EPSON
EXCEED YOUR VISION

S1S65010
Technical Manual

SEIKO EPSON CORPORATION

Rev.1.4

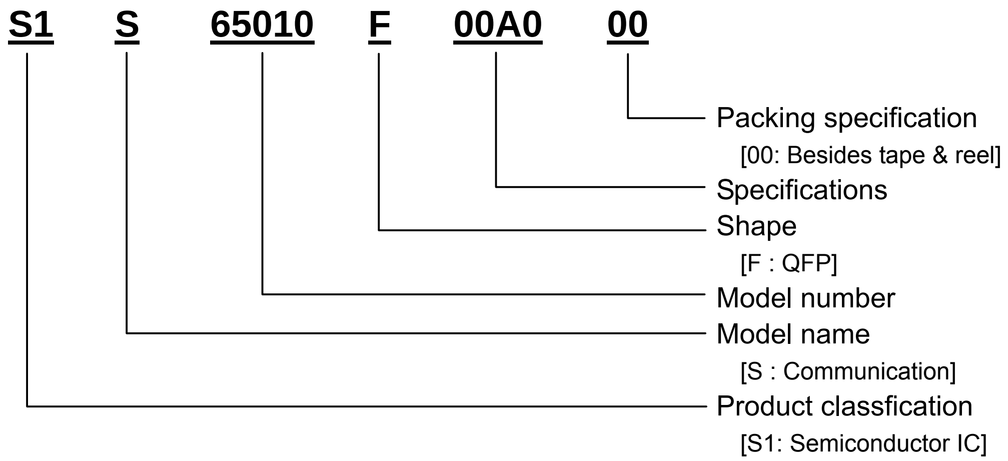
NOTICE

No part of this material may be reproduced or duplicated in any form or by any means without the written permission of Seiko Epson. Seiko Epson reserves the right to make changes to this material without notice. Seiko Epson does not assume any liability of any kind arising out of any inaccuracies contained in this material or due to its application or use in any product or circuit and, further, there is no representation that this material is applicable to products requiring high level reliability, such as, medical products. Moreover, no license to any intellectual property rights is granted by implication or otherwise, and there is no representation or warranty that anything made in accordance with this material will be free from any patent or copyright infringement of a third party. This material or portions thereof may contain technology or the subject relating to strategic products under the control of the Foreign Exchange and Foreign Trade Law of Japan and may require an export license from the Ministry of Economy, Trade and Industry or other approval from another government agency.

ARM POWERED® is a registered trademark of ARM.
CompactFlash is a registered trademark of Sandisk.
All other product names mentioned herein are trademarks and/or registered trademarks of their respective companies.

Configuration of product number

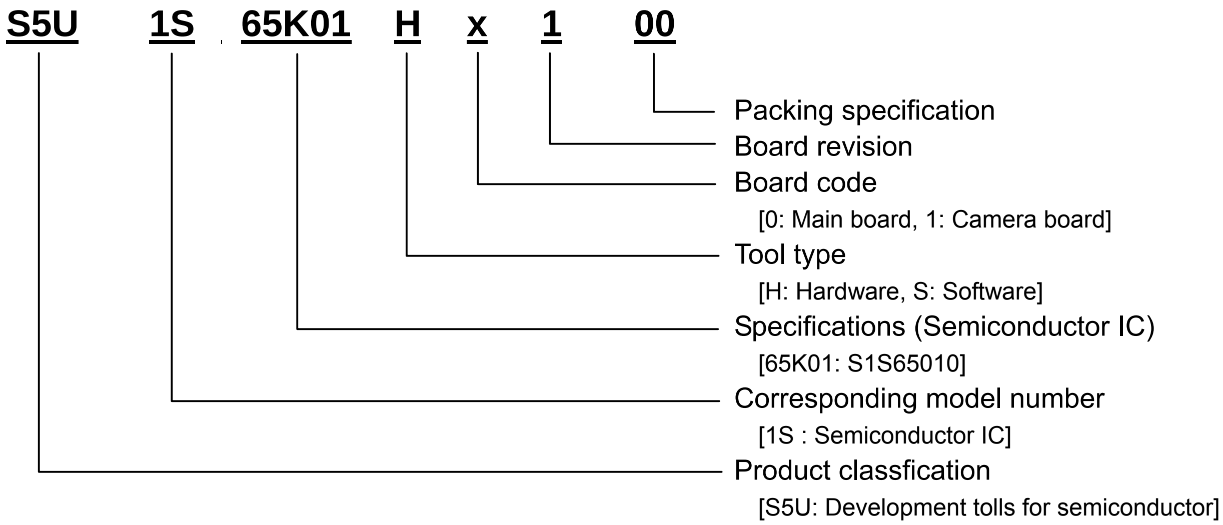
●Devices



S1 **S** **65010** **F** **00A0** **00**

Packing specification
[00: Besides tape & reel]
Specifications
Shape
[F : QFP]
Model number
Model name
[S : Communication]
Product classfication
[S1: Semiconductor IC]

●Development tools



S5U **1S** **65K01** **H** **x** **1** **00**

Packing specification
Board revision
Board code
[0: Main board, 1: Camera board]
Tool type
[H: Hardware, S: Software]
Specifications (Semiconductor IC)
[65K01: S1S65010]
Corresponding model number
[1S : Semiconductor IC]
Product classfication
[S5U: Development tolls for semiconductor]

Notes on Register Descriptions

Keep the following in mind when reading the register descriptions in this document.

Register descriptions in this document use the following abbreviations.

R/W : Read/write

RO : Read only

WO : Write only

RSV : Reserved bit or register field
In the absence of any indication to the contrary, always write 0 to these bits.

n/a : Not available
In the absence of any indication to the contrary, always write 0 to these bits.

In the absence of any indication to the contrary, always write 0 reserved bits. Writing 1 to a reserved bit can have unintended consequences.

Bits labeled “n/a” have no effect on hardware operation.

Some registers are accessible only under specific conditions. Read/write access is otherwise not allowed.

Table of Contents

1. OVERVIEW

This S1S65010 network camera controller is ideal for applications involving Internet cameras. It incorporates networking and protocol processing support, a camera interface, and a JPEG encoder, so constructing an Internet camera is as simple as connecting a camera module, an external Ethernet PHY device, and a Flash EEPROM containing the firmware.

Shutter commands from clients trigger image capture from the camera and JPEG encoding. Configuring this device as an HTTP server on a LAN, for example, allows clients to request image files. Image capture and transfer to the specified client can be continuous, at a fixed interval controlled by a built-in timer, or in response to a trigger from an interrupt request pin connected to an external sensor or other device. Image transfers can also take the form e-mail attachments.

This device improves upon the S1S65000 with higher frame rates (30 fps at VGA resolution), a higher maximum resolution (UXGA), two I^2S interfaces/modules for voice/audio data, and other new or enhanced functionality for constructing better Internet cameras.

This device allows network devices access to the GPIO ports and I^2C bus for use in specifying camera settings and controlling motors and other external equipment. This product ships with the necessary device drivers.

1.1 Features

- Internet camera operating totally independently of personal computers
- Pin compatibility and upward software compatibility with S1S65000
- Frame rate of 30 fps at VGA resolution
- Support for a broad range of camera modules, right up to 2-megapixel (about 2 million pixels)
- Audio support by I^2S
- Compression to JPEG format with hardware JPEG encoder (ISO 10918 compliant)
- Settings control over network
- E-mail delivery of images files
- Power saving by wake-up mode which reperats start, shooting and pause periodically
- Wireless LAN (802.11b) connectivity using Compact Flash (CF) interface
- Single-chip solution for lower system costs
- Built-in ARM720T Rev 4.3 CPU with 8 KB cache and running at up to 50 MHz

1.2 Internal Functional Blocks

CPU:

- 32-bit RISC ARM720T (maximum clock: 50 MHz)
- Free switching between full 32-bit instruction set and more efficient 16-bit Thumb code
- 31 general-purpose 32-bit registers
- Built-in multiplier

RAM:

- 78 KB of embedded RAM as workspace shared by CPU, JPEG, and Ethernet blocks

Camera Input and JPEG Encoder:

- 8-bit parallel interface using YUV 4-2-2 format
- Image sizes up to UXGA (1600 × 1200): UXGA, SXGA, XGA, VGA, QVGA, CIF, and QCIF
- Support ITU-R BT656 form
- Hardware JPEG encoder
- Max 30 fps at VGA resolution; 30 fps also at CIF resolution
- The pixel clock frequency for the camera data input is less than 2/3 of the CPU clocks.

JPEG Block:

- Hardware JPEG encoder
- Resize (subscreen extraction) function
- Dedicated line buffer
- Built-in variable-capacity FIFO for JPEG encoder output
- Built-in Enhanced DMA

Networking:

- 10/100 Base Ethernet MAC controller supporting both full duplex and half duplex operation
- IEEE 802.3 Clause 22 compliant media independent interface (MII)
- Built-in Enhanced DMA

External Memory Controller:

- 16-bit data bus
- Support for 2 to 128 MB of SDRAM
- Support for up to 16 MB of static memory (Flash EEPROM or SRAM)
- Support for three chapter select signals (SDRAM, Flash, plus one other)

CF Card Interface:

- Compliant with CF+ Specifications Rev 1.4
- Adaptable for use as interface to wireless LAN, PHS card, and other devices
- Support for true IDE mode

Standby Operation:

- HALT mode suspending the clock signal to the CPU when the latter is not needed
- I/O clock control suspending the clock signals to major I/O blocks

Timer and Watchdog Timer:

- Three 16-bit timers
- Choice of reload/cyclic or one-shot operation
- Support for toggle or port output upon underflow output
- Watchdog timer triggering interrupt request or reset signal

Serial Interfaces:

- UART: Software compatible with the 16550 interface
- UART Lite: Limited subset of 16550 software interface
- SPI: Clock synchronous interface
- I^{2}C master interface for camera interface and general-purpose applications
- Two I^{2}S interfaces/modules for audio data, compliant with the Philips I^{2}S standard

Interrupt Controller:

- Support for two fast (FIQ) and 32 normal (IRQ) interrupt requests

Real-Time Clock:

- Supports days, hours, minutes, and seconds
- Internal timer taps (1/128 to 1/2) for use as interrupt request sources
- Support for alarms and interrupt requests

GPIO:

- General-purpose I/O ports (maximum 57)
- Programmable I/O direction for all port pins
- Alternate I/O functions available for some port pins

Power Supplies:

• 3.3 V	I/O power supply
• 1.8 V	Core power supply
• 1.8 V	PLL analog power supply
• 2.4 V to 3.6 V	Camera I/O power supply

Package:

- 144-pin TQFP (TQFP24), 16 × 16 × 1 mm, 0.4 mm pin pitch

1.3 Supported Protocols

ARP, ICMP, IP, TCP, UDP, HTTPd, SMTP, DHCP, FTP, DNS resolver, telnet

Necessary protocols can be added or updated by rewriting Flash ROM.
Addition or update by the customer is also possible.

Protocols are prepared as EPSON’s sample software or partner’s products.

2. BLOCK DIAGRAM

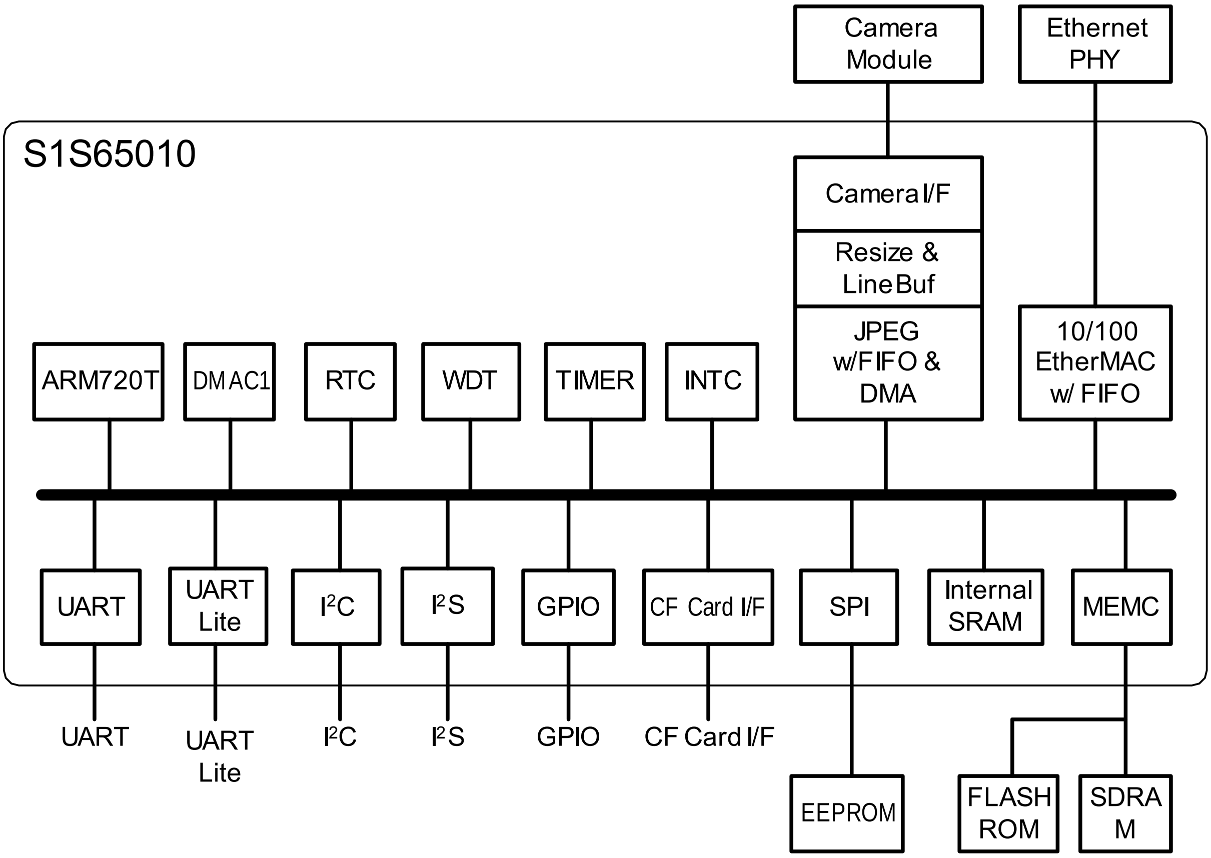


Fig.2.1 S1S65010 Block Diagram

3. PINS

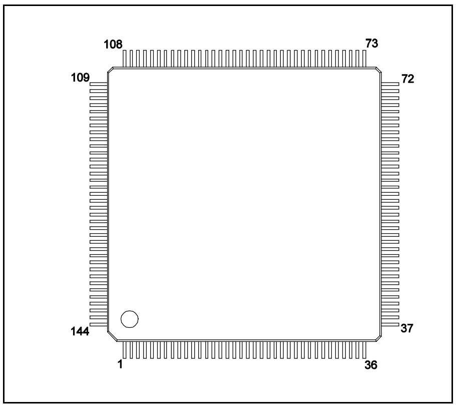


Fig.3.1 Pin Layout (Top View)

Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name
1	MA14	37	MD12	73	TRST#	109	CMDATA5
2	MA15	38	MD13	74	TCK	110	CMDATA6
3	MA16	39	MD14	75	TMS	111	CMDATA7
4	MA17	40	MD15	76	TDI	112	VSS
5	MA18	41	MDQML	77	TDO	113	LVDD
6	VSS	42	MDQMH	78	VSS	114	GPIOD0
7	MA19	43	HVDD1	79	GPIOA0	115	GPIOD1
8	MCS2#	44	VSS	80	GPIOA1	116	CFCE2#
9	MCS1#	45	MII_CRS	81	GPIOA2	117	CFCE1#
10	MCS0#	46	MII_COL	82	GPIOA3	118	CFIORD#
11	LVDD	47	MII_TXD3	83	GPIOA4	119	CFIOWR#
12	MOE#	48	MII_TXD2	84	GPIOA5	120	CFIREQ
13	MWE0#	49	MII_TXD1	85	GPIOA6	121	CFRST
14	MWE1#	50	LVDD	86	GPIOA7	122	VSS
15	HVDD1	51	MII_TXD0	87	HVDD1	123	HVDD1
16	MCLKEN	52	MII_TXEN	88	VSS	124	CFWAIT#
17	MCLK	53	MII_TXCLK	89	GPIOB0	125	CFSTSCHG#
18	VSS	54	MII_RXER	90	GPIOB1	126	CFDEN#
19	MRAS#	55	VSS	91	GPIOB2	127	CFDDIR
20	MCAS#	56	HVDD1	92	GPIOB3	128	MA0
21	MD0	57	MII_RXCLK	93	LVDD	129	MA1
22	MD1	58	MII_RXDV	94	GPIOB4	130	MA2
23	MD2	59	MII_RXD0	95	GPIOB5	131	MA3
24	MD3	60	MII_RXD1	96	GPIOB6	132	VSS
25	VSS	61	LVDD	97	GPIOB7	133	LVDD
26	LVDD	62	MII_RXD2	98	VSS	134	MA4
27	MD4	63	MII_RXD3	99	CMHREF	135	MA5
28	MD5	64	MII_MDC	100	CMVREF	136	MA6
29	MD6	65	MII_MDIO	101	CMCLKIN	137	MA7
30	MD7	66	VSS	102	CMCLKOUT	138	MA8
31	HVDD1	67	CLKI	103	CMDATA0	139	HVDD1
32	MD8	68	PLLVSS	104	CMDATA1	140	MA9
33	MD9	69	VCP	105	HVDD2	141	MA10
34	MD10	70	PLLVDD	106	CMDATA2	142	MA11
35	MD11	71	RESET#	107	CMDATA3	143	MA12
36	VSS	72	TESTEN	108	CMDATA4	144	MA13

Note: A sharp (#) to the right of the pin name indicates an active low signal.

3. PINS

3.1 Pin Descriptions

#: This symbol to the right of the pin name indicates a low active signal.
I: Input pin
O: Output pin
I/O: Bidirectional pin
P: Power supply pin

Table 3.1 Cell Types

Cell Type	Description	Pin Examples
ICS	LVCMOS Schmitt input	TCK, CLKI, RESET#
ICD1	LVCMOS input with pull-down resistor (50kΩ@3.3V)	TESTEN
ICU1	LVCMOS input with pull-up resistor (50kΩ@3.3V)	TMS, TDI
ICSU1	LVCMOS Schmitt input with pull-up resistor (50kΩ@3.3V)	TRST#
BLNC4	Low noise LVCMOS IO buffer (±4mA)	MII
BLNC4U1	Low noise LVCMOS IO buffer with pull-up resistor (50kΩ@3.3V) (±4mA)	CF interface
BLNC4D2	Low noise LVCMOS IO buffer with pull-down resistor (100kΩ@3.3V) (±4mA)	MD [15:0]
BLNS4	Low noise LVCMOS Schmitt IO buffer (±4mA)	GPIOA, GPIOB, GPIOD [1:0]
BLNS4D1	Low noise LVCMOS Schmitt IO buffer with pull-down resistor (50kΩ@3.3V) (±4mA)	Camera interface
OLN4	Low noise output buffer (±4mA)	MEMC interface (except MD pins)
OTLN4	Low noise Tri-state output buffer (±4mA)	TDO
OLTR	Low Voltage Transparent Output	VCP

Table 3.2 Pin Descriptions

Pin Name	Type	Cell Type	Pin No.	Description
(MA [23:22])	(I/O)	(BLNS4)	(97-96)	For further details on these pins, see the GPIOB[7:6] description.
(MA [21:20])	(I/O)	(BLNS4)	(114-115)	For further details on these pins, see the GPIOD[1:0] description.
MA [19:12]	O	OLN4	7, 1-5, 143-144	Address outputs 19 to 12 SDRAM uses MA[15:14] as the bank address (BA[1:0]).
MA 11	O	OLN4	142	This pin has more than one function. • MA11: Address output 11 (default pin function after reset) • CFREG#: Compact Flash (CF) interface REG signal specifying CF interface attribute and selecting I/O space
MA [10:0]	O	OLN4	128-131, 134-138, 140-141	These pins have more than one function. • MA[10:0]: Address outputs 10 to 0 (default pin function after reset) • CFADDR[10:0]: CF interface address outputs 10 to 0
MD [15:0]	I/O	BLNC4D2	21-24, 27-30, 32-35, 37-40	These pins have more than one function. • 16-bit Data bus to memory (default pin function after reset) • 16-bit Data bus to CF interface • MODESEL[15:0]: These input levels determine the internal operation mode. Sampling is at the end of a power on reset, when RESET# returns to High level from Low level. External pull-up resistances (approximately 4.7 to 10 kΩ) may therefore be necessary. For further details, see Section 4.1 “System Configuration.”
MCS [2:0]#	O	OLN4	8-10	Chip select signals for memory (SDRAM or static) MCS2# is for SDRAM.
MOE#	O	OLN4	12	This pin has more than one function. • MOE#: Memory output strobe signal (default pin function after reset) • CFOE#: CF interface output enable signal for attribute and common memory space
MWE0#	O	OLN4	13	This pin has more than one function. • MWE0#: Memory write enable signal for static memory (default pin function after reset) • CFWE#: CF interface write enable signal for attribute and common memory space
MWE1#	O	OLN4	14	Memory write Enable signal for SDRAM
MCLK	O	OLN4	17	SDRAM clock output This output has the same frequency as the internal operating clock (CPUCLK) signal.
MCLKEN	O	OLN4	16	Clock Enable signal for SDRAM
MRAS#	O	OLN4	19	RAS signal for SDRAM
MCAS#	O	OLN4	20	CAS signal for SDRAM
MDQML MDQMH	O	OLN4	41-42	These pins have more than one function. • Byte Enable signals for static memory • DQM signals for SDRAM MDQML accesses the lower bytes; MDQMH, the upper ones.

Pin Name	Type	Cell Type	Pin No.	Description
MII_TXCLK	I/O	BLNC4	53	This pin has more than one function. • MII_TXCLK: Media independent interface Ethernet PHY (MII PHY) transmit data output clock signal (TXCLK) input (non-GPIO function #1, the default pin function after reset) • GPIOF7 I/O
MII_TXEN	I/O	BLNC4	52	This pin has the following functions. • MII_TXEN: MII PHY sending output enable TXEN output （Pin function just after reset: Function 1 other than GPIO) • GPIOF6 I/O
MII_TXD3	I/O	BLNC4	47	This pin has the following functions. • MII_TXD3: MII PHY sending data TXD3 output (Pin function just after reset: Function 1 other than GPIO) • GPIOF2 I/O
MII_TXD2	I/O	BLNC4	48	This pin has the following functions. • MII_TXD2: MII PHY sending data TXD2 output (Pin function just after reset: Function 1 other than GPIO) • GPIOF3 I/O
MII_TXD1	I/O	BLNC4	49	This pin has the following functions. • MII_TXD1: MII PHY sending data TXD1 output (Pin function just after reset: Function 1 other than GPIO) • GPIOF4 I/O
MII_TXD0	I/O	BLNC4	51	This pin has the following functions. • MII_TXD0: MII PHY sending data TXD0 output (Pin function just after reset: Function 1 other than GPIO) • GPIOF5 I/O
MII_RXCLK	I/O	BLNC4	57	This pin has the following functions. • MII_RXCLK: MII PHY receiving data lock (RXCLK) input (Pin function just after reset: Function 1 other than GPIO) • GPIOG1I/O
MII_COL	I/O	BLNC4	46	This pin has the following functions. • MII_COL: MII PHY collision (COL) detection input (Pin function just after reset: Function 1 other than GPIO) • GPIOF1 I/O
MII_CRS	I/O	BLNC4	45	This pin has the following functions. • MII_CRS: MII PHY carrier sense (CRS) input (Pin function just after reset: Function 1 other than GPIO) • GPIOF0 I/O
MII_RXDV	I/O	BLNC4	58	This pin has the following functions. • MII_RXDV: MII PHY receiving data value (RXDV) input (Pin function just after reset: Function 1 other than GPIO) • GPIOG2 I/O
MII_RXD [3:0]	I/O	BLNC4	59-60, 62-63	This pin has the following functions. • MII_RXD[3:0]: MII PHY receiving data RXD[3:0] input (Pin function just after reset: Function 1 other than GPIO) • GPIOG[6:3] I/O
MII_RXER	I/O	BLNC4	54	This pin has more than one function. • MII_RXER: MII physical layer receive error (RXER) input (non-GPIO function #1, the default pin function after reset) • GPIOG0 I/O
MII_MDC	I/O	BLNC4	64	This pin has more than one function. • MII_MDC: MII physical layer management interface clock (MDC) output (non-GPIO function #1, the default pin function after reset) • GPIOG7 I/O

Pin Name	Type	Cell Type	Pin No.	Description
MII_MDIO	I/O	BLNC4	65	This pin has more than one function. • MII_MDIO: MII physical layer management interface data (MDIO) I/O (non-GPIO function #1, the default pin function after reset) • GPIOH0 I/O
CMDATA[7:0]	I/O	BLNS4D1	103-104, 106-111	These pins have more than one function. • CMDATA[7:0]: Camera YUV Data input These pins are set as GPIOC[7:0] inputs after reset. To use them for this alternate function, specify “non-GPIO function #1” in GPIOC Pin Function Register bits 15 to 0. • GPIOC[7:0] I/O (default pin function after reset)
CMVREF	I/O	BLNS4D1	100	This pin has more than one function. • CMVREF: Vertical synchronization input for camera Data input This pin is set as GPIOD4 input after reset. To use it for this alternate function, specify “non-GPIO function #1” in GPIOD Pin Function Register bits 9 to 8. • GPIOD4 I/O (default pin function after reset)
CMHREF	I/O	BLNS4D1	99	This pin has more than one function. • CMHREF: Horizontal synchronization input for camera Data input This pin is set as GPIOD5 input after reset. To use it for this alternate function, specify “non-GPIO function #1” in GPIOD Pin Function Register bits 11 to 10. • GPIOD5 I/O (default pin function after reset)
CMCLKOUT	I/O	BLNS4D1	102	This pin has more than one function. • CMCLKOUT: Basic clock output for camera This pin is set as GPIOD6 input after reset. To use it for this alternate function, specify “non-GPIO function #1” in GPIOD Pin Function Register bits 13 to 12. • GPIOD6 I/O (default pin function after reset)
CMCLKIN	I/O	BLNS4D1	101	This pin has more than one function. • CMCLKIN: Pixel clock for camera Data input This pin is set as GPIOD7 input after reset. To use it for this alternate function, specify “non-GPIO function #1” in GPIOD Pin Function Register bits 15 to 14. • GPIOD7 I/O (default pin function after reset)
CFCE2#	I/O	BLNC4U1	116	This pin has more than one function. • CFCE2#: Compact Flash (CF) memory interface card Enable 2 (CE2#) output This pin is set as GPIOD2 input after reset. To use it for this alternate function, specify “non-GPIO function #1” in GPIOD Pin Function Register bits 5 to 4. • GPIOD2 I/O (default pin function after reset)
CFCE1#	I/O	BLNC4U1	117	This pin has more than one function. • CFCE1#: CF card Enable 1 (CE1#) output This pin is set as GPIOD3 input after reset. To use it for this alternate function, specify “non-GPIO function #1” in GPIOD Pin Function Register bits 7 to 6. • GPIOD3 I/O (default pin function after reset)

Pin Name	Type	Cell Type	Pin No.	Description
CFIORD#	I/O	BLNC4U1	118	This pin has more than one function. • CFIORD#: CF IO read strobe output This pin is set as GPIOE0 input after reset. To use it for this alternate function, specify "non-GPIO function #1" in GPIOE Pin Function Register bits 1 to 0. • GPIOE0 I/O (default pin function after reset) • I2S0_SD: I2S0 serial data (non-GPIO function #2)
CFIOWR#	I/O	BLNC4U1	119	This pin has more than one function. • CFIOWR#: CF IO write strobe output This pin is set as GPIOE1 input after reset. To use it for this alternate function, specify "non-GPIO function #1" in GPIOE Pin Function Register bits 3 to 2. • GPIOE1 I/O (default pin function after reset) • I2S0_SCK: I2S0 serial clock (non-GPIO function #2)
CFWAIT#	I/O	BLNC4U1	124	This pin has more than one function. • CFWAIT#: CF wait request This pin is set as GPIOE2 input after reset. To use it for this alternate function, specify "non-GPIO function #1" in GPIOE Pin Function Register bits 5 to 4. • MWAIT#: Memory controller wait signal This signal shares the same pin as CFWAIT# (non-GPIO function #1). • GPIOE2 I/O (default pin function after reset)
CFRST	I/O	BLNC4U1	121	This pin has more than one function. • CFRST: Reset signal to the CF card This signal is at High level during a card reset and at Low level during normal card operation. This pin is set as GPIOE3 input after reset. To use it for this alternate function, specify "non-GPIO function #1" in GPIOE Pin Function Register bits 7 to 6. • GPIOE3 I/O (default pin function after reset) • I2S0_WS: I2S0 word select (non-GPIO function #2)
CFIREQ	I/O	BLNC4U1	120	This pin has more than one function. • CFIREQ: Interrupt request signal from CF card This pin is set as GPIOE4 input after reset. To use it for this alternate function, specify "non-GPIO function #1" in GPIOE Pin Function Register bits 9 to 8. • GPIOE4 I/O (default pin function after reset)
CFSTSCHG#	I/O	BLNC4U1	125	This pin has more than one function. • CFSTSCHG#: Status change signal from CF card This pin is set as GPIOE5 input after reset. To use it for this alternate function, specify "non-GPIO function #1" in GPIOE Pin Function Register bits 11 to 10. • GPIOE5 I/O (default pin function after reset) • I2S1_SD: I2S1 serial data (non-GPIO function #2)

Pin Name	Type	Cell Type	Pin No.	Description
CFDEN#	I/O	BLNC4U1	126	This pin has more than one function. • CFDEN#: Data bus Enable signal for CF card external buffer This pin is set as GPIOE6 input after reset. To use it for this alternate function, specify "non-GPIO function #1" in GPIOE Pin Function Register bits 13 to 12. • GPIOE6 I/O pin (default pin function after reset) • I2S1_SCK: I2S1 serial clock (non-GPIO function #2)
CFDDIR	I/O	BLNC4U1	127	This pin has more than one function. • CFDDIR: CF Data bus direction indicator output This pin goes to Low level during CF data reads. This pin is set as GPIOE7 input after reset. To use it for this alternate function, specify "non-GPIO function #1" in GPIOE Pin Function Register bits 15 to 14. • GPIOE7 I/O (default pin function after reset) • I2S1_WS: I2S1 word select (non-GPIO function #2)
GPIOA0	I/O	BLNS4	79	This pin has more than one function. • GPIOA0 I/O (default pin function after reset) • TXD0: UART transmit Data output (non-GPIO function #1)
GPIOA1	I/O	BLNS4	80	This pin has more than one function. • GPIOA1 I/O (default pin function after reset) • RXD0: UART receive Data input (non-GPIO function #1)
GPIOA2	I/O	BLNS4	81	This pin has more than one function. • GPIOA2 I/O (default pin function after reset) • SPI_SS: SPI chip select (non-GPIO function #1) • TXD1: UART Lite transmit Data output (non-GPIO function #2)
GPIOA3	I/O	BLNS4	82	This pin has more than one function. • GPIOA3 I/O (default pin function after reset) • SPI_SCLK: SPI serial clock (non-GPIO function #1) • RXD1: UART Lite receive Data input (non-GPIO function #2)
GPIOA4	I/O	BLNS4	83	This pin has more than one function. • GPIOA4 I/O (default pin function after reset) • SPI_MISO: SPI serial Data master input and slave output (non-GPIO function #1)
GPIOA5	I/O	BLNS4	84	This pin has more than one function. • GPIOA5 I/O (default pin function after reset) • SPI_MOSI: SPI serial Data master output and slave input (non-GPIO function #1)
GPIOA6	I/O	BLNS4	85	This pin has more than one function. • GPIOA6 I/O (default pin function after reset) • SCL: I2C clock I/O (non-GPIO function #1)
GPIOA7	I/O	BLNS4	86	This pin has more than one function. • GPIOA7 I/O (default pin function after reset) • SDA: I2C Data I/O (non-GPIO function #1)
GPIOB0	I/O	BLNS4	89	This pin has more than one function. • GPIOB0 I/O (default pin function after reset) • INT0 input • I2S0_WS: I2S0 word select (non-GPIO function #2)
GPIOB1	I/O	BLNS4	90	This pin has more than one function. • GPIOB1 I/O (default pin function after reset) • INT1 input • RTS0#: UART transmit request output (non-GPIO function #1) • I2S0_SCK: I2S0 serial clock (non-GPIO function #2)

Pin Name	Type	Cell Type	Pin No.	Description
GPIOB2	I/O	BLNS4	91	This pin has more than one function. • GPIOB2 I/O (default pin function after reset) • INT2 input • CTS0#: UART clear to send input (non-GPIO function #1) • I2S0_SD: I2S0 serial data (non-GPIO function #2)
GPIOB3	I/O	BLNS4	92	This pin has more than one function. • GPIOB3 I/O (default pin function after reset) • INT3 input • Timer 0 output (non-GPIO function #1) • I2S1_SD: I2S1 serial data (non-GPIO function #2)
GPIOB4	I/O	BLNS4	94	This pin has more than one function. • GPIOB4 I/O (default pin function after reset) • INT4 input • Timer 1 output (non-GPIO function #1)
GPIOB5	I/O	BLNS4	95	This pin has more than one function. • GPIOB5 I/O (default pin function after reset) • INT5 input • Timer 2 output (non-GPIO function #1)
GPIOB6	I/O	BLNS4	96	This pin has more than one function. • GPIOB6 I/O (default pin function after reset) • INT6 input • MA22: Address output 22 (non-GPIO function #1) • I2S1_SCK: I2S1 serial clock (non-GPIO function #2)
GPIOB7	I/O	BLNS4	97	This pin has more than one function. • GPIOB7 I/O (default pin function after reset) • INT7 input • MA23: Address output 23 (non-GPIO function #1) • I2S1_WS: I2S1 word select (non-GPIO function #2)
GPIOD0	I/O	BLNS4	114	This pin has more than one function. • GPIOD0 I/O (default pin function after reset) • INT8 input • MA20: Address output 20 (non-GPIO function #1)
GPIOD1	I/O	BLNS4	115	This pin has more than one function. • GPIOD1 I/O (default pin function after reset) • MA21: Address output 21 (non-GPIO function #1)
CLKI	I	ICS	67	32 kHz clock input This is the basic input clock signal for this device. The internal PLL multiplies this frequency to dozens of MHz for use as the basic clock signal for internal operation. This features Schmitt trigger input.
VCP	O	OLTR	69	Built-in PLL test pin This is for monitoring PLL output for testing purposes. Leave it open during normal use.
TRST#	I	ICSU1	73	JTAG interface reset This features Schmitt trigger input with pull-up resistance.
TCK	I	ICS	74	JTAG interface clock input This features Schmitt trigger input.
TMS	I	ICU1	75	JTAG interface TMS input This pin includes a built-in pull-up resistance.
TDI	I	ICU1	76	JTAG interface serial Data input This pin includes a built-in pull-up resistance.
TDO	O	OTLN4	77	JTAG interface serial Data output

Pin Name	Type	Cell Type	Pin No.	Description
TESTEN	I	ICD1	72	Test Enable (high active signal) This pin includes a built-in pull-down resistance. Connect this to VSS or leave it open during normal use.
RESET#	I	ICS	71	System reset signal Keep this input active (at Low level) for at least 100 ms after HVDD1 and LVDD have stabilized.
HVDD1	P	P	15, 31, 43, 56, 87, 123, 139	3.3 V power supply for all I/O cells except for the camera interface
HVDD2	P	P	105	3.0 V (Typical) power supply for the camera interface The supported range is 2.4 V to 3.6 V
LVDD	P	P	11, 26, 50, 61, 93, 113, 133	1.8 V power supply for core (internal circuitry)
PLLVDD	P	P	70	1.8 V analog power supply for PLL This must be treated as an analog power supply. Connect a stable power supply relatively free from noise.
PLLVSS	P	P	68	Analog ground for PLL This must be treated as an analog power supply. Connect a stable ground relatively free from noise.
VSS	P	P	6, 18, 25, 36, 44, 55, 66, 78, 88, 98, 112, 122, 132	Common grounds shared by I/O cells, the camera interface, and the core power supply

3.2 Multiplexed Function of GPIO Pins

Pin Name	Pin Function after Reset	GPIO	INT	Address Bus	UART/ UARTL	I2C	SPI / I2S	Timers	Camera Interface	CF Card Interface	MII PHY Interface
GPIOA0	GPIOA0	GPIOA0			TXD0						
GPIOA1	GPIOA1	GPIOA1			RXD0						
GPIOA2	GPIOA2	GPIOA2			TXD1		SPI_SS				
GPIOA3	GPIOA3	GPIOA3			RXD1		SPI_SCLK				
GPIOA4	GPIOA4	GPIOA4					SPI_MISO				
GPIOA5	GPIOA5	GPIOA5					SPI_MOSI				
GPIOA6	GPIOA6	GPIOA6				SCL					
GPIOA7	GPIOA7	GPIOA7				SDA					
GPIOB0	GPIOB0	GPIOB0	INT0				I2S0_WS				
GPIOB1	GPIOB1	GPIOB1	INT1		RTS0#		I2S0_SCK				
GPIOB2	GPIOB2	GPIOB2	INT2		CTS0#		I2S0_SD				
GPIOB3	GPIOB3	GPIOB3	INT3				I2S1_SD	Timer0out			
GPIOB4	GPIOB4	GPIOB4	INT4					Timer1out			
GPIOB5	GPIOB5	GPIOB5	INT5					Timer2out			
GPIOB6	GPIOB6	GPIOB6	INT6	MA22			I2S1_SCK				
GPIOB7	GPIOB7	GPIOB7	INT7	MA23			I2S1_WS				
CMDATA0	GPIOC0	GPIOC0							CMDATA0		
CMDATA1	GPIOC1	GPIOC1							CMDATA1		
CMDATA2	GPIOC2	GPIOC2							CMDATA2		
CMDATA3	GPIOC3	GPIOC3							CMDATA3		
CMDATA4	GPIOC4	GPIOC4							CMDATA4		
CMDATA5	GPIOC5	GPIOC5							CMDATA5		
CMDATA6	GPIOC6	GPIOC6							CMDATA6		
CMDATA7	GPIOC7	GPIOC7							CMDATA7		
GPIOD0	GPIOD0	GPIOD0	INT8	MA20							
GPIOD1	GPIOD1	GPIOD1		MA21							
CFCE2#	GPIOD2	GPIOD2								CFCE2#	
CFCE1#	GPIOD3	GPIOD3								CFCE1#	
CMVREF	GPIOD4	GPIOD4							CMVREF		
CMHREF	GPIOD5	GPIOD5							CMHREF		
CMCLKOUT	GPIOD6	GPIOD6							CMCLKOUT		
CMCLKIN	GPIOD7	GPIOD7							CMCLKIN		
CFIORD#	GPIOE0	GPIOE0					I2S0_SD			CFIORD#	
CFIOWR#	GPIOE1	GPIOE1					I2S0_SCK			CFIOWR#	
CFWAIT#	GPIOE2	GPIOE2								CFWAIT# / MWAIT#	
CFRST	GPIOE3	GPIOE3					I2S0_WS			CFRST	
CFIREQ	GPIOE4	GPIOE4								CFIREQ	
CFSTSCHG#	GPIOE5	GPIOE5					I2S1_SD			CFSTSCHG#	
CFDEN#	GPIOE6	GPIOE6					I2S1_SCK			CFDEN#	
CFDDIR	GPIOE7	GPIOE7					I2S1_WS			CFDDIR	
MII_CRS	MII_CRS	GPIOF0									MII_CRS
MII_COL	MII_COL	GPIOF1									MII_COL
MII_TXD3	MII_TXD3	GPIOF2									MII_TXD3
MII_TXD2	MII_TXD2	GPIOF3									MII_TXD2
MII_TXD1	MII_TXD1	GPIOF4									MII_TXD1
MII_TXD0	MII_TXD0	GPIOF5									MII_TXD0
MII_TXEN	MII_TXEN	GPIOF6									MII_TXEN
MII_TXCLK	MII_TXCLK	GPIOF7									MII_TXCLK
MII_RXER	MII_RXER	GPIOG0									MII_RXER
MII_RXCLK	MII_RXCLK	GPIOG1									MII_RXCLK
MII_RXDV	MII_RXDV	GPIOG2									MII_RXDV
MII_RXD0	MII_RXD0	GPIOG3									MII_RXD0
MII_RXD1	MII_RXD1	GPIOG4									MII_RXD1
MII_RXD2	MII_RXD2	GPIOG5									MII_RXD2
MII_RXD3	MII_RXD3	GPIOG6									MII_RXD3
MII_MDC	MII_MDC	GPIOG7									MII_MDC
MII_MDIO	MII_MDIO	GPIOH0									MII_MDIO

The lighter shading indicates non-GPIO function #1; the darker, non-GPIO function #2.

░ : Function 1 ▓ : Function 2

3.3 Pin Configurations During and After a Reset

Pin Name	I/O Direction During Reset	Level During Reset	Built-In Resistances	Description
MA[19:0]	Output	Low (However, only bit11 is High.)	None	
MD[15:0]	Input	Low	Pull-down resistance	100kΩ
MCS[2]#	Output	Low	None	
MCS[1]#	Output	High	None	
MCS[0]#	Output	High	None	
MOE#	Output	High	None	
MWE0#	Output	High	None	
MWE1#	Output	Low	None	
MCLK	Output	MCLK(32KHZ)	None	
MCLKEN	Output	High	None	
MRAS#	Output	Low	None	
MCAS#	Output	High	None	
MDQML	Output	Low	None	
MDQMH	Output	Low	None	
MII_TXCLK	Input	High-Z	None	Depends on external circuitry (normally MII-PHY)
MII_TXEN	Output	Low	None	
MII_TXD[3:0]	Output	Undefined	None	Undefined until initialized
MII_RXCLK	Input	High-Z	None	Depends on external circuitry (normally MII-PHY)
MII_COL	Input	High-Z	None	Depends on external circuitry (normally MII-PHY)
MII_CRS	Input	High-Z	None	Depends on external circuitry (normally MII-PHY)
MII_RXDV	Input	High-Z	None	Depends on external circuitry (normally MII-PHY)
MII_RXD[3:0]	Input	High-Z	None	Depends on external circuitry (normally MII-PHY)
MII_RXER	Input	High-Z	None	Depends on external circuitry (normally MII-PHY)
MII_MDC	Output	Low	None	
MII_MDIO	Input	High-Z	None	Depends on external circuitry (normally MII-PHY)
CMDATA[7:0]	Input	Low	Pull-down resistance	50kΩ
CMVREF	Input	Low	Pull-down resistance	50kΩ
CMHREF	Input	Low	Pull-down resistance	50kΩ
CMCLKOUT	Input	Low	Pull-down resistance	50kΩ
CMCLKIN	Input	Low	Pull-down resistance	50kΩ
CFCE2#	Input	High	Pull-up resistance	50kΩ
CFCE1#	Input	High	Pull-up resistance	50kΩ
CFIORD#	Input	High	Pull-up resistance	50kΩ
CFIOWR#	Input	High	Pull-up resistance	50kΩ
CFWAIT#	Input	High	Pull-up resistance	50kΩ
CFRST	Input	High	Pull-up resistance	50kΩ
CFIREQ	Input	High	Pull-up resistance	50kΩ
CFSTSCHG#	Input	High	Pull-up resistance	50kΩ
CFDEN#	Input	High	Pull-up resistance	50kΩ
CFDDIR	Input	High	Pull-up resistance	50kΩ
GPIOA[7:0]	Input	High-Z	None	Depends on external circuitry
GPIOB[7:0]	Input	High-Z	None	Depends on external circuitry
GPIOD[1:0]	Input	High-Z	None	Depends on external circuitry
CLKI	Input	High-Z	None	
VCP	Output	High-Z	None	Leave open
TRST#	Input	High	Pull-up resistance	50kΩ
TCK	Input	High-Z	None	
TMS	Input	High	Pull-up resistance	50kΩ
TDI	Input	High	Pull-up resistance	50kΩ
TDO	Output	High-Z	None	
TESTEN	Input	Low	Pull-down resistance	50kΩ
RESET#	Input	Low	None	

The following RESET value is decided depending on the content of each pin set.

4. FUNCTIONAL DESCRIPTION

4.1 System Configuration

At the end of a power on reset, when RESET# returns to High level from Low level, this device samples pin input levels from the MODESEL[15:0] pins, an alternate function for the MD bus pins (MD[15:0]), to determine internal configuration parameters.

Built-in pull-down resistances (typ. 100 kΩ) produce Low as the default for MD bus inputs. The developer need only provide pull-up resistances or other external components when High level inputs become necessary. Note, however, that adding such external pull-up resistances creates a constant current through the corresponding internal pull-down resistances. Detaching the internal pull-down resistance in software eliminates such constant current.

For further details, see Section 13 "System Controller."

Table 4.1 System Configuration Pins (MODESEL[15:0])

Pin Name	Pin Function	Level After Reset	
		Low	High
MD0	MODESEL0	32kHz Mode	Reserved for testing*
MD1	MODESEL1	Insert crystal oscillator stabilization interval (3 seconds)	Reserved for testing*
MD2	MODESEL2	Normal operation	Reserved for testing*
MD3	MODESEL3	Reserved (Always use Low.)	
MD4	MODESEL4	User settings	User settings
MD5	MODESEL5	User settings	User settings
MD6	MODESEL6	User settings	User settings
MD7	MODESEL7	User settings	User settings
MD8	MODESEL8	User settings	User settings
MD9	MODESEL9	User settings	User settings
MD10	MODESEL10	User settings	User settings
MD11	MODESEL11	User settings	User settings
MD12	MODESEL12	User settings	User settings
MD13	MODESEL13	User settings	User settings
MD14	MODESEL14	User settings	User settings
MD15	MODESEL15	User settings	User settings

* Using Settings "reserved for testing" risks permanently damaging this device.

MD0: **Clock input**

Low: 32 kHz (PLL input)

High: Reserved for testing (Do not use!)

MD1: **Crystal oscillator stabilization interval**

This setting controls the use of the crystal oscillator stabilization interval and thus determines how long the circuitry waits before starting the system (CPU) after a reset (power on or otherwise).

Low: Insert interval (3 seconds)

High: Reserved for testing (Do not use!)

MD2: **Clock signal monitor output**

Low: Normal operation

High: Reserved for testing (Do not use!)

MD3: **Reserved for testing. Always set to "Low."**

MD [15:4]: **These 12 pins are available for user applications via the corresponding bits in the system controller's chip configuration register.**

4.2 Memory Maps

After a reset, this device has the following memory maps for the ARM720T's 4 GB address space.

4.2.1 AHB1 Memory Map

The following is the memory map for AHB1, the AHB bus connected to the CPU.

Table 4.2 AHB1 Memory Map

Start Address	End Address	Size (MB)	Devices	External Chip Select	Device Bus Sizes (Bits)
0x0000_0000	0x07FF_FFFF	128	External ROM/SRAM	CS0/CS1	16
0x0800_0000	0x0FFF_FFFF	128	Reserved		
0x1000_0000	0x1FFF_FFFF	256	Reserved		
0x2000_0000	0x2FFF_FFFF	256	Internal SRAM		32
0x3000_0000	0x37FF_FFFF	128	External SDRAM	CS2	16
0x3800_0000	0x3FFF_FFFF	128	Reserved		
0x4000_0000	0x4FFF_FFFF	256	Reserved		
0x5000_0000	0x5FFF_FFFF	256	Reserved		
0x6000_0000	0x6FFF_FFFF	256	Reserved		
0x7000_0000	0x7FFF_FFFF	256	Reserved		
0x8000_0000	0x8FFF_FFFF	256	Reserved		
0x9000_0000	0x9FFF_FFFF	256	Reserved		
0xA000_0000	0xAFFF_FFFF	256	Reserved		
0xB000_0000	0xBFFF_FFFF	256	Reserved		
0xC000_0000	0xC7FF_FFFF	128	External ROM	CS0/CS1	16
0xC800_0000	0xCFFF_FFFF	128	Reserved		
0xD000_0000	0xDFFF_FFFF	256	Reserved		
0xE000_0000	0xEFFF_FFFF	256	Reserved		
0xF000_0000	0xFFFF_FFFF	256	Internal I/O area		32/16/8

CS0 connects to the boot device. The internal SRAM occupies only a fraction of the 256 MB space available. The remainder mirrors the internal SRAM contents over and over again.

Addresses 0xC000_0000 to 0xCFFF_FFFF mirror those starting at 0x0000_0000. In other words, this block is an alias for the device in the first block. Accessing this block actually accesses that block.

The internal I/O devices are at the top end of memory. Table 4.4 "Internal I/O Map" gives the layout in more detail.

4.2.2 AHB2 Memory Map

The following is the memory map for AHB2, the other AHB bus.

Table 4.3 AHB2 Memory Map

Start Address	End Address	Size (MB)	Devices	External Chip Select	Device Bus Sizes (Bits)
0x0000_0000	0x07FF_FFFF	128	External ROM/SRAM	CS0/CS1	16
0x0800_0000	0x0FFF_FFFF	128	Reserved		
0x1000_0000	0x1FFF_FFFF	256	Reserved		
0x2000_0000	0x2FFF_FFFF	256	Internal SRAM		32
0x3000_0000	0x37FF_FFFF	128	External SDRAM	CS2	16
0x3800_0000	0x3FFF_FFFF	128	Reserved		
0x4000_0000	0x4FFF_FFFF	256	Reserved		
0x5000_0000	0x5FFF_FFFF	256	Reserved		
0x6000_0000	0x6FFF_FFFF	256	Reserved		
0x7000_0000	0x7FFF_FFFF	256	Reserved		
0x8000_0000	0x8FFF_FFFF	256	Reserved		
0x9000_0000	0x9FFF_FFFF	256	Reserved		
0xA000_0000	0xAFFF_FFFF	256	Reserved		
0xB000_0000	0xBFFF_FFFF	256	Reserved		
0xC000_0000	0xC7FF_FFFF	128	External ROM/SRAM	CS0/CS1	16
0xC800_0000	0xCFFF_FFFF	128	Reserved		
0xD000_0000	0xDFFF_FFFF	256	Reserved		
0xE000_0000	0xEFFF_FFFF	256	JPEG DMA Port		32
0xF000_0000	0xFFFF_FFFF	256	Reserved		

The AHB1 and AHB2 memory maps show how bus masters both AHB buses share, as common resources, all memory connected to the external memory controller: the external ROM, SDRAM, and SRAM selected with the CS[2:0] signals as well as the internal SRAM.

4.3 I/O Map

The following Table lists the portions of the 256 MB internal I/O area (0xF000_0000 to 0xFFFF_FFFF) actually used.

The gaps labeled reserved do not contain devices. Reads return undefined data.

Table 4.4 Internal I/O Map

Base Address	Size (KB)	Description
0xFFFD_0000	64	Reserved
0xFFFE_0000	4	APB bridge
0xFFFE_1000	4	Reserved
0xFFFE_2000	4	Ethernet Mac
0xFFFE_3000	4	DMAC1
0xFFFE_4000	2	CF card attribute memory space
0xFFFE_4800	2	CF card common memory space
0xFFFE_5000	2	CF card I/O space
0xFFFE_5800	1	CF card True IDE CS1# space
0xFFFE_5C00	1	CF card True IDE CS2# space
0xFFFE_6000	4	CF Card Control Registers
0xFFFE_7000	4	Reserved
0xFFFE_8000	4	Camera interface
0xFFFE_9000	4	JPEG resize
0xFFFE_A000	4	JPEG module and FIFO control
0xFFFE_B000	4	JPEG codec
0xFFFE_C000	4	JPEG DMAC
0xFFFE_D000	4	I2C
0xFFFE_E000	4	I2S
0xFFFE_F000	4	(Interrupt controller)
0xFFFF_0000	4	Reserved
0xFFFF_1000	4	GPIO pin functions
0xFFFF_2000	4	SPI
0xFFFF_3000	4	Reserved
0xFFFF_4000	4	Reserved
0xFFFF_5000	4	UART
0xFFFF_6000	4	UART Lite (UARTL)
0xFFFF_7000	4	Reserved
0xFFFF_8000	4	RTC
0xFFFF_9000	4	DMAC2
0xFFFF_A000	4	Memory controller
0xFFFF_B000	4	Timers
0xFFFF_C000	4	Watchdog timer
0xFFFF_D000	4	System control
0xFFFF_E000	4	Reserved
0xFFFF_F000	4	Interrupt controller

4.4 Interrupt Controller

This device supports two fast (FIQ) and 32 normal (IRQ) interrupt requests. The following Table shows the source mappings that the interrupt controller uses for internal interrupt requests. For further details, see Section 15 “Interrupt Controller (INT).”

Table 4.5 Interrupt Request Source Connections

Type	Level	Source	Description
Fast interrupt request (FIQ)	FIQ0	Watchdog timer	
	FIQ1	GPIOB0 pin	Ex. Battery Low (*)
Normal interrupt request (IRQ)	IRQ0	Watchdog timer	
	IRQ1	Interrupt controller	Software interrupt request, produced by writing to a register
	IRQ2	ARM720T COMMRx	Debug Communication Port
	IRQ3	ARM720T COMMTx	Debug Communication Port
	IRQ4	Timer	16-bit timer channel 0
	IRQ5	Timer	16-bit timer channel 1
	IRQ6	Timer	16-bit timer channel 2
	IRQ7	Ethernet Mac & E-DMA	
	IRQ8	JPEG control	
	IRQ9	DMAC1	DMAC on AHB1 bus
	IRQ10	JPEG DMAC	
	IRQ11	Camera interface	
	IRQ12	Reserved	
	IRQ13	DMAC2	DMA INT, JPEG DMAC on AHB2 bus
	IRQ14 (**)	GPIOA or GPIOB	User specifies the GPIOA or GPIOB input pin to use as the source for interrupt requests.
	IRQ15	SPI0	SPI TXRDY/RXRDY
	IRQ16	I2C	Transfer complete
	IRQ17	UART	UART TXRDY/RXRDY
	IRQ18	RTC	Alarm or timer interval
	IRQ19	CF card interface	
	IRQ20 (*)	INT0	GPIOB0 input
	IRQ21 (*)	INT1	GPIOB1 input
	IRQ22 (*)	INT2	GPIOB2 input
	IRQ23	UARTL	UART Lite
	IRQ24 (*)	INT3	GPIOB3 input
	IRQ25 (*)	INT4	GPIOB4 input
	IRQ26 (*)	INT5	GPIOB5 input
	IRQ27 (*)	INT6	GPIOB6 input
	IRQ28 (*)	INT7	GPIOB7 input
	IRQ29 (*)	INT8	GPIOD0 input
	IRQ30	I2S0	I2S CH0
	IRQ31	I2S1	I2S CH1

* These represent direct inputs from the pins GPIOB[7:0] or GPIOD0. The defaults are active low Interrupt requests. An Interrupt controller Control Register provides the only way to change Enable, polarity, Level, and other Interrupt request Settings. They are thus different from IRQ14**, for which GPIO Control Register Settings are available.

** For further details on choosing the source from pins GPIOA[7:0] and GPIOB[7:0], see the detailed Register descriptions for GPIO[0x40] to GPIO[0x4C] in Section 25 “General-Purpose I/O (GPIO).”

4.5 Internal Functional Blocks

The following Table lists the many internal functional blocks that this device provides for creating a network camera controller.

Chapter	Functional Block	Abbreviation
5.	CPU	CPU
6.	DMA controller 1	DMAC1
7.	Camera interface	CAM
8.	JPEG controller	JPG
9.	JPEG_DMAC	JDMA
10.	DMA controller 2	DMAC2
11.	Ethernet MAC & E-DMA	ETH
12.	APB bridge	APB
13.	System controller	SYS
14.	Memory interface controller	MEMC
15.	Interrupt controller	INT
16.	UART	UART
17.	UART Lite	UARTL
18.	I2C Single Master Core Module	I2C
19.	I2S interface	I2S
20.	Serial peripheral interface	SPI
21.	Compact Flash (CF) card interface	CF
22.	Timers	TIM
23.	Real-time clock	RTC
24.	Watchdog timer	WDT
25.	GPIO	GPIO

5. CPU

5.1 Overview

The ARM720T CPU module contains an ARM7TDMI core with a unified 8 KB cache, a memory management unit (MMU), and an expansion write buffer. For further details, refer to the ARM720T Revision 4 (AMBA AHB Bus Interface Version) Core CPU Manual.

5.2 Block Diagram

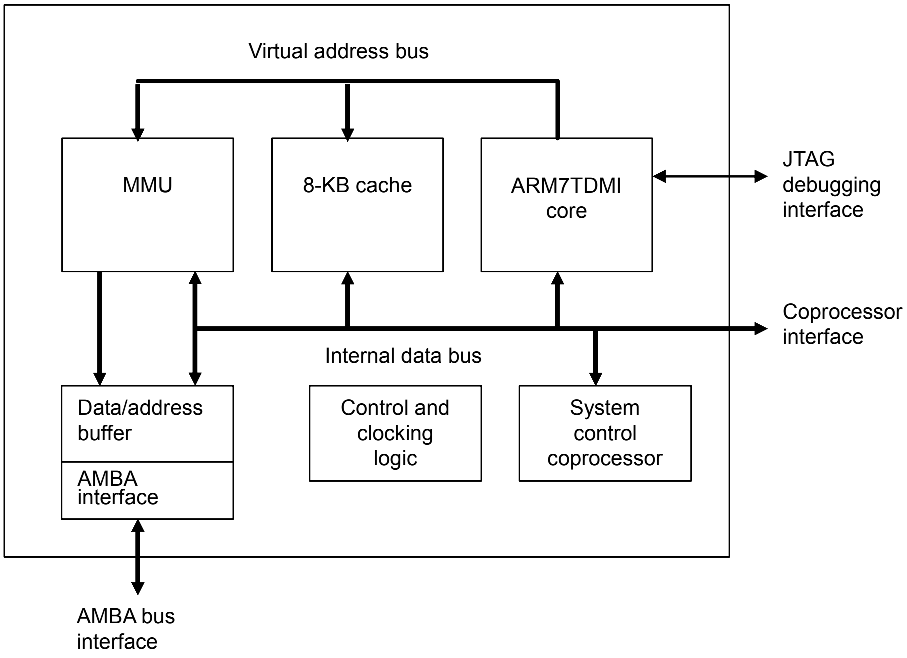


Fig.5.1 ARM720T Block Diagram

6. DMA CONTROLLER 1 (DMAC1)

6.1 Overview

This DMA controller, a bus master on the AHB1 bus, bypasses the CPU to transfer data directly between APB devices and memory (internal or external) or between memory devices.

This DMA controller supports dual-address transfers with two address phases. For each DMA request, reads the data first from the source address into an internal temporary register and then writes the data from there to the destination address. This cycle repeats until the number of transfers remaining goes to “0.”

The data size for these transfers can be 8, 16, or 32 bits.

6.2 Block Diagram

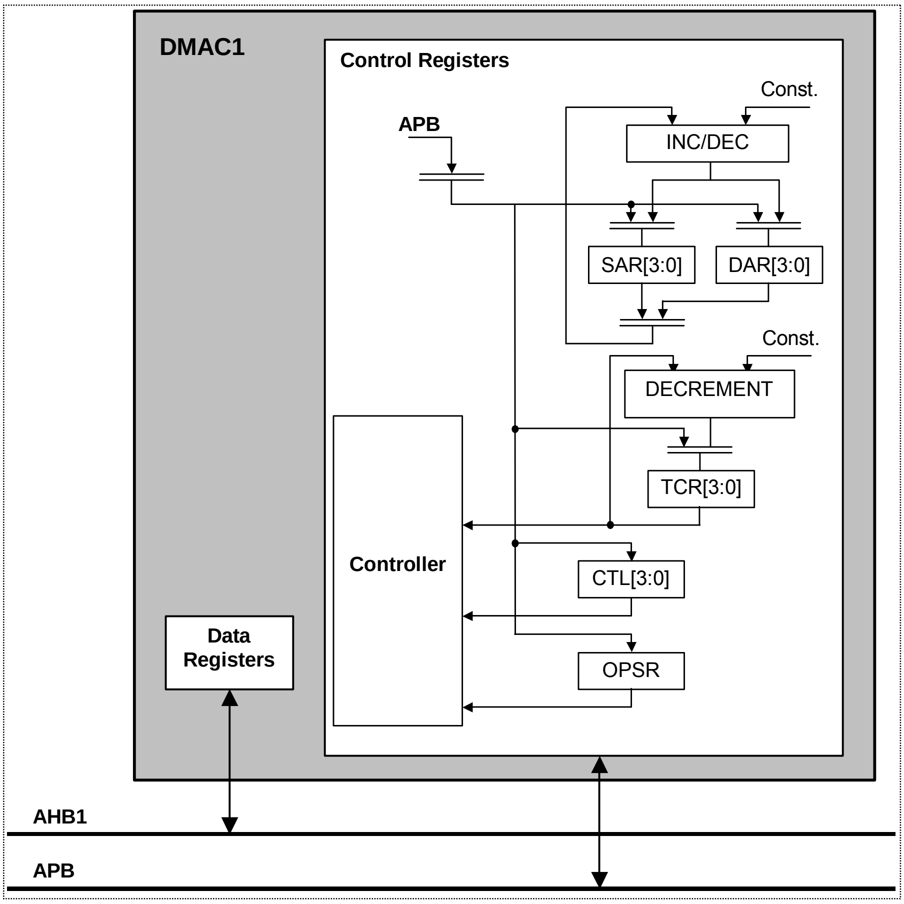


Fig.6.1 DMA Controller 1 (DMAC1) Block Diagram

6.3 External Pins

This block interacts with no external pins.

6.4 Registers

6.4.1 Register List

The base address for these registers is 0xFFFE_3000.
The register descriptions below sometimes use the following abbreviations.

R/W:	Read/Write
RO:	Read Only
WO:	Write Only
RSV:	Reserved bit or field*
n/a:	Bit or field not available--that is, not physically present in the hardware*

* In the absence of any indication to the contrary, set this to zero.

Table 6.1 DMAC1 Register List (Base Address: 0xFFFE_3000)

Address Offset	Register Name	Abbreviation	Default Value	R/W	Data Access Size (Bits)
0x00	DMA Channel 0 Source Address Register	SAR0	0xXXXX_XXXX	R/W	32
0x04	DMA Channel 0 Destination Address Register	DAR0	0xXXXX_XXXX	R/W	32
0x08	DMA Channel 0 Transfer Count Register	TCR0	0x00XX_XXXX	R/W	32
0x0C	DMA Channel 0 Control Register	CTL0	0x0000_0000	R/W	32
0x10	DMA Channel 1 Source Address Register	SAR1	0xXXXX_XXXX	R/W	32
0x14	DMA Channel 1 Destination Address Register	DAR1	0xXXXX_XXXX	R/W	32
0x18	DMA Channel 1 Transfer Count Register	TCR1	0x00XX_XXXX	R/W	32
0x1C	DMA Channel 1 Control Register	CTL1	0x0000_0000	R/W	32
0x20	DMA Channel 2 Source Address Register	SAR2	0xXXXX_XXXX	R/W	32
0x24	DMA Channel 2 Destination Address Register	DAR2	0xXXXX_XXXX	R/W	32
0x28	DMA Channel 2 Transfer Count Register	TCR2	0x00XX_XXXX	R/W	32
0x2C	DMA Channel 2 Control Register	CTL2	0x0000_0000	R/W	32
0x30	DMA Channel 3 Source Address Register	SAR3	0xXXXX_XXXX	R/W	32
0x34	DMA Channel 3 Destination Address Register	DAR3	0xXXXX_XXXX	R/W	32
0x38	DMA Channel 3 Transfer Count Register	TCR3	0x00XX_XXXX	R/W	32
0x3C	DMA Channel 3 Control Register	CTL3	0x0000_0000	R/W	32
0x60	DMA Channel Operating Select Register	OPSR	0x0000_0000	R/W	32

6.4.2 Detailed Register Descriptions

In the absence of any indication to the contrary, set all reserved bits to "0." Note that writing to a reserved bit risks unpredictable results.

Writing to bits labeled "n/a" has no hardware consequences.

Some registers only permit access under specific conditions. Reads and writes at other times are simply ignored.

DMA Channel 0 Source Address Register (SAR0)															
DMAC1[0x00]	Default = 0xXXXX_XXXX														Read/Write
Source Address [31:16]															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Source Address [15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 31 to 0: **DMA Channel 0 Source Address [31:0]**

This register specifies the source address for the DMA transfer on channel 0.

This address must fall on a memory boundary that matches the transfer data size. A 32-bit transfer, for example, requires 00b in Source Address Register bits 1 and 0.

After each successful transfer, the controller, in preparation for the next transfer, automatically updates this register by the transfer data size in bytes (TS: bits 4 to 3 in the Channel 0 Control Register) according to the source address mode (SAM: bits 13 to 12 in the in the Channel 0 Control Register).

DMA Channel 0 Destination Address Register (DAR0)															
DMAC1[0x04]	Default = 0xXXXX_XXXX														Read/Write
Destination Address [31:16]															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Destination Address [15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 31 to 0: **DMA Channel 0 Destination Address [31:0]**

This register specifies the destination address for the DMA transfer on channel 0.

This address must fall on a memory boundary that matches the transfer data size. A 32-bit transfer, for example, requires 00b in Source Address Register bits 1 and 0.

After each successful transfer, the controller, in preparation for the next transfer, automatically updates this register by the transfer data size in bytes (TS: bits 4 to 3 in the Channel 0 Control Register) according to the destination address mode (DAM: bits 15 to 14 in the in the Channel 0 Control Register).

DMA Channel 0 Transfer Count Register (TCR0)															
DMAC1[0x08]	Default = 0x00XX_XXXX														Read/Write
n/a								Transfer Count [23:16]							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Transfer Count [15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 23 to 0: **DMA Channel 0 Transfer Count [23:0]**

These bits specify the number of DMA transfers remaining. This count decrements after each successful transfer. Specifying "0" here specifies 2^{24}=16,777,216 transfers. Decrementing to zero triggers a DMA interrupt request.

Reads return zeros in bits 31 to 24.

DMA Channel 0 Control Register (CTL0)

DMAC1[0x0C] Default = 0x0000_0000 Read/Write

n/a								RSV				IDLE	RSV	AM	AL
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DAM		SAM		RS				RSV	RIM	TM	TS		IE	TE	DE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 23 to 20 (RSV): **Reserved (0)**

Bit 19 (IDLE): **Idle Delay Enable**

0: Disable (Normal operation)
1: Enable

The target device may require enabling, a delay in accepting the next request from the device. We therefore recommend setting this bit to “1” for write transfer from memory to I/O devices.

Bit 18 (RSV): **Reserved (0)**

Bit 17 (AM): **Acknowledge Mode**

Select the DMA cycle for DACK drive active signal output

0: Read cycle
1: Write cycle

Bit 16 (RSV): **Reserved (0)**

Bits 15 to 14 (DAM): **Destination Address Mode [1:0]**

This field specifies the strategy for updating the Destination Address Register after a successful transfer.

00: Leave fixed (Do not update)
01: Increment according to the transfer data size (8 bits: +1, 16 bits: +2, 32 bits: +4)
10: Decrement according to the transfer data size (8 bits: –1, 16 bits: –2, 32 bits: –4)
11: Reserved

Bits 13 to 12 (SAM): **Source Address Mode [1:0]**

This field specifies the strategy for updating the Source Address Register after a successful transfer.

00: Leave fixed (Do not update)
01: Increment according to the transfer data size (8 bits: +1, 16 bits: +2, 32 bits: +4)
10: Decrement according to the transfer data size (8 bits: –1, 16 bits: –2, 32 bits: –4)
11: Reserved

Bits 11 to 8 (RS): **Resource [3:0]**

This specifies the trigger for starting DMA transfers.

0000: I^{2}C output (WRREQ)
0001: I^{2}C input (RDREQ)
0010: I^{2}S I/O (I2S #0)
0011: I^{2}S I/O (I2S #1)
0100: UART output (TXRDY)
0101: UART input (RXRDY)
0110: SPI0 I/O (SPIIRQ)
0111-1110: Reserved
1111: Software request (SW-Request)
Setting all four bits to “1” specifies software DMA transfers.

Bit 7 (RSV): **Reserved (0)**

Bit 6 (RIM): **Request Input Mode**

This specifies the input mode for the DMA request signal from the specified resource.

0: Active low (level trigger)
1: Falling edge (edge trigger)

Bit 5 (TM): **Transfer Mode**

0: Single, one transfer per DMA request
1: Demand, continuous transfers until DMA request negated

Bits 4 to 3 (TS): **Transfer Size [1:0]**

These bits specify the data size for a transfer.

00: 8 bits
01: 16 bits
10: 32 bits
11: Reserved

Bit 2 (IE): **Interrupt Enable**

0: Disable
1: Enable

Setting this bit to "1" enables the transfer complete interrupt in the DMA channel 0.

Bit 1 (TE): **Transfer End**

0 (r): Transfers in progress or channel idle
1 (r): DMA transfer complete
0 (w): Clear this bit to "0"
1 (w): Ignored

This bit goes to "1" when all transfers are complete—that is, the DMA Channel 0 Transfer Count Register has decremented to zero. It retains this "1" setting until the software writes "0" to clear it to "0." DMA transfers on the channel are disabled until this bit returns to "0."

This bit also functions as an interrupt request source flag.

Bit 0 (DE): **DMA Enable**

0: Disable
1: Enable

Setting this bit to "1" enables DMA transfers on the channel 0.

DMA Channel 1 Source Address Register (SAR1)															
DMAC1[0x10]		Default = 0xXXXX_XXXX													Read/Write
Source Address [31:16]															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Source Address [15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 31 to 0: **DMA Channel 1 Source Address [31:0]**

This register specifies the source address for the DMA transfer on channel 1.

This address must fall on a memory boundary that matches the transfer data size. A 32-bit transfer, for example, requires 00b in Source Address Register bits 1 and 0.

After each successful transfer, the controller, in preparation for the next transfer, automatically updates this register by the transfer data size in bytes (TS: bits 4 to 3 in the Channel 1 Control Register) according to the source address mode (SAM: bits 13 to 12 in the in the Channel 1 Control Register).

DMA Channel 1 Destination Address Register (DAR1)
DMAC1[0x14] Default = 0xXXXX_XXXX Read/Write

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

Bits 31 to 0: **DMA Channel 1 Destination Address [31:0]**

This register specifies the destination address for the DMA transfer on channel 1.

This address must fall on a memory boundary that matches the transfer data size. A 32-bit transfer, for example, requires 00b in Source Address Register bits 1 and 0.

After each successful transfer, the controller, in preparation for the next transfer, automatically updates this register by the transfer data size in bytes (TS: bits 4 to 3 in the Channel 1 Control Register) according to the destination address mode (DAM: bits 15 to 14 in the in the Channel 1 Control Register).

DMA Channel 1 Transfer Count Register (TCR1)
DMAC1[0x18] Default = 0x00XX_XXXX Read/Write

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

Bits 23 to 0: **DMA Channel 1 Transfer Count [23:0]**

These bits specify the number of DMA transfers remaining. This count decrements after each successful transfer. Specifying “0” here specifies 224=16,777,216 transfers. Decrementing to zero triggers a DMA interrupt request.

Reads return zeros in bits 31 to 24.

DMA Channel 1 Control Register (CTL1)
DMAC1[0x1C] Default = 0x0000_0000 Read/Write

n/a								RSV				IDLE	RSV	AM	AL
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DAM		SAM		RS				RSV	RIM	TM	TS		IE	TE	DE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 23 to 20 (RSV): **Reserved (0)**

Bit 19 (IDLE): **Idle Delay Enable**

0: Disable (Normal operation)
1: Enable

The target device may require enabling, a delay in accepting the next request from the device. We therefore recommend setting this bit to “1” for write transfer from memory to I/O devices.

Bit 18 (RSV): **Reserved**

Bit 17 (AM): **Acknowledge Mode**

Select the DMA cycle for DACK active signal output

0: Read cycle
1: Write cycle

Bit 16 (RSV): **Reserved (0)**

Bits 15 to 14 (DAM): **Destination Address Mode [1:0]**

This field specifies the strategy for updating the Destination Address Register after a successful transfer.

00: Leave fixed (Do not update)
01: Increment according to the transfer data size (8 bits: +1, 16 bits: +2, 32 bits: +4)
10: Decrement according to the transfer data size (8 bits: –1, 16 bits: –2, 32 bits: –4)
11: Reserved

Bits 13 to 12 (SAM): **Source Address Mode [1:0]**

This field specifies the strategy for updating the Source Address Register after a successful transfer.

00: Leave fixed (Do not update)
01: Increment according to the transfer data size (8 bits: +1, 16 bits: +2, 32 bits: +4)
10: Decrement according to the transfer data size (8 bits: –1, 16 bits: –2, 32 bits: –4)
11: Reserved

Bits 11 to 8 (RS): **Resource [3:0]**

This specifies the trigger for starting DMA transfers.

0000: I2C output (WRREQ)
0001: I2C input (RDREQ)
0010: I2S I/O (I2S #0)
0011: I2S I/O (I2S #1)
0100: UART output (TXRDY)
0101: UART input (RXRDY)
0110: SPI0 I/O (SPIIRQ)
0111-1110: Reserved
1111: Software request (SW-Request)
Setting all four bits to “1” specifies software DMA transfers.

Bit 7 (RSV): **Reserved (0)**

Bit 6 (RIM): **Request Input Mode**

This specifies the input mode for the DMA request signal from the specified resource.

0: Active low (level trigger)
1: Falling edge (edge trigger)

Bit 5 (TM): **Transfer Mode**

0: Single, one transfer per DMA request
1: Demand, continuous transfers until DMA request negated

Bits 4 to 3 (TS): **Transfer Size [1:0]**

These bits specify the data size for a transfer.

00: 8 bits
01: 16 bits
10: 32 bits
11: Reserved

Bit 2 (IE): **Interrupt Enable**

0: Disable
1: Enable

Setting this bit to “1” enables the transfer complete interrupt in the DMA channel 1.

Bit 1 (TE): **Transfer End**

0 (r): Transfers in progress or channel idle
1 (r): DMA transfer complete
0 (w): Clear this bit to “0”
1 (w): Ignored

This bit goes to “1” when all transfers are complete—that is, the DMA Channel 1 Transfer Count Register has decremented to zero. It retains this “1” setting until the software writes “0” to clear it to “0.” DMA transfers on the channel are disabled until this bit returns to “0.”

This bit also functions as an interrupt request source flag.

Bit 0 (DE): **DMA Enable**

0: Disable
1: Enable

Setting this bit to “1” enables DMA transfers on the channel 1.

DMA Channel 2 Source Address Register (SAR2)															
DMAC1[0x20]		Default = 0xXXXX_XXXX													Read/Write
Source Address [31:16]															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Source Address [15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 31 to 0: **DMA Channel 2 Source Address [31:0]**

This register specifies the source address for the next DMA transfer on channel 2.

This address must fall on a memory boundary that matches the transfer data size. A 32-bit transfer, for example, requires 00b in Source Address Register bits 1 and 0.

After each successful transfer, the controller, in preparation for the next transfer, automatically updates this register by the transfer data size in bytes (TS: bits 4 to 3 in the Channel 2 Control Register) according to the source address mode (SAM: bits 13 to 12 in the in the Channel 2 Control Register).

DMA Channel 2 Destination Address Register (DAR2)															
DMAC1[0x24]		Default = 0xXXXX_XXXX													Read/Write
Destination Address [31:16]															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Destination Address [15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 31 to 0: **DMA Channel 2 Destination Address [31:0]**

This register specifies the destination address for the next DMA transfer on channel 2.

This address must fall on a memory boundary that matches the transfer data size. A 32-bit transfer, for example, requires 00b in Source Address Register bits 1 and 0.

After each successful transfer, the controller, in preparation for the next transfer, automatically updates this register by the transfer data size in bytes (TS: bits 4 to 3 in the Channel 2 Control Register) according to the destination address mode (DAM: bits 15 to 14 in the in the Channel 2 Control Register).

DMA Channel 2 Transfer Count Register (TCR2)															
DMAC1[0x28]		Default = 0x00XX_XXXX													Read/Write
n/a								Transfer Count [23:16]							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Transfer Count [15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 23 to 0: **DMA Channel 2 Transfer Count [23:0]**

These bits specify the number of DMA transfers remaining. This count decrements after each successful transfer. Specifying “0” here specifies 2^{24}=16,777,216 transfers. Decrementing to zero triggers a DMA interrupt request.

Reads return zeros in bits 31 to 24.

DMA Channel 2 Control Register (CTL2)
DMAC1[0x2C] Default = 0x0000_0000 Read/Write

n/a								RSV				IDLE	RSV	AM	AL
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DAM		SAM		RS				RSV	RIM	TM	TS		IE	TE	DE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 23 to 20 (RSV): **Reserved (0)**

Bit 19 (IDLE): **Idle Delay Enable**

0: Disable (Normal operation)
1: Enable

The target device may require enabling, a delay in accepting the next request from the device. We therefore recommend setting this bit to “1” for write transfer from memory to I/O devices.

Bit 18 (RSV): **Reserved (0)**

Bit 17 (AM): **Acknowledge Mode**

Select the DMA cycle for DACK active signal output

0: Read cycle
1: Write cycle

Bit 16 (RSV): **Reserved (0)**

Bits 15 to 14 (DAM): **Destination Address Mode [1:0]**

This field specifies the strategy for updating the Destination Address Register after a successful transfer.

00: Leave fixed (Do not update)
01: Increment according to the transfer data size (8 bits: +1, 16 bits: +2, 32 bits: +4)
10: Decrement according to the transfer data size (8 bits: –1, 16 bits: –2, 32 bits: –4)
11: Reserved

Bits 13 to 12 (SAM): **Source Address Mode [1:0]**

This field specifies the strategy for updating the Source Address Register after a successful transfer.

00: Leave fixed (Do not update)
01: Increment according to the transfer data size (8 bits: +1, 16 bits: +2, 32 bits: +4)
10: Decrement according to the transfer data size (8 bits: –1, 16 bits: –2, 32 bits: –4)
11: Reserved

Bits 11 to 8 (RS): **Resource [3:0]**

This specifies the trigger for starting DMA transfers.

0000: I^{2}C output (WRREQ)
0001: I^{2}C input (RDREQ)
0010: I^{2}S I/O (I2S #0)
0011: I^{2}S I/O (I2S #1)
0100: UART output (TXRDY)
0101: UART input (RXRDY)
0110: SPI0 I/O (SPIIRQ)
0111-1110: Reserved
1111: Software request (SW-Request)
Setting all four bits to “1” specifies software DMA transfers.

Bit 7 (RSV): **Reserved (0)**

Bit 6 (RIM): **Request Input Mode**

This specifies the input mode for the DMA request signal from the specified resource.

0: Active low (level trigger)
1: Falling edge (edge trigger)

Bit 5 (TM): **Transfer Mode**

0: Single, one transfer per DMA request
1: Demand, continuous transfers until DMA request negated

Bits 4 to 3 (TS): **Transfer Size [1:0]**

These bits specify the data size for a transfer.

00: 8 bits
01: 16 bits
10: 32 bits
11: Reserved

Bit 2 (IE): **Interrupt Enable**

0: Disable
1: Enable

Setting this bit to “1” enables the transfer complete interrupt in the DMA channel 2.

Bit 1 (TE): **Transfer End**

0 (r): Transfers in progress or channel idle
1 (r): DMA transfer complete
0 (w): Clear this bit to “0”
1 (w): Ignored

This bit goes to “1” when all transfers are complete—that is, the DMA Channel 2 Transfer Count Register has decremented to zero. It retains this “1” setting until the software writes “0” to clear it to “0.” DMA transfers on the channel are disabled until this bit returns to “0.”

This bit also functions as an interrupt request source flag.

Bit 0 (DE): **DMA Enable**

0: Disable
1: Enable

Setting this bit to “1” enables DMA transfers on the channel 2.

DMA Channel 3 Source Address Register (SAR3)															
DMAC1[0x30]		Default = 0xXXXX_XXXX													Read/Write
Source Address [31:16]															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Source Address [15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 31 to 0: **DMA Channel 3 Source Address [31:0]**

This register specifies the source address for the next DMA transfer on channel 3.

This address must fall on a memory boundary that matches the transfer data size. A 32-bit transfer, for example, requires 00b in Source Address Register bits 1 and 0.

After each successful transfer, the controller, in preparation for the next transfer, automatically updates this register by the transfer data size in bytes (TS: bits 4 to 3 in the Channel 3 Control Register) according to the source address mode (SAM: bits 13 to 12 in the in the Channel 3 Control Register).

DMA Channel 3 Destination Address Register (DAR3)
DMAC1[0x34] Default = 0xXXXX_XXXX Read/Write

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

Bits 31 to 0: **DMA Channel 3 Destination Address [31:0]**

This register specifies the destination address for the next DMA transfer on channel 3.

This address must fall on a memory boundary that matches the transfer data size. A 32-bit transfer, for example, requires 00b in Source Address Register bits 1 and 0.

After each successful transfer, the controller, in preparation for the next transfer, automatically updates this register by the transfer data size in bytes (TS: bits 4 to 3 in the Channel 3 Control Register) according to the destination address mode (DAM: bits 15 to 14 in the in the Channel 3 Control Register).

DMA Channel 3 Transfer Count Register (TCR3)
DMAC1[0x38] Default = 0x00XX_XXXX Read/Write

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

Bits 23 to 0: **DMA Channel 3 Transfer Count [23:0]**

These bits specify the number of DMA transfers remaining. This count decrements after each successful transfer. Specifying "0" here specifies 2^{24}=16,777,216 transfers. Decrementing to zero triggers a DMA interrupt request.

Reads return zeros in bits 31 to 24.

DMA Channel 3 Control Register (CTL3)
DMAC1[0x3C] Default = 0x0000_0000 Read/Write

n/a								RSV				IDLE	RSV	AM	AL
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DAM		SAM		RS				RSV	RIM	TM	TS		IE	TE	DE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 23 to 20 (RSV): **Reserved (0)**

Bit 19 (IDLE): **Idle Delay Enable**

0: Disable (Normal operation)
1: Enable

The target device may require enabling, a delay in accepting the next request from the device. We therefore recommend setting this bit to "1" for write transfer from memory to I/O devices.

Bit 18 (RSV): **Reserved**

Bit 17 (AM): **Acknowledge Mode**

Select the DMA cycle for DACK active signal output

0: Read cycle
1: Write cycle

Bit 16 (RSV): **Reserved (0)**

Bits 15 to 14 (DAM): **Destination Address Mode [1:0]**

This field specifies the strategy for updating the Destination Address Register after a successful transfer.

00: Leave fixed (Do not update)
01: Increment according to the transfer data size (8 bits: +1, 16 bits: +2, 32 bits: +4)
10: Decrement according to the transfer data size (8 bits: –1, 16 bits: –2, 32 bits: –4)
11: Reserved

Bits 13 to 12 (SAM): **Source Address Mode [1:0]**

This field specifies the strategy for updating the Source Address Register after a successful transfer.

00: Leave fixed (Do not update)
01: Increment according to the transfer data size (8 bits: +1, 16 bits: +2, 32 bits: +4)
10: Decrement according to the transfer data size (8 bits: –1, 16 bits: –2, 32 bits: –4)
11: Reserved

Bits 11 to 8 (RS): **Resource [3:0]**

This specifies the trigger for starting DMA transfers.

0000: I2C output (WRREQ)
0001: I2C input (RDREQ)
0010: I2S I/O (I2S #0)
0011: I2S I/O (I2S #1)
0100: UART output (TXRDY)
0101: UART input (RXRDY)
0110: SPI0 I/O (SPIIRQ)
0111-1110: Reserved
1111: Software request (SW-Request)
Setting all four bits to “1” specifies software DMA transfers.

Bit 7 (RSV): **Reserved (0)**

Bit 6 (RIM): **Request Input Mode**

This specifies the input mode for the DMA request signal from the specified resource.

0: Active low (level trigger)
1: Falling edge (edge trigger)

Bit 5 (TM): **Transfer Mode**

0: Single, one transfer per DMA request
1: Demand, continuous transfers until DMA request negated

Bits 4 to 3 (TS): **Transfer Size [1:0]**

These bits specify the data size for a transfer.

00: 8 bits
01: 16 bits
10: 32 bits
11: Reserved

Bit 2 (IE): **Interrupt Enable**

0: Disable
1: Enable

Setting this bit to “1” enables the transfer complete interrupt in the DMA channel 3.

Bit 1 (TE): **Transfer End**

0 (r): Transfers in progress or channel idle
1 (r): DMA block transfer complete
0 (w): Clear this bit to “0”
1 (w): Ignored

This bit goes to “1” when all transfers are complete—that is, the DMA Channel 3 Transfer Count Register has decremented to zero. It retains this “1” setting until the software writes “0” to clear it to “0.” DMA transfers on the channel are disabled until this bit returns to “0.”

This bit also functions as an interrupt request source flag.

Bit 0 (DE): **DMA Enable**

0: Disable
1: Enable

Setting this bit to “1” enables DMA transfers on the channel 3.

DMA Channel Operating Select Register (OPSR)															
DMAC1[0x60]		Default = 0x0000_0000													Read/Write
na															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
n/a						DPM		n/a							DGE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 9 to 8 (DPM): **DMA Channel Priority**

This specifies the order for assigning priority to simultaneous transfer requests on multiple channels.

00: CH0>CH1>CH2>CH3
01: CH0>CH2>CH3>CH1
10: CH2>CH0>CH1>CH3
11: Reserved

Bit 0 (DGE): **DMA Global Enable**

This simultaneously switches all DMA channels on and off.

0: Disable
1: Enable

7. CAMERA INTERFACE (CAM)

7.1 Overview

This Camera interface has the following features.

- Support for image sizes up to UXGA (1600 × 1200), provided that the camera matches this device's AC characteristics
- 8-bit data bus interface (YUV 4:2:2 format)
- Support for ITU-R BT.656 camera input
- Choice of capture frame settings

7.2 Block Diagram

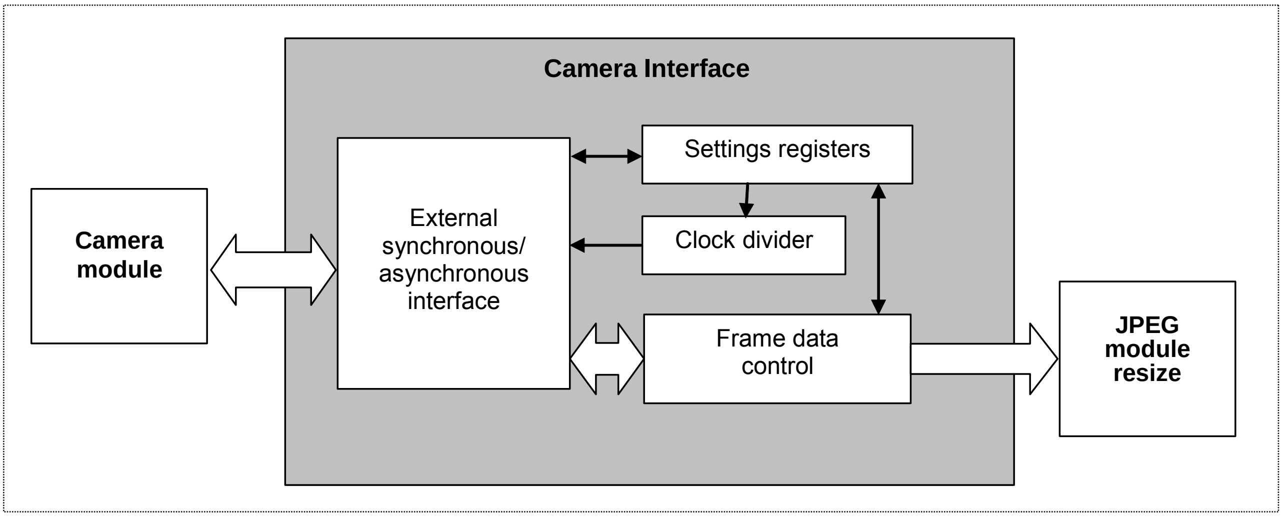


Fig.7.1 Camera Interface Block Diagram

7.3 External Pins

This block interacts with the following external pins.

Table 7.1 Camera Block External Pins

Pin Name	I/O	Pin Function	Multiplexed Pin*
CMDATA [7:0]	I	Camera data (YUV) inputs	GPIOC [7:0]
CMVREF	I	Vertical synchronous input from camera module	GPIOD4
CMHREF	I	Horizontal synchronous input from camera module	GPIOD5
CMCLKOUT	O	Basic clock output for camera	GPIOD6
CMCLKIN	I	Pixel clock for camera data input	GPIOD7

* These external pins is set as GPIO operation, so specify "non-GPIO function #1" in the GPIO Pin Function Register to configure them for this function.

7.4 Registers

7.4.1 Register List

The base address for these registers is 0xFFFE_8000.

Table 7.2 Register List (Base Address: 0xFFFE_8000)

Address Offset	Register Name	Default Value	R/W	Data Access Size (Bits)
0x00	Camera Clock Frequency Setting Register	0x0000	R/W	16
0x04	Camera Signal Setting Register	0x0000	R/W	16
0x08 to 0x1C	Reserved	—	—	—
0x20	Camera Mode Setting Register	0x0000	R/W	16
0x24	Camera Frame Control Register	0x0000	R/W	16
0x28	Camera Control Register	0x0000	WO	16
0x2C	Camera Status Register	0x0004	RO	16
0x30 to 0x5C	Reserved	—	—	—

7.4.2 Detailed Register Descriptions

Camera Clock Frequency Setting Register
CAM[0x00] Default = 0x0000 Read/Write

n/a							
15	14	13	12	11	10	9	8
n/a			Clock Frequency Select bits [4:0]				
7	6	5	4	3	2	1	0

Bits 4 to 0: **Clock Frequency Select bits [4:0]**

Theses bits set the frequency of output clock (CMCLKOUT).

Table 7.3 Output Clock (CMCLKOUT) Frequency

Value	Clock Frequency
00000	Internal clock 1/1
00001	Internal clock 1/2
00010	Internal clock 1/3
00011	Internal clock 1/4
00100	Internal clock 1/5
00101	Internal clock 1/6
00110	Internal clock 1/7
00111	Internal clock 1/8
01000	Internal clock 1/9
01001	Internal clock 1/10
01010	Internal clock 1/11
01011	Internal clock 1/12
01100	Internal clock 1/13
01101	Internal clock 1/14
01110	Internal clock 1/15
01111	Internal clock 1/16
10000	Internal clock 1/17
10001	Internal clock 1/18
10010	Internal clock 1/19
10011	Internal clock 1/20
10100	Internal clock 1/21
10101	Internal clock 1/22
10110	Internal clock 1/23
10111	Internal clock 1/24
11000	Internal clock 1/25
11001	Internal clock 1/26
11010	Internal clock 1/27
11011	Internal clock 1/28
11100	Internal clock 1/29
11101	Internal clock 1/30
11110	Internal clock 1/31
11111	Internal clock 1/32

Camera Signal Setting Register
CAM[0x04] Default = 0x0000 Read/Write

n/a							
15	14	13	12	11	10	9	8
n/a	Reserved (0)	Clock Mode Select	YUV Data Format Select bits [1:0]		HREF Active Select	VREF Active Select	Valid Input Clock Edge
7	6	5	4	3	2	1	0

Bit 6: **Reserved (0)**

Always set to "0."

Bit 5: **Clock Mode Select**

0: External (CMCLKIN)
1: Internal (CMCLKOUT)

Using the internal clock signal requires setting bits 4 to 0 in the Camera Clock Frequency Setting Register to a nonzero value to produce a frequency divider of at least 2. Delays on the board and inside the camera's image sensor also complicate synchronization between the internal clock signal and the data from the camera's image sensor. (Synchronization is not a problem when using the external clock signal.) Achieving stable operation requires setting the frequency divisor high enough to safely ignore those delays.

Bits 4 to 3: **YUV Data Format Select bits [1:0]**

This specifies the byte order for the YUV data input.

Table 7.4 YUV Data Formats

Setting	Format
00	(1st) CbYCrY (last)
01	(1st) CrYCbY (last)
10	(1st) YCbYCr (last)
11	(1st) YCrYCb (last)

Bit 2: **HREF Active Select**

Select the HREF data active level.

0: High
1: Low

Bit 1: **VREF Active Select**

Select the VREF data active level.

0: High
1: Low

Bit 0: **Valid Input Clock Edge**

Select the trigger edge of input clock for data capture timing.

0: Rising
1: Falling

This setting applies to both clock signals (external or internal) specified with bit 5.

Camera Mode Setting Register
CAM[0x20] Default = 0x0000 Read/Write

RSV (0)			RSV Camera Pin Active Pull-down Disable	n/a	Fast Sampling Mode	RSV (0)	
15	14	13	12	11	10	9	8
ITU-R BT656 Enable	RSV (0)			Clock Output Disable	RSV (0)		Camera Module Enable
7	6	5	4	3	2	1	0

Bits 15 to 13 (RSV): **Reserved (0)**

Bit 12 (RSV): **Reserved (0)**

Camera Active Pull-Down Disable
0: Enable
1: Disable

Note: This bit is not functional in this device. Use the system controller's GPIOC/GPIOD Resistor Control Register instead.

Bit 10: **Fast Sampling Mode**

0: Normal sampling
1: Fast sampling

Setting this bit to “1” doubles the sampling rate for the camera input data.

Bits 9 to 8 (RSV): **Reserved (0)**

Bit 7: **ITU-R BT.656 Enable**

0: Disable
1: Enable

Setting this bit to “1” switches to ITU-R BT.656 camera input.
This setting is only valid for the YUV 4:2:2 8-bit interface.

Bits 6 to 4 (RSV): **Reserved (0)**

Bit 3: **Clock Output (CMCLKOUT) Disable**

0: Enable (outputs CMCLKOUT)
1: Disable (fixed Low level)

The Camera Clock Frequency Setting Register specifies the CMCLKOUT frequency.
Disabling output fixes the pin output at Low level.

Bits 2 to 1 (RSV): **Reserved (0)**

Bit 0: **Camera Module Enable**

0: Disable
1: Enable

Setting this bit to “1” enables external clock output and other signals. Setting it to “0,” on the other hand, stops the clock signal to the camera module, helping reduce power consumption, but does not prevent read/write access to the Camera Interface Registers.

Camera Frame Control Register CAM[0x24] Default = 0x0000							Read/Write
n/a							JPEG Raw Data Capture Mode
15	14	13	12	11	10	9	8
Frame Capture Interrupt Control	Single Frame Capture Enable	Shutter Sync. Disable	Frame Sample Control bits [2:0]			Image Capture Interrupt Polarity	Image Capture Interrupt Request Enable
7	6	5	4	3	2	1	0

Bit 8: **JPEG Raw Data Capture Mode**

0: YUV data capture
1: JPEG data capture

Setting this bit to “1” switches data capture from YUV data to JPEG data.

Bit 7: **Frame Capture Interrupt Control**

This controls frame capture interrupt requests.
Frame capture complete interrupt requests (“1” in this bit) ignore the settings in bits 5 and 0. Setting this bit to “0” does not disable them.

Table 7.5 Frame Capture Interrupt Control

Image Capture Interrupt Polarity Bit	Interrupt
0	Valid frame capture
1	Frame capture complete

Bit 6: **Single Frame Capture Enable**

0: Disable

1: Enable

Setting this bit to "1" stops image capture one frame after the software sets CAM[0x28] bit 2 (frame capture start) to "1." Setting it to "0," on the other hand, produces repeated capture.

Note: Do not change this bit while CAM[0x20] bit 0 (camera module enable) is "1."

Bit 5: **Shutter Synchronization Disable**

Setting this bit to "1" disables the link between the shutter press and the start of frame capture interrupt status flag updates for each valid frame.

0: Disable flag updates until shutter is pressed

1: Enable flag updates at all times

Bits 4 to 2: **Frame Sample Control bits [2:0]**

This specifies the number of camera input frames to skip between frames.

Table 7.6 Frame Sampling Control

Frame Sample Control Bits [2:0]	Mode
000	0 (Accept all)
001	1 (Accept 1/2)
010	2 (Accept 1/3)
011	3 (Accept 1/4)
100	4 (Accept 1/5)
101	5 (Accept 1/6)
110	Reserved
111	Reserved (Accept none)

Bit 1: **Image Capture Interrupt Polarity**

This controls image capture interrupt request timing.

Table 7.7 Image Capture Interrupts

Image Capture Interrupt Polarity Bit	Interrupt
0	VREF level goes from data valid to data invalid
1	VREF level goes from data invalid to data valid

Bit 0: **Image Capture Interrupt Enable**

0: Disable

1: Enable

Camera Control Register
CAM[0x28] Default = 0x0000 Write Only

n/a						ITU-R BT656 Error Flag 1 Clear	ITU-R BT656 Error Flag 0 Clear
15	14	13	12	11	10	9	8
n/a				Frame Capture Stop	Frame Capture Start	Frame Interrupt Status Flag Clear	Camera Module Software Reset
7	6	5	4	3	2	1	0

Bit 9: **ITU-R BT.656 Error Flag 1 Clear**

0: (ignored)
1: Clear error flag 1

Writing "1" to this bit clears the error flag 1.

Bit 8: **ITU-R BT.656 Error Flag 0 Clear**

0: (ignored)
1: Clear error flag 0

Writing "1" to this bit clears the error flag 0.

Bit 3: **Frame Capture Stop**

0: (ignored)
1: Stop capture

Writing "1" to this bit stops image frame capture. After a reset, software should initialize the camera module by simultaneously writing "1" to both this bit and bit 0 (Camera Module Software Reset) to start it with capture off. The default state is on.

Bit 2: **Frame Capture Start**

0: (ignored)
1: Start capture

Writing "1" to this bit starts image frame capture. Simultaneously setting bit 6 (Single frame capture enable) to "1," however, automatically stops capture after a single frame.

Bit 1: **Frame Capture Interrupt Status Clear**

0: (ignored)
1: Clear flag

Writing "1" to this bit clears the status flag indicating frame capture interrupt requests from the camera.

Bit 0: **Camera Module Software Reset**

Writing "1" to this bit initializes the camera module.

0: (ignored)
1: Initialize

Note: Setting this bit to "1" does not initialize registers. It only resets CAM[0x20] bit 0 (Camera Module Enable) to "0."

Camera Status Register
CAM[0x2C] Default = 0x0004 Read Only

n/a						ITU-R BT656 Error Flag 1	ITU-R BT656 Error Flag 0
15	14	13	12	11	10	9	8
n/a	Camera VSYNC	RSV (1)	Effective Frame Status	Frame Capture Busy Status	Frame Capture Start/Stop Flag	Frame Capture Interrupt Status	n/a
7	6	5	4	3	2	1	0

Bit 9: **ITU-R BT.656 Error Flag 1**

This bit monitors reference command status in the ITU-R BT656 mode.

0: Normal operation
1: 2-bit error was detected during reference decoding.

Bit 8: **ITU-R BT.656 Error Flag 0**

This bit monitors reference command status in the ITU-R BT656 mode.

0: Normal operation
1: 1-bit error was detected during reference decoding.

Bit 6: **Camera VSYNC**

0: Vertical blanking
1: Valid data

This bit monitors the VSYNC signal from the camera module. In ITU-R BT656 mode, it monitors the status.

Note that VSYNC levels are constant regardless of the polarity setting.

Bit 5 (RSV): **Reserved (1)**

Bit 4: **Effective Frame Status**

0: Invalid (skipped)
1: Valid

This bit indicates the frame status when frame skipping is in effect.

Bit 3: **Frame Capture Busy Status**

0 Idle
1: Busy

This bit indicates the image frame capture status.

Bit 2: **Frame Capture Start/Stop Flag**

0: Stop
1: Start

This bit indicates the image frame capture setting.

Single frame capture, however, automatically resets this bit to “0” (stop) after a single frame.

Bit 1: **Frame Capture Interrupt Status**

0: There is no interrupt request
1: There is an interrupt request

This bit indicates frame capture interrupt request status.

Note: “1” in Camera Frame Control Register (CAM[0x24]) bit 7 or 0 indicates that there is a frame capture interrupt request.

7.5 Description of Operation

This device connects to camera modules (image sensors) with image sizes up to* UXGA (1600 × 1200). The camera interface features an 8-bit data bus and clock signals for receiving YUV 4:2:2 image data.

* Camera compatibility depends on its match with this device's AC characteristics.

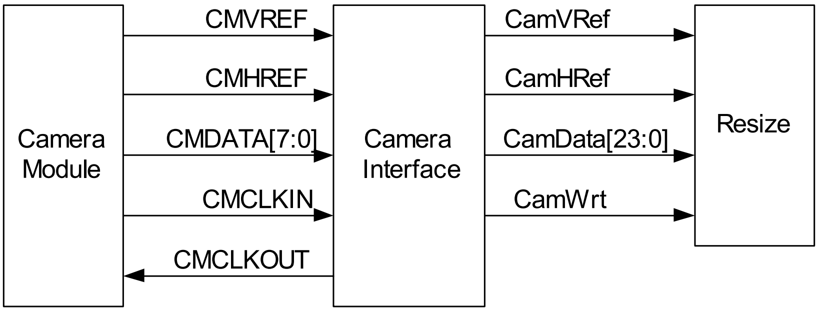


Fig.7.2 Camera Interface Connections

Table 7.8 CAM Internal Signals

Internal Signal Name	Description
CMVREF	Vertical data valid signal from camera module
CMHREF	Horizontal data valid signal from camera module
CMDATA[7:0]	8-bit data from camera module The ITU-R BT.601 format is also supported.
CMCLKIN	Pixel clock from camera module
CMCLKOUT	Operating clock for camera module
CamVRef	Vertical data valid signal to capture resizer
CamHRef	Horizontal data valid signal to capture resizer
CamData[23:0]	24-bit based on results of YUV 4:2:2 to YUV 4:4:4 conversion
CamWrt	Data valid signal to capture resizer

The camera interface synchronizes the pixel clock (CMCLKIN) signal and other asynchronous signals from the camera module with its internal system clock (SYSCLK) signal and sends the resulting camera image data to the capture resizer.

The following Figure shows a circuit using CMHREF and CMVREF for sampling the image data (CMDATA) from the camera module at CMCLKIN rising edges.

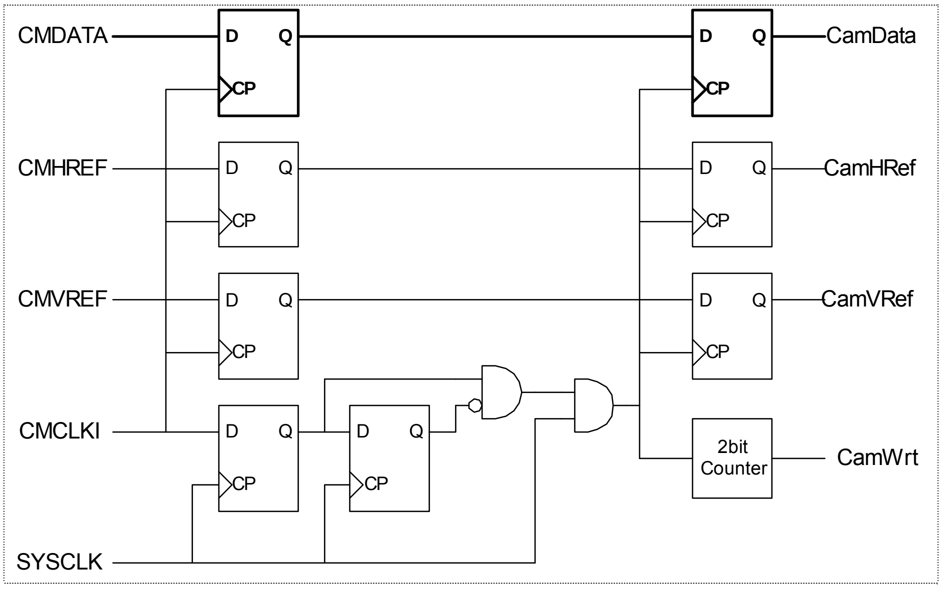


Fig.7.3 Data Sampling Circuit

In the normal sampling mode, the theory behind the clock edge detection mechanism says that the internal system clock signal sampling the pixel clock signal from the camera clock module must have at least twice the frequency. In real-life situations, however, clock duty ratios and other factors complicate things, making reliable operation at twice the frequency system dependent. We therefore recommend a multiplier of at least three.

In the fast sampling mode, the sampling rate is double of the normal sampling because of using the toggle buffering circuits.

7.5.1 Frame Capture Interrupt Requests

VREF input indicating the start of valid camera image data triggers interrupt requests needed by JPEG encoding and other image processing routines using that data.

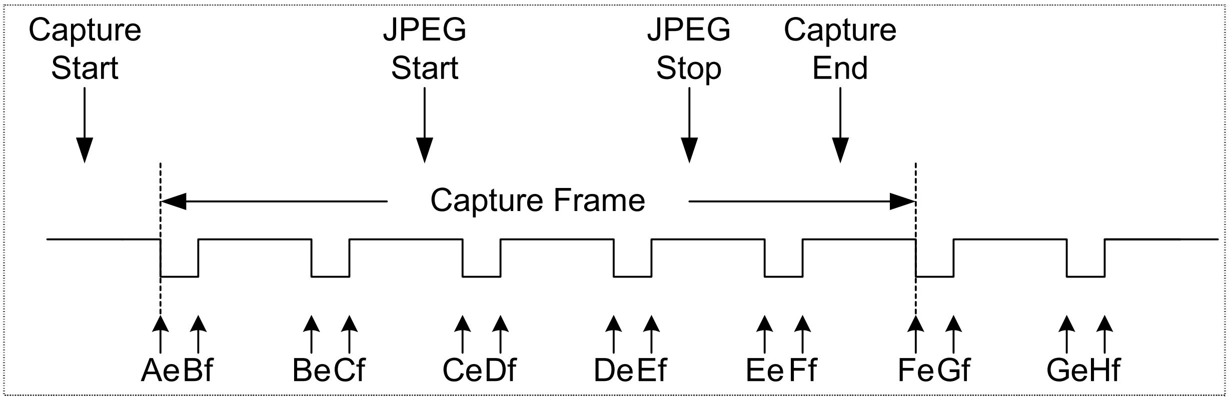


Fig.7.4 Interrupt Request Timing

The following is a timing chart showing the sequence of actions related to camera frame capture interrupt requests from enabling the camera interface capture through disabling it. “Capture Start” corresponds to writing “1” to the camera interface’s Camera Mode Setting Register bit 0; “Capture End,” writing “0” to that bit. Similarly, “JPEG Start” corresponds to writing “1” to the JPEG Start/Stop Control bit; “JPEG End,” writing “0” to that bit. Finally, “Capture Frame” indicates the resulting frame of camera module data flowing to the capture resizer. There are four register bits specifying the frame capture interrupt request timing. The following Table lists, using the labels at the bottom of the Figure, the trigger points for each setting combination. The fourth one, with shading, is the combination for normal use.

Table 7.9 Interrupt Request Timing

Camera Frame Control Register (CAM[0x24]) bits				Trigger Points
7	5	1	0	
0	x	x	0	None
0	0	0	1	Ce, De
0	0	1	1	Df, Ef
0	1	0	1	Be, Ce, De, Ee, Fe
0	1	1	1	Cf, Df, Ef, Ff, Gf
1	x	0	x	Fe
1	x	1	x	Gf

8. JPEG CONTROLLER (JPG)

8.1 Overview

This module provides encoding (to JPEG data) and capture (as YUV data) functions for the camera input image. It also includes a dedicated register access port used during YUV-JPEG and JPEG-YUV conversion.

The JPEG codec generally supports JPEG baseline methods for JPEG encoding and decoding and fully supports the arithmetic precision specified in JPEG Part 2 (ISO/IEC 10918-2).

This module supports image sizes up to UXGA (1600 × 1200). It supports JPEG encoding/decoding of images with resolutions greater than or equal to the minimum resolution for the YUV data format and image sizes that are multiples of the MCU size.

This module supports two quantization tables for compression and four for expansion.

This module supports two DC and AC Huffman tables each.

This module supports up to 36 bytes of user-specified markers during encoding. It processes SOI, SOF0, SOS, DQT, DHT, DRI, RSTm, and EOI markers, automatically decoding them during expansion. It does not support DNL markers.

This module supports four YUV data formats (4:4:4, 4:2:2, 4:2:0, and 4:1:1) for camera image JPEG encoding, but only two (4:2:2 and 4:2:0) for host JPEG encoding, host JPEG decoding, and YUV data capture.

This module does not support gray scale or RGB images.

The target processing time is a maximum of 1/30 seconds for the VGA size (640 × 480), but this is not guaranteed because throughput can vary widely with the quantization table settings, the Huffman table settings, camera input image details, and other factors.

8.2 Block Diagram

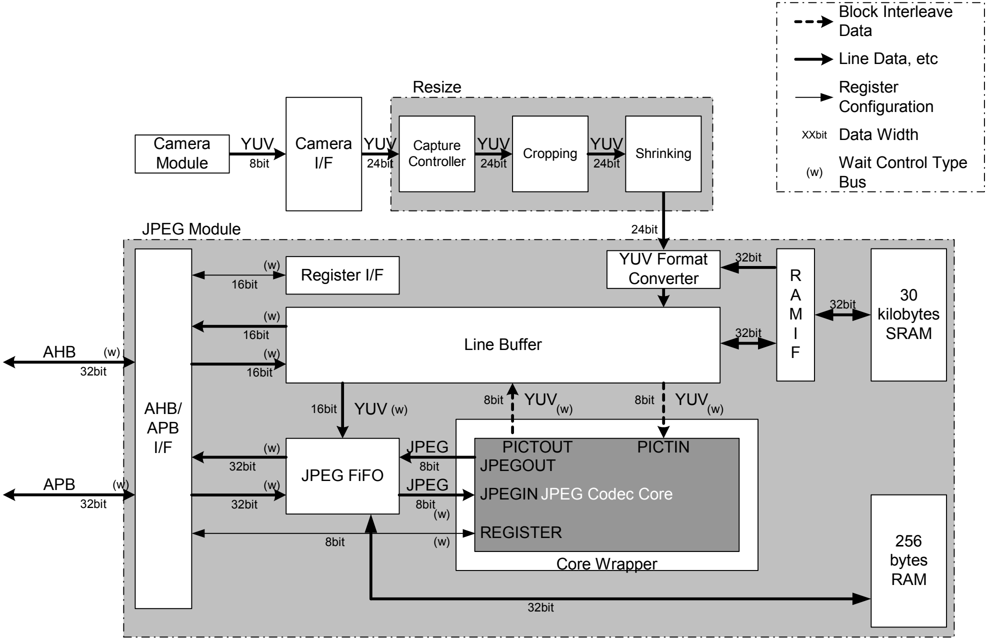


Fig.8.1 JPEG Controller Block Diagram

8.3 External Pins

This block interacts with no external pins.

8.4 Registers

8.4.1 Register List

Table 8.1 JPEG Controller Register List

Address Offset	Register Name	Default Value	R/W	Data Access Size (Bits)
Resizer Operation Registers (RSZ) : Base Address = 0xFFFE_9000				
0x60	Global Resizer Control Register	0x0000	WO	16
0x64	Capture Control State Register	0x0000	RO	16
0x68	Capture Data Setting Register	0x0000	R/W	16
0x70 to 0x7C	Reserved Registers	0x0000	R/W	16
0xC0	Capture Resizer Control Register	0x0000	R/W	16
0xC8	Capture Resizer Start X Position Register	0x0000	R/W	16
0xCC	Capture Resizer Start Y Position Register	0x0000	R/W	16
0xD0	Capture Resizer End X Position Register	0x027F	R/W	16
0xD4	Capture Resizer End Y Position Register	0x01DF	R/W	16
0xD8	Capture Resizer Scaling Rate Register	0x8080	R/W	16
0xDC	Capture Resizer Scaling Mode Register	0x0000	R/W	16
JPEG Module Registers (JCTL) : Base Address = 0xFFFE_A000				
0x00	JPEG Control Register	0x0000	R/W	16
0x04	JPEG Status Flag Register	0x8080	R/W	16
0x08	JPEG Raw Status Flag Register	0x8080	RO	16
0x0C	JPEG Interrupt Control Register	0x0000	R/W	16
0x10	Reserved Register	0x0080	RO	16
0x14	JPEG Codec Start/Stop Control Register	0x0000	WO	16
0x18 to 0x1C	Reserved Registers	—	—	16
0x20	Huffman Table Automatic Setting Register	0x0000	R/W	16
JPEG FIFO Settings Registers (JFIFO) : Base Address = 0xFFFE_A000				
0x40	JPEG FIFO Control Register	0x0000	R/W	16
0x44	JPEG FIFO Status Register	0x8001	RO	16
0x48	JPEG FIFO Size Register	0x003F	R/W	16
0x4C	JPEG FIFO Read/Write Port Register	0x0000 0000	R/W	32
0x50 to 0x58	Reserved Registers	—	—	16
0x60	Encode Size Limit Register 0	0x0000	R/W	16
0x64	Encode Size Limit Register 1	0x0000	R/W	16
0x68	Encode Size Result Register 0	0x0000	RO	16
0x6C	Encode Size Result Register 1	0x0000	RO	16
0x70 to 0x78	Reserved Registers	—	—	16
JPEG Line Buffer Setting Registers (JLB) : Base Address = 0xFFFE_A000				
0x80	JPEG Line Buffer Status Flag Register	0x0000	R/W	16
0x84	JPEG Line Buffer Raw Status Flag Register	0x0000	RO	16
0x88	JPEG Line Buffer Current Status Flag Register	0x0009	RO	16
0x8C	JPEG Line Buffer Interrupt Control Register	0x0000	R/W	16
0x90 to 0x9C	Reserved Registers	—	—	16
0xA0	JPEG Line Buffer Horizontal Pixel Support Size Register	0x2800	R/W	16
0xA4	JPEG Line Buffer Memory Address Offset Register	0x0030	R/W	16
0xA8 to 0xBC	Reserved Registers	—	—	16
0xC0	JPEG Line Buffer Read/Write Port Register	0x0000	R/W	16
JPEG Codec Registers (JCODEC) : Base Address = 0xFFFE_B000				
0x00	Operation Mode Setting Register	0x0000	R/W	16
0x04	Command Setting Register	Not applicable	WO	16
0x08	JPEG Operation Status Register	0x0000	RO	16
0x0C	Quantization Table Number Register	0x0000	R/W	16
0x10	Huffman Table Number Register	0x0000	R/W	16
0x14	DRI Setting Register 0	0x0000	R/W	16
0x18	DRI Setting Register 1	0x0000	R/W	16
0x1C	Vertical Pixel Size Register 0	0x0000	R/W	16
0x20	Vertical Pixel Size Register 1	0x0000	R/W	16

Address Offset	Register Name	Default Value	R/W	Data Access Size (Bits)
JPEG Codec Registers (JCODEC) : Base Address = 0xFFFE_B000				
0x24	Horizontal Pixel Size Register 0	0x0000	R/W	16
0x28	Horizontal Pixel Size Register 1	0x0000	R/W	16
0x2C to 0x34	Reserved Registers	—	—/—	16
0x38	RST Marker Operation Setting Register	0x0000	R/W	16
0x3C	RST Marker Operation Status Register	0x0000	RO	16
0x40 to 0xCC	Insertion Marker Data Registers	0x00FF	R/W	16
0x400 to 0x4FC	Quantization Table No. 0 Register	Not applicable	R/W	16
0x500 to 0x5FC	Quantization Table No. 1 Register	Not applicable	R/W	16
0x800 to 0x83C	DC Huffman Table No. 0 Register 0	Not applicable	WO	16
0x840 to 0x86C	DC Huffman Table No. 0 Register 1	Not applicable	WO	16
0x880 to 0x8BC	AC Huffman Table No. 0 Register 0	Not applicable	WO	16
0x8C0 to 0xB44	AC Huffman Table No. 0 Register 1	Not applicable	WO	16
0xC00 to 0xC3C	DC Huffman Table No. 1 Register 0	Not applicable	WO	16
0xC40 to 0xC6C	DC Huffman Table No. 1 Register 1	Not applicable	WO	16
0xC80 to 0xCBC	AC Huffman Table No. 1 Register 0	Not applicable	WO	16
0xCC0 to 0xF44	AC Huffman Table No. 1 Register 1	Not applicable	WO	16

The following describes these registers in detail.

8.4.2 Resizer Operation Registers (RSZ)

Note: These registers, apart from a few exceptions, cannot be modified while this device is accepting data from the camera interface.

Global Resizer Control Register							
RSZ[0x60]	Default = 0x0000						Write Only
n/a					Reserved		ACTAGAIN
15	14	13	12	11	10	9	8
n/a			Reserved		n/a	Reserved	
7	6	5	4	3	2	1	0

Bits 10 to 9 (RSV): **Reserved (0)**

Bit 8: **ACTAGAIN (Write Only)**

Writing "1" to this bit produces lock-step JPEG encoding, feeding frames to the JPEG codec with no gaps between frames. Whether such encoding is possible depends on the system and software specifications. For further details, see the description of operation below.

Bits 4 to 3: **Reserved (0)**

Always set to "0".

Bits 1 to 0: **Reserved (0)**

Always set to "0".

Capture Control State Register							
RSZ[0x64]	Default = 0x0000						Read Only
n/a							
15	14	13	12	11	10	9	8
n/a				State Value			
7	6	5	4	3	2	1	0

Bits 3 to 0: **State**

This gives the capture control sequence state machine's current state. For the meaning, see the description of operation in Section 8.5.1 "Capture Control."

Capture Data Setting Register							
RSZ[0x68]	Default = 0x0000						Read/Write
n/a							
15	14	13	12	11	10	9	8
n/a							Data Format Select
7	6	5	4	3	2	1	0

Bit 0: **Data Format Select for Image Capture**

0: YUV data

1: JPEG data

This specifies the data format for image capture.

Setting this bit to "1" invalidates all RSZ registers except RSZ[0xC0]. Bits in read-only registers may change, but their contents are not valid.

Reserved Registers RSZ[0x70-7C] Default = 0x0000							Read/Write
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0

Bits 15 to 0: **Reserved (0)**

Always set to "0".

Capture Resizer Control Register RSZ[0xC0] Default = 0x0000							Read/Write
n/a							
15	14	13	12	11	10	9	8
Capture Resizer Software Reset (WO)	n/a			Reserved (0)			Capture Resizer Enable
7	6	5	4	3	2	1	0

Bit 7: **Capture Resizer software reset (Write Only)**

Writing "1" to this bit produces a capture resizer software reset. Writing "0" to this bit does nothing.

Bits 3 to 1: **Reserved (0)**

Always set to "0".

Bit 0: **Capture Resizer Enable**

Setting this bit to "0" stops the clock signal to the capture resizer, helping reduce power consumption, but does not prevent read/write access to the its registers.

Write:

0: Disable

1: Enable

Read:

0: Disable

1: Enable

Capture Resizer Start X Position Register RSZ[0xC8] Default = 0x0000							Read/Write
n/a					Capture Resizer Start X Position bits [10:8]		
15	14	13	12	11	10	9	8
Capture Resizer Start X Position bits [7:0]							
7	6	5	4	3	2	1	0

Bits 10 to 0: **Capture Resizer Starting X Position [10:0]**

These bits determine the X start position for of the capture resizer.

Capture Resizer Start Y Position Register RSZ[0xCC] Default = 0x0000							Read/Write
n/a					Capture Resizer Start Y Position bits [10:8]		
15	14	13	12	11	10	9	8
Capture Resizer Start Y Position bits [7:0]							
7	6	5	4	3	2	1	0

Bits 10 to 0: **Capture Resizer Starting Y Position [10:0]**

These bits determine the Y start position for of the capture resizer.

Capture Resizer End X Position Register
RSZ[0xD0] Default = 0x027F Read/Write

n/a					Capture Resizer End X Position bits [10:8]		
15	14	13	12	11	10	9	8
Capture Resizer End X Position bits [7:0]							
7	6	5	4	3	2	1	0

Bits 10 to 0: **Capture Resizer End X Position [10:0]**
These bits determine the X start position for of the capture resizer.

Capture Resizer End Y Position Register
RSZ[0xD4] Default = 0x01DF Read/Write

n/a					Capture Resizer End Y Position bits [10:8]		
15	14	13	12	11	10	9	8
Capture Resizer End Y Position bits [7:0]							
7	6	5	4	3	2	1	0

Bits 10 to 0: **Capture Resizer End Y Position [10:0]**
These bits determine the Y start position for of the capture resizer.

Capture Resizer Scaling Rate Register
RSZ[0xD8] Default = 0x8080 Read/Write

Reserved (0)							
15	14	13	12	11	10	9	8
Reserved (0)				Capture Resizer Scaling Rate bits [3:0]			
7	6	5	4	3	2	1	0

Bits 15 to 4: **Reserved (0)**
Always set to "0".

Bits 3 to 0: **Capture Resizer Scaling Rate [3:0]**
These bits specify the scaling rate (1/n) with the capture resizer.
Note that the choices available depend on the scaling mode. For further details, see Table 8.4 "Capture Resizer Scaling Rate/Mode Selection" below.

Table 8.2 Capture Resizer Scaling Rate Setting

Bits [3:0]	Capture Resizer Scaling Rate Setting
0000	Reserved
0001	1/1
0010	1/2
0011	1/3 (V/H Reduction only, Capture Resizer Scaling Mode Register bit 1-0=01)
0100	1/4
0101	1/5 (V/H Reduction only, Capture Resizer Scaling Mode Register bit 1-0=01)
0110	1/6 (V/H Reduction only, Capture Resizer Scaling Mode Register bit 1-0=01)
0111	1/7 (V/H Reduction only, Capture Resizer Scaling Mode Register bit 1-0=01)
1000	1/8
1001-1111	Reserved

<table>
<tr><td colspan="8"><b>Capture Resizer Scaling Mode Register</b>
RSZ[0xDC] Default = 0x0000 Read/Write</td></tr>
<tr><td colspan="8">n/a</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="4">n/a</td><td colspan="2">Reserved
(0)</td><td colspan="2">Capture Resizer Scaling Mode
bits [1:0]</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
</table>

Bits 3 to 2 (RSV): **Reserved (0)**

Always set to “0”.

Bits 1 to 0: **Capture Resizer Scaling Mode bits [1:0]**

These bits specify the capture resizer scaling mode.

This setting affects the choices available in the Capture Resizer Scaling Rate Register. See Table 8.4 “Capture Resizer Scaling Rate/Mode Selection”.

Table 8.3 Capture Resizer Scaling Mode Selection

Bits [1:0]	Capture Resizer Scaling Mode
00	No resizer scaling
01	Both V and H reduction
10	Vertical sampling following by horizontal averaging V: Reduction, H: Average
11	Reserved

Table 8.4 Capture Resizer Scaling Rate/Mode Selection

		RSZ[0xDC] Bits [1:0]			
		00	**01**	**10**	**11**
RSZ[0xD8] Bits [3:0]	**0000**	1/1	Reserved	Reserved	Reserved
	0001	1/1	Reserved	Reserved	Reserved
	0010	1/1	1/2	1/2	Reserved
	0011	1/1	1/3	Reserved	Reserved
	0100	1/1	1/4	1/4	Reserved
	0101	1/1	1/5	Reserved	Reserved
	0110	1/1	1/6	Reserved	Reserved
	0111	1/1	1/7	Reserved	Reserved
	1000	1/1	1/8	1/8	Reserved
	other	Reserved	Reserved	Reserved	Reserved

8.4.3 JPEG Module Registers (JCTL)

JPEG Control Register
JCTL[0x00] Default = 0x0000 Read/Write

JPEG Encode Fast Mode	JPEG Marker Fast Output Mode	Reserved (0)					JPEG 180° Rotation Enable
15	14	13	12	11	10	9	8
JPEG Module SW Reset (WO)	Reserved (0)		UV Data Type Conversion	Operation Mode bits [2:0]			JPEG Module Enable
7	6	5	4	3	2	1	0

Bit15: **JPEG Encode Fast Mode**

0: (ignored)
1: Use fixed Huffman table to accelarate JPEG compression processing.

Note: Using this mode requires loading the JPEG Huffman tables (JCODEC [0x800]-[0xF44]) with the values specified in ISO/IEC 10918-1 Annex K or setting the appropriate bit in the Huffman Automatic Setting Register (JCTL[0x20]) to “1.”

Bit14: **JPEG Marker Fast Output Mode**

0: (ignored)
1: Use fixed Huffman table to accelarate JPEG marker output processing.

Note: This bit is only valid when bit 15 (JPEG encode fast mode) is set to “1.” Using this mode does not require loading the JPEG Huffman tables (JCODEC[0x800]-[0xF44]). The values specified in ISO/IEC 10918-1 Annex K are automatically used.

Bits 13 to 9 (RSV): **Reserved (0)**

Always set to “0”.

Bit 8: **JPEG 180° Rotation Enable**

0: Disable
1: Enable

This controls data rotation during JPEG encoding.

Note that hardware rotation applies only by specific image line, which the software must subsequently rearrange the data in the JPEG file. For further details, see Section 8.5.5.5 “JPEG 180° Rotation Encode.”

Bit 7: **JPEG Module Software Reset (Write Only)**

0: (ignored)
1: Reset

Writing “1” to this bit resets JPEG module except the JPEG codec and the registers. Always reset before starting JPEG encoding. Note that the write is ignored, however, if JCTL[0x00] bit 0 (JPEG module enable) is “0.”

Bits 6 to 5 (RSV): **Reserved (0)**

Always set to “0”.

Bit 4: **UV Data Type Conversion Disable**

0: Enable
1: Disable

Setting this bit to “1” disables automatic conversion of UV data input.

Table 8.5 UV Data Type

Bit 4	Camera Data Format	Internal Data Format
0 (Conversion)	0 ≤ U ≤ 255 0 ≤ V ≤ 255	–128 ≤ U ≤ 127 –128 ≤ V ≤ 127
	16 ≤ Cb ≤ 240 16 ≤ Cr ≤ 240	–112 ≤ Cb ≤ 112 –112 ≤ Cr ≤ 112
	–128 ≤ U ≤ 127 –128 ≤ V ≤ 127	0 ≤ U ≤ 255 0 ≤ V ≤ 255
	–112 ≤ Cb ≤ 112 –112 ≤Cr ≤ 112	16 ≤ Cb ≤ 240 16 ≤ Cr ≤ 240
1 (Non-conversion)	0 ≤ U ≤ 255 0 ≤ V ≤ 255	0 ≤ U ≤ 255 0 ≤ V ≤ 255
	16 ≤ Cb ≤ 240 16 ≤ Cr ≤ 240	16 ≤ Cb ≤ 240 16 ≤ Cr ≤ 240
	–128 ≤ U ≤ 127 –128 ≤ V ≤ 127	–128 ≤ U ≤ 127 –128 ≤ V ≤ 127
	–112 ≤ Cb ≤ 112 –112 ≤ Cr ≤ 112	–112 ≤ Cb ≤ 112 –112 ≤ Cr ≤ 112

Bits 3 to 1: **Operation Mode Selection**

This specifies the JPEG operation mode.

Note that specifying YUV data capture (“x11”) instead of JPEG encode or decode (“x00”) here stops the clock signal to the JPEG codec, preventing access to the JPEG Codec Registers. Changing register settings therefore requires switching back.

JCODEC[0x00] bits 1 to 0 specify the YUV data format for Camera Image JPEG encode and host input JPEG encode/decode, but these register bits specify it for YUV data capture.

Table 8.6 JPEG Operation Mode

Bits [3:1]	JPEG Operation Mode
000 (default)	Camera Image JPEG encode for YUV data formats 4:4:4*, 4:2:2, 4:1:1, or 4:2:0 * Converting from the YUV 4:4:4 format requires first scaling the Image with the capture resizer by a factor of at least 1/2.
001	Reserved
010	Reserved
011	YUV Data Capture (YUV 4:2:2)
100	Host Input JPEG encode/decode (YUV 4:2:2 or 4:2:0)
101	Reserved
110	Reserved
111	YUV data capture (YUV 4:2:0)

Bit 0: **JPEG Module Enable**

0: Disable (default)

1: Enable

This controls availability of the JPEG module.

Note that disabling the JPEG module stops the clock signal to it, preventing access to the JPEG Codec Registers.

Always disable the JPEG module before disabling the capture resizer.

JPEG Status Flag Register
JCTL[0x04] Default = 0x8080 Read/Write

Reserved (1) —	JPEG Codec File Out Status (RO)	JPEG FIFO Threshold Status [1:0] (RO)		Encode Size Limit Violation Flag (R/W)	JPEG FIFO Threshold Trigger Flag (R/W)	JPEG FIFO Full Flag (R/W)	JPEG FIFO Empty Flag (R/W)
15	14	13	12	11	10	9	8
Reserved (1) —			JPEG Decode Marker Detected Flag (R/W)	Reserved —	JPEG Line Buffer Overflow Flag (RO)	JPEG Codec Interrupt Flag (RO)	JPEG Line Buffer Interrupt Flag (RO)
7	6	5	4	3	2	1	0

Bit 15 (RSV): **Reserved (1)**

Always set to "0".

Bit 14: **JPEG Codec File Out Status (Read Only)**

0: Idle
1: Busy encoding or sending JPEG file

This indicates the JPEG codec output state for JPEG encoding.
Always write "1" to this bit.

Bits 13 to 12: **JPEG FIFO Threshold Status (Read Only)**

This gives the JPEG FIFO's current threshold status.
Always write "1" to these bits.

Table 8.7 JPEG FIFO Threshold Status

Bits [13:12]	JPEG FIFO Threshold Status
00	Empty
01	At least four bytes, but less than 1/4 full of FIFO size
10	At least 1/4 full, but less than 1/2 full of FIFO size
11	At least 1/2 full of FIFO size

Bit 11: **Encode Size Limit Violation Flag**

reads 0: There is no interrupt request
1: There is an interrupt request
writes 0: (ignored)
1: Clear

"1" in this bit during JPEG encode indicates that the JPEG file size exceeds the limit specified in the Encode Size Limit Register. Note that this flag is just a warning that does not stop JPEG encode.
Setting the corresponding bit in the JPEG Interrupt Control Register (JCTL[0x0C] bit 11) to "0" clears the interrupt request and this flag.

Bit 10: **JPEG FIFO Threshold Trigger Flag**

reads 0: There is no interrupt request
1: There is an interrupt request
writes 0: (ignored)
1: Clear

"1" in this bit indicates that the JPEG FIFO data size has exceeded the JPEG FIFO threshold (JPEG FIFO Control Register bits 5 to 4) at least once.
Setting the corresponding bit in the JPEG Interrupt Control Register (JCTL[0x0C] bit 10) to "0" clears the interrupt request and this flag.

Bit 9: **JPEG FIFO Full Flag**

reads 0: There is no interrupt request
1: There is an interrupt request
writes 0: (ignored)
1: Clear

"1" in this bit indicates that the JPEG FIFO has been full at least once.

Setting the corresponding bit in the JPEG Interrupt Control Register (JCTL[0x0C] bit 9) to "0" clears the interrupt request and this flag.

Bit 8: **JPEG FIFO Empty Flag**

reads 0: There is no interrupt request
1: There is an interrupt request
writes 0: (ignored)
1: Clear

"1" in this bit indicates that the JPEG FIFO has been empty at least once.

Setting the corresponding bit in the JPEG Interrupt Control Register (JCTL[0x0C] bit 8) to "0" clears the interrupt request and this flag.

Bits 7 to 5 (RSV): **Reserved (1)**

Always set to "1".

Bit 4: **JPEG Decode Marker Read Flag**

reads 0: There is no interrupt request
1: There is an interrupt request
writes 0: (ignored)
1: Clear

"1" in this bit indicates that a marker has been read from the JPEG file during JPEG decode.

Setting the corresponding bit in the JPEG Interrupt Control Register (JCTL[0x0C] bit 4) to "0" clears the interrupt request and this flag.

Bit 3: **Reserved (1)**

Always set to "1".

Bit 2: **JPEG Line Buffer Overflow Flag (Read Only)**

0: There is no interrupt request
1: There is an interrupt request

"1" in this bit indicates JPEG line buffer overflow.

Setting the corresponding bit in the JPEG Interrupt Control Register (JCTL[0x0C] bit 2) to "0" clears the interrupt request and this flag. Setting JCTL[0x00] bit 7 (JPEG Module Software Reset) to "1" also clears this flag.

Always write "1" to this bit.

Note that this flag is just a warning that does not stop JPEG module operation. The overflow destroys data, however.

Bit 1: **JPEG Codec Interrupt Flag (Read Only)**

0: There is no interrupt request
1: There is an interrupt request

"1" in this bit indicates that there is a JPEG codec interrupt request.

Setting the corresponding bit in the JPEG Interrupt Control Register (JCTL[0x0C] bit 1) to "0" clears the interrupt request and this flag.

Reading bit 0 in the JPEG Operation Status Register (JCODEC[0x08]) resets this bit to "0."

Bit 0: **JPEG Line Buffer Interrupt Flag (Read Only)**

0: There is no interrupt request

1: There is an interrupt request

"1" in this bit indicates that an interrupt has occurred during host JPEG encode/decode.

Setting the corresponding bit in the JPEG Interrupt Control Register (JCTL[0x0C] bit 0) to "0" clears the interrupt request and this flag. Clearing the JPEG Line Buffer Status Flag Register (JLB[0x80]) also clears this flag.

JPEG Raw Status Flag Register

JCTL[0x08] Default = 0x8080 Read Only

Reserved	Raw JPEG Codec File Out Status	Raw JPEG FIFO Threshold Status [1:0]		Raw Encode Size Limit Violation Flag	Raw JPEG FIFO Threshold Trigger Flag	Raw JPEG FIFO Full Flag	Raw JPEG FIFO Empty Flag
15	14	13	12	11	10	9	8
Reserved			Raw Decode Marker Detected Flag	Reserved	Raw JPEG Line Buffer Overflow Flag	Raw JPEG Codec Interrupt Request Flag	Raw JPEG Line Buffer Interrupt Request Flag
7	6	5	4	3	2	1	0

Bit 15: **Reserved**

Bit 14: **Raw JPEG Codec File Out Status**

0: Idle

1: Busy encoding or sending JPEG file

This indicates the JPEG codec output state for JPEG encoding.

Bits 13 to 12: **Raw JPEG FIFO Threshold Status**

This gives the JPEG FIFO's current data state.

Table 8.8 JPEG FIFO Threshold Status

Bits [13:12]	JPEG FIFO Threshold Status
00	Empty
01	At least four bytes, but less than FIFO full size
10	At least 1/4 full, but less than 1/2 FIFO full size
11	At least 1/2 FIFO full size

Bit 11: **Raw Encode Size Limit Violation Flag**

0: JPEG file size is within the limit

1: JPEG file size exceeds the limit

"1" in this bit during JPEG encoding indicates that the JPEG file size exceeds the limit specified in the Encode Size Limit Register 1 to 0 (JFIFO [0x60, 0x64]).

This bit is independent of the corresponding bit in the JPEG Interrupt Control Register (JCTL[0x0C]).

Setting the corresponding bit in the JPEG Status Flag Register (JCTL[0x04] bit 11) to "1" clears this flag.

Note that this flag is just a warning that does not stop JPEG encoding.

Bit 10: **Raw JPEG FIFO Threshold Trigger Flag**

0: JPEG FIFO data size is within the JPEG FIFO threshold

1: JPEG FIFO data size exceeds the JPEG FIFO threshold

"1" in this bit indicates that the JPEG FIFO data size has exceeded the JPEG FIFO threshold (JPEG FIFO Control Register bits 5 to 4) at least once.

This bit is independent of the corresponding bit in the JPEG Interrupt Control Register (JCTL[0x0C] bit 10). Setting the corresponding bit in the JPEG Status Flag Register (JCTL[0x04] bit 10) to "1" clears this flag.

Bit 9: **Raw JPEG FIFO Full Flag**

0: JPEG FIFO is not full

1: JPEG FIFO is full

"1" in this bit indicates that the JPEG FIFO has been full at least once.
This bit is independent of the corresponding bit in the JPEG Interrupt Control Register (JCTL[0x0C] bit 9).
Setting the corresponding bit in the JPEG Status Flag Register (JCTL[0x04] bit 9) to "1" clears the interrupt request and this flag.

Bit 8: **Raw JPEG FIFO Empty Flag**

0: JPEG FIFO is not empty

1: JPEG FIFO is empty

"1" in this bit indicates that the JPEG FIFO has been empty at least once.
This bit is independent of the corresponding bit in the JPEG Interrupt Control Register (JCTL[0x0C] bit 8).
Setting the corresponding bit in the JPEG Status Flag Register (JCTL[0x04] bit 8) to "1" clears the interrupt request and this flag.

Bits 7 to 5 (RSV): **Reserved**

Bit 4: **Raw JPEG Decode Marker Read Flag**

0: Marker read has not completed

1: Marker read has completed

"1" in this bit indicates that a marker has been read from the JPEG file during JPEG decoding.
This bit interacts with the corresponding bit in the JPEG Interrupt Control Register (JCTL[0x0C] bit 4). If the latter is "0," this bit never changes from "0" to "1." But changing the latter from "1" to "0" does not reset this bit to "0."
Setting the corresponding bit in the JPEG Status Flag Register (JCTL[0x04] bit 4) to "1" clears this flag.

Bit 3 (RSV): **Reserved**

Bit 2: **Raw JPEG Line Buffer Overflow Flag**

0: JPEG line buffer doesn't overflow

1: JPEG line buffer overflows

"1" in this bit indicates JPEG line buffer overflow. This bit is independent of the corresponding bit in the JPEG Interrupt Control Register (JCTL[0x0C] bit 2).
Setting JCTL[0x00] bit 7 (JPEG Module Software Reset) to "1" clears this flag.

Bit 1: **Raw JPEG Codec Interrupt Flag**

0: There is no JPEG codec interrupt request

1: There is a JPEG codec interrupt request

"1" in this bit indicates that a JPEG codec interrupt has occurred.
This bit is independent of the corresponding bit in the JPEG Interrupt Control Register (JCTL[0x0C] bit 1).
Reading bit 0 in the JPEG Operation Status Register (JCODEC[0x08]) clears this flag to "0."

Bit 0: **Raw JPEG Line Buffer Interrupt Flag**

0: There is no JPEG line buffer interrupt request

1: There is a JPEG line buffer interrupt request

"1" in this bit indicates that a JPEG line buffer interrupt request has occurred during YUV data capture.
This bit is independent of the corresponding bit in the JPEG Interrupt Control Register (JCTL[0x0C] bit 0).
Clearing the JPEG Line Buffer Status Flag Register (JLB [0x80]) clears this flag.

JPEG Interrupt Control Register
JCTL[0x0C] Default = 0x0000 Read/Write

Reserved (0)				Encode Size Limit Violation Interrupt Enable	JPEG FIFO Threshold Trigger Interrupt Enable	JPEG FIFO Full Interrupt Enable	JPEG FIFO Empty Interrupt Enable
15	14	13	12	11	10	9	8
Reserved (0)			Decode Marker Detected Interrupt Enable	Reserved (0)	JPEG Line Buffer Overflow Interrupt Enable	JPEG Codec Interrupt Enable	JPEG Line Buffer Interrupt Enable
7	6	5	4	3	2	1	0

Bits 15 to 12 (RSV): **Reserved (0)**
Always set to “0”.

Bit 11: **Encode Size Limit Violation Interrupt Enable**
0: Disable (default)
1: Enable
Setting this bit to “1” enables Encode size limit violation interrupt.

Bit 10: **JPEG FIFO Threshold Trigger Interrupt Enable**
0: Disable (default)
1: Enable
Setting this bit to “1” enables JPEG FIFO threshold trigger interrupt.

Bit 9: **JPEG FIFO Full Interrupt Enable**
0: Disable (default)
1: Enable
Setting this bit to “1” enables JPEG FIFO full interrupt.

Bit 8: **JPEG FIFO Empty Interrupt Enable**
0: Disable (default)
1: Enable
Setting this bit to “1” enables JPEG FIFO empty interrupt.

Bits 7 to 5 (RSV): **Reserved (0)**
Always set to “0”.

Bit 4: **JPEG Decode Marker Read Interrupt Enable**
0: Disable (default)
1: Enable
Setting this bit to “1” enables JPEG decode marker read interrupt.
If this bit set to “1” (Enable), detecting a JPEG decode marker read automatically suspends JPEG decode operation. Setting this bit to “0” resumes it.

Bit 3 (RSV): **Reserved (0)**
Always set to “0”.

Bit 2: **JPEG Line Buffer Overflow Interrupt Enable**
0: Disable (default)
1: Enable
Setting this bit to “1” enables JPEG line buffer overflow interrupt.

Bit 1: **JPEG Codec Interrupt Enable**
0: isable (default)
1: nable
Setting this bit to “1” enables JPEG codec interrupt.

Bit 0: **JPEG Line Buffer Interrupt Enable**

0: isable (default)
1: nable

Setting this bit to "1" enables JPEG line buffer interrupt.

This bit is the master enable control bit for the enable bits in the JPEG Line Buffer Interrupt Control Register (JLB[0x8C]). Setting this bit to "0" disables all enable bits "1" in that register. It does not affect JPEG line buffer overflow interrupt requests, however.

JPEG Codec Start/Stop Control Register							
JCTL[0x14]	Default = 0x0000						Write Only
n/a							
15	14	13	12	11	10	9	8
n/a							JPEG Start/Stop Control
7	6	5	4	3	2	1	0

Bit 0: **JPEG Start/Stop Control**

JPEG encode	0: Encode cancel (Cancel the encode start unless it has started encode)
	1: Encode start (Start from next frame)
YUV data capture	0: Capture stop (Stop after current frame)
	1: Capture start (Start from next frame)

This bit controls JPEG module operation—JPEG encoding and YUV data capture, but not JPEG decoding.

Huffman Table Automatic Setting Register							
JCTL[0x20]	Default = 0x0000						Read/Write
n/a							
15	14	13	12	11	10	9	8
n/a						Huffman Table Automatic Setting Non-standby Mode	Huffman Table Automatic Setting
7	6	5	4	3	2	1	0

Bit 1: **Huffman Table Automatic Setting Non-standby Mode**

When JPEG codec core is set for encode, setting this bit to "1" at the same time of writing "1" to bit 0 enables access to registers except the JCODEC Registers. The "0" setting enables the access to the other registers after Huffman table setting has completed.

Note that this bit is invalid except writing to bit 0 at the same time writing to this bit.

Reads always return "0."

Bit 0: **Huffman Table Automatic Setting**

When JPEG codec core is set for encode, writing "1" to this bit automatically loads the JPEG Huffman tables (JCODEC [0x800]-[0xF44]) with the values specified in ISO/IEC 10918-1 Annex K. Note that such a write is ignored, however, if JPEG encoding with the JPEG codec core is not enabled or the value written is "0."

Setting this bit to "1" disables access to the JCODEC Registers. (All become dummy writes.) It also disables write access to this register.

This bit automatically returns to "0" when Huffman table loading is complete.

8.4.4 JPEG FIFO Setting Registers (JFIFO)

JPEG FIFO Control Register
JFIFO[0x40] Default = 0x0000 Read/Write

15	14	13	12	11	10	9	8
Reserved (0)							
Reserved (0)		JPEG FIFO Trigger Threshold bits [1:0]		Reserved (0)	JPEG FIFO Clear	JPEG FIFO Direction (RO)	Reserved (0)
7	6	5	4	3	2	1	0

Bits 15 to 6 (RSV): **Reserved (0)**

Always set to “0”.

Bits 5 to 4: **JPEG FIFO Trigger Threshold**

These bits specify the JPEG FIFO trigger threshold.

Table 8.9 JPEG FIFO Trigger Thresholds

Bits [5:4]	JPEG FIFO Trigger Threshold
00	Never
01	At least four bytes
10	At least 1/4 FIFO full size
11	At least 1/2 FIFO full size

Bit 3 (RSV): **Reserved (0)**

Always set to “0”.

Bit 2: **JPEG FIFO Clear**

0: (ignored)
1: Clear JPEG FIFO

Writing “1” to this bit clears the JPEG FIFO. Always follow this with a write of “1” to JCTL[0x00] bit 7 (JPEG module software reset) to reset JPEG FIFO.

Bit 1: **JPEG FIFO Codec Direction (Read Only)**

0: Receive (JPEG encode)
1: Transmit (JPEG decode)

This indicates the JPEG FIFO direction.

Bit 0 (RSV): **Reserved (0)**

Always set to “0”.

JPEG FIFO Status Register
JFIFO[0x44] Default = 0x8001 Read Only

15	14	13	12	11	10	9	8
Reserved							
Reserved				JPEG FIFO Threshold Status bits [1:0]		JPEG FIFO Full Status	JPEG FIFO Empty Status
7	6	5	4	3	2	1	0

Bits 15 to 4: **Reserved**

Bits 3 to 2: **JPEG FIFO Threshold Status**

This gives the JPEG FIFO’s current data status.

Table 8.10 JPEG FIFO Threshold Status

Bits [3:2]	JPEG FIFO Threshold Status
00	Empty
01	At least four bytes, but less than 1/4 FIFO full size
10	At least 1/4 FIFO full size, but less than 1/2 FIFO full size
11	At least 1/2 FIFO full size

Bit 1: **JPEG FIFO Full Status**

0: Not full
1: Full

"1" in this bit indicates that the JPEG FIFO is currently full.

Bit 0: **JPEG FIFO Empty Status**

0: Not empty
1: Empty

"1" in this bit indicates that the JPEG FIFO is currently empty.

JPEG FIFO Size Register							
JFIFO[0x48] Default = 0x003F							Read/Write
Reserved (0)	JPEG FIFO Size bits [14:8]						
15	14	13	12	11	10	9	8
JPEG FIFO Size bits [7:0]							
7	6	5	4	3	2	1	0

Bit 15 (RSV): **Reserved (0)**

Always set to "0".

Bits 14 to 0: **JPEG FIFO Size**

The JPEG FIFO size is wording set. The maximum values of JPEG FIFO are 64 words. Because JPEG FIFO uses special RAM, 64 words that are the maximum values will usually be set.This register, less one, specifies the JPEG FIFO size in words.

JPEG FIFO size in words = Register bits [14:0] + 1

Note: The maximum possible setting and default is 0x003F, for a size of 64 words. We recommend using that setting because there is normally no reason to waste the JPEG FIFO's dedicated RAM by specifying a smaller size. The only other possible settings are all strings of 1s (2^{n-1}, n=0 to 5): 0x001F, 0x000F, 0x0007, 0x0003, 0x0001, and 0x0000. Do not use any other settings.

JPEG FIFO Read/Write Port Register							
JFIFO[0x4C] Default = 0x0000 0000							Read/Write
JPEG FIFO Read/Write Port bits [31:24]							
31	30	29	28	27	26	25	24
JPEG FIFO Read/Write Port bits [23:16]							
23	22	21	20	19	18	17	16
JPEG FIFO Read/Write Port bits [15:8]							
15	14	13	12	11	10	9	8
JPEG FIFO Read/Write Port bits [7:0]							
7	6	5	4	3	2	1	0

Bits 31 to 0: **JPEG FIFO Read/Write Port**

This is for reading data from the JPEG FIFO during JPEG encode and YUV data capture and for writing data to the JPEG FIFO during JPEG decode.

Reserved Registers							
JFIFO[0x50, 0x54, 0x58] Default = —							—/—
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0

Encode Size Limit Register 0							
JFIFO[0x60]	Default = 0x0000						Read/Write
Encode Size Limit bits [15:8]							
15	14	13	12	11	10	9	8□
Encode Size Limit bits [7:0]							
7	6	5	4	3	2	1	0

Encode Size Limit Register 1							
JFIFO[0x64]	Default = 0x0000						Read/Write
n/a							
15	14	13	12	11	10	9	8
Encode Size Limit bits [23:16]							
7	6	5	4	3	2	1	0

Encode Size Limit Register 1 bits [7:0]
Encode Size Limit Register 0 bits [15:0]

Encode Size Limit - bits[23:0]

Together these two registers specify, in bytes, a limit on JPEG file data size for JPEG encode. An encode size limit violation interrupt occurs if the JPEG file size exceeds the setting in this register. Note that this warning does not affect the JPEG encoding itself.

JPEG decode does not use these registers.

Encode Size Result Register 0							
JFIFO[0x68]	Default = 0x0000						Read Only
Encode Size Result bits [15:8]							
15	14	13	12	11	10	9	8
Encode Size Result bits [7:0]							
7	6	5	4	3	2	1	0

Encode Size Result Register 1							
JFIFO[0x6C]	Default = 0x0000						Read Only
n/a							
15	14	13	12	11	10	9	8
Encode Size Result bits [23:16]							
7	6	5	4	3	2	1	0

Encode Size Result Register 1 bits [7:0]
Encode Size Result Register 0 bits [15:0]

Encode Size Result bits[23:0]

Together these two registers give the JPEG file data size for JPEG encode. The contents are only correct when JPEG encode is complete.

JPEG decode does not use these registers.

Reserved Registers							
JFIFO[0x70, 0x74, 0x78]	Default = —						—/—
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0

8.4.5 JPEG Line Buffer Setting Registers (JLB)

JPEG Line Buffer Status Flag Register
JLB[0x80] Default = 0x0000 Read/Write

n/a							
15	14	13	12	11	10	9	8
n/a			Reserved (1)	JPEG Line Buffer Empty Flag	JPEG Line Buffer Full Flag	JPEG Line Buffer Half Full Flag	Reserved (1)
7	6	5	4	3	2	1	0

Bit 4 (RSV): **Reserved (1)**
Always set to “1”.

Bit 3: **JPEG Line Buffer Empty Flag**
“1” in this bit indicates that the JPEG line buffer has been empty at least once.
Setting the corresponding bit in the JPEG Line Buffer Interrupt Control Register (JLB[0x8C] bit 3) to “0” disables these interrupt requests.
reads 0: There is no interrupt request
1: There is an interrupt request
writes 0: (ignored)
1: Clear

Bit 2: **JPEG Line Buffer Full Flag**
“1” in this bit indicates that the JPEG line buffer has been full at least once.
Setting the corresponding bit in the JPEG Line Buffer Interrupt Control Register (JLB[0x8C] bit 2) to “0” disables these interrupt requests.
reads 0: There is no interrupt request
1: There is an interrupt request
writes 0: (ignored)
1: Clear

Bit 1: **JPEG Line Buffer Half Full Flag**
“1” in this bit indicates that the JPEG line buffer has been half full at least once.
Setting the corresponding bit in the JPEG Line Buffer Interrupt Control Register (JLB[0x8C] bit 1) to “0” disables these interrupt requests.
reads 0: There is no interrupt request
1: There is an interrupt request
writes 0: (ignored)
1: Clear

Bit 0: **Reserved (1)**
Always set to “1”.

JPEG Line Buffer Raw Status Flag Register
JLB[0x84] Default = 0x0000 Read Only

n/a							
15	14	13	12	11	10	9	8
n/a			Reserved (0)	Raw JPEG Line Buffer Empty Flag	Raw JPEG Line Buffer Full Flag	Raw JPEG Line Buffer Half Full Flag	Reserved (0)
7	6	5	4	3	2	1	0

Bit 4: **Reserved (0)**

Bit 3: **Raw JPEG Line Buffer Empty Flag**

"1" in this bit indicates that the JPEG line buffer has been empty at least once. This bit is independent of the corresponding bit in the JPEG Line Buffer Interrupt Control Register (JLB[0x8C]).

0: Not empty
1: Empty

Bit 2: **Raw JPEG Line Buffer Full Flag**

"1" in this bit indicates that the JPEG line buffer has been full at least once. This bit is independent of the corresponding bit in the JPEG Line Buffer Interrupt Control Register (JLB[0x8C]).

0: Not full
1: Full

Bit 1: **Raw JPEG Line Buffer Half Full Flag**

"1" in this bit indicates that the JPEG line buffer has been half full at least once. This bit is independent of the corresponding bit in the JPEG Line Buffer Interrupt Control Register (JLB[0x8C]).

0: Not half full
1: Half full

Bit 0: **Reserved (0)**

JPEG Line Buffer Current Status Flag Register
JLB[0x88] Default = 0x0009 Read Only

n/a							
15	14	13	12	11	10	9	8
n/a			Reserved	JPEG Line Buffer Empty Current Status	JPEG Line Buffer Full Current Status	JPEG Line Buffer Half Full Current Status	Reserved
7	6	5	4	3	2	1	0

Bit 4: **Reserved (0)**

Bit 3: **JPEG Line Buffer Empty Current Status**

0: Not empty
1: Empty

"1" in this bit indicates that the JPEG Line Buffer is currently empty.

Bit 2: **JPEG Line Buffer Full Current Status**

"1" in this bit indicates that the JPEG Line Buffer is currently full.

0: Not full
1: Full

Bit 1: **JPEG Line Buffer Half Full Current Status**

"1" in this bit indicates that the JPEG Line Buffer is currently half full.

0: Not half full
1: Half full

Bit 0: **Reserved (0)**

JPEG Line Buffer Interrupt Control Register							
JLB[0x8C]	Default = 0x0000						Read/Write
n/a							
15	14	13	12	11	10	9	8
n/a			Reserved (0)	JPEG Line Buffer Empty Interrupt Enable	JPEG Line Buffer Full Interrupt Enable	JPEG Line Buffer Half Full Interrupt Enable	Reserved (0)
7	6	5	4	3	2	1	0

Bit4 (RSV): **Reserved (0)**

Always set to “0”.

Bit 3: **JPEG Line Buffer Empty Interrupt Enable**

0: Disable (default)
1: Enable

Setting this bit to “1” enables JPEG line buffer empty interrupt.

Bit 2: **JPEG Line Buffer Full Interrupt Enable**

0: Disable (default)
1: Enable

Setting this bit to “1” enables JPEG line buffer full interrupt.

Bit 1: **JPEG Line Buffer Half Full Interrupt Enable**

0: Disable (default)
1: Enable

Setting this bit to “1” enables JPEG line buffer half full interrupt.

Bit 0: **Reserved (0)**

Always set to “0”.

JPEG Line Buffer Horizontal Pixel Support Size Register							
JLB[0xA0]	Default = 0x2800						Read/Write
Horizontal Support Size bits [10:4]							
15	14	13	12	11	10	9	8
Horizontal Support Size bits [3:0]				n/a	Horizontal Support Size Setting bits [2:0]		
7	6	5	4	3	2	1	0

Bits 15 to 4: **JPEG Line Buffer Horizontal Pixel Support Size bits [10:0] (Read Only)**

This gives the horizontal pixel support size specified in bits 2 to 0.

Bits 2 to 0: **JPEG Line Buffer Horizontal Pixel Support Size Setting [2:0]**

This specifies the horizontal pixel support size for the JPEG Line Buffer.

Note that using this register with a value other than the default requires modifying the settings in the JPEG Line Buffer Memory Address Offset Register (JLB[0xA4]) and System Controller Embedded Memory Control Register to match.

000: 640 (default)
001: 800
010: 1024
011: 1280
100: 1600
101-111: Reserved

JPEG Line Buffer Memory Address Offset Register
JLB[0xA4] Default = 0x0030 Read/Write

Reserved (0)							
15	14	13	12	11	10	9	8
Reserved (0)	JPEG Line Buffer Memory Address Offset bits [6:0]						
7	6	5	4	3	2	1	0

Bits 15 to 7 (RSV): **Reserved (0)**

Always set to “0.”

Bits 6 to 0: **JPEG Line Buffer Memory Address Offset**

This specifies, in KB, the internal memory address offset for the JPEG line buffer.

Note that using this register with a value other than the default requires modifying the setting in the System Controller Embedded Memory Control Register to match.

JPEG Line Buffer Read/Write Port Register
JLB[0xC0] Default = 0x0000_0000 Read/Write

JPEG Line Buffer Read/Write Port bits [31:24]							
31	30	29	28	27	26	25	24
JPEG Line Buffer Read/Write Port bits [23:16]							
23	22	21	20	19	18	17	16
JPEG Line Buffer Read/Write Port bits [15:8]							
15	14	13	12	11	10	9	8
JPEG Line Buffer Read/Write Port bits [7:0]							
7	6	5	4	3	2	1	0

Bits 31 to 0: **JPEG Line Buffer Read/Write Port**

This is used for write port to the JPEG line buffer during host JPEG encode and for read port from the JPEG Line Buffer during JPEG decode.

8.4.6 JPEG Codec Registers (JCODEC)

Operation Mode Settings Register
JCODEC[0x00] Default = 0x0000 Read/Write

n/a							
15	14	13	12	11	10	9	8
n/a			Reserved (0)	Marker Insert Enable	JPEG Operation Mode	YUV Format Select bits [1:0]	
7	6	5	4	3	2	1	0

Bit 4 (RSV): **Reserved (0)**
Always set to “0.”

Bit 3: **Marker Insertion Enable**
0: Disable (default)
1: Enable
Setting this bit to “1” during JPEG encode enables insertion into the JPEG file of the data in the Insert Marker Data Register.
JPEG decode does not use this bit.

Bit 2: **JPEG Operation Mode Select**
This specifies the JPEG operation mode.

Table 8.11 JPEG Operation Selection

Bit 2	JPEG Operation
0 (default)	Encode
1	Decode

Bits 1 to 0: **YUV Format Select [1:0]**
This specifies the YUV data format for JPEG encode. Note that JPEG encode from the YUV 4:4:4 format requires first scaling the image with the capture resizer.
During JPEG decode, this gives the YUV data format for the JPEG file.

Table 8.12 YUV Format Selection

Bits [1:0]	YUV Format
00 (default)	4:4:4
01	4:2:2
10	4:2:0
11	4:1:1

Command Setting Register
JCODEC[0x04] Default = not applicable Write Only

n/a							
15	14	13	12	11	10	9	8
JPEG Codec SW Reset	n/a						JPEG Operation Start
7	6	5	4	3	2	1	0

Do not read this register. Do not write to it during JPEG operation—unless it is for a reset.

Bit 7: **JPEG Codec Software Reset**
0: (ignored)
1: Reset
Writing “1” to this bit produces a JPEG Codec Software Reset, but does not reset JPEG Codec Registers.

Bit 0: **JPEG Operation Start**

0: (ignored)
1: Start JPEG operation

Writing “1” to this bit starts JPEG encode/decode and YUV data capture.

JPEG Operation Status Register JCODEC[0x08] Default = 0x0000 Read Only							
n/a							
15	14	13	12	11	10	9	8
n/a							JPEG Operation Status (RO)
7	6	5	4	3	2	1	0

Reading this register clears bit 1 in both the JPEG Status Flag Register and the JPEG Raw Status Flag Register.

Bit 0: **JPEG Operation Status**

This gives the JPEG codec operation status.

0: Idle
1: Busy with JPEG encode/decode

Quantization Table Number Register JCODEC[0x0C] Default = 0x0000 Read/Write							
n/a							
15	14	13	12	11	10	9	8
n/a					V Table Select	U Table Select	Y Table Select
7	6	5	4	3	2	1	0

Bit 2: **V Component Table Select**

This bit selects the Quantization Tables Number of V component for JPEG encode process. This bit is not used for JPEG decode process.

0: The Quantization Table No. 0 is used. (default)
1: The Quantization Table No. 1 is used.

Bit 1: **U Component Table Select**

This bit selects the Quantization Tables Number of U component for JPEG encode process. This bit is not used for JPEG decode process.

0: The Quantization Table No. 0 is used. (default)
1: The Quantization Table No. 1 is used.

Bit 0: **Y Component Table Select**

This bit selects the Quantization Tables Number of Y component for JPEG encode process. This bit is not used for JPEG decode process.

0: The Quantization Table No. 0 is used. (default)
1: The Quantization Table No. 1 is used.

Huffman Table Number Register JCODEC[0x10] Default = 0x0000 Read/Write							
n/a							
15	14	13	12	11	10	9	8
n/a		V ACTable Select	V DCTable Select	U ACTable Select	U DCTable Select	Y ACTable Select	Y DCTable Select
7	6	5	4	3	2	1	0

Bit 5: **V Component AC Table Select**

This bit selects the AC Huffman Table Number of V component for JPEG encode process. Select “1” in JPEG fast encode mode. This bit is not used for JPEG decode process.

0: The AC Huffman Table No. 0 is used (default)
1: The AC Huffman Table No. 1 is used

Bit 4: **V Component DC Table Select**

This bit selects the DC Huffman Table Number of V component for JPEG encode process. Select “1” in JPEG fast encode mode. This bit is not used for JPEG decode process.

0: The DC Huffman Table No. 0 is used (default)
1: The DC Huffman Table No. 1 is used

Bit 3: **U Component AC Table Select**

This bit selects the AC Huffman Table Number of U component for JPEG encode process. Select “1” in JPEG fast encode mode. This bit is not used for JPEG decode process.

0: The AC Huffman Table No. 0 is used (default)
1: The AC Huffman Table No. 1 is used

Bit 2: **U Component DC Table Select**

This bit selects the DC Huffman Table Number of U component for JPEG encode process. Select “1” in JPEG fast encode mode. This bit is not used for JPEG decode process.

0: The DC Huffman Table No. 0 is used (default)
1: The DC Huffman Table No. 1 is used

Bit 1: **Y Component AC Table Select**

This bit selects the AC Huffman Table Number of Y component for JPEG encode process. Select “0” in JPEG fast encode mode. This bit is not used for JPEG decode process.

0: The AC Huffman Table No. 0 is used (default)
1: The AC Huffman Table No. 1 is used

Bit 0: **Y Component DC Table Select**

This bit selects the DC Huffman Table Number of Y component for JPEG encode process. Select “0” in JPEG fast encode mode. This bit is not used for JPEG decode process.

0: The DC Huffman Table No. 0 is used (default)
1: The DC Huffman Table No. 1 is used

DRI Setting Register 0							
JCODEC[0x14]	Default = 0x0000						Read/Write
n/a							
15	14	13	12	11	10	9	8
DRI Value bits [15:8]							
7	6	5	4	3	2	1	0

DRI Setting Register 1							
JCODEC[0x18]	Default = 0x0000						Read/Write
n/a							
15	14	13	12	11	10	9	8
DRI Value bits [7:0]							
7	6	5	4	3	2	1	0

DRI Setting Register 0 - bits [7:0]
DRI Setting Register 1 - bits [7:0]

DRI Value bits[15:0]

Together these two registers specify the RST interval, the number of MCU blocks between RST markers, for JPEG encode.

Setting both to zero skips RST marker insertion, but not the RST Interval Definition Marker specifying zero as the interval.

JPEG decode does not use this field.

Vertical Pixel Size Register 0							
JCODEC[0x1C]	Default = 0x0000						Read/Write
n/a							
15	14	13	12	11	10	9	8
Y Pixel Size bits [15:8]							
7	6	5	4	3	2	1	0

Vertical Pixel Size Register 1							
JCODEC[0x20]	Default = 0x0000						Read/Write
n/a							
15	14	13	12	11	10	9	8
Y Pixel Size bits [7:0]							
7	6	5	4	3	2	1	0

Vertical Pixel Size Register 0 - bits [7:0]
Vertical Pixel Size Register 1 - bits [7:0]

Y Pixel Size bits[15:0]

This pair of registers specifies, in pixels, the vertical image size for JPEG encode and YUV data capture. During JPEG decode, this pair gives the JPEG file vertical image size in pixels.

Note: These registers are write only for YUV data capture (“x11” in JCTL[0x00] bits 3 to 1). Reads return undefined values.

Horizontal Pixel Size Register 0							
JCODEC[0x24]	Default = 0x0000						Read/Write
n/a							
15	14	13	12	11	10	9	8
X Pixel Size bits [15:8]							
7	6	5	4	3	2	1	0

Horizontal Pixel Size Register 1							
JCODEC[0x28]	Default = 0x0000						Read/Write
n/a							
15	14	13	12	11	10	9	8
X Pixel Size bits [7:0]							
7	6	5	4	3	2	1	0

Horizontal Pixel Size Register 0 - bits [7:0]
Horizontal Pixel Size Register 1 - bits [7:0]

X Pixel Size bits[15:0]

This pair of registers specifies, in pixels, the horizontal image size for JPEG encode and YUV data capture. During JPEG decode, this pair gives the JPEG file horizontal image size in pixels.

Note: These registers are write only for YUV data capture (“x11” in JCTL[0x00] bits 3 to 1). Reads return undefined values.

Reserved Registers							
JCODEC[0x2C-34]							
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0

RST Marker Operation Setting Register							
JCODEC[0x38]	Default = 0x0000						Read/Write
n/a							
15	14	13	12	11	10	9	8
n/a						RST Marker Operation bits [1:0]	
7	6	5	4	3	2	1	0

Bits 1 to 0: **RST Marker Operation Select**

This controls RST marker operation during JPEG decode.

JPEG encode does not use this setting.

Table 8.13 RST Marker Operation Modes

Bits [1:0]	RST Marker Operation
00 (default)	Error detection and data correction off Use this setting only in situations where the JPEG file for decode is known to be normal and free of errors. Otherwise, any errors in the JPEG file go undetected, yielding unreliable results.
01	Error detection on An error detected during decode immediately terminates JPEG decoding with a JPEG codec end interrupt request. RST Marker Operation Status (JCODEC[0x3C]) Register bits 6 to 3 (JPEG Decode Error Status) then give the error type. If there was an error, a JPEG codec core software reset is necessary before starting the next decoding operation.
10	Data correction on The JPEG codec responds to errors detected during decode by automatically skipping or adding data, allowing decode to continue through to the end of the JPEG file. Only then is there a JPEG codec end interrupt request.
11	Reserved

RST Marker Operation Status Register

JCODEC[0x3C] Default = 0x0000 Read Only

n/a							
15	14	13	12	11	10	9	8
Correction Code	JPEG Decode Error Status bits [3:0]				n/a		
7	6	5	4	3	2	1	0

Bit 7: **Correction Code**

0: No

1: Yes

"1" in this bit indicates that correction operation has been used during JPEG decode.

The contents are only valid, however, when bits 1 to 0 in the RST Marker Operation Setting Register are "10," enabling data correction for RST marker processing.

JPEG encode does not use this setting.

Bits 6 to 3: **JPEG Decode Error Status**

This gives an error code for JPEG decode.

The contents are only valid, however, when bits 1 to 0 in the RST Marker Operation Setting Register are "01," enabling error detection for RST marker processing.

JPEG encode does not use this setting.

Table 8.14 JPEG Error Codes

Bits [6:3]	JPEG Error Status
0000	No errors
0001 to 1010	Reserved
1011	Restart interval error
1100	Image size error
1101 to 1111	Reserved

Insert Marker Data Registers							
JCODEC[0x40-0xCC]	Default = 0x00FF						Read/Write
n/a							
15	14	13	12	11	10	9	8
Insert marker Data bits [7:0]							
7	6	5	4	3	2	1	0

These registers hold the marker data inserted during JPEG encode. Note that all 36 bytes are inserted regardless of the marker length.

JPEG decode does not use these registers.

Address Offsets 40h to 44h: Marker code for the sequence

Address Offsets 48h to 4Ch: Marker length (0002h to 0022h)

Address Offsets 50h to CCh: Marker data If the marker length is less than 32, fill the leftover registers with FFh.

Quantization Table No. 0 Register							
JCODEC[0x400-0x4FC]	Default = not applicable						Read/Write
n/a							
15	14	13	12	11	10	9	8
Quantization Table No. 0 bits [7:0]							
7	6	5	4	3	2	1	0

Bits 7 to 0: **Quantization Table No. 0**

This register specifies the value for quantization table No. 0 for JPEG encode.

JPEG decode does not use these registers.

Quantization Table No. 1 Register							
JCODEC[0x500-0x5FC]	Default = not applicable						Read/Write
n/a							
15	14	13	12	11	10	9	8
Quantization Table No. 1 Register 0 bits [7:0]							
7	6	5	4	3	2	1	0

Bits 7 to 0: **Quantization Table No. 1**

This register specifies the value for quantization table No. 1 for JPEG encode.

JPEG decode does not use this register.

DC Huffman Table No. 0 Register 0							
JCODEC[0x800-0x83C]	Default = not applicable						Write Only
n/a							
15	14	13	12	11	10	9	8
DC Huffman Table No. 0 Register 0 bits [7:0]							
7	6	5	4	3	2	1	0

DC Huffman Table No. 0

This register specifies the DC component Huffman table values for table 0 used during JPEG encode. This register specifies the number of codes for each code length. JPEG decode and Huffman table automatic loading do not use this register.

DC Huffman Table No. 0 Register 1							
JCODEC[0x840-0x86C]	Default = not applicable						Write Only
n/a							
15	14	13	12	11	10	9	8
Reserved (must be all 0)				DC Huffman Table No. 0 Register 1 bits [3:0]			
7	6	5	4	3	2	1	0

DC Huffman Table No. 0

This register specifies the DC component Huffman table values for table 0 used during JPEG encode. This register specifies the group numbers in order of the frequency of occurrence. Only the lower 4 bits of this register are used; the upper 4 bits must be set to “0000.” JPEG decode and Huffman table automatic loading do not use this register.

AC Huffman Table No. 0 Register 0							
JCODEC[0x880-0x8BC]	Default = not applicable						Write Only
n/a							
15	14	13	12	11	10	9	8
AC Huffman Table No. 0 Register 0 bits [7:0]							
7	6	5	4	3	2	1	0

AC Huffman Table No. 0

This register specifies the AC component Huffman table values for table 0 used during JPEG encode. This register specifies the number of codes for each code length. JPEG decode and Huffman table automatic loading do not use this register.

AC Huffman Table No. 0 Register 1							
JCODEC[0x8C0-0xB44]	Default = not applicable						Write Only
n/a							
15	14	13	12	11	10	9	8
AC Huffman Table No. 0 Register 1 bits [7:0]							
7	6	5	4	3	2	1	0

AC Huffman Table No. 0

This register specifies the AC component Huffman table values for table 0 used during JPEG encode. This register specifies the runs of zeros and group numbers in order of the frequency of occurrence. JPEG decode and Huffman table automatic loading do not use this register.

DC Huffman Table No. 1 Register 0							
JCODEC[0xC00-0xC3C]	Default = not applicable						Write Only
n/a							
15	14	13	12	11	10	9	8
DC Huffman Table No.1 Register 0 bits [7:0]							
7	6	5	4	3	2	1	0

DC Huffman Table No. 1

This register specifies the DC component Huffman table values for table 1 used during JPEG encode. This register specifies the number of codes for each code length. JPEG decode and Huffman table automatic loading do not use this register.

DC Huffman Table No. 1 Register 1							
JCODEC[0xC40-0xC6C]	Default = not applicable						Write Only
n/a							
15	14	13	12	11	10	9	8
Reserved (must be all 0)				DC Huffman Table No. 1 Register 1 bits [3:0]			
7	6	5	4	3	2	1	0

DC Huffman Table No. 1

This register specifies the DC component Huffman table values for table 1 used during JPEG encode. This register specifies the group numbers in order of the frequency of occurrence. Only the lower 4 bits of this register are used; the upper 4 bits must be set to “0000.” JPEG decode and Huffman table automatic loading do not use this register.

AC Huffman Table No. 1 Register 0							
JCODEC[0xC80-0xCBC]	Default = not applicable						Write Only
n/a							
15	14	13	12	11	10	9	8
AC Huffman Table No. 1 Register 0 bits [7:0]							
7	6	5	4	3	2	1	0

AC Huffman Table No. 1

This register specifies the AC component Huffman table values for table 1 used during JPEG encode. This register specifies the number of codes for each code length. JPEG decode and Huffman table automatic loading do not use this register.

AC Huffman Table No. 1 Register 1							
JCODEC[0xCC0-0xF44]	Default = not applicable						Write Only
n/a							
15	14	13	12	11	10	9	8
AC Huffman Table No. 1 Register 1 bits [7:0]							
7	6	5	4	3	2	1	0

AC Huffman Table No. 1

This register specifies the AC component Huffman table values for table 1 used during JPEG encode. This register specifies the runs of zeros and group numbers in order of the frequency of occurrence. JPEG decode and Huffman table automatic loading do not use this register.

8.5 Description of Operation

8.5.1 Capture Control

State machines control data capture. This is because camera image JPEG encode, YUV data capture, and other operations require data capture in frames, and such control in hardware helps to relieve time limitations.

These state machines use different states and state transition conditions for camera image JPEG encode and YUV data capture.

A register provides real-time access to the current state machine state (ID). The contents changes frequently, however, so such reads should be limited to software debugging purposes.

8.5.1.1 State Machine for Camera Image JPEG Encode

The following is the capture control state machine for camera image JPEG encode.

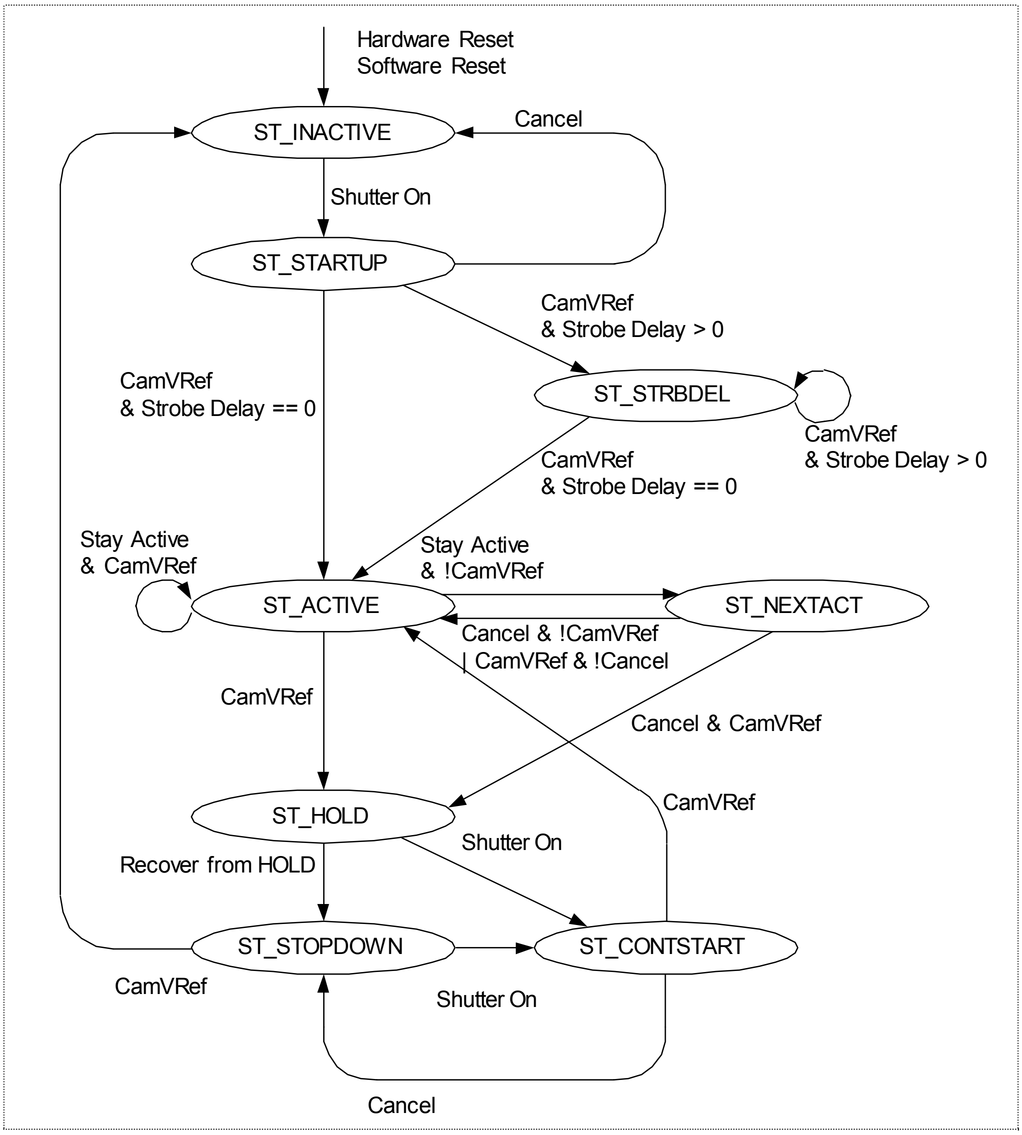


Fig.8.2 State Machine for Camera Image JPEG Encode

The following Table describes the states. A register provides read access to the state ID.

Table 8.15 State Descriptions in Camera Image JPEG Encode Mode

State (ID)	Description
ST_INACTIVE (0x0)	This is the initial state. This state captures no camera data.
ST_STARTUP (0x1)	Pressing the shutter shifts to this state from the ST_INACTIVE state. This state captures no camera data.
ST_STRBDEL (0x3)	This state represents the strobe frame delay. The state machine waits in this state for the specified number of frames to delay capture before shifting to the ST_ACTIVE state. This state captures no camera data.
ST_ACTIVE (0x7)	This state captures camera data. The only transitions out of this state involve the VREF input indicating the start of valid camera image data, so, as long as the cropping and scaling settings are correct, reliable frame data capture is guaranteed.
ST_NEXTACT (0xB)	In the ST_ACTIVE state, setting the Frame Capture Request bit to “1” shifts to this state. The next VREF input indicating the start of valid camera image data shifts back to the ST_ACTIVE state, enabling consecutive JPEG encode of frames. This state captures camera data.
ST_HOLD (0x6)	This state starts from the frame following the completion of capture. This state captures no camera data.
ST_STOPDOWN (0x4)	A hold cancel signal shifts to this state from the ST_HOLD state. The next frame shifts to the ST_INACTIVE state, terminating the hold state. This state captures no camera data.
ST_CONTSTART (0x5)	Pressing the shutter while the state machine is in the hold state resumes capture from the next frame. This state captures no camera data.

The following Table summarizes events and their actions.

Table 8.16 Event Description in Camera Image JPEG Encode Mode

Event	Action
Software Reset	This produces a software reset of the capture resizer.
Shutter On	This event corresponds to writing “1” to the JPEG start/stop control bit.
Cancel, Recover from HOLD	These events correspond to writing “0” to the JPEG start/stop Control bit.
CamVRef	This represents a change in the camera image data timing signal CMVREF from data valid level to data invalid level. This Section sometimes abbreviates this timing to simply VREF.
Stay Active	This writes “1” to the next frame capture request bit in the JPEG Control Register.
Strobe Delay	This decrements the strobe frame delay counter.

The following is a timing chart for JPEG encode of a single camera image.

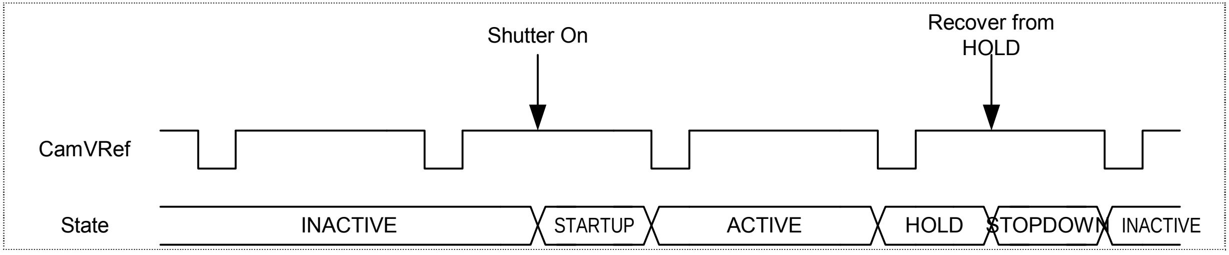


Fig.8.3 Timing Chart for Camera Image JPEG Encode (Single)

8.5.1.2 State Machine for YUV Data Capture

The following is the capture control state machine for YUV data capture.

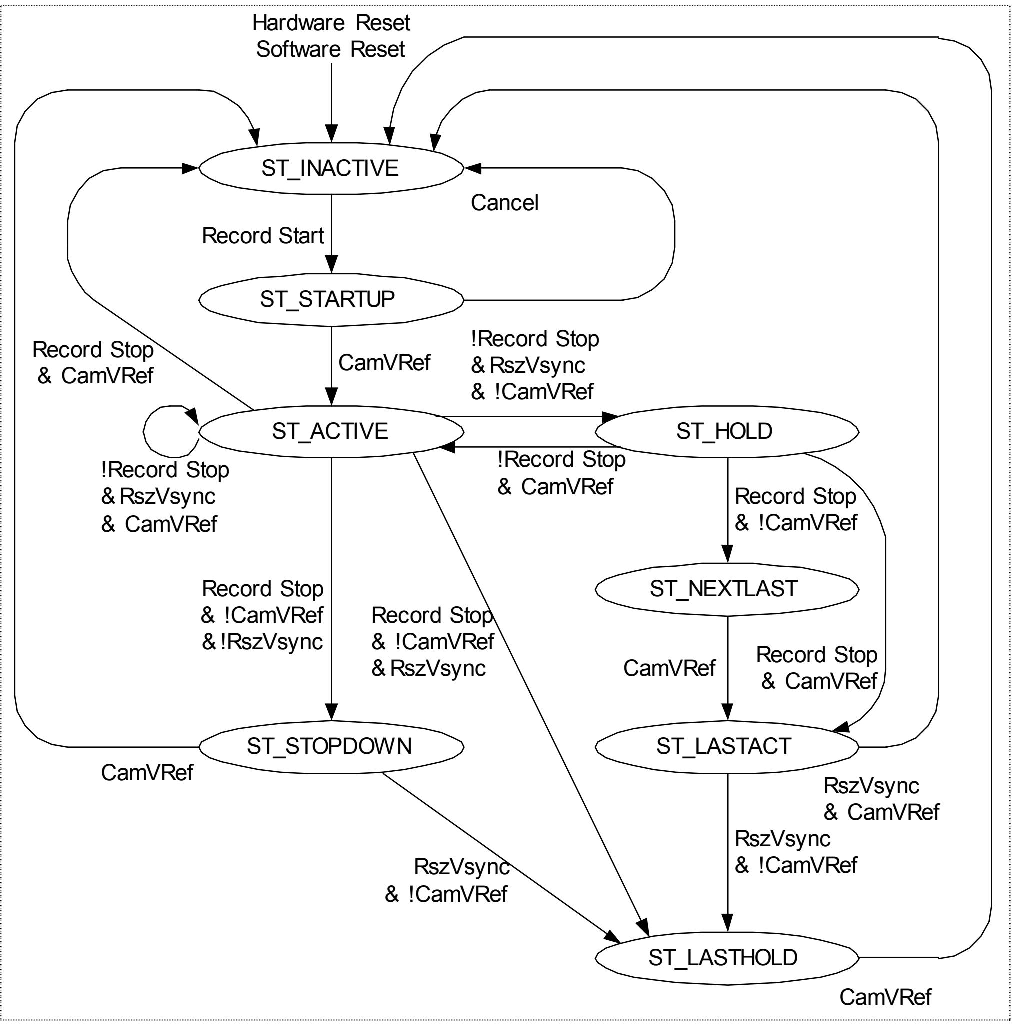


Fig.8.4 State Machine for YUV Data Capture

Table 8.17 State Descriptions in YUV Data Capture

State (ID)	Description
ST_INACTIVE (0x0)	This is the initial state. This state captures no camera data.
ST_STARTUP (0x1)	Active image recording start signal shifts to this state from the ST_INACTIVE state. This state captures no camera data.
ST_ACTIVE (0x7)	This state captures camera data.
ST_HOLD (0x6)	Capture data VREF in the middle of a frame shifts to this state from the ST_ACTIVE state. This state captures no camera data.
ST_STOPDOWN (0x4)	Active image recording stop signal shifts to this state from the ST_ACTIVE state. Capture ends with the current frame. This state captures camera data.
ST_NEXTLAST (0xE)	Active image recording stop signal shifts to this state from the ST_HOLD state. This state is necessary because JPEG module limitations prevent it from recognizing the end of movie capture between the image capture data VREF and the camera VREF. The JPEG module must therefore capture an additional frame after the active image recording stop signal following this VREF. This state captures no camera data.
ST_LASTACT (0xF)	This state is for capturing an additional frame to compensate for the limitation described above. This state captures camera data.
ST_LASTHOLD (0xD)	Capture data VREF during the final frame of the recording shifts to this state. This state captures no camera data.

The following Table summarizes events and their actions.

Table 8.18 Event Descriptions in YUV Data Capture

Event	Action
Software Reset	This produces a software reset of the capture resizer.
Record Start	This event corresponds to writing “1” to the JPEG Start/Stop Control bit.
Cancel, Record Stop	These events correspond to writing “0” to the JPEG Start/Stop Control bit.
CamVRef	This represents a change in the camera image data timing signal CMVREF from data valid level to data invalid level. This Section sometimes abbreviates this timing to simply VREF.
RszVsync	This represents the frame end signal from the capture resizer.

8.5.2 Capture Resizer

The capture resizer processes the image data from the camera interface in two stages: trimming and scaling.

The camera interface converts the YUV 4:2:2 image data from the camera module into YUV 4:4:4 format, allowing trimming at the individual pixel level.

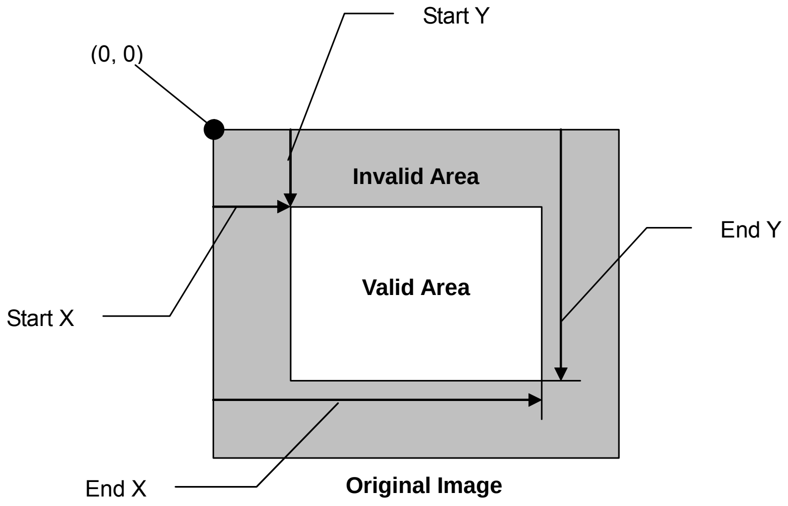
8.5.2.1 Trimming

Trimming in preparation for scaling discards border areas to minimize the image size using start and end (X,Y) position specified in registers. The origin (0,0) for these coordinates, in pixels, is at the upper left corner of the source image. The start coordinates specify the upper left corner of the image to retain; the end coordinates, the lower right.

Specifying a start or end point outside the boundaries of the input camera image produces invalid results.

Specifying an end position above or to the left of the start one produces invalid output, so always make the end coordinates greater than the corresponding starting ones.

To forcibly stop data output from the capture resizer, write “0” to RSZ[0xC0] bit 0 to disable the capture resizer.



Capture Resizer
Start X = RSZ[0xC8h] bits[10:0]
Start Y = RSZ[0xCCh] bits[10:0]
End X = RSZ[0xD0h] bits[10:0]
End Y = RSZ[0xD4h] bits[10:0]

Fig.8.5 Trimming Function

8.5.2.2 Scaling

Scaling shrinks the trimmed image with the specified square size. The image size after scaling must match that specified for JPEG encode.

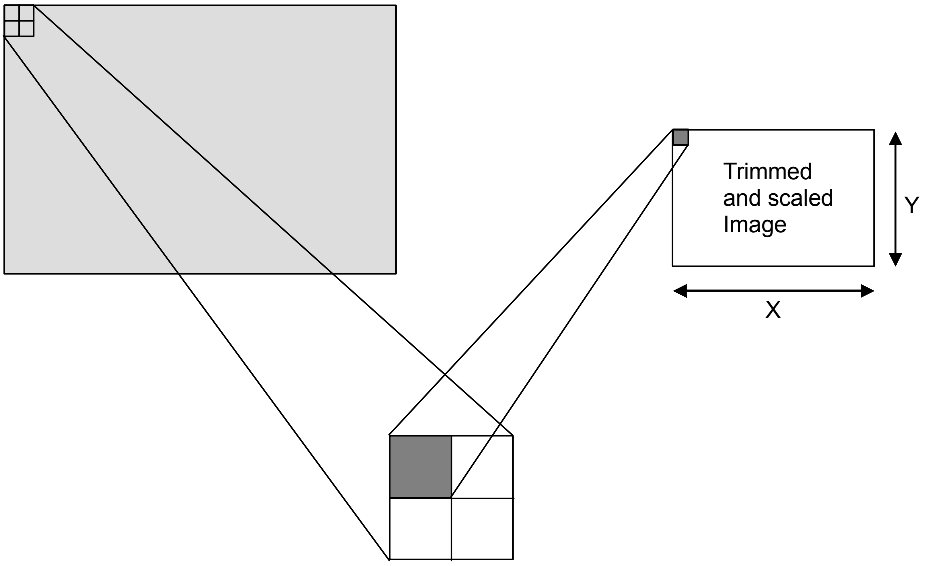
There are three scaling modes: sampling (1), averaging (2), and no scaling (0).

The first replaces squares 2 to 8* pixels on a side with single pixels from the centers of those squares.

The second averages a single line of pixels from the centers of those squares. There are three square sizes available, all powers of two*: 2, 4, or 8.

The last specifies no scaling, disabling the Capture Resizer Scaling Rate Register.

* Do not attempt to skip scaling by simply Setting the Scaling Rate Register to "1". Doing so produces invalid results for all modes other than no scaling (0).

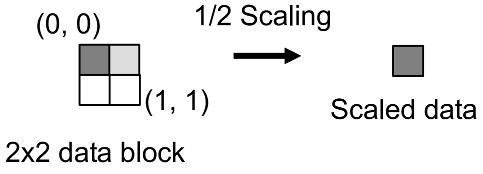


Capture Resizer
Scaling Rate = RSZ[0xD8h] bits[3:0]
Result X = JCODEC[0x24h], JCODEC[0x28h]
Result Y = JCODEC[0x1Ch], JCODEC[0x20h]

Fig.8.6 Scaling Example (1/2 Scaling)

8.5.2.2.1 1/2 Scaling

The 1/2 scaling function scales each 2x2 pixel block to a single pixel. This function provides two modes for this scaling ratio: reduction mode and averaging mode. Note that reduction mode is always used in the vertical direction regardless of which mode is selected.

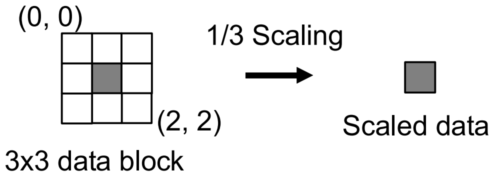


Scaled data = {(0, 0) + (0, 1)}/2 (average)
= (0, 0) (reduction)

Fig.8.7 1/2 Scaling

8.5.2.2.2 1/3 Scaling

The 1/3 scaling function scales each 3x3 pixel block to a single pixel. This scaling function only operates in reduction mode.

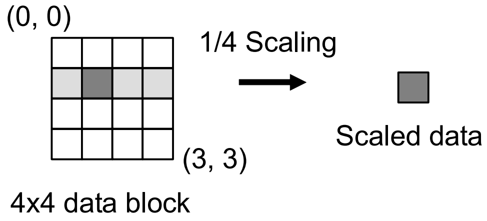


Scaled data = (1, 1)

Fig.8.8 1/3 Scaling

8.5.2.2.3 1/4 Scaling

The 1/4 scaling function scales each 4x4 pixel block to a single pixel. This function provides two modes for this scaling ratio: reduction mode and averaging mode. Note that reduction mode is always used in the vertical direction regardless of which mode is selected.

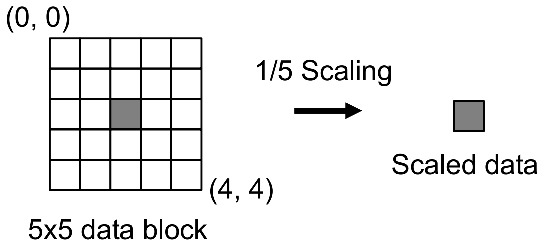


Scaled data = {(0, 1) + (1, 1) + (2, 1) + (3, 1)}/4 (average)
= (1, 1) (reduction)

Fig.8.9 1/4 Scaling

8.5.2.2.4 1/5 Scaling

The 1/5 scaling function scales each 5x5 pixel block to a single pixel. This scaling function only operates in reduction mode.

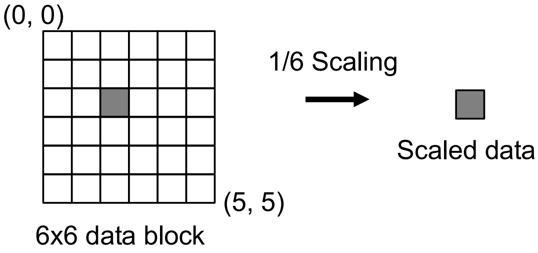


Scaled data = (2, 2)

Fig.8.10 1/5 Scaling

8.5.2.2.5 1/6 Scaling

The 1/6 scaling function scales each 6x6 pixel block to a single pixel. This scaling function only operates in reduction mode.

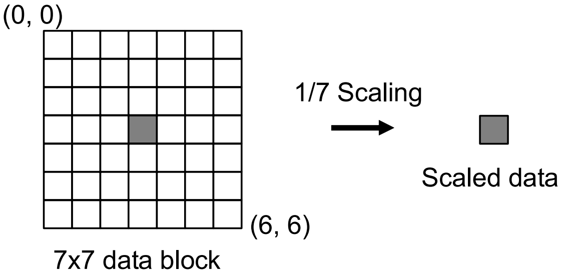


Scaled data = (2, 2)

Fig.8.11 1/6 Scaling

8.5.2.2.6 1/7 Scaling

The 1/7 scaling function scales each 7x7 pixel block to a single pixel. This scaling function only operates in reduction mode.

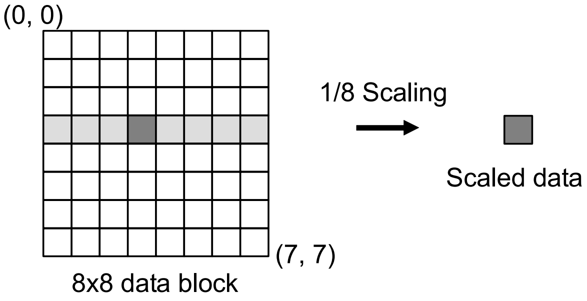


Scaled data = (3, 3)

Fig.8.12 1/7 Scaling

8.5.2.2.7 1/8 Scaling

The 1/8 scaling function scales each 8x8 pixel block to a single pixel. This function provides two modes for this scaling ratio: reduction mode and averaging mode. Note that reduction mode is always used in the vertical direction regardless of which mode is selected.



Scaled data = {(0, 3)+(1, 3)+(2, 3)+(3, 3)+(4, 3)+(5, 3)+(6, 3)+(7, 3)}/8 (average)
= (3, 3) (reduction)

Fig.8.13 1/8 Scaling

8.5.2.3 Usage Restrictions

Changing register contents while the capture resizer is receiving data from the camera interface invalidates that data. This is not an issue for software resets and enable operation. The general approach for changing capture resizer register settings while the camera is operational, therefore, is to monitor the VREF interrupt requests from the camera interface and change settings only during the VREF data blanking intervals between frames, when the camera interface does not accept data from the image sensor.

Specifying a start or end point outside the boundaries of the input camera image produces invalid results.

The settings in the Start and End Position Registers must specify a trimmed image with a vertical and horizontal pixels that are both multiples of the value specified in the Scaling Rate Register.

8.5.3 Image Processing Data Flow

This Section illustrates data flow for the various image processing modes.

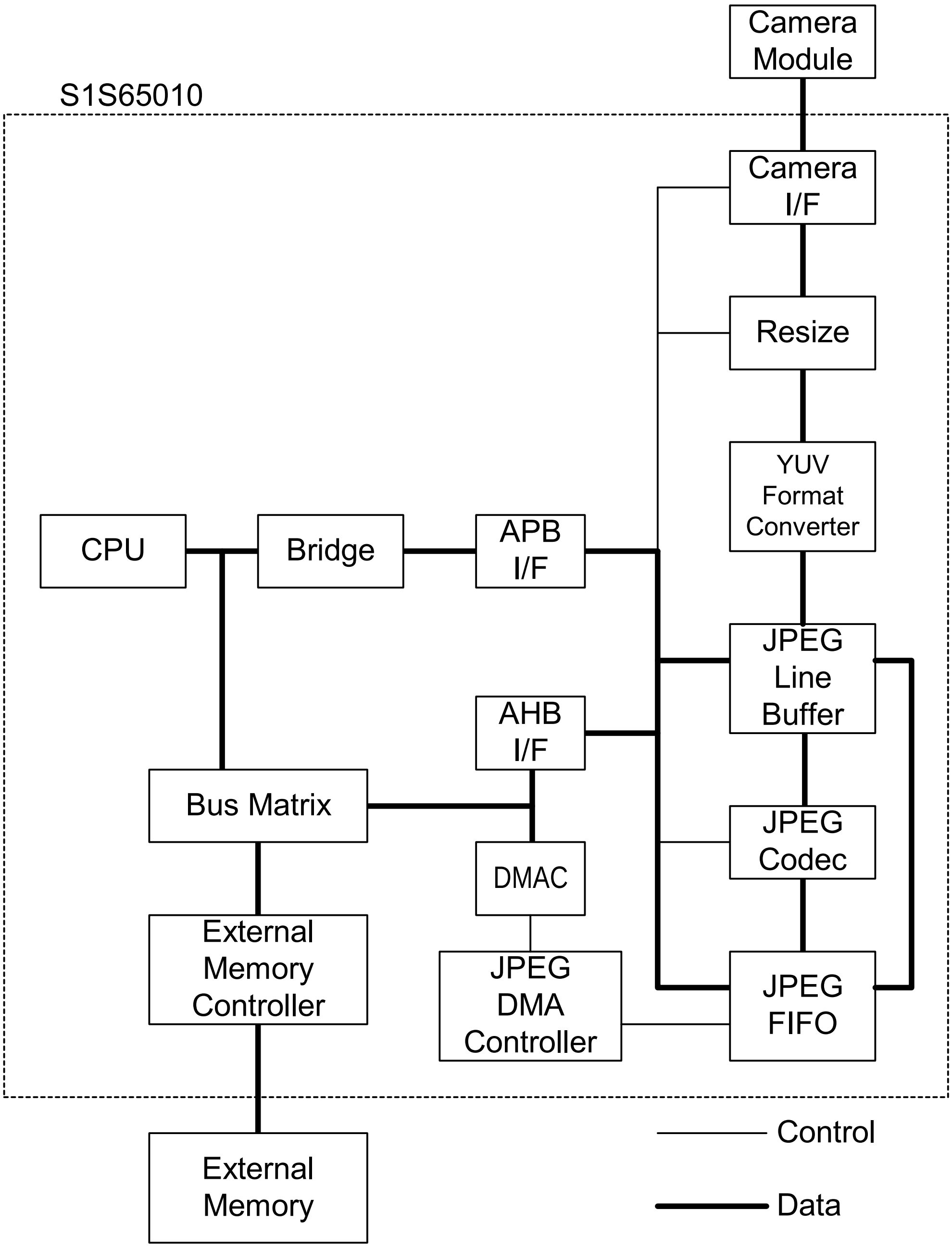


Fig.8.14 Image Processing Data Flow

8.5.3.1 Camera Image JPEG Encode

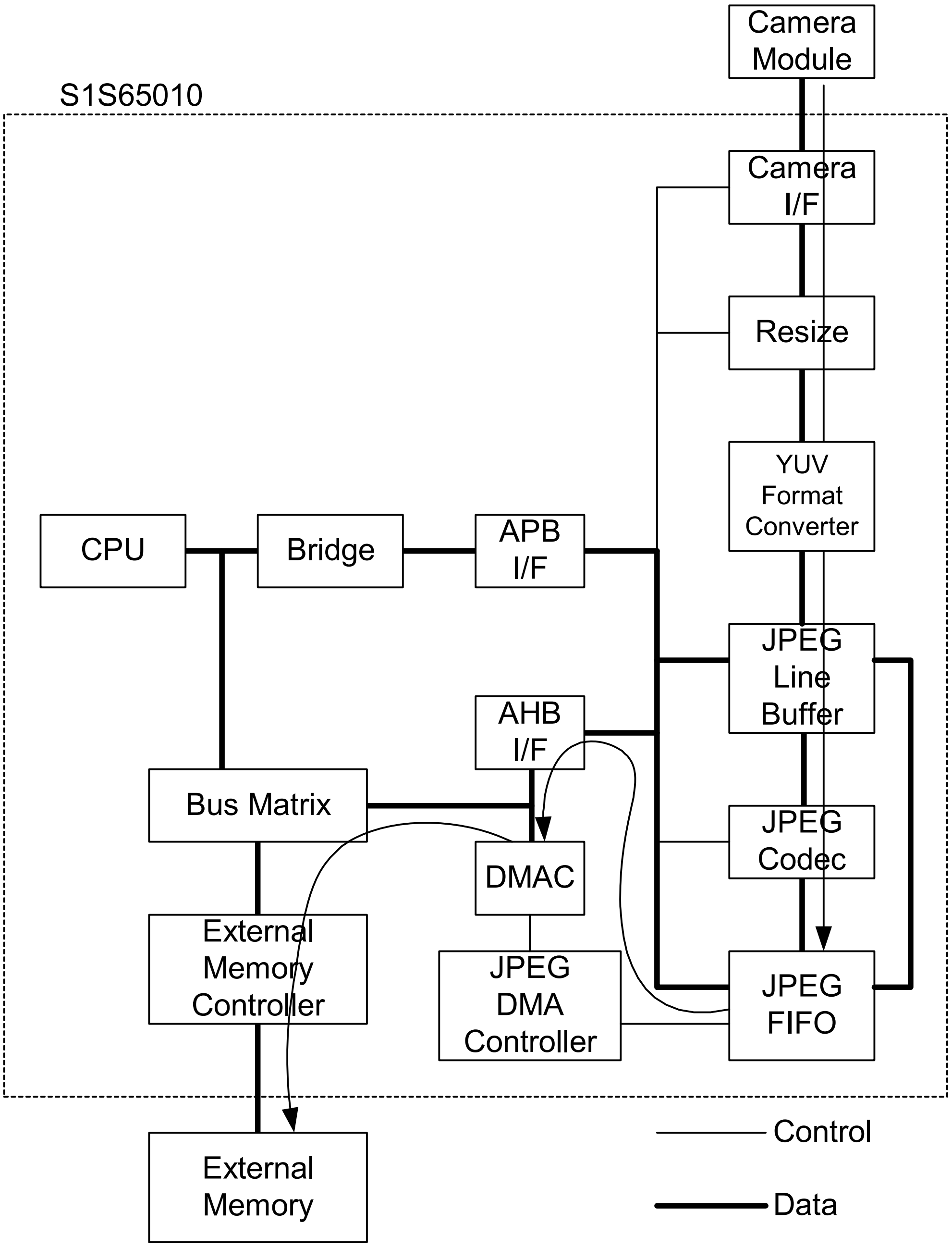


Fig.8.15 Camera Image JPEG Encode Data Flow

8.5.3.2 YUV Data Capture

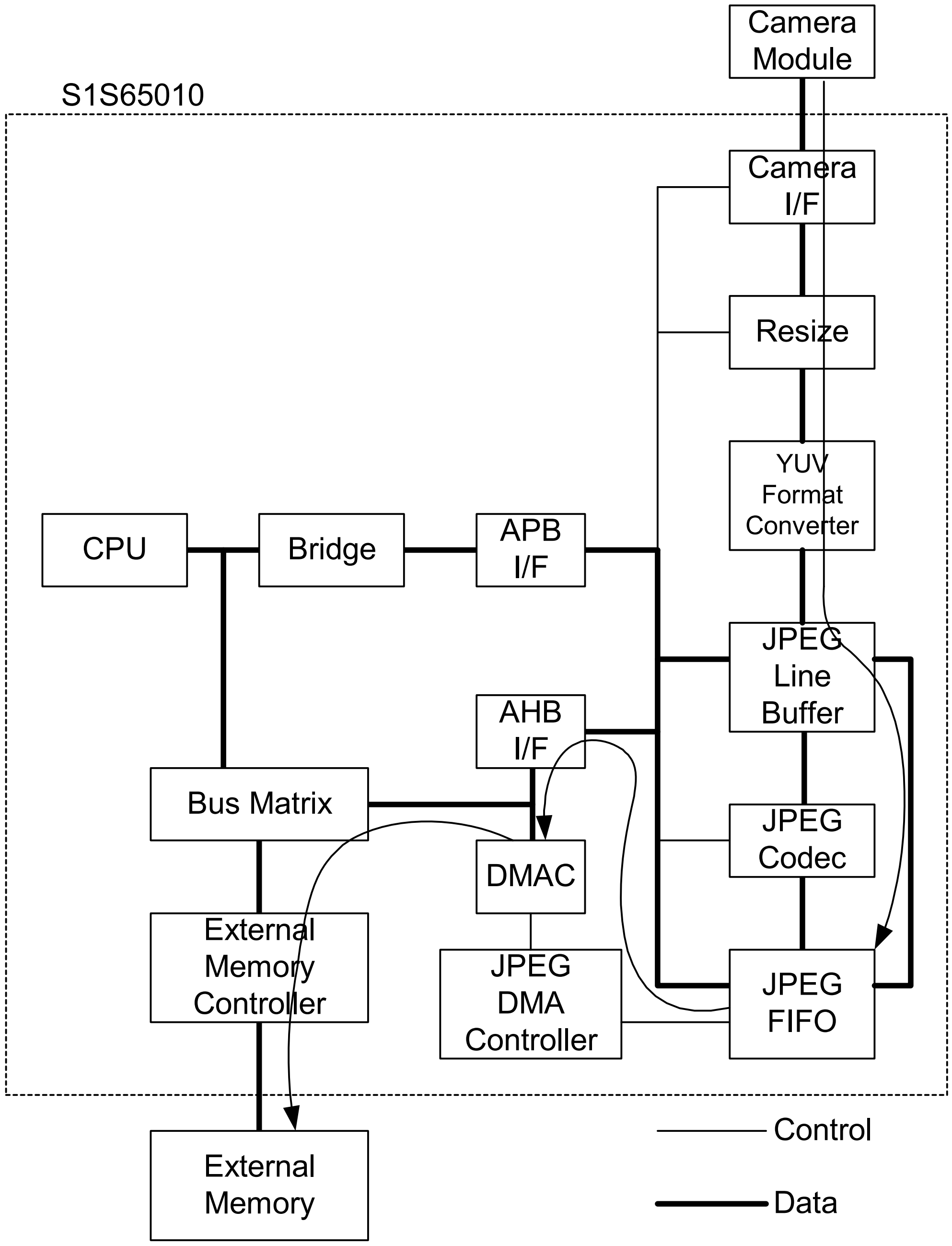


Fig.8.16 YUV Data Capture Data Flow

8.5.3.3 YUV Data JPEG Encode

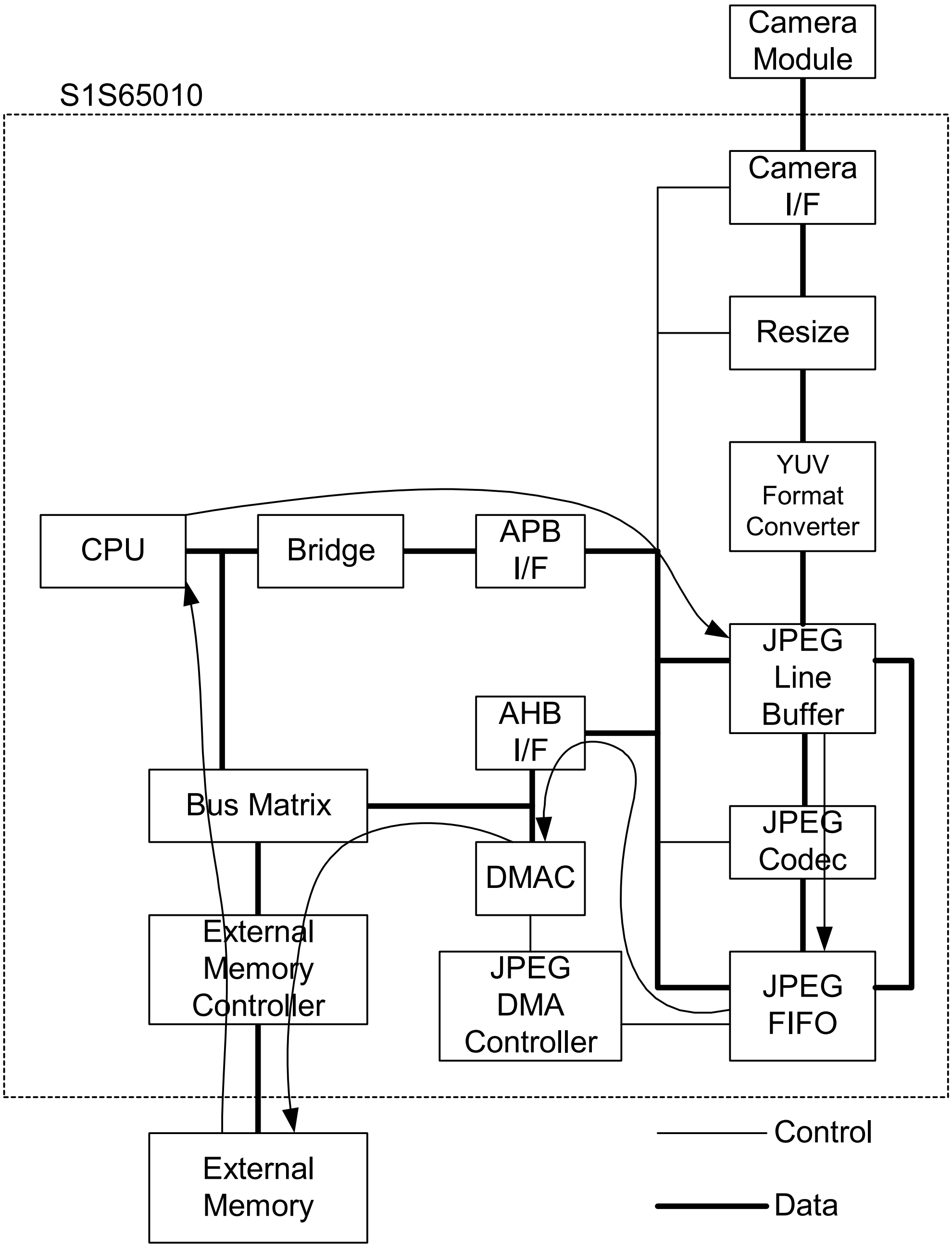


Fig.8.17 YUV Data JPEG Encode Data Flow

8.5.3.4 YUV Data JPEG Decode

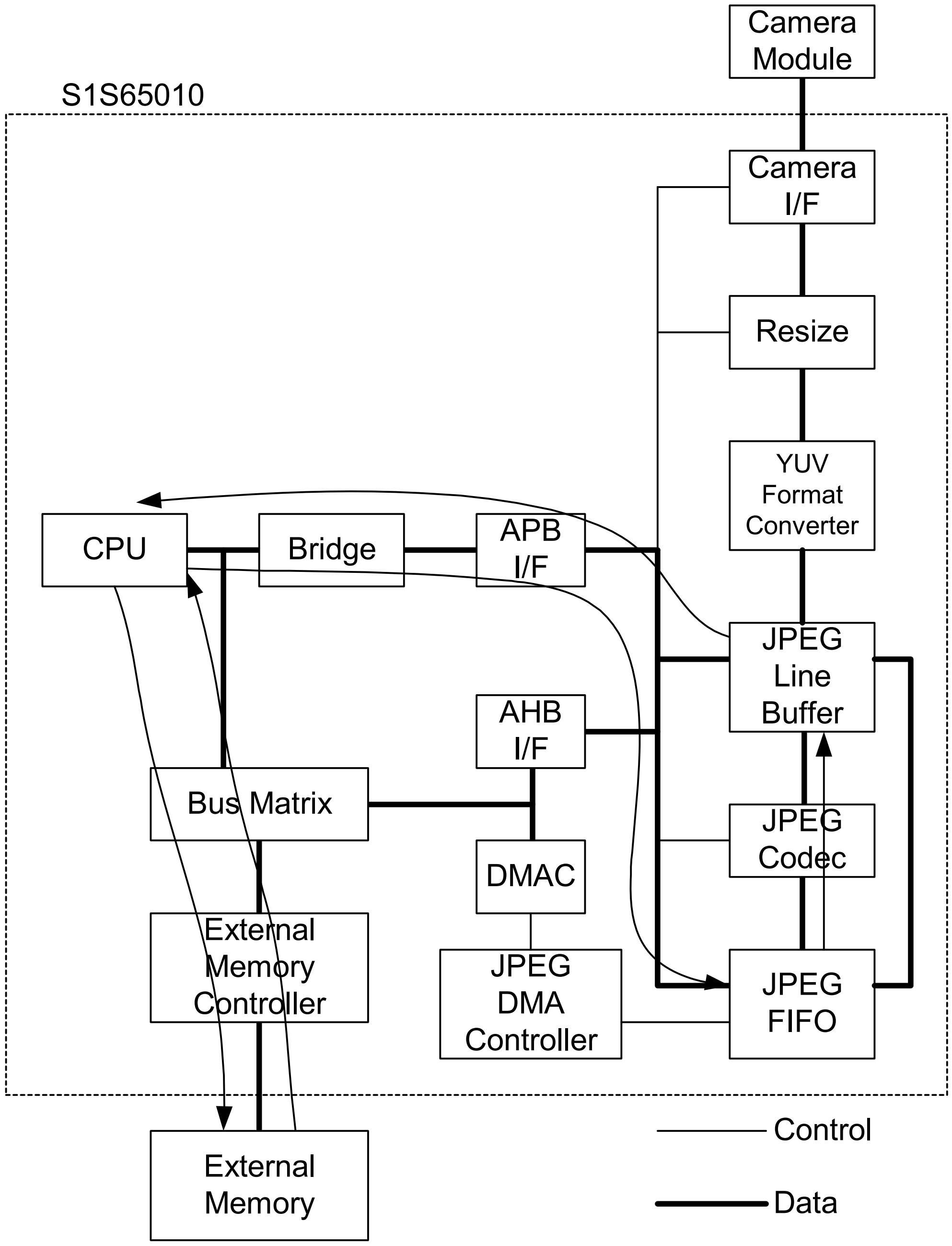


Fig.8.18 YUV Data JPEG Decode Data Flow

8.5.4 JPEG Codec Functions

The JPEG codec generally supports JPEG baseline methods and satisfies all compatibility requirements specified in the JPEG standard Part 2 (ISO/IEC 10918-2).

The JPEG codec supports images up to 1600 pixels horizontal wide and 2048 pixels vertical wide. It supports JPEG encode for image sizes up to UXGA (1600 × 1200).

The YUV data format converter converts the YUV 4:4:4 camera image data from the capture resizer to the YUV data format specified in JCODEC[0x00] bits 1 to 0. Note, however, that camera image JPEG encode, YUV data JPEG encode, JPEG decode, and YUV data capture impose the following minimum resolutions on image size. If the image is not a multiple of this minimum resolution, the software must modify the JPEG file's size information before decoding a JPEG file.

Table 8.19 Minimum Resolution Restrictions

YUV Format	Minimum Resolution
4:4:4	1 × 1
4:2:2	2 × 1
4:2:0	2 × 2
4:1:1	4 × 1

There are also minimum (MCU) sizes. Using images sizes below these limits does not produce reliable results.

Table 8.20 Minimum Size

YUV Format	MCU Size (Horizontal × Vertical)
4:4:4	(8 × 8)
4:2:2	16 × 8
4:2:0	16 × 16
4:1:1	32 × 8

There are two quantization tables for encode and four for decode. There are two Huffman tables each for both AC and DC components for both encode and decode.

JPEG encode supports insertion of markers up to 36 bytes long (including the marker identifier).

Decode processing automatically recognizes the markers SOI, SOF0, SOS, DQT, DHT, DRI, RSTm, DNL, and EOI and ignores all others.

Camera image JPEG encode supports YUV data formats 4:4:4, 4:2:2, 4:2:0, and 4:1:1. Note, however, that this support does not cover YUV 4:4:4 input without scaling, a combination that exceeds the camera image data capacity. RAM-based conversions between YUV data and JPEG data support only YUV data formats 4:2:2 and 4:2:0.

The target processing time is a maximum of 1/30 seconds for the VGA size (640×480), but this is not guaranteed because throughput can vary widely with the quantization table settings, the Huffman table settings, camera input image details, and other factors.

Ensuring stable operation during repeated JPEG encode requires leaving at least one frame interval between repetitions. The JPEG module offers support for consecutive JPEG encode, but whether such throughput is actually possible depends on the type of camera module used by the system and the software processing speed.

More specifically, the following inequality must be satisfied.

VREF inactive interval from the camera
> interrupt request response time
+ total time lost to processing higher-priority tasks
+ time spent setting up to encode next frame
+ JPEG codec marker output time

Enabling the JPEG marker fast output mode added with the S1S65010 reduces JPEG codec marker output times to 36 μs, otherwise approximately 2 ms. (All measurements with a system clock of 50 MHz.) Consecutive JPEG encode for the VGA size with JPEG marker fast output mode disabled runs an extremely high risk of overflow if the VREF inactive interval does not correspond to at least 15 lines.

8.5.4.1 Invalid JPEG Files

JPEG decode is not possible for the following types of files.

- Files that are not in JPEG format (even though they may have the file extension .“jpg”)
- Files with damaged markers
- Gray scale JPEG files with no UV data
- JPEG files with non-YUV color elements
- JPEG files using DNL markers
- JPEG files with damaged data in Huffman tables, quantization tables, or other non-image portions

The JPEG standard does not provide error correction at the bit level, and covering all the possible error states is too difficult in hardware, so this device expects prechecking by the software. If the software cannot guarantee that the input file contains valid JPEG data, it must first read the JPEG file markers to make sure that the JPEG codec can decode the JPEG file. Otherwise, the JPEG Decode Marker Read Flag does not indicate valid interrupt requests, and JPEG decode runs out of control. A single bit reversal in a marker code, for example, can prevent successful JPEG decoding. Software developers should be on the lookout for open-ended failure to complete decoding, an indicator for bugs.

8.5.4.2 Usage Restrictions for JPEG Codec Registers

The JPEG Codec Registers may not be accessed during JPEG encode or decode operations (the period between the point a JPEG codec operation is started until that operation completes). Even if the JPEG codec responds to a register access during this period, that access will either cause the codec to malfunction or will be invalid.

Accessing the JPEG Codec Registers generally does not yield valid results if there is no clock signal to the JPEG codec because the JPEG module is either disabled with “0” in JCTL[0x00] bit 0 (JPEG module enable) or enabled with a setting other than “x00” in bits 3 to 1 (JPEG operation mode) in the same register. The registers for configuring YUV data capture, however, are an exception. They remain accessible.

Reading JPEG Codec Registers labeled reserved or write only not only returns undefined values, but also risk invalidating JPEG codec operation itself. JPEG decode and JPEG encode require particular care because they are subject to additional register access restrictions.

Read the JPEG Codec Status and JPEG Codec Marker Status Registers only as necessary because the read itself changes the internal state.

When JPEG encode or JPEG decode ends, read the JPEG Codec Marker Status Register before reading the JPEG Codec Status Register. Skipping the first read leaves JPEG Codec processing incomplete, interfering with subsequent processing.

Quantization and Huffman tables must be reloaded after any switch between decode and encode processing or even changing an encode setting. Reloading can be skipped, however, if the same processing just continues.

8.5.5 Functions other than JPEG Codec

8.5.5.1 JPEG FIFO

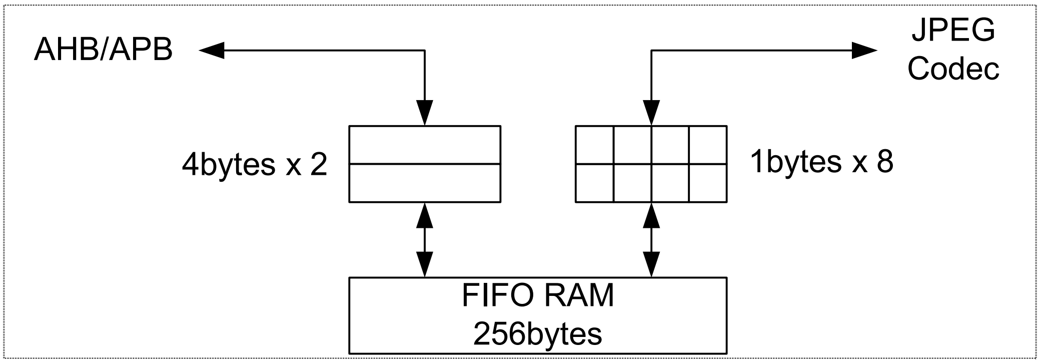


Fig.8.19 JPEG FIFO Overview

The JPEG FIFO holds up to 272 bytes of data in 256 bytes of RAM and two 4-byte read/write buffers. CPU read/write access to the FIFO while checking the JPEG FIFO status is best when using the maximum setting to access all 256 bytes of RAM.

The JPEG FIFO Status Register gives the JPEG FIFO status. The JPEG Interrupt Status Register contains most of those status bits as well. The former gives the real-time status in effect at the time of the read; the latter contains empty and full flags that, once set, retain their values until reset by writing “0” to the corresponding bit in the JPEG Interrupt Control Register (JCTL[0x0C]) to clear the interrupt request.

Camera image JPEG encode and YUV data JPEG encode have two ways to extract data from the JPEG FIFO: having the CPU read it from the JPEG FIFO Read/Write Port Register and using the JPEG DMA transfers.

1. Low-performance approach
   The CPU reads data from the JPEG FIFO Read/Write Port Register when the JPEG FIFO empty flag indicates that there is data available. The FIFO is too small to efficiently use FIFO full and threshold interrupt requests. This constant polling of the JPEG FIFO empty flag and reading data until the JPEG codec end interrupt request flag goes to “1” eats up a great deal of CPU time, so this approach does not lend itself to multitasking or real-time processing.

2. High performance approach
   Here the JPEG_DMAC transfers data into memory, automatically stopping when the JPEG FIFO sends it a frame end indication. All that is necessary is to allocate sufficient memory and to specify DMA settings for the maximum size possible. The register settings for each frame are minimal, and there is almost no CPU load involved in transferring from the FIFO.

JPEG decode processing repeatedly writes a 256-byte block of data to the JPEG FIFO Read/Write Port Register each time that the FIFO empty status bit goes to “1.”

8.5.5.2 JPEG Line Buffer

The JPEG line buffer has 30 KB of RAM for alternately converting between baseline data and block interleave data. It processes image horizontal widths up to 640 pixels.

JPEG files require image data in multiples of the basic MCU size, so the JPEG line buffer interpolates to make up for any shortfalls—expanding a 100×100 image in YUV 4:2:2 format to 112×104, for example, before sending the data to the JPEG codec. Note that this capability supplies only enough data to satisfy the MCU size requirement. It cannot expand a 60×60 image from the capture resizer to 112×104 or and other bigger sizes, for example.

The JPEG line buffer has a data port register for JPEG data YUV encode and JPEG decode.

The JPEG Line Buffer Status Register gives the JPEG line buffer status. The full and empty status bits are for triggering interrupt requests during YUV data JPEG encode and JPEG decode.

The JPEG FIFO or JPEG codec includes facilities for waiting JPEG encoding data input when they cannot process it fast enough. The JPEG line buffer, however, has no such facilities for stopping data from the capture resizer, so can overflow if the JPEG codec frequently keeps it waiting. The resulting JPEG line buffer overflow interrupt requests therefore indicates failure to maintain real-time encoding as the result of JPEG FIFO read delays, excessive JPEG encode load, or other factors.

Because the JPEG Line Buffer alternately converts between baseline data and block interleave data, it handles data in banks with the number of lines matching the basic MCU size width—16 lines per bank for the YUV 4:2:0 format and 8 for 4:2:2, for example—and adjusts the number of banks holding data in RAM according to the input image width to maximize usage of the 30 KB of RAM available.

Table 8.21 Input Image Width and Number of Banks

Input Image Width	Banks
≤32	32
≤64	16
≤128	8
≤256	4
>256	2

The total RAM used is the input image width times the number of banks.

8.5.5.3 YUV Data Format Converter

The YUV data format converter converts the YUV 4:4:4 camera image from the capture resizer to four YUV data formats. The following Figure shows the averaging formulas for the U component. The ones for the V component are similar. The Y component remains the same as in the source image.

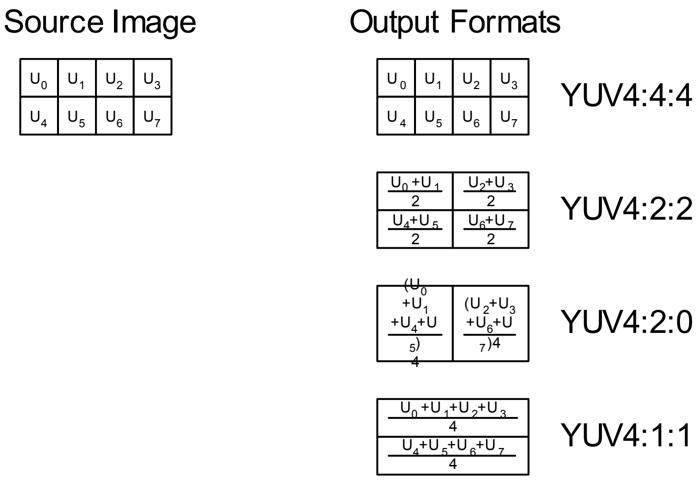


Fig.8.20 YUV Data Format Conversions

8.5.5.4 JPEG Module Interrupt Requests

The following describes how interrupt request flags are used.

1. JPEG codec interrupt request flag

Camera image JPEG encode and YUV data JPEG encode use this to signal the CPU to access the JPEG FIFO. When this bit goes to "1," the JPEG file size is known, so subtracting the number of data bytes read from the number in the encoded result gives the number remaining to be read at the end.

This bit never goes to "1" during YUV data capture.

JPEG decode uses this bit as the trigger for reading the last data from the JPEG line buffer.

2. JPEG line buffer overflow interrupt request flag

Only camera image JPEG encode uses this bit. Enabling overflow interrupt requests is essential here because the JPEG codec provides no guarantee that JPEG encode always proceeds in real time.

3. JPEG decode marker read flag

This bit goes to "1" when decoding is suspended. Only JPEG decode uses this bit. Do not rely upon it, however, in situations where it is impossible to guarantee that the JPEG codec can successfully decode the JPEG file. If necessary, the software must first check all markers in the JPEG file.

Disabling JPEG decode marker read Interrupt requests fixes this bit and its raw status counterpart to "0."

4. JPEG FIFO empty flag

JPEG decode sometimes uses this bit to signal the CPU to fill the now empty FIFO with the next block of data. Note, however, that having an empty FIFO suspends JPEG decode, so writing blocks half the FIFO size when the JPEG FIFO threshold status field indicates quarter full can boost JPEG decoding throughput.

Although JPEG encode can use this bit to check for the end of FIFO data, there is no particular need to do so because the bit has mostly a double-checking meaning.

5. JPEG FIFO full flag

Camera image JPEG encode and YUV data JPEG encode sometimes use this bit to signal the CPU to read the block of data now filling the FIFO. Note, however, that having a full state suspends JPEG encode, causing data to continue accumulating in the JPEG line buffer, so a more practical approach is to read blocks using the setting in the JPEG FIFO Size Register when the JPEG FIFO threshold status field indicates half or quarter full.

6. JPEG FIFO threshold trigger flag

This bit has no particular use because the FIFO is too small and the interrupt request response time overhead too big to make it practical to base FIFO access on interrupt requests triggered by a threshold.

7. Encode size limit violation flag

Camera image JPEG encode and YUV data JPEG encode use this bit. Using the JPEG DMAC's interrupt requests indicating the maximum number of DMA transfers does not destroy memory regions, so cannot serve as a direct method for limiting the maximum size of the JPEG file. The software can use this instead to detect when the limit is about to be exceeded and, for example, change quantization tables to bigger values.

8.5.5.5 JPEG 180° Rotation Encode

This function does not rotate the entire frame—only blocks containing the number of lines equal to the basic MCU width. The software must then rearrange the data. The hardware output leaves the first such block, for example, at the beginning of the JPEG file’s image data region. The software must move it to the end, its proper place in the rotated frame.

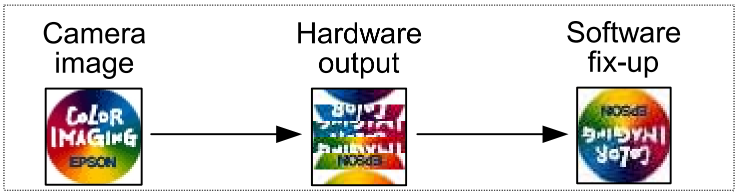


Fig.8.21 JPEG 180° Rotation Encode

Indicating the ends of these image subblocks to the software with RST markers is essential here because otherwise it is difficult to discern them from the encoded image data alone. The insertion spacing is the image horizontal width divided by the basic MCU size width and then rounded up to an integer. Rotating a YUV 4:2:0 image 100 pixels wide, for example, inserts an RST marker after every seventh (100/16=6.25) MCU size width in the JPEG output.

8.5.5.6 YUV Data Formats

YUV data JPEG encode, JPEG decode, and YUV data capture all involve handling YUV data in software. These processing modes support only YUV data formats 4:2:2 and 4:2:0, which have the following layouts.

	YUV 4:2:2	YUV 4:2:0
Nth line	UYVYUYVY	UYVYUYVY
N+1th line	UYVYUYVY	YYYYYYYY

The YUV data is in big-endian byte order, with U, Y, V, and Y values in ascending address order. YUV 4:2:0 shares U and V data between adjacent line pairs to make room for more Y data on the odd lines.

8.5.5.7 JPEG Module Software Reset

The frame is the basic unit for encode and decode with the JPEG codec. Repeating the same basic cycle yields encode or decode of a stream of frames. Before starting, however, we recommend a software reset to return the JPEG codec to its initial state and thus ensure stable operation. It resets only functional blocks, so registers retain all settings.

A JPEG module software reset affects the YUV data format converter, the JPEG line buffer, and the JPEG FIFO. We recommend this one prior to camera image JPEG encode because of processing lags between these components, but not during consecutive JPEG encode because there is a high possibility of overlap between the parallel processes of completing the current frame and starting the next one. Leaving at least one frame interval between frames, on the other hand, guarantees no overlap, so we recommend a JPEG module software reset before resuming JPEG encoding.

A JPEG module software reset is obligatory after decode or encode of an image that is not a multiple of the basic MCU size. In other words, consecutive JPEG encode is not possible with such frames.

8.5.5.8 JPEG Marker Fast Output Mode

JPEG markers distinguish data types inside JPEG files. The JPEG codec places most markers before the compressed data.

Enabling the JPEG marker fast output mode added with this version reduces JPEG codec marker output times to 36 μs, otherwise approximately 2 ms. (All measurements with a system clock of 50 MHz.) Note that disabling JPEG encode fast mode also, regardless of the register settings, disables JPEG marker fast output mode because the speed-up relies on the use of fixed values in the Huffman tables.

8.5.6 Sample Sequences

8.5.6.1 Camera Image JPEG Encode (Single Frame)

The following describes the camera image JPEG encode sequence for a single frame using DMA transfers to read data from the FIFO.

1. Configure the camera interface.
   For further details, see the camera interface functional description.
2. Simultaneously set JCTL[0x00] bit 0 (JPEG module enable) to “1” and bits 3 to 1 (JPEG operation mode) in the same register to “000.”
3. Set JCTL[0x00] bit 7 (JPEG module software reset) to “1.”
4. Initialize the JPEG Codec Registers.
   Note that, although the following generally initializes in basically ascending order, the only position-sensitive steps are those involving command registers.
   (a) Set JCODEC[0x04] bit 7 (JPEG codec software reset) to “1.”
   (b) Set JCODEC[0x00] bit 2 (JPEG operation mode) to “0” (encode).
   (c) If inserting user-specified markers, set JCODEC[0x00] bit 3 (marker insert enable) to “1” (enable).
   (d) Specify quantization table and Huffman table numbers in JCODEC[0x0C] and JCODEC[0x10].
   (e) If insert RST markers, specify the insertion spacing in JCODEC[0x14] and JCODEC[0x18].
   (f) Specify the input image size in JCODEC[0x1C], JCODEC[0x20], JCODEC[0x24], and JCODEC[0x28].
   (g) If insert RST markers, specify the marker data in JCODEC[0x40-0xCC].
   (h) Load the quantization tables (JCODEC[0x400-0x4FC] and JCODEC[0x500-0x5FC]) in the following order.

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

   (i) Load the Huffman tables (JCODEC[0x800-0xF44]).

The following example uses the values specified in ISO/IEC 10918-1 Annex K.

Copy pattern A into DC Huffman table No. 0 register 0 (JCODEC[0x800-0x83C]).
Copy pattern B into DC Huffman table No. 0 register 1 (JCODEC[0x840-0x86C]).
Copy pattern E into AC Huffman table No. 0 register 0 (JCODEC[0x880-0x8BC]).
Copy pattern F into AC Huffman table No. 0 register 1 (JCODEC[0x8C0-0xB44]).
Copy pattern C into DC Huffman table No. 1 register 0 (JCODEC[0xC00-0xC3C]).
Copy pattern D into DC Huffman table No. 1 register 1 (JCODEC[0xC40-0xC6C]).
Copy pattern G into AC Huffman table No. 1 register 0 (JCODEC[0xC80-0xCBC]).
Copy pattern H into AC Huffman table No. 1 register 1 (JCODEC[0xCC0-0xF44]).

A:	00h, 01h, 05h, ..., 00h, 00h	16 bytes
B:	00h, 01h, 02h, ..., 0Ah, 0Bh	12 bytes
C:	00h, 03h, 01h, ..., 00h, 00h	16 bytes
D:	00h, 01h, 02h, ..., 0Ah, 0Bh	12 bytes
E:	00h, 02h, 01h, 03h, ...1h, 7Dh	16 bytes
F:	01h, 02h, 03h, ..., F9h, FAh	162 bytes
G:	00h, 02h, 01h, 02h, ..., 02h, 77h	16 bytes
H:	00h, 01h, 02h, ..., F9h, FAh	162 bytes

5. Initialize the JPEG module.
   (a) Set JFIFO[0x48] to “0x3F” to set the JPEG FIFO size to use all dedicated RAM available.

(b) Specify the encode size limit in bytes (JFIFO[0x60], JFIFO[0x64]).
(c) Write “1” to JFIFO[0x40] bit 2 to clear the JPEG FIFO.

6. Write “1” to RSZ[0xC0] bit 7 to initialize the capture resizer.
   Make the image size after resizing (RSZ[0xC8], RSZ[0xCC], RSZ[0xD0], and RSZ[0xD4]) the same as that specified in step 4(f) above.
7. Configure interrupt requests.
   Write 0x0000FFFF to the JPEG Status Flag Register (JCTL[0x04]) to clear all interrupt requests.
   Set JCTL[0x0C] bits 2 and 11 both to “1” to enable JPEG line buffer overflow and encode size limit violation interrupt requests. Writing “1” to reserved bits in the JPEG Status Flag Register does not present problems.
   Enable JPEG module interrupt requests in the Interrupt Controller IRQ Enable Register (INT[0x008]).
8. Configure the JPEG_DMAC (JDMA[0x00] to [0x40]).
9. Start JPEG encode.
   (a) Set JCODEC[0x04] bit 0 to “1” to start the JPEG codec.
   (b) Set JCTL[0x14] bit 0 to “1” to start the JPEG module.
   Starting the JPEG module does not start capture until after the approximately 2 ms (with a system clock of 50 MHz) that the JPEG codec takes for JPEG marker output.
10. Wait for JPEG_DMAC frame end interrupt requests.
    Shut things down, however, if there is a JPEG line buffer overflow interrupt request or JPEG encode does not complete within a reasonable length of time.

8.5.6.2 Shutting Down

All processing modes can share a common shut down routine for recovering from any faults and returning to the start-up state.

Note that this sequence omits the dummy register reads that must follow writes to JPEG Codec Registers.

1. Write “0x0000” to the Global Resizer Control Register (RSZ[0x60]) just in case incorrect values have been written to reserved registers.
2. Set RSZ[0xC0] bit 0 (capture resizer enable) to “1” and then, write “1” to bit 7 (software reset) in the same register.
3. Set JCTL[0x00] bit 0 (JPEG module enable) to “1” and bits 3 to 1 (JPEG operation mode) to “000.”
4. Set JCODEC[0x04] bit 7 (JPEG Codec Software Reset) to “1.”
5. Perform a dummy read from the JPEG codec RST Marker Operation Status Register (JCODEC[0x3C]).
6. Perform a dummy read from the JPEG codec JPEG Operation Status Register (JCODEC[0x08]).
7. Write “0x00”to the JPEG Codec Operation Mode Setting Register (JCODEC[0x00]).
8. Set JCTL[0x00] bit 7 (JPEG module software reset) to “1.”
9. Write “0x0000” to the JPEG Line Buffer Interrupt Control Register (JLB[0x8C]) to disable all JPEG line buffer interrupt requests.
10. Write “0xFFFF” to the JPEG Line Buffer Status Flag Register (JLB[0x80]).
11. Write “0x0000” to the JPEG Interrupt Control Register (JCTL[0x0C]) to disable all JPEG interrupt requests.
12. Write “0xFFFF” to the JPEG Status Flag Register (JCTL[0x04]).
13. Write “1” to JDMA[0x20] bit 15 (JPEG_DMAC software reset).
14. Set JPEG_DMAC Control Register JDMA[0x0C] bits 0 (DMA enable) and 21 (JPEG interrupt enable) to “0” (disable).
15. Set JDMA[0x0C] bit 1 to “0” to clear JPEG_DMAC interrupt flags.
16. Disable JPEG controller and JPEG DMA interrupt requests to the interrupt controller.
17. Set JCTL[0x00] bit 0 (JPEG module enable) to “0” to disable the JPEG module.
18. Set RSZ[0xC0] bit 0 to “0” to disable the capture resizer.

Note: Always disable first the JPEG module and then the Capture resizer. Using the reverse order sometimes fails to stop the clock signal to the JPEG codec. If the Capture Resizer is already disabled—YUV data JPEG encode and JPEG decode, for example, do not use it—Enable it, disable the JPEG module, and then disable the Capture Resizer once again.

9. JPEG_DMAC (JDMA)

9.1 Overview

This DMA controller drives DMAC2 for a single key task, receiving image data from the camera interface. It sets up the DMA transfers in the DMAC2 registers. The FIFO inside the JPEG controller provides DMAC2 with request and acknowledge signals.

9.2 Block Diagram

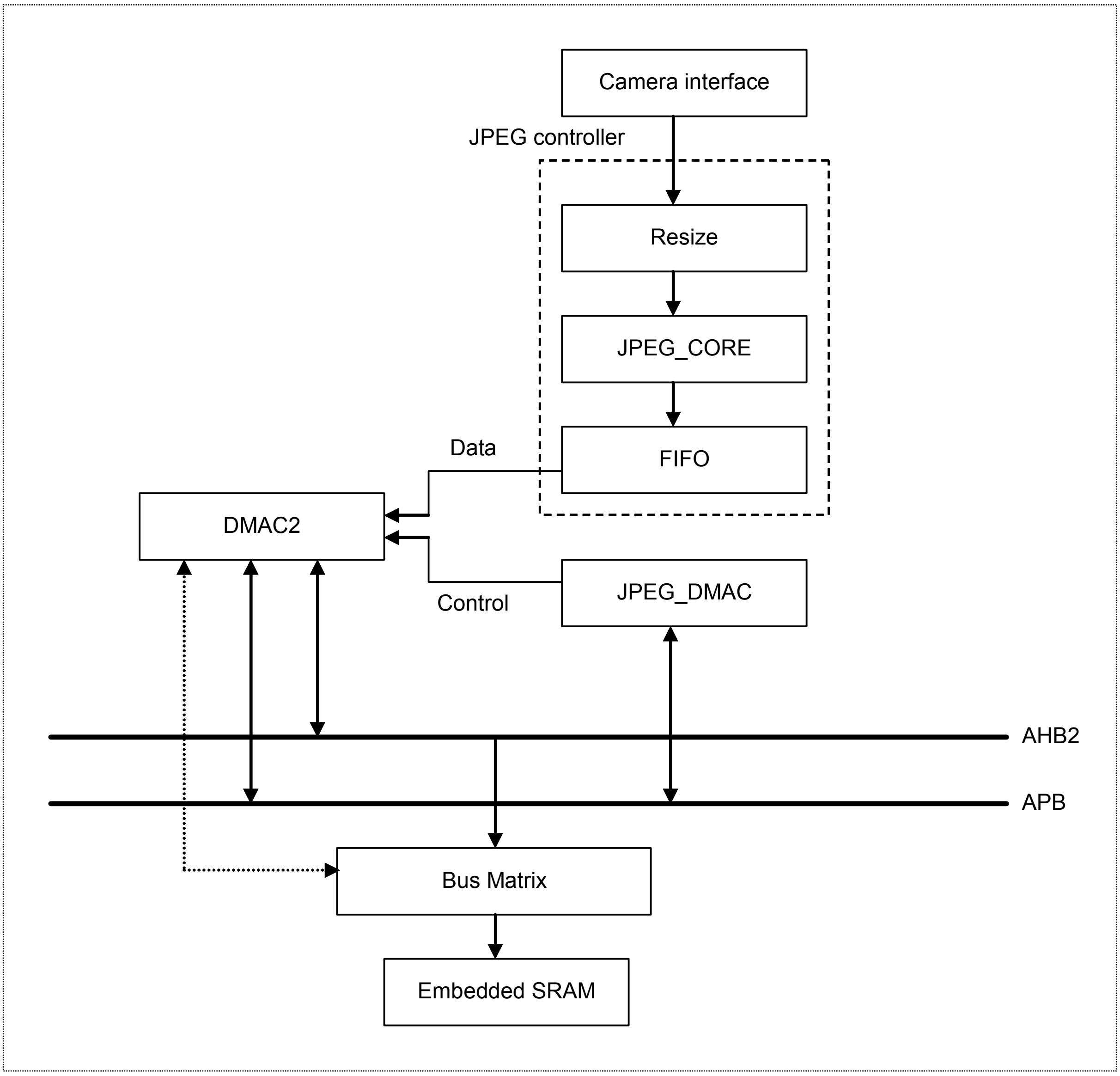


Fig.9.1 JPEG_DMAC Relationships with JPEG Controller, DMAC2, etc.

9.3 External Pins

This block interacts with no external pins.

9.4 Registers

9.4.1 Register List

The base address for these registers is 0xFFFE_C000.

Table 9.1 Register List (Base Address: 0xFFFE_C000)

Address Offset	Register Name	Abbreviation	Default Value*	R/W	Data Access Size (Bits)
0x00	JPEG DMA Source Address Register	JSAR	0xXXXX_XXXX	R/W	32
0x04	JPEG DMA Destination Address Register	JDAR	0xXXXX_XXXX	R/W	32
0x08	JPEG DMA Transfer Count Register	JTCR	0x0000_0000	R/W	32
0x0C	JPEG DMA Control Register	JCTL	0x0000_0000	R/W	32
0x10	JPEG DMA Block Count Register	JBCR	0x00XX_XXXX	R/W	32
0x14	JPEG DMA Destination Offset Address Register	JOFR	0x0000_0000	R/W	32
0x18	JPEG DMA Block End Count Register	JBER	0x00XX_XXXX	R/W	32
0x20	JPEG DMA Expansion Register	JHID	0x0000_0000	R/W	32
0x40	JPEG DMA FIFO Data Select Mode Register	JFSM	0x0000_0000	R/W	32

* X: Undefined value (hexadecimal digit)

9.4.2 Detailed Register Descriptions

JPEG DMA Source Address Register (JSAR)
JDMA[0x00] Default = 0xXXXX_XXXX Read/Write

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

Bits 31 to 0: **JPEG DMA Source Address [31:0]**
This register specifies the source address for JPEG DMA transfers. The hardware does not modify the contents of this register.

JPEG DMA Destination Address Register (JDAR)

JDMA[0x04] Default = 0xXXXX_XXXX Read/Write

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

Bits 31 to 0: **JPEG DMA Destination Address [31:0]**

This register specifies the destination address for JPEG DMA transfers. After each successful block transfer, the controller, in preparation for the next transfer, automatically updates the current contents of this register by adding the contents of the JPEG DMA Destination Offset Address Register (JOFR).

JPEG DMA Transfer Count Register (JTCR)

JDMA[0x08] Default = 0x0000_0000 Read/Write

n/a								JPEG DMA Transfer Count [23:16]							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
JPEG DMA Transfer Count [15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 23 to 0: **JPEG DMA Transfer Count [23:0]**

These bits specify the number of transfers with JPEG DMAC.
The hardware does not modify the contents of this register.
Reads return zeros in bits 31 to 24.

JPEG DMA Control Register (JCTL)

JDMA[0x0C] Default = 0x0000_0000 Read/Write

n/a								RSV	JS	JIE	JCS	RSV		AM	AL
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DAM		SAM		RS				RSV		TM	TS		IE	JTE	DE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit 23 (RSV): **Reserved (0)**

Bit 22 (JS): **JPEG DMA Transfer Start**

0: No transfer in progress
1: Transfer in progress

This bit automatically returns to “0” when the block transfer is complete or the hardware asserts FIFO_END.

Bit 21 (JIE): **JPEG Interrupt Enable**

0: Disable
1: Enable

When this bit is “1”, changing bit 1 (JTE) to “1” occurs a interrupt request.

Bit 20 (JCS): **JPEG DMA Channel Select**

This bit specifies the DMA channel to use.

0: DMA channel 0
1: DMA channel 1

Bits 19 to 18 (RSV): **Reserved (0)**

Bit 17 (AM): **Acknowledge Mode**

DACK signal output timing select

0: Active in Read cycle
1: Active in Write cycle

Bit 16 (AL): **Acknowledge Level**

DACK signal output polarity select
0: Low active
1: High active

Bits 15 to 14 (DAM): **Destination Address Mode**

This field specifies the strategy for updating the Destination Address Register after a successful transfer.
00: Leave fixed (Do not update)
01: Increment according to the transfer data size (8 bits: +1, 16 bits: +2, 32 bits: +4)
10: Decrement according to the transfer data size (8 bits: –1, 16 bits: –2, 32 bits: –4)
11: Reserved

Bits 13 to 12 (SAM): **Source Address Mode**

This field specifies the strategy for updating the Source Address Register after a successful transfer.
00: Leave fixed (Do not update)
01: Increment according to the transfer data size (8 bits: +1, 16 bits: +2, 32 bits: +4)
10: Decrement according to the transfer data size (8 bits: –1, 16 bits: –2, 32 bits: –4)
11: Reserved

Bits 11 to 8 (RS): **Resource Select**

Specify the source for starting the DMA transfer.
0010: This bit pattern is fixed.
Other: Reserved

Bits 7 to 6 (RSV): **Reserved (0)**

Bit 5 (TM): **Transfer Mode**

0: Single mode
1: Demand mode

Bits 4 to 3 (TS): **Transfer Size [1:0]**

00: 8 bits
01: 16 bits
10: 32 bits
11: Reserved

Bit 2 (IE): **Interrupt Enable**

0: Disable
1: Enable
Setting this bit to "1" produces an interrupt request after each block transfer.

Bit 1 (JTE): **JPEG DMA Transfer End**

0 (r): Transfers in progress or channel idle
1 (r): JPEG DMA block transfer complete
0 (w): Clear this bit to "0"
1 (w): (Ignored)

This bit goes to "1" when all transfers are complete—that is, the JPEG DMA Block Count Register has decremented to zero. It retains this "1" setting until the software writes "0" to clear it to "0." DMA transfers on the channel are disabled until this bit returns to "0."

This bit also functions as an interrupt flag.

Bit 0 (DE): **DMA Enable**

0: Disable

1: Enable

Setting this bit to “1” enables JPEG DMA transfers on the channel.

JPEG DMA Block Count Register (JBCR)

JDMA[0x10] Default = 0x00XX_XXXX Read/Write

n/a								JPEG DMA Block Count [23:16]							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
JPEG DMA Block Count [15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 23 to 0: **JPEG DMA Block Count**

These bits specify the total number of JPEG DMA block transfers.

After each successful block transfer, the controller automatically decrements this register.

Reads return zeros in bits 31 to 24.

JPEG DMA Destination Offset Address Register (JOFR)

JDMA[0x14] Default = 0x0000_0000 Read/Write

n/a								JPEG DMA Destination Offset Address [23:16]							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
JPEG DMA Destination Offset Address [15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 23 to 0: **JPEG DMA Destination Offset Address [23:0]**

These bits specify the offset added to the JPEG DMA Destination Address Register (JDAR) after each JPEG DMA block transfer.

The hardware does not modify the contents of this register.

Reads return zeros in bits 31 to 24.

JPEG DMA Block End Count Register (JBER)

JDMA[0x18] Default = 0x00XX_XXXX Read/Write

n/a								JPEG DMA Block End Count [23:16]							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
JPEG DMA Block End Count [15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 23 to 0: **JPEG DMA Block End Count [23:0]**

These bits specify the number of JPEG DMA block transfers. After each successful block transfer, the controller automatically increments this register. The register therefore normally displays the current number of blocks transferred.

Reads return zeros in bits 31 to 24.

JPEG DMA Expansion Register (JHID)

JDMA[0x20] Default = 0x0000_0000 Read/Write

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

Bit 15 (SW): **Software Reset**

This bit applies a software reset to all the JDMA Registers.

Writing “1” to this bit applies a software reset.

This bit reverts to “0” after the reset operation completes.

JPEG DMA FIFO Data Select Mode Register (JFSM)															
JDMA[0x40]		Default = 0x0000_0000												Read/Write	
n/a															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
n/a										RSV		FM	RSV		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 5 to 4 (RSV): **Reserved (0)**

Bit 3 (FM): **FIFO Mode**

Bus selection for the data output JPEG FIFO

0: APB bus

1: AHB bus

Bits 2 to 0 (RSV): **Reserved (0)**

10. DMA CONTROLLER 2 (DMAC2)

10.1 Overview

This DMA controller transfers data under the control of the JPEG_DMAC module or software.

10.2 Block Diagram

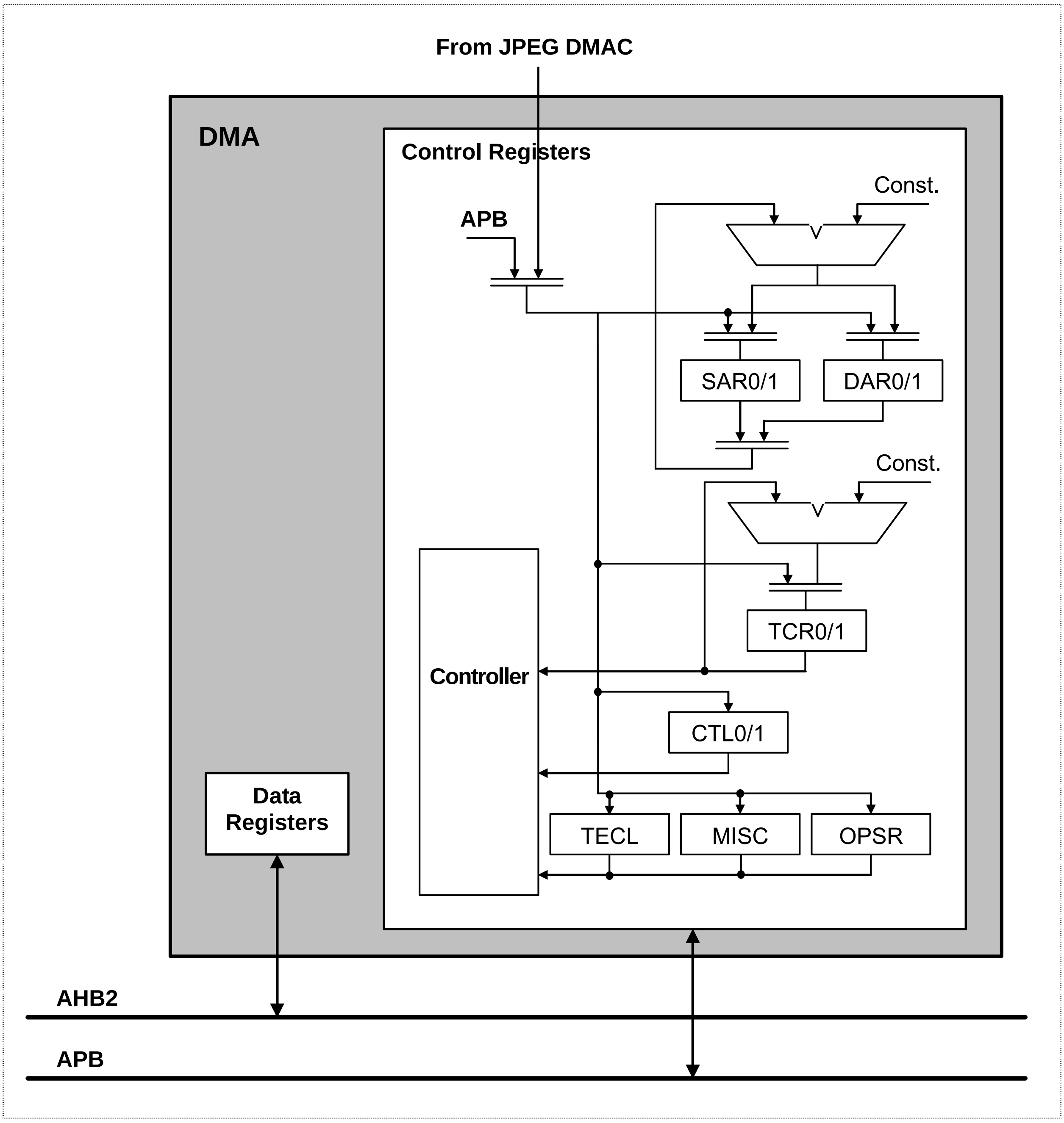


Fig.10.1 DMA Controller 2 (DMAC2) Block Diagram

10.3 External Pins

This block interacts with no external pins.

10.4 Registers

10.4.1 Register List

The base address for these registers is 0xFFFF_9000.

Table 10.1 Register List (Base Address: 0xFFFF_9000)

Address Offset	Register Name	Abbreviation	Default Value	R/W	Data Access Size (Bits)
0x00	DMA Channel 0 Source Address Register	SAR0	0xXXXX_XXXX	R/W	32
0x04	DMA Channel 0 Destination Address Register	DAR0	0xXXXX_XXXX	R/W	32
0x08	DMA Channel 0 Transfer Count Register	TCR0	0x00XX_XXXX	R/W	32
0x0C	DMA Channel 0 Control Register	CTL0	0x0000_0000	R/W	32
0x10	DMA Channel 1 Source Address Register	SAR1	0xXXXX_XXXX	R/W	32
0x14	DMA Channel 1 Destination Address Register	DAR1	0xXXXX_XXXX	R/W	32
0x18	DMA Channel 1 Transfer Count Register	TCR1	0x00XX_XXXX	R/W	32
0x1C	DMA Channel 1 Control Register	CTL1	0x0000_0000	R/W	32
0x60	DMA Channel Operating Select Register	OPSR	0x0000_0000	R/W	32
0x64	DMA Channel MISC Register	MISC	0x0000_0000	R/W	32
0x70	DMA Channel Transfer Complete Control Register	TECL	0x0000_0000	R/W	32

10.4.2 Detailed Register Descriptions

DMA Channel 0 Source Address Register (SAR0)

DMAC2[0x00] Default = 0xXXXX_XXXX Read/Write

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

Bits 31 to 0: **DMA Channel 0 Source Address [31:0]**

This register specifies the source address for the DMA transfer on channel 0.

This address must fall on a memory boundary that matches the transfer data size. A 32-bit transfer, for example, requires 00b in Source Address Register bits 1 and 0.

After each successful transfer, the controller, in preparation for the next transfer, automatically updates this register by the transfer data size in bytes (TS: bits 4 to 3 in the Channel 0 Control Register) according to the source address mode (SAM: bits 13 to 12 in the in the Channel 0 Control Register).

DMA Channel 0 Destination Address Register (DAR0)
DMAC2[0x04] Default = 0xXXXX_XXXX Read/Write

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

Bits 31 to 0: **DMA Channel 0 Destination Address [31:0]**

This register specifies the destination address for the DMA transfer on channel 0.

This address must fall on a memory boundary that matches the transfer data size. A 32-bit transfer, for example, requires 00b in Source Address Register bits 1 and 0.

After each successful transfer, the controller, in preparation for the next transfer, automatically updates this register by the transfer data size in bytes (TS: bits 4 to 3 in the Channel 0 Control Register) according to the destination address mode (DAM: bits 15 to 14 in the in the Channel 0 Control Register).

DMA Channel 0 Transfer Count Register (TCR0)
DMAC2[0x08] Default = 0x00XX_XXXX Read/Write

n/a								DMA Channel 0 Transfer Count [23:16]							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMA Channel 0 Transfer Count [15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 23 to 0: **DMA Channel 0 Transfer Count [23:0]**

These bits specify the number of DMA transfers remaining. This count decrements after each successful transfer. Starting at zero specifies 2^{24}=16,777,216 transfers. Decrementing to zero triggers a DMA interrupt request.

Reads return zeros in bits 31 to 24.

DMA Channel 0 Control Register (CTL0)
DMAC2[0x0C] Default = 0x0000_0000 Read/Write

n/a								RSV					IB4	AM	AL
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DAM		SAM		RS				RSV	RIM	TM	TS		IE	TE	DE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 23 to 19 (RSV): **Reserved (0)**

Bit 18 (IB4): **Increment Burst 4**

0: Disable
1: Enable

Writing "1" to this bit enables increment burst 4 transfers. Note, however, that such transfers require a transfer count evenly divisible by four. Set this bit to "0" to transfer any leftovers.

This functionality is for memory-to-memory block transfers only.

Bit 17 (AM): **Acknowledge Mode**

Select DACK signal output active timing

0: Active in DMA Read cycle
1: Active in DMA Write cycle

Bit 16 (AL): **Acknowledge Level**

Select DACK signal output polarity

0: Low active
1: High active

Bits 15 to 14 (DAM): **Destination Address Mode [1:0]**

This field specifies the strategy for updating the Destination Address Register after a successful transfer.

00: Leave fixed (Do not update)

01: Increment according to the transfer data size (8 bits: +1, 16 bits: +2, 32 bits: +4)

10: Decrement according to the transfer data size (8 bits: –1, 16 bits: –2, 32 bits: –4)

11: Reserved

Bits 13 to 12 (SAM): **Source Address Mode [1:0]**

This field specifies the strategy for updating the Source Address Register after a successful transfer.

00: Leave fixed (Do not update)

01: Increment according to the transfer data size (8 bits: +1, 16 bits: +2, 32 bits: +4)

10: Decrement according to the transfer data size (8 bits: –1, 16 bits: –2, 32 bits: –4)

11: Reserved

Bits 11 to 8 (RS): **Resource [3:0]**

1111: Software request (SW-Request)

Writing "1111" to bits 11 to 8 initiates DMA transfers. Note, however, that addresses must appear on the map in Section 4.2.2 "AHB2 Memory Map."

Other: Reserved

Bit 7 (RSV): **Reserved (0)**

Bit 6 (RIM): **Request Input Mode**

This specifies the input mode for the DMA request signal from the specified resource.

0: Low active (level trigger)

1: Falling edge (edge trigger)

Bit 5 (TM): **Transfer Mode**

0: Single transfer

1: Demand transfer

Bits 4 to 3 (TS): **Transfer Size [1:0]**

00: 8 bits

01: 16 bits

10: 32 bits

11: Reserved

Bit 2 (IE): **Interrupt Enable**

0: Disable

1: Enable

Setting this bit to "1" produces an interrupt request after DMA transfer completes.

Bit 1 (TE): **Transfer End**

0 (r): Transfers in progress or channel idle
1 (r): JPEG DMA transfer complete
0 (w): Clear this bit to “0”
1 (w): Ignored

This bit goes to “1” when all transfers are complete—that is, the DMA Channel 0 Transfer Count Register has decremented to zero. It retains this “1” setting until the software writes “0” to clear it to “0.” DMA transfers on the channel are disabled until this bit returns to “0.”

This bit also functions as an interrupt request source flag.

Bit 0 (DE): **DMA Enable**

0: Disable
1: Enable

Setting this bit to “1” enables DMA transfers on the channel 0.

DMA Channel 1 Source Address Register (SAR1)															
DMAC2[0x10]	Default = 0xXXXX_XXXX														Read/Write
DMA Channel 1 Source Address [31:16]															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMA Channel 1 Source Address [15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 31 to 0: **DMA Channel 1 Source Address [31:0]**

This register specifies the source address for the DMA transfer on channel 1.

This address must fall on a memory boundary that matches the transfer data size. A 32-bit transfer, for example, requires 00b in Source Address Register bits 1 and 0.

After each successful transfer, the controller, in preparation for the next transfer, automatically updates this register by the transfer data size in bytes (TS: bits 4 to 3 in the Channel 1 Control Register) according to the source address mode (SAM: bits 13 to 12 in the in the Channel 1 Control Register).

DMA Channel 1 Destination Address Register (DAR1)															
DMAC2[0x14]	Default = 0xXXXX_XXXX														Read/Write
DMA Channel 1 Destination Address [31:16]															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMA Channel 1 Destination Address [15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 31 to 0: **DMA Channel 1 Destination Address**

This register specifies the destination address for the DMA transfer on channel 1.

This address must fall on a memory boundary that matches the transfer data size. A 32-bit transfer, for example, requires 00b in Source Address Register bits 1 and 0.

After each successful transfer, the controller, in preparation for the next transfer, automatically updates this register by the transfer data size in bytes (TS: bits 4 to 3 in the Channel 1 Control Register) according to the destination address mode (DAM: bits 15 to 14 in the in the Channel 1 Control Register).

DMA Channel 1 Transfer Count Register (TCR1)															
DMAC2[0x18]	Default = 0x00XX_XXXX														Read/Write
n/a								DMA Channel 1 Transfer Count [23:16]							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMA Channel 1 Transfer Count [15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 23 to 0: **DMA Channel 1 Transfer Count [23:0]**

These bits specify the number of DMA transfers remaining. This count decrements after each successful transfer. Starting at zero specifies 2^{24}=16,777,216 transfers. Decrementing to zero triggers a DMA interrupt request.

Reads return zeros in bits 31 to 24.

DMA Channel 1 Control Register (CTL1)

DMAC2[0x1C] Default = 0x0000_0000 Read/Write

n/a								RSV					IB4	AM	AL
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DAM		SAM		RS				RSV	RIM	TM	TS		IE	TE	DE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 23 to 19 (RSV): **Reserved (0)**

Bit 18 (IB4): **Increment Burst 4**

0: Disable

1: Enable

Writing "1" to this bit enables increment burst 4 transfers. Note, however, that such transfers require a transfer count evenly divisible by four. Set this bit to "0" to transfer any leftovers.

This functionality is for memory-to-memory block transfers only.

Bit 17 (AM): **Acknowledge Mode**

Select DACK signal output active timing

0: Active in DMA read cycle

1: Active in DMA write cycle

Bit 16 (AL): **Acknowledge Level**

Select DACK signal output polarity

0: Low active

1: High active

Bits 15 to 14 (DAM): **Destination Address Mode [1:0]**

This field specifies the strategy for updating the Destination Address Register after a successful transfer.

00: Leave fixed (Do not update)

01: Increment according to the transfer data size (8 bits: +1, 16 bits: +2, 32 bits: +4)

10: Decrement according to the transfer data size (8 bits: –1, 16 bits: –2, 32 bits: –4)

11: Reserved

Bits 13 to 12 (SAM): **Source Address Mode [1:0]**

This field specifies the strategy for updating the Source Address Register after a successful transfer.

00: Leave fixed (Do not update)

01: Increment according to the transfer data size (8 bits: +1, 16 bits: +2, 32 bits: +4)

10: Decrement according to the transfer data size (8 bits: –1, 16 bits: –2, 32 bits: –4)

11: Reserved

Bits 11 to 8 (RS): **Resource [3:0]**

1111: Software request (SW-Request)

Writing "1111" to bits 11 to 8 initiates DMA transfers. Note, however, that addresses must appear on the map in Section 4.2.2 "AHB2 Memory Map."

Other: Reserved

Bit 7 (RSV): **Reserved (0)**

Bit 6 (RIM): **Request Input Mode**

This specifies the input mode for the DMA request signal from the specified resource.

0: Low active (level trigger)

1: Falling edge (edge trigger)

Bit 5 (TM): **Transfer Mode**

0: Single transfer
1: Demand transfer

Bits 4 to 3 (TS): **Transfer Size [1:0]**

00: 8 bits
01: 16 bits
10: 32 bits
11: Reserved

Bit 2 (IE): **Interrupt Enable**

0: Disable
1: Enable

Setting this bit to “1” produces an interrupt request after DMA transfer completes.

Bit 1 (TE): **Transfer End**

0 (r): Transfers in progress or channel idle
1 (r): JPEG DMA transfer complete
0 (w): Clear this bit to “0”
1 (w): Ignored

This bit goes to “1” when all transfers are complete—that is, the DMA Channel 1 Transfer Count Register has decremented to zero. It retains this “1” setting until the software writes “0” to clear it to “0.” DMA transfers on the channel are disabled until this bit returns to “0.”

This bit also functions as an interrupt request source flag.

Bit 0 (DE): **DMA Enable**

0: Disable
1: Enable

Setting this bit to “1” enables DMA transfers on the channel.

DMA Channel Operating Select Register (OPSR)
DMAC2[0x60] Default = 0x0000_0000 Read/Write

n/a															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
n/a						DPE	DPM	n/a							DGE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit 9 (DPE): **DMA Priority Toggling Enable**

0: Disable
1: Enable

Setting this bit to “1” toggles bit 8 (DPM) with a timing that depends on the mode.

Single transfers: after each transfer

Demand transfers: when the controlling request signal is negated, suspending transfers, and when the transfer counter goes to “0.”

Bit 8 (DPM): **DMA Priority Mode**

0: DMA channel 0 is high priority
1: DMA channel 1 is high priority

Bit 0 (DGE): **DMA Global Enable**

0: Disable
1: Enable

This simultaneously switches all DMA channels on and off.

DMA Channel MISC Register (MISC)

DMAC2[0x64] Default = 0x0000_0000 Read/Write

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

Bit 15 (SR): **Software Reset**

Writing “1” to this bit resets all DMAC2 registers to their initial values. In other words, the software must reconfigure the necessary registers before using the module.

Bits 1 to 0 (DPL): **DMA Polarities for each channel (DPL1 for channel 1; DPL0 for channel 0)**

0: Positive
1: Negative

DMA Channel Transfer Complete Control Register (TECL)

DMAC2[0x70] Default = 0x0000_0000 Read/Write

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

Bit 13 (STTE): **TE Set Enable Upon Transfer Complete Acceptance**

0: Disable
1: Enable

Writing “1” to this bit accepts of the transfer complete signal from the interrupt request source and sets the transfer end (TE) bit to “1.”
This bit is valid only when bit 12 (ENTE) is set to “1,”

Bit 12 (ENTE): **Transfer Complete Acceptance Enable**

0: Disable
1: Enable

Writing “1” to this bit accepts the transfer complete signal from the interrupt request source without setting the transfer end (TE) bit to “1.” Note, however, that setting TE to “1” requires writing “1” to bit 13 (STTE).

11. ETHERNET MAC & E-DMA (ETH)

11.1 Overview

The Ethernet DMA Controller for AHB (E-DMAC) includes a dedicated descriptor-based DMA controller and can perform Ethernet frame transfers efficiently without CPU load.

This block also includes a function for managing, in conjunction with its internal DMA controller, the external memory receive buffer free space status. If this function is enabled, the block can control operation so that pause frame transmissions are issued automatically before the receive buffer becomes full, thus preventing receive buffer overflows.

11.1.1 Features

- EPSON Fast Ethernet MAC
- 32-bit AHB master function
- Supports ring buffer structures based on the Descriptor Architechture.
- Supports multibuffer structures in which 1 descriptor = 1 frame.
- Dedicated DMA controller for the Fast Ethernet MAC
- The dedicated Ethernet MAC DMA supports burst transfers
- The results of Ethernet transmissions are reflected in the descriptor table.
- Automatic pause frame transmission function with IEEE 802.3x compliant.

11.2 Block Diagram

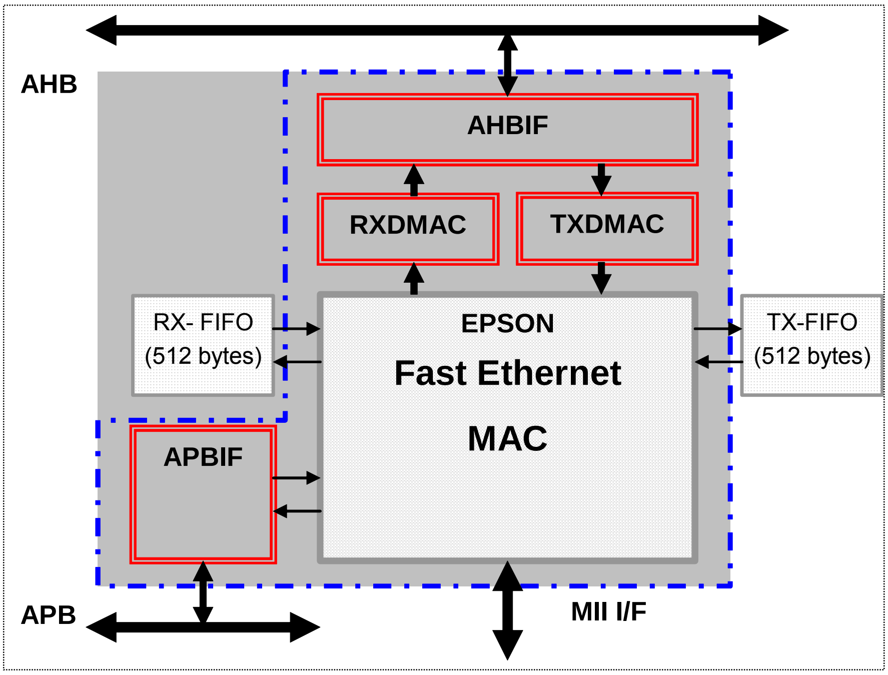


Fig.11.1 Block Diagram

11.3 External Pins

This block interacts with the following external pins.

Pin Name	I/O	Pin Function	Multiplexed Pin*
MII_TXCLK	Input	Media Independent Interface Ethernet PHY (MII_PHY) transmit data output clock TXCLK input	GPIOF7
MII_TXEN	Output	MII_PHY transmit enable TXEN output	GPIOF6
MII_TXD3	Output	MII_PHY transmit data TXD3 output	GPIOF2
MII_TXD2	Output	MII_PHY transmit data TXD2 output	GPIOF3
MII_TXD1	Output	MII_PHY transmit data TXD1 output	GPIOF4
MII_TXD0	Output	MII_PHY transmit data TXD0 output	GPIOF5
MII_RXCLK	Input	MII_PHY transmit data clock RXCLK input	GPIOG1
MII_COL	Input	MII_PHY collision detection COL input	GPIOF1
MII_CRS	Input	MII_PHY carrier sense CRS input	GPIOF0
MII_RXDV	Input	MII_PHY receive data enable RXDV input	GPIOG2
MII_RXD3	Input	MII_PHY receive data RXD3 input	GPIOG6
MII_RXD2	Input	MII_PHY receive data RXD2 input	GPIOG5
MII_RXD1	Input	MII_PHY receive data RXD1 input	GPIOG4
MII_RXD0	Input	MII_PHY receive data RXD0 input	GPIOG3
MII_RXER	Input	MII_PHY receive error RXER input	GPIOG0
MII_MDC	Output	MII_PHY management interface clock MDC output	GPIOG7
MII_MDIO	I/O	MII_PHY management interface data MDIO input/output	GPIOH0

Note*: While the ETH external pins are multiplexed with GPIO pin functions, these pins is set as the ETH pin function, i.e. the “non-GPIO function #1”. That is, they can be used as MII interface pins with the default settings. The settings in the GPIO Pin Function Register must be changed to use these pins for any other function.

11.4 Registers

11.4.1 Register List

The following table lists the registers for this block. The addresses shown in the table are APB bus addresses. The base address for these registers is 0xFFFE_2000.

Table 11.1 Register List (Base Address: 0xFFFE_2000)

Address Offset	Register Name	Default Value	R/W	Data Access Size (Bits)
0x00	Interrupt Status Register	0x 0000 0000	RO	32
0x04	Interrupt Enable Register	0x 0000 0000	R/W	32
0x08	Reset Register	0x 0000 2000	R/W	32
0x0C	PHY Status Register	0x 0000 0000	RO	32
0x10	DMA Command Register	0x 0000 0000	R/W	32
0x18	TX DMA Pointer Register	0x 0000 0000	R/W	32
0x1C	RX DMA Pointer Register	0x 0000 0000	R/W	32
0x20	Mode Register	0x 4000 0000	R/W	32
0x24	TX Mode Register	0x 0000 0000	R/W	32
0x28	RX Mode Register	0x 0000 0000	R/W	32
0x2C	MIIM Register	0x 0000 0000	R/W	32
0x30	MAC Address Register 1: Lower 32 bits	0x 0000 0000	R/W	32
0x34	MAC Address Register 1: Upper 16 bits	0x 0000 0000	R/W	32
0x38	MAC Address Register 2: Lower 32 bits	0x 0000 0000	R/W	32
0x3C	MAC Address Register 2: Upper 16 bits	0x 0000 0000	R/W	32
0x40	MAC Address Register 3: Lower 32 bits	0x 0000 0000	R/W	32
0x44	MAC Address Register 3: Upper 16 bits	0x 0000 0000	R/W	32
0x48	MAC Address Register 4: Lower 32 bits	0x 0000 0000	R/W	32
0x4C	MAC Address Register 4: Upper 16 bits	0x 0000 0000	R/W	32
0x50	MAC Address Register 5: Lower 32 bits	0x 0000 0000	R/W	32
0x54	MAC Address Register 5: Upper 16 bits	0x 0000 0000	R/W	32
0x58	MAC Address Register 6: Lower 32 bits	0x 0000 0000	R/W	32
0x5C	MAC Address Register 6: Upper 16 bits	0x 0000 0000	R/W	32
0x60	MAC Address Register 7: Lower 32 bits	0x 0000 0000	R/W	32
0x64	MAC Address Register 7: Upper 16 bits	0x 0000 0000	R/W	32
0x68	MAC Address Register 8: Lower 32 bits	0x 0000 0000	R/W	32
0x6C	MAC Address Register 8: Upper 16 bits	0x 0000 0000	R/W	32
0x70	Flow Control Register	0x 0000 0000	R/W	32
0x74	Pause Request Register	0x 0000 0000	R/W	32
0x78	Pause Frame Data Register 1	0x 0000 0000	R/W	32
0x7C	Pause Frame Data Register 2	0x 0000 0000	R/W	32
0x80	Pause Frame Data Register 3	0x 0000 0000	R/W	32
0x84	Pause Frame Data Register 4	0x 0000 0000	R/W	32
0x88	Pause Frame Data Register 5	0x 0000 0000	R/W	32
0x90	Buffer Management Enable Register	0x 0000 0000	R/W	32
0x94	Buffer Free Register	0x 0000 0000	R/W	32
0x98	Buffer Information Register	0x 03FF 03FF	R/W	32
0x9C	Pause Information Register	0x 0000 0000	R/W	32
0xA0 to 0xAC	Reserved	—	—	—
0xF0	TX FIFO Status Register	0x 4000 0000	RO	32
0xF4	RX FIFO Status Register	0x 4000 0000	RO	32
0xF8 to 0xFC	Reserved	—	—	—

11.4.2 Detailed Register Descriptions

Interrupt Status Register
ETH[0x00] Default = 0x0000_0000 Read Only

RX Complete	RX Descriptor Error	RX Access Error	Reserved	TX Complete	TX Descriptor END	TX Access Error	Reserved	RX FIFO Overflow	TX FIFO Underflow	Reserved					
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			Link Up	MIIM Access Complete	Pause Frame Transmit	Reserved									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Note that the Interrupt Status Register (ETH[0x00]) is a clear-on-read register. The interrupt status is cleared when this register is read.

Bit 31: **RX Complete**
Indicates that a receive DMA transfer completed normally.

Bit 30: **RX Descriptor Error**
Indicates that an error in the receive descriptor table was detected.

Bit 29: **Reserved**
This bit is unused and must always be set to “0.”
[**RX Access Error**
Indicates that a bus access error occurred during a receive DMA transfer.]

Bit 28: **Reserved**

Bit 27: **TX Complete**
Indicates that frame transmission completed.

Bit 26: **TX Descriptor END**
Indicates that the end of the chaining of transmittable descriptor tables has been reached.

Bit 25: **Reserved**
This bit is unused and must always be set to “0.”
[**TX Access Error**
Indicates that a bus access error occurred during a transmit DMA transfer.]

Bit 24: **Reserved**

Bit 23: **RX FIFO Overflow**
Indicates that an overflow occurred in the receive FIFO.

Bit 22: **TX FIFO Underflow**
Indicates that an overflow occurred in the transmit FIFO.

Bits 21 to 13: **Reserved**

Bit 12: **Reserved**
This bit is invalid in the S1S65010 and must always be set to “0.”
[**Link Up**
Indicates that a linkup event occurred.]

Bit 11: **MIIM Access Complete**
Indicates that access to the MIIM Register (ETH[0x2C]) completed.

Bit 10: **Pause Frame Transmit**

Indicates that a Pause Frame transmit operation completed.

Bits 9 to 0: **Reserved**

Interrupt Enable Register

ETH[0x04] Default = 0x0000_0000 Read/Write

RX Com-plete Enable	RX Descriptor Error Enable	RX Access Error Enable	Reserved (0)	TX Com-plete Enable	TX Descriptor Error Enable	TX Access Error Enable	Reserved (0)	RX FIFO Over-flow Enable	TX FIFO Under-flow Enable	Reserved (0)					
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved (0)			Link Up Enable (0)	MIIM Access Complete Enable	Pause Frame Transmit Enable	Reserved (0)									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit 31: **RX Complete Enable**

Enables the receive complete interrupt.

Bit 30: **RX Descriptor Error Enable**

Enables the receive descriptor error interrupt.

Bit 29: **Reserved (0)**

This bit is unused and must always be set to “0.”

RX Access Error Enable

Enables the receive DMA error interrupt.

Bit 28: **Reserved (0)**

Bit 27: **TX Complete Enable**

Enables the transmit complete interrupt.

Bit 26: **TX Descriptor Error Enable**

Enables the transmit descriptor error interrupt.

Bit 25: **Reserved (0)**

This bit is unused and must always be set to “0.”

TX Access Error Enable

Enables the transmit DMA error interrupt.

Bit 24: **Reserved (0)**

Bit 23: **RX FIFO Overflow Enable**

Enables the receive FIFO overflow interrupt.

Bit 22: **TX FIFO Underflow Enable**

Enables the transmit FIFO overflow interrupt.

Bits 21 to 13: **Reserved (0)**

Bit 12: **Reserved (0)**

This bit is invalid in the S1S65010 and must always be set to “0.”

Link Up Enable

Enables the link up interrupt.

Bit 11: **MIIM Access Complete Enable**

Enables the MIIM access complete interrupt.

Bit 10: **Pause Frame Transmit Enable**

Enables the pause frame transmit complete interrupt.

Bits 9 to 0: **Reserved (0)**

Reset Register

ETH[0x08] Default = 0x0000_2000 Read/Write

All Reset	Reserved (0)														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TX Reset	RX Reset	PHY Reset	Reserved (0)												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit 31: **All Reset**

Resets all modules. This bit automatically returns to "0."

Bits 30 to 16: **Reserved (0)**

Bit 15: **TX Reset**

Resets all modules related to transmit. This bit automatically returns to "0."

Bit 14: **RX Reset**

Resets all modules related to receive. This bit automatically returns to "0."

Bit 13: **Reserved (0)**

This bit is unused in the S1S65010 and must always be set to "0."

PHY Reset

Reset signal for the external PHY device. Since this bit does not return to "0" automatically, it must be cleared to "0" before starting a communication operation.

Bits 12 to 0: **Reserved (0)**

PHY Status Register

ETH[0x0C] Default = 0x0000_0000 Read Only

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

Bits 31 to 3: **Reserved**

Bit 2: **Reserved**

This bit is invalid in the S1S65010. The value read from this bit is undefined.

Link

Indicates the status of the PHY link.

Bit 1: **Reserved**

This bit is invalid in the S1S65010. The value read from this bit is undefined.

Speed*

Indicates the communication speed.

0: 10 Mbps

1: 100 Mbps

Bit 0: **Reserved**

This bit is invalid in the S1S65010. The value read from this bit is undefined.

Duplex*

Indicates the communication mode.

0: Half duplex

1: Full duplex

Note*: These bits have no meaning when the PHY is not linked up.

DMA Command Register

ETH[0x10] Default = 0x0000_0000 Read/Write

RX DMA Enable	RX FIFO Auto Recovery	Reserved (0)													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TX DMA Start	Reserved (0)														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit 31: **RX DMA Enable**

Enables receive DMA. The DMA transfer is started automatically when data is received from the line.

Bit 30: **RX FIFO Overflow Auto-Recovery**

When this bit is set to “1”, the receive FIFO is automatically reset if the receive FIFO overflows.

Bits 29 to 16: **Reserved (0)**

Bit 15: **TX DMA Start**

Starts the transmit DMA. This bit returns to “0” when the DMA transfers specified by the descriptor have all completed.

Bits 14 to 0: **Reserved (0)**

TX DMA Pointer Register

ETH[0x18] Default = 0x0000_0000 Read/Write

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

Bits 31 to 0: **TX DMA Pointer [31:0]**

Indicates the address of the descriptor referenced by the transmit DMA. Set this register to the address of the descriptor before starting a transmit DMA operation.

Note: The set value must be on a 4-byte boundary (the lower 2 bits must be “00”).

RX DMA Pointer Register															
ETH[0x1C]	Default = 0x0000_0000														Read/Write
RX DMA Pointer [31:16]															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RX DMA Pointer [15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 31 to 0: **RX DMA Pointer [31:0]**

Indicates the address of the descriptor referenced by the receive DMA. Set this register to the address of the descriptor before starting a receive DMA operation.

Note: The set value must be on a 4-byte boundary (the lower 2 bits must be “00”).

Mode Register															
ETH[0x20]	Default = 0x4000_0000														Read/Write
Big Endian	Auto Mode	Duplex Mode	Reserved (0)		Burst Length [2:0]			Reserved (0)							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved (0)															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit 31: **Big Endian**

Sets this block to function in big endian mode.

Caution: Since the S1S65010 does not support big endian mode, this bit must be set to “0”.

Bit 30: **Reserved (0)**

This bit is unused in the S1S65010 and must always be set to “0.”

Auto Mode

The duplex setting is determined by the status signal from the PHY.

Bit 29: **Duplex Mode**

Specifies the duplex mode when Auto Mode is “0.”

0: Half duplex
1: Full duplex

Bits 28 to 27: **Reserved (0)**

Bits 26 to 24: **Burst Length [2:0]**

Specifies the DMA burst length

000: 4 beats
001: 8 beats
010: 16 beats
011: Reserved (32 beats)
100: Reserved (64 beats)
101: Reserved (128 beats)
110: Reserved
111: Reserved

Bits 23 to 0: **Reserved (0)**

TX Mode Register
ETH[0x24] Default = 0x0000_0000 Read/Write

Long Packet Enable	Short Packet Enable	No Retransmission	Late Collision Retransmission	Reserved (0)								Store and Forward	Reserv-ed (0)	Transmission Start Threshold [1:0]	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Reserved (0)		TX FIFO almost Full Threshold [1:0]		Reserv-ed (0)	TX FIFO almost Empty Threshold [2:0]			Reserved (0)							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit 31: **Long Packet Enable**
Enables transmission of frames whose length exceeds the IEEE 802.3 standard.

Bit 30: **Reserved (0)**
This bit is unused and must always be set to “0.”

Short Packet Enable
Enables transmission of frames whose length is shorter than allowed by the IEEE 802.3 standard. Since this bit is for testing only, it must always be used with its default value.

Bit 29: **Reserved (0)**
This bit is unused and must always be set to “0.”

No Retransmission
In half duplex mode, retransmission is not performed even if a collision was detected.

Bit 28: **Reserved (0)**
This bit is unused and must always be set to “0.”

Late Collision Retransmission
In half duplex mode, the frame is retransmitted after a late collision is detected.

Bits 27 to 20: **Reserved (0)**

Bit 19: **Store and Forward**
Specifies that transmissions be performed in store and forward mode.
Note: When this mode is used, the maximum MTU size (IP packet size) will be:
512 – (18 + TX-FIFO Almost Full Threshold setting value) bytes.

Bit 18: **Reserved (0)**

Bits 17 to 16: **Transmission Start Threshold [1:0]**
The transmit operation is started when more data than the amount specified by this field has been written to the TX FIFO.

00: 4 words
01: 8 words
10: 16 words
11: 32 words

Bits 15 to 14: **Reserved (0)**

Bits 13 to 12: **TX FIFO Almost Full Threshold [1:0]**
The transmit DMA operation is stopped temporarily when the free space in the TX FIFO falls below the number of words specified by this field.

00: 4 words
01: 8 words
10: 16 words
11: 32 words

Bit 11: **Reserved (0)**

Bits 10 to 8: **TX FIFO Almost Empty Threshold [2:0]**

The transmit DMA operation is restarted when the amount of data in the TX FIFO falls below the number of words specified by this field.

000: 4 words
001: 8 words
010: 16 words
011: 32 words
100: 64 words
101: Reserved (128 words)
110: Reserved (256 words)
111: Reserved (512 words)

Bits 7 to 0: **Reserved (0)**

RX Mode Register
ETH[0x28] Default = 0x0000_0000 Read/Write

Address Filtering Enable	Multicast Filtering Enable	Reserved (0)											Read Trigger Threshold [2:0]		
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved (0)		RX FIFO almost Full Threshold [1:0]		Reserved (0)		RX FIFO almost Empty Threshold [1:0]		Reserved (0)							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit 31: **Address Filtering Enable**

Specifies that filtering be performed by to the Destination MAC address.

Bit 30: **Multicast Filtering Enable**

Specifies that filtering also be performed for multicast addresses.
This bit is valid when the Address Filtering Enable bit is “1.”

Bits 29 to 19: **Reserved (0)**

Bits 18 to 16: **Read Trigger Threshold [2:0]**

The receive DMA starts operation when the amount of data in the RX FIFO exceeds the number of words specified by this field.

000: 4 words
001: 8 words
010: 16 words
011: 32 words
100: 64 words
101: Reserved (128 words)
110: Reserved (256 words)
111: Reserved (512 words)

Bits 15 to 14: **Reserved (0)**

Bits 13 to 12: **RX FIFO Almost Full Threshold [1:0]**

The receive DMA starts operation when the remaining free space in the RX FIFO falls under the number of words specified by this field.

00: 4 words
01: 8 words
10: 16 words
11: 32 words

Bits 11 to 10: **Reserved (0)**

Bits 9 to 8: **RX FIFO Almost Empty Threshold [1:0]**

Receive DMA operation is temporarily stopped when the amount of data in the RX FIFO falls below the number of words specified by this field.

00: 4 words
01: 8 words
10: 16 words
11: 32 words

Bits 7 to 0: **Reserved (0)**

MIIM Register
ETH[0x2C] Default = 0x0000_0000 Read/Write

Reserved (0)					Operation (W) / Data Valid (R)	PHY Address [4:0]					Register Address [4:0]				
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MIIM Data [15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 31 to 27: **Reserved (0)**

Bit 26: **Operation (Write mode)**

0: An MIIM read operation is started.
1: An MIIM write operation is started.

Data Valid (Read mode)

Indicates whether or not this register's value is valid.

Bits 25 to 21: **PHY Address [4:0]**

Specifies the address of the PHY device accessed by MIIM.

Bits 20 to 16: **Register Address [4:0]**

Specifies the address of the register accessed by MIIM.

Bits 15 to 0: **MIIM Data [15:0]**

The data accessed by MIIM.

MAC Address Registers 1 to 8: Lower 32 bits
ETH[0x30, 0x38, 0x40, 0x48, 0x50, 0x58, 0x60, 0x68] Default = 0x0000_0000 Read/Write

MAC Address L32 [31:16]															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MAC Address L32 [15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 31 to 0: **MAC Address L32 [31:0]**

Specifies the lower 32 bits of the destination address to be received by address filtering.

Note: Up to 8 MAC Addresses can be Registered.

MAC Address Registers 1 to 8: Upper 16 bits
ETH[0x34, 0x3C, 0x44, 0x4C, 0x54, 0x5C, 0x64, 0x6C] Default = 0x0000_0000 Read/Write

Reserved (0)															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MAC Address U16 [15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 31 to 16: **Reserved**

(When read out, the value read will mirror the lower 16 bits.)

Bits 15 to 0: **MAC Address U16 [15:0]**

Specifies the upper 16 bits of the destination address to be received by address filtering.

Note: Up to 8 MAC Addresses can be Registered.

Flow Control Register
ETH[0x70] Default = 0x0000_0000 Read/Write

Flow Control Enable	Reserved (0)														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved (0)															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit 31: **Flow Control Enable**

Enables the transfer pause function based on pause frame receive.

Bits 30 to 0: **Reserved (0)**

Pause Request Register
ETH[0x74] Default = 0x0000_0000 Read/Write

Pause Frame Request	Reserved (0)														
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved (0)															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit 31: **Pause Frame Request**

Transmits a pause frame. This bit returns to “0” automatically after the pause frame transmission.

Bits 30 to 0: **Reserved (0)**

Pause Frame Data Registers 1 to 5
ETH[0x78, 0x7C, 0x80, 0x84, 0x88] Default = 0x0000_0000 Read/Write

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

Bits 31 to 0: **Pause Frame Data [31:0]**

Holds the pause frame data.

Buffer Management Enable Register ETH[0x90] Default = 0x0000_0000															Read/Write
Reserved (0)															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved (0)															Buffer Manage-Ment Enable
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 31 to 1: **Reserved**

Bit 0: **Buffer Management Enable**

Enables the receive buffer management function.

Buffer Free Register ETH[0x94] Default = 0x0000_0000															Read/Write
Reserved (0)															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved (0)															Buffer Free
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 31 to 1: **Reserved (0)**

Bit 0: **Buffer Free**

Reports that one frame of receive buffer has been freed.

When a “1” is written to this bit by the CPU, the block internal buffer remaining capacity counter is incremented and the bit is automatically returned to “0.”

Buffer Information Register ETH[0x98] Default = 0x03FF_03FF															Read/Write
Reserved (0)						Ability [9:0] (Read only)									
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved (0)						Capacity [9:0]									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 31 to 26: **Reserved (0)**

Bits 25 to 16: **Ability [9:0] (Read only)**

Indicates the amount of free space in the receive buffer.

Bits 15 to 10: **Reserved (0)**

Bits 9 to 0: **Capacity [9:0]**

Indicates the total receive buffer capacity.

Pause Information Register															
ETH[0x9C]		Default = 0x0000_0000													Read/Write
Pause Time [15:0]															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved (0)						Pause Transmission Threshold [9:0]									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 31 to 16: **Pause Time [15:0]**
Indicates the pause time for the pause frame.

Bits 15 to 10: **Reserved (0)**

Bits 9 to 0: **Pause Transmission Threshold [9:0]**
Indicates the threshold for receive buffer capacity at which a pause frame is transmitted.

TXFIFO Status Register															
ETH[0xF0]		Default = 0x4000_0000													Read Only
Almost Full	Almost Empty	TX FIFO Status			Frame Count			Reserved							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit 31: **Almost Full**
Indicates that the transmit FIFO is in the almost full state.

Bit 30: **Almost Empty**
Indicates that the transmit FIFO is in the almost empty state.

Bits 29 to 27: **TX FIFO Status**
Indicates the transmit FIFO status.

100: ACC NEW FR
Indicates that a new frame can be accepted.

101: WRITE ENABLE
Indicates that the current frame is being written.

110: CMPLT
Indicates that acquisition of a single frame has completed.

111: FULL
Indicates that the transmit FIFO is full

0xx: STOP
Indicates that the transmit FIFO is stopped (for, e.g., initialization).

Bits 26 to 24: **Frame Count**
Indicates the number of frames current in the transmit FIFO.

Bits 23 to 0: **Reserved**

RXFIFO Status Register
ETH[0xF4] Default = 0x4000_0000 Read Only

Almost Full	Almost Empty	Read Trigger	Receiving	Stored Words [11:0]											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit 31: **Almost Full**
Indicates that the receive FIFO is in the almost full state.

Bit 30: **Almost Empty**
Indicates that the receive FIFO is in the almost empty state.

Bit 29: **Read Trigger**
Indicates that the amount of data present in the receive FIFO is at or above the Read Trigger Threshold.

Bit 28: **Receiving**
Indicates that there is a valid frame in the receive FIFO.

Bits 27 to 16: **Stored Words**
Indicates the number of data present in the receive FIFO in words.

Bits 15 to 0: **Reserved**

11.5 Description of Operation

11.5.1 MAC Functions

11.5.1.1 Transmit

This block temporarily stores, in a FIFO, the data written by the ARM AHB bus via the DMA controller. This block then automatically removes that data from the FIFO, assembles it into frames stipulated by the IEEE 802.3 standard, and outputs that data to the PHY over the MII. The main features of this transmission function are as follows.

- Converts a 32-bit data stream to a 4-bit (nibble) format.
- Adds the preamble, calculates and adds the FCS (CRC).
- Supports retransmit on collision (up to 15 times) in half-duplex mode.
- Transmits a JAM signal when a transmission error occurs.
- Error handling based on the transmission byte count error (A JAM signal is transmitted for fewer than 64 bytes or 1519 bytes or over.)

In addition to the FIFO memory used to accumulate the transmit data, this block also includes a transmission result storage buffer. This transmission result storage buffer is provide to report to the user information such as whether or not frame transmission is proceeding normally.

11.5.1.2 Receive

This block analyzes the received data stream into the frame structure stipulated by the IEEE 802.3 standard, and temporarily accumulates the extracted receive frame data in a FIFO memory. The receive data accumulated in the FIFO memory can be read out by the user from the ARM AHB bus over the DMA controller. The main features of this receive function are as follows.

- The 4-bit (nibble) data stream is converted into a 32-bit wide stream.
- FCS (CRC) calculation
- Fragmental frames (frames with less than 64 bytes) are discarded automatically.
- The received byte count is confirmed.

This block can receive frames that exceed the maximum frame length (1518 bytes) stipulated by the IEEE 802.3 standard. If a frame that exceeds the maximum frame length is received, this block sets the Too Long flag to “1”, but it performs the FCS calculation and other operations for the content of the received data normally.

11.5.1.3 Flow Control

This block can be set to transmit a pause frame by writing “1” to the Pause Frame Request bit (bit 31 in the Pause Request Register ETH[0x74]). Pause frames can only be transmitted in full-duplex mode (when bit 29 in the Mode Register ETH[0x20] is “1”). Writes to the Pause Frame Request bit are ignored in half-duplex mode.

When this block receives a pause frame transmission request signal, it sends a pause frame immediately after the frame currently being transmitted completes. It inserts the content set in the Pause Frame Data Registers 1 to 5 (ETH[0x78, 0x7C, 0x80, 0x84, 0x88]) as the Destination Address, Source Address, Type, Opcode, and Pause Time fields in the pause frame. It automatically inserts zeros in the data block, adds the result of an FCS calculation, and transmits the result as the pause frame.

When the pause frame transmission completes, this block returns the Pause Frame Request bit to “0” and issues a Pause Frame Transmission interrupt.

When the Flow Control Enable bit (bit 31 in the Flow Control Register ETH[0x70]) is set to “1”, this block can implement transmission stop function based a pause frame receive. However, when the Flow Control Enable bit is “0”, this block will not stop the transmission operation when a pause frame is received.

If this block receives a pause frame, after the current frame being transmitted completes, it will stop transmission operation for the pause time indicated by the pause frame. The time for which transmission will be stopped will be the product of the slot time (the time for 512 bits: 5.12 μs at 100 Mbps or 51.2 μs at 10 Mbps) and the value indicated by the Pause Time field in the pause frame.

If another pause frame is received when this block has received a pause frame and has stopped a transmission operation, the timer internal to this block that is counting the transmission stop time will be updated to the value of the pause time indicated by the new pause frame. This allows the remote communicating system/terminal to perform operations such as releasing the pause or extending the time for which transmission is paused.

This block can be set to transmit a pause frame by setting the Pause Frame Request bit to “1” even during periods when the transmission operation is stopped due to the receive of a pause frame.

11.5.2 DMA Controller

11.5.2.1 Overview

This block includes a DMA controller that is directly connected to the transmit/receive Ethernet MAC units. Frame management is performed using descriptor tables, allowing high-efficiency transfer control that reduces the overhead on the system CPU load.

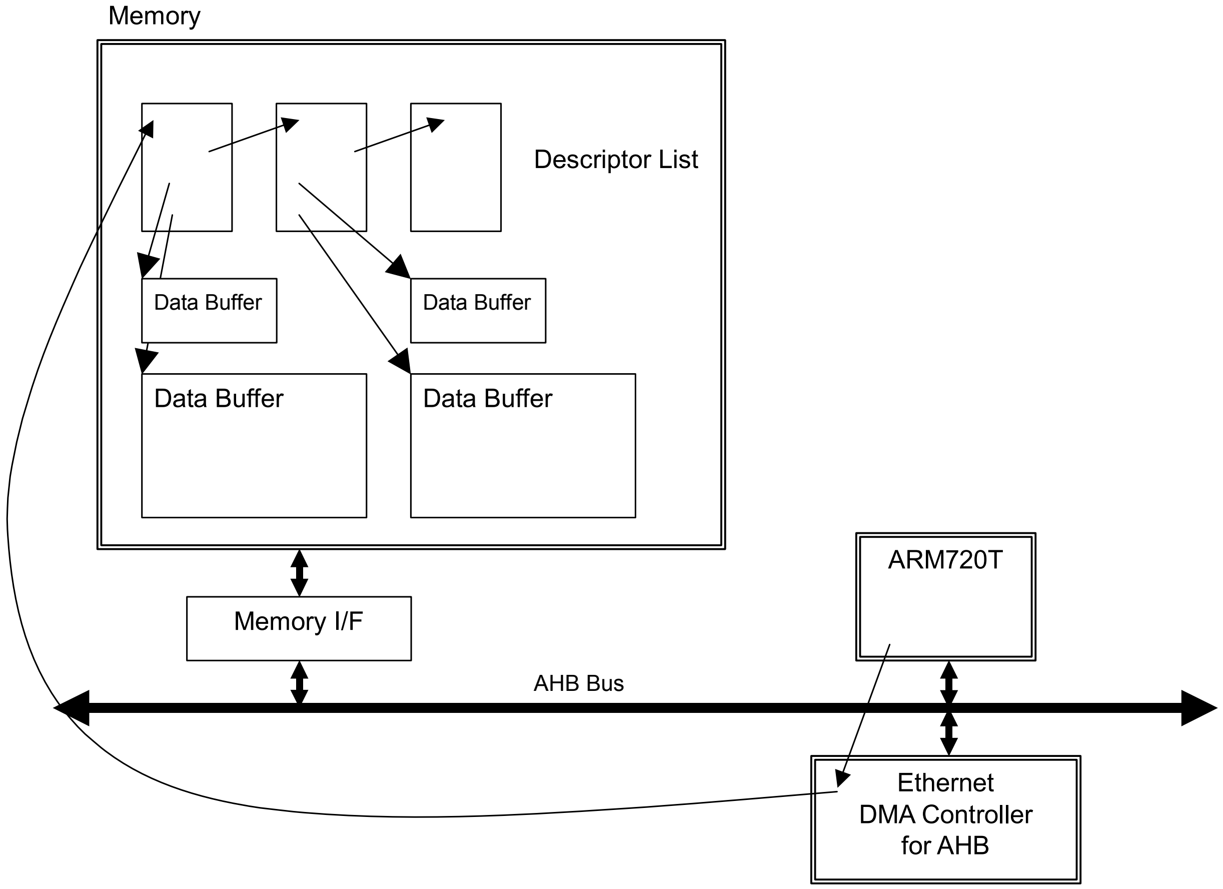


Fig.11.2 DMA Controller and Descriptor Architecture

The descriptor tables are in a one-to-one relationship with the transmit/receive frames. Multiple buffer areas can be specified from a single descriptor table, and data distributed within the memory space can be assembled into a single frame or a received frame can be stored distributed in multiple areas.

11.5.2.2 Descriptor Tables

Table 11.2 Transmit Descriptor Table

Offset Address	Name
0x00	TX Command / Status
0x04	TX Next Descriptor Pointer
0x08	TX Buffer Address 1st
0x0C	TX Buffer Size 1st
0x10	TX Buffer Address 2nd
0x14	TX Buffer Size 2nd
…	… (The remainder consists of repeated buffer address and size pairs.)

Table 11.3 Receive Descriptor Table

Offset Address	Name
0x00	RX Command / Status
0x04	RX Next Descriptor Pointer
0x08	RX Buffer Address 1st
0x0C	RX Buffer Size 1st
0x10	RX Buffer Address 2nd
0x14	RX Buffer Size 2nd
…	… (The remainder consists of repeated buffer address and size pairs.)

Transmit Descriptor Table

TX Command / Status
Offset Address = 0x00 Read/Write

Complete	Abort	Reserved	Usable	Carrier Sense Error	Too Short	Too Long	Under-Flow	Retry Count [3:0]				Late Collision	Excessive Collision	Multiple Collision	Single Collision
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Including VLAN Tag	Including CRC	Auto Padding	TX Octets [12:0]												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

When preparation of the transmit data has been completed, set bit 28 (Usable) to “1” and set bits 15 to 0 to an appropriate value. Clear all other bits to “0.”

Bit 31: **Complete**
Indicates that the transmit completed normally.

Bit 30: **Abort**
Indicates that it became impossible to continue the transmission due to some problem during that transmission.

Bit 29: **Reserved**

Bit 28: **Usable**
Indicates that the data specified by this descriptor is in the ready to transmit state. The user must set this bit to “1” after preparation of the transmit data has been completed. This block will clear this bit to “0” after the DMA transfer has completed.

Bit 27: **Carrier Sense Error**
Indicates that a carrier sense error occurred during the transmission.

Bit 26: **Too Short**
Indicates that a JAM was issued and the transmission was terminated due to an attempt to transmit a frame shorter than 64 bytes.

Bit 25: **Too Long**
Indicates that the transmission was terminated due to an attempt to transmit a frame longer than 1518 bytes.

Bit 24: **Underflow**
Indicates that a TX FIFO underflow occurred and the transmission was terminated.

Bits 23 to 20: **Retry Count [3:0]**
Indicates the number of retries due to collisions.

Bit 19: **Late Collision**
Indicates that a late collision occurred and the transmission was terminated.

Bit 18: **Excessive Collision**
Indicates that it was not possible to transmit the frame due to collisions, even though the transmission was retried 15 times.

Bit 17: **Multiple Collision**
Indicates that the transmission completed normally after multiple collisions were detected.

Bit 16: **Single Collision**
Indicates that the transmission completed normally after a single collision was detected.

Bit 15: **Including VLAN Tag**

Indicates that this is a frame to which a VLAN tag has been added. The maximum transmit frame length is expanded by 4 bytes when this bit is "1."

Bit 14: **Including CRC**

Indicates that FCS (CRC) has already been added to the frame data indicated by this descriptor. This block will not add FCS data when this bit is "1."

Bit 13: **Auto Padding**

Padding bits are added automatically when the transmit frame data length is less than 64 bytes.

Bits 12 to 0: **TX Octets [12:0]**

Specifies a value one less than the number of bytes in the transmit frame. The FCS (4 bytes) automatically added by this block are not included.

TX Next Descriptor Pointer															
Offset Address = 0x04															Read/Write
TX Next Descriptor Pointer [31:16]															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TX Next Descriptor Pointer [15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 31 to 0: **TX Next Descriptor Pointer [31:0]**

Specifies the start address of the next transmit descriptor.

TX Buffer Address 1st															
Offset Address = 0x08															Read/Write
TX Buffer Address 1st [31:16]															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TX Buffer Address 1st [15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 31 to 0: **TX Buffer Address 1st [31:0]**

Specifies the start address of the first buffer area that holds the transmit frame. The specified address must be set on a 4-byte boundary (the lower 2 bits must be "00"). (This block ignores the lower 2 bits.)

TX Buffer Size 1st															
Offset Address = 0x0C															Read/Write
Reserved															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			TX Buffer Size 1st [12:0]												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 31 to 13: **Reserved (0)**

Bits 12 to 0: **TX Buffer Size 1st [12:0]**

Specifies the size of the first buffer area that holds the transmit frame as a value one less than the byte count.

TX Buffer Address nth															
Offset Address = 0x10, 0x18, 0x20, 0x28…															Read/Write
TX Buffer Address nth [31:16]															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TX Buffer Address nth [15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 31 to 0: **TX Buffer Address nth [31:0]**

Specifies the start address of the nth buffer area that holds the transmit frame. The specified address must be set on a 4-byte boundary (the lower 2 bits must be "00"). (This block ignores the lower 2 bits.)

TX Buffer Size n^{th}
Offset Address = 0x14, 0x1C, 0x24, 0x2C… Read/Write

Reserved															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			TX Buffer Size n^{th} [12:0]												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 31 to 13: **Reserved (0)**

Bits 12 to 0: **TX Buffer Size n^{th} [12:0]**
Specifies the size of the nth buffer area that holds the transmit frame as a value one less than the byte count.

Receive Descriptor Table

RX Command / Status
Offset Address = 0x00 Read/Write

Received	Reserved		Usable	Reserved			Broadcast Frame	Multicast Frame	Individual Frame	Address Not Match	Too Long	Too Short	Not Octal	Nibble Error	CRC Error
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			RX Octets [12:0]												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Set bit 28 (Usable) to “1” and clear all the other bits to “0” to prepare receiving the next frame after processing the receive data in the buffer area has completed.

Bit 31: **Received**
Indicates that a frame was received and that the DMA transfer completed. This block sets this bit to “1” automatically.

Bits 30 to 29: **Reserved (0)**

Bit 28: **Usable**
Indicates that the data area specified by this descriptor is in the receive ready state. After the user has completed processing the received data in the buffer area, the user must set this bit to “1” to prepare for receive of the next frame. This block clears this bit to “0” after the DMA transfer completes.

Bits 27 to 25: **Reserved (0)**

Bit 24: **Broadcast Frame**
Indicates that the destination address was a broadcast address.

Bit 23: **Multicast Frame**
Indicates that the destination address was a multicast address.

Bit 22: **Individual Frame**
Indicates that the destination address was an address registered with the Address Recognizer.

Bit 21: **Address Not Match**
Indicates that the destination address was neither a broadcast, multicast, or individual address.

Bit 20: **Too Long**
Indicates that the frame exceeded 1518 bytes.

Bit 19: **Too Short**
Indicates that the frame was shorter than 64 bytes.

Bit 18: **Not Octal**

Indicates that the received frame data length was not a multiple of 8 bits.

Bit 17: **Nibble Error**

Indicates that a transmission coding error occurred during frame receive.

Bit 16: **CRC Error**

Indicates that the received frame's FCS was incorrect.

Bits 15 to 13: **Reserved (0)**

Bits 12 to 0: **RX Octets [12:0]**

Indicates a value one less than the byte count of the received frame. This byte count specifies the number of bytes from the destination address to the end of the FCS excluding the preamble and the SFD.

RX Next Descriptor Pointer
Offset Address = 0x04 Read/Write

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

Bits 31 to 0: **RX Next Descriptor Pointer [31:0]**

Specifies the start address of the next receive descriptor.

RX Buffer Address 1st
Offset Address = 0x08 Read/Write

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

Bits 31 to 0: **RX Buffer Address 1st [31:0]**

Specifies the head address of the first buffer that holds the received frame. The specified address must be set on a 4-byte boundary (the lower 2 bits must be "00"). (This block ignores the lower 2 bits.)

RX Buffer Size 1st
Offset Address = 0x0C Read/Write

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

Bits 31 to 13: **Reserved (0)**

Bits 12 to 0: **RX buffer Size 1st [12:0]**

Specifies the size of the first buffer area that holds the received frame as a value one less than the byte count. The specified size must be set on a 4-byte boundary (the lower 2 bits must be "11"). (This block ignores the lower 2 bits.)

RX Buffer Address nth
Offset Address = 0x10, 0x18, 0x20, 0x28… Read/Write

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

Bits 31 to 0: **RX Buffer Address nth [31:0]**

Specifies the head address of the nth buffer area that holds the received frame. The specified address must be set on a 4-byte boundary (the lower 2 bits must be “00”). (This block ignores the lower 2 bits.)

RX Buffer Size nth
Offset Address = 0x14, 0x1C, 0x24, 0x2C… Read/Write

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

Bits 31 to 13: **Reserved (0)**

Bits 12 to 0: **RX Buffer Size nth [12:0]**

Specifies the size of the nth buffer area that holds the received frame as a value one less than the byte count. The specified size must be set on a 4-byte boundary (the lower 2 bits must be “11”). (This block ignores the lower 2 bits.)

11.5.2.3 Transmit DMA Description of Operation

The transmit DMA is started, thus starting a frame transmission, by writing “1” to the TX DMA Start bit in the DMA Command Register (ETH[0x10]). When “1” is written to the TX DMA Start bit, a single descriptor table is read from the memory area pointed to by the TX DMA Pointer Register (ETH[0x18]). If the Usable bit in the read descriptor table is “1,” data readout from the buffer area starts according to the information in the descriptor table. When readout of the number of bytes of data indicated by the transfer byte count (TX Octets) specified in the descriptor table completes, the Usable bit is cleared to “0” and the next descriptor table is read out from the memory area specified by the TX Next Descriptor Pointer. Since the Next Descriptor Pointer is also loaded into the TX DMA Pointer Register, the CPU can find out the location of the descriptor for the current DMA transfer by reading the TX DMA Pointer Register. If the Usable bit in a read descriptor table is “0,” this block clears the TX DMA Start bit to “0” and stops the transmit DMA operation.

The descriptor table start addresses (TX DMA Pointer and TX Next Descriptor Pointer) must be specified to be addresses on a 4-byte boundary (the lower 2 bits must be “00”).

Multiple buffer areas can be specified in a single descriptor table. (This is called a multibuffer structure.) Data distributed in memory can be handled as a single transfer frame by using this multibuffer structure. For example, separate memory areas can be provided for the MAC header, the IP header, and the IP payload, and a frame can be formed by combining those. Also, a buffer can be used as a single buffer by specifying a value greater than TX Octets for the buffer size for the buffer specified as the first buffer (Buffer Size 1st). The transmit DMA block handles buffer addresses on 4-byte (32-bit) boundaries by ignoring the lower 2 bits. The buffer size can be specified in byte units, however.

Transmit DMA transfers are performed by writing data to the transmit FIFO in this block. Frame generation for the data written to the transmit FIFO is performed by this block’s MAC function. When the transmission operation for a single frame completes, the transmission result is reflected in the TX Command/Status field in the corresponding descriptor table and an interrupt is issued to the CPU. This interrupt can be masked using the Interrupt Enable Register (ETH[0x04]).

The transmit DMA controller performs flow control by monitoring the status of the transmit FIFO. When it receives an Almost Full signal from the transmit FIFO during a DMA transfer, it interrupts the DMA transfer and restarts the DMA transfer when it receives the Almost Empty signal. The Almost Full and Almost Empty signals can be set up with the TX FIFO Almost Full Threshold and the TX FIFO Almost Empty Threshold fields in the TX Mode Register (ETH[0x24]).

[EU1]

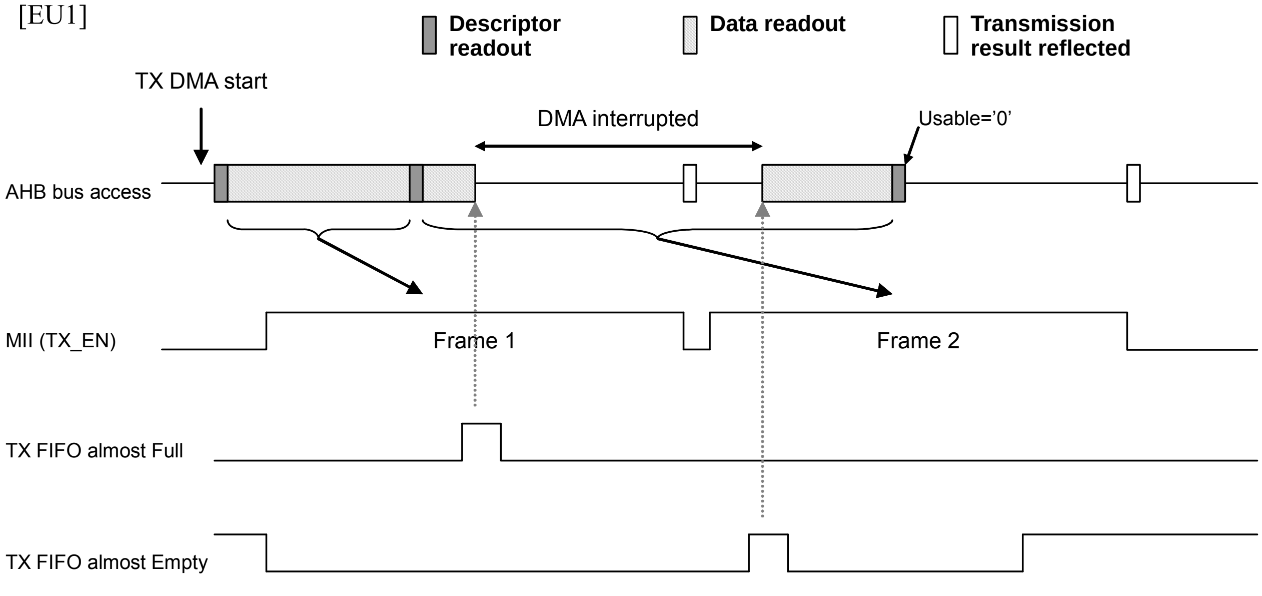


Fig.11.3 Behavior of Transmission

11.5.2.4 Receive DMA Description of Operation

The receive DMA function is set to the ready to receive (enabled) state by writing “1” to the RX DMA Enable bit in the DMA Control Register (ETH[0x10]). Even if enabled, the receive DMA function performs no operation if there is no frame to receive.

When this block receives a frame from the MII, it reads out a single descriptor table from the memory area pointed to by the RX DMA Pointer Register (ETH[0x1C]). If the Usable bit in the read descriptor table is “1,” the received data is written to the buffer area according to the information in the descriptor table. When all of the received data for a single frame has been written, the received result is reflected in the RX Command/Status field in the descriptor table and the Usable bit is cleared to “0.” When a receive DMA transfer for one frame completes, the RX DMA Pointer Register is updated to the value of the RX Next Descriptor Pointer and the circuit prepares for the next receive frame DMA transfer. The CPU can find out the location of the descriptor for the current DMA transfer by reading the RX DMA Pointer Register. If the Usable bit in the read descriptor table is “0,” an RX Descriptor Error interrupt is issued and the receive DMA operation is stopped. If an RX Descriptor Error interrupt occurs, the application should temporarily clear the RX DMA Enable bit to “0,” set up the descriptor table and the RX DMA Pointer again, and then set the RX DMA Enable bit back to “1.”

The descriptor table start addresses (RX DMA Pointer and RX Next Descriptor Pointer) must be specified to be addresses on a 4-byte boundary (the lower 2 bits must be “00”).

Multiple buffer areas can be specified in a single descriptor table. (This is called a multibuffer structure.) Applications can use this multibuffer structure to distribute the data in a received frame across different areas in memory. Also, a buffer can be used as a single buffer by specifying a value greater than 1518 bytes (the maximum Ethernet frame length) for the buffer size for the buffer specified as the first buffer (Buffer Size 1st). The receive DMA block handles buffer addresses and buffer sizes on 4-byte boundaries by ignoring the lower 2 bits.

The receive DMA controller performs flow control by monitoring the status of the receive FIFO. The receive DMA controller starts a DMA transfer when either receiving of one frame completes or when it receives a Read Trigger signal. If an Almost Empty signal is received from the FIFO during a DMA transfer, that DMA transfer is interrupted and only restarted when either receiving of one frame completes or when a Read Trigger signal is received. The Read Trigger and Almost Empty signals can be set up by setting the Read Trigger Threshold and RX FIFO Almost Empty Threshold fields in the RX Mode Register (ETH[0x28]).

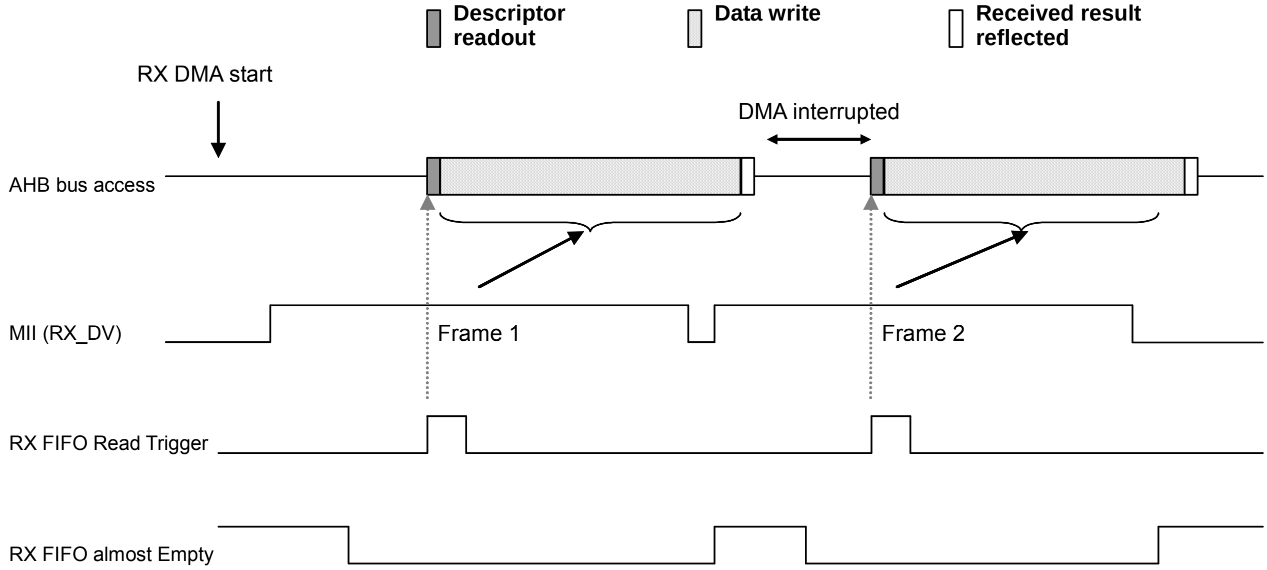


Fig.11.4 Behavior of Reception

11.5.2.5 DMA and MAC Operating Mode Settings

As described in section 11.5.2.3 and 11.5.2.4, DMA operation is intimately connected to the MAC operating mode. Keep the following points in mind when setting the DMA and MAC operating modes.

Burst Length: Mode register ETH[0x20] bits [26:24]

Sets the AHB Burst Length. Since the maximum burst length in AMBA 2.0 is 16 beats (INCR16), bursts of 32 beats or longer are handled by consecutively iterating INCR16 bursts multiple times.

Transmission Start Threshold: TX mode register ETH[0x24] bits [17:16]

This setting is ignored in Store and Forward mode.
In modes other than Store and Forward mode, frames smaller than the value specified by this settings remain stored in the FIFO until the next transmit frame is written. (For example, if this field is set to “11” (32 words), transmission will not be started for frames less than 128 bytes until the next transmit data is written, that is until more than 32 words have been written to the FIFO.
We recommend that this parameter be set to as small a value as possible so that TX FIFO Underflow errors do not occur. Also, you can reliably assure that all the frames have been transmitted by changing this parameter to a small value after the last frame has been written.

TX FIFO Almost Full Threshold: TX mode register ETH[0x24] bits[13:12]

This parameter must be set to a value larger than the Burst Length. (The value “11” (32 words) may be used if the burst length is 32 beats or over.) TX FIFO overflow errors may occur if this parameter is smaller than the burst length.

TX FIFO Almost Empty Threshold: TX mode register ETH[0x24] bits [10:8]

In Store and Forward mode, set this parameter to a value larger than the Maximum Frame Length. In the other modes, set it to a value larger than the Transmission Start Threshold.
If the value of this parameter is inappropriate, the operation may stop with the DMA in the wait state.

Read Trigger Threshold: RX mode register ETH[0x28] bits [18:16]

This parameter must be set to a value larger than both the Burst Length and the RX FIFO Almost Empty Threshold.

RX FIFO Almost Full Threshold: RX mode register ETH[0x28] bits [13:12]

If a delay occurs in the AHB response, set this parameter to as large a value as possible.

RX FIFO Almost Empty Threshold: RX mode register ETH[0x28] bits [9:8]

This parameter must be set to a value larger than the Burst Length. (The value “11” (32 words) may be used if the Burst Length is 32 beats or over.)

Recommended Setting Values

Burst Length	4	8	16
TX Start Threshold	16	16	16
TX FIFO Almost Full Threshold	8	16	32
TX FIFO Almost Empty Threshold	32	32	32
RX Read Trigger Threshold	16	32	64
RX FIFO Almost Full Threshold	32	32	32
RX FIFO Almost Empty Threshold	8	16	32

11.5.3 Address Filter

This block's address filtering function is enabled by setting the Address Filtering Enable bit (RX Mode Register ETH[0x28] bit 31) to "1." When the address filtering function is enabled, this block will only acquire received frames whose destination address field matches broadcast, multicast or the address registered in the Address Register. All other received frames are automatically discarded.

Additionally, the multicast filtering function will be enabled if the Multicast Filtering Enable bit (RX Mode Register ETH[0x28] bit 30) is set to "1" at the same time the Address Filtering Enable bit is. When the multicast filtering function is enabled, this block will only acquire received frames whose destination address field matches multicast or the address registered in the Address Register. All other received frames are automatically discarded. However, note that received frames that have a multicast address (01-80-C2-00-00-01) that is specified as a pause frame destination address are acquired unconditionally.

11.5.4 MIIM

11.5.4.1 Write Operation

Use the MIIM write operation provided by this block when you want to write to the MIIM Register inside PHY. When "1" is written to bit 26 in the MIIM Register (ETH[0x2C]), this block's MIIM function interprets that as a write operation, and performs a write operation to the MIIM register inside PHY over the MII management bus. The PHY address and register address output at this time are specified by bits [25:21] and bits [20:16] in the MIIM Register.

An MIIM access complete interrupt is generated when the MII management bus write operation completes and the write to the PHY Register has completed.

11.5.4.2 Read Operation

Use the MIIM read operation provided by this block when you want to read the MIIM Register inside PHY. When "0" is written to bit 26 in the MIIM Register, this block's MIIM function interprets that as a read operation, and performs a read operation on the MIIM Register inside PHY over the MII management bus. The PHY address and register address output at this time are specified by bits [25:21] and bits [20:16] in the MIIM Register. This block acquires the contents of the MIIM Register inside PHY over the MII management bus and reflects that content in the MIIM Register. When the read operation completes, this block sets bit 26 in the MIIM Register to "1" to indicate that bits [25:0] in the MIIM Register are valid.

An MIIM access complete interrupt is generated when the MII management bus read operation completes and the read from the PHY Register has completed.

11.5.5 Receive Buffer Management Function

The receive buffer management function manages the state of the receive buffer formed in this block's external memory and automatically transmits a pause frame when the receive buffer free space decreases below the amount specified by a certain register. This function is enabled by setting the Buffer Management Enable bit (bit 31 in the Buffer Management Enable Register, ETH[0x90]) to "1."

When the Buffer Management Enable bit is "0," this block loads the contents of the Capacity Register into an internal counter as the initial value for the receive buffer capacity. Therefore, the internal counter is initialized to Capacity if this function is disabled (if Buffer Management Enable is "0").

When this function is enabled (when Buffer Management Enable is "1") the internal counter counts the receive buffer free space. When a receive DMA transfer of 1 frame of data completes, the counter is decremented, and when "1" is written to the receive buffer release notification register (Buffer Free), the count is incremented. The current receive buffer free space can be checked by reading the Ability bits (Buffer Information Register ETH[0x98] bits[25:16]).

If the receive buffer free space becomes less that the value specified by the Pause Transmission Threshold bits (Pause Information Register ETH[0x9C] bits[9:0]), a pause frame is transmitted automatically. After the pause frame is transmitted, the function waits for the time set in the Pause Time register, and then checks the receive buffer free space again. Thus this function does not transmit any unnecessary pause frames.

11.6 Limitations on the Use of the Ethernet MAC and E-DMA (ETH)

There are limitations on the use of certain registers in this Ethernet unit in the S1S65010. The table below lists the limitations on register usage in this chip.

Offset Address	Register Bit Name	Limitation
ETH[0x00] bit 29	RX Access Error	May not be used
ETH[0x00] bit 25	TX Access Error	May not be used
ETH[0x00] bit 12	Link Up	May not be used
ETH[0x04] bit 29	RX Access Error Enable	May not be used
ETH[0x04] bit 25	TX Access Error Enable	May not be used
ETH[0x04] bit 12	Link Up Enable	May not be used
ETH[0x08] bit 13	PHY Reset	May not be used
ETH[0x0C] bit 2	Link	May not be used
ETH[0x0C] bit 1	Speed	May not be used
ETH[0x0C] bit 0	Duplex	May not be used
ETH[0x20] bit 30	Auto Mode	May not be used
ETH[0x20] bits [26:24]	Burst Length	Values in the range 011 to 111 may not be used.
ETH[0x24] bit 30	Short Packet Enable	May not be used
ETH[0x24] bit 29	No Retransmission	May not be used
ETH[0x24] bit 28	No Collision Retransmission	May not be used
ETH[0x24] bits [10:8]	TXFIFO almost Empty Threshold [2:0]	Values in the range 101 to 111 may not be used.
ETH[0x28] bits [18:16]	Read Trigger Threshold [2:0]	Values in the range 101 to 111 may not be used.

12. APB BRIDGE (APB)

12.1 Overview

This AHB bus slave acts as a bridge between the internal high-speed bus (AHB1) and low-speed devices connected to the APB bus (hereinafter abbreviated to “APB devices”). It provides AHB bus control on their behalf, freeing them from AHB bus control considerations, and thus allowing them to concentrate merely on the simpler APB bus control.

The programmer normally need not worry about this block and can just leave the settings* at their defaults in effect immediately after a reset. The minimum access time (initial setting) is two APB bus cycles, but there are also wait cycle settings (0 to 3 cycles) for individual APB devices.

(* APBWAIT0 is a major exception here.)

- Supports bus access for all bus width (8, 16, and 32-bit) types
- Supports 2-cycle bus operation (initial settings) for all APB devices
- Selectable wait cycles (0 to 3 wait states) for each APB device
- Supports appropriate byte lane manipulations for byte and half-word (16-bit) access operations
- Generates the basic APB timing
- Generates a PSEL internal signal for all APB devices
- Supplies the HBE and LBE internal signals as byte lane active signals

12.2 Block Diagram

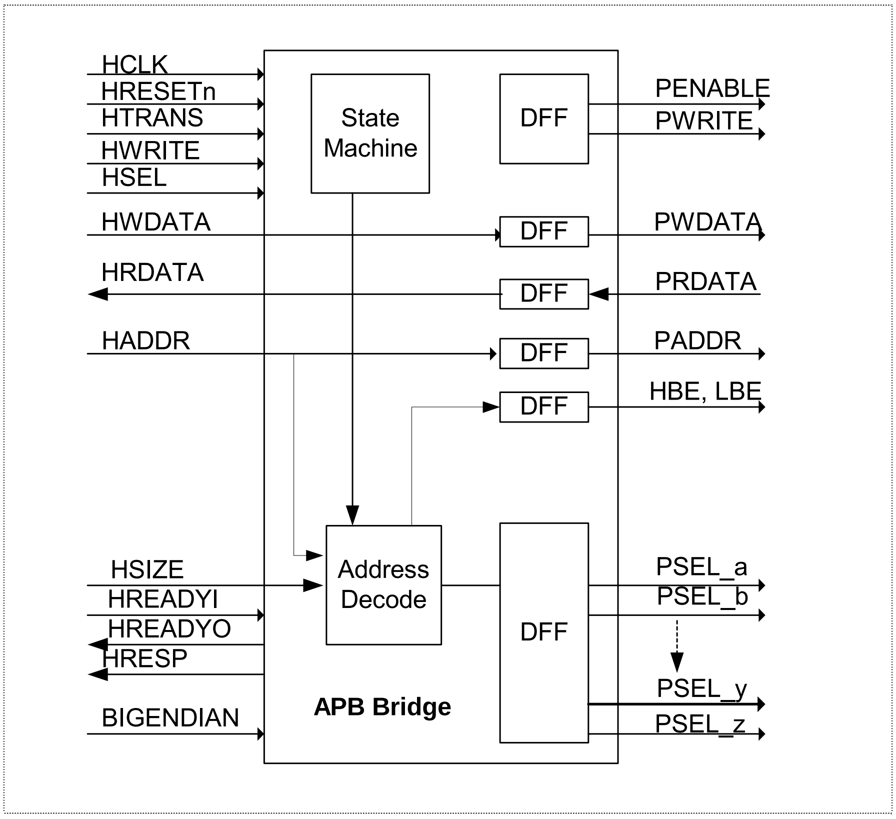


Fig.12.1 APB Bridge Block Diagram

12.3 External Pins

This block interacts with no external pins.

12.4 Registers

12.4.1 Register List

The base address for these registers is 0xFFFE_0000. The offsets in the following Table are relative to that address.

Table 12.1 APB Register List (Base Address: 0xFFFE_0000)

Address Offset	Register Name	Abbreviation	Default Value	R/W	Data Access Size (Bits)
0x00	APB WAIT0 Register	APBWAIT0	0x0040_0500	R/W	32
0x04	APB WAIT1 Register	APBWAIT1	0x0000_0000	R/W	32

12.4.2 Detailed Register Descriptions

APB WAIT0 Register (APBWAIT0)															
APB[0x00]		Default = 0x0040_0500												Read/Write	
PW0FCNF [1:0]		PW0ECNF [1:0]		PW0DCNF [1:0]		PW0CCNF [1:0]		PW0BCNF [1:0]		PW0ACNF [1:0]		PW09CNF [1:0]		PW08CNF [1:0]	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PW07CNF [1:0]		PW06CNF [1:0]		PW05CNF [1:0]		PW04CNF [1:0]		PW03CNF [1:0]		PW02CNF [1:0]		PW01CNF [1:0]		PW00CNF [1:0]	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 31 to 0: **PWxCNF[1:0] (x=00 to 0F)**

00: 0 wait, Basic 2-APB cycle (default)
01: 1 wait, 2-APB cycle + 1-wait cycle = 3-APB cycle
10: 2 wait, 2-APB cycle + 2-wait cycle = 4-APB cycle
11: 3 wait, 2-APB cycle + 3-wait cycle = 5-APB cycle

Note: To use this register, change the default value to “0x0050_0500.”

APB WAIT1 Register (APBWAIT1)															
APB[0x04]		Default = 0x0000_0000												Read/Write	
PW1FCNF [1:0]		PW1ECNF [1:0]		PW1DCNF [1:0]		PW1CCNF [1:0]		PW1BCNF [1:0]		PW1ACNF [1:0]		PW19CNF [1:0]		PW18CNF [1:0]	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PW17CNF [1:0]		PW16CNF [1:0]		PW15CNF [1:0]		PW14CNF [1:0]		PW13CNF [1:0]		PW12CNF [1:0]		PW11CNF [1:0]		PW10CNF [1:0]	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 31 to 0: **PWxCNF[1:0] (x=10 to 1F)**

00: 0 wait, Basic 2-APB cycle (default)
01: 1 wait, 2-APB cycle + 1-wait cycle = 3-APB cycle
10: 2 wait, 2-APB cycle + 2-wait cycle = 4-APB cycle
11: 3 wait, 2-APB cycle + 3-wait cycle = 5-APB cycle

Table 12.2 APB Device Mappings to APBWAIT[1:0] Registers

APBWAIT0		APBWAIT1	
PWxCNF	**APB Device**	**PWxCNF**	**APB Device**
PW00CNF	APB bridge	PW10CNF	Reserved
PW01CNF	Reserved	PW11CNF	GPIO
PW02CNF	Ethernet Mac	PW12CNF	Serial peripheral interface (SPI)
PW03CNF	DMA controller 1 (DMAC1)	PW13CNF	Reserved
PW04CNF	CF card attribute and common memory space	PW14CNF	Reserved
PW05CNF	CF card I/O space	PW15CNF	UART
PW06CNF	Compact Flash card setting	PW16CNF	UART Lite (UARTL)
PW07CNF	Reserved	PW17CNF	Reserved
PW08CNF	Camera interface	PW18CNF	Real-time clock (RTC)
PW09CNF	JPEG resize	PW19CNF	DMA controller 2 (DMAC2)
PW0ACNF	JPEG module/FIFO control	PW1ACNF	Memory controller
PW0BCNF	JPEG codec	PW1BCNF	Timers
PW0CCNF	JPEG DMAC	PW1CCNF	Watchdog timer
PW0DCNF	I2C	PW1DCNF	System controller
PW0ECNF	I2S	PW1ECNF	Reserved
PW0FCNF	(interrupt controller)	PW1FCNF	Interrupt controller

13. SYSTEM CONTROLLER (SYS)

13.1 Overview

This block mainly provides clock control, power management, memory mapping, and other functionality affecting the overall device operation.

Main Features

- Shifting to HALT mode (Low Speed/High Speed) by writing to the HALT Control Register
- Control stopping CPU and bus clocks in these HALT mode
- Dynamic control the CPU, AHB, and APB clock frequency
- Clock control stopping each clock signal to internal I/O blocks
- Support LOW-SPEED (32 kHz) mode
- Software reset
- Programmable clock generator for UART

13.2 Operation States

There are a state and four basic operating mode: POWER ON, LOW SPEED (32 kHz) MODE, LOW SPEED HALT MODE, HIGH SPEED MODE, and HIGH SPEED HALT MODE.

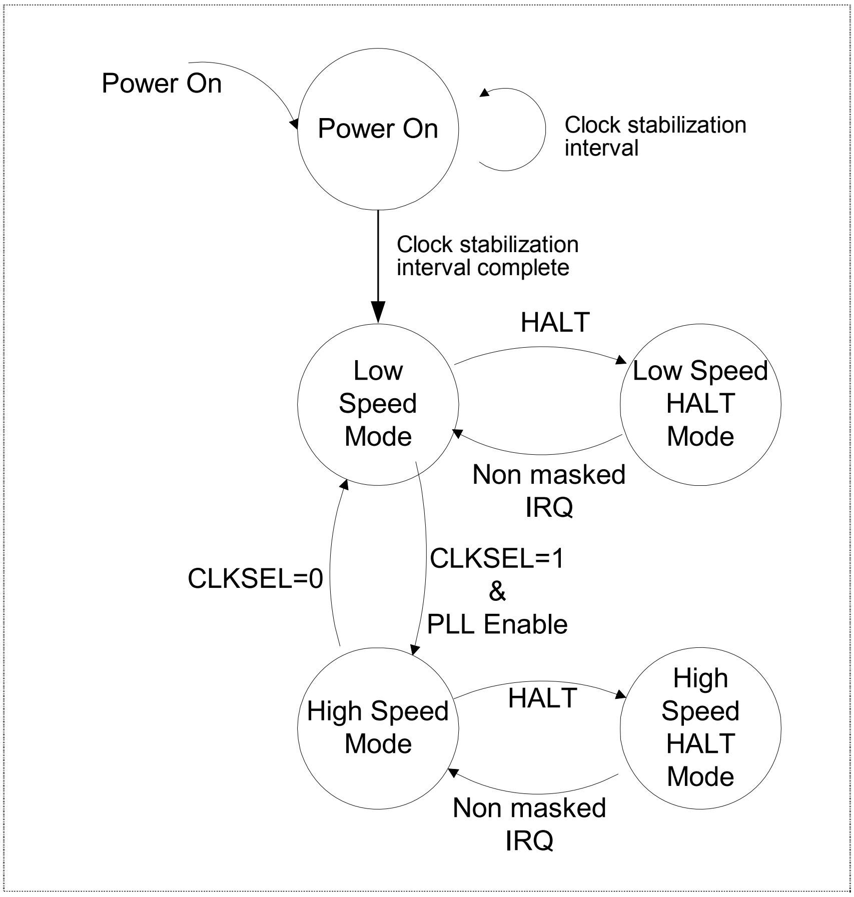


Fig.13.1 SYS State Transition Diagram

Note: The shift from LOW SPEED (32 kHz) MODE to HIGH SPEED MODE allows software to specify a PLL stabilization interval up to 100 ms.

The following pages describe these operating modes (state).

13.2.1 POWER ON state

A power on reset signal shifts to this state. Removal of the reset signal produces an automatic shift to the next state, LOW SPEED (32 kHz) mode, approximately 3 seconds later. This interval is to allow the external 32 kHz crystal oscillator assumed to be the source of the 32 kHz clock input to reach fully operational status.

13.2.2 LOW SPEED mode (32kHz mode)

This mode, the one immediately after reset, starts with the phase-locked loop (PLL) off. Immediately after power on reset, the CPU starts in this mode, too. The 32 kHz name sometimes used refers to the only clock signal available.

This mode is for configuring the PLL and specifying its frequency multiplier*. After allowing sufficient time for PLL operation to stabilize (up to 100 ms), the software switches to the HIGH SPEED mode for operation at the desired frequency. Changing PLL frequencies always requires shifting back to this state.

* This setting is subject to restrictions on the basic clock signals to the UART, timer, and other blocks.

13.2.3 LOW SPEED HALT mode

In the LOW SPEED mode, writing to the HALT Control Register shifts to this mode and, depending on register settings, stops clock signals to the internal I/O buses, internal I/O devices, etc. Turning them all off, for example, yields the state with the lowest power consumption.

An unmasked interrupt request returns from this mode to LOW SPEED mode. The interrupt sources include changes in GPIO input enabled interrupts, external interrupt request pin input, and interrupts from timers running on the 32 kHz clock signal.

13.2.4 HIGH SPEED mode

The PLL supplies the basic clock signal to the CPU, internal buses, and other components using the frequency multiplier specified. The software starts the PLL in LOW SPEED mode and then shifts to this mode.

13.2.5 HIGH SPEED HALT mode

In the HIGH SPEED mode, writing to the HALT Control Register shifts to this mode and, depending on register settings, stops clock signals to the CPU and internal I/O buses. Frequent use of this state—shifting to it whenever the CPU is not in use, for example—helps conserves power.

An unmasked interrupt request (IRQ or FIQ) shifts to the HIGH SPEED mode. Enabling timer or UART receive interrupt requests, for example, causes such input to produce an immediate return.

13.3 External Pins

This block interacts with the following external pins.

Pin Name	I/O	Pin Function	Multiplexed Pin
RESET#	I	Hardware reset	None
CLKI	I	32 kHz clock	None

13.4 Registers

13.4.1 Register List

The base address for these registers is 0xFFFF_D000.

Table 13.1 SYS Register List (Base Address: 0xFFFF_D000)

Address Offset	Register Name	Abbreviation	Default Value	R/W	Data Access Size (Bits)
0x00	Chip ID Register	CHIPID	0x0650_100X	RO	32
0x04	Chip Configuration Register	CHIPCFG	0x0000_XXXX	RO	16/32
0x08	PLL Setting Register 1	PLLSET1	0x0421_D46A	R/W	32
0x0C	PLL Setting Register 2	PLLSET2	0x0000_0000	(R/W)	16/32
0x10	HALT Mode Clock Control Register	HALTMODE	0x0000_0000	R/W	16/32
0x14	I/O Clock Control Register	IOCLKCTL	0x0000_0000	R/W	16/32
0x18	Clock Select Register	CLK32SEL	0x0000_0000	R/W	16/32
0x1C	HALT Control Register	HALTCTL	—	WO	16/32
0x20	Memory Remap Register	REMAP	0x0000_0000	R/W	16/32
0x24	Software Reset Register	SOFTRST	—	WO	32
0x28	UART Clock Divider Register	UARTDIV	0x0000_0000	R/W	16/32
0x2C	MD Bus Pull-down Control Register	MDPLDCTL	0x0000_0000	R/W	16/32
0x30	GPIOC Resistor Control Register	PORTCRCTL	0x0000_0000	R/W	16/32
0x34	GPIOD Resistor Control Register	PORTDRCTL	0x0000_0000	R/W	16/32
0x38	GPIOE Resistor Control Register	PORTERCTL	0x0000_0000	R/W	16/32
0x3C	Internal TEST Mode Register	ITESTM	0x0000_0000	R/W	32
0x40	Embedded Memory Control Register	EMBMEMCTL	0x0000_0010	R/W	16/32

13.4.2 Detailed Register Descriptions

Chip ID Register (CHIPID)
SYS[0x00] Default = 0x0650_100X Read Only

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

Bits 31 to 8: **Product ID Code [23:0]**
This is fixed at 065010h for this device.

Bits 7 to 3: **Reserved**

Bits 2 to 0: **Revision Code [2:0]**
This gives the revision number for this device. This number starts at 01h and increments with each version change.

Chip Configuration Register (CHIPCFG)
SYS[0x04] Default = 0x0000_XXXX Read Only

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

Bits 15 to 0: **CONF [15:0]**
These bits represent the results of sampling pin input levels from the MODESEL[15:0] pins, an alternate function for the data bus (MD[15:0]) pins, to determine internal configuration parameters at the rising edge of the hardware reset (RESET#) signal.
For further details, see Section 4.1 "System Configuration."

PLL Setting Register 1 (PLLSET1)
SYS[0x08] Default = 0x0421_D46A Read/Write

n/a	CS [1:0]		CP [4:0]				
31	30	29	28	27	26	25	24
RS [3:0]				VC[3:0]			
23	22	21	20	19	18	17	16
N-Counter [3:0]				W-Divider [1:0]]		L-Counter [9:8]	
15	14	13	12	11	10	9	8
L-Counter [7:0]							
7	6	5	4	3	2	1	0

These settings together determine the built-in PLL's frequency. Always set PLL settings register 2 (SYS[0x0C]) bit 1 (PLLEN) to "0" to power down the PLL before modifying them.

For specific setting recommendations, see Section 13.5 "Appendix A," 13.6 "Appendix B."

Bits 30 to 29: **CS [1:0]**
Capacity setting for built-in PLL's LPF

Bits 28 to 24: **CP [4:0]**
Current setting for built-in PLL's CP

Bits 23 to 20: **RS [3:0]**
Resistance setting for built-in PLL's LPF

Bits 19 to 16: **VC [3:0]**
Operation parameter for built-in PLL's VCO

Bits 15 to 12: **N-Counter [3:0]**
NN value*
N-Counter is used for the PLL output frequency with L-Counter. See the L-Counter.

Bits 11 to 10: **W-Divider [1:0]**
Built-in PLL's internal frequency divisor for deriving f_{POUT}, the PLL-Out frequency
00: Reserved. Do not use.
01: 1/2 f_{POUT}
10: 1/4 f_{POUT}
11: 1/8 f_{POUT}

Bits 9 to 0: **L-Counter [9:0]**
LL value*

* The N-Counter and L-Counter together determine the built-in PLL's frequency multiplier according to the following formula.

PLL Output = (N-counter+1) × (L-counter+1) × CLKI
where, = NN × LL × CLKI
NN=N-Counter+1, LL=L-counter+1, CLKI = external clock input(32.768 kHz)

PLL Setting Register 2 (PLLSET2)							
SYS[0x0C]	Default = 0x0000_0000						(Read/Write)
n/a							
31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
n/a							
15	14	13	12	11	10	9	8
Reserved —				n/a —			PLLEN (R/W)
7	6	5	4	3	2	1	0

Bits 7 to 4 (RSV): **Reserved**

Bit 0 (PLLEN): **PLL Enable**

0: Disable
1: Enable

Setting this bit to "0" powers down the PLL and enables write access to PLL Settings Register 1.

HALT Mode Clock Control Register (HALTMODE)							
SYS[0x10]	Default = 0x0000_0000						Read/Write
Reserved (0)							
31	30	29	28	27	26	25	24
Reserved (0)							
23	22	21	20	19	18	17	16
Reserved (0)							
15	14	13	12	11	10	9	8
CPUCKSEL [1:0]		n/a		HALT_MDCLK [3:0]			
7	6	5	4	3	2	1	0

This register specifies the frequency divisor for deriving the clock signals to the CPU and internal (AHB1, AHB2, and APB) buses and controls them in HALT mode.

Bits 31 to 8: **Reserved (0)**

Bits 7 to 6: **CPUCKSEL [1:0]**

Specifies frequency divisor for deriving the CPU, AHB1, AHB2, and APB clock signals from the PLL output

Note that the internal bus clock signals have the same frequency as the CPU clock (CPUCLK) signal. In other words, applying a frequency divisor to CPUCLK applies it to all internal bus clock signals as well. Changing this setting immediately produces a glitch-free change in frequency.

00: 1/1 PLL output frequency
01: 1/2 PLL output frequency
10: 1/4 PLL output frequency
11: 1/8 PLL output frequency

Bits 3 to 0: **HALT_MDCLK [3:0]**

Clock signal disable

0: Enable (Default)
1: Disable

Setting a bit to "1" stops the corresponding clock signal (CPU or internal bus) in HALT mode.

Be careful, however, not to stop the clock signals to the UART, Timer, Ethernet, GPIO, or other internal sources generating the unmasked interrupt requests for returning from the HALT mode.

Bit	Target
3	CPU (ARM720T)
2	AHB1*
1	AHB2
0	APB

Setting bits to "1" does not immediately stop the corresponding clock signals. These settings only take effect when the software writes to the HALT Control Register to shift to a HALT mode. Such shifts help reduce power consumption, for example, while the CPU is waiting, doing nothing, because it has no jobs to process or is waiting for the next interrupt request event.

Usage restriction: Stopping the AHB1 bus clock requires stopping the CPU clock (CPUCLK) signal as well. Stopping CPUCLK does not require stopping the AHB1 bus clock signal, however.

I/O Clock Control Register (IOCLKCTL)

SYS[0x14] Default = 0x0000_0000 Read/Write

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

Bits 31 to 9: **Reserved (0)**

Bits 8 to 0: **IOCLKCTL [8:0]**

Clock Enable

0: Disable

1: Enable

Setting a bit to "1" enables the clock signal to the corresponding I/O block (timer, UART, or SPI, for example). Writing "0" to the bits for unused I/O blocks lowers power consumption.

Bit 8 (I2S_CLKEN):	I^2S
Bit 7 (RSV):	Reserved (0)
Bit 6 (UART_CLKEN):	UART
Bit 5 (DMAC2_CLKEN):	DMAC2
Bit 4 (SPI_CLKEN):	SPI
Bit 3 (I2C_CLKEN):	I^2C
Bit 2 (TIMER_CLKEN):	Timers 0 to 2
Bit 1 (CF_CLKEN):	CF card interface
Bit 0 (Ether_CLKEN):	Ethernet

Clock Select Register (CLK32SEL)

SYS[0x18] Default = 0x0000_0000 Read/Write

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

Bit 0: **CLKSEL**

System clock select

0: 32 kHz

1: PLL output

Changing to PLL output ("1") requires first specifying the PLL parameters, enabling the PLL, and waiting an appropriate PLL stabilization interval (100 ms).

HALT Control Register (HALTCTL)							
SYS[0x1C] Default = —							Write Only
Halt Command [31:24]							
31	30	29	28	27	26	25	24
Halt Command [23:16]							
23	22	21	20	19	18	17	16
Halt Command [15:8]							
15	14	13	12	11	10	9	8
Halt Command [7:0]							
7	6	5	4	3	2	1	0

Bits 31 to 0: **Halt Command [31:0]**

Writing to this register shifts to a HALT mode. The value written does not matter.

Memory Remap Register (REMAP)							
SYS[0x20] Default = 0x0000_0000							Read/Write
n/a							
31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
n/a							
15	14	13	12	11	10	9	8
n/a						REMAP2	REMAP1
7	6	5	4	3	2	1	0

Bit1: **REMAP2**

AHB2 bus memory remap Enable

0: Disable
1: Enable

Bit0: **REMAP1**

AHB1 bus memory remap Enable

0: Disable
1: Enable

Setting a bit to “1” changes the memory map for the corresponding bus, assigning SDRAM to the space starting at address 0x0 after a reset.

This remapping is normally not needed, but it can improve operation for a RAM-based operating system. For further details, see Section 13.7 “Appendix C: AHB Memory Maps After Remapping”.

Notes: Changing memory maps requires careful attention to such details as running the code in a memory region not affected by the change.

To avoid inconsistency between AHB1 and AHB2 memory maps, we recommend always simultaneously setting both bits to the same value.

Software Reset Register (SOFTRST)							
SYS[0x24] Default = —							Write Only
Software Reset [31:24]							
31	30	29	28	27	26	25	24
Software Reset [23:16]							
23	22	21	20	19	18	17	16
Software Reset [15:8]							
15	14	13	12	11	10	9	8
Software Reset [7:0]							
7	6	5	4	3	2	1	0

Bits 31 to 0: **Software Reset [31:0]**

Writing “AA5555AAh” to this register initializes all registers inside this device and resets the CPU.

UART Clock Divider Register (UARTDIV) SYS[0x28] Default = 0x0000_0000							Read/Write
31	30	29	28	27	26	25	24
23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8
UARTCLKDIV [7:0]							
7	6	5	4	3	2	1	0

Bits 7 to 0: **UARTCLKDIV [7:0]**

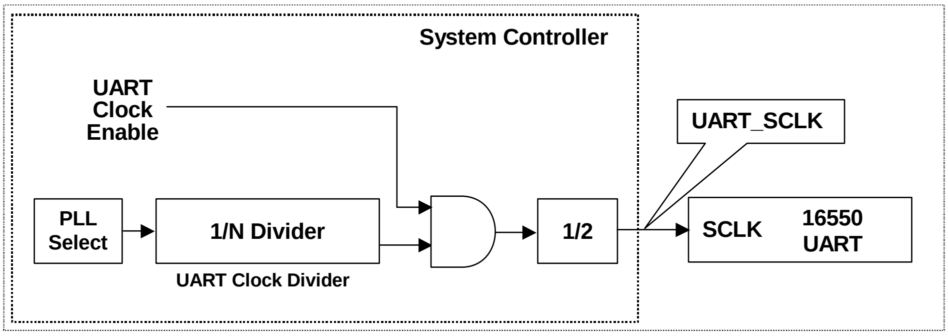
Frequency divisor, N = Bits [7:0] + 1

0: 1/1 frequency
1: 1/2 frequency
.
.
.
255: 1/256 frequency

This register specifies the frequency divisor for deriving the UART baud rate time base. UART_SCLK, the transfer rate generator clock, is the PCLK clock signal after passing through first this frequency divider and then another (1/2). (See Figure.)

UART_SCLK = (PCLK frequency) ÷ N ÷ 2

Note: This UART_SCLK signal is completely different from the SPI interface clock signal SCLK.



MD Bus Pull-down Control Register (MDPLDCTL) SYS[0x2C] Default = 0x0000_0000							Read/Write
Reserved							
31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
MDPLDNDIS [15:8]							
15	14	13	12	11	10	9	8
MDPLDNDIS [7:0]							
7	6	5	4	3	2	1	0

Bits 15 to 0: **MDPLDNDIS [15:0]**

MD[15:0] pin internal pull-down resistor disable

0: Enable (Default)
1: Disable

Setting a bit to “1” disables the internal pull-down resistor on the corresponding data bus (MD[15:0]) pin. This power-saving step becomes necessary for a pin with an external pull-up resistor because that resistor creates a constant current through the internal one after a reset.

GPIOC Resistor Control Register (PORTCRCTL)							
SYS[0x30]	Default = 0x0000_0000						Read/Write
Reserved							
31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
PORTCPDDIS [7:0]							
7	6	5	4	3	2	1	0

Bits 7 to 0: **PORTCPDDIS [7:0]**

GPIOC[7:0] pin internal pull-down resistor disable

0: Enable (Default)

1: Disable

Setting a bit to “1” disables the internal resistor on the corresponding GPIOC[7:0] pin.

GPIOD Resistor Control Register (PORTDRCTL)							
SYS[0x34]	Default = 0x0000_0000						Read/Write
Reserved							
31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
PORTDPUDDIS [7:2]						Reserved	
7	6	5	4	3	2	1	0

Bits 7 to 4: **PRTDPUDDIS [7:4]**

GPIOD[7:4] pin internal pull-down resistor disable

0: Enable (Default)

1: Disable

Setting a bit to “1” disables the internal resistor on the corresponding GPIOD[7:4] pin.

Bits 3 to 2: **PRTDPUDDIS [3:2]**

GPIOD[3:2] pin internal pull-up resistor disable

0: Enable (Default)

1: Disable

Setting a bit to “1” disables the internal resistor on the corresponding GPIOD[3:2] pin.

Bits 1 to 0: **Reserved (0)**

GPIOE Resistor Control Register (PORTERCTL)							
SYS[0x38]	Default = 0x0000_0000						Read/Write
Reserved							
31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
PORTEPUDIS [7:0]							
7	6	5	4	3	2	1	0

Bits 7 to 0: **PORTEPUDIS [7:0]**

GPIOE[7:0] pin internal pull-up resistor disable

0: Enable (Default)

1: Disable

Setting a bit to “1” disables the internal resistor on the corresponding GPIOE[7:0] pin.

Internal TEST Mode Register (ITESTM)							
SYS[0x3C]	Default = 0x0000_0000						— / —
31	30	29	28	27	26	25	24
23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8
7	6	5	4	3	2	1	0

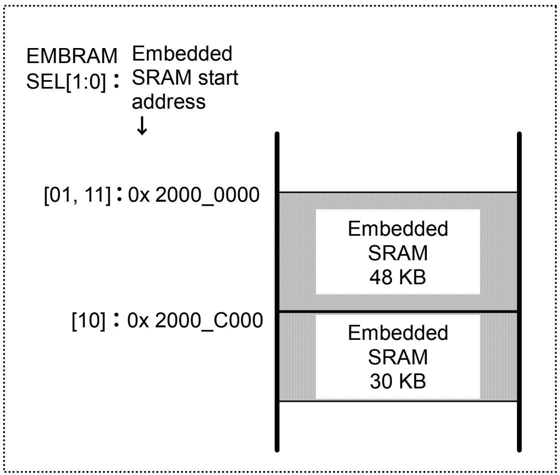
This register is for internal testing. Do not touch. Leave the contents as they are immediately after a reset.

Embedded Memory Control Register (EMBMEMCTL)							
SYS[0x40]	Default = 0x0000_0010						Read/Write
n/a							
31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
n/a							
15	14	13	12	11	10	9	8
Reserved		EMBRAMSEL[1:0]		Reserved		EMBWAITEN[1:0]	
7	6	5	4	3	2	1	0

Bits 31 to 6: **Reserved (0)**

Bits 5 to 4: **EMBRAMSEL [1:0]**
Embedded SRAM Select

00: Allocate all 78 KB to the JPEG line buffer
01: Allocate the first 48 KB (starting at 0x2000_0000) to internal SRAM and the remaining 30 KB (starting at 0x2000_C000) to the JPEG Line Buffer (S1S65000 configuration compatible)
10: Allocate the first 48 KB to the JPEG Line Buffer and the remaining 30 KB to internal SRAM
11: Allocate all 78 KB to internal SRAM



Bits 3 to 2: **Reserved (0)**

Bits 1 to 0: **EMBWAITEN [1:0]**
Embedded SRAM Wait Control

00: No wait cycle
01: Read Access Wait ON
(Read: 1 wait cycle, Write: No wait cycle)
10: Read Access Wait ON, Read Data Wait ON
(Read: 2 wait cycles, Write: No wait cycle)
11: Read Access Wait ON, Read Data Wait ON, Write Access Wait ON
(Read: 2 wait cycles, Write: 1 wait cycle)

13.5 Appendix A: Sample PLL Settings

The following two examples illustrate the procedures for configuring the PLL to derive the desired target system clock (f_{POUT}) frequency from the 32 kHz reference clock.

CPU Clock = 48.955392 MHz

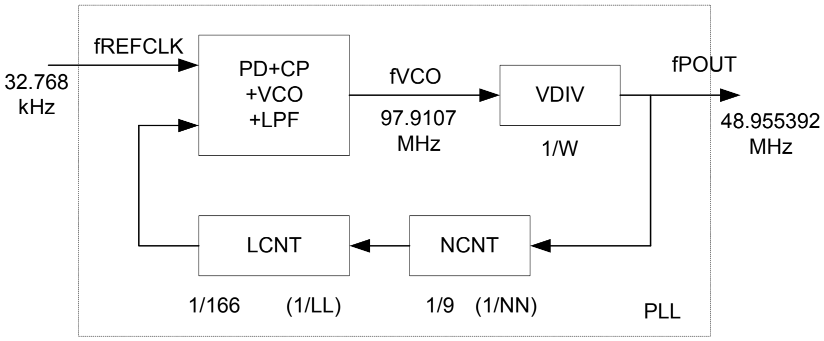
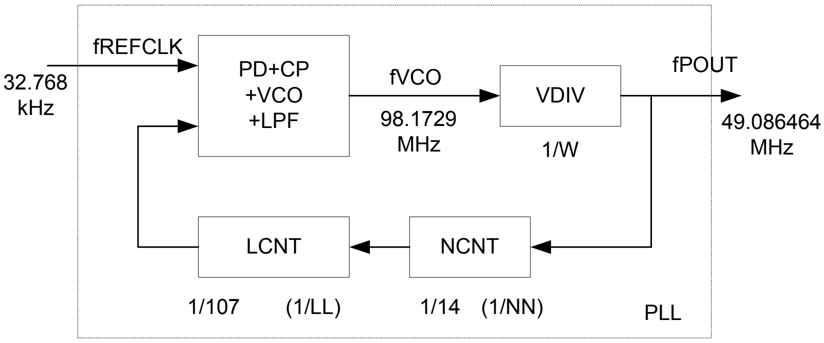


Fig.13.2 PLL Settings for 48.955392 MHz Output

First calculate the necessary frequency divisor.
f_{POUT} / f_{REFCLK} = 48.955392 MHz/32.768 kHz = 1494
Then factor it into NN and LL settings.
1494 = 2 × 3 × 3 × 83 = 9 ×166
Subtract one each for the register settings.
NN = 9: N-Counter = NN – 1 = 8 = 1000b
LL = 166: L-Counter = LL – 1 = 165 = 00_1010_0101b
Setting W to 2 produces the VCO frequency (f_{VCO}) closest to 100 MHz.
f_{VCO} = f_{POUT} × W = 48.955392 MHz × 2 = 97.910784 MHz
Subtract one for the register setting.
W = 2: W-Divider = W – 1 = 01b

CPU Clock = 49.086464 MHz



First calculate the necessary frequency divisor.
f_{POUT} / f_{REFCLK} = 49.086464 MHz/32.768 kHz = 1498
Then factor it into NN and LL settings.
1498 = 2 × 7 × 107 = 14 × 107
Subtract one each for the register settings.
NN = 14: N-Counter = NN – 1 = 13 = 1101b
LL = 107: L-Counter = LL – 1 = 106 = 00_0110_1010b
Setting W to 2 produces the VCO frequency (f_{VCO}) closest to 100 MHz.
f_{VCO} = f_{POUT} × W = 49.086464 MHz × 2 = 98.172928 MHz
Subtract one for the register setting.
W = 2: W-Divider = W – 1 = 01b

13.6 Appendix B: PLL Parameter Table

The following summarizes the above results, showing the PLL Settings Register 1 value for obtaining the desired output frequency.

Target Frequency	Frequency Multiplier for 32.768 kHz Source	PLL Setting Register 1 (Hex)	N-Counter (NN–1)	L-Counter (LL–1)
48.955392 MHz	1494 = 9 × 166	0x042184A5	1000b	00_1010_0101b
49.086464 MHz	1498 = 14 × 107	0x0421D46A	1101b	00_0110_1010b

Use the following W-Divider, VC, RS, CP, and CS settings as long as the VCO frequency stays within 90 to 100 MHz.

W-Divider	VC[3:0]	RS[3:0]	CP[4:0]	CS[1:0]
01b	0001b	0010b	0_0100b	00b

Note: Always allow the PLL output at least 100 ms to stabilize between changing the register settings (turning on the PLL if necessary) and setting the CLKSEL bit to "1" to actually switch from 32 kHz operation to the desired clock frequency.
Note that changing the PLL frequency requires both switching to 32 kHz operation and setting the PLLEN bit to "0" to disable the PLL. Direct changes at other frequencies are not allowed. Always go through the intermediary 32 kHz operation stage.

13.7 Appendix C: AHB Memory Maps After Remapping

13.7.1 AHB1 Memory Maps After Remapping

Remapping dynamically changes the memory map. ROM-based operating systems have no need for this capability, but RAM-based ones sometimes find the modified memory map more convenient.

Table 13.2 AHB1 Memory Map After Remapping

Starting Address	End Address	Size (MB)	Device	External Chip Select	Device Bus Size (Bits)
0x0000_0000	0x07FF_FFFF	128	External SDRAM	CS2	16
0x0800_0000	0x0FFF_FFFF	128	Reserved		
0x1000_0000	0x1FFF_FFFF	256	Reserved		
0x2000_0000	0x2FFF_FFFF	256	Internal SRAM		32
0x3000_0000	0x37FF_FFFF	128	External SDRAM	CS2	16
0x3800_0000	0x3FFF_FFFF	128	Reserved		
0x4000_0000	0x4FFF_FFFF	256	Reserved		
0x5000_0000	0x5FFF_FFFF	256	Reserved		
0x6000_0000	0x6FFF_FFFF	256	Reserved		
0x7000_0000	0x7FFF_FFFF	256	Reserved		
0x8000_0000	0x8FFF_FFFF	256	Reserved		
0x9000_0000	0x9FFF_FFFF	256	Reserved		
0xA000_0000	0xAFFF_FFFF	256	Reserved		
0xB000_0000	0xBFFF_FFFF	256	Reserved		
0xC000_0000	0xC7FF_FFFF	128	External ROM	CS0/CS1	16
0xC800_0000	0xCFFF_FFFF	128	Reserved		
0xD000_0000	0xDFFF_FFFF	256	Reserved		
0xE000_0000	0xEFFF_FFFF	256	Reserved		
0xF000_0000	0xFFFF_FFFF	256	Internal I/O area		32/16/8

13.7.2 AHB2 Memory Map After Remapping

Table 13.3 AHB2 Memory Map After Remapping

Start Address	End Address	Size (MB)	Device	External Chip Select	Device Bus Size (Bits)
0x0000_0000	0x07FF_FFFF	128	External SDRAM	CS2	16
0x0800_0000	0x0FFF_FFFF	128	Reserved		
0x1000_0000	0x1FFF_FFFF	256	Reserved		
0x2000_0000	0x2FFF_FFFF	256	Internal SRAM		32
0x3000_0000	0x37FF_FFFF	128	External SDRAM	CS2	16
0x3800_0000	0x3FFF_FFFF	128	Reserved		
0x4000_0000	0x4FFF_FFFF	256	Reserved		
0x5000_0000	0x5FFF_FFFF	256	Reserved		
0x6000_0000	0x6FFF_FFFF	256	Reserved		
0x7000_0000	0x7FFF_FFFF	256	Reserved		
0x8000_0000	0x8FFF_FFFF	256	Reserved		
0x9000_0000	0x9FFF_FFFF	256	Reserved		
0xA000_0000	0xAFFF_FFFF	256	Reserved		
0xB000_0000	0xBFFF_FFFF	256	Reserved		
0xC000_0000	0xC7FF_FFFF	128	External ROM/SRAM	CS0/CS1	16
0xC800_0000	0xCFFF_FFFF	128	Reserved		
0xD000_0000	0xDFFF_FFFF	256	Reserved		
0xE000_0000	0xEFFF_FFFF	256	JPEG DMA Port		32
0xF000_0000	0xFFFF_FFFF	256	Reserved		

13.8 Clock Control Block Diagram

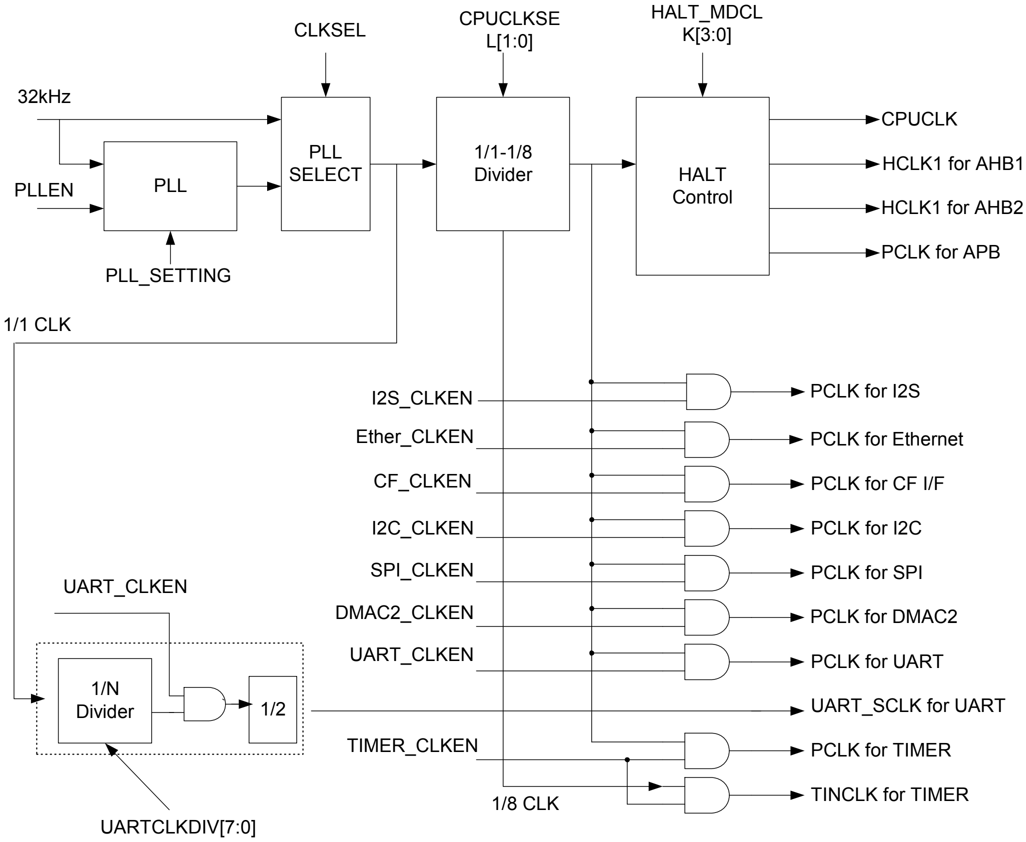


Fig.13.3 Clock Control Block Diagram

The PCLK signals represent the clock signals to the blocks for bus control and register control. UART_SCLK is the clock input for generating the UART transfer rate. TINCLK is the basic clock input for timer block counters.

13.9 Appendix D: Sample UART Clock Settings

CPU Clock = 48.955392 MHz

Baud rate	Ideal x16 Clock (Hz)	SYS[0x28]	16550 Divisor Value (DEC)	16550 Divisor Value (HEX)	Percent Error (%)	Actual x16 Clock (Hz)	UART_SCLK
110	1760	0	13908	3654	0.00	1760.0	24477696
300	4800	0	5100	13EC	0.01	4799.5	24477696
600	9600	0	2550	09F6	0.01	9599.1	24477696
1200	19200	0	1275	04FB	0.01	19198.2	24477696
2400	38400	0	637	027D	0.07	38426.5	24477696
4800	76800	0	319	013F	0.09	76732.6	24477696
9600	153600	0	159	009F	0.23	153947.8	24477696
14400	230400	0	106	006A	0.23	230921.7	24477696
19200	307200	0	80	0050	0.40	305971.2	24477696
38400	614400	0	40	0028	0.40	611942.4	24477696
57600	921600	0	27	001B	1.63	906581.3	24477696
115200	1843200	0	13	000D	2.15	1882899.7	24477696

CPU Clock = 49.086464 MHz

Baud rate	Ideal x16 Clock (Hz)	SYS[0x28]	16550 Divisor Value (DEC)	16550 Divisor Value (HEX)	Percent Error (%)	Actual x16 Clock (Hz)	UART_SCLK
110	1760	0	13945	3679	0.00	1760.0	24543232
300	4800	0	5113	13F9	0.00	4800.2	24543232
600	9600	0	2557	09FD	0.02	9598.4	24543232
1200	19200	0	1278	04FE	0.02	19204.4	24543232
2400	38400	0	639	027F	0.02	38408.8	24543232
4800	76800	0	320	0140	0.13	76697.6	24543232
9600	153600	0	160	00A0	0.13	153395.2	24543232
14400	230400	0	107	006B	0.44	229376.0	24543232
19200	307200	0	80	0050	0.13	306790.4	24543232
38400	614400	0	40	0028	0.13	613580.8	24543232
57600	921600	0	27	001B	1.37	909008.6	24543232
115200	1843200	0	13	000D	2.43	1887940.9	24543232

14. MEMORY CONTROLLER (MEMC)

14.1 Overview

This memory controller is an AHB bus interface controller that supports both asynchronous SRAM and SDRAM. It supports up to 4 asynchronous SRAM devices or up to 2 SDRAM devices (device 0 to device 3).

However, since the S1S65010 does not support device 3, up to 3 asynchronous SRAM devices or one SDRAM device can be used.

This memory controller provides the following features.

- Support for SRAM timing devices
- SDRAM support
- The refresh interval for SDRAM auto-refresh can be adjusted to match the device. If the memory controller cannot be used due to, for example, other memory accesses, it can save up multiple refreshes and perform them as a burst refresh at a point where the memory controller is free. Refresh operations are performed in the background.
- SDRAM self refresh is supported. It can also write to or read from SDRAM that has entered the self-refresh state. In this case, the application can select whether the SDRAM remains in the idle state or enters self-refresh mode after exiting from the self-refresh state.

14.1.1 SRAM Controller

- The SRAM controller supports the following devices.
  Asynchronous SRAM, ROM, FLASH, EEPROM
- External 16-bit data bus
- Support for burst transfers
- Programmable wait states
- Programmable WE# and OE# insertion timing
- Device 0 can be set to the external bus width at boot time. (This is not supported in the S1S65010.)
- The x16 type static memory in which the WE# signal as common and the byte enable signals are independent can be connected with no external logic.

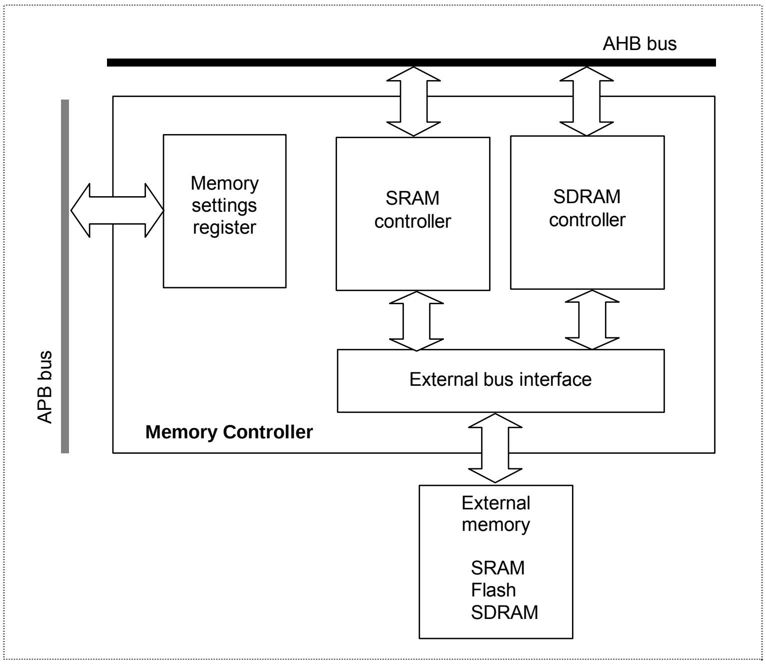
14.1.2 SDRAM Controller

- The SDRAM controller supports the following devices.
  SDRAM
- Bus width: 16 bits
- Burst transfer support
- The auto-precharge function can be turned on/off under program control.
- Supports automatic in/out setting mode for self-refresh mode devices.
- Supports the SDRAM initialization procedure mode.

14.1.3 External Bus Interface Module

- This module is required to use both the SRAM controller and the SDRAM controller with a shared external interface. This module arbitrates external bus access requests from the SRAM controller and the SDRAM controller. Bus requests adopt a handshake method. This unit can be expanded to handle multiple bus requests.
- The SDRAM controller takes first priority and the SRAM controller takes second priority. If there are bus requests on the same clock cycle, the SDRAM controller is given priority. If one memory controller is in the process of access the bus and the other memory controller issues an access request, the other controller must wait to use the bus until the controller using the bus releases the bus.

14.2 Block Diagram



14.3 External Pins

The memory controller interacts with the following external pins.

Table 14.1 Memory Controller Related External Pins

Pin Name	I/O	Pin Function	Multiplexed Pin*
MA23*	O	Memory address output signal 23	GPIOB7/INT7/I2S1_WS
MA22*	O	Memory address output signal 22	GPIOB6/INT6/I2S1_SCK
MA21*	O	Memory address output signal 21	GPIOD1
MA20*	O	Memory address output signal 20	GPIOD0/INT8
MA [19:12]	O	Memory address output signals [19:12]	
MA11	O	Memory address output signal 11	CFREG#**
MA [10:0]	O	Memory address output signals [10:0]	CFADDR [10:0]**
MD [15:0]	I/O	Memory data I/O signals [15:0]	MODSEL [15:0]***
MCS2#	O	Memory chip select signal 2	/SDRAM or other static memory
MCS [1:0]#	O	Memory chip select signals [1:0]	/FlashROM/ROM/SRAM
MOE#	O	Memory output strobe signal	CFOE#**/FlashROM/ROM/SRAM
MWE1#	O	Memory write enable signal	/SDRAM
MWE0#	O	Memory write enable signal	CFWE#*/FlashROM/ROM/SRAM
MCLK	O	SDRAM clock output signal	
MCLKEN	O	SDRAM clock enable output signal	
MRAS#	O	RAS signal for SDRAM	
MCAS#	O	CAS signal for SDRAM	
MDQMH	O	Byte enable signal (for static memory)	High-order byte DQM signal (SDRAM)
MDQML	O		Low-order byte DQM signal (SDRAM)
MWAIT	I	Memory controller wait signal	This signal shares a pin with CFWAIT#

Notes *: Since these memory controller external pins are multiplexed with GPIO or other pins, they can be used by selecting “Non-GPIO function #1” in the GPIO Pin Function Register.

**: When the compact flash (CF) interface is operating, these memory controller pins operate as CF external pins.

***: Operates as a Mode selection pin to determine the internal operating Mode at power on reset.

14.4 Memory Controller

14.4.1 Device Count

Up to 4 devices (devices 0 to 3) can be connected. However, in the S1S65010, only up to 3 devices (devices 0 to 2) can be used.

14.4.2 Memory Types

Devices 0 and 1 are limited to SRAM type devices. Device 2 supports both SRAM and SDRAM type devices. Set the memory type according to the type of the connected memory. The individual detailed settings are made in the SRAM and SDRAM Controller Registers. After a reset, device 0 is set up to be SRAM/ROM and devices 1 to 3 are set to the disabled state.

14.4.3 External Memory Bus Width

Only a width of 16 bits for each device is supported as the external memory bus width.

14.4.4 Device Segment Settings

A single bank of 128 MB is assigned to each of the SRAM and SDRAM type devices. For each device, segments can be set up freely within the one bank. Segments can be freely set up in sizes up to 128 MB in 1 MB units. The hardware will not cancel settings, even if overlapping segment areas are specified for devices belonging to the same memory type. When an overlapping area is accessed, the requested access is not executed and an AHB error is returned to the master. The segment area specifications are only valid if the device is enabled.

14.5 SRAM Control

14.5.1 Device Selection

The corresponding device control register is enabled when the MTYPE bit in the settings register for the device controller used is set to the SRAM type.

14.5.2 Timing Settings

The various device controller timing settings are set by multiplying the period of the clock supplied to the memory controller by coefficients. Set optimal values according to the clock period used.

The items that can be adjusted include the read and write timings and the OE and WE signal insertion timings.

14.5.3 Write Protection

If the WPROTECT bit in the corresponding control register is set to “1,” the corresponding device will be write protected.

14.6 SDRAM Control

14.6.1 Device Selection

The corresponding device control register is enabled when the MTYPE bit in the settings register for the device controller used is set to the SDRAM type.

14.6.2 Mode Register Settings

After setting the SDRAM Mode Register, issue an LMR instruction to the device with the Initialization Control Register. Normally, INIT_SD (MEMC[0x80] bit 15) should be used. Select the target device at the same time as issuing this command. The various device controller timing settings are set in period cycle units. Set optimal values according to the clock period used.

14.6.3 Burst Support

SDRAM supports burst lengths of 1, 2, 4, 8, and full page. (The S1S65010 does not support full page bursts.)

The address is limited to increment burst addressing. In SDRAM, there is no access penalty even if the burst length is exceeded within the same row. If an interrupt due to an access request to a different device occurs during read or write to a device, the SDRAM controller will automatically issue either a Precharge or a Burst Terminate command and quickly hand over the bus.

14.6.4 Auto Precharge Settings

This setting specifies whether or not a precharge is issued after a read or write operation completes. If “do not precharge” is selected, the row will remain selected making data reads or writes in the same page faster. The active row address will be precharged during an auto refresh, and then go to the idle state. It is also possible to precharge manually. See the description of the SDRAM Setting Register.

14.6.5 Power Saving

MCLKEN can be deasserted for a device that has gone to an idle state (including the self refresh state and the state where the row remains active) after a data access has completed. This function is enabled by setting the CKECTRL bit in the SDRAM Setting Register (MEMC[0x70]) to “1.”

If and only if the CKECTRL bit has been set to “1,” it is also possible to stop the clock supplied to the memory to reduce power consumption. This function is enabled by setting the CLKCTRL bit to “1.” If the accumulated value for auto refresh is exceeded and there is a read or write request to the device, data exchange can be restarted by asserting CKE high, even if the CKECTRL bit is “1.”

14.6.6 Stopping the Memory Clock

In addition to holding MCLKEN low with CKECTRL, it is also possible to achieve even lower power consumption by stopping clock supply to the SDRAM. This mode is enabled by setting the CLKCTRL bit to “1.”

The CKECTRL bit must also be enabled when the CLKCTRL bit is enabled. Stop the MCLK memory clock signal when all of the SDRAM connected to the memory controller is in one of the following states.

Memory uninitialized state, self refresh in progress state, or a chip idle state that is not a row active state.

In the memory stopped state, memory clock supply will be restarted if a memory read or write request or an auto refresh request occurs.

Note that as an exception, the MCLK output can be set to continue regardless of the SDRAM memory state by setting the CLKFORCE bit in the SDRAM Detailed Setting Register (MEMC[0x74]) to “1.”

14.6.7 Power Save Mode Support

When the system enters a power save mode (HALT mode), there are cases where clock supply to the memory controller itself will be stopped. While either auto refresh or self refresh is required to retain the data in SDRAM, since the memory controller's clock supply is lost, all of the SDRAM devices must enter self refresh mode.

Applications must implement this by explicitly setting all of the SDRAM devices to self refresh mode first and only then setting the system controller to power save mode (HALT mode).

14.6.8 Auto Refresh Control

The auto refresh operation is executed when at least one SDRAM has been initialized and is furthermore in an IDLE state (other than the self refresh state). The period is the number of HCLK cycles set in the SDRAM Refresh Timer Register. Since the memory controller refreshes connected SDRAM at the same time, the refresh period should be set to match the device that requires the most frequent refresh. While distributed refresh is taken to be the basic technique, multiple refresh requests can be accumulated in order to prevent data transfer from being split up by an auto refresh operation. The number of accumulations is specified with the AREFWAIT field [3:0] in the SDRAM Detailed Setting Register (MEMC[0x74]). If the number of refresh operations accumulated exceeds the specified count, the SDRAM controller recognizes a refresh request and starts an auto refresh at the next memory controller idle.

If a device is performing a bus access at the point that a refresh request occurs, the bus access take priority, and the auto refresh is performed at the point that there are no bus requests.

14.6.9 Self Refresh Control

This memory controller supports SDRAM self refresh. A device transitions to self refresh mode when the SELF bit for the corresponding device in the SDRAM Detailed Setting Register (MEMC[0x74]) is set to "1." To leave self refresh mode, set the corresponding register bit to "0."

If an access (read or write) is performed on an SDRAM device in self refresh mode, the device automatically leaves self refresh mode and the required commands are executed. While, in principle the device goes to an idle state after this access, this block can also be set to enter self refresh mode again. Set the RESELF bit in the SDRAM Detailed Setting Register (MEMC[0x74]) to "1" to use this mode. When the accesses, which may be as many as the count set in the SREFCNT field, have completed, the device will enter self refresh mode automatically.

14.6.10 Status Register

Applications can determined the status of the SDRAM controller and the connected devices.

14.7 Registers

14.7.1 Register List

The table below lists the memory controller control registers. The base address for these registers is 0xFFFF_A000.

Table 14.2 Register List (Base Address: 0xFFFF_A000)

Address Offset	Register Name	Abbreviation	Default Value	R/W	Data Access Size (Bits)
Common configuration registers					
0x00	Configuration Register for Device 0	CFG0	0x1F00_0041	R/W	32
0x04	Configuration Register for Device 1	CFG1	0x7F7F_0040	R/W	32
0x08	Configuration Register for Device 2	CFG2	0x7F7F_0040	R/W	32
0x0C	Reserved* (Configuration Register for Device 3)	CFG3	0x7F7F_0040	R/W	32
SRAM controller registers					
0x20	Timing Register for Device 0	RAMTMG0	0x0000_1C70	R/W	32
0x24	Control Register for Device 0	RAMCNTL0	0x0000_0001	R/W	32
0x30	Timing Register for Device 1	RAMTMG1	0x0000_1C70	R/W	32
0x34	Control Register for Device 1	RAMCNTL1	0x0000_0001	R/W	32
0x40	Timing Register for Device 2	RAMTMG2	0x0000_1C70	R/W	32
0x44	Control Register for Device 2	RAMCNTL2	0x0000_0001	R/W	32
0x50	Reserved* (Timing Register for Device 3)	RAMTMG3	0x0000_1C70	R/W	32
0x54	Reserved* (Control Register for Device 3)	RAMCNTL3	0x0000_0001	R/W	32
SDRAM controller registers					
0x60	Mode Register for SDRAM	SDMR	0x0000_0032	R/W	16/32
0x64	Reserved	—	—	—/—	—
0x68	Reserved	—	—	—/—	—
0x70	Configuration Register for SDRAM	SDCNFG	0x0600_C700	R/W	32
0x74	Advanced Configuration Register for SDRAM	SDADVCNFG	0x000F_0300	R/W	32
0x80	Initialization Control Register	SDINIT	0x0000_0000	R/W	16/32
0x90	Refresh Timer Register for SDRAM	SDREF	0x0000_00A0	R/W	16/32
0xA0	Status Register for SDRAM	SDSTAT	0x0000_0002	RO	32

Note*: Device 3 is not supported in the S1S65010.

14.7.2 Detailed Register Descriptions

Configuration Register for Device 0 (CFG0)															
MEMC[0x00]		Default = 0x1F00_0041													Read/Write
RSV (0)	EDAD [6:0]							RSV (0)	STAD [6:0]						
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV (0)								XBW [1:0]		RSV (0)		MTYPE [3:0]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit 31 (RSV): **Reserved (0)**
This bit is unused and must always be set to “0.”

Bits 30 to 24: **EDAD [6:0]**
Memory Segment End Address
Specifies the device segment end address in 1 MB units. (Initial value: 0x1F)

Bit 23 (RSV): **Reserved (0)**
This bit is unused and must always be set to “0.”

Bits 22 to 16: **STAD [6:0]**
Memory Segment Start Address
Specifies the device segment start address in 1 MB units. (Initial value: 0x00)

Bits 15 to 8 (RSV): **Reserved (0)**
These bits are unused and must always be set to “0.”

Bits 7 to 6: **XBW [1:0]**
External Bus Width
Selects the external bus width. (The S1S65010 only supports the 16-bit external bus width.)
00: Reserved
01: External bus width = 16 bits (Initial value)
10: Reserved (External bus width = 32 bits)
11: Reserved

Bits 5 to 4 (RSV): **Reserved (0)**
These bits are unused and must always be set to “0.”

Bits 3 to 0: **MTYPE [3:0]**
Memory Type
Selects the type of memory connected as device 0.
0000: Disabled
0001: ROM, SRAM, Flash ROM (Initial value)
All other values: Reserved

Configuration Register for Device 1 (CFG1)
MEMC[0x04] Default = 0x7F7F_0040 Read/Write

RSV (0)	EDAD [6:0]							RSV (0)	STAD [6:0]						
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV (0)								XBW [1:0]		RSV (0)		MTYPE [3:0]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit 31 (RSV): **Reserved**
This bit is unused and must always be set to "0."

Bits 30 to 24: **EDAD [6:0]**
Memory Segment End Address
Specifies the device segment end address in 1 MB units. (Initial value: 0x7F)

Bit 23 (RSV): **Reserved**
This bit is unused and must always be set to "0."

Bits 22 to 16: **STAD [6:0]**
Memory Segment Start Address
Specifies the device segment start address in 1 MB units. (Initial value: 0x7F)

Bits 15 to 8 (RSV): **Reserved**
These bits are unused and must always be set to "0."

Bits 7 to 6: **XBW [1:0]**
External Bus Width
Selects the external bus width. (The S1S65010 only supports the 16-bit external bus width.)
00: Reserved
01: External bus width = 16 bits (Initial value)
10: Reserved (External bus width = 32 bits)
11: Reserved

Bits 5 to 4 (RSV): **Reserved**
These bits are unused and must always be set to "0."

Bits 3 to 0: **MTYPE [3:0]**
Memory Type
Selects the type of memory connected as device 1.
0000: Disabled (Initial value)
0001: ROM, SRAM, Flash ROM
All other values: Reserved

Configuration Register for Device 2 (CFG2)
MEMC[0x08] Default = 0x7F7F_0040 Read/Write

RSV (0)	EDAD [6:0]							RSV (0)	STAD [6:0]						
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV (0)								XBW [1:0]		RSV (0)		MTYPE [3:0]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit 31 (RSV): **Reserved**
This bit is unused and must always be set to "0."

Bits 30 to 24: **EDAD [6:0]**
Memory Segment End Address
Specifies the device segment end address in 1 MB units. (Initial value: 0x7F)

Bit 23 (RSV): **Reserved**
This bit is unused and must always be set to "0."

Bits 22 to 16: **STAD[6:0]**
Memory Segment Start Address
Specifies the device segment start address in 1 MB units. (Initial value: 0x7F)

Bits 15 to 8 (RSV): **Reserved**
These bits are unused and must always be set to "0."

Bits 7 to 6: **XBW [1:0]**
External Bus Width
Selects the external bus width. (The S1S65010 only supports the 16-bit external bus width.)
00: Reserved
01: External bus width = 16 bits (Initial value)
10: Reserved (External bus width = 32 bits)
11: Reserved

Bits 5 to 4 (RSV): **Reserved**
These bits are unused and must always be set to "0."

Bits 3 to 0: **MTYPE [3:0]**
Memory Type
Selects the type of memory connected as device 2.
0000: Disabled (Initial value)
0001: ROM, SRAM, Flash ROM
1000: SDRAM
All other values: Reserved

Reserved Register (Configuration Register for Device 3 (CFG3))
MEMC[0x0C] Default = 0x7F7F_0040 Read/Write

RSV (0)	EDAD [6:0]							RSV (0)	STAD [6:0]						
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV (0)								XBW [1:0]		RSV (0)		MTYPE [3:0]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Note: This Register is not supported in the S1S65010.

Bit 31 (RSV): **Reserved**
This bit is unused and must always be set to “0.”

Bits 30 to 24: **EDAD [6:0]**
Memory Segment End Address
Specifies the device segment end address in 1 MB units. (Initial value: 0x7F)

Bit 23 (RSV): **Reserved**
This bit is unused and must always be set to “0.”

Bits 22 to 16: **STAD [6:0]**
Memory Segment Start Address
Specifies the device segment start address in 1 MB units. (Initial value: 0x7F)

Bits 15 to 8 (RSV): **Reserved**
These bits are unused and must always be set to “0.”

Bits 7 to 6: **XBW [1:0]**
External Bus Width
Selects the external bus width.
00: Reserved
01: External bus width = 16 bits (Initial value)
10: External bus width = 32 bits
11: Reserved

Bits 5 to 4 (RSV): **Reserved**
These bits are unused and must always be set to “0.”

Bits 3 to 0: **MTYPE [3:0]**
Memory Type
Selects the type of memory connected as device 2.
0000: Disabled (Initial value)
0001: ROM, SRAM, Flash ROM
1000: SDRAM
All other values: Reserved

Timing Register for Device [3*:0] (RAMTMG[3:0])
MEMC[0x20, 0x30, 0x40, 0x50*] Default = 0x0000_1C70 Read/Write

RSV (0)	WAITWE [4:0]					RSV (0)	WAITOE [4:0]					RSV (0)			
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV (0)	WAITWR [4:0]					RSV (0)	WAITRD [4:0]					RSV (0)			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Note: The Device 3 Timing Register is not supported in the S1S65010.

Bit 31 (RSV): **Reserved**
This bit is unused and must always be set to “0.”

Bits 30 to 26: **WAITWE [4:0]**
Write Enable Signal Delay Insertion Control
Sets the WE# signal insertion timing. (Initial value: 0x0)

Bit 25 (RSV): **Reserved**
These bits are unused and must always be set to “0.”

Bits 24 to 20: **WAITOE [4:0]**
Output Enable Signal Delay Insertion Control
Sets the OE# signal insertion timing. (Initial value: 0x0)

Bits 19 to 15 (RSV): **Reserved**
These bits are unused and must always be set to “0.”

Bits 14 to 10: **WAITWR [4:0]**
Write Cycle Wait Control
Sets the wait cycles during writes. Set the correct number of write wait cycles for the device used. (Initial value: 0x07)

Bit 9 (RSV): **Reserved**
This bit is unused and must always be set to “0.”

Bits 8 to 4: **WAITRD [4:0]**
Read Cycle Wait Control
Sets the wait cycles during reads. Set the correct number of read wait cycles for the SRAM/ROM device used. (Initial value: 0x07)

Bits 3 to 0 (RSV): **Reserved**
These bits are unused and must always be set to “0.”

Control Register for Device [3*:0] (RAMCNTL[3:0])
MEMC[0x24, 0x34, 0x44, 0x54*] Default = 0x0000_0001 Read/Write

RSV (0)															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV (0)												WPROTECT	MWAITPOL[1:0]		RBLE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Notes: When the connected memory is only Flash ROM/ROM/SRAM, this register is effective.
Notes: The Device 3 Control Register (MEMC[0x54]) is not supported in the S1S65010.

Bits 31 to 4 (RSV): **Reserved**
These bits are unused and must always be set to “0.”

Bit 3: **WPROTECT**
Write Protect
Writes are not executed for devices for which the write protect bit is set.
0: Write protect disabled (Initial value)
1: Write protect enabled

Bits 2 to 1: **MWAITPOL [1:0]**
Sets the polarity of the MWAIT signal.
00: Disabled (Initial value)
01: Enabled, active low
10: Enabled, active high
11: Reserved

Bit 0: **RBLE**
Byte Lane Control Setting
To implement byte control during writes for devices that do not provide byte lane control, set this bit to “0” and connect this device’s DQM signal to the corresponding memory device’s WE# signal.
0: During reads, DQM[1:0] go high, preventing writes.
1: During reads, DQM[1:0] stay low and all byte lanes are read out. (Initial value)

Mode Register for SDRAM (SDMR)
MEMC[0x60] Default = 0x0000_0032 Read/Write

RSV (0)															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV (0)						WBM	OP Mode [1:0]		CL [2:0]			BT	BL [2:0]		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 31 to 10 (RSV): **Reserved**

These bits are unused and must always be set to "0."

Bit 9: **WBM**

Write Burst Mode

0: Write with the set burst length (Initial value)
1: Single location access

Bits 8 to 7: **OP Mode [1:0]**

Operation Mode

00: Normal operation
xx: All other values are reserved.

Bits 6 to 4: **CL [2:0]**

CAS Latency

000: Reserved
001: CL=1
010: CL=2
011: CL = 3 (Initial value)
1xx: Reserved

Bit 3: **BT**

Burst Type

0: Sequential (Initial value)
1: Reserved

Bits 2 to 0: **BL [2:0]**

Burst Length

000: BL=1
001: BL=2
010: BL=4 (Initial value)
011: BL=8
100: Reserved
101: Reserved
110: Reserved
111: Reserved

(Full page (burst type = 0) mode is not supported by the S1S65010.)

Reserved Registers
MEMC[0x64, 0x68] Default = 0x xxxx_xxxx — / —

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

These registers are reserved and must not be accessed.

Configuration Register for SDRAM (SDCNFG)															
MEMC[0x70]		Default = 0x0600_C700													Read/Write
RSV (0)					CLK-CTRL	CKE-CTRL	RSV (0)	COLW [7:4]				RSV (0)			
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV (0)					TRCD [1:0]		APCG	REF [3:2]		RSV (0)		BNUM [3:2]		RSV (0)	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 31 to 27 (RSV): **Reserved**

These bits are unused and must always be set to “0.”

Bit 26 (CLKCTRL): **MCLK Control**

This bit can only be set to “1” when dynamic MCLKEN control (CKECTRL) is enabled.

0: MCLK is output continuously. However, SDRAM initialization is required.

1: MCLK is stopped when the SDRAM is in an idle state (including the self refresh state). (default)

Note: To change the value of this bit (CLKCTRL) after initializing SDRAM (MEMC[0x80] bit 15 INIT_SD = 1), temporarily set the SDRAM to self refresh mode. That is, first Enable self refresh mode, then change CLKCTRL, and then disable self refresh mode.

Bit 25 (CKECTRL): **Dynamic MCLKEN Control**

0: The MCLKEN = H state is output continuously (except in the self refresh state).

1: MCLKEN is set low when the SDRAM is in an idle state (including the self refresh state and the state where a bank is active but there is no access operation). (default)

Note: To change the value of this bit (CKECTRL) after initializing SDRAM (MEMC[0x80] bit 15 INIT_SD = 1), temporarily set the SDRAM to self refresh mode. That is, first Enable self refresh mode, then change CKECTRL, and then disable self refresh mode.

Bit 24 (RSV): **Reserved**

This bit is unused and must always be set to “0.”

Bits 23 to 20: **COLW [7:4] Column Address Width**

Sets the SDRAM column address width.

COLW[5:4]: Used for setting device 2

COLW[7:6]: Reserved (Used for setting device 3. This field must be set to “0.”)

00: Column address lines A0 to A7 (default)

01: Column address lines A0 to A8

10: Column address lines A0 to A9

11: Column address lines A0 to A9, A11

Bits 19 to 16 (RSV): **Reserved**

These bits are unused and must always be set to “0.”

Bits 15 to 11 (RSV): **Reserved**

These bits are unused and must always be set to “0.”

Bits 10 to 9: **TRCD [1:0] RAS to CAS Delay**

Sets the delay time between MRAS# and MCAS# (in cycles).

00: Reserved

01: 1 cycle

10: 2 cycles

11: 3 cycles (default)

Bit 8 (APCG): **Auto Precharge Control**

Sets the auto precharge mode.

0: No auto precharge (All banks are precharged with the auto refresh execution timing.)

1: Auto precharge enabled (default)

Bits 7 to 6: **REF [3:2] Refresh Cycle**

Sets the refresh cycle period.

REF2: Used for device 2

REF3: Reserved (Used for device 3. This field must be set to “0.”)

0: 2048 or 4096 cycles (default)

1: Reserved (8192 cycles)

Bits 5 to 4 (RSV): **Reserved**

These bits are unused and must always be set to “0.”

Bits [3:2] **BNUM [3:2] Bank Count**

Sets the bank number for the connected SDRAM.

BNUM2: Used for device 2

BNUM3: Reserved (Used for device 3. This field must be set to “0.”)

0: 4-bank device (default)

1: Reserved (2-bank device)

Bits 1 to 0 (RSV): **Reserved**

These bits are unused and must always be set to “0.”

Advanced Configuration Register for SDRAM (SDADVCNFG)

MEMC[0x74] Default = 0x000F_0300 Read/Write

RSV (0)												SREFCNT [3:0]			
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV (0)				AREFWAIT [3:0]				CLK-FORCE	RESELF	RSV (0)		SELF [3:2]		RSV (0)	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 31 to 20 (RSV): **Reserved**

These bits are unused and must always be set to “0.”

Bits 19 to 16: **SREFCNT [3:0]**

Cycle Count Before Reentry to Self Refresh Mode

Sets the number of cycles before reentering self refresh mode after accesses to SDRAM stop. The set value is only valid when the RESELF bit (MEMC[0x74] bit 6) is “1.” (default: 0xF)

Bits 15 to 12 (RSV): **Reserved**

These bits are unused and must always be set to “0.”

Bits 11 to 8: **AREFWAIT [3:0]**

Auto Refresh Hold Count

The refresh operation can be temporarily suspended and multiple refresh operations performed later with the distributed refresh execution timing when the memory controller interface has control of the memory. (default: 0x03)

Bit 7: **CLKFORCE**

0: The MCLK clock output depends on the memory controller state. (default)

1: The MCLK clock is output regardless of the memory controller state.

Bit 6: **RESELF**

Self Refresh Reentry Mode

This bit enables/disables the function returning to self refresh mode after read/write access to SDRAM that has entered self refresh mode by temporarily suspending that refresh operation.

0: The SDRAM does not reenter self refresh mode. (default)

1: SDRAM reenters self refresh mode if there are no accesses for SREFCNT[3:0] cycles.

Bits 5 to 4 (RSV): **Reserved**

These bits are unused and must always be set to "0."

Bits 3 to 2: **SELF [3:2]**

Self Refresh Mode On/Off

This field is automatically cleared to zero when SDRAM that has entered self refresh mode is woken up for a read or write operation. (Except for RESELF mode.)

SELF2: Used for device 2

SELF3: Reserved (Used for device 3. This field must be set to "0.")

0: Exit from self refresh mode (default)

1: Sets SDRAM to self refresh mode.

Bits 1 to 0 (RSV): **Reserved**

These bits are unused and must always be set to "0."

Initialization Control Register (SDINIT)

MEMC[0x80] Default = 0x0000_0000 Read/Write

RSV (0)															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INIT_ SD	RSV (0)							LMR	AREF	PCG-ALL	RSV (0)	DEVSEL [3:2]		RSV (0)	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Note: Do not set more than one command (field) in the Register at the same time.
Normally, INIT_SD should be used for SDRAM initialization.
The other commands should be used when manual initialization is required.
Select the device that is to be the object of the commands with DEVSEL (MEMC[0x80] bits [3:2]).

Bits 31 to 16 (RSV): **Reserved**

These bits are unused and must always be set to "0."

Bit 15 (INIT_SD): **SDRAM Initialization**

Automatically performs an SDRAM initialization sequence. Select the device that is to be the object of the commands with DEVSEL (bits [3:2] in this register).

0: No operation (default)

1: Perform an SDRAM initialization operation. (This bit is automatically returned to the "0" state after the initialization completes.)

Bits 14 to 8 (RSV): **Reserved**

These bits are unused and must always be set to "0."

Bit 7 (LMR): **Load Mode Register**

Issues a load mode register command to the SDRAM.

0: No operation (default)

1: Issue a load mode register command. (This bit is automatically returned to the "0" state after the command is executed.)

Bit 6 (AREF): **Auto Refresh**

Issues an auto refresh command to the SDRAM.

0: No operation (default)

1: Issue an auto refresh command. (This bit is automatically returned to the "0" state after the command is executed.)

Bit 5 (PCGALL): **Precharge All**

Issues a precharge all banks command to the SDRAM.

0: No operation (default)

1: Issue a precharge all banks command. (This bit is automatically returned to the “0” state after the command is executed.)

Bit 4 (RSV): **Reserved**
This bit is unused and must always be set to “0.”

Bits 3 to 2: **DEVSEL [3:2]**
Device Selection
Selects the device to which commands are issued.
DEVSEL2: Used for device 2
DEVSEL3: Reserved (Used for device 3. This field must be set to “0.”)
0: Device not selected (default)
1: Device selected

Bits 1 to 0 (RSV): **Reserved**
These bits are unused and must always be set to “0.”

Refresh Timer Register for SDRAM (SDREF)															
MEMC[0x90]	Default = 0x0000_00A0														Read/Write
RSV (0)															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV (0)				REFTIME [11:0]											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 31 to 12 (RSV): **Reserved**
These bits are unused and must always be set to “0.”

Bits 11 to 0: **REFTIME [11:0]**
Refresh Time
Sets the execution interval for the distributed refresh operation.
Set an HCLK cycle count that matches the distributed auto refresh interval.
For example, If the refresh interval is 16 µs and the system clock HCLK frequency is 10 MHz, then REFTIME should be set as follows.
16 µs × 10 MHz = 160 cycles (= 1010_0000b)

Status Register for SDRAM (SDSTAT)															
MEMC[0xA0]	Default = 0x0000_0002														Read Only
RSV															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV				RSV (DEVST3 [3:0])				RSV				DEVST2 [3:0]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 31 to 12 (RSV): **Reserved**
These bits are unused and must always be set to “0.”

Bits 11 to 8 (RSV): **Reserved (DEVST3[3:0] and data related to device 3 are invalid.)**
Indicates the current status of device 3.

DEVST3[0]	1:	The device is usable (The state where the device is initialized and enabled)
	0:	The device is not usable.
DEVST3[1]	1:	The device is either in an idle state or a suspended state.
	0:	Some other state
DEVST3[2]	1:	One of the device banks’ row is activated.
	0:	No row is activated.
DEVST3[3]	1:	The device is in self refresh mode.
	0:	The device is in a mode other than self refresh mode.

Bits 7 to 4 (RSV): **Reserved**
These bits are unused and must always be set to “0.”

Bits 3 to 0: **DEVST2 [3:0]**
Device 2 Status
Indicates the current status of device 2.

DEVST2[0]	1:	The device is usable (The state where the device is initialized and enabled)
	0:	The device is not usable.
DEVST2[1]	1:	The device is either in an idle state or a suspended state.
	0:	Some other state
DEVST2[2]	1:	One of the device banks’ row is activated.

	0: No row is activated.
DEVST2[3]	1: The device is in self refresh mode.
	0: The device is in a mode other than self refresh mode.

14.8 Limitations on the Use of the Memory Controller (MEMC)

In the S1S65010, since device 3 cannot be used, there are limitation on the use of certain registers. The table below lists the limitations on the use of registers in this chip.

Offset Address	Bit Name	Limitation
MEMC[0x0C] bits[31:0]	All bits	Cannot be used
MEMC[0x50] bits[31:0]	All bits	Cannot be used
MEMC[0x54] bits[31:0]	All bits	Cannot be used
MEMC[0x70] bits[23:22]	COL[7:6]	Cannot be used
MEMC[0x70] bit 7	REF3	Cannot be used
MEMC[0x70] bit 3	BNUM3	Cannot be used
MEMC[0x74] bit 3	SELF3	Cannot be used
MEMC[0x80] bit 3	DEVSEL3	Cannot be used
MEMC[0xA0] bit [11:8]	DEVST3[3:0]	Cannot be used

14.9 Configuration Register for Device[2:0] Setting Example

When ROM, SRAM, and SDRAM are set up as follows:

Memory Type	Device	Capacity	Address
ROM	Device 0	1 MB	0x0000_0000 to 0x000F_FFFF
SRAM	Device 1	512 KB	0x0010_0000 to 0x0017_FFFF
SDRAM	Device 2	8 MB	0x3000_0000 to 0x307F_FFFF

Since memory segments are specified in 1 MB units, a 1 MB area is allocated for SRAM. Accordingly, the Configuration Register for Device[2:0] must be set as follows.

Configuration Register for Device 0: MEMC [0x00] = 0x0000_0041
Configuration Register for Device 1: MEMC [0x04] = 0x0101_0041
Configuration Register for Device 2: MEMC [0x08] = 0x0700_0048

15. INTERRUPT CONTROLLER (INT)

15.1 Overview

This block supports two fast (FIQ) and 32 normal (IRQ) interrupt requests. The Table maps these inputs to signals from peripheral circuits and functional blocks.

Immediately after a reset, the external interrupt request (INT[8:0]) inputs have Low active level triggers; all others, High active level triggers. The former, however, have register settings offering a choice of trigger types (level or edge) and of input signal polarity to match the needs of the intended user application system.

This block handles the FIQ and IRQ interrupt requests and outputs to the ARM720T core's two interrupt request signals: nFIQ and nIRQ, respectively,

The hardware does not assign a priority order to interrupt requests.

Table 15.1 Interrupt Request Sources

Type	Level	Source	Description
Fast (FIQ)	FIQ0	Watchdog timer	
	FIQ1	GPIOB0 pin	Example: Battery Low (*)
Normal (IRQ)	IRQ0	Watchdog timer	
	IRQ1	Interrupt controller	Software request from register
	IRQ2	ARM720T COMMRx	Debug Communication Port
	IRQ3	ARM720T COMMTx	Debug Communication Port
	IRQ4	Timer	16-bit timer channel 0
	IRQ5	Timer	16-bit timer channel 1
	IRQ6	Timer	16-bit timer channel 2
	IRQ7	Ethernet Mac & E-DMA	
	IRQ8	JPEG control	
	IRQ9	DMAC1	DMAC on AHB1 bus
	IRQ10	JPEG DMAC	
	IRQ11	Camera interface	
	IRQ12	Reserved	
	IRQ13	DMAC2	DMA INT (JPEG DMAC on AHB2 bus)
	IRQ14 (**)	GPIOA[7:0] and GPIOB[7:0]	Wide choice of interrupt request input pins: GPIOA[7:0] and GPIOB[7:0]
	IRQ15	SPI	SPI TXRDY/RXRDY
	IRQ16	I2C	Transfer Complete
	IRQ17	UART	UART TXRDY/RXRDY
	IRQ18	RTC	Alarm or Timer tick
	IRQ19	CF card interface	
	IRQ20 (*)	INT0	GPIOB0 direct input
	IRQ21 (*)	INT1	GPIOB1 direct input
	IRQ22 (*)	INT2	GPIOB2 direct input
	IRQ23	UARTL	UART Lite
	IRQ24 (*)	INT3	GPIOB3 direct input
	IRQ25 (*)	INT4	GPIOB4 direct input
	IRQ26 (*)	INT5	GPIOB5 direct input
	IRQ27 (*)	INT6	GPIOB6 direct input
	IRQ28 (*)	INT7	GPIOB7 direct input
	IRQ29 (*)	INT8	GPIOD0 direct input
	IRQ30	I2S0	I2S CH0
	IRQ31	I2S1	I2S CH1

Notes:

* These represent direct inputs from the pins GPIOB[7:0] and GPIOD0. The default Settings specify Low active triggers. An Interrupt controller Control Register provides the only way to change Enable, polarity, Level, and other Interrupt request Settings. They are thus different from IRQ14**, for which GPIO Control Register Settings are available.

** IRQ14 interrupt request is chosen from pins GPIOA[7:0] and GPIOB[7:0]. For further details, see the detailed register descriptions for GPIO[0x40] to GPIO[0x4C] in Section 25 "General-Purpose I/O (GPIO)."

15.2 Block Diagram

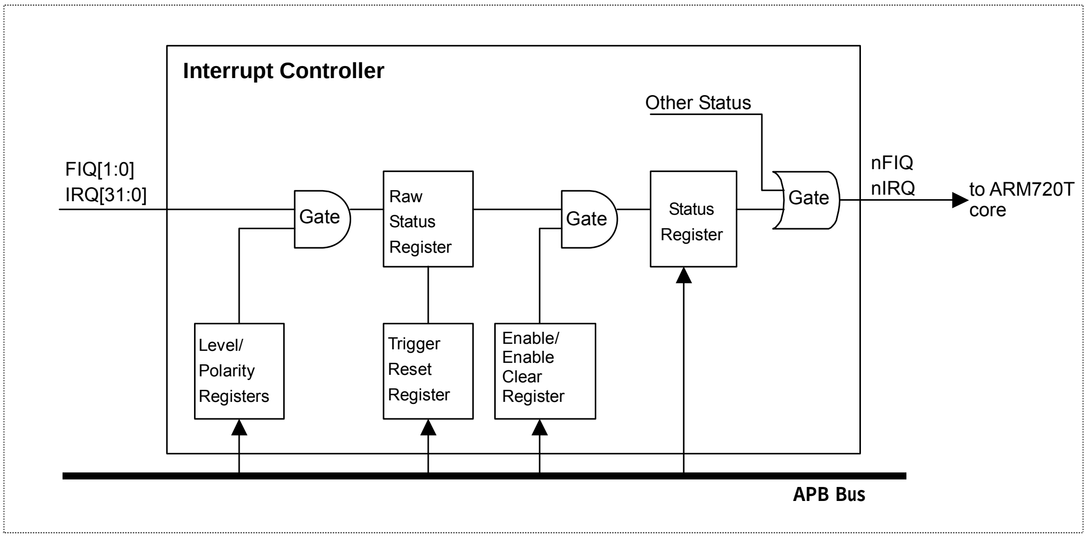


Fig.15.1 Interrupt Controller Block Diagram

15.3 Fast (FIQ) Interrupt Requests

FIQ0 and FIQ1 are assigned as the interrupt request from the watchdog timer and port pin GPIOB0, respectively. These two signals trigger nFIQ interrupt requests to the ARM720T core.

15.4 Normal (IRQ) Interrupt Requests

The interrupt controller has 32 interrupt request inputs: IRQ0 to IRQ31, each with an internal device as their interrupt request source. ORing them together yields the nIRQ interrupt request signal to the ARM720T core. Each source has its own bit in the Interrupt Control Register—bit 0 for IRQ0, bit 1 for IRQ1, etc. Table 15.1 above lists the sources.

15.5 External Pins

This block interacts with the following external pins.

Pin Name	I/O	Pin Function	Multiplexed Pin
FIQ1	I	Fast interrupt request pin 1	GPIOB0/I2S0_WS
INT0	I	External interrupt request pin 0	GPIOB0/I2S0_WS
INT1	I	External interrupt request pin 1	GPIOB1/RTS0#/I2S0_SCK
INT2	I	External interrupt request pin 2	GPIOB2/CTS0#/I2S0_SD
INT3	I	External interrupt request pin 3	GPIOB3/Timer0out/I2S1_SD
INT4	I	External interrupt request pin 4	GPIOB4/ Timer1out
INT5	I	External interrupt request pin 5	GPIOB5/ Timer2out
INT6	I	External interrupt request pin 6	GPIOB6/MA22/I2S1_SCK
INT7	I	External interrupt request pin 7	GPIOB7/MA23/I2S1_WS
INT8	I	External interrupt request pin 8	GPIOD0/MA20

Note: The external interrupt request pins INT0 to INT8 represent GPIO pin inputs. Using these GPIO pins as interrupt request inputs requires configuring internal Interrupt Control registers. Using one for a different function requires specifying non-GPIO function #1 or #2 in the multiplexed Pin Function Register.

15.6 Registers

15.6.1 Register List

The base address for these registers is 0xFFFF_F000.

Table 15.2 INT Register List (Base Address: 0xFFFF_F000)

Address Offset	Register Name	Default Value	R/W	Data Access Size (Bits)
0x000	IRQ Status Register	0x0000_0000	RO	32
0x004	IRQ Raw Status Register	0x0000_0000*1	RO	32
0x008	IRQ Enable Register	0x0000_0000	R/W	32
0x00C	IRQ Enable Clear Register	0x0000_0000	WO	32
0x010	Software IRQ Register	0x0000_0000	WO	32
0x080	IRQ Level Register	0x0000_0000	R/W	32
0x084	IRQ Polarity Register	0xFFFF_FFFF	R/W	32
0x088	IRQ Trigger Reset Register	0x0000_0000	WO	32
0x100	FIQ Status Register	0x0000_0000	RO	32
0x104	FIQ Raw Status Register	0x0000_0000*1	RO	32
0x108	FIQ Enable Register	0x0000_0000	R/W	32
0x10C	FIQ Enable Clear Register	0x0000_0000	WO	32
0x180	FIQ Level Register	0x0000_0000	R/W	32
0x184	FIQ Polarity Register	0x0000_0003	R/W	32
0x188	FIQ Trigger Reset Register	0x0000_0000	WO	32

*1: The Default Value of the IRQ or FIQ Raw Status Register varies depending on system configuration conditions.

15.6.2 Detailed Register Descriptions

In the absence of any indication to the contrary, register bits not labeled reserved all is set as "0."

IRQ Status Register
INT[0x000] Default = 0x0000_0000 Read Only

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

Bits 31 to 0: **IRQ [31:0] Status**
IRQ Status After Masking
0: There is no interrupt request
1: There is an interrupt request
"1" in a bit indicates that there is an interrupt request from the corresponding source after masking with the IRQ Enable Register (INT[0x008])—that is, only if that register enables interrupt requests for that source.
"1" in a bit also triggers an interrupt request to the CPU.
Setting a bit in the IRQ Raw Status Register to "0" simultaneously sets the corresponding bit in this register to "0" as well.
There is a 1:1 correspondence between the bit numbers and the IRQ[31:0] signals. See Table 15.1 above for a list.

IRQ Raw Status Register															
INT[0x004]		Default = 0x0000_0000													Read Only
IRQ [31:16] Status															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IRQ [15:0] Status															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 31 to 0: **IRQ [31:0] Status**

IRQ Status Before Masking

0: There is no interrupt request

1: There is an interrupt request

"1" in a bit indicates that there is an interrupt request from the corresponding source before masking with the IRQ Enable Register (INT[0x008])—that is, even when that register disables interrupt requests for that source.

Resetting a bit to "0" requires clearing the source's interrupt request flag for level trigger interrupt requests and writing "1" to the corresponding bit in the IRQ Trigger Reset Register (INT[0x088]) for edge trigger interrupt requests.

There is a 1:1 correspondence between the bit numbers and the IRQ[31:0] signals. See Table 15.1 above for a list.

IRQ Enable Register															
INT[0x008]		Default = 0x0000_0000													Read/Write
IRQ [31:16] Enable															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IRQ [15:0] Enable															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 31 to 0: **IRQ [31:0] Enable**

Interrupt Request Enable Bits

0(r): There is no interrupt request

1(r): There is an interrupt request

0(w): Disable

1(w): Enable

For reads, a "1" in a bit indicates that the corresponding interrupt request is currently enabled.

Writing "1" to a bit enables the corresponding source, causing the interrupt controller to pass interrupt request input from it on to the CPU.

Writing "0" to a bit does nothing (no operation). It does not disable interrupt request input from the corresponding source, for example. Clearing a bit to "0" requires writing "1" to the corresponding bit in the IRQ Enable Clear Register (INT[0x00C]).

Reset operation clears all bits in this register, disabling all interrupt requests.

There is a 1:1 correspondence between the bit numbers and the IRQ[31:0] signals. See Table 15.1 above for a list.

IRQ Enable Clear Register															
INT[0x00C]		Default = 0x0000_0000													Write Only
IRQ [31:16] Enable Clear															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IRQ [15:0] Enable Clear															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 31 to 0: **IRQ [31:0] Enable Clear**

Interrupt Request Mask Bits

0: (ignored)

1: Disable interrupt request input

Writing “1” to a bit clears the corresponding IRQ Enable bit in the IRQ Enable Register (INT[0x008]) to “0,” disabling (masking) interrupt request (IRQ) input from the corresponding source. Writing “0” to a bit does nothing (no operation).

There is a 1:1 correspondence between the bit numbers and the IRQ[31:0] signals. See Table 15.1 above for a list.

Software IRQ Register

INT[0x010] Default = 0x0000_0000 Write Only

n/a															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
n/a														Soft-ware IRQ	RSV
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit 1: **Software IRQ Control**

0 (w): Negate

1 (w): Assert

This bit controls the software interrupt request (IRQ1) signal.

IRQ Raw Status Register (INT[0x004]) bit 1 gives the current software interrupt request status.

Bit 0 (RSV): **Reserved**

IRQ Level Register

INT[0x080] Default = 0x0000_0000 Read/Write

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

Bits [31:0]: **IRQ [31:0] Level**

0: Level trigger mode

1: Edge trigger mode

These bits specify the IRQ input trigger modes—that is, whether sampling the interrupt request (IRQ) signal by levels or edges. The IRQ Polarity Register (INT[0x084]) specifies the signal polarity—that is, which level (Low or High) or edge (falling or rising).

There is a 1:1 correspondence between the bit numbers and the IRQ[31:0] signals. See Table 15.1 above for a list.

Note: Normally leave the register contents as they were immediately after reset.

IRQ Polarity Register

INT[0x084] Default = 0xFFFF_FFFF Read/Write

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

Bits 31 to 0: **IRQ [31:0] Polarity**

Interrupt Request Polarities

0: Low level/falling edge

1: High level/rising edge

These bits specify the polarity for sampling the corresponding IRQ signal. The IRQ Level Register (INT[0x080]) specifies the trigger type (level or edge).

There is a 1:1 correspondence between the bit numbers and the IRQ[31:0] signals. See Table 15.1 above for a list.

Note: Normally leave the register contents as they were immediately after reset.

IRQ Trigger Reset Register															
INT[0x088]	Default = 0x0000_0000														Write Only
IRQ [31:16] Trigger Reset															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IRQ [15:0] Trigger Reset															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 31 to 0: **IRQ [31:0] Trigger Reset**

Interrupt Request Trigger Resets

0: (ignored)

1: Clear interrupt request status

Writing “1” to a bit clears the corresponding bit in the IRQ Raw Status Register (INT[0x004]) to “0”—only if the corresponding interrupt request uses an edge trigger.

There is a 1:1 correspondence between the bit numbers and the IRQ[31:0] signals. See Table 15.1 above for a list.

FIQ Status Register															
INT[0x100]	Default = 0x0000_0000														Read Only
n/a															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
n/a														FIQ [1:0] Status	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 1 to 0: **FIQ [1:0] Status**

FIQ status after masking.

0: There is no interrupt request

1: There is an interrupt request

“1” in a bit indicates that there is an interrupt request from the corresponding source after masking with the FIQ Enable Register (INT[0x108])—that is, only if that register enables interrupt requests for that source.

“1” in a bit also triggers an interrupt request to the CPU.

Clearing a bit in the FIQ Raw Status Register to “0” simultaneously sets the corresponding bit in this register to “0” as well.

There is a 1:1 correspondence between the bit numbers and the FIQ[1:0] signals.

FIQ Raw Status Register															
INT[0x104]	Default = 0x0000_0000														Read Only
n/a															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
n/a														FIQ [1:0] Raw Status	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 1 to 0: **FIQ [1:0] Raw Status**

FIQ status before masking.

0: There is no interrupt request

1: There is an interrupt request

“1” in a bit indicates that there is an interrupt request from the corresponding source before masking with the FIQ Enable Register (INT[0x108])—that is, even when that register disables interrupt requests for that source.

Resetting a bit to “0” requires clearing the source’s interrupt request flag for level trigger interrupt requests and writing “1” to the corresponding bit in the FIQ Trigger Reset Register (INT[0x188]) for edge trigger interrupt requests.

There is a 1:1 correspondence between the bit numbers and the FIQ[1:0] signals.

FIQ Enable Register															
INT[0x108]	Default = 0x0000_0000													Read/Write	
n/a															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
n/a														FIQ [1:0] Enable	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 1 to 0: **FIQ [1:0] Enable**

Fast Interrupt Request Enable Bits

0(r): There is no interrupt request
1(r): There is an interrupt request
0(w): Disable
1(w): Enable

For reads, a “1” in a bit indicates that the corresponding interrupt request is currently enabled.

Writing “1” to a bit enables the corresponding source, causing the interrupt controller to pass interrupt request input from it on to the CPU.

Writing “0” to a bit does nothing (no operation). It does not disable interrupt request input from the corresponding source, for example. Resetting a bit to “0” requires writing “1” to the corresponding bit in the FIQ Enable Clear Register (INT[0x10C]).

Reset operation clears all bits in this register, disabling all interrupt requests.

There is a 1:1 correspondence between the bit numbers and the FIQ[1:0] signals.

FIQ Enable Clear Register															
INT[0x10C]	Default = 0x0000_0000													Write Only	
n/a															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
n/a														FIQ [1:0] Enable Clear	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 1 to 0: **FIQ [1:0] Enable Clear**

Fast Interrupt Request Mask Bits

0: (ignored)
1: Disable interrupt request input

Writing “1” to a bit clears the corresponding FIQ Enable bit in the FIQ Enable Register (INT[0x108]) to “0,” disabling (masking) interrupt request (FIQ) input from the corresponding source. Writing “0” to a bit does nothing (no operation).

There is a 1:1 correspondence between the bit numbers and the FIQ[1:0] signals.

FIQ Level Register															
INT[0x180]	Default = 0x0000_0000													Read/Write	
n/a															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
n/a														FIQ [1:0] Level	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 1 to 0: **FIQ [1:0] Level**

0: Level trigger mode
1: Edge trigger mode

These bits specify the FIQ input trigger modes—that is, whether sampling the interrupt request (FIQ) signal by levels or edges. The FIQ Polarity Register (INT[0x184]) specifies the signal polarity—that is, which level (Low or High) or edge (falling or rising).

There is a 1:1 correspondence between the bit numbers and the FIQ[1:0] signals.

FIQ Polarity Register
INT[0x184] Default = 0x0000_0003 Read/Write

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

Bits 1 to 0: **FIQ [1:0] Polarity**

Interrupt Request Polarities

0: Low level/falling edge

1: High level/rising edge

These bits specify the polarity for sampling the corresponding FIQ signal. The FIQ Level Register (INT[0x180]) specifies the trigger type (level or edge).

There is a 1:1 correspondence between the bit numbers and the FIQ[1:0] signals.

FIQ Trigger Reset Register
INT[0x188] Default = 0x0000_0000 Write Only

n/a															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
n/a														FIQ [1:0] Trigger Reset	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 1 to 0: **FIQ [1:0] Trigger Reset**

Interrupt Request Trigger Resets

0: (ignored)

1: Clear interrupt request status

Writing “1” to a bit clears the corresponding bit in the FIQ Raw Status Register (INT[0x104]) to “0”—only if the corresponding interrupt request uses an edge trigger.

There is a 1:1 correspondence between the bit numbers and the FIQ[1:0] signals.

16. UART

16.1 Overview

This block provides an asynchronous data transfer interface compatible with the industry standard, 16550. It converts parallel data from the CPU into serial data for transmission to peripheral devices and, going in the opposite direction, serial data received from peripheral devices into parallel data.

16.2 Block Diagram

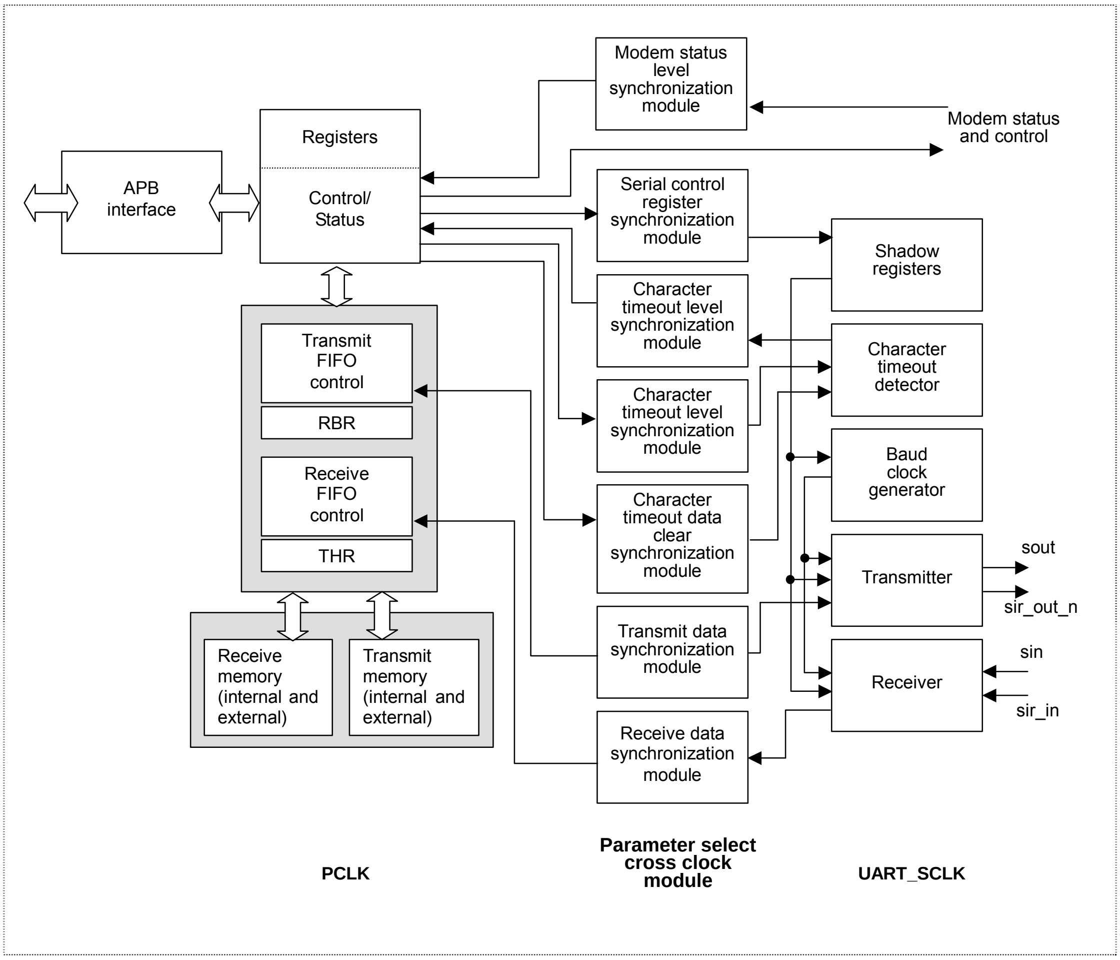

Modem status level synchronization module
Modem status and control
Registers
APB interface
Control/ Status
Serial control register synchronization module
Shadow registers
Character timeout level synchronization module
Character timeout detector
Transmit FIFO control
RBR
Character timeout level synchronization module
Baud clock generator
Receive FIFO control
THR
Character timeout data clear synchronization module
Transmitter
sout
sir_out_n
Receive memory (internal and external)
Transmit memory (internal and external)
Transmit data synchronization module
Receiver
sin
sir_in
Receive data synchronization module
PCLK
Parameter select cross clock module
UART_SCLK

Fig.16.1 UART Block Diagram

16.3 External Pins

This block interacts with the following external pins.

Pin Name	I/O	Pin Function	Multiplexed Pin*
TXD0	Output	UART transmit data	GPIOA0
RXD0	Input	UART received data	GPIOA1
RTS0#	Output	UART ready to send	GPIOB1/INT1/I2S0_SCK
CTS0#	Input	UART clear to send	GPIOB2/INT2/I2S0_SD

* These external pins are multiplexed with GPIO pin and other functions pin, so specify “non-GPIO function #1” in the GPIO Pin Function Register to configure them for this function.

16.4 Registers

The default base address for these registers is 0xFFFF_5000.
In the absence of any indication to the contrary, register bits not labeled reserved all is set as “0.”

16.4.1 Register List

Table 16.1 UART Register List (Base Address: 0xFFFF_5000)

Address Offset	DLAB	Register Name	Abbreviation	Default Value	R/W	Data Access Size (Bits)*
0x00	0	Receive Buffer Register	RBR	0x00	RO	8/16/32
0x00	0	Transmit Holding Register	THR	—	WO	8/16/32
0x00	1	Divisor Latch LSB Register	DLL	0x00	R/W	8/16/32
0x04	0	Interrupt Enable Register	IER	0x00	R/W	8/16/32
0x04	1	Divisor Latch MSB Register	DLM	0x00	R/W	8/16/32
0x08	—	Interrupt Identify Register	IIR	0x01	RO	8/16/32
0x08	—	FIFO Control Register	FCR	—	WO	8/16/32
0x0C	—	Line Control Register	LCR	0x00	R/W	8/16/32
0x10	—	Modem Control Register	MCR	0x00	R/W	8/16/32
0x14	—	Line Status Register	LSR	0x00	RO	8/16/32
0x18	—	Modem Status Register	MSR	0x00	RO	8/16/32
0x1C	—	Scratch Register	SCR	0x00	R/W	8/16/32
0x20	—	Test 0 Register	T0	0x00	R/W	8/16/32
0x24	—	Test 1 Register	T1	0x00	R/W	8/16/32
0x28	—	Test Status 0 Register	TS0	—	RO	8/16/32
0x2C	—	Test Status 1 Register	TS1	0x00	RO	8/16/32
0x30	—	Test Status 2 Register	TS2	0x00	RO	8/16/32
0x3C	—	Test Status 3 Register	TS3	0x00	RO	8/16/32

Note*: This block supports all bus access widths (8, 16, and 32 bits), but only at 32-bit memory boundaries.

16.4.2 Important Notes on Register Access

Accessing the gaps—byte access to offset 01h, for example—between these 8-bit control registers does not yield reliable results. Always access only the specified word boundary offsets.

The test registers (20h to 3CHh) are for debugging the block itself. Do not use them for other purposes because specifications are subject to change.

16.4.3 Detailed Register Descriptions

Receive Buffer Register (RBR)
UART[0x00] DLAB [0] Default = 0x00 Read Only

Serial Received Data (RBR[7:0])							
7	6	5	4	3	2	1	0

Bits 7 to 0: **Serial Received Data RBR [7:0]**

When DLAB (bit 7 in the Line Control Register (UART[0x0C])) is “0,” reads from this address access the receive buffer (RBR), returning the byte data received via serial port.
Note, however, that this data is only valid when this block’s data ready bit (bit 0 in the Line Status Register (UART[0x14])) is “1.”
If FIFO is enabled, read data from this register is the data at the head of receive FIFO.
If the receive FIFO is full, any new data that arrives is discarded to protect the FIFO contents.

Transmit Holding Register (THR)
UART[0x00] DLAB [0] Default = — Write Only

Serial Transmit Data (THR [7:0])							
7	6	5	4	3	2	1	0

Bits 7 to 0: **Serial Transmit Data THR [7:0]**

When DLAB (bit 7 in the Line Control Register (UART[0x0C])) is “0,” writes to this address access the transmit buffer (THR). If FIFO is enabled, enabling to write up to 16 byte in the transmit FIFO. Such writes are ignored, however, if this FIFO is full.

Divisor latch LSB Register (DLL)
UART[0x00] DLAB [1] Default = 0x00 Read/Write

Divisor Latch LSB (DL[7:0])							
7	6	5	4	3	2	1	0

Bits 7 to 0: **Divisor Latch DL [7:0]**

Divisor latch lower half.

Interrupt Enable Register (IER)
UART[0x04] DLAB [0] Default = 0x00 Read/Write

Programmable Transmit Holding Register Empty Interrupt Enable (EPTBEI)	Reserved (0)			Modem Status Interrupt Enable (EDSSI)	Receive Line Status Interrupt Enable (ELSI)	Transmit Holding Register Empty Interrupt Enable (ETBEI)	Ready to Receive Data Interrupt Enable (ERBFI)
7	6	5	4	3	2	1	0

When DLAB (bit 7 in the Line Control Register (UART[0x0C])) is “0,” this address accesses the Interrupt Enable Register (IER) controlling interrupt requests from five sources.

Creating 16550-compatible firmware requires fixing at bit 7 “0”.

Bit 7: **EPTBEI**
Programmable Transmit Holding Register Empty Interrupt Enable
0: Disable
1: Enable
"1" in this bit is only valid during FIFO operation with Transmit Holding Register empty interrupt requests enabled ("1" in both UART[0x08]) bit 0 and (UART[0x04] bit 1).
Note: Creating 16550-compatible firmware requires fixing this bit at "0".

Bit 3: **EDSSI**
Modem Status Interrupt Enable
0: Disable
1: Enable
Note, however, that CTS input changes do not trigger interrupt requests in auto CTS control mode ("1" in both Modem Control Register (UART[0x10]) bit 5 (AFCE) and FIFO Control Register (UART[0x08]) bit 0).

Bit 2: **ELSI**
Receive Line Status Interrupt Enable
0: Disable
1: Enable

Bit 1: **ETBEI**
Transmit Holding Register Empty Interrupt Enable
0: Disable
1: Enable

Bit 0: **ERBFI**
Ready to Receive Data Interrupt Enable
0: Disable
1: Enable

For further details on interrupt request sources, see Table 16.2.

Divisor latch MSB Register (DLM) UART[0x04] DLAB [1] Default = 0x00							Read/Write
Divisor Latch MSB (DL[15:8])							
7	6	5	4	3	2	1	0

Bits 7 to 0: **Divisor Latch DL [15:8]**
Divisor latch upper half.

When DLAB (bit 7 in the Line Control Register at offset 0x0C) is "1," these two registers together specify the frequency divisor for deriving the baud rate from the source clock signal. Specify the lower half (LSB) in UART[0x00] and the upper half (MSB) in UART[0x04].
The following is the formula for the baud rate.

baud rate = input clock ÷ DL[15:0] ÷ 16

Table 16.4 gives the relationship between baud rate and frequency divisor for a 24 MHz source clock signal.

Interrupt Identify Register (IIR) UART[0x08] Default = 0x01							Read Only
FIFO Enable (FFEN [1:0])		Reserved		Interrupt ID (IID [3:0])			
7	6	5	4	3	2	1	0

Bits 7 to 6: **FFEN [1:0]**
FIFO Enable Status bits

00: FIFO disabled
11: FIFO enabled

These bits give the buffering status—both on or both off.

Bits 3 to 0: **IID [3:0]**
Interrupt ID

This field identifies the source for the current interrupt request.

Table 16.2 UART Interrupt Requests

IID [3:0]	Interrupt Type	Source	IID Reset Procedure	Priority Level
0001	None	None	n/a	n/a
0110	Receive line status interrupt	• Overrun error, parity error, framing error, break received	• Read Line Status Register.	1 (Highest)
0100	Ready to receive data interrupt	• Ready to receive data	• Read Line Status Register.	2
	Receive trigger level reached interrupt	• Receive FIFO data level has risen to trigger threshold	• Bring data level in receive FIFO below trigger level.	
1100	Character timeout interrupt	• Nonempty receive FIFO with no data reads or input for the equivalent time of four characters	• Read from Receive Buffer Register (RBR).	2
0010	Transmit Holding Register empty interrupt	• Transmit Holding Register empty	• Read Interrupt Identify Register • Write to the Transmit Holding Register.	3
	Transmit trigger level reached interrupt	• Transmit FIFO data level has fallen to trigger level	• Read Interrupt Identify Register • Write enough data in the transmit FIFO for the trigger level.	
0000	Modem status interrupt	• Change CTS#, DSR#, RI#, or DCD# input	• Read Modem Status Register.	4 (Lowest)

FIFO Control Register (FCR)
UART[0x08] Default = — Write Only

Receive Data Trigger Level (RCVRT[1:0])		Transmit Data Trigger Level (XMITT[1:0])		DMA Mode Select (DMAMS)	Transmit FIFO Reset (XMITFR)	Receive FIFO Reset (RCVRFR)	FIFO Enable (EFIFO)
7	6	5	4	3	2	1	0

Bits 7 to 6: **RCVRT [1:0]**
Receive Data Trigger Level

00 (w):1 byte
01 (w): 4 bytes
10 (w): 8 bytes
11 (w): 14 bytes

If bit 0 (FIFO enable) is "1," enabling buffering with the FIFO, this field specifies the receive FIFO data level triggering a receive trigger level reached interrupt request. A data level at or above this setting triggers an interrupt request. The interrupt request automatically clears when reading from the Receive Buffer Register (RBR) reduces the data level below the setting.

Bits 5 to 4: **XMITT [1:0]**

Transmit Data Trigger Level

00 (w): 0 bytes
01 (w): 2 bytes
10 (w): 4 bytes
11 (w): 8 bytes

If bit 0 (FIFO enable) is "1," enabling buffering with the FIFO, this field specifies the transmit FIFO data level triggering a programmable Transmit Holding Register empty interrupt request—if enabled by a "1" in the corresponding enable bit, UART[0x04] bit 7. A data level at or below this setting triggers an interrupt request. The interrupt request automatically clears when writing to the Transmit Buffer Register raises the data level above this setting.

Note: Creating 16550-compatible firmware requires fixing this bit at "0".

Bit 3: **DMAMS**

DMA mode select

0 (w): Single word
1 (w): Multiple words

This specifies the operation mode for the internal signal DMA controller status.

Bit 2: **XMITFR**

Transmit FIFO reset

0: (ignored)
1: Clear

If bit 0 (FIFO enable) is "1," enabling buffering with the FIFO, writing "1" to this bit clears data in the transmit FIFO. It does not reset the shift register, however. This bit automatically returns to "0" afterwards.

Note: During this operation, the hardware has enough time to transmit up to two characters in the transmit FIFO.

Bit 1: **RCVRFR**

Receive FIFO reset

0: (ignored)
1: Clear

If bit 0 (FIFO enable) is "1," enabling buffering with the FIFO, writing "1" to this bit clears data in the receive FIFO. It does not reset the shift register, however.

This bit automatically returns to "0" afterwards.

Note: This operation does not necessarily clear the Data Ready bit in the Line Status Register (UART[0x14]). Continue reading from the Receive Buffer Register (RBR) (UART[0x00]) until the Data Ready bit goes to "0."

Bit 0: **EFIFO**

FIFO Enable

0: Disable
1: Enable

This bit simultaneously enables both the transmit and receive FIFO.

Line Control Register (LCR) UART[0x0C] Default = 0x00 Read/Write							
Divisor Latch Access Bit (DLAB)	Break Control (SBRK)	Reserved (0)	Even Parity (EPS)	Parity Enable (PEN)	Number of Stop Bits (STB)	Word Length Bits (WLS[1:0])	
7	6	5	4	3	2	1	0

Bit 7 (DLAB): **Divisor Latch Access Bit**

0: Disable

1: Enable

Setting this bit to "1" switches UART[0x00] and UART[0x04] accesses to the divisor latch LSB/MSB Registers instead of the Receive Buffer (RBR), Transmit Holding (THR), and Interrupt Request Enable Registers (IER).

Bit 6 (SBRK): **Break Control**

0: Normal output

1: Break signal output

Setting this bit to "1" drives serial output pins at Low level until the software resets it to "0."

Bit 4 (EPS): **Even Parity**

0: Odd

1: Even

This bit specifies odd or even parity. This setting is only valid when Parity Enable (bit 3) is "1."

Bit 3 (PEN): **Parity Enable**

0: Disable

1: Enable

Setting this bit to "1" enables parity checking for received data and adds parity bits to transmit data.

Bit 2 (STB): **Number of Stop Bits**

0: 1

1: 1.5 for 5 bits of data length; 2 for 6, 7, and 8 bits of data length

This field specifies the number of stop bits added to transmit data. This block only checks the first stop bit for received data.

Bits 1 to 0: **WLS [1:0]**

Word Length Bits

00: 5 bits

01: 6 bits

10: 7 bits

11: 8 bits

This field specifies the bit length in the transmit/receive characters. This count does not include parity or stop bits added at the end.

Modem Control Register (MCR)
UART[0x10] Default = 0x00 Read/Write

n/a		Auto Flow Control Enable (AFCE)	Loop Back (LOOP)	Output #2 Control (OUT2)	Output #1 Control (OUT1)	RTS# Control (RTS)	DTR# Control (DTR)
7	6	5	4	3	2	1	0

Bit 5 (AFCE): **Auto Flow Control Enable**

0: Disable
1: Enable

Setting this bit to "1" switches from manual flow control to auto flow control using the modem signals CTS# and RTS#.

If UART[0x08] bit 0 (FIFO enable) is "1," enabling buffering with the FIFO, setting this bit to "1" activates auto CTS# control mode, which automatically suspends data transmission when the CTS# input goes to High level and resumes data transmission when the CTS# input goes to Low level—if transmit FIFO has data, that is.

Setting Modem Control Register bit 1 to "1" as well activates auto RTS control mode, which automatically drives the RTS# output at High level when the receive FIFO data level reaches the trigger threshold. This RTS output automatically returns to Low level when the receive FIFO is empty.

Note: Creating 16550-compatible firmware requires fixing this bit at "0".

Bit 4 (LOOP): **Loop Back**

0: Normal operation
1: Local loop test

Setting this bit to "1" configures the interface for a modem local loop test, connecting the serial output lines to serial input lines to feed serial data and modem control outputs back as inputs, making it possible to perform simple self-diagnostics with no additional equipment. Note that the coverage includes interrupt requests, breaks, and other functionality as well.

The following Table summarizes the differences between the two modes.

Table 16.3 Differences Between Loop Back and Normal Modes

Location		Loop Back	Normal
DTR# pin output		Constant High level (inactive)	Inverse of DTR bit
RTS# pin output		Constant High level (inactive)	Inverse of RTS bit
TXD pin output		Constant High level (mark state)	Sequential serial data from Transmit Shift Register (TSR)
Modem Status Register (MSR)	CTS bit	Reads return RTS bit	Reads return inverse of CTS# pin state
	DSR bit	Reads return DTR bit	Reads return inverse of DSR# pin state
	RI bit	Reads return OUT1 bit	Reads return inverse of RI# pin state
	DCD bit	Reads return OUT2 bit	Reads return inverse of DCD# pin state
	DCTS bit	Tracks changes in RTS bit	Tracks changes in CTS# pin
	DDSR bit	Tracks changes in DTR bit	Tracks changes in DSR# pin
	TERI bit	Tracks OUT1 bit falling edges	Tracks changes in RI# pin
	DDCD bit	Tracks changes in OUT1 bit	Tracks changes in DCD# pin
Receive shift register		Sequential serial data from Transmit Shift Register (TSR)	Serial data from RXD pin

Bit 3 (OUT2): **Output #2 Control**

0: High level
1: Low level

This bit directly controls OUT2# output. Loop back mode, however, drives the pin at a constant High level and connects to the internal signal equivalent to the DCD# signal.

Note: This bit only applies to loop back mode.

Bit 2 (OUT1): **Output #1 Control**

0: High level
1: Low level

This bit directly controls OUT1# output. Loop back mode, however, drives the pin at a constant High level and connects to the internal signal equivalent to the RI# signal.

Note: This bit only applies to loop back mode.

Bit 1 (RTS): **RTS# Control**

This bit directly controls RTS# output in the manual flow control mode (Modern Control Register bit 5 (AFCE) is set to “0.”).

0: High level
1: Low level

In the auto flow control mode (AFCE bit = “1”), this bit enables auto RTS control mode.

0: Disable
1: Enable

Loop back mode, however, drives the pin at a constant High level and connects to the internal signal equivalent to the CTS# signal.

Bit 0 (DTR): **DTR# control**

0: High level
1: Low level

This bit directly controls DTR# output. Loop back mode, however, drives the pin at a constant High level and connects to the internal signal equivalent to the DSR# signal.

Note: This bit only applies to loop back mode.

Line Status Register (LSR)
UART[0x14] Default = 0x00 Read Only

Receive FIFO Error (RCVRE)	Transmit Empty (TEMT)	Transmit Holding Register Empty (THRE)	Break Interrupt (BI)	Framing Error (FE)	Parity Error (PE)	Overrun Error (OE)	Data Ready (DR)
7	6	5	4	3	2	1	0

Bit 7 (RCVRE): **Receive FIFO Error**

0: No error
1: Error detected

“1” in this bit indicates a receive FIFO error: parity error, framing error, or break. This bit is only valid when UART[0x08] bit 0 (FIFO enable) is set to “1,” enabling buffering with the FIFO. Otherwise, it remains “0.”

Reading this register resets this bit to “0” if the data at the head of the receive FIFO is the only one with such an error.

Bit 6 (TEMT): **Transmit Empty**

0: Not empty
1: Empty

“1” in this bit indicates that there is no data to transmit from the Transmit Shift Register and transmit FIFO if UART[0x08] bit 0 (FIFO enable) is set to “1” (or Transmit Holding Register if FIFO enable is set to “0”).

Writing transmit data returns this bit to “0.”

Bit 5 (THRE): **Transmit Holding Register Empty**

The meaning of this bit depends on Interrupt Enable Register (UART[0x04]) bit 7 (EPTBEI) which controls Programmable Transmit Holding Register empty interrupt requests.

If EPTBEI is "0" (disable), a "1" in this bit indicates that there is no data to transmit from the transmit FIFO (or Transmit Holding Register if UART[0x08] bit 0 (FIFO enable) is "0," disabling buffering with the FIFO). It also triggers an interrupt request if Transmit Holding Register empty interrupt requests are enabled.

0: Not empty
1: Empty

This bit goes to "1" only when regular transmit operation empties both the transmit FIFO or the Transmit Holding Register. Resets, FIFO clears, or other operations emptying the transmit FIFO do not affect this bit.
Writing transmit data returns this bit to "0."

Setting EPTBEI to "1" (enable) reverses the meaning, so that a "1" in this bit indicates that the transmit FIFO is full; a "0," that it is safe to write data to the transmit FIFO.

0: Not full
1: Full

In other words, by enabling Programmable Transmit Holding Register empty interrupt, the software can refill the transmit FIFO by alternately checking this bit and writing the next data to transmit. Keeping the transmit FIFO well supplied with data boosts data transfer efficiency even in systems that cannot respond immediately to interrupt requests.

Bit 4 (BI): **Break Interrupt**

0: There is no interrupt request
1: There is an interrupt request

"1" in this bit indicates that there is a break interrupt request. This bit goes to "1" when the input line remains at Low level for the equivalent interval to one character during a receive operation.
If UART[0x08] bit 0 (FIFO enable) is "1," enabling buffering with the FIFO, a "1" in this bit indicates that the error has occurred in the data at the head of the FIFO.
Enabling receive line status interrupt requests causes this flag to trigger interrupt requests.
Reading this register resets this bit to "0."

Bit 3 (FE): **Framing Error**

0: No error
1: Error detected

"1" in this bit indicates that there has been a framing error—that is, when the received data did not have a valid stop bit.
If UART[0x08] bit 0 (FIFO enable) is "1," enabling buffering with the FIFO, a "1" in this bit indicates that the error has occurred in the data at the head of the FIFO.
Enabling receive line status interrupt requests causes this flag to trigger interrupt requests.
Reading this register resets this bit to "0."

Bit 2 (PE): **Parity Error**

0: No error
1: Error detected

"1" in this bit indicates that there has been a parity error in the received data when the parity enable bit is set to "1."
If UART[0x08] bit 0 (FIFO enable) is "1," enabling buffering with the FIFO, a "1" in this bit indicates that the error has occurred in the data at the head of the FIFO.
Enabling receive line status interrupt requests causes this flag to trigger interrupt requests.
Reading this register resets this bit to "0."

Bit 1 (OE): **Overrun Error**

0: No error

1: Error detected

"1" in this bit indicates an overrun, arrival of new data with no place to store it because the FIFO (UART[0x08] bit 0 (FIFO enable) is "1," enabling buffering with the FIFO) is full or the software has not read the current data in the Receive Buffer Register yet.

Enabling receive line status interrupt requests causes this flag to trigger interrupt requests.

Reading this register resets this bit to "0."

Bit 0 (DR) **Data Ready**

0: No data available

1: Data available

"1" in this bit indicates that there is a valid data in the receive FIFO (or Receive Buffer Register (RBR) if UART[0x08] bit 0 (FIFO enable) is "0," disabling buffering with the FIFO).

Enabling ready to receive data interrupt requests causes this flag to trigger interrupt requests.

Reading all data from the receive FIFO or RBR resets this bit to "0."

Modem Status Register (MSR)
UART[0x18] Default = 0x00 Read Only

DCD Status (DCD) 7	RI Status (RI) 6	DSR Status (DSR) 5	CTS Status (CTS) 4	DCD Change (DDCD) 3	RI Falling Change (TERI) 2	DSR Change (DDSR) 1	CTS Change (DCTS) 0

Bit 7 (DCD): **DCD Status**

0: High level

1: Low level

This bit indicates the DCD# pin input state.

Note: This bit only applies to loop back mode.

Bit 6 (RI): **RI Status**

0: High level

1: Low level

This indicates the RI# pin input state.

Note: This bit only applies to loop back mode.

Bit 5 (DSR): **DSR Status**

0: High level

1: Low level

This indicates the DSR# pin input state.

Note: This bit only applies to loop back mode.

Bit 4 (CTS): **CTS Status**

0: High level

1: Low level

This indicates the CTS# pin input state.

Bit 3 (DDCD): **DCD change**

0: No change

1: Change detected

This indicates whether the DCD# pin input level has changed since the last read. Loop back mode ("1" in UART[0x10] bit 4), however, ignores the pin and instead monitors UART[0x10] bit 3 (output 2 control) for changes.

Reading this register resets this bit to "0."

Note: This bit only applies to loop back mode.

Bit 2 (TERI): **RI Falling Change**

0: No falling edge

1: Falling edge detected

This indicates whether there has been an RI# falling edge since the last read. Loop back mode ("1" in UART[0x10] bit 4), however, ignores the pin and instead monitors UART[0x10] bit 2 (output 1 control) for falling edges.

Reading this register resets this bit to "0."

Note: This bit only applies to loop back mode.

Bit 1 (DDSR): **DSR Change**

0: No change

1: Change detected

This indicates whether the DSR# pin input level has changed since the last read. Loop back mode ("1" in UART[0x10] bit 4), however, ignores the pin and instead monitors UART[0x10] bit 0 (DTR) control for changes.

Reading this register resets this bit to "0."

Note: This bit only applies to loop back mode.

Bit 0 (DCTS): **CTS Change**

0: No change

1: Change detected

This indicates whether the CTS# pin input level has changed since the last read. Loop back mode ("1" in UART[0x10] bit 4), however, ignores the pin and instead monitors UART[0x10] bit 1 (RTS control) for changes.

Reading this register resets this bit to "0."

Scratch Register (SCR)							
UART[0x1C]	Default = 0x00						Read/Write
Scratch bits (SCR [7:0])							
7	6	5	4	3	2	1	0

Bits 7 to 0: **Scratch bits [7:0]**

This general-purpose register has no effects on hardware operation, so is available for software use.

Test 0 Register (T0)							
UART[0x20]	Default = 0x00						Read/Write
n/a							Test Mode
7	6	5	4	3	2	1	0

Bit 0: **Test Mode**

0: Normal mode

1: Test mode

Setting this bit to "1" specifies test mode, for testing equivalent to loop back testing.

Test 1 Register (T1) UART[0x24] Default = 0x00							Read/Write
n/a				DCD Test	RI Test	DSR Test	CTS Test
7	6	5	4	3	2	1	0

Bit 3: **DCD Test**

0: Low level (active)
1: High level (inactive)

In test mode ("1" in UART[0x20] bit 0), this bit controls DCD# input.

Bit 2: **RI Test**

0: Low level (active)
1: High level (inactive)

In test mode ("1" in UART[0x20] bit 0), this bit controls RI# input.

Bit 1: **DSR Test**

0: Low level (active)
1: High level (inactive)

In test mode ("1" in UART[0x20] bit 0), this bit controls DSR# input.

Bit 0: **CTS Test**

0: Low level (active)
1: High level (inactive)

In test mode ("1" in UART[0x20] bit 0), this bit controls CTS0# input.

Test Status 0 Register (TS0) UART[0x28] Default = —							Read Only
n/a				DCD Raw Status	RI Raw Status	DSR Raw Status	CTS Raw Status
7	6	5	4	3	2	1	0

Bit 3: **DCD Raw Status**

0: Low level (active)
1: High level (inactive)

This always indicates the input status for DCD# signal.

Bit 2: **RI Raw Status**

0: Low level (active)
1: High level (inactive)

This always indicates the input status for RI# signal.

Bit 1: **DSR Raw Status**

0: Low level (active)
1: High level (inactive)

This always indicates the input status for DSR# signal.

Bit 0: **CTS Raw Status**

0: Low level (active)
1: High level (inactive)

This always indicates the input status for CTS0# signal.

Test Status 1 Register (TS1) UART[0x2C] Default = 0x00							Read Only
n/a				DCD Status	RI Status	DSR Status	CTS Status
7	6	5	4	3	2	1	0

Bit 3: **DCD Status**

0: Low level (active)
1: High level (inactive)

This indicates the input status for the DCD# signal recognized by UART circuit.

Bit 2: **RI Status**

0: Low level (active)
1: High level (inactive)

This indicates the input status for the RI# signal recognized by UART circuit.

Bit 1: **DSR Status**

0: Low level (active)
1: High level (inactive)

This indicates the input status for the DSR# signal recognized by UART circuit.

Bit 0: **CTS Status**

0: Low level (active)
1: High level (inactive)

This indicates the input status for the CTS0# signal recognized by UART circuit.

Test Status 2 Register (TS2) UART[0x30] Default = 0x00							Read Only
n/a			BAUDOUT Status	OUT2 Status	OUT1 Status	RTS Status	DTR Status
7	6	5	4	3	2	1	0

Bit 4: **BAUDOUT Status**

0: Low level
1: High level

This indicates the output status for the BAUDOUT# signal from UART circuit.

Bit 3: **OUT2 Status**

0: Low level
1: High level

This indicates the output status for the OUT2# signal from UART circuit.

Bit 2: **OUT1 Status**

0: Low level
1: High level

This indicates the output status for the OUT1# signal from UART circuit.

Bit 1: **RTS Status**

0: Low level (active)
1: High level (inactive)

This indicates the output status for the RTS0# signal from UART circuit.

Bit 0: **DTR Status**

0: Low level (active)
1: High level (inactive)

This indicates the output status for the DTR# signal from UART circuit.

Test Status 3 Register (TS3)
UART[0x3C] Default = 0x00 Read Only

n/a					TXRDY Status	RXRDY Status	INTR Status
7	6	5	4	3	2	1	0

Bit 2: **TXRDY Status**

0: Low level (active)
1: High level (inactive)

This indicates the output status for the TXRDY# signal from UART circuit.

Bit 1: **RXRDY Status**

0: Low level (active)
1: High level (inactive)

This indicates the output status for the RXRDY# signal from UART circuit.

Bit 0: **INTR Status**

0: Low level (active)
1: High level (inactive)

This indicates the output status for the INTR signal from UART circuit.

16.4.4 Sample Baud Rate Settings

The following formula determines the clock frequency divisor for the baud rate setting.

frequency divisor = input clock frequency UART_SCLK (Hz) ÷ baud rate (bps) ÷ 16

The following Table gives the relationship between baud rate and frequency divisor for a 24 MHz source clock signal.

Table 16.4 Baud Rate and Frequency Divisor

Baud Rate	**Theoretical 16 clock cycles**	**24.00256 MHz UART Source Clock**		
		Frequency Divisor for 16 clock cycles	**Deviation (%)**	**Actual 16 clock cycles**
300	4800	5000	0.01	4800.5
600	9600	2500	0.01	9601.0
1200	19200	1250	0.01	19202.0
2400	38400	625	0.01	38404.1
4800	76800	312	0.17	76931.3
9600	153600	156	0.17	153862.6
14400	230400	104	0.17	230793.8
19200	307200	78	0.17	307725.1
28800	460800	52	0.17	461587.7
38400	614400	39	0.17	615450.3
57600	921600	26	0.17	923175.4
115200	1843200	13	0.17	1846350.8
125000	2000000	12	0.01	2000213.3
250000	4000000	6	0.01	4000426.7
500000	8000000	3	0.01	8000853.3
750000	12000000	2	0.01	12001280.0
1500000	24000000	1	0.01	24002560.0

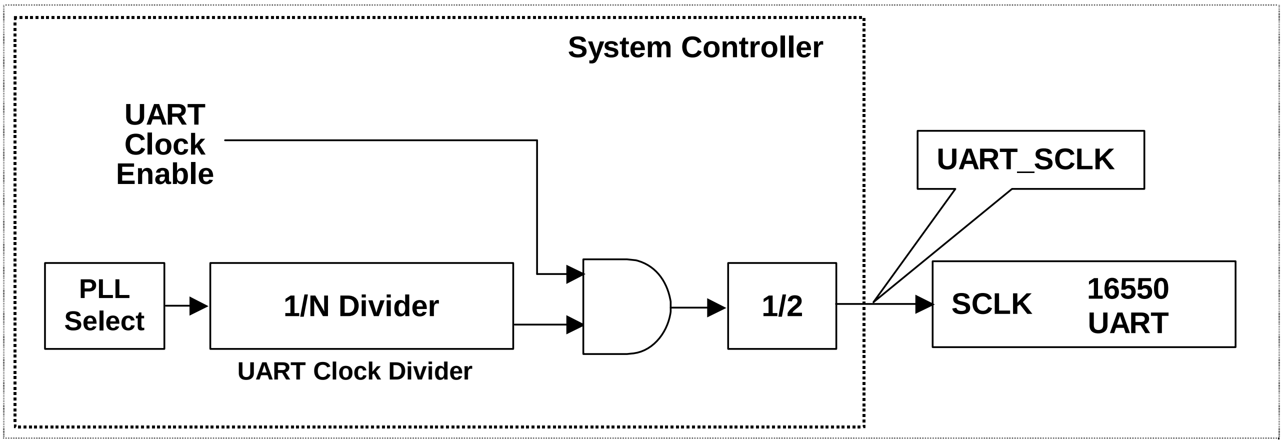


Fig.16.2 UART Clock Signals

16.5 Usage Limitations

Although this UART internally provides most 16550 functionality, it does not provide all the requisite signal I/O pins. The following Tables lists register bits that have usage restrictions as a result.

Offset Address	Register Bit Name	Limitation
UART[0x10]Bit 0	DTR: DTR # control	Can only be used in loop back mode.
UART[0x10]Bit 2	OUT1: Output #1 control	Can only be used in loop back mode.
UART[0x10]Bit 3	OUT2: Output #2 control	Can only be used in loop back mode.
UART[0x18]Bit 1	DDSR: DSR change	Can only be used in loop back mode.
UART[0x18]Bit 2	TERI: RI falling edge	Can only be used in loop back mode.
UART[0x18]Bit 3	DDCD: DCD change	Can only be used in loop back mode.
UART[0x18]Bit 5	DSR: DSR status	Can only be used in loop back mode.
UART[0x18]Bit 6	RI: RI status	Can only be used in loop back mode.
UART[0x18]Bit 7	DCD: DCD status	Can only be used in loop back mode.

The following register settings are present in the 16550, but are not present in this UART.

Offset Address	Register Bit Name	Limitation
UART[0x0C]Bit 5	Sticky parity	This bit is always unusable.

The following register settings are not present in the 16550, but are present in the UART. These bits must not be used when creating 16550-compatible software.

Offset Address	Register Bit Name	Limitation
UART[0x04]Bit 7	EPTBEI: Programmable Transmit Holding Register empty interrupt request enable	Not compatible with the 16550.
UART[0x08]Bit [5:4]	XMITT[1:0]: Transmit data trigger level setting	Not compatible with the 16550.
UART[0x10]Bit 5	AFCE: Auto flow control enable	Not compatible with the 16550.

17. UART LITE

17.1 Overview

This block provides an asynchronous data transfer interface compatible with the industry standard, 16550. It converts parallel data from the CPU into serial data for transmission to peripheral devices and, going in the opposite direction, serial data received from peripheral devices into parallel data.

This block strips this interface down to the bare minimum functionality required for a debugging console.

- Fixed frame format: 8 data bits, 1 stop bit, no parity
- Single-byte buffers only in both directions
- Overrun error, framing error, and break detection interrupt requests
- Receive ready and Transmit Holding Register empty interrupt requests
- Divisor settings of 0 to 65535

17.2 Block Diagram

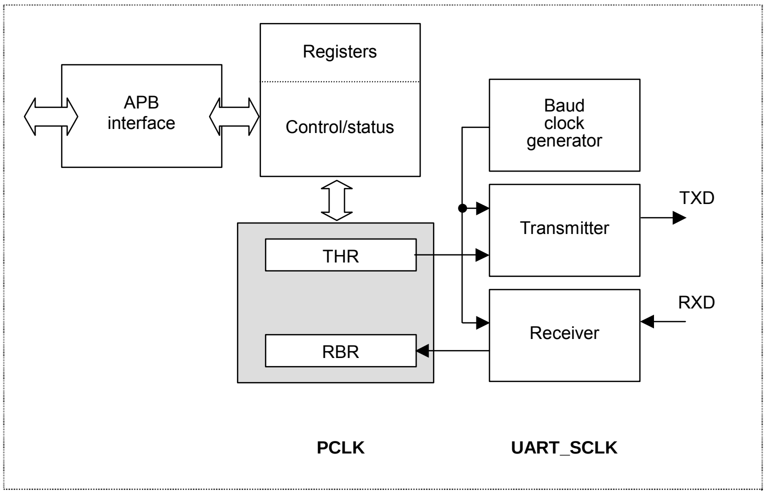


Fig.17.1 UART Lite Block Diagram

17.3 External Pins

This block interacts with the following external pins.

Pin Name	I/O	Pin Function	Multiplexed Pin*
TXD1	Output	Transmit data	GPIOA2/SPI_SS
RXD1	Input	Receive data	GPIOA3/SPI_SCLK

Note*: These external pins are multiplexed with GPIO pins and other function pins, so specify "non-GPIO function #2" in the GPIO Pin Function Register to configure them for this function.

17.4 Registers

The default base address for these registers is 0xFFFF_6000.
In the absence of any indication to the contrary, register bits not labeled reserved all is set as “0.”

17.4.1 Register List

Table 17.1 UART Lite Register List (Base Address: 0xFFFF_6000)

Address Offset	DLAB	Register Name	Abbreviation	Default Value	R/W	Data Access Size (Bits)
0x00	0	Receive Buffer Register	RBR	0x00	RO	8/16/32
0x00	0	Transmit Holding Register	THR	—	WO	8/16/32
0x00	1	Divisor Latch LSB Register	DLL	0x00	R/W	8/16/32
0x04	0	Interrupt Enable Register	IER	0x00	R/W	8/16/32
0x04	1	Divisor Latch MSB Register	DLM	0x00	R/W	8/16/32
0x08	—	Interrupt Identify Register	IIR	0x01	RO	8/16/32
0x0C	—	Line Control Register	LCR	0x03	R/W	8/16/32
0x14	—	Line Status Register	LSR	0x00	RO	8/16/32
0x30	—	Test Status 2 Register	TS2	0x00	RO	8/16/32
0x3C	—	Test Status 3 Register	TS3	0x00	RO	8/16/32

* This block supports all bus access widths (8, 16, and 32 bits), but only at 32-bit memory boundaries.

17.4.2 Important Notes on Register Access

Accessing the gaps—byte access to offset 01h, for example—between these 8-bit control registers does not yield reliable results. Always access only the specified word boundary offsets.

The test registers (30h to 3Ch) are for debugging the block itself. Do not use them for other purposes because specifications are subject to change.

17.4.3 Detailed Register Descriptions

Receive Buffer Register (RBR)
UARTL[0x00] DLAB [0] Default = 0x00 Read Only

Serial Received Data (RBR[7:0])							
7	6	5	4	3	2	1	0

Bits 7 to 0: **Serial Received Data (RBR [7:0])**

When DLAB (bit 7 in the Line Control Register at offset 0x0C) is “0,” reads from this address access the receive buffer (RBR), returning a byte of data received via the serial port.

Note, however, that this data is only valid when this block’s data ready bit (bit 0 in the Line Status Register at offset 0x14) is “1.”

If there already is data in the receive buffer when the next data arrives, the newly arrived data overwrites the data in the receive buffer.

Transmit Holding Register (THR)
UARTL[0x00] DLAB [0] Default = — Write Only

Serial Transmit Data (THR [7:0])							
7	6	5	4	3	2	1	0

Bits 7 to 0: **Serial Transmit Data (THR [7:0])**

When DLAB (bit 7 in the Line Control Register at offset 0x0C) is “0,” writes to this address access the transmit buffer (THR).

Divisor Latch LSB Register (DLL)
UARTL[0x00] DLAB [1] Default = 0x00 Read/Write

Divisor Latch LSB (DL[7:0])							
7	6	5	4	3	2	1	0

Bits 7 to 0: **Divisor Latch LSB (DL [7:0])**

Divisor latch lower half.

Interrupt Enable Register (IER)
UARTL[0x04] DLAB [0] Default = 0x00 Read/Write

Reserved (0)					Receive Line Status Interrupt Enable (ELSI)	Transmit Holding Register Empty Interrupt Enable (ETBEI)	Ready to Receive Data Interrupt Enable (ERBFI)
7	6	5	4	3	2	1	0

When DLAB (bit 7 in the Line Control Register at offset 0x0C) is “0,” this address accesses the Interrupt Enable Register (IER) controlling interrupt requests from three sources.

Bits 7 to 3: **Reserved (0)**

Bit 2: **ELSI**

Receive Line Status Interrupt Enable

0: Disable
1: Enable

Bit 1: **ETBEI**

Transmit Holding Register Empty Interrupt Enable

0: Disable
1: Enable

Bit 0: **ERBFI**

Ready to Receive Data Interrupt Enable

0: Disable
1: Enable

For further details on interrupt request sources, see Table 17.2.

Divisor Latch MSB Register (DLM)
UARTL[0x04] DLAB [1] Default = 0x00 Read/Write

Divisor Latch MSB (DL[15:8])							
7	6	5	4	3	2	1	0

Bits 7 to 0: **Divisor Latch MSB (DL [15:8])**

Divisor latch upper half.

When DLAB (bit 7 in the Line Control Register at offset 0x0C) is “1,” these two registers together specify the frequency divisor for deriving the baud rate from the source clock signal. Specify the lower half (LSB) in UARTL[0x00] and the upper half (MSB) in UARTL[0x04].
The following is the formula for the baud rate.

$$\text{baud rate} = \text{input clock} \div DL[15:0] \div 16$$

Table 17.3 gives the relationship between baud rate and frequency divisor for a 24 MHz source clock signal.

Interrupt Identifier Register (IIR)
UARTL[0x08] Default = 0x01 Read Only

Reserved				Interrupt ID (IID [3:0])			
7	6	5	4	3	2	1	0

Bits 3 to 0: **IID [3:0]**

Interrupt ID

This field identifies the source for the current interrupt request.

Table 17.2 UART Interrupt

IID [3:0]	Interrupt Type	Source	IID Reset Procedure	Priority Level
0001	None	None	n/a	n/a
0110	Receive line status interrupt	• Overrun error, framing error, break received	• Read Line Status Register.	1 (Highest)
0100	Ready to receive data interrupt	• Ready to receive data	• Read Line Status Register.	2
0010	Transmit Holding Register empty interrupt	• Transmit Holding Register empty	• Read Interrupt Identifier Register • Write to the Transmit Holding Register.	3 (Lowest)

Line Control Register (LCR)							
UARTL[0x0C]	Default = 0x03						Read/Write
DLAB	Break Control (SBRK)	Reserved (0)		Parity Enable (PEN)	Number of Stop Bits (STB)	Word Length (WLS[1:0])	
7	6	5	4	3	2	1	0

Bit 7 (DLAB): **Divisor Latch Access Bit**

0: Disable
1: Enable

Setting this bit to "1" switches UARTL[0x00] and UARTL[0x04] accesses to the Divisor Latch LSB/MSB registers instead of the Receive Buffer (RBR), Transmit Holding (THR), and Interrupt Request Enable (IER) Registers.

Bit 6 (SBRK): **Break Control**

0: Normal output
1: Break signal output

Setting this bit to "1" drives serial output pins at Low level until the software resets it to "0."

Bits 5 to 4: **Reserved (0)**

Bit 3 (PEN): **Parity Enable (Read Only)**

"1" in this bit indicates that parity checking for received data and the addition of parity bits to transmit data is disabled. This function is invalid in this UARTL.

0: Disable (fixed)

Bit 2 (STB): **Number of Stop Bits (Read Only)**

0: 1 bit (fixed)

This field indicates the number of stop bits added to transmit data.

Bits 1 to 0 (WLS): **Word Length [1:0] (Read Only)**

11: 8 (fixed)

This field indicates the number of data bits in the character. This count does not include a stop bit added at the end.

This setting is fixed at 8 bits, so do not change from the default.

Line Status Register (LSR) UARTL[0x14] Default = 0x00							Read Only
Reserved	Transmit Empty (TEMT)	Transmit Holding Register Empty (THRE)	Break Interrupt (BI)	Framing Error (FE)	Reserved	Overrun Error (OE)	Data Ready (DR)
7	6	5	4	3	2	1	0

Bit 7 (RSV): **Reserved**

Bit 6 (TEMT): **Transmit Empty**

0: Not empty
1: Empty

"1" in this bit indicates that there is no data to transmit from the Transmit Shift Register and Transmit Holding Register.

Writing transmit data returns this bit to "0."

Bit 5 (THRE): **Transmit Holding Register Empty**

0: Not empty
1: Empty

This bit goes to "1" only when regular transmit operation empties the Transmit Holding Register. Resets or other operations do not affect this bit.

Writing transmit data returns this bit to "0."

Bit 4 (BI): **Break Interrupt**

0: There is no interrupt request
1: There is an interrupt request

"1" in this bit indicates that there is a break interrupt request. This bit goes to "1" when the input line remains at Low level for the equivalent interval to one character during a receive operation.

Enabling receive line status interrupt requests causes this flag to trigger interrupt requests.

Reading this register resets this bit to "0."

Bit 3 (FE): **Framing Error**

0: No error
1: Error detected

"1" in this bit indicates that there has been a framing error—that is, when the received data did not have a valid stop bit.

Enabling receive line status interrupt requests causes this flag to trigger interrupt requests.

Reading this register resets this bit to "0."

Bit 2 (RSV): **Reserved**

Bit 1 (OE): **Overrun Error**

0: No error
1: Error detected

"1" in this bit indicates an overrun, arrival of new data byte with no place to store it because the software has not read the current data in the Receive Buffer Register (RBR) yet.

Enabling receive line status interrupt requests causes this flag to trigger interrupt requests.

Reading this register resets this bit to "0."

Bit 0 (DR): **Data Ready**

0: No data available
1: Data available

"1" in this bit indicates that there is a valid data in the receive buffer.

Enabling ready to receive data interrupt requests causes this flag to trigger interrupt requests.

Reading the receive buffer resets this bit to "0."

Test Status 2 Register (TS2) UARTL[0x30] Default = 0x00							Read Only
n/a			BAUDOUT Status	n/a			
7	6	5	4	3	2	1	0

Bit 4: **BAUDOUT Status**

0: Low level
1: High level

This indicates the output status for the BAUDOUT signal from UART circuit.

Test Status 3 Register (TS3) UARTL[0x3C] Default = 0x00							Read Only
n/a							INTR Status
7	6	5	4	3	2	1	0

Bit 0: **INTR Status**

0: Low level (inactive)
1: High level (active)

This indicates the output status for the INTR signal from UART circuit.

17.4.4 Sample Baud Rate Settings

The following formula determines the clock frequency divisor for the baud rate setting.

$$\text{frequency divisor} = \text{input clock frequency (Hz)} \div \text{baud rate (bps)} \div 16$$

The following Table gives the relationship between baud rate and frequency divisor for a 24 MHz source clock signal.

Table 17.3 Baud Rate and Frequency Divisor

Baud Rate	**Theoretical 16 clock cycles**	**24.00256 MHz UARTL Source Clock**		
		Frequency Divisor for 16 clock cycles	**Deviation (%)**	**Actual 16 clock cycles**
300	4800	5000	0.01	4800.5
600	9600	2500	0.01	9601.0
1200	19200	1250	0.01	19202.0
2400	38400	625	0.01	38404.1
4800	76800	312	0.17	76931.3
9600	153600	156	0.17	153862.6
14400	230400	104	0.17	230793.8
19200	307200	78	0.17	307725.1
28800	460800	52	0.17	461587.7
38400	614400	39	0.17	615450.3
57600	921600	26	0.17	923175.4
115200	1843200	13	0.17	1846350.8
125000	2000000	12	0.01	2000213.3
250000	4000000	6	0.01	4000426.7
500000	8000000	3	0.01	8000853.3
750000	12000000	2	0.01	12001280.0
1500000	24000000	1	0.01	24002560.0

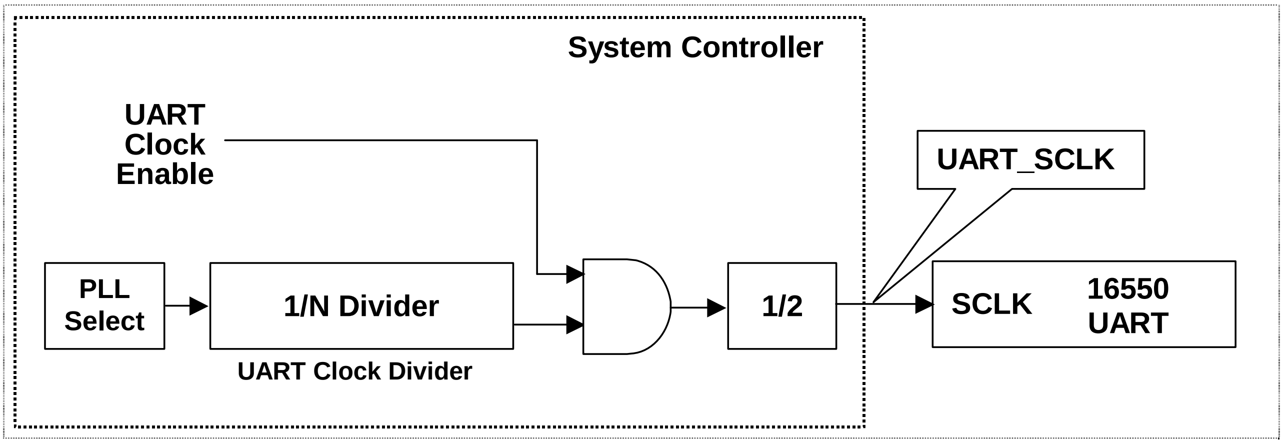


Fig.17.2 UARTL Clock Signals

17.5 Usage Limitations

As the Lite in the name indicates, this block deliberates limits functionality, so sometimes cannot handle source code written with full 16550 compatibility in mind. The lack of FIFO in particular makes higher baud rates impractical.

A debugging console can automatically determine whether it is communicating with UART Lite or a 16550-compatible UART with FIFO by writing “1” to UARTL[0x08] bit 0 and checking whether UARTL[0x08] bits 7 to 6 remain at “00” or go to “11” respectively.

18. I^{2}C SINGLE MASTER CORE MODULE (I2C)

18.1 Overview

18.1.1 Master Mode

- Support for I^{2}C bus I^{2}C single master mode
- No support for I^{2}C bus I^{2}C multi master mode
- Support for multiple slave devices on the I^{2}C bus
- Transmit and receive buffers (TBUF and RBUF) to mitigate the read/write timing by software other than I^{2}C data transfer shift registers
- Status register for reporting bus errors detected
- Support for I^{2}C clock (SCL) wait

18.1.2 Slave Mode

No support for slave mode

18.2 Block Diagram

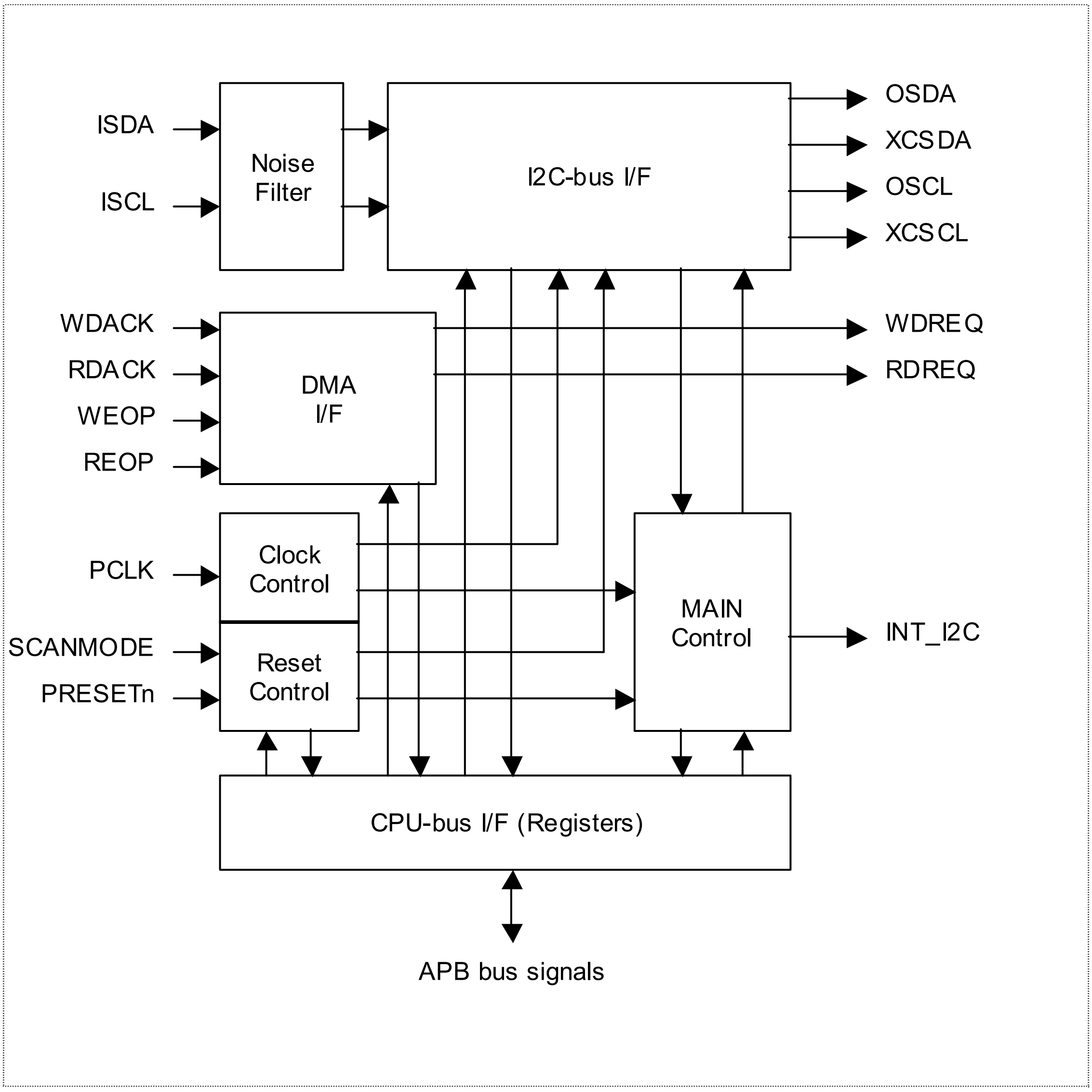


Fig.18.1 I^{2}C Block Diagram

18.3 External Pins

Pin Name	I/O	Pin Function	Multiplexed Pin*
SCL	I/O	I^{2}C clock signal	GPIOA6
SDA	I/O	I^{2}C data	GPIOA7

Note*: These external pins is set as GPIO operation, so specify “non-GPIO function #1” in the GPIO Pin Function Register to configure them for this function.

18.4 Registers

18.4.1 Register List

The base address for these registers is 0xFFFE_D000. The offsets in the following Table are relative to that address.

Table 18.1 I^{2}C Register List (Base Address: 0xFFFE_D000)

Address Offset	Register Name	Default Value	R/W	Data Access Size*1 (Bits)
0x00	I^{2}C Transmit Data Register	0000 0000b	R/W	8 (16/32)
0x04	I^{2}C Receive Data Register	0000 0000b	RO	8 (16/32)
0x08	I^{2}C Control Register	0000 0000b	R/W	8 (16/32)
0x0C	I^{2}C Bus Status Register	00xx 0000b *2	RO	8 (16/32)
0x10	I^{2}C Error Status Register	0000 0000b	RO	8 (16/32)
0x14	I^{2}C Interrupt Control/Status Register	0000 0000b	R/W	8 (16/32)
0x18	I^{2}C-Bus Sample Clock Frequency Divisor Register	0000 0000b	R/W	8 (16/32)
0x1C	I^{2}C SCL Clock Frequency Divisor Register	0000 0000b	R/W	8 (16/32)
0x20	I^{2}C I/O Control Register	0000 0000b	R/W	8 (16/32)
0x24	I^{2}C DMA Mode Register	0000 0000b	R/W	8 (16/32)
0x28	I^{2}C DMA Count Value (LSB) Register	0000 0000b	R/W	8 (16/32)
0x2C	I^{2}C DMA Count Value (MSB) Register	0000 0000b	R/W	8 (16/32)
0x30	I^{2}C DMA Status Register	0000 1000b	RO	8 (16/32)
0x34 to 0x38	Reserved	—	—	—

Notes *1: All registers in this list are 8 bits wide, so firmware normally uses 8-bit access. Only the bottom 8 bits are valid for 16- and 32-bit reads.

*2: Bits 5 (SDA) and 4 (SCL) in this register monitor the input states for external pins SDA and SCL, which specify initial values. External pull-up resistances normally set these bits both to “1,” but the external pin configurations can change these.

18.4.2 Detailed Register Descriptions

The following describes these registers in detail.

I^{2}C Transmit Data Register							
I2C[0x00]	Default = 0000 0000b						Read/Write
I^{2}C Transmit Data TD [7:0]							
7	6	5	4	3	2	1	0

Bits 7 to 0: **TD [7:0]**

I^{2}C Transmit Data

This register is an 8-bit buffer holding data to transmit over the I^{2}C-BUS.
The bit order is from the top bit (MSB) to the bottom (LSB) one. (See Figure.)

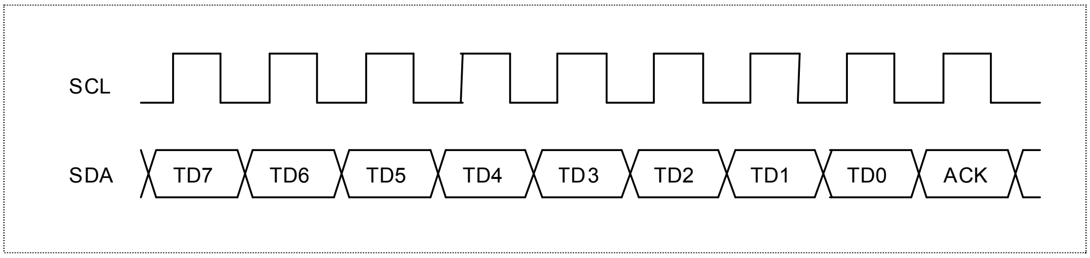


Fig.18.2 Transmit Bit Layout

I^{2}C Receive Data Register							
I2C[0x04]	Default = 0000 0000b						Read Only
I^{2}C Receive Data RD [7:0]*							
7	6	5	4	3	2	1	0

Bits 7 to 0: **RD [7:0]**

I^{2}C Receive Data

This register is an 8-bit buffer holding received data from the I^{2}C-BUS.
The bit order is from the top bit (MSB) to the bottom (LSB) one. (See Figure.)

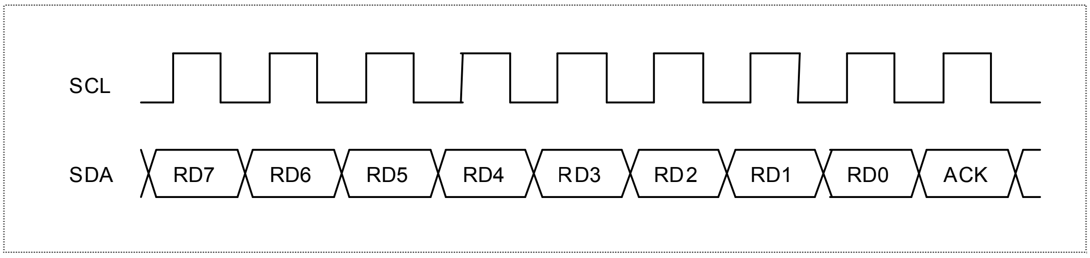


Fig.18.3 Receive Bit Layout

I^{2}C Control Register
I2C[0x08] Default = 0000 0000b Read/Write

Reserved (0)		SR	CLKW*	TACK*	TRNS [2:0]*		
7	6	5	4	3	2	1	0

* A software reset can initialize these bits.

This register controls specifying start/stop conditions, starting data transfers, etc.
Writing to the TRNS bits initiates the specified operation—but only if I^{2}C Bus Status Register (I2C[0x0C]) bit 7 (RUN) is “1.”

Bits 7 to 6: **Reserved (0)**

Bit 5 (SR): **Software Reset**
0: Release
1: Reset
Writing “1” to this bit forces a software reset, initializing this module.

Bit 4 (CLKW): **Clock Wait Mode Enable**
0: Disable
1: Enable
This bit controls the use of clock waits.

Bit3 (TACK): **Data Receive Acknowledge Enable (receive only)**
0: Disable
1: Enable
This bit specifies whether data receive operations send an acknowledge signal to the slave device.
Note: DMA transfer mode uses this bit value only for the final byte transferred. The rest of the time, send always “0.”

Bits 2 to 0: **TRNS[2:0]**
Transmit Control Command
These bits start a new operation.
001: I^{2}C start condition
010: I^{2}C stop condition
011: I^{2}C data receive
100: I^{2}C data transfer
101, 110: Do not use
000, 111: Clear error flags in Error Status Register

I^{2}C Bus Status Register I2C[0x0C] Default = 00xx 0000b							Read Only
RUN* 7	Reserved 6	SDA 5	SCL 4	Using* 3	Busy* 2	Error* 1	Finish* 0

Notes *: A software reset can initialize this field.
x: Unknown binary digit

This register indicates the I^{2}C-BUS status.

Bit 7 (RUN): **Command Execution Status**

0: Idle
1: Executing

This bit indicates whether a command is executing.

Bit 6: **Reserved**

Bits 5 to 4: **SDA and SCL**

I^{2}C-BUS monitors

These monitor the SDA and SCL signal states. External pull-up resistances normally fix these inputs at High level.

Bit 3 (Using): **Using I^{2}C-BUS**

0: No
1: Yes

This bit indicates whether the I2C single master is using the I^{2}C-BUS.

Bit 2 (Busy): **I^{2}C-BUS Busy**

0: Free
1: Busy

This bit indicates whether the I2C bus is busy (or free).

Bit 1 (Error): **Error Status**

0: No error
1: Error detected

This bit indicates whether there has been an error.
Writing to the I^{2}C Control Register resets this bit to “0.”

Bit 0 (Finish): **Command Execution Complete**

0: No command or command executing
1: Complete

“1” in this bit indicates completion of command execution; a “0,” otherwise—either there is no command or the command is still executing.
This bit is cleared to “0” during a reset (while RESET# is at Low level), when I^{2}C Control Register bit 5 (software reset) is “1,” and when the firmware writes a valid command to I^{2}C Register Control bits 2 to 0 (TRNS).

I^{2}C Error Status Register
I2C[0x10]   Default = 0000 0000b   Read Only

Reserved			Receive Acknowledgement Error*	SCL Mismatch Error*	SDA Mismatch Error*	Stop Condition*	Start Condition*
7	6	5	4	3	2	1	0

Note *: A software reset can initialize this field.

This register indicates the error status.

Bits 7 to 5: **Reserved**

Bit 4: **Receive Acknowledgment Error**

0: None detected
1: Detected

"1" in this bit indicates an error during receive acknowledgment.

Bit 3: **SCL Mismatch Error**

0: None detected
1: Detected

"1" in this bit indicates a clock mismatch—in other words, a mismatch between ISCL and OSCL values.

Bit 2: **SDA Mismatch Error**

0: None detected
1: Detected

"1" in this bit indicates a data mismatch—in other words, a mismatch between ISDA and OSDA values.

Bit 1: **Stop Condition Detection**

0: None detected
1: Detected

"1" in this bit indicates a stop condition other than one specified by a command.

Bit 0: **Start Condition Detection**

0: None detected
1: Detected

"1" in this bit indicates a start condition other than one specified by a command.

Note: These bits are cleared to "0" during a reset (while RESET# is at Low level), when I^{2}C Control Register bit 5 (software reset) is "1," and when the firmware writes a valid command to I^{2}C Control Register bits 2 to 0 (TRNS).

I^2C Interrupt Control/Status Register I2C[0x14] Default = 0000 0000b							Read/Write
Reserved (0)				Error Interrupt Status Flag*	Command Complete Interrupt Status Flag*	Error Interrupt Enable*	Command Complete Interrupt Enable*
7	6	5	4	3	2	1	0

Note *: A software reset can initialize this field.

Bits 7 to 4: **Reserved (0)**

Bit 3: **Error Interrupt Status Flag**

0: No error

1: Error interrupt

This bit is only valid when bit 1 is “1.”

Writing “1” to this bit clears this bit.

Bit 2: **Command Complete Interrupt Status Flag**

0: Command executing or Bit 0 (Command complete interrupt enable bit) is “0”.

1: Command complete interrupt

This bit is only valid when bit 0 is “1.” It is always “0” if bit 0 (command complete interrupt request enable) is “0.”

Writing “1” to this bit clears this bit.

Bit 1: **Error Interrupt Enable**

0: Disable (mask)

1: Enable

This bit controls masking of error interrupt requests.

Bit 0: **Command Complete Interrupt Enable**

0: Disable (mask)

1: Enable

This bit controls masking of command complete interrupt requests.

I^2C-Bus Sample Clock Frequency Divisor Register I2C[0x18] Default = 0000 0000b							Read/Write
Reserved (0)				I^2C Sample Frequency Divisor Setting [3:0]			
7	6	5	4	3	2	1	0

Bits 7 to 4: **Reserved (0)**

Bits 3 to 0: **I^2C Sample Frequency Divisor Setting [3:0] (*m*)**

This field specifies the frequency divisor for deriving the I^2C-BUS sample clock from the master clock.

The following is the formula for calculating the frequency ($f_{I2Csample}$) from this setting, *m*:

$$f_{I2Csample} = f_{PCLK} \div (4 * m) \text{ [Hz]}$$

Note: The formula is different for *m* = 0:

$$f_{I2Csample} = f_{PCLK} / 2 \text{ [Hz]}$$

I^2C SCL Clock Frequency Divisor Register

I2C[0x1C] Default = 0000 0000b Read/Write

Reserved (0)					SCL Frequency Divisor Setting [2:0]		
7	6	5	4	3	2	1	0

Bits 7 to 3: **Reserved (0)**

Bits 2 to 0: **SCL Frequency Divisor Setting [2:0] (*n*)**

This field specifies the frequency divisor for deriving the SCL clock for I^2C bus transfers from the I^2C-BUS sample clock.

The following is the formula for calculating the frequency (f_{SCL}) from this setting, *n*, and *m*, the I^2C sample clock frequency divisor setting, bits 3 to 0 in the I^2C-BUS Sample Clock Frequency Divisor Register, above.

$$\begin{aligned} f_{SCL} &= f_{I2Csampling} \div (2^n * 4) \\ &= f_{PCLK} \div (4 * m) \div (2^n * 4) \\ &= f_{PCLK} \div (16 * m * 2^n) \text{ [Hz]} \end{aligned}$$

Note: The formula is different for $m = 0$:

$$\begin{aligned} f_{SCL} &= f_{PCLK} / \{ 2 * (2^n * 4) \} \\ &= f_{PCLK} \div (8 * 2^n) \text{ [Hz]} \end{aligned}$$

I^2C I/O Control Register

I2C[0x20] Default = 0001 0001b Read/Write

Reserved (0)		SDA High Drive Enable	SDA Sampling Enable	Reserved (0)		SCL High Drive Enable	SCL Sampling Enable
7	6	5	4	3	2	1	0

This register specifies the output mode and controls use of the noise filter.

Bits 7 to 6: **Reserved (0)**

Bit 5: **SDA High Drive Enable**

0: Disable
1: Enable

Setting this bit to “0” relies on the external pull-up resistance for SDA High level output. The “1” setting uses internal drive circuitry.

Bit 4: **SDA Sampling Enable**

0: 1 data sampling (fixed)
1: 2 data sampling

This field specifies the number of data samples, made with the I^2C-BUSsampling clock, as SDA input.

Note: Fix this bit at “0.”

Bits 3 to 2: **Reserved (0)**

Bit 1: **SCL High Drive Enable**

0: Disable
1: Enable

Setting this bit to “0” relies on the external pull-up resistance for SCL High level output. The “1” setting uses internal drive circuitry.

Bit 0: **SCL Sampling Enable**

0: 1 data sampling (fixed)
1: 2 data sampling

This field specifies the number of data samples, made with the I^2C-BUS sampling clock, as SCL input.

Note: Fix this bit at “0.”

I^{2}C DMA Mode Register							
I2C[0x24] Default = 0000 0000b							Read/Write
Reserved (0)						DMA_MODE [1:0]	
7	6	5	4	3	2	1	0

Bits 7 to 2: **Reserved (0)**

Bits 1 to 0: **DMA_MODE[1:0]**

DMA Mode Setting

00: DMA transfer Off
01: Single address mode
10: Dual address mode with DMA counter, not EOP.
11: Dual address mode with EOP

Do not switch DMA modes (single, dual without EOP, or dual with EOP) in the middle of a transfer. The only changes allowed are to and from the 00b (off) setting.

Burst transfers are not supported.

I^{2}C DMA Counter Value Register (LSB)							
I2C[0x28] Default = 0000 0000b							Read/Write
DMA Counter Value (LSB)							
7	6	5	4	3	2	1	0

I^{2}C DMA Counter Value Register (MSB)							
I2C[0x2C] Default = 0000 0000b							Read/Write
DMA Counter Value (MSB)							
7	6	5	4	3	2	1	0

These registers provide read/write access to the lower and upper halves of the DMA counter, respectively.

I^{2}C DMA Status Register I2C[0x30] Default = 0000 1000b							Read Only
Reserved				TBUF Empty*	RBUF Update*	RDREQ Monitor*	WDREQ Monitor*
7	6	5	4	3	2	1	0

Note *: A software reset can initialize this field.

Bits 7 to 4: **Reserved**

Bit 3: **TBUF_Empty**
Transmit buffer empty

0: Not empty
1: Empty, write enable

“1” in this bit indicates that it is safe to write to the I^{2}C transmit data buffer (TBUF); a “0,” that there is data to transmit.

Bit 2: **RBUF_Update**
Receive buffer update

0: Not update
1: Update

“1” in this bit indicates a data is updated in the I^{2}C receive data buffer (RBUF).

Bit 1: **Reserved (RDREQ Monitor)**

(RDREQ signal monitor
0: Low level
1: High level
This indicates the RDREQ pin level.)

Note: This bit changes, but in no usable or reliable fashion.

Bit 0: **Reserved (WDREQ Monitor)**

(WDREQ signal monitor
0: Low level
1: High level
This indicates the WDREQ pin level.)

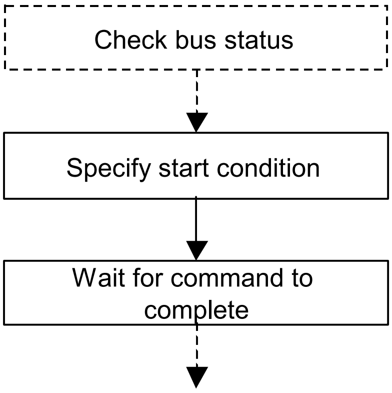
Note: This bit changes, but in no usable or reliable fashion.

18.5 Description of Operation (Sample Bus Control Commands)

The following gives examples of I^2C-BUS control using this module. Actual systems generally require status checks, error recovery, and other processing as well.

Note that the specific control procedures depend on slave device specifications.

18.5.1 Sample Start (S) Flowchart

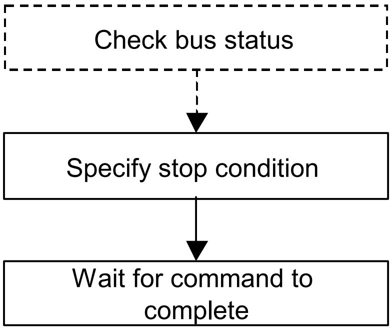


Make sure that I2C[0x0C] bit 2 is “0.”

Start condition setting:
Write “001b” to I2C[0x08] bits 2 to 0.

Wait for I2C[0x0C] bit 0 to go to “1.”

18.5.2 Sample Stop (P) Flowchart

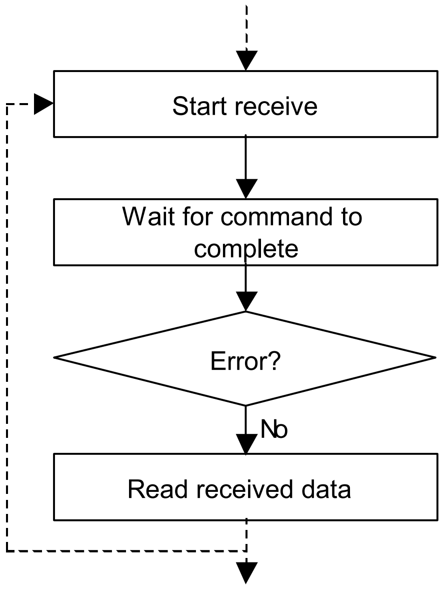


Make sure that I2C[0x0C] bit 3 is “1.”

Start condition setting:
Write “010b” to I2C[0x08] bits 2 to 0.

Wait for I2C[0x0C] bit 0 to go to “0.”

18.5.3 Sample Receive (R) Flowchart



Receive setting:
Write “011b” to I2C[0x08] bits 2 to 0.

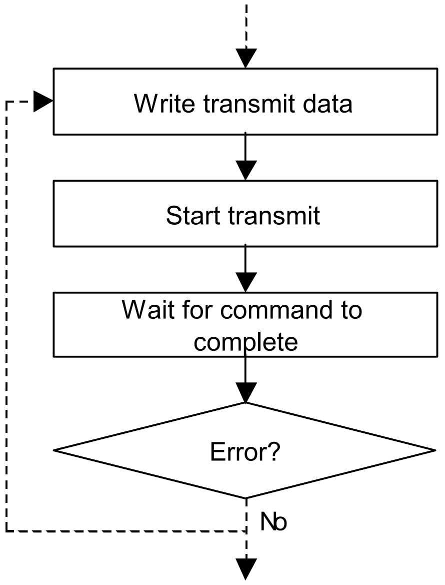
Wait for I2C[0x0C] bit 0 to go to “1.”

Check I2C[0x0C] bit 1:

A “1” in bit 1 indicates an error.

Read data received from I2C[0x04].

18.5.4 Sample Transfer (T) Flowchart



Write data to transmit in I2C[0x00].

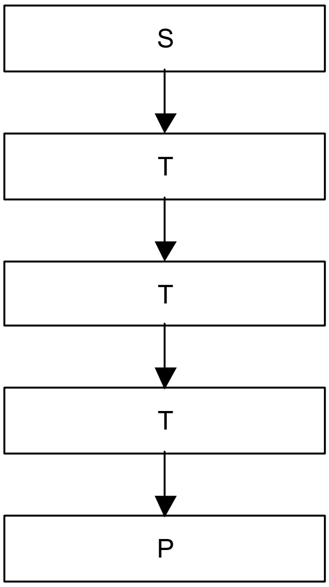
Transmit setting:
Write “100b” to I2C[0x08] bits 2 to 0.

Wait for I2C[0x0C] bit 0 to go to “1.”

Check I2C[0x0C] bit 1:
A “1” in bit 1 indicates an error.

18.5.5 Sample Sequence for Write to Slave Device

The S, P, R, and T in the Figure refer to the start, stop, receive, and transmit flowcharts above.



Start condition satisfied.

Transmit slave address and write mode.

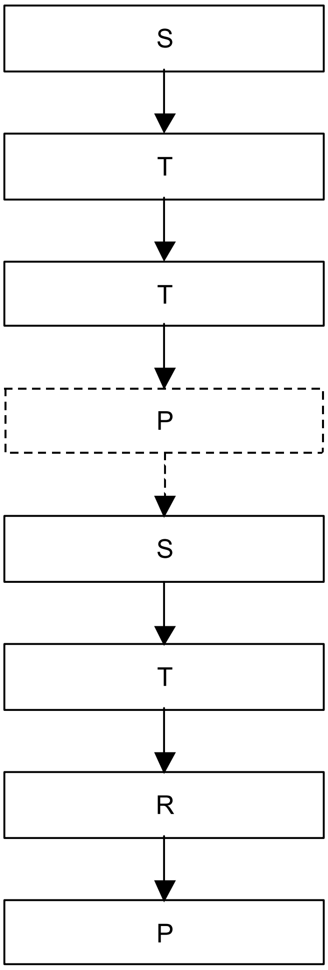
Transmit the internal address of slave device.

Transmit write data.

Stop condition satisfied.

18.5.6 Sample Sequence for Read from Slave Device

The S, P, R, and T in the Figure refer to the start, stop, receive, and transmit flowcharts above.



Start condition satisfied.

Transmit slave address and write mode.

Transmit the internal address of slave device.

Stop condition satisfied.

Stop condition satisfied.

Transmit slave address and read mode.

Receive read data.

Stop condition satisfied.

18.6 Usage Limitations

The following Table lists register bits with usage restrictions.

Bit Address	Register Bit Name	Limitations
I2C[0x30] bit 0	WDREQ signal monitor	Reserved. Do not use.
I2C[0x30] bit 1	RDREQ signal monitor	Reserved. Do not use.

19. I^{2}S (I2S)

19.1 Overview

This module, which compliant with the Philips I^{2}S standard, is mainly for audio data transfers. It supplies two channel communication with independent transmit/receive controls for simultaneously receiving and transmitting data from/to one audio device or simultaneously receiving data from two different audio devices, for example.

19.1.1 Features

This module has the following features.

- Choice of master (SCK and WS output) or slave (SCK and WS input) operation
- Choice of direction for SD pin: transmit mode (output) or receive mode (input)
- Support data widths: 16, 14, or 8 bits
- Choice of stereo or monaural
- Support frame cycles: 32 fs, 64 fs, 128 fs, or 256 fs
- Choice of 256 source clock frequency divisors from 2 to 512 (master mode only)
- DMA support
- Clock sharing—using the same clock signal for 2 channels
- FIFO over- and underflow detection
- Interrupt requests indicating six states of FIFO
- Zero output when FIFO underflows during transmission
- Monaural to stereo conversion (transmit mode only)

19.2 Block Diagram

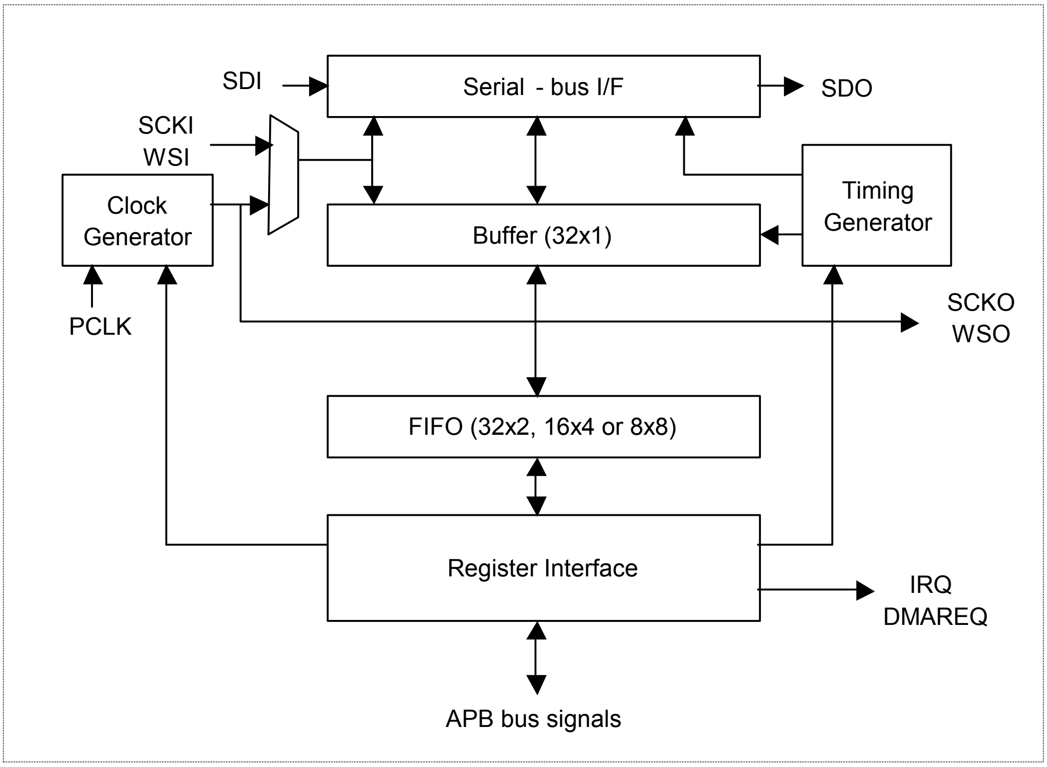


Fig.19.1 I2S Block Diagram

19.3 External Pins

This module interacts with the following external pins.

Pin Name	I/O	Pin Function	Multiplexed Pin*
I2S0_SCK	I/O	I2S0 serial clock	GPIOB1/RTS0#* or GPIOE1/CFIOWR#*
I2S0_WS	I/O	I2S0 word select	GPIOB0* or GPIOE3/CFRST*
I2S0_SD	I/O	I2S0 serial data	GPIOB2/CTS0#* or GPIOE0/CFIORD#*
I2S1_SCK	I/O	I2S1 serial clock	GPIOB6/MA22* or GPIOE6/CFDEN#*
I2S1_WS	I/O	I2S1 word select	GPIOB7/MA23* or GPIOE7/CFDDIR*
I2S1_SD	I/O	I2S1 serial data	GPIOB3/TIMER0OUT* or GPIOE5/CFSTSCHG#*

Note*: These external pins are multiplexed with GPIO pins and other function pins, so specify “non-GPIO function #2” in the GPIO Pin Function Register to configure them for this function.

19.4 Registers

19.4.1 Register List

The default base address for these registers is 0xFFFE_E000.
In the absence of any indication to the contrary, register bits not labeled reserved all is set as “0.”

Table 19.1 I2S[1:0] Register List

Address Offset	Register Name	Default Value	R/W	Data Access Size (Bits)
I2S0 Control Registers: 0xFFFE_E000				
0x00	I2S0 Control Register	0x0000	R/W	16/32
0x04	I2S0 Clock Frequency Divisor Register	0x0000	R/W	16/32
0x08	I2S0 Transfer Port Register	—	R/W	8/16/32
0x10	I2S0 Interrupt Status Register	0x0000	R/W	16/32
0x14	I2S0 Interrupt Raw Status Register	0x0009	RO	16/32
0x18	I2S0 Interrupt Enable Register	0x0000	R/W	16/32
0x1C	I2S0 Current Status Register	0x0009	RO	16/32
I2S1 Control Registers: 0xFFFE_E000				
0x40	I2S1 Control Register	0x0000	R/W	16/32
0x44	I2S1 Clock Frequency Divisor Register	0x0000	R/W	16/32
0x48	I2S1 Transfer Port Register	—	R/W	8/16/32
0x50	I2S1 Interrupt Status Register	0x0000	R/W	16/32
0x54	I2S1 Interrupt Raw Status Register	0x0009	RO	16/32
0x58	I2S1 Interrupt Enable Register	0x0000	R/W	16/32
0x5C	I2S1 Current Status Register	0x0009	RO	16/32

19.4.2 Detailed Register Descriptions

I2S[1:0] Control Registers
I2S0[0x00], I2S1[0x40] Default = 0x0000 Read/Write

n/a			CNVM2S	FRAMECYC [1:0]		CLKOUTEN	CLKSEL
15	14	13	12	11	10	9	8
SFTRST (WO)	DATAWIDTH [1:0]		MONO/ STEREO	DMAEN	TX/RX	MST/SLV	I2SEN
7	6	5	4	3	2	1	0

Bit 12: **CNVM2S**

Convert Monaural to Stereo (transmit mode only)

0: Disable
1: Enable

Setting this bit to “1” converts monaural data to stereo output—more precisely, duplicates the left channel output on the right channel.

Bits 11 to 10: **FRAMECYC [1:0]**

Frame Cycles (master mode only)

00: 32 fs
01: 64 fs
10: 128 fs
11: 256 fs

These register bits specify the number of frame cycles (32 × 2n) for master mode. Slave mode uses an even number between twice the data width and 256 fs.

Bit 9: **CLKOUTEN**

Clock Output Enable (master mode only)

0: Disable
1: Enable

This bit controls clock output.

Bit 8: **CLKSEL**

Clock Select (slave mode only)

0: Disable
1: Enable

Setting this bit to “1” uses the clock (SCK) and word select (WS) signals from the other channel. Otherwise (“0”), the interface uses the external signals from the SDI and WSI pins (slave mode) or internally generated ones (master mode).

Note: “1” in this Register bit requires Setting bit 1 in the same Register to “0” (slave). Clock sharing is not available in master mode.

Bit 7: **SFTRST**

Software Reset (Write Only)

0: n/a
1: Reset

Writing “1” to this bit clears the FIFO and shift register and resets the internal control circuitry. It does not reset the master mode clock (SCK) generator. Neither does it reset the master mode word select (WS) generator for receiving. For transmitting, it resets the WS output to High level.

Bits 6 to 5: **DATAWIDTH [1:0]**

Data Width in Bits

00: 16 bits
01: 14 bits
10: 8 bits
11: Reserved

This field specifies the data width in bits.

Bit 4: **MONO/ STEREO**

Data Type Select

0: Stereo

1: Monaural

This bit specifies the data format: stereo or monaural.

Bit 3: **DMAEN**

DMA Enable

0: DMA Disable

1: DMA Enable

Setting this bit to “1” issues DMA requests to DMAC1. The “0” setting does not issue DMA request.

Bit 2: **TX/RX**

Transfer Mode Select

0: Receive mode (Data input)

1: Transmit mode (Data output)

This bit specifies the transfer mode (data direction).

Bit 1: **MST/SLV**

Master/Slave Select

0: Slave mode

1: Master mode

Setting this bit to “1” specifies I2S master mode, where the interface provides SCK and WS output. The “0” (slave) setting uses external inputs for these two signals.

Bit 0: **I2SEN**

I2S Module Enable

0: Disable

1: Enable

Setting this bit to “1” enables I2S module.

I2S[1:0] Clock Frequency Divisors Register

I2S0[0x04], I2S1[0x44] Default = 0x0000 Read/Write

n/a							
15	14	13	12	11	10	9	8
CLKDIV [7:0]							
7	6	5	4	3	2	1	0

Bits 7 to 0: **CLKDIV [7:0]**

Clock Frequency Divisor (master mode only)

This field specifies the frequency divisor for deriving the master mode output clock from the source clock.

clock frequency divisor = (CLKDIV + 1) × 2

The following is thus the formula for calculating the sampling frequency.

sampling frequency = source clock frequency ÷ (clock frequency divisor × frame cycles)

Note: For this device, the source clock frequency is the same as the system clock frequency.

I2S[1:0] Transfer Port Register

I2S0[0x08], I2S1[0x48] Default = — Read/Write

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

Bits 31 to 0: TXD/RXD [31:0]

Transfer Port

Receive mode: Reads the I^2S data from the FIFO.
Writes: n/a

Transmit mode: Reads: n/a
Writes the I^2S data to the FIFO.

The data layout in this register depends on the data width (16, 14, or 8 bits) and the type (stereo or monaural).

16-bit stereo:	TXD/RXD [31:16]	Right channel data
	TXD/RXD [15:0]	Left channel data
14-bit stereo:	TXD/RXD [31:30]	Padding data (“0”)
	TXD/RXD [29:16]	Right channel data
	TXD/RXD [15:14]	Padding data (“0”)
	TXD/RXD [13:0]	Left channel data
8-bit stereo:	TXD/RXD [31:16]	Invalid data
	TXD/RXD [15:8]	Right channel data
	TXD/RXD [7:0]	Left channel data
16-bit monaural:	TXD/RXD [31:16]	Invalid data
	TXD/RXD [15:0]	Monaural data
14-bit monaural:	TXD/RXD [31:16]	Invalid data
	TXD/RXD [15:14]	Padding data (“0”)
	TXD/RXD [13:0]	Monaural data
16-bit monaural:	TXD/RXD [31:8]	Invalid data
	TXD/RXD [7:0]	Monaural data

Note: Register reads and writes must use an access width large enough to cover a complete set of data. Reading 16-bit stereo data, for example, requires 32-bit access to simultaneously read both right and left data because using 16-bit access reads only the left data, discards the right channel data by next data access.

I2S[1:0] Interrupt Status Registers							
I2S0[0x10], I2S1[0x50]		Default = 0x0000					Read/Write
n/a							
15	14	13	12	11	10	9	8
n/a		OVERFLOWFLG	UNDERFLOWFLG	NOTFULLFLG	NOTEMPTYFLG	FULLFLG	EMPTYFLG
7	6	5	4	3	2	1	0

This register gives the result of the bitwise logical AND of the Raw Interrupt Status Register (I2S0[0x14] or I2S1[0x54]) and the Interrupt Enable Register (I2S0[0x18] or I2S1[0x58]).

Bit 5: **OVERFLOWFLG - FIFO Overflow Interrupt Flag**

Indicates whether or not a FIFO overflow has ever occurred.

Write 0: n/a
Write 1: Tries to clear the flag.
Read 0: Indicates either that this flag is not enabled or that the FIFO has never overflowed.
Read 1: Indicates that the FIFO has overflowed at least once.

Whether or not this flag is cleared when a “1” is written differs depending on mode.

Transmit mode: The flag will always be cleared. All data written after the overflow is lost, but the FIFO still contains the preceding data.

Receive mode: The attempt succeeds if it follows at least one I2S clock cycle with the FIFO not full. It also succeeds after a software reset.

Bit 4: **UNDERFLOWFLG - FIFO Underflow Interrupt Flag**

Indicates whether or not a FIFO underflow has ever occurred.

Write 0: n/a

Write 1: Tries to clear the flag.

Read 0: Indicates either that this flag is not enabled or that a FIFO underflow has never occurred.

Read 1: Indicates that a FIFO underflow has occurred at least once.

Whether or not this flag is cleared when a “1” is written differs depending on the mode.

Transmit mode: The attempt succeeds if it follows at least one I2S clock cycle after data was written to the FIFO. It also succeeds after a software reset.

Receive mode: The flag will always be cleared. All received data during the underflow is not guaranteed.

Bit 3: **NOTFULLFLG - FIFO Not Full Interrupt Flag**

Indicates whether or not the FIFO has ever been in the not full state.

Write 0: n/a

Write 1: Tries to clear the flag.

Read 0: Indicates either that this flag is not enabled or that the FIFO has never been in the not full state.

Read 1: Indicates that the FIFO has been in the not full state at least once.

Whether or not this flag is cleared when a “1” is written differs depending on the FIFO state.

FIFO full state: The flag will be cleared.

FIFO not full state: The flag will not be cleared.

Bit 2: **NOTEMPTYFLG - FIFO Not Empty Interrupt Flag**

Indicates whether or not the FIFO has ever been in the not empty state.

Write 0: n/a

Write 1: Tries to clear the flag.

Read 0: Indicates either that this flag is not enabled or that the FIFO has never been in the not empty state.

Read 1: Indicates that the FIFO has been in the not empty state at least once.

Whether or not this flag is cleared when a “1” is written differs depending on the FIFO state.

FIFO empty state: The flag will be cleared.

FIFO not empty state: The flag will not be cleared.

Bit 1: **FULLFLG - FIFO Full Interrupt Flag**

Indicates whether or not the FIFO has ever been in the full state.

Write 0: n/a

Write 1: Tries to clear the flag.

Read 0: Indicates either that this flag is not enabled or that the FIFO has never been in the full state.

Read 1: Indicates that the FIFO has been in the full state at least once.

Whether or not this flag is cleared when a “1” is written differs depending on the FIFO state.

FIFO full state: The flag will not be cleared.

FIFO not full state: The flag will be cleared.

Bit 0: **EMPTYFLG - FIFO Empty Interrupt Flag**

Indicates whether or not the FIFO has ever been in the empty state.

Write 0: n/a

Write 1: Tries to clear the flag.

Read 0: Indicates either that this flag is not enabled or that the FIFO has never been in the empty state.

Read 1: Indicates that the FIFO has been in the empty state at least once.

Whether or not this flag is cleared when a “1” is written differs depending on the FIFO state.

FIFO empty state: The flag will not be cleared.

FIFO not empty state: The flag will be cleared.

I2S[1:0] Interrupt Raw Status Registers
I2S0[0x14], I2S1[0x54] Default = 0x0009 Read Only

n/a							
15	14	13	12	11	10	9	8
n/a		RAWOVERFLOWFLG	RAWUNDERFLOWFLG	RAWNOTFULLFLG	RAWNOTEMPTYFLG	RAWFULLFLG	RAWEMPTYFLG
7	6	5	4	3	2	1	0

“1” in a bit indicates that the FIFO has satisfied the corresponding trigger condition (overflow, underflow, not full, not empty, full, or empty) at least once; a “0,” otherwise.

Bit 5: **RAWOVERFLOWFLG - Raw FIFO Overflow Interrupt Flag**
Indicates whether or not a FIFO overflow has ever occurred.
0: A FIFO overflow has never occurred.
1: A FIFO overflow has occurred at least once

Bit 4: **RAWUNDERFLOWFLG - Raw FIFO Underflow Interrupt Flag**
Indicates whether or not a FIFO underflow has ever occurred.
0: A FIFO underflow has never occurred.
1: A FIFO underflow has occurred at least once

Bit 3: **RAWNOTFULLFLG - Raw FIFO Not Full Interrupt Flag**
Indicates whether or not the FIFO has ever been in the not full state.
0: The FIFO has never been in the not full state.
1: The FIFO has been in the not full state at least once.

Bit 2: **RAWNOTEMPTYFLG - Raw FIFO Not Empty Interrupt Flag**
Indicates whether or not the FIFO has ever been in the not empty state.
0: The FIFO has never been in the not empty state.
1: The FIFO has been in the not empty state at least once.

Bit 1: **RAWFULLFLG - Raw FIFO Full Interrupt Flag**
Indicates whether or not the FIFO has ever been in the full state.
0: The FIFO has never been in the full state.
1: The FIFO has been in the full state at least once.

Bit 0: **RAWEMPTYFLG - Raw FIFO Empty Interrupt Flag**
Indicates whether or not the FIFO has ever been in the empty state.
0: The FIFO has never been in the empty state.
1: The FIFO has been in the empty state at least once.

I2S[1:0] Interrupt Enable Registers							
I2S0[0x18], I2S1[0x58]		Default = 0x0000					Read/Write
n/a							
15	14	13	12	11	10	9	8
n/a		OVERFLOW IRQEN	UNDERFLOW IRQEN	NOTFULL IRQEN	NOTEMPTY IRQEN	FULL IRQEN	EMPTY IRQEN
7	6	5	4	3	2	1	0

These bits control whether the corresponding trigger condition (overflow, underflow, not full, not empty, full, or empty), even once, actually triggers an interrupt request.

Bit 5: **OVERFLOWIRQEN**
FIFO Overflow Interrupt Enable
0: Disable (mask)
1: Enable

Bit 4: **UNDERFLOWIRQEN**
FIFO Underflow Interrupt Enable
0: Disable (mask)
1: Enable

Bit 3: **NOTFULLIRQEN**
FIFO Not Full Interrupt Enable
0: Disable (mask)
1: Enable

Bit 2: **NOTEMPTYIRQEN**
FIFO Not Empty Interrupt Enable
0: Disable (mask)
1: Enable

Bit 1: **FULLIRQEN**
FIFO Full Interrupt Enable
0: Disable (mask)
1: Enable

Bit 0: **EMPTYIRQEN**
FIFO Empty Interrupt Enable
0: Disable (mask)
1: Enable

I2S[1:0] Current Status Registers
I2S0[0x1C], I2S1[0x5C] Default = 0x0009 Read Only

FIFOWPNTR [3:0]				FIFORPNTR [3:0]			
15	14	13	12	11	10	9	8
DMASTS	n/a			NOTFULLSTS	NOTEMPTYSTS	FULLSTS	EMPTYSTS
7	6	5	4	3	2	1	0

Bits 15 to 12: **FIFOWPNTR [3:0]**
FIFO Write Pointer

Bits 11 to 8: **FIFORPNTR [3:0]**
FIFO Read Pointer

These fields give the FIFO's current write and read pointers (0x0 to 0xF).
The number of valid bits depends on the data size and data type (stereo or monaural).

lowest bit:	16-bit stereo, 14-bit stereo
lowest two bits:	16-bit monaural, 14-bit monaural, 8-bit monaural
lowest three bits:	8-bit monaural

"1" in bits 3 to 0 indicates that the FIFO currently satisfies the corresponding trigger condition (not full, not empty, full, or empty).

Bit 7: **DMASTS**
DMA Status
0: No DMA request
1: DMA request
This bit indicates whether there is currently a DMA request.

Bit 3: **NOTFULLSTS**
FIFO Not Full Current Status
0: FIFO full
1: FIFO not full

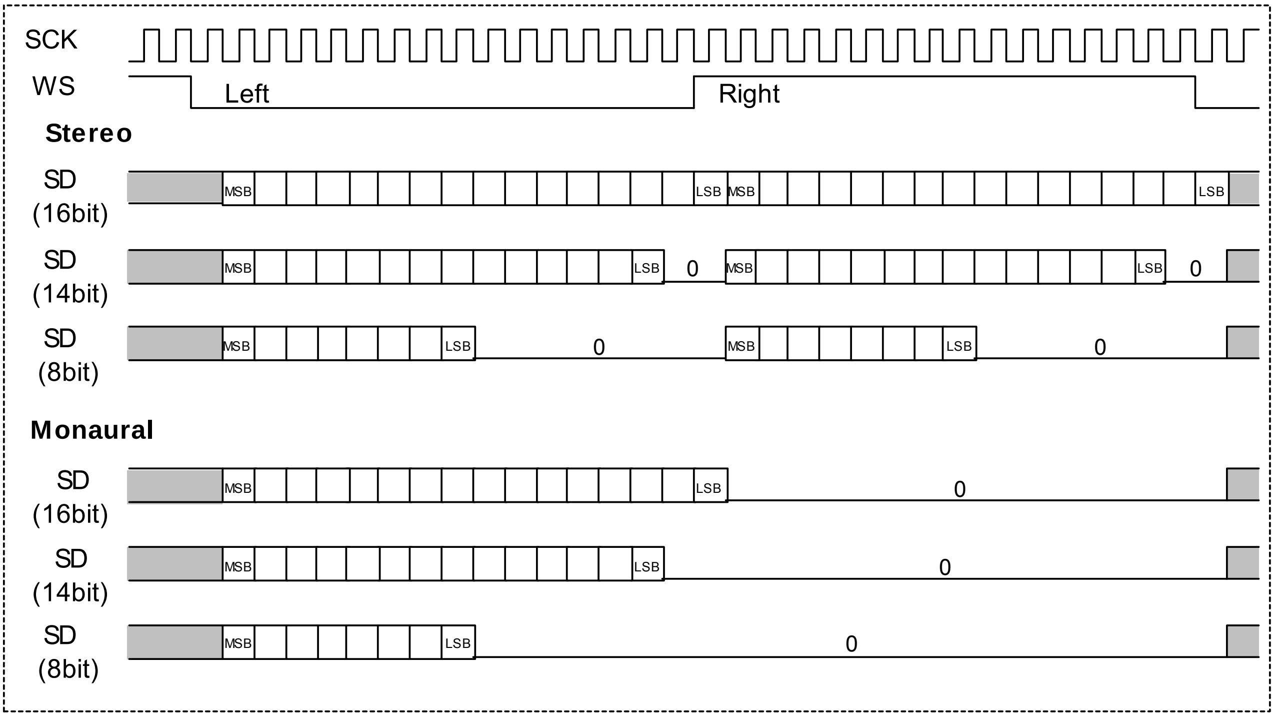
Bit 2: **NOTEMPTYSTS**
FIFO Not Empty Current Status
0: FIFO empty
1: FIFO not empty

Bit 1: **FULLSTS**
FIFO Full Current Status
0: FIFO not full
1: FIFO full

Bit 0: **EMPTYSTS**
FIFO Empty Current Status
0: FIFO not empty
1: FIFO empty
"1" in this bit indicates that the FIFO is currently empty.

19.5 Functional Description

19.5.1 I^{2}S Timing Chart (32 fs)



19.5.2 Data Width and Number of FIFO Stages

The data layout depends on the data width (16, 14, or 8 bits) and the type (stereo or monaural).

16-bit stereo:	TXD/RXD [31:16] Right channel data
	TXD/RXD [15:0] Left channel data
14-bit stereo:	TXD/RXD [31:30] Padding data ("0")
	TXD/RXD [29:16]Right channel data
	TXD/RXD [15:14] Padding data ("0")
	TXD/RXD [13:0] Left channel data
8-bit stereo:	TXD/RXD [31:16] Invalid data
	TXD/RXD [15:8]Right channel data
	TXD/RXD [7:0] Left channel data
16-bit monaural:	TXD/RXD [31:16] Invalid data
	TXD/RXD [15:0] Monaural data
14-bit monaural:	TXD/RXD [31:16] Invalid data
	TXD/RXD [15:14] Padding data ("0")
	TXD/RXD [13:0] Monaural data
8-bit monaural:	TXD/RXD [31:8] Invalid data
	TXD/RXD [7:0] Monaural data

Note: Register reads and writes must use an access width large enough to cover a complete set of data. Reading 16-bit stereo data, for example, requires 32-bit access to simultaneously read both right and left data because using 16-bit access reads only the left data discards the right channel data by next data access.

The depth of FIFO depends on the data width (16, 14, or 8 bits) and the type (stereo or monaural).

FIFO Depth	Data Condition
2	16-bit stereo, 14-bit stereo
4	16-bit monaural, 14-bit monaural, 8-bit stereo
8	8-bit monaural

19.5.3 DMA Transfers

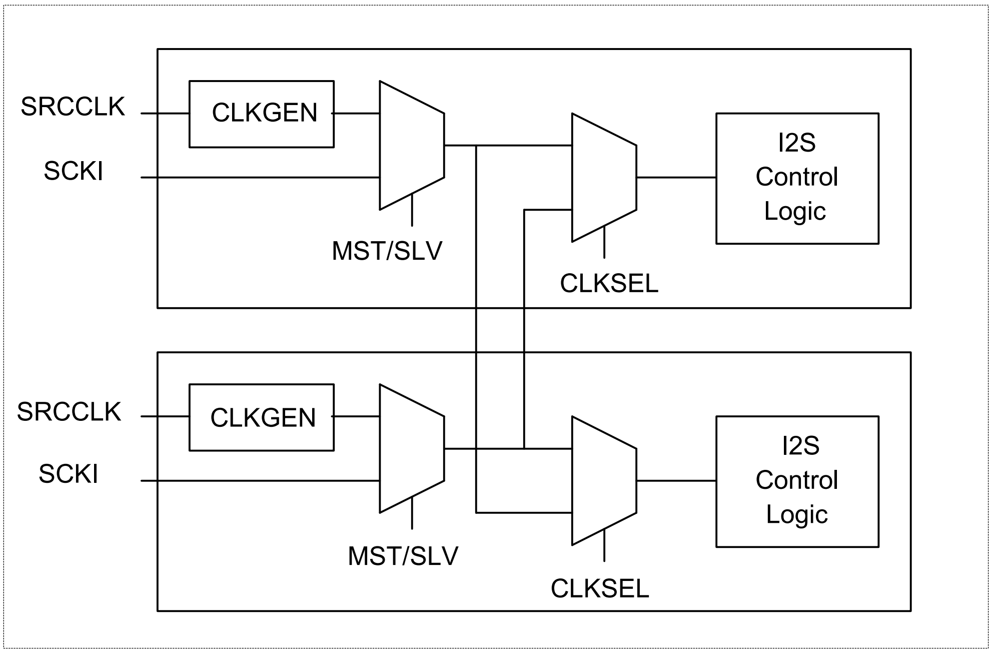
Setting I2S0[0x00] or I2S1[0x40] bit 3 to "1" enables DMA transfers with DMAC1 for the corresponding I^2S interface.

This module asserts the DMA request signal line as appropriate for the transfer direction: when the FIFO is not full (transmit mode) or not empty (receive mode).

As a measure to block unnecessary activity by the corresponding I^2S interface, FIFO overflow negates the DMA request.

19.5.4 Sharing Clock Signals

Setting I2S0[0x00] or I2S1[0x40] bit 8 to "1" enables clock sharing. The following Figure shows the related circuitry.



Using the clock signal from the other channel, however, prevents master clock output from this one. The firmware therefore must set bit 1 in the same register to "0" (slave mode). Note that the one supplying the clock signal can use either master or slave mode.

Having simultaneous I^2S transfers over the two interfaces use the same clock signal reduces the number of signal pins used to four.

19.5.5 Conversion from Monaural to Stereo

Setting I2S0[0x00] or I2S1[0x40] bit 12 to “1” converts monaural data to stereo output—more precisely, duplicates the left channel output on the right channel.

There are no facilities for converting stereo data input into monaural output.

19.6 Sample Settings

This example gives I^{2}S register settings for connecting the four signals SCK, WS, SDI, and SDO to an audio chip supporting transfers and receiving its clock and word select outputs.

Note that the firmware must first adjust GPIO register settings to switch these I^{2}S I/O pins from other I/O functions used before making these settings.

I2S0 Settings

- Slave mode
- Transmit mode
- DMA enabled
- 16-bit stereo
- 32 fs

I2S1 Settings

- Slave mode
- Receive mode
- DMA enabled
- 16-bit stereo
- 32 fs
- Shared clock

Procedure

```
I2S0[0x00] = 0x00000005 # slave mode, transmit mode, 16-bit stereo, 32 fs
I2S0[0x18] = 0x00000030 # enable over- and underflow interrupt requests
I2S0[0x00] = 0x00000085 # software reset
I2S0[0x00] = 0x0000000D# enable DMA transfers
I2S1[0x40] = 0x00000101 # slave mode, receive mode, 16-bit stereo, 32 fs, clock sharing
I2S1[0x58] = 0x00000030 # enable over- and underflow interrupt requests
I2S1[0x40] = 0x00000181 # software reset
I2S1[0x40] = 0x00000109 # enable DMA transfers
```

20. SERIAL PERIPHERAL INTERFACE (SPI)

20.1 Overview

The S1S65010 includes a single serial peripheral interface (SPI) channel.

The SPI supports both master and slave mode, and transfers from 1 to 32 bits of data. A delay time of from 0 to 65,535 clock cycles can be inserted between individual data transfers and it can also generate internal interrupts. This block includes data buffers for both transmit and receive. Four external pins are allocated to the SPI. The SRDY# signal is fixed at the low level internally, and cannot be used as an external pin signal.

20.1.1 Master Mode

When the SPI is set to master mode, it controls the data transfers with the slave devices connected to the SPI bus. The serial clock signal is supplied by the SPI to the slave device via SCLK pin, and serial data is output from the MOSI pin and input to the MISO pin. The SPI block provides an SS (slave select) pin. Although this pin is not required for data transfer, it can be used to detect mode violation errors. In a multi-master SPI system, a mode violation error occurs when two or more devices are set to master mode at the same time. When the SPI is in master mode, if the block detects that the SS pin is at the active level a mode violation interrupt is generated and the SPI is automatically reset to slave mode to prevent signal collision. If there is no need to detect mode violation errors, the SS pin can be used as a general-purpose I/O pin.

After enabling the SPI (setting it to the active state), a data transfer starts when the application writes the transmit data to the Transmit Data Register (TXD).

Figure 20-1 shows the control and operation flow in master mode.

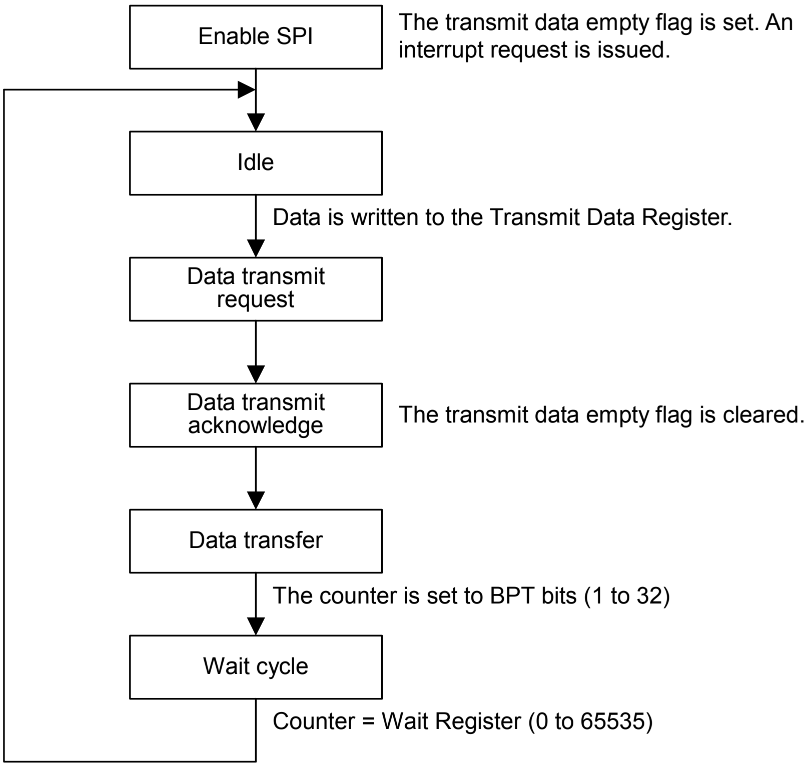

Enable SPI
The transmit data empty flag is set. An interrupt request is issued.
Idle
Data is written to the Transmit Data Register.
Data transmit request
Data transmit acknowledge
The transmit data empty flag is cleared.
Data transfer
The counter is set to BPT bits (1 to 32)
Wait cycle
Counter = Wait Register (0 to 65535)

Fig.20.1 SPI Master Mode Transmit Flowchart

20.1.2 Slave Mode

When the SPI is set to slave mode, an external SPI master controls data transfers with this SPI. A signal output by the external master is used as the operating clock input to the SCLK pin. Serial data is input to the MOSI pin and output from the MISO pin. The SS (slave select) pin becomes an input.

When the SS pin goes to the active level, serial clock input and transfer operation are enabled.

After enabling the SPI (setting it to operation enabled state), the transfer is started by the external SPI master. The SPI circuit includes an internal counter operated by the SCLK clock signal, the circuit uses that counter to control the transmit or receive signals for the set transfer bit count.

If the SCLK clock is input for a number of cycles that exceeds the set transfer bit count, only the set number of transfer bits of transfer data is guaranteed.

Fig.20.2 shows the control and operation flow in slave mode.

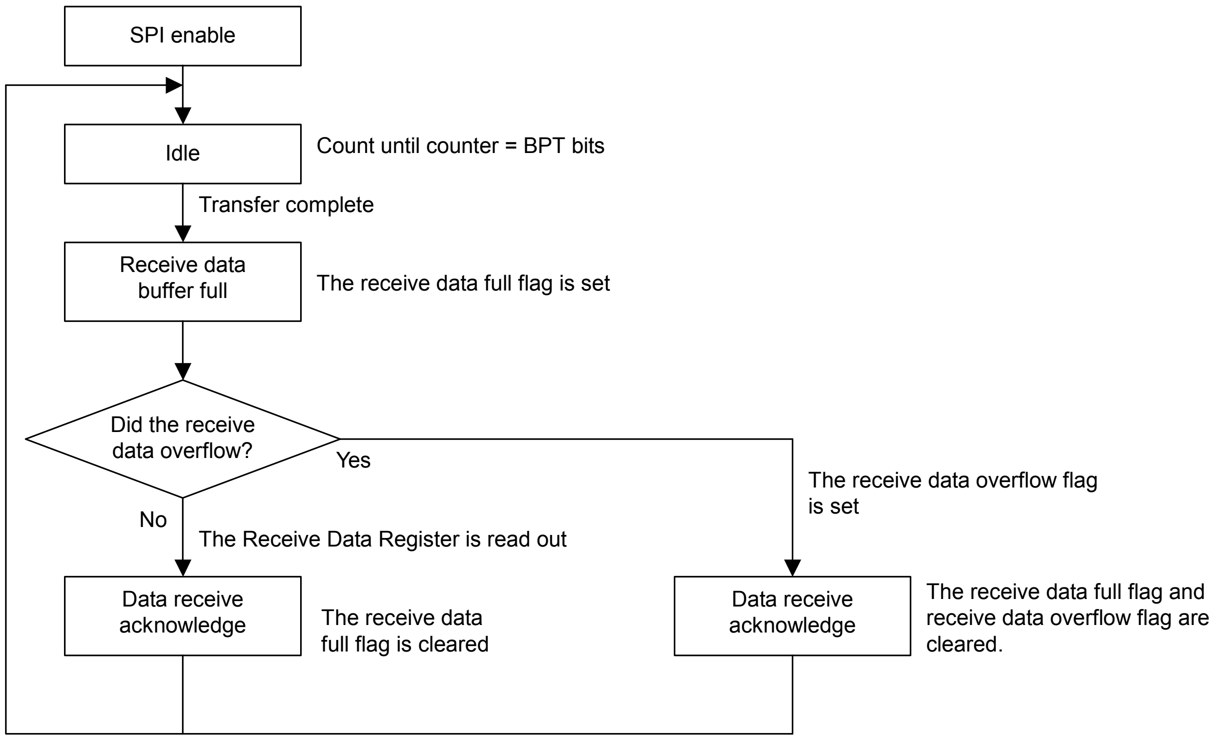

SPI enable
Idle
Count until counter = BPT bits
Transfer complete
Receive data buffer full
The receive data full flag is set
Did the receive data overflow?
Yes
The receive data overflow flag is set
No
The Receive Data Register is read out
Data receive acknowledge
The receive data full flag is cleared
Data receive acknowledge
The receive data full flag and receive data overflow flag are cleared.

Fig.20.2 SPI Slave Mode Receive Flowchart

20.2 Block Diagram

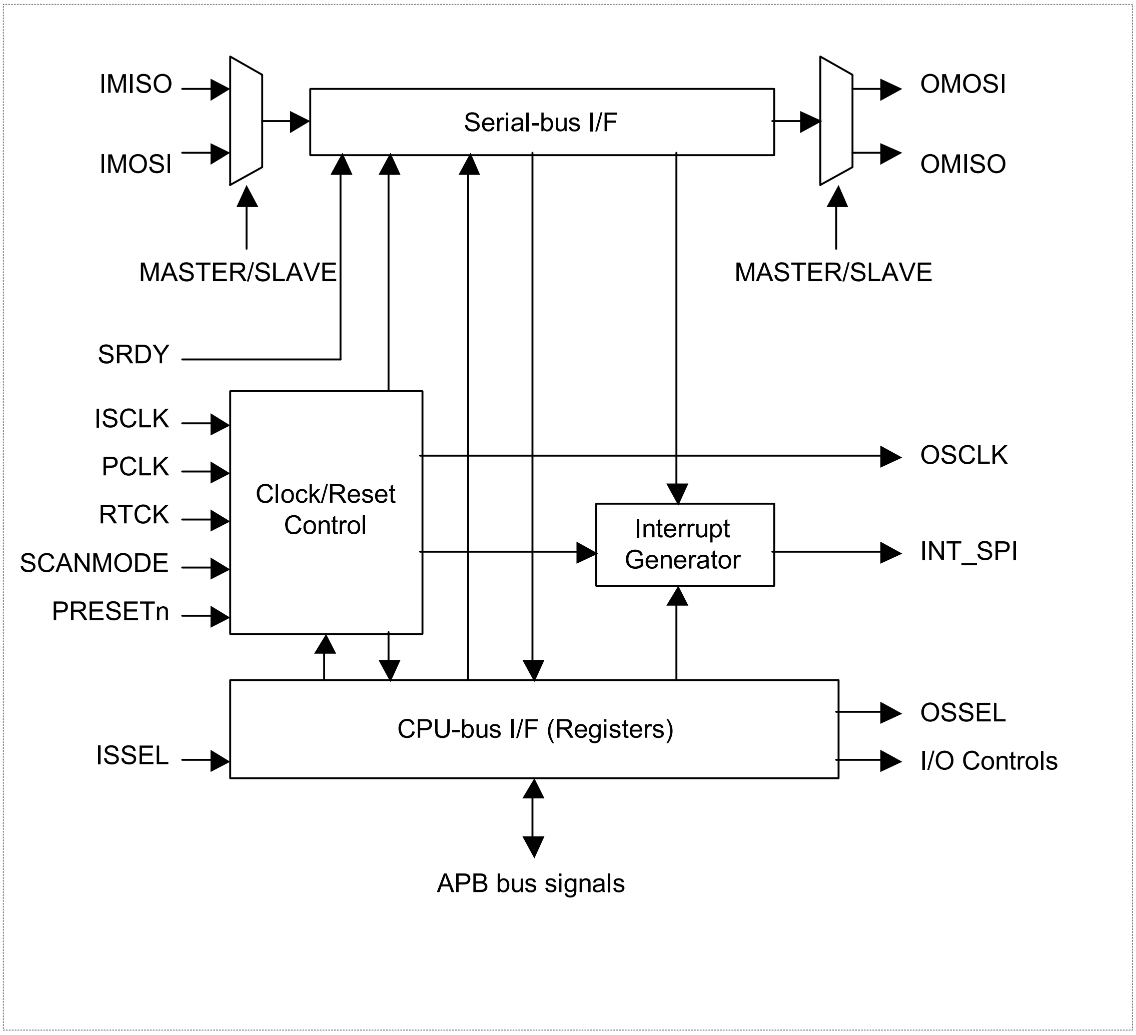


Fig.20.3 SPI Block Diagram

20.3 External Pins

The serial peripheral interface interacts with the following pins.

Pin Name	I/O	Pin Function	Multiplexed Pin*
SPI_SCLK	I/O	SPI serial clock	GPIOA3/RXD1*
SPI_SS	I/O	SPI chip select	GPIOA2/TDX1*
SPI_MISO	I/O	SPI serial data master input/slave output	GPIOA4*
SPI_MOSI	I/O	SPI serial data master output/slave input	GPIOA5*

Note*: These external pins is set as GPIO operation, so specify “non-GPIO function #1” in the GPIO Pin Function Register to configure them for this function.

20.4 Clock and Data Transfer Timing

When the SPI is used in master mode, the internal SCLK clock is used to operate the shift register that performs transfer data input and output. One of four types of signal, based on the combination of the phase and polarity, can be selected for SCLK.

The clock phase is selected with the CPHA bit (bit 9 in the SPI Control Register 1). If CPHA is set to “0,” the output data changes on the clock falling edge (data is output from the shift register), and the input data is captured in the shift register on the clock rising edge (the bits in the shift register are shifted in sequence). When data is written to the Transmit Data Register, the MSB is output. If CPHA is set to “1,” the output changes on the rising edge and the input is acquired on the falling edge. The MSB in the data is output on the first SCLK rising edge.

The clock polarity is selected with the CPOL bit (bit 8 in SPI Control Register 1). When CPOL is “0,” High is active, and when CPOL is “1,” Low is active. The above description of CPHA presents the I/O timing when the clock is in the active high polarity. If CPOL is “1.” the rising and falling edges are reversed. Note, however, that the SPI internal edge trigger event timing is not reversed.

Fig.20.4 shows the SCLK clock waveforms in the master mode when these bits are selected. This flexibility covers the majority of serial peripherals on the market today.

Fig.20.5 shows the SCLK clock waveforms in the slave mode.

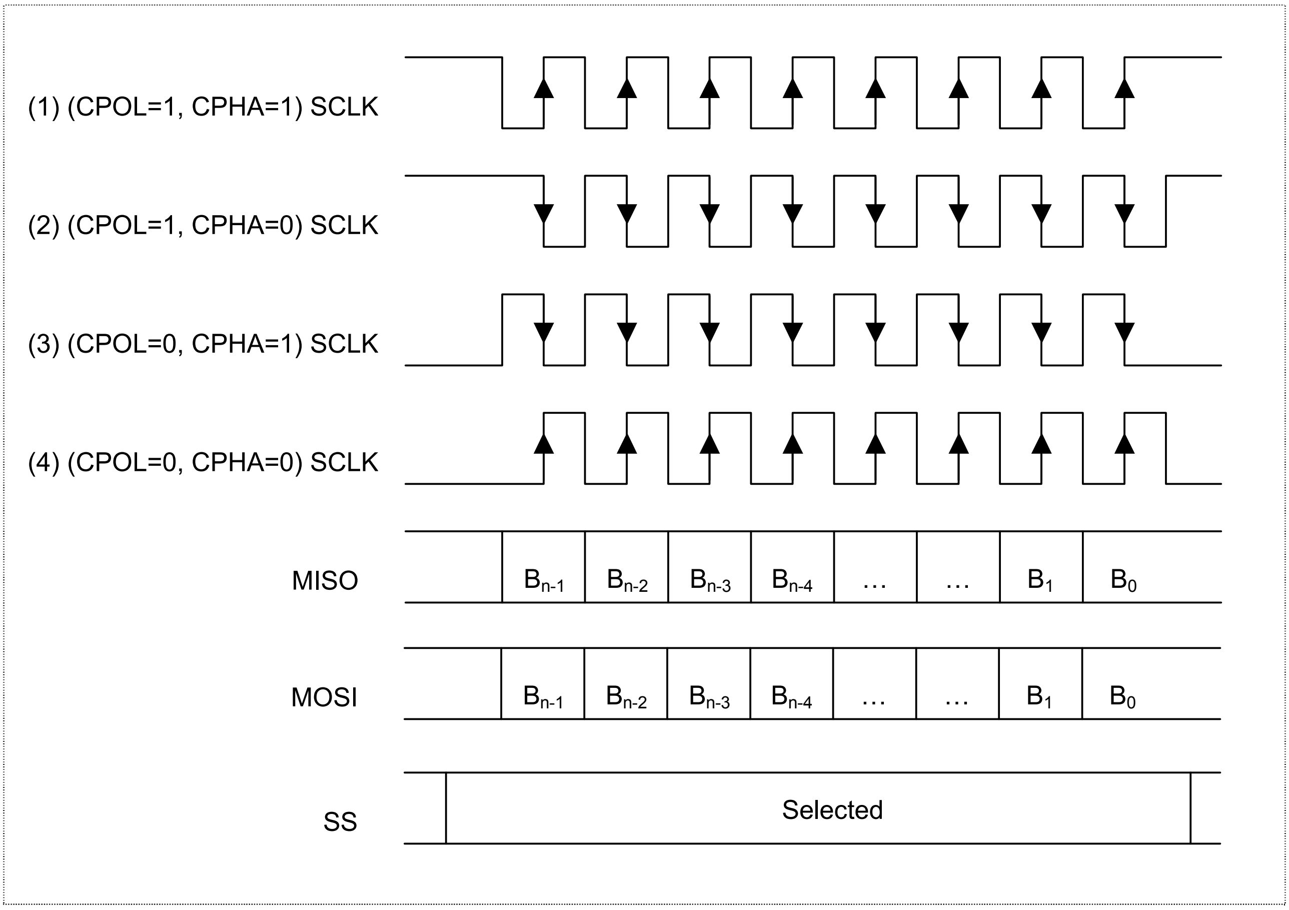

(1) (CPOL=1, CPHA=1) SCLK
(2) (CPOL=1, CPHA=0) SCLK
(3) (CPOL=0, CPHA=1) SCLK
(4) (CPOL=0, CPHA=0) SCLK
MISO
B$_{n-1}$ B$_{n-2}$ B$_{n-3}$ B$_{n-4}$ … … B$_1$ B$_0$
MOSI
B$_{n-1}$ B$_{n-2}$ B$_{n-3}$ B$_{n-4}$ … … B$_1$ B$_0$
SS
Selected

Fig.20.4 Clock Settings in SPI Master Mode (Where n is the transfer Data bit count)

Fig.20.5 shows the SCLK clock waveforms in the slave mode as reference information.

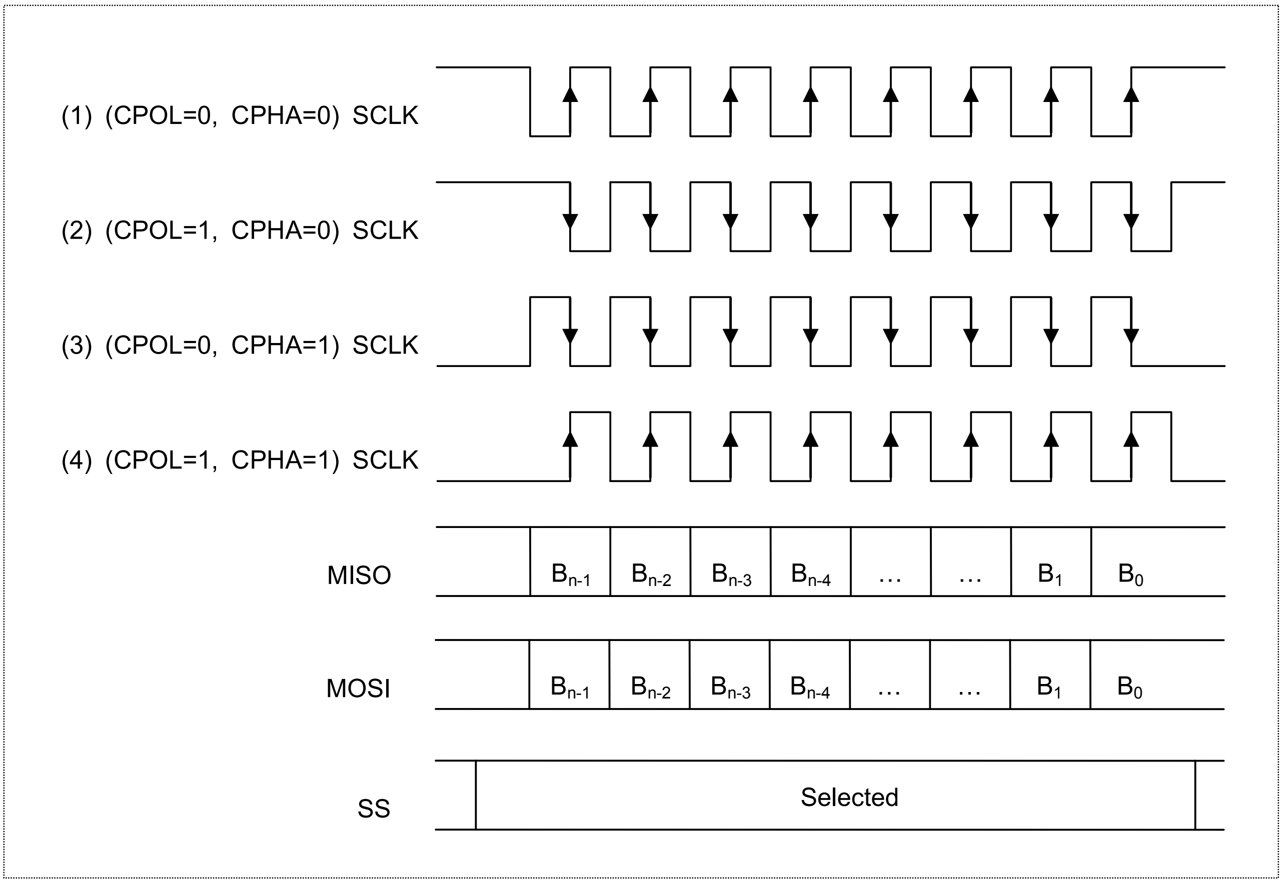

(1) (CPOL=0, CPHA=0) SCLK
(2) (CPOL=1, CPHA=0) SCLK
(3) (CPOL=0, CPHA=1) SCLK
(4) (CPOL=1, CPHA=1) SCLK
MISO
Bn-1
Bn-2
Bn-3
Bn-4
…
…
B1
B0
MOSI
Bn-1
Bn-2
Bn-3
Bn-4
…
…
B1
B0
SS
Selected

図 20.5 Clock Settings in SPI Slave Mode (When the number of transfer data bits is n)

20.5 Registers

20.5.1 Register List

The SPI control registers are allocated with a default base address of 0xFFFF_2000.
In the absence of any specification to the contrary, the default value for unreserved bits is “0.”

Table 20.1 SPI Register List (Base Address: 0xFFFF_2000)

Address Offset	Register Name	Default Value	R/W	Data Access Size (Bits)
SPI Registers				
0x00	SPI Receive Data Register	0x0000_0000	RO	32
0x04	SPI Transmit Data Register	0x0000_0000	R/W	32
0x08	SPI Control Register 1	0x0000_0000	R/W	32
0x0C	SPI Control Register 2	0x0000_0000	R/W	32
0x10	SPI Wait Register	0x0000_0000	R/W	32
0x14	SPI Status Register	0x0000_0010	RO	32
0x18	SPI Interrupt Control Register	0x0000_0000	R/W	32

20.5.2 Detailed Register Description

SPI Receive Data Register
SPI[0x00] Default = 0x0000_0000 Read Only

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

Bits 31 to 0: **Receive Data [31:0]**
The data received from the external serial peripheral device can be read out.

SPI Transmit Data Register
SPI[0x04] Default = 0x0000_0000 Read/Write

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

Bits [31:0]: **Transmit Data [31:0]**
Data buffer to which the transmit data is written. Data can be written when the TDEF bit (bit 4 in the SPI Status Register), which indicates whether or not this register is empty, is “1.”

SPI Control Register 1
SPI[0x08] Default = 0x0000_0000 Read/Write

n/a															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
n/a	BPT [4:0]					CPHA	CPOL	n/a	MCBR [2:0]			CLKS	RX RAW	Mode	ENA
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 14 to 10: **BPT[4:0]**
Transfer Bit Count
Specifies the length of the data bits transmitted or received in a single transfer.

00000: 1 bit
00001: 2 bits
.
.
.
11110: 31 bits
11111: 32 bits

Bit 9: **CPHA**
Serial Clock Phase Select
Selects the phase of the serial clock.
0: Generate a clock pulse at the second half in the data cycle (see Figure 20-4 (2)(4))
1: Generate a clock pulse at the first half in the data cycle (see Figure 20-4 (1)(3))

Bit 8: **CPOL**
Serial Clock Polarity Select
Selects the polarity of the serial clock.
0: High active (Generate High pulse as clock; see Figure 20-4 (3)(4))
1: Low active (Generate Low pulse as clock; see Figure 20-4 (1)(2))

Bits 6 to 4: **MCBR [2:0]**
Master Clock Bit Rate Select
Sets the SCLK speed in master mode. The source clock (bus clock) divisor is set as shown below by the setting of this field.
Divisor = $4 * 2^{MCBR[2:0]}$
Thus the SPI master clock will operate as follows.
Master clock frequency (f_{SCLK}) = bus clock frequency / ($4 * 2^{MCBR[2:0]}$)
Note: This field is invalid in slave Mode and when the real-time clock (32.768 kHz) is used as the source clock in master Mode (when bit 3 in this Register is set to 1).

Bit 3: **CLKS**
Source Clock Select
Selects the source clock used to generate the SCLK in master mode.
0: Bus clock
1: Real-time clock (32.768 kHz)

Bit 2: **RXDATA RAW**
0: The RXDATA is masked by the BPT width
1: The RXDATA is shift register data without masking

Bit 1: **Mode**
SPI Mode Select
Selects whether this interface operates in master mode or slave mode.
0: Slave mode
1: Master mode

Bit 0: **ENA**
SPI Enable
Enables the SPI transmit and receive circuits.
0: Disable
1: Enable

SPI Control Register 2

SPI[0x0C] Default = 0x0000_0000 Read/Write

n/a															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
n/a				SSA	SS	SSP	SSC	n/a					Reserved (0)		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit 11: **SSA**

Slave Select Pin (SS) Automatic Control

This bit sets the following operation in master mode.

0: If the SS pin is set to output, the SS pin is controlled by SS (bit 10)

1: If the SS pin is set to output, the SS pin is controlled by the internal transfer timing

This bit is invalid in slave mode.

Bit 10: **SS**

Slave Select Pin (SS) Control

In master mode, this bit controls the SS pin output when the SS pin is set to output.

0: The SS pin outputs the inactive level

1: The SS pin outputs the active level

In slave mode, this bit indicates the following when the SS pin is set to the disabled input state (SSC = 0).

0: This SPI is not selected

1: This SPI is selected

This bit is invalid in all other cases.

Bit 9: **SSP**

Slave Select pin (SS) Polarity Select

0: Low active

1: High active

Bit 8: **SSC**

Slave Select Pin (SS) Setting

In master mode, switches the I/O direction of the SS pin.

0: Input (mode violation detection)

1: Output (slave select output)

The output level (High or Low) can be set with the SS bit (bit 10 in this register).

Mode violation detection is not performed.

In slave mode:

0: The SS pin is set to the disabled input state. The selection specified by the SS bit (bit 10) is becomes valid.

1: The SS pin is set to the enabled input state.

The table below summarizes the settings provided by bits[10:8].

Table 20.2 SS Pin State Settings

Mode Select*	**Bit 8: SSC (SS setting)**	**Bit 11: SSA (SS automatic select)**	**Bit 9: SSP (SS polarity select)**	**Bit 10: SS (SS control)**	**SS pin state (Active Level)**
Master mode	0: SS pin input (Mode violation detection)	Invalid	0: Low active	Invalid	SS input (low)
			1: High active		SS input (high)
	1: SS pin output	0: Controlled by the SS bit	0: Low active	0: Inactive	SS output = high (low)
				1: Active	SS output = low (low)
			1: High active	0: Inactive	SS output = low (high)
				1: Active	SS output = high (high)
		1: Automatic control	0: Low active	Invalid	SS output = controlled automatically (low)
			1: High active		SS output = controlled automatically (high)
Slave mode	0: SS pin disabled input	Invalid	Invalid	0: Not selected	SS input (high) <not selected>
				1: Selected	SS input (high) <selected>
	1: SS pin enabled input		0: Low active	Invalid	SS input (low)
			1: High active		SS input (high)

*: The Mode bit (bit 1 in SPI Control Register 1) selects the mode.

Bits 2 to 0: **Reserved (0)**

SPI Wait Register SPI[0x10] Default = 0x0000_0000															Read/Write
n/a															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WAIT Cycles [15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 15 to 0: **WAIT Cycles [15:0]**

Wait cycles

Sets the wait time inserted between each data transmit/receive operation in SCLK cycles. (Wait time = WAIT Cycles[15:0] × SCLK period)

0000h (w): 0 clock cycles
0001h (w): 1 clock cycle
0002h (w): 2 clock cycles
.
.
.
FFFFh (w): 65535 clock cycles

Note: This register setting is only valid in master mode.

SPI Status Register
SPI[0x14] Default = 0x0000_0010 Read Only

n/a															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
n/a									BSYF	MFEF	TDEF	RDOF	RDFF	n/a	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit 6: **BSYF**
Transfer Busy Flag
Indicates whether or not an SPI transmit or receive operation is in progress.
0: Standby
1: A transmit or a receive operation is in progress.
This flag is set automatically by the start of a transmit or a receive operation. When set, the BSYF flag is automatically cleared when the transfer completes and the SPI block returns to the standby state.
This flag is only valid in master mode and is always "0" in slave mode.

Bit 5: **MFEF**
Mode Violation Error Flag
Indicates whether or not a mode violation error has occurred.
0: No error detected
1: An error occurred.
This flag is set when the SPI is in master mode and the SS pin is set to the active level by the external device. This flag is automatically cleared when the error is resolved. To clear this error, the SPI block will go to slave mode while this bit and the MFIE bit (bit 5 in the SPI Interrupt Control Register) are set and disable all outputs without starting any data transfer operations.

Bit 4: **TDEF**
Transmit Data Empty Flag
Indicates that the Transmit Data Register is empty.
0: Transmit data present
1: No transmit data present (default)
This flag is set when the transmit data written to the Transmit Data Register has been transmitted over the serial interface (or when the serial interface is reset). Setting this flag makes it possible for the next transmit data to be written to the Transmit Data Register. Note that when this flag is set, it is cleared by writing to the Transmit Data Register.

Bit 3: **RDOF**
Receive Data Overflow Flag
Indicates that a receive data overflow has occurred.
0: No overflow
1: An overflow has occurred.
This flag is set when the next receive data is sent to the Receive Data Register from the serial interface in the state where the receive data full flag is set (and the receive data has not been read out). When this flag is set, it is cleared when the Receive Data Register is read.

Bit 2: **RDFF**
Receive Data Full Flag
Indicates that there is receive data in the Receive Data Register.
0: No receive data
1: Receive data is present
This flag is set when receive data is sent from the serial interface to the Receive Data Register. When this flag is set, it is cleared by reading the Receive Data Register.

Note: This Register is cleared when the SPI Enable bit (bit 0 in the SPI Control Register 1) is set to "0" and also when the SPI block is disabled.

SPI Interrupt Control Register
SPI[0x18] Default = 0x0000_0000 Read/Write

n/a															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
n/a										MFIE	TEIE	ROIE	RFIE	MIRQ	IRQE
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit 5: **MFIE**

Mode Violation Interrupt Enable

Enables/disables the mode violation interrupt.

0: Disable
1: Enable

This setting is valid only when the SPI is in master mode and the SS pin is set up for mode violation detection use.

Bit 4: **TEIE**

Transmit Data Register Empty Interrupt Enable

Enables or disables the transmit data register empty interrupt.

0: Disable
1: Enable

Bit 3: **ROIE**

Receive Data overflow Error Interrupt Enable

Enables or disables the receive data overflow error interrupt.

0: Disable
1: Enable

Bit 2: **RFIE**

Receive Data Register Full Interrupt Enable

Enables or disables the receive data register full interrupt.

0: Disable
1: Enable

Bit 1: **MIRQ**

Manual Interrupt Request Set/Clear

Sets or clears the SPI manual interrupt request.

0: Clears
1: Sets

This bit allows software to generate an SPI interrupt. The control provided by this bit is invalid when the IRQE (bit 0 in this register) is set to “0” (interrupts disabled).

Bit 0: **IRQE**

Interrupt Request Enable

Enables or disables SPI interrupt requests.

0: Disable
1: Enable

21. COMPACT FLASH INTERFACE (CF)

21.1 Overview

This module has the following features.

- CF card attribute memory space (2 KB)
- CF card common memory space (2 KB)
- CF card I/O space (2 KB)
- Interrupt request outputs STSCHG# and IREQ
- Command strobe timing output for internal clock (PCLK) frequencies from 6 MHz to 60 MHz
- Programmable idle cycle insertion and programmable command cycle insertion for strobe outputs (CFIORD# and CFIOWR#)
- CF card interface support for True IDE operation
  Note, however, that external circuitry must provide the CFOE# pull down resistance, the CSSEL signal, and the low active reset signal.

Note: The limited number of pins available prevents this device from providing the following signal lines. Internally, they are all fixed at Low level.

- CD [2:1]#
- VS [2:1]#
- BVD2#
- WP/IOIS16#

21.2 Block Diagram

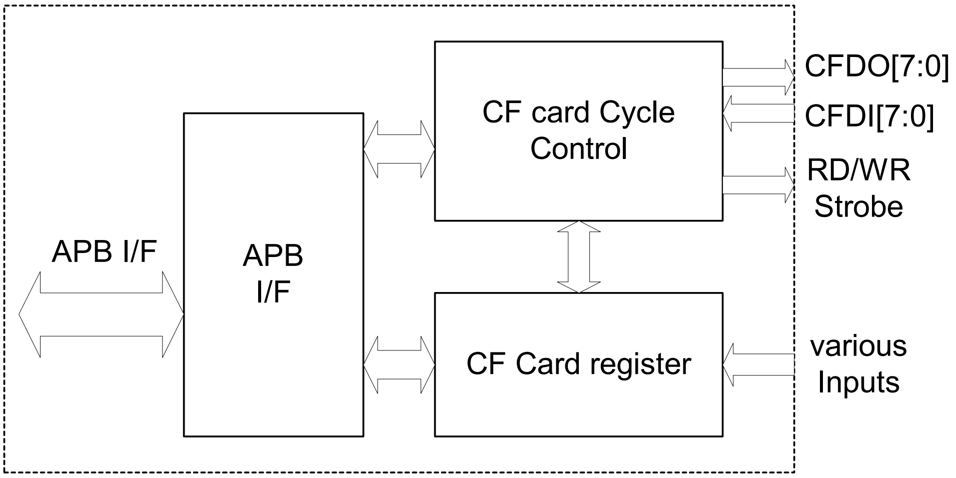


Fig.21.1 Compact Flash Interface (CF) Block Diagram

21.3 Compact Flash (CF) Card Interface Memory Map

Table 21.1 CF Card Interface Memory Map

Description	Address Range	Size (KB)
CF card attribute memory space	0xFFFE4000 to 0xFFFE47FF	2
CF card common memory space	0xFFFE4800 to 0xFFFE4FFF	2
CF card I/O space	0xFFFE5000 to 0xFFFE57FF	2
CF card True IDE CS1# space	0xFFFE5800 to 0xFFFE5BFF	1
CF card True IDE CS2# space	0xFFFE5C00 to 0xFFFE5FFF	1
CF card interface settings	0xFFFE6000 to 0xFFFE6FFF	4

Note: The CF card spaces do not support 32-bit access. Only the 8- and 16-bit data sizes function properly. The CF card interface settings registers, in contrast, support only 16- and 32-bit access.

True IDE

Simply accessing the specified spaces is not sufficient to create a full True IDE implementation using this CF card interface. The user application system must provide certain signal operations at the board level—starting with proper implementation of the following two key signals.

OE# (a.k.a. ATASEL): The CF card interface OE# signal must be at Low level during a power on reset, the state following when the power supply goes from OFF to ON.

CSSEL: The board must also pull up, pull down, or leave open CSSEL, the signal determines the operation mode (master or slave) for IDE devices, corresponding to the user application system.

True IDE also imposes the following restriction on addresses given in the Table above for the CS1# and CS2# spaces.

The CS1# space allows access to registers in the address range with 0x0 (“000b”) to 0x7 (“111b”) in the lowest three bits; the CS2# space restricts access to addresses with 0x6 (“110b”) or 0x7 (“111b”) in their lowest three bits. Alternate Status Registers, for example, are assigned to CS2# addresses with 111b in their lowest three bits.

21.4 External Pins

This module interacts with the following external pins.

Table 21.2 Compact Flash Interface (CF) External Pins

Pin Name	I/O	Pin Function	Multiplexed Pin
CFCE2#	O	CF card enable 2 (CE2#)	GPIOD2*
CFCE1#	O	CF card enable 1 (CE1#)	GPIOD3*
CFIORD#	O	CF I/O read strobe	GPIOE0*
CFIOWR#	O	CF I/O write strobe	GPIOE1*
CFWAIT#	I	Wait request from CF card	GPIOE2*
CFRST	O	Reset signal to the CF card	GPIOE3*
CFIREQ	I	Interrupt request signal from CF card	GPIOE4*
CFSTSCHG#	I	Status change signal from CF card	GPIOE5*
CFDEN#	O	Data enable signal for CF card external buffer	GPIOE6*
CFDDIR	O	CF data bus direction indicator	GPIOE7*
CFREG#	O	CF card interface REG signal for attribute and I/O spaces	MA11**
CFADDR [10:0]	O	CF card interface address bus	MA [10:0]**
CFDATA [15:0]	I/O	CF card interface 16-bit data bus	MD [15:0]**
CFOE#	O	CF card interface output enable signal for attribute and common memory spaces	MOE#**
CFWE#	O	CF card interface write enable signal for attribute and common memory spaces	MWE0#**

Notes * These external pins is set as GPIO operation, so specify “non-GPIO function #1” inthe GPIO Pin Function Register to configure them for this function.

** When the CF card interface is in operation, it uses memory controller pins as its external pins.

21.5 Registers

21.5.1 Register List

The base address for these registers is 0xFFFE_6000.

Table 21.3 CF Register List (Base Address: 0xFFFE_6000)

Address Offset	Register Name	Abbreviation	Default Value	R/W	Data Access Size (Bits)
0x00	CF Card Interface Control Register	CFCTL	0x1000	(R/W)	16 (/32)
0x04	CF Card Pin Status Register	CFPINSTS	0x0XXX	RO	16 (/32)
0x08	CF Card IRQ Source & Clear Register	CFINTRSTS	0x0XXX	R/W	16 (/32)
0x0C	CF Card IRQ Enable Register	CFINTMSTS	0x0000	R/W	16 (/32)
0x10	CF Card IRQ Status Register	CFINTSTS	0x0000	RO	16 (/32)
0x14	CF Card MISC Register	CFMISC	0x0000	R/W	16 (/32)

21.5.2 Detailed Register Descriptions

CF Card Interface Control Register (CFCTL)							
CF[0x00]	Default = 0x1000						(Read/Write)
PROG CYCEN	PROG IDLE [2:0]			PROG CYC [3:0]			
R/W							
15	14	13	12	11	10	9	8
Reserved (0)	IOIS8_IO	IOIS8_MEM	PROG IDLE EN	CFRST	CFCARDEN	PCKMD[1:0]	
RO	R/W						
7	6	5	4	3	2	1	0

Bit 15: **PROG_CYCEN**

Strobe Output (CFIORD#/CFIOWR#) Programmable Command Cycle Insertion Enable

0: Disable (default)
1: Enable

Normally leave this disabled, the default setting.

Bits 14 to 12: **PROG_IDLE[2:0]**

Number of Idle Cycles for CF Card Interface

The default setting is “1”.

Bits 11 to 8: **PROG_CYC[3:0]**

Number of Command Active Cycles for CFIORD# and CFIOWR# Strobe Outputs

This setting is only valid when PROG_CYCEN is “1.”

Bit 7: **Reserved**

Bit 6: **IOIS8_IO**

Device Size for CF Card I/O space

0: 16 bits
1: 8 bits

Bit 5: **IOIS8_MEM**

Device Size for CF Card Common Memory Space

0: 16 bits
1: 8 bits

Bit 4: **PROG_IDLE_EN**

Strobe Output (CFIORD#/CFIOWR#) Programmable Idle Cycle Insertion Enable

0: Disable
1: Enable

The specifies the timing at which the command output (CFCE1#/CFCE2#) goes active.
Normally leave this disabled, the default setting.

Bit 3: **CFRST**

Direct CFRST Pin Control when CF Card Interface Selected

0: Low level
1: High level

Bit 2: **CFCARDEN**

CF Card Interface Enable

0: Disable
1: Enable

Bits 1 to 0: **PCKMD[1:0]**

Clock Frequency Setting

Selects the appropriate value for PCLK frequency to operate the CF card interface normaly.

00: When PCLK clock rate is 25 MHz to 50 MHz
01: When PCLK clock rate is around 24 MHz
10: When PCLK clock rate is around 12 MHz
11: When PCLK clock rate is around 6 MHz

CF Card Pin Status Register (CFPINSTS)
CF[0x04] Default = 0x0XXX Read Only

0							IREQ#2
15	14	13	12	11	10	9	8
WP	IREQ#1	BVD2#	BVD1#/ STSCHG	VS2#	VS1#	CD2#	CD1#
7	6	5	4	3	2	1	0

These bits monitor pin input levels. Bits 1 to 0 provide noise filtering on their inputs (CD2# and CD1#). All the others directly reflect the corresponding pin inputs.

Bit 8: **IREQ#2 Pin Input**

This bit directly monitors the CF card interface IREQ# pin level. It is particularly handy when the IREQ# pin is used as high active interrupt request input.

Bit 7: **WP Pin Input**

This bit monitors the CF card interface WP pin input.

Bit 6: **RDY/BSY, IREQ Pin Input** (The name changes with CF card interface mode.)

This bit monitors the inverse of the CF card interface RDY/BSY or IREQ pin. It is particularly handy when the IREQ# pin is used as low active interrupt request input.

Bit 5: **BVD2# Pin Input**

This bit monitors the CF card interface BVD2# pin input.

Bit 4: **BVD1#/STSCHG# Pin Input**

This bit monitors the CF card interface BVD1#/STSCHG# pin input.

Bit 3: **VS2# Pin Input**

This bit monitors the CF card interface VS2# pin input.

Bit 2: **VS1# Pin Input**

This bit monitors the CF card interface VS2# pin input.

Bit 1: **CD2# Pin Input**

This bit monitors the CF card interface CD2# pin input after noise removal.

Bit 0: **CD1# Pin Input**

This bit monitors the CF card interface CD1# pin input after noise removal.

Note: This device does not support bits 7, 5, or 3 to 0.

CF Card IRQ Source & Clear Register (CFINTRSTS)							
CF[0x08]	Default = 0x0XXX						Read/Write
Reserved							IREQ#2
15	14	13	12	11	10	9	8
Reserved	IREQ#1	Reserved	BVD1/ STSCHG	Reserved		CD2	CD1
7	6	5	4	3	2	1	0

This register indicates raw (unmasked) interrupt request sources. Writing “1” to a bit clears the corresponding interrupt request.

Bits 15 to 7: **Reserved (0)**
Reads always return “0.”

Bit 8: **IREQ#2 Input**
This bit is particularly handy when the IREQ# pin is used as high active interrupt request input.
In CF card I/O mode, a “1” in this bit indicates that there is an interrupt request from the IREQ pin.

Bit 6: **IREQ#1 (Inverse Input)**
This bit is particularly handy when the IREQ# pin is used as low active interrupt request input.
In CF card I/O mode, a “1” in this bit indicates that there is an interrupt request from the IREQ pin.

Bit 4: **BVD1/STSCHG#**
Pin Status Change
In CF card I/O mode, a “0” in this bit indicates that the status change signal STSCHG# is “0”—that is, that there has been a change in the RDY/BSY# or WP signal. “1” indicates no particular change in the signals.

Bit 1 (CD2): **CD2 Pin Status Change**
This bit goes to “1” at rising and falling edges in the CD2# signal after noise removal.

Bit 0 (CD1): **CD1 Pin Status Change**
This bit goes to “1” at rising and falling edges in the CD1# signal after noise removal.

Note: This device does not support bits 1 to 0.

CF Card IRQ Enable Register (CFINTMSTS)							
CF[0x0C]	Default = 0x0000						Read/Write
Reserved							IRQEN#EN2
15	14	13	12	11	10	9	8
Reserved	IREQ#EN1	Reserved	BVD1EN/ STSCHGEN	Reserved		CD2EN	CD1EN
7	6	5	4	3	2	1	0

These bits represent the interrupt request enable counterparts to the interrupt request status flag bits in the CF card IRQ Source & Clear Register.

0: Disable (mask)
1: Enable

Note: This device does not support bits 1 to 0.

CF Card IRQ Status Register (CFINTSTS)
CF[0x10] Default = 0x0000 Read Only

Reserved							IREQ#2
15	14	13	12	11	10	9	8
Reserved	IREQ#1	Reserved	BVD1/ STSCHG	Reserved		CD2	CD1
7	6	5	4	3	2	1	0

This register gives the result of ANDing the IRQ Source and IRQ Enable Registers.

"1" in a bit indicates an interrupt request from the corresponding source; a "0," otherwise—either there is no interrupt request or the interrupt request is masked.

Note: This device does not support bits 1 to 0.

CF Card MISC Register (CFMISC)
CF[0x14] Default = 0x0000 Read/Write

Reserved							
15	14	13	12	11	10	9	8
Reserved							CSRDEN
7	6	5	4	3	2	1	0

This register is for use by the hardware. Normally do not write to it.

Bits 15 to 2: **Reserved (0)**

Bit 0: **CSRDEN**
Double Output Enable
0: Disable
1: Enable
Setting this bit to "1" makes CFCE1# and CFCE2# both active during reads.

21.6 Usage Limitations

The limited number of pins available prevents this device from providing the following signal lines. Internally, they are all fixed at Low level.

- CD [2:1]#
- VS [2:1]#
- BVD2#
- WP/IOIS16#

The following Table lists unsupported register bits.

Bit Address	Register Bit Name	Limitations
CF[0x04] bit 7	WP pin input	Do not use.
CF[0x04] bit 5	BVD2# pin input	Do not use.
CF[0x04] bit 3	VS2# pin input	Do not use.
CF[0x04] bit 2	VS1# pin input	Do not use.
CF[0x04] bit 1	CD2# pin input	Do not use.
CF[0x04] bit 0	CD1# pin input	Do not use.
CF[0x08] bit 1	CD2 pin status change	Do not use.
CF[0x08] bit 0	CD1 pin status change	Do not use.
CF[0x0C] bit 1	CD2EN	Do not use.
CF[0x0C] bit 0	CD1EN	Do not use.
CF[0x10] bit 1	CD2	Do not use.
CF[0x10] bit 0	CD1	Do not use.

22. TIMERS (TIM)

22.1 Overview

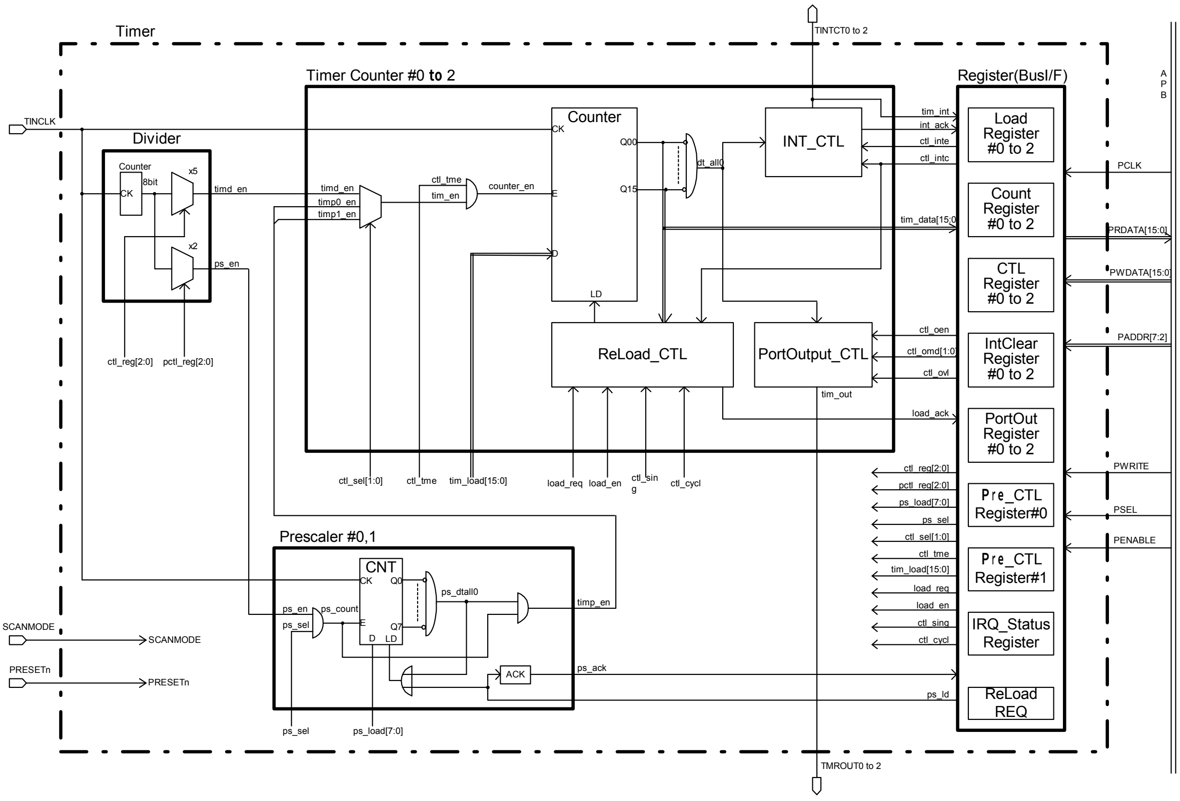
This module has the following features.

- Three 16-bit count down timers with identical structures
- Two timer modes: cyclic and single
- Maskable interrupt request when timer counter goes to "0"
- Frequency divider using 8-bit counter for divisors 1 to 256
- Two prescalers using 8-bit counters to tune the clock signal from the divider block
- Three choices for timer output upon underflow: underflow signal, fixed level output pulses, and output toggle indicating underflow frequency

22.2 Block Diagram

The following block diagram shows block components: registers (and bus interface to them), frequency divider (Divider), two prescalers (#0 and #1), and three timer/counters (#0 to #2). It shows only the detailed block structures for prescaler #0 and timer/counter #0 because the others have identical structures. The divider block and control registers are connected to all five numbered blocks.

The clock signal (TINCLK) from the system controller is 1/8 the PCLK (APB bus clock) frequency. For further details, see Section 13 "System Controller."



<16bit-TIMER BLOCK>

Fig.22.1 TIM Block Diagram

22.3 External Pins

This module interacts with the following external pins.

Pin Name	I/O	Pin Function	Multiplexed Pin*
Timer0out	Output	Timer 0 output	GPIOB3/INT3/I2S1_SD
Timer1out	Output	Timer 1 output	GPIOB4/INT4
Timer2out	Output	Timer 2 output	GPIOB5/INT5

Note*: These external pins are multiplexed with GPIO pins and other function pins, so specify "non-GPIO function #1" in the GPIO Pin Function Register to configure them for this function.

22.4 Registers

22.4.1 Register List

The base address for these registers is 0xFFFF_B000.

Table 22.1 TIM Register List (Base Address: 0xFFFF_B000)

Address Offset	Register Name	Abbreviation	Default Value	R/W	Data Access Size (Bits)
0x00	Timer 0 Load Register	TM0LD	0x0000	R/W	16 (/32) *1
0x04	Timer 0 Count Register	TM0CNT	0x0000	RO	16 (/32) *1
0x08	Timer 0 Control Register	TM0CTRL	0x0000	(R/W)	16 (/32) *1
0x0C	Timer 0 IRQ Flag Clear Register	TM0IRQ	—	WO	8 (/16/32) *2
0x10	Timer 0 Port Output Control Register	TM0POUT	0x0000	(R/W)	8 (/16/32) *2
0x20	Timer 1 Load Register	TM1LD	0x0000	R/W	16 (/32) *1
0x24	Timer 1 Count Register	TM1CNT	0x0000	RO	16 (/32) *1
0x28	Timer 1 Control Register	TM1CTRL	0x0000	(R/W)	16 (/32) *1
0x2C	Timer 1 IRQ Flag Clear Register	TM1IRQ	—	WO	8 (/16/32) *2
0x30	Timer 1 Port Output Control Register	TM1POUT	0x0000	(R/W)	8 (/16/32) *2
0x40	Timer 2 Load Register	TM2LD	0x0000	R/W	16 (/32) *1
0x44	Timer 2 Count Register	TM2CNT	0x0000	RO	16 (/32) *1
0x48	Timer 2 Control Register	TM2CTRL	0x0000	(R/W)	16 (/32) *1
0x4C	Timer 2 IRQ Flag Clear Register	TM2IRQ	—	WO	8 (/16/32) *2
0x50	Timer 2 Port Output Control Register	TM2POUT	0x0000	(R/W)	8 (/16/32) *2
0x60 to 0x9C	Reserved	—	—	—	—
0xA0	Prescaler 0 Control Register	PS0CTRL	0x0000	(R/W)	16 (/32) *1
0xA4	Prescaler 1 Control Register	PS1CTRL	0x0000	(R/W)	16 (/32) *1
0xB0	Timer IRQ Status Register	TMIRQSTS	0x0000	RO	8 (/16/32) *2

Notes *1: Registers support 16- and 32-bit access.
*2: Registers support 8-, 16- and 32-bit access.

22.4.2 Detailed Register Descriptions

Timer 0 Load Register (TM0LD)							
TIM[0x00]	Default = 0x0000						Read/Write
Timer 0 Load Value [15:8]							
15	14	13	12	11	10	9	8
Timer 0 Load Value [7:0]							
7	6	5	4	3	2	1	0

Bits 15 to 0: **TM0LD [15:0]**

Timer 0 Load Value

This register specifies the 16-bit value loaded into the counter as its count starting value.

Timer 0 Count Register (TM0CNT)							
TIM[0x04]	Default = 0x0000						Read only
Timer 0 Current Count Value [15:8]							
15	14	13	12	11	10	9	8
Timer 0 Current Count Value [7:0]							
7	6	5	4	3	2	1	0

Bits 15 to 0: **TM0CNT [15:0]**

Timer 0 Current Count Value

This returns the current contents of the counter.

Timer 0 Control Register (TM0CTRL)							
TIM[0x08]	Default = 0x0000						(Read/Write)
n/a RO						Divider/Prescaler Select [1:0] R/W	
15	14	13	12	11	10	9	8
Timer 0 Enable R/W	n/a RO	Mode Select R/W	Divider Divisor [2:0] R/W			Immediate Load Request R/W	IRQ Request R/W
7	6	5	4	3	2	1	0

Bits 9 to 8: **Divider/Prescaler Select [1:0]**

0x: Divider

10: Prescaler #0

11: Prescaler #1

This specifies the block acting as the frequency divider for deriving the count clock.

Bit 7: **Timer 0 Enable**

0: Disable

1: Enable

Setting this bit to “1” starts counting down using the mode specified by bit 5 (mode).

Bit 5: **Mode Select**

0: Cyclic mode

1: Single mode

Bits 4 to 2: **Divider Divisor (Ratio) [2:0]**

000: 1 (no divide) (1/1)

001: 4 (1/4)

010: 8 (1/8)

011: 16 (1/16)

100: 32 (1/32)

101: 64 (1/64)

110: 128 (1/128)

111: 256 (1/256)

Bit 1: **Immediate Load Request (cyclic mode only)**

0: Normal timing
1: Immediately

In cyclic mode, setting this bit to "1" cause a write to the Load Register to immediately reload the counter from that register instead of the next time that it reaches zero.

Note: Single Mode ignores this bit. Writing to the Load Register always produces an immediate reload.

Bit 0: **Timer 0 Interrupt Request Enable**

0: Disable (mask)
1: Enable

Timer 0 IRQ Flag Clear Register (TM0IRQ)							
TIM[0x0C]	Default = —						Write Only
arbitrary data							
15	14	13	12	11	10	9	8
arbitrary data							
7	6	5	4	3	2	1	0

Writing to this register resets the timer's interrupt request status flag, bit 0 in the Timer IRQ Status Register (TMIRQSTS), to "0." The value written is don't-care.

Timer 0 Port Output Control Register (TM0POUT)							
TIM[0x10]	Default = 0x0000						(Read/Write)
n/a RO							
15	14	13	12	11	10	9	8
n/a RO				Output Mode R/W		Output Enable R/W	Output Level R/W
7	6	5	4	3	2	1	0

Bits 3 to 2: **Output Mode**

00: Output level specified by bit 0
01: Underflow pulse
10: Toggle output level
11: Reserved

These pins specify the timer output mode when underflow has occurred.

Bit 1: **Output Enable**

0: Disable
1: Enable

Configuring the corresponding GPIO pin for port output and setting this bit to "1" drives that pin as specified by bits 3 to 2 (mode).

Bit 0: **Output Level**

This specifies the output for "00" in bits 3 to 2 (mode).

0: Low
1: High

Timer 1 Load Register (TM1LD)							
TIM[0x20]	Default = 0x0000						Read/Write
Timer 1 Load Value [15:8]							
15	14	13	12	11	10	9	8
Timer 1 Load Value [7:0]							
7	6	5	4	3	2	1	0

Bits 15 to 0: **TM1LD [15:0]**
Timer 1 Load Value

This register specifies the 16-bit value loaded into the counter as its count starting value.

Timer 1 Count Register (TM1CNT)							
TIM[0x24]	Default = 0x0000						Read Only
Timer 1 Current Count Value [15:8]							
15	14	13	12	11	10	9	8
Timer 1 Current Count Value [7:0]							
7	6	5	4	3	2	1	0

Bits 15 to 0: **TM1CNT [15:0]**

Timer 1 Current Count Value

This returns the current contents of the counter.

Timer 1 Control Register (TM1CTRL)							
TIM[0x28]	Default = 0x0000						(Read/Write)
n/a RO						Divider/Prescaler Select [1:0] R/W	
15	14	13	12	11	10	9	8
Timer 1 Enable R/W	n/a RO	Mode Select R/W	Divider Divisor [2:0] R/W			Immediate Load Request R/W	IRQ Request R/W
7	6	5	4	3	2	1	0

Bits 9 to 8: **Divider/Prescaler Select [1:0]**

0x: Divider

10: Prescaler #0

11: Prescaler #1

This specifies the block acting as the frequency divider for deriving the count clock.

Bit 7: **Timer 1 Enable**

0: Disable

1: Enable

Setting this bit to “1” starts counting down using the mode specified by bit 5 (mode).

Bit 5: **Mode Select**

0: Cyclic mode

1: Single mode

Bits 4 to 2: **Divider Divisor (Ratio) [2:0]**

000: No divide (1/1)

001: 4 (1/4)

010: 8 (1/8)

011: 16 (1/16)

100: 32 (1/32)

101: 64 (1/64)

110: 128 (1/128)

111: 256 (1/256)

Bit 1: **Immediate Load Request (cyclic mode only)**

0: Normal timing

1: Immediately

In cyclic mode, setting this bit to “1” cause a write to the Load Register to immediately reload the counter from that register instead of the next time that it reaches zero.

Note: Single Mode ignores this bit. Writing to the Load Register always produces an immediate reload.

Bit 0: **Timer 1 Interrupt Request Enable**

0: Disable (mask)

1: Enable

Timer 1 IRQ Flag Clear Register (TM1IRQ)							
TIM[0x2C]	Default = —						Write Only
arbitrary data							
	14	13	12	11	10	9	8
arbitrary data							
7	6	5	4	3	2	1	0

Writing to this register resets the timer's interrupt request status flag, bit 1 in the Timer IRQ Status Register (TMIRQSTS), to "0." The value written is don't-care.

Timer 1 Port Output Control Register (TM1POUT)							
TIM[0x30]	Default = 0x0000						(Read/Write)
n/a RO							
15	14	13	12	11	10	9	8
n/a RO				Output Mode R/W		Output Enable R/W	Output Level R/W
7	6	5	4	3	2	1	0

Bits 3 to 2: **Output Mode**

00: Output level specified by bit 0
01: Underflow pulse
10: Toggle output level
11: Reserved

These pins specify the timer output mode when underflow has occurred.

Bit 1: **Output Enable**

0: Disable
1: Enable

Configuring the corresponding GPIO pin for port output and setting this bit to "1" drives that pin as specified by bits 3 to 2 (mode).

Bit 0: **Output Level**

This specifies the output for "00" in bits 3 to 2 (mode).

0: Low
1: High

Timer 2 Load Register (TM2LD)							
TIM[0x40]	Default = 0x0000						Read/Write
Timer 2 Load Value [15:8]							
15	14	13	12	11	10	9	8
Timer 2 Load Value [7:0]							
7	6	5	4	3	2	1	0

Bits 15 to 0: **TM2LD [15:0]**

Timer 2 Load Value

This register specifies the 16-bit value loaded into the counter as its count starting value.

Timer 2 Count Register (TM2CNT)							
TIM[0x44]	Default = 0x0000						Read only
Timer 2 Current Count Value [15:8]							
15	14	13	12	11	10	9	8
Timer 2 Current Count Value [7:0]							
7	6	5	4	3	2	1	0

Bits 15 to 0: **TM2CNT [15:0]**

Timer 2 Current Count Value

This returns the current contents of the counter.

Timer 2 Control Register (TM2CTRL)							
TIM[0x48]	Default = 0x0000						(Read/Write)
n/a RO						Divider/Prescaler Select [1:0] R/W	
15	14	13	12	11	10	9	8
Timer 2 Enable R/W	n/a RO	Mode Select R/W	Divider Divisor [2:0] R/W			Immediate Load Request R/W	IRQ Request R/W
7	6	5	4	3	2	1	0

Bits 9 to 8: **Divider/Prescaler Select [1:0]**

0x: Divider
10: Prescaler #0
11: Prescaler #1

This specifies the block acting as the frequency divider for deriving the count clock.

Bit 7: **Timer 2 Enable**

0: Disable
1: Enable

Setting this bit to "1" starts counting down using the mode specified by bit 5 (mode).

Bit 5: **Mode Select**

0: Cyclic mode
1: Single mode

Bits 4 to 2: **Divider Divisor (Ratio) [2:0]**

000: No divide (1/1)
001: 4 (1/4)
010: 8 (1/8)
011: 16 (1/16)
100: 32 (1/32)
101: 64 (1/64)
110: 128 (1/128)
111: 256 (1/256)

Bit 1: **Immediate Load Request (cyclic mode only)**

0: Normal timing
1: Immediately

In cyclic mode, setting this bit to "1" cause a write to the Load Register to immediately reload the counter from that register instead of the next time that it reaches zero.

Note: Single Mode ignores this bit. Writing to the Load Register always produces an immediate reload.

Bit 0: **Timer 2 Interrupt Request Enable**

0: Disable (mask)
1: Enable

Timer 2 IRQ Flag Clear Register (TM2IRQ)							
TIM[0x4C]	Default = —						Write Only
arbitrary data							
15	14	13	12	11	10	9	8
arbitrary data							
7	6	5	4	3	2	1	0

Writing to this register resets the timer's interrupt request status flag, bit 2 in the Timer IRQ Status Register (TMIRQSTS), to "0." The value written is don't-care.

Timer 2 Port Output Control Register (TM2POUT)							
TIM[0x50]	Default = 0x0000						(Read/Write)
n/a RO							
15	14	13	12	11	10	9	8
n/a RO				Output Mode R/W		Outpu Enable R/W	Output Level R/W
7	6	5	4	3	2	1	0

Bits 3 to 2: **Output Mode**

00: Output level specified by bit 0
01: Underflow pulse
10: Toggle output level
11: Reserved

These pins specify the timer output mode when underflow has occurred.

Bit 1: **Output Enable**

0: Disable
1: Enable

Configuring the corresponding GPIO pin for port output and setting this bit to "1" drives that pin as specified by bits 3 to 2 (mode).

Bit 0: **Output Level**

This specifies the output for "00" in bits 3 to 2 (mode).

0: Low
1: High

Prescaler 0 Control Register (PS0CTRL)							
TIM[0xA0]	Default = 0x0000						(Read/Write)
Divider Divisor [2:0] R/W			n/a RO				
15	14	13	12	11	10	9	8
Prescaler 0 Load Value [7:0] R/W							
7	6	5	4	3	2	1	0

Bits 15 to 13: **Divider Divisor (Ratio) [2:0]**

Set the divisor using in the divider.

000: No divide (1/1)
001: 4 (1/4)
010: 8 (1/8)
011: 16 (1/16)
100: 32 (1/32)
101: 64 (1/64)
110: 128 (1/128)
111: 256 (1/256)

Bits 7 to 0: **Prescaler 0 Load Value [7:0]**

These bits specify the count value for the clock divided by the divider in the Prescaler 0.

Prescaler 1 Control Register (PS1CTRL)
TIM[0xA4] Default = 0x0000 (Read/Write)

Divider Divisor R/W			n/a RO				
15	14	13	12	11	10	9	8
Prescaler 1 Load Value [7:0] R/W							
7	6	5	4	3	2	1	0

Bits 15 to 13: **Divider Divisor (Ratio) [2:0]**

Set the divisor using in the divider.
000: No divide (1/1)
001: 4 (1/4)
010: 8 (1/8)
011: 16 (1/16)
100: 32 (1/32)
101: 64 (1/64)
110: 128 (1/128)
111: 256 (1/256)

Bits 7 to 0: **Prescaler 1 load value [7:0]**

These bits specify the count value for the clock divided by the divider in the Prescaler 1.

Timer IRQ Status Register (TMIRQSTS)
TIM[0xB0] Default = 0x0000 Read Only

n/a							
15	14	5	4	3	2	1	0
n/a			Reserved		Timer2 IRQ	Timer1 IRQ	Timer0 IRQ
7	6	5	4	3	2	1	0

Bits 4 to 3: **Reserved**

Bit 2: **Timer 2 IRQ**
Timer 2 Interrupt Status

0: There is no interrupt request.
1: There is an interrupt request.

"1" in a bit indicates an interrupt request from the corresponding timer. Writing to the Timer 2 IRQ Flag Clear Register resets the bit to "0."

Bit 1: **Timer 1 IRQ**
Timer 1 Interrupt Status

0: There is no interrupt request.
1: There is an interrupt request.

"1" in a bit indicates an interrupt request from the corresponding timer. Writing to the Timer 1 IRQ Flag Clear Register resets the bit to "0."

Bit 0: **Timer 0 IRQ**
Timer 0 Interrupt Status

0: There is no interrupt request.
1: There is an interrupt request.

"1" in a bit indicates an interrupt request from the corresponding timer. Writing to the Timer 0 IRQ Flag Clear Register resets the bit to "0."

22.5 Loading (and Reloading) Timer Counters

22.5.1 Timer/Counter Modes

There are two timer modes. The following describes the procedures for specifying Load Values.

(1) Cyclic Mode

The counter repeatedly counts down to zero from the value specified in the Load Register.

Timer Control Register bit 1 (immediate load request) offers two choices for reload timing.

a) Setting that bit to “1” and writing to the Load Register immediately forcibly reloads the counter regardless of the setting in Timer Control Register bit 7 (timer enable).

b) Setting that bit to “0” produces an interrupt request when the counter counts down to zero. The counter then reloads with the value specified in the Load Register and resumes counting down.

Note: Specifying FFFFh in the Load Register produces a free-running counter.

(2) Single Mode

The counter counts down to zero from the value specified in the Load Register. When the counter reaches zero, there is an interrupt request, and the counter stops. Clearing the interrupt request by writing to the timer’s IRQ Flag Clear Register does not restart the counter. Single mode reload timing ignores Timer Control Register bits 1 (immediate load request bit) and 7 (timer enable). Writing to the Load Register always produces an immediate reload.

22.6 Sample: Timer Clock Settings (1 kHz, 1 MHz)

22.6.1 Setting for Divider and prescaler

The frequency divisor and load value settings in the Prescaler Control Registers offer the three timers a choice of two timer periods. Alternatively, a timer can skip the prescalers and use just the divider and the frequency divisor in its Control Register.

The following Table gives sample Prescaler Control Register settings for deriving 1 ms and 1 μs periods (1 kHz and 1 MHz frequencies, respectively) from a timer input clock (TINCLK) frequency of 6 MHz—1/8 that for an APB bus running at (PCLK) 48 MHz.

Note that the prescaler setting is one less than the divisor: 0 for 1/1, 1 for 1/2, 2 for 1/3, etc.

Table 22.2 Prescaler Register Settings for 1 ms and 1 μs Periods

Period	Bits 15 to 13 (Divisor)	Prescaler Load Value (Bits 7 to 0)	Frequency
1 ms	100b (32)	0xBA (187)	1.002673 kHz
	101b (64)	0x5D (94)	0.997340 kHz
1 μs	000b (1)	0x05 (6)	1.000000 MHz

Note: These Settings are for f_{TINCLK}=6MHz (f_{PCLK}=48MHz).

For other period, use the following formula and adjust the frequency divisor and load value settings in the Prescaler Control Register to produce the frequency closest to the desired one.

$$f = \underbrace{f_{PCLK}}_{f_{TINCLK}} \div 8 \div \underbrace{\text{PSnCTRL[15:13] setting}}_{\text{Frequency divisor}} \div \underbrace{\text{PSnCTRL[7:0] setting}}_{\text{Prescaler load value}}$$

Examples (from Table 22.2)

1) 1 kHz ≠ 48 MHz ÷ 8 ÷ 32 ÷ 187 = 1.002673 kHz

2) 1 kHz ≠ 48 MHz ÷ 8 ÷ 64 ÷ 94 = 0.997340 kHz

3) 1 MHz = 48 MHz ÷ 8 ÷ 1 ÷ 6 = 1.000000 MHz

22.7 Timing Charts

22.7.1 Cyclic Mode, Immediate Load Request

The following is the timing chart for a cyclic mode immediate load request, which produces an immediate reload. The prescaler has a counter enable cycle setting of 1/3.

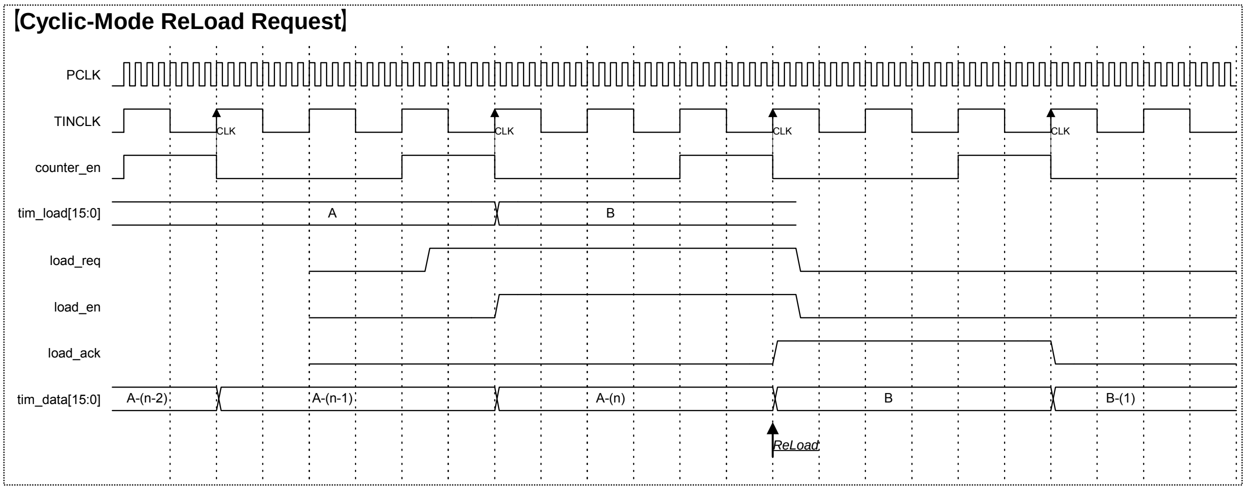


Fig.22.2 Cyclic Mode, Immediate Load Request

22.7.2 Cyclic Mode, Normal Reload

The following is the timing chart for normal cyclic count down operation, with no immediate load request. The frequency divisor is 1. When the counter reaches zero, there is an interrupt request, and the counter reloads.

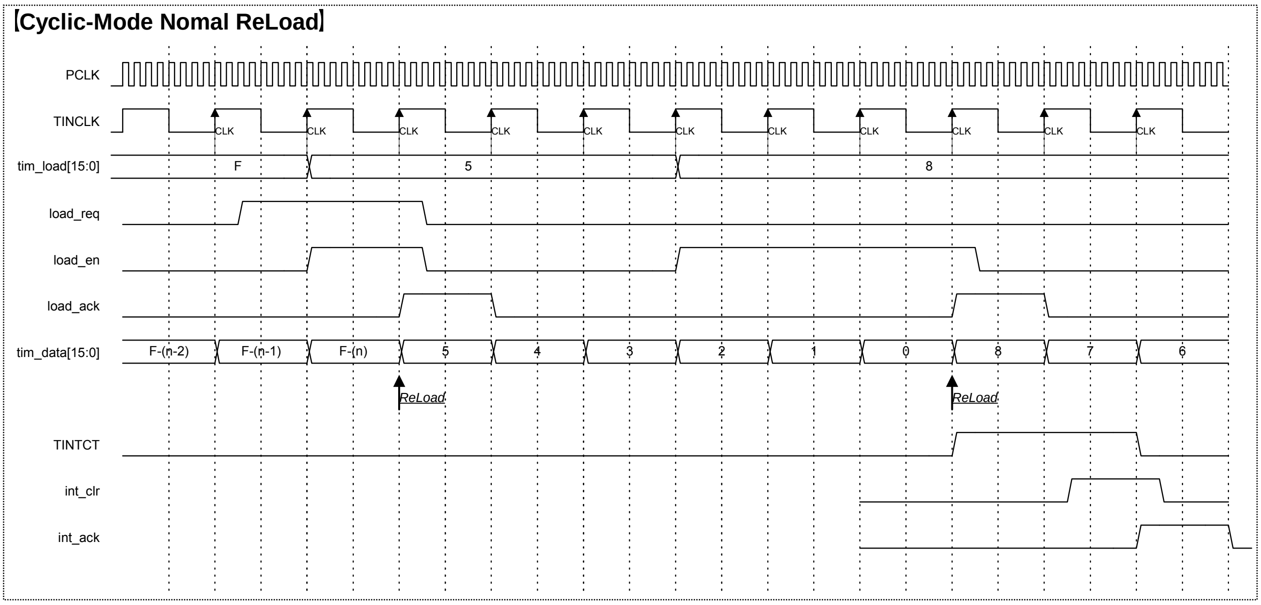


Fig.22.3 Cyclic Mode, Normal Reload

22.7.3 Single Mode, Normal Reload

The following is the timing chart for single mode after writing to the Load Register. The frequency divisor is 1. When the counter reaches zero, there is an interrupt request.

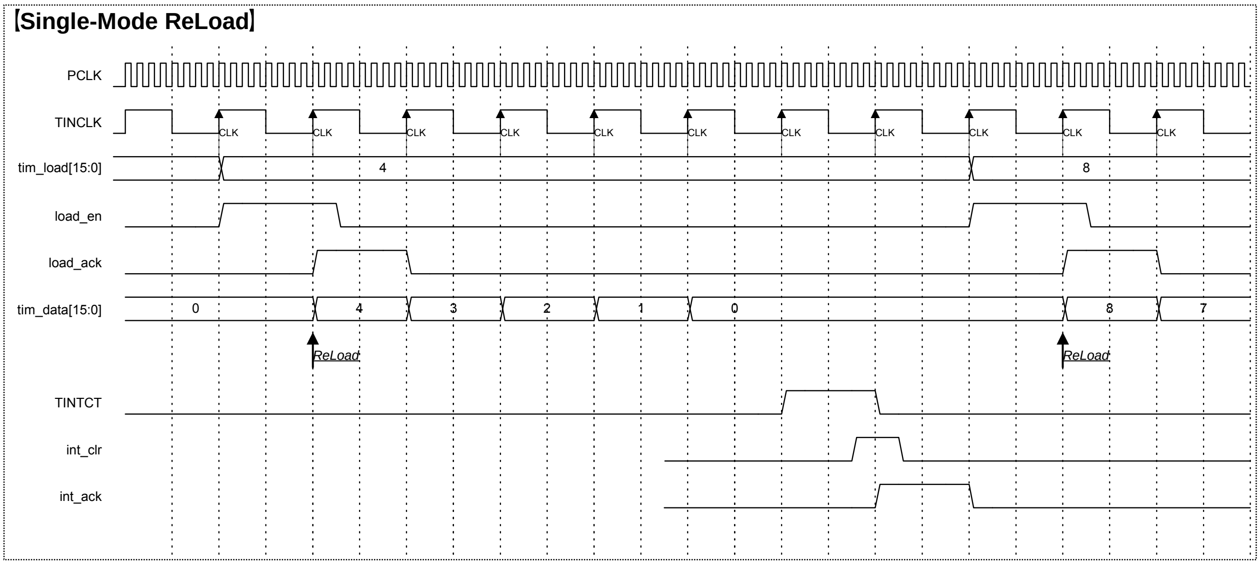


Fig.22.4 Single Mode, Normal Reload

22.7.4 Port Output

The following timing charts are for port output triggered by the counter reaching zero. They show the waveforms with the default settings (1/1) for both the divider and prescaler blocks.

(1) Mode 00b: Output Level specified by bit 0

When the counter reaches zero, the timer drives the output at the level specified by Port Output Control Register bit 0 (output value)

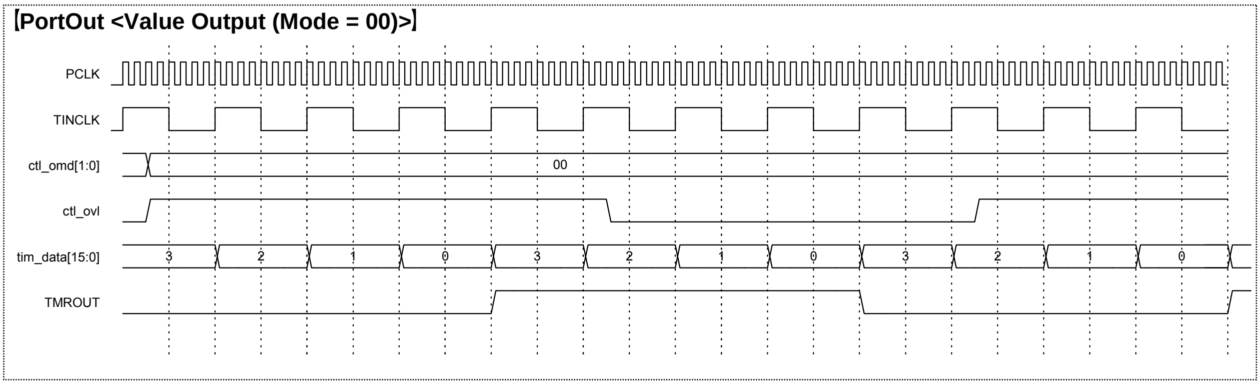


Fig.22.5 Specified Value Output

(2) Mode 01b: Underflow pulse

When the counter reaches zero, the timer outputs its underflow (TMROUT) signal. The pulse width is the same as the down count data width. It coincides with the first count cycle after the counter reaches zero and reloads—in the figure, when the counter is set to "3."

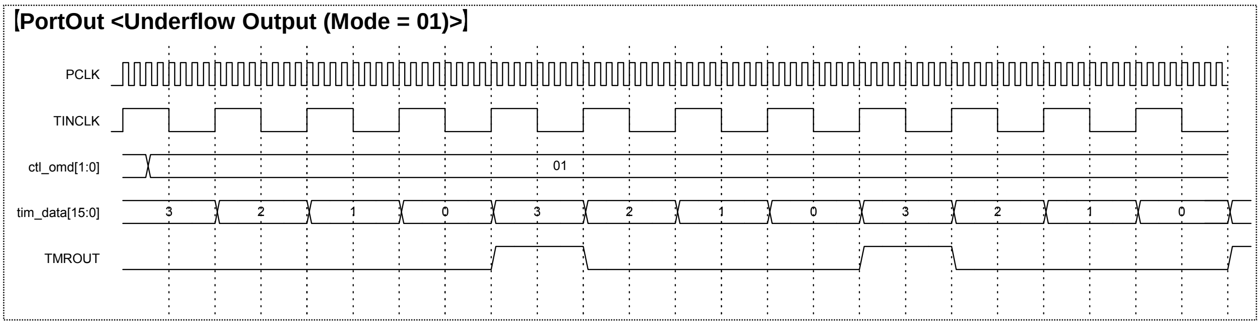


Fig.22.6 Underflow Pulse

(3) Mode 10b: Toggle output Level

When the counter reaches zero, the timer reverses the output signal.

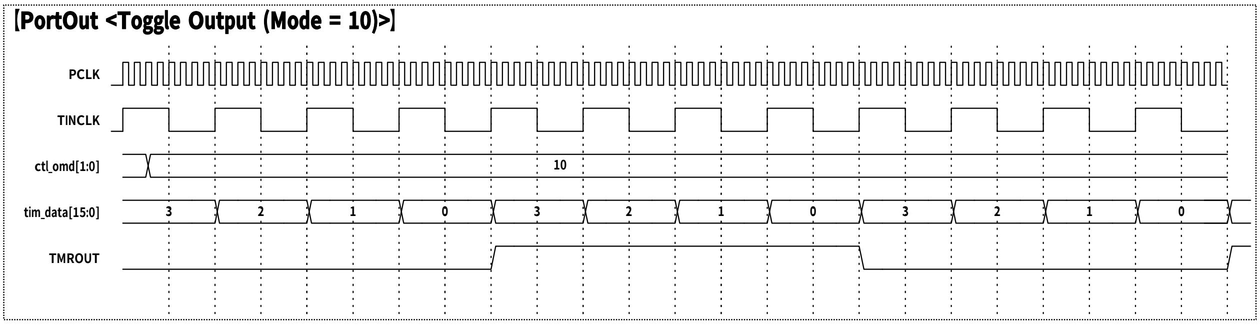


Fig.22.7 Toggle Output

23. REAL-TIME CLOCK (RTC)

23.1 Overview

This module passes the 32.768 kHz input clock signal through first a prescaler with 256 Hz output and then an 8-bit frequency divider to counters (seconds, minutes, hours, and days) that mark time for clock, stopwatch, and other timekeeping functions. The firmware has read access to these counters.

This module also provides interrupt requests with various frequencies for use as periodic interrupt requests, wake-up sources, etc.: 32 Hz, 8 Hz, and 2 Hz plus every second, minute, hour, or day. This support includes alarms at specific minutes, hours, and days for use as wake-up sources, user-specified alarms as part of calendar software, etc.

This module remains operational as long as it receives the 32.768 kHz input clock signal—even when the CPU and other on-chip peripherals are on standby.

System resets do not affect timekeeping. It continues during external reset input.

23.2 Block Diagram

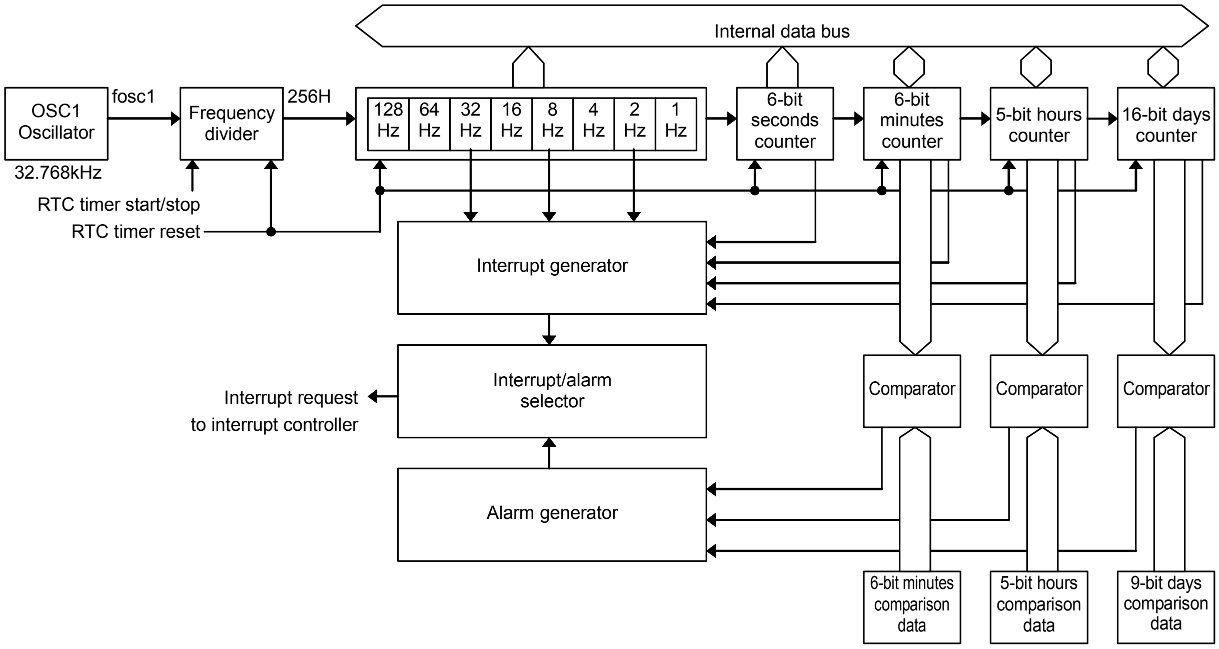


Fig.23.1 RTC Block Diagram

23.3 External Pins

This module interacts with no external pins.

23.4 Registers

23.4.1 Register List

The base address for these registers is 0xFFFF_8000.

Table 23.1 RTC Register List (Base Address: 0xFFFF_8000)

Address Offset	Register Name	Default Value	R/W	Data Access Size (Bits)
0x000	RTC Run/Stop Control Register	x--- ---xb	(R/W)	8
0x004	RTC Interrupt Register	1110 0000b	R/W	8
0x008	RTC Timer Divider Register	xxxx xxxxb	R/(W)	8
0x00C	RTC Second Counter Register	--xx xxxxb	R/W	8
0x010	RTC Minute Counter Register	--xx xxxxb	R/W	8
0x014	RTC Hour Counter Register	---x xxxxb	R/W	8
0x018	RTC Day Counter Register	0x XXXX	R/W	16
0x020	RTC Alarm Minute Compare Register	--xx xxxxb	R/W	8
0x024	RTC Alarm Hour Compare Register	---x xxxxb	R/W	8
0x028	RTC Alarm Day Compare Register	x xxxx xxxxb	R/W	16
0x02C	RTC Test Register	---0 0000b	R/W	8
0x030	RTC Prescaler Register	-xxx xxxxb	R/(W)	8
0x034	RTC Test Clock Register	---- ----b	WO	8

23.4.2 Detailed Register Descriptions

In the absence of any indication to the contrary, set all reserved bits to “0.” Ignore the values returned by reads.

RTC Run/Stop Control Register (8 bits)

RTC[0x000] Default = x--- ---x b Read/Write

BUSY	BUSYWIDTH		Reserved			TCRST	TCRUN
x	xx		—			—	0
RO	R/W					WO	R/W
7	6	5	4	3	2	1	Bit0

x: Undefined binary digit

Bit 7: **BUSY (Read Only)**

Read/write access to the RTC Registers is only possible when this bit is “0.”

“1” in this bit indicates that the block is in the middle of an internal update cycle. This update cycle arises once a second and lasts for the interval specified in bits 6 to 5 (BUSYWIDTH). If this bit is “1,” successful access therefore requires waiting that interval before trying again.

Bits 6 to 5: **BUSYWIDTH**

Busy Interval

This specifies the approximate length of the internal update cycle.

00: 244 µs
01: 122 µs
10: 61 µs
11: Reserved

A system reset does not initialize this bit.

To ensure proper operation, stop this module before reset. This module only accepts resets when it is stopped.

Bits 4 to 2: **Reserved**

Bit 1: **TCRST**

RTC Counter Reset (Write Only)

0: Normal operation
1: Reset

Reads always return "0."

Simultaneously writing "1" to this bit and "0" to bit 0 in the RTC run/stop control register resets the prescalers and dividing timer.

To ensure proper operation, stop this module before reset. This module only accepts resets when it is stopped.

Bit 0: **TCRUN**

RTC Status/Control

reads 0: Idle
1: Busy
writes 0: Stop
1: Start*

* There is a delay of approximately 30 to 61 μs while the block synchronizes with the 32 kHz clock signal.
Immediately after a power on reset, this module is in an undefined state.
A system reset does not initialize this bit.

RTC Interrupt Register (8 bits)
RTC[0x004] Default = 0xE0 Read/Write

TCISE[2:0]			TCASE[2:0]			TCIF	TCAF
7	6	5	4	3	2	1	0

Bits 7 to 5: **TCISE[2:0]**

Interrupt Request Source

000: Carry from 32 Hz counter (32 times per second)
001: Carry from 8 Hz counter (8 times per second)
010: Carry from 2 Hz counter (twice per second)
011: Carry from 1 Hz counter (once per second)
100: Carry from Minutes counter (once per minute)
101: Carry from Hours counter (once per hour)
110: Carry from Days counter (once per day)
111: No interrupt requests (default)

This use of counter carries means that there are interrupt requests the specified fixed interval apart.

A system reset initializes this field to 111b.

Bits 4 to 2: **TCASE[2:0]**

RTC Alarm Source Enable

000: No alarm (default)
xx1: Minutes alarm
x1x: Hours alarm
1xx: Days alarm

Setting a bit to "1" produces an interrupt request when there is a match between the counter and the alarm comparison data for the corresponding alarm source. Setting multiple bits produces one when there is a match in any pair.

The interrupt request continues as long as there is such a match. Writing "1" to bit 0 (TCAF) does not clear the interrupt request here. To wait for the next interrupt request with the same alarm settings, disable all sources until the counter for the smallest one changes. If the alarm uses hour and day settings, for example, avoid using the same alarm source settings for an hour after an interrupt request.

A system reset initializes this field to 000b.

Bit 1: **TCIF**

RTC Timer Interrupt Request Flag

reads 0: None pending
1: Pending
writes 0: (ignored)
1: Clear

A system reset initializes this bit to “0.”

Bit 0: **TCAF**

Alarm Interrupt Request Flag

reads 0: None pending
1: Pending
writes 0: (ignored)
1: Clear

A system reset initializes this bit to “0.”

RTC Timer Divider Register (8 bits)
RTC[0x008] Default = xxxx xxxx b Read/(Write)

TCD7	TCD6	TCD5	TCD4	TCD3	TCAD2	TCD1	TCD0
7	6	5	4	3	2	1	0

x: Undefined binary digit

Bit 7 (TCD7):	**1 Hz indicator**
Bit 6 (TCD6):	**2 Hz indicator**
Bit 5 (TCD5):	**4 Hz indicator**
Bit 4 (TCD4):	**8 Hz indicator**
Bit 3 (TCD3):	**16 Hz indicator**
Bit 2 (TCD2):	**32 Hz indicator**
Bit 1 (TCD1):	**64 Hz indicator**
Bit 0 (TCD0):	**128 Hz indicator**

0: Low
1: High

“1” in a bit indicates the contents in the corresponding 128 to 1 Hz counter.
Writing “1” to RTC Run/Stop Control Register (RTC[0x000]) bit 1 (TCRST) resets this register to zero.
Firmware can consider this register an up counter.
This register does not include circuitry for synchronizing it with the 32 kHz system clock, so the firmware must not trust reads until they return the same value twice in a row.
Writes are only possible when the RTC Test Register enables writes to divider counters (Test mode: 0000).
Note, however, that this hardware test function is outside the scope of these specifications, so does not guarantee proper results for such writes.

RTC Seconds Counter Register (8 bits)
RTC[0x00C] Default = --xx xxxx b Read/Write

Reserved		TCMD[5:0]					
		R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0

x: Undefined binary digit

Bits 7 to 6: **Reserved (0)**

Bits 5 to 0: **TCMD[5:0]**

Seconds (0-59)

Indicates the binary data of seconds.
TCMD5 = MSB, TCMD0 = LSB.
Writing a value bigger than 59 sets this field to zero.
A system reset does not initialize this register.

RTC Minutes Counter Register (8 bits)

RTC[0x010] Default = --xx xxxx b Read/Write

Reserved		TCHD[5:0]					
		R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0

x: Undefined binary digit

Bits 7 to 6: **Reserved (0)**

Bits 5 to 0: **TCHD[5:0]**

Minutes (0-59)

Indicates the binary data of minutes.

TCHD5 = MSB, TCHD0 = LSB.

Writing a value bigger than 59 sets this field to zero.

A system reset does not initialize this register.

RTC Hours Counter Register (8 bits)

RTC[0x014] Default = ---x xxxx b Read/Write

Reserved			TCDD[4:0]				
—	—	—	x	x	x	x	x
			R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0

x: Undefined binary digit

Bits 7 to 5: **Reserved (0)**

Bits 4 to 0: **TCDD[4:0]**

Hours (0-23)

Indicates the binary data of hours.

TCDD4 = MSB, TCDD0 = LSB.

Writing a value bigger than 23 sets this field to zero.

A system reset does not initialize this register.

RTC Days Counter Register (16 bits)

RTC[0x018] Default = 0xXXXX Read/Write

TCND [15:8]							
15	14	13	12	11	10	9	8
TCND [7:0]							
7	6	5	4	3	2	1	0

x: Undefined hexadecimal digit

Bits 15 to 0: **TCND[15:0]**

Days (0-65535)

Indicates the binary data of days.

TCND15 = MSB, TCND0 = LSB.

A system reset does not initialize this register.

RTC Alarm Minutes Compare Register (8 bits)
RTC[0x020] Default = --xx xxxx b Read/Write

Reserved		TCCH[5:0]					
—	—	x	x	x	x	x	x
		R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0

x: Undefined binary digit

Bits 7 to 6: **Reserved (0)**

Bits 5 to 0: **TCCH[5:0]**
Minutes (0-59)
Indicates the binary data of alarm minutes.
TCCH5 = MSB, TCCH0 = LSB.
Writing a value bigger than 59 succeeds, but prevents any matches with the corresponding counter.
A system reset does not initialize this register.

RTC Alarm Hours Compare Register (8 bits)
RTC[0x024] Default = ---x xxxx b Read/Write

Reserved			TCCD[4:0]				
—	—	—	x	x	x	x	x
			R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0

x: Undefined binary digit

Bits 7 to 5: **Reserved (0)**

Bits 4 to 0: **TCCD[4:0]**
Hours (0-23)
Indicates the binary data of alarm hours.
TCCD4 = MSB, TCCD0 = LSB.
Writing a value bigger than 23 succeeds, but prevents any matches with the corresponding counter.
A system reset does not initialize this register.

RTC Alarm Days Compare Register (16 bits)
RTC[0x028] Default = ---- ---x xxxx xxxx b Read/Write

Reserved							TCCN8
—	—	—	—	—	—	—	x
							R/W
15	14	13	12	11	10	9	8
TCCN[7:0]							
x	x	x	x	x	x	x	x
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0

x: Undefined binary digit

Bits 15 to 9: **Reserved (0)**

Bits 8 to 0: **TCCN[8:0]**
Days (0-511)
Indicates the binary data of alarm days.
TCCN8 = MSB, TCCN0 = LSB.
A system reset does not initialize this register.

RTC Test Register (8 bits)							
RTC[0x02C]	Default = ---0 0000 b						Read/Write
Reserved			RTST4	RTST3	RTST2	RTST1	RTST0
—			R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0

Bits 7 to 5: **Reserved (0)**

Bits 4 to 1: **RTST[4:1]**

Test Mode

0000: Enable writes to divider counters

xx10: Use test clock

This mode uses the pulse generated by writing to the RTC Test Clock Register instead of the 32 kHz input clock signal.

x1xx: Bypass counter carries

This mode uses carries from the seconds, minutes, hours, and days counters as clock pulses.

1xxx: Bypass divider counter carries

This mode uses carries from the divider counter as clock pulses.

A system reset initializes this field to 0000b.

Bit 0: **RTST0**

Test Mode Enable

Writing first “1,” then “0” switches to the test mode.

A system reset initializes both this bit and the record of the last write to “0.”

RTC Prescaler Register (8 bits)							
RTC[0x030]	Default = -xxx xxxx b						Read/(Write)
Reserved	TCP [6:0]						
—	R/(W)						
7	6	5	4	3	2	1	0

Bits 7: **Reserved**

Bits 6 to 0: **TCP[6:0]**

Prescaler Contents

Indicates the binary data of prescaler:

TCP6 = MSB, TCP0 = LSB.

A system reset does not initialize this register.

Writes are only possible when the RTC Test Register enables writes to divider counters. Note, however, that this hardware test function is outside the scope of these specifications, so does not guarantee proper results for such writes.

RTC Test Clock Register (8 bits)							
RTC[0x034]	Default = -xxx xxxx b						Write Only
TSTCLK							
WO							
7	6	5	4	3	2	1	0

Bits 7 to 0: **TSTCLK**

Test Clock

In test clock mode a write to this register produces a clock pulse. The value written is don’t-care. Note, however, that this hardware test function is outside the scope of these specifications.

23.5 Configuring Real-Time Clock Registers

The Test Register, Test Clock Register, and test modes using them are outside the scope of these specifications. The following description therefore ignores them entirely.

23.5.1 After Power On Reset

Immediately after a power on reset, this module is in an undefined state, so the firmware must start by writing “0” to RTC Run/Stop Control Register bit 0 to stop the real-time clock. The bit goes to “1” to indicate that the block is operational—in this case, synchronizing with the 32 kHz circuitry’s run/stop control signal. The firmware must therefore wait until that bit returns to “0,” indicating that the block is currently stopped. Such confirmation is necessary because it takes 30 to 61 μs to synchronize register settings with the 32 kHz circuitry.

The next step is writing “10b” to RTC Run/Stop Control Register bits 1 to 0 to reset the prescalers and 128 to 1 Hz counters to zero because their contents are undefined when the power is first applied. Note that this step involves simultaneously writing “0” (stop) to bit 0 and not just writing “1” (reset) to bit 1.

The block is now ready to accept settings for the functionality used: counters (seconds, minutes, hours, and days) for time and other timekeeping functions, alarm comparison data (minutes, hours, and days) for alarms, interrupt request settings for period and interrupts, etc.

The firmware must also specify a busy interval (BUSYWIDTH) greater than the time required to ensure completion of firmware processing. If such processing, including any suspensions by other processes, completes within 100 μs, for example, the setting to use is “01” (122 μs).

Note: The real-time clock is still stopped at this point, so the above setting steps—from counters to BUSYWIDTH—can be in any order.

If the system uses real-time clock interrupt requests, clear interrupt request and alarm source status flags in this module and enable real-time clock interrupt requests in the interrupt controller module.

Finally, write “1” to RTC Run/Stop Control Register (RTC[0x000]) bit 0 to start the real-time clock. Operation starts 30 to 61 μs later.

23.5.2 Stopping and Restarting

To pause the real-time clock, write “0” to RTC Run/Stop Control Register bit 0 and wait until that bit goes to “0” (approximately 30 to 61 μs). The counters retain their current contents, ready for further counting when operation resumes.

To resume operation, clear any interrupt request flags and write “10” to RTC Run/Stop Control Register bits 1 to 0 to reset the prescalers and only the 128 to 1 Hz counters.

23.5.3 Reconfiguring On the Fly

RTC Run/Stop Control Register bit 0 offers the only means for making changes during operation. In particular, note that the firmware cannot reset the prescalers and 128 to 1 Hz counters during operation.

RTC Interrupt Register settings and Alarm Compare Register (minutes, hours, and days) can be modified at any time, but always disable real-time clock interrupt requests in the interrupt controller for the duration and clear any interrupt request flags before re-enabling. This bracketing is necessary to prevent modifications from triggering spurious interrupt requests.

Counter Registers (seconds, minutes, hours, and days) can be modified whenever the BUSY bit is “0.” Disabling all interrupt requests from other peripherals for the duration to prevent interruption by other processes is necessary enough to check the BUSY bit once because the register modification is instantaneous.

23.5.4 System Reset During Operation

A system reset initializes only the interrupt request circuitry and the interrupt request settings. The other real-time clock continue operating without initialization: the divider block, the prescalers, the counters (seconds, minutes, hours, and days), or the alarm comparison data (minutes, hours, and days). In other words, the block correctly keeps time and performs its other timekeeping functions, but stops sending interrupt requests. The firmware must therefore set up the relevant registers all over again.

23.5.5 Important Notes on Programming

- A power on reset initializes only interrupt requests. It is up to the firmware to specify the alarm compare data, operation control settings, and other register contents using the procedures in Section 23.5.1 “After Power On Reset.”
- To reset the prescalers and 128 to 1 Hz counters, stop the real-time clock and write “10b” to RTC Run/Stop Control Register bits 1 to 0. Note that this step involves simultaneously writing “0” (stop) to bit 0 and not just writing “1” (reset) to bit 1.
- Before modifying the interrupt request source and alarm source settings, always disable real-time clock interrupt requests in the interrupt controller for the duration and clear any interrupt request flags before re-enabling. This bracketing is necessary to prevent modifications from triggering spurious interrupt requests.
- A system reset initializes only the interrupt request circuitry and the interrupt request settings. The firmware must therefore set up the relevant registers all over again.

24. WATCHDOG TIMER (WDT)

24.1 Overview

This module monitors whether the system is running properly with a programmable 16-bit counter. This counter counts down from its initial setting and, when it reaches zero, triggers an interrupt request or reset request as specified in WDT[0x08] bit 4 (watchdog timer output). The firmware tries to prevent this by reloading the counter at regular intervals. In other words, an interrupt request or reset request indicates that the firmware is no longer executing normally.

The firmware can read the counter contents at any time. The counter resets to 0xFFFF after reaching zero, after a system reset, and when the firmware writes “0” to WDT[0x08] bit 5 (watchdog timer enable) to stop the counter.

The watchdog timer remains in operation in HALT mode.

The watchdog timer uses a built-in prescaler to derive its programmable count clock frequency from the APB clock (PCLK) input signal.

24.2 Block Diagram

This module consists of read, load, and control registers (and APB bus interface to them), a 16-bit count down counter, and a prescaler containing an 11-bit count down counter.

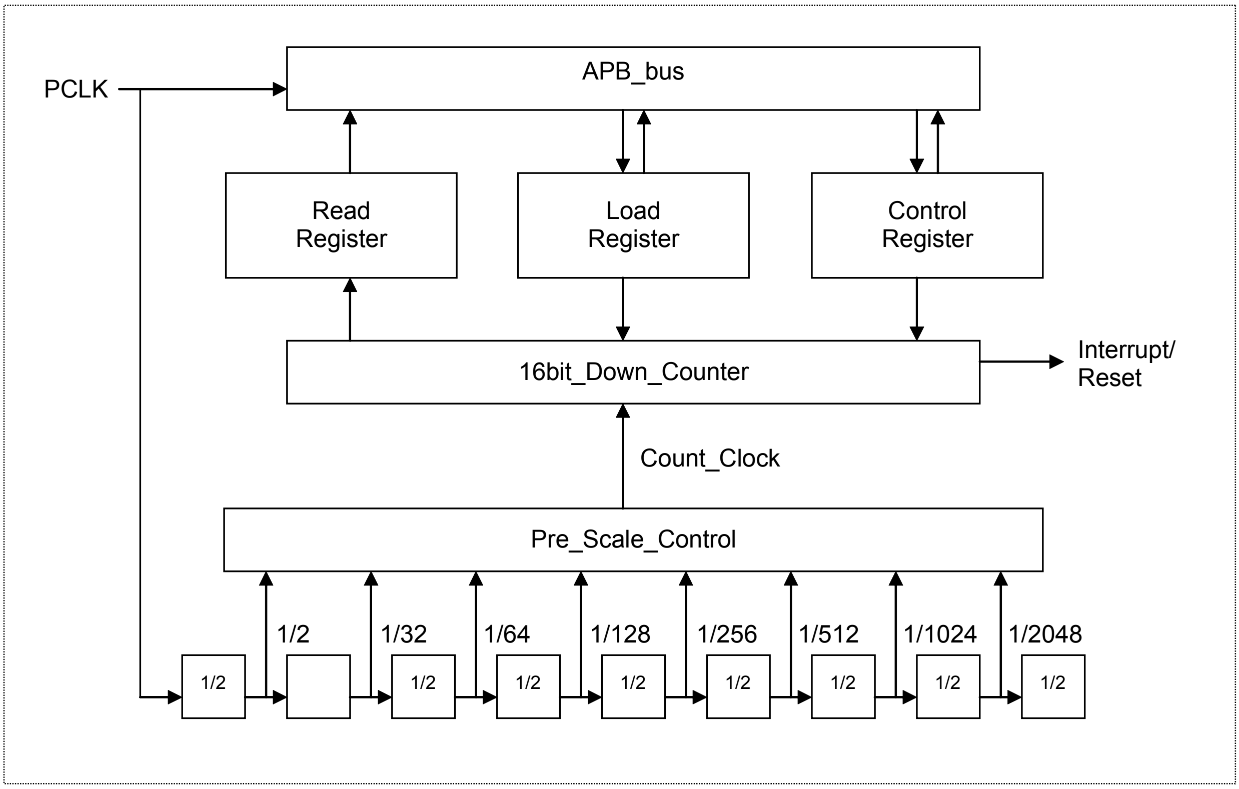


Fig.24.1 WDT Block Diagram

24.3 External Pins

This module interacts with no external pins.

24.4 Registers

24.4.1 Register List

The base address for these registers is 0xFFFF_C000.

Table 24.1 WDT Register List (Base Address: 0xFFFF_C000)

Address Offset	Register Name	Default Value	R/W	Data Access Size (Bits)
0x00	Watchdog Timer Load Register	0x0000_FFFF	R/W	16 (/32)
0x04	Watchdog Timer Count Register	0x0000_FFFF	RO	16 (/32)
0x08	Watchdog Timer Control Register	0x0000_0000	R/W	16 (/32)

24.4.2 Detailed Register Descriptions

In the absence of any indication to the contrary, register bits not labeled reserved all is set as “0.”

Watchdog Timer Load Register
WDT[0x00] Default = 0x0000_FFFF Read/Write

n/a															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Timer Load Value															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 15 to 0: **Timer Load Value [15:0]**
Writing to this register loads that data into the counter.

Watchdog Timer Count Register
WDT[0x04] Default = 0x0000_FFFF Read Only

n/a															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Current Counter Value															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 15 to 0: **Current Counter Value [15:0]**
Reads return the current counter value.

Watchdog Timer Control Register
WDT[0x08] Default = 0x0000_0000 Read/Write

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
n/a															
Reserved (Fixed at A5h)								n/a	WDT Status (RO)	WDT Enable	WDT Output	n/a	Prescaler Frequency Divisor Setting		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits 15 to 8: **Reserved**

Always write 0xA5.

Bit 6: **Watchdog Timer Status (Read Only)**

0: Nonzero
1: Zero

This bit goes to “1” when the counter goes to “0.” It stays “1” until the firmware writes “0” to bit 5 in this register (watchdog timer enable) to reset the watchdog timer.

Bit 5: **Watchdog Timer Enable**

0: Disable
1: Enable

Writing “0” to this bit stops the counter, resets the counter to 0xFFFF, and resets bit 6 in this register (watchdog timer status) to “0.”
Writing “1” to this bit immediately starts the watchdog timer counting down.

Bit 4: **Watchdog Timer Output**

0: Reset request
1: Interrupt request

This specifies the signal output indicating that the counter has gone to “0.”
Internal reset request signal (WRST) is a High level to the system controller. This signal returns to Low level when the counter changes to a nonzero value.
The internal watchdog timer interrupt request signal (WINT) is High level output to the interrupt controller. This signal returns to Low level when the firmware writes “0” (disable) to bit 5 in this register (watchdog timer enable).

Bits 2 to 0: **Prescaler frequency divisor Setting [2:0]**

This field specifies the frequency divisor that the prescaler uses to derive the count clock from the APB clock (PCLK) input signal.

Bits [2:0]	Prescaler Frequency Divisor *n*
000	2
001	32
010	64
011	128
100	256
101	512
110	1024
111	2048

*: Prescaler divisor $n = 2$: When Bits[2:0] = 0.
$2(4 + \text{Bits}[2:0])$: When Bits[2:0] ≠ 0.
count clock frequency = $f_{\mathrm{PCLK}} \div n$
where
f_{PCLK} is the APB system clock (PCLK) frequency.

25. GENERAL PURPOSE I/O (GPIO)

25.1 Overview

This block provides general purpose I/O (GPIO) pins and registers for switching these pins between GPIO and alternate functions.

This block has the following features.

- Eight 8-bit GPIO ports: GPIOA to GPIOH
- Individual I/O direction control for each GPIO pin
- Alternate I/O functionality available for some pins
- Pin function registers for switching these multifunction I/O pins
- Support for using GPIOA and GPIOB pin inputs as interrupt request sources
- Choice of triggers for these GPIOA/GPIOB interrupt requests: falling edges, rising edges, Low level, or High level

25.2 External Pins

This block interacts with the following external pins.

Port	Pin Name	I/O Direction	GPIO Pin Function	Alternate Functions		
				Shared	Function 1	Function 2
A	GPIOA0	I/O	General-purpose I/O pin A0		TXD0	
	GPIOA1	I/O	General-purpose I/O pin A1		RXD0	
	GPIOA2	I/O	General-purpose I/O pin A2		SPI_SS	TXD1
	GPIOA3	I/O	General-purpose I/O pin A3		SPI_SCLK	RXD1
	GPIOA4	I/O	General-purpose I/O pin A4		SPI_MISO	
	GPIOA5	I/O	General-purpose I/O pin A5		SPI_MOSI	
	GPIOA6	I/O	General-purpose I/O pin A6		SCL	
	GPIOA7	I/O	General-purpose I/O pin A7		SDA	
B	GPIOB0	I/O	General-purpose I/O pin B0	INT0/FIQ1		I2S0_WS
	GPIOB1	I/O	General-purpose I/O pin B1	INT1	RTS0#	I2S0_SCK
	GPIOB2	I/O	General-purpose I/O pin B2	INT2	CTS0#	I2S0_SD
	GPIOB3	I/O	General-purpose I/O pin B3	INT3	Timer0out	I2S1_SD
	GPIOB4	I/O	General-purpose I/O pin B4	INT4	Timer1out	
	GPIOB5	I/O	General-purpose I/O pin B5	INT5	Timer2out	
	GPIOB6	I/O	General-purpose I/O pin B6	INT6	MA22	I2S1_SCK
	GPIOB7	I/O	General-purpose I/O pin B7	INT7	MA23	I2S1_WS
C	GPIOC0	I/O	General-purpose I/O pin C0		CMDATA0	
	GPIOC1	I/O	General-purpose I/O pin C1		CMDATA1	
	GPIOC2	I/O	General-purpose I/O pin C2		CMDATA2	
	GPIOC3	I/O	General-purpose I/O pin C3		CMDATA3	
	GPIOC4	I/O	General-purpose I/O pin C4		CMDATA4	
	GPIOC5	I/O	General-purpose I/O pin C5		CMDATA5	
	GPIOC6	I/O	General-purpose I/O pin C6		CMDATA6	
	GPIOC7	I/O	General-purpose I/O pin C7		CMDATA7	
D	GPIOD0	I/O	General-purpose I/O pin D0	INT8	MA20	
	GPIOD1	I/O	General-purpose I/O pin D1		MA21	
	GPIOD2	I/O	General-purpose I/O pin D2		CFCE2#	
	GPIOD3	I/O	General-purpose I/O pin D3		CFCE1#	
	GPIOD4	I/O	General-purpose I/O pin D4		CMVREF	
	GPIOD5	I/O	General-purpose I/O pin D5		CMHREF	
	GPIOD6	I/O	General-purpose I/O pin D6		CMCLKOUT	
	GPIOD7	I/O	General-purpose I/O pin D7		CMCLKIN	

Port	Pin Name	I/O Direction	GPIO Pin Function	Alternate Functions		
E	GPIOE0	I/O	General-purpose I/O pin E0		CFIORD#	I2S0_SD
	GPIOE1	I/O	General-purpose I/O pin E1		CFIOWR#	I2S0_SCK
	GPIOE2	I/O	General-purpose I/O pin E2		CFWAIT#/MWAIT#	
	GPIOE3	I/O	General-purpose I/O pin E3		CFRST	I2S0_WS
	GPIOE4	I/O	General-purpose I/O pin E4		CFIREQ	
	GPIOE5	I/O	General-purpose I/O pin E5		CFSTSCHG#	I2S1_SD
	GPIOE6	I/O	General-purpose I/O pin E6		CFDEN#	I2S1_SCK
	GPIOE7	I/O	General-purpose I/O pin E7		CFDDIR	I2S1_WS
F	GPIOF0	I/O	General-purpose I/O pin F0		MII_CRS	
	GPIOF1	I/O	General-purpose I/O pin F1		MII_COL	
	GPIOF2	I/O	General-purpose I/O pin F2		MII_TXD3	
	GPIOF3	I/O	General-purpose I/O pin F3		MII_TXD2	
	GPIOF4	I/O	General-purpose I/O pin F4		MII_TXD1	
	GPIOF5	I/O	General-purpose I/O pin F5		MII_TXD0	
	GPIOF6	I/O	General-purpose I/O pin F6		MII_TXEN	
	GPIOF7	I/O	General-purpose I/O pin F7		MII_TXCLK	
G	GPIOG0	I/O	General-purpose I/O pin G0		MII_RXER	
	GPIOG1	I/O	General-purpose I/O pin G1		MII_RXCLK	
	GPIOG2	I/O	General-purpose I/O pin G2		MII_RXDV	
	GPIOG3	I/O	General-purpose I/O pin G3		MII_RXD0	
	GPIOG4	I/O	General-purpose I/O pin G4		MII_RXD1	
	GPIOG5	I/O	General-purpose I/O pin G5		MII_RXD2	
	GPIOG6	I/O	General-purpose I/O pin G6		MII_RXD3	
	GPIOG7	I/O	General-purpose I/O pin G7		MII_MDC	
H	GPIOH0	I/O	General-purpose I/O pin H0		MII_MDIO	

25.3 Registers

25.3.1 Register List

The base address for these registers is 0xFFFF_1000.

Table 25.1 GPIO Register List (Base Address: 0xFFFF_1000)

Address Offset	Register Name	Abbreviation	Default Value	R/W	Data Access Size (Bits)
0x00	GPIOA Data Register	GPIOA_DATA	0x0000	R/W	8 (/16/32) *1
0x04	GPIOA Pin Function Register	GPIOA_FNC	0x0000	R/W	16 (/32) *2
0x08	GPIOB Data Register	GPIOB_DATA	0x0000	R/W	8 (/16/32) *1
0x0C	GPIOB Pin Function Register	GPIOB_FNC	0x0000	R/W	16 (/32) *2
0x10	GPIOC Data Register	GPIOC_DATA	0x0000	R/W	8 (/16/32) *1
0x14	GPIOC Pin Function Register	GPIOC_FNC	0x0000	R/W	16 (/32) *2
0x18	GPIOD Data Register	GPIOD_DATA	0x0000	R/W	8 (/16/32) *1
0x1C	GPIOD Pin Function Register	GPIOD_FNC	0x0000	R/W	16 (/32) *2
0x20	GPIOE Data Register	GPIOE_DATA	0x0000	R/W	8 (/16/32) *1
0x24	GPIOE Pin Function Register	GPIOE_FNC	0x0000	R/W	16 (/32) *2
0x28	GPIOF Data Register	GPIOF_DATA	0x0000	R/W	8 (/16/32) *1
0x2C	GPIOF Pin Function Register	GPIOF_FNC	0x5555	R/W	16 (/32) *2
0x30	GPIOG Data Register	GPIOG_DATA	0x0000	R/W	8 (/16/32) *1
0x34	GPIOG Pin Function Register	GPIOG_FNC	0x5555	R/W	16 (/32) *2
0x38	GPIOH Data Register	GPIOH_DATA	0x0000	R/W	8 (/16/32) *1
0x3C	GPIOH Pin Function Register	GPIOH_FNC	0x0001	R/W	16 (/32) *2
0x40	GPIOA&B IRQ Type Register	GPIOAB_ITYP	0x0000	R/W	16 (/32) *2
0x44	GPIOA&B IRQ Polarity Register	GPIOAB_IPOL	0x0000	R/W	16 (/32) *2
0x48	GPIOA&B IRQ Enable Register	GPIOAB_IEN	0x0000	R/W	16 (/32) *2
0x4C	GPIOA&B IRQ Status & Clear Register	GPIOAB_ISTS	0x0000	R/W	16 (/32) *2

Notes: *1: Registers support 8-, 16-, and 32-bit access.
*2: Registers support 16- and 32-bit access.
*3: The defaults of the GPIOA to GPIOE data registers vary depending on system configuration conditions because the pin selection function is set to the port input as the default.

25.3.2 Detailed Register Descriptions

25.3.2.1 Data and Function Registers

GPIOA Data Register (GPIOA_DATA)
GPIO[0x00] Default = 0x0000_0000 Read/Write

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

This is the GPIOA Data Register. It supports both write and read access.

Reads return the pin input state if the corresponding pin is configured for input and the register bits value otherwise.

Writes specify the pin output level if the pin is configured for output.

Note: The default is set to the value corresponding to the GPIOA pin.

GPIOA Pin Function Register (GPIOA_FNC)

GPIO[0x04] Default = 0x0000_0000 Read/Write

n/a							
31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
GPA7MD [1:0]		GPA6MD [1:0]		GPA5MD [1:0]		GPA4MD [1:0]	
15	14	13	12	11	10	9	8
GPA3MD [1:0]		GPA2MD [1:0]		GPA1MD [1:0]		GPA0MD [1:0]	
7	6	5	4	3	2	1	0

Each 2-bit field specifies the pin function for the corresponding GPIOA pin.

Table 25.2 Port A Pin Functions

GPAxMD1	GPAxMD0	Pin Function
0	0	GPIOAx pin input (default)
0	1	Non-GPIO function #1
1	0	GPIOAx pin output
1	1	Non-GPIO function #2 (only available for pins 3 and 2)

25.3.2.2 GPIOB Registers

GPIOB Data Register (GPIOB_DATA)

GPIO[0x08] Default = 0x0000_0000 Read/Write

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

This is the GPIOB Data Register. It supports both write and read access.

Reads return the pin input state if the corresponding pin is configured for input and the register bits value otherwise.

Writes specify the pin output level if the pin is configured for output.

Note: The default is set to the value corresponding to the GPIOB pin.

GPIOB Pin Function Register (GPIOB_FNC)

GPIO[0x0C] Default = 0x0000_0000 Read/Write

n/a							
31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
GPB7MD [1:0]		GPB6MD [1:0]		GPB5MD [1:0]		GPB4MD [1:0]	
15	14	13	12	11	10	9	8
GPB3MD [1:0]		GPB2MD [1:0]		GPB1MD [1:0]		GPB0MD [1:0]	
7	6	5	4	3	2	1	0

Each 2-bit field specifies the pin function for the corresponding GPIOB pin.

Table 25.3 Port B Pin Functions

GPBxMD1	GPBxMD0	Pin Function
0	0	GPIOBx pin input (default)
0	1	Non-GPIO function #1(not available for pins 0)
1	0	GPIOBx pin output
1	1	Non-GPIO function #2 (only available for pins 7, 6, 3, 2, 1, and 0)

25.3.2.3 GPIOC Registers

GPIOC Data Register (GPIOC_DATA)
GPIO[0x10] Default = 0x0000_0000 Read/Write

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

This is the GPIOC Data Register. It supports both write and read access.

Reads return the pin input state if the corresponding pin is configured for input and the register bits value otherwise.

Writes specify the pin output level if the pin is configured for output.

Note: The default is set to the value corresponding to the GPIOC pin.

GPIOC Pin Function Register (GPIOC_FNC)
GPIO[0x14] Default = 0x0000_0000 Read/Write

n/a							
31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
GPC7MD [1:0]		GPC6MD [1:0]		GPC5MD [1:0]		GPC4MD [1:0]	
15	14	13	12	11	10	9	8
GPC3MD [1:0]		GPC2MD [1:0]		GPC1MD [1:0]		GPC0MD [1:0]	
7	6	5	4	3	2	1	0

Each 2-bit field specifies the pin function for the corresponding GPIOC pin.

Table 25.4 Port C Pin Functions

GPCxMD1	GPCxMD0	Pin Function
0	0	GPIOCx pin input (default)
0	1	Non-GPIO function #1
1	0	GPIOCx pin output
1	1	Reserved

25.3.2.4 GPIOD Registers

GPIOD Data Register (GPIOD_DATA)
GPIO[0x18] Default = 0x0000_0000 Read/Write

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

This is the GPIOD Data Register. It supports both write and read access.

Reads return the pin input state if the corresponding pin is configured for input and the register bits value otherwise.

Writes specify the pin output level if the pin is configured for output.

Note: The default is set to the value corresponding to the GPIOD pin.

GPIOD Pin Function Register (GPIOD_FNC)							
GPIO[0x1C]	Default = 0x0000_0000						Read/Write
n/a							
31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
GPD7MD [1:0]		GPD6MD [1:0]		GPD5MD [1:0]		GPD4MD [1:0]	
15	14	13	12	11	10	9	8
GPD3MD [1:0]		GPD2MD [1:0]		GPD1MD [1:0]		GPD0MD[1:0]	
7	6	5	4	3	2	1	0

Each 2-bit field specifies the pin function for the corresponding GPIOD pin.

Table 25.5 Port D Pin Functions

GPDxMD1	GPDxMD0	Pin Function
0	0	GPIODx pin input (default)
0	1	Non-GPIO function #1
1	0	GPIODx pin output
1	1	Reserved

25.3.2.5 GPIOE Registers

GPIOE Data Register (GPIOE_DATA)							
GPIO[0x20]	Default = 0x0000_0000						Read/Write
n/a							
31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
n/a							
15	14	13	12	11	10	9	8
GPIOE Data [7:0]							
7	6	5	4	3	2	1	0

This is the GPIOE Data Register. It supports both write and read access.

Reads return the pin input state if the corresponding pin is configured for input and the register bits value otherwise.

Writes specify the pin output level if the pin is configured for output.

Note: The default is set to the value corresponding to the GPIOE pin.

GPIOE Pin Function Register (GPIOE_FNC)							
GPIO[0x24]	Default = 0x0000_0000						Read/Write
n/a							
31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
GPE7MD [1:0]		GPE6MD [1:0]		GPE5MD [1:0]		GPE4MD [1:0]	
15	14	13	12	11	10	9	8
GPE3MD [1:0]		GPE2MD [1:0]		GPE1MD [1:0]		GPE0MD [1:0]	
7	6	5	4	3	2	1	0

Each 2-bit field specifies the pin function for the corresponding GPIOE pin.

Table 25.6 Port E Pin Functions

GPExMD1	GPExMD0	Pin Function
0	0	GPIOEx pin input (default)
0	1	Non-GPIO function #1
1	0	GPIOEx pin output
1	1	Non-GPIO function #2 (only available for pins 7, 6, 5, 3, 1, and 0)

25.3.2.6 GPIOF Registers

GPIOF Data Register (GPIOF_DATA)
GPIO[0x28] Default = 0x0000_0000 Read/Write

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

This is the GPIOF Data Register. It supports both write and read access.

Reads return the pin input state if the corresponding pin is configured for input and the register bits value otherwise.

Writes specify the pin output level if the pin is configured for output.

GPIOF Pin Function Register (GPIOF_FNC)
GPIO[0x2C] Default = 0x0000_5555 Read/Write

n/a							
31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
GPF7MD [1:0]		GPF6MD [1:0]		GPF5MD [1:0]		GPF4MD [1:0]	
15	14	13	12	11	10	9	8
GPF3MD [1:0]		GPF2MD [1:0]		GPF1MD [1:0]		GPF0MD [1:0]	
7	6	5	4	3	2	1	0

Each 2-bit field specifies the pin function for the corresponding GPIOF pin.

Table 25.7 Port F Pin Functions

GPFxMD1	GPFxMD0	Pin Function
0	0	GPIOFx pin input
0	1	Non-GPIO function #1 (default)
1	0	GPIOFx pin output
1	1	Reserved

25.3.2.7 GPIOG Registers

GPIOG Data Register (GPIOG_DATA)
GPIO[0x30] Default = 0x0000_0000 Read/Write

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

This is the GPIOG Data Register. It supports both write and read access.

Reads return the pin input state if the corresponding pin is configured for input and the register bits value otherwise.

Writes specify the pin output level if the pin is configured for output.

GPIOG Pin Function Register (GPIOG_FNC)
GPIO[0x34] Default = 0x0000_5555 Read/Write

n/a							
31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
GPG7MD [1:0]		GPG6MD [1:0]		GPG5MD [1:0]		GPG4MD [1:0]	
15	14	13	12	11	10	9	8
GPG3MD [1:0]		GPG2MD [1:0]		GPG1MD [1:0]		GPG0MD [1:0]	
7	6	5	4	3	2	1	0

Each 2-bit field specifies the pin function for the corresponding GPIOG pin.

Table 25.8 Port G Pin Functions

GPGxMD1	GPGxMD0	Pin Function
0	0	GPIOGx pin input
0	1	Non-GPIO function #1 (default)
1	0	GPIOGx pin output
1	1	Reserved

25.3.2.8 GPIOH Registers

GPIOH Data Register (GPIOH_DATA)
GPIO[0x38] Default = 0x0000_0000 Read/Write

n/a							
31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
n/a							
15	14	13	12	11	10	9	8
Reserved (0)							GPIOH Data 0
7	6	5	4	3	2	1	0

This is the GPIOH Data Register. It supports both write and read access.

Reads return the pin input state if the corresponding pin is configured for input and and the register bits value otherwise.

Writes specify the pin output level if the pin is configured for output.

GPIOH Pin Function Register (GPIOH_FNC)							
GPIO[0x3C]	Default = 0x0000_0001						Read/Write
n/a							
31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
Reserved (0)		Reserved (0)		Reserved (0)		Reserved (0)	
15	14	13	12	11	10	9	8
Reserved (0)		Reserved (0)		Reserved (0)		GPH0MD [1:0]	
7	6	5	4	3	2	1	0

Each 2-bit field specifies the pin function for the corresponding GPIOH pin.

Table 25.9 Port H Pin Functions

GPH0MD1	GPH0MD0	Pin Function
0	0	GPIOH0 pin input
0	1	Non-GPIO function #1 (default)
1	0	GPIOH0 pin output
1	1	Reserved

25.3.2.9 GPIOA/GPIOB Interrupt Request Registers

The four registers GPIO[0x40] to GPIO[0x4C] are for IRQ14 interrupt requests based on GPIOA/GPIOB pin input.

GPIOA&B IRQ TYPE

GPIOA&B IRQ Type Register (GPIOAB_ITYP)							
GPIO[0x40]	Default = 0x0000_0000						Read/Write
n/a							
31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
PORTB_IRQ_TYPE [7:0]							
15	14	13	12	11	10	9	8
PORTA_IRQ_TYPE [7:0]							
7	6	5	4	3	2	1	0

Bits 15 to 8: **PORTB_IRQ_TYPE [7:0]**

Bits 7 to 0: **PORTA_IRQ_TYPE [7:0]**

These bits offer a choice of trigger types for the interrupts.
0: Level trigger
1: Edge trigger

GPIOA&B IRQ Polarity

GPIOA&B IRQ Polarity Register (GPIOAB_IPOL)							
GPIO[0x44]	Default = 0x0000_0000						Read/Write
n/a							
31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
PORTB_IRQ_POL [7:0]							
15	14	13	12	11	10	9	8
PORTA_IRQ_POL [7:0]							
7	6	5	4	3	2	1	0

Bits 15 to 8: **PORTB_IRQ_POL [7:0]**

Bits 7 to 0: **PORTA_IRQ_POL [7:0]**

These bits offer a choice of polarities for the interrupts.

0: High (level trigger) or
rising (edge trigger)

1: Low (level trigger) or
falling (edge trigger)

GPIOA&B IRQ ENABLE

GPIOA&B IRQ Enable Register (GPIOAB_IEN)							
GPIO[0x48]	Default = 0x0000_0000						Read/Write
n/a							
31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
PORTB_IEN [7:0]							
15	14	13	12	11	10	9	8
PORTA_IEN [7:0]							
7	6	5	4	3	2	1	0

Bits 15 to 8: **PORTB_IEN [7:0]**

Bits 7 to 0: **PORTA_IEN [7:0]**

Interrupt request enable

0: Disable

1: Enable

“1” in a bit enables interrupt requests from the corresponding GPIOA/GPIOB pin.

GPIOA&B IRQ STATUS & Clear

GPIOA&B IRQ Status & Clear Register (GPIOAB_ISTS) GPIO[0x4C] Default = 0x0000_0000							Read/Write
n/a							
31	30	29	28	27	26	25	24
n/a							
23	22	21	20	19	18	17	16
PORTB_IRQ [7:0]							
15	14	13	12	11	10	9	8
PORTA_IRQ [7:0]							
7	6	5	4	3	2	1	0

Bits 15 to 8: **PORTB_IRQ [7:0]**

Bits 7 to 0: **PORTA_IRQ [7:0]**

Reads return the interrupt request status for the corresponding GPIOA/GPIOB pin.
0: There is no interrupt request
1: There is an interrupt request
Writing “1” to a bits clears the interrupt request for the corresponding GPIOA/GPIOB pin.
0: (ignored)
1: Clear

Note: The hardware ORs the interrupt requests from the GPIOA/GPIOB pins to create a single interrupt request (IRQ14) to the interrupt controller. The IRQ handler must therefore start by determining which GPIO pin triggered the interrupt request.

25.4 GPIOA/GPIOB Interrupt Request Logic

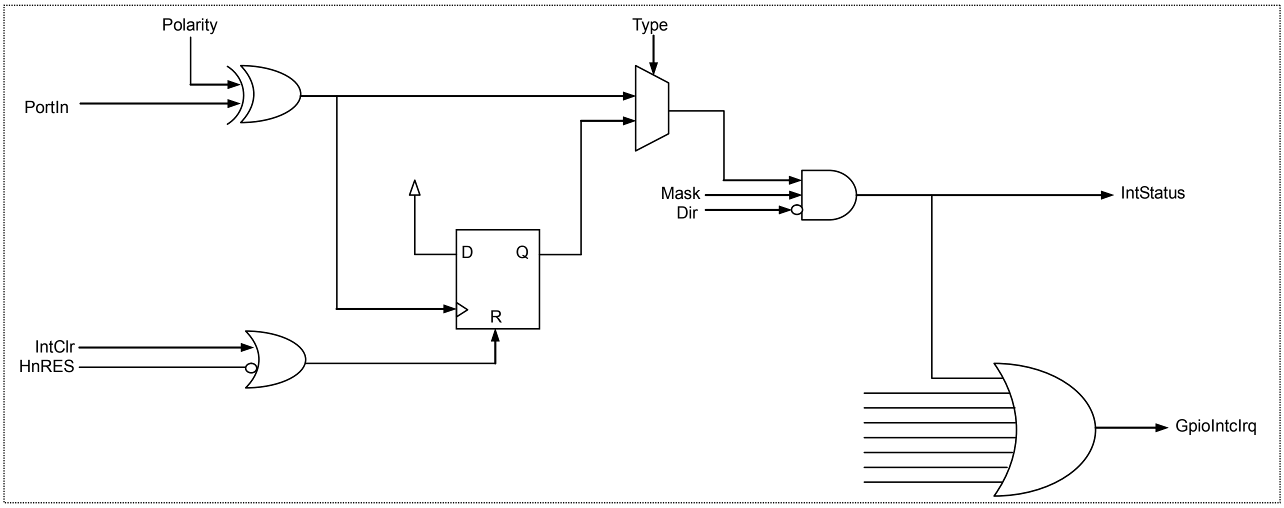


Fig.25.1 GPIOA/GPIOB Interrupt Request Logic

Note: Always clear all Interrupt requests before changing Interrupt request Trigger types or polarities. Otherwise, there may be spurious Interrupt requests.

26. ABSOLUTE MAXIMUM RATINGS

26.1 Absolute Maximum Ratings

(VSS = 0 V)

Item	Symbol	Rating	Unit
Power supply voltage	HVDD*	–0.3 to 4.0	V
	LVDD*	–0.3 to 2.5	V
Input voltage	HVI	–0.3 to HVDD+0.5	V
	LVI	–0.3 to LVDD+0.5	V
Output voltage	HVO	–0.3 to HVDD+0.5	V
	LVO	–0.3 to LVDD+0.5	V
Output current/pin	I_{OUT}	± 10	mA
Storage temperature	T_{stg}	–65 to 150	°C

*: HVDD ≥ LVDD

26.2 Recommend Operation Conditions (2 Power Supplies, 3.3 V I/O Buffers)

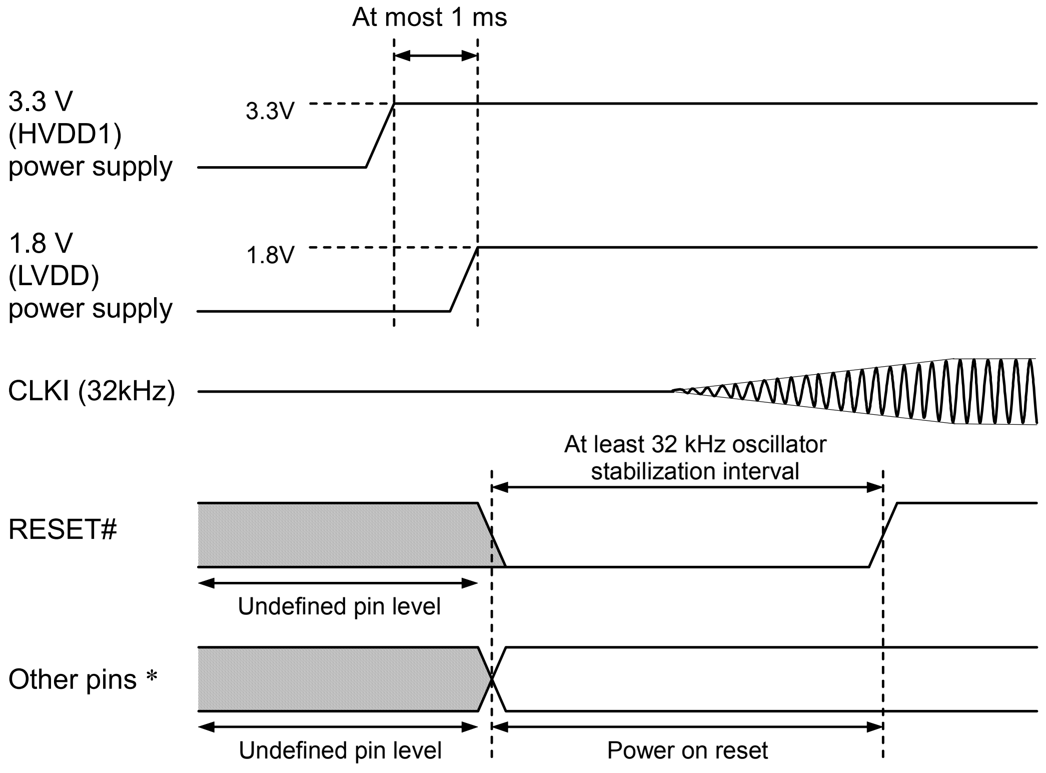
(VSS = PLLVSS = 0 V)

Item		Symbol	Min.	Typ.	Max.	Unit
Power supply voltage (high voltage)	I/O cell power supply	HVDD1	3.00	3.30	3.60	V
	Camera interface power supply	HVDD2	2.40	3.00	3.60	V
Power supply voltage (low voltage)	Core (internal) power supply	LVDD	1.65	1.80	1.95	V
	PLL analog power supply	PLLVDD	1.65	1.80	1.95	V
Input voltage	I/O cell power supply	HV_{I1}	VSS	—	HVDD1	V
	Camera interface power supply	HV_{I2}	VSS	—	HVDD2	V
	Core (internal) power supply	LV_I	VSS	—	LVDD	V
	PLL analog power supply	$PLLV_I$	PLLVSS	—	PLLVDD	V
Ambient temperature		Ta	–40	25	85*	°C
Input rising time (normal input)		t_{ri}	—	—	50	ns
Input falling time (normal input)		t_{fa}	—	—	50	ns
Input rising time (Schmitt trigger input)		t_{ri}	—	—	5	ms
Input falling time (Schmitt trigger input)		t_{fa}	—	—	5	ms

* This temperature assumes a recommended ambient temperature (Tj) range of –40 to 125°C.

26.3 Power Supply Timing

Always apply the 3.3 V (HVDD1) power supply and then, within 1 ms, the 1.8 V (LVDD) power supply. We recommend keeping this interval as short as possible. Even after the two have stabilized, keep RESET# at Low level while the 32 kHz oscillator starts up—at least 100 ms, for example.



*** Pins other than the power supplies (and grounds), RESET#, clock signals, and analog pins.**

Fig.26.1 Power Supply Timing

26.4 Shut Down Timing

Shut down in reverse order, removing first the 1.8 V (LVDD) power supply and then, within 1 ms, the 3.3 V (HVDD1) power supply. We recommend keeping this interval as short as possible.

Cutting off the 1.8 V power supply alone leaves the pins in undefined states. Designs must take appropriate measures to prevent these from interfering with proper system operation.

27. ELECTRICAL CHARACTERISTICS

27.1 DC Characteristics

Table 27.1 DC Characteristics

(HVDD = 3.3 V ± 0.3 V, VSS = 0 V, Ta= -40 to 85°C)

Item	Symbol	Conditions		Min.	Typ.	Max.	Unit
Input leakage current	I_{LI}	—		-5	—	5	μA
Off state leakage current	I_{OZ}	—		-5	—	5	μA
High level output voltage*	V_{OH}	IOH = –4mA HVDD=Min.		HVDD -0.4	—	—	V
Low level output voltage*	V_{OL}	IOL = 4mA HVDD=Min.		—	—	0.4	V
High level input voltage	V_{IH1}	LVCMOS level, HVDD=Max.		2.2	—		V
Low level input voltage	V_{IL1}	LVCMOS level, HVDD=Min.		—	—	0.8	V
High level input voltage	V_{T1+}	LVCMOS Schmitt trigger input		1.4	—	2.7	V
Low level input voltage	V_{T1-}	LVCMOS Schmitt trigger input		0.6	—	1.8	V
Hysteresis voltage	V_{H1}	LVCMOS Schmitt trigger input		0.3	—	—	V
High level input voltage	V_{IH2}	LVTTL level, HVDD=Max.		2.0	—	—	V
Low level input voltage	V_{IL2}	LVTTL level, HVDD=Min.		—	—	0.8	V
Pull-up resistance	PPU	VI=0V		25	50	120	kΩ
Pull-down resistance	PPD	VI=HVDD	Others*2	25	50	120	kΩ
			MD[15:0] pin	50	100	240	
Input pin capacitance	C_{I}	f=1 MHz, HVDD = 0 V		—	—	8	pF
Output pin capacitance	C_{O}	f=1 MHz, HVDD = 0 V		—	—	8	pF
I/O pin capacitance	C_{IO}	f=1 MHz, HVDD = 0 V		—	—	8	pF
Current consumption (LVDD)	I_{LOW}	LOW SPEED (32 kHz) mode		—	100	—	μA
	I_{LHALT}	LOW SPEED HALT mode		—	7	—	μA
	I_{FO}	HIGH SPEED mode*3		—	70	—	mA
	I_{HALT}	HIGH SPEED HALT mode*4		—	1	—	mA
	I_{DDQ}	Clock signals and PLL off	Ta = 25°C	—	5	—	μA
			Ta = 85°C	—	15	300	μA
Current consumption (PLLVDD)	I_{DDPLL}	PLL frequency = 50 MHz		—	1	—	mA

Notes *: All output and bidirectional pins.
*2: All pins with pull-down resistances—except the Data bus (MD[15:0]) pins
*3: For Image transfers at VGA resolution at 7.5 fps
*4: With all clock signals stopped—including the one from MII PHY

27.2 AC Characteristics

27.2.1 AC Characteristics Measuring Conditions

HVDD1 = 3.3 V ± 0.3 V
HVDD2 = 3.0 V ± 0.6 V
LVDD = 1.8 V ± 0.15 V
T_A = -40°C to 85°C
C_L = 50 pF (unless otherwise mentioned)

27.2.2 AC Characteristics Timing

27.2.2.1 Clock Timing

Table 27.2 Clock (CLKI) Timing

Item	Symbol	Min.	Typ.	Max.	Unit	Notes
CLKI frequency	f_{OSC}	—	32.768	—	kHz	—
CLKI input period	t_{OSC}	—	1/ f_{OSC}	—	s	—
CLKI High level pulse width	t_{CLKIH}	5	—	—	μs	—
CLKI Low level pulse width	t_{CLKIL}	5	—	—	μs	—
CLKI rising time (10% to 90%)	t_{CLKIR}	—	—	12	μs	—
CLKI falling time (90% to 10%)	t_{CLKIF}	—	—	12	μs	—
System clock frequency	f_{sys}	—	—	50	MHz	—
System clock period	T_S	1/f_{sys}	—	—	ns	—

27.2.2.2 CPU Control Signal Timing

Table 27.3 CPU Control Signal Timing

Item	Symbol	Min.	Typ.	Max.	Unit	Notes
RESET# pulse width	t_{RESW}	10	—	—	T_{CLKI}	*1
IRQ/FIQ pulse width	t_{IRQW}	10	—	—	T_S	*2
Clock restart time	t_{WAK}	—	—	4	T_S	*2
PLL stabilization interval	t_{PLLST}	—	—	100	ms	—

*1: The unit is T_{CLKI}, the 32 kHz clock period. The input must have an amplitude above the threshold voltage.

*2: T_S = system clock period

27.2.2.3 Camera Interface (CAM) Timing

Table 27.4 Camera Interface (CAM) Timing

Item	Symbol	Min.	Typ.	Max.	Unit	Notes
CMVREF rising edge to CMHREF rising edge	t_{CAM1}	0	—	—	Tc	*3
Horizontal blanking interval	t_{CAM2}	4	—	—	Tc	*3
CMHREF falling edge to CMVREF falling edge	t_{CAM3}	0	—	—	Tc	*3
Vertical blanking interval	t_{CAM4}	1	—	—	Line	—
Camera input clock period	t_{CAM5}	2 (4)	—	—	Ts	*2
Camera input clock Low level pulse width	t_{CAM6}	1 (2)	—	—	Ts	*2
Camera input clock High level pulse width	t_{CAM7}	1 (2)	—	—	Ts	*2
Data setup time	t_{CAM8}	10	—	—	ns	—
Data hold time	t_{CAM9}	10	—	—	ns	—
CMVREF/CMHREF setup time	t_{CAM10}	10	—	—	ns	—
CMVREF/CMHREF hold time	t_{CAM11}	10	—	—	ns	—

*2: Ts = system clock period, the values in the fast sampling mode (the values in the parentheses are in the normal sampling mode).

*3: Tc = camera interface input clock period

27.2.2.4 Media Independent Interface Ethernet PHY (MII PHY) Timing

Table 27.5 MII Timing

Item	Symbol	Min.	Typ.	Max.	Unit	Notes
MII output data delay	t_{TXD}	7	—	20	ns	—
MII input data setup time	t_{RXS}	10	—	—	ns	—
MII input data hold time	t_{RXH}	10	—	—	ns	—
MDIO output delay	t_{MOD}	—	1	—	Ts	*2
MDIO data setup time	t_{MIS}	10	—	—	ns	—
MDIO data hold time	t_{MIH}	0	—	—	ns	—
MDC period	t_{MDC}	—	64	—	Ts	*2

*2: Ts = system clock period

27.2.2.5 Memory Controller (MEMC) Timing

- Static Memory Controller Timing

Table 27.6 Static Memory Read Timing

Item	Symbol	Min.	Typ.	Max.	Unit	Notes
Address delay	t_{DADD}			10	ns	—
MCS0# active delay	t_{MCS0AD}			10	ns	—
MCS0# inactive delay	t_{MCS0ID}			10	ns	
MOE# active delay	t_{MOEAD}			10	ns	
MOE# inactive delay	t_{MOEID}			10	ns	
Read data setup time	t_{RDS}	10			ns	
Read data hold time	t_{RDH}	0			ns	

Table 27.7 Static Memory Write Timing

Item	Symbol	Min.	Typ.	Max.	Unit	Notes
Address delay	t_{DADD}			10	ns	—
MCS0# active delay	t_{MCS0AD}			10	ns	—
MCS0# inactive delay	t_{MCS0ID}			10	ns	
MWE0# active delay	t_{MWE0AD}			10	ns	
MWE0# inactive delay	t_{MWE0ID}			10	ns	
MDQM delay	t_{MDQD}			10	ns	
Write data delay	t_{WDD}	1		10	ns	

- SDRAM Controller Timing

Table 27.8 SDRAM Controller Timing

Item	Symbol	Min.	Typ.	Max.	Unit	Notes
CKE delay	t_{CKED}	1	—	10	ns	—
MCS2# delay	t_{CSD}	1	—	10	ns	—
MRAS# delay	t_{RASD}	1	—	10	ns	—
MCAS# delay	t_{CASD}	1	—	10	ns	—
MWE1# delay	t_{WED}	1	—	10	ns	—
MDQML/MDQMH delay	t_{DQMD}	1	—	10	ns	—
Address delay	t_{ADD}	1	—	10	ns	—
Write data delay	t_{WDD}	1	—	10	ns	—
Read data setup time	t_{RDS}	10	—	—	ns	—
Read data hold time	t_{RDH}	0	—	—	ns	—

27.2.2.6 I^{2}C Single Master Core Module (I^{2}C) Timing

Table 27.9 I^{2}C Timing

Item	Symbol	Min.	Typ.	Max.	Unit	Notes
SCL period	$t_{C(SCL)}$	8	—	30720	T_S	*2
SCL High level pulse width	$t_{WH(SCL)}$	—	1/2	—	$T_{C(SCL)}$	*4
SCL Low level pulse width	$t_{WL(SCL)}$	—	1/2	—	$T_{C(SCL)}$	*4
SDA output delay	$t_{D(OSDA)}$	—	1/4	—	$T_{C(SCL)}$	*4
SDA input setup time	$t_{SU(ISDA)}$	0	—	—	ns	*5
SDA input hold time	$t_{HD(ISDA)}$	0	—	—	ns	*5
SDA sample time	$t_{SMP(SDA)}$	—	1/4	—	$T_{C(SCL)}$	*4
Start condition start time	$t_{S(ST)}$	1/4	—	—	$T_{C(SCL)}$	*4
Start condition complete time	$t_{E(ST)}$	1/2	—	—	$T_{C(SCL)}$	*4
Stop condition start time	$t_{S(SP)}$	1/4	—	—	$T_{C(SCL)}$	*4
Stop condition complete time	$t_{E(SP)}$	1/2	—	—	$T_{C(SCL)}$	*4

*2: T_S = system clock period
*4: $T_{C(SCL)}$ = I^{2}C clock (SCL) period
*5: See SDA sample time ($T_{smp(SDA)}$).

27.2.2.7 I^{2}S Timing

Table 27.10 I^{2}S Timing

Item	Symbol	Min.	Typ.	Max.	Unit	Notes
SCK period	t_{SCKCT}	2	—	512	T_S	*1
SCK High level pulse width	t_{SCKWH}	1	—	—	T_S	*1
SCK Low level pulse width	t_{SCKWL}	1	—	—	T_S	*1
SCK duty ratio	t_{SCKDT}	—	50	—	%	*2
WS period	t_{WSCT}	32	—	256	t_{SCKCT}	*3
WS output delay	t_{WSOD}	-1	—	1	T_S	*1
WS input setup time	t_{WSISU}	1	—	—	T_S	*1
WS input hold time	t_{WSIHD}	1	—	—	T_S	*1
SD output delay	t_{SDOD}	-1	—	1	T_S	*1
SD input setup time	t_{SDISU}	1	—	—	T_S	*1
SD input hold time	t_{SDIHD}	1	—	—	T_S	*1

*1: T_S = system clock period
*2: $t_{SCKDT} = t_{SCKWH} \div (t_{SCKWH} + t_{SCKWL})$
*3: t_{SCKCT} = SCK period

27.2.2.8 Serial Peripheral Interface (SPI) Timing

Table 27.11 SPI Timing

Item	Symbol	Min.	Typ.	Max.	Unit	Notes
SCLK period	$t_{C(SCLK)}$	4	—	512	Ts	*2
SCLK pulse width (leading half)	$t_{WH1(SCLK)}$	—	1/2	—	$T_{C(SCLK)}$	*6
SCLK pulse width (trailing half)	$t_{WH2(SCLK)}$	—	1/2	—	$T_{C(SCLK)}$	*6
SS output start time (automatic control)	$t_{S(OSS)}$	3	—	—	Ts	*2
SS output complete time (automatic control)	$t_{E(OSS)}$	1	—	—	Ts	*2
SS input setup time	$t_{SU(ISS)}$	3	—	—	Ts	*2
SS input hold time	$t_{HD(ISS)}$	1	—	—	Ts	*2
MISO input setup time	$t_{SU(MI)}$	30	—	—	ns	
MISO input hold time	$t_{HD(MI)}$	0	—	—	ns	
MISO output delay	$t_{D(SO)}$	—	—	30	ns	
MOSI input setup time	$t_{SU(SI)}$	10	—	—	ns	
MOSI input hold time	$t_{HD(SI)}$	10	—	—	ns	
MOSI output delay	$t_{D(MO)}$	—	—	0	ns	

*2: Ts = system clock period

*6: $T_{C(SCLK)}$ = clock SPI (SCLK) period = (4×2^{MCBR}) Ts

27.2.2.9 Compact Flash Interface (CF) Timing

■ CF Card Attribute Memory Timing

Table 27.12 CF Card Attribute Memory Read Timing

Item	Symbol	Min.	Typ.	Max.	Unit	Notes
Read cycle time	t_{ATRC}	—	20	—	Ts	*2
Address setup time	t_{ADSAR}	—	4	—	Ts	*2
Address hold time (from MOE# inactive)	t_{ADHMOE}	—	2	—	Ts	*2
CE valid time before read	t_{CEVBR}	—	3	—	Ts	*2
CE valid time after read	t_{CEVAR}	—	2	—	Ts	*2
MOE# active time	t_{MOEW}	—	14	—	Ts	*2
Read data setup time	t_{RDS}	1Ts+13	—	—	ns	
Read data hold time	t_{RDH}	0	—	—	ns	

*2: Ts = system clock period

Table 27.13 CF Card Attribute Memory Write Timing

Item	Symbol	Min.	Typ.	Max.	Unit	Notes
Write cycle time	t_{ATWC}	—	16	—	Ts	*2
Address setup time	t_{ADSAW}	—	3	—	Ts	*2
MWE0# active time	t_{MWE0W}	—	9	—	Ts	*2
Write recovery time	t_{WREC}	—	2	—	Ts	*2
Write data valid time 1	t_{WDV1}	—	11	—	Ts	*2
Write data valid time 2	t_{WDV2}	—	2	—	Ts	*2

*2: Ts = system clock period

■ CF Common Memory Timing

Table 27.14 CF Common Memory Read Timing

Item	Symbol	Min.	Typ.	Max.	Unit	Notes
Read cycle time	t_{CMRC}	—	17	—	Ts	*2
Address setup time	t_{CRADS}	—	4	—	Ts	*2
Address hold time (from MOE# inactive)	t_{ADHMOE}	—	2	—	Ts	*2
CE valid time before read	t_{CEVBR}	—	3	—	Ts	*2
CE valid time after read	t_{CEVAR}	—	2	—	Ts	*2
Wait active tolerance time after read	t_{WTATAR}	—	—	6	Ts	*2
Data setup time after wait release	t_{DSAWT}	—	—	0	Ts	*2
Wait active time	t_{WTW}			3000	ns	
Read data setup time	t_{RDS}	1Ts+13	—	—	ns	
Read data hold time	t_{RDH}	0	—	—	ns	

*2: Ts = system clock period

Table 27.15 CF Common Memory Write Timing

Item	Symbol	Min.	Typ.	Max.	Unit	Notes
Write cycle time	t_{CMWC}	—	17	—	Ts	*2
Address setup time	t_{ADS}	—	4	—	Ts	*2
Address hold time	t_{ADH}	—	4	—	Ts	*2
CE valid time before write	t_{CEVBW}	—	3	—	Ts	*2
CE valid time after write	t_{CEVAW}	—	2	—	Ts	*2
MWE0# active time	t_{MWE0W}	—	9	—	Ts	*2
Data valid time before write	t_{DVBW}	—	11	—	Ts	*2
Data valid time after write	t_{DVAW}	—	2	—	Ts	*2
Write recovery time	t_{WREC}	—	2	—	Ts	*2
Wait active tolerance time after write	t_{WTATAW}	—	—	6	Ts	*2
Write active time after wait release	t_{WWAWT}	—	—	3	Ts	*2
Wait active time	t_{WTW}	—	—	3000	ns	

*2: Ts = system clock period

■ CF I/O Space or IDE Timing

Table 27.16 CF I/O Space or IDE Read Timing

Item	Symbol	Min.	Typ.	Max.	Unit	Notes
Read cycle time	t_{IORC}	—	20	—	Ts	*2
IORD# active time	t_{IORW}	—	10	—	Ts	*2
Address setup time	t_{ADSIO}	—	6	—	Ts	*2
Address hold time	t_{ADHIO}	—	4	—	Ts	*2
CE valid time before I/O read	$t_{CEVBIOR}$	—	5	—	Ts	*2
CE valid time after I/O read	$t_{CEVAIOR}$	—	3	—	Ts	*2
REG valid time before I/O read	$t_{REGVBIOR}$	—	6	—	Ts	*2
REG valid time after I/O read	$t_{REGVAIOR}$	—	4	—	Ts	*2
Wait tolerance time after I/O read active	$t_{WTATIOR}$	—	—	6	Ts	*2
Data delay tolerance time after wait release	t_{DATAWT}	—	—	0	Ts	*2
Wait active time	t_{WTW}	—	—	3000	ns	
Read data setup time	t_{RDS}	1Ts+13	—		ns	
Read data hold time	t_{RDH}	0			ns	

*2: Ts = system clock period

Table 27.17 CF I/O Space or IDE Write Timing

Item	Symbol	Min.	Typ.	Max.	Unit	Notes
Write cycle time	t_{IOWC}	—	20	—	Ts	*2
IOWR# active time	t_{IOWW}	—	10	—	Ts	*2
Address setup time	t_{ADSIO}	—	6	—	Ts	*2
Address hold time	t_{ADHIO}	—	4	—	Ts	*2
CE valid time before I/O write	$t_{CEVBIOW}$	—	5	—	Ts	*2
CE valid time after I/O write	$t_{CEVAIOW}$	—	3	—	Ts	*2
REG valid time before I/O write	$t_{REGVBIOW}$	—	6	—	Ts	*2
REG valid time after I/O write	$t_{REGVAIOW}$	—	4	—	Ts	*2
Data valid time before I/O write	t_{DVBIOW}	—	14	—	Ts	*2
Data valid time after I/O write	t_{DVAIOW}	—	3	—	Ts	*2
Wait tolerance time after I/O write	$t_{WTATIOW}$	—	—	6	Ts	*2
I/O write inactive time after wait release	t_{WITAWT}	—	—	2	Ts	*2
Wait active time	t_{WTW}	—	—	3000	ns	

*2: Ts = system clock period

27.2.3 Timing Charts

27.2.3.1 Clock Timing

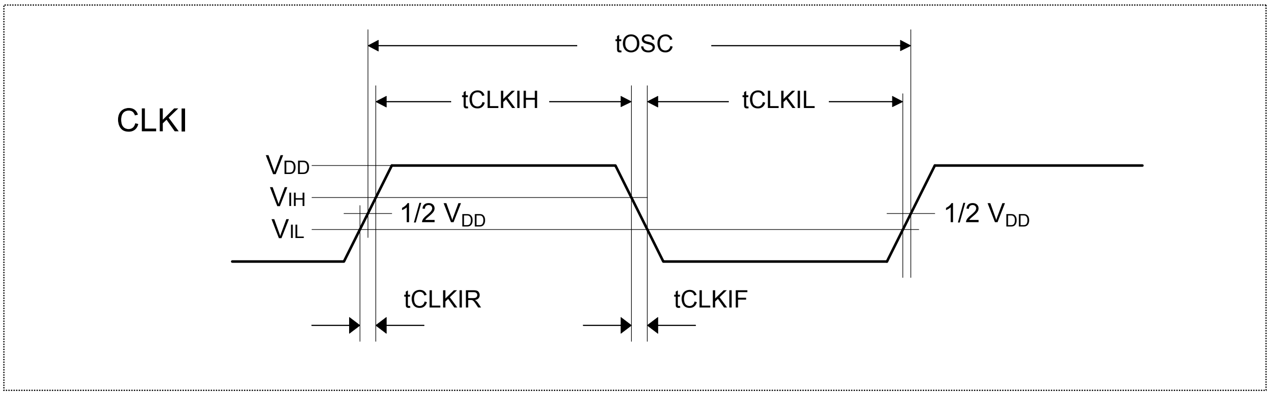


Fig.27.1 Clock Timing

27.2.3.2 CPU Control Signal Timing

■ RESET# Timing

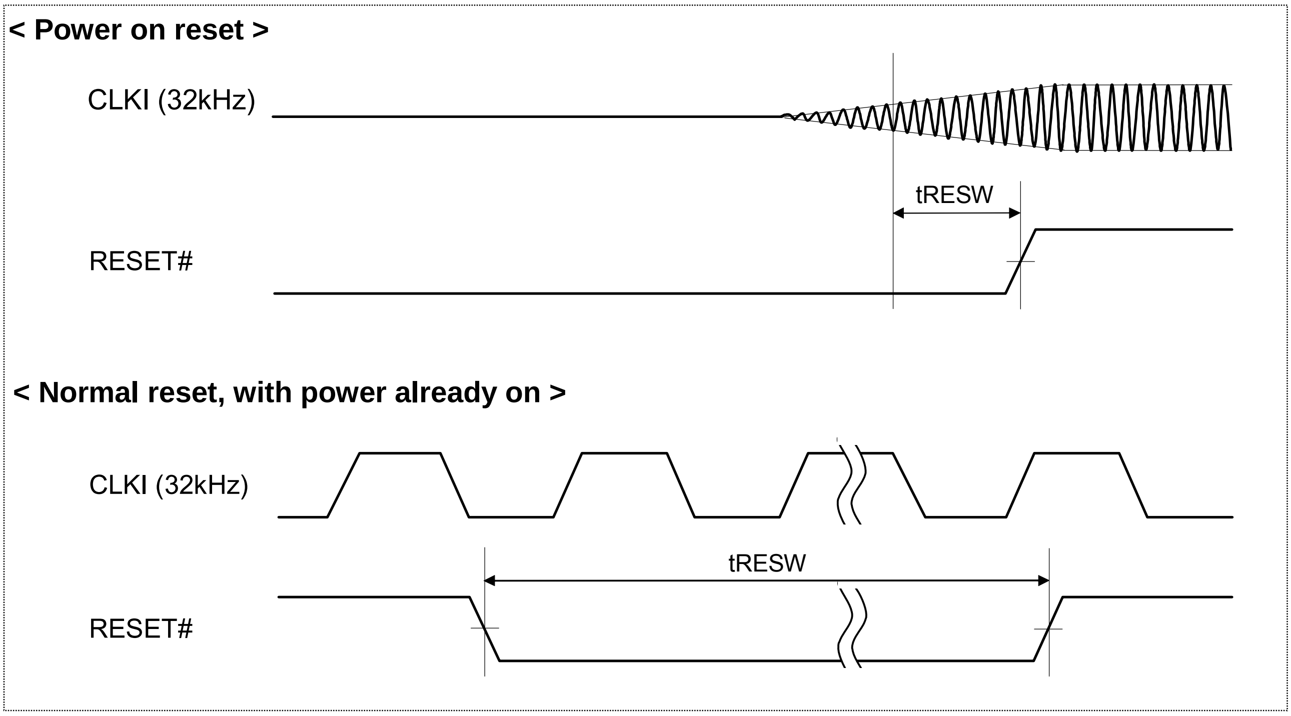


Fig.27.2 RESET# Timing

■ Interrupt Request Signal Timing

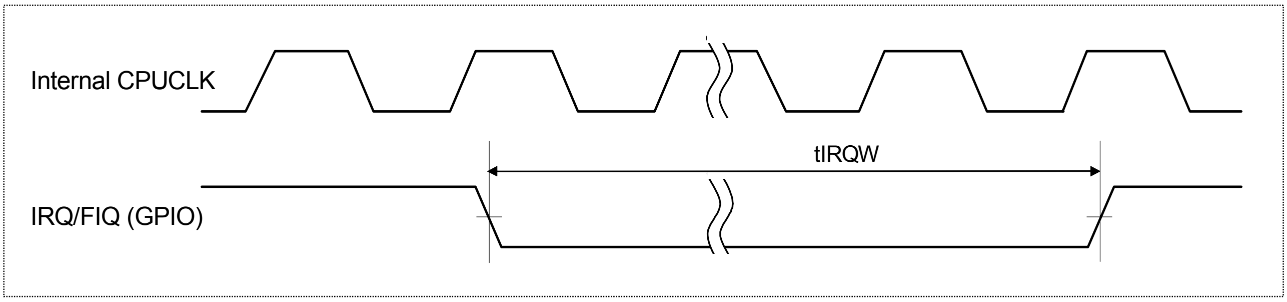
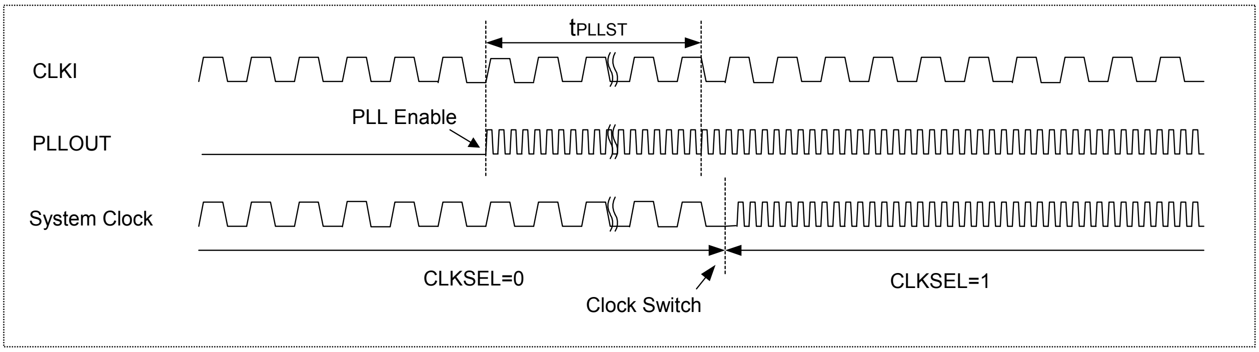


Fig.27.3 Interrupt Request Signal Timing

■ PLL Timing

(1) Clock Switching 1 (PLL Enable)



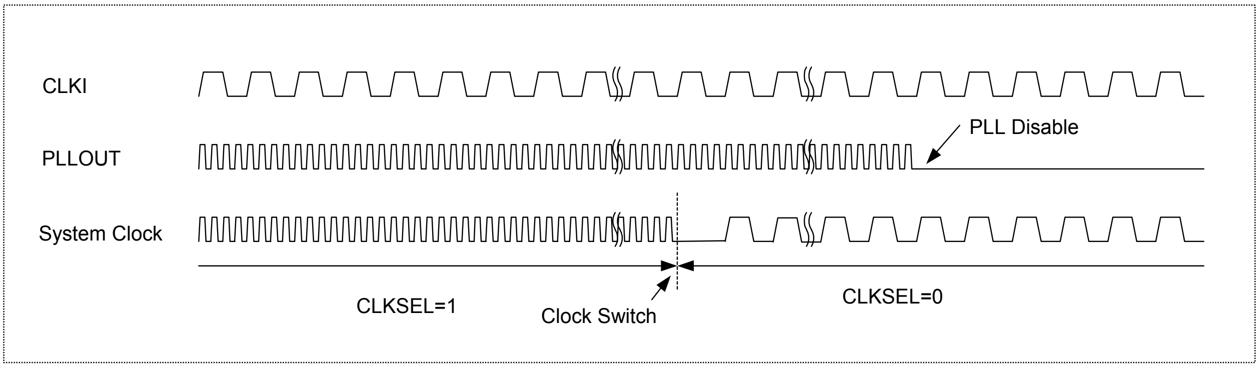
System clocks = CPUCLK, HCLK1, HCLK2, and PCLK

Fig.27.4 Clock Switching 1 (PLL Enable)

This shows the timing for changing the internal system clock frequency from 32 kHz to PLL output.

Set system controller PLL Settings Register 2 (SYS[0x0C]) bit 0 (PLLEN) to “1” to enable the PLL, wait the PLL stabilization interval (tPLLST). and set Clock Select Register (SYS[0x18]) bit 0 (CLKSEL) to “1” to switch the system clocks to PLL output.

(2) Clock Switching 2 (PLL Disable)



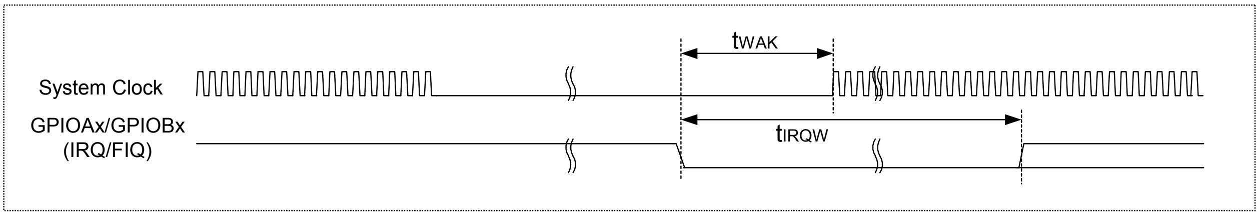
System clocks = CPUCLK, HCLK1, HCLK2, and PCLK

Fig.27.5 Clock Switching 2 (PLL Disable)

This shows the timing for changing the internal system clock frequency from PLL output to 32 kHz.

Set Clock Select Register (SYS[0x18]) bit 0 (CLKSEL) to “0” to switch the system clock to CLKI (32 kHz) and then set system controller PLL Settings Register 2 (SYS[0x0C]) bit 0 (PLLEN) to “0” to disable the PLL.

(3) Restarting Clock with Interrupt Request



System clocks = CPUCLK, HCLK1, HCLK2, and PCLK

Fig.27 6 Clock Restart Timing

This shows the timing for shifting, triggered by an interrupt request, from HIGH SPEED HALT mode to HIGH SPEED mode. The system clock signal restarts after a clock restart interval (tWAK). The interrupt request pulse width (tIRQW) must be sufficiently longer than this interval.

27.2.3.3 Camera Interface Timing

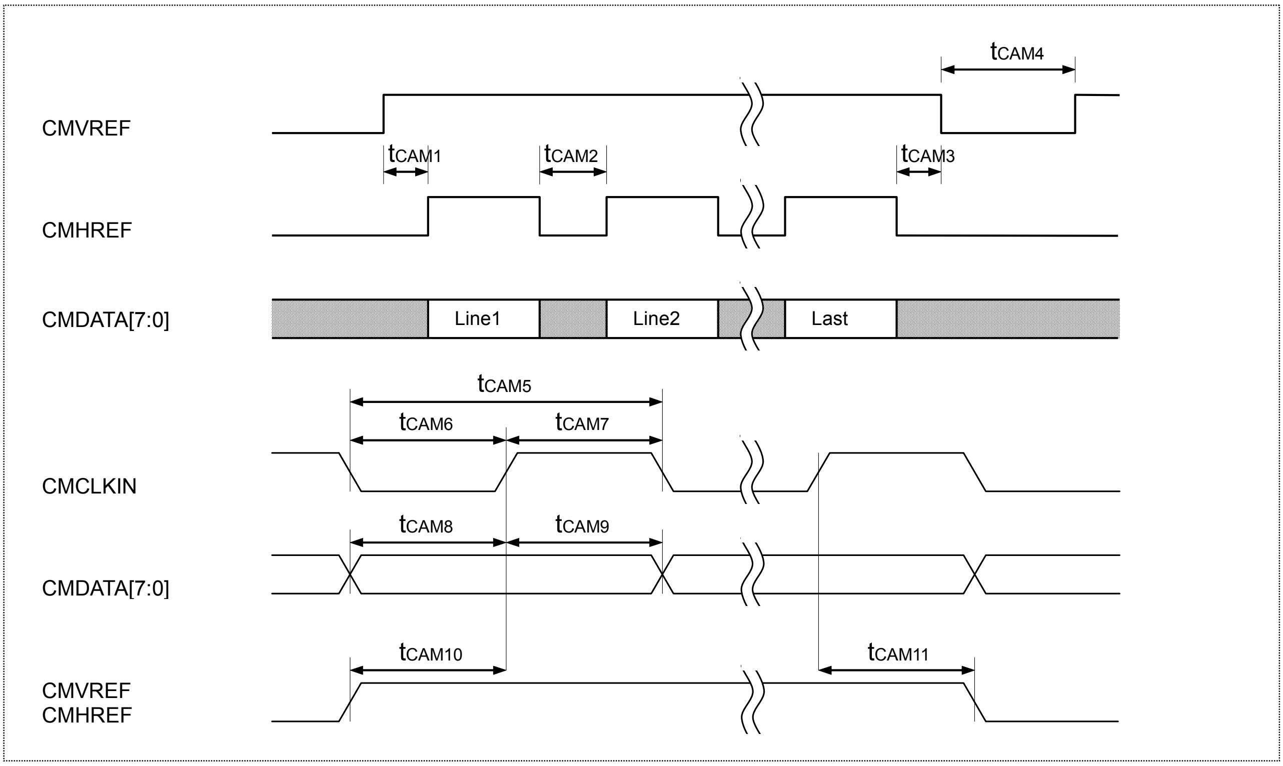


Fig.27.7 Camera Interface Timing

A register setting specifies the valid CMCLKIN edge. The above Figure gives the data capture timing for CMCLKIN rising edges.

27.2.3.4 Media Independent Interface Ethernet PHY (MII PHY) Timing

■ MII Transmit

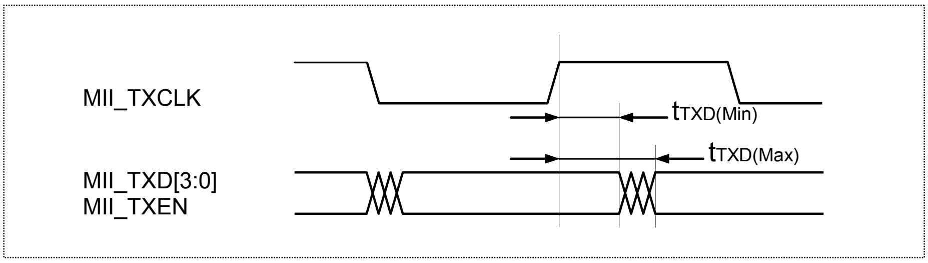


Fig.27.8 MII Transmit Timing

■ MII Receive

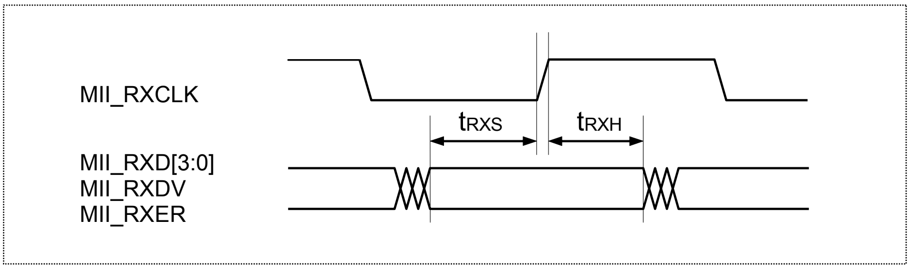


Fig.27.9 MII Receive Timing

■ MDIO Output

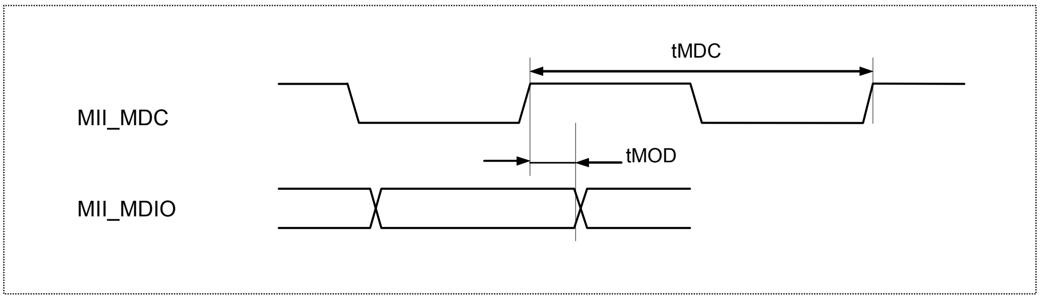


Fig.27.10 MDIO Output Timing

■ MDIO Input

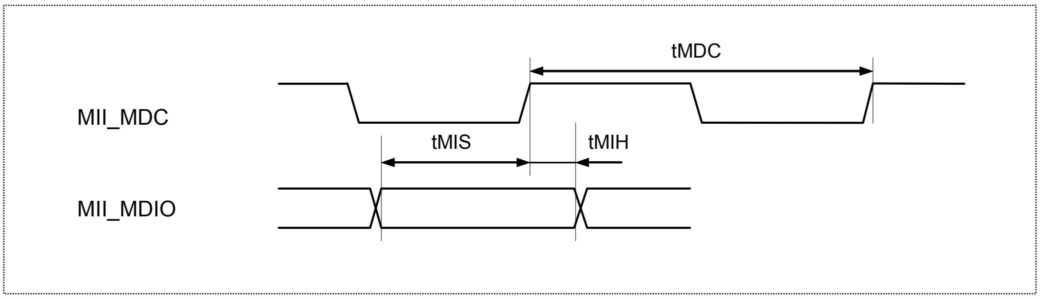
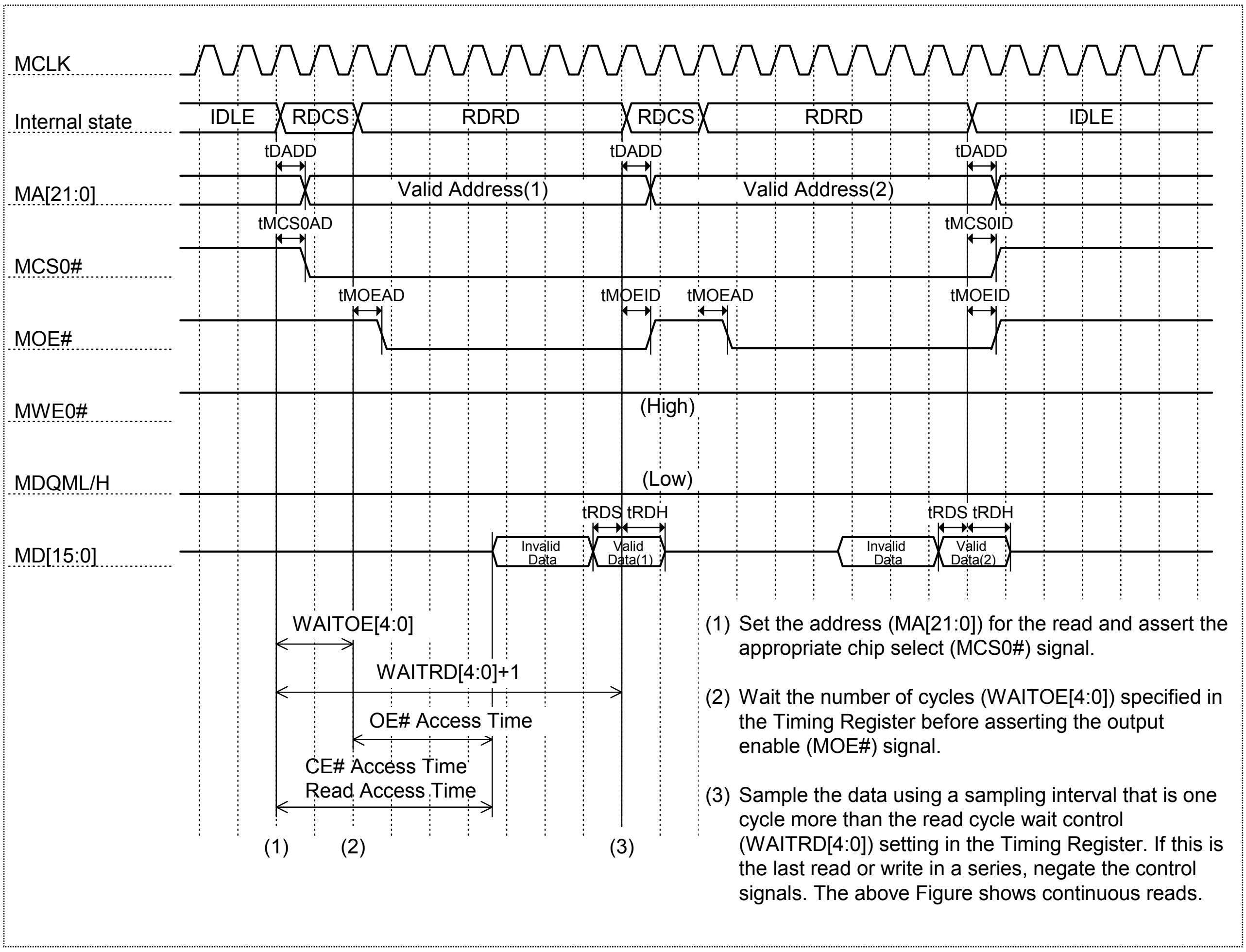


Fig.27.11 MDIO Input Timing

27.2.3.5 Memory Interface Controller

27.2.3.5.1 Static Memory Controller Timing (Flash EEPROM, SRAM, etc.)

■ Static Memory Read Timing



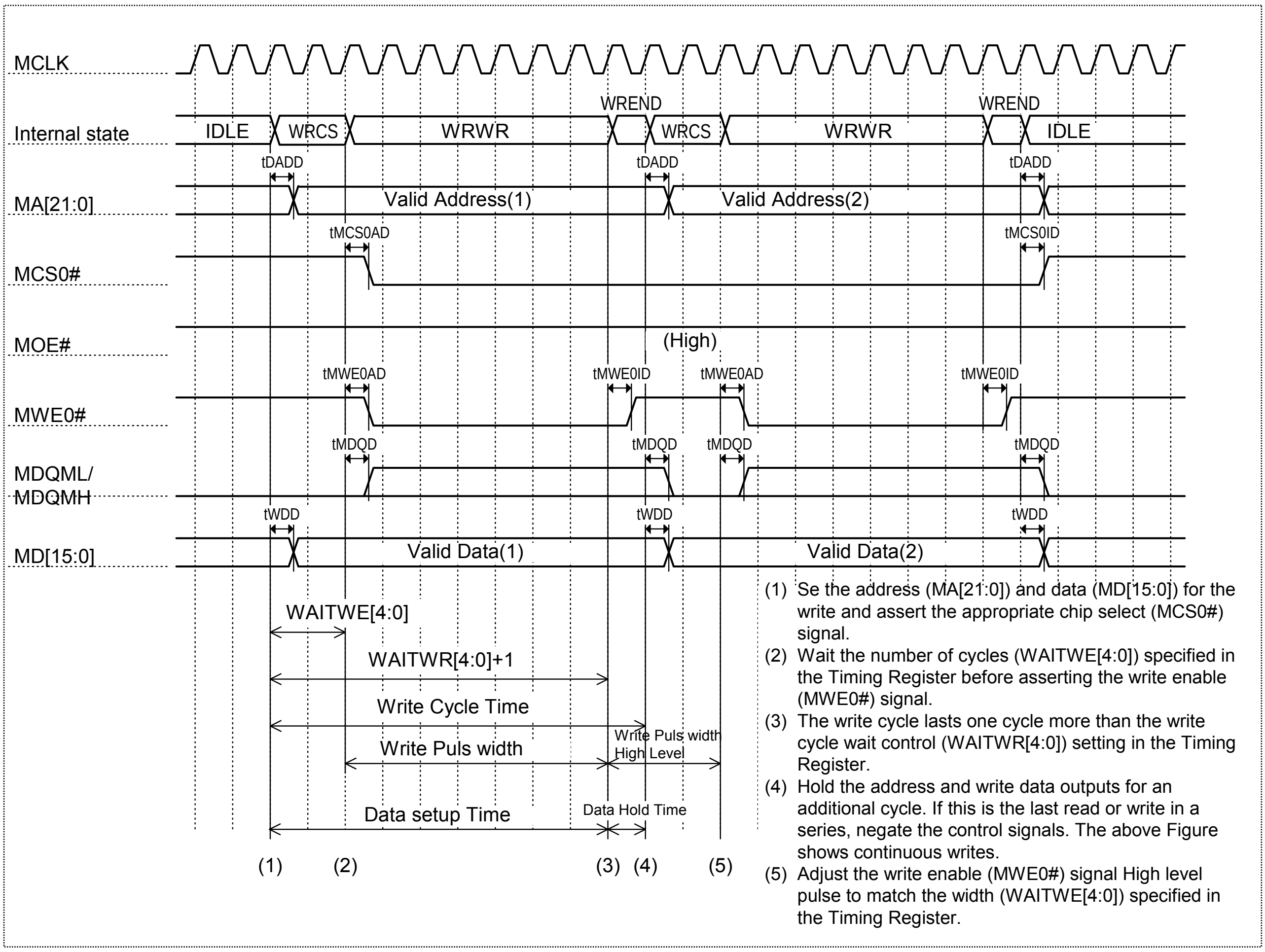
(1) Set the address (MA[21:0]) for the read and assert the appropriate chip select (MCS0#) signal.

(2) Wait the number of cycles (WAITOE[4:0]) specified in the Timing Register before asserting the output enable (MOE#) signal.

(3) Sample the data using a sampling interval that is one cycle more than the read cycle wait control (WAITRD[4:0]) setting in the Timing Register. If this is the last read or write in a series, negate the control signals. The above Figure shows continuous reads.

Fig.27.12 Static Memory Read Timing

■ Static Memory Write Timing



(1) Se the address (MA[21:0]) and data (MD[15:0]) for the write and assert the appropriate chip select (MCS0#) signal.

(2) Wait the number of cycles (WAITWE[4:0]) specified in the Timing Register before asserting the write enable (MWE0#) signal.

(3) The write cycle lasts one cycle more than the write cycle wait control (WAITWR[4:0]) setting in the Timing Register.

(4) Hold the address and write data outputs for an additional cycle. If this is the last read or write in a series, negate the control signals. The above Figure shows continuous writes.

(5) Adjust the write enable (MWE0#) signal High level pulse to match the width (WAITWE[4:0]) specified in the Timing Register.

Fig.27.13 Static Memory Write Timing

27.2.3.5.2 SDRAM Controller AC Timing

The following Figures show the SDRAM controller AC timing.
The following Table summarizes the commands used in the Figures.

Command	Function	MCS2#	MRAS#	MCAS#	MWE1#	Address, etc.
ACT	Bank active	L	L	H	H	Bank/Row
RD	Read	L	H	L	H	Bank/Col
WR	Write	L	H	L	L	Bank/Col
BT	Burst terminate	L	H	H	L	—
PCGA	Prechage all banks	L	L	H	L	A10 = HIGH
PCG	Precharge	L	L	H	L	A10 = LOW
AREF	Auto refresh	L	L	L	H	MCLKEN = HIGH
SELF_IN	Self refresh start	L	L	L	H	MCLKEN = LOW
SELF_OUT	Self refresh end	H	x	x	x	MCLKEN = HIGH
LMR	Load mode register	L	L	L	L	—

■ SDRAM Read Cycle

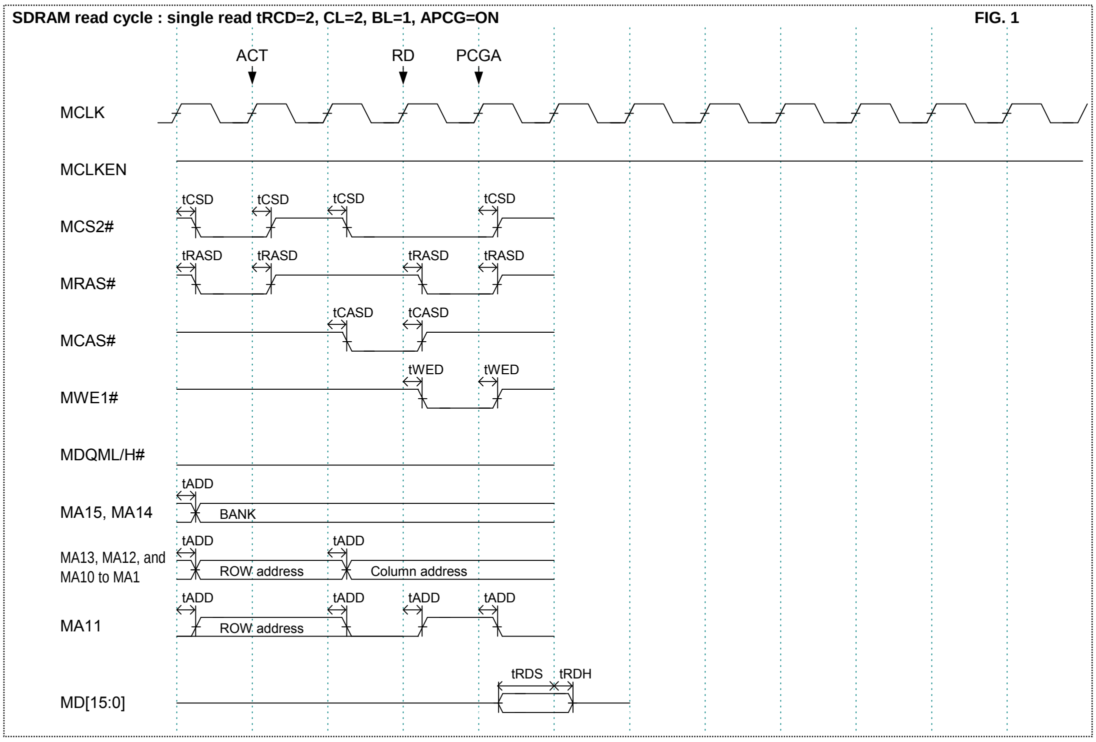


Fig.27.14 SDRAM Read Cycle 1: Single Read; t_{RCD}=2, CL=2, BL=1, APCG=ON

SDRAM read cycle : single read tRCD=2, CL=2, BL=4, APCG=ON

FIG. 2

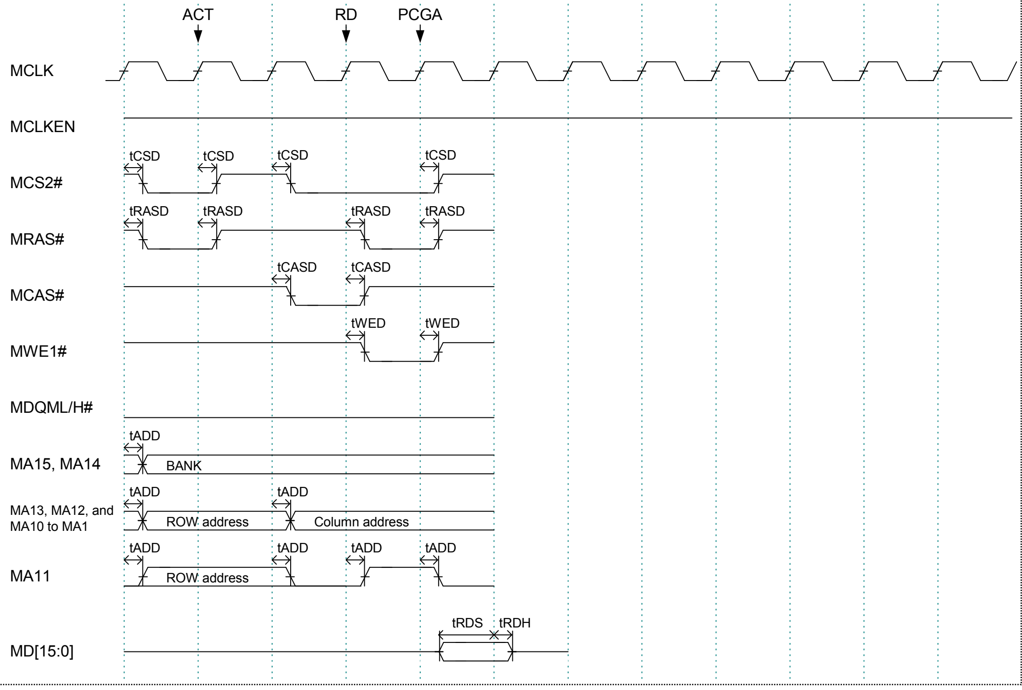


Fig.27.15 SDRAM Read Cycle 2: Single Read; t_{RCD}=2, CL=2, BL=4, APCG=ON

SDRAM read cycle : 6 burst read tRCD=2, CL=2, BL=1, APCG=ON

FIG. 3

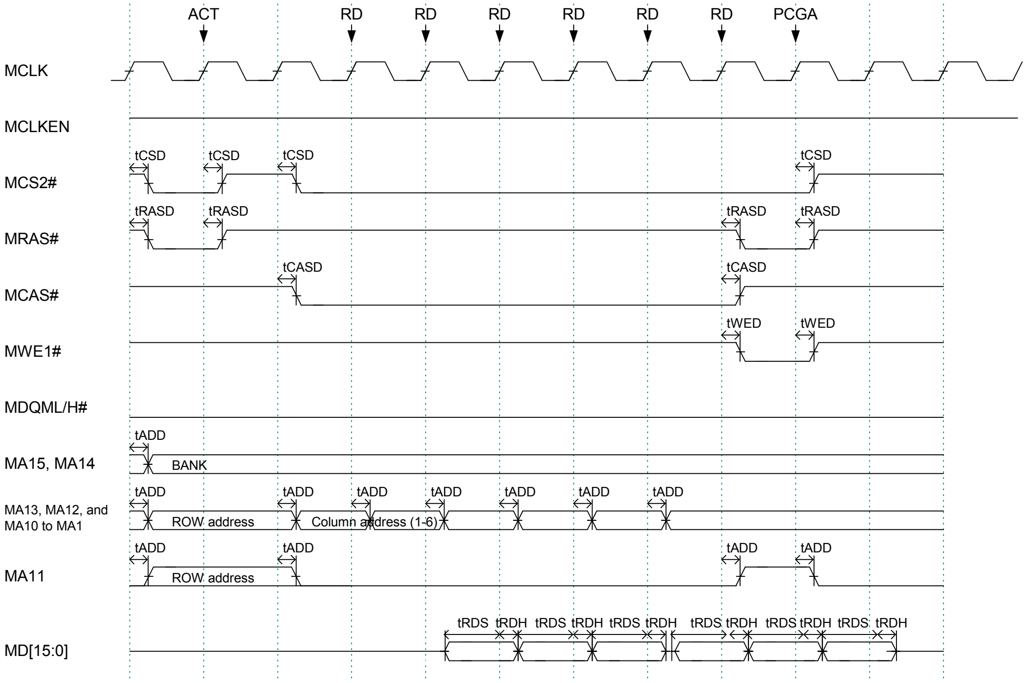


Fig.27.16 SDRAM Read Cycle 3: 6 Burst Read; t_{RCD}=2, CL=2, BL=1, APCG=ON

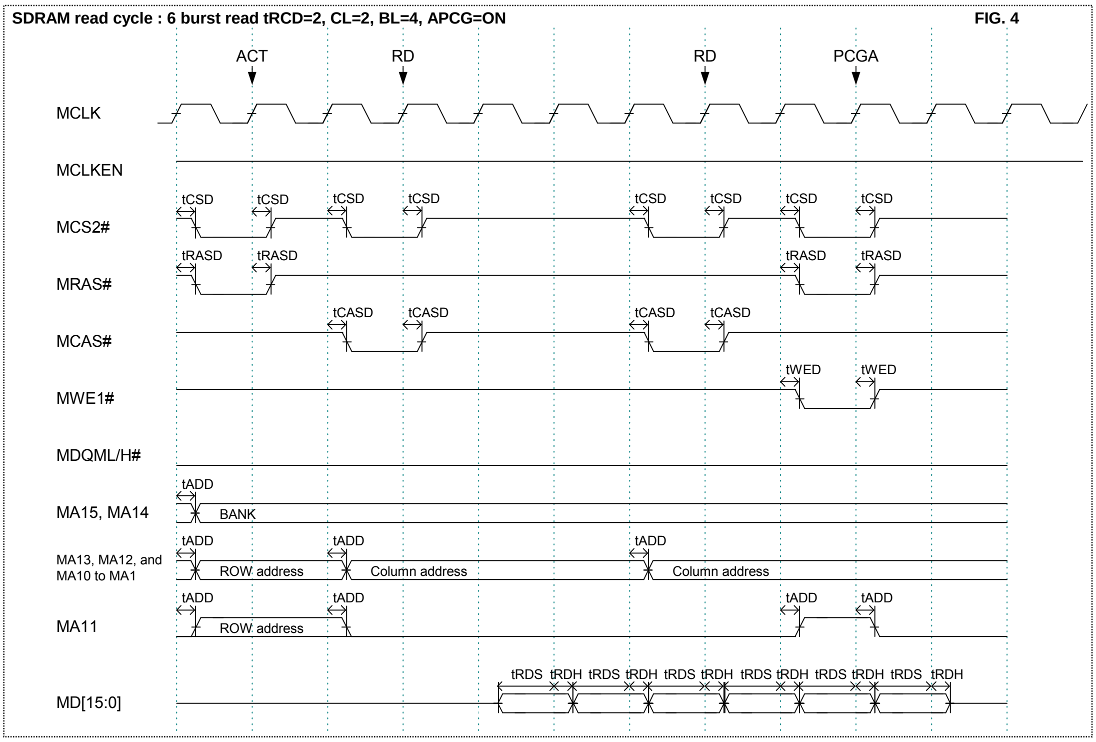


Fig.27.17 SDRAM Read Cycle 4: 6 Burst Read; tRCD=2, CL=2, BL=4, APCG=ON

■ SDRAM Write Cycle

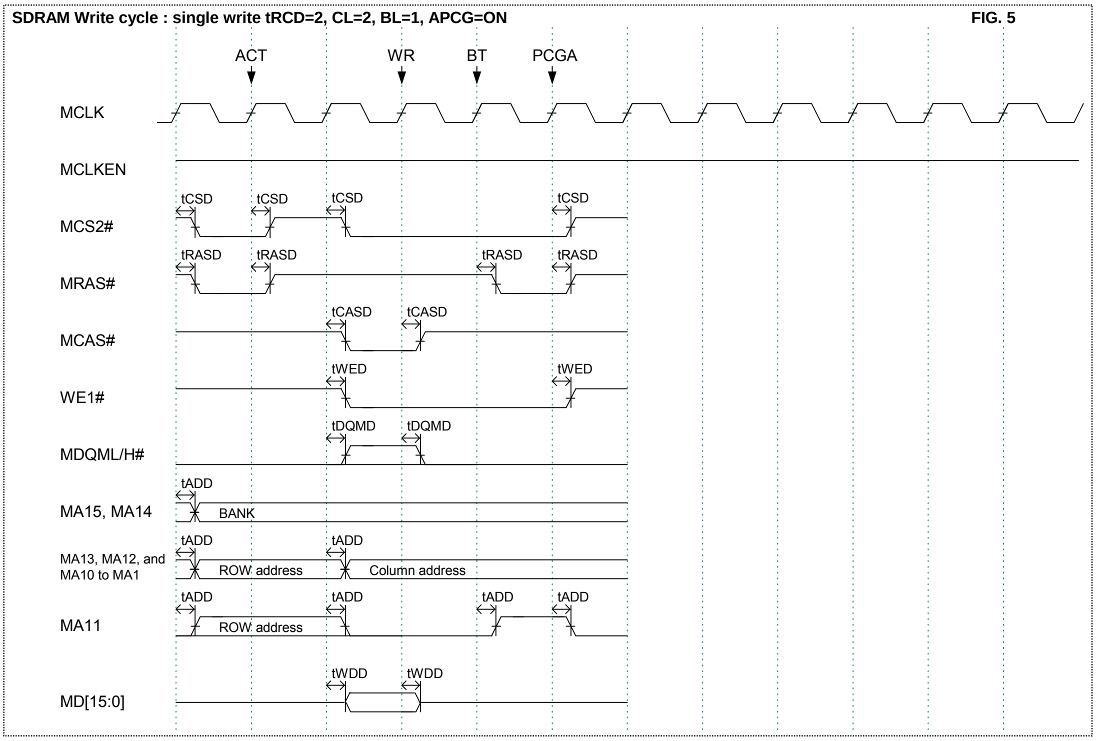


Fig.27.18 SDRAM Write Cycle 1: Single Write; tRCD=2, CL=2, BL=1, APCG=ON

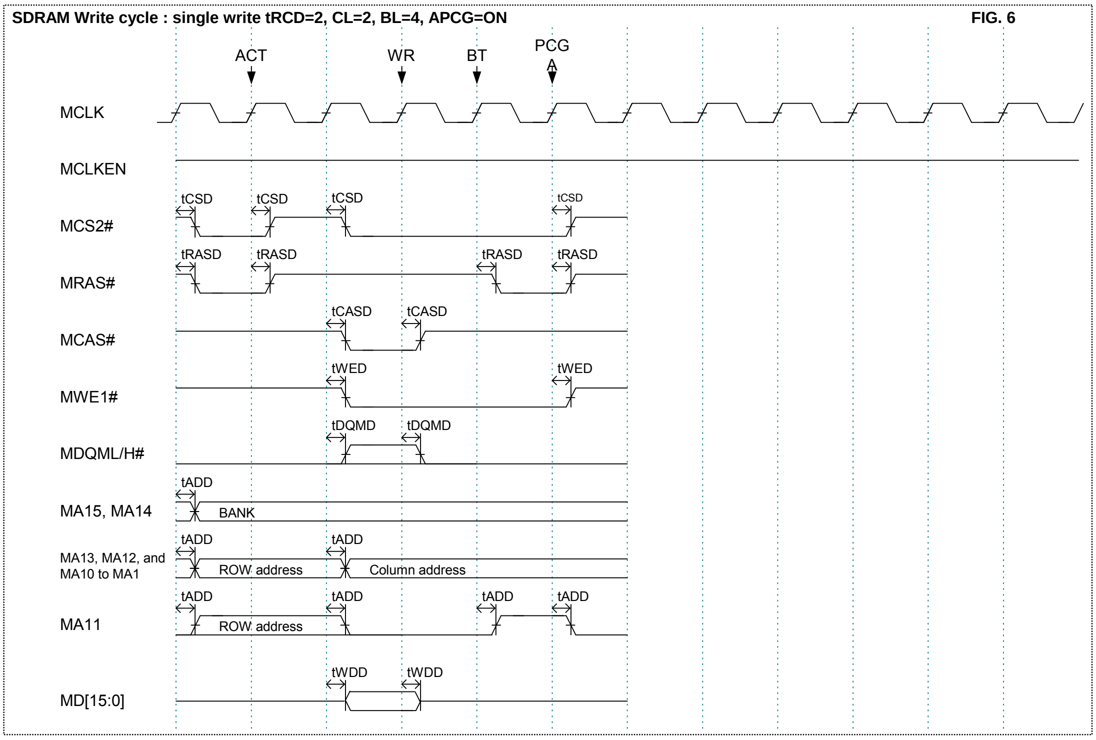


Fig.27.19 SDRAM Write Cycle 2: Single Write; t_{RCD}=2, CL=2, BL=4, APCG=ON

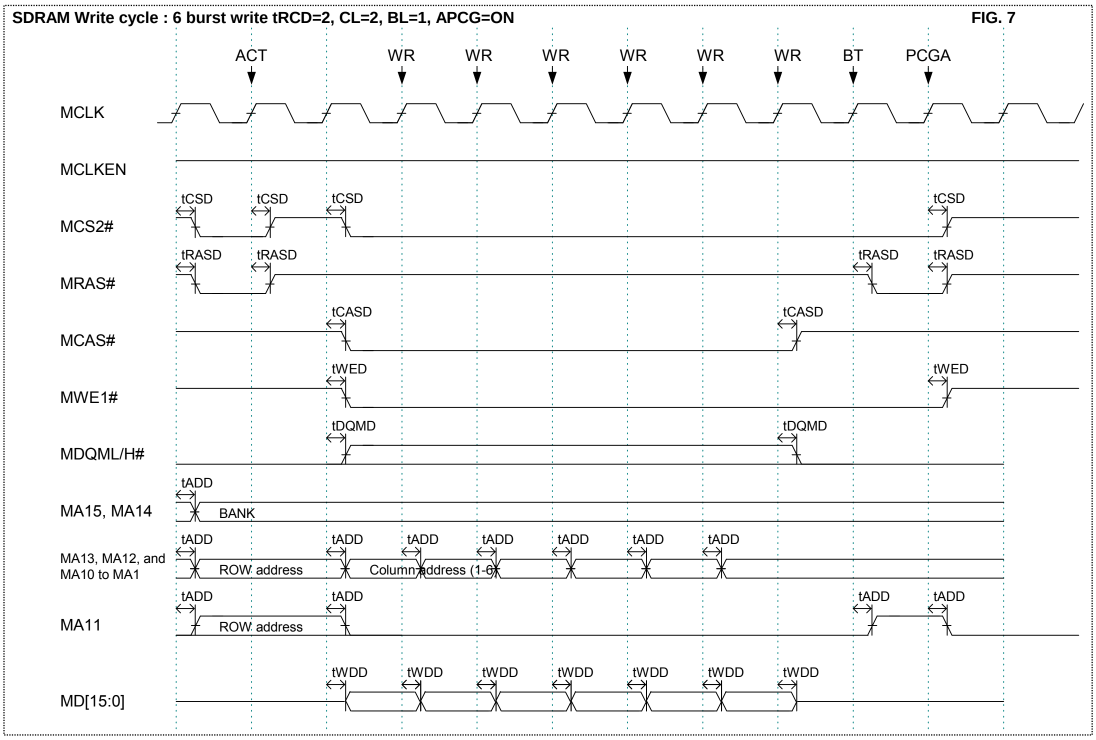


Fig.27.20 SDRAM Write Cycle 3: 6 Burst Write; t_{RCD}=2, CL=2, BL=1, APCG=ON

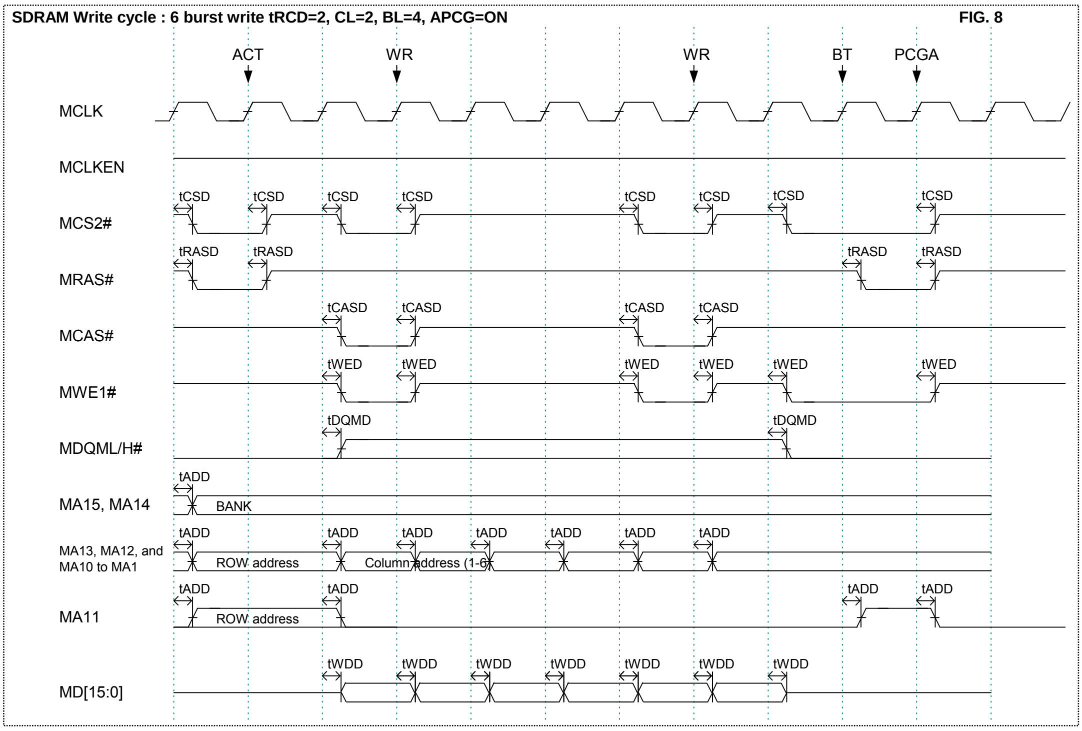


Fig.27.21 SDRAM Write Cycle 4: 6 Burst Write; tRCD=2, CL=2, BL=4, APCG=ON

■ SDRAM Read Cycle (Row Active Mode)

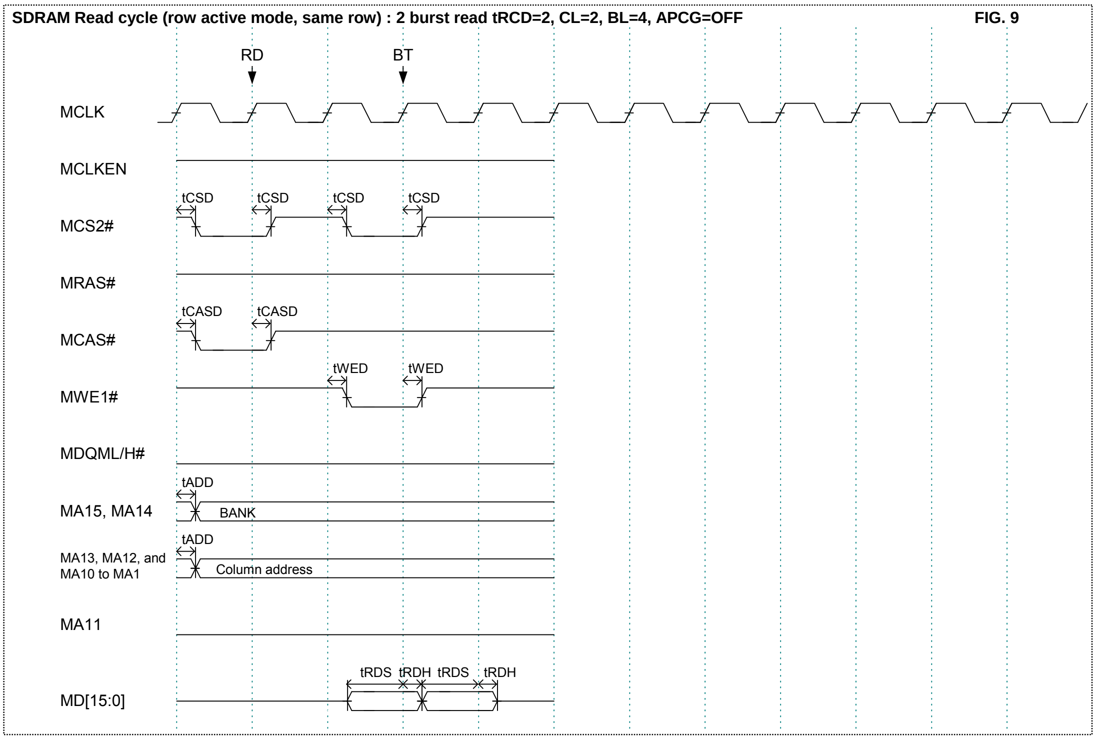


Fig.27.22 SDRAM Read Cycle Row Active Mode 1 (Same Row):
2 Burst Read; tRCD=2, CL=2, BL=4, APCG=OFF

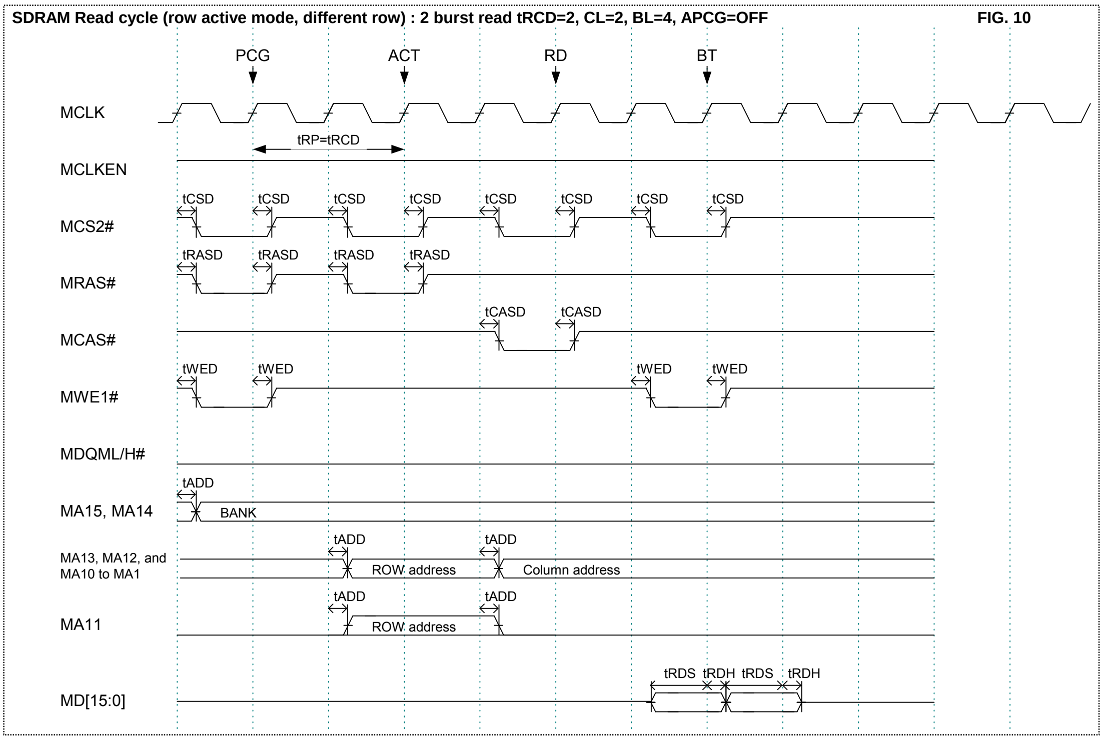


Fig.27.23 SDRAM Read Cycle Row Active Mode 2 (Different Rows): Burst Read; tRCD=2, CL=2, BL=4, APCG=OFF

■ SDRAM Write Cycle (Row Active Mode)

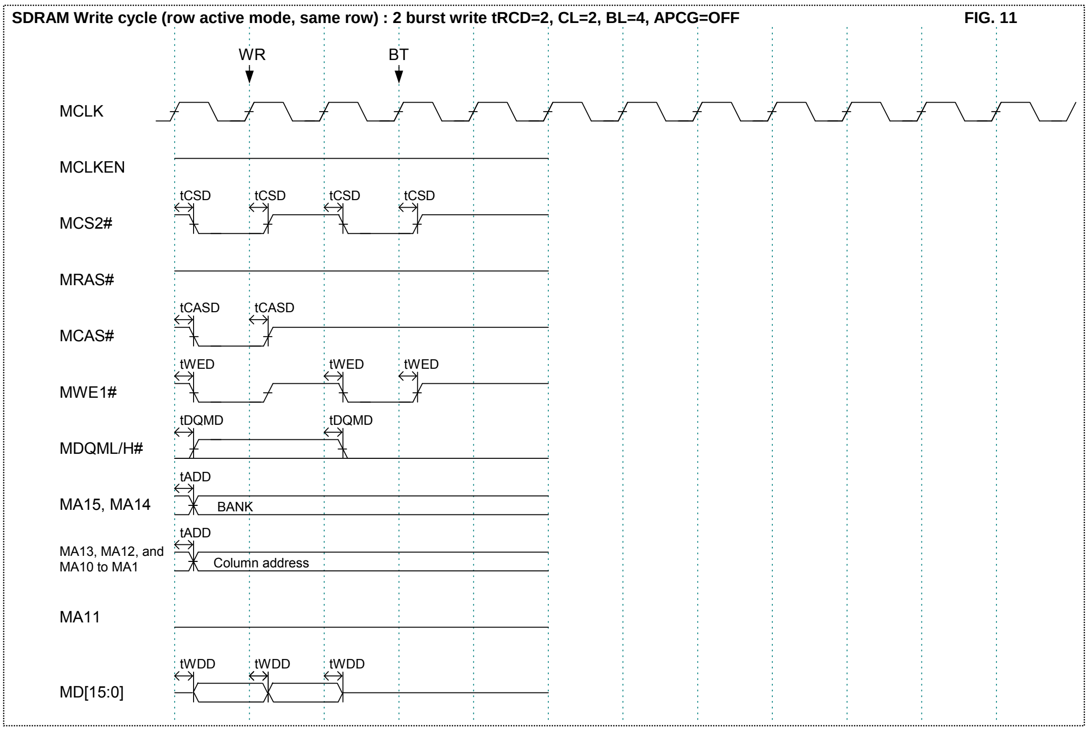

SDRAM Write cycle (row active mode, same row) : 2 burst write tRCD=2, CL=2, BL=4, APCG=OFF
FIG. 11
WR
BT
MCLK
MCLKEN
tCSD
MCS2#
MRAS#
tCASD
MCAS#
tWED
MWE1#
tDQMD
MDQML/H#
tADD
MA15, MA14
BANK
MA13, MA12, and MA10 to MA1
Column address
MA11
tWDD
MD[15:0]

Fig.27.24 SDRAM Write Cycle Row Active Mode 1 (Same Row): 2 Burst Write; tRCD=2, CL=2, BL=4, APCG=OFF

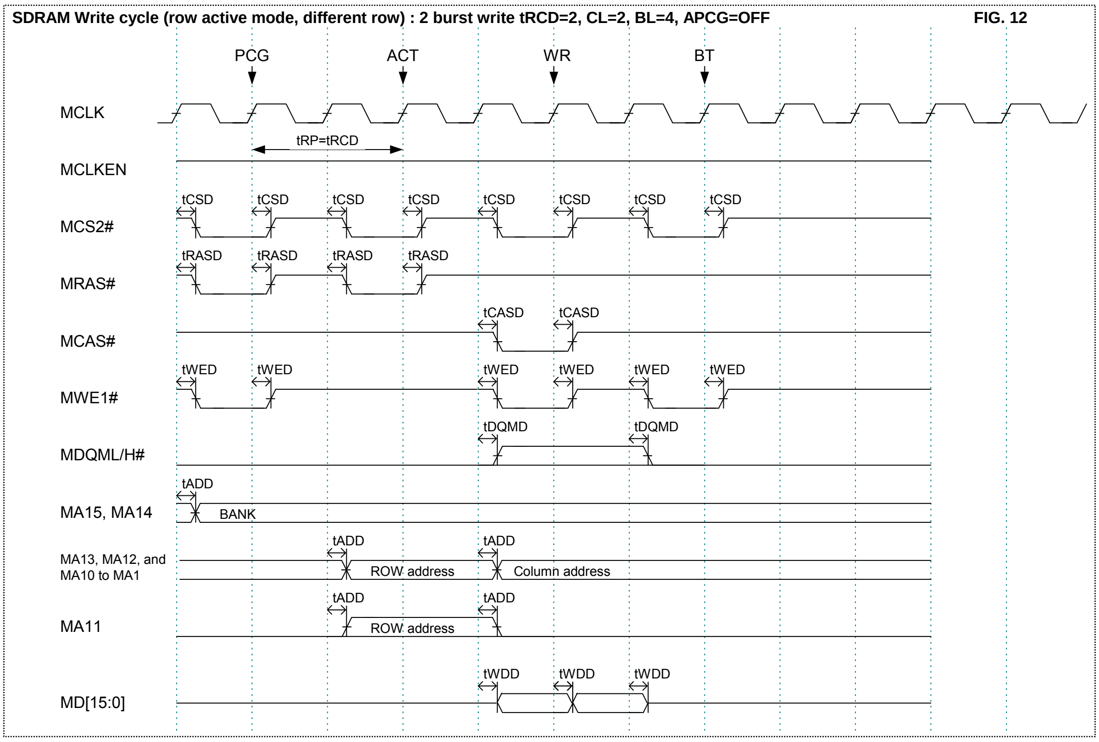


Fig.27.25 SDRAM Write Cycle Row Active Mode 2 (Different Row): 2 Burst Write; tRCD=2, CL=2, BL=4, APCG=OFF

■ SDRAM Auto Refresh Cycle

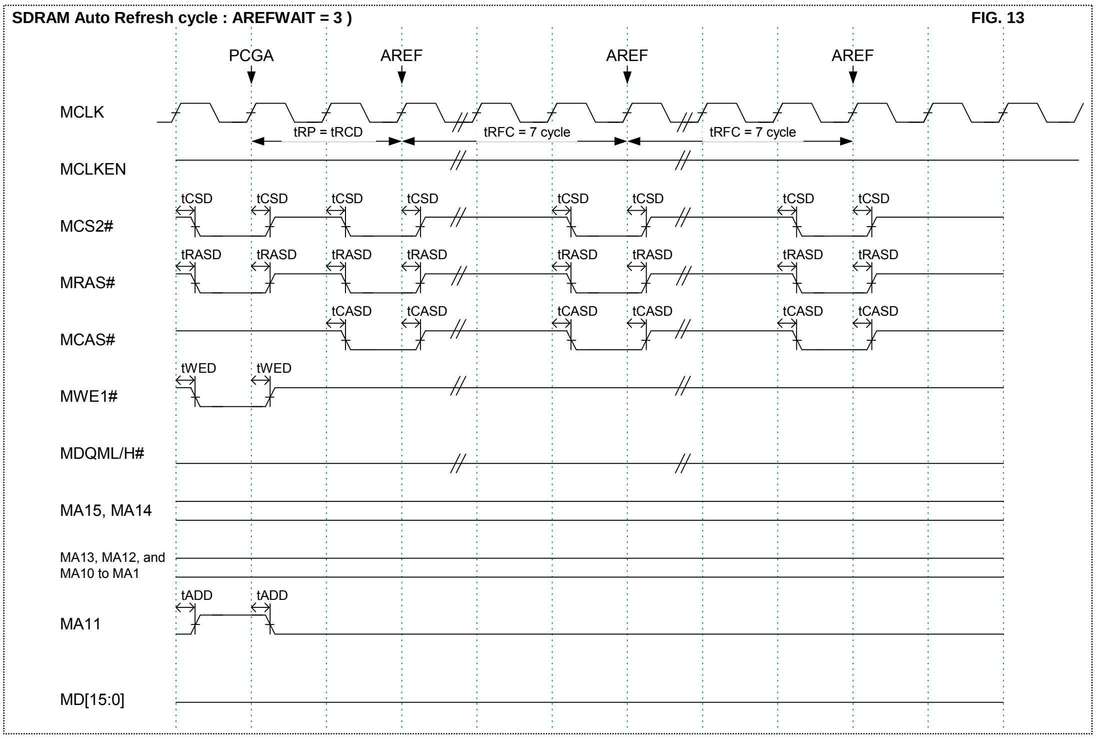


Fig.27.26 SDRAM Auto Refresh Cycle: AREFWAIT=3

■ SDRAM Self Refresh Cycle

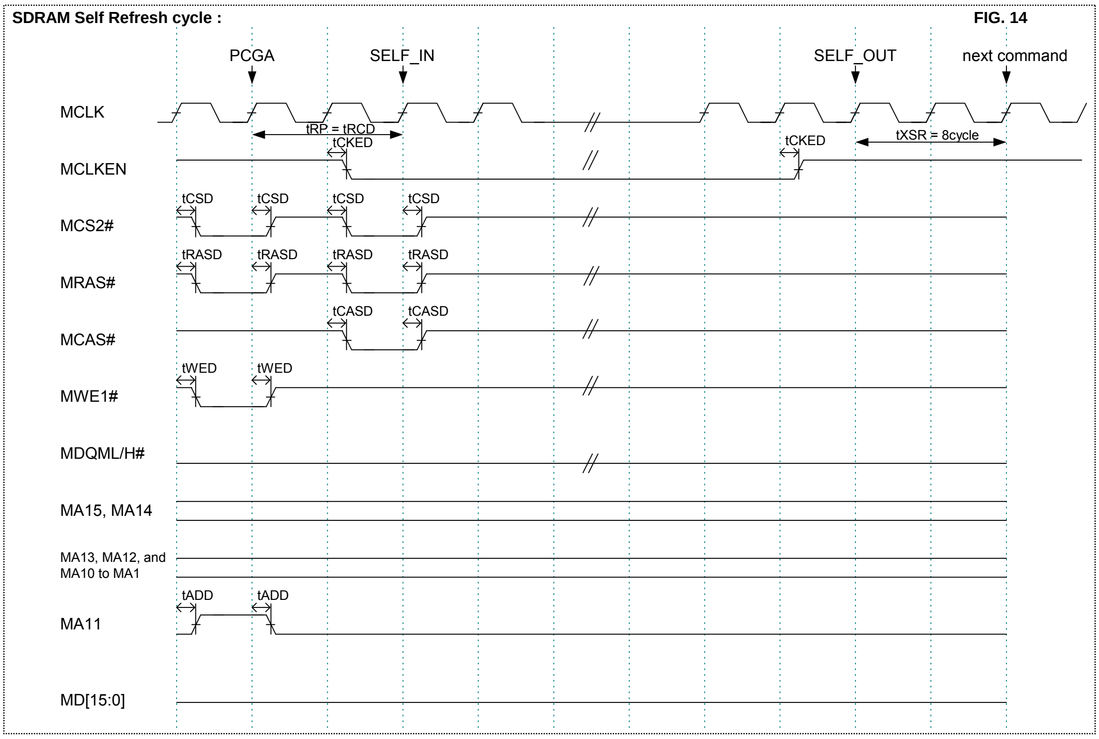


Fig.27.27 SDRAM Self Fresh Cycle

■ SDRAM Initialization Cycle

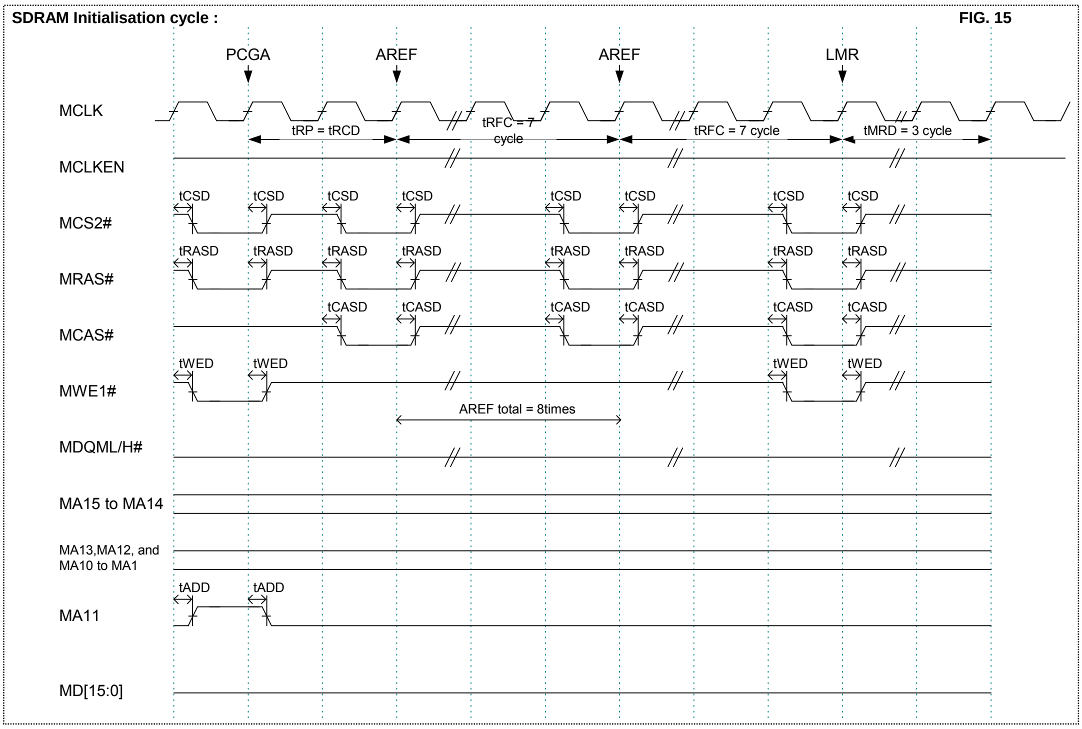


Fig.27.28 SDRAM Initialization Cycle

■ MCLK/MCLKEN Control

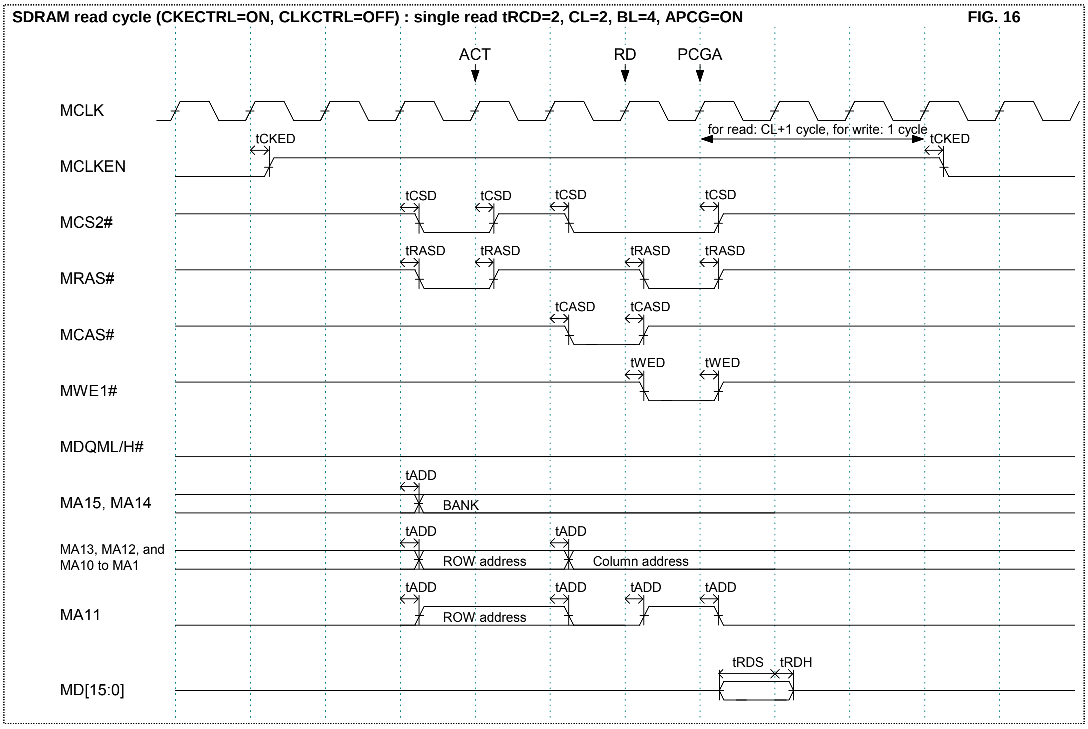

SDRAM read cycle (CKECTRL=ON, CLKCTRL=OFF) : single read tRCD=2, CL=2, BL=4, APCG=ON
FIG. 16
ACT
RD
PCGA
MCLK
MCLKEN
MCS2#
MRAS#
MCAS#
MWE1#
MDQML/H#
MA15, MA14
MA13, MA12, and MA10 to MA1
MA11
MD[15:0]
tCKED
for read: CL+1 cycle, for write: 1 cycle
tCSD
tRASD
tCASD
tWED
tADD
BANK
ROW address
Column address
tRDS
tRDH

Fig.27.29 MCLK/MCLKEN Control 1 (CKECTRL=ON, CLKCTRL=OFF): Single Read; t_{RCD}=2, CL=2, BL=4, APCG=ON

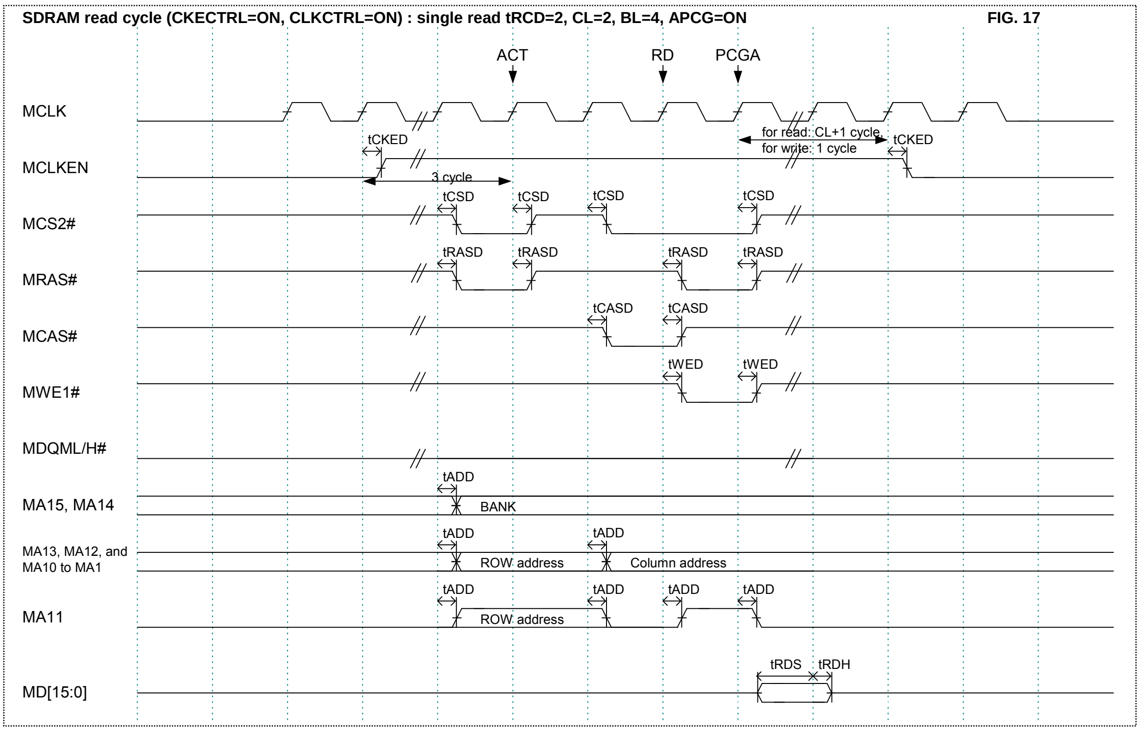


Fig.27.30 MCLK/MCLKEN Control 2 (CKECTRL=ON, CLKCTRL=ON): Single Read; t_{RCD}=2, CL=2, BL=4, APCG=ON

27.2.3.6 I^{2}C Single Master Core Module Timing

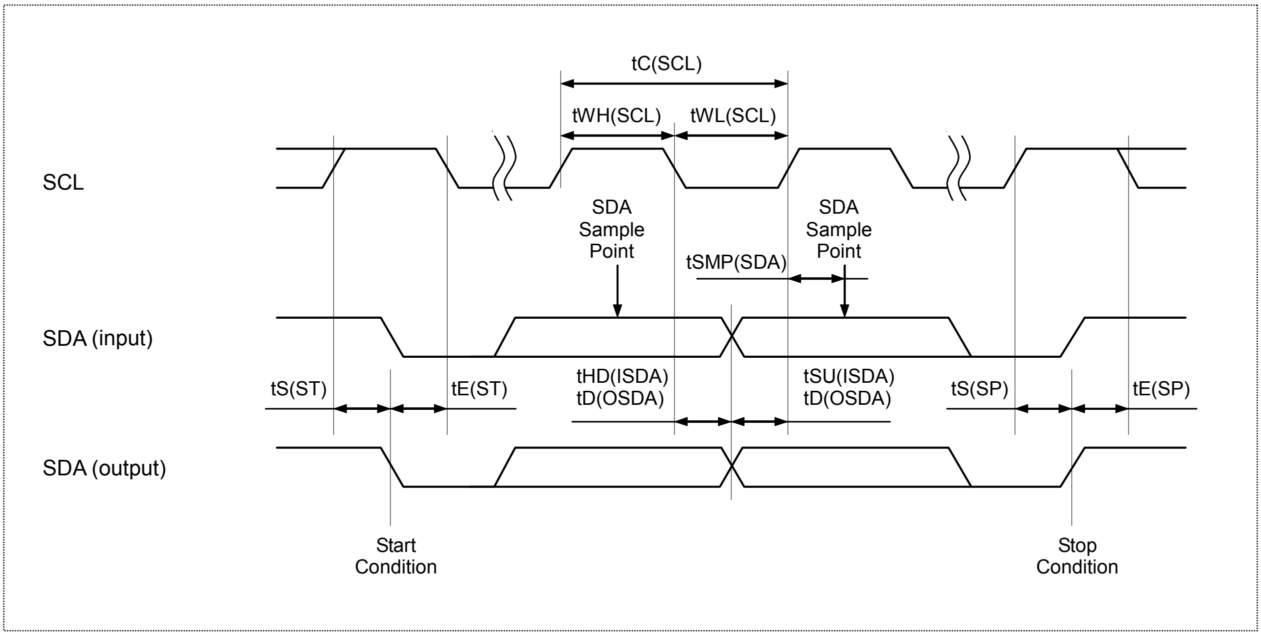


Fig.27.31 I^{2}C Single Master Core Module Timing

27.2.3.7 I^{2}S Timing

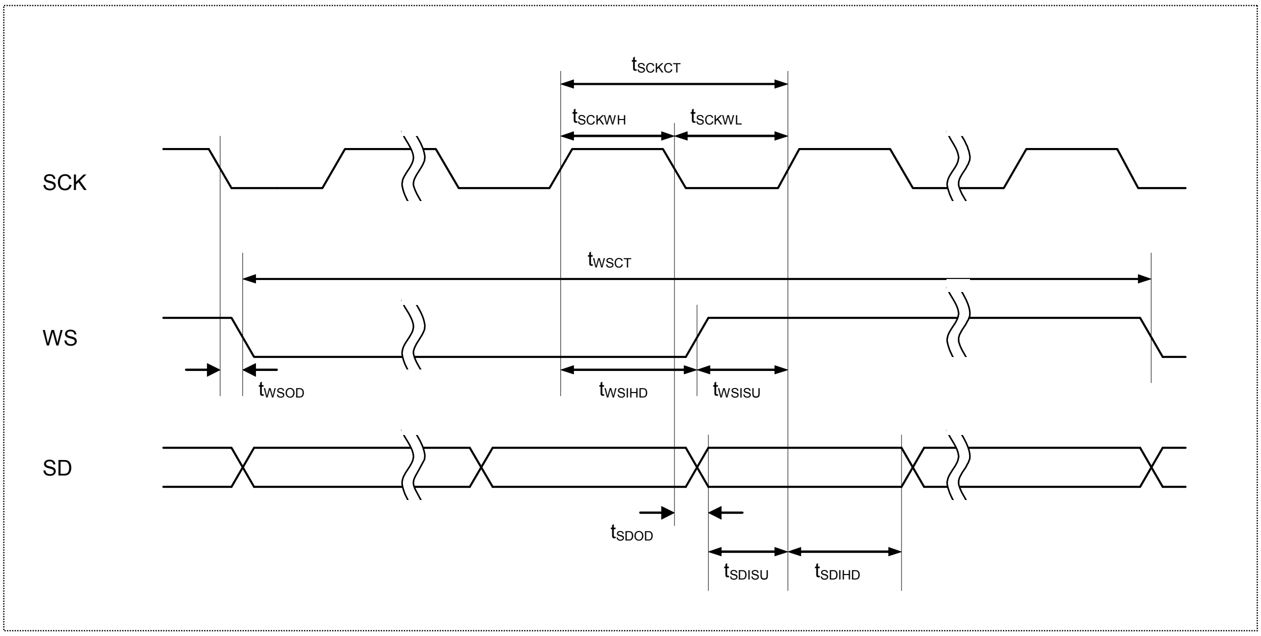


Fig.27.32 I2S Timing

27.2.3.8 Serial Peripheral Interface (SPI) Timing

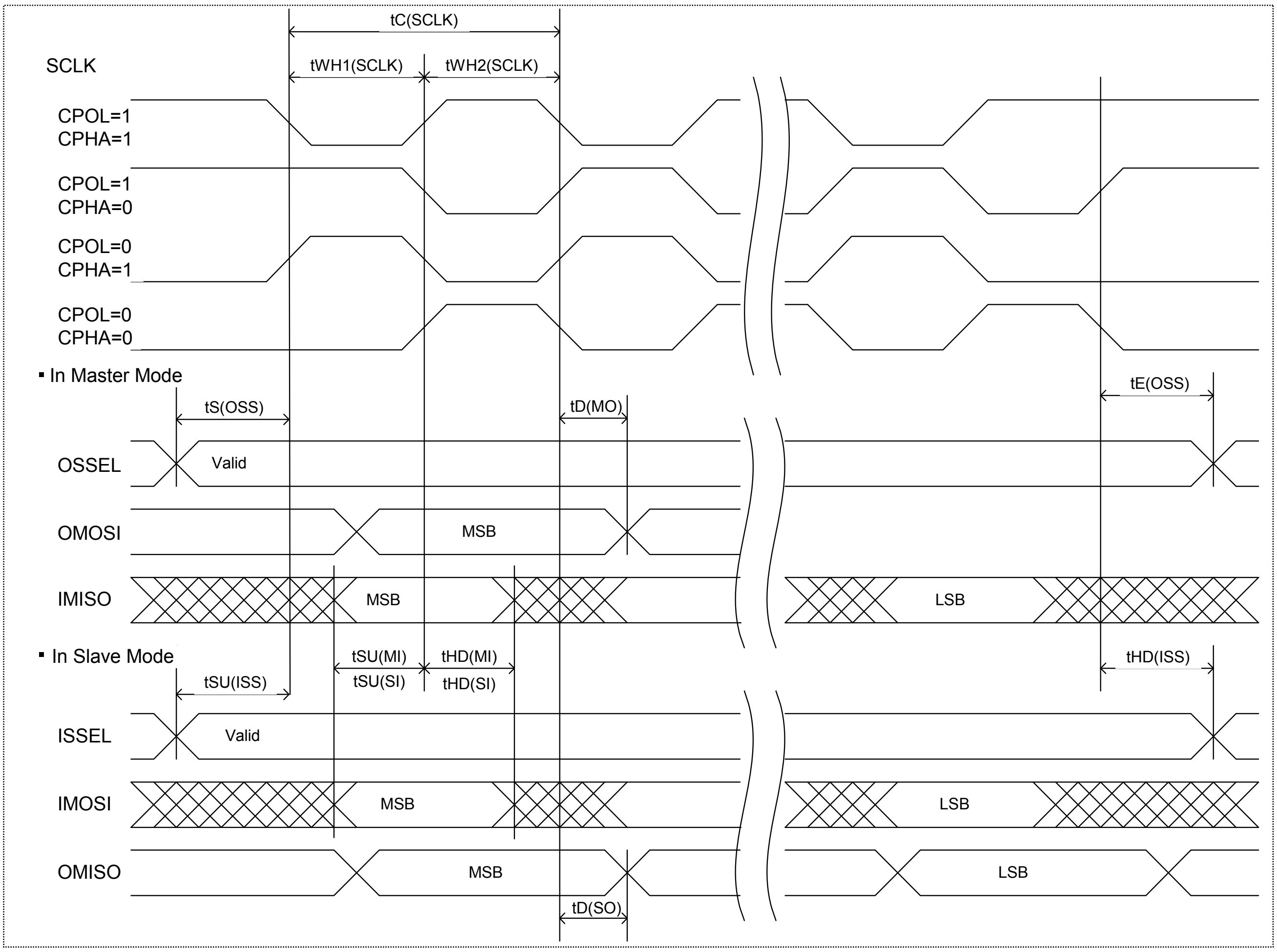


Fig.27.33 Serial Peripheral Interface (SPI) Timing

27.2.3.9 Compact Flash Interface (CF) Timing

■ CF Attribute Memory Read Cycle

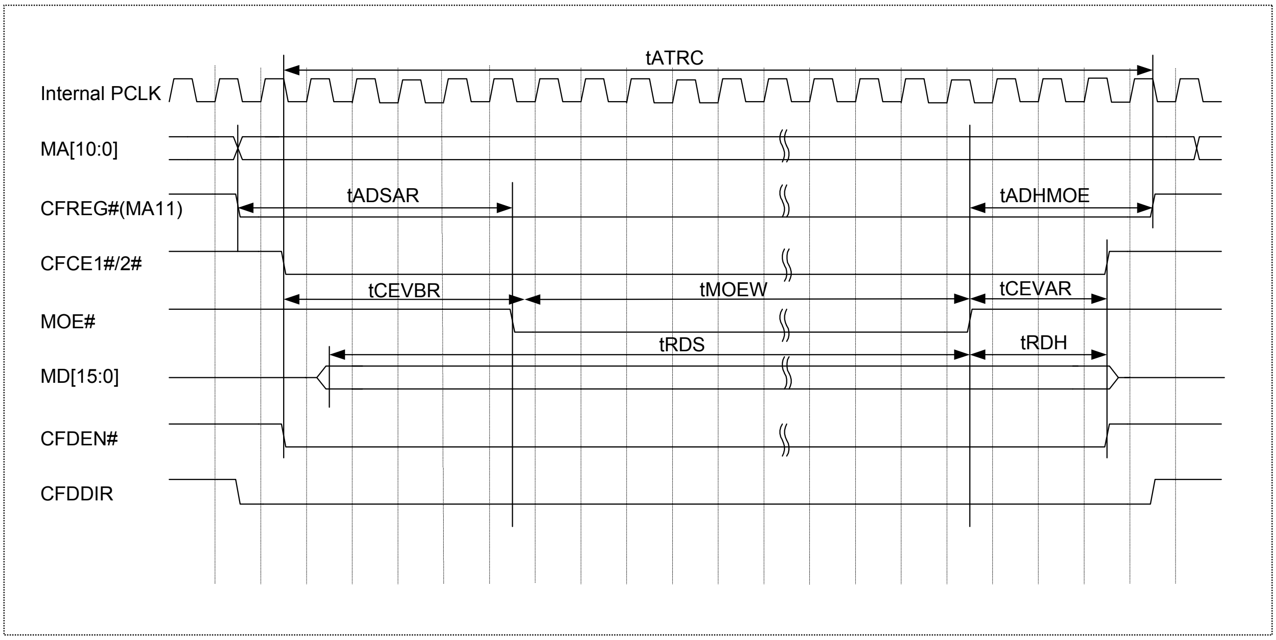


Fig.27.34 CF Attribute Memory Read Cycle

■ CF Attribute Memory Write Cycle

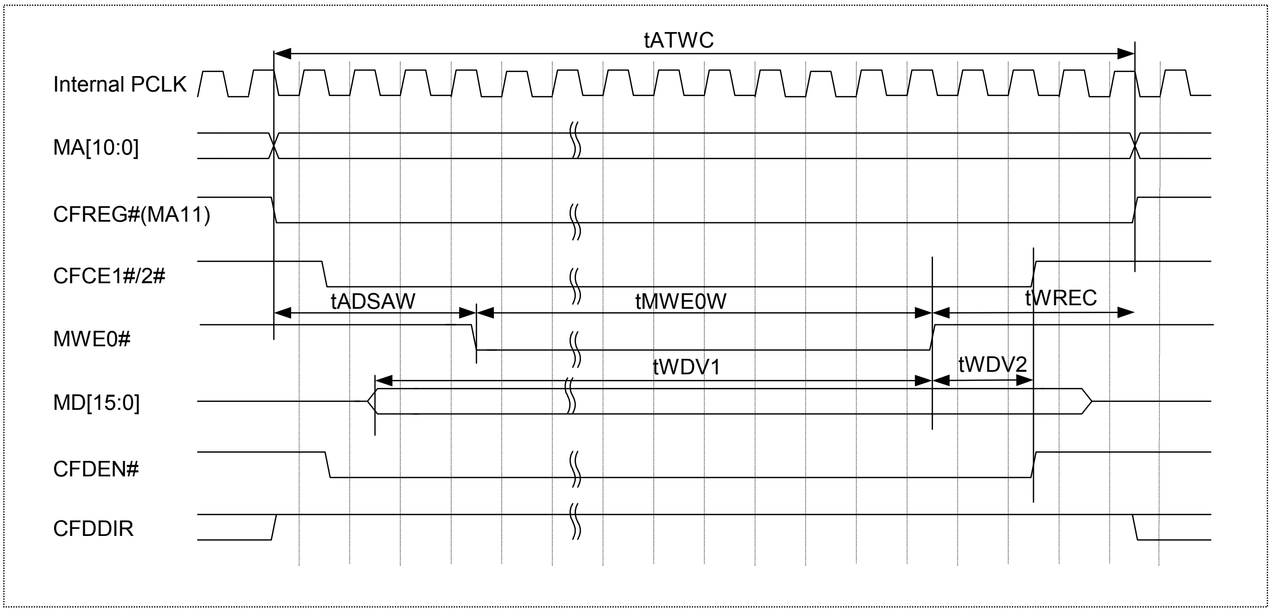


Fig.27.35 CF Attribute Memory Write Cycle

■ CF Common Memory Read Cycle

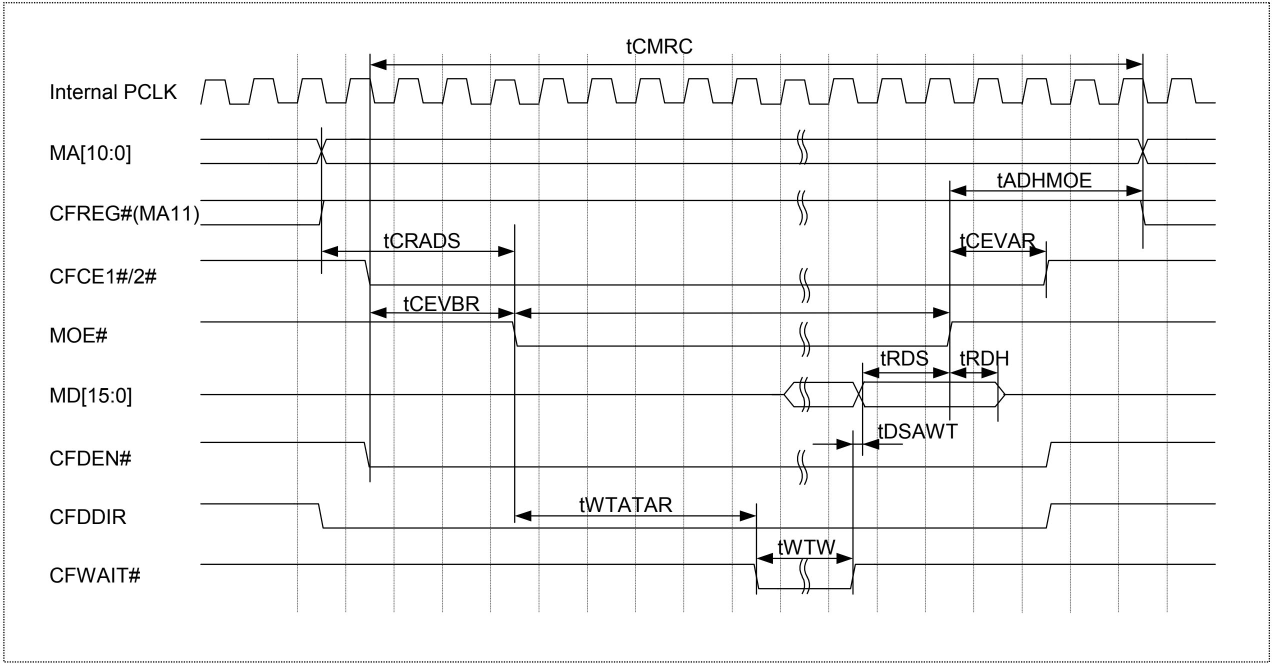


Fig.27.36 CF Common Memory Read Cycle

■ CF Common Memory Write Cycle

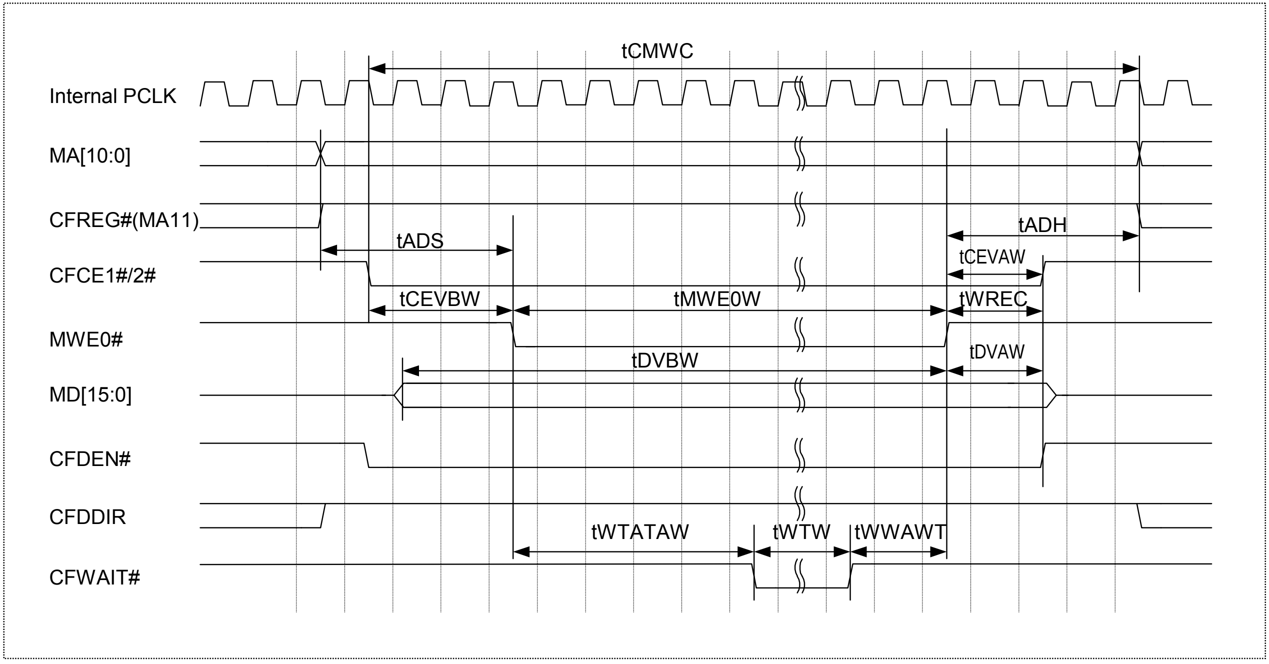


Fig.27.37 CF Common Memory Write Cycle

■ CF I/O Space or IDE Read Cycle

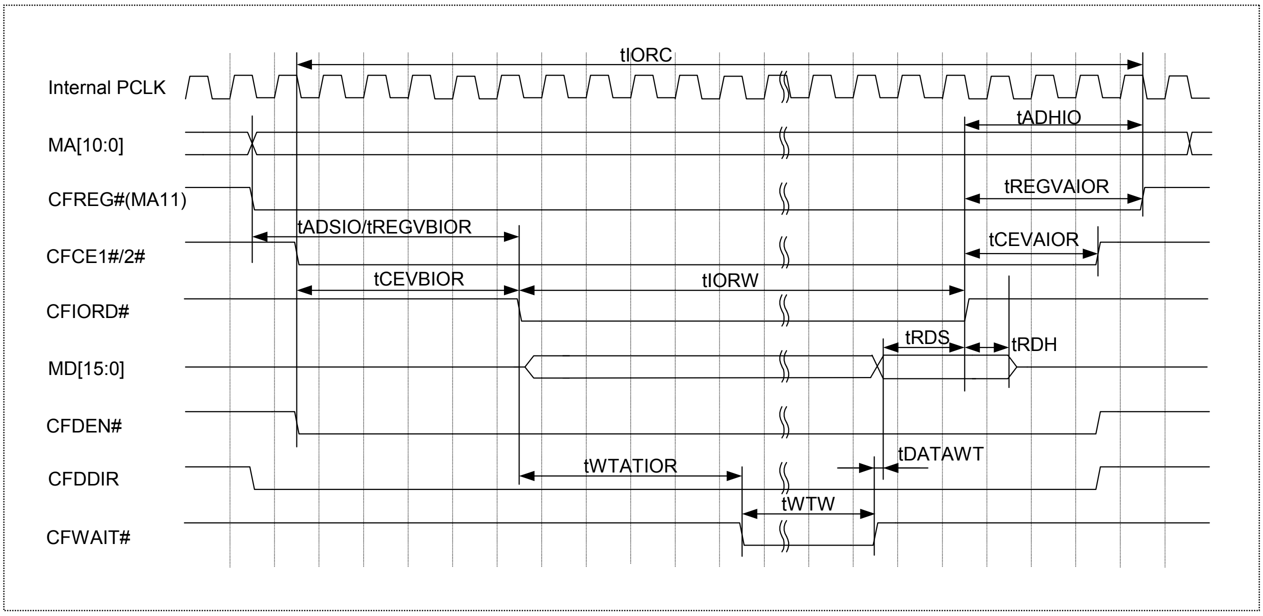


Fig.27.38 CF I/O Space or IDE Read Cycle

■ CF I/O Space or IDE Write Cycle

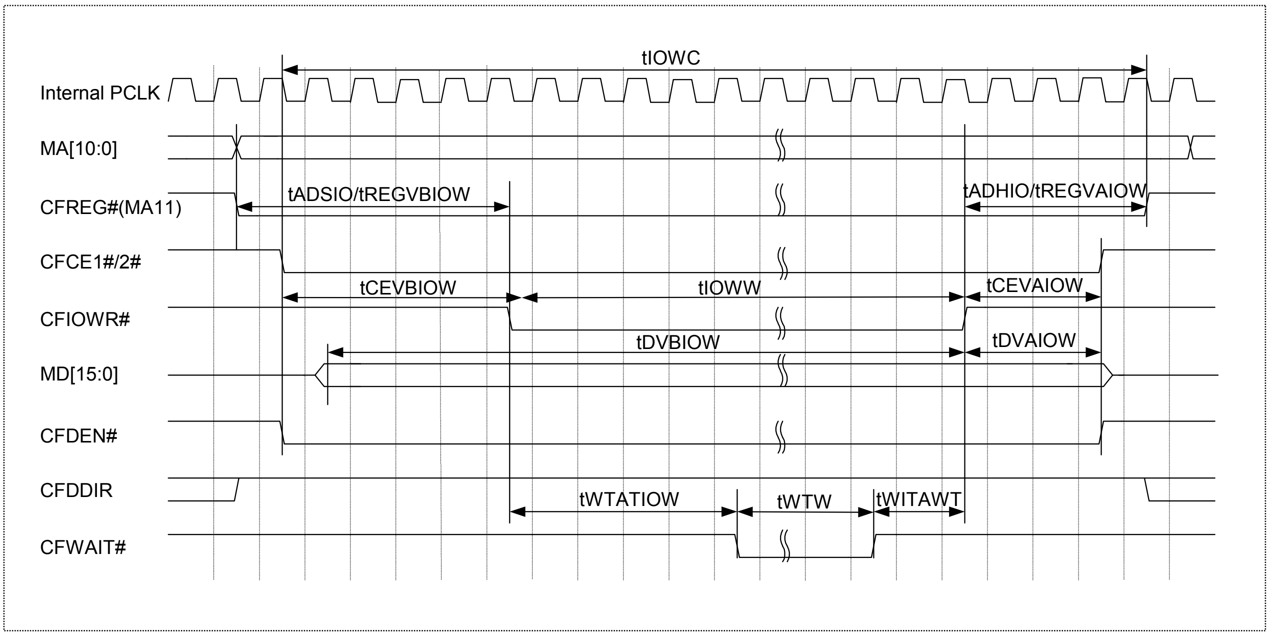


Fig.27.39 CF I/O Space or IDE Write Cycle

28. REFERENCE: SAMPLE EXTERNAL CONNECTIONS

28.1 Sample Memory Connections

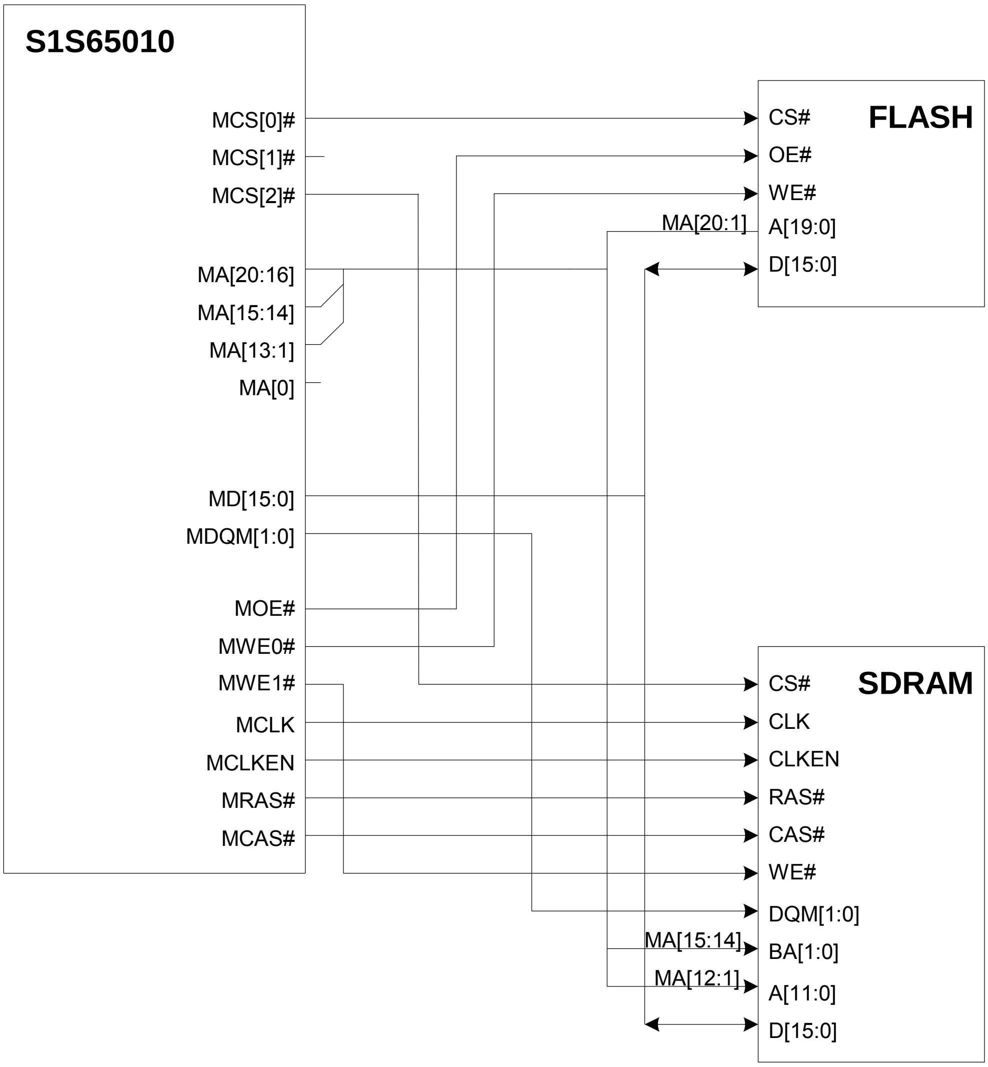


Fig.28.1 Sample Memory Connections 1

Note: Connect MA[15:14] to the SDRAM bank Address (BA[1:0]).

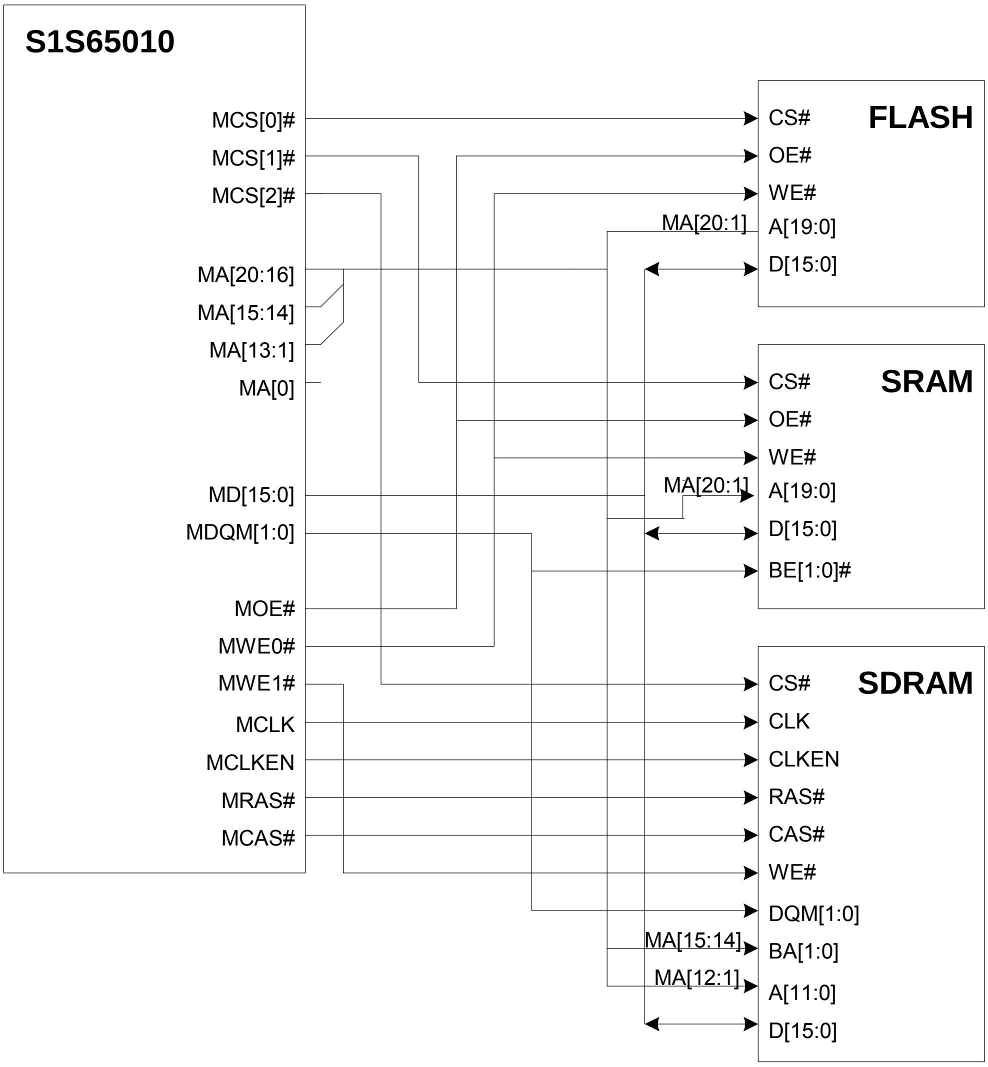


Fig.28.2 Sample Memory Connections 2

Note: Connect MA[15:14] to the SDRAM bank Address (BA[1:0]).

28.2 Sample Compact Flash Connections: 16-Bit Bus

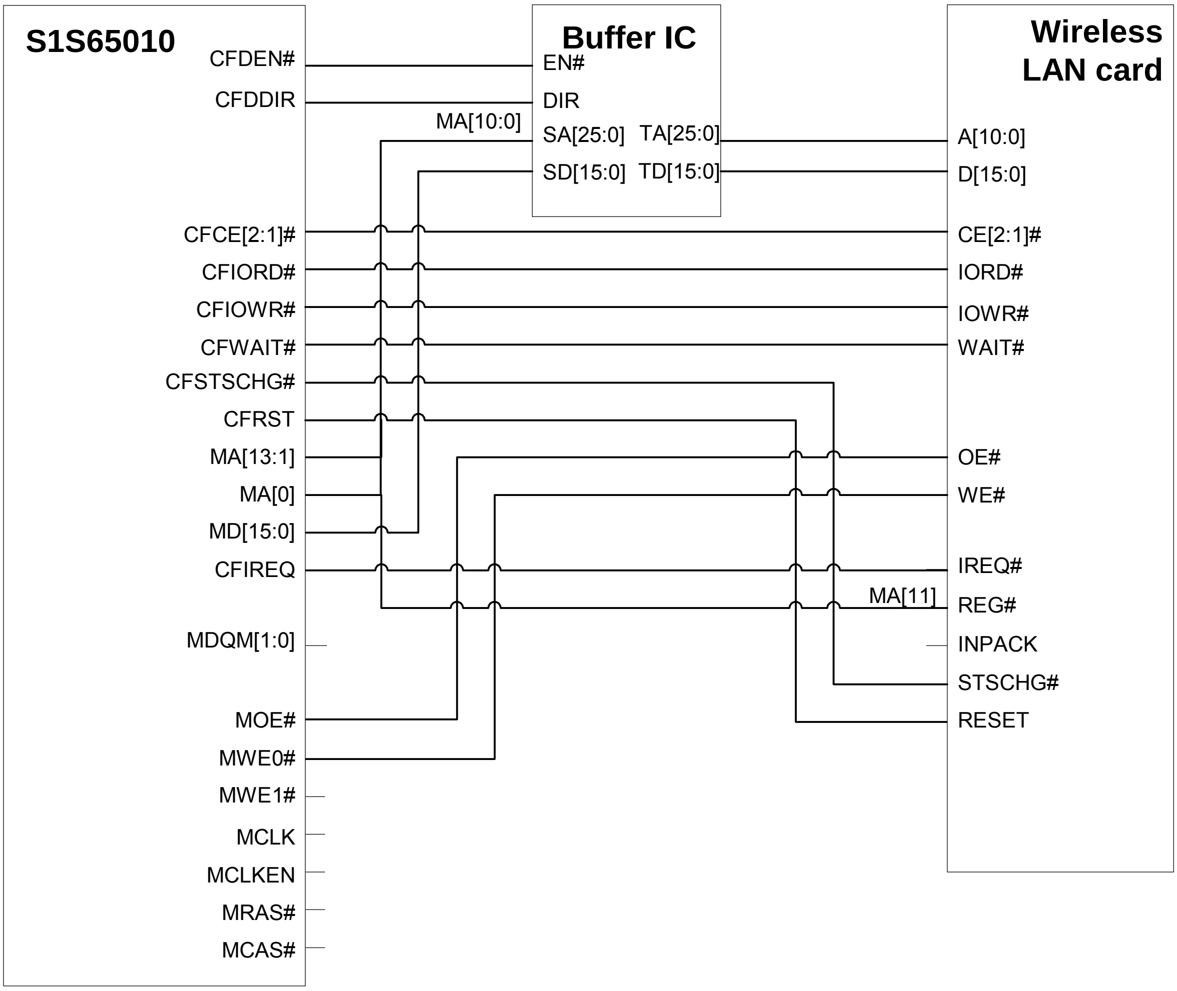
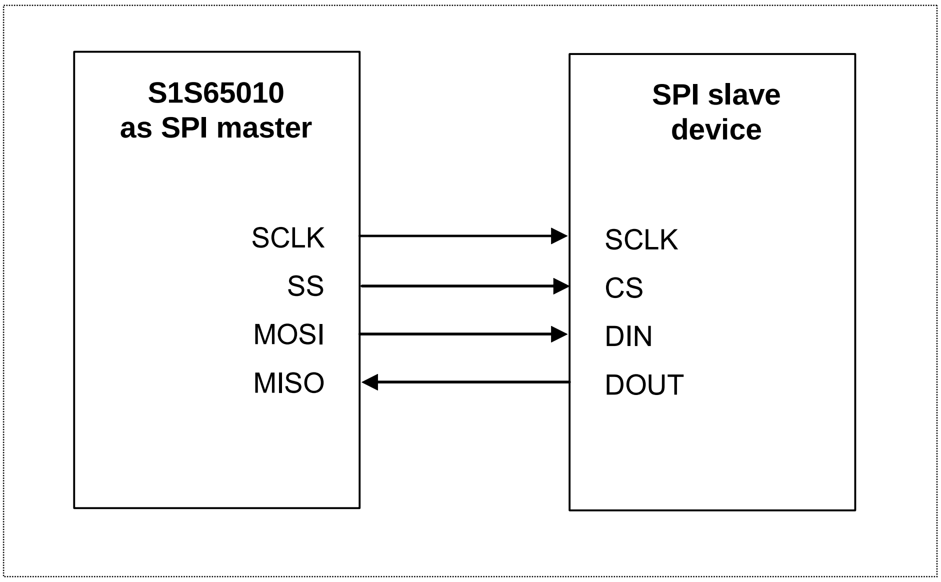


Fig.28.3 Sample Compact Flash Connections: Wireless LAN Card

28.3 Sample Serial Peripheral Interface (SPI) Connections

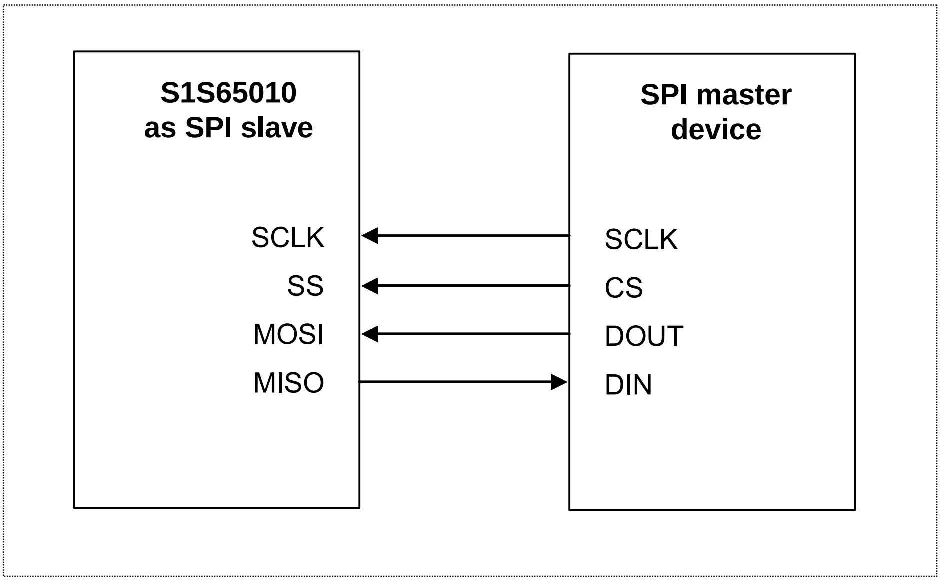
28.3.1 As Master

This is a sample connection for this device as master.



28.3.2 As Slave

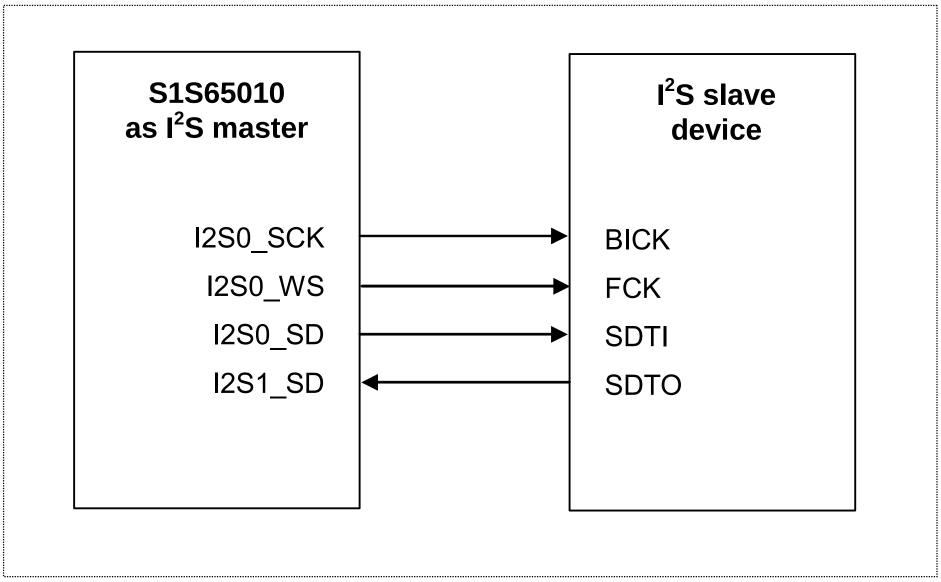
This is a sample connection for this device as slave.



28.4 Sample I^2S Connections

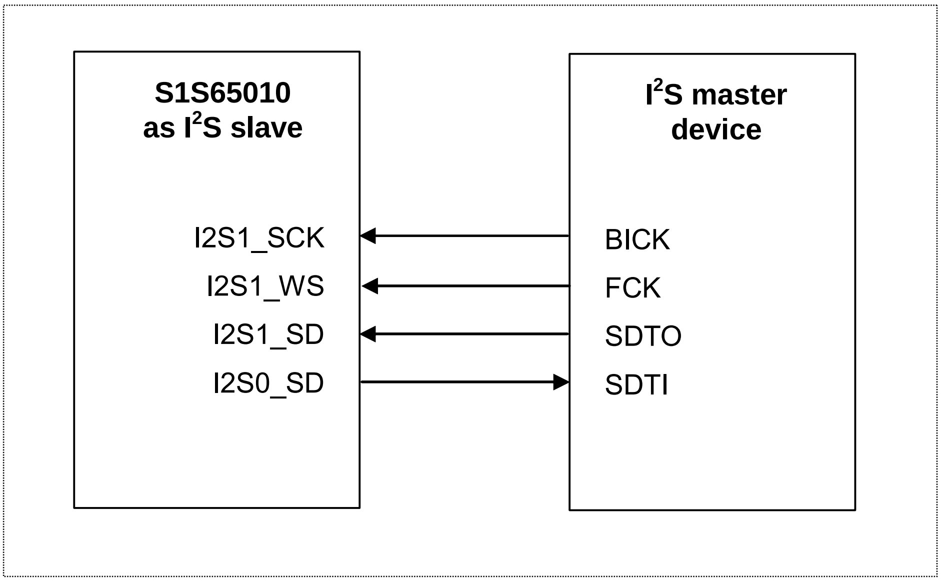
28.4.1 As Master

This is a sample connection for this device as master.



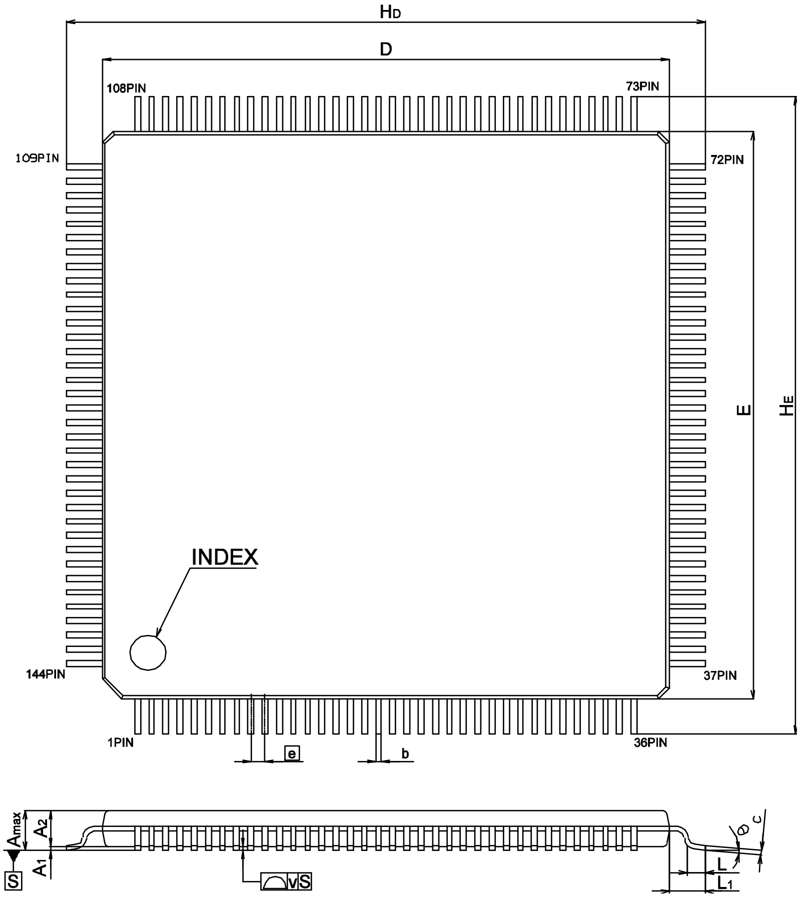
28.4.2 As Slave

This is a sample connection for this device as slave.



29. EXTERNAL DIMENSIONS

29.1 Plastic TQFP: 144 Pins, Body Size 16x16x1 mm (TQFP24)



Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
E	—	16	—
D	—	16	—
A_{max}	—	—	1.2
A_1	—	0.1	—
A_2	—	1	—
$\boxed{e}$	—	0.4	—
b	0.13	—	0.23
c	0.09	—	0.2
θ	0°	—	8°
L	0.3	—	0.7
L_1	—	1	—
H_E	—	18	—
H_D	—	18	—
y	—	—	0.08

Fig.29.1 TQFP24-144PIN Package Dimensions

30. REVISION HISTORY

Revision	Revised on	Description		
		Revised part	Before revision	After revision
0.1	2004/04/19	New creation （Preliminary）		
0.2	2004/06/21	Error correction: The major corrections are as shown below. The others, for example, corrections of character, format, and font errors, are not related to the contents of this manual.		
		p.3: Supported Protocols: Modification of description order and addition of description and notes	ARP, ICMP, IP, TCP, DHCP, FTP, HTTPd, SMTP, DNS resolver:	ARP, ICMP, IP, TCP, UDP (added), HTTPd, SMTP, DHCP*, FTP*, DNS resolver*, telnet* (added)： Note (*): Handled as sample codes.
		p.8: PINS, Table 3.1: RESET#	ICS	ICSU1
		p.15: Table 3.2 RESET# Addition of description		Change to the Schmidt trigger input with pull-up resistance.
		p.17: Pin Configurations During and After a Reset (RESET#)	None —	With pull-up resistance 50kΩ
		p.56, 60: Resizer Operation Register RSZ[0xD8] Default Value	0x0001	0x8080
		p.78: JPEG Line Buffer Memory Address Offset Register	JLB[0xA0]	JLB[0xA4]
		p.129, 135: ETH Mode register Default Value	0x4000_0040	0x4000_0000
		p.135: ETH Mode register ETH[0x 20] bit [7:0]	Same description as high-order 1 byte	Delete the description. Reserved. （0）
		p.139: ETH MAC Address Registers 1 to 8: High-order 16 bits ETH [0x34, 0x44, 0x4C, 0x54, 0x5C, 0x64, 0x6C] Bits [31:16] Description	Reserved.	Reserved. Add the description in parentheses ( ).
		p.140: ETH Buffer Management Register ETH [0x90] Buffer Management Enable bit	bit 31	Move the bit value to bit 0.
		p.141: ETH Buffer Free Register ETH [0x94] Buffer Free bit	bit 31	Move the bit value to bit 0.
		p.141: ETH Buffer Information Register Default Value	0x0000_0000	0x03FF_03FF
		p.159: APB APB device correction for Table 12.2 (See the following.)		
		PW01CNF	DMAC1	Reserved.
		PW03CNF	Reserved.	DMAC1
		PW0ACNF	JPEG FIFO	JPEG module FIFO control
		PW0BCNF	JPEG encoder	JPEG codec
		PW0CCNF	JPEG master	JPEG DMAC
		PW16CNF	Reserved.	UARTL (UART Lite)
		p.163, 164: SYS [0x08] Register Correction of Default Value	0x 0421_84AE	0x 0421_D46A
		p.165: SYS [0x10] register Bits [3:0] 1st line in Description	Because "0" was written to the relevant bit.	Because "0" was written to the relevant bit.
		p.199: INT Table 15.1 Note (*): Change of description about IRQ [29:24, 22:20]		
		p.248 and after: I2C 18.5: All changes of operation description		

Revision	Revised on	Description		
		Revised part	Before revision	After revision
0.2 (Continued)	2004/06/21	p.288, 295 ： TIM[0xA0] [0xA4] Register, Correction of Default Value	0x 00XX	0x 0000
		p.339, 360: AC characteristics Addition of Table 27.10 and Fig. 27.32		Add I2S timing.
		p.369: Addition of I2S connection example		Add I2S connection example. (In master or slave mode)
		p.375 to 382: Appendix 1	—	Correct a writing error in the register list.
0.3	2004/07/06	6. DMA Controller 1 (DMAC1) and the Related Description		
		p.29 to 42: Change the description about the offset address in register details.	DMA[0xXX]	DMAC1[0xXX]
		8. JPEG Controller (JPG) and the Related Description		
		p.64, 65: JCTL [0x04] Register Delete the function of bit 3.	JPEG Codec I/O Error Flag	Reserved
		p.67, 68: JCTL [0x08] Register Delete the function of bit 3.	Raw JPEG Codec I/O Error Flag	Reserved
		p.6: JCTL [0x0C] Register Delete the function of bit 3.	JPEG Codec I/O Error Interrupt Enable	Reserved
		9. JPEG_DMAC (JDMA) and the Related Description		
		p.114: JDMA [0x0C] Register Change the bit-5 selection function.	1: Burst transfer	1: Demand transfer
		10. DMAController 2 (DMAC2) and the Related Description		
		p.119 to 126: Change the description about the offset address in register details.	DMA[0xXX]	DMAC2[0xXX]
		p.118, 126: DMAC2 [0x70] Register Addition of new register	—	Add a new DMA Channel Transfer End Control Register (TECL).
		p.121: DMAC2 [0x0C] Register Change the selection function of bit 5.	1: Burst transfer	1: Demand transfer
		p.124: DMAC2 [0x1C] Register Change the selection function of bit 5.	1: Burst transfer	1: Demand transfer
		p.125: DMAC2 [0x60] Register Add a new function of bit 9.	n/a	DPE DMA Priority Change Enable
		13. SYSTEM CONTROLLER (SYS) and the Related Description		
		p.162: 13.2.4 Full Operation Mode: Delete the description. (Last 2 lines)	Set to the CPU operating frequency.	Deleted.
		19. I^2S (I2S) and the related description		
		p.256: I2S0 [0x08], I2S1 [0x48] Change the register details.	The transfer port access size...	The data that can be read and written by one access...
		p.261: 1.5.2: All changes of description about data width and FIFO step count	The transfer port access size...	The data that can be read and written by one access....
0.4	2004/08/05	Correction of writing error: The major corrections are shown below. The others, for example, corrections of character, format, and font errors, are not related to the contents of this manual.		
		6. DMA Controller 1 (DMAC1)		
		p.27: 6.2: Block Diagram	—	Fig. 6.1 Addition of figure number and table title
		7. Camera Interface (CAM) and the related information		
		p.52: 7.5 Description of Operation 1st line	of maximum VGA size...	of maximum UXGA size...
		8. JPEG Controller (JPG) and the related description		

30. REVISION HISTORY

Revision	Revised on	Description		
		Revised part	Before revision	After revision
0.4 (Continued)	2004/08/05	p.62: JCTL [0x00] Register Bit 15　Bit name	Encode Fast Mode（Figure） High-speed encode mode (Description)	JPEG Encode Fast Mode（Figure） JPEG high-speed encode mode (Description)
		p.62: JCTL [0x00] Register Bit 14　Bit name	None.	JPEG Marker Fast Output Mode（Figure）
		p.63: Table 8.5	=<	≦
		p.65: JCTL [0x04] bits 7, 6, and 5, Description	Same description for bits 7, 6, and 5	Collected to bit [7:5].
		p.78: JLB [0xC0] Register: Default Value	0x0000	0x0000_0000
		p.103-104: 8.5.4 JPEG Codec Function: Revise the entire description.		
		p.103: 8.5.4 JPEG Codec Function: Description: 3rd line	Width: 640 pixels	Width: 1600 pixels
		Same as above	Height: 2048 lines	Height: 2048 pixels
		Same as above	VGA size	UXGA size
		p.103-104: 8.5.4 JPEG Codec Function	Codec circuit Codec core	Standardized to Codec Circuit.
		p.103-104: 8.5.4 JPEG Codec Function	Resize circuit Resize	Standardized to Resize.
		p.103: Inequality symbol at the bottom of this page	Time required until the JPEG codec core ...	VREF inactive time of camera.
		p.103: 2nd line at the bottom of this page and after	...Encoded by the JPEG codec code. (Delete 6 lines.)	The marker output time of the JPEG codec circuit is raised. （Add new 5 lines.)
		p.104: Delete Section "8.5.4.2 Software Reset Processing"		
		p.106: Table 8.21	<=	≦
		p.xxx: Add Section "8.5.5.7 JPEG Module Software Reset".		
		p.xxx: Add Section "8.5.5.8 Marker High Output Mode."		
		10. DMA Controller 2 (DMAC2)		
		p.117: 10.2 Block Diagram	—	Fig. 10.1 Add the figure number and table title.
		11.Ethernet MAC & E-DMA（ETH）		
		p.128: External Pin List Multiplex pins	SPI2_SCLK SPI1_SCLK SPI1_MOSI SPI2_SS SPI2_MOSI SPI1_MISO SPI1_SS SPI2_MISO	Delete all.
0.5	2004/09/01	Correction of writing error: The major corrections are shown below. The others, for example, corrections of character, format, and font errors, are not related to the contents of this manual.		
		p.1,2: 1:1. Overview	Voice	Voice/audio (data)
		p.1: 1.1 Features - Addition of item	—	Support voice/audion data by I2S.
		p.1: 1.1 Last Line of Features	ARM720T Rev4.2	ARM720T Rev4.3
		p.174:Embedded Memory Control Register Bit [5:4]：EMBRAMSEL[1:0]	—	Add and correct the description. Describe the differences of the start address when using as a built-in SRAM.

Revision	Revised On	Description		
		Revised part	Before revision	After revision
0.5 (Continued)	2004/09/01	p.253: 19.1 Overview	Noise data	Voide/audio data
		p.282: CF Card Interface Control Register	—	Add the description about bits 6 and 5.
		p.369: Fig. 28.3 Compact Flash I/F Connection Example	Inverter installed.	Delete the inverter connected to the S1S65010/CFIREQ pin (in ther center of the figure).
		p.379-38: Appendix S1S65010 internal register	—	Correct the description omission or writing error.
1.0	2004/10/05	Correction of writing error: The major corrections are shown below. The others, for example, corrections of character, format, and font errors, are not related to the contents of this manual.		
		p.8,15,17: Pins Correct the invalid cell type of the RESET# pin without pull-up resistance.	Set to ICSU1 Schmidt trigger input with pull-up resistance.	ICS Delete the description at the left.
		p.14,16: Pins	RTS0 CTS0	RTS0# CTS0#
		p.104: JPEG Controller 8.5.4.2 JPEG Codec Register: Restrictions	—	Add 3 lines to the beginning of this section.
		p.216：UART UART[0x08] bit3 Available.	Reserved	DMAMS DMA mode selection bit
		p.227：UART[0x08]bit3 Available	UART[0x08]bit3 Available	Delete from the use restrictions.
		p.271：SPI Fig. 20.4 SPI Clock Setting	—	Add numbers (1) to (4).
		p.273：SPI SPI[0x08]　bit9, 8	—	Change the description. Use the contents of Fig. 20.4 as a description.
1.1	2004/11/04	p.337: Electrical Characteristics: Table 27.1 Current consumption IDDQ (Ta=25℃) Max.	30 (μA)	—
1.2	2004/12/15	Correction of writing error: The major corrections are shown below. The others, for example, corrections of character, format, and font errors, are not related to the contents of this manual.		
		Add before "Table of Contents": Add before "Table of Contents". Product Model No. System		
		Add before "Table of Contents": Notes on use		
		p.27: Fig. 6.1 Correct a register writing error.	SAR0/1 DAR0/1 TCR0/1 CTL0/1 Delete	SAR[3:0] DAR[3:0] TCR[3:0] CTL[3:0] Delete
		p.28: Move the description of the registers in 7.4.1 to 6.4.1.	—	The registers in this chapter and after may be omitted as shown below. R/W, RO, WO, RSV. n/a... （Omitted）
		p.29: Move the description about the registers in Section 7.4.2 to Section 6.4.2.	—	Unless especially specified, ... (omitted) ..., no influence
		p.55: 8.1 Overview 2nd line at the bottom of the description	1/15 sec. or less	1/30 sec.
		p.63: Table 8.5 Title	YUV Data Type	UV Data Type

30. REVISION HISTORY

Revision	Revised On	Description		
		Revised part	Before revision	After revision
1.2 (Continued)	2004/12/15	p.67: JCTL[0x08] bit 14 Register, In the figure	JPEG Codec File Out Status	JPEG Codec File Out Raw Status
		p.75, 76 : JLB[0x80, 0x84] bit 2, In the description	JPEG Line Buffer Interrupt Control Register (Bit 4) ...	to JPEG Line Buffer Interrupt Control Register (Bit 4) ...
		p.76 : JLB[0x88] bits [3:1] Register, In the figure	Raw JPEG …….. Status	JPEG …….. Status
		p.79:.JCODEC[0x00] Bit 2 In the description	Of the view resizer	Delete one sentence.
		p.119: Fig. 10.1 Register: Correct a writing error.	OPSR（Left side）	Change to MISC. Add TECL.
		p.163: Fig. 13.1, p.164: 13.2.2 to 13.2.5 p.335: Table 27.1 p.346: Description of Fig. 27.6 Standard the mode names.	LOW Power State Low Power IDE FULL Operation IDLE mode	Low Speed Mode Low Speed HALT mode High Speed mode High Speed HALT mode
		p.213: UART[0x00] In description of bit	Deviser latch MSB register (UART[0x4C])	Deviser latch MSB register (UART[0x04])
		p.216: UART[0x08] Bit 3 In description of bit	Note: This bit...	Delete the note.
		p.298: 22.6 Change the header.	Set value of 1ms order, 1 μ s order	Timer internal clock setting example (1kHz, 1MHz)
		p.371: 29 External Dimensions Fig. 29.1		Change to the latest version.
1.3	2005/06/08	Correction of writing error: The major corrections are shown below. The others, for example, corrections of character, format, and font errors, are not related to the contents of this manual.		
		Front cover: Change of corporate logo p.1: 1.2 Built-in Functions Add the description about the camera input or JPEG encoder.	EPSON logo	TAGLINE logo
			ARM logo	Relocate. Avoid this information from being written together with the TAGLINE logo.
				The pixel clock for camera data input is less than 2/3 of the CPU clock frequency.
		p.16: Table 4.1 System Configuration Pins: Description	0	Low
			1	High
			Stability time (10ms)	Reserved. (For test)*
		p.26, 29, 31, 34: Channels 0 to 3, DMAC1 Control Register Bits [11:8]: Resource selection	SPIO0 I/O (SPIIRQ)	Reserved.
		p.53: Below Fig. 7.3 Add a description.	— （1st line）	In the normal sampling mode,
			— (5th line and after)	In the high-speed sampling mode, (Added)
		p.93: Fig. 8.5: Correction of register writing error	RSZ[0xC8h]bits[9:0] RSZ[0xCCh]bits[8:0] RSZ[0xD0h]bits[9:0] RSZ[0xD4h]bits[8:0]	RSZ[0xC8h]bits[10:0] RSZ[0xCCh]bits[10:0] RSZ[0xD0h]bits[10:0] RSZ[0xD4h]bits[10:0]
		8.5.4 JPEG Codec Function p.103: 14th line at the bottom	1/15	1/30
		p.104: 1st line	15fps	Deleted.

Revision	Revised On	Description		
		Revised part	**Before revision**	**After revision**
1.3 (Continued)	2005/06/08	p.226, 237: Fig. 16.2 Concept of UART Clock Fig. 17.2 Concept of UART Lite Clock	—	Change: 13. SYSTEM CONTROLLER: UART Clock Divider Register (SYS[0x28]) Change to the same figure.
		p.277: 21.1 Overview of CF	60MHz-6MHz	50MHz-6MHz
		p.336: AC characteristics	（TBD）	Deleted.
		p.337: Table 27.4 CAM Timings t_{CAM5}, t_{CAM6}, t_{CAM7}	t_{CAM5} ： 4	1.6 (3.2)
			t_{CAM6} ： 2	0.8 (1.6)
			t_{CAM7} ： 2	0.8 (1.6)
		*2: Descripton (added)	—	The Min value is used for high-speed sampling; the Min value enclosed in ( ) is used for normal sampling.
		p.349: Table 27.13 (MCS0#)	—	Change waveforms.
		p.362: Table 27.34 (CFDIR)	—	Change waveforms.
		p.362 to 364: Figures 27.34 to 39	CFDIR	CFDDIR
1.4	2008/01/28	p.15: 3.3 Pin Configuration During and After a Reset: List	Value after Reset: described.	Delete the description.
			Value during reset of MA[19.0] Low	Low (Bit 11 only: High)
			Value during reset of MCS[2:0]# High	MCS[2]#:Low MCS[1]#:High MCS[0]#:High
			Value during reset of MWE1# High	Low
			Value during reset of MCLK Low	MCLK(32KHz)
			Value during reset of MCLKEN Low	High
			Value during reset of MRAS# High	Low
		p.37, 43: Camera Status Register Default Value	Default ＝ 0x0034	Default ＝ 0x0004
		p.47: Table 8.1 Register List 0x18-0x1C Reserved Registers Default Value	0x0000	—
		0x90-0xB8 Reserved Registers	0x90-0xB8	0x90-0x9C 0xA8-0xBC
		p.147: Table 13.1 Register List Embedded Memory Control Register Default Value	0x 0000_0000	0x 0000_0010

30. REVISION HISTORY

Revision	Revised On	Description		
		Revised part	Before revision	After revision
1.4 (Continued)	2008/01/28	p.164,175: Table 14.2 Register List SDRAM Status Register Default Value	0x 0000_0202	0x 0000_0002
		p.179: Table 15.2 Register List IRQ or FIQ Raw Status Register	—	*1 The Default Value of the IRQ or FIQ Raw Status Register varies depending on system configuration conditions.
		p.186, 188: Interrupt Identify Register Default value	0x00	0x01
		p.202, 204: Interrupt Identify Register Default Value	0x00	0x01
		p.239, 240:	—	Fig. 20.5 Clock Setting in Slave Mode
		p.249, 251, 252: Table 21.3 Register List CF Card Pin Status Register CF IRQ Source & Clear Register Default Value	0x00XX 0x0000	0x0XXX 0x0XXX
		p.269: Set bit 1 in the RTC Run/Stop control register to "1."	Simultaneously writing data resets only the prescaler counter and 128-1Hz counter.	Simultaneously writing “1” to this bit and “0” to bit 0 in the RTC run/stop control register resets the prescalers and dividing timer.
		p.277, 278: Table 24.1 Register List Watchdog Timer Control Register Default Value	0x0000_A500	0x0000_0000
		P281, 282, 283, 284: GPIOA/B/C/D/E Data Register	—	The default is set to the value corresponding to the GPIOA to GPIOE pin.

31. Appendix 1. S1S65010 Internal Register List

Address (h)	Register Name	Abbreviation	Default Value*1 (h)	R/W	Data Access Size*2 (Bits)
0x FFFE_0000	**APB Bridge Registers**	**APB**			
0x FFFE_0000	APB WAIT0 Register	APBWAIT0	0x 0040_0500	R/W	32
0x FFFE_0004	APB WAIT1 Register	APBWAIT1	0x 0000_0000	R/W	32
0xFFFE_2000	**Ethernet MAC & E-DMA Registers**	**ETH**			
0x FFFE_2000	Interrupt Status Register		0x 0000_0000	RO	32
0x FFFE_2004	Interrupt Enable Register		0x 0000_0000	R/W	32
0x FFFE_2008	Reset Register		0x 0000_2000	R/W	32
0x FFFE_200C	PHY Status Register		0x 0000_0000	RO	32
0x FFFE_2010	DMA Command Register		0x 0000_0000	R/W	32
0x FFFE_2018	TX DMA Pointer Register		0x 0000_0000	R/W	32
0x FFFE_201C	RX DMA Pointer Register		0x 0000_0000	R/W	32
0x FFFE_2020	Mode Register		0x 4000_0000	R/W	32
0x FFFE_2024	TX Mode Register		0x 0000_0000	R/W	32
0x FFFE_2028	RX Mode Register		0x 0000_0000	R/W	32
0x FFFE_202C	MIIM Register		0x 0000_0000	R/W	32
0x FFFE_2030	MAC Address Register 1: Lower 32 bits		0x 0000_0000	R/W	32
0x FFFE_2034	MAC Address Register 1: Upper 16 bits		0x 0000_0000	R/W	32
0x FFFE_2038	MAC Address Register 2: Lower 32 bits		0x 0000_0000	R/W	32
0x FFFE_203C	MAC Address Register 2: Upper 16 bits		0x 0000_0000	R/W	32
0x FFFE_2040	MAC Address Register 3: Lower 32 bits		0x 0000_0000	R/W	32
0x FFFE_2044	MAC Address Register 3: Upper 16 bits		0x 0000_0000	R/W	32
0x FFFE_2048	MAC Address Register 4: Lower 32 bits		0x 0000_0000	R/W	32
0x FFFE_204C	MAC Address Register 4: Upper 16 bits		0x 0000_0000	R/W	32
0x FFFE_2050	MAC Address Register 5: Lower 32 bits		0x 0000_0000	R/W	32
0x FFFE_2054	MAC Address Register 5: Upper 16 bits		0x 0000_0000	R/W	32
0x FFFE_2058	MAC Address Register 6: Lower 32 bits		0x 0000_0000	R/W	32
0x FFFE_205C	MAC Address Register 6: Upper 16 bits		0x 0000_0000	R/W	32
0x FFFE_2060	MAC Address Register 7: Lower 32 bits		0x 0000_0000	R/W	32
0x FFFE_2064	MAC Address Register 7: Upper 16 bits		0x 0000_0000	R/W	32
0x FFFE_2068	MAC Address Register 8: Lower 32 bits		0x 0000_0000	R/W	32
0x FFFE_206C	MAC Address Register 8: Upper 16 bits		0x 0000_0000	R/W	32
0x FFFE_2070	Flow Control Register		0x 0000_0000	R/W	32
0x FFFE_2074	Pause Request Register		0x 0000_0000	R/W	32
0x FFFE_2078	Pause Frame Data Register 1		0x 0000_0000	R/W	32
0x FFFE_207C	Pause Frame Data Register 2		0x 0000_0000	R/W	32
0x FFFE_2080	Pause Frame Data Register 3		0x 0000_0000	R/W	32
0x FFFE_2084	Pause Frame Data Register 4		0x 0000_0000	R/W	32
0x FFFE_2088	Pause Frame Data Register 5		0x 0000_0000	R/W	32
0x FFFE_2090	Buffer Management Enable Register		0x 0000_0000	R/W	32
0x FFFE_2094	Buffer Free Register		0x 0000_0000	R/W	32
0x FFFE_2098	Buffer Information Register		0x 03FF_03FF	R/W	32
0x FFFE_209C	Pause Information Register		0x 0000_0000	R/W	32
0x FFFE_20A0 to 0x FFFE_20AC	Reserved		—	—	—
0x FFFE_20F0	TX FIFO Status Register		0x 4000_0000	RO	32
0x FFFE_20F4	RX FIFO Status Register		0x 4000_0000	RO	32
0x FFFE_20F8 to 0x FFFE_20FC	Reserved		—	—	—
0x FFFE_3000	**DMA Controller 1 Registers**	**DMAC1**			
0x FFFE_3000	DMA Channel 0 Source Address Register	SAR0	0x XXXX_XXXX	R/W	32
0x FFFE_3004	DMA Channel 0 Destination Address Register	DAR0	0x XXXX_XXXX	R/W	32
0x FFFE_3008	DMA Channel 0 Transfer Count Register	TCR0	0x 00XX_XXXX	R/W	32
0x FFFE_300C	DMA Channel 0 Control Register	CTL0	0x 0000_0000	R/W	32

Address (h)	Register Name	Abbreviation	Default Value*1 (h)	R/W	Data Access Size*2 (Bits)
0x FFFE_3010	DMA Channel 1 Source Address Register	SAR1	0x XXXX_XXXX	R/W	32
0x FFFE_3014	DMA Channel 1 Destination Address Register	DAR1	0x XXXX_XXXX	R/W	32
0x FFFE_3018	DMA Channel 1 Transfer Count Register	TCR1	0x 00XX_XXXX	R/W	32
0x FFFE_301C	DMA Channel 1 Control Register	CTL1	0x 0000_0000	R/W	32
0x FFFE_3020	DMA Channel 2 Source Address Register	SAR2	0x XXXX_XXXX	R/W	32
0x FFFE_3024	DMA Channel 2 Destination Address Register	DAR2	0x XXXX_XXXX	R/W	32
0x FFFE_3028	DMA Channel 2 Transfer Count Register	TCR2	0x 00XX_XXXX	R/W	32
0x FFFE_302C	DMA channel 2 Control Register	CTL2	0x 0000_0000	R/W	32
0x FFFE_3030	DMA Channel 3 Source Address Register	SAR3	0x XXXX_XXXX	R/W	32
0x FFFE_3034	DMA channel 3 Destination Address Register	DAR3	0x XXXX_XXXX	R/W	32
0x FFFE_3038	DMA Channel 3 Transfer Count Register	TCR3	0x 00XX_XXXX	R/W	32
0x FFFE_303C	DMA Channel 3 Control Register	CTL3	0x 0000_0000	R/W	32
0x FFFE_3060	DMA Channel Operating Select Register	OPSR	0x 0000_0000	R/W	32
0x FFFE_6000	**Compact Flash Interface Registers**	**CF**			
0x FFFE_6000	CF Card Interface Control Register	CFCTL	0x 1000	(R/W)	16
0x FFFE_6004	CF Card Pin Status Register	CFPINSTS	0x 0XXX	RO	16
0x FFFE_6008	CF Card IRQ Source & Clear Register	CFINTRSTS	0x 0XXX	R/W	16
0x FFFE_600C	CF Card IRQ Enable Register	CFINTMSTS	0x 0000	R/W	16
0x FFFE_6010	CF Card IRQ Status Register	CFINTSTS	0x 0000	RO	16
0x FFFE_6014	CF Card MISC Register	CFMISC	0x 0000	R/W	16
0x FFFE_8000	**Camera Interface Registers**	**CAM**			
0x FFFE_8000	Camera Clock Frequency Setting Register		0x 0000	R/W	16
0x FFFE_8004	Camera Signal Setting Register		0x 0000	R/W	16
0x FFFE_8008 to 0x FFFE_801C	Reserved		—	—	—
0x FFFE_8020	Camera Mode Setting Register		0x 0000	R/W	16
0x FFFE_8024	Camera Frame Control Register		0x 0000	R/W	16
0x FFFE_8028	Camera Control Register		0x 0000	WO	16
0x FFFE_802C	Camera Status Register		0x 0004	RO	16
0x FFFE_8030 to 0x FFFE_805C	Reserved		—	—	—
0x FFFE_9000	**Resizer Operation Registers**	**RSZ**			
0x FFFE_9060	Global Resizer Control Register		0x 0000	WO	16
0x FFFE_9064	Capture Control State Register		0x 0000	RO	16
0x FFFE_9068	Capture Data Setting Register		0x 0000	R/W	16
0x FFFE_9070 to 0Xx FFFE_907C	Reserved Registers		0x 0000	R/W	16
0x FFFE_90C0	Capture Resizer Control Register		0x 0000	R/W	16
0x FFFE_90C8	Capture Resizer Start X Position Register		0x 0000	R/W	16
0x FFFE_90CC	Capture Resizer Start Y Position Register		0x 0000	R/W	16
0x FFFE_90D0	Capture Resizer End X Position Register		0x 027F	R/W	16
0x FFFE_90D4	Capture Resizer End Y Position Register		0x 01DF	R/W	16
0x FFFE_90D8	Capture Resizer Scaling Rate Register		0x 8080	R/W	16
0x FFFE_90DC	Capture Resizer Scaling Mode Register		0x 0000	R/W	16

Address (h)	Register Name	Abbreviation	Default Value*1 (h)	R/W	Data Access Size*2 (Bits)
0x FFFE_A000	**JPEG Module Registers**	**JCTL**			
0x FFFE_A000	JPEG Control Register		0x 0000	R/W	16
0x FFFE_A004	JPEG Status Flag Register		0x 8080	R/W	16
0x FFFE_A008	JPEG Raw Status Flag Register		0x 8080	RO	16
0x FFFE_A00C	JPEG Interrupt Control Register		0x 0000	R/W	16
0x FFFE_A010	Reserved Register		0x 0080	RO	16
0x FFFE_A014	JPEG Codec Start/Stop Control Register		0x 0000	WO	16
0x FFFE_A018 to 0x FFFE_A01C	Reserved Registers		—	—	16
0x FFFE_A020	Huffman Table Automatic Setting Register		0x 0000	R/W	16
0x FFFE_A040	**JPEG FIFO Setting Registers**	**JFIFO**			
0x FFFE_A040	JPEG FIFO Control Register		0x 0000	R/W	16
0x FFFE_A044	JPEG FIFO Status Register		0x 8001	RO	16
0x FFFE_A048	JPEG FIFO Size Register		0x 003F	R/W	16
0x FFFE_A04C	JPEG FIFO Read/Write Port Register		0x 0000_0000	R/W	32
0x FFFE_A050 to 0x FFFE_A058	Reserved Registers		—	—	16
0x FFFE_A060	Encode Size Limit Register 0		0x 0000	R/W	16
0x FFFE_A064	Encode Size Limit Register 1		0x 0000	R/W	16
0x FFFE_A068	Encode Size Result Register 0		0x 0000	RO	16
0x FFFE_A06C	Encode Size Result Register 1		0x 0000	RO	16
0x FFFE_A070 to 0x FFFE_A078	Reserved Registers		—	—	16
0x FFFE_A080	**JPEG Line Buffer Setting Registers**	**JLB**			
0x FFFE_A080	JPEG Line Buffer Status Flag Register		0x 0000	R/W	16
0x FFFE_A084	JPEG Line Buffer Raw Status Flag Register		0x 0000	RO	16
0x FFFE_A088	JPEG Line Buffer Current Status Flag Register		0x 0009	RO	16
0x FFFE_A08C	JPEG Line Buffer Interrupt Control Register		0x 0000	R/W	16
0x FFFE_A090 to 0x FFFE_A0B8	Reserved Registers		—	—	16
0x FFFE_A0A0	JPEG Line Buffer Horizontal Pixel Support Size Register		0x 2800	R/W	16
0x FFFE_A0A4	JPEG Line Buffer Memory Address Offset Register		0x 0030	R/W	16
0x FFFE_A0A8 to 0x FFFE_A0BC	Reserved Registers		—	—	16
0x FFFE_A0C0	JPEG Line Buffer Read/Write Port Register		0x 0000	R/W	16
0x FFFE_B000	**JPEG Codec Registers**	**JCOCEC**			
0x FFFE_B000	Operation Mode Setting Register		0x 0000	R/W	16
0x FFFE_B004	Command Setting Register		Not applicable	WO	16
0x FFFE_B008	JPEG Operation Status Register		0x 0000	RO	16
0x FFFE_B00C	Quantization Table Number Register		0x 0000	R/W	16
0x FFFE_B010	Huffman Table Number Register		0x 0000	R/W	16
0x FFFE_B014	DRI Setting Register 0		0x 0000	R/W	16
0x FFFE_B018	DRI Setting Register 1		0x 0000	R/W	16
0x FFFE_B01C	Vertical Pixel Size Register 0		0x 0000	R/W	16
0x FFFE_B020	Vertical Pixel Size Register 1		0x 0000	R/W	16
0x FFFE_B024	Horizontal Pixel Size Register 0		0x 0000	R/W	16
0x FFFE_B028	Horizontal Pixel Size Register 1		0x 0000	R/W	16
0x FFFE_B02C to 0x FFFE_B034	Reserved Registers		—	—	16
0x FFFE_B038	RST Marker Operation Setting Register		0x 0000	R/W	16
0x FFFE_B03C	RST Marker Operation Status Register		0x 0000	RO	16

Address (h)	Register Name	Abbreviation	Default Value*1 (h)	R/W	Data Access Size*2 (Bits)
0x FFFE_B040 to 0x FFFE_B0CC	Insertion Marker Data Registers		0x 00FF	R/W	16
0x FFFE_B400 to 0x FFFE_B4FC	Quantization Table No. 0 Register		Not applicable	R/W	16
0x FFFE_B500 to 0x FFFE_B5FC	Quantization Table No. 1 Register		Not applicable	R/W	16
0x FFFE_B800 to 0x FFFE_B83C	DC Huffman Table No. 0 Register 0		Not applicable	WO	16
0x FFFE_B840 to 0x FFFE_B86C	DC Huffman Table No. 0 Register 1		Not applicable	WO	16
0x FFFE_B880 to 0x FFFE_B8BC	AC Huffman Table No. 0 Register 0		Not applicable	WO	16
0x FFFE_B8C0 to 0x FFFE_BB44	AC Huffman Table No. 0 Register 1		Not applicable	WO	16
0x FFFE_BC00 to 0x FFFE_BC3C	DC Huffman Table No. 1 Register 0		Not applicable	WO	16
0x FFFE_BC40 to 0x FFFE_BC6C	DC Huffman Table No. 1 Register 1		Not applicable	WO	16
0x FFFE_BC80 to 0x FFFE_BCBC	AC Huffman Table No. 1 Register 0		Not applicable	WO	16
0x FFFE_BCC0 to 0x FFFE_BF44	AC Huffman Table No. 1 Register 1		Not applicable	WO	16
0x FFFE_C000	**JPEG DMAC Registers**	**JDMA**			
0x FFFE_C000	JPEG DMA Source Address Register	JSAR	0x XXXX_XXXX	R/W	32
0x FFFE_C004	JPEG DMA Destination Address Register	JDAR	0x XXXX_XXXX	R/W	32
0x FFFE_C008	JPEG DMA Transfer Count Register	JTCR	0x 0000_0000	R/W	32
0x FFFE_C00C	JPEG DMA Control Register	JCTL	0x 0000_0000	R/W	32
0x FFFE_C010	JPEG DMA Block Count Register	JBCR	0x 00XX_XXXX	R/W	32
0x FFFE_C014	JPEG DMA Destination Offset Address Register	JOFR	0x 0000_0000	R/W	32
0x FFFE_C018	JPEG DMA Block End Count Register	JBER	0x 00XX_XXXX	R/W	32
0x FFFE_C020	JPEG DMA Expansion Register	JHID	0x 0000_0000	R/W	32
0x FFFE_C040	JPEG DMA FIFO Data Select Mode Register	JFSM	0x 0000_0000	R/W	32
0x FFFE_D000	**I^{2}C Registers**	**I2C**			
0x FFFE_D000	I^{2}C Transmit Data Register		0000 0000 b	R/W	8 (16/32)
0x FFFE_D004	I^{2}C Receive Data Register		0000 0000 b	RO	8 (16/32)
0x FFFE_D008	I^{2}C Control Register		0000 0000 b	R/W	8 (16/32)
0x FFFE_D00C	I^{2}C Bus Status Register		00xx 0000 b	RO	8 (16/32)
0x FFFE_D010	I^{2}C Error Status Register		0000 0000 b	RO	8 (16/32)
0x FFFE_D014	I^{2}C Interrupt Control/Status Register		0000 0000 b	R/W	8 (16/32)
0x FFFE_D018	I^{2}C-Bus Sample Clock Frequency Divisor Register		0000 0000 b	R/W	8 (16/32)
0x FFFE_D01C	I^{2}C SCL Clock Frequency Divisor Register		0000 0000 b	R/W	8 (16/32)
0x FFFE_D020	I^{2}C I/O Control Register		0001 0001 b	R/W	8 (16/32)
0x FFFE_D024	I^{2}C DMA Mode Register		0000 0000 b	R/W	8 (16/32)
0x FFFE_D028	I^{2}C DMA Count Value (LSB) Register		0000 0000 b	R/W	8 (16/32)
0x FFFE_D02C	I^{2}C DMA Count Value (MSB) Register		0000 0000 b	R/W	8 (16/32)
0x FFFE_D030	I^{2}C DMA Status Register		0000 1000 b	RO	8 (16/32)
0x FFFE_D034 to 0x FFFE_D038	Reserved		—	—	—
0x FFFE_E000	**I2S Registers**	**I2S**			
0x FFFE_E000	I2S0 Control Register		0x 0000	R/W	16/32
0x FFFE_E004	I2S0 Clock Frequency Divisor Register		0x 0000	R/W	16/32
0x FFFE_E008	I2S0 Transfer Port Register		—	R/W	8/16/32
0x FFFE_E010	I2S0 Interrupt Status Register		0x 0000	R/W	16/32
0x FFFE_E014	I2S0 Interrupt Raw Status Register		0x 0009	RO	16/32
0x FFFE_E018	I2S0 Interrupt Enable Register		0x 0000	R/W	16/32

Address (h)	Register Name		Abbreviation	Default Value*1 (h)	R/W	Data Access Size*2 (Bits)
0x FFFE_E01C	I2S0 Current Status Register			0x 0009	RO	16/32
0x FFFE_E040	I2S1 Control Register			0x 0000	R/W	16/32
0x FFFE_E044	I2S1 Clock Frequency Divisor Register			0x 0000	R/W	16/32
0x FFFE_E048	I2S1 Transfer Port Register			—	R/W	8/16/32
0x FFFE_E050	I2S1 Interrupt Status Register			0x 0000	R/W	16/32
0x FFFE_E054	I2S1 Interrupt Raw Status Register			0x 0009	RO	16/32
0x FFFE_E058	I2S1 Interrupt Enable Register			0x 0000	R/W	16/32
0x FFFE_E05C	I2S1 Current Status Register			0x 0009	RO	16/32
0x FFFF_1000	**General Purpose I/O Registers**		**GPIO**			
0x FFFF_1000	GPIOA Data Register		GPIOA_DATA	0x 0000_0000	R/W	8 (/16/32)
0x FFFF_1004	GPIOA Pin Function Register		GPIOA_FNC	0x 0000_0000	R/W	16 (/32)
0x FFFF_1008	GPIOB Data Register		GPIOB_DATA	0x 0000_0000	R/W	8 (/16/32)
0x FFFF_100C	GPIOB Pin Function Register		GPIOB_FNC	0x 0000_0000	R/W	16 (/32)
0x FFFF_1010	GPIOC Data Register		GPIOC_DATA	0x 0000_0000	R/W	8 (/16/32)
0x FFFF_1014	GPIOC Pin Function Register		GPIOC_FNC	0x 0000_0000	R/W	16 (/32)
0x FFFF_1018	GPIOD Data Register		GPIOD_DATA	0x 0000_0000	R/W	8 (/16/32)
0x FFFF_101C	GPIOD Pin Function Register		GPIOD_FNC	0x 0000_0000	R/W	16 (/32)
0x FFFF_1020	GPIOE Data Register		GPIOE_DATA	0x 0000_0000	R/W	8 (/16/32)
0x FFFF_1024	GPIOE Pin Function Register		GPIOE_FNC	0x 0000_0000	R/W	16 (/32)
0x FFFF_1028	GPIOF Data Register		GPIOF_DATA	0x 0000_0000	R/W	8 (/16/32)
0x FFFF_102C	GPIOF Pin Function Register		GPIOF_FNC	0x 0000_5555	R/W	16 (/32)
0x FFFF_1030	GPIOG Data Register		GPIOG_DATA	0x 0000_0000	R/W	8 (/16/32)
0x FFFF_1034	GPIOG Pin Function Register		GPIOG_FNC	0x 0000_5555	R/W	16 (/32)
0x FFFF_1038	GPIOH Data Register		GPIOH_DATA	0x 0000_0000	R/W	8 (/16/32)
0x FFFF_103C	GPIOH Pin Function Register		GPIOH_FNC	0x 0000_0001	R/W	16 (/32)
0x FFFF_1040	GPIOA&B IRQ Type Register		GPIOAB_ITYP	0x 0000_0000	R/W	16 (/32)
0x FFFF_1044	GPIOA&B IRQ Polarity Register		GPIOAB_IPOL	0x 0000_0000	R/W	16 (/32)
0x FFFF_1048	GPIOA&B IRQ Enable Register		GPIOAB_IEN	0x 0000_0000	R/W	16 (/32)
0x FFFF_104C	GPIOA&B IRQ Status & Clear Register		GPIOAB_ISTS	0x 0000_0000	R/W	16 (/32)
0x FFFF_2000	**SPI Registers**		**SPI**			
0x FFFF_2000	SPI Receive Data Register			0x 0000_0000	RO	32
0x FFFF_2004	SPI Transmit Data Register			0x 0000_0000	R/W	32
0x FFFF_2008	SPI Control Register 1			0x 0000_0000	R/W	32
0x FFFF_200C	SPI Control Register 2			0x 0000_0000	R/W	32
0x FFFF_2010	SPI Wait Register			0x 0000_0000	R/W	32
0x FFFF_2014	SPI Status Register			0x 0000_0010	RO	32
0x FFFF_2018	SPI Interrupt Control Register			0x 0000_0000	R/W	32
0x FFFF_5000	DLAB	**UART Registers**	**UART**			
0x FFFF_5000	0	Receive Buffer Register	RBR	0x 00	RO	8 (/16/32)
0x FFFF_5000	0	Transmit Holding Register	THR	—	WO	8 (/16/32)
0x FFFF_5000	1	Divisor Latch LSB Register	DLL	0x 00	R/W	8 (/16/32)
0x FFFF_5004	0	Interrupt Enable Register	IER	0x 00	R/W	8 (/16/32)
0x FFFF_5004	1	Divisor Latch MSB Register	DLM	0x 00	R/W	8 (/16/32)
0x FFFF_5008	Interrupt Identify Register		IIR	0x 01	RO	8 (/16/32)
0x FFFF_5008	FIFO Control Register		FCR	—	WO	8 (/16/32)
0x FFFF_500C	Line Control Register		LCR	0x 00	R/W	8 (/16/32)
0x FFFF_5010	Modem Control Register		MCR	0x 00	R/W	8 (/16/32)
0x FFFF_5014	Line Status Register		LSR	0x 00	RO	8 (/16/32)
0x FFFF_5018	Modem Status Register		MSR	0x 00	RO	8 (/16/32)
0x FFFF_501C	Scratch Register		SCR	0x 00	R/W	8 (/16/32)
0x FFFF_5020	Test 0 Register		T0	0x 00	R/W	8 (/16/32)
0x FFFF_5024	Test 1 Register		T1	0x 00	R/W	8 (/16/32)
0x FFFF_5028	Test Status 0 Register		TS0	—	RO	8 (/16/32)
0x FFFF_502C	Test Status 1 Register		TS1	0x 00	RO	8 (/16/32)
0x FFFF_5030	Test Status 2 Register		TS2	0x 00	RO	8 (/16/32)
0x FFFF_503C	Test Status 3 Register		TS3	0x 00	RO	8 (/16/32)
0x FFFF_6000	**UART Lite Registers**		**UARTL**			
0x FFFF_6000	0	Receive Buffer Register	RBR	0x 00	RO	8 (/16/32)
0x FFFF_6000	0	Transmit Holding Register	THR	—	WO	8 (/16/32)

31. Appendix 1. S1S65010 Internal Register List

Address (h)	Register Name		Abbreviation	Default Value*1 (h)	R/W	Data Access Size*2 (Bits)
0x FFFF_6000	1	Divisor Latch LSB Register	DLL	0x 00	R/W	8 (/16/32)
0x FFFF_6004	0	Interrupt Enable Register	IER	0x 00	R/W	8 (/16/32)
0x FFFF_6004	1	Divisor Latch MSB Register	DLM	0x 00	R/W	8 (/16/32)
0x FFFF_6008	Interrupt Identify Register		IIR	0x 01	RO	8 (/16/32)
0x FFFF_600C	Line Control Register		LCR	0x 03	R/W	8 (/16/32)
0x FFFF_6014	Line Status Register		LSR	0x 00	RO	8 (/16/32)
0x FFFF_6030	Test Status 2 Register		TS2	0x 00	RO	8 (/16/32)
0x FFFF_603C	Test Status 3 Register		TS3	0x 00	RO	8 (/16/32)
0x FFFF_8000	**RTC Registers**		**RTC**			
0x FFFF_8000	RTC Run/Stop Control Register			x--- ---x b	(R/W)	8
0x FFFF_8004	RTC Interrupt Register			1110 0000 b	R/W	8
0x FFFF_8008	RTC Timer Divider Register			xxxx xxxx b	R/(W)	8
0x FFFF_800C	RTC Second Counter Register			--xx xxxx b	R/W	8
0x FFFF_8010	RTC Minute Counter Register			--xx xxxx b	R/W	8
0x FFFF_8014	RTC Hour Counter Register			---x xxxx b	R/W	8
0x FFFF_8018	RTC Day Counter Register			0x XXXX	R/W	16
0x FFFF_8020	RTC Alarm Minute Compare Register			--xx xxxx b	R/W	8
0x FFFF_8024	RTC Alarm Hour Compare Register			---x xxxx b	R/W	8
0x FFFF_8028	RTC Alarm Day Compare Register			x xxxx xxxx b	R/W	16
0x FFFF_802C	RTC Test Register			---0 0000 b	R/W	8
0x FFFF_8030	RTC Prescaler Register			-xxx xxxx b	R/(W)	8
0x FFFF_8034	RTC Test Clock Register			---- ---- b	WO	8
0x FFFF_9000	**DMA controller 2 Registers**		**DMAC2**			
0x FFFF_9000	DMA Channel 0 Source Address Register		SAR0	0x XXXX_XXXX	R/W	32
0x FFFF_9004	DMA Channel 0 Destination Address Register		DAR0	0x XXXX_XXXX	R/W	32
0x FFFF_9008	DMA Channel 0 Transfer Count Register		TCR0	0x 00XX_XXXX	R/W	32
0x FFFF_900C	DMA Channel 0 Control Register		CTL0	0x 0000_0000	R/W	32
0x FFFF_9010	DMA Channel 1 Source Address Register		SAR1	0x XXXX_XXXX	R/W	32
0x FFFF_9014	DMA Channel 1 Destination Address Register		DAR1	0x XXXX_XXXX	R/W	32
0x FFFF_9018	DMA Channel 1 Transfer Count Register		TCR1	0x 00XX_XXXX	R/W	32
0x FFFF_901C	DMA Channel 1 Control Register		CTL1	0x 0000_0000	R/W	32
0x FFFF_9060	DMA Channel Operating Select Register		OPSR	0x 0000_0000	R/W	32
0x FFFF_9064	DMA Channel MISC Register		MISC	0x 0000_0000	R/W	32
0x FFFF_9070	DMA Channel Transfer Complete Control Register		TECL	0x 0000_0000	R/W	32
0x FFFF_A000	**Memory Controller Registers**		**MEMC**			
0x FFFF_A000	Configuration Register for Device 0		CFG0	0x 1F00_0041	R/W	32
0x FFFF_A004	Configuration Register for Device 1		CFG1	0x 7F7F_0040	R/W	32
0x FFFF_A008	Configuration Register for Device 2		CFG2	0x 7F7F_0040	R/W	32
0x FFFF_A00C	Reserved* (Configuration Register for Device 3)		CFG3	0x 7F7F_0040	R/W	32
0x FFFF_A020	Timing Register for Device 0		RAMTMG0	0x 0000_1C70	R/W	32
0x FFFF_A024	Control Register for Device 0		RAMCNTL0	0x 0000_0001	R/W	32
0x FFFF_A030	Timing Register for Device 1		RAMTMG1	0x 0000_1C70	R/W	32
0x FFFF_A034	Control Register for Device 1		RAMCNTL1	0x 0000_0001	R/W	32
0x FFFF_A040	Timing Register for Device 2		RAMTMG2	0x 0000_1C70	R/W	32
0x FFFF_A044	Control Register for Device 2		RAMCNTL2	0x 0000_0001	R/W	32
0x FFFF_A050	Reserved* (Timing Register for Device 3)		RAMTMG3	0x 0000_1C70	R/W	32

Address (h)	Register Name	Abbreviation	Default Value*1 (h)	R/W	Data Access Size*2 (Bits)
0x FFFF_A054	Reserved* (Control Register for Device 3)	RAMCNTL3	0x 0000_0001	R/W	32
0x FFFF_A060	Mode Register for SDRAM	SDMR	0x 0000_0032	R/W	16/32
0x FFFF_A064	Reserved	—	—	—/—	—
0x FFFF_A068	Reserved	—	—	—/—	—
0x FFFF_A070	Configuration Register for SDRAM	SDCNFG	0x 0600_C700	R/W	32
0x FFFF_A074	Advanced Configuration Register for SDRAM	SDADVCNFG	0x 000F_0300	R/W	32
0x FFFF_A080	Initialization Control Register	SDINIT	0x 0000_0000	R/W	16/32
0x FFFF_A090	Refresh Timer Register for SDRAM	SDREF	0x 0000_00A0	R/W	16/32
0x FFFF_A0A0	Status Register for SDRAM	SDSTAT	0x 0000_0002	RO	32
0x FFFF_B000	**Timers Registers**	**TIM**			
0x FFFF_B000	Timer 0 Load Register	TM0LD	0x 0000	R/W	16 (/32)
0x FFFF_B004	Timer 0 Count Register	TM0CNT	0x 0000	RO	16 (/32)
0x FFFF_B008	Timer 0 Control Register	TM0CTRL	0x 0000	(R/W)	16 (/32)
0x FFFF_B00C	Timer 0 IRQ Flag Clear Register	TM0IRQ	—	WO	8 (/16/32)
0x FFFF_B010	Timer 0 Port Output Control Register	TM0POUT	0x 00	(R/W)	8 (/16/32)
0x FFFF_B020	Timer 1 Load Register	TM1LD	0x 0000	R/W	16 (/32)
0x FFFF_B024	Timer 1 Count Register	TM1CNT	0x 0000	RO	16 (/32)
0x FFFF_B028	Timer 1 Control Register	TM1CTRL	0x 0000	(R/W)	16 (/32)
0x FFFF_B02C	Timer 1 IRQ Flag Clear Register	TM1IRQ	—	WO	8 (/16/32)
0x FFFF_B030	Timer 1 Port Output Control Register	TM1POUT	0x 00	(R/W)	8 (/16/32)
0x FFFF_B040	Timer 2 Load Register	TM2LD	0x 0000	R/W	16 (/32)
0x FFFF_B044	Timer 2 Count Register	TM2CNT	0x 0000	RO	16 (/32)
0x FFFF_B048	Timer 2 Control Register	TM2CTRL	0x 0000	(R/W)	16 (/32)
0x FFFF_B04C	Timer 2 IRQ Flag Clear Register	TM2IRQ	—	WO	8 (/16/32)
0x FFFF_B050	Timer 2 Port Output Control Register	TM2POUT	0x 00	(R/W)	8 (/16/32)
0x FFFF_B060 to 0x FFFF_B09C	Reserved	—	—	—	—
0x FFFF_B0A0	Prescaler 0 Control Register	PS0CTRL	0x 0000	(R/W)	16 (/32)
0x FFFF_B0A4	Prescaler 1 Control Register	PS1CTRL	0x 0000	(R/W)	16 (/32)
0x FFFF_B0B0	Timer IRQ Status Register	TMIRQSTS	0x 00	RO	8 (/16/32)
0x FFFF_C000	**Watchdog Timer Registers**	**WDT**			
0x FFFF_C000	Watchdog Timer Load Register		0x 0000_FFFF	R/W	16 (/32)
0x FFFF_C004	Watchdog Timer Count Register		0x 0000_FFFF	RO	16 (/32)
0x FFFF_C008	Watchdog Timer Control Register		0x 0000_0000	R/W	16 (/32)
0x FFFF_D000	**System Controller Registers**	**SYS**			
0x FFFF_D000	Chip ID Register	CHIPID	0x 0650_100X	RO	32
0x FFFF_D004	Chip Configuration Register	CHIPCFG	0x 0000_XXXX	RO	16 (/32)
0x FFFF_D008	PLL Setting Register 1	PLLSET1	0x 0421_D46A	R/W	32
0x FFFF_D00C	PLL Setting Register 2	PLLSET2	0x 0000_0000	(R/W)	16 (/32)
0x FFFF_D010	HALT Mode Clock Control Register	HALTMODE	0x 0000_0000	R/W	16 (/32)
0x FFFF_D014	I/O Clock Control Register	IOCLKCTL	0x 0000_0000	R/W	16 (/32)
0x FFFF_D018	Clock Select Register	CLK32SEL	0x 0000_0000	R/W	16 (/32)
0x FFFF_D01C	HALT Control Register	HALTCTL	—	WO	16 (/32)
0x FFFF_D020	Memory Remap Register	REMAP	0x 0000_0000	R/W	16 (/32)
0x FFFF_D024	Software Reset Register	SOFTRST	—	WO	32
0x FFFF_D028	UART Clock Divider Register	UARTDIV	0x 0000_0000	R/W	16 (/32)
0x FFFF_D02C	MD Bus Pull-down Control Register	MDPLDCTL	0x 0000_0000	R/W	16 (/32)
0x FFFF_D030	GPIOC Resistor Control Register	PORTCRCTL	0x 0000_0000	R/W	16 (/32)
0x FFFF_D034	GPIOD Resistor Control Register	PORTDRCTL	0x 0000_0000	R/W	16 (/32)
0x FFFF_D038	GPIOE Resistor Control Register	PORTERCTL	0x 0000_0000	R/W	16 (/32)
0x FFFF_D03C	Internal TEST Mode Register	ITESTM	0x 0000_0000	—/—	—
0x FFFF_D040	Embedded Memory Control Register	EMBMEMCTL	0x 0000_0010	R/W	16 (/32)
0x FFFF_F000	**Interrupt controller Registers**	**INT**			
0x FFFF_F000	IRQ Status Register		0x 0000_0000	RO	32
0x FFFF_F004	IRQ Raw Status Register		0x 0000_0000	RO	32
0x FFFF_F008	IRQ Enable Register		0x 0000_0000	R/W	32

Address (h)	Register Name	Abbreviation	Default Value*1 (h)	R/W	Data Access Size*2 (Bits)
0x FFFF_F00C	IRQ Enable Clear Register		0x 0000_0000	WO	32
0x FFFF_F010	Software IRQ Register		0x 0000_0000	WO	32
0x FFFF_F080	IRQ Level Register		0x 0000_0000	R/W	32
0x FFFF_F084	IRQ Polarity Register		0x FFFF_FFFF	R/W	32
0x FFFF_F088	IRQ Trigger Reset Register		0x 0000_0000	WO	32
0x FFFF_F100	FIQ Status Register		0x 0000_0000	RO	32
0x FFFF_F104	FIQ Raw Status Register		0x 0000_0000	RO	32
0x FFFF_F108	FIQ Enable Register		0x 0000_0000	R/W	32
0x FFFF_F10C	FIQ Enable Clear Register		0x 0000_0000	WO	32
0x FFFF_F180	FIQ Level Register		0x 0000_0000	R/W	32
0x FFFF_F184	FIQ Polarity Register		0x 0000_0003	R/W	32
0x FFFF_F188	FIQ Trigger Reset Register		0x 0000_0000	WO	32

Notes *1: These values are normally in hexadecimal, but some have the suffix b, indicating binary. “X” indicates an undefined hexadecimal digit; “x,” an undefined binary one.

*2: The notations “8 (/16/32)” and “16 (/32)” indicate a minimum access size (8 or 16). All such registers support 16- and 32-bit access. Only the bottom 8 bits are valid for 16- and 32-bit reads.

EPSON

International Sales Operations

AMERICA

EPSON ELECTRONICS AMERICA, INC.
HEADQUARTERS
2580 Orchard Parkway
San Jose , CA 95131,USA
Phone: +1-800-228-3964 FAX: +1-408-922-0238

SALES OFFICES
Northeast
301 Edgewater Place, Suite 210
Wakefield, MA 01880, U.S.A.
Phone: +1-800-922-7667 FAX: +1-781-246-5443

EUROPE

EPSON EUROPE ELECTRONICS GmbH
HEADQUARTERS
Riesstrasse 15 Muenchen Bayern,
80992 GERMANY
Phone: +49-89-14005-0 FAX: +49-89-14005-110

ASIA

EPSON (CHINA) CO., LTD.
7F, Jinbao Bldg.,No.89 Jinbao St.,
Dongcheng District,
Beijing 100005, China
Phone: +86-10-6410-6655 FAX: +86-10-6410-7320

SHANGHAI BRANCH
7F, Block B, Hi-Tech Bldg., 900, Yishan Road,
Shanghai 200233, CHINA
Phone: +86-21-5423-5522 FAX: +86-21-5423-5512

EPSON HONG KONG LTD.
20/F., Harbour Centre, 25 Harbour Road
Wanchai, Hong Kong
Phone: +852-2585-4600 FAX: +852-2827-4346
Telex: 65542 EPSCO HX

EPSON (CHINA) CO., LTD.
SHENZHEN BRANCH
12/F, Dawning Mansion, Keji South 12th Road,
Hi- Tech Park, Shenzhen
Phone: +86-755-2699-3828 FAX: +86-755-2699-3838

EPSON TAIWAN TECHNOLOGY & TRADING LTD.
14F, No. 7, Song Ren Road,
Taipei 110
Phone: +886-2-8786-6688 FAX: +886-2-8786-6660

EPSON SINGAPORE PTE., LTD.
1 HarbourFront Place,
#03-02 HarbourFront Tower One, Singapore 098633
Phone: +65-6586-5500 FAX: +65-6271-3182

SEIKO EPSON CORPORATION
KOREA OFFICE
50F, KLI 63 Bldg., 60 Yoido-dong
Youngdeungpo-Ku, Seoul, 150-763, KOREA
Phone: +82-2-784-6027 FAX: +82-2-767-3677

GUMI OFFICE
2F, Grand B/D, 457-4 Songjeong-dong,
Gumi-City, KOREA
Phone: +82-54-454-6027 FAX: +82-54-454-6093

SEIKO EPSON CORPORATION
SEMICONDUCTOR OPERATIONS DIVISION

IC Sales Dept.
IC International Sales Group
421-8, Hino, Hino-shi, Tokyo 191-8501, JAPAN
Phone: +81-42-587-5814 FAX: +81-42-587-5117

Document Code: 405250202
First Issue April 2008
Printed in JAPAN Ⓗ