4) Enable the corresponding bit of EIENB register.

The EIENB register must be set disabled when setting the register within this module. It is also necessary to clear the EIREQ register before setting the EIENB register enabled. This is done in order to prevent the accidental occurrence of an interrupt source at register setting or with interrupts enabled.

28. Programmable Pulse Generators (PPGs)

This chapter describes the Programmable Pulse Generators of the MB88F333.

28.1. Overview

Programmable Pulse Generators (PPGs) are used to produce a one-shot (rectangular wave) output or pulse width modulation (PWM) output. With their software-programmable cycle and duty capability, the PPGs provide a broad scope of applications.

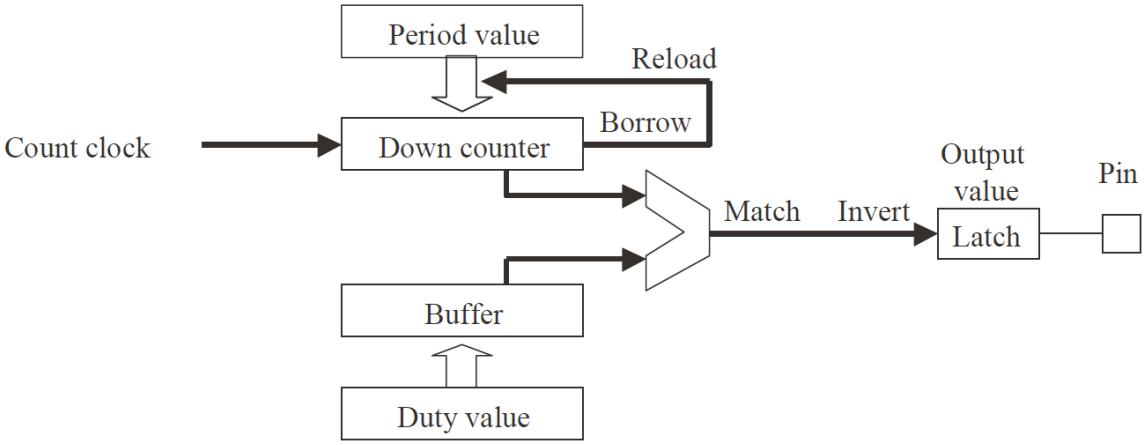


Figure 29-1 Diagram of PPG (Programmable Pulse Generator)

28.2. Features

- Output waveforms: The PPGs can generate the following six kinds of waveforms:

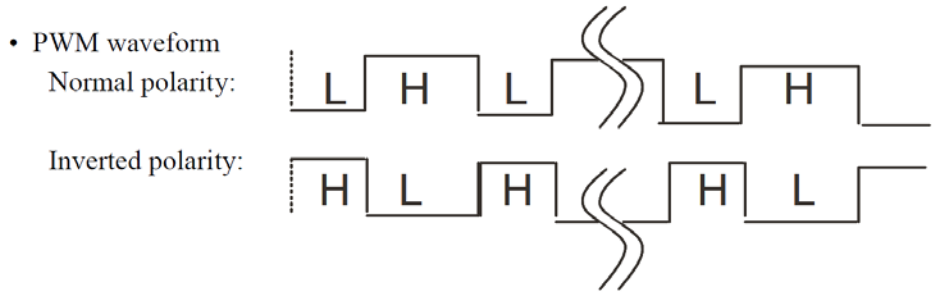


Figure 29-2 Output waveforms

- One-shot waveforms (Rectangular wave)

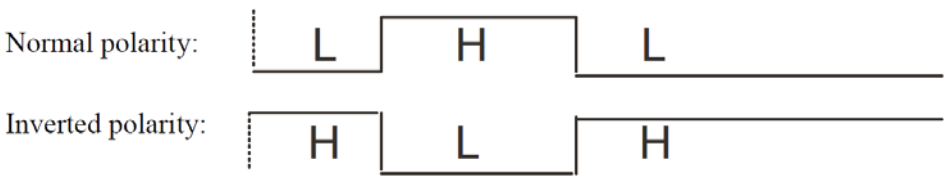


Figure 29-3 One-shot waveform (Rectangular wave)

- Clamped output
  Normal polarity: “L” Clamped output
  Inverted polarity: “H” Clamped output
- Quantity: 2 groups (Output: 8 channels PPG4 - PPG11)
- Count clock: Choose from four choices.
  ×1, ×1/4, ×1/16, ×1/64 of the peripheral clock (CLKP)
- Period: Setting range = Duty value ~ 65535 (specified with a 16-bit register)
  Period = Count clock × (PCSR register value + 1)
  (Example) Count clock = 41.66MHz(24ns), PCSR value = 41666
  Period = 24ns × (41666+1) = 1ms
- Duty: Setting range = 0 ~ Period value (specified with a 16-bit register)
  Duty = Count clock × (PDUT register value + 1)
- Interrupt: Choose from four choices:
  - Software trigger
  - Counter borrow (cycle match)
  - Duty match
  - Counter borrow (cycle match) or duty match
- Activation trigger:
  - Software trigger
  - Internal triggers
    Reload timer output ×2 (TOT2) available as trigger for PPG4-PPG7
    Reload timer output ×3 (TOT3) available as trigger for PPG4-PPG7
    Reload timer output ×4 (TOT4) available as trigger for PPG8-PPG11
    Reload timer output ×5 (TOT5) available as trigger for PPG8-PPG11

28.3. Registers

28.3.1. Format of Register Descriptions

- Endian

  Only Little Endian access corresponds to the register of this module.

- Address

  “Address” shows the address of the register. (Base address + Offset address)

- Bit

  “Bit” shows the bit number of the register.

- Name

  “Name” shows the bit field name of the register.

- R/W

  “R/W” shows the attribute of Read/Write in each Bit field.
  R0: The reading value is always "0".
  R1: The reading value is always "1".
  W0: The writing value is always "0".When “1” is written, it is disregarded.
  W1: The writing value is always "1". When “0” is written, it is disregarded.
  R: Read
  W: Write
  RX: not readable
  WX: not writeable

  Note : If a value is written to registers/bitfields that list R0, R1 and R in the following descriptions, then this value will not be changed in those registers/bitfields.

- Initial value

  “Initial value” is an initial value when reset is released.
  0: It becomes "0".
  1: It becomes "1".
  X: It becomes irregular.

28.3.2. PCSR: PPG Cycle Setting Register

Controls the cycle of the PPG.

- PCSR04 (PPG4): Address 0132h (Access: Half-word)
- PCSR05 (PPG5): Address 013Ah (Access: Half-word)
- PCSR06 (PPG6): Address 0142h (Access: Half-word)
- PCSR07 (PPG7): Address 014Ah (Access: Half-word)
- PCSR08 (PPG8): Address 0152h (Access: Half-word)
- PCSR09 (PPG9): Address 015Ah (Access: Half-word)
- PCSR10 (PPG10): Address 0162h (Access: Half-word)
- PCSR11 (PPG11): Address 016Ah (Access: Half-word)

PCSR04_H (PPG4): Address 0132h
PCSR04_L (PPG4): Address 0133h

15	14	13	12	11	10	9	8	Bit
PCSR04[15]	PCSR04[14]	PCSR04[13]	PCSR04[12]	PCSR04[11]	PCSR04[10]	PCSR04[9]	PCSR04[8]	
X	X	X	X	X	X	X	X	Initial value
RW	RW	RW	RW	RW	RW	RW	RW	Attribute

7	6	5	4	3	2	1	0	Bit
PCSR04[7]	PCSR04[6]	PCSR04[5]	PCSR04[4]	PCSR04[3]	PCSR04[2]	PCSR04[1]	PCSR04[0]	
X	X	X	X	X	X	X	X	Initial value
RW	RW	RW	RW	RW	RW	RW	RW	Attribute

PCSR05_H (PPG5): Address 013Ah
PCSR05_L (PPG5): Address 013Bh

15	14	13	12	11	10	9	8	Bit
PCSR05[15]	PCSR05[14]	PCSR05[13]	PCSR05[12]	PCSR05[11]	PCSR05[10]	PCSR05[9]	PCSR05[8]	
X	X	X	X	X	X	X	X	Initial value
RW	RW	RW	RW	RW	RW	RW	RW	Attribute

7	6	5	4	3	2	1	0	Bit
PCSR05[7]	PCSR05[6]	PCSR05[5]	PCSR05[4]	PCSR05[3]	PCSR05[2]	PCSR05[1]	PCSR05[0]	
X	X	X	X	X	X	X	X	Initial value
RW	RW	RW	RW	RW	RW	RW	RW	Attribute

PCSR06_H (PPG6): Address 0142h
PCSR06_L (PPG6): Address 0143h

15	14	13	12	11	10	9	8	Bit
PCSR06[15]	PCSR06[14]	PCSR06[13]	PCSR06[12]	PCSR06[11]	PCSR06[10]	PCSR06[9]	PCSR06[8]	
X	X	X	X	X	X	X	X	Initial value
RW	RW	RW	RW	RW	RW	RW	RW	Attribute

7	6	5	4	3	2	1	0	Bit
PCSR06[7]	PCSR06[6]	PCSR06[5]	PCSR06[4]	PCSR06[3]	PCSR06[2]	PCSR06[1]	PCSR06[0]	
X	X	X	X	X	X	X	X	Initial value
RW	RW	RW	RW	RW	RW	RW	RW	Attribute

PCSR07_H (PPG7): Address 014Ah
PCSR07_L (PPG7): Address 014Bh

15	14	13	12	11	10	9	8	Bit
PCSR07[15]	PCSR07[14]	PCSR07[13]	PCSR07[12]	PCSR07[11]	PCSR07[10]	PCSR07[9]	PCSR07[8]	
X	X	X	X	X	X	X	X	Initial value
RW	RW	RW	RW	RW	RW	RW	RW	Attribute

7	6	5	4	3	2	1	0	Bit
PCSR07[7]	PCSR07[6]	PCSR07[5]	PCSR07[4]	PCSR07[3]	PCSR07[2]	PCSR07[1]	PCSR07[0]	
X	X	X	X	X	X	X	X	Initial value
RW	RW	RW	RW	RW	RW	RW	RW	Attribute

PCSR08_H (PPG8): Address 0152h
PCSR08_L (PPG8): Address 0153h

15	14	13	12	11	10	9	8	Bit
PCSR08[15]	PCSR08[14]	PCSR08[13]	PCSR08[12]	PCSR08[11]	PCSR08[10]	PCSR08[9]	PCSR08[8]	
X	X	X	X	X	X	X	X	Initial value
RW	RW	RW	RW	RW	RW	RW	RW	Attribute

7	6	5	4	3	2	1	0	Bit
PCSR08[7]	PCSR08[6]	PCSR08[5]	PCSR08[4]	PCSR08[3]	PCSR08[2]	PCSR08[1]	PCSR08[0]	
X	X	X	X	X	X	X	X	Initial value
RW	RW	RW	RW	RW	RW	RW	RW	Attribute

PCSR09_H (PPG9): Address 015Ah
PCSR09_L (PPG9): Address 015Bh

15	14	13	12	11	10	9	8	Bit
PCSR09[15]	PCSR09[14]	PCSR09[13]	PCSR09[12]	PCSR09[11]	PCSR09[10]	PCSR09[9]	PCSR09[8]	
X	X	X	X	X	X	X	X	Initial value
RW	RW	RW	RW	RW	RW	RW	RW	Attribute

7	6	5	4	3	2	1	0	Bit
PCSR09[7]	PCSR09[6]	PCSR09[5]	PCSR09[4]	PCSR09[3]	PCSR09[2]	PCSR09[1]	PCSR09[0]	
X	X	X	X	X	X	X	X	Initial value
RW	RW	RW	RW	RW	RW	RW	RW	Attribute

PCSR10_H (PPG10): Address 0162h
PCSR10_L (PPG10): Address 0163h

15	14	13	12	11	10	9	8	Bit
PCSR10[15]	PCSR10[14]	PCSR10[13]	PCSR10[12]	PCSR10[11]	PCSR10[10]	PCSR10[9]	PCSR10[8]	
X	X	X	X	X	X	X	X	Initial value
RW	RW	RW	RW	RW	RW	RW	RW	Attribute

7	6	5	4	3	2	1	0	Bit
PCSR10[7]	PCSR10[6]	PCSR10[5]	PCSR10[4]	PCSR10[3]	PCSR10[2]	PCSR10[1]	PCSR10[0]	
X	X	X	X	X	X	X	X	Initial value
RW	RW	RW	RW	RW	RW	RW	RW	Attribute

PCSR11_H (PPG11): Address 016Ah
PCSR11_L (PPG11): Address 016Bh

15	14	13	12	11	10	9	8	Bit
PCSR11[15]	PCSR11[14]	PCSR11[13]	PCSR11[12]	PCSR11[11]	PCSR11[10]	PCSR11[9]	PCSR11[8]	
X	X	X	X	X	X	X	X	Initial value
RW	RW	RW	RW	RW	RW	RW	RW	Attribute

7	6	5	4	3	2	1	0	Bit
PCSR11[7]D7	PCSR11[6]	PCSR11[5]	PCSR11[4]	PCSR11[3]	PCSR11[2]	PCSR11[1]	PCSR11[0]	
X	X	X	X	X	X	X	X	Initial value
RW	RW	RW	RW	RW	RW	RW	RW	Attribute

- The PPG Period Setting registers are buffered. Transfers from the buffers to the counter take place automatically at counter overflow or underflow.
- After the PPG Period Setting registers have been written, be sure to set PPG Duty Setting registers PDUT.
- Always access the PPG Period Setting registers in a half-word (16-bit) format.
  (See “Caution”)
- The read value is all "1".

28.3.3. PDUT: PPG Duty Setting Register

Sets the duty of the PPG output waveform.

- PDUT04 (PPG4): Address 0134h (Access: Half-word)
- PDUT05 (PPG5): Address 013Ch (Access: Half-word)
- PDUT06 (PPG6): Address 0144h (Access: Half-word)
- PDUT07 (PPG7): Address 014Ch (Access: Half-word)
- PDUT08 (PPG8): Address 0154h (Access: Half-word)
- PDUT09 (PPG9): Address 015Ch (Access: Half-word)
- PDUT10 (PPG10): Address 0164h (Access: Half-word)
- PDUT11 (PPG11): Address 016Ch (Access: Half-word)

PDUT04_H (PPG4): Address 0134h (Access: Half-word)

PDUT04_L (PPG4): Address 0135h (Access: Half-word)

15	14	13	12	11	10	9	8	Bit
PDUT04[15]	PDUT04[14]	PDUT04[13]	PDUT04[12]	PDUT04[11]	PDUT04[10]	PDUT04[9]	PDUT04[8]	
X	X	X	X	X	X	X	X	Initial value
RW	RW	RW	RW	RW	RW	RW	RW	Attribute

7	6	5	4	3	2	1	0	Bit
PDUT04[7]	PDUT04[6]	PDUT04[5]	PDUT04[4]	PDUT04[3]	PDUT04[2]	PDUT04[1]	PDUT04[0]	
X	X	X	X	X	X	X	X	Initial value
RW	RW	RW	RW	RW	RW	RW	RW	Attribute

PDUT05_H (PPG5): Address 013Ch (Access: Half-word)

PDUT05_L (PPG5): Address 013Dh (Access: Half-word)

15	14	13	12	11	10	9	8	Bit
PDUT05[15]	PDUT05[14]	PDUT05[13]	PDUT05[12]	PDUT05[11]	PDUT05[10]	PDUT05[9]	PDUT01[8]	
X	X	X	X	X	X	X	X	Initial value
RW	RW	RW	RW	RW	RW	RW	RW	Attribute

7	6	5	4	3	2	1	0	Bit
PDUT05[7]	PDUT05[6]	PDUT05[5]	PDUT05[4]	PDUT05[3]	PDUT05[2]	PDUT05[1]	PDUT05[0]	
X	X	X	X	X	X	X	X	Initial value
RW	RW	RW	RW	RW	RW	RW	RW	Attribute

PDUT06_H (PPG6): Address 0144h (Access: Half-word)

PDUT06_L (PPG6): Address 0145h (Access: Half-word)

15	14	13	12	11	10	9	8	Bit
PDUT06[15]	PDUT06[14]	PDUT06[13]	PDUT06[12]	PDUT06[11]	PDUT06[10]	PDUT06[9]	PDUT06[8]	
X	X	X	X	X	X	X	X	Initial value
RW	RW	RW	RW	RW	RW	RW	RW	Attribute

7	6	5	4	3	2	1	0	Bit
PDUT06[7]	PDUT06[6]	PDUT06[5]	PDUT06[4]	PDUT06[3]	PDUT06[2]	PDUT06[1]	PDUT06[0]	
X	X	X	X	X	X	X	X	Initial value
RW	RW	RW	RW	RW	RW	RW	RW	Attribute

PDUT07_H (PPG7): Address 014Ch (Access: Half-word)
PDUT07_L (PPG7): Address 014Dh (Access: Half-word)

15	14	13	12	11	10	9	8	Bit
PDUT07[15]	PDUT07[14]	PDUT07[13]	PDUT07[12]	PDUT07[11]	PDUT07[10]	PDUT07[9]	PDUT07[8]	
X	X	X	X	X	X	X	X	Initial value
RW	RW	RW	RW	RW	RW	RW	RW	Attribute

7	6	5	4	3	2	1	0	Bit
PDUT07[7]	PDUT07[6]	PDUT07[5]	PDUT07[4]	PDUT07[3]	PDUT07[2]	PDUT07[1]	PDUT07[0]	
X	X	X	X	X	X	X	X	Initial value
RW	RW	RW	RW	RW	RW	RW	RW	Attribute

PDUT08_H (PPG8): Address 0154h (Access: Half-word)
PDUT08_L (PPG8): Address 0155h (Access: Half-word)

15	14	13	12	11	10	9	8	Bit
PDUT08[15]	PDUT08[14]	PDUT08[13]	PDUT08[12]	PDUT08[11]	PDUT08[10]	PDUT08[9]	PDUT08[8]	
X	X	X	X	X	X	X	X	Initial value
RW	RW	RW	RW	RW	RW	RW	RW	Attribute

7	6	5	4	3	2	1	0	Bit
PDUT08[7]	PDUT08[6]	PDUT08[5]	PDUT08[4]	PDUT08[3]	PDUT08[2]	PDUT08[1]	PDUT08[0]	
X	X	X	X	X	X	X	X	Initial value
RW	RW	RW	RW	RW	RW	RW	RW	Attribute

PDUT09_H (PPG9): Address 015Ch (Access: Half-word)
PDUT09_L (PPG9): Address 015Dh (Access: Half-word)

15	14	13	12	11	10	9	8	Bit
PDUT09[15]	PDUT09[14]	PDUT09[13]	PDUT09[12]	PDUT09[11]	PDUT09[10]	PDUT09[9]	PDUT09[8]	
X	X	X	X	X	X	X	X	Initial value
RW	RW	RW	RW	RW	RW	RW	RW	Attribute

7	6	5	4	3	2	1	0	Bit
PDUT09[7]	PDUT09[6]	PDUT09[5]	PDUT09[4]	PDUT09[3]	PDUT09[2]	PDUT09[1]	PDUT09[0]	
X	X	X	X	X	X	X	X	Initial value
RW	RW	RW	RW	RW	RW	RW	RW	Attribute

PDUT10_H (PPG10): Address 0164h (Access: Half-word)
PDUT10_L (PPG10): Address 0165h (Access: Half-word)

15	14	13	12	11	10	9	8	Bit
PDUT10[15]	PDUT10[14]	PDUT10[13]	PDUT10[12]	PDUT10[11]	PDUT10[10]	PDUT10[9]	PDUT10[8]	
X	X	X	X	X	X	X	X	Initial value
RW	RW	RW	RW	RW	RW	RW	RW	Attribute

7	6	5	4	3	2	1	0	Bit
PDUT10[7]	PDUT10[6]	PDUT10[5]	PDUT10[4]	PDUT10[3]	PDUT10[2]	PDUT10[1]	PDUT10[0]	
X	X	X	X	X	X	X	X	Initial value
RW	RW	RW	RW	RW	RW	RW	RW	Attribute

PDUT11_H (PPG11): Address 016Ch (Access: Half-word)
PDUT11_L (PPG11): Address 016Dh (Access: Half-word)

15	14	13	12	11	10	9	8	Bit
PDUT11[15]	PDUT11[14]	PDUT11[13]	PDUT11[12]	PDUT11[11]	PDUT11[10]	PDUT11[9]	PDUT11[8]	
X	X	X	X	X	X	X	X	Initial value
RW	RW	RW	RW	RW	RW	RW	RW	Attribute

7	6	5	4	3	2	1	0	Bit
PDUT11[7]	PDUT11[6]	PDUT11[5]	PDUT11[4]	PDUT11[3]	PDUT11[2]	PDUT11[1]	PDUT11[0]	
X	X	X	X	X	X	X	X	Initial value
RW	RW	RW	RW	RW	RW	RW	RW	Attribute

- The PPG Duty Setting registers are buffered. Transfers from the buffers to the counter take place automatically at counter overflow or underflow.
- Set a value smaller than the setting of PPG Period Setting register PCSR in a PPG Duty Setting register. (See “Caution”)
- If the same value as set in PPG Period Setting register PCSR is set in a PPG Duty Setting register,
  - “H” is always output to (OSEL=”0”) at normal polarity time.
  - “L” is always output to (OSEL=”1”) at inverted polarity time.
    (The OSEL bit is an output polarity specification bit of the PPG control register PCN.)
- Always access the PPG Duty Setting registers in a half-word (16-bit) format.
  (See “Caution”)
- The read value is all "1".

28.3.4. PCN: PPG Control Status register

Controls the operations and status of PPGs.

- PCN04 (PPG4): Address 0136h (Access: Byte, Half-word)
- PCN05 (PPG5): Address 013Eh (Access: Byte, Half-word)
- PCN06 (PPG6): Address 0146h (Access: Byte, Half-word)
- PCN07 (PPG7): Address 014Eh (Access: Byte, Half-word)
- PCN08 (PPG8): Address 0156h (Access: Byte, Half-word)
- PCN09 (PPG9): Address 015Eh (Access: Byte, Half-word)
- PCN10 (PPG10): Address 0166h (Access: Byte, Half-word)
- PCN11 (PPG11): Address 016Eh (Access: Byte, Half-word)

PCN04_H (PPG4): Address 0136h (Access: Byte, Half-word)
PCN04_L (PPG4): Address 0137h (Access: Byte, Half-word)

15	14	13	12	11	10	9	8	Bit
CNTE04	STGR04	MDSE04	RTRG04	CKS04[1]	CKS04[0]	PGMS04		
0	0	0	0	0	0	0	X	Initial value
R/W	R0/W	R/W	R/W	R/W	R/W	R/W	RX/WX	Attribute
O	O							Rewrite during operation

7	6	5	4	3	2	1	0	Bit
EGS04[1]	EGS04[0]	IREN04	IRQF04	IRS04[1]	IRS04[0]		OSEL04	
0	0	0	0	0	0	X	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	RX/WX	R/W	Attribute
		O	O					Rewrite during operation

PCN05_H (PPG5): Address 013Eh (Access: Byte, Half-word)
PCN05_L (PPG5): Address 013Fh (Access: Byte, Half-word)

15	14	13	12	11	10	9	8	Bit
CNTE05	STGR05	MDSE05	RTRG05	CKS05[1]	CKS05[0]	PGMS05		
0	0	0	0	0	0	0	X	Initial value
R/W	R0/W	R/W	R/W	R/W	R/W	R/W	RX/WX	Attribute
O	O							Rewrite during operation

7	6	5	4	3	2	1	0	Bit
EGS05[1]	EGS05[0]	IREN05	IRQF05	IRS05[1]	IRS05[0]		OSEL05	
0	0	0	0	0	0	X	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	RX/WX	R/W	Attribute
		O	O					Rewrite during

operation

PCN06_H (PPG6): Address 0146h (Access: Byte, Half-word)
PCN06_L (PPG6): Address 0147h (Access: Byte, Half-word)

15	14	13	12	11	10	9	8	Bit
CNTE06	STGR06	MDSE06	RTRG06	CKS06_1	CKS06_0	PGMS06		
0	0	0	0	0	0	0	X	Initial value
R/W	R0/W	R/W	R/W	R/W	R/W	R/W	RX/WX	Attribute
O	O							Rewrite during operation

7	6	5	4	3	2	1	0	Bit
EGS06[1]	EGS06[0]	IREN06	IRQF06	IRS06[1]	IRS06[0]		OSEL06	
0	0	0	0	0	0	X	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	RX/WX	R/W	Attribute
		O	O					Rewrite during operation

PCN07_H (PPG7): Address 014Eh (Access: Byte, Half-word)
PCN07_L (PPG7): Address 014Fh (Access: Byte, Half-word)

15	14	13	12	11	10	9	8	Bit
CNTE07	STGR07	MDSE07	RTRG07	CKS07_1	CKS07_0	PGMS07		
0	0	0	0	0	0	0	X	Initial value
R/W	R0/W	R/W	R/W	R/W	R/W	R/W	RX/WX	Attribute
O	O							Rewrite during operation

7	6	5	4	3	2	1	0	Bit
EGS07[1]	EGS07[0]	IREN07	IRQF07	IRS07[1]	IRS07[0]		OSEL07	
0	0	0	0	0	0	X	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	RX/WX	R/W	Attribute
		O	O					Rewrite during operation

PCN08_H (PPG8): Address 0156h (Access: Byte, Half-word)
PCN08_L (PPG8): Address 0157h (Access: Byte, Half-word)

15	14	13	12	11	10	9	8	Bit
CNTE08	STGR08	MDSE08	RTRG08	CKS08[1]	CKS08[0]	PGMS08		
0	0	0	0	0	0	0	X	Initial value
R/W	R0/W	R/W	R/W	R/W	R/W	R/W	RX/WX	Attribute
O	O							Rewrite during operation

7	6	5	4	3	2	1	0	Bit
EGS08[1]	EGS08[0]	IREN08	IRQF08	IRS08[1]	IRS08[0]		OSEL08	
0	0	0	0	0	0	X	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	RX/WX	R/W	Attribute
		O	O					Rewrite during operation

PCN09_H (PPG9): Address 015Eh (Access: Byte, Half-word)
PCN09_L (PPG9): Address 015Fh (Access: Byte, Half-word)

15	14	13	12	11	10	9	8	Bit
CNTE09	STGR09	MDSE09	RTRG09	CKS09[1]	CKS09[0]	PGMS09		
0	0	0	0	0	0	0	X	Initial value
R/W	R0/W	R/W	R/W	R/W	R/W	R/W	RX/WX	Attribute
O	O							Rewrite during operation

7	6	5	4	3	2	1	0	Bit
EGS09[1]	EGS09[0]	IREN09	IRQF09	IRS09[1]	IRS09[0]		OSEL09	
0	0	0	0	0	0	X	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	RX/WX	R/W	Attribute
		O	O					Rewrite during operation

PCN10_H (PPG10): Address 0166h (Access: Byte, Half-word)
PCN10_L (PPG10): Address 0167h (Access: Byte, Half-word)

15	14	13	12	11	10	9	8	Bit
CNTE10	STGR10	MDSE10	RTRG10	CKS10[1]	CKS10[0]	PGMS10		
0	0	0	0	0	0	0	X	Initial value
R/W	R0/W	R/W	R/W	R/W	R/W	R/W	RX/WX	Attribute
O	O							Rewrite during operation

7	6	5	4	3	2	1	0	Bit
EGS10[1]	EGS10[0]	IREN10	IRQF10	IRS10[1]	IRS10[0]		OSEL10	
0	0	0	0	0	0	X	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	RX/WX	R/W	Attribute
		O	O					Rewrite during operation

PCN11_H (PPG11): Address 016Eh (Access: Byte, Half-word)
PCN11_L (PPG11): Address 016Fh (Access: Byte, Half-word)

15	14	13	12	11	10	9	8	Bit
CNTE11	STGR11	MDSE11	RTRG11	CKS11[1]	CKS11[0]	PGMS11		
0	0	0	0	0	0	0	X	Initial value
R/W	R0/W	R/W	R/W	R/W	R/W	R/W	RX/WX	Attribute
O	O							Rewrite during operation

7	6	5	4	3	2	1	0	Bit
EGS11[1]	EGS11[0]	IREN11	IRQF11	IRS11[1]	IRS11[0]		OSEL11	
0	0	0	0	0	0	X	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	RX/WX	R/W	Attribute
		O	O					Rewrite during operation

O: Rewritable, x: Not writable (See “Caution”)

- Bit 15: Timer enable operation

CNTE	Operation
0	Stop
1	Operation

This bit enables the operation of the PPG

.

- Bit 14: Software trigger

STGR	Operation
0	The operation is unaffected by writing (The read value always equals “0”).
1	Software trigger activation

When the Software Trigger bit is set to “1”, a software trigger is generated to activate the PPG, separately from the generation of an internal trigger (EN bit, reload timer output).

- Bit 13: Mode selection

MDSE	Mode
0	PWM operation
1	One-shot operation

When the Mode Selection bit is set to “0”, a PWM operation is enabled to generate pulses in sequence. When the Mode Selection bit is set to “1”, pulse output takes place only once.

- Bit 12: Restart enable

RTRG	Operation
0	Disable restart.
1	Enable restart.

When the Enable Restart bit is set to “1“, a trigger (software/internal) leads to a restart of the PPG, (even if the PPG is enabled (CNTE = “1“)). If the Enable Restart bit is set to “0” the trigger has no effect on the

function of the PPG.

- Bits 11-10: Counter clock selection

CKS1	CKS0	Down Counter Count Clock Selection
0	0	Peripheral clock (CLKP)
0	1	Peripheral clock divided by 4
1	0	Peripheral clock divided by 16
1	1	Peripheral clock divided by 64

- Bit 9: PPG output mask selection

PGMS	Operation
0	No output mask
1	Output mask (Output “L” level latched:OSEL=“0”)

When the PPG Output Mask Selection bit is set to “1”, the PPG output can be clamped at “L” or “H” regardless of the mode, cycle, and duty settings.
The output level can be specified using the Output Polarity Specification bit (PCN.OSEL).

- Bit 8: Undefined. The operation is unaffected by writing. The read value is indeterminate.
- Bits 7-6: Trigger input edge selection

EGS1	EGS0	Selected Edge
0	0	The operation is unaffected by writing.
0	1	Rising edge
1	0	Falling edge
1	1	Both edges (rising edge, or, falling edge)

Select an edge to trigger the activation of the trigger input selected with the Trigger Specification bits (GCN11[15:12]), (GCN11[11:8]), (GCN11[7:4]), and (GCN11[3:0]) of PPG7 to PPG4, (GCN12[15:12]), (GCN12[11:8]), (GCN12[7:4]), and (GCN12[3:0]) of PPG11 to PPG8, using the Trigger Input Edge Selection bit (EGS[1:0]).

- Bit 5: Interrupt request enable

IREN	Operation
0	Interrupt request disable
1	Enable interrupt requests.

- Bit 4: interrupt request flag

IRQF	Read Operation	Write Operation
0	No interrupt request(Read)	Clear the Interrupt Request flag.(Write)
1	Interrupt request (Read)	The operation is unaffected by writing. (Write)

If the Interrupt Request flag (IRQF) equals “1” and writing “0” to the flag take place at the same time, the setting of the Interrupt Request flag (IRQF=“1”) overrides.

- Bit 3-2: Interrupt cause selection

IRS1	IRS0	Selection
0	0	Software trigger, or, trigger input
0	1	Counter borrow
1	0	The counter matches the duty value.
1	1	Counter borrow, or the counter equals the duty value.

Select the operation in which to generate an interrupt request.

- Bit 1: Undefined. The operation is unaffected by writing. The read value is indeterminate.
- Bit 0: PPG output polarity specification

OSEL	Operation
0	Normal polarity
1	Inverted polarity

When the PPG Output Mask Selection bit (PCN.PGMS) has been set to “1”, if the Output Polarity Specification bit (OSEL) is set to “0”, the output is clamped at “L”; if the Output Polarity Specification bit is set to “1”, the output is clamped at “H”.

28.3.5. GCN1: General Control register 1

Selects a trigger input to PPG4-PPG7 and PPG8-PPG11.

- GCN11 (PPG4-PPG7): Address 0104h (Access: Half-word)
- GCN12 (PPG8-PPG11): Address 0108h (Access: Half-word)

GCN11_H (PPG4-PPG7): Address 0104h (Access: Half-word)
GCN11_L (PPG4-PPG7): Address 0105h (Access: Half-word)

15	14	13	12	11	10	9	8	Bit
GCN11_TSEL3[3]	GCN11_TSEL3[2]	GCN11_TSEL3[1]	GCN11_TSEL3[0]	GCN11_TSEL2[3]	GCN11_TSEL2[2]	GCN11_TSEL2[1]	GCN11_TSEL2[0]	
0	0	1	1	0	0	1	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	Bit
GCN11_TSEL1[3]	GCN11_TSEL1[2]	GCN11_TSEL1[1]	GCN11_TSEL1[0]	GCN11_TSEL0[3]	GCN11_TSEL0[2]	GCN11_TSEL0[1]	GCN11_TSEL0[0]	
0	0	0	1	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

PPG4-PPG7:

- GCN11: Bits 15-12 (GCN11_TSEL3[3:0]) PPG7 trigger specification
- GCN11: Bits 11-8 (GCN11_TSEL2[3:0]) PPG6 trigger specification
- GCN11: Bits 7-4 (GCN11_TSEL1[3:0]) PPG5 trigger specification
- GCN11: Bits 3-0 (GCN11_TSEL0[3:0]) PPG4 trigger specification

GCN12_H (PPG8-PPG11): Address 0108h (Access: Half-word)
GCN12_L (PPG8-PPG11): Address 0109h (Access: Half-word)

15	14	13	12	11	10	9	8	Bit
GCN12_TSEL3[3]	GCN12_TSEL3[2]	GCN12_TSEL3[1]	GCN12_TSEL3[0]	GCN12_TSEL2[3]	GCN12_TSEL2[2]	GCN12_TSEL2[1]	GCN12_TSEL2[0]	
0	0	1	1	0	0	1	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	Bit
GCN12_TSEL1[3]	GCN12_TSEL1[2]	GCN12_TSEL1[1]	GCN12_TSEL1[0]	GCN12_TSEL0[3]	GCN12_TSEL0[2]	GCN12_TSEL0[1]	GCN12_TSEL0[0]	
0	0	0	1	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

PPG8-PPG11:

- GCN12: Bits 15-12 (GCN12_TSEL3[3:0]) PPG11 trigger specification
- GCN12: Bits 11-8 (GCN12_TSEL2[3:0]) PPG10 trigger specification
- GCN12: Bits 7-4 (GCN12_TSEL1[3:0]) PPG9 trigger specification
- GCN12: Bits 3-0 (GCN12_TSEL0[3:0]) PPG8 trigger specification

TSEL				Activation trigger specification
0	0	0	0	EN0 bit (GCN2 register)
0	0	0	1	EN1 bit (GCN2 register)
0	0	1	0	EN2 bit (GCN2 register)
0	0	1	1	EN3 bit (GCN2 register)
0	1	0	0	16-bit reload timer 0/2/4/6
0	1	0	1	16-bit reload timer 1/3/5/7
None of the above				Disabled (See “Caution”)

- PPG4 to PPG11 as selected are activated when the edge specified by the Trigger Input Edge Selection bits (PCN.EGS[1:0]) are detected during the specified activation trigger.
- For detailed setting of each channel see chapter 29.7.7 What activation triggers are available and how

are they selected? (Page No.29-30)

28.3.6. GCN2: General Control Register 2

Generates PPG4-PPG7 and PPG8-PPG11 internal trigger levels using software.

- GCN21 (PPG4-PPG7): Address 0107h (Access: Byte)
- GCN22 (PPG8-PPG11): Address 010Bh (Access: Byte)

GCN21 (PPG4-PPG7): Address 0107h (Access: Byte)

7	6	5	4	3	2	1	0	bit
				GCN21_EN3	GCN21_EN2	GCN21_EN1	GCN21_EN0	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

- Bits 7-4: Undefined. Always write “0”. The read value is the value as written. (See “Caution”)
- Bit 3: GCN21_EN3 trigger input
- Bit 2: GCN21_EN2 trigger input
- Bit 1: GCN21_EN1 trigger input
- Bit 0: GCN21_EN0 trigger input

GCN21_EN0, GCN21_EN1, GCN21_EN2, and GCN21_EN3	Internal Triggers GCN21_EN0, GCN21_EN1, GCN21_EN2, and GCN21_EN3
0	Set the level to “L”.
1	Set the level to “H”.

- Set the levels of internal triggers GCN21_EN0, GCN21_EN1, GCN21_EN2, and GCN21_EN3.
- If any of the EN trigger inputs (GCN21_EN0, GCN21?EN1, GCN21_EN2, GCN21_EN3) is selected with the trigger specification bits (GCN11_TSEL0[3:0], GCN11_TSEL1[3:0], GCN11_TSEL2[3:0], and GCN11_TSEL3[3:0]) of PPG4, PPG5, PPG6, PPG7, then the selected EN serves as a PPG trigger input bit.
- If the state selected with the trigger input edge selection bit (EGS[1:0]) is generated by software using the trigger input bit (selected GCN21_EN0, GCN21_EN1, GCN21_EN2, or GCN21_EN3), the choice serves as an activation trigger to activate the PPG.

GCN22 (PPG8-PPG11): Address 010Bh (Access: Byte)

7	6	5	4	3	2	1	0	bit
				GCN22_EN3	GCN22_EN2	GCN22_EN1	GCN22_EN0	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

- Bits 7-4: Undefined. Always write “0”. The read value is the value as written. (See “Caution”)
- Bit 3: GCN22_EN3 trigger input
- Bit 2: GCN22_EN2 trigger input
- Bit 1: GCN22_EN1 trigger input
- Bit 0: GCN22_EN0 trigger input

GCN22_EN0, GCN22_EN1, GCN22_EN2, and GCN22_EN3	Internal Triggers EN0, EN1, EN2, and EN3
0	Set the level to “L”.
1	Set the level to “H”.

- Set the levels of internal triggers GCN22_EN0, GCN22_EN1, GCN22_EN2, and GCN22_EN3.
- If any of the EN trigger inputs (HCN22_EN0, GCN22_EN1, GCN22_EN2, GCN22_EN3) is selected

with the trigger specification bits (GCN12_TSEL0[3:0], GCN12_TSEL1[3:0], GCN12_TSEL2[3:0], and GCN12_TSEL3[3:0]) of PPG8, PPG9, PPG10, PPG11, then the selected EN serves as a PPG trigger input bit.

- If the state selected with the trigger input edge selection bit (EGS[1:0]) is generated by software using the trigger input bit (selected GCN22_EN0, GCN22_EN1, GCN22_EN2, or GCN22_EN3), the choice serves as an activation trigger to activate the PPG.

28.3.7. PTMR: PPG Timer Register

Reads the counts of PPG4-PPG7 and PPG8-PPG11.

- PTMR04 (PPG4): Address 0130h (Access: Half-word)
- PTMR05 (PPG5): Address 0138h (Access: Half-word)
- PTMR06 (PPG6): Address 0140h (Access: Half-word)
- PTMR07 (PPG7): Address 0148h (Access: Half-word)
- PTMR08 (PPG8): Address 0150h (Access: Half-word)
- PTMR09 (PPG9): Address 0158h (Access: Half-word)
- PTMR10 (PPG10): Address 0160h (Access: Half-word)
- PTMR11 (PPG11): Address 0168h (Access: Half-word)

PTMR04_H (PPG4): Address 0130h (Access: Half-word)
PTMR04_L (PPG4): Address 0131h (Access: Half-word)

15	14	13	12	11	10	9	8	Bit
PTMR04[15]	PTMR04[14]	PTMR04[13]	PTMR04[12]	PTMR04[12]	PTMR04[10]	PTMR04[9]	PTMR04[8]	
1	1	1	1	1	1	1	1	Initial value
R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	Attribute

7	6	5	4	3	2	1	0	Bit
PTMR04[7]	PTMR04[6]	PTMR04[5]	PTMR04[4]	PTMR04[3]	PTMR04[2]	PTMR04[1]	PTMR04[0]	
1	1	1	1	1	1	1	1	Initial value
R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	Attribute

PTMR05_H (PPG5): Address 0138h (Access: Half-word)
PTMR05_L (PPG5): Address 0139h (Access: Half-word)

15	14	13	12	11	10	9	8	Bit
PTMR05[15]	PTMR05[14]	PTMR05[13]	PTMR05[12]	PTMR05[12]	PTMR05[10]	PTMR05[9]	PTMR05[8]	
1	1	1	1	1	1	1	1	Initial value
R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	Attribute

7	6	5	4	3	2	1	0	Bit
PTMR05[7]	PTMR05[6]	PTMR05[5]	PTMR05[4]	PTMR05[3]	PTMR05[2]	PTMR05[1]	PTMR05[0]	
1	1	1	1	1	1	1	1	Initial value
R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	Attribute

PTMR06_H (PPG6): Address 0140h (Access: Half-word)
PTMR06_L (PPG6): Address 0141h (Access: Half-word)

15	14	13	12	11	10	9	8	Bit
PTMR06[15]	PTMR06[14]	PTMR06[13]	PTMR06[12]	PTMR06[12]	PTMR06[10]	PTMR06[9]	PTMR06[8]	
1	1	1	1	1	1	1	1	Initial value
R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	Attribute

7	6	5	4	3	2	1	0	Bit
PTMR06[7]	PTMR06[6]	PTMR06[5]	PTMR06[4]	PTMR06[3]	PTMR06[2]	PTMR06[1]	PTMR06[0]	
1	1	1	1	1	1	1	1	Initial value
R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	Attribute

PTMR07_H (PPG7): Address 0148h (Access: Half-word)
PTMR07_L (PPG7): Address 0149h (Access: Half-word)

15	14	13	12	11	10	9	8	Bit
PTMR07[15]	PTMR07[14]	PTMR07[13]	PTMR07[12]	PTMR07[12]	PTMR07[10]	PTMR07[9]	PTMR07[8]	
1	1	1	1	1	1	1	1	Initial value
R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	Attribute

7	6	5	4	3	2	1	0	Bit
PTMR07[7]	PTMR07[6]	PTMR07[5]	PTMR07[4]	PTMR07[3]	PTMR07[2]	PTMR07[1]	PTMR07[0]	
1	1	1	1	1	1	1	1	Initial value
R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	Attribute

PTMR08_H (PPG8): Address 0150h (Access: Half-word)
PTMR08_L (PPG8): Address 0151h (Access: Half-word)

15	14	13	12	11	10	9	8	Bit
PTMR08[15]	PTMR08[14]	PTMR08[13]	PTMR08[12]	PTMR08[12]	PTMR08[10]	PTMR08[9]	PTMR08[8]	
1	1	1	1	1	1	1	1	Initial value
R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	Attribute

7	6	5	4	3	2	1	0	Bit
PTMR08[7]	PTMR08[6]	PTMR08[5]	PTMR08[4]	PTMR08[3]	PTMR08[2]	PTMR08[1]	PTMR08[0]	
1	1	1	1	1	1	1	1	Initial value
R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	Attribute

PTMR09_H (PPG9): Address 0158h (Access: Half-word)
PTMR09_L (PPG9): Address 0159h (Access: Half-word)

15	14	13	12	11	10	9	8	Bit
PTMR09[15]	PTMR09[14]	PTMR09[13]	PTMR09[12]	PTMR09[12]	PTMR09[10]	PTMR09[9]	PTMR09[8]	
1	1	1	1	1	1	1	1	Initial value
R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	Attribute

7	6	5	4	3	2	1	0	Bit
PTMR09[7]	PTMR09[6]	PTMR09[5]	PTMR09[4]	PTMR09[3]	PTMR09[2]	PTMR09[1]	PTMR09[0]	
1	1	1	1	1	1	1	1	Initial value
R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	Attribute

PTMR10_H (PPG10): Address 0160h (Access: Half-word)
PTMR10_L (PPG10): Address 0161h (Access: Half-word)

15	14	13	12	11	10	9	8	Bit
PTMR10[15]	PTMR10[14]	PTMR10[13]	PTMR10[12]	PTMR10[12]	PTMR10[10]	PTMR10[9]	PTMR10[8]	
1	1	1	1	1	1	1	1	Initial value
R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	Attribute

7	6	5	4	3	2	1	0	Bit
PTMR10[7]	PTMR10[6]	PTMR10[5]	PTMR10[4]	PTMR10[3]	PTMR10[2]	PTMR10[1]	PTMR10[0]	
1	1	1	1	1	1	1	1	Initial value
R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	Attribute

PTMR11_H (PPG11): Address 0168h (Access: Half-word)
PTMR11_L (PPG11): Address 0169h (Access: Half-word)

15	14	13	12	11	10	9	8	Bit
PTMR11[15]	PTMR11[14]	PTMR11[13]	PTMR11[12]	PTMR11[12]	PTMR11[10]	PTMR11[9]	PTMR11[8]	
1	1	1	1	1	1	1	1	Initial value
R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	Attribute

7	6	5	4	3	2	1	0	Bit
PTMR11[7]	PTMR11[6]	PTMR11[5]	PTMR11[4]	PTMR11[3]	PTMR11[2]	PTMR11[1]	PTMR11[0]	
1	1	1	1	1	1	1	1	Initial value
R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	Attribute

- The count of the 16-bit down counter can be read.
- Be sure to access the PPG Timer register PTMR in half words (16 bits).
- The register will not be read correctly if it is byte-accessed.

28.4. Operation

The MB88F333 features a maximum of 8 programmable pulse generators (PPGs), which provide programmable pulse outputs independently or jointly.
The individual modes of operation are described below.

28.4.1. PWM Operation

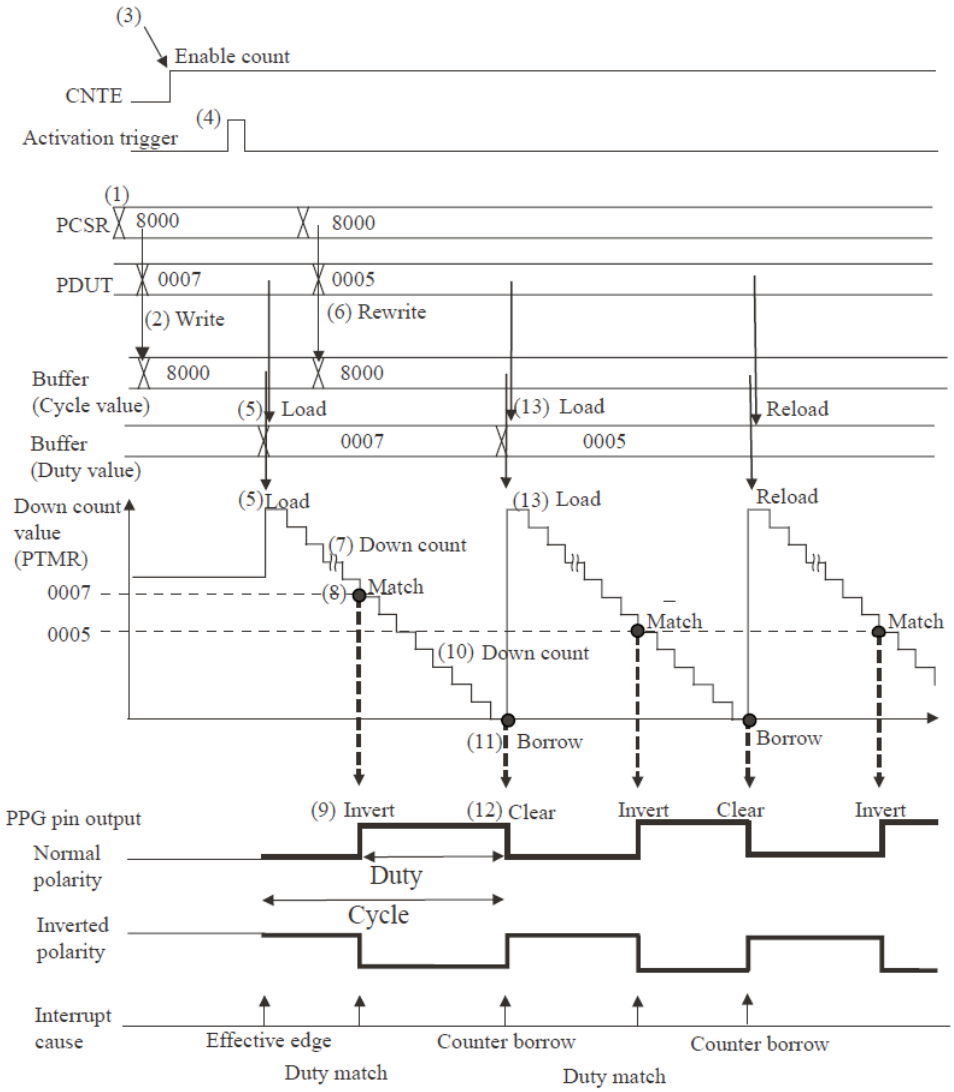


Figure 29-5 In PWM operation, variable-duty pulses are generated from the PPG pin

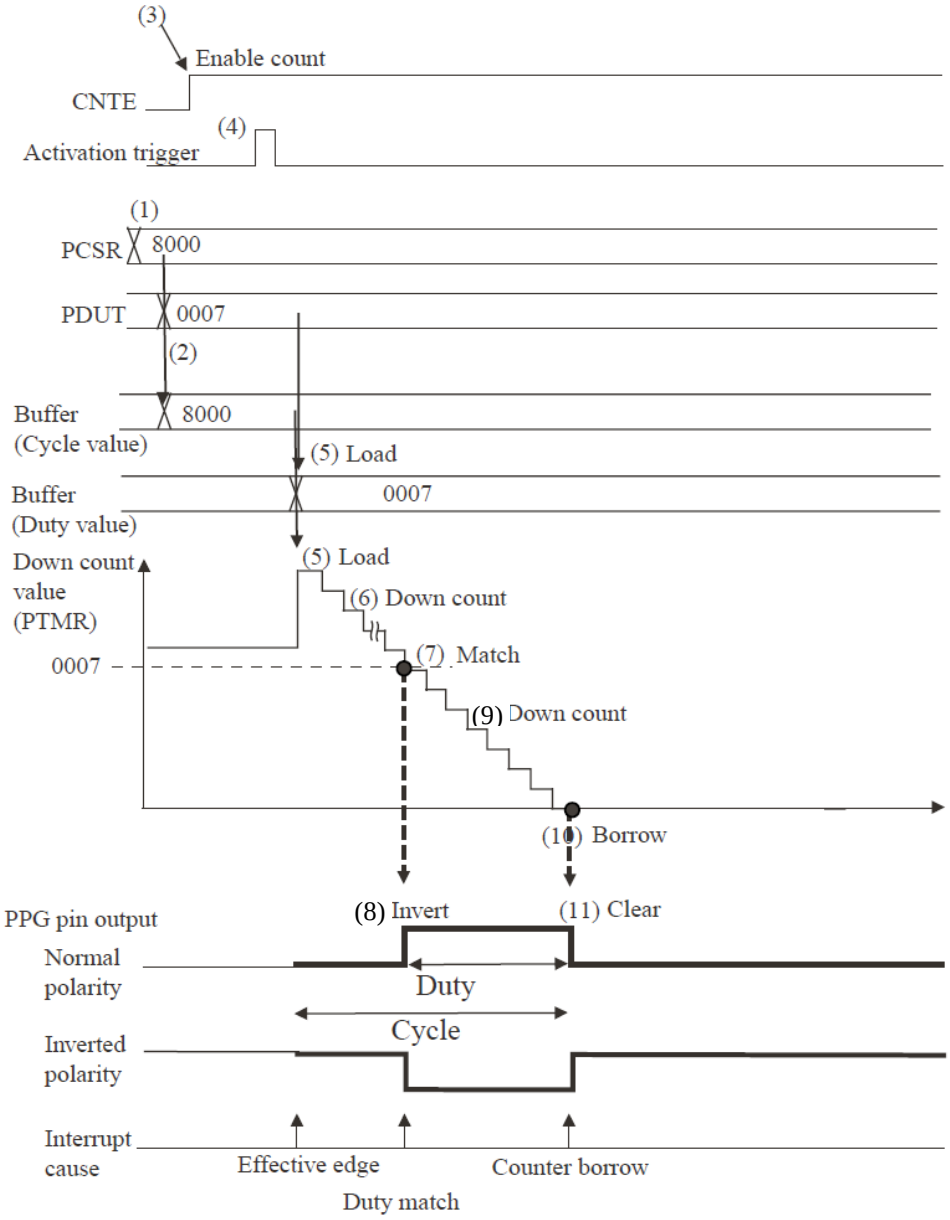
(1) Write a cycle value.
(2) Write a duty value and transfer the cycle value to buffers.
(3) Enable PPG operation.
(4) Generate an activation trigger.
(5) Load the cycle and duty values.
(6) Rewrite the duty value and transfer the cycle value to buffers.
(7) Counter down count
(8) The down counter equals the duty value.
(9) Inverses the PPG pin output level.

(10) Counter down count
(11) Counter borrow
(12) Clear the PPG pin output level (return to normal).
(13) Reload the cycle value.
(14) Reload the duty value.
(15) Steps from (7) to (14) are iterated.
(See 'Caution')

Equation
Period = {Period value (PCSR) + 1} x Count clock
Duty = {Duty value (PDUT) + 1} x Count clock
Width up to pulse output = {Period value (PCSR) – Duty value (PDUT)} x Count clock

28.4.2. One-Shot Operation

In one-shot operation, one-shot pulses are generated from the PPG pin.



(1) Write a cycle value.
(2) Write a duty value and transfer the cycle value to buffers.
(3) Enable PPG operation.
(4) Generate an activation trigger.
(5) Load the cycle and duty values.
(6) Counter down count
(7) Down counter value and duty value
(8) Inverse the PPG pin output level.
(9) Counter down count
(10) Counter borrow
(11) Clear the PPG pin output level (return to normal).
(12) The operating sequence is now completed.
(See “Caution”)

28.4.3. Restart Operation

The restart operation is described below.

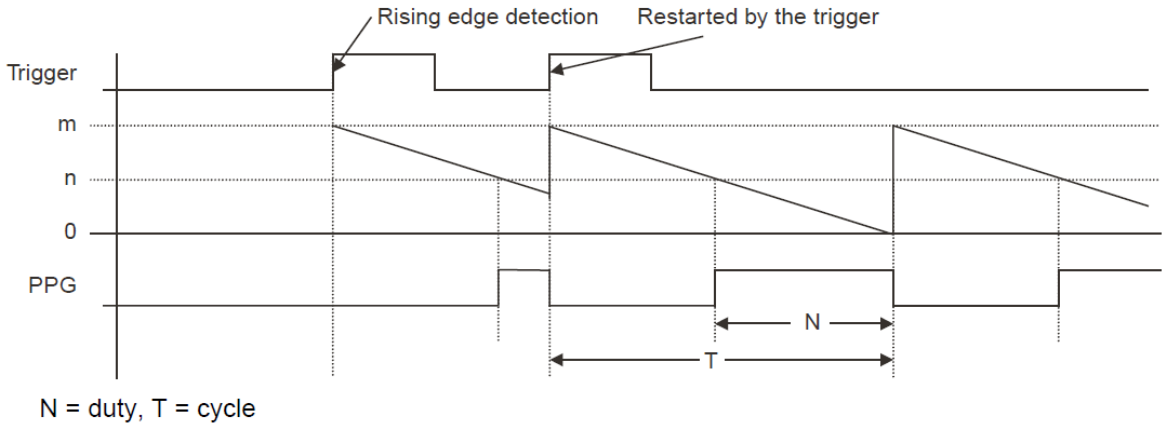


Figure 29-6 Restart available in PWM operation

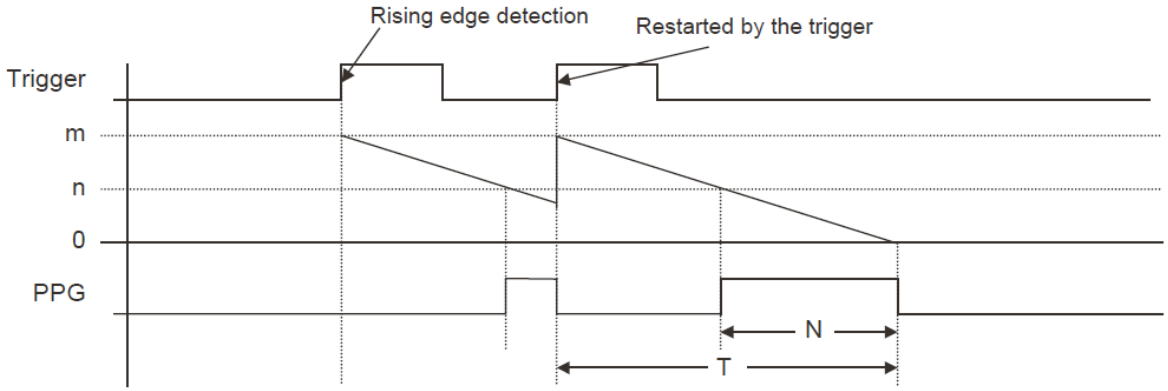


Figure 29-7 Restart available in one-shot operation

If a restart is not available, the second and subsequent triggers have no effect in both PWM and one-shot operations.
(The second and subsequent triggers following a shutdown of the down counter are functional.)

28.5. Setting

Table 29-1 Settings Needed to Start the PPG

Setting	Setting Registers	Setting Procedure*
Period and duty value settings	PPG cycle settings (PCSR04-PCSR11) PPG duty settings (PDUT04-PDUT11)	29.7.1
Enable PPG operation.	PPG control status (PCN04-PCN11)	29.7.2
Operation mode selection (PWM/one-shot)		29.7.3
Enable restart.		29.7.4
Count clock selection		29.7.5
PPG output mask selection		29.7.6
Trigger selection Software Internal trigger	General Control 1 (, GCN11, GCN12, GCN13)	29.7.7
Output polarity specification		29.7.8
PPG pin output setting	Port functions	29.7.9
Trigger generation (software trigger) (Reload timer) (GCN2.EN bit)	PPG Control Status (PCN04-PCN11)	29.7.10
	See “Chapter 31 Reload Timer”.	
	General Control 2 (GCN20, GCN21, GCN22, GCN23)	

* For refer to the section indicated by the number.

Table 29-2 Settings Needed to Stop the PPG

Setting	Setting Registers	Setting Procedure*
PPG Timer enable operation setting	PPG control status (PCN04-PCN11)	29.7.11

*For the setting procedure, refer to the section indicated by the number.

Table 29-3 Settings Needed to Clamp the Output Level

Setting	Setting Registers	Setting Procedure*
Output polarity specification	PPG control status (PCN04-PCN11)	29.7.8
PPG output mask selection		29.7.6
Period value = Duty value setting	PPG duty settings (PDUT04-PDUT11)	29.7.6

*For the setting procedure, refer to the section indicated by the number.

Table 29-4 Settings Needed to Implement PPG Interrupts

Setting	Setting Registers	Setting Procedure*
PPG interrupt cause selection (Generate an activation trigger, borrow, and duty match)	PPG control status (PCN04-PCN11)	29.7.12
PPG interrupt setting Clear interrupt requests. Enable interrupt requests.		29.7.13

*For the setting procedure, please refer to the section stated.

28.6. Q & A

28.6.1. How do I set (rewrite) a cycle and a duty?

Period and duty value settings

- Set each cycle value in PPG Period Setting Register PCSR.
- Set each duty value in PPG Duty Setting Register PDUT.
- The PPG Period Setting and the PPG Duty Setting registers each have a buffer to allow the user to ignore the write timing.
  - Equation
    PCSR register value = {Cycle/Count clock} –1
    PDUT register value = {“H” width (duty)*/Count clock} –1

*: Normal polarity (OSEL= 0)

- Allowed range

PCSR register value = PCSR register value - FFFFh (65535)
PDUT register value = 0 - PCSR register value

Note : Be sure to set a 'duty' cycle following the setting of a cycle. (See “Caution”)

28.6.2. How do I enable or disable PPG operations?

Enabling the PPG operation
Use the PPG operation enable bit (PCN.CNTE).

Control	PPG Operation Enable Bit (CNTE)
To stop a PPG operation	Set “0”.
To enable a PPG operation	Set “1”.

Enable PPG operation before starting the PPG.
(See “Caution”)

28.6.3. How do I set the PPG operation mode (PWM operation/one-shot operation)?

Operation mode selection
Use the mode selection bit (PCN.MDSE).

Operation Mode	Mode Selection Bit (MDSE)
To implement a PWM operation	Set “0”.
To implement a one-shot operation	Set “1”.

(See “Caution”)

28.6.4. How do I restart it?

Enable restart.
A restart of a PPG can be enabled while the PPG is in operation.
Use the Enable Restart bit (PCN.RTRG) to set.
(See “Caution”)

28.6.5. What count clocks are available and how are they selected?

Count clock selection

The count clock is selectable out of the four choices listed below.

Use the count clock selection bit (PCN.CKS[1:0]).

Count Clock	Count Clock Selection Bit		(Example) CLKP = 41.66MHz	
	CKS1	CKS0	Count Clock	Period (1 - FFFFh)
CLKP	0	0	41.66MHz	48ns – 1.573ms
CLKP/4	0	1	10.42MHz	192ns – 6.289ms
CLKP/16	1	0	2.6MHz	769ns – 25.206ms
CLKP/64	1	1	650kHz	3.1µs – 100.825ms

(See “Caution”)

28.6.6. How do I clamp the PPG pin output level?

PPG output mask selection

The level of PPG pin output can be clamped.

Use the PPG Output Mask Selection bit (PCN.PGMS) and the duty value (PDUT) to set.

PPG Pin Output	PPG Output Polarity Specification Bit (OSEL)	Setting Procedure
To clamp the “L” level under normal polarity	When “0”	Set the PPG Output Mask Selection bit (PGMS) to “1”.
To clamp the “H” level under normal polarity	When “0”	Period value (PCSR) = Set a duty value (PDUT).
To clamp the “H” level under inverted polarity	When “1”	Set the PPG Output Mask Selection bit (PGMS) to “1”.
To clamp the “L” level under inverted polarity	When “1”	Period value (PCSR) = Set a duty value (PDUT).

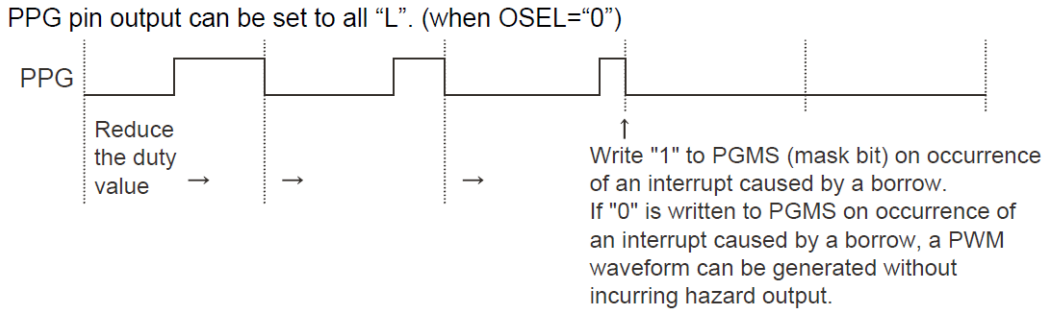


Figure 29-8 PPG pin output can be set to all “L”. (When OSEL=“0”)

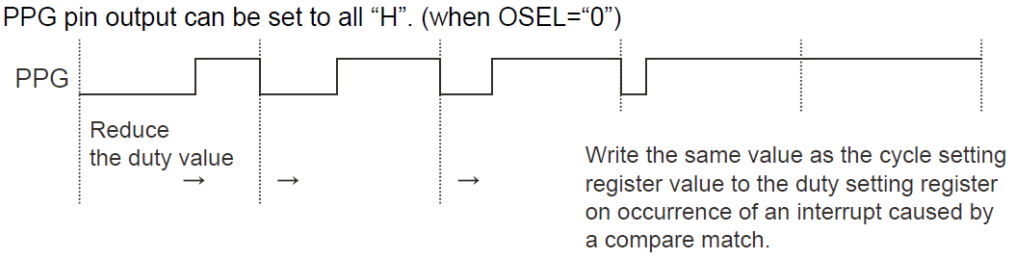


Figure 29-9 PPG pin output can be set to all “H”. (When OSEL=“0”)

PPG output will also equal all “H” if “0” is set in both the PPG Period Setting Register (PCSR) and PPG Duty Setting Register (PDUT). (When OSEL=“0”)

28.6.7. What activation triggers are available and how are they selected?

- Trigger selection
  - Activation triggers are broadly grouped into software triggers and internal triggers.
  - Software triggers work at all times.

A trigger is set using the trigger specification bits (GCN1.TSEL0[3:0]), (GCN1.TSEL1[3:0]), (GCN1.TSEL2[3:0]), and (GCN1.TSEL3[3:0]).

Triggers are selectable for PPG4, PPG5, PPG6, and PPG7 independently.

Internal Trigger	PPG4	PPG5	PPG6	PPG7
	Internal Trigger Specification Bit			
	TSEL0[3:0]	TSEL1[3:0]	TSEL2[3:0]	TSEL3[3:0]
To select the EN0 bit of the GCN21 register	Set “0000”	Set “0000”	Set “0000”	Set “0000”
To select the EN1 bit of the GCN21 register	Set “0001”	Set “0001”	Set “0001”	Set “0001”
To select the EN2 bit of the GCN21 register	Set “0010”	Set “0010”	Set “0010”	Set “0010”
To select the EN3 bit of the GCN21 register	Set “0011”	Set “0011”	Set “0011”	Set “0011”
To select reload timer 2	Set “0100”	Set “0100”	Set “0100”	Set “0100”
To select reload timer 3	Set “0101”	Set “0101”	Set “0101”	Set “0101”

Triggers are selectable for PPG8, PPG9, PPG10, and PPG11 independently.

Internal Trigger	PPG8	PPG9	PPG10	PPG11
	Internal Trigger Specification Bit			
	TSEL0[3:0]	TSEL1[3:0]	TSEL2[3:0]	TSEL3[3:0]
To select the EN0 bit of the GCN22 register	Set “0000”	Set “0000”	Set “0000”	Set “0000”
To select the EN1 bit of the GCN22 register	Set “0001”	Set “0001”	Set “0001”	Set “0001”
To select the EN2 bit of the GCN22 register	Set “0010”	Set “0010”	Set “0010”	Set “0010”
To select the EN3 bit of the GCN22 register	Set “0011”	Set “0011”	Set “0011”	Set “0011”
To select reload timer 4	Set “0100”	Set “0100”	Set “0100”	Set “0100”
To select reload timer 5	Set “0101”	Set “0101”	Set “0101”	Set “0101”

The same trigger can be specified for a group of PPGs to activate all these PPGs simultaneously. (See “Caution”)

Trigger edge selection

Trigger edges are set using trigger input edge selection bits (PCN.EG[1:0]).

Trigger Edge Selection	Trigger Input Edge Selection Bits (EG1-EG0)
When not detected (software trigger only)	Set “00”.
“L” -> “H” Trigger generated on the rising edge	Set “01”.
“H” -> “L” Trigger generated on the falling edge	Set “10”.
Trigger generated on both edges	Set “11”.

(See “Caution”)

28.6.8. How do I invert the output polarity?

Output polarity specification

The polarity in the normal state can be specified as follows:

Use the PPG Output Polarity Specification bit (PCN.OSEL) to set.

(“Normal state” means the state in which pulse output is not executed.)

Output Level in Normal State	PPG Output Polarity Specification Bit (OSEL)
To enable “L” level output (normal polarity)	Set “0”.
To enable “H” level output (inverted polarity)	Set “1”.

(See “Caution”)

28.6.9. How do I program a pin as a PPG output pin?

Please refer to "1.7.1 Pin Multiplexing".

28.6.10. How do I generate an activation trigger?

Generating a trigger

Methods of generating an activation trigger are described below.

- Activating a software trigger
  Use the Software Trigger bit (PCN.STGR) to set.
  Write “1” to the Software Trigger bit (STGR) to generate an activation trigger.
  Always functional, regardless of the internal trigger.
- Activating PPGs with reload timers
  The reload timers need to be set up and activated. For more information, see “Chapter 31 Reload Timer)”.
  An activation trigger is generated when the edge specified by the reload timer output signal is generated with the reload timer underflow.
- Activating a PPG with the EN trigger input bits (GCN2.EN0) - (GCN2.EN3)
  An activation trigger can be generated by rewriting the level of the EN trigger input bits (GCN2.EN0) - (GCN2.EN3).

Edge	Software-Based Setting Procedure (EN0, EN1, EN2, EN3)
Rising edge	First, set the EN bit to “0”, then the EN bit to “1”.
Falling edge	First, set the EN bit to “1”, then to “0”.

- Activating multiple PPGs concurrently
- The same trigger (trigger input bit) can be specified with the PPG trigger specification bits to activate all the PPGs simultaneously when the trigger is generated.
- Even if an activation trigger is generated before the operation of a PPG is enabled, that PPG would not be activated. Be sure to enable the operation of a PPG before generating a trigger to activate it. (See “ How do I enable or disable PPG operations?”.)

28.6.11. How do I stop a PPG operation?

PPG Timer enable operation setting (See “How do I enable or disable PPG operations?”)

28.6.12. What interrupts are available and how are they selected?

Interrupt cause selection

Four kinds of interrupts are selectable as follows:

Use the Interrupt Cause Setting bit (PCN.IRS[1:0]) to set.

Interrupt Cause	Interrupt Cause Setting Bit (IRS[1:0])
Software trigger or Internal trigger generation (PPG4-PPG11)	Set “00”.
Down counter borrow (cycle match)	Set “01”.
Duty match	Set “10”.
Down counter borrow (cycle match) or Duty match	Set “11”.

28.6.13. How do I enable, disable and clear interrupts?

Interrupt Request Enable flag, Interrupt Request flag

Use the interrupt request enable bit (PCN.IREN) to enable interrupts.

	Interrupt Request Enable Bit (IREN)
To disable interrupt requests	Set “0”.
To enable interrupt requests	Set “1”.

Use the interrupt request bit (PCN.IRQF) to clear interrupt requests.

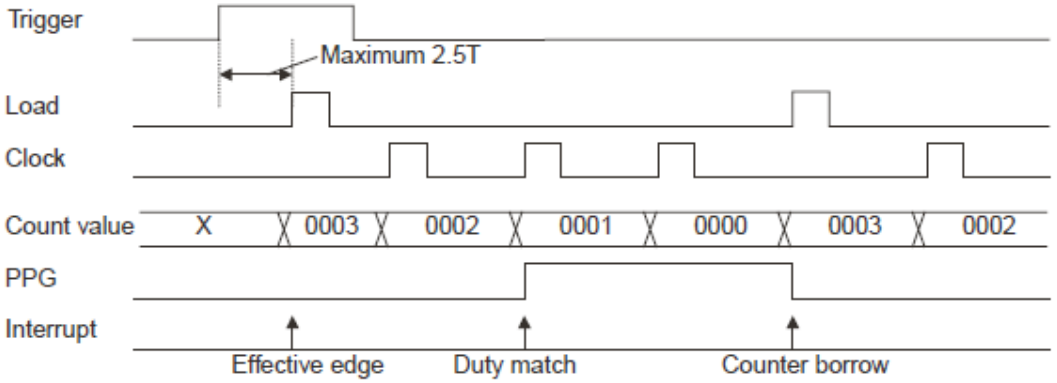
	Interrupt Request Bit (IRQF)
To select interrupt request	Write “0”.

(See “Caution”)

28.7. Caution

- If the Interrupt Request flag (PCN.IRQF) equals “1” and the Interrupt Request flag is set to “0” at the same time, the setting of the Interrupt Request flag to “1” overrides the flag clear request.

- The first load comes with a maximum delay of 2.5T after the activation trigger. (T: Count clock)
  If the down counter is loaded and counts at the same time, the load operation overrides.



- Be sure to write duty value PDUT after cycle PCSR has been initialized and rewritten.
  (Always write in the order of (1)PCSR and (2)PDUT.)
  Only the PDUT can be written for rewriting the duty.

- Set the duty value PDUT smaller than the cycle value of PCSR. If a larger value has been set, disable the operation of the PPG before replacing the duty with a smaller value.

- Always access PPG Period Setting registers PCSR and PPG Duty Setting registers in a half-word (16-bit) format. If these registers are byte-accessed, no values would be written to their upper and lower bit positions.

- To activate a PPG, it is necessary to set the Timer Operation Enable bits (PCN.CNTE) to “1” before or concurrently with the activation to enable the PPG operation.

- The values of mode (MDSE), restart enable (RTRG), count clock (CKS[1:0]), trigger input edge (EGS[1:0]), interrupt cause (IRS), internal trigger (TSEL) and output polarity specification (OSEL) may not be changed while the PPG is operating.
  If any of these values has been changed while the PPG was operating, disable the operation of the PPG before reloading the register.

- Whenever writing a value to GCN2, be sure to write “0” to any undefined part of the upper 4 bits. If “1” is written, disable the operation of the PPG before reloading the register.

- If any value outside the specified range (0110 - 1111) is set in Activation Trigger Specification bits (TSEL0[3:0]), (TSEL1[3:0]), (TSEL2[3:0]), (TSEL3[3:0]) has been set, disable the operation of the PPG and then write the specified value to let the register return to normal.

- If the Timer Operation Enable bit (PCN.CNTE) is set to “0” to disable PPGn while it is operating, the PPG stops and sets its output value to the initial value (“1” if (OSEL “1“) else “0“). The PPG timer is latched until the operation of the timer is enabled by setting the Timer Operation enable bit (PCN.CNTE) to “1”.
- If (PCN.CNTE) is set to “1“ to enable the PPG, it restarts (PPG timer is set to initial value).

29. A/D Converter

This chapter describes the ADC unit of the MB88F333.

29.1. Overview

The A/D converter converts analog input voltages into digital values and provides the following features.

Features of A/D converter: Conversion time: minimum 3µs per channel.
RC type successive approximation conversion with sample & hold circuit
10-bit or 8-bit resolution
Program section analog input from 9 channels
Single conversion mode: conversion of one selected channel
Scan conversion mode: continuous conversion of multiple channels, programmable for up to 9 channels

Single conversion mode:	Convert the specified channel only once.
Continuous mode:	Repeatedly convert the specified channels.
Stop mode:	Convert one channel then temporarily halt until the next activation. (enables the synchronization of the conversion start timing).

29.1.1. Input impedance

The sampling circuit of the A/D converter is schematically shown below:

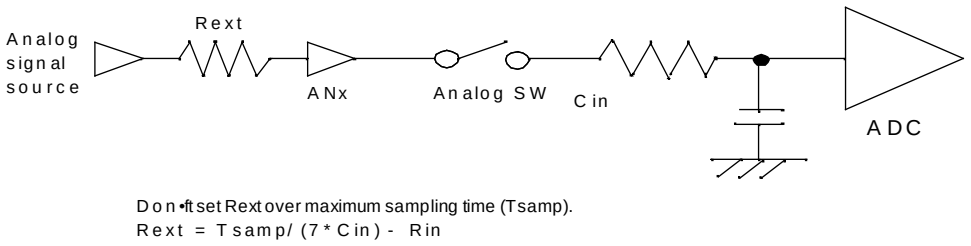


Figure 30-1 Input Impedance

29.2. Block Diagram of the A/D Converter

Following figure shows a block diagram of the A/D converter.

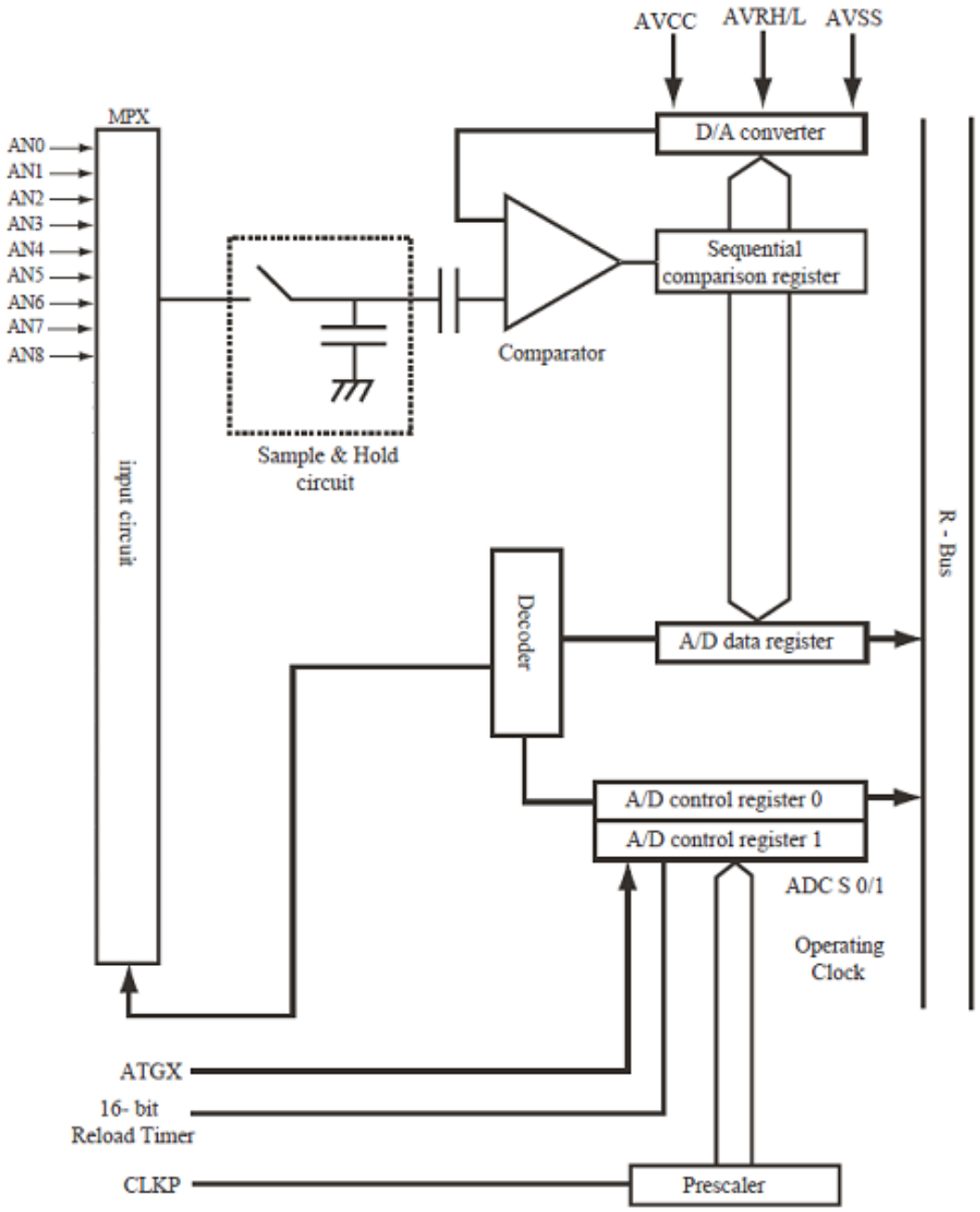


Figure 30-2 Block diagram of A/D converter

29.3. Registers of the A/D Converter

The A/D converter has the following registers.

- A/D enable register (ADER)
- A/D control status register (ADCS)
- Data register (ADCR1,ADCR0)
- Sampling timer setting register (ADCT)
- A/D channel setting register (ADSCH,ADSEH)

29.3.1. Register list

- ADERH: Address 0x01A0 (Access: Word, Half-word, Byte)

15	14	13	12	11	10	9	8	Bit
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	Bit
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

- ADERL: Address 0x01A2 (Access: Word, Half-word, Byte)

15	14	13	12	11	10	9	8	Bit
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	ADE8	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	Bit
ADE7	ADE6	ADE5	ADE4	ADE3	ADE2	ADE1	ADE0	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

- ADCS1: Address 0x01A4 (Access: Half-word, Byte)

15	14	13	12	11	10	9	8	Bit
BUSY	INT	INTE	reserved	STS1	STS0	STRT	reserved	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

- ADCS0: Address 0x01A5 (Access: Half-word, Byte)

7	6	5	4	3	2	1	0	Bit
MD1	MD0	S10	ACH4	ACH3	ACH2	ACH1	ACH0	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R	R	R	R	R	Attribute

- ADCR1: Address 0x01A6 (Access: Word, Half-word, Byte)

15	14	13	12	11	10	9	8	Bit
-	-	-	-	-	-	D9	D8	
0	0	0	0	0	0	X	X	Initial value
R0	R0	R0	R0	R0	R0	R	R	Attribute

- ADCR0: Address 0x01A7 (Access: Word, Half-word, Byte)

7	6	5	4	3	2	1	0	Bit
D7	D6	D5	D4	D3	D2	D1	D0	
X	X	X	X	X	X	X	X	Initial value
R	R	R	R	R	R	R	R	Attribute

- ADCT1: Address 0x01A8 (Access: Word, Half-word, Byte)

15	14	13	12	11	10	9	8	Bit
CT5	CT4	CT3	CT2	CT1	CT0	ST9	ST8	
0	0	0	1	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

- ADCT0: Address 0x01A9 (Access: Word, Half-word, Byte)

7	6	5	4	3	2	1	0	Bit
ST7	ST6	ST5	ST4	ST3	ST2	ST1	ST0	
0	0	1	0	1	1	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

- ADSCH (ADC0): Address 0x01AA (Access: Word, Half-word, Byte)

15	14	13	12	11	10	9	8	Bit
-	-	-	ANS4	ANS3	ANS2	ANS1	ANS0	
-	-	-	0	0	0	0	0	Initial value
RX, W0	RX, W0	RX, W0	R/W	R/W	R/W	R/W	R/W	Attribute

- ADECH (ADC0): Address 0x01AB (Access: Word, Half-word, Byte)

7	6	5	4	3	2	1	0	Bit
-	-	-	ANE4	ANE3	ANE2	ANE1	ANE0	
-	-	-	0	0	0	0	0	Initial value
RX, W0	RX, W0	RX, W0	R/W	R/W	R/W	R/W	R/W	Attribute

29.3.2. A/D Enable Register (ADER)

While a pin is being used as an analog input, the corresponding bit in the ADER register must be set to 1.

29.3.2.1. A/D enable register (ADER)

- ADERH (ADC0): Address 0x01A0 (Access: Word, Half-word, Byte)

15	14	13	12	11	10	9	8	Bit
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	Bit
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

- ADERL (ADC0): Address 0x01A2 (Access: Word, Half-word, Byte)

15	14	13	12	11	10	9	8	Bit
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	ADE8	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

7	6	5	4	3	2	1	0	Bit
ADE7	ADE6	ADE5	ADE4	ADE3	ADE2	ADE1	ADE0	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

[ADE8:0]: A/D Input Enable

ADEn	Function
0	Disable [Initial value]
1	Analog input

Reset clears to 0.
Be sure to set the start channel and end channel to 1.

29.3.3. A/D Control Status Register (ADCS)

The A/D control status register controls and shows the status of the A/D converter. Do not overwrite the ADCS0 register during A/D conversion operation.

29.3.3.1. A/D control status register 1 (ADCS1)

- ADCS1 (ADC0): Address 0x01A4 (Access: Half-word, Byte)

15	14	13	12	11	10	9	8	Bit
BUSY	INT	INTE	reserved	STS1	STS0	STRT	reserved	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

[bit 15] BUSY (busy flag and stop)

BUSY	Function
Reading	A/D converter operation indication bit. Set on activation of A/D conversion and cleared on completion.
Writing	Writing "0" to this bit during A/D conversion forcibly terminates conversion. Use to forcibly terminate in continuous and stop modes.

Cleared on the completion of an A/D conversion in single conversion mode.
In continuous and stop mode, the flag is not cleared until conversion is terminated by writing "0".
Initialized to "0" by a reset.
Do not execute manual termination and software activation (BUSY="0" and STRT="1") at the same time.

[bit 14] INT (interrupt)

This bit is set when the conversion data is stored in ADCR.
If bit 13 (INTE) is "1" when this bit is set, an interrupt request is generated.
Only clear this bit by writing "0" when A/D conversion is halted.
Initialized to "0" by a reset.

[bit 13] INTE (Interrupt enable)

This bit is enables or disables the conversion completion interrupt.

INTE	Function
0	Disable interrupt [Initial value]
1	Enable interrupt

Cleared by a reset.

[bit 12] reserved bit

Always write "0" to this bit.

[bit 11, 10] STS1, STS0 (Start source select)

These bits initialized "00" by reset.
These bits select the A/D activation source.

STS1	STS0	Function
0	0	Software activation [Initial value]
0	1	External trigger pin activation and software activation
1	0	Software activation
1	1	External trigger pin activation and software activation

In multiple-activation modes, the first factor to occur starts A/D conversion.
The activation source changes immediately on writing to the register. Therefore care is required when switching activation mode during A/D operation.
The A/D converter detects falling edges on the external trigger pin. When external trigger level is "L" and if these bits are changed to external trigger activation mode, A/D converting may starts.

[bit 9] STRT (Start)

Writing "1" to this bit starts A/D conversion (software activation).
Write "1" again to restart conversion.
Initialized to "0" by a reset.
In continuous and stop mode, restarting does not occur. Check the BUSY bit (BUSY=0) before writing "1". (Activate conversion after clearing).
Do not execute forced manual termination and software activation (BUSY="0" and STRT="1") at the same time.

[bit 8] reserved bit

Always write "0" to this bit.

29.3.3.2. A/D control status register 0 (ADCS0)

- ADCS0 (ADC0): Address 0x01A5 (Access: Half-word, Byte)

7	6	5	4	3	2	1	0	Bit
MD1	MD0	S10	ACH4	ACH3	ACH2	ACH1	ACH0	
0	0	0	0	0	0	0	0	Initial value
R/W	R/W	R/W	R	R	R	R	R	Attribute

[bit 7, 6] MD1, MD0 (A/D converter mode set)

These bits select the operation mode.

MD1	MD0	Operating mode
0	0	Single mode ; all restarts conversion during operation enabled
0	1	Single mode ; restarts conversion during operation disabled
1	0	Continuous mode ; restarts conversion during operation disabled
1	1	Stop mode ; restarts conversion during operation disabled

Single mode: Continuous A/D conversion from selected channel(s) ANS4 to ANS0 to selected channel(s) ANE4 to ANE0 with a pause after every conversion cycle.

Continuous mode: Repeated A/D conversion cycles from selected channels ANS4 to ANS0 to selected channels ANE4 to ANE0.

Stop mode: A/D conversion for each channel from selected ANS4 to ANS0 to selected channels ANE4 to ANE0, followed by a pause. Restart is determined by the occurrence of a start source.

When A/D conversion is started in continuous mode or stop mode, conversion operation continues until stopped by the BUSY bit.
Conversion is stopped by writing "0" to the BUSY bit.
On activation after forcibly being stopped, conversion starts from the channel selected from ANS4 to ANS0.
All restarts are disabled for external trigger and software start sources in single, continuous and stop modes.

[bit 5] S10

This bit defines the resolution of A/D conversion. If this bit set to "0", the resolution is 10-bit. Otherwise resolution is 8-bit and the conversion result is stored in ADCR0.
Initialized to "0" by a reset.

[bit 4 to 0] ACH4-0 (Select analog conversion channel)

These bits show the current conversion channel.

ACH4	ACH3	ACH2	ACH1	ACH0	Conversion channel
0	0	0	0	0	AN0
0	0	0	0	1	AN1
0	0	0	1	0	AN2
0	0	0	1	1	AN3
0	0	1	0	0	AN4
0	0	1	0	1	AN5
0	0	1	1	0	AN6
0	0	1	1	1	AN7
0	1	0	0	0	AN8

ACH	Function
Reading	During active A/D conversion (BUSY='1') the number of the channel that is actively converting is shown. If conversion is manually halted, the number of the channel that is stopped is shown. If a conversion is not currently active (BUSY='0'), the number of the channel of the last conversion is shown. Hint for applications using remote handler event message ID 76 (ADC interrupt): The payload of this event contains (amongst other items) the register fields ACH, BUSY and ADCR. If BUSY=0, then the result value ADCR belongs to channel number ACH. If BUSY = 1 then the result value ADCR belongs to the previously conversion channel. Please pay additional attention if you use masked channels (Register ADER) to identify the previous conversion channel.
Writing	No effect to these bits.

Initialized to "00000" by reset.

29.3.4. Data Register (ADCR1, ADCR0)

These registers store the conversion results of the A/D converter. ADCR0 stores the lower 8-bits. ADCR1 stores the upper 2-bits. The register values are updated on the completion of each conversion. The registers normally store the results of the previous conversion.

29.3.4.1. Data register (ADCR1, ADCR0)

- ADCR1 (ADC0): Address 0x01A6 (Access: Word, Half-word, Byte)

15	14	13	12	11	10	9	8	Bit
-	-	-	-	-	-	D9	D8	
-	-	-	-	-	-	X	X	Initial value
R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R0, W0	R	R	Attribute

- ADCR0 (ADC0): Address 0x01A7 (Access: Word, Half-word, Byte)

7	6	5	4	3	2	1	0	Bit
D7	D6	D5	D4	D3	D2	D1	D0	
X	X	X	X	X	X	X	X	Initial value
R	R	R	R	R	R	R	R	Attribute

Bits 15 to 10 of ADCR1 read "0".

The A/D converter has a conversion data protection function. See the "Operation" section for further information.

29.3.5. Sampling Timer Setting Register (ADCT)

The ADCT register controls the sampling time and comparison time of an analog input. This register sets the A/D conversion time. Do not update the value of this register during A/D conversion operation.

29.3.5.1. Sampling timer setting register (ADCT)

- ADCT1 (ADC0): Address 0x01A8 (Access: Word, Half-word, Byte)

15	14	13	12	11	10	9	8	Bit
CT5	CT4	CT3	CT2	CT1	CT0	ST9	ST8	
0	0	0	1	0	0	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

- ADCT0 (ADC0): Address 0x01A9 (Access: Word, Half-word, Byte)

7	6	5	4	3	2	1	0	Bit
ST7	ST6	ST5	ST4	ST3	ST2	ST1	ST0	
0	0	1	0	1	1	0	0	Initial value
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Attribute

[bit 15 to 10] CT5-0 (A/D comparison time set)

These bits specify the clock division of comparison time.
Setting "000001" means one division (=CLKP).
Do not set these bits to "000000".
Reset initializes these bits to "000100"
Comparison time = CT value * CLKP cycle * 10 + (4 * CLKP)

Remarks: Do not set a comparison time over 500 µs.

[bit 9 to 0] ST9-0 (Analog input sampling time set)

These bits specify the sampling time for analog input.
Reset initializes these bits to "0000101100"

Sampling time = ST value * CLKP cycle

Remarks: Do not set a sampling time below 1.2 µs when AVCC is below 4.5 V.

The necessary sampling time and ST value are calculated as follows:
The necessary sampling time (Tsamp) = (Rext + Rin) * Cin * 7
ST9 to ST0 = Tsamp / CLKP cycle

ST has to be set so that the sampling time is over Tsamp.
e.g. CLKP = 41.66MHz, AVCC >= 4.5V, Rext = 200Kohm
Tsamp = (200Kohm +2.6Kohm) *8.5pF * 7 = 12.054 [µs]
ST = 12.054[µs] / 24[ns] = 502.25
ST has to be set over 503_D (111110111_B).

Tsamp is decided by Rext. Thus the conversion time should be considered together with Rext.

29.3.6. A/D Channel Setting Register (ADSCH, ADECH)

These registers specify the channels for the A/D converter to convert. Do not update these registers while the A/D conversion is operating.

29.3.6.1. A/D channel setting register (ADSCH, ADECH)

- ADSCH (ADC0): Address 0x01AA (Access: Word, Half-word, Byte)

15	14	13	12	11	10	9	8	Bit
-	-	-	ANS4	ANS3	ANS2	ANS1	ANS0	
-	-	-	0	0	0	0	0	Initial value
RX, W0	RX, W0	RX, W0	R/W	R/W	R/W	R/W	R/W	Attribute

- ADECH (ADC0): Address 0x01AB (Access: Word, Half-word, Byte)

7	6	5	4	3	2	1	0	Bit
-	-	-	ANE4	ANE3	ANE2	ANE1	ANE0	
-	-	-	0	0	0	0	0	Initial value
RX, W0	RX, W0	RX, W0	R/W	R/W	R/W	R/W	R/W	Attribute

These bits set the start and end channel for A/D converter.
Setting of ANE4 to ANE0 the same channel as in ANS4 to ANS0 specifies conversion for that channel only. (Single conversion)
In continuous or stop mode, conversion is performed up to the channel specified by ANE4 to ANE0. Conversion then starts again from the start channel specified by ANS4 to ANS0.
If ANS > ANE, conversion starts with the channel specified by ANS, continuous up to channel AN11, starts again from AN0, and ends with the channel specified by ANE.
Initialized to ANS="00000", ANE="00000" by a reset.

E.g. Channel Setting ANS=AN3, ANE=AN8, single conversion mode
Operation: Conversion AN3 -> AN4 -> AN5 -> AN6 -> AN7 -> AN8 end

[bit 12 to 8] ANS4-0 (Analog start channel set)
[bit 4 to 0] ANE4-0 (Analog end channel set)

ANS4 ANE4	ANS3 ANE3	ANS2 ANE2	ANS1 ANE1	ANS0 ANE0	Start / End Channel
0	0	0	0	0	AN0
0	0	0	0	1	AN1
0	0	0	1	0	AN2
0	0	0	1	1	AN3
0	0	1	0	0	AN4
0	0	1	0	1	AN5
0	0	1	1	0	AN6
0	0	1	1	1	AN7
0	1	0	0	0	AN8
The settings other than the above-mentioned are prohibited.					

29.4. Operation of the A/D Converter

The A/D converter operates using a successive approximation method with a 10-bit or 8-bit resolution. As only one 16-bit register(Data register) is provided to store conversion results, the data registers (ADCR0 and ADCR1) are updated each time a conversion is completed. The following describes the operation modes.

Single Mode

In single conversion mode, the analog input signals selected by the ANS bits and ANE bits are converted in succession until the completion of conversion on the end channel determined by the ANE bits. A/D conversion then ends. If the start and end channel are the same (ANS=ANE), only a single channel conversion is performed.

Example:
ANS=00000b, ANE=00011b
Start -> AN0 -> AN1 -> AN2 -> AN3 -> End

ANS=00010b, ANE=00010b
Start -> AN2 -> End

Continuous Mode

In continuous mode, the analog input signals selected by the ANS bits and ANE bits are converted in succession until the completion of conversion on the end channel determined by the ANE bits is attained. Then the converter returns to the ANS channel for analog input and repeats the process continuously. If the start and end channels are the same (ANS=ANE), conversion is performed continuously for that channel.

Example:
ANS=00000b, ANE=00011b
Start -> AN0 -> AN1 -> AN2 -> AN3 -> AN0 ... -> repeat

ANS=00010b, ANE=00010b
Start -> AN2 -> AN2 -> AN2 ... -> repeat

In continuous mode, conversion is repeated until '0' is written to the BUSY bit (writing '0' to the BUSY bit forcibly stops the conversion operation). Note that a forced termination operation halts the current conversion in mid-conversion (if the operation is forcibly terminated, the value in the register is the result of the most recently completed conversion).

Stop Mode

In stop mode, the analog input signal selected by the ANS bits and ANE bits are converted in succession, but the conversion operation pauses for each channel. The pause is released by applying another start signal.
On the completion of conversion on the end channel as determined by the ANE bits, the converter returns to the ANS channel for analog input signal and repeats the conversion process continuously. If the start and end channel are the same (ANS=ANE), only a single channel conversion is performed.

Example:
ANS=00000b, ANE=00011b
Start -> AN0 -> stop -> start -> AN1 -> stop -> start -> AN2 -> stop -> start -> AN3 -> stop -> start -> AN0 ... -> repeat

ANS=00010b, ANE=00010b
Start -> AN2 -> stop -> start -> AN2 -> stop -> start -> AN2 ... -> repeat

In stop mode, the startup source is only the source determined by the STS1, STS0 bits. This mode enables the synchronization of the conversion start signal.

29.5. Settings

Table 30-1 Settings needed to use A/D - Single-Shot Conversion Mode

Setting	Setting Registers	Setting Procedure*
Mode selection (Single-shot conversion)	A/D control (ADCS)	See 30.6.1
Bit length selection		See 30.6.2
Channel selection		
Conversion time setting	sampling timer setting (ADCT)	See 30.6.3
A/D activation trigger selection	A/D control (ADCS)	See 30.6.4
A/D activation trigger generation Software trigger -> Software trigger bit setting		See 30.6.5
External trigger -> Inputs a trigger to the ADTG(0, 1) pin.	External input	
Conversion end flag check	A/D control (ADCS)	See 30.6.6
Conversion value read	Data register (ADCR)	See 30.6.7

*: For the setting procedure, refer to the section indicated by the number.

Table 30-2 Settings needed to use A/D - Scan Conversion Mode

Setting	Setting Registers	Setting Procedure*
Mode selection (Scan conversion)	A/D control (ADCS)	See 30.6.1
Bit length selection		See 30.6.2
Starting channel selection		
Conversion time setting	sampling timer setting (ADCT)	See 30.6.3
A/D activation trigger selection	A/D control (ADCS)	See 30.6.4
A/D activation trigger generation Software trigger -> Software trigger bit setting		See 30.6.5
External trigger -> Inputs a trigger to the ATGX pin.	External input	
Conversion end flag check	A/D control (ADCS)	See 30.6.6
Conversion value read	Data register (ADCR)	See 30.6.7

*: For the setting procedure, please refer to the section stated.

Table 30-3 Forcing A/D operations to Stop

Setting	Setting Registers	Setting Procedure*
Forced stop	A/D control (ADCS)	See 30.6.8

*: For the setting procedure, refer to the section indicated by the number.

Table 30-4 Items needed to enable A/D Interrupts

Setting	Setting Registers	Setting Procedure*
A/D interrupt cause selection (A/D conversion end)	A/D control registers (ADCS)	See 30.6.9
A/D interrupt setting Clear interrupt requests. Enable interrupt requests.		See 30.6.10.

*: For the setting procedure, refer to the section indicated by the number.

29.6. Q & A

29.6.1. What conversion modes are available and how are they selected?

Two modes of conversion are available:

- Single-shot conversion mode, in which the conversion takes place only once.
- Scan conversion mode, in which a specified sequence of channels are converted.

Mode selection is made using the conversion mode selection bits (ADCS.MD[1:0]).

MD1	MD0	Operating mode
0	0	Single mode ; all restarts conversion during operation enabled
0	1	Single mode ; restarts conversion during operation disabled
1	0	Continuous mode ; restarts conversion during operation disabled
1	1	Stop mode ; restarts conversion during operation disabled

29.6.2. How do I specify a bit length?

Configure the conversion result storage bit length setting (ADCS.S10).

Operation Mode	Conversion Result Storage Bit Length (S10)
10bit : Subordinate position 8BIT is stored in the ADCR0 register as for the conversion result, and high rank 2BIT is stored in the ADCR1 register.	Set “0”.
8bit : The conversion result is stored in the ADCR0 register	Set “1”.

29.6.3. How do I set a conversion time?

Use Sampling Timer Setting registers ADCT to set a conversion time.

[bit 15 to 10] CT5-0 (A/D comparison time set)

These bits specify clock division of comparison time.
Setting "000001" means one division (=CLKP).
Do not set these bits to "000000".
Reset initializes these bits to "000100"
Comparison time = CT value * CLKP cycle * 10 + (4 * CLKP)
Remarks : Do not set the comparison time to over 500 µs.

[bit 9 to 0] ST9-0 (Analog input sampling time set)

These bits specify the sampling time of the analog input.
Reset initializes these bits to "0000101100"

Sampling time = ST value * CLKP cycle
Remarks : Do not set the sampling time to below 1.2 µs when AVCC is below 4.5 V.

Necessary sampling time and ST value are calculated as follows:
Necessary sampling time (Tsamp) = (Rext + Rin) * Cin * 7
ST9 to ST0 = Tsamp / CLKP cycle

ST has to be set so that the sampling time is over Tsamp.
e.g. CLKP = 41.66MHz, AVCC >= 4.5V, Rext = 200Kohm
Tsamp = (200Kohm + 2.6Kohm) * 8.5pF * 7 = 12.054 [µs]
ST = 12.054 [µs] / 24 [ns] = 502.25
ST has to be set over 503_D (111110111_B).

Tsamp is decided by Rext. Thus conversion time should be considered together with Rext.

29.6.4. To select how to activate the A/D converter

There are two types of activation triggers:

- Software trigger
- External trigger input falling signal

To set an activation trigger, use Activation Trigger Selection bits (ADCS.STS[1: 0]).

A/D Activation Trigger	Activation Trigger Selection bit (STS[1: 0])
To specify a software trigger	Set “00”.
To specify an external trigger/software trigger	Set “01”.
To specify a software trigger	Set “10”.
To specify an external trigger/ software trigger	Set “11”.

The A/D converter is activated on the first instance of any one of these causes selected.

29.6.5. To activate the A/D converter

- Generating a software trigger

A software trigger is generated using A/D Conversion Software Trigger bits (ADCS.STRT).

Operation	A/D Conversion Software Trigger Bit (STRT)
To generate a software trigger	Write “1”.

Activating A/D converter with an external trigger

Use external trigger input pin ATGX to generate an external trigger.

29.6.6. To verify the end of a conversion

There are two ways to verify the end of a conversion, as follows:

- Checking the A/D Conversion End Interrupt Request bits (ADCS.INT)

(INT)	Description
If the read value is “0”	No A/D conversion end interrupt request
If the read value is “1”	A/D conversion end interrupt request

- Checking the Operation Verification bits (ADCS.BUSY)

(BUSY)	Setting
If the read value is “0”	A/D conversion end (stop)
If the read value is “1”	A/D conversion in progress

29.6.7. How do I read a conversion value?

The conversion value can be read from Data register ADCR.

29.6.8. How do I force an A/D conversion operation to stop?

Use the Forced Stop bits (ADCS.BUSY)

Operation	Forced Stop Bit (BUSY)
To force an A/D conversion operation to a stop	Write “0”.

The operation of the A/D is unaffected by writing “1” to the Forced Stop bit (BUSY).

29.6.9. What interrupts are available?

A/D Conversion End interrupt only. No interrupt cause selection bit is available.

29.6.10. How do I enable, disable, clear interrupts?

Interrupts are enabled using the Interrupt Request Enable bits (ADCS.INTE).

	Interrupt Request Enable Bit (INTE)
To disable interrupt requests	Set “0”.
To enable interrupt requests	Set “1”.

Interrupt request are cleared using the Interrupt Request bits (ADCS.INT).

	Interrupt Request Bit (INT)
To clear interrupt requests	Write “0” or activate A/D. (See “To activate the A/D converter (Page No.30-16)”).

(See “Caution (Page No.30-18)”).

29.7. Caution

Caution on using the A/D converter are summarized as follows:

Power-on sequence
Be sure to turn on the MCU power (VDD*) before turning on the power to the A/D converter (AD_AVCC, AD_AVRH) and applying a voltage to the analog input.

Input impedance of the analog input pin
The A/D converter has a built-in sample hold circuit to receive the voltage present on the analog input pin in the sample hold capacitor after the activation of an A/D conversion. Therefore, if the analog input external circuit has a high output impedance, it is possible that the analog input voltage fails to stabilize within the sampling cycle. For this reason, keep the output impedance of the external circuit sufficiently low.
If the output impedance of the external circuit cannot be kept sufficiently low, lengthen the sampling time to a maximum.

As AVRH-AV_{SS} decreases, the error grows in proportion.

29.7.1. Definitions of A/D Converter Terms

- Resolution

Analog change identifiable to an A/D converter.

- Linearity error

Deviation between the straight line connecting zero transition point
(00 0000 0000 <- -> 00 0000 0001) and full-scale transition point
(11 1111 1110 <- -> 11 1111 1111) from actual conversion characteristics

- Differential linearity error

Deviation of the input voltage required for changing the output by one LSB, from its ideal value

$$1LSB = \frac{VFST - VOT}{1022} [V]$$

VOT: Voltage at which digital output transit from (000)H to (001)H
VFST: Voltage at which digital output transit from (3FE)H to (3FF)H

$$\text{Digital output N Linearity error} = \frac{VNT - \{1LSB \times (N-1) + VOT\}}{1LSB} [LSB]$$

$$\text{Digital output N Differential linearity error} = \frac{V(N+1)T - VNT}{1LSB} - 1 [LSB]$$

V_{NT}: Voltage at which digital output transit from (N+1) to N

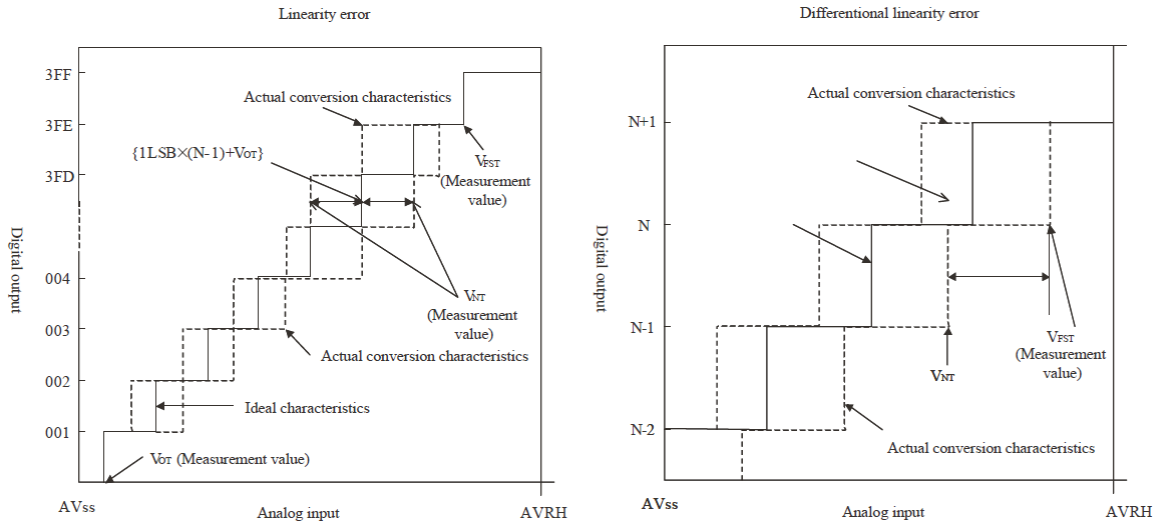


Figure 30-3 Differential linearity error of A/D converter

- Overall error

The difference between an actual value and a theoretical value, containing a zero transition error/full transition error/linearity error

$$1LSB'(\text{Ideal value}) = \frac{AVRH - AVSS}{1024} \quad [V]$$

$$VOT'\ (\text{Ideal value}) = AVSS + 0.5LSB' \quad [V]$$

$$VFST'\ (\text{Ideal value}) = AVRH - 1.5LSB' \quad [V]$$

$$\text{Overall error of digital output } N = \frac{VNT - \{1LSB' \times (N - 1) + 0.5LSB'\}}{1LSB'}$$

VNT: Voltage at which digital output transit from (N+1) to N

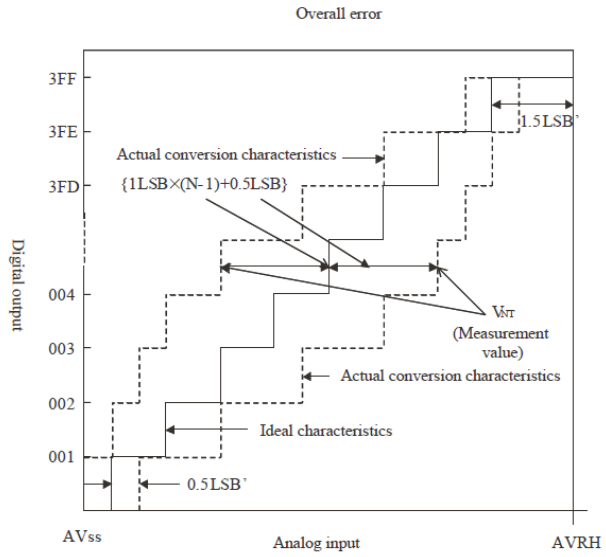


Figure 30-4 Overall error of A/D converter

30. Reload Timer

This chapter describes the Reload Timer of the MB88F333.

30.1. Overview

The reload timer uses a 16 bit down counter to detect the input signal trigger and perform a countdown. The count length is 16 bits.

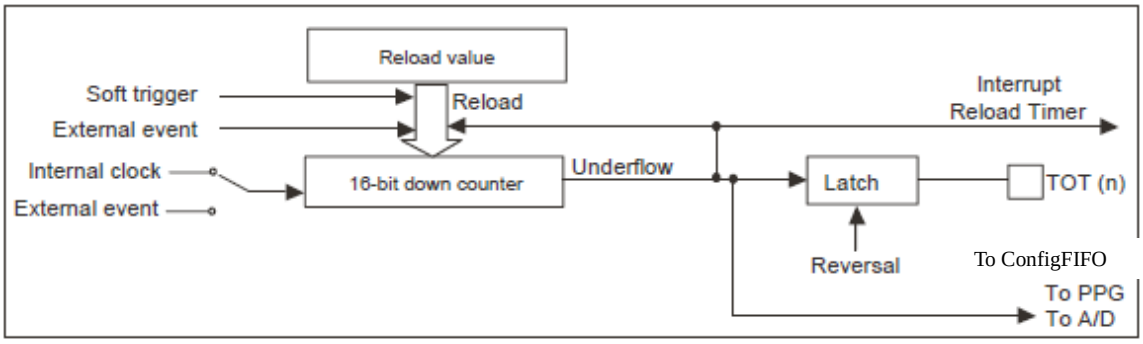


Figure 31-1 Block diagram of the Reload Timer

30.2. Features

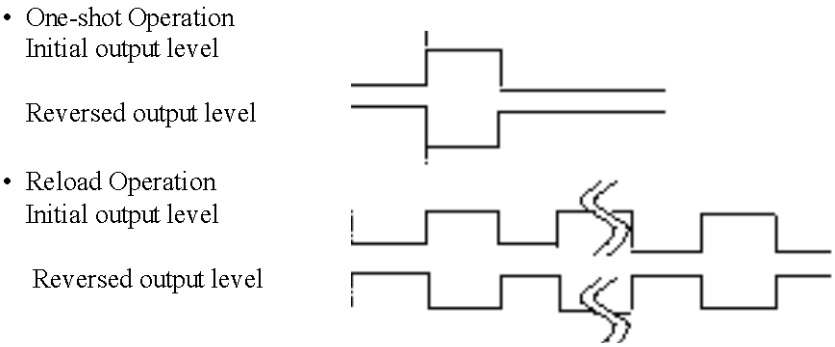


Figure 31-2 Operation: 2 kinds of operation are possible

Format: 16 bit down counter with reload register
Quantity: 16 (Output: 16 channels TOT[0-15])
Clock mode: Select from two modes

- Internal clock mode
  Count clock: CLKP/2, CLKP/8, CLKP/32, CLKP/64, CLKP/128
  Activation triggers (4 types)

- External event clock mode
  Count clock : External event (RLT[4:0] pins)
  Count active edge : Rising/falling/both edges of external event
  Activation trigger : Software trigger
  Cycle : Cycle = count clock x (reload value + 1)

(Example) When count clock = 20.83MHz, reload value = 20829

Cycle = 48ns x (20829+1) = 1.0ms

Count active edge: When in external event mode, choose from 3 types.

- External trigger (rising /falling/both edges)
  Interrupt: Request generated by underflow
  Other 1: Counter stop in software/can be reopened
  Other 2: Control of other peripheral functions possible
- PPG activation trigger source:
  Reload Timer 2: PPG4, PPG5, PPG 6 , PPG7
  Reload Timer 3: PPG4, PPG5, PPG 6 , PPG7
  Reload Timer 4: PPG8, PPG9, PPG 10 , PPG11
  Reload Timer 5: PPG8, PPG9, PPG 10 , PPG11

same as PPG chapter

- A/D converter activation trigger source
  Reload timer 7: A/D
- ConfigFIFO activation trigger source:
  Reload timer 8: ConfigFIFO1
  Reload timer 9: ConfigFIFO2
  Reload timer 10: ConfigFIFO3
  Reload timer 11: ConfigFIFO4
  Reload timer 12: ConfigFIFO5
  Reload timer 13: ConfigFIFO6
- Other activation trigger source:
  Reload timer 6: Can be used by the RH as an event trigger
  Reload timer 14: Can be used by the RH as an event trigger
  Reload timer 15: Can be used by the RH as an event trigger

30.3. Configuration

Reload Timer 0 (Internal clock count)

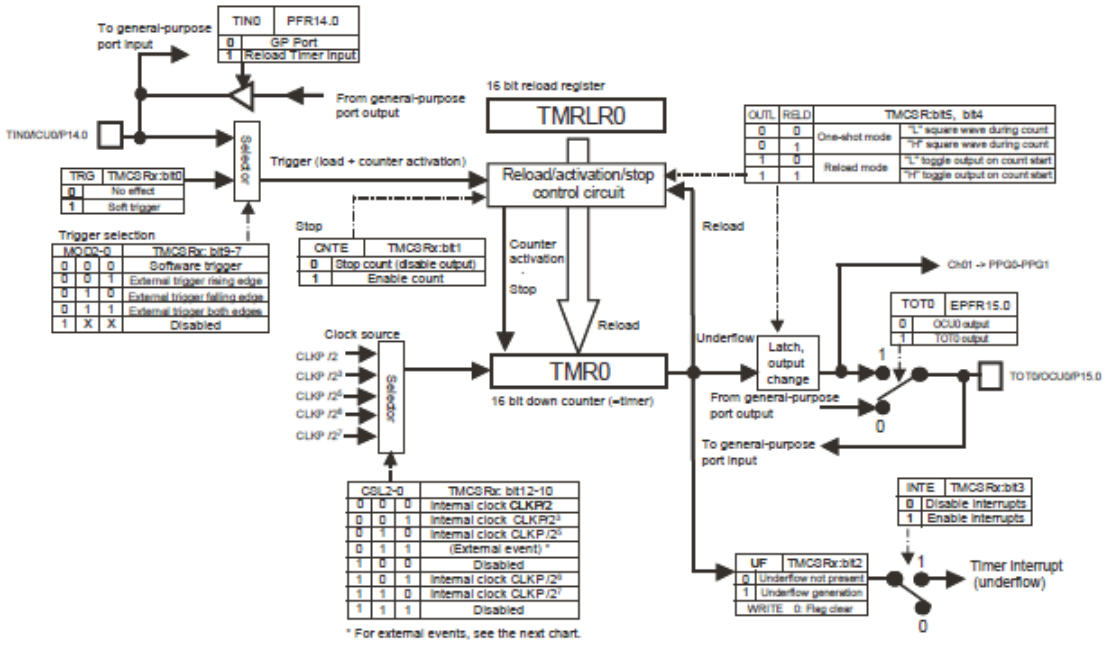


Figure 31-3 Configuration Diagram of the Reload Timer (Internal Clock Count)

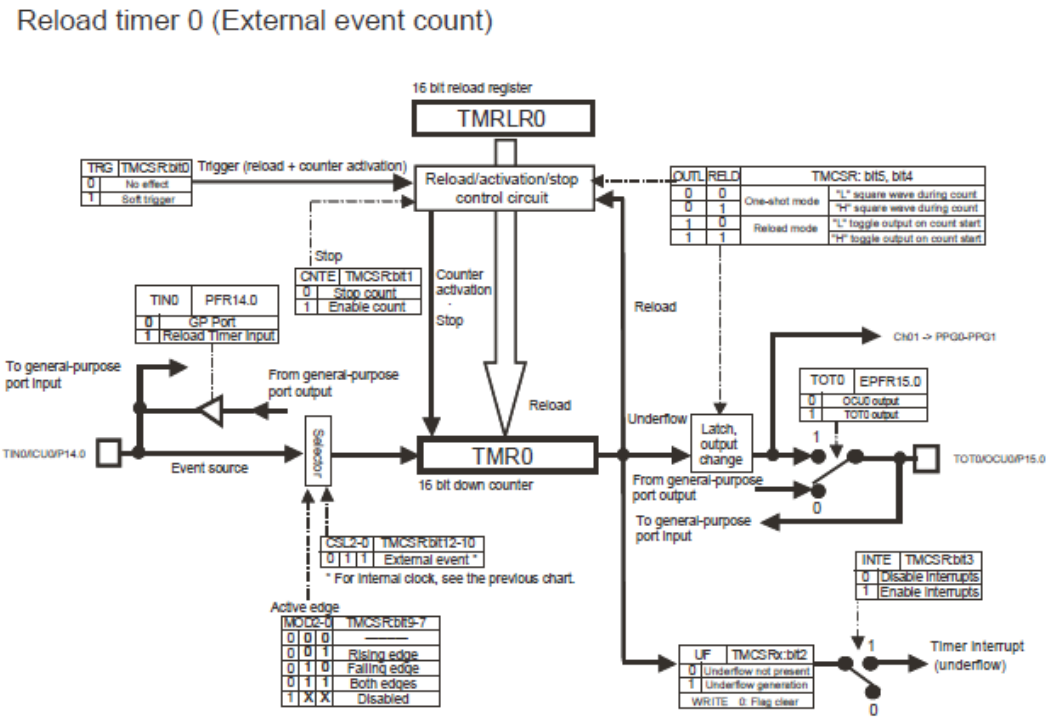


Figure 31-4 Configuration Diagram of the Reload Timer (External event count)

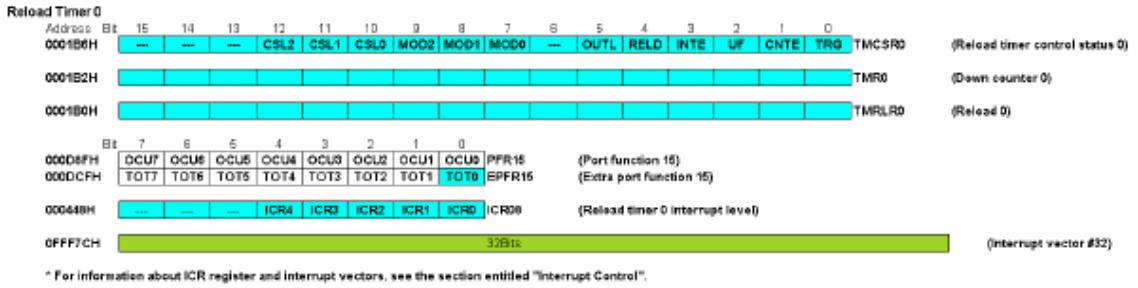


Figure 31-5 Register List

30.4. Registers

30.4.1. TMCSR: Reload Timer Control Status Register

The control status register controls the operation mode of the reload timer and interrupts.

- TMCSR0 (Reload timer 0): Address: 0x001B6 (Access: Byte, Half-word)
- TMCSR1 (Reload timer 1): Address: 0x001BE (Access: Byte, Half-word)
- TMCSR2 (Reload timer 2): Address: 0x001C6 (Access: Byte, Half-word)
- TMCSR3 (Reload timer 3): Address: 0x001CE (Access: Byte, Half-word)
- TMCSR4 (Reload timer 4): Address: 0x001D6 (Access: Byte, Half-word)
- TMCSR5 (Reload timer 5): Address: 0x001DE (Access: Byte, Half-word)
- TMCSR6 (Reload timer 6): Address: 0x001E6 (Access: Byte, Half-word)
- TMCSR7 (Reload timer 7): Address: 0x001EE (Access: Byte, Half-word)
- TMCSR8 (Reload timer 8): Address: 0x00596 (Access: Byte, Half-word)
- TMCSR9 (Reload timer 9): Address: 0x0059E (Access: Byte, Half-word)
- TMCSR10 (Reload timer 10): Address: 0x005A6 (Access: Byte, Half-word)
- TMCSR11 (Reload timer 11): Address: 0x005AE (Access: Byte, Half-word)
- TMCSR12 (Reload timer 12): Address: 0x005B6 (Access: Byte, Half-word)
- TMCSR13 (Reload timer 13): Address: 0x005BE (Access: Byte, Half-word)
- TMCSR14 (Reload timer 14): Address: 0x005C6 (Access: Byte, Half-word)
- TMCSR15 (Reload timer 15): Address: 0x005CE (Access: Byte, Half-word)

15	14	13	12	11	10	9	8	bit
			CSL2	CSL1	CSL0	MOD2	MOD1	
-	-	-	0	0	0	0	0	Initial Value
RX/WX	RX/WX	RX/WX	R/WX	R/W	R/W	R/W0	R/W	Attribute
x	x	x	x	x	x	x	x	Rewrite during operation

7	6	5	4	3	2	1	0	bit
MOD0		OULT	RELD	INTE	UF	CNTE	TRG	
0	-	0	0	0	0	0	0	Initial Value
R/W	RX/WX	R/W	R/W	R/W	R(RM1),W	R/W	R0/W	Attribute
x	x	x	x	x	O	O	O	Rewrite during operation

(O: can be rewritten, x: cannot be rewritten)

- bit15-14: Undefined

Writing has no effect on the operation. The read value is “0”.

- bit13: Undefined (reload timer 0 - reload timer 2)

Always write “0”. The read value is “0”.

- bit12-10: Count clock selection CLKP: peripheral clock

CSL2	CSL1	CSL0	Count clock	Remarks
0	0	0	Internal clock CLKP/2	
0	0	1	Internal clock CLKP/8	
0	1	0	Internal clock CLKP/32	
0	1	1	External event (external clock)	Not supported for RLT[7:5]
1	0	1	Internal clock CLKP/64	
1	1	0	Internal clock CLKP/128	

Notes: Depending on whether an internal clock or an external event is selected, the meaning of the operation mode selection bit (MOD[2:0]) changes.

- bit9-7: Operation mode selection

Reload trigger when internal clock is selected

MOD2	MOD1	MOD0	Reload trigger
0	0	0	Software trigger
0	0	1	External trigger (rising edge)
0	1	0	External trigger (falling edge)
0	1	1	External trigger (both edges)

When the selected reload trigger is input, the value of reload register TMRLR is loaded to the down

counter and the count operation is started.

Count trigger when external event is selected

MOD2	MOD1	MOD0	Count trigger
0	0	0	--------
0	0	1	External trigger (rising edge)
0	1	0	External trigger (falling edge)
0	1	1	External trigger (both edges)

Counts an external event using the selected count trigger.

Always set MOD2 to “0”. The read value is the written value.

- bit6: Undefined

Writing has no effect on the operation. The read value is “0”.

- bit5: Output level setting

OUTL	One-shot mode (RELD=“0”)	Reload mode (RELD=“1”)
0	During count “H” square wave	During count start “L” toggle output
1	During count “L” square wave	During count start “H” toggle output

During one-shot mode, a pulse is output during the count and during reload mode the output is toggled.

For output level setting bit “0” and “1” the output level is reversed.

- bit4: Enable reload

RELD	Enable reload
0	One-shot mode (reload disabled)
1	Reload mode (reload enabled)

In reload mode, down counter underflow (0000H -> FFFFH) causes the value set to reload register (TMRLR) to be loaded to the down counter, and the count operation continues.

In one-shot mode, down counter underflow (0000H -> FFFFH) causes the count operation to stop.

- bit3: Enable timer interrupt requests

INTE	Enable timer interrupt requests
0	Disable interrupt requests
1	Enable interrupt requests

When timer interrupt requests are enabled, the timer interrupt request flag (UF) becomes “1” and interrupt requests are generated.

- bit2: Timer interrupt request flag

UF	Timer interrupt request flag	
	When read	When write
0	Underflow not present	Clear interrupt requests
1	Underflow present	No effect

Upon down counter underflow (0000H -> FFFFH) generation, the timer interrupt request flag becomes “1”.

If the interrupt request is enabled (INTE=“1”) an interrupt request is generated.

- bit1: Enable timer count

CNTE	Enable timer count
0	Stop count operation
1	Enable count operation (waiting for activation trigger)

If timer count is enabled, it waits for an activation trigger, and when an activation trigger is generated, the count operation starts. The activation trigger can be a software trigger or an external trigger.

- bit0: Software trigger

TRG	Software trigger
0	No effect. (The read value is “0”.)
1	Start count operation after data load.

If the count operation is enabled (CNTE=“1”) and the software trigger bit is set to “1”, the value of the reload register (TMRLR) is loaded to the down counter and the count operation starts.

If the count operation is not enabled (CNTE=“0”), the software trigger has no effect.

30.4.2. TMR: Timer Register

- TMR0 (Reload timer 0): Address: 01B2H (Access: Half-word)
- TMR1 (Reload timer 1): Address: 01BAH (Access: Half-word)
- TMR2 (Reload timer 2): Address: 01C2H (Access: Half-word)
- TMR3 (Reload timer 3): Address: 01CAH (Access: Half-word)
- TMR4 (Reload timer 4): Address: 01D2H (Access: Half-word)
- TMR5 (Reload timer 5): Address: 01DAH (Access: Half-word)
- TMR6 (Reload timer 6): Address: 01E2H (Access: Half-word)
- TMR7 (Reload timer 7): Address: 01EAH (Access: Half-word)
- TMR8 (Reload timer 8): Address: 0592H (Access: Half-word)
- TMR9 (Reload timer 9): Address: 059AH (Access: Half-word)
- TMR10 (Reload timer 10): Address: 05A2H (Access: Half-word)
- TMR11 (Reload timer 11): Address: 05AAH (Access: Half-word)
- TMR12 (Reload timer 12): Address: 05B2H (Access: Half-word)
- TMR13 (Reload timer 13): Address: 05BAH (Access: Half-word)
- TMR14 (Reload timer 14): Address: 05C2H (Access: Half-word)
- TMR15 (Reload timer 15): Address: 05CAH (Access: Half-word)

15	14	13	12	11	10	9	8	bit
D15	D14	D13	D12	D11	D10	D9	D8	
X	X	X	X	X	X	X	X	Initial Value
R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	Attribute

7	6	5	4	3	2	1	0	bit
D7	D6	D5	D4	D3	D2	D1	D0	
X	X	X	X	X	X	X	X	Initial Value
R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	R/WX	Attribute

The reload timer count value can be read out through the timer register TMR.

Please perform the read out using half-word access.

30.4.3. TMRLR: Reload register

- TMRLR0 (Reload timer 0): Address: 01B0H (Access: Half-word)
- TMRLR1 (Reload timer 1): Address: 01B8H (Access: Half-word)
- TMRLR2 (Reload timer 2): Address: 01C0H (Access: Half-word)
- TMRLR3 (Reload timer 3): Address: 01C8H (Access: Half-word)
- TMRLR4 (Reload timer 4): Address: 01D0H (Access: Half-word)
- TMRLR5 (Reload timer 5): Address: 01D8H (Access: Half-word)
- TMRLR6 (Reload timer 6): Address: 01E0H (Access: Half-word)
- TMRLR7 (Reload timer 7): Address: 01E8H (Access: Half-word)
- TMRLR8 (Reload timer 8): Address: 0590H (Access: Half-word)
- TMRLR9 (Reload timer 9): Address: 0598H (Access: Half-word)
- TMRLR10 (Reload timer 10): Address: 05A0H (Access: Half-word)
- TMRLR11 (Reload timer 11): Address: 05A8H (Access: Half-word)
- TMRLR12 (Reload timer 12): Address: 05B0H (Access: Half-word)
- TMRLR13 (Reload timer 13): Address: 05B8H (Access: Half-word)
- TMRLR14 (Reload timer 14): Address: 05C0H (Access: Half-word)
- TMRLR15 (Reload timer 15): Address: 05C8H (Access: Half-word)

15	14	13	12	11	10	9	8	bit
D15	D14	D13	D12	D11	D10	D9	D8	
X	X	X	X	X	X	X	X	Initial Value
RX/W	RX/W	RX/W	RX/W	RX/W	RX/W	RX/W	RX/W	Attribute

7	6	5	4	3	2	1	0	bit
D7	D6	D5	D4	D3	D2	D1	D0	
X	X	X	X	X	X	X	X	Initial Value
RX/W	RX/W	RX/W	RX/W	RX/W	RX/W	RX/W	RX/W	Attribute

The reload value for the down counter is stored in reload register TMRLR.

Please write using half-word access.

30.5. Operation

30.5.1. Internal Clock/Reload Mode

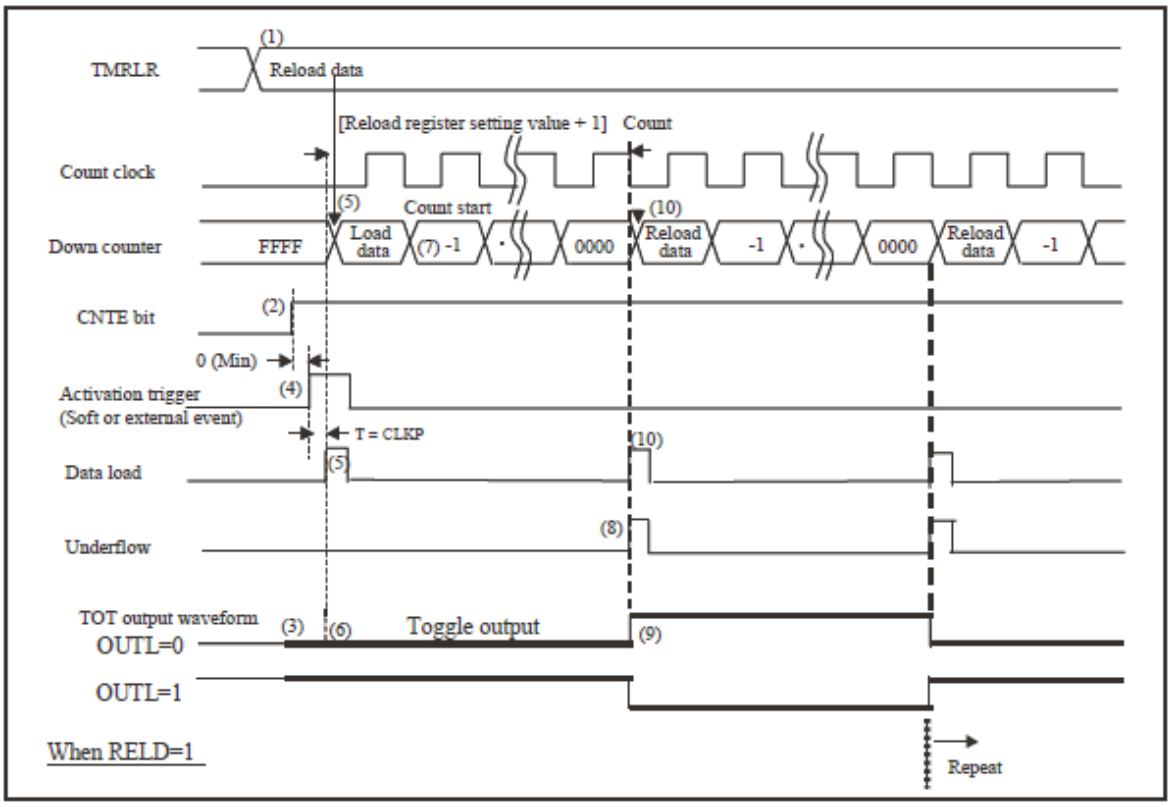


Figure 31-6 Operation of Reload Timer in reload mode (pulse with a 50% duty ratio is output)

(1) Set reload value to reload register
(2) Enable reload timer count operation
(3) TOT pin output
(4) Generate reload trigger (activation): soft trigger or external event trigger
(5) Load reload value
(6) TOT toggle output start
(7) Counter count down (internal clock synchronous)
(8) Generate counter underflow
(9) TOT pin output level reversal (toggle output)
(10) Reload reload value
(11) Repeat steps (7) to (10)
(See “Caution (Page No.31-17)”.)

30.5.2. Internal Clock/One-shot Mode

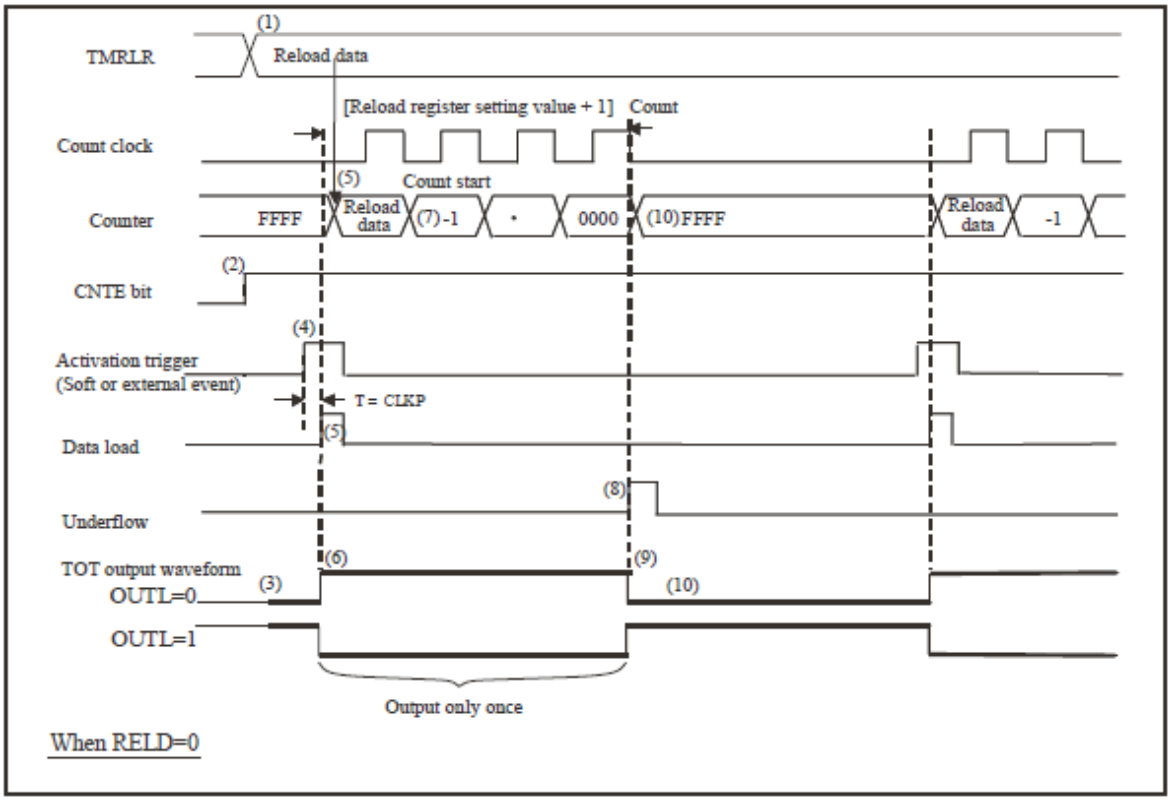


Figure 31-7 Operation of Reload Timer in one-shot mode (one-shot pulse is output)

(1) Set reload value to reload register
(2) Enable reload timer count operation
(3) TOT pin output
(4) Generate reload trigger (activation): soft trigger or external event trigger
(5) Load reload value
(6) Square wave output (during count, “H” output/OUTL=“0”)
(7) Counter count down (internal clock synchronous)
(8) Generate counter underflow
(9) Return TOT pin output level
(10) Count stop, wait for next activation trigger

(See “Caution (Page No.31-17)”.)

30.5.3. External Event Clock Reload Mode

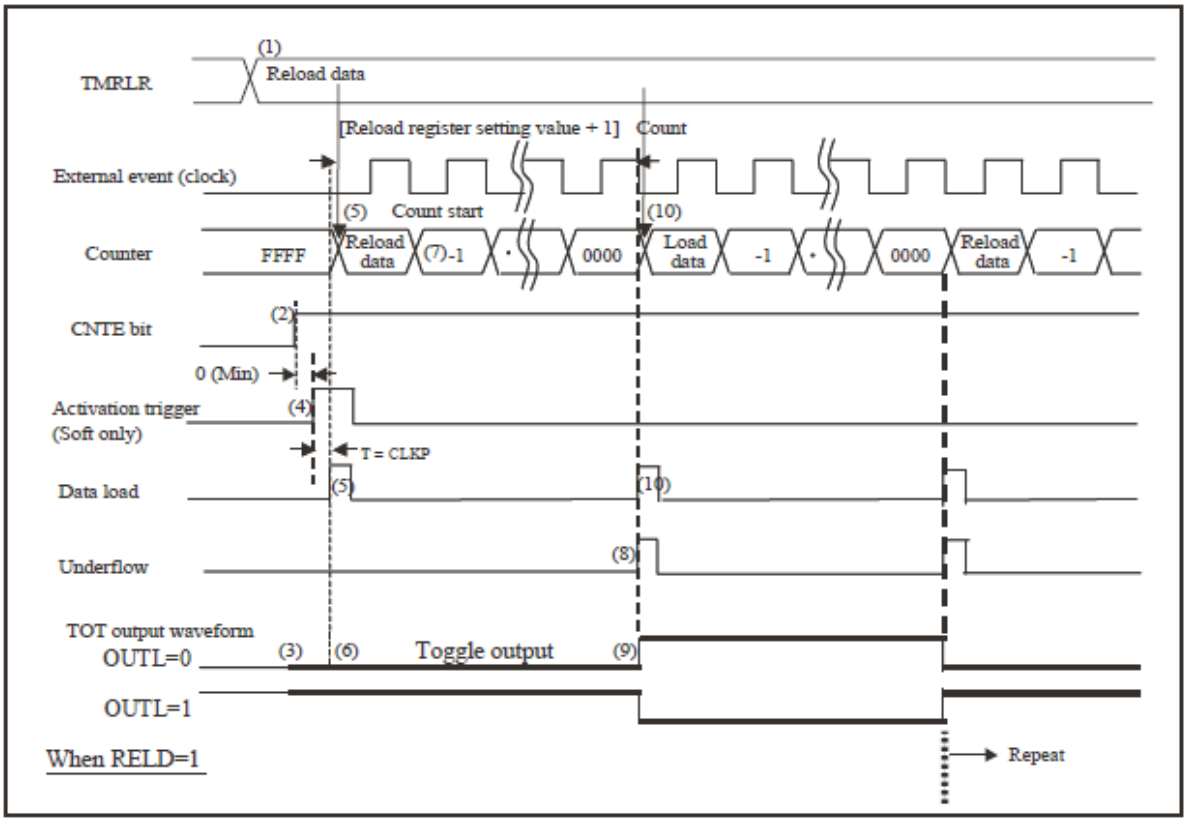


Figure 31-8 Operation in external event clock reload mode (outputs a pulse with a 50% duty ratio)

(1) Set reload value to reload register
(2) Enable reload timer count operation
(3) TOT pin output
(4) Generate reload trigger (activation): software trigger only
(5) Load reload value
(6) TOT pin output (initial value)
(7) Counter count down (external event synchronous)
(9) TOT pin output level reversal
(10) Reload reload value
(11) Repeat steps (6) to (9)
(See “Caution (Page No.31-17)”.)

30.5.4. External Event Clock/One-shot Mode

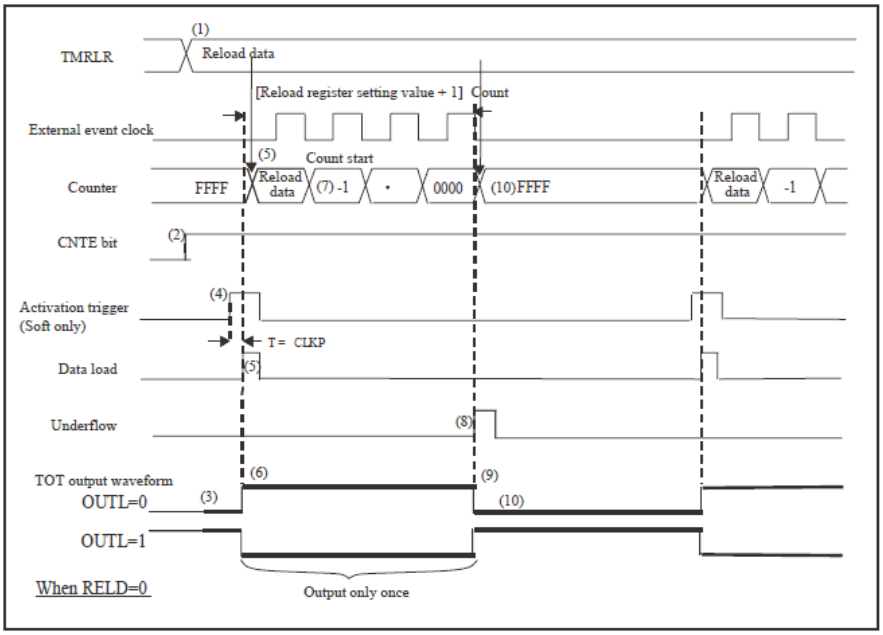


Figure 31-9 Operation in external event clock one-shot mode, (one-shot pulse is output)

(1) Set reload value to reload register
(2) Enable reload timer count operation
(3) TOT pin output
(4) Generate reload trigger (activation): soft trigger only
(5) Load reload value
(6) TOT pin output (during count, output “H”/OUTL=“0”)
(7) Counter count down (via external events)
(8) Generate counter underflow
(9) TOT pin output reversal
(10) Stop counter, wait for next activation trigger

Note:The first reload will be delayed by a maximum of 1 T (T: count clock).

30.5.5. Operation during Reset

A reset (reset on INITX signal, watchdog reset, software reset) will cause the registers in the reload timer to be initialized. The initial value of reload registers is indeterminate.
For detailed information on initial values, see the explanation of registers.

30.5.6. Operation during Sleep Mode

Even after making the transition to sleep mode, the operation of the reload timer will continue.

30.5.7. Operation during Stop Mode

When the transition is made to stop mode, the operation of the reload timer stops.
Afterwards, when returning from stop mode, it will return to the state it was in before the transition to stop mode.

30.5.8. Operation when Returning from Stop Mode

When returning due to an external interrupt, the reload timer will continue operation from its stopped state. When returning from a reset (INITX), it will return to the initial state (down counter stopped, no TOT pin output).

30.5.9. Status Transition

The status of the counter is decided by the CNTE bit of the reload timer control register and the internal WAIT signal.

Settable statuses are:

STOP status: Stopped (CNTE="0", WAIT="1")

WAIT status: Waiting for activation trigger (CNTE="1", WAIT="1")

RUN status : Count operation running (CNTE="1", WAIT="0")

LOAD status: Loading value to counter (from RUN/WAIT, TRG="1" or underflow: CNTE="1", WAIT="0")

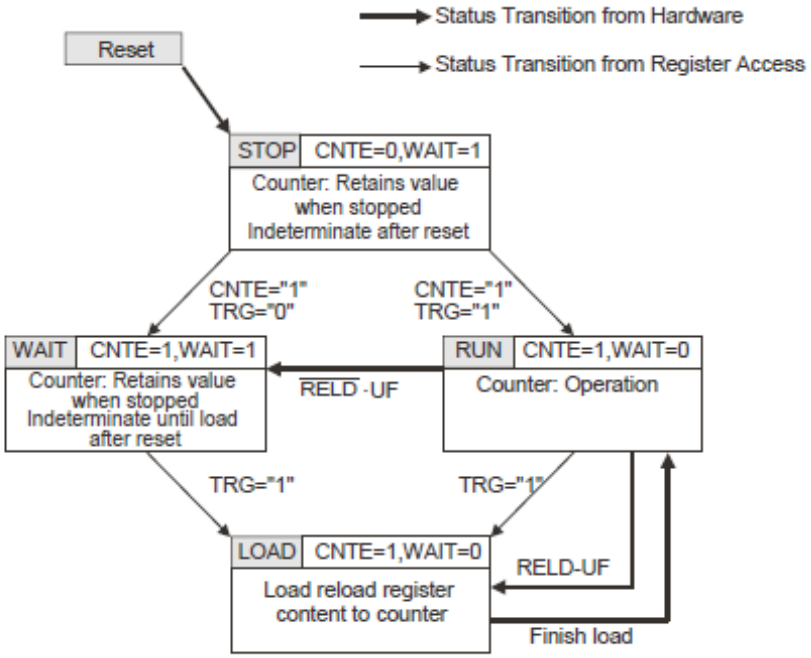


Figure 31-10 Status Transition Diagram of the Reload Timer

30.6. Setting

Table 31-1 Settings Necessary for Moving the Reload Timer (Internal Clock Operation)

Setting	Setting Registers	Setting Procedure*
Reload value settings	Reload (TMRLR0-TMRLR15)	See 31.7.1
Count clock selection (internal clock selection)	Reload timer control status (TMCSR0-TMCSR15)	See 31.7.2
Enable reload timer count operation		See 31.7.3
Mode selection (reload /one-shot)		See 31.7.4
Output reversal specification		See 31.7.5
Reload trigger selection (activation selection) Soft trigger External trigger (Rising edge/falling edge/both edges)		See 31.7.6
Generate activation trigger		See 31.7.8
Soft trigger -> Software trigger bit setting	Reload timer control status (TMCSR0-TMCSR15)	
External trigger -> Input trigger to TIN pin	External input	

*: For the setting procedure, refer to the section indicated by the number.

Table 31-2 Settings Necessary for Moving the Reload Timer (External Event Operation)

Setting	Setting Registers	Setting Procedure*
Reload value setting	Reload (TMRLR0-TMRLR15)	See 31.7.1
Count clock selection (external event clock selection)	Reload timer control status (TMCSR0-TMCSR15)	See 31.7.2
Enable reload timer count operation		See 31.7.3
Mode selection (reload /one-shot)		See 31.7.4
Output reversal specification		See 31.7.5
External event clock active edge selection (Rising edge/falling edge/both edges)		See 31.7.7
TIN0 - TIN7 pin external event input		See CCNT PIN Multiplex
Generate activation trigger Soft trigger -> Software trigger bit setting	Reload timer control status (TMCSR0-TMCSR15)	See 31.7.8

*: For the setting procedure, refer to the section indicated by the number.

Table 31-3 Items Necessary for Performing Reload Timer Interrupts

Setting	Setting Registers	Setting Procedure*
Reload timer interrupt settings Interrupt request clear Enable interrupt requests	Reload timer control status (TMCSR0-TMCSR15)	See 31.7.9

*: For the setting procedure, refer to the section indicated by the number.

Table 31-4 Settings Necessary for Stopping the Reload Timer

Setting	Setting Registers	Setting Procedure*
Reload timer stop bit setting	Reload timer control status (TMCSR0-TMCSR15)	See 31.7.10

*: For the setting procedure, refer to the section indicated by the number.

30.7. Q & A

30.7.1. What is the reload value setting (rewriting) procedure?

The reload value is set by the 16 bit reload registers TMRLR0-TMRLR15.
The equation for the values to be set is as follows.

- Formula
  TMRLR register value = {reload interval/count clock}-1
- Allowed Range
  TMRLR register value = 0~FFFh (65535)

30.7.2. What are the kinds of count clocks and how are they selected?

The count clock is chosen from the 6 types in the table below.
Selection is done via the count clock selection bit.

Table 31-5 TMCSR.CSL[2:0]

Count Clock	Counter clock selection bit			Count clock example		
	CSL2	CSL1	CSL0	When CLKP= 41.66MHz	When CLKP= 20.83MHz	When CLKP= 10.42MHz
CLKP/2	0	0	0	48ns	96ns	192ns
CLKP/8	0	0	1	192ns	384ns	768ns
CLKP/32	0	1	0	768ns	1.54μs	3.07μs
External event	0	1	1	Pulse width: 2/CLKP min		
CLKP/64	1	0	1	1.54μs	3.07μs	6.14μs
CLKP/128	1	1	0	3.07μs	6.14μs	12.29μs
Disabled *	1	0	0	-----		
	1	1	1	-----		

(*: See “Caution (Page No.31-17)”.)

30.7.3. How to I enable/disable the reload timer count operation?

Use the timer count enable bit (TMCSR.CNTE).

Control Details	RLT operation permission bit (CNTE)
To stop the reload timer	Set to “0”
To enable the reload timer’s count operation	Set to “1”

Cannot be reopened from the stopped state. Enable before activation or simultaneous with activation.

30.7.4. How do I set the reload timer mode (reload/one-shot)?

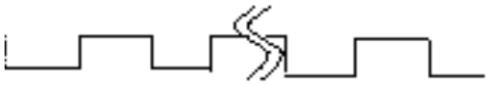
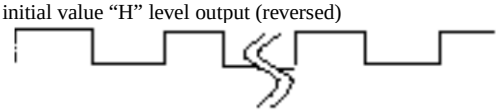
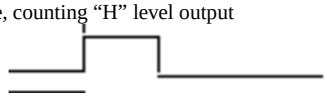
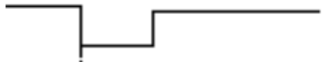
Use mode selection bit (TMCSR.RELD).

Operation Mode	Mode selection bit (RELD)
To set to one-shot mode	Set to “0”
To set to reload	Set to “1”

30.7.5. How do I reverse the output level?

The settings for the output level are detailed in the following table.

The setting is done via timer output level bit (TMCSR.OUTL).

Output level	Timer output level bit (OUTL)
Reload mode, Initial value “L” level output 	Set to “0”
Reload mode, initial value “H” level output (reversed) 	Set to “1”
One-shot mode, counting “H” level output 	Set to “0”
One-shot mode, counting “L” level output (reversed) 	Set to “1”

30.7.6. What are the kinds of triggers, and how do I select them?

- Selection is done via the operation mode selection bit (TMCSR.MOD[2:0]).

There are 4 types of reload triggers when an internal clock is selected.

Trigger	Operation mode selection bit (MOD[2:0])
Software trigger (TRG bit set)	Set to “000”
External trigger from TINx pin (rising edge)	Set to “001”
External trigger from TINx pin (falling edge)	Set to “010”
External trigger from TINx pin (both edges)	Set to “011”
	“100”, “101”, “110”, “111” are disabled *

Reload is repeated on down counter underflow.

(*: See “Caution (Page No.31-17)”.)

- The reload trigger (activation) when an external event is selected is a software trigger.

  Reload is repeated on down counter underflow.

30.7.7. What are the types of external event clock active edges and how do I select them?

The setting is done via the trigger selection bit (TMCSR.MOD[1:0]).

There are three types of active edges.

Active edge	Trigger selection bit (MOD1-MOD0)
Rising edge	Set to “01”
Falling edge	Set to “10”
Both edges	Set to “11”

MOD2 settings have no meaning, no matter if they are set to “0” or “1”.

30.7.8. How do I generate an activation trigger?

- Generating a soft trigger

The setting is done via the software trigger bit (TMCSR.TRG).

When the software trigger bit (TGR) is set to“1”, a trigger is generated.

To enable operation and activate at the same time, set the count permission bit (TMCSR.CNTE) and the soft trigger bit (TMCSR.TRG) simultaneously.

- Generating an external trigger

By inputting the edge specified by the trigger selection bit to the trigger pin corresponding to each reload timer, a trigger is generated.

Timer	Trigger pin
Reload timer 0	TIN0
Reload timer 1	TIN1
Reload timer 2	TIN2
Reload timer 3	TIN3
Reload timer 4	TIN4
Reload timer 5	TIN5
Reload timer 6	TIN6
Reload timer 7	TIN7

30.7.9. How do I enable interrupts?

Enabling interrupts, interrupt request flag

Enabling of interrupts is done via the interrupt request permission bit (TMCSR0.INTE) ~ (TMCSR7.INTE).

	Interrupt request permission bit (INTE)
To disable interrupt requests	Set to “0”
To enable interrupt requests	Set to “1”

Clearing of interrupt requests is done via the interrupt request bit (TMCSR0.UF) ~ (TMCSR7.UF).

	Interrupt request bit (UF)
To disable interrupt requests	Set to “0”

30.7.10. How do I stop the reload timer?

This setting is done via the reload timer stop bit.

See “How to I enable/disable the reload timer count operation? (Page No.31-13)”.

30.8. Caution

- Count source select bit (TMCSR.CSL[2:0]) settings not in the table: “100”, “111” are disabled.
  If they are set, disable the reload timer operation before resetting the count source select bit.

- Operation mode bit (TMCSR.MOD2) must be set to “0”. If it is set to “1”, disable the reload timer count operation before resetting it. Also the value written during read/modify/write access may be read.

- Control bits (Count source select, operation mode, reload permission) must not be rewritten during operation.
  If they are set during operation, disable the reload timer count operation before resetting them.

- From activation timing, it takes T cycle for the reload value to be loaded to the down counter. (Cycle = 1/ CLKP, CLKP = peripheral clock)

- About output signal internal connections
  - Reload timer TOT0-TOT1 outputs are connected to the HDMAC0/1 trigger inputs.
  - Reload timer TOT2-TOT5 outputs are connected to the PPG4-PPG11 internal trigger inputs.
  - Reload timer TOT8-TOT13 outputs are connected to the ConfigFIFO internal trigger inputs.

- Rewriting of the count clock selection bit (CSL[2:0]), operation mode selection bit (MOD[2:0]), output level setting bit (OUTL), reload permission bit (RELD), and timer interrupt request permission bit (INTE) should be done when the reload timer is stopped (TMCSR.CNTE=“0”).

- The internal prescaler should be already set when the timer count permission bit (TMCSR.CNTE) is set to “1”.

- If interrupt request flag set timing and clear timing overlap, the flag setting will be given priority and the clear operation will be made invalid.

- When writing to the reload register and the reload timing overlap, the old data will be loaded to the counter. The new data will be loaded during the next reload timing.

- If the loading and counting of the timer register overlap, the load (reload) operation is given priority.

- If you want to enable the count at the same time as you start the count operation, set both the timer count permission bit (TMCSR.CNTE) and the software trigger bit (TMCSR.TRG) to “1”.

31. Electrical Characteristics

31.1. Maximum Ratings

Table 32-1 Maximum Ratings

Parameter	Symbol	Value	Unit
Supply voltage	VDDI, VDDA VDDE VDD_RSDS VD5 HVDD	-0.5 to 2.5(*1) -0.5 to 4.0(*2) -0.5 to 6.0(*3)	V
Input voltage	V_I	-0.5 to VDDI+0.5(< 2.5V) -0.5 to VDDE+0.5(< 4.0V) -0.5 to VD5+0.5(< 6.0V)	V
Output voltage	V_O	-0.5 to VDDI+0.5(< 2.5V) -0.5 to VDDE+0.5(< 4.0V) -0.5 to VD5+0.5(< 6.0V)	V
Storage temperature	T_{ST}	-55 to 125	Degrees C
Junction temperature	T_J	-40 to 125	Degrees C

(*1)power supply for internal or APIX

(*2)power supply for I/O part

(*3)power supply for internal or I/O part

Notes:

- Applying stress exceeding the maximum ratings (voltage, current, temperature, etc.) may cause damage to semiconductor devices. Never exceed the ratings above.
- Never connect IC outputs or I/O pins directly, or connect them to VD5,VDDI,VDDA,VDDE, VDD_RSDS or VSS directly; otherwise thermal destruction of elements will result, but which does not apply to pins designed to prevent signal collision.
- Provide ESD protection, such as grounding when handling the product; otherwise externally-charged electric charge flows inside the IC and discharges, which may result in damage to the circuit.
- Applying voltage higher than VD5 or lower than VSS to I/O pins of CMOS IC, or applying voltage higher than the ratings between VD5,VDDI,VDDA,VDDE, VDD_RSDS and VSS may cause latch up. The latch up increases supply current, resulting in thermal destruction of elements. When handling the product, never exceed the maximum ratings.

Table 32-2 ADC Maximum Ratings

Parameter	Symbol	Value	Unit
Supply Voltage	AD_AVCC	-0.5 to 6.0	V
Input voltage	AD_AVRH AD_AVCC AD_AVSS AD_ATGX AD_AN[5:0]	-0.5 to VD5+0.5(< 6.0V)	V
Junction Temperature	T_J	-40 to 125	degree

31.2. Recommended Operating Conditions

Table 32-3 3.3V Standard CMOS I/O Recommended Operating Conditions

Parameter		Symbol	Rating			Unit
			Min.	Typ.	Max.	
Power supply voltage		VDDE,VDD_RSDS VDDI, VDDA	3.0 1.65	3.3 1.8	3.6 1.95	V
Input voltage (High level)	3.3 V CMOS	VIH	0.8*VDDE		VDDE	V
Input voltage (Low level)	3.3 V CMOS	VIL	VSS		0.2*VDDE	V
Operating ambient temperature		TA	-40		105	degree
Junction temperature		TJ	-40		125	degree

Table 32-4 5.0V Standard CMOS I/O Recommended Operating Conditions

Parameter		Symbol	Rating			Unit
			Min.	Typ.	Max.	
Power supply voltage		VD5,HVDD AD_AVCC	3.0	5.0	5.5	V
Input voltage (High level)	5.0 V CMOS	VIH	0.8*VD5		VD5	V
Input voltage (Low level)	5.0 V CMOS	VIL	VSS		0.2*VD5	V
Operating ambient temperature		TA	-40		105	degree
Junction temperature		TJ	-40		125	degree

Notes:
The recommended operating conditions are primarily intended to assure the normal operation of semiconductor device. The values of electrical characteristics are guaranteed under the requirements above, so use the product accordingly. Using the product without observing the conditions may affect the product's reliability. Performance of this product is not guaranteed if used under unspecified conditions and by an unspecified combination of logic. Be sure to contact Fujitsu group when using the product under such conditions.

31.3. Precautions at Power ON

31.3.1. Recommended Power ON/OFF Sequence

Please observe the following power supply ON/OFF sequences:

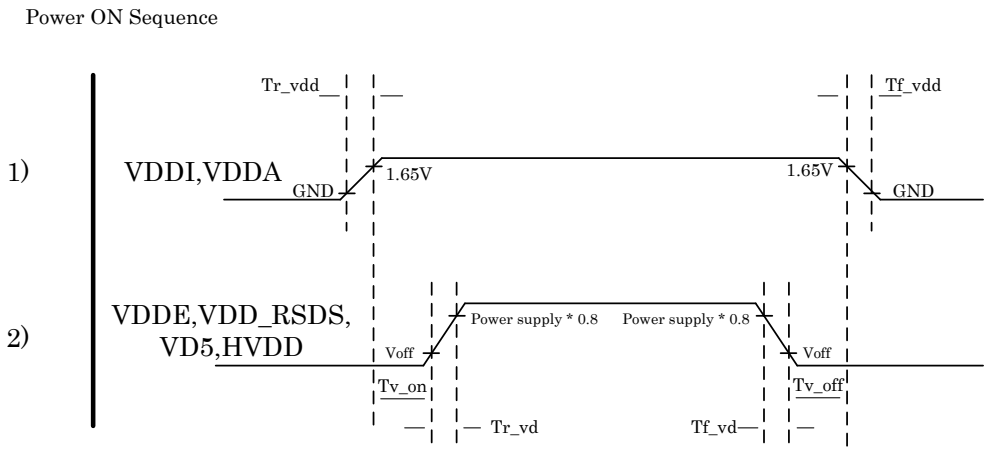


Figure 32-1 Recommended Power ON/OFF Sequence

Table 32-5 Power ON/OFF standard

Item	Sign	Standard		Unit
		min	max	
Power supply standing up time.(VDDI,VDDA) Power supply fall time.(VDDI,VDDA)	Tr_vdd Tf_vdd	0	1	s
Time to start-up of power supply. (VDDE,VDD_RSDS,VD5,HVDD)	Tv_on	0	950	ms
Time to turning off power supply.(VDDI,VDDA)	Tv_off	0	1	s
Power supply standing up time. (VDDE,VDD_RSDS,VD5,HVDD)	Tr_vd	0	50	ms
Power supply fall time. (VDDE,VDD_RSDS,VD5,HVDD)	Tf_vd	0	1	s

Notes:

Power ON:

- We recommend that 1.8V signals (VDDI, VDDA) are switched on before other signals
- Note: If VDDI is switched on at the same time or later, correct and reliable internal POR (Power On Reset) operation can not be guaranteed.

Power OFF:

- Switch off all power supplies within 1 second
- Note: Switch off VD5, VDDE, VDD_RSDS before VDDI to avoid undefined outputs

You must switch on *all* power supplies, enabling a subset of power supplies is prohibited!
Exception: If the complete SMC block is not used, then the HVDDx pins can be connected to GND (see also the 'Unused Pins' section).

Please note that if you are using the XRST solution, the power sequence can be freely selected and that the

minimum wait time for a power-on after a power off (1 ms) also no longer applies.

31.3.2. Power ON Reset

After the VDDI supply has been switched on and VD5 has exceeded 2.7V, the internal power reset module counts upwards until it reaches a waiting threshold of 8 ms (2^{15} cycles of XTAL0), assuming an XTAL0 clock of 4MHz. After this point in this time the power on reset is diasserted.
Note also that it is necessary to fulfill the conditions in Table32-6 to attain a stable power-on reset operation.

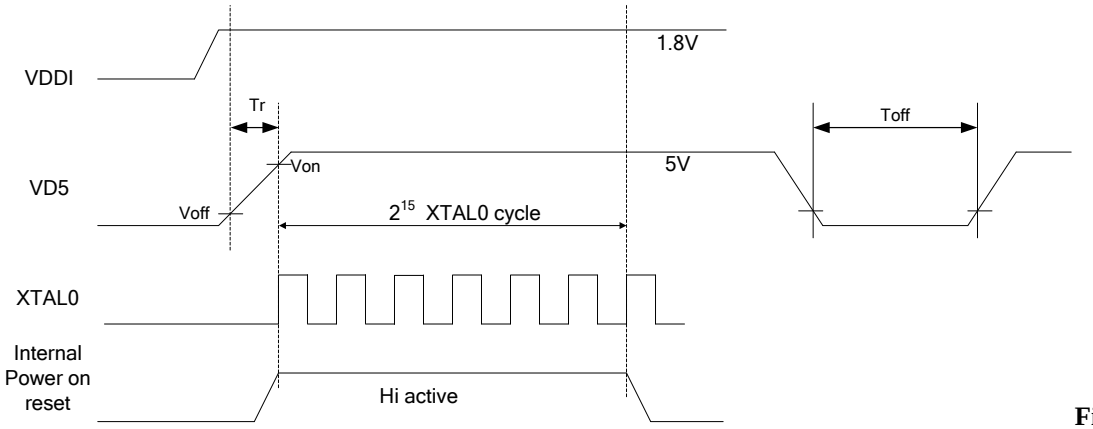


Figure 32-2 Power ON Sequence

Table 32-6 Power ON standard

Item	Sign	Standard		Unit
		MIN	MAX	
Power supply standing up time	Tr	-	50	ms
Power supply beginning time	Voff	-	0.2	V
Power supply attainment voltage	Von	2.7	-	V
Power supply determination time (see also next section!)	Toff	1	-	ms

31.3.3. Restriction of minimum interval time of power off and on

There is a restriction of interval of time between power off and power on again.
Internal power on reset module don't work correctly less than 1 ms power off time. So we must take more than 1ms interval time between power off and power on again.

Please note that if you are using the XRST solution, the power sequence can be freely selected and that the minimum wait time for a power-on after a power off (1 ms) also no longer applies.

31.4. DC Characteristics

31.4.1. 3.3 V Standard CMOS I/O

Table 32-7 Standard CMOS I/O DC Characteristics

Measurement condition: VDDE = 3.0 to 3.6V, VSS = 0 V, Ta = -40 to 105degree

Parameter	Symbol	Condition		Rating			unit
				Min.	Typ.	Max.	
H level input voltage	VIH	-		0.8*VDDE		VDDE	V
L level input voltage	VIL	-		VSS		0.2*VDDE	V
H level output voltage	VOH	Driving capability 1	IOH = -4 mA	VDDE-0.5		-	V
		Driving capability 2	IOH = -8 mA				
L level output voltage	VOL	Driving capability 1	IOH = 4 mA	-		0.4	V
		Driving capability 2	IOH = 8 mA				
Input leakage current	IL	-	-	-5	-	5	µA

Driving capabilities 1 to 2 in the table above indicate the following external pins:

Driving capability 1: TCON_TSIG[11:0]
Driving capability 2:none

31.4.2. 5.0 V Standard CMOS I/O

Table 32-8 Standard CMOS I/O DC Characteristics

Measurement condition: VD5 = 3.0 to 5.5 V, VSS = 0 V, Ta = -40 to 105degree

Parameter	Symbol	Condition		Rating			Unit
				Min.	Typ.	Max.	
H level input voltage	VIH	-		0.8*VD5	-	VD5	V
L level input voltage	VIL	-		VSS	-	0.2*VD5	V
H level output voltage	VOH	Driving capability 1	4.5V < VD5 =< 5.5V IOH = -2 mA	VD5-0.5	-	-	V
			3.0V =< VD5 =<4.5V IOH = -1.6 mA				
		Driving capability 2	3.0V =< VD5 =< 5.5V IOH = -3 mA				
		Driving capability 3	4.5V < VD5 =< 5.5V IOH = -5 mA				
			3.0V =< VD5 =<4.5V IOH = -3 mA				
		Driving capability 4	4.5V < VD5 =< 5.5V Ta = -40degree IOH = -40 mA				
			4.5V < VD5 =< 5.5V IOH = -30 mA				
			3.0V =< VD5 =<4.5V IOH = -20 mA				
L level output voltage	VOL	Driving capability 1	4.5V < VD5 =< 5.5V IOH = 2 mA	-	-	0.4	V
			3.0V =< VD5 =<4.5V IOH = 1.6 mA				
		Driving capability 2	3.0V =< VD5 =< 5.5V IOH = 3 mA				
		Driving capability 3	4.5V < VD5 =< 5.5V IOH = 5 mA				
			3.0V =< VD5 =<4.5V IOH = 3 mA				
		Driving capability 4	4.5V < VD5 =< 5.5V Ta = -40degree IOH = 40 mA				
			4.5V < VD5 =< 5.5V IOH = 30 mA				
			3.0V =< VD5 =<4.5V IOH = 20 mA				
Input leakage current	IL	-	-	-5	-	5	µA

Driving capabilities 1 to 4 in the table above indicate the following external pins:

Driving capability 1: SG_SGA,SG_SGO,PWM_O[11:4],GPIO[7:0], HOST_DO,HOST_INT, I2C_SCL,I2C_SDA
Driving capability 2: none
Driving capability 3: SPI0_DO,SPI0_SCK,
Driving capability 4: SMC_1P[0],SMC_1M[0],SMC_2P[0],SMC_2M[0]

31.4.3. RSDS IO

The relay terminal of RSDS I/O is DISPP[9:0] and DISPN[9:0].

Table32-9 is only valid for Differential Mode. BOOST is mode signal.

Table 32-9 Standard RSDS I/O Characteristics

Measurement condition: VD5 = 3.0 to 3.6 V, VSS = 0 V, Ta = -40 to 105degree

Parameter	Symbol	Condition		Rating			Unit
				Min.	Typ.	Max.	
Differential Output Voltage	VOD	Driving capability 1	BOOST=H R=50Ohm	105	170	325	mV
		Driving capability 2	BOOST=L R=100Ohm	105	170	325	
Offset Voltage	VOS			0.5	1.2	1.5	V
RSDS Driver current	Irsds	Driving capability 1	BOOST=H	2.10	3.4	6.50	mV
		Driving capability 2	BOOST=L	1.05	1.7	3.25	

Table32-10 is valid in TTL mode.

Table 32-10 Standard CMOS I/O DC Characteristics

Measurement condition: VD5 = 3.0 to 3.6 V, VSS = 0 V, Ta = -40 to 105degree

Parameter	Symbol	Condition		Rating			Unit
				Min.	Typ.	Max.	
H level output voltage	VOH	Driving capability 1	BOOST=H IOH=-5mA	2.5	-	-	V
		Driving capability 2	BOOST=L IOH=-2mA	2.6	-	-	
L level output voltage	VOL	Driving capability 1	BOOST=H IOH=5mA	-	-	0.4	V
		Driving capability 2	BOOST=L IOH=2mA	-	-	0.3	

31.4.4. ADC

Table 32-11 External pinlist for ADC

Pin Name	I/O	I/F	Function
AD_AN[8:0]	I	5V	Analog signal input
AD_ATGX	I	5V	digital signal input
AD_AVSS	I	GND	Analog GND input
AD_AVCC	I	5V	5V analog power supply
AD_AVRL	I	0V	Reference supply (-)
AD_AVRH	I	5V	Reference supply (+)

Table 32-12 Recommended Operating Conditions

Parameter	Symbol	Rating			Unit
		Min.	Typ.	Max.	
5V analog power supply	AVCC	2.7	5.0	5.5	V
1.8V digital power supply	VDDI	1.6	1.8	2.0	V
Analog Input signal	AD_AN[8:0]	AVSS	-	AVCC	V
Operating ambient temperature	Ta	-40	25	105	degree

Table 32-13 ADC electric characteristics

condition : AD_AVCC= 2.7V~5.5V, Ta= -40~105degree, AD_AVSS=VSS=0V

Parameter	symbol	Rating			unit	Remarks
		MIN.	TYP.	MAX.		
Resolution		-	-	13	bit	
Total margin		-3.0	-	+3.0	LSB	
Non linearity margin	INL	-2.5	-	+2.5	LSB	
Differential linearity margin	DNL	-1.9	-	+1.9	LSB	
Zero transition voltage	VOT	AVRL-1.5	AVRL+0.5	AVRL+2.5	LSB	
Full scale transition voltage	VFST	AVRH-3.5	AVRH-1.5	AVRH+0.5	LSB	
5V analog source current	IAVCC	-	1.9	3.7	mA	
Reference source current	Iref	-	520	810	uA	
Output impedance of Analog signal source	ext	-	-	4.2	kΩ	AVCC≧2.7V, Sampling time minimum
Analog input equivalent resistance	Rin	-	-	2.6	kΩ	AVCC≧4.5V
		-	-	12.1	kΩ	AVCC≧2.7V
Analog input equivalent capacitance	Cin	-	-	8.5	pF	

31.4.5. APIX characteristics

Power Supply

The analog logic portion (CML) of the APIX analog IP block shall be supplied by dedicated supply pads. (Bit-Error performance is impacted by the quality of the provided analog supply. Values beyond specification can lead to significant increase in the bit-error rate.)

Digital logic within the APIX analog block shall be provided from the core supply system via the digital hard macro. The digital hard macro is supplied from the core.

Table 32-14 APIX electric characteristics

Parameter	Min	Max	Comment
VDDA	1.65V	1.95V	DC Supply Voltage Analog and CML Termination
VDD	1.65V	1.95V	DC Supply Voltage from Digital Core
IDD_ANA		65 mA	Supply Current analog, without serial upstream output
Idriver	0.6 mA	20 mA	Nominal output driver current upstream
IDD_ANA_pwrdwn		430 uA	Supply Current analog, when power down

31.5. AC Characteristics

31.5.1. Host Interface signal timings

Table 32-15 AC Timing of Host I/F signals

Signal Name	Symbol	Description	Value			unit
			Min	Typ	Max	
HOST_SCK	t_{cyc}	Clock Cycle time	48.3	-	-	ns
HOST_XCS	t_{cssr}	Chip Select setup time	*1)	-	-	ns
	t_{cshr}	Chip Select hold time	*1)	-	-	ns
HOST_DO	t_{do}	Data output delay time	4.0	-	25.8	ns
HOST_DI	t_{dsr}	Data input setup time	5	-	-	ns
	t_{dhr}	Data input hold time	0	-	-	ns

*1)Please refer Chapter4 Host Interface

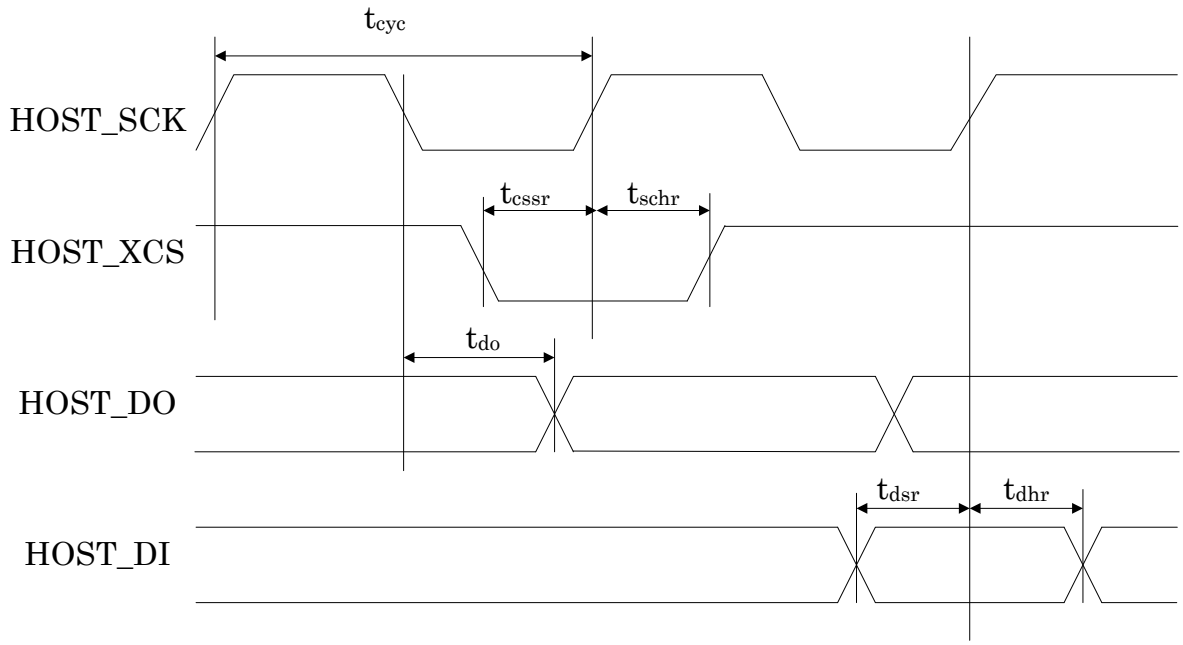


Figure 32-3 Host I/F Timings

31.5.2. GPIO signal timings

Table 32-16 AC Timing of GPIO signals

Signal Name	Symbol	Description	Value			unit
			Min	Typ	Max	
GPIO[19:12, 7:0]	t_{dw}	Input Data-width	24.1	-	-	ns

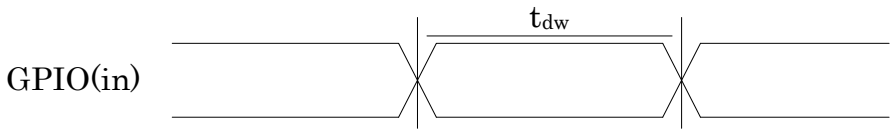


Figure 32-4 GPIO Timings

31.5.3. External Interrupt signal timings

Table 32-17 AC Timing of External Interrupt signals

Signal Name	Symbol	Description	Value			unit
			Min	Typ	Max	
INT[4:0]	t_{dw}	Input Data-width	24.1	-	-	ns

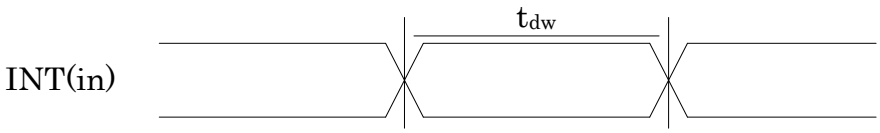


Figure 32-5 External Interrupt Timings

31.5.4. PWM signal timings

Table 32-18 AC timing of PWM signals

Signal Name	Symbol	Description	Value			unit
			Min	Typ	Max	
VPWM_O[3:0]	t_{dw1}	Output Data-width	24.1	-	-	ns
PWM_O[11:4]	t_{dw2}	Output Data-width	24.1	-	-	ns

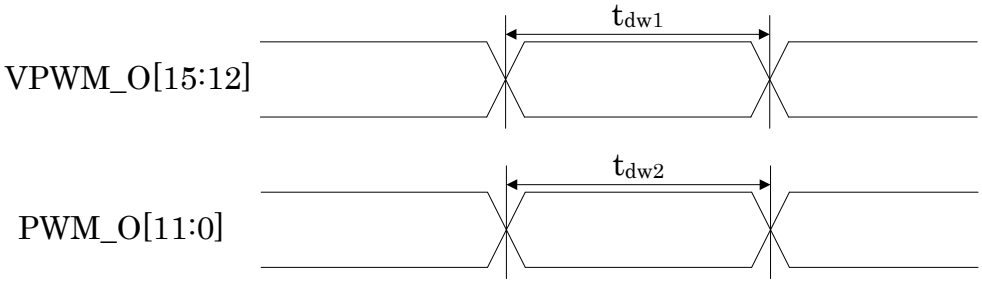


Figure 32-6 PWM Output timing

31.5.5. I²C Bus timings

Table 32-19 AC timing of I²C signals

Signal Name	Symbol	Description	Value		unit	Remark
			Min	max		
I2C_SCL	f_{SCL}	SCL clock frequency	0	400	kHz	
	$t_{HD;STA}$	Hold time (repeated) START condition. After this period, the first clock pulse is generated	0.6	-	us	
	t_{LOW}	LOW period of SCL clock	1.3	-	us	
	t_{HIGH}	HIGH period of SCL clock	0.6	-	us	
I2C_SDA	$t_{SU;STA}$	Setup time for a repeated START condition	0.6	-	us	

	$t_{HD;DAT}$	Data hold time for I2C-bus devices	0	0.9	us	
	$t_{SU;DAT}$	Data set up time	100	-	ns	
	t_r	Rise time of both SDA and SCL signals	20+0.1Cb	300	ns	
	t_f	Fall time of both SDA and SCL signals	20+0.1Cb	300	ns	
	$t_{SU;STO}$	Setup time for STOP condition	0.6	-	us	
	t_{BUF}	Bus free time between a STOP and START condition	1.3	-	us	
	C_b	Capacitive load fir each bus line	-	400	pF	
	t_{SP}	Pulse width of spike suppressed by input filter	0	1..1.5* t_{CLKP}	ns	*1

(*1) The noise filter will suppress signal spikes with a pulse width of 0ns and between (1 to 1.5) cycles of peripheral clock, depending on the phase relationship between I2C signals (SDA,SCL) and peripheral clock.

Note: t_{CLKP} is the cycle time of the peripheral clock.

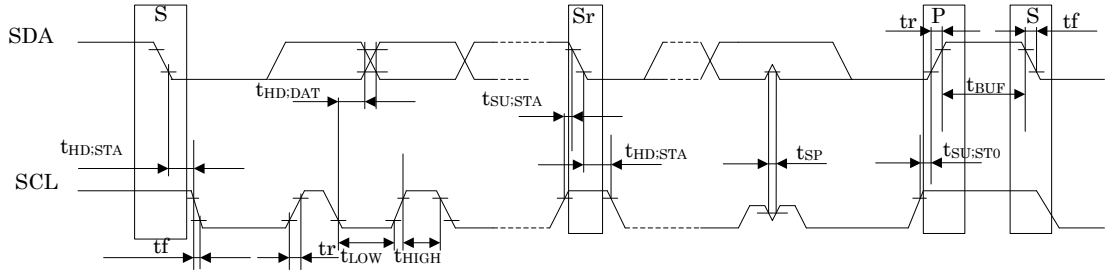


Figure 32-7 I²C access timing

31.5.6. SPI signal timings

Table 32-20 AC Timing of SPI signals

Signal name	Condition	Symbol	Description	Value		unit
				Min	Max	
SPIn_SCK	Internal clock operation (master mode)	t_{cyc}	Serial clock cycle time	4 t_{clkp}	-	ns
SPIn_DO		t_{SLOVI}	SPI_SCK fall to SPI_DO delay time	-20	20	ns
		t_{OVSHI}	SPI_DO to SPI_SCK delay time	m* t_{clkp} -20 (*1)	-	ns
SPIn_DI		t_{IVSHI}	Valid SPI_DI to SPI_SCK rise setup time	t_{clkp} +45	-	ns
		t_{SHIXI}	SPI_SCK rise to valid SPI_DI hold time	0	-	ns
SPIn_SCK	External clock operation (slave mode)	t_{SHSLE}	Serial clock “H” pulse width	t_{clkp}+10		ns
		t_{SLDHE}	Serial clock “L” pulse width	t_{clkp}+10		ns
SPIn_DO		t_{SLOVE}	SPI_SCK fall to SPI_DO delay time		2t_{clkp}+ 45	ns
SPIn_DI		t_{IVSHE}	Valid SPI_DI to SPI_SCK rise setup time	10		ns
		t_{SHIXE}	SPI_SCK rise to valid SPI_DI hold time	t_{clkp}+10		ns
SPIn_SCK		t_{FE}	SCK rising time		20	ns
		t_{RE}	SCK falling time		20	ns

*1):Parameter m depends on t_{cyc} and can be calculated as:

・ if t_{cyc} =2*k* t_{clkp},then m = k,where k is an integer > 2

・if t_{cyc} =(2*k+1)* t_{clkp},then m = k+1,where k is an integer > 1

Notes: ・The above values are AC characteristics for CLK synchronous mode.

・t_{clkp} is the cycle time of the peripheral clock.

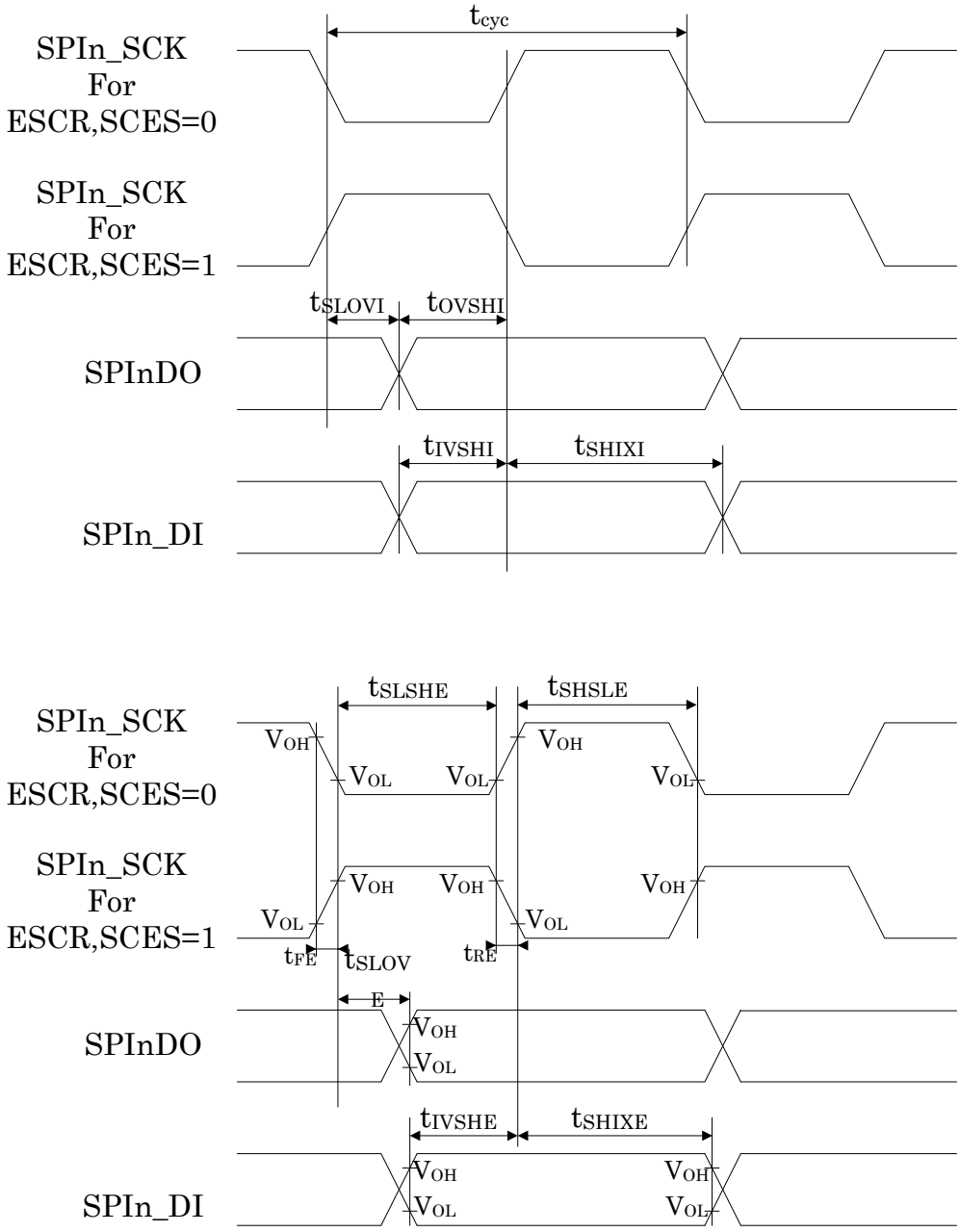


Figure 32-8 SPI Timings

*The polarity of SPI0/1_SCK is decided by the register setting.

31.5.7. Display signal timings

Table 32-21 AC Timing of Display signals

Symbol	Description	Unit	min	typ	max	Condition
RSDS operation mode						
RSSU	setup time	ns	3.5			C_L=15pF, Delay[i]=0
RSHD	hold time	ns	3.3			C_L=15pF, Delay[i]=0
f_RSCK	Frequency	MHz			42.0	
t_RSCK	Period	ns	23.810			
RSCKH	High Period	ns		10.405		C_L=5pF
RSCKL	Low Period	ns		10.405		C_L=5pF
	Duty cycle	%	48	50	52	
RSTr/f	Rise/Fall Time	ns			1.5	
TSIGSU	setup time	ns	6.0			C_L=15pF, SSWITCH[i]=0
TSIGHD	hold time	ns	6.0			C_L=15pF, SSWITCH[i]=0
TTL operation mode not bypass mode						
DISPSU	setup time	ns	4.3			C_L=5pF Boost=0, Delay[i]=1
			4.6			C_L=10pF Boost=1, Delay[i]=1
			4.1			C_L=15pF Boost=1, Delay[i]=1
			4.1			C_L=20pF Boost=1, Delay[i]=1
DISPHD	hold time	ns	4.3			C_L=5pF Boost=0, Delay[i]=1
			4.6			C_L=10pF Boost=1, Delay[i]=1
			4.1			C_L=15pF Boost=1, Delay[i]=1
			4.1			C_L=20pF Boost=1, Delay[i]=1
f_TTLCK	Frequency	MHz			42.0	
t_TTLCK	Period	ns	23.810			
TTLCKH	High Period	ns		10.405		C_L=5pF
TTLCKL	Low Period	ns		10.405		C_L=5pF
	Duty cycle	%	48	50	52	
t_rise/fall	Rise/Fall Time	ns			1.5	
TSIGSU	setup time	ns	6.9			C_L=5pF, Boost=0
			6.1			C_L=10pF, Boost=1
			6.5			C_L=15pF, Boost=1
			7.1			C_L=20pF, Boost=1
TSIGHD		ns	5.3			C_L=5pF, Boost=0
			6.4			C_L=10pF, Boost=1
			5.5			C_L=15pF, Boost=1
			4.8			C_L=20pF, Boost=1

Symbol	Description	Unit	min	typ	max	Condition
TTL operation mode bypass mode						
DISPSU	setup time	ns	0.9			C_L=20pF Boost=0,
DISPHD	hold time	ns	9.2			C_L=20pF Boost=0
f_TTLCK	Frequency	MHz			42.0	
t_TTLCK	Period	ns	23.810			
TTLCKH	High Period	ns		10.405		C_L=5pF
TTLCKL	Low Period	ns		10.405		C_L=5pF
	Duty cycle	%	48	50	52	
t_rise/fall	Rise/Fall Time	ns			1.5	
TSIGSU	setup time	ns	5.8			C_L=20pF, Boost=0
TSIGHD	Hold time	ns	7.8			C_L=20pF, Boost=0

The Bypass mode is not so flexible than not bypass mode. If more flexible display interface timing is needed please use TCON module.

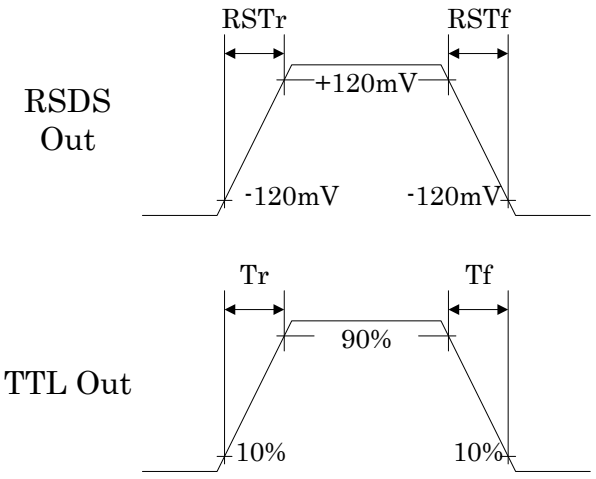


Figure 32-9 Rise Fall Times

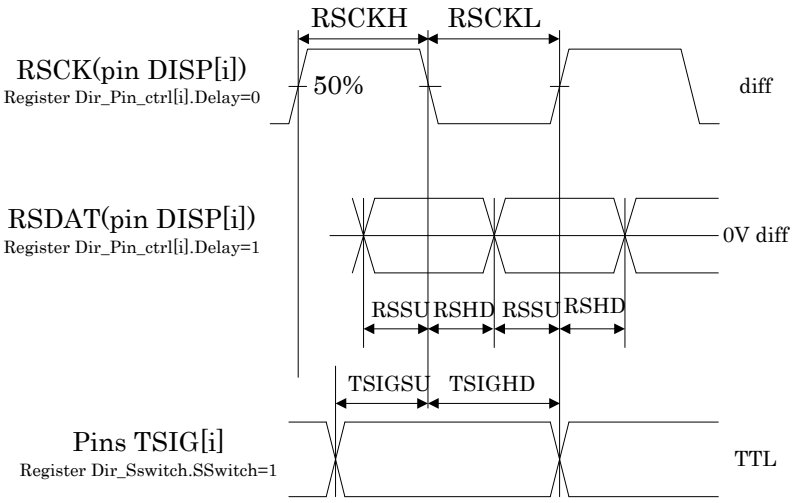


Figure 32-10 RSDS operation Output Timing

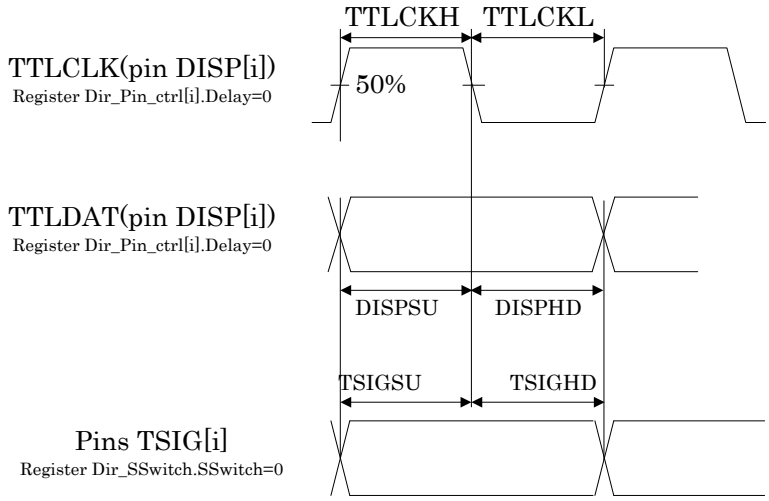


Figure 32-11 TTL operation output timing (not bypass mode) (1)

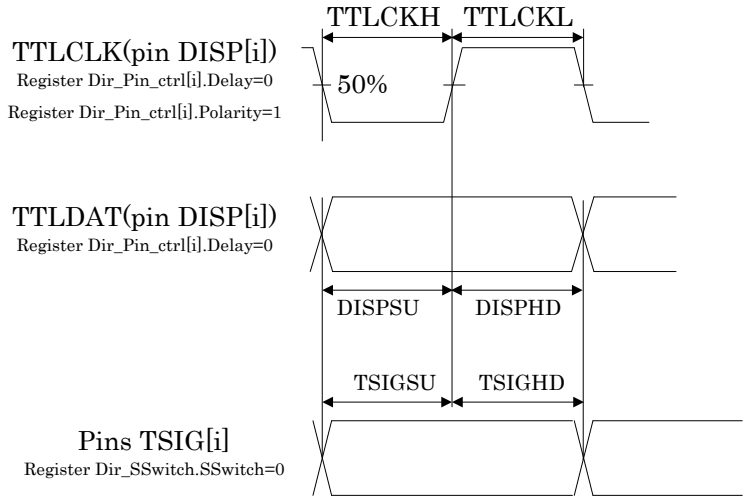
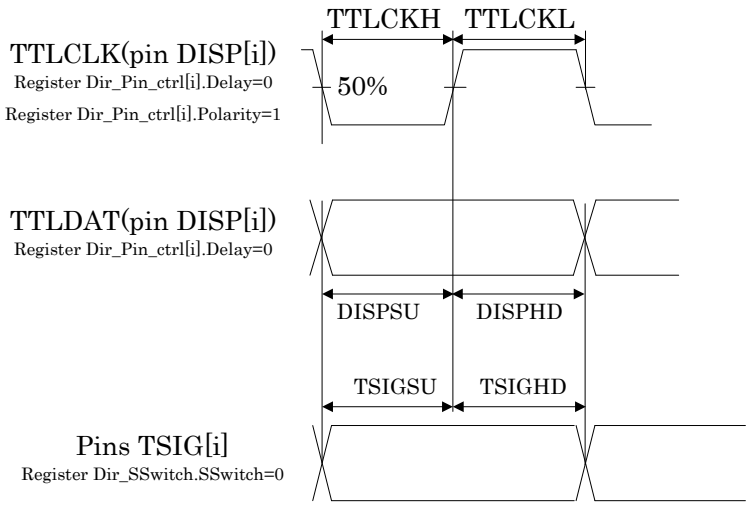


Figure 32-12 TTL operation output timing (not bypass mode) (2)



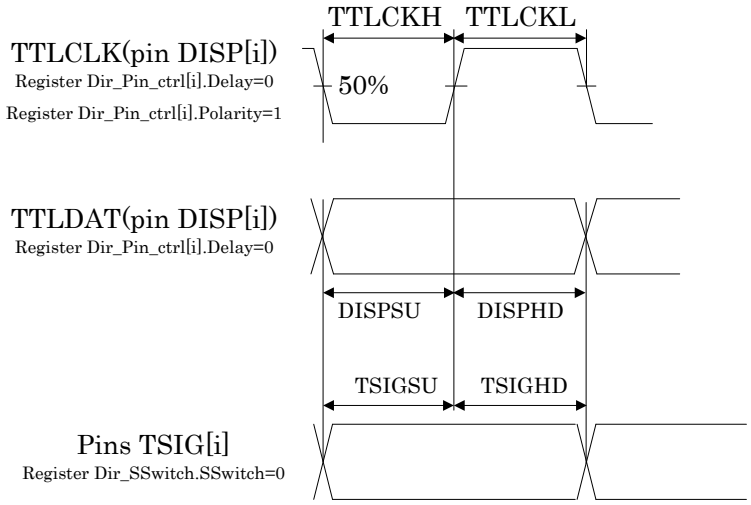


Figure 32-14 TTL operation output timing (bypass mode)

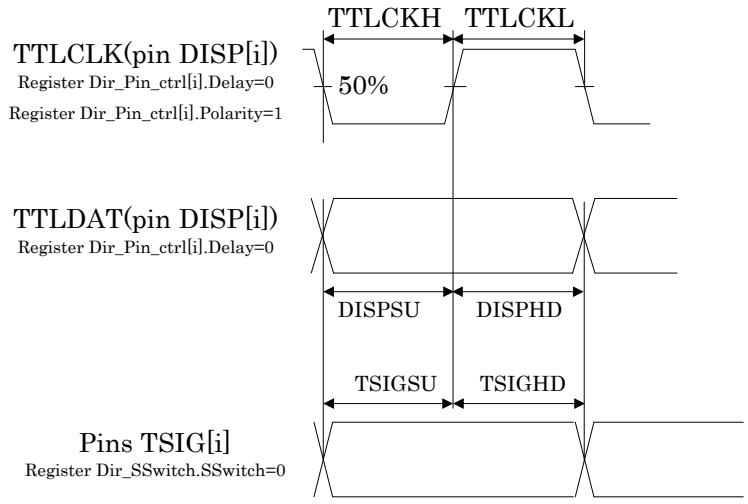


Figure 32-14 TTL operation output timing (bypass mode)