EPSON
EXCEED YOUR VISION

S2S65A30
Technical Manual (1)

SEIKO EPSON CORPORATION

Rev.1.3

NOTICE

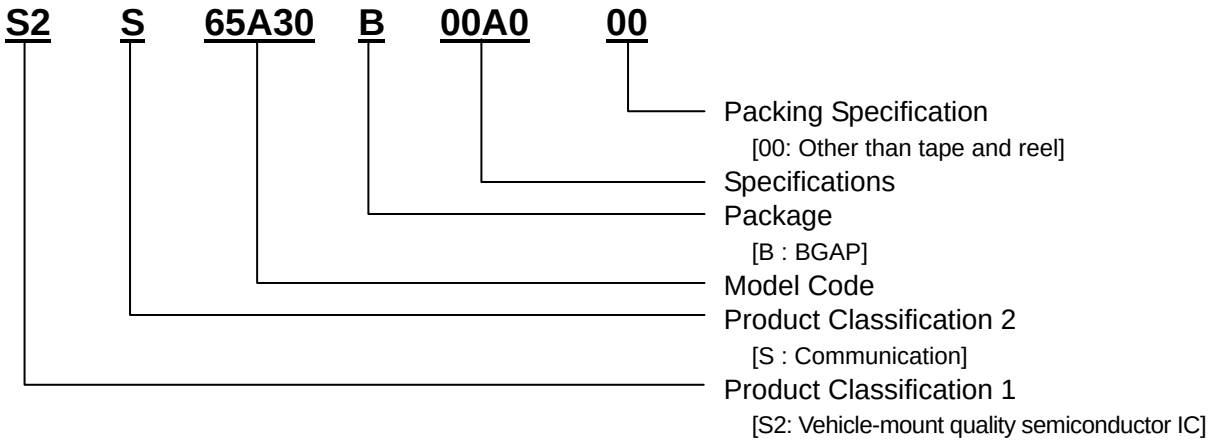
No part of this material may be reproduced or duplicated in any form or by any means without the written permission of Seiko Epson. Seiko Epson reserves the right to make changes to this material without notice. Seiko Epson does not assume any liability of any kind arising out of any inaccuracies contained 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.

is a registered trademark of ARM.
CompactFlash is a registered trademark of Sandisk.

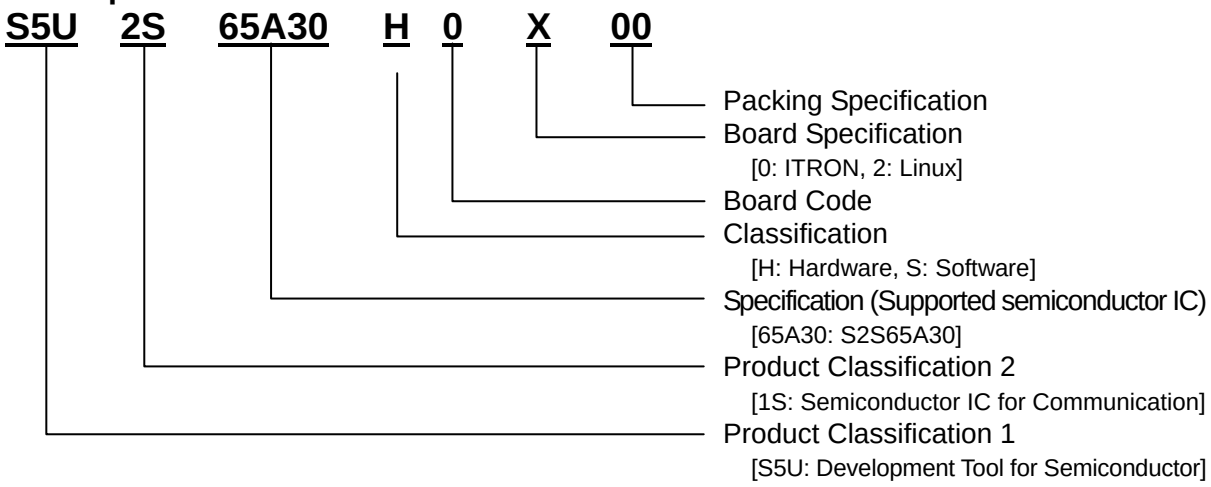
All brands or product names mentioned herein are trademarks and/or registered trademarks of their respective companies.

Configuration of product number

● **Device**



● **Development tool**



Precautions in Use

For descriptions of the registers in this document, be careful with the following:

For descriptions of the registers in this document, the following abbreviations may be used.

R/W:	Read and Write
RO:	Read Only
WO:	Write Only
RSV:	Reserved bit/register (write down “0”, if not otherwise specified)
n/a:	not available (write down “0”, if not otherwise specified)

If not otherwise specified, set “0” in the reserved bits for the registers. If a write operation performed on a reserved bit, unexpected results may occur. The bits specified as “n/a” have no impact on the hardware.

Some of the resisters can be accessed only under certain conditions. Read/write to the non-accessible registers is ineffective.

Table of Contents

1. DESCRIPTION

This product, S2S65A30, is an IC specifically for a drive recorder. It has camera interfaces and JPEG encoder/decoder functions, as well as embedded interlace/progressive converter, image corrector, CF, SD memory, USB 2.0 device (high-speed mode supported) interfaces, and 8-ch ADC. A drive recorder can easily be configured by connecting S2S65A30 with camera modules, SDRAM, an external storage (CF or SD memory card), and Flash ROM containing firmware. For example, JPEG data generated by the embedded encoder may constantly be accumulated into the SDRAM. In response to an external trigger, such as detection of a rapid change in speed by the acceleration sensor IC, the accumulated image data can be transferred to and saved in an external storage (CF or SD memory card). In addition, S2S65A30 is equipped with GPIO and I^2C bus, which enable you to set cameras and control external devices.

1.1 Features

- One-chip solution, which can reduce system cost.
- Provides JPEG encoding by using 30 fps @VGA hardware (ISO 10918 compliant).
- Up to two camera modules can be connected.
- Each camera module has two hardware JPEG encoders.
- Provides moving-object detection function to support motion detection.
- Supports I^2S for voice data.
- Has a CompactFlash interface for a CF memory card or a wireless LAN interface (802.11b/g).
- An SD memory interface for SD memory card connection.
- ARM720T 50MHz operation.
- USB 2.0 device (High-Speed) function support, which enables connection to a PC.
- Supports 8-ch ADC for connection with various analog sensors.
- Contains event counter timers.
- Memory bus: 2 ports (6bit-Bus: FROM/SRAM, 16/32bit-Bus: SDRAM).
- Interlace / progressive conversion
- JPEG decoding with DMA function
- Image contrast correction

1.2 Built-In Functions

CPU:

- 32-bit RISC ARM720T (maximum of 50MHz, with 8KB Cache).
- 32-bit long command codes and 16-bit long command codes called Efficient Thumb Code can be used by switching them.
- 32-bit general purpose register (×31).
- A multiplier is included in the CPU.

RAM:

- 56 KB Built-in RAM for CPU/JPEG Work (CPU Work: 32KB Max.).

Standby Function

- A HALT function to stop the CPU clock when any CPU operation is not required.
- An I/O clock stop function to stop each clock of the main I/O blocks.

Camera Input/ Interlace/Progressive Conversion:

- 8-bit parallel interface × 2 ports
- 2 camera modules can be connected.
- Up to 640×480 resolution (VGA, QVGA, CIF, QCIF).
- YUV4-2-2 progressive (both ports)
- Interlace signal conversion by the interlace/progressive converting function (both ports).
- Pixel clock frequency for inputting camera data is less than 2/3 of CPU clock frequency.

Image Correction:
- Contrast correction of camera channel 2.

JPEG:
- Hardware JPEG encoder×2
- Throughput greater than 30 fps @VGA (when 1 camera module connected).
- Resize function (screen can be cut off)
- Dedicated line buffer
- Variable volume FIFO built in the JPEG encoder output.
- An enhanced DMA is included in the network.
- Hardware JPEG decoder

USB2.0 Device:
- Supports HS (480Mbps) and FS (12Mbps) transfer.
- Has built-in FS/HS Termination function (external circuit not required).
- VBUS 5V Interface (external protection circuit required).
- Supports control bulk and interrupt transfer.
- Supports 8 end points shared by control (End Point 5) and bulk/interrupt
- Has a 16-bit or 8-bit width general purpose CPU interface.
- Littler Endian is supported.
- Addition to and deletion form the register table is performed based on the HS-Device section of S1R72V05.
- 12MHz or 24MHz crystal transducer input is supported as clock input for USB.
- As internal clock use, the following frequencies are available based on the clock for the input USB.
- Clock for input USB: 12MHz or 24MHz.
- Clock for internal USB: 60MHz (via PLL for built-in USB).

Memory Controller:
- AHB bus interface memory controller.
- Supports up to four SRAM timing devices.
- Supports up to eight SDRAMs.
- Refresh interval of SDRAM auto-refresh can be adjusted to the device.
- SDRAM burst refresh support.
- Supports SDRAM self-refresh.

CF Card Interface:
- Complies with CF+ Specification Rev.1.4.
- Can be used as the interface of wireless LAN, PHS card, etc.
- Supports the True IDE mode.

SD Memory Interface:
- Complies with SD Memory Card Physical Layer Spec. ver.2.0.

Interrupt Controller:
- Supports 32 IRQs and 2 FIQs.

Serial interface:
- UART: Compatible with 16550 software × 3 channel
- SPI: Clock synchronous type × 1 channel
- I^2C master interface (camera interface and multipurpose use)
- I^2S interface (voice/audio data supported, I^2S compliant)

Timer A:
- 16-bit timer × 3-channel timer
- Re-load/cyclic or one shot operation mode
- Supports toggle outputs resulted from underflow output or port outputs.

Timer B: for Event Counter

- 16-bit upcount timer
- Four COMMON registers are implemented, each of which can be configured as an output register or an input capture register.

Watchdog Timer

- Interrupt output or re-settable watchdog timer.

Real Time Clock

- Supports year, month, day, hour, minute, and second, as well as leap year.
- The internal timer tap from 1/128 to 1/2 can be used as the interrupt source as well.
- Supports alarm function and interrupt.

GPIO:

- General-purpose I/O port (up to 82)
- Programmable setting of directions is possible for all ports.
- Partly selects other I/O functions.

AD Converter:

- 8-channel analog signal inputs.
- 10-bits resolution AD converter.
- AD conversion time is 20μsec or less.

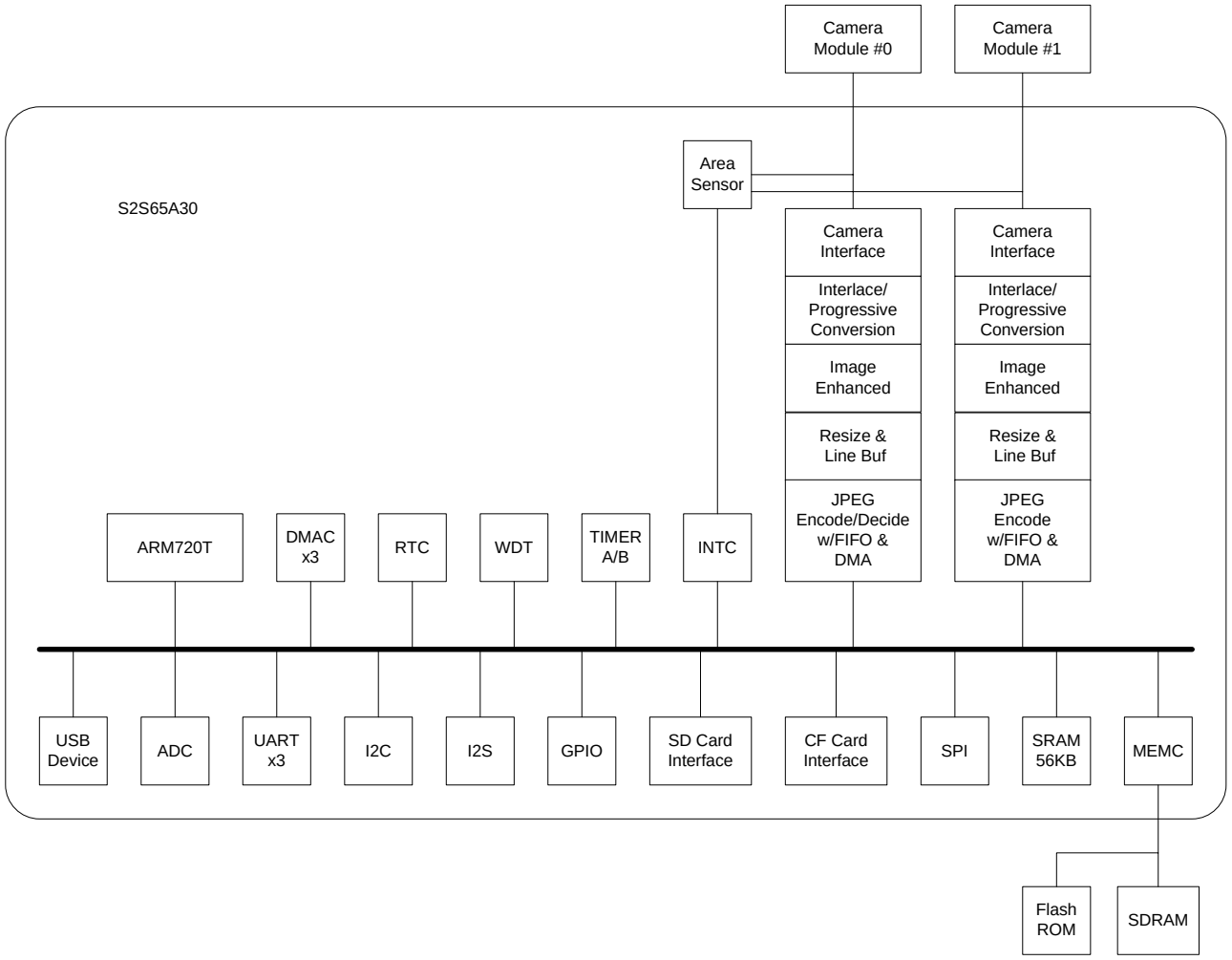
Power Supply:

- 3.3V (I/O power supply)
- 3.3V (USB)
- 1.8V (core power supply)
- 2.4 to 3.6V (Camera 1/2 I/O power supply)
- 2.7 to 3.6V (SDRAM I/O power supply)
- 3.3V (A/D power supply)
- 1.8V (USB/PLL/RTC)

Package:

- PFBGA 280 Pin (PFBGA16UX280) 16 × 16 × 1.2 mm, 0.8 mm ball pitch

2. BLOCK DIAGRAM



Note (*1) : Internal SRAM is shared with Line Buffer of JPG[2:1].When JPG[2:1] is used, CPU-Work cannot be used.

Figure 2.1 S2S65A30 Internal Block Diagram

3. PIN

3.1 Pin Assignment

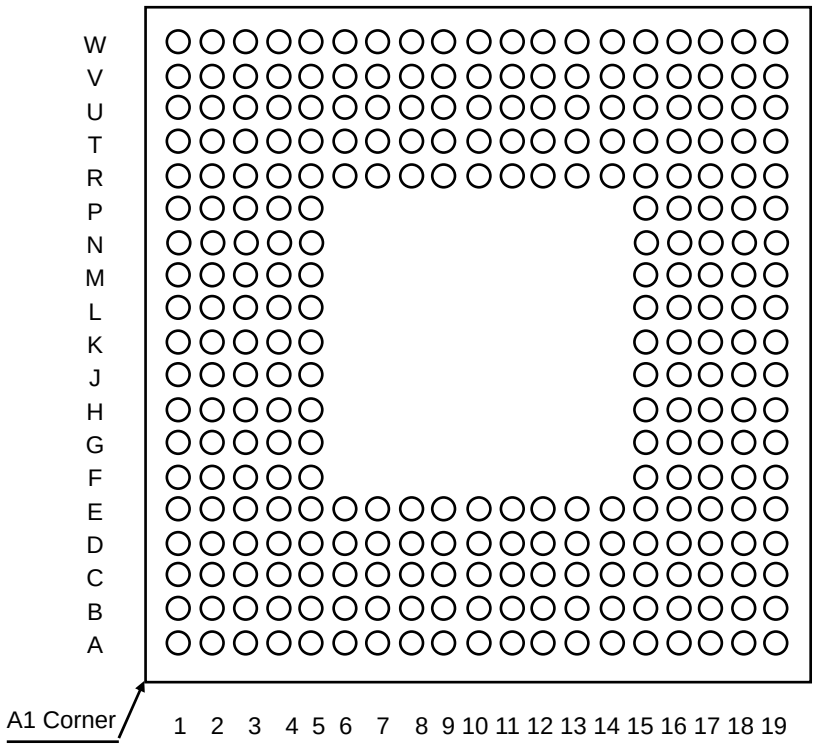


Figure 3.1 Pin Assignment (Bottom View)

Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name
A1	NC	D14	HVDD	K15	LVDD	T7	CM2DATA2
A2	CM1CLKIN	D15	GPIOA2	K16	CFDEN#	T8	GPIOD2
A3	CM1CLKOUT	D16	GPIOA4	K17	CFDDIR	T9	GPIOD1
A4	C1VDD	D17	GPIOA0	K18	LVDD	T10	MA17
A5	SYS_OSCO	D18	LVDD	K19	CFSTSCHG#	T11	MA15
A6	SYS_OSCI	D19	GPIOA1	L1	Vss	T12	MA11
A7	RTCVDD	E1	LVDD	L2	GPIOJ6	T13	MA7
A8	PLLVSS	E2	Vss	L3	GPIOJ3	T14	Vss
A9	SYSVCP	E3	SDA6	L4	GPIOJ5	T15	MA2
A10	SYSCLKI	E4	VSS	L5	LVDD	T16	MCS1#
A11	LVDD	E5	SDA0	L15	HVDD	T17	MD13
A12	Vss	E6	GPIOK5	L16	CFWAIT#	T18	MD12
A13	UXVDD	E7	GPIOK4	L17	CFRST	T19	MD10
A14	DP	E8	Vss	L18	Vss	U1	SDD0
A15	DM	E9	CM1DATA0	L19	CFIRQ	U2	SDDQM1#
A16	UVDD3	E10	TESTEN1	M1	GPIOJ0	U3	CM2HREF
A17	R1	E11	Vss	M2	GPIOJ2	U4	Vss
A18	UPVDD	E12	LVDD	M3	GPIOJ1	U5	CM2DATA4
A19	NC	E13	TDI	M4	SDD15	U6	AVSS
B1	SDA13	E14	GPIOB3	M5	Vss	U7	CM2DATA5
B2	SDA14	E15	GPIOB1	M15	CFCE2#	U8	CM2DATA7
B3	CM1HREF	E16	GPIOB0	M16	MD3	U9	GPIOD3

3. PIN

Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name
B4	Vss	E17	GPIOA3	M17	CFCE1#	U10	GPIOD0
B5	CM1DATA3	E18	HVDD	M18	CFIORD#	U11	MA16
B6	CM1DATA6	E19	GPIOA5	M19	CFIOWR#	U12	MA12
B7	Vss	F1	SDA1	N1	SDD11	U13	MA8
B8	BUP#	F2	SDVDD	N2	Vss	U14	MA4
B9	PLLVDD	F3	SDA5	N3	SDD13	U15	MA1
B10	SYSCKSEL	F4	SDA2	N4	SDD14	U16	Vss
B11	TRST#	F5	GPIOK1	N5	SDD12	U17	MBEL#
B12	TESTCK	F15	LVDD	N15	MD5	U18	MD14
B13	XVSS	F16	GPIOB2	N16	MD1	U19	MD15
B14	UVDD3	F17	GPIOA6	N17	Vss	V1	GPIOD6
B15	UVSS	F18	LVDD	N18	MD2	V2	Vss
B16	Vss	F19	GPIOA7	N19	MD0	V3	CM2VREF
B17	UVSS	G1	GPIOK7	P1	SDVDD	V4	CM2DATA0
B18	PVSS	G2	GPIOK6	P2	SDD10	V5	AVDD
B19	USBCK_OSCO	G3	GPIOK3	P3	SDD9	V6	ADIN7
C1	SDA11	G4	GPIOK0	P4	SDD8	V7	ADIN5
C2	Vss	G5	Vss	P5	Vss	V8	ADIN3
C3	SDVDD	G15	GPIOC1	P15	MD9	V9	ADIN1
C4	SDA12	G16	HVDD	P16	MD8	V10	AVSS
C5	CM1VREF	G17	GPIOB5	P17	MD6	V11	MA19
C6	CM1DATA5	G18	Vss	P18	MD4	V12	MA14
C7	CM1DATA7	G19	GPIOB4	P19	LVDD	V13	Vss
C8	Vss	H1	LVDD	R1	SDD6	V14	MA3
C9	HVDD	H2	Vss	R2	SDD5	V15	GPIOD5
C10	Vss	H3	GPIOK2	R3	SDD4	V16	MCS0#
C11	TESTEN0	H4	SDWE#	R4	SDD7	V17	MWE#
C12	TDO	H5	SDCAS#	R5	SDD3	V18	LVDD
C13	Vss	H15	GPIOC6	R6	SDD2	V19	Vss
C14	VBUS	H16	GPIOC0	R7	CM2DATA1	W1	NC
C15	Vss	H17	GPIOB6	R8	CM2DATA6	W2	CM2CLKIN
C16	HVDD	H18	HVDD	R9	Vss	W3	CM2CLKOUT
C17	Vss	H19	GPIOB7	R10	HVDD	W4	C2VDD
C18	Vss	J1	SDCS1#	R11	MA13	W5	AVDD
C19	USBCK_OSCI	J2	SDCS0#	R12	MA9	W6	ADIN6
D1	SDA8	J3	SDRAS#	R13	MA6	W7	ADIN4
D2	SDA9	J4	GPIOJ7	R14	MA5	W8	ADIN2
D3	SDA7	J5	GPIOJ4	R15	HVDD	W9	ADIN0
D4	SDA10	J15	GPIOC7	R16	GPIOD4	W10	AVDD
D5	SDA4	J16	GPIOC5	R17	MD11	W11	MA18
D6	SDA3	J17	GPIOC4	R18	Vss	W12	HVDD
D7	CM1DATA1	J18	GPIOC3	R19	MD7	W13	MA10
D8	CM1DATA2	J19	GPIOC2	T1	Vss	W14	LVDD
D9	CM1DATA4	K1	SDCLK	T2	LVDD	W15	HVDD
D10	LVDD	K2	SDCLKEN	T3	SDD1	W16	MA0
D11	TCK	K3	SDVDD	T4	GPIOD7	W17	MOE#
D12	TMS	K4	SDVDD	T5	SDDQM0#	W18	MBEH#
D13	RESET#	K5	Vss	T6	CM2DATA3	W19	NC

Note: # at the right end of pin name indicates to be an active low signal.

3.2 Pin Description

: # at the right end of pin name indicates to be an active low signal.
I : Input pin
O : Output pin
IO : Bi-directional pin
P : Power supply

Table 3.1 Cell Type Description

Cell Type	Description	Example of Pin being Used	
		Pin Name	Power Supply
ILS	Low Voltage LVCMOS Schmitt input	BUP#	RTCVDD
ICD1	LVCMOS input with pull-down resistor (50kΩ@3.3V)	TESTEN[1:0],TESTCK	HVDD
ICU1	LVCMOS input with pull-up resistor (50kΩ@3.3V)	TMS, TDI	HVDD
ICSU1	LVCMOS Schmitt input with pull-up resistor (50kΩ@3.3V)	TRST#,TCK,RESET#	HVDD
ICSD1	LVCMOS Schmitt input with pull-down resistor (50kΩ@3.3V)	SYSCKSEL	HVDD
ILTR	Low Voltage Transparent Input	SYS_OSCI	RTCVDD
		USBCK_OSCI	UPVDD
		R1	UVDD3
IHTR	High Voltage Transparent Input	ADIN[7:0]	ADVDD
OLTR	Low Voltage Transpatent Output	SYS_OSCO,SYSVCP	PLLVDD
		USBCK_OSCO	UPVDD
BLNC4U1	Low noise LVCMOS IO buffer with pull-up resistor (50kΩ@3.3V) (±4mA)	CF I/F	HVDD
BLNC4D2	Low noise LVCMOS IO buffer with pull-down resistor (100kΩ@3.3V) (±4mA)	MD [15:0]	HVDD
		SDD[15:0],GPIOJ,GPIOK	SDVDD
BLNS4	Low noise LVCMOS Schmitt IO buffer (±4mA)	GPIOA, GPIOB, GPIOC,GPIOD, SYSCLKI	HVDD
BLNS4D1	Low noise LVCMOS Schmitt IO buffer with pull-down resistor (50kΩ@3.3V) (±4mA)	Camera I/F	C1VDD, C2VDD
OLN4	Low noise output buffer (±4mA)	SRAM Device I/F (excluding MD)	HVDD
		SDRAM I/F (excluding SDD[15:0])	SDVDD
OTLN4	Low noise Tri-state output buffer (±4mA)	TDO	HVDD
USBDM	USB DM buffer	DM	UVDD3
USBDP	USB DP buffer	DP	UVDD3
USBVBUS	USB VBUS output buffer	VBUS	UVDD3

Table 3.2 Pin Description

Pin Name	Type	Cell Type	Pin No.	Description
(MA [23:20])	(I/O)	(BLNS4)		For information on these pins, see the description of **GPIOD [3:0]**.
MA [19:12]	O	OLN4	V11,W11, T10,U11, T11,V12, R11,U12	**Address Output Signal for Flash-ROM/SRAM [19:12]**
MA 11	O	OLN4	T12	This pin has the following functions: • **MA11: Address output signal for Flash-ROM/SRAM [11]** (Pin function right after reset) • **CFREG# Output Signal** When the compact flash (CF) interface is in operation, this signal functions as the REG signal selecting attribute of the CF interface and I/O space.
MA [10:0]	O	OLN4	W13,R12, U13,T13, R13,R14, U14,V14, T15,U15, W16	These pins have the following functions: • **MA [10:0]: Address output signal for Flash-ROM/SRAM [10:0]** (Pin function right after reset) • **CFADDR [10:0] Output signal** When the CF interface is in operation, this signal becomes the CF interface address signal [10:0].
MBEL#	I/O	OLN4	U17	Data bus low byte enable output for **Flash-ROM/SRAM**
MBEH#	I/O	OLN4	W18	Data bus high byte enable output for **Flash-ROM/SRAM**
MD [15:0]	I/O	BLNC4D2	U19,U18, T17,T18, R17,T19, P15,P16, R19,P17, N15,P18, M16,N18, N16,N19	These pins have the following functions: • **16-bit data bus for Flash-ROM/SRAM** (Pin function right after reset) • **When the CF interface is in operation, this pin becomes good for 16-bit data.** • **MODESEL[15:0]** Sampled to determine the internal operation mode, at power-on resetting (RESET# transition from Low to High). For details, see section “4.1 System Configuration”. Here, to determine the operation mode, a pull-up resistance may be required externally. (Resistance in the range from 4.7 to 10kΩ)
MCS [3:2]#	O	(BLNS4)	V15,R16	For information on these pins, see the description of **GPIOD [5:4]**.
MCS [1:0]#	O	OLN4	T16,V16	**Chip select signal for Flash-ROM/SRAM [1:0]** (Active low signal)
MOE#	O	OLN4	W17	This pin has the following functions: (Active low signal) • **MOE#: Strobe signal for Flash-ROM/SRAM** (Pin function right after reset) • **CFOE# output signal** When the CF interface is in operation, this signal becomes the output enable signal of CF interface memory and attribute spaces.

Pin Name	Type	Cell Type	Pin No.	Description
MWE#	O	OLN4	V17	This pin has the following functions: (Active low signal) • **MWE#: Write enable signal for Flash-ROM/SRAM** (for static memory) (Pin function right after reset) • **CFWE# output signal** When the CF interface is in operation, this signal becomes the write enable signal of CF interface memory and attribute spaces.
SDA [14:0]	O	OLN4	B2,B1, C4,C1, D4,D2, D1,D3, E3,F3, D5,D6, F4,F1, E5	**Address output for SDRAM [14:0]**
SDD[31:16]	(I/O)	(BLNC4D2)		For information on these pins, see the descriptions of **GPIOK [7:0] and GPIOJ [7:0]**.
SDD [15:0]	I/O	BLNC4D2	M4,N4, N3,N5, N1,P2, P3,P4, R4,R1, R2,R3, R5,R6, T3,U1	**Data I/O for SDRAM [15:0]**
SDWE#	O	OLN4	H4	**Write enable signal for SDRAM**
SDCLK	O	OLN4	K1	**Outputting clock for SDRAM** The same frequency as internal operation frequency (CPUCLK) is output.
SDCLKEN	O	OLN4	K2	**Clock enable signal for SDRAM**
SDRAS#	O	OLN4	J3	**RAS signal for SDRAM** (Active low signal)
SDCAS#	O	OLN4	H5	**CAS signal for SDRAM** (Active low signal)
SDCS[1:0]#	O	OLN4	J1,J2	**Chip select signal for SDRAM** (Active low signal)
SDDQM[3:2]#	(I/O)	(BLNS4)		For information on these pins, see the description of **GPIOD [7:6]**.
SDDQM[1:0]#	O	OLN4	U2,T5	**DQM signal for SDRAM** (Active low signal) SDDQM0# corresponds to the lower bytes; SDDQM1#, to the higher bytes.
CM1DATA[7:0]	I/O	BLNS4D1	C7,B6, C6,D9, B5,D8, D7,E9	These pins have the following functions. The pins are set to GPIOE[7:0] input after reset. • **CM1DATA[7:0]: Progressive camera 1 YUV data input** To use as the CM1DATA[7:0] pin, set bits [15:0] of GPIOE Pin Function Register to “Function 1 of other than GPIO”. • **IPC1DATA[7:0]: Interlace camera 1 YUV data input** To use as the IPC1DATA[7:0] pin, set bits [15:0] of GPIOE Pin Function Register to “Function 2 of other than GPIO”. • **GPIOE[7:0] I/O** (Pin function right after reset)

Pin Name	Type	Cell Type	Pin No.	Description
CM1VREF	I/O	BLNS4D1	C5	This pin has the following functions. The pin is set to GPIOFO input after reset. • **CM1VREF: Vertical sync input during progressive camera 1 data input** To use as the CM1VREF pin, set bits [1:0] of GPIOF Pin Function Register to "Function 1 of other than GPIO". • **IPC1VREF: Vertical sync input during interlace camera 1 data input** To use as the IPC1VREF pin, set bits [1:0] of GPIOF Pin Function Register to "Function 2 of other than GPIO". • **GPIOF0 I/O** (Pin function right after reset)
CM1HREF	I/O	BLNS4D1	B3	This pin has the following functions. The pin is set to GPIOF1 input after reset. • **CM1HREF: Horizontal sync input during progressive camera 1 data input** To use as the CM1HREF pin, set bits [3:2] of GPIOF Pin Function Register to "Function 1 of other than GPIO". • **IPC1VREF: Horizontal sync input during interlace camera 1 data input** To use as the IPC1HREF pin, set bits [3:2] of GPIOF Pin Function Register to "Function 2 of other than GPIO". • **GPIOF1 I/O** (Pin function right after reset)
CM1CLKOUT	I/O	BLNS4D1	A3	This pin has the following functions. The pin is set to GPIOF2 input after reset. • **CM1CLKOUT: Basic clock output for progressive camera 1** To use as the CM1CLKOUT pin, set bits [5:4] of GPIOF Pin Function Register to "Function 1 of other than GPIO". • **IPC1FIELD: Field identification signal input for interlace camera 1** To use as the IPC1FIELD pin, set bits [5:4] of GPIOF Pin Function Register to "Function 2 of other than GPIO". • **GPIOF2 I/O (Pin function right after reset)**
CM1CLKIN	I/O	BLNS4D1	A2	This pin has the following functions. The pin is set to GPIOF3 input after reset. • **CM1CLKIN: Pixel clock for progressive camera 1 data input** To use as the CM1CLKIN pin, set bits [7:6] of GPIOF Pin Function Register to "Function 1 of other than GPIO". • **IPC1CLKIN: Pixel clock for interlace camera 1 data input** To use as the IPC1CLKIN pin, set bits [7:6] of GPIOF Pin Function Register to "Function 2 of other than GPIO". • **GPIOF3 I/O** (Pin function right after reset)

Pin Name	Type	Cell Type	Pin No.	Description
CM2DATA[7:0]	I/O	BLNS4D1	U8,R8 U7,U5 T6,T7, R7,V4	These pins have the following functions. The pins are set to GPIOG[7:0] input after reset. • **CM2DATA[7:0]: Progressive camera 2 YUV data input** To use as the CM2DATA [7:0] pin, set bits [15:0] of GPIOG Pin Function Register to “Function 1 other than GPIO”. • **IPC2DATA[7:0]: Interlace camera 2 YUV data input** To use as the IPC2DATA[7:0] pin, set bits [15:0] of GPIOG Pin Function Register to “Function 2 of other than GPIO”. • **GPIOG[7:0] I/O** (Pin function right after reset)
CM2VREF	I/O	BLNS4D1	V3	This pin has the following functions. The pin is set to GPIOF4 input after reset. • **CM2VREF: Vertical sync input during progressive camera 2 data input** To use as the CM2VREF pin, set bits [9:8] of GPIOF Pin Function Register to “Function 1 of other than GPIO”. • **IPC2VREF: Vertical sync input during interlace camera 2 data input** To use as the IPC2VREF pin, set bits [9:8] of the GPIOF pin function register to “Function 2 of other than GPIO”. • **GPIOF4 I/O** (Pin function right after reset)
CM2HREF	I/O	BLNS4D1	U3	This pin has the following functions. The pin is set to GPIOF5 input after reset. • **CM1HREF: Horizontal sync input during progressive camera 2 data input** To use as the CM2HREF pin, set bits [11:10] of GPIOF Pin Function Register to “Function 1 of other than GPIO”. • **IPC2HREF: Horizontal sync input during interlace camera 2 data input** To use as the IPC2HREF pin, set bits [11:10] of GPIOF Pin Function Register to “Function 2 of other than GPIO”. • **GPIOF5 I/O** (Pin function right after reset)
CM2CLKOUT	I/O	BLNS4D1	W3	This pin has the following functions. The pin is set to GPIOF6 input after reset. • **CM2CLKOUT: Basic clock output for progressive camera 2** To use as the CM2CLKOUT pin, set bits [13:12] of GPIOF Pin Function Register to “Function 1 of other than GPIO”. • **IPC2FIELD: Field identification signal input for interlace camera 2** To use as the IPC2FIELF pin, set bits [13:12] of GPIOF Pin Function Register to “Function 2 of other than GPIOF”. • **GPIOF6 I/O** (Pin function right after reset)

Pin Name	Type	Cell Type	Pin No.	Description
CM2CLKIN	I/O	BLNS4D1	W2	This pin has the following functions. The pin is set to GPIOF7 input after reset. • **CM2CLKIN: Pixel clock for progressive camera 2 data input** To use as the CM2CLKIN pin, set bits [15:14] of GPIOF Pin Function Register to "Function 1 of other than GPIO". • **IPC2CLKIN: Pixel clock for interlace camera 2 data input** To use as the IPC2CLKIN pin, set bits [15:14] of GPIOF Pin Function Register to "Function 2 of other than GPIO". • **GPIOF7 I/O** (Pin function right after reset)
CFCE2#	I/O	BLNC4U1	M15	This pin has the following functions: • **CFCE2#: Card enable 2 (CE2#) output** (active low signal) **for Compact Flash Memory Interface (hereafter referred to as CF)** This pin, when reset, becomes good for inputting GPIOH0. To use as the CFCE2# pin, set bits [1:0] of the GPIOH pin function register to "Function 1 of other than GPIO". • **GPIOH0 I/O** (Pin function right after reset) • **SDMDATA0: Data I/O 0 for SD card** (Pin Function 2 of other than GPIO)
CFCE1#	I/O	BLNC4U1	M17	This pin has the following functions: • **CFCE1#: Card Enable 1 (CE1#) Output for CF** (Active low signal) This pin, when reset, becomes good for inputting GPIOH1. To use as the CFCE1# pin, set bits [3:2] of the GPIOH pin function register to "Function 1 of other than GPIO". • **GPIOH1 I/O** (Pin function right after reset) • **SDMDATA1: Data I/O 1 for SD card** (Pin Function 2 of other than GPIO)
CFIORD#	I/O	BLNC4U1	M18	This pin has the following functions: • **CFIORD#: IO read strobe output for CF** (Active low signal) This pin, when reset, becomes good for inputting GPIOH2. To use as the CFIORD# pin, set bits [5:4] of the GPIOH pin function register to "Function 1 other than GPIO". • **GPIOH2 I/O** (Pin function right after reset) • **SDMDATA2: Data I/O 2 for SD card** (Pin Function 2 of other than GPIO)
CFIOWR#	I/O	BLNC4U1	M19	This pin has the following functions: (Active low signal) • **CFIOWR#: IO write strobe output for CF** This pin, when reset, becomes good for inputting GPIOH3. To use as the CFIOWR# pin, set bits [7:6] of the GPIOH pin function register to "Function 1 other than GPIO". • **GPIOH3 I/O** (Pin function right after reset) • **SDMDATA3: Data I/O 3 for SD card** (Pin Function 2 of other than GPIO)

Pin Name	Type	Cell Type	Pin No.	Description
CFWAIT#	I/O	BLNC4U1	L16	This pin has the following functions: • **CFWAIT#: Wait request input for CF** (active low signal) This pin, when reset, becomes good for inputting GPIOH4. To use as the CFWAIT# pin, set bits [9:8] of the GPIOH pin function register to "Function 1 other than GPIO". • **GPIOH4 I/O** (Pin function right after reset) • **SDMCMD: Command I/O for SD card** (Function 2 of other than GPIO)
CFRST	I/O	BLNC4U1	L17	This pin has the following functions: • **CFRST: Reset signal to CF card** The signal is HIGH when the card is reset and LOW when the card is under normal operation. This pin, when reset, becomes good for inputting GPIOH5. To use as the CFRST pin, set bits [11:10] of the GPIOH pin function register to "Function 1 other than GPIO". • **GPIOH5 I/O** (Pin function right after reset) • **SDMCLK: Clock output for SD card** (Function 2 of other than GPIO)
CFIRQ	I/O	BLNC4U1	L19	This pin has the following functions: • **CFIRQ: Interrupt request signal from CF card** This pin, when reset, becomes good for inputting GPIOH6. To use as the CFIRQ pin, set bits [13:12] of the GPIOH pin function register to "Function 1 other than GPIO". • **GPIOH6 I/O** (Pin function right after reset) • **SDMCD#: Card detect input for SD card** (Function 2 of other than GPIO)
CFSTSCHG#	I/O	BLNC4U1	K19	This pin has the following functions: • **CFSTSCHG#: Status change signal from CF card** (active low signal) This pin, when reset, becomes good for inputting GPIOH7. To use as the CFSTSCHG# pin, set bits [15:14] of the GPIOH pin function register to "Function 1 other than GPIO". • **GPIOH7 I/O** (Pin function right after reset) • **SDMWP: Write protect input for SD card** (Function 2 of other than GPIO)
CFDEN#	I/O	BLNC4U1	K16	This pin has the following functions: • **CFDEN#: Data bus enable signal for external buffer of CF card** (active low signal) This pin, when reset, becomes good for inputting GPIOI0. To use as the CFDEN# pin, set bits [1:0] of the GPIOI pin function register to "Function 1 other than GPIO". • **GPIOI0 I/O pin** (Pin function right after reset) • **SDMGPO: General-purpose output for SD card** (Function 2 of other than GPIO)

Pin Name	Type	Cell Type	Pin No.	Description
CFDDIR	I/O	BLNC4U1	K17	This pin has the following functions: • **CFDDIR: Data bus directional instruction output for CF** When CF data is read, this pin becomes Low. Also, this pin, when reset, becomes good for inputting GPIOI1. To use as the CFDDIR pin, set bits [3:2] of the GPIOI pin function register to "Function 1 other than GPIO". • **GPIOI1 I/O** (Pin function right after reset)
R1	I	ILTR	A17	USB Device internal operation setting pin
DM	I/O	USBDM	A15	USB Device D- I/O
DP	I/O	USBDP	A14	USB Device D+ I/O
VBUS	I	USBVBUS	C14	USB Device VBUS input
ADIN[7:0]	I	IHTR	V6,W6, V7,W7, V8,W8, V9,W9	Analog signal input
GPIOA0	I/O	BLNS4	D17	This pin has the following functions: • **GPIOA0 I/O** (Pin function right after reset) • **TXD1: UART1 transmit data output** (Function 1 of other than GPIO)
GPIOA1	I/O	BLNS4	D19	This pin has the following functions: • **GPIOA1 I/O** (Pin function right after reset) • **RXD1: UART1 receive data input** (Function 1 of other than GPIO)
GPIOA2	I/O	BLNS4	D15	This pin has the following functions: • **GPIOA2 I/O** (Pin function right after reset) • **RTS1: UART1 request to send output** (Function 1 of other than GPIO) • **I2S1_WS: Word select for I2S1** (Function 2 of other than GPIO)
GPIOA3	I/O	BLNS4	E17	This pin has the following functions: • **GPIOA3 I/O** (Pin function right after reset) • **CTS1: UART1 clear to send input** (Function 1 of other than GPIO) • **I2S1_SCK: Serial clock for I2S1** (Function 2 of other than GPIO)
GPIOA4	I/O	BLNS4	D16	This pin has the following functions: • **GPIOA4 I/O** (Pin function right after reset) • **TXD2: UART2 transmit data output** (Function 1 of other than GPIO)
GPIOA5	I/O	BLNS4	E19	This pin has the following functions: • **GPIOA5 I/O** (Pin function right after reset) • **RXD2: UART2 receive data input** (Function 1 of other than GPIO)
GPIOA6	I/O	BLNS4	F17	This pin has the following functions: • **GPIOA6 I/O** (Pin function right after reset) • **RTS2: UART2 request to send output** (Function 1 of other than GPIO) • **SCL: I^2C clock I/O** (Function 2 of other than GPIO)
GPIOA7	I/O	BLNS4	F19	This pin has the following functions: • **GPIOA7 I/O** (Pin function right after reset) • **CTS2: UART2 clear to receive input** (Function 1 of other than GPIO) • **SDA: I^2C data I/O** (Function 2 of other than GPIO)

Pin Name	Type	Cell Type	Pin No.	Description
GPIOB0	I/O	BLNS4	E16	This pin has the following functions: • **GPIOB0 I/O** (Pin function right after reset) • **INT0 input** • **I2S0_WS: Word select for I2S0** (Function 1 of other than GPIO)
GPIOB1	I/O	BLNS4	E15	This pin has the following functions: • **GPIOB1 I/O** (Pin function right after reset) • **INT1 input** • **I2S0_SCK: Serial clock for I2S0** (Function 1 of other than GPIO)
GPIOB2	I/O	BLNS4	F16	This pin has the following functions: • **GPIOB2 I/O** (Pin function right after reset) • **INT2 input** • **I2S0_SD: Serial data for I2S0** (Function 1 of other than GPIO)
GPIOB3	I/O	BLNS4	E14	This pin has the following functions: • **GPIOB3 I/O** (Pin function right after reset) • **INT3 input** • **I2S1_SD: Serial data for I2S1** (Function 1 of other than GPIO)
GPIOB4	I/O	BLNS4	G19	This pin has the following functions: • **GPIOB4 I/O** (Pin function right after reset) • **INT4 input** • **TimerA0Out** (Function 1 other than GPIO)
GPIOB5	I/O	BLNS4	G17	This pin has the following functions: • **GPIOB5 I/O** (Pin function right after reset) • **INT5 input** • **TimerA1Out** (Function 1 other than GPIO) • **DREQ#** (Function 2 other than GPIO)
GPIOB6	I/O	BLNS4	H17	This pin has the following functions: • **GPIOB6 I/O** (Pin function right after reset) • **INT6 input** • **TimerA2Out** (Function 1 other than GPIO)
GPIOB7	I/O	BLNS4	H19	This pin has the following functions: • **GPIOB7 I/O** (Pin function right after reset) • **INT7 input** • **TimerBIn** (Function 1 other than GPIO)
GPIOC0	I/O	BLNS4	H16	This pin has the following functions: • **GPIOC0 I/O** (Pin function right after reset) • **TimerB0IO** (Function 1 other than GPIO)
GPIOC1	I/O	BLNS4	G15	This pin has the following functions: • **GPIOC1 I/O** (Pin function right after reset) • **TimerB1IO** (Function 1 other than GPIO) • **DACK#** (Function 2 other than GPIO)
GPIOC2	I/O	BLNS4	J19	This pin has the following functions: • **GPIOC2 I/O** (Pin function right after reset) • **TimerB2IO** (Function 1 other than GPIO) • **CFRST#** (Function 2 other than GPIO)
GPIOC3	I/O	BLNS4	J18	This pin has the following functions: • **GPIOC3 I/O** (Pin function right after reset) • **TimerB3IO** (Function 1 other than GPIO) • **UART3_CLK** (Function 2 of other than GPIO)
GPIOC4	I/O	BLNS4	J17	This pin has the following functions: • **GPIOC4 I/O** (Pin function right after reset) • **SPI_SS: Chip select for SPI** (Function 1 of other than GPIO) • **TXD3: UART3 transmit data output** (Function 2 of other than GPIO)

Pin Name	Type	Cell Type	Pin No.	Description
GPIOC5	I/O	BLNS4	J16	This pin has the following functions: • **GPIOC5 I/O** (Pin function right after reset) • **SPI_SCLK: Serial clock for SPI** (Function 1 of other than GPIO) • **RXD3: UART3 receive data input** (Function 2 of other than GPIO)
GPIOC6	I/O	BLNS4	H15	This pin has the following functions: • **GPIOC6 I/O** (Pin function right after reset) • **SPI_MISO: Serial data master input/slave output for SPI** (Function 1 of other than GPIO) • **RTS3: UART3 request to send output** (Function 2 of other than GPIO)
GPIOC7	I/O	BLNS4	J15	This pin has the following functions: • **GPIOC6 I/O** (Pin function right after reset) • **SPI_MOSI: Serial data master output/slave input for SPI** (Function 1 of other than GPIO) • **CTS3: UART3 clear to receive input** (Function 2 of other than GPIO)
GPIOD[3:0]	I/O	BLNS4	U9,T8, T9,U10	This pin has the following functions: • **GPIOD [3:0] I/O** (Pin function right after reset) • **MA [23:20]: Address output signal [23:20]** (Function 1 of other than GPIO)
GPIOD[5:4]	I/O	BLNS4	V15,R16	This pin has the following functions: • **GPIOD [5:4] I/O** (Pin function right after reset) • **MCS [3:2]#: Chip select output signal for memory** (Function 1 of other than GPIO)
GPIOD6	I/O	BLNS4	V1	This pin has the following functions: • **GPIOD6 I/O** (Pin function right after reset) • **DQM2# signal for SDRAM** (active low signal) • **SDDQM2#: Corresponds to the lower byte of the higher 16 bits of SDRAM 32-bit data width** (Function 1 of other than GPIO)
GPIOD7	I/O	BLNS4	T4	This pin has the following functions: • **GPIOD7 I/O** (Pin function right after reset) • **DQM3# signal for SDRAM** (active low signal) • **SDDQM3##: Corresponds to the higher byte of the higher 16 bits of SDRAM 32-bit data width** (Function 1 of other than GPIO)
GPIOJ[7:0]	I/O	BLNC4D2	J4,L2, L4,J5, L3,M2, M3,M1	This pin has the following functions: • **GPIOJ [7:0] I/O** (Pin function right after reset) • **SDD [23:16]: Data I/O for SDRAM** (Function 1 of other than GPIO)
GPIOK[7:0]	I/O	BLNC4D2	G1,G2, E6,E7, G3,H3, F5,G4	This pin has the following functions: • **GPIOK [7:0] I/O** (Pin function right after reset) • **SDD [31:24]: Data I/O for SDRAM** (Function 1 of other than GPIO)
SYSCLKI	I/O	BLNS4	A10	**32KHz system clock input** Basic clock input when SYSCKSEL is “HIGH”. 1/4 of system clock or 32KHz becomes output by setting it when SYSCKSEL is “LOW”.
SYS_OSCI	I	ILTR	A6	**Connection pin for crystal transducer** This is an operation clock oscillator pin. It connects a 32KHz crystal transducer.
SYS_OSCO	O	OLTR	A5	**Connection pin for crystal transducer** This is an operation clock oscillator pin. It connects a 32KHz crystal transducer.

Pin Name	Type	Cell Type	Pin No.	Description
SYSVCP	O	OLTR	A9	**System test pin for built-in PLL** This pin is used to monitor outputs of the PLL at the time of system test. Make the pin open at the time of normal operation.
SYSCKSEL	I	ICSD1	B10	**32KHz system clock input crystal transducer/oscillator select signal** Crystal oscillator is used when SYSCKSEL is "HIGH". Crystal transducer is used when SYSCKSEL is "LOW".
USBCK_OSCI	I	ILTR	C19	**Connection pin for crystal transducer** This is an operation clock oscillator pin specifically for USB. It connects a 12/24MHz crystal transducer.
USBCK_OSCO	O	OLTR	B19	**Connection pin for crystal transducer** This is an operation clock oscillator pin specifically for USB. It connects a 12/24MHz crystal transducer.
TRST#	I	ICSD1	B11	**Resetting for JTAG Interface** (active low signal) This signal is to be input with a Schmitt trigger with pull-down resistor.
TCK	I	ICSU1	D11	**Clock Input Pin for JTAG Interface** This clock is to be input with a Schmitt trigger.
TMS	I	ICU1	D12	**TMS Pin for JTAG Interface** This pin has a built-in pull-up resistor.
TDI	I	ICU1	E13	**Serial Data Input Pin for JTAG Interface** This pin has a built-in pull-up resistor.
TDO	O	OTLN4	C12	**Serial Data Output Pin for JTAG Interface**
TESTEN0	I	ICD1	C11	**Test enable 0 (active high signal)** This pin has a built-in pull-down resistor. Connect this pin to Vss or make it open at the time of normal operation.
TESTEN1	I	ICD1	E10	**Test enable 1** (active high signal) This pin has a built-in pull-down resistor. Connect this pin to Vss or make it open at the time of normal operation.
TESTCK	I	ICD1	B12	**Test clock** This pin has a built-in pull-down resistor. Connect this pin to Vss or make it open at the time of normal operation.
RESET#	I	ICSU1	D13	**System reset signal** (active low signal) Even after HVDD and LVDD become stable, keep RESET# active (LOW) for 100ms.
BUP#	I	ILS	B8	**Standby signal** (active low signal) 1.8V signal
HVDD	P	P	C9,C16, D14,E18, G16,H18, L15,R10, R15,W12, W15	**Power supply for I/O cell :** 3.3V (Typical) 3.0V (Min.) - 3.6V (Max.)
C1VDD	P	P	A4	**Power supply for camera 1 interface:** 3.0 (Typical) 2.4V (Min.) - 3.6V (Max.)
C2VDD	P	P	W4	**Power supply for camera 2 interface:** 3.0 (Typical) 2.4V (Min.) - 3.6V (Max.)

Pin Name	Type	Cell Type	Pin No.	Description
SDVDD	P	P	C3,F2, K3,K4, P1	**Power supply for SDRAM:** 3.3V (Typical) 2.7V (Min.) - 3.6V (Max.)
AVDD	P	P	V5,W5, W10	**Power supply for A/D C:** 3.3V (Typical) 3.0V (Min.) - 3.6V (Max.)
UVDD3	P	P	A16,B14	**Power supply for USB:** 3.3V (Typical) 3.0V (Min.) - 3.6V (Max.)
LVDD	P	P	A11,D10, D18,E1, E12,F15, F18,H1, K15,K18, L5,P19, T2,V18, W14	**Power supply for core (internal):** 1.8V (Typical) 1.65V (Min.) - 1.95V (Max.)
UPVDD	P	P	A18	**Power supply for USB:** 1.8V (Typical) 1.65V (Min.) - 1.95V (Max.)
UXVDD	P	P	A13	**Power supply for USB:** 1.8V (Typical) 1.65V (Min.) - 1.95V (Max.)
PLLVDD	P	P	B9	**Power supply for analog (PLL):** 1.8V (Typical) 1.65V (Min.) - 1.95V (Max.) Handling as an analog power supply is required. Supply stable power that generates less noise.
RTCVDD	P	P	A7	**Power supply for RTC:** 1.8V (Typical) 1.65V (Min) – 1.95V (Max)
UVSS	P	P	B15,B17	**Ground for USB**
PVSS	P	P	B18	**Ground for USB**
XVSS	P	P	B13	**Ground for USB**
AVSS	P	P	U6,V10	**Ground for A/D C**
PLLVSS	P	P	A8	**Ground for analog (PLL)** Handling as an analog power supply is required. Supply stable ground that generates less noise.
Vss	P	P	A12,B4, B7,B16, C2,C8, C10,C13, C15,C17, C18,E2, E4,E8, E11,G5, G18,H2, K5,L1, L18,M5, N2,N17, P5,R9, R18,T1, T14,U4, U16,V2, V13,V19	**Ground common to I/O cell, camera interface, and core power supplies**

3.3 Multiplex Pin Function of GPIO Pins, Pin Function Right after Reset

S2S65A30 Pin Name	Pin Function Right after Reset	FlashROM/ SAM Extension	SDRAM Extension	UART	I2C/ I2S/ SPI	Timer	Progressive Camera Interface	Camera Interface	CF Card	SD Card
GPIOA0	GPIOA0			TXD1						
GPIOA1	GPIOA1			RXD1						
GPIOA2	GPIOA2			RTS1	I2S1_WS					
GPIOA3	GPIOA3			CTS1	I2S1_SCK					
GPIOA4	GPIOA4			TXD2						
GPIOA5	GPIOA5			RXD2						
GPIOA6	GPIOA6			RTS2	SCL					
GPIOA7	GPIOA7			CTS2	SDA					
GPIOB0	GPIOB0				I2S0_WS					
GPIOB1	GPIOB1				I2S0_SCK					
GPIOB2	GPIOB2				I2S0_SD					
GPIOB3	GPIOB3				I2S1_SD					
GPIOB4	GPIOB4					TimerA0out				
GPIOB5	GPIOB5	DREQ#				TimerA1out				
GPIOB6	GPIOB6					TimerA2out				
GPIOB7	GPIOB7					TimerBIN				
GPIOC0	GPIOC0					TimerB0IO				
GPIOC1	GPIOC1	DACK#				TimerB1IO				
GPIOC2	GPIOC2					TimerB2IO			CFRST	
GPIOC3	GPIOC3			UART3_CLK		TimerB3IO				
GPIOC4	GPIOC4			TXD3	SPI_SS					
GPIOC5	GPIOC5			RXD3	SPI_SCLK					
GPIOC6	GPIOC6			RTS3	SPI_MISO					
GPIOC7	GPIOC7			CTS3	SPI_MOSI					
GPIOD0	GPIOD0	MA20								
GPIOD1	GPIOD1	MA21								
GPIOD2	GPIOD2	MA22								
GPIOD3	GPIOD3	MA23								
GPIOD4	GPIOD4	MCS2#								
GPIOD5	GPIOD5	MCS3#								
GPIOD6	GPIOD6		SDDQM2#							
GPIOD7	GPIOD7		SDDQM3#							
CM1DATA0	GPIOE0						CM1DATA0	IPC1DATA0		
CM1DATA1	GPIOE1						CM1DATA1	CM1DATA1		
CM1DATA2	GPIOE2						CM1DATA2	IPC1DATA2		
CM1DATA3	GPIOE3						CM1DATA3	IPC1DATA3		
CM1DATA4	GPIOE4						CM1DATA4	IPC1DATA4		
CM1DATA5	GPIOE5						CM1DATA5	IPC1DATA5		
CM1DATA6	GPIOE6						CM1DATA6	IPC1DATA6		
CM1DATA7	GPIOE7						CM1DATA7	IPC1DATA7		
CM1VREF	GPIOF0						CM1VREF	IPC1VREF		
CM1HREF	GPIOF1						CM1HREF	IPC1HREF		
CM1CLKOUT	GPIOF2						CM1CLKOUT	IPC1FIELD		
CM1CLKIN	GPIOF3						CM1CLKIN	IPC1CLKIN		
CM2VREF	GPIOF4						CM2VREF	IPC2VREF		
CM2HREF	GPIOF5						CM2HREF	IPC2HREF		
CM2CLKOUT	GPIOF6						CM2CLKOUT	IPC2FIELD		
CM2CLKIN	GPIOF7						CM2CLKIN	IPC2CLKIN		
CM2DATA0	GPIOG0						CM2DATA0	IPC2DATA0		
CM2DATA1	GPIOG1						CM2DATA1	IPC2DATA1		
CM2DATA2	GPIOG2						CM2DATA2	IPC2DATA2		
CM2DATA3	GPIOG3						CM2DATA3	IPC2DATA3		
CM2DATA4	GPIOG4						CM2DATA4	IPC2DATA4		
CM2DATA5	GPIOG5						CM2DATA5	IPC2DATA5		
CM2DATA6	GPIOG6						CM2DATA6	IPC2DATA6		
CM2DATA7	GPIOG7						CM2DATA7	IPC2DATA7		

S2S65A30 Pin Name	Pin Function Right after Reset	FlashROM/ SAM Extension	SDRAM Extension	UART	I2C/ I2S/ SPI	Timer	Progressive Camera Interface	Camera Interface	CF Card	SD Card
CFCE2#	GPIOH0								CFCE2#	SDMDATA0
CFCE1#	GPIOH1								CFCE1#	SDMDATA1
CFIORD#	GPIOH2								CFIORD#	SDMDATA2
CFIOWR#	GPIOH3								CFIOWR#	SDMDATA3
CFWAIT#	GPIOH4								CFWAIT#	SDMCMD
CFRST	GPIOH5								CFRST	SDMCLK
CFIRQ	GPIOH6								CFIRQ	SDMCD#
CFSTSCHG#	GPIOH7								CFSTSCHG#	SDMWP
CFDEN#	GPIOI0								CFDEN#	SDMGPO
CFDDIR	GPIOI1								CFDDIR	
GPIOJ0	GPIOJ0		SDD16							
GPIOJ1	GPIOJ1		SDD17							
GPIOJ2	GPIOJ2		SDD18							
GPIOJ3	GPIOJ3		SDD19							
GPIOJ4	GPIOJ4		SDD20							
GPIOJ5	GPIOJ5		SDD21							
GPIOJ6	GPIOJ6		SDD22							
GPIOJ7	GPIOJ7		SDD23							
GPIOK0	GPIOK0		SDD24							
GPIOK1	GPIOK1		SDD25							
GPIOK2	GPIOK2		SDD26							
GPIOK3	GPIOK3		SDD27							
GPIOK4	GPIOK4		SDD28							
GPIOK5	GPIOK5		SDD29							
GPIOK6	GPIOK6		SDD30							
GPIOK7	GPIOK7		SDD31							

Function 1 : Function 1 Function 2 : Function 2

3.4 Pin Status during Reset

Pin Name	Direction during RESET	Value during RESET	Presence of Internal Resistor	Description
MA[19:0]	Output	Low (However, only bit11 is High.)	No	
MD[15:0]	Input	High-Z(Pull-down)	Yes, Pull Down register	100kΩ
MCS[1:0]#	Output	High	No	
MCS[3:2]#	Input	High-Z	No	Depends on the external circuit. GPIOD[5:4]
MBEL#	Output	Low	No	
MBEH#	Output	Low	No	
MOE#	Output	High	No	
MWE#	Output	High	No	
SDA[14:0]	Output	Low	No	
SDD[15:0]	Input	High-Z(Pull-down)	Yes, Pull Down register	100kΩ
SDCS[1:0]#	Output	High	No	
SDWE#	Output	High	No	
SDCLK	Output	SDCLK(32KHz)	No	
SDLKEN	Output	Low	No	
SDRAS#	Output	High	No	
SDCAS#	Output	High	No	
SDDQM[1:0]#	Output	Low	No	
CM1DATA[7:0]	Input	High-Z(Pull-down)	Yes, Pull Down register	50kΩ
CM1VREF	Input	High-Z(Pull-down)	Yes, Pull Down register	50kΩ
CM1HREF	Input	High-Z(Pull-down)	Yes, Pull Down register	50kΩ
CM1CLKOUT	Input	High-Z(Pull-down)	Yes, Pull Down register	50kΩ
CM1CLKIN	Input	High-Z(Pull-down)	Yes, Pull Down register	50kΩ
CM2DATA[7:0]	Input	High-Z(Pull-down)	Yes, Pull Down register	50kΩ
CM2VREF	Input	High-Z(Pull-down)	Yes, Pull Down register	50kΩ
CM2HREF	Input	High-Z(Pull-down)	Yes, Pull Down register	50kΩ
CM2CLKOUT	Input	High-Z(Pull-down)	Yes, Pull Down register	50kΩ
CM2CLKIN	Input	High-Z(Pull-down)	Yes, Pull Down register	50kΩ
CFCE2#	Input	High-Z(Pull-up)	Yes, Pull Up register	50kΩ
CFCE1#	Input	High-Z(Pull-up)	Yes, Pull Up register	50kΩ
CFIORD#	Input	High-Z(Pull-up)	Yes, Pull Up register	50kΩ
CFIOWR#	Input	High-Z(Pull-up)	Yes, Pull Up register	50kΩ
CFWAIT#	Input	High-Z(Pull-up)	Yes, Pull Up register	50kΩ
CFRST	Input	High-Z(Pull-up)	Yes, Pull Up register	50kΩ
CFIRQ	Input	High-Z(Pull-up)	Yes, Pull Up register	50kΩ
CFSTSCHG#	Input	High-Z(Pull-up)	Yes, Pull Up register	50kΩ
CFDEN#	Input	High-Z(Pull-up)	Yes, Pull Up register	50kΩ
CFDDIR	Input	High-Z(Pull-up)	Yes, Pull Up register	50kΩ
GPIOA[7:0]	Input	High-Z	No	Depends on the external circuit.
GPIOB[7:0]	Input	High-Z	No	Depends on the external circuit.
GPIOC[7:0]	Input	High-Z	No	Depends on the external circuit.
GPIOD[7:0]	Input	High-Z	No	Depends on the external circuit.
GPIOJ[7:0]	Input	High-Z(Pull-down)	Yes, Pull Down register	100kΩ
GPIOK[7:0]	Input	High-Z(Pull-down)	Yes, Pull Down register	100kΩ
SYSCLKI	–	High-Z	No	Depends on the SYSCKSEL Pin
SYSVCP	Output	High-Z	No	Leave this pin Open when it is used
SYSCKSEL	Input	High-Z(Pull-down)	Yes, Pull Down register	50kΩ
TRST#	Input	High-Z(Pull-down)	Yes, Pull Down register	50kΩ
TCK	Input	High-Z(Pull-up)	Yes, Pull Up register	50kΩ
TMS	Input	High-Z(Pull-up)	Yes, Pull Up register	50kΩ
TDI	Input	High-Z(Pull-up)	Yes, Pull Up register	50kΩ
TDO	Output	High-Z	No	
TESTEN0	Input	High-Z(Pull-down)	Yes, Pull Down register	50kΩ
TESTEN1	Input	High-Z(Pull-down)	Yes, Pull Down register	50kΩ
TESTCK	Input	High-Z(Pull-down)	Yes, Pull Down register	50kΩ
RESET#	Input	Low	No	

4. FUNCTIONAL DESCRIPTION

4.1 System Configuration

S2S65A30 determines internal chip operation based on the MD bus.
Specifically, the IC's operation mode is determined by sampling internally and establishing the values of the MD [15:0] bus while it resets for the period during which RESET# is active and which it is starting up. Usually, you do not have to configure anything because the MD bus have a built-in pull-down resistor (Typ: 100kΩ). Depending on your system, you can add an external pull-up resistor and change the operation mode.
If an external pull-up resistor is used, steady current through the external pull-up resistor and the internal pull-down resistor should be present. In S2S65A30, however, this steady current can be turned off by the software that disconnects the internal pull-down resistor.
For more information, see "13. SYSTEM CONTROLLER".

Table 4.1 System Configuration Pins MODESEL [15:0]

Pin Name	Pin Functions	Value at Resetting	
		Low	High
MD0	MODESEL0	32KHz Mode (normal operation)	Reserved (For test)*
MD1	MODESEL1	Crystal oscillator signal stabilization wait time 00: Approx. 3 sec. 01: Approx. 100 msec. 10: Approx. 20 msec. 11: Approx. 2 msec.	
MD2	MODESEL2		
MD3	MODESEL3	For user setting	For user setting
MD4	MODESEL4	For user setting	For user setting
MD5	MODESEL5	For user setting	For user setting
MD6	MODESEL6	For user setting	For user setting
MD7	MODESEL7	For user setting	For user setting
MD8	MODESEL8	For user setting	For user setting
MD9	MODESEL9	For user setting	For user setting
MD10	MODESEL10	For user setting	For user setting
MD11	MODESEL11	For user setting	For user setting
MD12	MODESEL12	For user setting	For user setting
MD13	MODESEL13	For user setting	For user setting
MD14	MODESEL14	For user setting	For user setting
MD15	MODESEL15	For user setting	For user setting

Note *: Do not use the setting of where Reserved (for test) is specified. If the setting of Reserved (for test) is used, this IC may be broken.

MD0: Clock input setting
Low: 32KHz (PLL input: normal operation)
High: Reserved (for test; cannot be used by users)

MD[2:1]: Crystal oscillator signal stabilization wait time setting
00: Approx. 3 sec.
01: Approx. 100 msec.
10: Approx. 20 msec
11: Approx. 2 msec.

MD [15:3]: Provides 12-bit portion for users.
Set values are reflected to the chip configuration register inside the system controller. Users can use this for their purpose.

4.2 Memory Map

There are three AHB buses in the IC: The one that is connected to ARM720T is hereafter referred to as AHB1 bus, the one connected to DMAC2 is referred to as AHB2, and the other connected to DMAC3 is referred to as AHB3. ARM720T, DMAC2, and DMAC3 can gain access respectively to AHB1, AHB2, and AHB3 buses.

4.2.1 Memory Map (AHB1)

Table 4.2 AHB1 Memory Map

Start Address	End Address	Size	AHB1	AHB2	AHB3	Chip Select Pin
0x0000_0000	0x00FF_FFFF	16MB	External ROM			MCS0#
0x0100_0000	0x03FF_FFFF	48MB	Reserved			
0x0400_0000	0x04FF_FFFF	16MB	External SRAM			MCS1#
0x0500_0000	0x07FF_FFFF	48MB	Reserved			
0x0800_0000	0x08FF_FFFF	16MB	External SRAM			MCS2#
0x0900_0000	0x0BFF_FFFF	48MB	Reserved			
0x0C00_0000	0x0CFF_FFFF	16MB	External SRAM			MCS3#
0x0D00_0000	0x0FFF_FFFF	48MB	Reserved			
0x1000_0000	0x1FFF_FFFF	256MB	Reserved			
0x2000_0000	0x2FFF_FFFF	256MB	Internal SRAM			
0x3000_0000	0x3FFF_FFFF	256MB	Reserved			
0x4000_0000	0x4FFF_FFFF	256MB	External SDRAM			SDCS0#
0x5000_0000	0x5FFF_FFFF	256MB	External SDRAM			SDCS1#
0x6000_0000	0x6FFF_FFFF	256MB	Reserved			
0x7000_0000	0x7FFF_FFFF	256MB	Reserved			
0x8000_0000	0x8FFF_FFFF	256MB	SDMMC I/F			
0x9000_0000	0x9FFF_FFFF	256MB	Reserved			
0xA000_0000	0xAFFF_FFFF	256MB	Reserved			
0xB000_0000	0xBFFF_FFFF	256MB	Reserved			
0xC000_0000	0xC0FF_FFFF	16MB	External ROM			MCS0#
0xC100_0000	0xC3FF_FFFF	48MB	Reserved			
0xC400_0000	0xC4FF_FFFF	16MB	External SRAM			MCS1#
0xC500_0000	0xC7FF_FFFF	48MB	Reserved			
0xC800_0000	0xC8FF_FFFF	16MB	External SRAM			MCS2#
0xC900_0000	0xCBFF_FFFF	48MB	Reserved			
0xCC00_0000	0xCCFF_FFFF	16MB	External SRAM			MCS3#
0xCD00_0000	0xCFFF_FFFF	48MB	Reserved			
0xD000_0000	0xDFFF_FFFF	256MB	Reserved	JPEG2 Port	Reserved	
0xE000_0000	0xEFFF_FFFF	256MB	Reserved	JPEG1 Port	Reserved	
0xF000_0000	0xFFFF_FFFF	256MB	I/O Register	Reserved	USB DMA Port	

The device connected to CS0 becomes the boot device.
The area from 0xC000_0000 to 0xCFFF_FFFF is used as the alias space area for the device assigned to 0x0000_0000. This means that you can see the same information by accessing this space.
The portion shown at the bottom of the memory map is used by S2S65A30's built-in I/O device. The layout of this built-in I/O is shown in "Specifications: IO Map".
Memories connected to S2S65A30's (external) memory controller, i.e., external ROM, SRAM, and SDRAM, as well as internal SRAM, can be accessed as shared resource via both AHB buses (master).

4.3 I/O Map

The internal I/O area uses the following portion of the 256 MB area from 0xF000_0000 to 0xFFFF_FFFF. No device exists for the blank (Reserved) sections. If they are accessed, only indefinite data will be read.

Table 4.3 Built-in I/O Map

Base Address	Space Size	S2S65A30
0xFFFD_0000	4KB	Reserved
0xFFFD_1000	1KB	IPC timing & Scaler2
0xFFFD_1400	1KB	IPC2
0xFFFD_1800	2KB	AME2
0xFFFD_2000	4KB	Reserved
0xFFFD_3000	4KB	Timer B
0xFFFD_4000	4KB	UART2
0xFFFD_5000	4KB	UART3
0xFFFD_6000	4KB	Reserved
0xFFFD_7000	4KB	Reserved
0xFFFD_8000	4KB	Camera Interface_2
0xFFFD_9000	4KB	JPEG resize_2
0xFFFD_A000	4KB	JPEG module/FIFO_2
0xFFFD_B000	4KB	JPEG codec_2
0xFFFD_C000	4KB	ADC
0xFFFD_D000	4KB	Reserved
0xFFFD_E000	4KB	Reserved
0xFFFD_F000	4KB	USB_2.0_Device
0xFFFE_0000	4KB	APB bridge
0xFFFE_1000	1KB	IPC timing & Scaler1
0xFFFE_1400	1KB	IPC1
0xFFFE_1800	2KB	AME1
0xFFFE_2000	4KB	Reserved
0xFFFE_3000	4KB	DMAC1
0xFFFE_4000	2KB	CF attribute
0xFFFE_4800	2KB	CF common
0xFFFE_5000	2KB	CF I/O
0xFFFE_5800	1KB	CF ture IDE CS1#
0xFFFE_5C00	1KB	CF ture IDE CS2#
0xFFFE_6000	4KB	CF control
0xFFFE_7000	4KB	ARS
0xFFFE_8000	4KB	Camera Interface_1
0xFFFE_9000	4KB	JPEG resize_1
0xFFFE_A000	4KB	JPEG module/FIFO_1
0xFFFE_B000	4KB	JPEG codec_1
0xFFFE_C000	4KB	JPEG DMAC
0xFFFE_D000	4KB	I2C
0xFFFE_E000	4KB	I2S
0xFFFE_F000	4KB	(Interrupt controller)
0xFFFF_0000	4KB	Reserved
0xFFFF_1000	4KB	GPIO/pin function
0xFFFF_2000	4KB	SPI
0xFFFF_3000	4KB	DMAC3
0xFFFF_4000	4KB	Reserved
0xFFFF_5000	4KB	UART1
0xFFFF_6000	1KB	Reserved
0xFFFF_7000	4KB	Reserved
0xFFFF_8000	4KB	RTC
0xFFFF_9000	4KB	DMAC2
0xFFFF_A000	4KB	Memory controller
0xFFFF_B000	4KB	Timer A
0xFFFF_C000	4KB	WDT
0xFFFF_D000	4KB	System controller
0xFFFF_E000	4KB	Reserved
0xFFFF_F000	4KB	Interrupt controller

4.4 Interrupt Controller

S2S65A30 can handle two sources for FIQ and up to 32 interrupt sources for IRQ.
Based on the IRQ mapping, internal interrupt requests are connected to the interrupt controller as shown below.
For more information, see “15. INTERRUPT CONTROLLER”.

Type	Level	S2S65A30
FIQ	FIQ0	Watchdog timer
	FIQ1	GPIOB0
IRQ	IRQ0	Watchdog timer
	IRQ1	Interrupt controller
	IRQ2	ARM720T COMMRx
	IRQ3	ARM721T COMMTx
	IRQ4	Timer A Ch.0
	IRQ5	Timer A Ch.1
	IRQ6	Timer A Ch.2
	IRQ7	Reserved
	IRQ8	JPEG control_1
	IRQ9	DMA1C
	IRQ10	JPEG DMAC_1
	IRQ11	Camera Interface_1 / IPC_1
	IRQ12	ARS
	IRQ13	DMA2C
	IRQ14	GPIOA or GPIOB
	IRQ15	SPI
	IRQ16	I^2C
	IRQ17	UART_1
	IRQ18	RTC
	IRQ19	CF card Interface
	IRQ20	INT_GPIOB (GPIOB)
	IRQ21	Timer B
	IRQ22	DMAC_3
	IRQ23	JPEG control_2
	IRQ24	JPEG DMAC_2
	IRQ25	Camera Interface_2 / IPC_2
	IRQ26	UART2
	IRQ27	UART3
	IRQ28	SD memory
	IRQ29	USB_DEV
	IRQ30	ADC
	IRQ31	I^2S

4.5 Built-in Functions of S2S65A30

S2S65A30 has many built-in functional blocks to realize drive recorder function.
These functions are listed below.

Chapter Number	Functional Block Name	Abbr. for Function Name
5.	CPU	CPU
6.	DMA Controller 1	DMAC1
7.	Camera Interface [1:0]	CAM[1:0]
8.	JPEG Controller [1:0]	JPG[1:0]
9.	JPEG_DMAC	JDMA
10.	DMA Controller 2	DMAC2
11.	USB2.0 Device	USB2d
12.	APB Bridge	APB
13.	System Controller	SYS
14.	Memory Controller	MEMC
15.	Interrupt Controller	INT
16.	UART[2:0]	UART[2:0]
17.	I^2C Single Master Core Module	I^2C
18.	I^2S Interface	I^2S
19.	Serial Peripheral Interface	SPI
20.	Compact Flash Card Interface	CF
21.	SD Memory Controller Interface	SD/SDMMC
22.	Timer A	TIMA
23.	Timer B	TIMB
24.	Real Time Clock	RTC
25.	Watchdog Timer	WDT
26.	GPIO	GPIO
27.	A/D Converter	ADC
28.	Area Sensor	ARS
29	IP Conversion Module	IPC
30	DMA Controller 3	DMAC3

5. CPU

5.1 Description

The ARM720T core is used as the S2S65A30 CPU module. ARM720T, whose core is ARM7TDMI, is equipped with a unified 8k-byte cache, a memory management unit, and an extended write buffer.
For more information on the ARM720T core, see *"ARM720T Revision 4 (AMBA AHB Bus Interface Version) core CPU Manual"*.

5.2 ARM720T Block Diagram

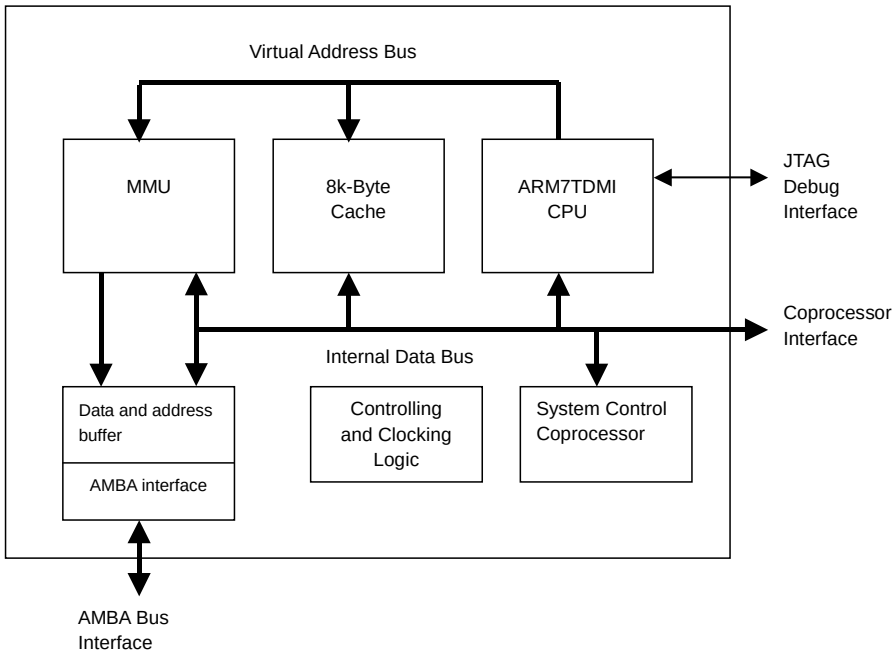


Figure 5.1 ARM720T Block Diagram

6. DMA CONTROLLER 1 (DMAC1)

6.1 Description

DMAC1, which is placed as the bus master on AHB1, is a DMA controller that transfers data between an APB device and a memory (internal or external memory) or between memories (internal and external memories) without using the CPU.
This DMA controller employs a dual-address transfer method where two address phases are used. In this method, data from the transfer source address is read into the temporary register within DMAC1 and then written to the destination address, each time a DMA request arises. This operation is repeated until the number of transfers reaches "0". You can specify 8, 16, or 32 bits as the transfer size.

6.2 Block Diagram

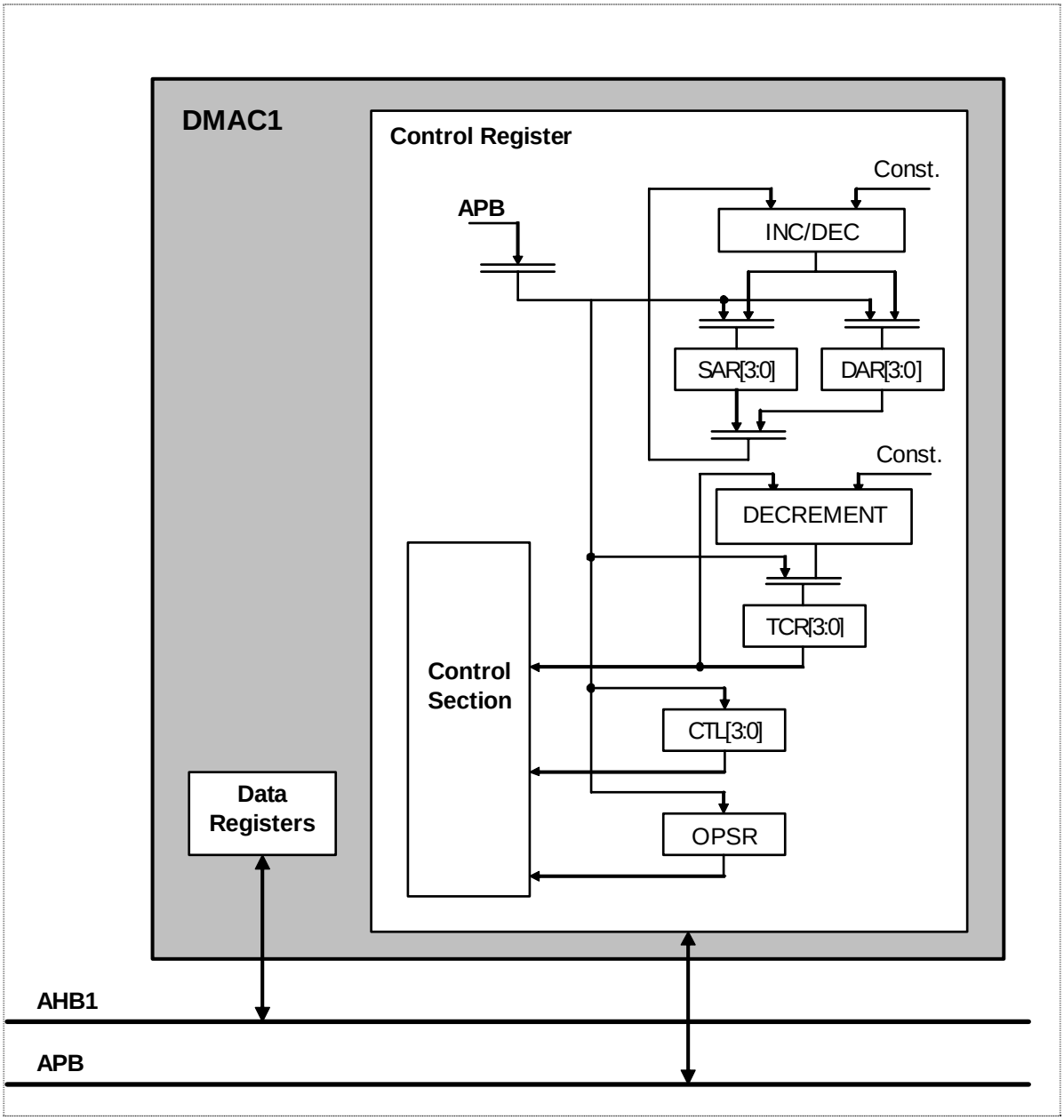


Figure 6.1 DMA Controller 1 (DMAC1) Block Diagram

6.3 External Pins

There is no external pin that relates to DMA Controller 1.

6.4 Registers

6.4.1 List of Registers

The base address of a DMAC1-related register is 0xFFFE_3000.
For the registers described in this chapter and latter chapters, the following abbreviations may be used.

R/W:	Read and Write
RO:	Read Only
WO:	Write Only
RSV:	Reserved bit/register (write down “0”, if not otherwise specified)
n/a:	not available (write down “0”, if not otherwise specified)

Table 6.1 List of Registers (Base Address: 0xFFFE_3000)

Address Offset	Register Name	Register Abbreviation	Default Value	R/W	Data Access Size
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	0x00XX_XXXX	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	0x00XX_XXXX	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	0x00XX_XXXX	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	0x00XX_XXXX	R/W	32
0x60	DMA Channel Operating Select Register	OPSR	0x0000_0000	R/W	32

6.4.2 Details of Registers

If not otherwise specified, set "0" in the reserved bits. If a write operation performed on a reserved bit, unexpected results may occur. The bits specified as "n/a" have no impact on the hardware.
Some of the resisters can be accessed only under certain conditions. Read/write to the non-accessible registers is ineffective.

DMA Channel [3:0] Source Address Register (SAR [3:0])
DMAC1[0x00] [0x10] [0x20] [0x30] Default value = 0xXXXX_XXXX Read/Write

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

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

A transfer source address for DMA transfer on channel [3:0] is set by the software. A transfer source address must be an boundary address appropriate for the transferred data size. For example, the bits [1:0] of this resister must be 00b for 32-bit transfer. Once DMA transfer begins, the transfer source address automatically updates to the next address every time when a transfer ends, according to the transferred data size (TS: channel [3:0] control register bits [4:3]) and the source address mode (SAM: channel [3:0] control register bits [13:12]).

DMA Channel [3:0] Destination Address Register (DAR [3:0])
DMAC1[0x04] [0x14] [0x24] [0x34] Default value = 0xXXXX_XXXX Read/Write

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

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

A transfer destination address for DMA transfer on channel [3:0] is set by the software. A transfer destination address must be an boundary address appropriate for the transferred data size. For example, the bits [1:0] of this resister must be 00b for 32-bit transfer. Once DMA transfer begins, the transfer destination address automatically updates to the next address every time when a transfer ends, according to the transferred data size (TS: channel [3:0] control register bits [4:3]) and the destination address mode (DAM: channel [3:0] control register bits [15:14]).

DMA Channel [3:0] Transfer Count Register (TCR [3:0])
DMAC1[0x08] [0x18] [0x28] [0x38] Default value = 0x00XX_XXXX Read/Write

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

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

In these bits, the number of transfers for DMA transfer is set by the software. Once DMA transfer begins, the number decrements every time when a transfer ends. If "0" is set here, the number of transfers will be 2^{24} = 16777216. A DMA interruption occurs when the number counts down to "0".
If this register is read, "0" are appended to bits [31:24].

DMA Channel [1:0] Control Resister (CTL [1:0])

DMAC1[0x0C] [0x1C] Default value = 0x00XX_XXXX Read/Write

n/a								RSV				IDLE	RSV	AM	RSV
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 [23:20]: **RSV Reserved (0)**

Bit 19: **IDLE Idle Delay Enable**

Some DMA target devices may require this setting to be enabled.
It is recommended to set this to "1" for write transfer to a device (transfer from memory to device).

0: Normal operation
1: Delays the timing for the next request from a device to be received.

Bit 18: **RSV Reserved**

Bit 17: **AM Acknowledge Mode**

Selects an output timing of DACK signal.

0: Active during DMA read cycle
1: Active during DMA write cycle

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

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

Selects an update mode for the destination address register for the update that is taken place after one transfer end.

00: Fixes the transfer destination address (no update)
01: Increments the transfer destination address according to the transferred data size
(8-bit address increments by 1; 16-bit, by 2; 32-bit, by 4)
10: Decrements the transfer destination address according to the transferred data size
(8-bit address decrements by 1; 16-bit, by 2; 32-bit, by 4)
11: Reserved

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

Selects an update mode for the source address register for the update that is taken place after one transfer ends.

00: Fixes the source destination address (no update)
01: Increments the transfer source address according to the transferred data size
(8-bit address increments by 1; 16-bit, by 2; 32-bit, by 4)
10: Decrements the transfer source address according to the transferred data size
(8-bit address decrements by 1; 16-bit, by 2; 32-bit, by 4)
11: Reserved

Bits [11:8]: **RS Resource Select [3:0]**

Selects a trigger to start DMA transfer.

0000: Reserved
0001: Reserved
0010: Reserved
0011: Reserved
0100: UART#1 output (TX)
0101: UART#1 input (RX)
0110: UART#2 output (TX)
0111: UART#2 input (RX)
1000: UART#3 output (TX)
1001: UART#3 input (RX)
1010: SPI I/O (SPI-TX/RX)

1011: Reserved
1100: Reserved
1101: Reserved
1110: Reserved
1111: SW-Request Software Request
Set bits [11:8] to "1111" to select software DMA transfer.

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

Bit 6: **RIM Request Input Mode**

Selects an input mode for DMA request signal.
0: Active LOW (level trigger)
1: Falling edge (edge trigger)

Bit 5: **TM Transfer Mode**

Selects a transfer mode for DMA transfer.
0: Single transfer (one transfer per DMA request)
1: Demand transfer (continues the transfer while the DMA request exists)

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

Selects data size to be transferred in a single transfer.
00: 8 bits
01: 16 bits
10: 32 bits
11: Reserved

Bit 2: **IE Interrupt Enable**

Permits/inhibits transfer end interrupt on the DMA channel [1:0].
0: Interrupt inhibited.
1: Interrupt permitted.

Bit 1: **TE DMA Transfer End**

0 (Read): Transferring or waiting
1 (Read): DMA transfer ended
0 (Write): Clears this bit
1 (Write): Ineffective

This bit is set when the value of the DMA channel 0 transfer count register becomes "0" as a result of DMA transfer. Once this bit is set, it preserves "1" until "0" is written to clear it. DMA transfer on this channel is inhibited until this bit is cleared. This bit also works as an interrupt source flag.

Bit 0: **DE DMA Enable**

DMA transfer on channel [1:0] is permitted based on this bit.
0: DMA transfer inhibited
1: DMA transfer permitted

DMA Channel [3:2] Control Resister (CTL [3:2])

DMAC1[0x2C] [0x3C] Default value = 0x00XX_XXXX Read/Write

n/a								RSV				IDLE	RSV	AM	RSV
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 [23:20]: **RSV Reserved (0)**

Bit 19: **IDLE Idle Delay Enable**

Some DMA target devices may require this setting to be enabled.
It is recommended to set this to "1" for write transfer to a device (transfer from memory to device).

0: Normal operation
1: Delays the timing for the next request from a device to be received.

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

Bit 17: **AM Acknowledge Mode**

Selects an output timing of DACK signal.

0: Active during DMA read cycle
1: Active during DMA write cycle

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

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

Selects an update mode for the destination address register for the update that is taken place after one transfer end.

00: Fixes the transfer destination address (no update)
01: Increments the transfer destination address according to the transferred data size (8-bit address increments by 1; 16-bit, by 2; 32-bit, by 4)
10: Decrements the transfer destination address according to the transferred data size (8-bit address decrements by 1; 16-bit, by 2; 32-bit, by 4)
11: Reserved

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

Selects an update mode for the source address register for the update that is taken place after one transfer ends.

00: Fixes the source destination address (no update)
01: Increments the transfer source address according to the transferred data size (8-bit address increments by 1; 16-bit, by 2; 32-bit, by 4)
10: Decrements the transfer source address according to the transferred data size (8-bit address decrements by 1; 16-bit, by 2; 32-bit, by 4)
11: Reserved

Bits [11:8]: **RS Resource Select [3:0]**

Selects a trigger to start DMA transfer.

0000: Reserved
0001: Reserved
0010: Reserved
0011: Reserved
0100: I^2C output (I^2C-TX)
0101: I^2C input (I^2C-RX)
0110: I^2S I/O (I^2S-CH0)
0111: I^2S I/O (I^2S-CH1)
1000: Reserved
1001: Reserved
1010: SPI I/O (SPI-TX/RX)

1011: Reserved
1100: Reserved
1101: Reserved
1110: Reserved
1111: SW-Request Software Request
Set bits [11:8] to "1111" to select software DMA transfer.

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

Bit 6: **RIM Request Input Mode**

Selects an input mode for DMA request signal.
0: Active LOW (level trigger)
1: Falling edge (edge trigger)

Bit 5: **TM Transfer Mode**

Selects a transfer mode for DMA transfer.
0: Single transfer (one transfer per DMA request)
1: Demand transfer (continues the transfer while the DMA request exists)

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

Selects data size to be transferred in a single transfer.
00: 8 bits
01: 16 bits
10: 32 bits
11: Reserved

Bit 2: **IE Interrupt Enable**

Permits/inhibits transfer end interrupt on the DMA channel [3:2].
0: Interrupt inhibited.
1: Interrupt permitted.

Bit 1: **TE DMA Transfer End**

0 (Read): Transferring or waiting
1 (Read): DMA transfer ended
0 (Write): Clears this bit
1 (Write): Ineffective

This bit is set when the value of the DMA channel 3 transfer count register becomes "0" as a result of DMA transfer. Once this bit is set, it preserves "1" until "0" is written to clear it. DMA transfer on this channel is inhibited until this bit is cleared. This bit also works as an interrupt source flag.

Bit 0: **DE DMA Enable**

DMA transfer on channel [3:2] is permitted based on this bit.
0: DMA transfer inhibited.
1: DMA transfer permitted.

DMA Channel Operating Select Resister (OPSR)

DMAC1[0x60] Default value = 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

Bit [9:8]: **DPM Priority Mode**

Determines the priority when there are transfer requests for multiple channels simultaneously.

00: CH0>CH1>CH2>CH3
01: CH1>CH0>CH3>CH2
10: CH2>CH3>CH0>CH1
11: Simple round robin mode

Bit 0: **DGE DMA Global Enable**

Enables/disables all channels for the DMA.

0: Disable
1: Enable

7. CAMERA INTERFACE [2:1] (CAM[2:1])

7.1 Description

The camera interface has the following features.

- 8-bit parallel interface × 2 ports (One of the ports is reserved for IP conversion included in S2S65A30.)
- Can support cameras whose maximum resolution size is VGA (640×480)
- 8-bit data bus interface (YUV4:2:2 format)
- Supports the format extending the ITU-R BT.656 and BT.601 types to Progressive Scan.
- Capture frames can be configured.
- Pixel clock frequency for inputting camera data is less than 2/3 of CPU clock.

7.2 Block Diagram

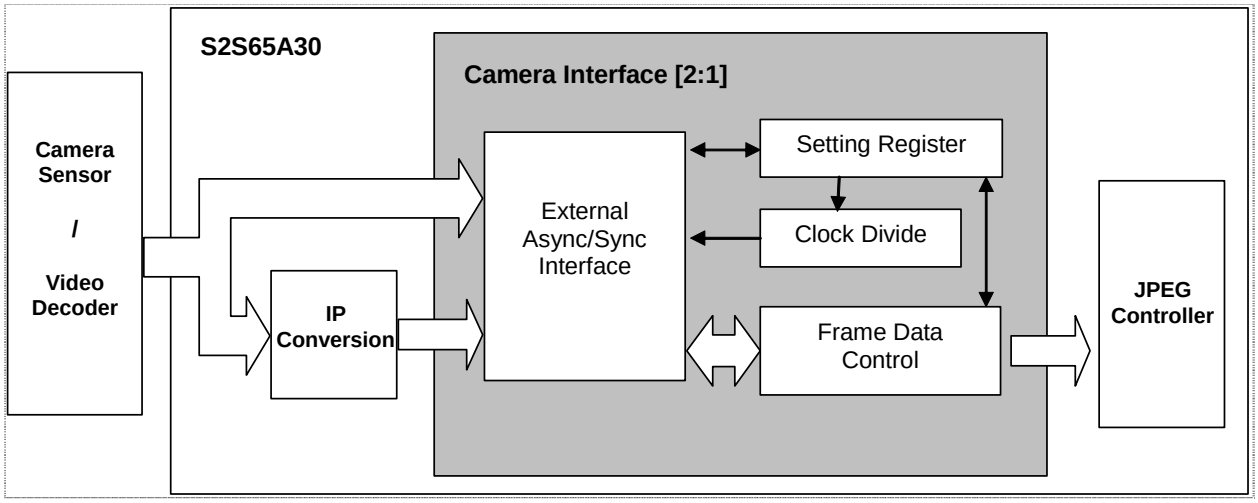


Figure 7.1 Camera Interface [2:1] Block Diagram

7.3 External Pins

The Camera Interface [2:1]-related external pins are listed below.

Table 7.1 External Pins

Pin Name	Input/Output	Pin Functions	Multiplex Pin*/Remarks
CM1DATA [7:0]	I	Camera 1 data (YUV) input	GPIOE [7:0], IPC1DATA[7:0]
CM1VREF	I	Vertical sync input at Camera 1 data input	GPIOF0, IPC1VREF
CM1HREF	I	Horizontal sync input at Camera 1 data input	GPIOF1, IPC1HREF
CM1CLKOUT	O	Basic clock output for Camera 1	GPIOF2, IPC1FIELD
CM1CLKIN	I	Pixel clock for inputting Camera 1 data	GPIOF3, IPC1CLKIN
CM2DATA [7:0]	I	Camera 2 data (YUV) input	GPIOG [7:0], IPC2DATA[7:0]
CM2VREF	I	Vertical sync input at Camera 2 data input	GPIOF4, IPC2VREF
CM2HREF	I	Horizontal sync input at Camera 2 data input	GPIOF5, IPC2HREF
CM2CLKOUT	O	Basic clock output for Camera 2	GPIOF6, IPC2FIELD
CM2CLKIN	I	Pixel clock for inputting Camera 2 data	GPIOF7, IPC2CLKIN

Note(*): External pins for the camera interface (CAM) are multiplexed with GPIO pins. To use as an external pin for CAM, set the value to “Function 1 of other than GPIO” by using the GPIO pin function register. To use IP conversion included in S2S65A30, set the value to “Function 2 of other than GPIO”.

7.4 Registers

7.4.1 List of Registers

The registers for camera interface [2:1] are listed below. The base addresses of these registers are as follows.
Camera interface 1: 0xFFFE_8000
Camera interface 2: 0xFFFD_8000

Table 7.2 List of Registers

Address Offset	Register Name	Default value	R/W	Data Access Size
Camera Interface : Base Address 0xFFFE_8000(CAM1) 0xFFFD_8000(CAM2)				
0x00	Camera [2:1] input 1 clock cycle setting register	0x0000	R/W	16bit
0x04	Camera [2:1] input 1 signal setting register	0x0000	R/W	16bit
0x08	Reserved	—	—	—
0x0C	Camera [2:1] input 2 signal setting register	0x0000	R/W	16bit
0x20	Camera [2:1] mode setting register	0x0000	R/W	16bit
0x24	Camera [2:1] frame control register	0x0000	R/W	16bit
0x28	Camera [2:1] control register	0x0000	WO	16bit
0x2C	Camera [2:1] status register	0x0004	RO	16bit
0x30 - 0x5C	Reserved	—	—	—

7.4.2 Detailed Description of Registers

Camera [2:1] Clock Cycle Setting Register							
CAM2 [0x00], CAM1 [0x00] Default value = 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 [4:0]: **Clock Frequency Select bits [4:0]**
Sets the cycle for the output clock (CMCLKOUT).

Table 7.3 Camera Clock Cycle Settings

Bits [4:0]	Clock Frequency
00000	Internal clock 1/1
00001	Internal clock 1/2 (Recommended value)
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 [2:1] Signal Setting Register							
CAM2 [0x04], CAM1 [0x04] Default value = 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 bit (0)**

If necessary, write "0" to this bit.

Bit 5: **Clock Mode Select**

Selects a clock to use for the internal operation. If an internal divided clock is used, the cycle must be set to greater than or equal to 1/2. Unlike external clock input, using an internal divided click makes it difficult to synchronize output data with the internal divided clock because of delays occur on the board and within the camera (image sensor). For stable operation, the dividing ratio must be large enough for these delays to be ignored.

0: External clock input (CMCLKIN input is used)

1: Internal divided clock (The clock set by the camera clock cycle setting register (the same clock as =CMCLKOUT) is internally used.)

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

Selects the order of YUV data that is input by byte.

Table 7.4 YUV Format Select

Bits [4:3]	YUV Format (8bit)
00	(1st) CbYCrY (last)
01	(1st) CrYCbY (last)
10	(1st) YCbYCr (last)
11	(1st) YCrYCb (last)

Bit 2: **HREF Active Select**

Sets the HREF data active level.

0: Recognizes at High level while HREF data is effective.

1: Recognizes at Low level while HREF data is effective.

Bit 1: **VREF Active Select**

Sets the VREF data active level.

0: Recognizes at High level while VREF data is effective.

1: Recognizes at Low level while VREF data is effective.

Bit 0: **Valid Input Clock Edge**

Selects the valid edge for input clock.

This is effective on both external and internal clocks selected by setting bit 5.

0: Data is read when the clock changes from Low to High.

1: Data is read when the clock changes from High to Low.

Camera [2:1] Input 2 Signal Setting Register							
CAM2 [0x0C], CAM1 [0x0C] Default value = 0x0000							Read/Write
n/a							
15	14	13	12	11	10	9	8
(RSV)	Reserved (0)	(RSV)	YUV Data Format Select bits [1:0]		HREF Active Select	VREF Active Select	(RSV)
7	6	5	4	3	2	1	0

* This register sets input from the IP conversion module included in S2S65A30. When you set output from the IP conversion module to default, this register must also be kept to default.

Bits [15:5]: **Reserved bit (0)**
If necessary, set this bit to "0".

Bits [4:3]: **YUV Data Format Select bits[1:0]**
Selects the order of YUV data that is input by byte.

Table 7.5 YUV Format Select

Bits [4:3]	YUV Format (8bit)
00	(1st) CbYCrY (last)
01	(1st) CrYCbY (last)
10	(1st) YCbYCr (last)
11	(1st) YCrYCb (last)

Bit 2: **HREF Active Select**
Sets the HREF data active level.
0: Recognizes at High level while HREF data is valid
1: Recognizes at Low level while HREF data is valid.

Bit 1: **VREF Active Select**
Sets the VREF data active level.
0: Recognizes at High level while VREF data is valid.
1: Recognizes at Low level while VREF data is valid.

Bit 0: **Reserved bit (0)**
If necessary, set this bit to "0".

Camera [2:1] Mode Setting Register

CAM2 [0x20], CAM1 [0x20] Default value = 0x0000 Read/Write

15	14	13	12	11	10	9	8
RSV (0)			RSV Camera Active Pull-down Disable	n/a	Fast Sampling Mode	RSV (0)	

7	6	5	4	3	2	1	0
ITU-R BT656 Enable	RSV (0)		Camera Port Select	Clock Output Disable	RSV (0)		Camera Module Enable

Bit [15:11]: **RSV Reserved (0)**

Bit 10: **Fast Sampling Mode**

Sets the operation that samples input data from a camera at twice as fast as normal.

0: Normal sampling
1: Fast sampling

Bit [9:8]: **RSV Reserved (0)**

Bit 7: **ITU-R BT656 Enable**

Selects input from ITU-R BT656 compliant cameras.
This mode is available only in YUV422-8-bit interface mode.

0: Normal camera mode
1: ITU-R BT656 compliant camera mode

Bits [6:5]: **RSV Reserved (0)**

Bit 4: **Camera Port Select**

Selects the camera input port.

0: Camera input 1 (Uses input from an external pin for S2S65A30)
1: Camera input 2 (Uses input from the IP conversion module for S2S65A30)

Bit 3: **Clock Output Disable**

Sets the operation of clock output (CMCLKOUT).

0: Clock output (CMCLKOUT output is used. The cycle is set by the camera cycle setting register.)
1: Clock output inhibited (CMCLKOUT is not output; the pin is fixed to Low level.)

Bits [2:1]: **RSV Reserved (0)**

Bit 0: **Camera Module Enable**

Sets an operation of camera module.
External clock output and other operations will be enabled when the setting of this bit is “enabled”.
Setting this bit to “0” stops clock supply to the camera module, and thus is effective to reduce power consumption. Read/write operation on the camera interface registers is possible even if this bit is set to “0”.

0: Camera module disabled
1: Camera module enabled

Camera [2:1] Frame Control Register

CAM2 [0x24], CAM1 [0x24] Default value = 0x0000 Read/Write

15	14	13	12	11	10	9	8
n/a							Jpeg Raw Data Capture Mode
Frame Capture Interrupt Control	Single Frame Capture Enable	Shutter Sync. Disable	Frame Sample Control bits [2:0]			Frame Capture Interrupt Polarity	Frame Capture Interrupt Enable
7	6	5	4	3	2	1	0

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

Specifies the JPEG data capture mode.

0: YUV data capture setting

1: Reserved

Bit 7: **Frame Capture Interrupt Control**

Controls frame interrupts.

If frame capture end interrupt is specified, it does not affected by the setting in bit 5.

If frame capture end interrupt is specified, it does not affected by the setting in bit 0 and interrupt is enabled.

Table 7.6 Frame Interrupt Control

Image Capture Interrupt Polarity Bit	Interrupt
0	Interrupt is generated when an effective frame is captured.
1	Interrupt is generated when a frame capture ends.

Bit 6: **Single Frame Capture Enable**

Specifies the frame capture mode.

0: Repeats capturing.

1: Stops capturing after one frame is captured when Image frame capture operation (CAM[0x28] bit 2 = 1) has been set.

Note: This bit must not be changed when Camera Module Enable (CAM[0x20] bit 0) is set to “1”.

Bit 5: **Shutter Synchronization Disable**

Inhibits shutter synchronization of the frame interrupt flag.

0: After the shutter is pressed, outputs frame interrupt status for each effective frame.

1: Always outputs frame interrupt status for each effective frame.

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

Controls frame thinning-out of camera input.

Table 7.7 Frame Thinning-out Control

Frame Sample Control Bits[2:0]	Mode
000	No thinning-out
001	Thins out to 1/2
010	Thins out to 1/3
011	Thins out to 1/4
100	Thins out to 1/5
101	Thins out to 1/6
110	Reserved
111	Reserved (Thins out all frames)

Bit 1: **Image Capture Interrupt Polarity**

Controls interrupt generation during image capture.

Table 7.8 Frame Interrupt Control

Image Capture Interrupt Polarity Bit	Interrupt
0	When the VREF data active level changes to the VREF data inactive level
1	When the VREF data inactive level changes to the VREF data active level

Bit 0: **Image Capture Interrupt Enable**

Enables interrupt during image capture.

0: Disabled

1: Enabled

Camera [2:1] Control Register

CAM2 [0x28], CAM1 [0x28] Default value = 0x0000 Write Only

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

Bit 9: **ITU-R BT656 Error Flag 1 Clear**

The state can be cleared by writing "1" to this bit.

0: No operation

1: Clears the value of error flag 1.

Bit 8: **ITU-R BT656 Error Flag 0 Clear**

The state can be cleared by writing "1" to this bit.

0: No operation

1: Clears the value of error flag 0.

Bit 3: **Frame Capture Stop**

Specifies the setting for stopping image frame capture.

Although the default after reset is the capture operation state, a camera module can be started with the capture stopped state by setting this bit along with bit 0 Camera Module Soft Reset.

0: No operation

1: Specifies the setting for stopping capture operation

Bit 2: **Frame Capture Start**

Specifies the setting for executing image frame capture.

Stops the operation after one frame is captured if Single Frame Capture has been enabled.

0: No operation

1: Specifies the setting for executing capture operation

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

The frame interrupt status from a camera is cleared by writing "1" to this bit.

0: Does not clear the frame interrupt status.

1: Clears the frame interrupt status.

Bit 0: **Camera Module Soft Reset**

Initializes the circuit section of the camera.

0: Does not initialize the circuit section of the camera.

1: Initializes the circuit section of the camera.

Note: When this bit is set to "1", only CAM [0x20] bit 0 (Camera Module Enable) is initialized to "0" and the register is not initialized.

Camera [2:1] Status Register

CAM2 [0x2C], CAM1 [0x2C] Default value = 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 Interrupt Status	n/a
7	6	5	4	3	2	1	0

Bit 9: **ITU-R BT656 Error Flag 1**

In the ITU-R BT656 mode, the state of the reference command can be determined.

0: Normal operation

1: Detects 2-bit error during reference decoding.

Bit 8: **ITU-R BT656 Error Flag 0**

In the ITU-R BT656 mode, the state of the reference command can be determined.

0: Normal operation

1: Detects 1-bit error correction during reference decoding.

Bit 6: **Camera VSYNC**

The VSYNC value from a camera module can be determined.

In the ITU-R BT656 mode, the value after decoding the reference code can be determined.

The VSYNC value is constant irrespective of the polarity setting.

0: In the vertical blank period

1: In the valid data period

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

Bit 4: **Effective Frame Status**

By reading this bit, whether a frame is effective (sampled) or ineffective (thinned out) can be determined.

0: The frame is ineffective (thinned out)

1: The frame is effective

Bit 3: **Frame Capture Busy Status**

The image frame capture status can be determined.

0: Frame capture has stopped.

1: Frame capture is in progress.

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

The setting status of image frame capture can be determined.

After the setting is set to Start, the setting automatically changes to Stop when frame capture ends, if Single Frame Capture has been enabled.

0: Frame Capture Stop setting

1: Frame Capture Start setting

Bit 1: **Frame Capture Interrupt Status**

By reading this bit, the frame interrupt status can be determined.

0: Frame interrupt has not been generated.

1: Frame interrupt has been generated.

Note: Frame interrupt is generated only when bit 7 = 1 or bit 0 = 1 in the camera frame control register (CAM[0x24]).

7.5 Explanation of Operation

The camera interface uses an 8 bit-data bus to receive image data in YUV4:2:2 format using clock. Make sure to check the AC characteristics to determine if a camera can be used or not.

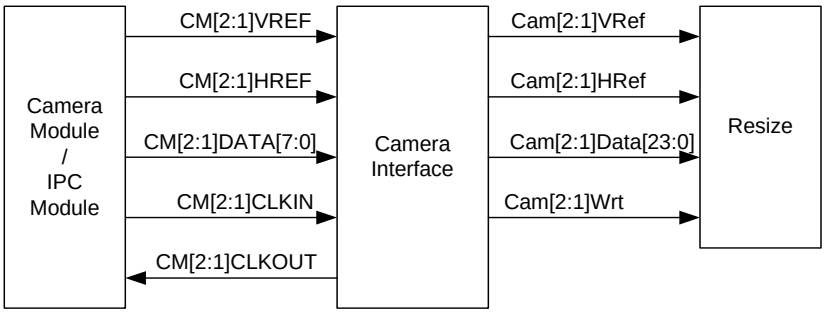


Figure 7.2 Camera Interface Connection Diagram

Table 7.9 Internal Signal

Internal Signal Name	Description
CM[2:1]VREF	Incoming vertical data enable signal from the camera module
CM[2:1]HREF	Incoming horizontal data enable signal from a camera module
CM[2:1]DATA[7:0]	Incoming 8-bit data signal from a camera module The ITU-R BT.601 format is also supported.
CM[2:1]CLKIN	Incoming pixel clock from a camera module
CM[2:1]CLKOUT	Clock that makes a camera module operate
Cam[2:1]VREF	Outgoing vertical data enable signal to the resizing circuit
Cam[2:1]HREF	Outgoing horizontal data enable signal to the resizing circuit
Cam[2:1]Data[23:0]	Signal that is converted to 24-bit after converting YUV4:2:2 to YUV4:4:4.
Cam[2:1]Wrt	Outgoing data enable signal to the resizing circuit

The camera interface synchronizes the incoming internal image processing clock and synchronous signal from the camera module with the internal image processing clock and outputs camera image data to the resizing circuit. The figure below shows a circuit that samples image data output from a camera module using CM[2:1]DATA, CM[2:1]HREF, and CM[2:1]VREF, on the rising edge of CM[2:1]CLKIN.

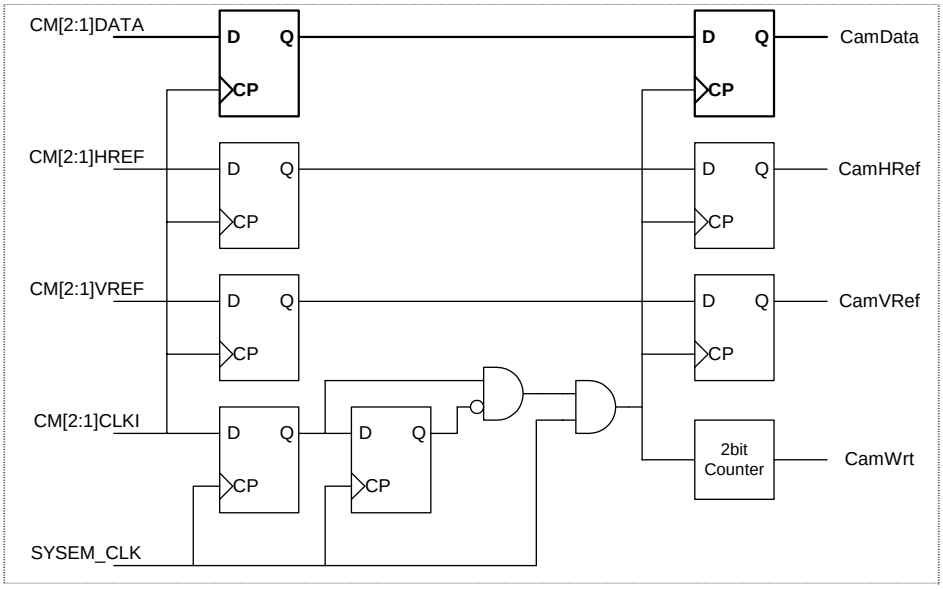


Figure 7.3 Data Sampling Circuit

In normal sampling mode (camera mode setting register CAMx[0x20] bit10=0), where an internal system clock samples the incoming clock from a camera clock module and detects clock edges, the system clock frequency must be at least twice as high as the incoming pixel clock from the camera module, in theory. In reality, whether actual operation is performed at doubled frequency or not depends on the system because there are factors such as clock duty ratio. Therefore, the frequency should at least be tripled to ensure the operation.
In fast sampling mode (camera mode setting register CAMx[0x20] bit10=1), the sampling performance is doubled because two systems of the circuit are used (just one system is used in normal sampling mode) and toggled after each cycle.

7.5.1 Frame Capture Interrupt

Interrupt can be generated from a VREF signal of camera image data. This interrupt is necessary for the image processing routine involving camera images, such as JPEG encoding.

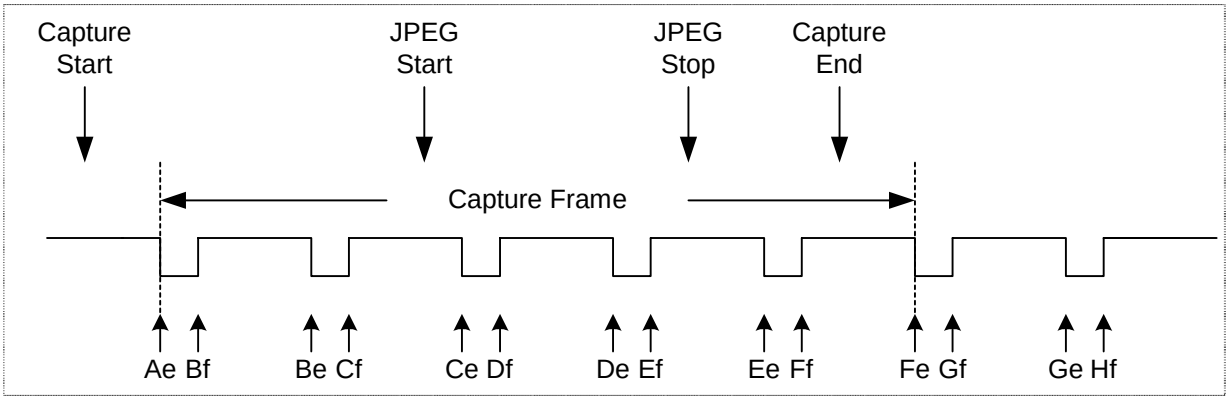


Figure 7.4 Timing of Interrupt Generation

Figure 7.4 is a timing chart that starts with enabling camera interface capture and ends with disabling it, showing actions related to a camera frame interrupt. Capture Start means that bit 0 in the camera mode setting register of the camera interface is set to “1”. Similarly, Capture End means that bit 0 in the camera mode setting register is set to “0”. JPEG Start means that “1” is written to the JPEG start/stop control register; JPEG End means that “0” is written to it. Capture Frame represents a frame where incoming data from the camera module is sent to the resizing circuit.

There are four register settings in total that determine the timing of the frame capture interrupt. Table 7.10 shows where in the figure above an interrupt is generated based on the settings of these four register bits. Usually, the fourth register setting from the top (indicated by hatching) is used.

Table 7.10 Timing of Interrupt Generation

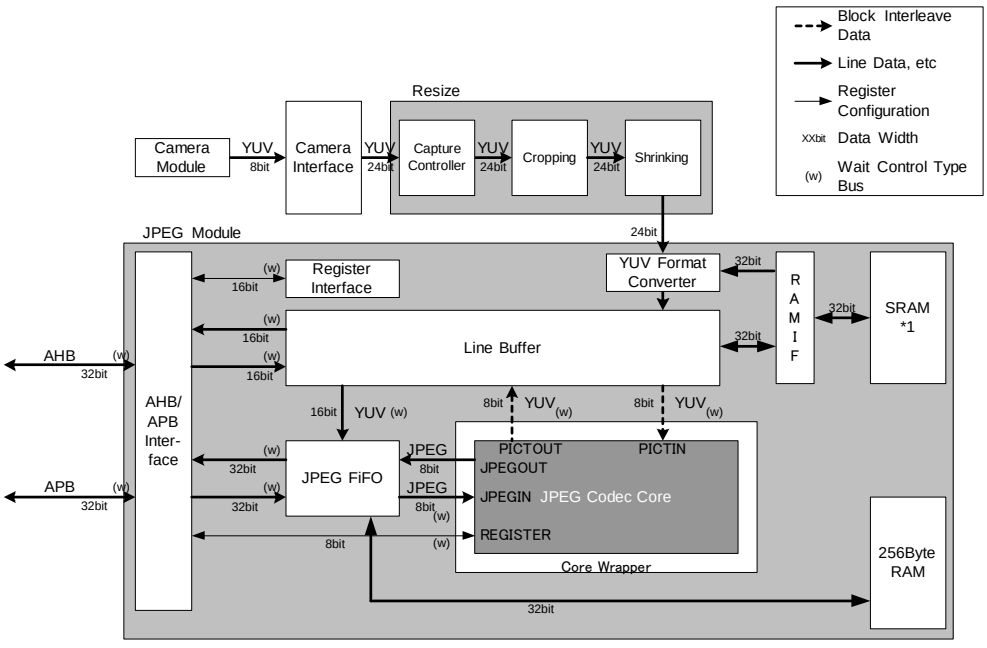
Camera Frame Control Register CAM[0x24]				Camera Frame Interrupt Generation Timing
Bit 7	**Bit 5**	**Bit 1**	**Bit 0**	
0	x	x	0	No Interrupt Generation
0	0	0	1	Ce, De
0	0	1	1	Df, Ef
0	1	0	1	Be, Ce, De, Ee, Fe
0	1	1	1	Cf, Df, Ef, Ff, Gf
1	x	0	x	Fe
1	x	1	x	Gf

8. JPEG CONTROLLER (JPG[2:1])

8.1 Description

This controller provides JPEG encoding and decoding of camera input images, as well as a function that captures camera input images as YUV data. The JPEG encoding and decoding methods nearly comply with baseline JPEG standard. The calculation accuracy complies with JPEG Part2 (ISO/IEC10918-2) and thus adequate accuracy is provided. Available maximum image size is VGA standard because of restriction of internal line buffer memory size. Images that can be JPEG-encoded/decoded are minimum unit YUV-formatted images of MCU size or larger. Two quantization tables can be configured for compression and up to four tables are supported for expansion. Each of the DC and AC Huffman tables can handle two tables. Up to 36 bytes optional markers can be inserted during encoding. Processed makers are SOI, SOF0, SOS, DQT, DHT, DRI, RSTm, and EOI, which are automatically decoded during expansion. DNL makers are not supported. Supported YUV formats are YUV 4:4:4, YUV 4:2:2, YUV 4:1:1, and YUV 4:2:0 for encoding camera images and YUV 4:2:2 and YUV 4:2:0 for capturing YUV data. Because camera interface only supports YUV 4:2:2, use the YUV 4:2:2 in usual environments. Gray images and RGB images are not supported. The processing speed for a VGA size image is 1/30 second at minimum. However, it cannot be guaranteed that this minimum speed is always achieved because processing is greatly affected by the image data, Huffman table, and quantization tables.
To support dual camera connection, S2S56A30 has two channels of JPEG controllers [2:1] that have exactly equivalent functions.

8.2 Block Diagram



*1 : Refer to bit[5:4](EMBRAMSEL[1:0]) of SYS[0x68](EMBMEMCTL) for the allocation of SRAM.

Figure 8.1 JPEG Controller Block Diagram

8.3 External Pins

There is no external pin that relates to the JPEG controller.

8.4 Registers

The following table lists the registers. Two identical channels of JPEG controllers (JPEG1 and JPEG2) are provide. Although the registers for these two channels have different base addresses, their functions are exactly same.

8.4.1 List of Registers

Table 8.1 List of Registers

Address Offset	Register Name	Default Value	R/W	Data Access Size
Resizer Operation Registers (RSZ[2:1]) : Base Address = 0xFFFE_9000(JPG1) 0xFFFD_9000(JPG2)				
0x60	Global resizer control register	0x0000	WO	16bit
0x64	Capture control state register	0x0000	RO	16bit
0x68	Capture data setting register	0x0000	R/W	16bit
0x70-0x7C	Reserved register	—	—	—
0xC0	Capture resizer control register	0x0000	R/W	16bit
0xC8	Capture resizer start X position register	0x0000	R/W	16bit
0xCC	Capture resizer start Y position register	0x0000	R/W	16bit
0xD0	Capture resizer end X position register	0x027F	R/W	16bit
0xD4	Capture resizer end Y position register	0x01DF	R/W	16bit
0xD8	Capture resizer scaling rate register	0x8080	R/W	16bit
0xDC	Capture resizer scaling mode register	0x0000	R/W	16bit
JPEG Module Registers (JCTL[2:1]) : Base Address = 0xFFFE_A000(JPG1) 0xFFFD_A000(JPG2)				
0x00	JPEG control register	0x0000	R/W	16bit
0x04	JPEG status flag refresh register	0x8080	R/W	16bit
0x08	JPEG raw status flag register	0x8080	RO	16bit
0x0C	JPEG interrupt control register	0x0000	R/W	16bit
0x10	Reserved register	—	—	—
0x14	JPEG codec start stop control register	0x0000	WO	16bit
0x18 - 0x1C	Reserved register	—	—	—
0x20	Huffman table auto setting register	0x0000	R/W	16bit
JPEG FIFO Setting Registers (JFIFO[2:1]) : Base Address = 0xFFFE_A000(JPG1) 0xFFFD_A000(JPG2)				
0x40	JPEG FIFO control resister	0x0000	R/W	16bit
0x44	JPEG FIFO status register	0x8001	RO	16bit
0x48	JPEG FIFO size register	0x003F	R/W	16bit
0x4C	JPEG FIFO read/write port resister	0x0000 0000	R/W	32bit
0x50 - 0x58	Reserved register	—	—	—
0x60	Encode size limit register 0	0x0000	R/W	16bit
0x64	Encode size limit register 1	0x0000	R/W	16bit
0x68	Encode size result register 0	0x0000	RO	16bit
0x6C	Encode size result register 1	0x0000	RO	16bit
0x70 - 0x78	Reserved register	—	—	—
JPEG Line Buffer Setting Registers (JLB[2:1]) : Base Address = 0xFFFE_A000(JPG1) 0xFFFD_A000(JPG2)				
0x80	JPEG line buffer status flag register	0x0000	R/W	16bit
0x84	JPEG line buffer raw status flag register	0x0000	RO	16bit
0x88	JPEG line buffer current status flag register	0xX009	RO	16bit
0x8C	JPEG line buffer interrupt control register	0x0000	R/W	16bit
0x90 - 0x9C	Reserved register	—	—	—

Address Offset	Register Name	Default Value	R/W	Data Access Size
0xA0	JPEG line buffer horizontal pixel acceptable size register	0x2800	R/W	16bit
0xA4	JPEG line buffer memory address offset register	0x0020	R/W	16bit
0xA8 - 0xBC	Reserved register	—	—	—
0xC0	JPEG line buffer read/write port resister	0x0000 0000	R/W	32bit
JPEG Codec Registers (JCODEC[2:1]) : Base Address = 0xFFFE_B000(JPG0) 0xFFFD_B000(JPG1)				
0x00	Operation mode setting register	0x0000	R/W	16bit
0x04	Command setting register	Not applicable	WO	16bit
0x08	JPEG operation status register	0x0000	RO	16bit
0x0C	Quantization table number register	0x0000	R/W	16bit
0x10	Huffman table number register	0x0000	R/W	16bit
0x14	DRI setting register 0	0x0000	R/W	16bit
0x18	DRI setting register 1	0x0000	R/W	16bit
0x1C	Vertical pixel size register 0	0x0000	R/W	16bit
0x20	Vertical pixel size register 1	0x0000	R/W	16bit
0x24	Horizontal pixel size register 0	0x0000	R/W	16bit
0x28	Horizontal pixel size register 1	0x0000	R/W	16bit
0x2C - 0x34	Reserved register	—	—	—
0x38	RST marker operation setting register	0x0000	R/W	16bit
0x3C	RST marker operation status register	0x0000	RO	16bit
0x40 - 0xCC	Inserted marker data register	0x00FF	R/W	16bit
0x400 - 0x4FC	Quantization table No.0 register	Not applicable	R/W	16bit
0x500 - 0x5FC	Quantization table No.1 register	Not applicable	R/W	16bit
0x800 - 0x83C	DC Huffman table No.0 register 0	Not applicable	WO	16bit
0x840 - 0x86C	DC Huffman table No.0 register 1	Not applicable	WO	16bit
0x880 - 0x8BC	AC Huffman table No.0 register 0	Not applicable	WO	16bit
0x8C0 - 0xB44	AC Huffman table No.0 register 1	Not applicable	WO	16bit
0xC00 - 0xC3C	DC Huffman table No.1 register 0	Not applicable	WO	16bit
0xC40 - 0xC6C	DC Huffman table No.1 register 1	Not applicable	WO	16bit
0xC80 - 0xCBC	AC Huffman table No.1 register 0	Not applicable	WO	16bit
0xCC0 - 0xF44	AC Huffman table No.1 register 1	Not applicable	WO	16bit

Details of these registers are described in the following sections.

8.4.2 Resizer Operation Registers (RSZ[2:1])

Note: Most of the resizer registers cannot be changed while receiving data from a camera interface.

Global Resizer Control Register

RSZ[0x60] Default value = 0x0000 Write Only

15	14	13	12	11	10	9	8
n/a					Reserved		ACTAGAIN
7	6	5	4	3	2	1	0
n/a			Reserved		n/a	Reserved	

Bit [10:9]: **Reserved**
Be sure to set each bit to “0”.

Bit 8: **ACTAGAIN (Write only)**
This bit is used when continuous encoding is performed. If you write “1” to this bit, the frame that follows the current frame is also sent to the JPEG codec circuit.
Whether continuous encoding is possible or not depends on the system and software specifications. For details, see Explanation of Operation.

Bit [4:3]: **Reserved**
Be sure to set each bit to “0”.

Bits [1:0]: **Reserved**
Be sure to set each bit to “0”.

Capture Control State Register

RSZ[0x64] Default value = 0x0000 Read Only

15	14	13	12	11	10	9	8
n/a							
7	6	5	4	3	2	1	0
n/a				State Value			

Bit [3:0]: **State Value**
Indicates the value for the current state of the capture control sequence state machine. For the meanings of state values, see Explanation of Operation, “8.5.1 Capture Control”.

Capture Data Setting Register

RSZ[0x68] Default value = 0x0000 Read/Write

15	14	13	12	11	10	9	8
n/a							
7	6	5	4	3	2	1	0
n/a							Data Format Select

Bit 0: **Captured Image Data Format Select**
Selects a captured image data format. If this bit is set to “1”, all settings in RSZ[*] registers except RSZ[0xC0] are disabled. (The statuses of some Read-Only registers may change but the values are not effective.)

0: YUV data
1: Reserved (JPEG data)

Reserved Register

RSZ[0x70-7C] Default value = 0x0000 Read/Write

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

Bit [15:0]: **Reserved**

Be sure to set each bit to “0”.

Capture Resizer Control Register

RSZ[0xC0] Default value = 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)**

By writing “1” to this bit, the capture resizer is software-reset. Nothing happens when “0” is written to it.

Bits [3:1]: **Reserved**

Be sure to set each bit to “0”.

Bit 0: **Capture Resizer Enable**

Setting this bit to “0” stops clock supply to the resize module, and thus is effective to reduce power consumption. Read/write operation on the resize module registers is possible even if this bit is set to “0”.

Write:

0: Capture resizer is disabled.

1: Capture resizer is enabled.

Read:

0: Capture resizer has been disabled.

1: Capture resizer has been enabled.

Capture Resizer Start X Position Register

RSZ[0xC8] Default value = 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 [10:0]: **Capture Resizer Start X Position [10:0]**

These bits specify the X position where the capturing of the capture resizer starts.

Capture Resizer Start Y Position Register

RSZ[0xCC] Default value = 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 [10:0]: **Capture Resizer Start Y Position [10:0]**

These bits specify the Y position where the capturing of the capture resizer starts.

Capture Resizer End X Position Register

RSZ[0xD0] Default value = 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 [10:0]: **Capture Resizer End X Position [10:0]**

These bits specify the X position where the capturing of the capture resizer ends.

Capture Resizer End Y Position Register

RSZ[0xD4] Default value = 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 [10:0]: **Capture Resizer End Y Position [10:0]**

These bits specify the Y position where the capturing of the capture resizer ends.

Capture Resizer Scaling Rate Register

RSZ[0xD8] Default value = 0x8080 Read/Write

15	14	13	12	11	10	9	8
Reserved (0)							

7	6	5	4	3	2	1	0
Reserved (0)				Capture Resizer Scaling Rate bits [3:0]			

Bits [15:4]: **Reserved**

Be sure to set each bit to "0".

Bits [3:0]: **Capture Resizer Scaling Rate [3:0]**

These bits specify the capture resizer scaling rate. Some of the scaling rates cannot be used depending on the setting in the capture resizer scaling mode register. For details, see "*Table 8.4 Capture Resizer Scaling Rate/Mode Selection*", which is a part of the description on the capture resizer scaling mode register.

Table 8.2 Capture Resizer Scaling Rate Settings

bits [3:0]	Capture Resizer Scaling Rate Settings
0000	Reserved
0001	1/1
0010	1/2
0011	1/3 (Only in vertical thinning-out mode, capture resizer scaling mode register bit 1-0=01)
0100	1/4
0101	1/5 (Only in vertical thinning-out mode, capture resizer scaling mode register bit 1-0=01)
0110	1/6 (Only in vertical thinning-out mode, capture resizer scaling mode register bit 1-0=01)
0111	1/7 (Only in vertical thinning-out mode, capture resizer scaling mode register bit 1-0=01)
1000	1/8
1001-1111	Reserved

Capture Resizer Scaling Mode Register

RSZ[0xDC] Default value = 0x0000 Read/Write

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

Bits [3:2]: **Reserved**

Be sure to set each bit to "0".

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

These bits specify the scaling mode of the capture resizer. Depending on the scaling mode, some of the scale rates cannot be set in the capture resizer scaling rate register. For details, 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 scaling
01	Thinning-out in both vertical and horizontal directions
10	Thinning-out in vertical direction, averaging in horizontal direction
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[2:1])

JPEG Control Register

JCTL[0x00] Default value = 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 Encoding Fast Mode**

0: No impact.

1: Expedites JPEG compression process by using the fixed Huffman table.

Note: When JPEG compression is performed with this bit set to "1", the JPEG Huffman table value (effective address in JCODEC[0x800]-[0xF44]) must be changed to one listed in "ISO/IEC 10918-1 Annex K" or the JCTL[0x20] Huffman auto setting register must be set to "1".

Bit14: **JPEG Marker Fast Output Mode**

0: No impact.

1: Expedites JPEG maker output by using the fixed Huffman table.

Note: The setting of this bit is effective only when the encode fast mode (Bit15) is set to "1". When JPEG compression is performed with this bit set to "1", you do not have to write the JPEG Huffman table value (effective address in JCODEC[0x800]-[0xF44]). The one listed in "ISO/IEC 10918-1 Annex K" will be used.)

Bits [13:9]: **Reserved**

Be sure to set each bit to "0".

Bit 8: **JPEG 180° Rotation Enable**

Specifies whether the JPEG encode data should be rotated. Because the hardware rotates the data on a specific line basis, you may need to rearrange the data in the JPEG file by using the software. For details, see "*8.5.5.5 JPEG 180°Rotation Encode*"

0: No rotation

1: Rotates by 180°

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

Resets JPEG modules except JPEG codec. The registers are not reset. Be sure to reset the modules before starting JPEG encoding. To execute this reset, JPEG Module Enable must be set to "1". If the JPEG modules have not been enabled, the software reset may not be executed even if "1" is written to this bit.

0: No impact

1: Reset

Bit [6:5]: **Reserved**

Be sure to set each bit to "0".

Bit 4: **UV Data Type Conversion**

Converts the UV type input from a camera.

Table 8.5 UV Data Type

Bit 4	Camera Data Type	Internal Data Type
0 (Converts)	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 (Not Convert)	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 [3:1] : **Operation Mode Select**

Specifies the JPEG operation mode. When YUV data capture is selected, clock supply to the JPEG codec stops and the JPEG codec register cannot be accessed. Therefore, if you change the setting of the JPEG codec register, camera image JPEG encode (other than YUV data capture) should be set.

While the YUV format for camera image JPEG encoding/decoding is set by JCODEC[0x00]bit1-0YUV format select, the YUV format for YUV data capture is set by this resister bit.

In case of VGA image with YUV4:2:0 format, more than 31KB memory is needed as line-buffer. Then only one channel can operate with YUV4:2:0 format, due to S2S65A30's internal memory limitation.

Table 8.6 JPEG Operation Mode

bits [3:1]	JPEG Operation Mode
000	Camera image JPEG encode (YUV4:4:4, YUV4:2:2, YUV4:1:1, and YUV4:2:0. YUV4:4:4 supports images resized to less than 1/2 only.) (Default value)
001	Reserved
010	Reserved
011	YUV data capture (YUV 4:2:2)
100	JPEG decoding (YUV 4:2:2)
101	Reserved
110	Reserved
111	YUV data capture (YUV 4:2:0)

Bit 0: **JPEG Module Enable**

Enables the JPEG modules. When the modules are disabled, clock supply to the JPEG modules stops and the JPEG codec register cannot be accessed. Be sure to disable the JPEG modules before disabling the resizer.

0: Disables (Default value)

1: Enables

JPEG Status Flag Register

JCTL[0x04] Default value = 0x8080 Read/Write

Reserved (1) —	JPEG Codec File Out Status (RO)	JPEG FIFO Threshold Status bits [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) —			Decode Maker Read 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: **Reserved**

Be sure to set "1".

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

Shows the JPEG codec output state during encoding.

0: JPEG file output has been stopped

1: JPEG encoding or JPEG file output is in progress.

Be sure to set "1" when a write operation is performed.

Bits [13:12]: **JPEG FIFO Threshold Status (Read only)**

Shows the current data status of the JPEG FIFO.

Be sure to set "1" when a write operation is performed.

Table 8.7 JPEG FIFO Threshold Status

bits [13:12]	JPEG FIFO Threshold Status
00	Empty
01	Greater than or equal to 4 bytes and less than 1/4 of the FIFO size
10	Greater than or equal to 1/4 and less than 1/2 of the FIFO size
11	Greater than or equal to 1/2 of the FIFO size

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

This is an interrupt flag that indicates that, during JPEG encoding, the JPEG file size has exceeded the encode size limits set in the encode size limit register. This flag can be disabled by using the JPEG interrupt control register (JCTL[0x0C] bit 11). This function just reports that the file size has exceeded the limits and the JPEG encoding will continue.

Read:

0: No Interrupt

1: Interrupt exists (encode was oversized)

Write:

0: No impact

1: Clear

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

This is an interrupt flag that indicates that the data size of the JPEG FIFO has exceeded the JPEG FIFO threshold (JPEG FIFO control register bits 5-4) at least once. This flag can be disabled by using the JPEG interrupt control register (bit 10).

Read:

0: No Interrupt

1: Interrupt exists (JPEG FIFO threshold was violated)

Write:

0: No impact

1: Clear

Bit 9: **JPEG FIFO Full Flag**

This is an interrupt flag that indicates that the JPEG FIFO has been full at least once. This flag can be disabled by using the JPEG interrupt control register (bit 9).

Read:

0: No Interrupt
1: Interrupt exists (JPEG FIFO full occurred)

Write:

0: No impact
1: Clear

Bit 8: **JPEG FIFO Empty Flag**

This is an interrupt flag that indicates that the JPEG FIFO has been empty at least once. This flag can be disabled by using the JPEG interrupt control register (bit 8).

Read:

0: No Interrupt
1: Interrupt exists (JPEG FIFO empty occurred)

Write:

0: No impact
1: Clear

Bit [7:5]: **Reserved**

Be sure to set each bit to "1".

Bit 4: **Decoding Marker Flag**

This is an interrupt flag that indicates that the marker in a JPEG file is read during JPEG decoding.

Read:

0: No interrupt has occurred
1: An interrupt has occurred (the marker has been read)

Write:

0: No impact
1: Clear

To clear the interrupt, disable bit 4 of JPEG Interrupt Control Register previously.

Bit 3: **Reserved**

Be sure to set "1".

Bit 2: **JPEG Buffer Interrupt Flag (Read only)**

This is an interrupt flag that indicates that the JPEG line buffer has overflowed. This flag can be disabled by using the JPEG interrupt control register (bit 2).

This flag is cleared by the setting of JPEG Module Software Reset (bit 7).

0: No Interrupt
1: Interrupt exists (JPEG line buffer overflow occurred)

Be sure to set "1" when a write operation is performed.

This function just reports that the line buffer has overflowed and the JPEG modules is continuing the operation. However, the data has been destroyed by the overflow.

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

This is an interrupt flag that indicates that the JPEG codec has generated an interrupt. This flag can be disabled by using the JPEG interrupt control register (bit 1). This flag is cleared by reading the JPEG operation status (bit 0).

0: No Interrupt
1: Interrupt exists (JPEG codec interrupt occurred)

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

This is a flag that is used during JPEG decoding, and indicates that a JPEG line buffer interrupt has been occurred. This flag can be disabled by JPEG Interrupt Control Register (bit 0). This flag is cleared by clearing JPEG Line Buffer Status Flag Register.

0: No interrupt has occurred
1: An interrupt (JPEG line buffer interrupt) has occurred

JPEG raw status flag register

JCTL[0x08] Default value = 0x8080 Read Only

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

Bit 15: **Reserved**

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

Shows the JPEG codec output state during encoding.

0: JPEG file output has been stopped
1: JPEG encoding or JPEG file output is in progress.

Bits [13:12]: **Raw JPEG FIFO Threshold Status**

Shows the current data status of the JPEG FIFO.

Table 8.8 JPEG FIFO Threshold Status

Bits [13:12]	JPEG FIFO Threshold Status
00	Empty
01	Greater than or equal to 4 bytes, less than FIFO size
10	Greater than or equal to 1/4 and less than 1/2 of the FIFO size
11	Greater than or equal to 1/2 of the FIFO size

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

This is a flag that indicates that, during JPEG encoding, the JPEG file size has exceeded the encode size limits (encode size limit register 0, 1). This flag is not affected by the JPEG interrupt control register (bit 11). Write "1" to the JPEG status flag register bit 11 to clear this flag. This function just reports that the file size has exceeded the limits and the JPEG encoding will continue.

0: No oversized encoding occurred
1: Oversized encoding occurred

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

This is an flag that indicates that the data size of the JPEG FIFO has exceeded the JPEG FIFO threshold (JPEG FIFO control register bits 5-4) at least once. This flag is not affected by the JPEG interrupt control register (bit 10). Write "1" to the JPEG status flag register bit 10 to clear this flag.

0: No JPEG FIFO threshold violation occurred
1: JPEG FIFO threshold violation occurred

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

This is a flag that indicates that the JPEG FIFO has been full at least once. This flag is not affected by the JPEG interrupt control register (bit 9). Write "1" to the JPEG status flag register bit 9 to clear this flag.

0: No JPEG FIFO full occurred
1: JPEG FIFO full occurred

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

This is a flag that indicates that the JPEG FIFO has been empty at least once. This flag is not affected by the JPEG interrupt control register (bit 8). Write "1" to the JPEG status flag register bit 8 to clear this flag.

0: No JPEG FIFO empty occurred
1: JPEG FIFO empty occurred

Bit [7:5]: **Reserved**

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

This is a flag that indicates that the marker in a JPEG file is read during JPEG decoding. While JPEG Interrupt Control Register (bit 4) is set to "0", the value of this flag does not change from "0" to "1". On the other hand, if this flag has been set to "1", the value does not change to "0" even if the setting of JPEG Interrupt Control Register (bit 4) is changed from "1" to "0". This flag is cleared by setting bit 4 of JPEG Status Flag Register to "1".

0: The marker has not yet been read
1: The marker has been read

Bit 3: **Reserved**

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

This is a flag that indicates that the JPEG line buffer has overflowed. This flag is not affected by the JPEG interrupt control register (JCTL[0x0C] bit 2). This flag is cleared by the setting of JPEG Module Software Reset (JCTL[0x00] bit 7).

0: No JPEG line buffer overflow occurred
1: JPEG line buffer overflow occurred

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

This is a flag that indicates that the JPEG codec has generated an interrupt. This flag is not affected by the JPEG interrupt control register (bit 1). This flag is cleared by reading the JPEG operation status register (bit 0).

0: No JPEG codec interrupt generated
1: JPEG codec interrupt generated

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

This is a flag is used in YUV data capture and indicates that a JPEG line buffer interrupt has been generated. This flag is not affected by the JPEG interrupt control register (bit 0). This flag is cleared by clearing the JPEG line buffer interrupt flag register.

0: No JPEG line buffer interrupt generated
1: JPEG line buffer interrupt generated

JPEG Interrupt Control Register

JCTL[0x0C] Default value = 0x0000 Read/Write

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

Bits [15:12]: **Reserved**

Be sure to set each bit to "0".

Bit 11: **Encode Size Limit Violation Interrupt Enable**

Enables encode size limit violation interrupts.

0: Disabled (default value)
1: Enabled

Bit 10: **JPEG FIFO Threshold-Triggered Interrupt Enable**

Enables JPEG FIFO threshold-triggered interrupts.

0: Disabled (default value)
1: Enabled

Bit 9: **JPEG FIFO-Full Interrupt Enable**

Enables JPEG FIFO-full interrupts.

0: Disabled (default value)
1: Enabled

Bit 8: **JPEG FIFO Empty Interrupt Enable**

Enables JPEG FIFO empty interrupts.

0: Disabled (default value)
1: Enabled

Bits [7:5]: **Reserved**

Be sure to set each bit to "0".

Bit 4: **JPEG Decoding Marker Read Interrupt Enable**

Enables JPEG decoding marker read interrupts. It can also pause JPEG decoding operation in a decoding marker read state. The pause state is released by disabling this bit.

0: Disabled (Default)
1: Enabled

Bit 3: **Reserved**

Be sure to set "0".

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

Enables JPEG line buffer overflow interrupts.

0: Disabled (default value)
1: Enabled

Bit 1: **JPEG Codec Interrupt Enable**

Enables JPEG Codec interrupts.

0: Disabled (default value)
1: Enabled

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

Enables JPEG line buffer interrupts. This bit serves as a basis for controlling enable-conditions of the JPEG line buffer interrupt control register. If this bit is disabled, enabling any bit of the JPEG line buffer interrupt control register does not cause JPEG line buffer interrupts that are controlled by the JPEG line buffer interrupt control register to be enabled. This control bit is not related to the JPEG line buffer overflow interrupt.

0: Disabled (default value)
1: Enabled

JPEG Codec Start/Stop Control Register

JCTL[0x14] Default value = 0x0000 Write Only

15	14	13	12	11	10	9	8
n/a							

7	6	5	4	3	2	1	0
n/a							JPEG Start/Stop Control

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

Controls the operation of the JPEG module (including YUV data capture). These bits are not used in JPEG decoding.

For JPEG encoding

0: Cancel encoding (cancel the initiation of encoding if encoding process has not been applied)
1: Initiate encoding (encode the next frame)

On YUV data capture

0: Stops capture (stops capture after current frame)
1: Starts capture (starts capture on the subsequent frames)

Huffman Table Auto Setting Register

JCTL[0x20] Default value = 0x0000 Read/Write

15	14	13	12	11	10	9	8
n/a							

7	6	5	4	3	2	1	0
n/a						Huffman Table Auto Setting Non-Standby Mode	Huffman Table Auto Setting

Bit 1: **Non-Standby Mode that is triggered when automatically setting Huffman tables**

By writing a “1” in both of Bit 0 and Bit 1 with the JPEG Codec Core set to Encode, it is possible to access to registers other then the JCODEC Registers.

By writing a “1” in Bit 0 and a “0” at the same time, it is possible to access to other registers after setting up Huffman tables.

This bit is valid only when it is simultaneously written on write of “1” to Bit 0.

“0” is read.

Bit 0: **Huffman Table Auto Setting**

With the JPEG Codec core being configured for encoding, writing a 1 to this bit causes the value defined in Annex K of ISO/IEC 10918-1 to be automatically loaded into the JPEG Huffman table value (valid addresses between JCODEC[0x800] and [0xF44).

Nothing happens at times except when setting encoding operations and when “0” is written to it.

When this bit is set to “1”, JCODEC Register accesses are prohibited. (It is regarded as a dummy write.)

When the bit is active, a write access to the register address is prohibited.
This bit will be automatically reset to "0" upon completion of Huffman table setting after a "1" is written.

8.4.4 JPEG FIFO Setting Register (JFIFO[2:1])

JPEG FIFO Control Resister

JFIFO[0x40] Default value = 0x0000 Read/Write

15	14	13	12	11	10	9	8
Reserved (0)							

7	6	5	4	3	2	1	0
Reserved (0)		JPEG FIFO Trigger Threshold bits [1:0]		Reserved (0)	JPEG FIFO Clear	JPEG FIFO Direction (RO)	Reserved (0)

Bits [15:6]: **Reserved**
Be sure to set each bit to "0".

Bits [5:4]: **JPEG FIFO trigger threshold**
Set the threshold that triggers JPEG FIFO.

Table 8.9 JPEG FIFO Trigger Threshold Selection

Bits [5:4]	JPEG FIFO Trigger Threshold
00	Not triggered
01	Triggered when the size exceeds 4 bytes.
10	Triggered when the value exceeds 1/4 of the FIFO size.
11	Triggered when the value exceeds 1/2 of the FIFO size.

Bit 3: **Reserved**
Be sure to set "0".

Bit 2: **JPEG FIFO Clear**
Clears JPEG FIFOs. IF a JPEG FIFO is cleared, be sure to reset the JPEG module (bit 7 of JPEG Control Register).
0: No effect.
1: Clear JPEG FIFO

Bit 1: **JPEG FIFO Codec Direction (Read only)**
Indicates the direction of JPEG FIFO.
0: Receive (set JPEG encoding)
1: Send (set JPEG Decoding)

Bit 0: **Reserved**
Be sure to set "0".

JPEG FIFO Status Register

JFIFO[0x44] Default value = 0x8001 Read Only

Reserved							
15	14	13	12	11	10	9	8
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 [15:4]: **Reserved**

Bits [3:2]: **JPEG FIFO Threshold Status**

Indicates the current data status of the JPEG FIFO.

Table 8.10 JPEG FIFO Threshold Status

Bits [3:2]	JPEG FIFO Threshold Status
00	Empty
01	Greater than or equal to 4 bytes and less than 1/4 of the FIFO size
10	Greater than or equal to 1/4 and less than 1/2 of the FIFO size
11	Greater than or equal to 1/2 of the FIFO size

Bit 1: **JPEG FIFO Full Status**

Indicates that the JPEG FIFO is full.

0: Not full.
1: Full.

Bit 0: **JPEG FIFO Empty Status**

Indicates that the JPEG FIFO is empty.

0: Not empty.
1: Empty.

JPEG FIFO Size Register

JFIFO[0x48] Default value = 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: **Reserved**

Be sure to set “0”.

Bits [14:0]: **JPEG FIFO Size**

Specify the size of JPEG FIFO in words. The maximum size of JPEG FIFO is 64 words. Since the JPEG FIFO has a dedicated RAM, the user usually specify 64 words, or the maximum size.
JPEG FIFO Size (words) = JPEG FIFO Size bits[14:0]+1

Note: This register can be set to “0x003F” at maximum. We recommended you use “0x003F” (Default). It can also be set to “0x001F”, “0x000F”, “0x0007”, “0x0003”, “0x0001”, and “0x0000”, but do not set it to other values.

JPEG FIFO Read/Write Port Register

JFIFO[0x4C] Default value = 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 [31:0]: **JPEG FIFO Read/Write Port**

These bits are the JPEG FIFO read port during JPEG encoding and YUV data capturing (a source of DMA transfer of JPEG files and YUV data). During JPEG decoding, JPEG FIFO becomes the write port (a destination of DMA transfer of JPEG files).

To configure this port as a source or destination of DMA transfer, set 0xE000_004C(JPEG1) or 0xD000_004C(JPEG2), an address viewed from the AHB bus of this port.

Reserved Register

JFIFO[0x50, 0x54, 0x58] Default value = — —/—

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 value = 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 value = 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]

Specify the limit on the size of data in a JPEG file for JPEG encoding in bytes. When a JPEG file whose size is larger than the setting in this register is encoded, an encoding size limit violation interrupt occurs. This will not affect the JPEG encoding process itself. These bits are not used in JPEG decoding.

Encode Size Result Register 0

JFIFO[0x68] Default value = 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 value = 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]

Indicate the size of a JPEG file in JPEG encoding. It gives the correct size only after the JPEG encoding process is completed. These bits are not used in JPEG decoding.

Reserved Register

JFIFO[0x70, 0x74, 0x78] Default value = — —/—

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 Register (JLB[2:1])

JPEG Line Buffer Status Flag Register

JLB[0x80] Default value = 0x0000 Read/Write

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

Bit 4: **Reserved**

Be sure to set “0”.

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

This is an interrupt flag that indicates that the JPEG line buffer has reached an empty state at least once. This flag can be disabled by using the JPEG line buffer interrupt control register (bit 3).

Read:

0: Interrupt disabled

1: Interrupt enabled (JPEG line buffer empty state occurred)

Write:

0: No effect.

1: Clear.

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

This is an interrupt flag that indicates that the JPEG line buffer has been full at least once. This flag can be disabled by using the JPEG line buffer interrupt control register (bit 2).

Read:

0: Interrupt disabled

1: Interrupt enabled (JPEG line buffer full state occurred)

Write:

0: No effect.

1: Clear.

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

This is an interrupt flag that indicates that the JPEG line buffer has been half full at least once. This flag can be disabled by using the JPEG Line Buffer Interrupt Control Register (bit 1).

Read:

0: Interrupt disabled

1: Interrupt enabled (JPEG line buffer half full state occurred)

Write:

0: No effect.

1: Clear.

Bit 0: **Reserved**

Be sure to set “1”.

JPEG Line Buffer Row Status Flag Register

JLB[0x84] Default value = 0x0000 Read Only

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

Bit 4: **Reserved**

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

This is a flag that indicates that the JPEG line buffer has been empty at least once. This flag is not affected by the JPEG line buffer interrupt control register (bit 3).

0: No JPEG line buffer empty state occurred.
1: JPEG line buffer empty state occurred.

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

This is a flag that indicates that the JPEG line buffer has been full at least once. This flag is not affected by the JPEG line buffer interrupt control register (bit 2).

0: No JPEG line buffer full state occurred.
1: JPEG line buffer full state occurred

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

This is a flag that indicates that the JPEG FIFO has been half full at least once. This flag is not affected by the JPEG line buffer interrupt control register (bit 1).

0: No JPEG line buffer half full state occurred.
1: JPEG line buffer half full state occurred

Bit 0: **Reserved**

JPEG Line Buffer Current Status Flag Register

JLB[0x88] Default value = 0xX009 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**

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

Indicates whether or not the JPEG line buffer is empty.

0: Not empty.
1: Empty.

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

Indicates whether or not the JPEG line buffer is full.

0: Not full.
1: Full.

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

Indicates whether or not the JPEG line buffer is half full.

0: Not half full.

1: Half full.

Bit 0: **Reserved**

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

Bit4: **Reserved**
Be sure to set “0”.

Bit 3: **JPEG Line Buffer Empty Interrupt Enable**
Enables JPEG line buffer empty interrupts.
0: Disabled (Default value)
1: Enabled

Bit 2: **JPEG Line Buffer Full Interrupt Enable**
Enables JPEG line buffer full interrupts.
0: Disabled (Default value)
1: Enabled

Bit 1: **JPEG Line Buffer Half Full Interrupt Enable**
Enables JPEG line buffer half full interrupts.
0: Disabled (Default value)
1: Enabled

Bit 0: **Reserved**
Be sure to set “0”.

JPEG Line Buffer Horizontal Pixel Allowable Size Register							
JLB[0xA0] Default value = 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 [15:4]: **JPEG Line Buffer Horizontal Pixel Allowable Size (read only)**
Represent the horizontal allowable size specified by bits [2:0].

Bits [2:0]: **JPEG Line Buffer Horizontal Pixel Allowable Size Setting**
Specify the horizontal pixel size that is accepted by the JPEG line buffer. When using this register with a value other than its default value, it is required to modify the setting in JLB[0xA4] and in the appropriate memory allocation register in the system controller.
000: Allows up to 640 horizontal pixels (default value).
001: Allows up to 800 horizontal pixels.
010: Allows up to 1024 horizontal pixels.
011: Allows up to 1280 horizontal pixels.
100: Allows up to 1600 horizontal pixels.
101-111: Not available for setting.

JPEG Line Buffer Memory Address Offset Register

JLB[0xA4] Default value = 0x0020 Read/Write

15	14	13	12	11	10	9	8
Reserved (0)							

7	6	5	4	3	2	1	0
Reserved (0)	JPG-LB Memory Address Offset bits [6:0]						

Bits [15:7]: **Reserved**

Be sure to set “0”.

Bits [6:0]: **JPEG Line Buffer Memory Address Offset**

Specify the address offset of the internal memory that is used by the JPEG line buffer in multiples of 1 KB. When using this register with a value other than its default value, it is required to modify the appropriate memory allocation register in the system controller.

JPEG Line Buffer Read/Write Port Register

JLB[0xC0] Default value = 0x0000_0000 Read/Write

31	30	29	28	27	26	25	24
JPEG Line Buffer Read/Write Port bits [31:24]							

23	22	21	20	19	18	17	16
JPEG Line Buffer Read/Write Port bits [23:16]							

15	14	13	12	11	10	9	8
JPEG Line Buffer Read/Write Port bits [15:8]							

7	6	5	4	3	2	1	0
JPEG Line Buffer Read/Write Port bits [7:0]							

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

These bits become the JPEG line buffer read port (a source of DMA transfer of YUV data) during JPEG decoding.

For DMA transfer, set 0xE000_00C0(JPEG1) or 0xD000_00C0(JPEG2), an address viewed from the AHB bus of this port.

8.4.6 JPEG Codec Registers (JCODEC[2:1])

Operation Mode Setting Register

JCODEC[0x00] Default value = 0x0000 Read/Write

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

Bit 4: **Reserved**

Be sure to set “0”.

Bit 3: **Marker Insert Enable**

This bit enables insertion of data written in the Insert Marker Data Register in a JPEG file. These bits are not used in JPEG decoding.

0: Disabled (marker not inserted) (default value)

1: Enabled (marker inserted)

Bit 2: **JPEG Operation Selection**

Specifies the JPEG operation mode.

Table 8.11 JPEG Operation Selection

Bit 2	JPEG Operation
0 (Default value)	Encode
1	Decode

Bits [1:0]: **YUV Format Selection**

Specify the YUV data format for JPEG encoding. JPEG encoding in YUV 4:4:4 format is not available if scaling has not been performed by the resizer. The bits indicate YUV data format of a JPEG file during JPEG decoding.

Table 8.12 YUV Format Selection

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

Command Setting Register

JCODEC[0x04] Default value = not applicable Write Only

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

The user should not attempt to read this register. Also, do not attempt to write it when the JPEG controller is in operation. (A reset operation is allowed.)

Bit 7: **JPEG Codec Software Reset**

This bit resets the JPEG Codec through software. It does not reset the JPEG Codec registers.

0: No effect.
1: Reset

Bit 0: **JPEG Operation Start**

Initiates the JPEG operation (including YUV Data Capture).

0: No effect.
1: Start JPEG operation.

JPEG Operation Status Register

JCODEC[0x08] Default value = 0x0000 Read Only

15	14	13	12	11	10	9	8
n/a							
7	6	5	4	3	2	1	0
n/a							JPEG Operation Status (RO)

Bit 1 of JPEG Status Flag Register and bit 1 of JPEG Raw Status Flag Register are cleared by reading this register.

Bit 0: **JPEG Operation Status.**

Indicates the operating state of the JPEG Codec.

0: Stopped.
1: JPEG encoding and decoding in progress

Quantization Table Number Register

JCODEC[0x0C] Default value = 0x0000 Read/Write

15	14	13	12	11	10	9	8
n/a							
7	6	5	4	3	2	1	0
n/a					V Table Select	U Table Select	Y Table Select

Bit 2: **V Table Select**

Specifies the quantization table number used for V component in JPEG Encoding Mode. This bit is not used in JPEG decoding.

0: Use quantization table 0 (default value).
1: Use quantization table 1.

Bit 1: **U Table Select**

Specifies the quantization table number used for U component in JPEG Encoding Mode. This bit is not used in JPEG decoding.

0: Use quantization table 0 (default value).
1: Use quantization table 1.

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

Specifies the quantization table number used for Y component in JPEG Encoding Mode. This bit is not used in JPEG decoding.

0: Use quantization table 0 (default value).
1: Use quantization table 1.

Huffman Table Number Register

JCODEC[0x10] Default value = 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 AC Table Select**

Specifies the AC Huffman table number used for V component in JPEG Encoding Mode. Set this bit to "1" for the JPEG Fast Encoding Mode. This bit is not used in JPEG decoding.

0: Use AC Huffman table 0 (default value).
1: Use AC Huffman table 1.

Bit 4: **V DC Table Select**

Specifies the DC Huffman table number used for V component in JPEG Encoding Mode. Set this bit to "1" for the JPEG Fast Encoding Mode. This bit is not used in JPEG decoding.

0: Use DC Huffman table 0 (default value).
1: Use DC Huffman table 1.

Bit 3: **U AC Table Select**

Specifies the AC Huffman table number used for U component in JPEG Encoding Mode. Set this bit to "1" for the JPEG Fast Encoding Mode. This bit is not used in JPEG decoding.

0: Use AC Huffman table 0 (default value).
1: Use AC Huffman table 1.

Bit 2: **U DC Table Select**

Specifies the DC Huffman table number used for U component in JPEG Encoding Mode. Set this bit to "1" for the JPEG Fast Encoding Mode. This bit is not used in JPEG decoding.

0: Use DC Huffman table 0 (default value).
1: Use DC Huffman table 1.

Bit 1: **Y AC Table Select**

Specifies the AC Huffman table number used for Y component in JPEG Encoding Mode. Set this bit to "0" for the JPEG Fast Encoding Mode. This bit is not used in JPEG decoding.

0: Use AC Huffman table 0 (default value).
1: Use AC Huffman table 1.

Bit 0: **Y DC Table Select**

Specifies the DC Huffman table number used for Y component in JPEG Encoding Mode. Set this bit to "0" for the JPEG Fast Encoding Mode. This bit is not used in JPEG decoding.

0: Use DC Huffman table 0 (default value).
1: Use DC Huffman table 1.

Set DRI Register 0

JCODEC[0x14] Default value = 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

Set DRI Register 1

JCODEC[0x18] Default value = 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

Set DRI Register 0 bits [7:0]
Set DRI Register 1 bits [7:0]:

DRI Value bits [15:0]

Specify the number of MCUs into which a RST marker is inserted in JPEG encoding. (If these bits are set to “0”, a RST marker is not inserted, but a “0” is inserted into the Define RST Interval Marker as the value defining the intervals.
This bit is not used in JPEG decoding.

Vertical Pixel Size Register 0

JCODEC[0x1C] Default value = 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 value = 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]

These bits specify the image size in the vertical direction during JPEG encoding and YUV Data Capture.
The bits indicate the vertical image size of a JPEG file during JPEG decoding.

Note: These bits are write-only bits when the YUV Data Capture mode is specified (JCTL [0x00], Bits 3-1 = 011 or 111). A read on the register during that time yields an unknown value.

Horizontal Pixel Size Register 0

JCODEC[0x24] Default value = 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 value = 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]

These bits specify the image size in the horizontal direction during JPEG encoding and YUV Data Capture.
The bits indicate the horizontal image size of a JPEG file during JPEG decoding.

Note: These bits are write-only bits when the YUV Data Capture mode is specified (JCTL [0x00], Bits 3-1 = 011 or 111). A read on the register during that time yields an unknown value.

Reserved Register
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 value = 0x0000 Read/Write

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

Bits [1:0]: **RST Marker Operation Select**
These bits set RST marker operation during decoding. These bits are not used in JPEG encoding.

Table 8.13 RST Marker Selection

Bits [1:0]	RST Marker Operation
00 (Default)	Disables error detection/data correction This setting is used only in the conditions assuring that the JPEG file to be decoded is normal and error-free. If any error is found in the JPEG file during this setting, error detection process will not start, and the decoding will not terminate successfully.
01	Enables error detection In this setting, the JPEG decoding terminates upon error detection during the decoding, and a JPEG codec end interrupt occurs. To check the error type, read JPEG decode error status of the JCODEC[0x3C]RST marker operation status register bits [6:3]. If an error is detected, software reset of JPEG codec core is required to start the next decoding process.
10	Enables data correction In this setting, the data is skipped or added automatically upon error detection during the decoding, and the decoding process continues until all JPEG file are decoded. When the JPEG decoding is completed, a JPEG codec end interrupt occurs.
11	Reserved

RST Marker Operation Status Register
JCODEC[0x3C] Default value = 0x0000 Read Only

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

Bit 7: **Revise Code**
This bit indicates the status of correcting operation during JPEG decoding. It is enabled when the RST marker processing is set to data correction enabled (the RST marker operation setting register bits [1:0] are set to "10"). This bit is not used in JPEG encoding.

0: Correction operation has not been done
1: Correction operation has been done

Bits [6:3]: **JPEG Decode Error Status**

These bits indicate the error code in JPEG decoding. The setting is enabled when the RST marker processing is set to error detection enabled (the RST marker operation setting register bits [1:0] are set to "01"). These bits are not used in JPEG encoding.

Table 8.14 JPEG Error Status

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

Insert Marker Data Register

JCODEC[0x40-0xCC] Default value = 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

A 36-byte register that inserts a marker during JPEG encoding. All of 36 bytes are3 inserted regardless of the length to the market. This register is not used in JPEG decoding.

Address Offset [40h-44h]: Specifies the marker code to be inserted.

Address Offset [48h-4Ch]: Specifies the length of the marker. The allowable range is from 0002h to 0022h.

Address Offset [50h-CCh]: Specifies the marker data. Up to 32 bytes. The data portion exceeding the length of marker must be padded with FFh.

Quantization Table No.0 Register							
JCODEC[0x400-0x4FC] Default value = 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

Quantization Table No.0
Specifies the quantization table value that corresponds to table number 0 used for JPEG encoding. These bits are not used in JPEG decoding.

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

Quantization Table No.1
Specifies the quantization table value that corresponds to table number 1 used for JPEG encoding. These bits are not used in JPEG decoding.

DC Huffman Table No.0 Register 0
JCODEC[0x800-0x83C] Default value = 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
Specifies the DC Huffman table value that corresponds to table number 0 used for JPEG encoding. It specifies the number of symbols for each length in the Huffman table. This bit is not used when Huffman tables are automatically setup. These bits are not used during JPEG decoding and automatic setup of Huffman tables.

DC Huffman Table No. 0 Register 1
JCODEC[0x840-0x86C] Default value = 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
Specifies the DC Huffman table value that corresponds to table number 0 used for JPEG encoding. It specifies a group number of all symbols in order of probability of occurrence. Use the low-order 4 bits to specify the value. The high-order 4 bits must be set to “0000”. This bit is not used when Huffman tables are automatically setup. These bits are not used during JPEG decoding and automatic setup of Huffman tables.

AC Huffman Table No. 0 Register 0
JCODEC[0x880-0x8BC] Default value = 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
Specifies the AC Huffman table value that corresponds to table number 0 used for JPEG encoding. It specifies the number of symbols for each length in the Huffman table. This bit is not used when Huffman tables are automatically setup. These bits are not used during JPEG decoding and automatic setup of Huffman tables.

AC Huffman Table No.0 Register 1
JCODEC[0x8C0-0xB44] Default value = 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
Specifies the AC Huffman table value that corresponds to table number 0 used for JPEG encoding. It specifies a run-length of zeros and group number of all symbols in order of probability of occurrence. This bit is not used when Huffman tables are automatically setup. These bits are not used during JPEG decoding and automatic setup of Huffman tables.

DC Huffman Table No. 1 Register 0

JCODEC[0xC00-0xC3C] Default value = not applicable Write Only

n/a							
15	14	13	12	11	10	9	8
DC Huffman Table 1 Register No. 0 bits [7:0]							
7	6	5	4	3	2	1	0

DC Huffman Table No.1

Specifies the DC Huffman table value that corresponds to table number 1 used for JPEG encoding. It specifies the number of symbols for each length in the Huffman table. This bit is not used when Huffman tables are automatically setup. These bits are not used during JPEG decoding and automatic setup of Huffman tables.

DC Huffman Table No.1 Register 1

JCODEC[0xC40-0xC6C] Default value = 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

Specifies the DC Huffman table value that corresponds to table number 1 used for JPEG encoding. It specifies a group number of all symbols in order of probability of occurrence. Use the low-order 4 bits to specify the value. The high-order 4 bits must be set to “0000”. This bit is not used when Huffman tables are automatically setup. These bits are not used during JPEG decoding and automatic setup of Huffman tables.

AC Huffman Table No.1 Register 0

JCODEC[0xC80-0xCBC] Default value = 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

Specifies the AC Huffman table value that corresponds to table number 1 used for JPEG encoding. It specifies the number of symbols for each length in the Huffman table. This bit is not used when Huffman tables are automatically setup. These bits are not used during JPEG decoding and automatic setup of Huffman tables.

AC Huffman Table No.1 Register 1

JCODEC[0xCC0-0xF44] Default value = 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

Specifies the AC Huffman table value that corresponds to table number 1 used for JPEG encoding. It specifies a run-length of zeros and group number of all symbols in order of probability of occurrence. This bit is not used when Huffman tables are automatically setup. These bits are not used during JPEG decoding and automatic setup of Huffman tables.

8.5 Explanation of Operations

8.5.1 Capture Control

Refer to “S2S65A30 JPEG decode driver Design Specification”(Linux) for JPEG decoding flow.

The device controls data capture using state machines. This is because JPEG encoding of camera images or YUV capture requires frame by frame capture and it is expected that restrictions on time will be relaxed by controlling it through hardware.
For state machines, conditions for state transitions and states have different meaning depending upon the JPEG encoding of camera images and YUV capture.
The current state of a state machine can be read from the relevant register. Since the machine is a signal changing in real time, it is desired to read only for software debugging purposes.

8.5.1.1 State Machine for JPEG Encoding of Camera Image

Figure 8.2 illustrates a state machine for controlling capture of camera images in JPEG encoding mode.

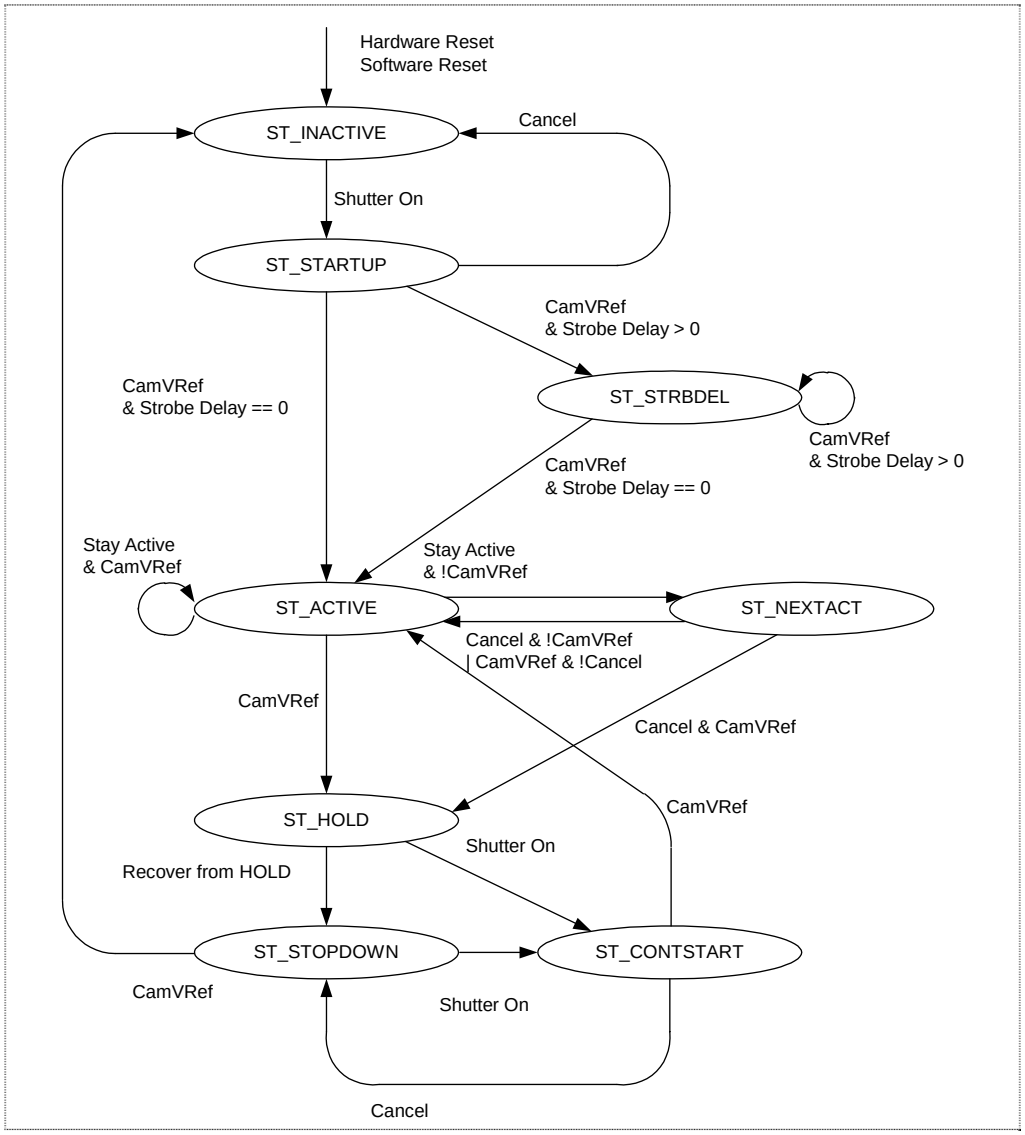


Figure 8.2 State Machine for Camera Image JPEG Encode

The meaning of each state is described in the table below. State IDs are values that are read by reading the appropriate registers.

Table 8.15 State Description in Camera Image JPEG Encode Mode

State	Description
ST_INACTIVE State ID (0x0)	An initial state at which the state machine has not captured camera data yet.
ST_STARTUP State ID (0x1)	A state changes from ST_INACTIVE to a state at which a shutter button is pressed, but the state machine has not captured camera data yet.
ST_STRBDEL State ID (0x3)	The state machine waits for Strobe Frame Delay. If Strobe Delay is detected, the state machine causes Capture to be delayed by the number of frames specified before going into ST_ACTIVE. The state machine has not captured camera data yet.
ST_ACTIVE State ID (0x7)	A state at which the state machine captures camera data. In ST_ACTIVE, a state transition only occurs if a VREF sent from the camera is detected. This ensures that frame data is captured as long as trimming and scaling parameters are correctly specified.
ST_NEXTACT State ID (0xB)	The state machine goes to this state by asserting Request Next Frame Capture while ST_ACTIVE is triggered. It goes back to ST_ACTIVE as soon as a VREF from the camera is detected. This state is provided to encode frames in succession. The state machine has captured camera data.
ST_HOLD State ID (0x6)	The state machine goes to this state for the next frame which has been captured. The state machine has not captured camera data yet.
ST_STOPDOWN State ID (0x4)	The state machine receives the Reset Hold signal in response to ST_HOLD. It transitions to ST-INACTIVE on the subsequent frames, whereby the HOLD state is cleared. The state machine has not captured camera data yet.
ST_CONTSTART State ID (0x5)	The state machine initiates capturing of frames when “Shutter press” is detected in HOLD state. The state machine has not captured camera data yet.

The association between events and actions are shown below.

Table 8.16 Event Description in Camera Image JPEG Encode Mode

Event	Action
Software Reset	Reset the resizer through software.
Shutter On	Write a “1” to the JPEG Start/Stop Register.
Cancel, Recover from HOLD	Write a “0” to the JPEG Start/Stop Register.
CamVRef	The timing of transition of camera image data’s VREF data level from active to inactive. The term “VREF” used alone in this section refers to this timing.
Stay Active	Write a “1” to the Request Next Frame bit of JPEG Control Register.
Strobe Delay	Strobe Frame Delay Count.

Figure 8.3 shows a timing chart for a single JPEG encoding sequence of a camera image.

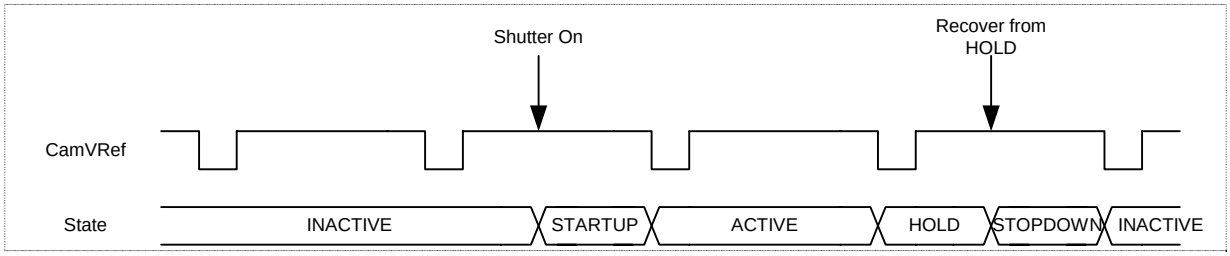


Figure 8.3 Timing Chart for Camera Image JPEG Encode (Single)

8.5.1.2 State Machine for YUV Capture

Figure 8.4 illustrates a state machine for controlling the capture in the YUV capture mode.

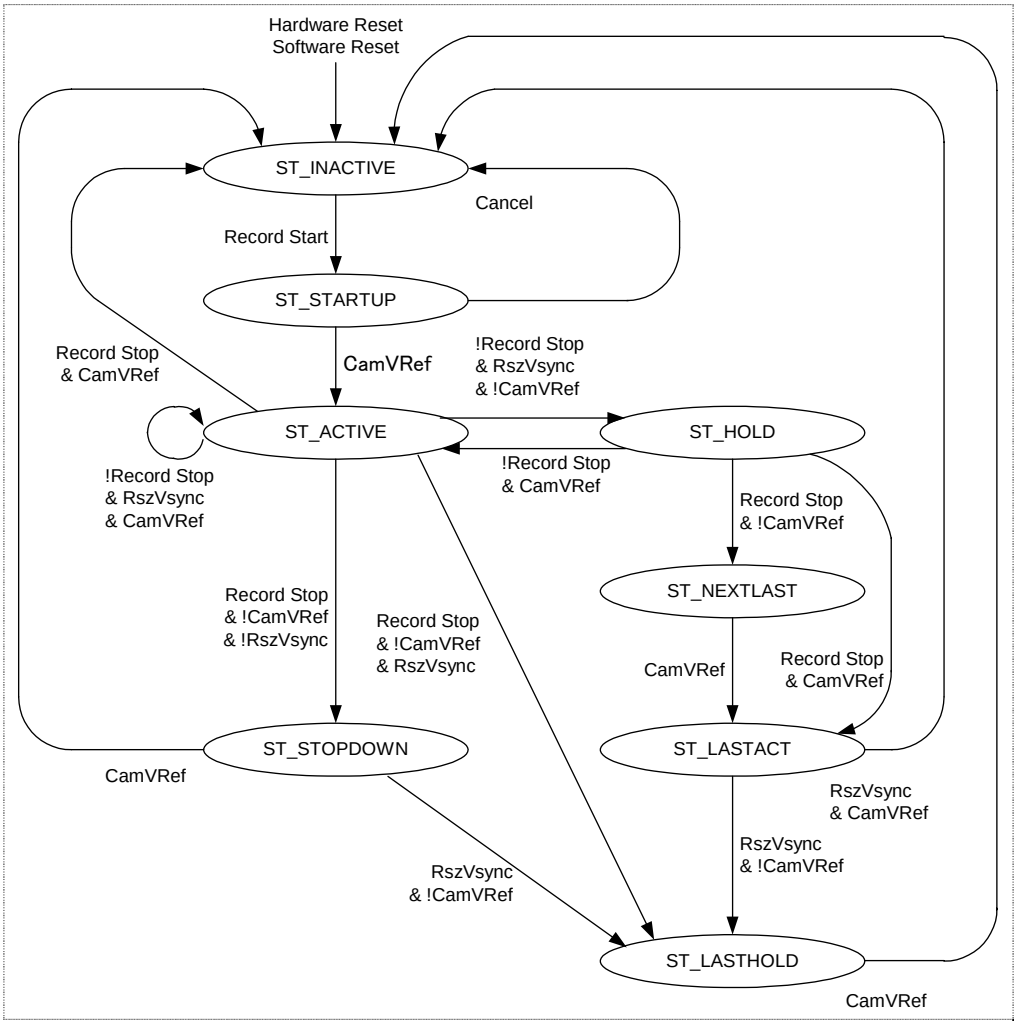


Figure 8.4 State Machine for YUV Capture

Table 8.17 State Description in YUV Capture Mode

State	Description
ST_INACTIVE State ID (0x0)	An initial state at which the state machine has not captured camera data yet.
ST_STARTUP State ID (0x1)	The state machine transitions from ST_INACTIVE to a state at which it receives a Record Start signal. The state machine has not captured camera data yet.
ST_ACTIVE State ID (0x7)	A state at which the state machine captures camera data. The state machine has captured camera data.
ST_HOLD State ID (0x6)	The state machine transitions from ST_ACTIVE to ST_HOLD when a Capture Data VREF is generated in the frame. The state machine has not captured camera data yet.
ST_STOPDOWN State ID (0x4)	The state machine transitions from ST_INACTIVE to a state at which it receives a Record Stop signal. Capture process completes with the current frame. The state machine has captured camera data.
ST_NEXTLAST State ID (0xE)	The state machine goes to this state upon reception of a Record Stop signal while ST_HOLD is triggered. It is not possible for the JPEG module to recognize the completion of motion image capture between the moment a Capture Data VREF is generated and the moment a Camera VREF is generated. To resolve this problem, this state is provided to let the Record Stop signal capture another one frame after receiving a Capture Data VREF. The state machine has not captured camera data yet.
ST_LASTACT State ID (0xF)	The state machine goes to a state at which it captures another frame as described above. The state machine has captured camera data.
ST_LASTHOLD State ID (0xD)	The state machine transitions to ST_LASTHOLD when a Capture Data VREF is generated in the last recording frame. The state machine has not captured camera data yet.

The association between events and actions are shown below.

Table 8.18 Event Description in YUV Capture Mode

Event	Action
Software Reset	Reset the resizer through software.
Record Start	Write a “1” to the JPEG Start/Stop Register.
Cancel, Record Stop	Write a “0” to the JPEG Start/Stop Register.
CamVRef	The timing of transition of camera image data’s VREF data level from active to inactive. The term “VREF” used alone in this section refers to this timing.
RszVsync	An end-of-frame signal that is output by the resizer.

8.5.2 Resizing

The resizer performs trimming and downsizing for image data sent from a camera interface. This bit is not used in JPEG decoding. It has two stages – trimming and downsizing. Since image data sent from a camera module in YUV 4:2:2 format is converted to YUV 4:4:4 format by a camera interface, trimming can be performed on a single pixel basis (1x1) by the resizer.

8.5.2.1 Trimming

Trimming is performed as the preprocessing stage for scaling to trim the edges of an image to a size required for scaling process. The trimming process uses the register that contains starting X and Y coordinates and ending X and Y coordinates. The upper left corner of the original image is defined as coordinates (0, 0). The starting coordinates correspond to the upper left corner of a trimmed image and the ending coordinates correspond to the upper right corner of a trimmed image. These settings can be specified on a single pixel basis (a pixel matrix of 1 x 1), they cannot be greater than the maximum coordinate value of an image from a camera. If a specified ending coordinate value is less than a starting coordinate value, the resizer fails to operate correctly, outputting some piece of data. For this reason, be sure to specify ending coordinates greater than starting coordinates. To forcefully stop data output from the resizer, disable the circuit by writing a 0 to RSZ[0xC0], bit 0.

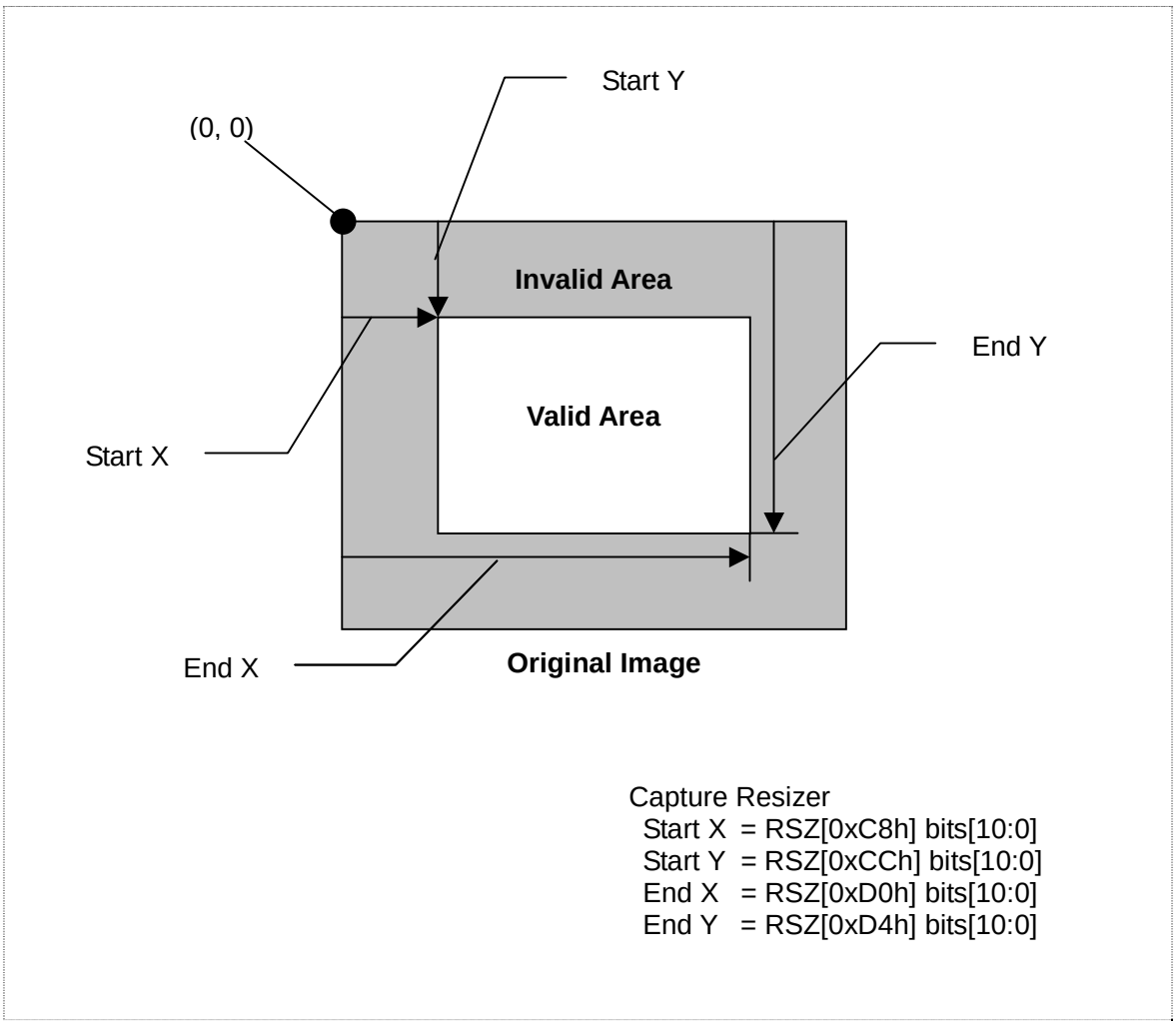


Figure 8.5 Trimming Function

8.5.2.2 Scaling

Scaling is performed as the postprocessing stage of trimming to reduce the image by a factor specified for the trimmed image. The size of the image that was scaled down must be identical to the one to be specified in JPEG Encoding Mode.
The scaling process has three scaling modes – “Downsizing”, “Averaging” and “No Scaling”. In Downsizing mode, select a representative point from an image block and use it for a single pixel of a scaled down image. The scaling can be set by selecting any scaling factor between 1/2 and 1/8. In Averaging mode, the scaling in the vertical direction is achieved by downsizing and the scaling in the horizontal direction is achieved by averaging. For an image that is selected for downsizing, all pixels in each column of an image block are added and their average value is used for a single pixel in the scaled down image block. The scaling factors available in Averaging mode are 1/2, 1/4 and 1/8. Specify No Scaling mode when images are not to be scaled down. In No Scaling mode, the value set in the Set Scaling Factor Register becomes invalid. If a “1” is written in the above register in any other modes, operations are not guaranteed. Be sure to select the No Scaling mode when images are not to be scaled down.

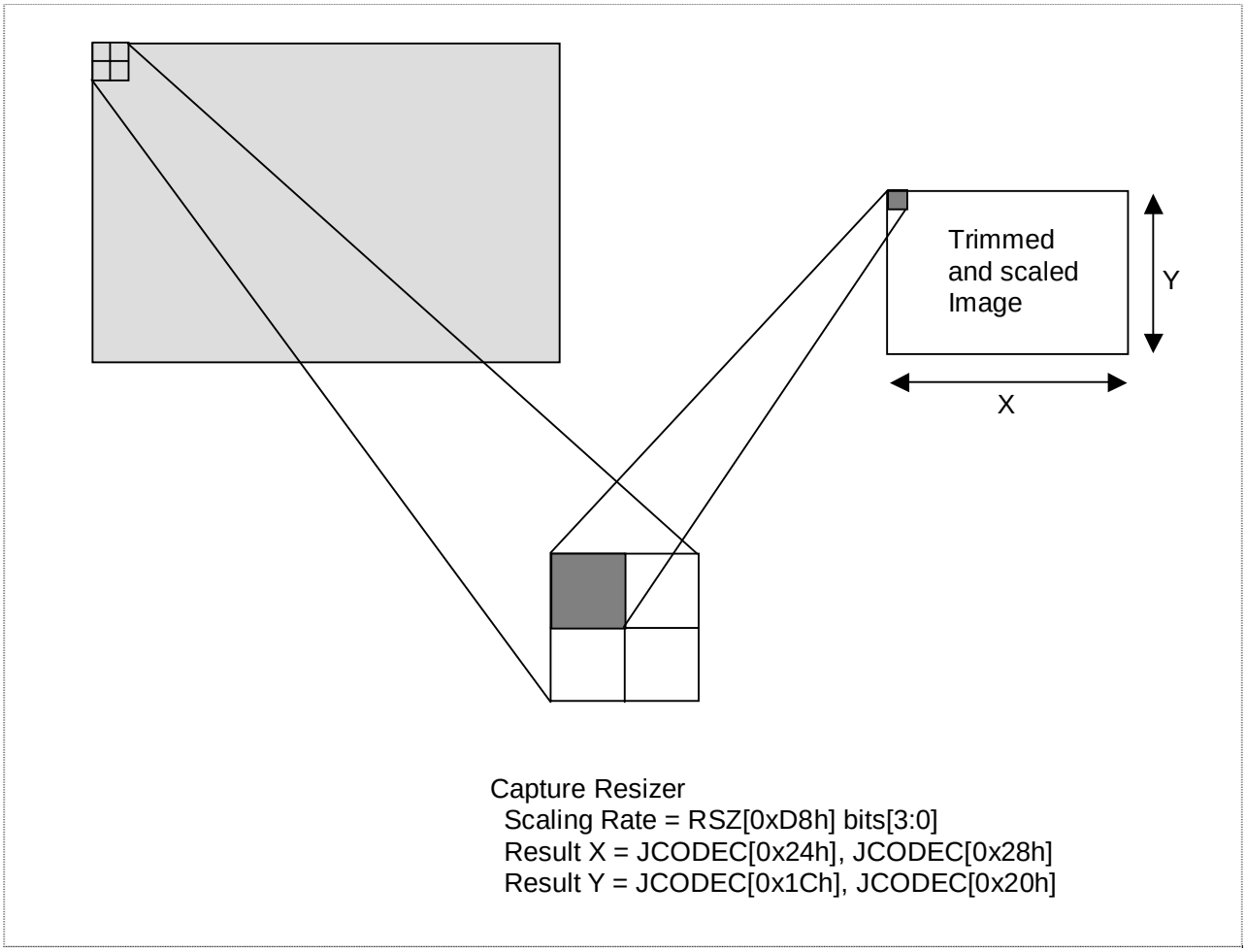


Figure 8.6 Scaling Example (1/2 Scaling)

8.5.2.2.1 Scaling Down by a Factor of 1/2

Scaling down by a factor of 1/2 reduces every 2 x 2 pixel block to one pixel. For this scaling factor, both of two scaling modes - Downsizing and Averaging - are available. Note that, for both Downsizing and Averaging modes, the scaling in the vertical direction is fixed to Downsizing mode.

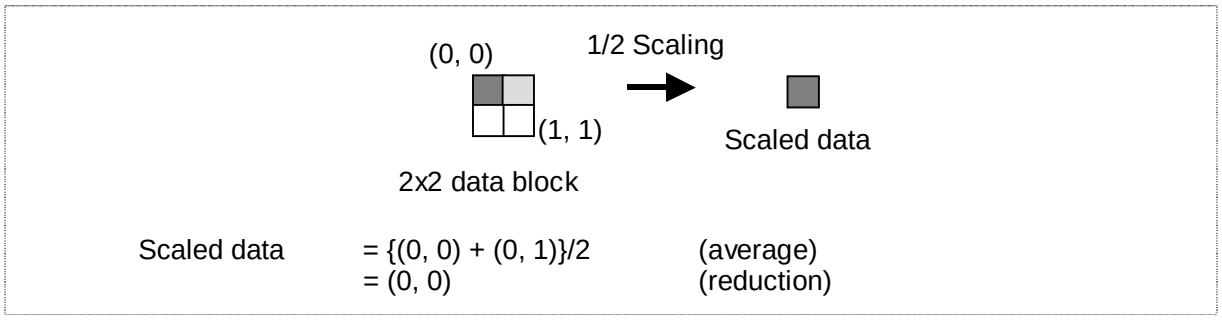


Figure 8.7 1/2 Compression

8.5.2.2.2 Scaling Down by a Factor of 1/3

Scaling down by a factor of 1/3 reduces every 3 x 3 pixel block to one pixel. For this scaling factor, only Downsizing mode is available.

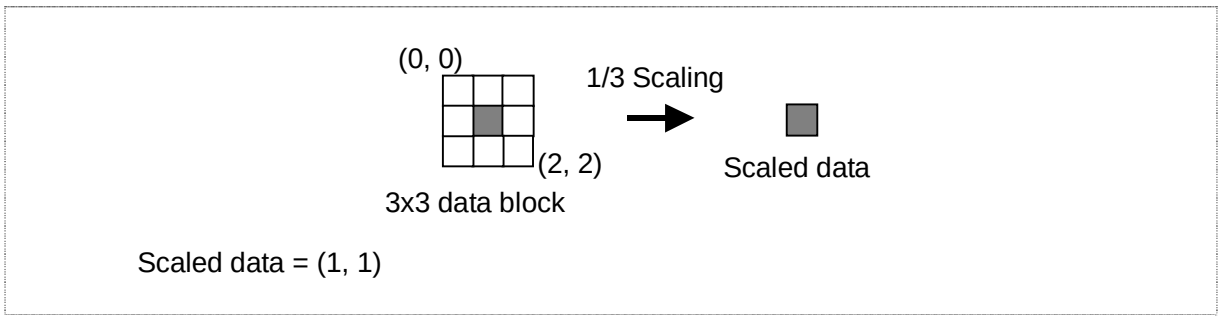


Figure 8.8 1/3 Scaling

8.5.2.2.3 Scaling Down by a Factor of 1/4

Scaling down by a factor of 1/4 reduces every 4 x 4 pixel block to one pixel. For this scaling factor, both of two scaling modes - downsizing and averaging - are available. Note that, for both Downsizing and Averaging modes, the scaling in the vertical direction is fixed to Downsizing mode.

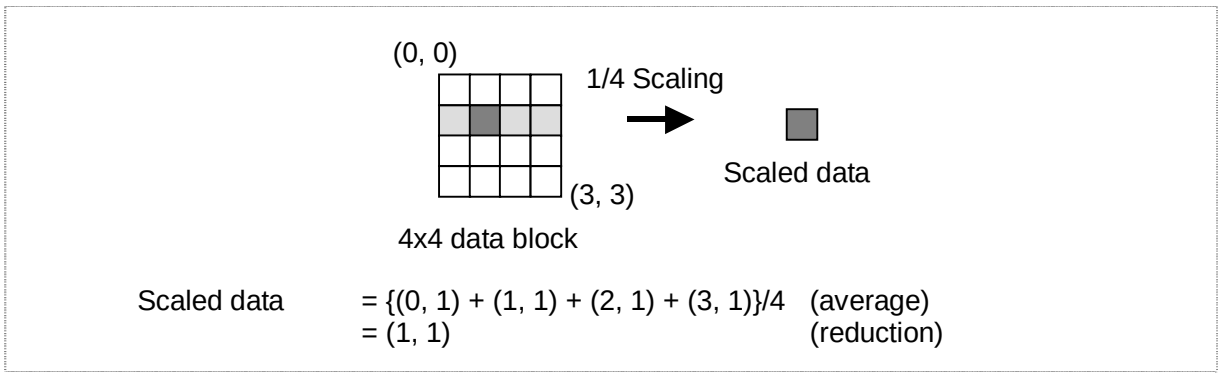


Figure 8.9 1/4 Scaling

8.5.2.2.4 Scaling Down by a Factor of 1/5

Scaling down by a factor of 1/5 reduces every 5 x 5 pixel block to one pixel. For this scaling factor, only Downsizing mode is available.

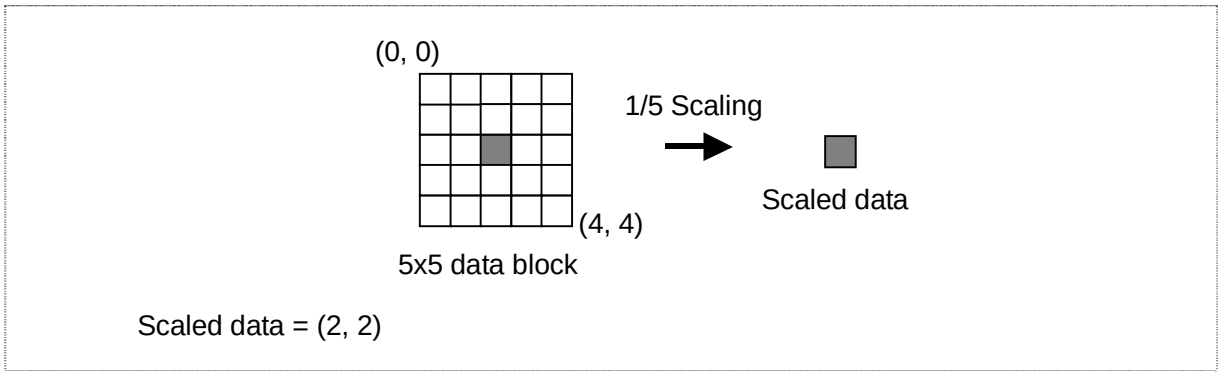


Figure 8.10 1/5 Scaling

8.5.2.2.5 Scaling Down by a Factor of 1/6

Scaling down by a factor of 1/6 reduces every 6 x 6 pixel block to one pixel. For this scaling factor, only Downsizing mode is available.

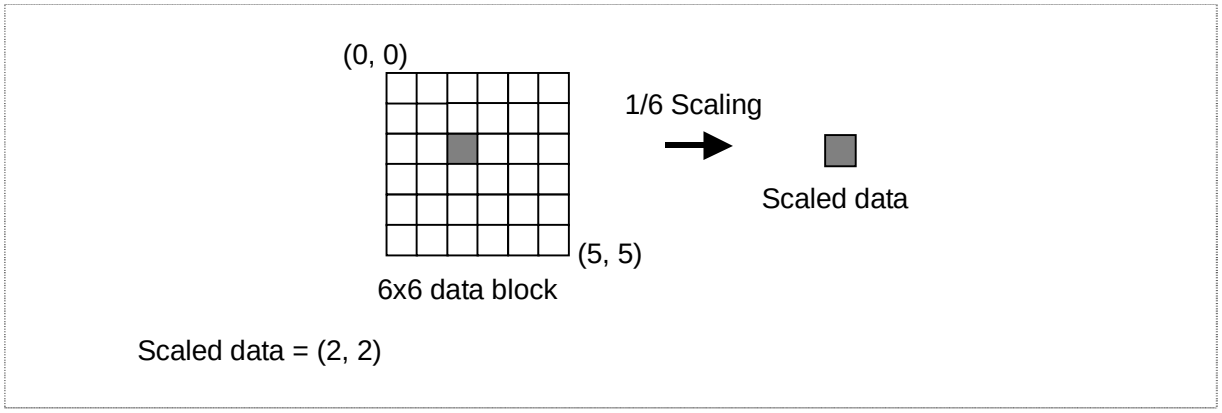


Figure 8.11 1/6 Scaling

8.5.2.2.6 Scaling Down by a Factor of 1/7

Scaling down by a factor of 1/7 reduces every 7 x 7 pixel block to one pixel. For this scaling factor, only Downsizing mode is available.

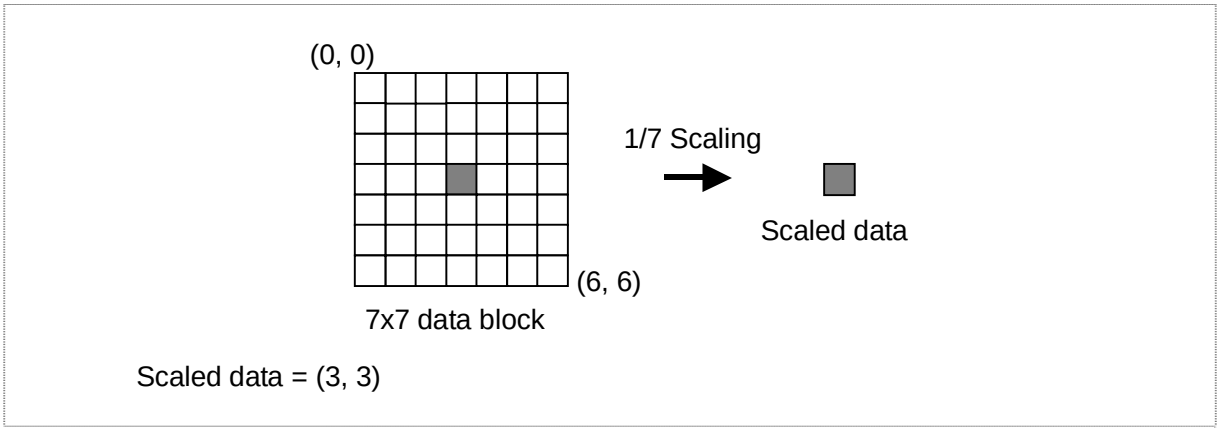


Figure 8.12 1/7 Scaling

8.5.2.2.7 Scaling Down by a Factor of 1/8

Scaling down by a factor of 1/8 reduces every 8 x 8 pixel block to one pixel. For this scaling factor, both of two scaling modes - Downsizing and Averaging - are available. Note that, for both Downsizing and Averaging modes, the scaling in the vertical direction is fixed to Downsizing mode.

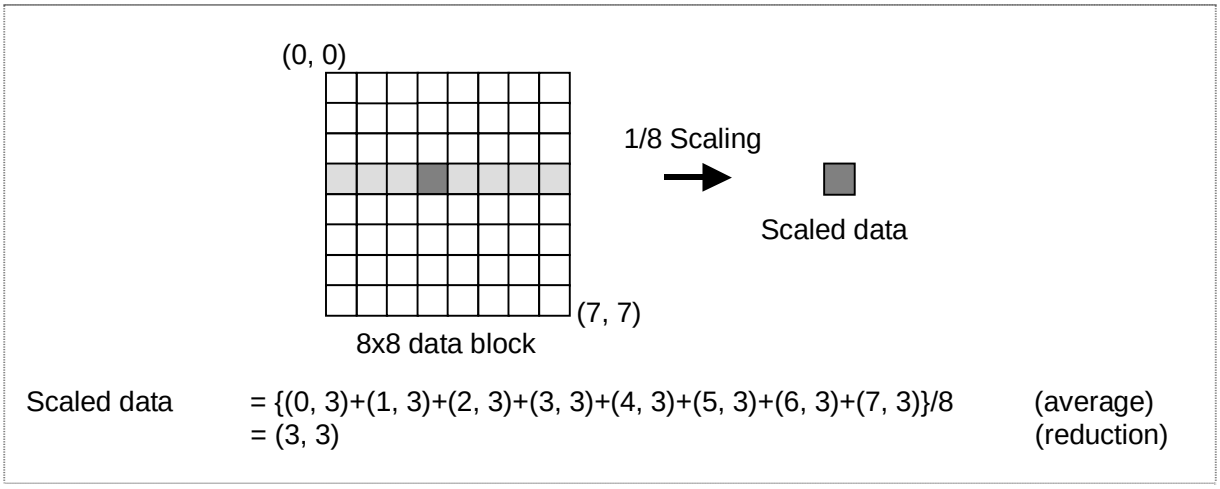


Figure 8.13 1/8 Scaling

8.5.2.3 Restrictions on the Use of Resizer

It is not possible to change the registers in the resizer, except Software Rest or Enable, when data is being transmitted from a camera interface. Any attempt to do so results in data corruption. To change settings of registers in the resizer when a camera is operating, it is common to change them in a period of time during which the camera interface is not receiving data from the image sensor between frames (VREF data bank period) using a VREF interrupt from the camera interface.
It is not possible to specify any values greater than those values for a camera image in any of the Resizer Start X, Start Y, End X and End Y Coordinates Registers. If such a value is specified, the resizer will not operate correctly.
An image that is trimmed by specifying the Resizer Start X, Start Y, End X and End Y Coordinates Registers must be divisible by a value specified in the Set Scaling Factor Register.

8.5.3 Data Flow of Image Processing

This section provides a diagram showing the data flow in various image processing modes.

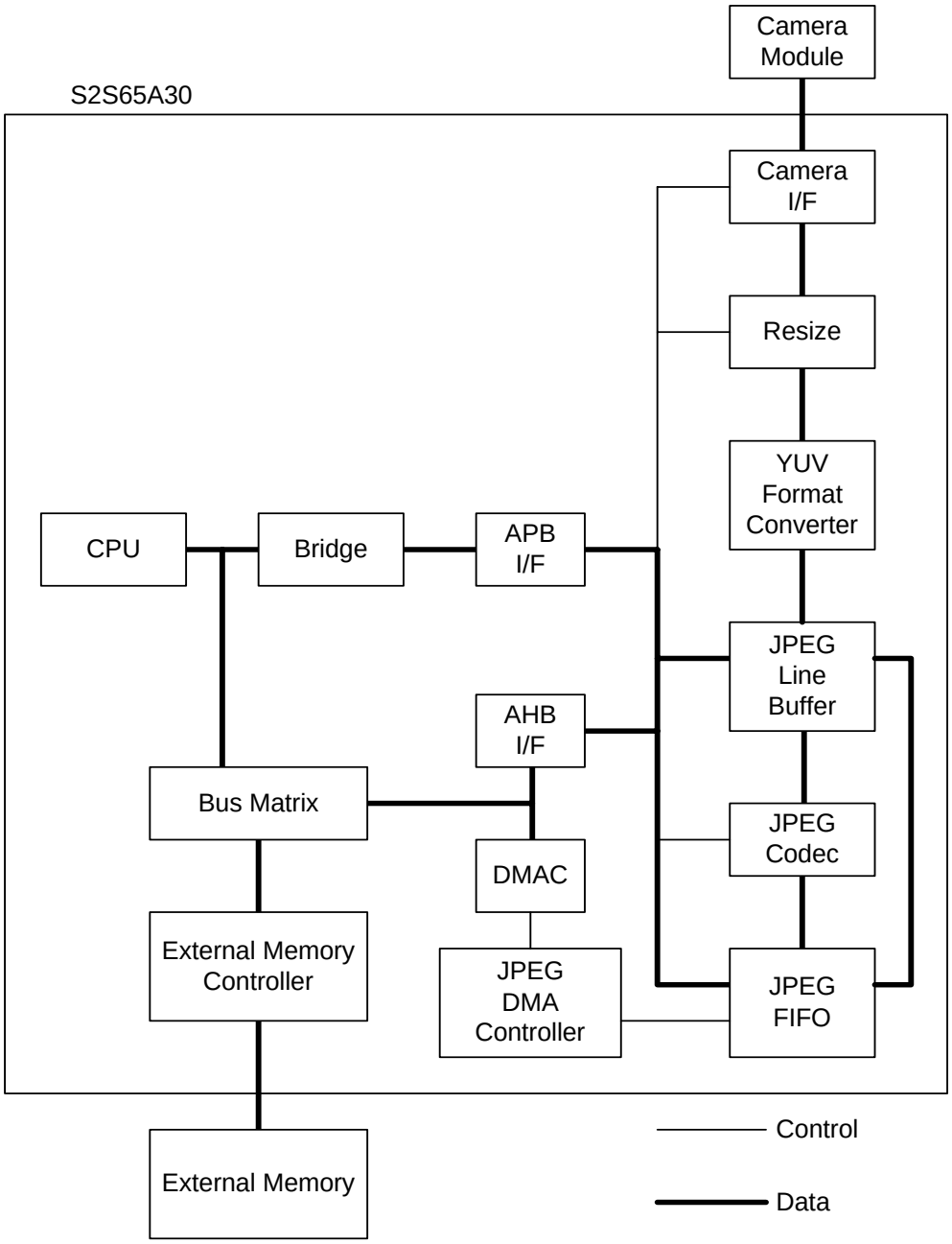


Figure 8.14 Data Flow of Image Processing

8.5.3.1 Camera Image JPEG Encoding

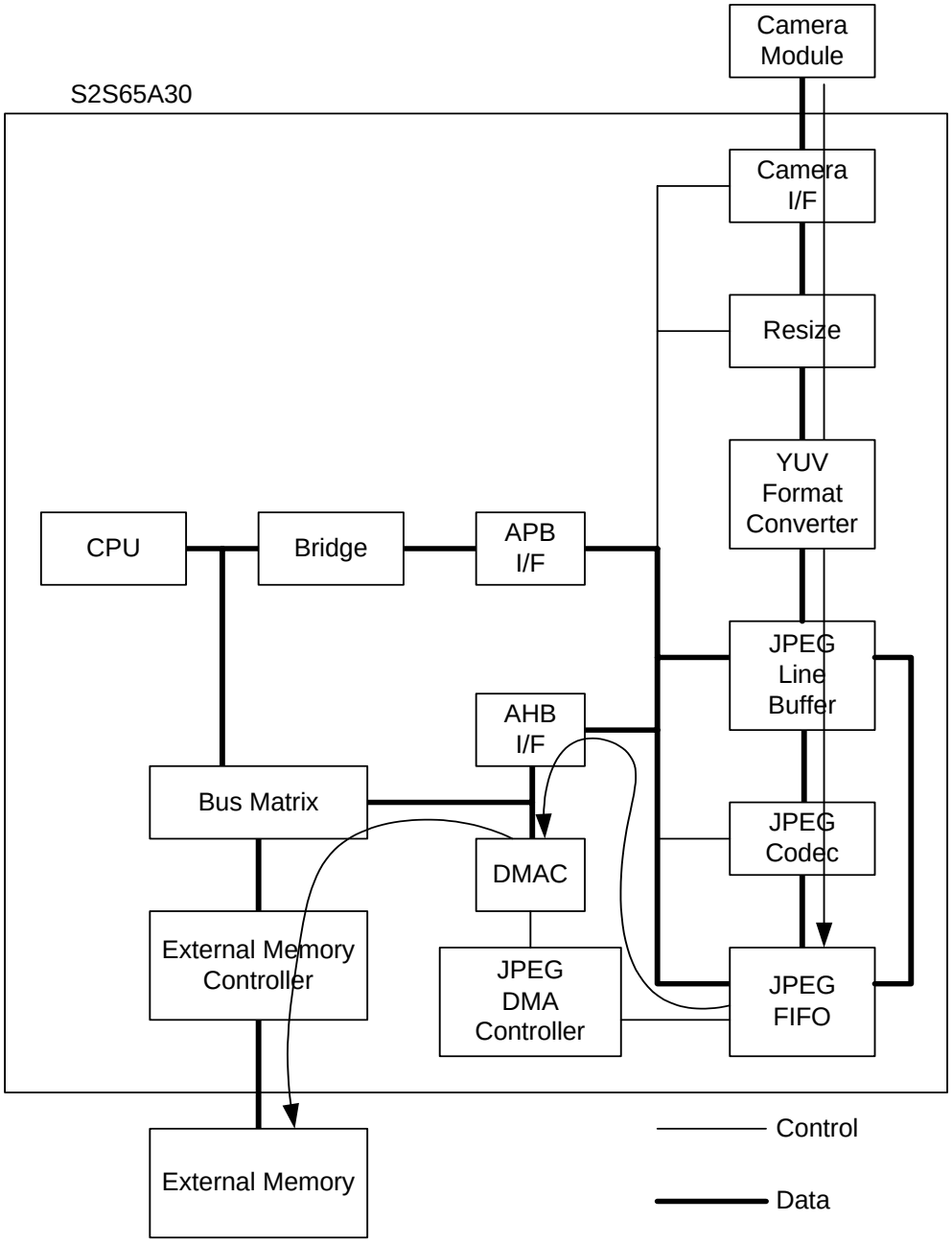


Figure 8.15 Data Flow of Camera Image JPEG Encoding

8.5.3.2 YUV Data Capture

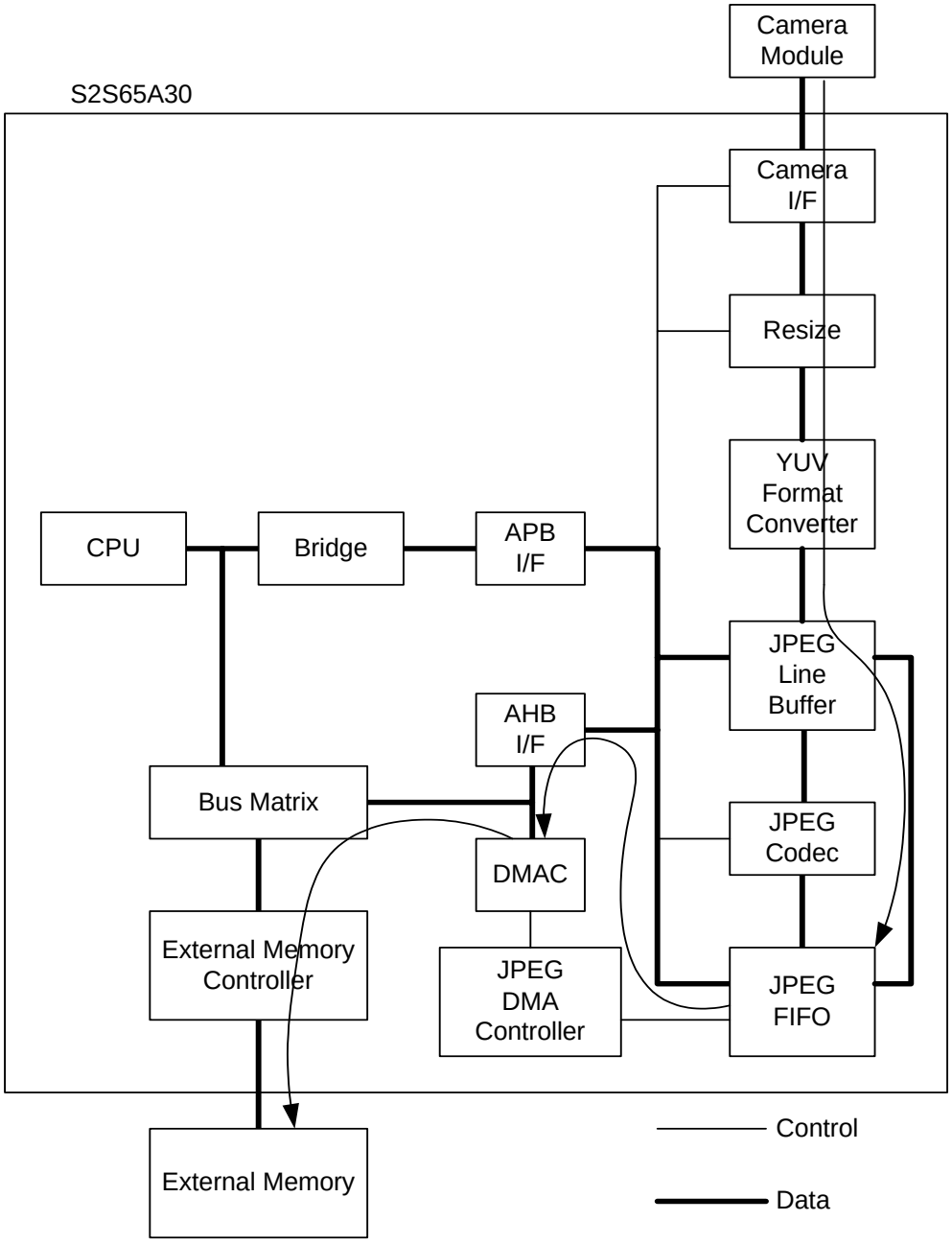


Figure 8.16 Data Flow of YUV Data Capture

8.5.3.3 JPEG Decoding (Using DMA)

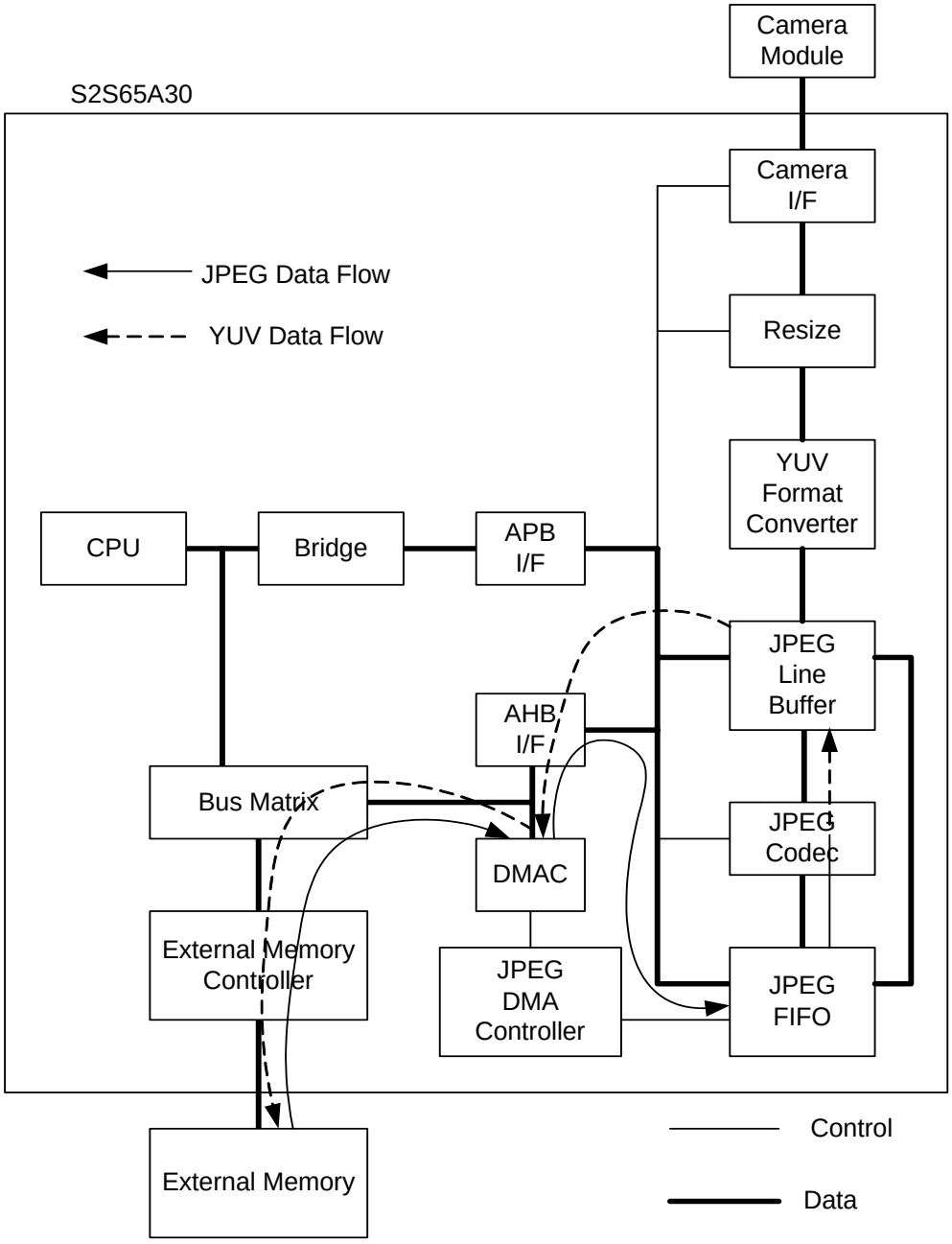


Figure 8.17 JPEG Decoding Data Flow (Using DMA)

8.5.4 JPEG Codec

The JPEG Codec circuit contained in the device almost complies with the baseline JPEG standard and satisfies the conformance tests required by Part 2 of JPEG standard (ISO/IEC 10918-2). The JPEG Codec circuit support up to 1600 pixels width and 2048 pixels height. Because of restriction of internal RAM capacity used for line buffer, however, S2S65A30 supports only up to 640 pixels width and, among general image size formats, up to VGA of JPEG encoding.
YUV 4:4:4 formatted data sent from the resizer is converted by the YUV format converter into a YUV format specified by the JCODEC[0x00] bits [1:0]. For JPEG encoding/decoding of camera images and YUV capture, the resolution of image to be processed must meet the following minimum restriction.

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 restrictions on the minimum size of images: operations of images smaller than that size is not guaranteed.

Table 8.20 Minimum Size

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

Two quantization tables can be used for encoding process, and four for decoding. Two each of the AC and DC Huffman tables can be used for JPEG encoding and decoding processes. For the JPEG encoding process, a marker of up to 36 bytes, including a marker identifier, can be inserted. Markers SOI, SOF0, SOS, DQT, DHT, DRI, RSTm, DNL, and EOI are automatically identified during decoding process. Other markers are neglected.
The JPEG encoding process for camera images supports YUV 4:4:4, YUV 4:2:2, YUV 4:2:0, and YUV 4:1:1. For YUV 4:4:4, the amount of data is larger than that of a camera image. It is, therefore, not supported when encoding camera images without scaling. Only YUV 4:2:2 and YUV 4:2:0 are supported for decoding from JPEG data to YUV data. The image data processing capability is less than one-thirtieth of a second for an image size of 640 x 480 pixels. This performance may not be achieved due to such factors as values of quantization and/or Huffman tables, and resolution of input images. The processing speed is, for JPEG decoding, approximately one-twentieth at the fastest because of limited bandwidth of internal bus and efficiency of DMA transfer. The speed becomes much slower if it includes transfer to additional destinations (e.g., external display controller) of YUV data after the decoding process.
An approach for performing JPEG encoding successively in a consistent manner would be to alternatively activate and deactivate encoding of one frame. The controller is capable of performing JPEG encoding successively on a frame by frame basis. Whether or not this capability can be achieved depends upon the type of camera modules employed in your system or the performance of software. Specifically, the following inequality must be satisfied.

Time period during which a camera VREF is held inactive
> Interrupt response time
+ Total amount of time a process is interrupted by higher priority tasks
+ Time to process settings for encoding the next frame
+ Time required to output markers from JPEG Codec circuit

The JPEG Codec circuits outputs a marker at a rate of 36 microseconds at 50 MHz system clock if the marker fast output mode added to the S2S65A30 is used, whereas it outputs it at a rate of approximately 2 milliseconds

if that mode is not used. To perform JPEG encoding successively on a VGA sized frame basis without using the marker fast output mode, it is very likely to generate an overflow unless there are alt least 15 lines in the period of time during which the VREF remains inactive.

8.5.4.1 Files That Cannot be JPEG Decoded

The following JPEG files cannot be decode.

- Non-JPEG files (even if they have a “jpg” extension).
- Files with a broken marker
- Gray image JPEG files (without UV data)
- JPEG files with non-YUV color elements
- JPEG files using a DNL marker
- Files with a broken Huffman or quantization table, data other than image.

If you cannot assure that a JPEG file input to the JPEG Codec circuit is normal, you must previously read the marker of the JPEG file with software, and make sure that it can be decoded by the JPEG codec circuit, so that overrun of the JPEG decoding process will be avoided. Because error correction by bit is not performed for JPEG, any type of error state can occur if the contents of the file is not assured. It is difficult for hardware to check all possible error states, and thus this device expects preliminary checking performed by software.JPEG. The decode marker read flag is an effective interrupt when a file is assured to be normal. In an environment where a JPEG file cannot be assured to be normal, however, the decode marker read flag should not be used.
For example, only single bit inversion of a marker code can cause JPEG decoding process to work abnormally. It is required for software to determine the occurrence of error when a decoding process does not end persistently.

8.5.4.2 Restrictions on JPEG Codec Registers

Do not modify any JPEG Codec register while it is in operation (between the start of the JPEG Codec and the end of the JPEG Codec). A response to an access during that period should cause a malfunction or should become invalid.
Accessing the JPEG Codec register with no clock supplied to the JPEG Codec circuit will be invalidated. The conditions that cause the JPEG Codec clock to stop are:

- The JPEG modules are disabled.
- The JPEG modules are enabled and the JPEG processing mode is neither 000 nor 100.

Registers that must be configured for YUV capture are designed so that they can be configured even if no clock is supplied to the JPEG Codec circuit.
If a read access is attempted on a register reserved for the JPEG Codec circuit or a write-only register, a read value as well as the operation of the JPEG Codec circuit are not guaranteed. Because restriction of access is dependent on JPEG decoding and encoding processes, pay particular attention so that registers should not be accessed improperly.
Reads from JPEG Codec Status Registers and JPEG Codec Marker Status Registers cause their internal state to change. Therefore, they must be read them only when needed. **Upon completion of JPEG encoding read JPEG Codec Status Register, and JPEG decoding read JPEG Codec Marker Status Register before reading JPEG Codec Status Register. This completes the operations of the JPEG Codec circuit. Subsequent operations should result in failure if the JPEG Codec Marker Status Register is not read.**
For both of the quantization and Huffman tables, a write to table can be omitted only when the same operation is performed in succession. The tables must be rewritten if the process is changed from decoding to encoding, or settings are modified even during encoding.

8.5.5 Functions Other Than JPEG Codec

8.5.5.1 JPEG FIFO

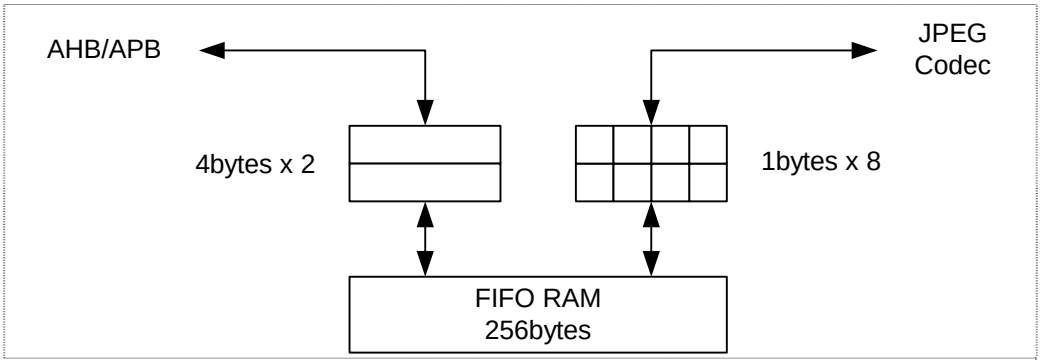


Figure 8.18 JPEG FIFO Overview

The JPEG FIFO is capable of holding 272 bytes of data using a 256-byte RAM and four 4-byte RAM buffers (two for each of read and write). To perform JPEG FIFO read/write operations by the CPU by checking statuses of the JPEG FIFO, it is recommended that the maximum value that can be read from and written into a 256-byte RAM be applied.
The JPEG FIFO Status Register can be used to check JPEG FIFO statuses. Further, most of the JPEG FIFO statuses can also be checked by the JPEG Interrupt Status Register. The JPEG FIFO Status Register indicates the status of the JPEG FIFO when it is read. The Empty Status of the JPEG Interrupt Register indicates that the FIFO has been empty at least once. The Full Status of the JPEG Interrupt Register indicates that the FIFO has been full at least once. The JPEG Interrupt Control Register generates an interrupt based on the above status information and retains it until an interrupt is cleared.
For JPEG encoding of camera images and of YUV data, there are two ways to retrieve data from the JPEG FIFO. One method is to let the CPU read the JPEG FIFO Read/Write Port Register. The other method is to use a JPEG DAM controller.

1. Low performance - An approach in which the JPEG FIFO Empty Flag is used to determine whether or not the FIFO is empty to let the CPU read the JPEG FIFO Read/Write Port Register. This is an impractical approach since the FIFO is not large enough, using interrupts that are triggered by FIFO level (full) or threshold level are inefficient. Since the CPU polls the JPEG FIFO Empty Status Flag to read data until the JPEG Codec End Interrupt Flag is set, this approach consumes large amounts of CPU. It is, therefore, not suitable for multitasking or real-time processing.

2. High performance - An approach in which a JPEG DMA controller is used to transfer data to memory areas. The JPEG DMA controller completes a data transfer upon reception of Frame End sent from the JPEG FIFO. The user can implement required transfer simply by setting the memory size circuit reserved as memory area with DAM setting to the maximum amount of data to be transferred. The user requires to set only a small number of registers for each frame and transfer from the FIFO causes virtually no CPU load.

Because, during JPEG decoding, the DMA controller writes to FIFO properly while monitoring empty state of FIFO, CPU does not need to know the FIFO status during DMA transfer.

8.5.5.2 JPEG Line Buffer

The JPEG line buffer contains 32K-byte and 24K-byte RAMs and enables data interchange between line-base and block interleaved methods. When creating a JPEG file, image data whose size is represented in MCUs is required. The line buffer complements for deficit in the data. For example, to encode a 100 × 100 pixel image in YUV4:2:2, the line buffer compensates for the image to 112 × 104 pixels before sending data to the JPEG Codec circuit. The line buffer simply compensates for data whose MCUs do not contain much details and does not serve to compensate for, for example, a 60 × 60 pixel data sent from the resizer to a 112 × 104.

It contains a data port register to retrieve YUV data during JPEG decoding, and can check the line buffer status by JPEG Line Buffer Status Register. Because YUV data is usually retrieved using the DMA controller, the line buffer and empty flag need not to be perceived during DMA transfer.
It is capable of detecting its own overflow state and generating an interrupt. In JPEG encoding, if the JPEG FIFO or JPEG Codec circuit is unable to continue processing, it waits for input data. If the JPEG Codec circuit frequently keeps incoming inputs from the resizer waiting, the buffer overflows. Whenever real-time encoding cannot be performed due to JPEG FIFO reads or JPEG encoding workloads, line buffer overflow interrupts are detected.
The line buffer handles data containing the lines for width in MCUs as a bank to implement data interchange between line-base and block interleaved methods. It is capable of varying the number of banks that can be stored in the 30 KB RAM based on the widths of input images to efficiently use the RAM.
For YUV 4:2:0, 16 lines make up one bank. For YUV 4:2:2, 8 lines makes up one bank.

Table 8.21 Amount of RAM Data (Number of Banks) Based on Size of Input Image

Width of input image	Amount of data
≤32	Width of input image × 32banks
≤64	Width of input image × 16banks
≤128	Width of input image × 8banks
≤256	Width of input image × 4banks
>256	Width of input image × 2banks

8.5.5.3 YUV Format Converter

The YUV format converter converts YUV 4:4:4 image sent form the resizer into four types of YUV formats. As illustrated below, averaging is adopted for the YUV conversion. Figure 8.18 shows an example of YUV format conversion of U component. The V component has the same formula for the averaging. Y component is the same as the original image for the YUV format.

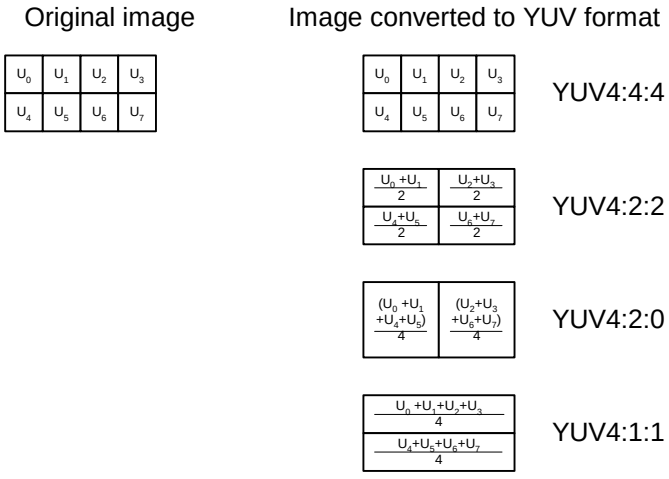


Figure 8.19 YUV Format Conversion

8.5.5.4 JPEG Module Interrupts

This section describes how to use interrupt flags.

1. JPEG Codec Interrupt Flag
   For JPEG-based camera image encoding, the JPEG Codec Interrupt Flag is used to access to a FIFO through a CPU. This flag is not used for JPEG encoding using DMA transfer. When this flag is turned on, the size of the JPEG file is determined. The amount of remaining data to be read last corresponds to the value obtained by subtracting the amount of data that has been read from the size of encoding result. This flag is not turned on for YUV Capture. For JPEG decoding, it is used as a trigger to read the last unit of data from the line buffer.

2. JPEG Line Buffer Overflow Interrupt Flag
   The JPEG Line Buffer Overflow Interrupt Flag is used for encoding JPEG camera images. Since the JPEG Codec circuit does not guarantee that JPEG encoding process always completes successfully, it is required to enable this interrupt when encoding JPEG camera images.

3. JPEG Decoding Marker Read Flag
   The JPEG Decoding Marker Read Flag is used for JPEG encoding. In an environment where a JPEG file is not assured to be decoded using the JPEG Code circuit, however, this flag in not used because software must be used to read the marker of the JPEG file to check the normal operation.
   The decoding process stops while this flag is set. If the JPEG decoder marker read interrupt is disabled, the flag is not set and stayed in low state.

4. JPEG FIFO Empty Flag
   The FIFO status can be checked at the start of JPEG decoding. After that, the status does not need to be checked because the DMA controller automatically performs DMA transfer by handshaking with the FIFO control circuit. If JPEG decoding does not terminate successfully or other error occurs, check this flag to isolate the cause of the error.
   In JPEG encoding, there is a way to use the JPEG FIFO Empty Flag to determine whether or not the FIFO is empty. It has strong implications of using it for just in case. It is, therefore, not necessary to perform encoding process by checking the state of this flag.

5. JPEG FIFO Full Flag
   The JPEG FIFO Full Flag may be possibly used to let CPU read the JPEG FIFO for JPEG-based camera image encoding, in which case, the number of bytes corresponding to the FIFO size is read from the FIFO when it is full. This flag is not used for JPEG encoding of camera images using DMA transfer. While the FIFO is full, the JPEG Codec circuit aborts encoding process, data accumulating in the line buffer. The user will actually adopt an approach to read the value from the FIFO Effective Size Register by using the JPEG FIFO Threshold Status Register to determine whether or not the value is greater than or equal to 1/2 or 1/4 of the FIFO size.
   This flag does not need to be checked during JPEG decoding because the DMA controller automatically determines the full state and stops DMA transfer.

6. JPEG FIFO Threshold Trigger Flag
   Because the size of FIFO is small, it is virtually impractical to use FIFO access that is driven by threshold-triggered interrupts should increase the overhead involved with the interrupt response time. For that reason, this flag does not server any special purpose.

7. Encode Size Limit Over Flag
   The Encode Limit Over Flag is used for encoding JPEG camera images. When the JPEG DMA controller is used, interrupts that are triggered by the maximum number of DMA transfers are used to prevent memory corruption. It is, therefore, the flag is not used to directly limit the maximum size of the JPEG file. Instead, it is used to detect whether or not the size of the JPEG file is close to reaching the limit. This allow, for example, the user to use it to change the value of a quantization table to a larger value.

8.5.5.5 JPEG 180° Rotation Encode

In JPEG rotation by 180°, rotation is not performed on one frame but performed in increments of lines that corresponds to MCU width of frame. It is, therefore, necessary to sort the data that is rotated by lines via software. For instance, the data in lines is written at the beginning of image data area in the JPEG file and, when rotation is performed on one frame, it is written at the end of the image area. It is, therefore, necessary to move the location to the end of the image area using software.

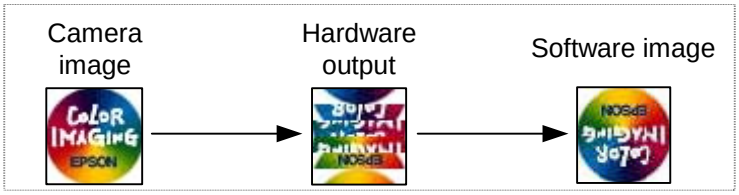


Figure 8.20 JPEG 180° Rotation Encode

Since it is difficult for software to recognize the end of line simply by detecting encoded image data, insert a RST marker at the end of each line to make the software recognize it. Thus, it is mandatory to insert RST markers. For the interval for an RST marker to be inserted, divide the width of an image by horizontal MCUs and then round the result to 0 places. For example, the interval for RST marker to be inserted when 100 pixel wide image is rotated by 180° and encoded in YUV 4:2:2 is 100/16 = 6.25 → 7.

8.5.5.6 YUV Data Format

In this device, software handles YUV data during JPEG encoding and decoding, and YUV capturing of the YUV data. In these processing modes, only YUV 4:2:2 and YUV 4:2:0 are processible formats, and YUV data units are lined up as follows.

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

YUV data is in big endian format, and the data units are lined up as U, Y, V, and Y in ascending order of address. YUV 4:2:0 uses UV data on an even line also as UV data on an odd line. Therefore, only Y data appears on odd lines.

8.5.5.7 Software Reset

In the JPEG Codec circuit, a single encoding or decoding sequence is performed on a frame-by-frame basis. Successive encoding or decoding is achieved by repeating the single encoding sequence. It is recommended that the JPEG Codec circuit be reset through software before it performs an operation. This initialization ensures stable operations. Since software reset only resets functional blocks, the values set in the registers remain intact.
In addition to the software reset on the JPEG Codec circuit, JPEG module software reset is provided to reset the YUV format converter, JPEG line buffer and JPEG FIFO components. Note that software reset recommended for camera images when performing JPEG encode process differs. This is attributable to time lag caused by processing time for each component.
To perform JPEG encoding successively, we recommend that the JPEG module be not reset through software. This is because the end of processing of current frame operates in parallel with the start of processing of the next frame and no timing at which software reset is performed is most likely to occur.
On the other hand, when the user provides an interval for one or more frames after encoding one frame, there is no possibility of overlapping of encoding process between the current and next frames. We, therefore, recommend that software reset be performed before initiating the JPEG encoding.
It is also necessary to reset the JPEG modules through software after encoding images grouped in MCUs. This

means that successive JPEG encoding must be performed on images grouped in MCUs.

8.5.5.8 Marker Fast Output Mode

The types of JPEG data are identified by markers called JPEG markers. In the JPEG Codec circuit, most of the markers are located in front of the compression data part. With earlier versions of the products, these markers are output at a clock rate of approximately 2 ms at 50 MHz system clock. For S2S65A30, new Marker Fast Output Mode is added and it is possible to complete marker outputs at a clock rate of approximately 36 μs at 50 MHz system clock by using this mode together with the JPEG Fast Encoding Mode.
Since processing is accelerated by making the Huffman tables fixed values, the Marker Fast Output Mode will be disabled if the JPEG Fast Encoding Mode is disabled regardless of the setting of the relevant register.

8.5.6 Example of Sequence

8.5.6.1 JPEG Encoding of Camera Image (Single Frame)

This section describes a sequence for encoding a single camera image. The DMA is used to read data from the FIFO.

1. Set up the camera interface. For details, see Explanation of Operation on the camera interface.
2. Enable the JPEG module (JCTL[0x00], Bit 0 = 1) and set the JPEG Operation Mode Setting Register (bits [3:1]) to “000”.
3. Reset the JPEG module through software (JCTL[0x00], bit 7 = 1).
4. Initialize the JPEG Codec Registers in ascending order. There is no dependency on the order except for the Command Register.
   (a) Reset the JPEG circuit through software (JCODEC[0x04], bit 7 = 1).
   (b) Set the JPEG encoding mode (JCODEC[0x00], bit 2 = 0).
   (c) Enable appropriate Insert Marker (JCODEC[0x00], bit 3 = 1)) to insert a desired marker.
   (d) Specify the quantization table number (JCODEC[0x0C]) and Huffman table number (JCODEC[0x10]).
   (e) Set an interval for inserting RST markers, if they are to be inserted (JCODEC[0x14], [0x18]).
   (f) Enter the size of image (JCODEC[0x1C], [0x20], [0x24], [0x28]).
   (g) Enter the marker to be inserted (JCODEC[0x04 - 0xCC]).
   (h) Set the quantization tables in the order of JCODEC[0x400 - 0x4FC] and [0x500 - 0x5FC].

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) Set the Huffman tables, for example, using data given in the ISO/IEC 10918 Annex K.

   Set A to the DC Huffman Table No.1 Register 0 (JCODEC[0x800 - 0x83C]).
   Set B to the DC Huffman Table No.0 Register 1 (JCODEC[0x840 - 0x86C]).
   Set E to the DC Huffman Table No.0 Register 0 (JCODEC[0x880 - 0x8BC]).
   Set F to the DC Huffman Table No.0 Register 1 (JCODEC[0x8C0 - 0xB44]).
   Set C to the DC Huffman Table No.1 Register 0 (JCODEC[0xC00 - 0xC3C]).
   Set D to the DC Huffman Table No.1 Register 1 (JCODEC[0xC40 - 0xC6C]).
   Set G to the DC Huffman Table No.1 Register 0 (JCODEC[0xC80 - 0xCBC]).
   Set H to the DC Huffman Table No.1 Register 1 (JCODEC[0xCC0 - 0xF44]).

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

5. Initialize the JPEG module.
   (a) Specify the size of the JPEG FIFO. Since the FIFO is a dedicated RAM, enter 0x3F. (JFIFO[0x48] = 0x3F)
   (b) Specify the limit on the size for encoding in bytes (JFIFO[0x60], [0x64]).
   (c) Clear the JPEG FIFO (JCODEC[0x04], bit 2 = 1).

6. Initialize the resizer (RSZ[0xC0], bit 7 = 1). Set it so that the size of resized image is the same value as the one specified in step 4 (f). (RSZ[0xC8], [0xCC], [0xD0], [0xD4])
7. Specify interrupts. Clear the interrupt by writing 0x000FFF in the JPEG Status Flag Register (JCTL[0x04]) and then enable the Line Buffer Overflow Interrupt and Encode Size Limit Violation Interrupt (JCTL[0x0C], bit 2 = bit 11 = 1). For the JPEG Status Flag Register, reserved bits can be written with a "1". Enable the Interrupt Enable for the JPEG module in the interrupt controller (INT[0x008]).
8. Configure the DMA controller (JDMA[0x00] to [0x40]).
9. Starts the JPEG encoding operation.
   (a) Start the JPEG circuit (JCODEC[0x04], bit 0 = 1).
   (b) Start the JPEG module (JCODEC[0x14], bit 0 = 1).
   To output the JPEG marker after starting the JPEG Codec circuit, approximately 2 ms are required at 50 MHz system clock. When the JPEG module is started before 2 ms have elapsed, the capture process actually starts after 2 ms have elapsed.
10. Wait for Frame Capture End Interrupt of the JPEG controller. If a line buffer overflow interrupt occurs during the process or the JPEG encoding process does not complete within a few minutes, perform the termination processing.

8.5.6.2 Termination

The user can use the same termination processing for all processing modes. A failed operation can be initialized by performing termination processing. Write access to the JPEG Codec circuit must be followed by a dummy read of another register, which is not included in the sequence described below.

1. Write 0x0000 in the Global Control Register (RSZ[0x60]) for the resizer in case a wrong value is written in a reserved register.
2. Enable the resizer (RSZ[0xC0], bit 0 = 1) and enable the software reset (RSZ[0xC0], bit 7 = 1).
3. Enable the JPEG module circuit (JCTL[0x00], Bit 0 = 1) and write 000 in the JPEG Operation Mode Setting Register (JCTL[0x00], bits 3-1).
4. Reset the JPEG Codec circuit through software (JCODEC[0x04], bit 7 = 1).
5. Read (dummy read) the RST Marker Operation Status Register (JCODEC[0x3C]) in the JPEG Codec circuit.
6. Read (dummy read) the JPEG Operation Status Register (JCODEC[0x08]) in the JPEG Codec circuit.
7. Write 0x00 in the Operation Mode Setting Register (JCODEC[0x00]) in the JPEG Codec circuit.
8. Reset the JPEG module circuit through software (JCTL[0x00], bit 7 = 1).
9. Write 0x0000 in the JPEG Line Buffer Interrupt Control Register (JLB[0x8C]) and disable all line buffer interrupts.
10. Write 0xFFFF in the JPEG Line Buffer Status Flag Register (JLB[0x80]).
11. Write 0x0000 in the JPEG Interrupt Control Register (JCTL[0x0C]) and disable all JPEG interrupts.
12. Write 0xFFFF in the JPEG Status Flag Register (JCTL[0x04]).
13. Reset the JPEG DMA circuit through software (JDMA[0x20], bit 15 = 1).
14. Disable the DMA Enable and JPEG Interrupt Enable in the JPEG DMA circuit (JDMA[0x0C], bit 0 = 1, JDMA[0x0C], bit 21 = 0).
15. Clear the Interrupt Flag in the JPEG DMA circuit (JDMA[0x0C], bit 1 = 0).
16. Disable the JPEG Controller Interrupt and JPEG DAM Interrupt in the interrupt controller.
17. Disable the JPEG module (JCTL[0x00], bit 0 = 0).
18. Disable the resizer (RSZ[0xC0], bit 0 = 0).

Note: The JPEG module must be disabled before disabling the resizer. If the JPEG module is disabled last, clock supply to the JPEG Coded circuit may not stop. In JPEG decoding, the resizer has been disabled previously because the decoding process does not use the resizer. If you want to disable the JPEG module, enable the resizer once before disabling the JPEG module, and then disable the resizer again.

9. JPEG_DMAC (JDMA)

9.1 Description

JPEG_DMAC is a control circuit of DMA controller, dedicated to the processing of image data sent from the camera interface.
According to the status of JPEG controller [2:1], JPEG_DMAC writes necessary register data to DMAC controller 2 and controls the data. Thus, JPEG_DMAC can properly perform DMA transfer of image data sent from the camera interface using DMA controller 2.
Request/acknowledge to DMA controller 2 is assumed by the FIFO/line buffer control circuit built in the JPEG controller.

9.2 Block Diagram

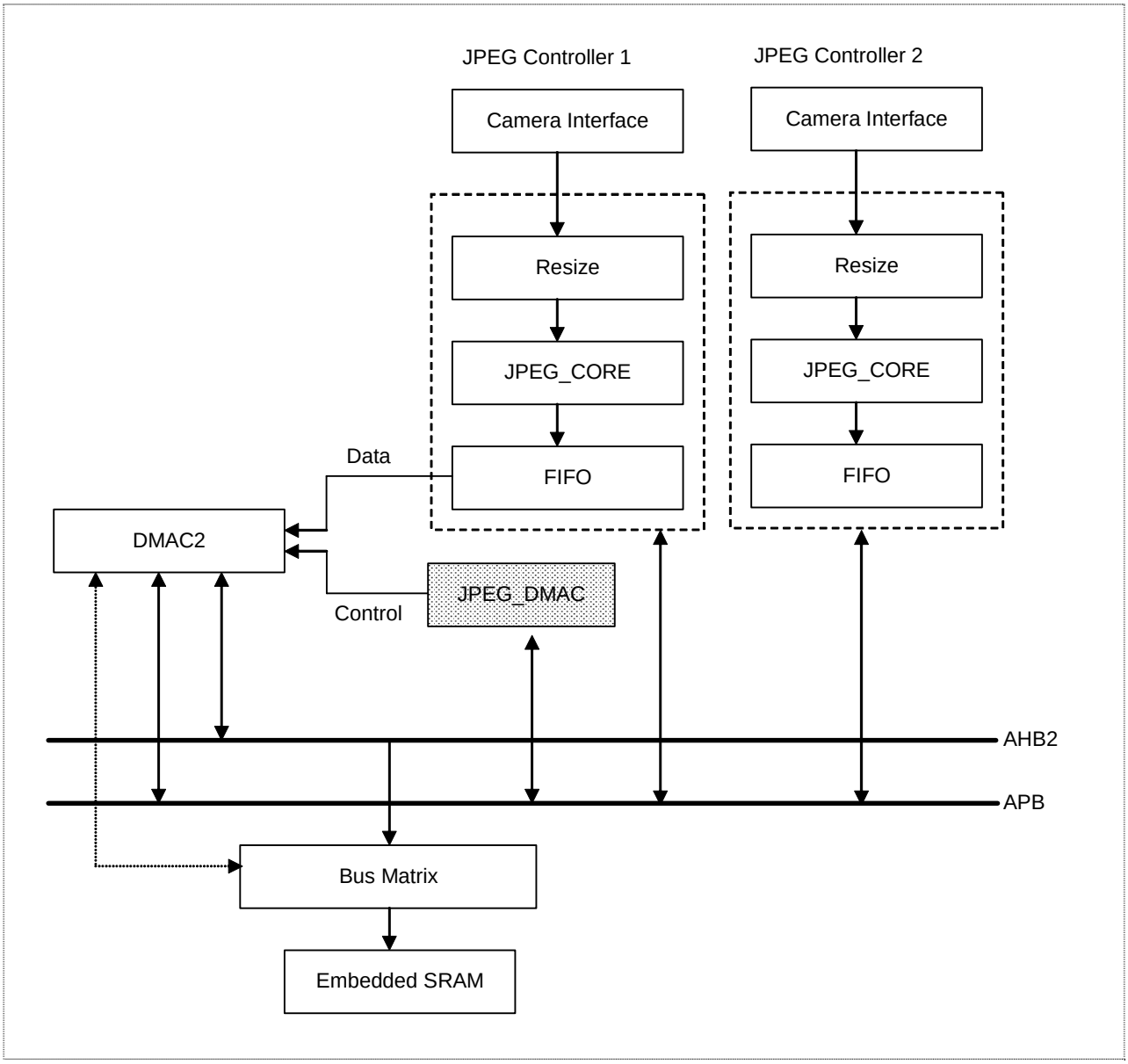


Figure 9.1 Block diagram of the relation between JPEG_DMAC and JPEG controllers, DMAC2, etc.

9.3 External Pins

The IC has no external pins for JPEG_DMAC.

9.4 Registers

9.4.1 List of Registers

The base address of the registers of JPEG_DMAC is 0xFFFE_C000.

Table 9.1 List of registers (Base address: 0xFFFE_C000)

Address offset	Register name	Register name's	Default Value *	R/W	Data access size
0x00	DMA channel 0 JPEG source address register	JSAR0	0xXXXX_XXXX	R/W	32
0x04	DMA channel 0 JPEG destination address register	JDAR0	0xXXXX_XXXX	R/W	32
0x08	DMA channel 0 JPEG transfer count register	JTCR0	0x0000_0000	R/W	32
0x0C	DMA channel 0 JPEG control register	JCTL0	0x0000_0000	R/W	32
0x10 - 0x18	—	—	—	—	—
0x1C	DMA Channel 0 JPEG encoding result size register	JERS0-	-0x0000_0000	RO-	32-
0x20	DMA channel 1 JPEG source address register	JSAR1	0xXXXX_XXXX	R/W	32
0x24	DMA channel 1 JPEG destination address register	JDAR1	0xXXXX_XXXX	R/W	32
0x28	DMA channel 1 JPEG transfer count register	JTCR1	0x0000_0000	R/W	32
0x2C	DMA channel 1 JPEG control register	JCTL1	0x0000_0000	R/W	32
0x30 - 0x38	—	—	—	—	—
0x3C	DMA Channel 1 JPEG encoding result size register	JERS1-	-0x0000_0000	RO-	32-
0x40	DMA channel JPEG FIFO data selection mode register	JFSM	0x0000_0000	R/W	32
0x48	DMA channel JPEG extended register	JHID	0x0000_0000	R/W	32
	The following lists registers added for decoding				
0x50	DMA channel 2 JPEG source address register	JSAR2	0xXXXX_XXXX	R/W	32
0x54	DMA channel 2 JPEG destination address register	JDAR2	0xXXXX_XXXX	R/W	32
0x58	DMA channel 2 JPEG transfer count register	JTCR2	0x0000_0000	R/W	32
0x5C	DMA channel 2 JPEG control register	JCTL2	0x0000_0000	R/W	32

* Note: X: Undecided (h)
Supplemental remark: It is recommended to use DMA channel 0 with JPEG controller 1 and DMA channel 1 with JPEG controller 2.

9.4.2 Detailed Description of Registers

DMA channel 0/1 JPEG source address register (JSAR0/1)

JDMA[0x00], [0x20] Default value = 0xXXXX_XXXX Read/Write

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

Bits [31:0]: **DMA channel 0/1 JPEG source address [31:0]**

Set the source addresses used for the JPEG_DMA transfer with the software. The settings are not updated by hardware.

For encoding, set the source addresses as follows:

・To use JPEG1, set 0xE000_004C(JPEG FIFO AHB address).

・To use JPEG2, set 0xD000_004C(JPEG FIFO AHB address).

For decoding, set the source addresses as follows:

・Specify the SDRAM address where the JPEG image resides.

・DMA channel 0 can be used only when using JPEG1 for decoding.

・DMA channel 1 can be used only when using JPEG2 for decoding.

DMA channel 0/1 JPEG destination address register (JDAR0/1)

JDMA[0x04], [0x24] Default value = 0xXXXX_XXXX Read/Write

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

Bits [31:0]: **DMA channel 0/1 JPEG destination address [31:0]**

Set the destination addresses of JPEG_DMA transfer with software. Each time the DMA transfer of one block is completed, the offset value set for JPEG Destination Offset Address Register of the DMA channel is added automatically to the current value.

For encoding, set the destination addresses as follows:

・Specify the external memory (SDRAM) address of the destination to which the JPEG file is transferred.

For decoding, set the destination addresses as follows.

・JPEG[2:1] Specify the FIFO address.

・Specify the JPEG1 FIFO address (0xE000_004C) for DMA channel 0.

・Specify the JPEG2 FIFO address (0xD000_004C) for DMA channel 1.

DMA channel 0/1 JPEG transfer count register (JTCR0/1)

JDMA[0x08], [0x28] Default value = 0x0000_0000 Read/Write

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

Bits [23:0]: **DMA channel 0/1 JPEG transfer count [23:0]**

Set the count of the JPEG_DMA transfer with the software. The settings are not updated by hardware

Bits [31:24] get “0” if this register is read.

DMA channel 0/1 JPEG control register (JCTL0/1)

JDMA[0x0C], [0x2C] Default value = 0x0000_0000 Read/Write

n/a						JENC	JATM	RSV	JS	JIE	JCS1	JCS0	IB4	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 25: **JENC JPEG encode destination count-up stop**

0: Usually written (compatible with previous models)

1: tops counting the destination addresses when data input from core exceeds the specified size.

Bits [24:23]: **RSV Reserved (0)**

Bit 22: **JS JPEG_DMA transfer start**

0: This bit is cleared automatically to “0” upon completion of the transfer of each one block or upon assertion of FIFO_END.

1: This bit is set to “1” upon start of JPEG_DMA transfer.

Bit 21: **JIE JPEG interrupt enable**

If this bit is set to “1”, interrupt is asserted upon setting of JTE (Bit 1).

0: Interrupt disabled

1: Interrupt enabled

Bit [20:19]: **JCS[1:0] DMA channel select**

Selects a DMA channel of the DMAC2 used.

00: DMA channel 0 (JCTL0)

01: DMA channel 1 (JCTL1)

10,11: Reserved

Bit 18: **IB4 Increment burst 4-beat transfer**

Set this bit to “1” only when transferring YUV data via DMA in YUV data capturing.

0: Single transfer

1: 4-beat burst transfer enabled

If this bit is set to “1”, the JPEG transfer counter value of DMA channels 0/1 must be a multiple of 4.

Bit 17: **AM Acknowledge mode**

Select the output timing for the DACK signal.

0: Active while in the DMA read cycle

1: Active while in the DMA write cycle

Bit 16: **AL Acknowledge level**

Select the output polarity for the DACK signal.

0: LOW active

1: HIGH active

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

Select an update mode entered upon completion of one count of transfer for the destination address register.

00: Destination address fixed (not updated)

01: Destination address incremented corresponding to the size of data transferred.
(+1 for 8 bits; +2 for 16 bits, +4 for 32 bits)

10: Destination address decremented corresponding to the size of data transferred.
(-1 for 8 bits; -2 for 16 bits, -4 for 32 bits)

11: Reserved

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

Select an update mode entered upon completion of one count of transfer for the source address register.

00: Source address fixed (not updated)

01: Source address incremented corresponding to the size of data transferred.
(+1 for 8 bits; +2 for 16 bits, +4 for 32 bits)
10: Source address decremented corresponding to the size of data transferred.
(-1 for 8 bits; -2 for 16 bits, -4 for 32 bits)
11: Reserved

Bits [11:8]: **RS Resource selection [3:0]**

Select the cause which starts up the DMA transfer.
0010: JPEG1 encoding (JCTL0 is recommended)
0011: JPEG2 encoding (JCTL1 is recommended)
1100: JPEG1 decoding (Only JCTL0 can be specified)
1101: JPEG2 decoding (Only JCTL1 can be specified)
Other: Reserved

Bits [7:6]: **RSV Reserved (0)**

Bit 5: **TM Transmission Mode**

Select a transmission mode for the DMA transfer.
0: Single transfer
1: Demand transfer

Bits [4:3]: **TS Size of data transferred**

Select the size of data sent by one count of transfer.
00: 8 bits
01: 16 bits
10: 32 bits
11: Reserved

Bit 2: **IE Interrupt enable**

0: No interrupt is caused upon completion of one count of block transfer.
1: Interrupt is caused upon completion of one count of block transfer.

Bit 1: **JTE JPEG_DMA transfer completed**

0 (when reading): Transferring or standing by
1 (when writing): JPEG_DMA transfer completed
0 (when writing): Clears this bit.
1 (when writing): Invalid

This bit is set if the DMA channel's JPEG transfer block count register gets a value "0". Once set, it holds "1" until it is cleared by "0" written in. The DMA transfer is disabled for this channel until this bit is cleared. This bit functions also as an interrupt flag.
This bit doesn't operate at the YUV capture mode. The end of the YUV capture mode uses the interruption of DMAC2.

Bit 0: **DE DMA enable**

This bit can enable the JPEG_DMA transfer.
0: JPEG_DMA transfer disabled
1: JPEG_DMA transfer enabled

JPEG encode result size register (JERD0/1)

JDMA[0x1C], [0x3C] Default value = 0x0000_0000 Read Only

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

Bits [31]: **JPEG Encode Result**

Set to "1" when encoding result exceeds the specified size.

Bits [23:0]: **JPEG Encode Result Size [23:0]**

The size is set upon completion of JPEG encoding.

9.4.3 Register Details (Registers Used for YUV Data DMA Transfer during JPEG Decoding)

DMA channel 2 JPEG source address register (JSAR2)

JDMA[0x50] Default value = 0xXXXX_XXXX Read/Write

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

Bits [31:0]: **DMA channel 2 JPEG source address [31:0]**

Set the source addresses of the JPEG_DMA transfer with software. The settings are not updated by hardware.

For decoding, set the source addresses as follows.

- Because JPEG files are stored in the line buffer inside the JPEG controller after being decoded, specify the line buffer address.
- To decode the file with JPEG1, specify 0xE000_00C0.
- To decode the file with JPEG2, specify 0xD000_00C0.
- Change the line buffer mode setting of DMA Channel JPEG FIFO Data Select Mode Register (JDMA[0x40]) into access via the AHB bus.

DMA channel 2 JPEG destination address register (JDAR2)

JDMA[0x54]] Default value = 0xXXXX_XXXX Read/Write

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

Bits [31:0]: **DMA channel 2 JPEG destination address [31:0]**

Set the destination addresses of JPEG_DMA transfer with software. Each time the DMA transfer of one block is completed, the offset value set for JPEG Destination Offset Address Register of the DMA channel is added automatically to the current value.

For decoding, set the destination addresses as follows.

- Specify the external memory (SDRAM) address in which YUV data is stored after being decoded.

DMA channel 2 JPEG transfer count register (JTCR2)

JDMA[0x58] Default value = 0x0000_0000 Read/Write

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

Bits [23:0]: **DMA channel 2 JPEG transfer count [23:0]**

Set the count of the JPEG_DMA transfer with software. The settings are not updated by hardware.
If this register is read, "0" are appended to bits [31:24].

DMA channel 2 JPEG transfer count register (JCTL2)

JDMA[0x5C] Default value = 0x0000_0000 Read/Write

RSV									JS	JIE	JCS1	JCS0	IB4	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

Bits [31:23]: **RSV Reserved (0)**

Bit 22: **JS JPEG_DMA transfer start**

0: This bit is cleared automatically to "0" upon completion of the transfer of each one block or upon assertion of FIFO_END.
1: This bit is set to "1" upon start of JPEG_DMA transfer.

Bit 21: **JIE JPEG interrupt enable**

If this bit is set to "1", interrupt is asserted upon setting of JTE (Bit 1).
0: Interrupt disabled
1: Interrupt enabled

Bit [20:19]: **JCS[1:0] DMA channel select**

Selects a DMA channel of the DMAC2 used.
00: DMA channel 0
01: DMA channel 1
10: DMA channel 2
11: DMA channel 3

Bit 18: **IB4 Increment burst 4-beat transfer**

This bit is used only when transfer YUV data via DMA. Set this bit to "1" usually in JPEG decoding. If this bit is set to "1", the JPEG transfer counter value of DMA channels 0/1 must be a multiple of 4.
0: Single transfer
1: 4-beat burst transfer enabled

Bit 17: **AM Acknowledge mode**

Select the output timing for the DACK signal.
0: Active while in the DMA read cycle
1: Active while in the DMA write cycle

Bit 16: **AL Acknowledge level**

Select the output polarity for the DACK signal.
0: Active Low
1: Active High

Bits [15:14]: **DAM destination address mode [1:0]**

Select an update mode entered upon completion of one count of transfer for the destination address register.
00: Fixes the transfer destination address (no update)

01: Increments the transfer destination address according to the transferred data size (+1 for 8-bit size, +2 for 16-bit, or +4 for 32-bit)
10: Decrements the transfer destination address according to the transferred data size (-1 for 8-bit size, -2 for 16-bit, or -4 for 32-bit)
11: Reserved

Bits [13:12]: **SAM source address mode [1:0]**

Select an update mode entered upon completion of one count of transfer for the source address register.
00: Fixes the transfer source address (no update)
01: Increments the transfer source address according to the transferred data size (+1 for 8-bit size, +2 for 16-bit, or +4 for 32-bit)
10: Decrements the transfer source address according to the transferred data size (-1 for 8-bit size, -2 for 16-bit, or -4 for 32-bit)
11: Reserved

Bits [11:8]: **RS Resource select [3:0]**

Select the cause which starts up the DMA transfer.
0110: JPEG1 YUV
0111: JPEG2 YUV
Other: Reserved

Bits [7:6]: **RSV Reserved (0)**

Bit 5: **TM transmission mode**

Select a transmission mode for the DMA transfer.
0: Single transfer
1: Demand transfer

Bits [4:3]: **TS Size of data transferred**

Select the size of data transferred by one count of transfer.
00: 8 bits
01: 16 bits
10: 32 bits
11: Reserved

Bit 2: **IE Interrupt enable**

0: No interrupt is caused upon completion of one count of block transfer.
1: Interrupt is caused upon completion of one count of block transfer.

Bit 1: **JTE JPEG_DMA transfer completed**

0 (Read): Transferring or waiting
1 (Read): DMA transfer ended
0 (Write): Clears this bit
1 (Write): Invalid

This bit is set if the DMA Channel JPEG Transfer Block Count Register is turned to “0”. Once this bit is set, the “1” value is retained until it is cleared to “0”. The DMA transfer is disabled for this channel until this bit is cleared. This bit functions also as an interrupt flag.
During JPEG decoding, an interrupt caused by DMA transfer writing to JPEG FIFO of a JPEG file (interrupt by the JTE bit of JCTL[1:0]) does not occur. Determine the termination of JPEG decoding by the interrupt of this bit, a cause of terminating transfer of decoded YUV data to external memory.

Bit 0: **DE DMA enable**

This bit can enable the JPEG_DMA transfer.
0: JPEG_DMA transfer disabled
1: JPEG_DMA transfer enabled

DMA channel JPEG FIFO data selection mode register (JFSM)

JDMA[0x40] Default value = 0x0000_0000 Read/Write

n/a		RSV									JL1B	JF1B	RSV		
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
n/a		RSV									JL0B	JF0B	RSV		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [29:24]: **RSV Reserved (0)**
Bits [21:20]: **RSV Reserved (0)**
Bits [18:16]: **RSV Reserved (0)**
Bits [13:8]: **RSV Reserved (0)**
Bits [5:4]: **RSV Reserved (0)**

Bit 20, 4: **JL[1:0]B Line buffer mode**

Makes the settings for the data output bus of JPEG Line[1:0]. This bit is available when decoding JPEG images.

0: Responds to access from the APB bus side.
1: Responds to access from the AHB bus side.

JF1B responds to JPEG2, and JF0B to JPEG1.

Bit 19, 3: **JF[1:0]B FIFO mode**

Makes the settings for the data output bus of JPEG FIFO[1:0].

0: Responds to access from the APB bus side.
1: Responds to access from the AHB bus side.

JF1B responds to JPEG2, and JF0B to JPEG1.

Bits [2:0]: **RSV Reserved (0)**

DMA channel JPEG extended register (JHID)

JDMA[0x48] Default value = 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	SRCH2	SRCH1	SRCH0	n/a										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit 15: **SW Software reset**

Causes the software reset of all the registers in JDMA. The software reset is performed by “1” written in this bit. Upon completion of the software reset, this bit returns to “0” automatically.

Bit 13: **SRCH2 Software reset CH2**

Causes the software reset of all the CH2 registers in JDMA. The software reset is performed by setting this bit to “1”. Upon completion of the software reset, this bit returns to “0” automatically.

Bit 12: **SRCH1 Software reset CH1**

Causes the software reset of all the CH1 registers in JDMA. The software reset is performed by setting this bit to “1”. Registers used also by CH1/0 are not reset. Upon completion of the software reset, this bit returns to “0” automatically.

Bit 11: **SRCH0 Software reset CH0**

Causes the software reset of all the CH0 registers in JDMA. The software reset is performed by setting this bit to “1”. Registers used also by CH1/0 are not reset. Upon completion of the software reset, this bit returns to “0” automatically.

10. DMA CONTROLLER 2 (DMAC2)

10.1 Description

DMAC2 is a DMA controller which performs the DMA transfer based on the control information sent from JPEG_DMAC or set by software.

10.2 Block Diagram

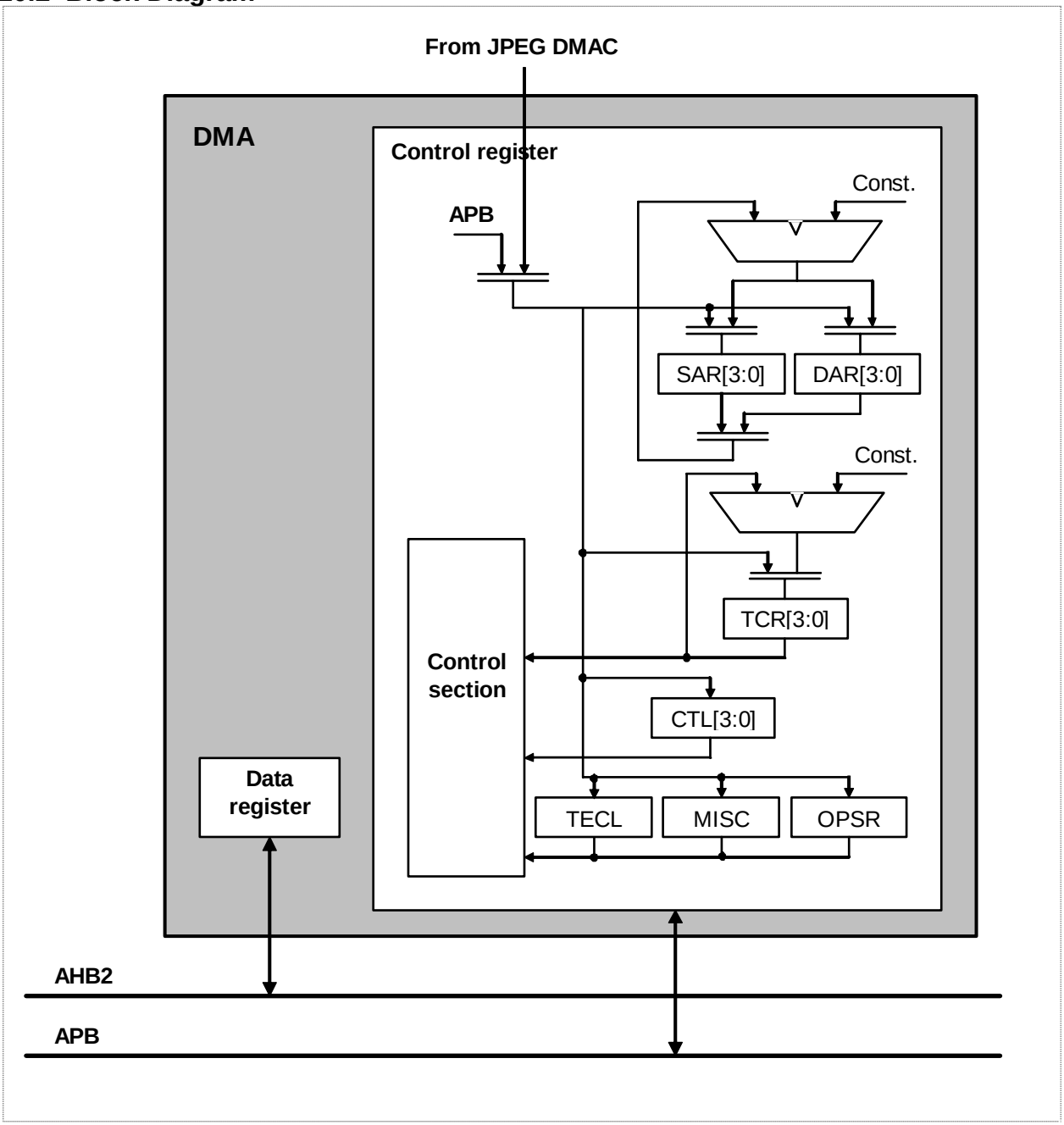


Figure 10.1 Block diagram of DMA controller 2

10.3 External Pins

The IC has no external pins for DMA controller 2.

10.4 Registers

10.4.1 List of Registers

The base address of the registers for DMAC2 is 0xFFFF_9000.

Table 10.1 List of registers (Base address: 0xFFFF_9000)

Address offset	Register name	Register name's abbreviation	Default Value	R/W	Data access size
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
0x40 -0x5C	Reserved	—	—	—	—
0x60	DMA channel Operating selection register	OPSR	0x0000_0000	R/W	32
0x64	DMA channel MISC register	MISC	0x0000_0000	R/W	32
0x70	DMA channel Transfer completion control register	TECL	0x0000_0000	R/W	32

10.4.2 Detailed Description of Registers

DMA channel [3:0] source address register (SAR[3:0])

DMAC2[0x00], [0x10], [0x20], [0x30] Default value = 0xXXXX_XXXX Read/Write

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

Bits [31:0]: **DMA channel [3:0] source address [31:0]**

Set the source address for the DMA transfer through channel [3:0] with the software. The source address set must be a boundary address compatible with the size of data transferred. In the case of 32-bit transfer, for example, "00b" must be set to bits [1:0]. Once the DMA transfer is started, the source address is updated automatically to another one for the next transfer each time one count of transfer completes according to the size of data transferred (TS: channel [3:0] control register bits [4:3]) and the source address mode (SAM: channel [3:0] control register bits [13:12]).

DMA channel [3:0] destination address register (DAR[3:0])

DMAC2[0x04], [0x14], [0x24], [0x34] Default value = 0xXXXX_XXXX Read/Write

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

Bits [31:0]: **DMA channel [3:0] destination address [31:0]**

Set the destination address for the DMA transfer through channel [3:0] with the software. The destination address set must be a boundary address compatible with the size of data transferred. In the case of 32-bit transfer, for example, "00b" must be set to bits [1:0]. Once the DMA transfer is started, the destination address is updated automatically to another one for the next transfer each time one count of transfer completes according to the size of data transferred (TS: channel [3:0] control register bits [4:3]) and the destination address mode (DAM: channel [3:0] control register bits [15:14]).

DMA channel [3:0] transfer count register (TCR[3:0])

DMAC2[0x08], [0x18], [0x28], [0x38] Default value = 0x00XX_XXXX Read/Write

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

Bits [23:0]: **DMA channel [3:0] transfer count [23:0]**

Set the DMA transfer count with the software. Once the DMA transfer starts, the setting is decremented each time one count of transfer completes. If "0" is set, the transfer count is 2^{24}=16777216. The counter value counted-down to "0" causes the DMA interrupt.

Bits [31:24] get "0" if this register is read.

DMA channel [3:0] control register (CTL[3:0])

DMAC2[0x0C], [0x1C], [0x2C], [0x3C] Default value = 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 [23:19]: **RSV Reserved (0)**

Bit 18: **IB4 Increment burst 4**

Bit value "1" causes the increment burst 4 transfer to be performed. The size of the data transferred by the increment burst 4 transfer, though, must be a multiple of four data-sizes. Otherwise, any fraction of the size of the data transferred must be transferred by separate DMA transfer by setting this bit to "0".
This function is used only for block transfer from a memory to another.

Bit 17: **AM Acknowledge mode**

Select the output timing for the DACK signal.

0: Active while in the DMA read cycle
1: Active while in the DMA write cycle

Bit 16: **AL Acknowledge level**

Select the output polarity for the DACK signal.

0: LOW active
1: HIGH active

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

Select an update mode entered upon completion of one count of transfer for the destination address register.

00: Destination address fixed (not updated)
01: Destination address incremented corresponding to the size of data transferred.
(+1 for 8 bits; +2 for 16 bits, +4 for 32 bits)
10: Destination address decremented corresponding to the size of data transferred.
(-1 for 8 bits; -2 for 16 bits, -4 for 32 bits)
11: Reserved

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

Select an update mode entered upon completion of one count of transfer for the source address register.

00: Source address fixed (not updated)
01: Source address incremented corresponding to the size of data transferred.
(+1 for 8 bits; +2 for 16 bits, +4 for 32 bits)
10: Source address decremented corresponding to the size of data transferred.
(-1 for 8 bits; -2 for 16 bits, -4 for 32 bits)
11: Reserved

Bits [11:8]: **RS Resource selection [3:0]**

Select the cause which starts up the DMA transfer.

1111: **SW-Request** Software request
Setting bits [11:8] to "1111" starts up the DMA transfer of the software.
Only the addresses mapped in 4.2.2 Memory Map (AHB2) can be set.
Other: Reserved

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

Bit 6: **RIM Request input mode**

Select an input mode for the DMA request signal.

0: LOW active (level trigger)
1: Fall edge (edge trigger)

Bit 5: **TM Transfer mode**

Select a transfer mode for the DMA transfer.

0: Single transfer
1: Demand transfer

Bits [4:3]: **TS Transfer size [1:0]**

Select the size of data transferred by one count of transfer.

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

Bit 2: **IE Interrupt enable**

Enables/disables the interrupt into DMA channel [3:0] upon completion of transfer.

0: Interrupt disabled
1: Interrupt enabled

Bit 1: **TE DMA transfer completed**

0 (when reading): Transferring or standing by
1 (when reading): JPEG_DMA transfer completed
0 (when writing): Clears this bit.
1 (when writing): Disabled

This bit is set if the transfer block count register of DMA channel [3:0] gets a value “0”. Once set, it holds “1” until it is cleared by “0” written in. The DMA transfer is disabled for this channel until this bit is cleared. This bit functions also as an interrupt flag.

Bit 0: **DE DMA enable**

This bit can enable the DMA transfer through channel [3:0].

0: DMA transfer disabled
1: DMA transfer enabled

DMA channel operating selection register (OPSR)

DMAC2[0x60] Default value = 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 change enable**

0: Priority not changed

1: Priority changed

Bit value "1" changes DPM of bit 8 under the following conditions:

Single transfer: Changes DPM at each count of transfer.

Demand transfer: Changes DPM if transfer is suspended because of negated request or if transfer counter gets "0".

Bit 8: **DPM DMA priority mode**

0: Priority is CH0>CH1>CH2>CH3.

1: Priority is CH1>CH0>CH2>CH3.

Bit 0: **DGE DMA global enable**

Enables/disables all the DMA channels.

0: Disables

1: Enables

DMA channel MISC register (MISC)

DMAC2[0x64] Default value = 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**

"1" written in this bit initializes the registers in DMAC2.

All the registers in DMAC2 are initialized, so make settings again for registers as required.

Bits [3:0]: **DPL DMA polarity selection [1:0]**

DPL3 Select the polarity of DMA channel 3.

0: Positive

1: Negative

DPL2 Select the polarity of DMA channel 2.

0: Positive

1: Negative

DPL1 Select the polarity of DMA channel 1.

0: Positive

1: Negative

DPL0 Select the polarity of DMA channel 0.

0: Positive

1: Negative

DMA channel transfer completion control register (TECL)

DMAC2[0x70] Default value = 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 upon reception of completion of transfer**

"1" written in this bit sets TE upon receiving the transfer completion signal from a request-issuing cause. This bit becomes valid when bit 12 is set to "1".

Bits 12: **ENTE Transfer completion reception enable**

"1" written in this bit enables reception of the transfer completion signal from a request-issuing cause, without setting TE; Write "1" to bit 13 to make TE set.

11. USB HS-DEVICE

11.1 Description

This USB HS-Device Macro is made by extracting the USB HS-Device Controller section mounted on the USB2.0-compatible SEIKO EPSON High Speed USB-Host/Device Combination Controller (S1R72V05) and rebuilding it as a macro for the use on a SoC (System On Chip).
See "Appendix2 USB Device Controller" for more details of the HS-Device Macro.

11.2 Block Diagram

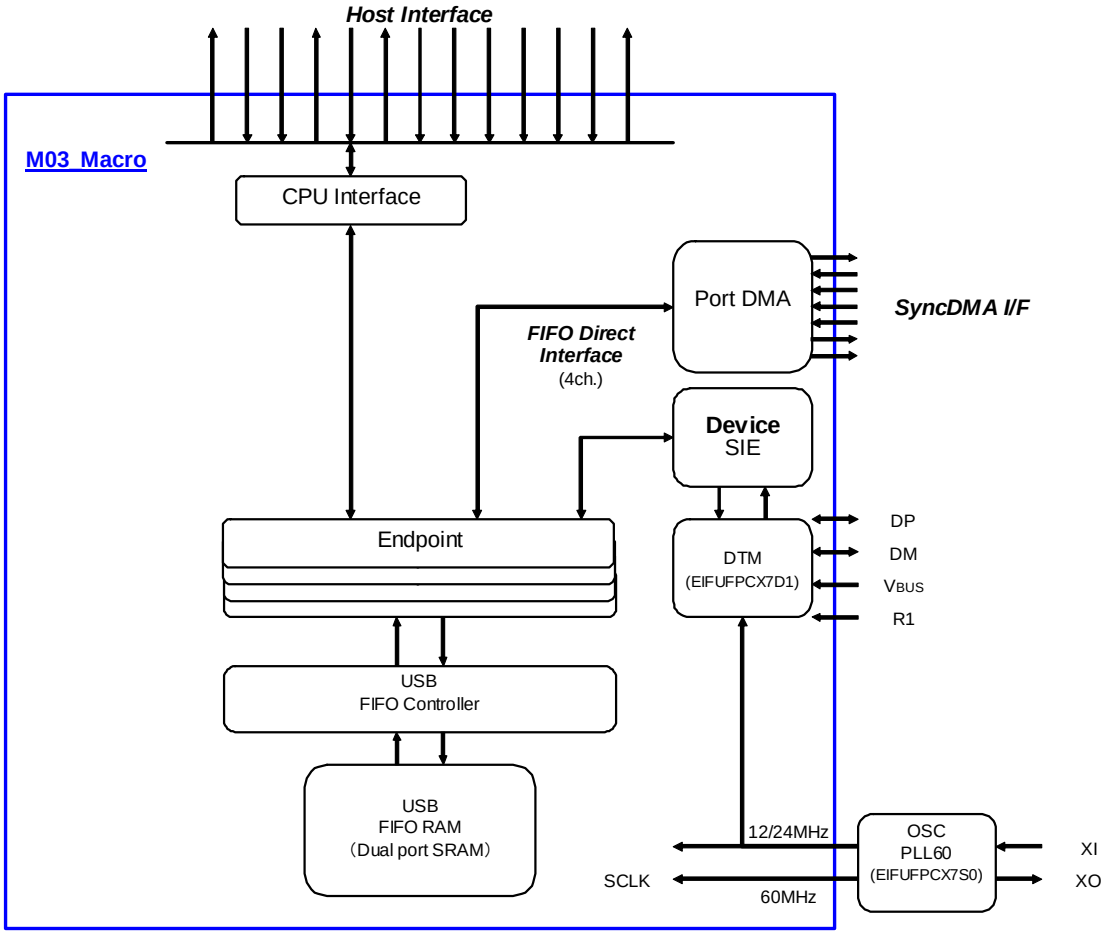


Figure 11.1 Block diagram of USB HS-Device

11.3 Features

The USB HS-Device Macro features the following basic functions.

<USB2.0 Device Function>
・Supports transfer of HS (480Mbps) and FS (12Mbps).
・Built in FS/HS Termination function (no external circuit needed)
・VBUS 5V Interface (an external protection circuit needed)
・Supports Control Bulk and Interrupt Transfer.
・Supports the End Points (5 in total) shared by Control (End Point 0) and Bulk/Interrupt.

<CPU Interface>
・General-purpose CPU Interface with the 16-bit or 8-bit width
・Compatible with Littler Endian.
・Register Table compiled by addition/deletion made to that of the HS-Device of S1R72V05.

< Oscillation Circuit >
・USB clock input compatible with the 12MHz or 24MHz crystal transducer.
(1MΩ built in for the oscillation circuit and as the feedback resistor)
・The following frequencies are available for internal clock based on the USB input clock.
USB input clock: 12MHz or 24MHz
Internal USB clock: 60MHz (via PLL for the built-in USB)

12. APB BRIDGE (APB)

12.1 Description

This module, mounted on the APB bus connecting the internal high-speed bus AHB1 and the low-speed APB-functional devices (APB Devices), is a slave device on the AHB bus performing the bridge function to control the AHB bus on behalf of each APB Device. With this APB bridge, each APB Device is discharged from the responsibility to control the AHB bus, and it has only to control the simple APB bus. This APB bridge is irrelevant to a software programmers usually; they are required occasionally, though, to make a setting of the wait function (1 to 3 wait) to the APB Devices. Use it normally as it is reset.

12.2 Block Diagram

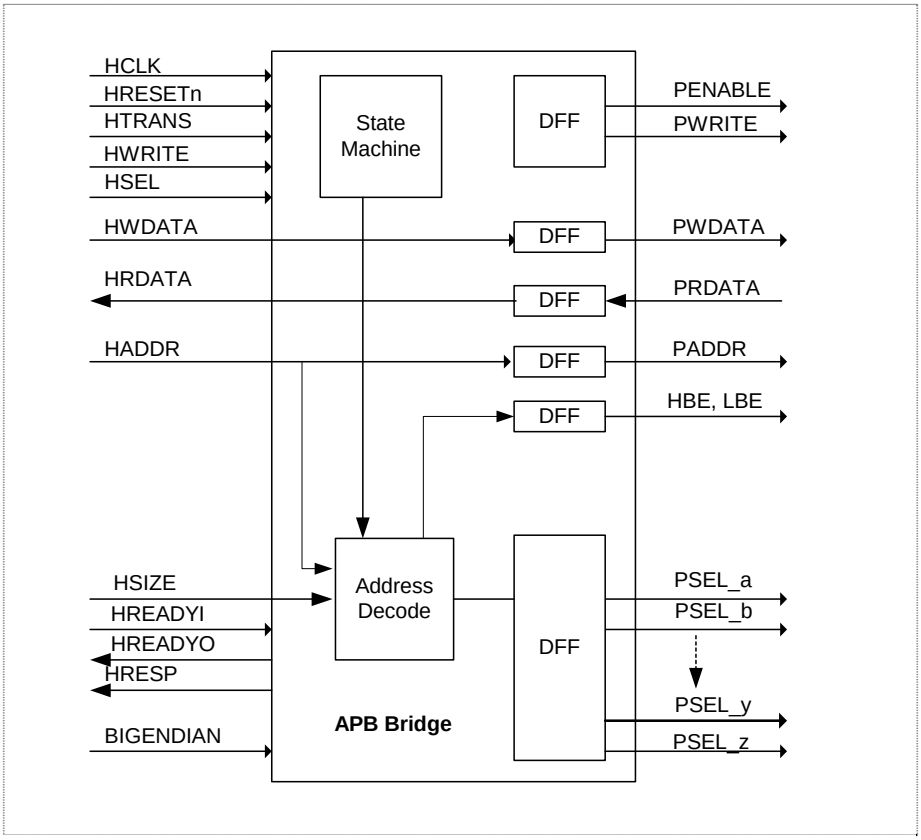


Figure 12.1 Block diagram of the APB bridge

12.3 External Pins

The IC has no external pins for the APB bridge.

12.4 Registers

12.4.1 List of Registers

The base address of the registers for APB is 0xFFFE_0000.

Table 12.1 List of registers (Base address: 0xFFFE_0000)

Address offset	Register name	Register name's abbreviation	Default	R/W	Data access size
0x00	APB WAIT0 Register	APBWAIT0	0x0050_0500	R/W	32bit
0x04	APB WAIT1 Register	APBWAIT1	0x0000_0000	R/W	32bit
0x08	APB WAIT2 Register	APBWAIT2	0x0050_0000	R/W	32bit

12.4.2 Detailed Description of Registers

APB WAIT0 register (APBWAIT0)

APB[0x00] Default value = 0x0050_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 [31:0]: **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

APB WAIT1 register (APBWAIT1)

APB[0x04] Default value = 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 [31:0]: **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

APB WAIT2 register (APBWAIT2)

APB[0x08] Default value = 0x0050_0000 Read/Write

PW2FCNF [1:0]		PW2ECNF [1:0]		PW2DCNF [1:0]		PW2CCNF [1:0]		PW2BCNF [1:0]		PW2ACNF [1:0]		PW29CNF [1:0]		PW28CNF [1:0]	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PW27CNF [1:0]		PW26CNF [1:0]		PW25CNF [1:0]		PW24CNF [1:0]		PW23CNF [1:0]		PW22CNF [1:0]		PW21CNF [1:0]		PW20CNF [1:0]	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [31:0]: **PWxCNF[1:0] (x=20 to 2F)**

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 List of APBWAIT[2:0] registers and corresponding APB Devices

APBWAIT0		**APBWAIT1**		**APBWAIT2**	
PW0xCNF	APB Device	PW1xCNF	APB Device	PW2xCNF	APB Device
PW00CNF	APB bridge	PW10CNF	Reserved	PW20CNF	Reserved
PW01CNF	IPC-1	PW11CNF	GPIO	PW21CNF	IPC-2
PW02CNF	Reserved	PW12CNF	SPI	PW22CNF	Reserved
PW03CNF	DMA1C	PW13CNF	DMAC3	PW23CNF	Timer B
PW04CNF	CF Attribute CF Common	PW14CNF	Reserved	PW24CNF	UART2
PW05CNF	CF I/O	PW15CNF	UART1	PW25CNF	UART3
PW06CNF	CF CTL	PW16CNF	Reserved	PW26CNF	Reserved
PW07CNF	ARS	PW17CNF	Reserved	PW27CNF	Reserved
PW08CNF	Camera Interface-1	PW18CNF	RTC	PW28CNF	Camera Interface-2
PW09CNF	JPEG Resize-1	PW19CNF	DMA2C	PW29CNF	JPEG Resize-2
PW0ACNF	JPEG module/FIFO-1	PW1ACNF	Memory CNTL	PW2ACNF	JPEG module/FIFO-2
PW0BCNF	JPEG Codec-1	PW1BCNF	Timer A	PW2BCNF	JPEG Codec-2
PW0CCNF	JPEG-DMAC	PW1CCNF	WDT	PW2CCNF	ADC
PW0DCNF	I^2C	PW1DCNF	SYS-CNTL	PW2DCNF	Reserved
PW0ECNF	I^2S	PW1ECNF	Reserved	PW2ECNF	Reserved
PW0FCNF	(INT-CNTL)	PW1FCNF	INT-CNTL	PW2FCNF	USB_2.0 Device

Table 12.3 Built-in I/O map

Base Address	Space Size	S2S65A30
0xFFFD_0000	4KB	Reserved
0xFFFD_1000	4KB	**IPC-2**
0xFFFD_2000	4KB	Reserved
0xFFFD_3000	4KB	**Timer B**
0xFFFD_4000	4KB	**UART2**
0xFFFD_5000	4KB	**UART3**
0xFFFD_6000	4KB	Reserved
0xFFFD_7000	4KB	Reserved
0xFFFD_8000	4KB	**Camera Interface-2**
0xFFFD_9000	4KB	**JPEG Resize-2**
0xFFFD_A000	4KB	**JPEG module/FIFO-2**
0xFFFD_B000	4KB	**JPEG Codec-2**
0xFFFD_C000	4KB	**ADC**
0xFFFD_D000	4KB	Reserved
0xFFFD_E000	4KB	Reserved
0xFFFD_F000	4KB	**USB_2.0 Device**
0xFFFE_0000	4KB	**APB bridge**
0xFFFE_1000	4KB	**IPC-1**
0xFFFE_2000	4KB	Reserved
0xFFFE_3000	4KB	**DMA1C**
0xFFFE_4000	2KB	CF attribute
0xFFFE_4800	2KB	CF common
0xFFFE_5000	2KB	CF I/O
0xFFFE_5800	1KB	CF ture IDE CS1#
0xFFFE_5C00	1KB	CF ture IDE CS2#
0xFFFE_6000	4KB	CF control
0xFFFE_7000	4KB	**ARS**
0xFFFE_8000	4KB	**Camera Interface-1**
0xFFFE_9000	4KB	**JPEG Resize-1**
0xFFFE_A000	4KB	**JPEG module/FIFO-1**
0xFFFE_B000	4KB	**JPEG Codec-1**
0xFFFE_C000	4KB	**JPEG-DMAC**
0xFFFE_D000	4KB	**I^2C**
0xFFFE_E000	4KB	**I^2S**
0xFFFE_F000	4KB	(INT-CNTL)
0xFFFF_0000	4KB	Reserved
0xFFFF_1000	4KB	**GPIO**
0xFFFF_2000	4KB	**SPI**
0xFFFF_3000	4KB	**DMA3C**
0xFFFF_4000	4KB	Reserved
0xFFFF_5000	4KB	**UART1**
0xFFFF_6000	1KB	Reserved
0xFFFF_7000	4KB	Reserved
0xFFFF_8000	4KB	**RTC**
0xFFFF_9000	4KB	**DMA2C**
0xFFFF_A000	4KB	**Memory CNTL**
0xFFFF_B000	4KB	**Timer A**
0xFFFF_C000	4KB	**WDT**
0xFFFF_D000	4KB	**SYS-CNTL**
0xFFFF_E000	4KB	Reserved
0xFFFF_F000	4KB	**INT-CNTL**

13. SYSTEM CONTROLLER (SYS)

13.1 Description

This block controls clocks, means to get the system lower powered, matters involving the chip and the entire system, and memory map mainly.
This block features:

- Supports transition to the low-power mode (IDLE) by means of HALT.
- Allows selection of ON/OFF of the CPU and bus clocks within HALT.
- Allows dynamic switching of frequencies of the CPU/AHB/APB clocks.
- Allows turning ON/OFF the clock for each I/O built in.
- Supports the operation mode in 32KHz.
- Supports the means to switch the PLL clock.
- Supports software reset.
- Programmable Clock generator for UART

13.2 Operation mode

S2S65A30 has four operational states: Low-speed, low-speed HALT, high-speed, and high-speed HALT modes.

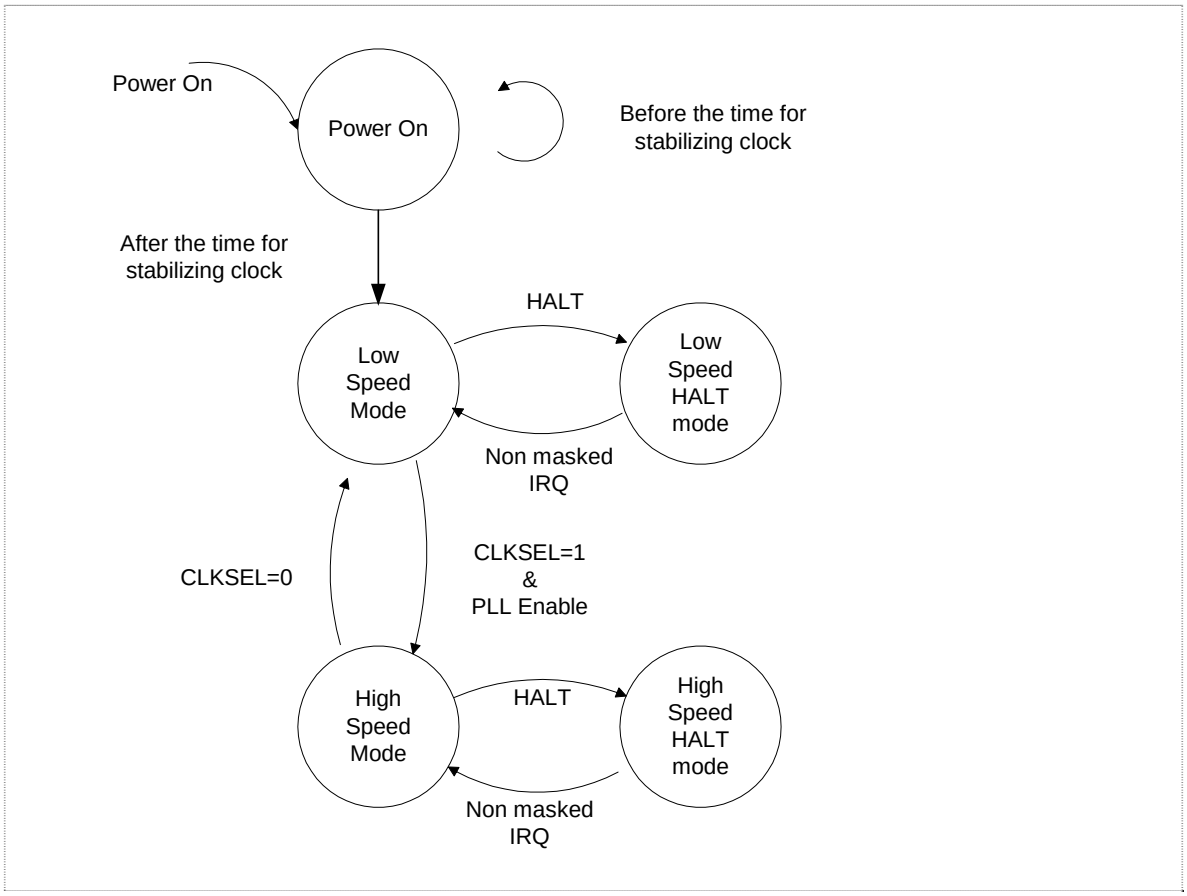


Figure 13.1 Operational states inside the system

Note: Transition from the Low Speed Mode to High Speed Mode can be made by the software after waiting the PLL stabilizing time (Max. 100ms).

Each operational state (mode) is described in the following pages.

13.2.1 Power-On State

This is not a special operational mode. The reset signal generated by turning power on makes the system enter this mode. The clock input of 32KHz is used. Assuming external 32KHz, it is necessary to secure 3 seconds at the maximum for the rise time, and this state is held for such duration. The state transits to the next operational mode, or the low-speed mode, about 3 seconds after the resetting is released. The state transits to the low-speed mode, the next operation mode when the time set by RSTTIM (SYS[0x04]) is passed after the resetting is released.

13.2.2 Low-Speed Mode

PLL is stopped in this mode immediately after resetting. This mode is operational only in 32KHz. The CPU starts in this state immediately after power-on resetting.
Setting PLL in this mode makes it possible to set any multiplying factor to the clock (though there are some restrictions on UART and the basic clock generated by a timer). PLL is operated at the frequency set (i.e. in the high-speed mode) switched to by the software after waiting some time under the software control because PLL takes some time (Max:100ms) before it is stabilized. Any change in the PLL settings required in the high-speed mode must be made in the low-speed mode.

13.2.3 Low-Speed HALT Mode

A command issued in the low-speed mode instructing the HALT operation makes the system transit into this mode, which consumes power the least among all the modes. This mode can stop both built-in I/O bus and the supply of the clock to it, which are dependent on their register settings. Exit from this mode (returning to the low-speed mode always) is caused by unmasked interrupt. For example, change in the interrupt-enabled GPIO input, external interrupt pin, or interrupt from the timer operating at 32KHz can cause exit from this mode.

13.2.4 High-Speed Mode

The system transits into this mode under the software control after PLL is started in the low-speed mode, operating with the basic clock set to the PLL at a multiplying factor which is supplied to the CPU and internal buses.

13.2.5 High-Speed HALT Mode

A command issued in the high-speed mode instructing the HALT operation makes the system transit into this mode. In this mode, both clock-stop of each I/O bus and CPU set before entering this mode are valid at the same time. Accordingly, the low-power operation is attained by using this mode frequently, i.e. by issuing HALT whenever CPU is not required to operate. Unmasked interrupt returns this mode to the high-speed mode. For example, enabled timer-interrupt or UART reception interrupt can cause interrupt, allowing subsequent transition to the high-speed mode promptly.

13.3 External Pins

The IC has the external pins for the system controller as follows.

Pin name	Input/Output	Pin functions	Multiplex pin/Remarks
RESET#	I	Hardware reset input	None
SYSCLKI	I/O	32kHz oscillator clock input/output	32kHz output is available when crystal oscillator is used
SYS_OSCI	I	32kHz crystal oscillator input	None
SYS_OSCO	O	32kHz crystal oscillator output	None
SYS_CLKSEL	I	32kHz clock source select	None

13.4 Registers

13.4.1 List of Registers

Shown below are the system control registers. The base address of these registers is 0xFFFF_D000.

Table 13.1 List of registers (Base address: 0xFFFF_D000)

Address Offset	Register Name	Abbreviation Name	Initial Value	R/W	Data Access Size
0x00	Chip ID Register	CHIPID	0x065A_300X	RO	32bit
0x04	Chip Configuration Register	CHIPCFG	0x0000_XXXX	RO	16/32bit
0x08	PLL Setting Register 1	PLLSET1	0x0423_4930	R/W	32bit
0x0C	PLL Setting Register 2	PLLSET2	0x0000_0000	(R/W)	16/32bit
0x10	HALT Mode Clock Control Register	HALTMODE	0x0000_0000	R/W	16/32bit
0x14	IO Clock Control Register	IOCLKCTL	0x0000_0000	R/W	16/32bit
0x18	Clock Select Register	CLK_SEL	0x0000_0000	R/W	16/32bit
0x1C	HALT Control Register	HALTCTL	—	WO	16/32bit
0x20	Memory Remap Register	REMAP	0x0000_0000	R/W	16/32bit
0x24	Software Reset Register	SOFTRST	—	WO	32bit
0x28	UART1 Clock Divider Register	UART1DIV	0x0000_0000	R/W	16/32bit
0x2C	UART2 Clock Divider Register	UART2DIV	0x0000_0000	R/W	16/32bit
0x30	UART3 Clock Divider Register	UART3DIV	0x0000_0000	R/W	16/32bit
0x34	Timer-B Clock Select Register	TIMBCLSEL	0x0000_0000	R/W	16/32bit
0x38	ADC Sampling Clock Divide Register	ADCCKDIV	0x0000_000A	R/W	16/32bit
0x40	MD Bus Pull-down Control Register	MDPLDCTL	0x0000_0000	R/W	16/32bit
0x44	SDD Bus Pull-down Control Register	SDDPLDCTL	0x0000_0000	R/W	16/32bit
0x48	GPIOE Resistor Control Register	PORTERCTL	0x0000_0000	R/W	16/32bit
0x4C	GPIOF Resistor Control Register	PORTFRCTL	0x0000_0000	R/W	16/32bit
0x50	GPIOG Resistor Control Register	PORTGRCTL	0x0000_0000	R/W	16/32bit
0x54	GPIOH Resistor Control Register	PORTHRCTL	0x0000_0000	R/W	16/32bit
0x58	GPIOI Resistor Control Register	PORTIRCTL	0x0000_0000	R/W	16/32bit
0x5C	GPIOJ Resistor Control Register	PORTJRCTL	0x0000_0000	R/W	16/32bit
0x60	GPIOK Resistor Control Register	PORTKRCTL	0x0000_0000	R/W	16/32bit
0x64	MISC Resister Control Register	MISCRCTL	0x0000_0000	R/W	16/32bit
0x68	Embedded Memory Control Register	EMBMEMCTL	0x0000_0000	R/W	32bit
0x6C	(Reserved)	(Reserved)	0x0000_0000	R/W	32bit
0x70	(Reserved)	(Reserved)	0x0000_0000	R/W	32bit

13.4.2 Detailed Description of Registers

Chip ID Register (CHIPID)

SYS[0x00] Default value = 0x065A_300X — Read Only

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

Bits[31:8]: **Product ID Code [23:0]**
This chip has “065A30h” in the hexadecimal format embedded here.

Bits[7:3]: **Reserved**

Bits[2:0]: **Revision Code [2:0]**
Shows the revision number of this IC. It is 01h for S2S65A30B00A000.

Chip Configuration Register (CHIPCFG)

SYS[0x04] Default value = 0x0000_XXXX — Read Only

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
CONF [15:8]							
7	6	5	4	3	2	1	0
CONF [7:3]					RSTTIM[1:0]		Reserved

Bits[15:0]: **CONF [15:3]**
CNF[15:3] are used to decide the internal operation by sampling the state of the pull-up/pull-down resistor connected to the MD[15:3] data bus at the rise of RESET#.
For more detailed functions, see “4.1 System Configuration”.

Bits[2:1]: **RSTTIM [1:0]**
RSTTIM[1:0] is used to set an interval between the time when the RESET# pin is turned to High and the time when the resetting inside IC is released. Be sure to use the bits with proper setting according to oscillation stability time specified for connected crystal oscillator.
This register is read only, and cannot be set other than for pulling up/down the MD[1:0] pin via resistor.

00: Approx. 3 sec. (compatible with previous models)
01: Approx. 100 msec.
10: Approx. 20 msec.
11: Approx. 2 msec.

Bit 0: **Reserved**
Reserved for test use.
Pull down the MDO pin via pull-down resistor.

PLL Setting Register 1 (PLLSET1)

SYS[0x08] Default value = 0x0423_4930 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

This register decides the operational frequency of the built-in PLL. To change a setting, be sure to put PLL in the power-down state first by setting "0" to bit 1 (PLLN) of the PLL Setting Register 2 (SYS[0x0C]). For more concrete values recommended, see Appendices A and B. For details of recommended values, see "PLL Setting Example" and "PLL Parameter table".

Bits [31]: **Reserved (0)**

Bits [30:29]: **CS [1:0]**
Adjust the capacity of LPF of the built-in PLL. Set the recommended values.

Bits [28:24]: **CP [4:0]**
Adjust the current of CP of the built-in PLL. Set the recommended values.

Bits [23:20]: **RS [3:0]**
Adjust the resistance of LPF of the built-in PLL. Set the recommended values.

Bits [19:16]: **VC[3:0]**
Parameters to operate VCO in the built-in PLL. Set the recommended values.

Bits [15:12]: **NN value N-Counter [3:0]**
N-Counter is used to decide the multiplying factor for PLL together with the L-Counter mentioned below. For more details, see the descriptions of LL Value and the examples of settings for PLL.

Bits [11:10]: **V-Divider [1:0]**
Specify a value to divide the PLL-Out frequency with in the built-in PLL.
00: Disabled
01: 1/2 division
10: 1/4 division
11: 1/8 division
Set the recommended values.

Bits [9:0]: **LL Value L-Counter [9:0]**
Values which decide the multiplying factor for the built-in PLL. Set the recommended values.

PLL Output = (N-Counter+1) × (L-Counter+1) × CLKI_L
= NN × LL × CLKI_L

NN = N-Counter, LL = L-Counter, CLKI_L = External clock input (32.768KHz)

PLL Setting Register 2 (PLLSET2)

SYS[0x0C] Default value = 0x0000_0000 (Read/Write)

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							
7	6	5	4	3	2	1	0
							PLLEN

Bits [7:1]: **Reserved (0)**

Bit 0: **PLLEN**

Specify whether Enable or Disable the built-in PLL. The value of PLL Setting Register1 can be changed only if PLLEN = 0.

0: PLL is Disabled. (PLL stays in the power-down state.)

1: PLL is Enabled.

HALT Mode Clock Control Register (HALTMODE)

SYS[0x10] Default value = 0x0000_0000 Read/Write

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
Reserved (0)							
7	6	5	4	3	2	1	0
CPUCKSEL [1:0]		n/a		HALT_MDCLK [4:0]			

This register decides the clock frequencies of CPU and internal buses (AHB1, AHB2, APB Bus) and which clock to stop during the HALT mode.

Bits [31:8]: **Reserved (0)**

Bits [7:6]: **CPUCKSEL [1:0]**

Specify the clock division for CPU/AHB1/AHB2/ AHB3/APB.

CPU, AHB1, AHB2, and APB have all the same clock value. If the CPU clock frequency is divided, the AHB1, AHB2, and APB clock frequencies all get the same value. Any change in the clock frequency is reflected immediately after it is written in without any glitch.

00: 1/1 of the PLL output is supplied to CPU.

01: 1/2 of the PLL output is supplied to CPU.

10: 1/4 of the PLL output is supplied to CPU.

11: 1/8 of the PLL output is supplied to CPU.

Bits [3:0]: **HALT_MDCLK [3:0]**

Bits [4:0], if "0" is written in a bit or bits, stop supplying clock to the corresponding internal bus or buses (CPU, AHB1-Bus, AHB2-Bus, or APB-bus, or all of them) during the HALT mode. Unmasked interrupt causes exit from the HALT mode. Accordingly, it is necessary to keep supplying clock to an internal device (UART, Timer, Ethernet, or GPIO) in which to cause interrupt.

0: Clock keeps to be supplied. (Clock On)

1: Clock stops to be supplied. (Clock Off)

Bit3: Specify clock On/Off for ARM720T.

Bit2: Specify clock On/Off for AHB1.*

Bit1: Specify clock On/Off for AHB2 and AHB3.

Bit0: Specify clock On/Off for APB.

"1" written in bits [3:0] to specify clock Off does not stop the clock immediately; this specification becomes valid only when the HALT mode is entered in by means of HALT Control Register. This function serves for saving power while CPU is idle and standing by, for example when the CPU has no job to execute or it is waiting for some interrupt event.

* **Restriction on use: If AHB1 bus clock is stopped by HALT, CPUCLK must be specified to be stopped by HALT, too. CPUCLK may be specified to be stopped by HALT without restriction independently from the AHB1 bus clock.**

IO Clock Control Register (IOCLKCTL)

SYS[0x14] Default value = 0x0000_0000 Read/Write

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

Each bit of this register specifies whether to supply or stop supplying clock to the corresponding I/O (Timer, UART, SPI, etc.). "1" written in causes supply of clock, and "0" stops the supply. Writing "0" for any unused I/O function serves for saving power consumption.

Bit 18: **Reserved (0)**

Bits [17:0]: **IOCLKCTL [17:0]**

Bit17 (IPC2_CLKEN) : Specify clock On/Off for IPC 2.
Bit16 (IPC1_CLKEN) : Specify clock On/Off for IPC 1.
Bit15 (DMA3C_CLKEN) : Specify clock On/Off for DMA3C.
Bit14 (ADC_CLKEN) : Specify clock On/Off for ADC.
Bit13 (UART1_CLKEN) : Specify clock On/Off for UART1.
Bit12 (UART2_CLKEN) : Specify clock On/Off for UART2.
Bit11 (UART3_CLKEN) : Specify clock On/Off for UART3.
Bit10 (DMA1C_CLKEN) : Specify clock On/Off for DMA1C.
Bit9 (DMA2C_CLKEN) : Specify clock On/Off for DMA2C.
Bit8 (SPI_CLKEN) : Specify clock On/Off for SPI.
Bit7 (I2C_CLKEN) : Specify clock On/Off for I^2C.
Bit6 (I2S_CLKEN) : Specify clock On/Off for I^2S.
Bit5 (TIMERA_CLKEN) : Specify clock On/Off for Timer Ch0/1/2.
Bit4 (TIMERB_CLKEN) : Specify clock On/Off for Timer Ch0/1/2.
Bit3 (CF_CLKEN) : Specify clock On/Off for CF Card Interface.
Bit2 (SDMMC-IF) : Specify clock On/Off for SDMMC-Interface.
Bit1 (USB_CLKEN) : Specify clock On/Off for USB Interface.
Bit0 (Reserved) : Reserved

Clock Select Register (CLK_SEL)

SYS[0x18] Default value = 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		Reserved		N/a			CLKSEL
7	6	5	4	3	2	1	0

Bit [31:6]: **Reserved (0)**

Bit [5] : **SYSCLKOEN**

Select whether or not to output 32 kHz clock from the SYSCLKI pin when the SYSCLKSEL pin is turned to Low, that is, when 32 kHz clock is supplied by the crystal oscillator.

0: The clock is not output from the SYSCLKI pin (Default)
1: The clock is output from the SYSCLKI pin

Bits[4:1]: **Reserved (0)**

Bit 0: **CLKSEL**

Decides which of 32KHz or PLL output to use as the system clock. To specify “use PLL output”, first make settings for the PLL parameters, enable PLL, secure proper PLL stabilization time (100ms), and set “1” to this bit for changing to PLL output.
While PLL is in a power-down state, the system clock is not switched to PLL even if this bit is set to “1”. If you want to use PLL output as the system clock, also set the SYS[0x0C] register.
Also when PLL is powered down while it is operated as the system clock, 32kHz output is used for the system clock.

0: 32KHz
1: PLL output

HALT Control Register (HALTCTL)

SYS[0x1C] Default value = — Write Only

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

Bits [31:0]: **Halt Control [31:0]**

Any value written in this register causes the chip to enter the HALT mode.

Memory Remap Register (REMAP)

SYS[0x20] Default value = 0x0000_0000 Read/Write

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							
7	6	5	4	3	2	1	0
						REMAP2	REMAP1

This register allows making changes in memory map after reset. The Remap function assigns the SDRAM space to the space visible at address "0x0" after resetting. For more details, see Appendix to SYSTEM CONTROLLER. The register needs not be used usually. If an OS running on a RAM basis is used, though, the remapping may make the use of the memory map easier.

Bit1: **REMAP2**

Allows memory map on the AHB2 and AHB3 bus side to be changed.

0: Applies the memory map after reset.

1: Makes a change in the memory map.

Bit0: **REMAP1**

Allows making a change in memory map on the AHB1 Bus side.

0: Applies the memory map after reset.

1: Makes a change in the memory map

Note: Take care in the operation to make a change in the memory map by executing the change code in a field unaffected by the memory map, for example.
Also it is recommended to set REMAP1 and REMAP2 simultaneously to avoid any contradiction in memory maps between AHB1/AHB2/AHB3.

Software Reset Register (SOFTRST)

SYS[0x24] Default value = — Write Only

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

Bits [31:0]: **Software Reset [31:0]**

Writing "AA5555AAh" in this register initializes all the registers in this IC (S2S65A30) and resets the CPU.

UART1/2 Clock Divider Register (UART1/2DIV)

SYS[0x28/2C] Default value = 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
UART 1/2 CLKDIV [7:0]							
7	6	5	4	3	2	1	0

This register is used for a baud-rate TIME-based divider for UART1/2. In this IC, the PCLK clock is divided basically into its division rate × 1/2 frequency, and supplied to the UART's SCLK (= baud-rate generating clock).

Bit [31]: **UART1/2 DIVMODE**

Set division rate 1/N for the last step of the frequency divider.

0: 1/2 division

1: 1/3 division

Bits [7:0]: **UART1/2CLKDIV [7:0]**

Generate a division rate 1/N. (N = Bits[7:0]+1)

0: 1/1

1: 1/2 division

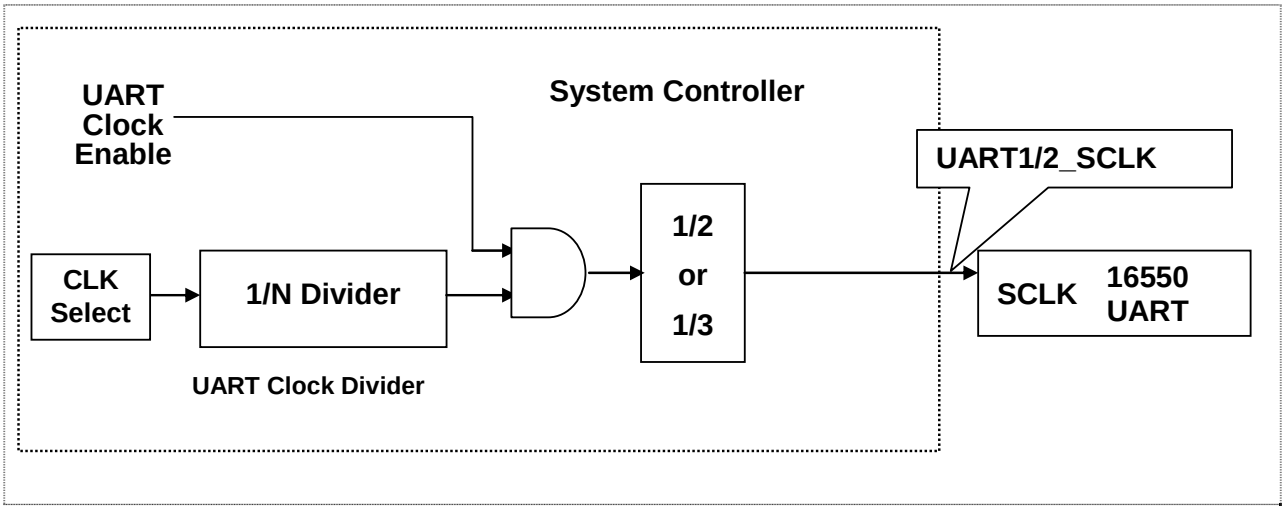
•

•

255: 1/256 division

UART1/2 is supplied with UART1/2_SCLK which is a clock divided into 1/2 of the divided UART1/2CLKDIV. Thus, UART1/2_SCLK = (PCLK's frequency) * 1/N * 1/2. (See the following chart.)

Note: This UART1/2_SCLK is very different from SCLK, the clock for SPI Interface.
To use UART1/2, be sure to write to this register once.



UART3 Clock Divider Register (UART3DIV)							
SYS[0x30] Default value = 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							UART3_CKSEL
15	14	13	12	11	10	9	8
UART 3 CLKDIV [7:0]							
7	6	5	4	3	2	1	0

This register is used for a baud-rate TIME-based divider for UART1/2/3. In this IC, the PCLK clock is divided basically into its division rate × 1/2 frequency, and supplied to the UART's SCLK (= baud-rate generating clock).

Bit [31]: **UART3 DIVMODE**

Set division rate 1/N for the last step of the frequency divider.

0: 1/2 division
1: 1/3 division

Bits8: **UART3CKSEL**

Allows selection of the external input clock for UART3 only.

0: Internal clock (Refer to Bit[7:0] for division rate.)
1: External clock

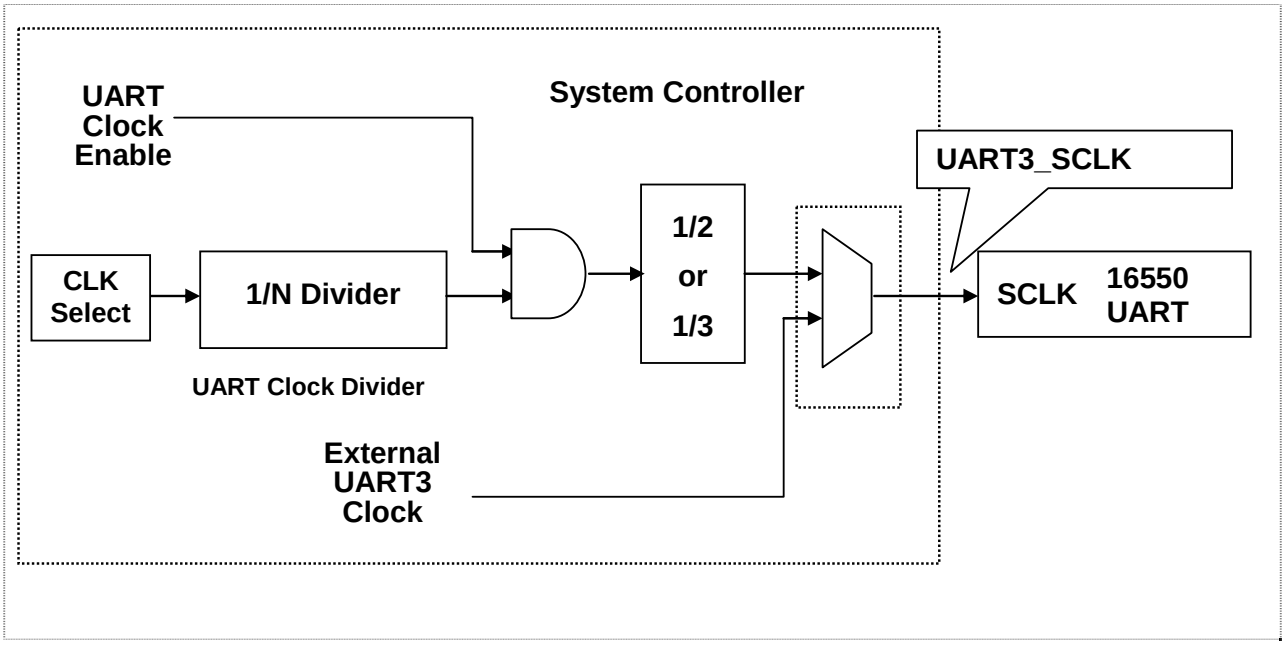
Bits [7:0]: **UART3CLKDIV [7:0]**

Generate a division rate 1/N. (N = Bits[7:0]+1)

0: 1/1
1: 1/2 division
•
•
255: 1/256 division

When you select internal clock, UART3 is supplied with UART3_SCLK which is a clock divided into 1/2 of the divided UART3CLKDIV.
Thus, UART3_SCLK = (PCLK's frequency) * 1/N * 1/2.
When you select external clock, input external clock is supplied to UART3 as it is.
(See the following chart.)

Note: This UART3_SCLK is very different from SCLK, the clock for SPI Interface.
To use UART3, be sure to write to this register once.



Timer B Clock Select Register (TIMBCKSEL)

SYS[0x34] Default value = 0x0000_0000 Read/Write

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							
7	6	5	4	3	2	1	0
N/a							CLKSEL

Bit 0: **CLKSEL**

This register selects the clock supplied to Timer-B.

0: 1/8 of PLL output

1: 32KHz

ADC Sampling Clock Divide Register (ADCCKDIV)

SYS[0x38] Default value = 0x0000_000A Read/Write

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							
7	6	5	4	3	2	1	0
ADCKDIVN3[3:0]				ADCKDIVN2		ADCKDIVN1	

Bits [7:4]: **ADCKDIVN3[3:0]**

This register sets the division rate of the clock supplied to ADC. One part of (ADCKDIVN3+1) clock that is divided by ADCKDIV1 and ADCKDIV2 is set to ADC as High and Low periods of sampling clock. Use the default setting in usual environments.

Division setting: (ADCKDIVN1 × ADCKDIVN2) × 1/((ADCKDIVN3+1)×2)

Bits [3:2]: **ADCKDIVN2[1:0]**

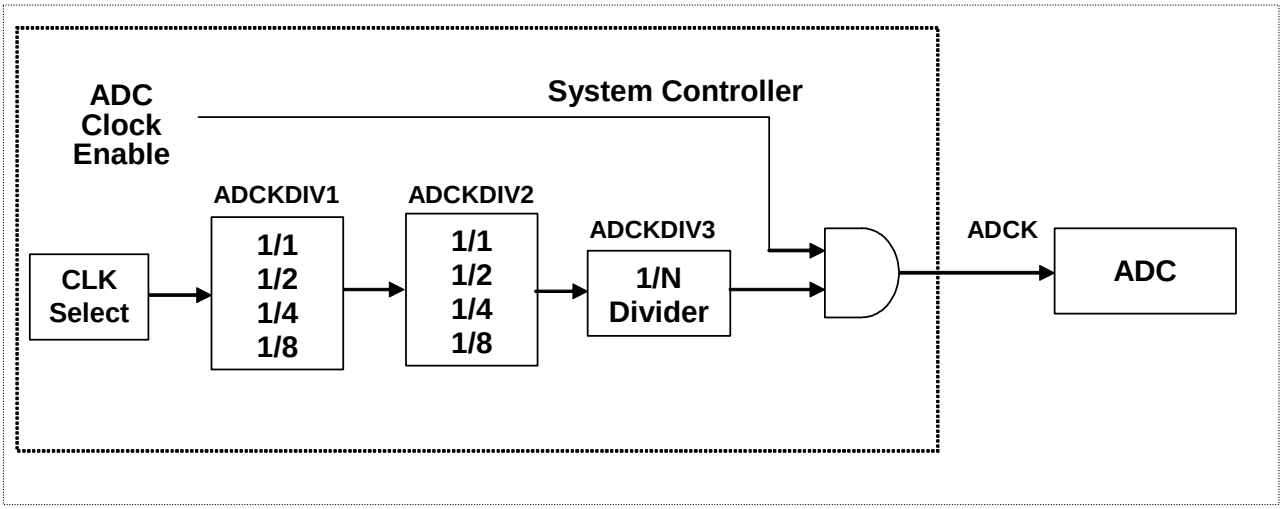
This register sets the division rate of the clock supplied to ADC. ADCKDIV2 divides the clock divided by ADCKDIV1, and the divided clock is used as a reference clock for ADCKDIV3. Use the default setting in usual environments.

00: 1/1
01: 1/2
10: 1/4 (default)
11: 1/8

Bits [1:0]: **ADCKDIVN1[1:0]**

This register sets the division rate of the clock supplied to ADC. The clock divided by ADCKDIV1 is used as a reference clock for ADCKDIV2. Use the default setting in usual environments.

00: 1/1
01: 1/2
10: 1/4 (default)
11: 1/8



MD Bus Resistor Control Register (MDPLDCTL)

SYS[0x40] Default value = 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 [15:0]: **MDPLDNDIS [15:0]**

Control the connection/separation of the pull-down resistor inside the MD [15:0] bus. After resetting, the resistor can be separated as required (according to presence or absence of external pull-up resistor particularly for a low-power application).

Each bit corresponds to each pin of MD [15:0].

0: Pull-down resistor enabled (this state is default after resetting)
1: Pull-down resistor disabled

SDD Bus Resistor Control Register (SDDPLDCTL)

SYS[0x44] Default value = 0x0000_0000 Read/Write

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
SDDPLDNDIS [15:8]							
7	6	5	4	3	2	1	0
SDDPLDNDIS [7:0]							

Bits [15:0]: **SDDPLDNDIS [15:0]**

Control the connection/separation of the pull-down resistors built in SDD[15:0] Bus. The separation is possible as required (in particular, if an external pull-up resistor is connected for a low-power application) after resetting.

Each bit corresponds to each pin of MD [15:0].

0: Pull-down resistor enabled (default after resetting)

1: Pull-down resistor disabled

GPIOE Resistor Control Register (PORTERCTL)

SYS[0x48] Default value = 0x0000_0000 Read/Write

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
Reserved							
7	6	5	4	3	2	1	0
PORTEPDDIS [7:0]							

Bits [7:0]: **PORTEPDDIS [7:0]**

This register controls the connection/separation of the pull-down resistors built in GPIOE [7:0] pins.

Each bit corresponds to each pin of GPIOE [7:0].

0: Pull-down resistor enabled (default after resetting)

1: Pull-down resistor disabled

GPIOF Resistor Control Register (PORTFRCTL)

SYS[0x4C] Default value = 0x0000_0000 Read/Write

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
Reserved							
7	6	5	4	3	2	1	0
PORTFPDDIS [7:0]							

Bits [7:4]: **PORTFPDDIS [7:0]**

Control the connection/separation of the pull-down resistors built in GPIOF [7:0] pins.

Each bit corresponds to each pin of GPIOF [7:0].

0: Pull-down resistor enabled (default after resetting)

1: Pull-down resistor disabled

GPIOG Resistor Control Register (PORTGRCTL)

SYS[0x50] Default value = 0x0000_0000 Read/Write

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
Reserved							
7	6	5	4	3	2	1	0
PORTGPDDIS [7:0]							

Bits [7:0]: **PORTGPDDIS [7:0]**

This register controls the connection/separation of the pull-down resistors built in GPIOG [7:0] pins. Each bit corresponds to each pin of GPIOG [7:0].

0: Pull-down resistor enabled (default after resetting)
1: Pull-down resistor disabled

GPIOH Resistor Control Register (PORTHRCTL)

SYS[0x54] Default value = 0x0000_0000 Read/Write

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
Reserved							
7	6	5	4	3	2	1	0
PORTHPUDIS [7:0]							

Bits [7:0]: **PORTHPUDIS [7:0]**

This register controls the connection/separation of the pull-up resistors built in GPIOH [7:0] pins. Each bit corresponds to each pin of GPIOH [7:0].

0: Pull-up resistor enabled (default after resetting)
1: Pull-up resistor disabled

GPIOI Resistor Control Register (PORTIRCTL)

SYS[0x58] Default value = 0x0000_0000 Read/Write

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
Reserved							
7	6	5	4	3	2	1	0
Reserved						PORTIPUDIS [1:0]	

Bits [7:0]: **PORTIPUDIS [1:0]**

This register controls the connection/separation of the pull-up resistors built in GPIOI [1:0] pins. Each bit corresponds to each pin of GPIOI [1:0].

0: Pull-up resistor enabled (default after resetting)
1: Pull-up resistor disabled

GPIOJ Resistor Control Register (PORTJRCTL)

SYS[0x5C] Default value = 0x0000_0000 Read/Write

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
Reserved							
7	6	5	4	3	2	1	0
PORTJPDDIS [7:0]							

Bits [7:0]: **PORTJPDDIS [7:0]**

This register controls the connection/separation of the pull-down resistors built in GPIOJ [7:0] pins. Each bit corresponds to each pin of GPIOJ [7:0].

0: Pull-down resistor enabled (default after resetting)
1: Pull-down resistor disabled

GPIOK Resistor Control Register (PORTKRCTL)

SYS[0x60] Default value = 0x0000_0000 Read/Write

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
Reserved							
7	6	5	4	3	2	1	0
PORTKPDDIS [7:0]							

Bits [7:0]: **PORTKPDDIS [7:0]**

This register controls the connection/separation of the pull-down resistors built in GPIOK [7:0] pins. Each bit corresponds to each pin of GPIOK [7:0].

0: Pull-down resistor enabled (default after resetting)
1: Pull-down resistor disabled

MISC Resister Control Register (MISCRCTL)

SYS[0x64] Default value = 0x0000_0000 Read/Write

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
Reserved							
7	6	5	4	3	2	1	0
Reserved							TRSTNDUDIS

Bit [0]: **TRSTNPDDIS**

This register controls the connection/separation of the pull-up resistor inside the TRST# pin.

0: Pull-down resistor enabled (this state is default after resetting.)
1: Pull-down resistor disabled

Embedded Memory Control Register (EMBMEMCTL)

SYS[0x68] Default value = 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		EMBRAMSEL[1:0]		Reserved		EMBWAITEN[1:0]	
7	6	5	4	3	2	1	0

Bits [31:6]: **Reserved (0)**

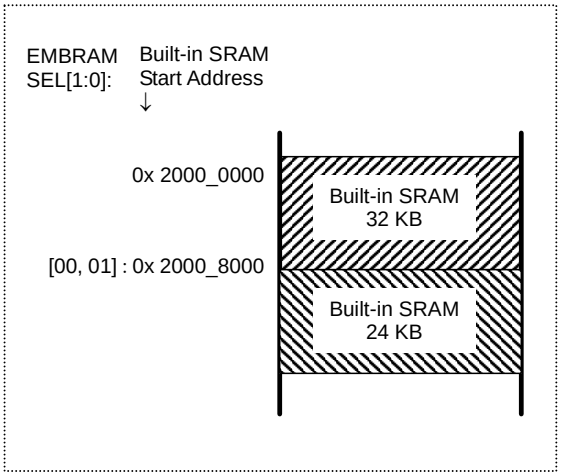
Bits [5:4]: **EMBRAMSEL[1:0] Embedded SRAM Select**

00: Built-in SRAM 24KB is allocated to the JPEG1 line buffer.
(Use JPEG Line Buffer [0xA4] of JPEG controller 1 by setting it to default "0x0020") 32KB is made available as the built-in SRAM with the start address "0x 2000_0000".

01: 4KB is allocated to the JPEG1 line buffer, and 32KB to the JPEG2 line buffer.
(Use JPEG Line Buffer [0xA4] of JPEG controller 1 by setting it to "0x0000", and that of JPEG controller 2 by setting it to the default "0x0020").

10: 56KB is allocated to the JPEG1 line buffer. (Use JPEG Line Buffer [0xA4] of JPEG controller 1 by setting it to "0x0000".)

11: The full size 56KB is made available for the built-in SRAM with the start address "0x 2000_0000".



Bits [3:2]: **Reserved (0)**

Bits [1:0]: **EMBWAITEN[1:0] Embedded SRAM Wait Control**

00: No Wait
01: Read Access Wait ON (Read: 1 wait, Write: no wait)
10: Read Access Wait ON, Read Data Wait ON,
(Read: 2 wait, Write: no wait)
11: Read Access Wait ON, Read Data Wait ON, Write Access Wait On
(Read: 2 wait, Write: 1 wait)

MISC Register (MISC)

SYS[0x6C] Default value = 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
Reserved							
7	6	5	4	3	2	1	0

Bits [31:0]: **Reserved (0)**

Internal TEST Mode Register (ITESTM)

SYS[0x70] Default value = 0x0000_0000 Read/Write

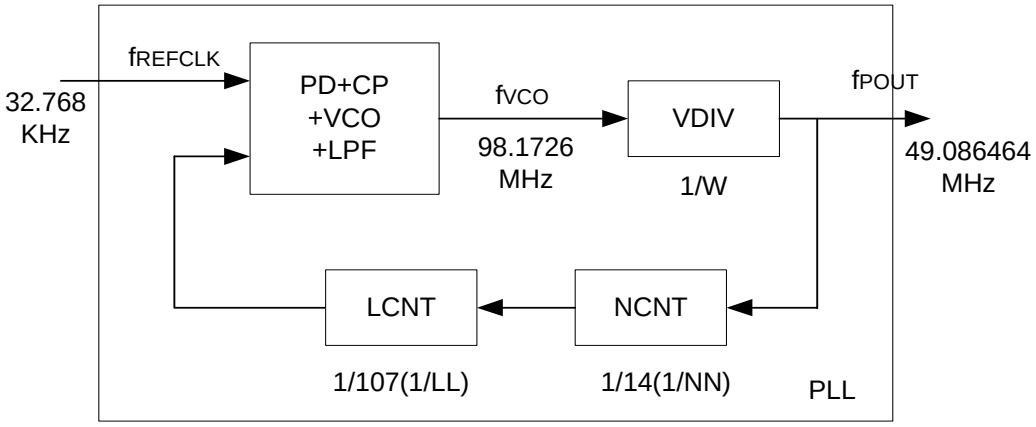
31	30	29	28	27	26	25	24
23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8
7	6	5	4	3	2	1	0

This register is used for the internal testing of the LSI. No user may operate it.
Leave it as it is reset.

Bits [31:0]: **Reserved (0)**

13.5 PLL Setting Example

Shown below are examples of the PLL settings.
If CPU Clock = 49.086464 MHz:



If f_{POUT}= 49.086464 MHz is selected:
f_{REFCLK} = 32.768KHz
f_{POUT} / f_{REFCLK} = 49.086464 MHz / 32.768KHz = 1498 (division rate)
Division rate: 1498 = 2 × 7 × 207 = 107 × 14
Thus, NN = 14 and LL = 107 can be selected as the division values.
NN = 14, or N-Counter [3:0] = 13 = 1101 (Binary)
LL = 107, or L-Counter [9:0] = 106 = 00_0110_1010 (Binary)

V_{CO} frequency: f_{VCO} = f_{POUT} × W = 49.086464 MHz × 2 = 98.172928 MHz
Here, W = 2, or about 100MHz;
thus W = 2 or W-Divider [1:0] = 01 (Binary) suffices.

13.6 PLL Parameter table

To make settings for PLL Setting Register1, use the values shown below corresponding to the frequency range used. PLL requires 100ms at the maximum for its output stabilizing time. Be sure to wait for this time to change 32KHz after turning on PLL, then switch it to desired clock by setting CLKSEL=1. The PLL's frequency can be changed only if the system is running at 32KHz and with PLLEN=0 (= PLL Disable). If it is running at an frequency other than 32KHz, the frequency may not changed to another directly. Switch it to 32KHz first, then to the frequency desired. Shown below is an example of the PLL Setting Register1 settings.

Target frequency (example)	Multiplying factor for 32.768KHz	PLL_Setting Register1 (Hex)	NN N-Counter	LL L-Counter
49.086464MHz	1498 = 14 x 107	0x0421_D46A	1101b	00_0110_1010b
49.971200MHz	1525 = 5 x 305	0x0423_4930	0100b	01_0011_0000b
50.003968MHz	1526 = 7 x 218	0x0423_68D9	0110b	00_1101_1001b

Use the values shown below for W-Divider, VC, RS, CP and CS if VCO frequency is 45 - 50MHz.

VV_Divider	VC[3:0]	RS[3:0]	CP[4:0]	CS[1:0]
01b	0001b	0010b	0_0100b	00b
10b	0011b	0010b	0_0100b	00b

13.7 Memory Map after Remapping

13.7.1 Memory Map after Remapping (AHB1)

The remap function allows changing the memory map after resetting.
The memory map after remapped may be easier to use if an OS running on a RAM basis is used, though this does not apply to the OS running on a ROM basis.

Table 13.2 Memory Map of AHB1 after Remapping

Start Address	End Address	Size	Device	External Chip Select	Device Bus size
0x0000_0000	0x0FFF_FFFF	256MByte	External SDRAM	SDCS0#	32
0x1000_0000	0x1FFF_FFFF	256MByte	Reserved		
0x2000_0000	0x2FFF_FFFF	128MByte	Built-in SRAM		32
0x3000_0000	0x3FFF_FFFF	256MByte	Reserved		
0x4000_0000	0x4FFF_FFFF	256MByte	External SDRAM	SDCS0#	32
0x5000_0000	0x5FFF_FFFF	256MByte	External SDRAM	SDCS1#	32
0x6000_0000	0x6FFF_FFFF	256Mbyte	Reserved		
0x7000_0000	0x7FFF_FFFF	256Mbyte	Reserved		
0x8000_0000	0x8FFF_FFFF	256Mbyte	SDMMC-IF		
0x9000_0000	0x9FFF_FFFF	256Mbyte	Reserved		
0xA000_0000	0xAFFF_FFFF	256Mbyte	Reserved		
0xB000_0000	0xBFFF_FFFF	256Mbyte	Reserved		
0xC000_0000	0xC1FF_FFFF	32MByte	External ROM	MCS0#	16
0xC200_0000	0xC3FF_FFFF	32MByte	Reserved		
0xC400_0000	0xC5FF_FFFF	32MByte	External SRAM	MCS1#	16
0xC600_0000	0xC7FF_FFFF	32MByte	Reserved		
0xC800_0000	0xC9FF_FFFF	32MByte	External SRAM	MCS2#	16
0xCA00_0000	0xCBFF_FFFF	32MByte	Reserved		
0xCC00_0000	0xCDFF_FFFF	32MByte	External SRAM	MCS3#	16
0xCE00_0000	0xCFFF_FFFF	32MByte	Reserved		
0xD000_0000	0xDFFF_FFFF	256Mbyte	Reserved		
0xE000_0000	0xEFFF_FFFF	256Mbyte	Reserved		
0xF000_0000	0xFFFF_FFFF	256Mbyte	Built-in I/O Area		32/16/8

Table 13.3 Memory Map of AHB1 before Remapping

Start Address	End Address	Size	Device	External Chip Select	Device Bus size
0x0000_0000	0x01FF_FFFF	32MByte	External ROM	MCS0#	16
0x0200_0000	0x03FF_FFFF	32MByte	Reserved		
0x0400_0000	0x05FF_FFFF	32MByte	External SRAM	MCS1#	16
0x0600_0000	0x07FF_FFFF	32MByte	Reserved		
0x0800_0000	0x09FF_FFFF	32MByte	External SRAM	MCS2#	16
0x0A00_0000	0x0BFF_FFFF	32MByte	Reserved		
0x0C00_0000	0x0DFF_FFFF	32MByte	External SRAM	MCS3#	16
0x0E00_0000	0x0FFF_FFFF	32MByte	Reserved		
0x1000_0000	0x1FFF_FFFF	256MByte	Reserved		
0x2000_0000	0x2FFF_FFFF	256Mbyte	Built-in SRAM		32
0x3000_0000	0x3FFF_FFFF	256MByte	Reserved		
0x4000_0000	0x4FFF_FFFF	256MByte	External SDRAM	SDCS0#	16/32
0x5000_0000	0x5FFF_FFFF	256MByte	External SDRAM	SDCS1#	16/32
0x6000_0000	0x6FFF_FFFF	256Mbyte	Reserved		
0x7000_0000	0x7FFF_FFFF	256Mbyte	Reserved		
0x8000_0000	0x8FFF_FFFF	256Mbyte	SDMMC-IF		
0x9000_0000	0x9FFF_FFFF	256Mbyte	Reserved		
0xA000_0000	0xAFFF_FFFF	256Mbyte	Reserved		
0xB000_0000	0xBFFF_FFFF	256Mbyte	Reserved		
0xC000_0000	0xC1FF_FFFF	32MByte	External ROM	MCS0#	16
0xC200_0000	0xC3FF_FFFF	32MByte	Reserved		
0xC400_0000	0xC5FF_FFFF	32MByte	External SRAM	MCS1#	16
0xC600_0000	0xC7FF_FFFF	32MByte	Reserved		
0xC800_0000	0xC9FF_FFFF	32MByte	External SRAM	MCS2#	16
0xCA00_0000	0xCBFF_FFFF	32MByte	Reserved		
0xCC00_0000	0xCDFF_FFFF	32MByte	External SRAM	MCS3#	16
0xCE00_0000	0xCFFF_FFFF	32MByte	Reserved		
0xD000_0000	0xDFFF_FFFF	256Mbyte	Reserved		
0xE000_0000	0xEFFF_FFFF	256Mbyte	Reserved		
0xF000_0000	0xFFFF_FFFF	256Mbyte	Built-in I/O Area		32/16/8

13.7.2 Memory Map after Remapping (AHB2)

Table 13.4 Memory Map after Remapping (AHB2)

Start Address	End Address	Size	Device	External Chip Select	Device Bus size
0x0000_0000	0x0FFF_FFFF	256MByte	External SDRAM	SDCS0#	32
0x1000_0000	0x1FFF_FFFF	256MByte	Reserved		
0x2000_0000	0x2FFF_FFFF	128MByte	Built-in SRAM		32
0x3000_0000	0x3FFF_FFFF	256MByte	Reserved		
0x4000_0000	0x4FFF_FFFF	256MByte	External SDRAM	SDCS0#	32
0x5000_0000	0x5FFF_FFFF	256MByte	External SDRAM	SDCS1#	32
0x6000_0000	0x6FFF_FFFF	256Mbyte	Reserved		
0x7000_0000	0x7FFF_FFFF	256Mbyte	Reserved		
0x8000_0000	0x8FFF_FFFF	256Mbyte	SDMMC-IF		
0x9000_0000	0x9FFF_FFFF	256Mbyte	Reserved		
0xA000_0000	0xAFFF_FFFF	256Mbyte	Reserved		
0xB000_0000	0xBFFF_FFFF	256Mbyte	Reserved		
0xC000_0000	0xC1FF_FFFF	32MByte	External ROM	MCS0#	16
0xC200_0000	0xC3FF_FFFF	32MByte	Reserved		
0xC400_0000	0xC5FF_FFFF	32MByte	External SRAM	MCS1#	16
0xC600_0000	0xC7FF_FFFF	32MByte	Reserved		
0xC800_0000	0xC9FF_FFFF	32MByte	External SRAM	MCS2#	16
0xCA00_0000	0xCBFF_FFFF	32MByte	Reserved		
0xCC00_0000	0xCDFF_FFFF	32MByte	External SRAM	MCS3#	16
0xCE00_0000	0xCFFF_FFFF	32MByte	Reserved		
0xD000_0000	0xDFFF_FFFF	256Mbyte	JPEG2 DMA Port		
0xE000_0000	0xEFFF_FFFF	256Mbyte	JPEG1 DMA Port		32
0xF000_0000	0xFFFF_FFFF	256Mbyte	Reserved		

Table 13.5 Memory Map of AHB2 before Remapping

Start Address	End Address	Size	Device	External Chip Select	Device Bus size
0x0000_0000	0x01FF_FFFF	32MByte	External ROM	MCS0#	16
0x0200_0000	0x03FF_FFFF	32MByte	Reserved		
0x0400_0000	0x05FF_FFFF	32MByte	External SRAM	MCS1#	16
0x0600_0000	0x07FF_FFFF	32MByte	Reserved		
0x0800_0000	0x09FF_FFFF	32MByte	External SRAM	MCS2#	16
0x0A00_0000	0x0BFF_FFFF	32MByte	Reserved		
0x0C00_0000	0x0DFF_FFFF	32MByte	External SRAM	MCS3#	16
0x0E00_0000	0x0FFF_FFFF	32MByte	Reserved		
0x1000_0000	0x1FFF_FFFF	256MByte	Reserved		
0x2000_0000	0x2FFF_FFFF	256Mbyte	Built-in SRAM		32
0x3000_0000	0x3FFF_FFFF	256MByte	Reserved		
0x4000_0000	0x4FFF_FFFF	256MByte	External SDRAM	SDCS0#	16/32
0x5000_0000	0x5FFF_FFFF	256MByte	External SDRAM	SDCS1#	16/32
0x6000_0000	0x6FFF_FFFF	256Mbyte	Reserved		
0x7000_0000	0x7FFF_FFFF	256Mbyte	Reserved		
0x8000_0000	0x8FFF_FFFF	256Mbyte	SDMMC-IF		
0x9000_0000	0x9FFF_FFFF	256Mbyte	Reserved		
0xA000_0000	0xAFFF_FFFF	256Mbyte	Reserved		
0xB000_0000	0xBFFF_FFFF	256Mbyte	Reserved		
0xC000_0000	0xC1FF_FFFF	32MByte	External ROM	MCS0#	16
0xC200_0000	0xC3FF_FFFF	32MByte	Reserved		
0xC400_0000	0xC5FF_FFFF	32MByte	External SRAM	MCS1#	16
0xC600_0000	0xC7FF_FFFF	32MByte	Reserved		
0xC800_0000	0xC9FF_FFFF	32MByte	External SRAM	MCS2#	16
0xCA00_0000	0xCBFF_FFFF	32MByte	Reserved		
0xCC00_0000	0xCDFF_FFFF	32MByte	External SRAM	MCS3#	16
0xCE00_0000	0xCFFF_FFFF	32MByte	Reserved		
0xD000_0000	0xDFFF_FFFF	256Mbyte	JPEG2 DMA Port		
0xE000_0000	0xEFFF_FFFF	256Mbyte	JPEG1 DMA Port		32
0xF000_0000	0xFFFF_FFFF	256Mbyte	Reserved		

13.7.3 Memory Map after Remapping (AHB3)

Table 13.6 Memory Map after Remapping (AHB3)

Start Address	End Address	Size	Device	External Chip Select	Device Bus size
0x0000_0000	0x0FFF_FFFF	256MByte	External SDRAM	SDCS0#	32
0x1000_0000	0x1FFF_FFFF	256MByte	Reserved		
0x2000_0000	0x2FFF_FFFF	128MByte	Built-in SRAM		32
0x3000_0000	0x3FFF_FFFF	256MByte	Reserved		
0x4000_0000	0x4FFF_FFFF	256MByte	External SDRAM	SDCS0#	32
0x5000_0000	0x5FFF_FFFF	256MByte	External SDRAM	SDCS1#	32
0x6000_0000	0x6FFF_FFFF	256Mbyte	Reserved		
0x7000_0000	0x7FFF_FFFF	256Mbyte	Reserved		
0x8000_0000	0x8FFF_FFFF	256Mbyte	SDMMC-IF		
0x9000_0000	0x9FFF_FFFF	256Mbyte	Reserved		
0xA000_0000	0xAFFF_FFFF	256Mbyte	Reserved		
0xB000_0000	0xBFFF_FFFF	256Mbyte	Reserved		
0xC000_0000	0xC1FF_FFFF	32MByte	External ROM	MCS0#	16
0xC200_0000	0xC3FF_FFFF	32MByte	Reserved		
0xC400_0000	0xC5FF_FFFF	32MByte	External SRAM	MCS1#	16
0xC600_0000	0xC7FF_FFFF	32MByte	Reserved		
0xC800_0000	0xC9FF_FFFF	32MByte	External SRAM	MCS2#	16
0xCA00_0000	0xCBFF_FFFF	32MByte	Reserved		
0xCC00_0000	0xCDFF_FFFF	32MByte	External SRAM	MCS3#	16
0xCE00_0000	0xCFFF_FFFF	32MByte	Reserved		
0xD000_0000	0xDFFF_FFFF	256Mbyte	Reserved		
0xE000_0000	0xEFFF_FFFF	256Mbyte	Reserved		32
0xF000_0000	0xFFFF_FFFF	256Mbyte	USB DMA Port		

Table 13.7 Memory Map of AHB3 before Remapping

Start Address	End Address	Size	Device	External Chip Select	Device Bus size
0x0000_0000	0x01FF_FFFF	32MByte	External ROM	MCS0#	16
0x0200_0000	0x03FF_FFFF	32MByte	Reserved		
0x0400_0000	0x05FF_FFFF	32MByte	External SRAM	MCS1#	16
0x0600_0000	0x07FF_FFFF	32MByte	Reserved		
0x0800_0000	0x09FF_FFFF	32MByte	External SRAM	MCS2#	16
0x0A00_0000	0x0BFF_FFFF	32MByte	Reserved		
0x0C00_0000	0x0DFF_FFFF	32MByte	External SRAM	MCS3#	16
0x0E00_0000	0x0FFF_FFFF	32MByte	Reserved		
0x1000_0000	0x1FFF_FFFF	256MByte	Reserved		
0x2000_0000	0x2FFF_FFFF	256Mbyte	Built-in SRAM		32
0x3000_0000	0x3FFF_FFFF	256MByte	Reserved		
0x4000_0000	0x4FFF_FFFF	256MByte	External SDRAM	SDCS0#	16/32
0x5000_0000	0x5FFF_FFFF	256MByte	External SDRAM	SDCS1#	16/32
0x6000_0000	0x6FFF_FFFF	256Mbyte	Reserved		
0x7000_0000	0x7FFF_FFFF	256Mbyte	Reserved		
0x8000_0000	0x8FFF_FFFF	256Mbyte	SDMMC-IF		
0x9000_0000	0x9FFF_FFFF	256Mbyte	Reserved		
0xA000_0000	0xAFFF_FFFF	256Mbyte	Reserved		
0xB000_0000	0xBFFF_FFFF	256Mbyte	Reserved		
0xC000_0000	0xC1FF_FFFF	32MByte	External ROM	MCS0#	16
0xC200_0000	0xC3FF_FFFF	32MByte	Reserved		
0xC400_0000	0xC5FF_FFFF	32MByte	External SRAM	MCS1#	16
0xC600_0000	0xC7FF_FFFF	32MByte	Reserved		
0xC800_0000	0xC9FF_FFFF	32MByte	External SRAM	MCS2#	16
0xCA00_0000	0xCBFF_FFFF	32MByte	Reserved		
0xCC00_0000	0xCDFF_FFFF	32MByte	External SRAM	MCS3#	16
0xCE00_0000	0xCFFF_FFFF	32MByte	Reserved		
0xD000_0000	0xDFFF_FFFF	256Mbyte	Reserved		
0xE000_0000	0xEFFF_FFFF	256Mbyte	Reserved		32
0xF000_0000	0xFFFF_FFFF	256Mbyte	USB DMA Port		

13.8 Clock Control Block Diagram

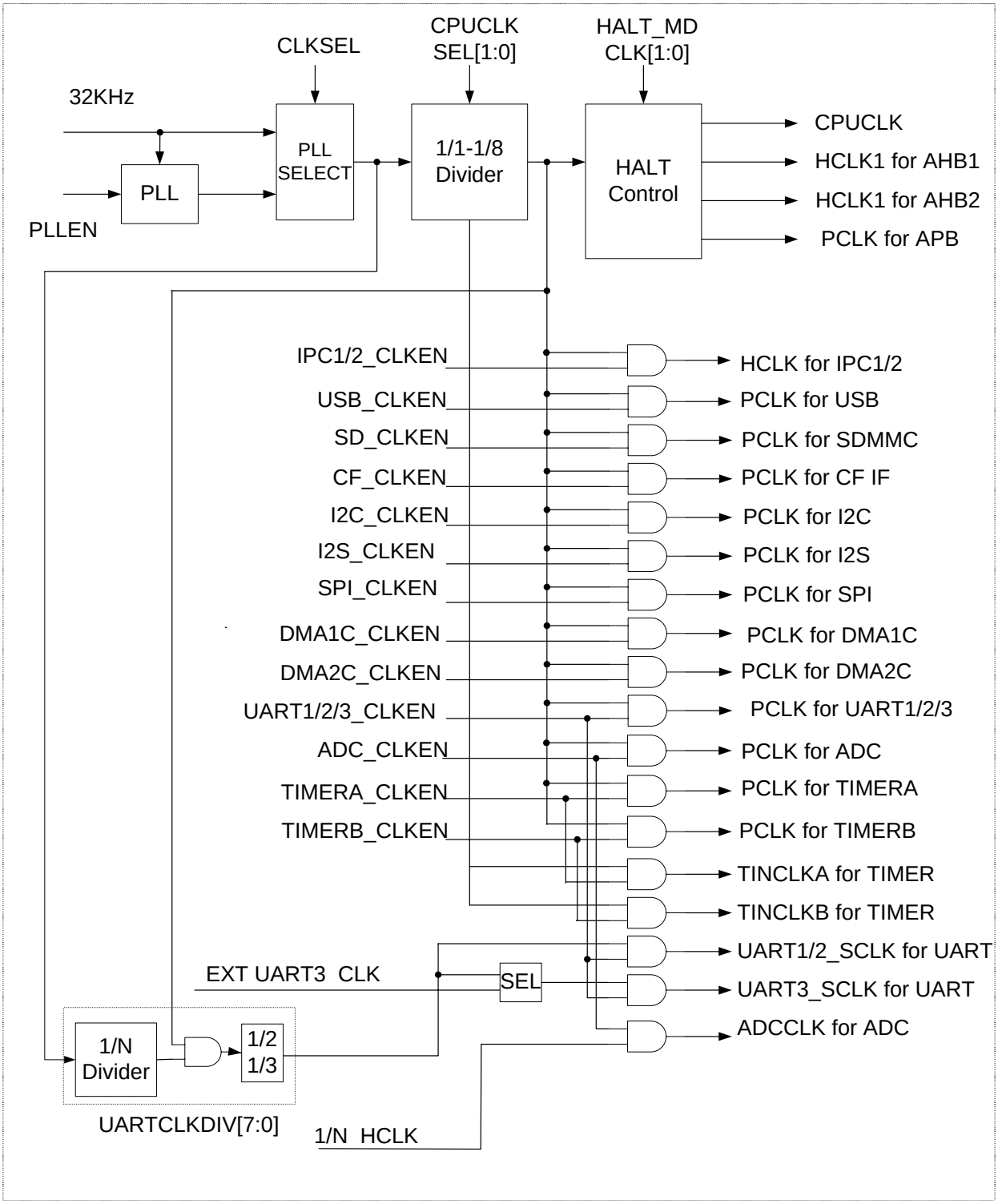


Figure 13.2 Clock Control Block Diagram

A clock signal name “PCLK_for_block-name” in the diagram indicates the clock used for controlling the buses and registers in each block. Also, UART1/2/3_SCLK is used for the clock input for generating the baud-rate at UART1/2/3, respectively; TINCLKA/B is used for the basic clock input for counting in the TimerA/B block, respectively.

13.9 Example of Settings for UART Clock

If CPU Clock = 49.086464 MHz:

Baudrate	Ideal x16 Clock (Hz)	SYS[0x28]	16550 Divisor Value (DEC)	16550 Divisor Value (HEX)	Percent Error %	Actual x16 Clock (Hz)
110	1760	0	13945	3679	0.00	1760.0
300	4800	0	5113	13F9	0.00	4800.2
600	9600	0	2557	09FD	0.02	9598.4
1200	19200	0	1278	04FE	0.02	19204.4
2400	38400	0	639	027F	0.02	38408.8
4800	76800	0	320	0140	0.13	76697.6
9600	153600	0	160	00A0	0.13	153395.2
14400	230400	0	107	006B	0.44	229376.0
19200	307200	0	80	0050	0.13	306790.4
38400	614400	0	40	0028	0.13	613580.8
57600	921600	0	27	001B	1.37	909008.6
115200	1843200	0	9*	0009	1.37	1818017.2

If CPU Clock = 50.003968 MHz:

Baudrate	Ideal x16 Clock (Hz)	SYS[0x28]	16550 Divisor Value (DEC)	16550 Divisor Value (HEX)	Percent Error %	Actual x16 Clock (Hz)
110	1760	0	14206	377E	0.00	1760.0
300	4800	0	5209	1459	0.00	4699.8
600	9600	0	2604	0A2C	0.01	9601.4
1200	19200	0	1302	0516	0.01	19202.8
2400	38400	0	651	028B	0.01	38405.5
4800	76800	0	326	0146	0.14	76693.2
9600	153600	0	163	00A3	0.14	153386.4
14400	230400	0	109	006D	0.44	2293760
19200	307200	0	81	0051	0.48	308666.5
38400	614400	0	41	0029	0.75	609804.5
57600	921600	0	27	001B	0.48	925999.4
115200	1843200	0	9*	0009	0.48	1851998.8

*Set the DIVMODE bit of the UART1/2/3 Clock Divider Register (SYS[0x28/2C/30]) to “1”, and set division rate for the last step of the frequency divider to 1/3.

2% of errors need to be included to the calculation of the Error rate of the UART baud-rate as Period Jitter of the PLL clock.

14. MEMORY CONTROLLER (MEMC)

14.1 Description

The memory controller, built in the AHB bus interface, controls the asynchronous SRAM- and SDRAM-type devices. It supports up to four asynchronous SRAM-type and up to two SDRAM-type devices.
Shown below are the features of the memory controller.

- Compatible with the SRAM timing devices.
- Compatible with the SDRAM devices.
- Allows adjustment of the SDRAM auto-refresh intervals in accordance with devices. If the memory controller is unavailable because it is engaged in memory access or otherwise, it can accumulate plural waiting refreshing operations, performing the burst refreshing as soon as it becomes available. It performs refreshing on the background.
- Supports the SDRAM's self refreshing. Also it can read/write the SDRAM in the self-refreshing state. In this case, it allows selection whether to keep the SDRAM in idle state or to get the SDRAM enter in the self-refresh mode again.

14.1.1 SRAM Controller

- Supports the following devices.
  Asynchronous SRAM, ROM, FLASH, EEPROM
- External 8/16-bit bus
- Supports burst transfer.
- Wait state is programmable.
- Timing of inserting MWE# and MOE# is programmable.
- Connectable to a x16-type static memory with common MWE# and independent byte-enable without any external logic.

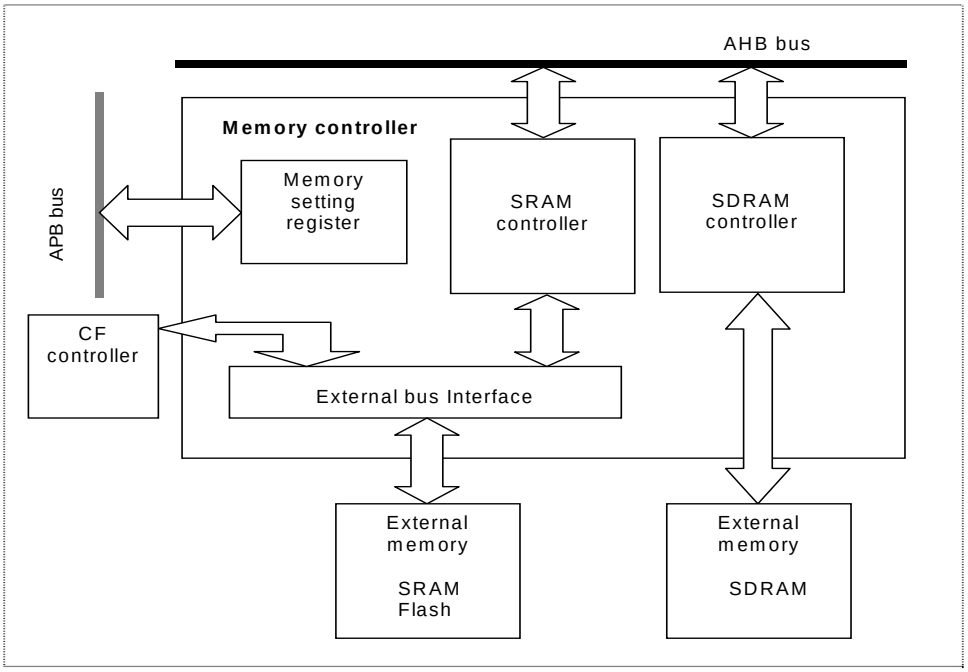
14.1.2 SDRAM Controller

- Supports the following devices.
  SDRAM
- 16/32-bit bus
- Supports burst transfer.
- Auto-precharge ON/OFF is programmable.
- Supports the mode of auto-setting of IN/OUT from/to a device in the self-refresh mode.
- Supports the manual SDRAM initialization mode.

14.1.3 External Bus Interface Module

- This module is required to use the SRAM controller and CF controller modules with common address/data buses. It arbitrates the requirements for the external bus from the SRAM controller and CF controller modules to access external memories. The requirements for the bus are met by handshaking.

14.2 Block Diagram



14.3 External Pins

The IC has the external pins for the memory controller as follows.

Table 14.1 List of external pins for memory controller

Pin name	I/O	Pin functions	Multiplex pin/Remarks
MA[23:20]	O	SRAM address output signal [23:20]	GPIOD[3:0]*
MA [19:12]	O	SRAM address output signal [19:12]	
MA11	O	SRAM address output signal 11	CFREG#**
MA [10:0]	O	SRAM address output signal [10:0]	CFADDR [10:0]**
MD [15:0]	I/O	SRAM data I/O signal [15:0]	MODESEL[15:0]***
MCS [3:2]#	O	Chip select signal for SRAM [3:2]	GPIOD[5:4]*
MCS [1:0]#	O	Chip select signal for SRAM [1:0]	
MOE#	O	Strobe signal for SRAM output	CFOE#**
MWE#	O	Write-enable signal for SRAM	CFWE#**
MBEH#	O	Upper-bites-enable signal for SRAM	
MBEL#	O	Lower-bites-enable signal for SRAM	
MWAIT	I	Wait signal for memory controller	The pin is shared with CFWAIT#.

Note (*):These external pins for the memory controller is multiplexed with GPIO, etc. They become available by setting “Non-GPIO function 1” in the register of the GPIO pin functions.

Note (**):The pin for the memory controller operates as the external pin for the compact flash interface (CF) when CR is in operation.

Note (***):Operates as the MODESEL pin to decide the internal operation mode at the time of power-on reset.

Table 14.2 List of external pins for memory controller

Pin name	I/O	Pin functions	Multiplex pin/Remarks
SDA [14:13]	O	SDRAM bank address output signal [1:0]	
SDA [12:0]	O	SDRAM address output signal [12:0]	
SDD [31:16]	I/O	SDRAM data I/O signal [31:16]	GPIOJ[7:0]*,GPIOK[7:0]*
SDD [15:0]	I/O	SDRAM data I/O signal [15:0]	
SDCS[1:0]#	O	Chip select signal for SDRAM [1:0]	
SDWE#	O	Write-enable signal for SDRAM	
SDCLK	O	Clock output signal for SDRAM	
SDCLKEN	O	Clock-enable output signal for SDRAM	
SDRAS#	O	RAS signal for SDRAM	
SDCAS#	O	CAS signal for SDRAM	
SDDQM3#	I/O	DQM signal for highest-order byte (SDRAM)	GPIOD7*
SDDQM2#	I/O	DQM signal for upper-middle-order byte (SDRAM)	GPIOD6*
SDDQM1#	O	DQM signal for lower-middle-order byte (SDRAM)	
SDDQM0#	O	DQM signal for lowest-order byte (SDRAM)	

Note (*):These external pins for the memory controller is multiplexed with GPIO, etc. They become available by setting “Non-GPIO function 1” in the register of the GPIO pin functions.

14.4 Memory controller

14.4.1 Number of Devices

Up to four SRAM timing devices (device 0 to 3) can be connected to the controller.
Up to eight (512MB at the maximum) SDRAM devices with 8-bit data width can be connected by using two chip-select signal lines.

14.4.2 External Memory Width

The bus width of external memory is switchable between 16 bits for the SRAM timing devices and 16 or 32 bits for the SDRAM devices by making a corresponding setting.

14.4.3 Setting of Device Segments

The SRAM interface has up to 32MB of space allocated for each chip-select signal. 64MB (16MBx4) is available for the SRAM timing space. Whereas the SDRAM interface has up to 256MB of space allocated for each chip-select signal, having up to 512MB assigned to the signals (256MB to each).

14.5 SRAM control

14.5.1 Setting of Timing

The timing is set for each device controller by multiplying a factor to the clock cycle supplied by the memory controller. Set the most appropriate timing values matching the clock cycle.
Adjustable timings include those of reading, writing, and inserting the MOE# or MWE# signal.

14.5.2 Write Protect

Setting “1” to the WPROTECT bit of a control register disables writing in the relevant device. Attempting to write in the prohibited area aborts bus access and causes an error notification to CPU; treat such conditions by software properly.

14.6 SDRAM control

14.6.1 Setting of Mode Register

Set first the SDRAM mode register, then issue the LMR instruction to devices with the initialization control register. Use INIT_SD(MEMC [0x80] bit15) normally. Select at the same time the target devices the instruction is issued for. Set the timing for each device controller with number of clocks. Set the most appropriate timing values matching the clock cycle.

14.6.2 Compatibility with Burst

The SDRAM is compatible with burst lengths 1, 2, 4, and 8. Also, the addresses are limited to the increment burst. The SDRAM incurs no access penalty even if a burst length is surpassed in the same ROW. If an interrupt is caused in R/W of a device by an requirement for access to a different device, the SDRAM controller issues the pre-charge or burst-terminate command automatically to pass over the bus promptly.

14.6.3 Setting of Auto-Precharge

Set whether to issue pre-charge after completion of the R/W operation. Selecting “No precharge” keeps the ROW system to be selected, accelerating the speed of R/W of the data in the same page. An active ROW address is precharged at the time of auto-refreshing, then goes idle. Also manual precharge operation is available. See SDRAM Setting Register (MEMC [0x70]).

14.6.4 Reduction of Power Consumption

If a device is in the state left idle after data access (including self-refresh and ROW active states), it can have MCLKEN de-asserted. This is made effective by “1” set to the CKECTRL bit of the SDRAM setting register (MEMC [0x70]).
If the CKECTRL bit has “1” set, it is possible to stop the clock itself supplied to the memory for the purpose of further reduction of power consumption. This is made effective by “1” set to the CLKCTRL bit.
If the accumulated number of auto-refreshing requirements surpasses the value set, or R/W is requested of a device, CKE is asserted as “HIGH” and data transfer is resumed even if the CKECTRL bit has “1”.

14.6.5 Stopping Memory Clock

Stopping the supply of clocks to SDRAM can lead to further reduction of power consumption other than keeping MCLKEN at “LOW” with CKECTRL. This mode is made effective by setting the CLKCTRL bit to “1”. Be sure to make the CKECTRL bit effective for making the CLKCTRL bit effective. The memory controller stops the memory clock MCLK if all the SDRAMs connected are in one of the following states:
memory not initialized yet, or chip idle in the self-refresh mode or with ROW inactive.
Any request for R/W or auto-refreshing of memory causes supply of clocks to be resumed even if in the state where the memory clock is stopped.
As an exceptional case, setting “1” to the CLKFORCE bit in the SDRAM detailed settings register (MEMC [0x74]) causes the output from MCLK to be continued regardless of the state of the SDRAM memory.

14.6.6 Support of Energy-Saving Mode

When the system enters the energy-saving mode, the supply of clocks to the memory controller itself may be stopped. An SDRAM requires to be auto-refreshed or self-refreshed to hold the data in it. Stop of supply of clocks to the memory controller necessitates that all the SDRAMs be entered into the self-refresh mode.
First enter all the SDRAMs into the self-refresh mode with the software, then set the system controller to the energy-saving mode.

14.6.7 Control of Auto-Refresh

Auto-refreshing is performed when at least one SDRAM has been initialized and is in the IDLE state (but not self-refreshing). Its cycle is equal to the number of HCLK set in the SDRAM refresh timer register. The memory controller also refreshes the SDRAM connected at the same time; so set the refreshing cycle in accordance with the device which requires the most frequent refreshing.
The refreshing is performed in a distributed way basically; but plural requirements for refreshing can be accumulated to prevent data transfer from being severed by auto-refreshing. Specify the number of requirements accumulable by setting AREFWAIT bits [3:0] in the SDRAM detailed setting register (MEMC [0x74]). If the refreshing requirements accumulated surpass the specified count, The SDRAM controller recognizes the requirements, starting auto-refreshing next time the memory controller goes idle.
If the device is accessing a bus at the timing its refreshing is requested, the access to the bus takes priority, and the auto-refreshing is performed upon clearance of the bus request.

14.6.8 Control of Self-Refresh

The memory controller supports the SDRAM's self-refreshing. To enter a device into the self-refresh mode, set "1" to the SELF bit for the relevant device in the SDRAM detailed setting register (MEMC [0x74]). To exit from the self-refresh mode, set "0" instead.
If a SDRAM in the self-refresh mode is accessed (for Read/Write), it exit from the mode automatically to execute the command required. After that, the device goes idle basically; it can be entered again into the self-refresh mode, though, by making an appropriate setting. To do so, set "1" to the RESELF bit in the SDRAM detailed setting register (MEMC [0x74]). The SDRAM enters the self-refresh mode automatically if the count of accesses to it as set in the SREFCNT bits is exhausted.

14.6.9 Status Register

Shows the statuses of the SDRAM controller and the devices connected.

14.7 Registers

14.7.1 List of Registers

Shown below is the list of the memory controller registers. The base address of these registers is 0xFFFF_A000.

Table 14.3 List of registers (Base address: 0xFFFF_A000)

Address offset	Register name	Register name's abbreviation	Default Value	R/W	Data access size
SRAM controller registers					
0x00	SRAM device 0 bus width setting register	RAMWIDTH0	0x0000_0040	R/W	32
0x04	SRAM device 1 bus width setting register	RAMWIDTH 1	0x0000_0040	R/W	32
0x08	SRAM device 2 bus width setting register	RAMWIDTH 2	0x0000_0040	R/W	32
0x0C	SRAM device 3 bus width setting register	RAMWIDTH 3	0x0000_0040	R/W	32
0x20	SRAM device 0 timing register	RAMTMG0	0x0000_1C7C	R/W	32
0x24	SRAM device 0 control register	RAMCNTL0	0x0000_0001	R/W	32
0x30	SRAM device 1 timing register	RAMTMG1	0x0000_1C7C	R/W	32
0x34	SRAM device 1 control register	RAMCNTL1	0x0000_0001	R/W	32
0x40	SRAM device 2 timing register	RAMTMG2	0x0000_1C7C	R/W	32
0x44	SRAM device 2 control register	RAMCNTL2	0x0000_0001	R/W	32
0x50	SRAM device 3 timing register	RAMTMG3	0x0000_1C7C	R/W	32
0x54	SRAM device 3 control register	RAMCNTL3	0x0000_0001	R/W	32
SDRAM controller registers					
0x60	SDRAM mode register	SDMR	0x0000_0032	R/W	16/32
0x64	Reserved	—	—	—/—	—
0x68	Reserved	—	—	—/—	—
0x70	SDRAM setting register	SDCNFG	0x0600_C700	R/W	32
0x74	SDRAM detailed setting register	SDADVCNFG	0x000F_0300	R/W	32
0x80	Initialization control register	SDINIT	0x0000_0000	R/W	16/32
0x90	SDRAM refresh timer register	SDREF	0x0000_00A0	R/W	16/32
0xA0	SDRAM status register	SDSTAT	0x0000_0202	RO	32

14.7.2 Detailed Description of Registers

SRAM device [3:0] bus width setting register (RAMWIDTH[3:0])

MEMC[0x00, 0x04, 0x08, 0x0C] Default value = 0x0000_0040 Read/Write

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

Bits [30:8]: **RSV Reserved**

These bits are reserved. Be sure to set "0".

Bits [7:6]: **WIDTH [1:0] SRAM bus width setting**

00: 8-bit

01: 16 bits (default)

1x: Reserved

Bits [5:0]: **RSV Reserved**

These bits are reserved. Be sure to set "0".

SRAM device [3:0] Timing register (RAMTMG[3:0])

MEMC[0x20, 0x30, 0x40, 0x50] Default value = 0x0000_1C7C Read/Write

RSV (0)	WAITWE [4:0]					RSV (0)	WAITOE [4:0]					RSV (0)		WAITWH[1:0]	
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)		WAITRH[1:0]	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit 31: **RSV Reserved**
This bit is reserved. Be sure to set “0”.

Bits [30:26]: **WAITWE [4:0] Write-enable signal insertion delay control**
Set the timing to insert the MWE# signal. (Default = 0x0)

Bit 25: **RSV Reserved**
This bit is reserved. Be sure to set “0”.

Bits [24:20]: **WAITOE [4:0] Output-enable signal insertion delay control**
Set the timing to insert the MOE# signal. (Default = 0x0)

Bits [19:18]: **RSV Reserved**
This bit is reserved. Be sure to set “0”.

Bits [17:16]: **WAITWH[1:0]**
MWE# signal hold cycle setting. (Default = 0x0)
Set the number of cycles between the rise of a MWE# signal and the rise of MCS#[3:0].
(The hold cycle of a WE# signal is the set value of this register plus 1 cycle.)

Bit 15: **RSV Reserved**
This bit is reserved. Be sure to set “0”.

Bits [14:10]: **WAITWR [4:0] Write cycle wait control**
Writing wait cycle setting. Set the number of wait cycles for writing to a SRAM or ROM device. (Default = 0x07)

Bit 9: **RSV Reserved**
This bit is reserved. Be sure to set “0”.

Bits [8:4]: **WAITRD [4:0] Read cycle wait control**
Set the number of reading wait cycle for to an SRAM/ROM device. (Default = 0x07)

Bits [3:2]: **RSV Reserved**
This bit is reserved. Be sure to set “0”.

Bits [1:0]: **WAITRH[1:0]**
MOE# signal hold cycle setting. (Default = 0x0)
Set the number of cycles between the rise of a MOE# signal and the rise of MCS#[3:0].

SRAM device [3:0] Control register (RAMCNTL[3:0])

MEMC[0x24, 0x34, 0x44, 0x54] Default value = 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

Bits [31:4]: **RSV Reserved**

This bit is reserved. Be sure to set “0”.

Bit 3: **WPROTECT Write protect**

No write can be performed in a device having the write-protect bit set.

0: Not write-protected (default)

1: Write-protected

Bits [2:1]: **MWAITPOL [1:0]**

Set the polarity for the MWAIT signal.

00: Disabled (default)

01: Enabled if LOW-active

10: Enabled if HIGH-active

11: Reserved

Bit 0: **RBLE Byte lane control setting**

To implement the byte control on writing with a device lacking in byte-lane control, set this bit to “0” and connect DQM of this device to WE# of the target device.

0: Reading makes DQM [1:0] HIGH, preventing writing.

1: Reading makes DQM [1:0] LOW, allowing reading of all the byte lanes. (Default)

SDRAM mode register (SDMR)

MEMC[0x60] Default value = 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 [31:10]: **RSV Reserved**

This bit is reserved. Be sure to set “0”.

Bit 9: **WBM Write burst mode**

0: Written with the burst length specified (default)

1: Single location access

Bits [8:7]: **OP Mode [1:0] Operation mode**

00: Normal operation

xx: All the other combinations of bits are reserved.

Bits [6:4]: **CL [2:0] CAS latency**

000: Reserved

001: CL=1

010: CL=2

011: CL=3 (default)

1xx: Reserved

Bit 3: **BT Burst type**

0: Sequential (default)

1: Reserved

Bits [2:0]: **BL [2:0] Burst length**

000: BL=1

001: BL=2

010: BL=4 (default)

011: BL=8

100: Reserved

101: Reserved

110: Reserved

111: Reserved

Reserved register

MEMC[0x64, 0x68] Default value = 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

Reserved register; don’t access it.

SDRAM setting register (SDCNFG)

MEMC[0x70] Default value = 0x0600_C700 Read/Write

RSV (0)					CLK-CTRL	CKE-CTRL	XBW	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	RSV(0)		RSV (0)		BNUM [1:0]		RSV (0)	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [31:27]: **RSV Reserved**

This bit is reserved. Be sure to set "0".

Bit 26: **CLKCTRL MCLK control**

This bit can be set to "1" only if the dynamic MCLKEN control (CKECTRL) is enabled.

0: MCLK is output continuously. Initialization of SDRAM is required, though.

1: MCLK is stopped if SDRAM is idle (including the self-refreshing state). (Default)

Note: To change the value of this bit (CLKCTRL) after initializing SDRAM (MEMC [0x80] bit15:INIT_SD = 1), first enter the SDRAM into the self-refresh mode. That is, perform: self-refresh mode IN, change of CLKCTRL, then self-refresh mode OUT.

Bit 25: **CKECTRL Dynamic MCLKEN control**

0: MCLKEN=H is output continuously (except in the self-refreshing state).

1: MCLK is put into LOW if SDRAM is idle (including the self-refreshing state and the bank active state getting no access). (Default)

Note: To change the value of this bit (CKECTRL) after initializing SDRAM (MEMC [0x80] bit15:INIT_SD = 1), first enter the SDRAM into the self-refresh mode. That is, perform: self-refresh mode IN, change of CKECTRL, then self-refresh mode OUT.

Bit 24: **XBW External bus width**

Select the external bus width.

0: External bus width =16 bits (default)

1: External bus width =32 bits

Bits [23:20]: **COLW [7:4] Column address width**

Set the column address width for SDRAM.

COLW [5:4]: For setting for SDRAM device 0.

COLW [7:6]: For setting for SDRAM device 1.

00: Column address A0 - A7 (default)

01: Column address A0 - A8

10: Column address A0 - A9

11: Column address A0 - A9, A11

Bits [19:16]: **RSV Reserved**

This bit is reserved. Be sure to set "0".

Bits [15:11]: **RSV Reserved**

This bit is reserved. Be sure to set "0", though the bit defaults to "0b11000".

Bits [10:9]: **TRCD [1:0] RAS-CAS delay**

Set the delay time from MRAS# to MCAS# (in number of cycles).

00: Reserved

01: 1 cycle

10: 2 cycles

11: 3 cycles (default)

Bit 8: **APCG Auto-precharge control**

Make setting for auto-precharge.

0: No auto-precharge (All the banks are precharged at the timing auto-refresh is performed.)

1: Auto-precharged (default)

Bits [7:6]: **RSV Reserved**
This bit is reserved. Be sure to set "0".

Bits [5:4]: **RSV Reserved**
This bit is reserved. Be sure to set "0".

Bits [3:2]: **BNUM [1:0] Number of banks**
Set the number of banks in the SDRAM connected.
BNUM0: For setting for SDRAM device 0
BNUM1: For setting for SDRAM device 1
0: 4-bank device (default)
1: Reserved (for 2-bank device)

Bits [1:0]: **RSV Reserved**
This bit is reserved. Be sure to set "0".

SDRAM detailed setting register (SDADVCNFG)
MEMC[0x74] Default value = 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
AREFWAIT [3:0]								CLK-FORCE	RESELF	RSV (0)		SELF [1:0]		RSV (0)	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [31:20]: **RSV Reserved**
This bit is reserved. Be sure to set "0".

Bits [19:16]: **SREFCNT [3:0] Number of cycles before re-entering self-refresh mode**
Set the number of cycles before the self-refresh mode is re-entered after SDRAM lost accesses. The setting is effective only when "1" is set to the RESELF bit (MEMC [0x74] bit 6). (Default = 0x07)

Bits [15:12]: **RSV Reserved**
This bit is reserved. Be sure to set "0".

Bits [11:8]: **AREFWAIT [3:0] Count of auto-refresh held**
If the memory controller interface is occupied at the timing of distributed refreshing performed, the requirements for refreshing are held, and are met after the occupancy is released. (Default = 0x03)

Bit 7: **CLKFORCE**
0: MCLK clock is output subject to the state of the memory controller. (Default)
1: MCLK clock is output regardless of the state of the memory controller.

Bit 6: **RESELF Self-refresh re-entry mode**
Controls whether to re-enter the SDRAM into the self-refresh mode after the SDRAM in the self-refresh mode is activated for R/W.
0: The SDRAM is not re-entered in the self-refresh mode. (Default)
1: The SDRAM is re-entered in the self-refresh mode if it gets no access for SREFCNT [3:0] cycles.

Bits [5:4]: **RSV Reserved**
This bit is reserved. Be sure to set "0".

Bits [3:2]: **SELF [1:0] Self-refresh mode ON/OFF**
Cleared to zero automatically if the SDRAM in the self-refresh mode is activated for R/W (except the RESELF mode is set).
SELF0: For setting for SDRAM device 0
SELF1: For setting for SDRAM device 1
0: Exits from the self-refresh mode. (Default)

1: The SDRAM is set to the self-refresh mode.

Bits [1:0]: **RSV Reserved**
This bit is reserved. Be sure to set "0".

Initialization control register (SDINIT)
MEMC[0x80] Default value = 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 [1:0]		RSV (0)	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Note: Don't set plural instructions in this register at the same time.
Use normally INIT_SD to initialize the SDRAM.
Use other instructions if manual initialization is required.
Select the target devices with DEVSEL[1:0] (MEMC[0x80] bits[3:2]) for each instruction.

Bits [31:16]: **RSV Reserved**
This bit is reserved. Be sure to set "0".

Bit 15: **INIT_SD Initialize SDRAM**
Performs the sequence for initializing the SDRAM automatically. Select the target devices with DEVSEL (bits [3:0] in this register).
0: No action (default)
1: SDRAM initialized. (Returns to "0" automatically upon completion of initialization.)

Bits [14:8]: **RSV Reserved**
This bit is reserved. Be sure to set "0".

Bit 7: **LMR Mode register set**
Issues the mode register set instruction to SDRAM.
0: No action (default)
1: Issues the mode register set instruction. (Returns to "0" automatically upon issuance.)

Bit 6: **AREF Auto-refresh**
Issues the auto-refresh instruction to SDRAM.
0: No action (default)
1: Issues the auto-refresh instruction. (Returns to "0" automatically upon issuance.)

Bit 5: **PCGALL Precharge all**
Issues the all-bank precharge instruction to SDRAM.
0: No action (default)
1: Issues the all-bank precharge instruction. (Returns to "0" automatically upon issuance.)

Bit 4: **RSV Reserved**
This bit is reserved. Be sure to set "0".

Bits [3:2]: **DEVSEL [1:0] Select device**
Selects the device an instruction is issued to.
DEVSEL0: For setting for SDRAM device 0
DEVSEL1: For setting for SDRAM device 1
0: No selection (default)
1: Selected

Bits [1:0]: **RSV Reserved**
This bit is reserved. Be sure to set "0".

SDRAM refresh timer register (SDREF)

MEMC[0x90] Default value = 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 [31:12]: **RSV Reserved**

This bit is reserved. Be sure to set “0”.

Bits [11:0]: **REFTIME [11:0] Refreshing interval**

Set the interval for issuing distributed refresh.

Set the number of the HCLK cycles compatible with the interval of distributed auto-refreshing.

For example, if the refreshing interval is 16μsec and the frequency of the system clock HCLK is 10MHz, then set:

16μs × 10MHz = 160 cycles (=1010_0000b).

SDRAM status register (SDSTAT)

MEMC[0xA0] Default value = 0x0000_0202 Read Only

RSV															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV				DEVST1 [3:0]				RSV				DEVST0 [3:0]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits [31:12]: **RSV Reserved**

This bit is reserved. Be sure to set “0”.

Bits [11:8]: **DEVST1 [3:0] Status of SDRAM device 1**

Shows the current state of SDRAM device 1.

DEVST1 [0] 1: Device available (enabled and initial settings completed)
0: Device unavailable

DEVST1 [1] 1: Device idle or suspended
0: Device in other states

DEVST1 [2] 1: ROW of either bank of the device is activated.
0: ROW inactive

DEVST1 [3] 1: Device in the self-refresh mode
0: Device not in the self-refresh mode

Bits [7:4]: **RSV Reserved**

This bit is reserved. Be sure to set “0”.

Bits [3:0]: **DEVST0 [3:0] Status of SDRAM device 0**

Shows the current state of SDRAM device 0.

DEVST0 [0] 1: Device available (enabled and initial settings completed)
0: Device unavailable

DEVST0 [1] 1: Device idle or suspended
0: Device in other states

DEVST0 [2] 1: ROW of either bank of the device is activated.
0: ROW inactive

DEVST0 [3] 1: Device in the self-refresh mode
0: Device not in the self-refresh mode

15. INTERRUPT CONTROLLER (INT)

15.1 Description

This interrupt controller (INTC) can handle two fast interrupt requests (FIQs) and 32 normal interrupt requests (IRQs). Table 15.1 shows the correspondence of peripheral circuits/functions to their respective interrupt request inputs. After reset, all interrupt requests are active HIGH level trigger inputs. Although INT [7:0] is set to the low level trigger setting right after reset, this interrupt controller has made it possible to select the polarity and level/edge trigger for the input signal if necessary.
The interrupt controller processes FIQ and IRQ inputs and then outputs two interrupt request signals to the ARM720T core: nFIQ and nIRQ.

Table 15.1 List of Interrupt Sources

Interrupt Type	Interrupt Level	Interrupt 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 request by the register
	IRQ2	ARM720T COMMRx	Debug Communication Port
	IRQ3	ARM720T COMMTx	Debug Communication Port
	IRQ4	Timer A	16-bit Timer Channel 0
	IRQ5	Timer A	16-bit Timer Channel 1
	IRQ6	Timer A	16-bit Timer Channel 2
	IRQ7	Reserved	
	IRQ8	JPEG1 Control	
	IRQ9	DMAC1	DMAC on AHB1
	IRQ10	JPEG1 DMAC	
	IRQ11	Camera 1 I/F IPC-1	ORed camera 1 and IPC-1
	IRQ12	ARS	
	IRQ13	DMAC2	DMA INT (DMAC for JPEG on AHB2)
	IRQ14 (*)	GPIOA or GPIOB	Selects an interrupt input pin between GPIOA and GPIOB
	IRQ15	SPI	SPI TXRDY/RXRDY
	IRQ16	I^2C	Transfer Complete
	IRQ17	UART1	UART TXRDY/RXRDY
	IRQ18	RTC	Alarm or Timer tick
	IRQ19	CF card Interface	
	IRQ20 (**)	INTGB	GPIOB7 to GPIOB0 direct input representative bit
	IRQ21	Timer B	
	IRQ22	Reserved	
	IRQ23	JPEG2 Control	
	IRQ24	JPEG2 DMAC	
	IRQ25	Camera 2 I/F IPC-2	ORed camera 2 and IPC-2
	IRQ26	UART2	
	IRQ27	UART3	
	IRQ28	SD Memory	
	IRQ29	USB DEV	
	IRQ30	ADC	
	IRQ31	I^2S	

Note (*): Sets an interrupt request based on GPIOA [7:0] or GPIOB [7:0]. For details, see the detailed description of registers from GPIO [0x60] to GPIO [0x6C] in "26. GPIO Detailled Description of Registers".
Note (**): Sets an interrupt request based on GPIOB [7:0]. For details, see the detailed description of IRQxx registers.
Note (***): Direct input from the GPIOB0 pin. (The default is active-low level interrupt request. However, the interrupt level on the inside becomes active-high level because it reverses internally.)

Table 15.2 List of Interrupt Sources

Interrupt Type	Interrupt Level	Interrupt Source	Description
GPIOB Interrupt Request IRQxx	IRQGB0(*)	INT0	GBIOB0 direct input
	IRQGB1(*)	INT1	GBIOB1 direct input
	IRQGB2(*)	INT2	GBIOB2 direct input
	IRQGB3(*)	INT3	GBIOB3 direct input
	IRQGB4(*)	INT4	GBIOB4 direct input
	IRQGB5(*)	INT5	GBIOB5 direct input
	IRQGB6(*)	INT6	GBIOB6 direct input
	IRQGB7(*)	INT7	GBIOB7 direct input

Note (*): Direct input from the GPIOB [7:0] pin. (The default is active-low level interrupt request. However, the interrupt level on the inside becomes active-high level because it reverses internally.) These interrupt settings (enable, polarity, level, etc.) can be changed only by the control register of the interrupt controller. Unlike IRQ14 (*), they cannot be changed by the GPIO[0x60] ~ GPIO[0x6C].
In case of using this interrupt function, must hold the level of external interrupt signal in at least 4 system terms, after the level is changed. Otherwise use the interrupt function of GPIO module (IRQ14).

15.2 Block Diagram

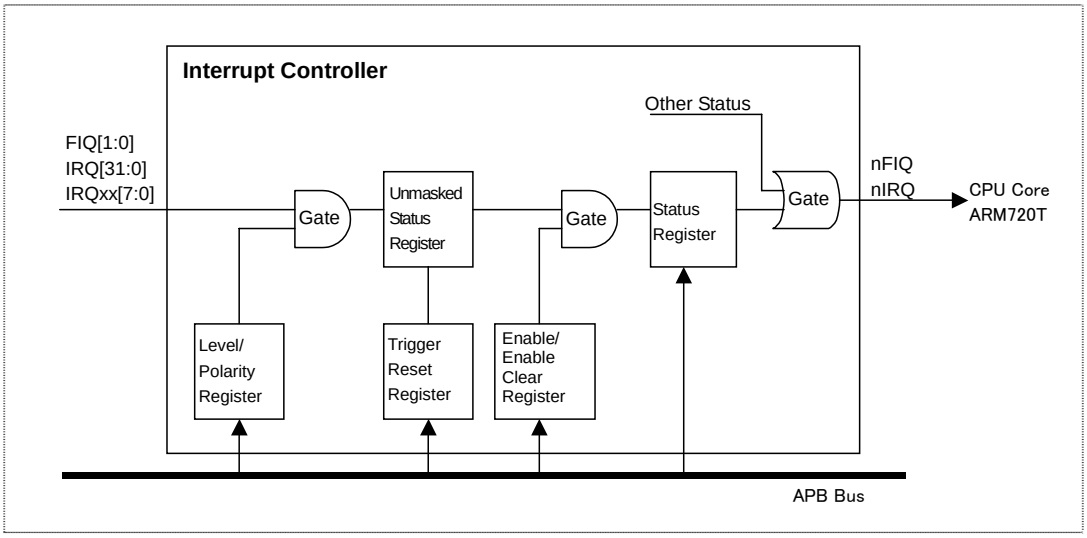


Figure 15.1 Interrupt Controller Block Diagram

15.3 FIQ

As an interrupt output, a watchdog timer and GPIO port B [0] are assigned to FIQ0 and FIQ1, respectively. These two FIQ signals are used to generate nFIQ to be input to ARM720T.

15.4 IRQ

Interrupt requests from the internal devices are assigned to the IRQs. All IRQ interrupt requests are ORed to generate nIRQ signal to be input to ARM720T. The interrupt controller has a total of 32 interrupt request inputs, IRQ31-IRQ0, each of which corresponds to a bit of the interrupt control register with a ratio of 1:1. For example, IRQ0 is assigned to Bit 0, while IRQ1 is assigned to Bit 1. For the correspondence, see Table 15.1.
For the external interrupt, INT0-INT7, interrupt requests are received at IRQGB0-IRQGB7 and then assigned as an external representative interrupt to IRQ20.

15.5 Priority Level

Priority levels (from 0=Lowest to 3=Highest) of interrupt can be set for each IRQ. If several interrupt causes with the same precedence level occur, the interrupt with the smallest IRQ number is preceded.
If this function is not used (if INTC registers added to S2S65A30 are not set), the operation is compatible with S2S65A00.
This function is used in the sample software of μITRON version, but not used in Linux version.

15.6 External Pins

The following lists the external pins related to the interrupt controller (INT).

Pin Name	Input/Output	Pin Functions	Multiplex Pin*/Remarks
FIQ1	I	Fast Interrupt Pin 1	GPIOB0/I^2S_WS
INT0	I	External Interrupt Pin 0	GPIOB0/I^2S_WS
INT1	I	External Interrupt Pin 1	GPIOB1/I^2S_SCK
INT2	I	External Interrupt Pin 2	GPIOB2/I^2S_SDO
INT3	I	External Interrupt Pin 3	GPIOB3/I^2S_SDI
INT4	I	External Interrupt Pin 4	GPIOB4/TimerA0out
INT5	I	External Interrupt Pin 5	GPIOB5/TimerA1out
INT6	I	External Interrupt Pin 6	GPIOB6/TimerA2out
INT7	I	External Interrupt Pin 7	GPIOB7/TimerBIn

Note (*): The INT-related external interrupt pins use the GPIO pin inputs. If you use a GPIO pin as an interrupt input, set the internal register for interrupt control. To use for other pin functions, set the value to “Function 1 or 2 of other than GPIO” by using the GPIO pin function register.

15.7 Registers

15.7.1 List of Registers

The base address where the control register of the interrupt controller is located is 0xFFFF_F000.

Table 15.3 List of Registers (Base Address: 0xFFFF_F000)

Address Offset	Register Name	Default Value	R/W	Data Access Size
0x000	IRQ Status Register	0x0000_0000	RO	32
0x004	IRQ Unmasked 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
0x020	IRQxx Status Register	0x0000_0000	RO	32
0x024	IRQxx Unmasked Status Register	0x0000_0000 (*1)	RO	32
0x028	IRQxx Enable Register	0x0000_0000	R/W	32
0x02C	IRQxx Enable Clear Register	0x0000_0000	WO	32
0x040	IRQ Priority Register 0	0x0000_0000	R/W	32
0x044	IRQ Priority Register 1	0x0000_0000	R/W	32
0x048	IRQ Priority Register 2	0x0000_0000	R/W	32
0x04C	IRQ Priority Register 3	0x0000_0000	R/W	32
0x050	IRQ Request Number Register	0x0000_FFFF	RO	32
0x054	IRQ Present Priority Register	0x0000_00FF	R/W	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
0x0A0	IRQxx Level Register	0x0000_0000	R/W	32
0x0A4	IRQxx Polarity Register	0x0000_0FFF	R/W	32
0x0A8	IRQxx Trigger Reset Register	0x0000_0000	WO	32
0x100	FIQ Status Register	0x0000_0000	RO	32
0x104	FIQ Unmasked 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

Note (*1): Default value of IRQ/IRQxx/FIQ Unmask Status Registers are different according to the system configuration.

Note (*2): Registers between 0x040 and 0x054 are newly added. They are not used in Linux version sample software.

15.7.2 Detailed Description of Registers

If not otherwise specified, the default value of any register bit that is not reserved is “0”.

IRQ Status Register

INT[0x000] Default value = 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 [31:0]: **IRQ[31:0] Status**

Indicates the (masked) generation status of an interrupt request from each unit.

0 (r): No interrupt request exists

1 (r) : An interrupt request exists

These status bits indicate the status of the IRQ for which interrupt is enabled by the IRQ enable register (INT[0x008]). If an interrupt request from an interrupt disabled unit is generated, the bit will not be “1”. The interrupt request that is set to “1” is sent to the CPU. These bits return to “0” by clearing the corresponding bits of the IRQ unmasked status register.

Bits [31:0] correspond to IRQ [31:0] with the same number on one-to-one basis. For information on the interrupt that each bit indicates, see Table 15.1.

IRQ Unmasked Status Register

INT[0x004] Default value = 0x0000_0000 Read Only

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

Bits [31:0]: **IRQ[31:0] Unmasked Status**

Indicates the (unmasked) generation status of an interrupt cause of each unit.

0 (r) : No interrupt cause exists

1 (r) : An interrupt cause exists

These status bits indicate the status of the IRQ where the interrupt has not been masked by the IRQ enable register (INT[0x008]). If an interrupt cause from an interrupt disabled unit is generated, the bit will also be “1”. For a level trigger interrupt, these bits return to “0” by clearing the interrupt flag of each unit. For an edge trigger interrupt, they return to “0” by writing “1” to the corresponding bits of the IRQ trigger reset register (INT[0x088])

Bits [31:0] correspond to IRQ [31:0] with the same number on one-to-one basis. For information on the interrupt that each bit indicates, see Table 15.1.

IRQ Enable Register

INT[0x008] Default value = 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 [31:0]: **IRQ[31:0] Enable**

Enables interrupt request (IRQ) input from each unit.

0 (r): Interrupt request is disabled
1 (r): Interrupt request is enabled
0 (w): Invalid
1 (w): Enables interrupt request input

By reading this register, you can determine for each interrupt request (IRQ) input whether it is currently enabled (bit is set to "1") or disabled (bit is set to "0").

When performing a write operation to the register, writing "1" enables interrupt request (IRQ) input. The interrupt controller receives the IRQ input corresponding to the bit and then outputs an interrupt request to the CPU. When "0" is written to the register, it is ignored and accessing the register does not disable interrupt request (IRQ) input. To clear the bits, the IRQ enable clear register (INT[0x00C]) is used.

When reset, all interrupts are set to disabled.

Bits [31:0] correspond to IRQ [31:0] with the same number on one-to-one basis. For information on the interrupt that each bit indicates, see Table 15.1.

IRQ Enable Clear Register

INT[0x00C] Default value = 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 [31:0]: **IRQ[31:0] Enable Clear**

Disables (masks) the interrupt request (IRQ) input from each unit.

0 (w): Invalid
1 (w): Disables interrupt request input

Writing "1" to these bits clears the corresponding IRQ enable bits of the IRQ enable register (INT[0x008]) and the interrupt request (IRQ) input is disabled. Writing "0" is ignored.

Bits [31:0] correspond to IRQ [31:0] with the same number on one-to-one basis. For information on the interrupt that each bit indicates, see Table 15.1.

Software IRQ Register

INT[0x010] Default value = 0x0000_0000 Write Only

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

Bit 1: **Software IRQ**

Controls an IRQ1 software interrupt.

0 (w): Clears a software interrupt request

1 (w): Generates a software interrupt request

The status of the software interrupt cause that is currently set can be read from bit 1 of the IRQ unmasked status register (INT[0x004]).

Bit 0: **RSV Reserved**

IRQxx Status Register

INT[0x020] Default value = 0x0000_0000 Read Only

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

Bits [31:0]: **IRQxx[31:0] Status**

IRQxx[31:12] : n/a

IRQxx[11:8] : Reserved

IRQxx[7:0] : IRQxx (IRQGB[7:0]) status

Indicates the (masked) generation status of an interrupt request from each unit.

0 (r) : No interrupt request exists

1 (r) : An interrupt request exists

These status bits indicate the status of the IRQxx for which interrupt is enabled by the IRQxx enable register (INT[0x028]). If an interrupt request from an interrupt disabled unit is generated, the bit will not be "1". The interrupt request that is set to "1" is sent to the CPU. These bits return to "0" by clearing the corresponding bits of the IRQ unmasked status register.

Bits [7:0] correspond to IRQxx [7:0] with the same number on one-to-one basis. For information on the interrupt that each bit indicates, see Table 15.2.

IRQxx Unmasked Status Register

INT[0x024] Default value = 0x0000_0000 Read Only

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

Bits [31:0]: **IRQxx[31:0] Unmasked Status**

IRQxx[31:12] : n/a
IRQxx[11:8] : Reserved
IRQxx[7:0] : IRQxx (IRQGB[7:0]) Unmasked Status

Indicates the (unmasked) generation status of an interrupt cause of each unit.

0 (r): No interrupt cause exists
1 (r): An interrupt cause exists

These status bits indicate the status of the IRQxx where the interrupt has not been masked by the IRQxx enable register (INT[0x028]). If an interrupt cause from an interrupt disabled unit is generated, the bit will also be "1". For a level trigger interrupt, these bits return to "0" by clearing the interrupt flag of each unit. For an edge trigger interrupt, they return to "0" by writing "1" to the corresponding bits of the IRQxx trigger reset register (INT[0x0A8])

Bits [7:0] correspond to IRQxx [7:0] with the same number on one-to-one basis. For information on the interrupt that each bit indicates, see Table 15.2.

IRQxx Enable Register

INT[0x028] Default value = 0x0000_0000 Read/Write

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

Bits [31:0]: **IRQxx[31:0] Enable**

IRQxx[31:12] : n/a
IRQxx[11:8] : Reserved
IRQxx[7:0] : IRQxx (IRQGB[7:0]) Enable Register

Enables the interrupt request (IRQ) input from each unit.

0 (r): Interrupt request is disabled
1 (r): Interrupt request is enabled
0 (w): Ignored
1 (w): Enables interrupt request input

By reading this register, you can determine for each interrupt request (IRQxx) input whether it is currently enabled (bit is set to "1") or disabled (bit is set to "0").

When performing a write operation to the register, writing "1" enables interrupt request (IRQxx) input. The interrupt controller receives the IRQxx input corresponding to the bit and then outputs an interrupt request to the CPU. When "0" is written to the register, it is ignored and accessing the register does not disable the interrupt request (IRQxx) input. To clear the bits, the IRQxx enable clear register (INT[0x02C]) is used.

When reset, all interrupts are set to disabled.

Bits [7:0] correspond to IRQ [7:0] with the same number on one-to-one basis. For information on the interrupt that each bit indicates, see Table 15.2.

IRQxx Enable Clear Register

INT[0x02C] Default value = 0x0000_0000 Write Only

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

Bits [31:0]: **IRQxx[31:0] Enable Clear**

IRQxx[31:12] : n/a
IRQxx[11:8] : Reserved
IRQxx[7:0] : IRQxx (IRQGB[7:0]) Enable Clear

Disables (masks) the interrupt request (IRQxx) input from each unit.

0 (w): Ignored
1 (w): Disables interrupt request input

Writing "1" to these bits clears the corresponding IRQxx enable bits of the IRQxx enable register (INT[0x028]) and the interrupt request (IRQxx) input is disabled. Writing "0" is ignored.

Bits [7:0] correspond to IRQxx [7:0] with the same number on one-to-one basis. For information on the interrupt that each bit indicates, see Table 15.2.

IRQ Priority Register 0

INT[0x040] Default value = 0x0000_0000 Read/Write

RSV		IRQ07PR		RSV		IRQ06PR		RSV		IRQ05PR		RSV		IRQ04PR	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV		IRQ03PR		RSV		IRQ02PR		RSV		IRQ01PR		RSV		IRQ00PR	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

IRQ Priority Register 1

INT[0x044] Default value = 0x0000_0000 Read/Write

RSV		IRQ15PR		RSV		IRQ14PR		RSV		IRQ13PR		RSV		IRQ12PR	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV		IRQ11PR		RSV		IRQ10PR		RSV		IRQ09PR		RSV		IRQ08PR	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

IRQ Priority Register 2

INT[0x048] Default value = 0x0000_0000 Read/Write

RSV		IRQ23PR		RSV		IRQ22PR		RSV		IRQ21PR		RSV		IRQ20PR	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV		IRQ19PR		RSV		IRQ18PR		RSV		IRQ17PR		RSV		IRQ16PR	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

IRQ Priority Register 3

INT[0x04C] Default value = 0x0000_0000 Read/Write

RSV		IRQ31PR		RSV		IRQ30PR		RSV		IRQ29PR		RSV		IRQ28PR	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV		IRQ27PR		RSV		IRQ26PR		RSV		IRQ25PR		RSV		IRQ24PR	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

IRQxxPR

Sets the priority level of interrupt requests by 2 bits from 0 (lowest) to 3 (highest). If several interrupt causes with the same priority level occur, the interrupt with the smallest IRQ number is preceded.

RSV Reserved

IRQ Request Number Register

INT[0x050] Default value = 0xFFFF — Read Only

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

Bits [31:16]: **RSV[31:16] Reserved**

Bits [15:8]: **IRQ Req Priority[15:8]**

Indicates the priority level set to the IRQ number of occurred interrupt cause. Indicates 0xFF if no interrupt cause has occurred.

Bits [7:0]: **IRQ Number[7:0]**

Indicates the IRQ number with the current highest priority for processing. Indicates 0xFF if no interrupt cause has occurred.

IRQ Present Priority Register

INT[0x054] Default value = 0x00FF — Read/Write

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

Bits [31:4]: **RSV[31:4] Reserved**

Bits [3:0]: **IRQPR[3:0]**

Indicates the current priority level of interrupts. Interrupt causes with a priority level equal to or lower than that set in this register are automatically masked by the interrupt controller, and do not request CPU for interrupt. 0xF means a value lower than priority level 0.

This register is changed by software. It is not set by hardware.

IRQ Level Register

INT[0x080] Default value = 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**

Sets a trigger mode for each IRQ input.

0 (r/w): Level trigger mode

1 (r/w): Edge trigger mode

This setting determines whether interrupt request (IRQ) signal is sampled according to the level or falling/rising edge of the signal. The polarity of a signal (LOW level/rising edge or HIGH level/falling edge) is set by the IRQ polarity register (INT[0x084]). Bits [31:0] correspond to IRQ [31:0] with the same number on one-to-one basis. For information on the interrupt that each bit indicates, see Table 15.1.

Note: Normally, use the value of this register that is set at reset without changing it.

IRQ Polarity Register

INT[0x084] Default value = 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 [31:0]: **IRQ[31:0] Polarity**

Sets the polarity for each IRQ signal.

0 (r/w): LOW level/falling edge

1 (r/w): HIGH level/rising edge

An interrupt request (IRQ) signal is sampled according to this setting. Level/edge selection is done by the IRQ level register (INT[0x080]). Bits [31:0] correspond to IRQ [31:0] with the same number on one-to-one basis. For information on the interrupt that each bit indicates, see Table 15.1.

Note: Normally, use the value of this register that is set at reset without changing it.
Please set bit28 to 0 when you use the interrupt of the SD card detection for the factor.

IRQ Trigger Reset Register

INT[0x088] Default value = 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 [31:0]: **IRQ[31:0] Trigger Reset**

Clears the unmasked status of the interrupt that is set to edge trigger mode.

0 (w): Invalid

1 (w): Clears the interrupt status

Writing “1” clears the corresponding status bits in the IRQ unmasked status register (INT[0x004]) that are set to edge trigger mode. Bits [31:0] correspond to IRQ [31:0] with the same number on one-to-one basis. For information on the interrupt that each bit indicates, see Table 15.1.

IRQxx Level Register

INT[0x0A0] Default value = 0x0000_0000 Read/Write

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

Bits [31:0]: **IRQxx[31:0] Level**

IRQxx[31:12] : n/a
IRQxx[11:8] : Reserved
IRQxx[7:0] : IRQxx (IRQGB[7:0]) Level

Sets a trigger mode for each IRQxx input.

0 (r/w): Level trigger mode
1 (r/w): Edge trigger mode

This setting determines whether interrupt request (IRQxx) signal is sampled according to the level or falling/rising edge of the signal. The polarity of a signal (LOW level/rising edge or HIGH level/falling edge) is set by the IRQxx polarity register (INT[0x0A4]).

Bits [7:0] correspond to IRQxx [7:0] with the same number on one-to-one basis. For information on the interrupt that each bit indicates, see Table 15.2.

IRQxx Polarity Register

INT[0x0A4] Default value = 0x0000_0FFF Read/Write

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

Bits [31:0]: **IRQxx[31:0] Polarity**

IRQxx[31:12] : n/a
IRQxx[11:8] : Reserved
IRQxx[7:0] : IRQxx (IRQGB[7:0]) Polarity

Sets the polarity for each IRQxx signal.

0 (r/w): LOW level/falling edge
1 (r/w): HIGH level/rising edge

An interrupt request (IRQxx) signal is sampled according to this setting. Level/edge selection is done by the IRQxx level register (INT[0x0A0]).

Bits [7:0] correspond to IRQxx [7:0] with the same number on one-to-one basis. For information on the interrupt that each bit indicates, see Table 15.2.

IRQxx Trigger Reset Register

INT[0x0A8] Default value = 0x0000_0000 Write Only

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

Bits [31:0]: **IRQxx[31:0] Trigger Reset**

IRQxx[31:12] : n/a
IRQxx[11:8] : Reserved
IRQxx[7:0] : IRQxx (IRQGB[7:0]) Trigger Reset

Clears the unmasked status of the interrupt that is set to edge trigger mode.

0 (w): Ignored
1 (w): Clears the interrupt status

Writing "1" clears the corresponding status bits in the IRQxx unmasked status register (INT[0x024]) that are set to edge trigger mode.

Bits [7:0] correspond to IRQxx [7:0] with the same number on one-to-one basis. For information on the interrupt that each bit indicates, see Table 15.2.

FIQ Status Register

INT[0x100] Default value = 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 [1:0]: **FIQ [1:0] Status**

Indicates the (masked) generation status of an FIQ interrupt request.

0 (r) : No interrupt request exists
1 (r) : An interrupt request exists

These status bits indicate the status of the FIQ for which interrupt is enabled by the FIQ enable register (INT[0x108]). If an interrupt request from an interrupt disabled unit is generated, the bit will not be "1". The interrupt request that is set to "1" is sent to the CPU. These bits return to "0" by clearing the corresponding bits of the FIQ unmasked status register.

Bits [1:0] correspond to FIQ [1:0] with the same number on one-to-one basis.

FIQ Unmasked Status Register

INT[0x104] Default value = 0x0000_0000 Read Only

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
n/a															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
n/a														FIQ[1:0] Unmasked Status	

Bits [1:0]: **FIQ[1:0] Unmasked Status**

Indicates the (unmasked) generation status of an FIQ interrupt cause.

0 (r) : No interrupt cause exists

1 (r) : An interrupt cause exists

These status bits indicate the status of the FIQ where the interrupt has not been masked by the FIQ enable register (INT[0x108]). If an interrupt cause from an interrupt disabled unit is generated, the bit will also be "1". For a level trigger interrupt, these bits return to "0" by clearing the interrupt flag of each unit. For an edge trigger interrupt, they return to "0" by writing "1" to the corresponding bits of the FIQ trigger reset register (INT[0x188]).

Bits [1:0] correspond to FIQ [1:0] with the same number on one-to-one basis.

FIQ Enable Register

INT[0x108] Default value = 0x0000_0000 Read/Write

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
n/a															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
n/a														FIQ [1:0] Enable	

Bits [1:0]: **FIQ [1:0] Enable**

Enables the fast interrupt request (FIQ) input.

0 (r) : Interrupt request is disabled

1 (r) : Interrupt request is enabled

0 (w) : Invalid

1 (w): Enables interrupt request input

By reading this register, you can determine for each interrupt request (FIQ) input whether it is currently enabled (bit is set to "1") or disabled (bit is set to "0").

When performing a write operation to the register, writing "1" enables interrupt request (FIQ) input. The interrupt controller receives the FIQ input corresponding to the bit and then outputs an interrupt request to the CPU. When "0" is written to the register, it is ignored and accessing the register does not disable the interrupt request (FIQ) input. To clear the bits, the FIQ enable clear register (INT[0x10C]) is used.

When reset, all interrupts are set to disabled.

Bits [1:0] correspond to FIQ [1:0] with the same number on one-to-one basis.

FIQ Enable Clear Register

INT[0x10C] Default value = 0x0000_0000 Write Only

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
n/a															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
n/a														FIQ [1:0] Enable Clear	

Bits [1:0]: **FIQ [1:0] Enable Clear**

Disables (masks) the fast interrupt request (FIQ) input.

0 (w) : Invalid

1 (w) : Disables interrupt request input

Writing "1" to these bits clears the corresponding FIQ enable bits of the FIQ enable register (INT[0x108]) and the interrupt request (FIQ) input is disabled. Writing "0" is ignored. Bits [1:0] correspond to FIQ [1:0] with the same number on one-to-one basis.

FIQ Level Register

INT[0x180] Default value = 0x0000_0000 Read/Write

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
n/a															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
n/a														FIQ [1:0] Level	

Bits [1:0]: **FIQ [1:0] Level**

Sets a trigger mode for each FIQ input.

0 (r/w): Level trigger mode

1 (r/w): Edge trigger mode

This setting determines whether interrupt request (FIQ) signal is sampled according to the level or falling/rising edge of the signal. The polarity of a signal (LOW level/rising edge or HIGH level/falling edge) is set by the FIQ polarity register (INT[0x184]). Bits [1:0] correspond to FIQ [1:0] with the same number on one-to-one basis.

FIQ Polarity Register

INT[0x184] Default value = 0x0000_0003 Read/Write

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
n/a															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
n/a														FIQ [1:0] Polarity	

Bits [1:0]: **FIQ [1:0] Polarity**

Sets the polarity for each FIQ signal.

0 (r/w): LOW level/falling edge

1 (r/w): HIGH level/rising edge

An interrupt request (FIQ) signal is sampled according to this setting. Level/edge selection is done by the FIQ level register (INT[0x180]). Bits [1:0] correspond to FIQ [1:0] with the same number on one-to-one basis.

FIQ Trigger Reset Register

INT[0x188] Default value = 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 [1:0]: **FIQ [1:0] Trigger Reset**

Clears the unmasked status of the interrupt that is set to edge trigger mode.

0 (w): Invalid

1 (w): Clears the interrupt status

Writing “1” to these bits clears the corresponding status bits in the FIQ unmasked status register (INT[0x104]) that are set to edge trigger mode. Bits [1:0] correspond to FIQ [1:0] with the same number on one-to-one basis.

16. UART1/2/3

16.1 Description

UART1/2/3 are asynchronous data transfer interfaces that are compatible with the industry standard 16550. They perform serial conversion of CPU parallel data and transfer it to the peripheral devices, as well as receive serial data from the peripheral devices and convert it to parallel data.
S2S65A30 has three UART modules (UART1/2/3) whose functionalities are identical.

16.2 Block Diagram

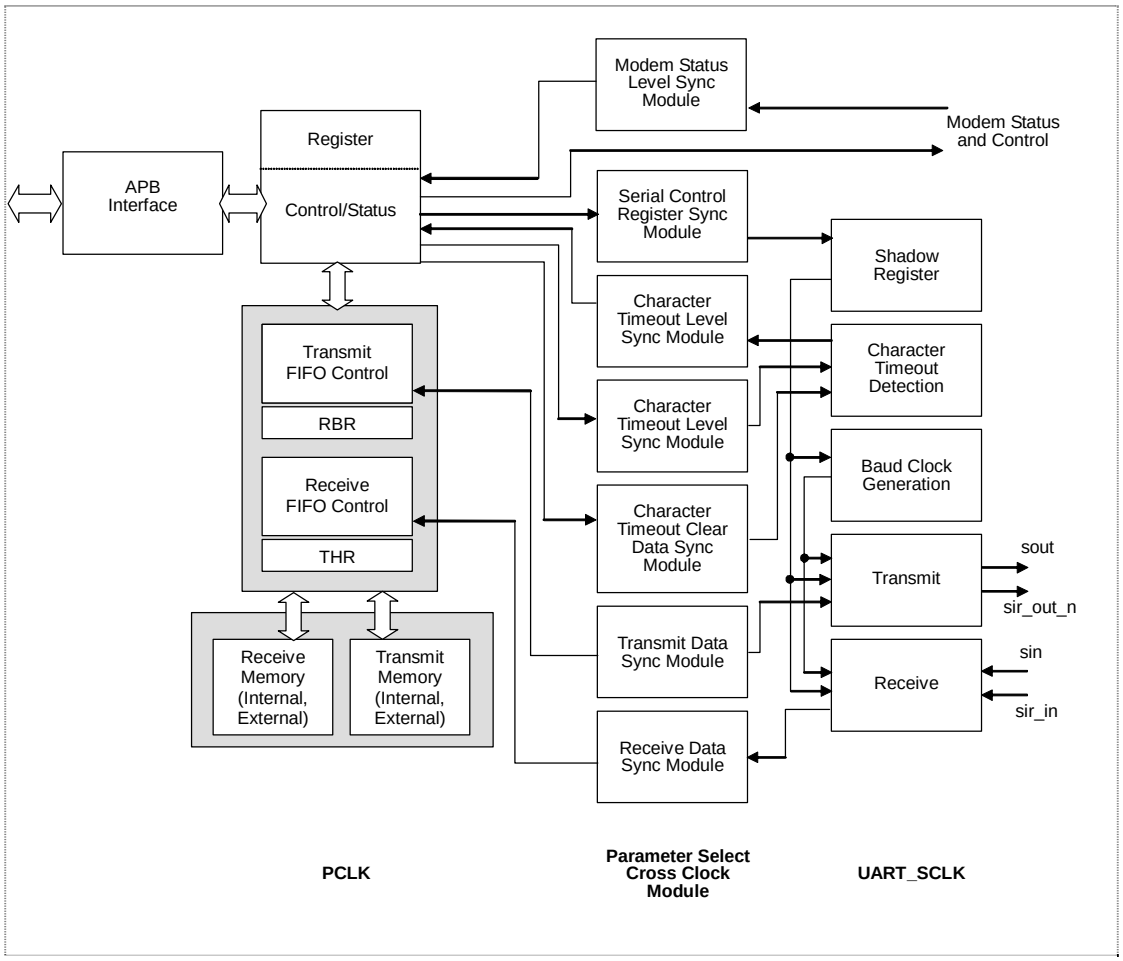


Figure 16.1 UART1/2/3 Block Diagram

16.3 External Pins

The external pins related to UART1/2/3 are shown below.

Pin Name	Input/Output	Pin Functions	Multiplex Pin
TXD1	Output	UART1 transmit data output	GPIOA0(*)
RXD1	Input	UART1 transmit data input	GPIOA1(*)
RTS1	Output	UART1 request to send output	GPIOA2(*)
CTS1	Input	UART1 clear to send input	GPIOA3(*)
TXD2	Output	UART2 transmit data output	GPIOA4(*)
RXD2	Input	UART2 transmit data input	GPIOA5(*)
RTS2	Output	UART2 request to send output	GPIOA6(*)
CTS2	Input	UART2 clear to send input	GPIOA7(*)
UART3_CLK	Input	UART3 clock input	GPIOC3(**)
TXD3	Output	UART3 transmit data output	GPIOC4(**)
RXD3	Input	UART3 transmit data input	GPIOC5(**)
RTS3	Output	UART3 request to send output	GPIOC6(**)
CTS3	Input	UART3 clear to send input	GPIOC7(**)

Note (*): The external pins for UART1/2 are multiplexed with GPIO pins or other pins. You can use the functions for UART1/2 by setting “Function 1 of other than GPIO” through the GPIO pin function register.

Note (**):The external pins for UART3 are multiplexed with GPIO pins or other pins. You can use the functions for UART1/2 by setting “Function 2 of other than GPIO” through the GPIO pin function register.

16.4 Description of Registers

The default base addresses where the UART1/2/3 control registers are located are as follows.
UART1: 0x FFFF_5000
UART2: 0x FFFD_4000
UART3: 0x FFFD_5000

If not otherwise specified, the default value of any register bit that is not reserved is “0”.

Note: Accesses to UART1/2/3 can read and write regardless whether the access size is 8 bits, 16 bits, or 32 bits, as long as UART1/2/3 are accessed at 32-bit boundary offset.

16.4.1 List of Registers

Table 16.1 List of Registers
Base Address: 0xFFFF_5000 (UART1)
0xFFFD_4000 (UART2)
0xFFFD_5000 (UART3)

Address Offset	DLAB	Register Name	Register Abbreviation	Default Value	R/W	Data Access Size
0x00	0	Receive Buffer Register	RBR	0x00	RO	8/16/32bit
0x00	0	Transmit Holding Register	THR	—	WO	8/16/32bit
0x00	1	Divisor Latch LSB Register	DLL	0x00	R/W	8/16/32bit
0x04	0	Interrupt Enable Register	IER	0x00	R/W	8/16/32bit
0x04	1	Divisor Latch MSB Register	DLM	0x00	R/W	8/16/32bit
0x08	—	Interrupt Identify Register	IIR	0x01	RO	8/16/32bit
0x08	—	FIFO Control Resister	FCR	—	WO	8/16/32bit
0x0C	—	Line Control Register	LCR	0x00	R/W	8/16/32bit
0x10	—	Modem Control Register	MCR	0x00	R/W	8/16/32bit
0x14	—	Line Status Register	LSR	0x60	RO	8/16/32bit
0x18	—	Modem Status Register	MSR	0xEX	RO	8/16/32bit
0x1C	—	Scratch Register	SCR	0x00	R/W	8/16/32bit
0x20	—	Test 0 Register	T0	0x00	R/W	8/16/32bit
0x24	—	Test 1 Register	T1	0x00	R/W	8/16/32bit
0x28	—	Test Status 0 Register	TS0	—	RO	8/16/32bit
0x2C	—	Test Status 1 Register	TS1	0x01	RO	8/16/32bit
0x30	—	Test Status 2 Register	TS2	0x0F	RO	8/16/32bit
0x3C	—	Test Status 3 Register	TS3	0x02	RO	8/16/32bit

16.4.2 Precautions on Register Access

The behavior when an address between the UART1/2/3 control registers are accessed is not guaranteed. For example, the behavior when address offset 01h is accessed in byte units is not guaranteed. Be sure to use the specified address offsets to access the registers.
Registers 20h through 3Ch are for debugging UART itself. They cannot be used for other than debugging. The specifications of these registers may be changed in the future.

16.4.3 Detailed Description of Registers

Receive Buffer Register (RBR)

UART1/2/3[0x00] DLAB [0] Default value = 0x00 Read Only

Serial Receive Data (RBR[7:0])							
7	6	5	4	3	2	1	0

Bits [7:0]: **RBR[7:0] Serial Receive Data Bits [7:0]**

This register works as a receive buffer during a read operation when the DLAB bit (UART1/2/3[0x0C] Bit 7) of the line control register is "0". From Bits [7:0], the byte data received on a serial port can be read. The read data is effective only if the data ready bit (UART1/2/3[0x14] Bit 0) of the line status register is "1". Accessing this register reads the first data in the receive FIFO. When the receive FIFO is full, data that has already received in the FIFO is retained but data that is sent after that is lost.

Transmit Holding Register (THR)

UART1/2/3[0x00] DLAB [0] Default value = — Write Only

Serial Transmit Data (THR [7:0])							
7	6	5	4	3	2	1	0

Bits [7:0]: **THR[7:0] Serial Transmit Data Bits [7:0]**

This register works as a transmit buffer during a write operation when the DLAB bit (UART1/2/3[0x0C] Bit 7) of the line control register is "0". Up to 16 bytes of data can be written until the transmit FIFO becomes full. When the transmit FIFO becomes full, data written to this register after that is lost.

Divisor Latch LSB Register (DLL)

UART1/2/3[0x00] DLAB [1] Default value = 0x00 Read/Write

Divisor Latch LSB (DL[7:0])							
7	6	5	4	3	2	1	0

Bits [7:0]: **DL[7:0] Divisor Latch LSB Bits**

This register is used to set the dividing ratio of the source clock that determines the baud rate, when the DLAB bit (UART1/2/3[0x0C] Bit 7) of the line control register is "1". A setting value is 16 bits and this register is for the lower 8 bits. The higher 8 bits are set in the divisor latch MSB register (UART1/2/3[0x04]). A baud rate is determined by the following expression.

Baud rate = Input clock/(16×DL[15:0])

Table 16.4 shows the correspondence of baud rates and division values.

Interrupt Enable Register (IER)

UART1/2/3[0x04] DLAB [0] Default value = 0x00 Read/Write

7	6	5	4	3	2	1	0
Programmable Transmit Holding Empty Interrupt Enable (EPTBEI)	Reserved (0)			Modem Status Interrupt Enable (EDSSI)	Receive Line Status Interrupt Enable (ELSI)	Transmit Holding Empty Interrupt Enable (ETBEI)	Receive Data Ready Interrupt Enable (ERBFI)

This register works as an interrupt enable register for UART1/2/3 when the DLAB bit (UART1/2/3[0x0C] Bit 7) of the line control register is "0", and enables/disables five UART1/2/3 interrupts. The bit 7 function does not exist for 16550. Do not use Bit 7 to create a firmware compatible with 16550.

Bit 7: **EPTBEI Programmable Transmit Holding Register Empty Interrupt Enable**

Becomes effective only if the operation is in FIFO mode (UART1/2/3[0x08] Bit 0 = 1) and the transmit holding register empty interrupt is enabled (UART1/2/3[0x04] Bit 1 = 1).

0 (r/w): Interrupt is disabled

1 (r/w): Interrupt is enabled

Note: this bit does not exist in 16550. (not compatible with 16550.)

Bit 3: **EDSSI Modem Status Interrupt Enable**

0 (r/w): Interrupt is disabled

1 (r/w): Interrupt is enabled

Note that in auto CTS control mode (when the modem control register Bit 5 is "1" and the FIFO control register Bit 0 is set to "1"), interrupt does not occur because the CTS# input changes.

Bit 2: **ELSI Receive Line Status Interrupt Enable**

0 (r/w): Interrupt is disabled

1 (r/w): Interrupt is enabled

Bit 1: **ETBEI Transmit Holding Register Empty Interrupt Enable**

0 (r/w): Interrupt is disabled

1 (r/w): Interrupt is enabled

Bit 0: **ERBFI Receive Data Ready Interrupt Enable**

0 (r/w): Interrupt is disabled

1 (r/w): Interrupt is enabled

For description of each interrupt cause, see Table 16.2.

Divisor Latch MSB Register (DLM)

UART1/2/3[0x04] DLAB [1] Default value = 0x00 Read/Write

7	6	5	4	3	2	1	0
Divisor Latch MSB (DL[15:8])							

Bits [7:0]: **DL[15:8] Divisor Latch MSB Bits**

This register is used to set the dividing ratio of the source clock that determines the baud rate, when the DLAB bit (UART1/2/3[0x0C] Bit 7) of the line control register is "1". A setting value is 16 bits and this register is for the higher 8 bits. The lower 8 bits are set in the divisor latch LSB register (UART1/2/3[0x00]). A baud rate is determined by the following expression.

Baud rate = Input clock/(16×DL[15:0])

For the correspondence of baud rates and division values, see Table 16.4.

Interrupt Identify Register (IIR)

UART1/2/3[0x08] Default value = 0x01 Read Only

FIFO Enable (FFEN [1:0])		Reserved		Interrupt ID (IID [3:0])			
7	6	5	4	3	2	1	0

Bits [7:6]: **FFEN [1:0] FIFO Enable Status Bits**

Indicate if the transmit and receive FIFOs are enabled or disabled.

00: FIFO is disabled

11: FIFO is enabled

Bits [3:0]: **IID [3:0] Interrupt ID Bits [3:0]**

Identify the source of the interrupt that currently occurs on UART1/2/3.

Table 16.2 UART1/2/3 Interrupt

IID [3:0]	Interrupt Type	Interrupt Source	How to Reset Interrupt ID	Priority
0001	No	No	n/a	n/a
0110	Receive line status interrupt	・Overrun error ・Parity error ・Framing error ・Break received	・Read out the line status register.	1 (Highest)
0100	Receive data ready interrupt	・Receive data ready	・Read out the line status register.	2
	Receive data trigger level reached interrupt	・In the receive FIFO, data has been received until reaching the trigger level.	・Reduce the amount of the data In the receive FIFO below the trigger level.	
1100	Character timeout interrupt	・When the receive FIFO has one or more characters of data, data is not read from the FIFO or is not input to the FIFO for the period equivalent to four characters.	・Read out the receive buffer register.	2
0010	Transmit holding register empty interrupt	・The transmit holding register is empty	・Read out the interrupt identify register. ・Write data to the transmit holding register.	3
	Transmit data trigger level reached interrupt	・Data in the transmit FIFO has been transmitted until reaching the trigger level.	・Read out the interrupt identify register. ・Write data to the transmit FIFO so that there is more data than the trigger level.	
0000	Modem status interrupt	・CTS# input. DSR input, RI input, or DCD input is changed.	・Read out the modem status register.	4 (Lowest)

FIFO Control Resister (FCR)

UART1/2/3[0x08] Default value = — 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 [7:6]: **RCVRT[1:0] Receive Data Trigger Level Setting**

Specifies the maximum number of bytes received in the receive FIFO above which a receive data trigger level reached interrupt should occur if the FIFO is enabled (FIFO enable (Bit 0)= 1).

00 (w): 1 byte
01 (w): 4 bytes
10 (w): 8 bytes
11 (w): 14 bytes

When the number of data bytes in the receive FIFO exceeds the value, a receive data trigger level reached interrupt occurs. The receive data interrupt cause is cleared when receive data is read out from the receive buffer register and the number of data bytes in the FIFO drops below the value.

Bits [5:4]: **XMITT[1:0] Transmit Data Trigger Level Setting**

Specifies the minimum number of data bytes left in the transmit FIFO below which a transmit data trigger level reached interrupt should occur if the programmable transmit holding register empty interrupt is enabled (UART1/2/3[0x04] Bit 7=1) and the FIFO is enabled (FIFO enable (Bit 0)= 1). This bit function does not exist for 16550. Do not use this bit to create a firmware compatible with 16550.

00 (w): 0 byte
01 (w): 2 bytes
10 (w): 4 bytes
11 (w): 8 bytes

When the number of data bytes in the transmit FIFO drops to the value, a transmit data trigger level reached interrupt occurs. The transmit data interrupt cause is cleared when transmit data is written to the transmit buffer register and the number of data bytes in the FIFO exceeds the value.

Note: This bit does not exist in 16550. (not compatible with 16550.)

Bit 3: **DMAMS DMA Mode Selection Bit**

Sets an operation mode of the status signal (internal signal) for the DMA controller.

0 (w): Single word mode
1 (w): Multi-word mode

Bit 2: **XMITFR Transmit FIFO Reset**

Clears data in the transmit FIFO if the FIFO is enabled (FIFO enable (Bit 0)= 1). The shift register is not reset. A "1" written to this bit is automatically reset.

0 (w): Disabled
1 (w): Clear

Note: If data is left in the FIFO, up to two characters are sent before the data is cleared.

Bit 1: **RCVRFR Receive FIFO Reset**

Clears data in the receive FIFO if the FIFO is enabled (FIFO enable (Bit 0)= 1). The shift register is not reset. A "1" written to this bit is automatically reset.

0 (w): Disabled
1 (w): Clear

Note: Depends on the receive FIFO reset, the data ready bit, which indicates that there is receive data, is not cleared. After you clear the receive FIFO, be sure to clear the data ready bit by reading out the receive buffer register (UART1/2/3[0x00]).

Bit 0: **EFIFO FIFO Enable**

Sets the transmit and receive FIFOs to enabled or disabled.

0 (w): Disabled. The FIFOs will not be used for transmission and reception.

1 (w): Enabled. Transmission and reception is done via the FIFOs.

Line Control Register (LCR)

UART1/2/3[0x0C] Default value = 0x00 Read/Write

DLAB	Break Control (SBRK)	Reserved (0)	Even Parity (EPS)	Parity Enable (PEN)	No. of Stop Bits (STB)	Word Length (WLS[1:0])	
7	6	5	4	3	2	1	0

Bit 7: **DLAB Divisor Latch Access Bit**

Specifies whether the divisor latch LSB/MSB register should be accessed or the receive buffer register, transmit holding register, and interrupt enable register should be accessed from UART1/2/3[0x00] and UART1/2/3[0x04].

0 (r/w): Accesses to the receive buffer register, transmit holding register, and interrupt enable register.

1 (r/w): Accesses to the divisor latch LSB/MSB register.

Bit 6: **SBRK Break Control**

Controls the output of the break signal.

0 (r/w): Normal output.

1 (r/w): Break signal output. When this bit is set to "1" (before "0" is written to it), the serial output is LOW.

Bit 4: **EPS Even Parity**

Selects even/odd parity.

0 (r/w): Odd parity

1 (r/w): Even parity

This setting is effective only if the parity enable (Bit 3) is "1".

Bit 3: **PEN Parity Enable**

Sets the parity check and parity bit attachment to enabled/disabled.

0 (r/w): Parity disabled. Receive data is treated as data without parity. No parity bit is attached to transmit data.

1 (r/w): Parity enabled. Data is received as data with a parity bit attached and parity check is performed. A parity bit is attached to transmit data.

Bit 2: **STB No. of Stop Bits**

Selects the number of stop bits.

0 (r/w): 1 bit

1 (r/w): 2 bits (if the data length is 6, 7, or 8 bits)
1.5 bits (if the data length is 5 bits)

The specified stop bits are attached to the transmit data. When receiving data, only the first 1 bit is checked regardless the specified number of stop bits.

Bits [1:0]: **WLS[1:0] Word Length Bits [1:0]**

Specifies the number of bits for transmit/receive characters (excluding parity and stop bits).

00 (r/w): 5 bits

01 (r/w): 6 bits

10 (r/w): 7 bits

11 (r/w): 8 bits

Modem Control Register (MCR)

UART1/2/3[0x10] Default value = 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**

Enables auto flow control that uses CTS# and RTS# signals from the modem.

0 (r/w): Manual flow control mode

1 (r/w): Auto flow control mode

Setting this bit to "1" and Bit 0 of the FIFO control register to "1" enables auto CTS control mode, and automatically aborts data transmission when the CTS# input becomes HIGH while there is data in the transmit FIFO. When the CTS# input becomes LOW, data transmission is automatically resumed. In addition, setting Bit 1 of the modem control register to "1" enables auto RTS control mode, and changes the RTS# output to HIGH when the receive FIFO reaches the receive data trigger level. When the receive FIFO becomes empty, the RTS# output is automatically changed to LOW.

Note: This bit does not exist in 16550. (not compatible with 16550.)

Bit 4: **LOOP Loop Back**

Runs a local loop test on the modem.

0 (r/w): Normal operation

1 (r/w): Local loop test

When this bit is set to "1", the serial output line is linked to the serial input line and the test can be run by inputting transmit data.

The table below shows differences between loop back mode and normal mode.

As shown the table, you can run simple stand-alone self-diagnoses in loop back mode, since the output system for serial data and modem control feeds back to the input system. Even in this mode, the interrupt and break functions continue their operation.

Table 16.3 Differences between Loop Back Mode and Normal Mode

Difference Location		Loop Back Mode	Normal Mode
DTR# pin		Always outputs high level (inactive)	Outputs an inverted value of the DTR bit
RTS# pin		Always outputs high level (inactive)	Outputs an inverted value of the RTS bit
TXD pin		Always outputs high level (mark state)	Sequentially outputs serial data from the transmit shift register (TSR)
Modem Status Register	CTS bit	The setting of the RTS bit is read.	The inverted value of the CTS# pin is read.
	DSR bit	The setting of the DTR bit is read.	The inverted value of the DSR# pin is read.
	RI bit	The setting of the OUT1 bit is read.	The inverted value of the RI# pin is read.
	DCD bit	The setting of the OUT2 bit is read.	The inverted value of the DCD# pin is read.
	DCTS bit	Captures changes of the RTS bit.	Captures changes of the CTS# pin.
	DDSR bit	Captures changes of the DTR bit.	Captures changes of the DSR# pin.
	TERI bit	Captures falling edge changes of the OUT1 bit.	Captures changes of the RI# pin.
	DDCD bit	Captures changes of the OUT1 bit.	Captures changes of the DCD# pin.
Receive shift register		Captures serial data from the transmit shift register (TSR).	Captures serial data from the RXD pin.

Bit 3: **OUT2 Output 2# Control**

Directly controls OUT2# output. However, in loop back test mode, it is always high level output and connects to internal signal equivalent to DCD#.

0 (r/w): OUT2# = HIGH

1 (r/w): OUT2# = LOW

Note: This bit is available only in loop back mode.

Bit 2: **OUT1 Output 1# Control**

Directly controls OUT1# output. However, in loop back test mode, it is always high level output and connects to internal signal equivalent to RI#.

0 (r/w): OUT1# = HIGH

1 (r/w): OUT1# = LOW

Note: This bit is available only in loop back mode.

Bit 1: **RTS RTS Control**

Directly controls RTS# output if manual flow mode is enabled (modem control register Bit 5 is set to 0).

0 (r/w): RTS# = HIGH

1 (r/w): RTS# = LOW

Becomes auto RTS control mode if auto flow mode is enabled (modem control register Bit 5 is set to 1).

0 (r/w): No control for RTS

1 (r/w): Auto RTS control mode

However, in loop back test mode, it is always high level output and connects to internal signal equivalent to CTS#.

Bit 0: **DTR DTR# Control**

Directly controls DTR# output. However, in loop back test mode, it is always high level output and connects to internal signal equivalent to DSR#.

0 (r/w): DTR# = HIGH

1 (r/w): DTR# = LOW

Note: This bit is available only in loop back mode.

Line Status Register (LSR)

UART1/2/3[0x14] Default value = 0x60 Read Only

Receive FIFO Error	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**

Indicates if there is a receive FIFO error (parity error, framing error, or break indicator). This bit is effective only if the FIFO is enabled (the FIFO control register UART1/2/3[08h] Bit 0 (FIFO enable bit)= 1); if the FIFO is disabled (the FIFO enable bit = 0), this bit is always set to 0.

0: No error

1: Error occurred

If there is no data in the FIFO other than the error data to be read next from the receive buffer, reading this register resets this flag to “0”.

Bit 6: **TEMT Transmit Empty**

Indicates that the transmit shift register and the FIFO/transmit holding register are empty.

0: Transmit data exists

1: No transmit data exists

Indicates that both the transmit FIFO register and the transmit shift register are empty, when the FIFO is enabled (FIFO enable bit = 1). Indicates that both the transmit holding register and the transmit shift register are empty, when the FIFO is disabled (FIFO enable bit = 0). This bit returns to 0 when transmit data is written.

Bit 5: **THRE Transmit Holding Register Empty**

The behavior of this bit varies depending on the setting of the interrupt enable register UART1/2/3[0x04] Bit 7 (EPTBEI bit).

When programmable transmit holding register empty interrupt is disabled (EPTBEI bit = 0), this bit indicates that the transmit FIFO/holding registers are empty.

0: Transmit data exists
1: No transmit data exists

Indicates that the transmit FIFO register is empty, when the FIFO is enabled (FIFO enable bit = 1). Indicates that the transmit holding register is empty, when the FIFO is disabled (FIFO enable bit = 0). When this flag is set, an interrupt request is generated if the transmit holding register empty interrupt is enabled. Note that this bit is set when the transmit FIFO/holding registers become empty as a result of transmission. It is not set when the registers become empty as a result of another operation such as reset and FIFO clear. This bit returns to 0 when transmit data is written.

When programmable transmit holding register empty interrupt is enabled (EPTBEI bit = 1), this bit indicates that the transmit FIFO is full.

0 (r) : Data can be written to the transmit FIFO
1 (r) : The transmit FIFO is full

By reading this bit before writing data to the transmit FIFO when a programmable transmit holding register empty interrupt occurs, transmit data can be written to the transmit FIFO until it becomes full. Using this bit and programmable transmit holding register empty interrupt ensures that the transmit FIFO always has data, and efficient data transfer can be carried out even in a system that cannot respond to an interrupt immediately.

Bit 4: **BI Break Interrupt Flag**

Indicates a break interrupt.

0: No break interrupt exists
1: A break interrupt exists

This flag is set if the input line is 0 during a period equivalent to one character.
It indicates that the error is occurring at the first character in the FIFO, when the FIFO is enabled (FIFO enable bit = 1). If receive line status interrupt is enabled, an interrupt request is generated when this flag is set. This flag resets to “0” when the register is read out.

Bit 3: **FE Framing Error**

Indicates that a framing error occurs.

0: No error exists
1: An error exists

This flag is set if the receive data contains no valid stop bit. It indicates that the error is occurring at the first character in the FIFO, when the FIFO is enabled (FIFO enable bit = 1). If receive line status interrupt is enabled, an interrupt request is generated when this flag is set. This flag resets to “0” when the register is read out.

Bit 2: **PE Parity Error**

Indicates that a parity error occurs.

0: No error exists
1: An error exists

This flag is set when a parity error is detected in the parity check that is performed when the parity enable bit is set . It indicates that the error is occurring at the first character in the FIFO, when the FIFO is enabled (FIFO enable bit = 1). If receive line status interrupt is enabled, an interrupt request is generated when this flag is set. This flag resets to “0” when the register is read out.

Bit 1: **OE Overrun Error**

Indicates that an overrun error occurs.

0: No error exists
1: An error exists

An overrun error occurs when the next character is received before the current character is read from the receive buffer register, if the FIFO is disabled (FIFO enable bit = 0). An overrun error also occurs when a new character is received while the FIFO is full, if the FIFO is enabled (FIFO enable bit = 1). If receive line status interrupt is enabled, an interrupt request is generated when this flag is set. This flag resets to "0" when the register is read out.

Bit 0: **DR Data Ready**

Indicates that there is receive data.

0: No receive data exists
1: Receive data exists

This bit is set to "1" if there is data of one or more valid characters in the receive buffer register or the FIFO. If receive data ready interrupt is enabled, an interrupt request is generated when this flag is set. This flag resets to "0" when all receive data is read from the receive FIFO.

Modem Status Register (MSR)

UART1/2/3[0x18] Default value = 0xEX Read Only

DCD Status (DCD) 7	RI Status (RI) 6	DSR Status (DSR) 5	CTS Status (CTS) 4	DCD Change (DDCD) 3	RI Falling Edge Change (TERI) 2	DSR Change (DDSR) 1	CTS Change (DCTS) 0

Bit 7: **DCD DCD Status**

Indicates the input state of the DCD pin.

0: DCD input = HIGH
1: DCD input = LOW

Note: This bit is available only in loop back mode.

Bit 6: **RI RI Status**

Indicates the input state of the RI pin.

0: RI input = HIGH
1: RI input = LOW

Note: This bit is available only in loop back mode.

Bit 5: **DSR DSR Status**

Indicates the input state of the DSR pin.

0: DSR input = HIGH
1: DSR input = LOW

Note: This bit is available only in loop back mode.

Bit 4: **CTS CTS Status**

Indicates the input state of the CTS# pin.

0: CTS1# input = HIGH
1: CTS#1 input = LOW

Bit 3: **DDCD DCD Change**

Indicates if there has been any change in the input of the DCD pin since the previous read operation. Note that, however, the change in the output 2 control bit (UART1/2/3[0x10] Bit 3) is reflected regardless the setting of the DCD pin if loop back mode is enabled (UART1/2/3[0x10] Bit 4 = 1).

0: No change exists
1: Change exists

This bit is cleared when the register is read out.

Note: This bit is available only in loop back mode.

Bit 2: **TERI RI Falling Edge Change**

Indicates if there has been any falling edge change in the input of the RI pin since the previous read operation. Note that, however, the falling edge change in the output 1 control bit (UART1/1/3[0x10] Bit 2) is reflected regardless the setting of the RI pin if loop back mode is enabled (UART1/2/2[0x10] Bit 4 = 1).

0: No falling edge change exists
1: Falling edge change exists

This bit is cleared when the register is read out.

Note: This bit is available only in loop back mode.

Bit 1: **DDSR DSR Change**

Indicates if there has been any change in the input of the DSR pin since the previous read operation. Note that, however, the change in the DTR bit (UART1/2/3[0x10] Bit 0) is reflected regardless the setting of the DSR pin if loop back mode is enabled (UART1/2/0[0x10] Bit 4 = 1).

0: No change exists
1: Change exists

This bit is cleared when the register is read out.

Note: This bit is available only in loop back mode.

Bit 0: **DCTS CTS Change**

Indicates if there has been any change in the input of the CTS# pin since the previous read operation. Note that, however, the change in the RTS control bit (UART1/2/3[0x10] Bit 1) is reflected regardless the setting of the CTS# pin if loop back mode is enabled (UART1/2/1[0x10] Bit 4 = 1).

0: No change exists
1: Change exists

This bit is cleared when the register is read out.

Scratch Register (SCR)

UART1/2/3[0x1C] Default value = 0x00 Read/Write

Scratch Bits (SCR [7:0])							
7	6	5	4	3	2	1	0

Bits [7:0]: **SCR [7:0] Scratch Bits**

Can be used as a general purpose register that does not affect the hardware operations.

Test 0 Register (T0)

UART1/2/3[0x20] Default value = 0x00 Read/Write

n/a							Test Mode
7	6	5	4	3	2	1	0

Bit 0: **Test Mode**

Select the test mode. In test mode, a test equivalent to a loop back test can be carried out.

0 (r/w): Normal mode
1 (r/w): Test mode

Test 1 Register (T1)
UART1/2/3[0x24] Default value = 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**
Controls the DCD input in test mode (UART1/2/3[0x20] Bit 0=1).
0 (r/w): Inputs low level (active)
1 (r/w): Inputs high level (inactive).

Bit 2: **RI Test**
Controls the RI input in test mode (UART1/2/3[0x20] Bit 0=1).
0 (r/w): Inputs low level (active)
1 (r/w): Inputs high level (inactive).

Bit 1: **DSR Test**
Controls the DSR input in test mode (UART1/2/3[0x20] Bit 0=1).
0 (r/w): Inputs low level (active)
1 (r/w): Inputs high level (inactive).

Bit 0: **CTS Test**
Controls the CTS# input in test mode (UART1/2/3[0x20] Bit 0=1).
0 (r/w): Inputs low level (active)
1 (r/w): Inputs high level (inactive).

Test Status 0 Register (TS0)
UART1/2/3[0x28] Default value = — 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**
Always indicates the input state of the DCD pin.
0: Low level (active) is input
1: High level (inactive) is input

Bit 2: **RI Raw Status**
Always indicates the input state of the RI pin.
0: Low level (active) is input
1: High level (inactive) is input

Bit 1: **DSR Raw Status**
Always indicates the input state of the DSR pin.
0: Low level (active) is input
1: High level (inactive) is input

Bit 0: **CTS Raw Status**
Always indicates the input state of the CTS pin.
0: Low level (active) is input
1: High level (inactive) is input

Test Status 1 Register (TS1)

UART1/2/3[0x2C] Default value = 0x01 Read Only

n/a				DCD Status	RI Status	DSR Status	CTS Status
7	6	5	4	3	2	1	0

Bit 3: **DCD Status**

Indicates the input state of the DCD signal that the UART1/2/3 circuits recognize.

0: Low level (active) is input
1: High level (inactive) is input

Bit 2: **RI Status**

Indicates the input state of the RI signal that the UART1/2/3 circuits recognize.

0: Low level (active) is input
1: High level (inactive) is input

Bit 1: **DSR Status**

Indicates the input state of the DSR signal that the UART1/2/3 circuits recognize.

0: Low level (active) is input
1: High level (inactive) is input

Bit 0: **CTS Status**

Indicates the input state of the CTS# signal that the UART1/2/3 circuits recognize.

0: Low level (active) is input
1: High level (inactive) is input

Test Status 2 Register (TS2)

UART1/2/3[0x30] Default value = 0x0F 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**

Indicates the state of the BAUDOUT signal that the UART1/2/3 circuits output.

0: Outputs low level
1: Outputs high level

Bit 3: **OUT2 Status**

Indicates the state of the OUT2 signal that the UART1/2/3 circuits output.

0: Outputs low level
1: Outputs high level

Bit 2: **OUT1 Status**

Indicates the state of the OUT1 signal that the UART1/2/3 circuits output.

0: Outputs low level
1: Outputs high level

Bit 1: **RTS Status**

Indicates the state of the RTS# signal that the UART1/2/3 circuits output.

0: Outputs low level (active)
1: Outputs high level (inactive)

Bit 0: **DTR Status**

Indicates the state of the DTR signal that the UART1/2/3 circuits output.

0: Outputs low level (active)
1: Outputs high level (inactive)

Test Status 3 Register (TS3)

UART1/2/3[0x3C] Default value = 0x02 Read Only

7	6	5	4	3	2	1	0
n/a					TXRDY Status	RXRDY Status	INTR Status

Bit 2: **TXRDY Status**

Indicates the state of the TXRDY signal that the UART1/2/3 circuits output.

0: Outputs low level (active)
1: Outputs high level (inactive)

Bit 1: **RXRDY Status**

Indicates the state of the RXRDY signal that the UART1/2/3 circuits output.

0: Outputs low level (active)
1: Outputs high level (inactive)

Bit 0: **INTR Status**

Indicates the state of the INTR signal that the UART1/2/3 circuits output.

0: Outputs low level (inactive)
1: Outputs high level (active)

16.4.4 Baud Rate Setup Example

The clock division value to set up the baud rate in UART1/2/3 is obtained from the following expression.

Division value = UART1/2/3_SCLK input clock frequency (Hz) ÷ baud rate (bps) ÷ 16

For example, the clock division values when input clock to UART1/2/3 (=UART1/2/3_SCLK) is at 24.543232 MHz are as follows.

Table 16.4 Baud Rate and Division Value

Baud Rate	×16 Ideal Clock Value	24.543232MHz UART1/2/3 Source Clock		
		×16 Clock Division Value	Error Percent %	Actual ×16 Clock
300	4800	5113	0.00	4800.2
600	9600	2557	0.02	9598.4
1200	19200	1278	0.02	19204.4
2400	38400	639	0.02	38408.8
4800	76800	320	0.13	76697.6
9600	153600	160	0.13	153395.2
14400	230400	107	0.44	229376.0
19200	307200	80	0.13	306790.4
28800	460800	53	0.49	463079.8
38400	614400	40	0.13	613580.8
57600	921600	27	1.37	909008.6
115200	1843200	13	2.43	1887940.9
125000	2000000	12	2.26	2045269.3
250000	4000000	6	2.26	4090538.7
500000	8000000	3	2.26	8181077.3
750000	12000000	2	2.26	12271616.0
1500000	24000000	1	2.26	24002560.0

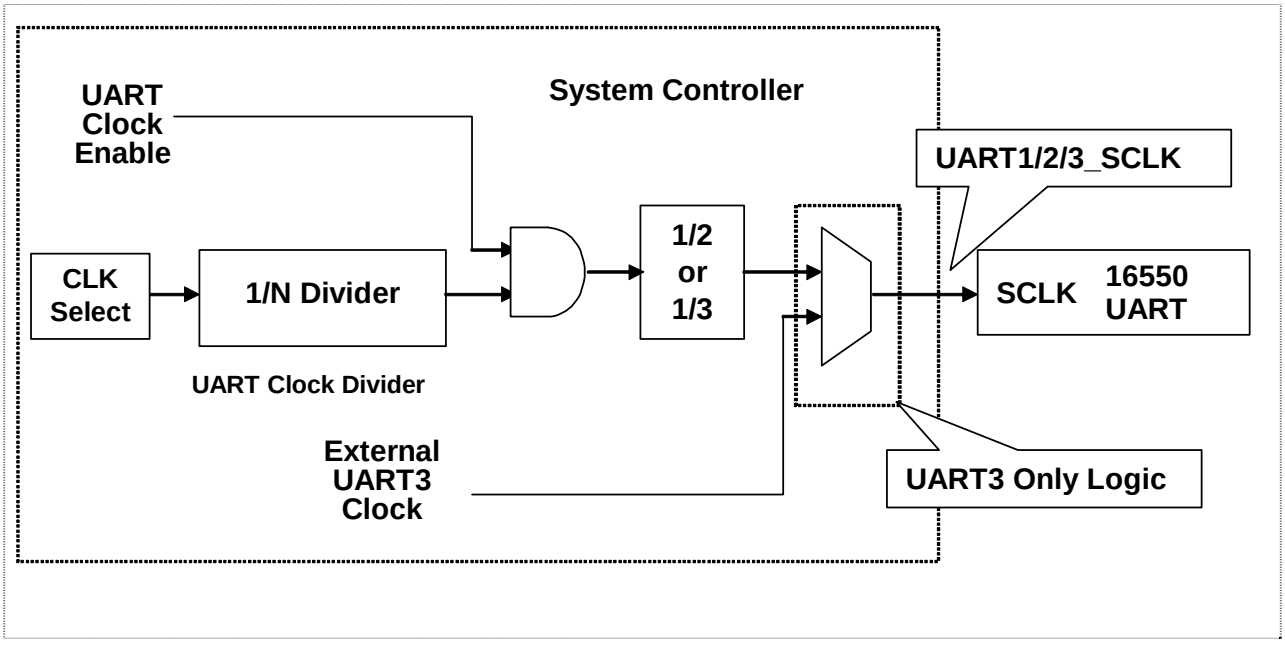


Figure 16.2 Conceptual Diagram of UART1/2/3 Clock

16.5 Usage Restrictions for This UART1/2/3

Although the chip circuits themselves have most functions for 16550, restrictions are applied to some registers because this chip does not provide all signals as I/O pins that are compatible with 16550. This chip applies restrictions to the following registers.

Offset Address	Register Bit Name	Restrictions
UART1/2/3[0x10]Bit 0	DTR: DTR# Control	Only loop back mode available
UART1/2/3[0x10]Bit 2	OUT1: Output 1# Control	Only loop back mode available
UART1/2/3[0x10]Bit 3	OUT2: Output 2# Control	Only loop back mode available
UART1/2/3[0x18]Bit 1	DDSR: DSR Change	Only loop back mode available
UART1/2/3[0x18]Bit 2	TERI: RI Falling Edge Change	Only loop back mode available
UART1/2/3[0x18]Bit 3	DDCD: DCD Change	Only loop back mode available
UART1/2/3[0x18]Bit 5	DSR: DSR Status	Only loop back mode available
UART1/2/3[0x18]Bit 6	RI: RI Status	Only loop back mode available
UART1/2/3[0x18]Bit 7	DCD: DCD Status	Only loop back mode available

The following register setting is available for 16550 but not for this UART1/2/3.

Offset Address	Register Bit Name	Restrictions
UART1/2/3[0x0C]Bit 5	Stick parity	Always unavailable

The following register settings are available for this UART1/2/3 but not for 16550: These registers should not be used when you create a firmware compatible with 16550.

Offset Address	Register Bit Name	Restrictions
UART1/2/3[0x04]Bit 7	EPTBEI: Programmable Transmit Holding Register Empty Interrupt Enable	Not compatible with 16550
UART1/2/3[0x08]Bit [5:4]	XMITT[1:0]: Transmit Data Trigger Level Setting	Not compatible with 16550
UART1/2/3[0x10]Bit 5	AFCE: Auto Flow Control Enable	Not compatible with 16550

17. I²C SINGLE MASTER CORE MODULE (I²C)

17.1 Description

17.1.1 Master Mode

- Supports I²C bus I²C single master mode.
- Does not support I²C bus I²C multi-master mode.
- Multiple slave devices can be connected to the I²C bus.
- In addition to the shift register involved in I²C data transfer, this module has transmit and receive buffers (TBUF and RBUF) to ease the timing of software read/write.
- Detects bus error statuses and can report them through the status register.
- Supports the I²C clock wait function.

17.1.2 Slave Mode

- Does not support slave mode.

17.2 Block Diagram

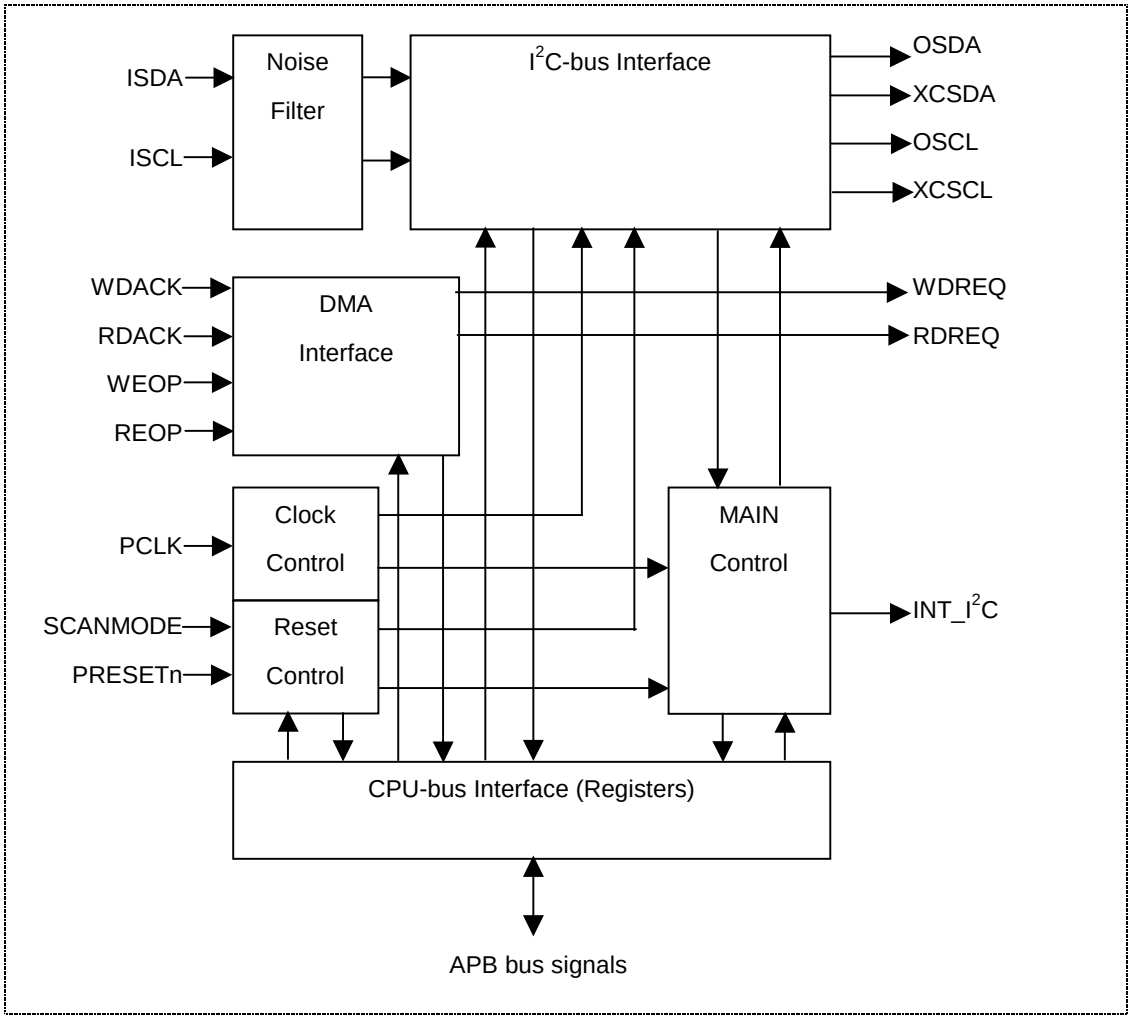


Figure 17.1 I^2C Internal Block Diagram

17.3 External Pins

Pin Name	Input/Output	Pin Functions	Multiplex Pin*/Remarks
SCL	Input and output	I^2C clock input and output	GPIOA6
SDA	Input and output	I^2C data input and output	GPIOA7

Note (*): The external pins for I^2C are multiplexed with GPIO pins. You can use the functions for I^2C by setting "Function 2 of other than GPIO" through the GPIO pin function register.

17.4 Registers

17.4.1 List of Registers

The address offset values in the table below are offsets from the base address that is assigned to the I2C logic module. The base address of these registers is 0xFFFE_D000.

Table 17.1 List of Registers (Base Address: 0xFFFE_D000)

Address Offset	Register Name	Default Value	R/W	Access Size *1
0x00	I^2C transmit data register	0000 0000b	R/W	8 (16/32)
0x04	I^2C receiving data register	0000 0000b	RO	8 (16/32)
0x08	I^2C control register	0000 0000b	R/W	8 (16/32)
0x0C	I^2C bus status register	00xx 0000b *2	RO	8 (16/32)
0x10	I^2C error status register	0000 0000b	RO	8 (16/32)
0x14	I^2C interrupt control/status register	0000 0000b	R/W	8 (16/32)
0x18	I^2C–BUS sample clock dividing setting register	0000 0000b	R/W	8 (16/32)
0x1C	I^2C SCL clock dividing setting register	0000 0000b	R/W	8 (16/32)
0x20	I^2C I/O control register	0000 0000b	R/W	8 (16/32)
0x24	I^2C DMA mode register	0000 0000b	R/W	8 (16/32)
0x28	I^2C DMA counter value (LSB) register	0000 0000b	R/W	8 (16/32)
0x2C	I^2C DMA counter value (MSB) register	0000 0000b	R/W	8 (16/32)
0x30	I^2C DMA status register	0000 1000b	RO	8 (16/32)
0x34 – 0x38	Reserved	—	—	—

*1: The registers in the List of Registers are defined with 8 bits and firmware usually uses these registers in 8-bit access. Although 16-bit or 32-bit access is possible if the firmware uses 16-bit or 32-bit access instruction, only the lower 8 bits can be used as valid values.

*2: Bit 5 (SDA) and Bit 4 (SCL) of this register monitor the external pins: SDA pin and SCL pin. The status of these pins determine the default value. Because these pins are generally pulled up externally, the settings are SDA=1 and SCL=1, which correspond to the settings of the external pins.

17.4.2 Detailed Description of Registers

The following explains the details of each register:

I²C transmit data register

I²C[0x00] Default value = 0000 0000b — Read/write

I²C Transmit Data TD [7:0]							
7	6	5	4	3	2	1	0

Bits [7:0]: **TD[7:0] I²C Transmit Data**

This register is an 8-bit buffer that stores transmit data of I²C-BUS transfer.
The MSB bit through the LSB bit are transmitted in sequence when data is transmitted from the master. (See the diagram below.)

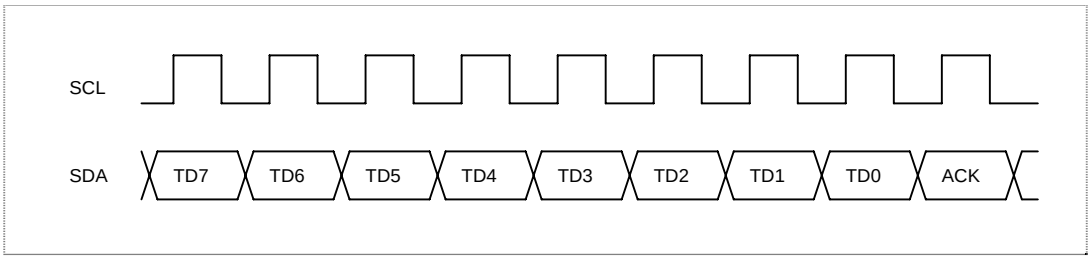


Figure 17.2 Bit Sequence When transmitting

I²C receiving data register

I²C[0x04] Default value = 0000 0000b — Read Only

I²C Receive Data RD [7:0]*							
7	6	5	4	3	2	1	0

Bits [7:0]: **RD[7:0] I²C Receive Data**

This register is an 8-bit buffer that stores receive data of I²C-BUS transfer.
Transmitted data from a slave device, the MSB bit through the LSB bit, is received in sequence. (See the diagram below.)

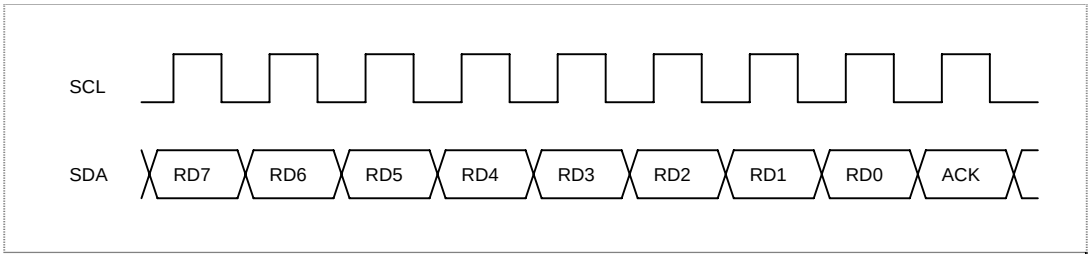


Figure 17.3 Bit Sequence When Receiving

I²C control register

I²C[0x08] Default value = 0000 0000b Read/write

Reserved (0)		SR	CLKW*	TACK*	TRNS [2:0] *		
7	6	5	4	3	2	1	0

Note: (*) can also be initialized by software reset.

This register controls occurrence of start and stop conditions, data transmission/reception start, and others.
When TRNS[2:0] are written, the specified operation is performed.
If TRNS[2:0] are written when the I²C bus status register I²C[0x0C] bit 7 is set to RUN =1, the command is not executed.

Bits [7:6]: **Reserved (0)**

Bit 5: **SR Software Reset**

Forces reset by the software to perform an initialization.

0: Disables software reset

1: Enables software reset

Bit 4: **CLKW Clock wait Mode Enable**

Selects whether to use the clock wait function.

0: Clock wait mode disable

1: Clock wait mode enable

Bit3: **TACK Data Reception Acknowledgement Setting (only for receive [Read] mode)**

Sets an acknowledgement to send to the slave device at data reception.

0: Does not send an acknowledgement to the slave device.

1: Sends an acknowledgement to the slave device.

Note: In DMA transfer mode, the value of this bit (TACK) is used only when the last byte is transferred; “0” is sent in all other byte transfer.

Bits [2:0]: **TRNS[2:0] Transmit Control Command**

Specifies the operation and instructs about the start.

001: I²C start condition

010: I²C stop condition

011: I²C data receive

100: I²C data transfer

101, 110: Unavailable

000, 111: Clears the error flag of the error status register.

I²C bus status register							
I²C[0x0C] Default value = 00xx 0000b							Read Only
RUN*	Reserved	SDA	SCL	Using*	Busy*	Error*	Finish*
7	6	5	4	3	2	1	0

Note: (*) can also be initialized by software reset.
x: unknown (Irregularity)

This register represents the status of the I²C-BUS.

Bit 7: **RUN Command Execution Status**

Indicates if there is a command operation.

0: I²C-BUS (command) wait state
1: I²C-BUS (command) execution is in progress

Bit 6: **Reserved**

Bit [5:4]: **SDA, SCL I²C-BUS Monitor**

Indicates the current status of the SDA and SCL. Because SDA and SCL are usually pulled up externally, these bits are set to HIGH and indicate values accordingly.

Bit 3: **Using I²C-BUS Use**

Indicates if the I²C single master uses the I²C-BUS.

0: The I²C single master is not using the I²C-BUS.
1: The I²C single master is using the I²C-BUS.

Bit 2: **Busy I²C-BUS Busy**

Indicates the operating state of the I²C-BUS.

0: I²C-BUS BusFree state
1: I²C-BUS is in use

Bit 1: **Error Error occurrence**

Indicates if an error has occurred.

0: No error exists
1: Error state has occurred
This bit can be cleared by writing operation to the I²C control register.

Bit 0: **Finish Command Execution Finished**

Indicates that the command finished.

0: Command is not executed or currently in progress
1: Command execution finished
This bit is cleared when reset is enabled (RESET# = LOW), when the I²C control register bit 5 (software reset) is set to 1, or by writing a valid command to the I²C control register bits[2:0] (TRNS).

I^2C error status register

I2C[0x10] Default value = 0000 0000b Read Only

Reserved			Reception Acknowledgement *	SCL Mismatch *	SDA Mismatch *	Stop Condition *	Start Condition *
7	6	5	4	3	2	1	0

Note: (*): Can also be initialized by software reset.

This register represents error states.

Bits [7:5]: **Reserved**

Bit 4: **Reception Acknowledgement Error**

Indicates if there is an error in a reception acknowledgement.

0: No error exists
1: An error exists

Bit 3: **SCL Mismatch Error**

Indicates if there is an error due to a mismatch between ISCL and OSCL.

0: Normal operation
1: Clock mismatch is detected.

Bit 2: **SDA Mismatch Error**

Indicates if there is an error due to a mismatch between ISDA and OSDA.

0: Normal operation
1: Data mismatch is detected.

Bit 1: **Stop Condition Detection**

Indicates if there is an error due to an occurrence of a stop condition.

0: Normal operation
1: A stop condition other than one caused by a command is detected.

Bit 0: **Start Condition Detection**

Indicates if there is an error due to an occurrence of a start condition.

0: Normal operation
1: A start condition other than one caused by a command is detected.

Note: These error statuses are cleared when reset is enabled (RESET# = LOW), when the I^2C control register bit 5 (software reset) is set to 1, or by writing a valid command to the I^2C control register bits[2:0] (TRNS).

I²C interrupt control/status register

I²C[0x14] Default value = 0000 0000b Read/Write

Reserved (0)				Error INT Status Flag *	Completion INT Status Flag *	Error INT Enable *	Completion INT Enable *
7	6	5	4	3	2	1	0

Note: (*): Can also be initialized by software reset.

Bits [7:4]: **Reserved (0)**

Bit 3: **Status Flag for Interrupt Caused by Error Occurrence**

This bit is effective only when Bit 1 is set to "1".

0: No error has occurred

1: An interrupt request caused by error occurrence has been generated
This flag is cleared by writing "1" to this bit.

Bit 2: **Command Completion Interrupt Status Flag**

This bit is effective only when Bit 0 is set to "1".

0: The command completion interrupt enable bit (Bit 0) is set to 0, or the command has not been completed.

1: An interrupt request caused by command completion has been generated
This flag is cleared by writing "1" to this bit.

Bit 1: **Error Occurrence Interrupt Enable**

Controls interrupts caused by error occurrence.

0: Disables Interrupt generation caused by error occurrence

1: Enables interrupt generation caused by error occurrence

Bit 0: **Command Completion Interrupt Enable**

Controls interrupt generation caused by command completion.

0: Disables interrupt generation caused by command completion

1: Enables interrupt generation caused by command completion

I²C-BUS sample clock dividing setting register

I²C[0x18] Default value = 0000 0000b Read/Write

Reserved (0)				Dividing Ratio (I²C Sample) [3:0]			
7	6	5	4	3	2	1	0

Generates an I²C-BUS sample clock by dividing the master clock.

Bits [7:4]: **Reserved bit (0)**

Bits [3:0]: **Dividing Ratio (I²C Sample) [3:0]**

Sets the dividing ratio to generate an I²C-BUS sample clock from the master clock.

The frequency of the I²C-BUS sample clock ($f_{I2C\ sample}$) is obtained from the following expression.

Bits [3:0] > 0:

$f_{I2C\ sample} = f_{PCLK}/(4*m)$[Hz]

Bits [3:0] = 0:

$f_{I2C\ sample} = f_{PCLK}/2$[Hz]

Note: In the expression above, m = dividing ratio (I²C sample) [3:0].

I²C SCL clock dividing setting register

I²C[0x1C] Default value = 0000 0000b Read/Write

Reserved (0)					Dividing Ratio (SCL) [2:0]		
7	6	5	4	3	2	1	0

This register is for setting the dividing ratio to generate SCL.

Bits [7:3]: **Reserved (0)**

Bit [2:0]: **Dividing Ratio (SCL) [2:0]**

Sets the dividing ratio to generate SCL from an I²C-BUS sample clock.

The frequency of the SCL for I²C-BUS transfer is obtained from the following expression.

Dividing Ratio (I²C Sample)[3:0] > 0:

$f_{SCL} = f_{I2C\ sample}/(2n*4)$

$= f_{PCLK}/\{(4*m) * (2n*4)\}$

$= f_{PCLK}/(16*m*2n)$[Hz]

Dividing ratio (I²C sample)[3:0] = 0:

$f_{SCL} = f_{PCLK}/\{2*(2n*4)\}$

$= f_{PCLK}/(8*2n)$ [Hz]

Note: In the expression above, “m” and “n” are specified as follows:

m = dividing ratio (I²C sample)[3:0]

(See I²C-bus sample clock dividing setting register Bits[3:0].)

n = dividing ratio (SCL) [2:0]

I²C I/O control register

I²C[0x20] Default value = 0000 0000b Read/Write

Reserved (0)		High Drive SDA	Sample SDA	Reserved (0)		High Drive SCL	Sample SCL
7	6	5	4	3	2	1	0

This register is for selecting output mode and determining whether the noise filter should be enabled or disabled.

Bits [7:6]: **Reserved (0)**

Bit 5: **SDA HIGH Drive Enable**

0: “SDA=1” is controlled by the pull-up resistor external to the IC.

1: “SDA=1” is controlled by this IC itself by driving “High”.

Bit 4: **SDA Sampling Enable**

0: As SDA input, one data sample is sampled with the I²C-BUS sample clock.

1: As SDA input, two data samples are sampled with the I²C-BUS sample clock.

Note: Use this bit with the value of “0”.

Bits [3:2]: **Reserved (0)**

Bit 1: **SCL HIGH Drive Enable**

0: “SCL=1” is controlled by the pull-up resistor external to the IC.

1: “SCL=1” is controlled by this IC itself by driving “High”.

Bit 0: **SCL Sampling Enable**

0: As SCL input, one data sample is sampled with the I²C-BUS sample clock.

1: As SCL input, two data samples are sampled with the I²C-BUS sample clock.

Note: Use this bit with the value of "0".

I²C DMA mode register

I²C[0x24] Default value = 0000 0000b Read/Write

Reserved (0)						DMA_MODE [1:0]	
7	6	5	4	3	2	1	0

Bits[7:2]: **Reserved (0)**

Bits [1:0]: **DMA_MODE[1:0] DMA Mode Setting**

00: Do not use DMA transfer.

01: Single address mode

10: Dual address mode (without EOP). Use the DMA counter.

11: Dual address mode (with EOP).

Although DMA modes can be turned ON/OFF during a transfer (from Start(restart) to Stop), do not switch the DMA mode (Single, Dual (with EOP), Dual (without EOP)) that you intend to use.

Burst transfer is not supported.

I²C DMA counter value (LSB) register

I²C[0x28] Default value = 0000 0000b Read/Write

DMA Counter Value (LSB)							
7	6	5	4	3	2	1	0

Write: This register is for setting the lower byte [7:0] of the DMA counter value.
Read: This register returns the lower byte [7:0] of the DMA counter value.

I²C DMA counter value (MSB) register

I²C[0x2C] Default value = 0000 0000b Read/Write

DMA Counter Value (MSB)							
7	6	5	4	3	2	1	0

Write: This register is for setting the higher byte [15:8] of the DMA counter value.
Read: This register returns the higher byte [15:8] of the DMA counter value.

I²C DMA status register

I²C[0x30] Default value = 0000 1000b — Read Only

Reserved				TBUF Empty*	RBUF Update*	RDREQ Monitor*	WDREQ Monitor*
7	6	5	4	3	2	1	0

Note: (*):Can also be initialized by software reset.

Bits[7:4]: **Reserved**

Bit 3: **TBUF_Empty Transmit Buffer Empty**

Indicates whether the write operation to the transmit data buffer (TBUF) of the I²C single master core is possible.

0: The transmit buffer has data to transmit.

1: The transmit buffer is empty. Write operation is possible.

Bit 2: **RBUF_Update Receive Buffer Update**

Indicates whether the data in the I²C receive data buffer (RBUF) is updated.

0: The receive buffer has not been updated.

1: The data in the receive buffer has been updated.

Bit 1: **Reserved (RDREQ Monitor RDREQ Signal Monitor)**

Indicates the state of the RDREQ signal.

0: Indicates that the signal on the RDREQ pin is Low.

1: Indicates that the signal on the RDREQ pin is High.

Note: The state may change but you cannot use this bit.

Bit 0: **Reserved (WDREQ Monitor WDREQ Signal Monitor)**

Indicates the state of the WDREQ signal.

0: Indicates that the signal on the WDREQ pin is Low.

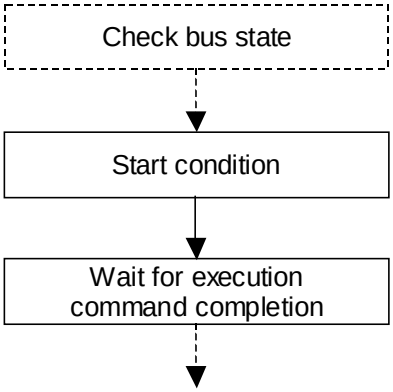
1: Indicates that the signal on the WDREQ pin is Low.

Note: The state may change but you cannot use this bit.

17.5 Explanation of Operation (Example of Use: Bus Control Command)

The following is examples for controlling I^2C bus with this module. In practice, it may be required to check the states or respond to an error. In addition, a specific control method depends on the specifications of the slave device.

17.5.1 Start (S) Flow Example

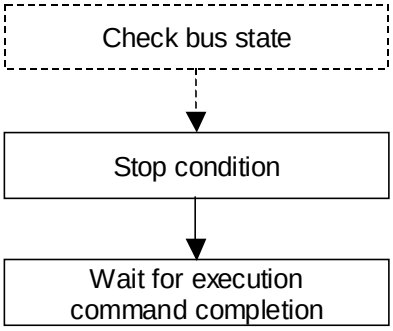


Check that I2C[OxOC] bit 2 is set to “0”

Start condition setting:
Set the I2C[OxO8] bit [2:0] to “001”

Wait for the I2C[OxOC] bit 2 to be set to “1”

17.5.2 Stop (P) Flow Example

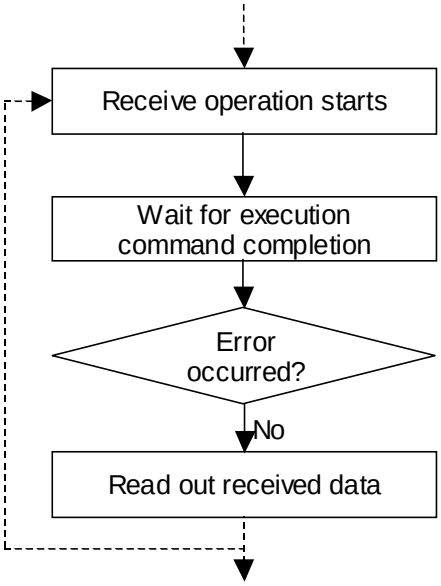


Check that I2C[OxOC] bit 3 is set to “1”

Stop condition setting:
Set the I2C[OxO8] bit [2:0] to “010”

Wait for the I2C[OxOC] bit 0 to be set to “1”

17.5.3 Receive (R) Flow Example



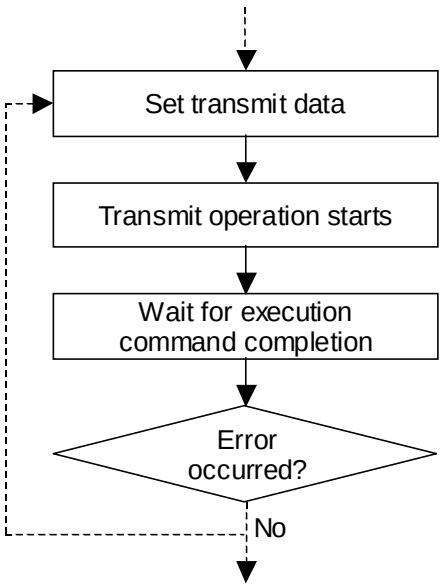
Receive setting:
Set the I2C[OxO8] bit [2:0] to “011”

Wait for the I2C[OxOC] bit 0 to be set to “1”

Check the I2C[OxOC] bit 1
“0” indicates that no error exists; “1”, an error has occurred.

Read out received data of I2C[OxO4]

17.5.4 Transfer (T) Flow Example



Set transmit data in I2C[OxO0]

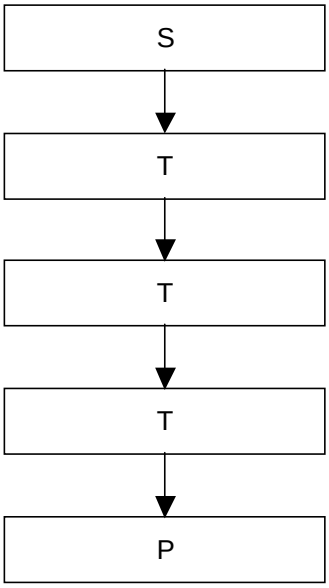
Transfer setting:
Wait for the I2C[OxOC] bit 0 to be set to "100"

Wait for the I2C[OxOC] bit 0 to be set to "1"

Check the I2C[OxOC] bit 1
"0" indicates that no error exists; "1", an error has occurred.

17.5.5 Example of Sequence When Writing to Slave Device

The S/P/R/T in the figure below correspond to above-mentioned Start/Stop/Receive/Transfer flows.



Start condition occurs

Transmits the slave address and Write mode

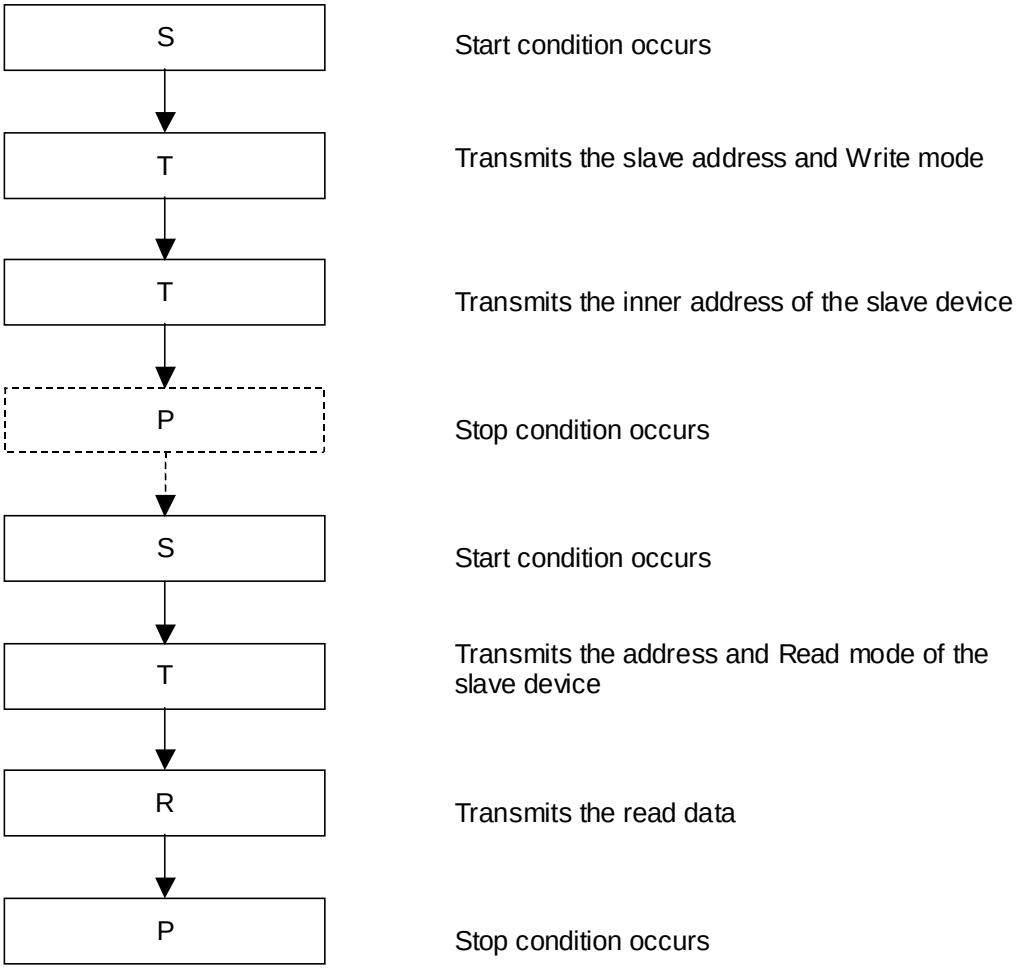
Transmits the internal address of the slave device

Transmits the write data

Stop condition occurs

17.5.6 Example of Sequence When Reading from Slave

The S/P/R/T in the figure below correspond to above-mentioned Start/Stop/Receive/Transfer flows.



17.6 Usage Restrictions for This I²C Single Master Core Module (I²C)

This chip applies restrictions to the following registers.

Offset Address	Register Bit Name	Restrictions
I²C[0x30] Bit 0	WDREQ Signal Monitor	Unavailable
I²C[0x30] Bit 1	RDREQ Signal Monitor	Unavailable

18. I²S (I2S)

18.1 Description

The I²S module complies with the I²S standard defined by Philips. This module is mainly used for voice/audio data communications. I²S supports 2-Channel communications. You can choose transmit or receive for each channel. For example, you can not only receive voice/audio data from an audio device while transmitting voice/audio data to an audio device, but also receive data from two different audio device at the same time.

18.1.1 Function

Functions that I²S provides.

- Allows you to select between master mode (outputs SCK and WS) and slave mode (inputs SCK and WS)
- Allows you to select between transmit mode (outputs SD) and receive mode (inputs SD)
- Supports 16-, 14-, an 8-bit data widths
- Allows you to select between stereo and monaural
- Supports frame cycles, 32fs, 64fs, 128fs, and 256fs
- Allows you to select a clock dividing setting from 1/2 to 1/512 (256 steps) of the source clock (only for master mode)
- Supports DMA
- Clock sharing function (Makes both channels use the same clock)
- Detects FIFO overflow and underflow
- Interrupt generation based on the six types of the FIFO states
- 0 output function (in transmit mode and when the FIFO underflows)
- Conversion from monaural to stereo (only for transmit mode)

18.2 Block Diagram

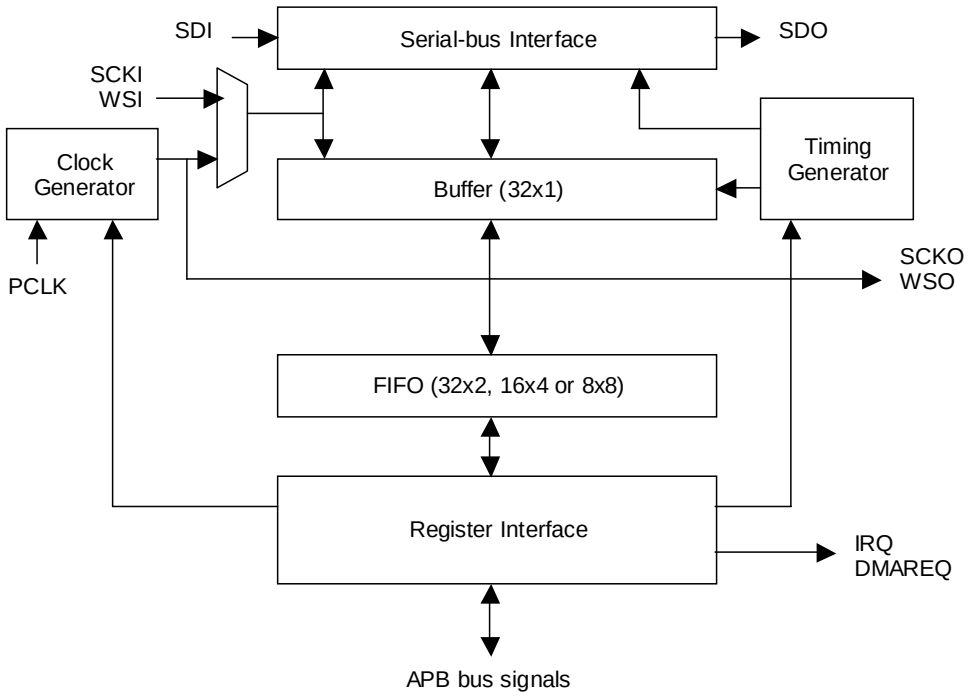


Figure 18.1 I²S Block Diagram

18.3 External Pins

The external pins related to I^2S are as follows.

Pin Name	Input/Output	Pin Functions	Multiplex Pin/Remarks
I^2S_SCK (I^2S0_SCK)	Input and output	I^2S0 serial clock	GPIOB1 (*)
I^2S_WS (I^2S0_WS	Input and output	I^2S0 word select	GPIOB0 (*)
I^2S_SDO (I^2S0_SD)	Input and output	I^2S0 serial data	GPIOB2 (*)
(I^2S1_SCK)	Input and output	I^2S1 serial clock	GPIOA3 (**)
(I^2S1_WS)	Input and output	I^2S1 word select	GPIOA2 (**)
I^2S_SDI (I^2S1_SD)	Input and output	I^2S1 serial data	GPIOB3 (*)

Note (*): The external pins for I^2S are multiplexed with GPIO pins or other pins. You can use the functions for I^2S by setting "Function 2 of other than GPIO" through the GPIO pin function register.

Note (**): For the external WS/SCK pins for I^2S1, set transmit mode on I^2S0 and receive mode on I^2S1, and you do not need external connections. When you connect the external pins, note that they are multiplexed with GPIO pins or other pins. You can use the functions for them by setting "Function 2 of other than GPIO" through the GPIO pin function register.

18.4 Description of Registers

18.4.1 List of Registers

The default base address where the I^2S control registers are located is 0xFFFE_E000.
If not otherwise specified, the default value of any register bit that is not reserved is "0".

Table 18.1 I^2S[1:0] List of Registers

Address Offset	Register Name	Default Value	R/W	Data Access Size
I^2S0 Control Register Base Address: 0xFFFE_E000				
0x00	I^2S0 Control Register	0x0000	R/W	16/32
0x04	I^2S0 Clock Dividing Register	0x0000	R/W	16/32
0x08	I^2S0 Transmit/Receive Port Register	—	R/W	8/16/32
0x10	I^2S0 Interrupt Status Register	0x0000	R/W	16/32
0x14	I^2S0 Interrupt Raw Status Register	0x0009	RO	16/32
0x18	I^2S0 Interrupt Enable Register	0x0000	R/W	16/32
0x1C	I^2S0 Current Status Register	0x0009	RO	16/32
I^2S1 Control Register Base Address: 0xFFFE_E000				
0x40	I^2S1 Control Register	0x0000	R/W	16/32
0x44	I^2S1 Clock Dividing Register	0x0000	R/W	16/32
0x48	I^2S1 Transmit/Receive Port Register	—	R/W	8/16/32
0x50	I^2S1 Interrupt Status Register	0x0000	R/W	16/32
0x54	I^2S1 Interrupt Raw Status Register	0x0009	RO	16/32
0x58	I^2S1 Interrupt Enable Register	0x0000	R/W	16/32
0x5C	I^2S1 Current Status Register	0x0009	RO	16/32

18.4.2 Detailed Description of Registers

I²S[1:0] Control Register							
I²S0[0x00], I²S1[0x40]		Default value = 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 Conversion from Monaural to Stereo (Only for transmit mode)**

Converts monaural data to stereo data before transmitting. Concretely speaking, monaural data to be transmitted on the L channel is also transmitted on the R channel.

0: Disables monaural to stereo conversion
1: Enables monaural to stereo conversion

Bits [11:10]: **FRAMECYC [1:0] Frame Cycle Select (Only for master mode)**

Controls the number of frame cycles. This register bit is effective only in master mode.
In slave mode, an even number of frame cycles that is more than or equal to twice as much as the data width and less than or equal to 256fs can be used.

00: 32fs
01: 64fs
10: 128fs
11: 256fs

Bit 9: **CLKOUTEN Clock Output Enable (Only for master mode)**

Controls clock output.

0: Clock output disable
1: Clock output enable

Bit 8: **CLKSEL Clock Select**

Selects the clock source.

0: Uses SDI/WSI from outside (slave mode) or self-generated clock/word select (master mode).
1: Shares the clock and word select that the other channel uses.

This register bit is used to share I²S clock and I²S word select signals with the I²S of the other channel.

Note: If you set this register bit to "1", set Bit 1 of the same register to "0" to enable slave mode. You cannot use the clock sharing function in master mode.

Bit 7: **SFTRST Software Reset (Write Only)**

Resets the FIFO data, shift register, and internal control circuit. This does not reset the clock generation circuit used in master mode. In receive mode, this does not reset the word select generation circuit, which is also used in master mode. In transmit mode, however, this resets the circuit so that the output becomes "1".

0: n/a
1: Software reset

Bits [6:5]: **DATAWIDTH [1:0] Data Bit Width**

Selects the bit width of the data.

00: 16 bits
01: 14 bits
10: 8 bits
11: Reserved

Bit 4: **MONO/ STEREO Monaural/Stereo Selection**

Selects a data type between stereo and monaural.

0: Stereo type
1: Monaural type

Bit 3: **DMAEN DMA Enable**

Selects whether to use DMA.

0: DMA Disable (Does not issue any DMA request)
1: DMA enable (Issues a DMA request to DMA controller 1)

Bit 2: **TX/RX Transmit/Receive Select**

Selects whether to use the I^2S for transmit mode or for receive mode.

0: Receive mode (Inputs data)
1: Transmit mode (Outputs data)

Bit 1: **MST/SLV Master/Slave Select**

Selects whether to use the I^2S for master mode or for slave mode.

0: Slave mode (Inputs clock and word select)
1: Master mode (Outputs clock and word select)

Bit 0: **I2SEN I^2S Enable**

Controls enable/disable of the I^2S module.

0: I^2S disable
1: I^2S enable

I^2S[1:0] Clock Division Register

I^2S0[0x04], I^2S1[0x44] Default value = 0x0000 Read/Write

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

Bit[7:0]: **CLKDIV [7:0] Clock Division (Only for master mode)**

In master mode, sets the number of divisions for the output clock based on the source clock.

Number of divisions = (CLKDIV + 1) × 2

With this setting, the sampling frequency is obtained from the following expression.

Sampling frequency =
source clock frequency / (number of clock divisions × frame cycle)

In S2S65A30, the source clock frequency is the same as the system clock frequency.

I^2S[1:0] Transmit/Receive Port Register

I^2S0[0x08], I^2S1[0x48] Default value = — 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

Bit[31:0]: **TXD/RXD [31:0] Transmit/Receive Port [31:0]**

In receive mode

Read: Reads I^2S data accumulated in the FIFO.

Write: n/a

In transmit mode

Read: n/a

Write: Writes I^2S data to the FIFO.

The amount of data that can be read or write at one time differs depending on the data width setting and the stereo/monaural setting.

16-bit stereo:

TXD/RXD[31:16] Right data

TXD/RXD[15:0] Left data

14-bit stereo:

TXD/RXD[31:30] 0 data

TXD/RXD[29:16] Right data

TXD/RXD[15:14] 0 data

TXD/RXD[13:0] Left data

8-bit stereo:

TXD/RXD[31:16] Invalid data

TXD/RXD[15:8] Right data

TXD/RXD[7:0] Left 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] 0 data

TXD/RXD[13:0] Monaural data

8-bit monaural:

TXD/RXD[31:8] Invalid data

TXD/RXD[7:0] Monaural data

Note: A whole piece of sampling data must be read or written by one register access because the transmit/receive port register handles data on a sampling basis. For example, if you try to read 16-bit stereo data by 16-bit access, only the left data may be read and the right data may be overwritten by the next data. So, in this case, you should perform 32-bit access to read both the left data and the right data at one time.

I²S[1:0] Interrupt Status Register

I²S0[0x10], I²S1[0x50] Default value = 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 represents a result that is obtained by bit ANDing of the raw interrupt status register (I2S0[0x14], I2S1[0x54]) and the interrupt enable register (I2S0[0x18], I2S1[0x58]).

Bit 5: **OVERFLOWFLG FIFO Overflow Interrupt Flag**

Indicates whether the FIFO has ever overflowed.

Write 0: n/a
Write 1: Attempts to clear the flag.
Read 0: This flag is not enabled or the FIFO has never overflowed.
Read 1: The FIFO has overflowed at least once.

Whether writing "1" to this bit clears the flag depends on modes.

Transmit mode: The flag is always cleared. Overflowing data is lost at the time when the overflow occurs. Data that is already in the FIFO remains there.
Receive mode: The flag is cleared if the write operation is performed after the FIFO has not been full for a period of one I2S clock. The flag is also cleared if the operation performed after executing a software reset.

Bit 4: **UNDERFLOWFLG FIFO Underflow Interrupt Flag**

Indicates whether the FIFO has ever underflowed.

Write 0: n/a
Write 1: Attempts to clear the flag.
Read 0: This flag is not enabled or the FIFO has never underflowed.
Read 1: The FIFO has underflowed at least once.

Whether writing "1" to this bit clears the flag depends on modes.

Transmit mode: The flag is cleared if the write operation is performed in one I²S clock later after writing data to the FIFO. The flag is also cleared if the operation performed after executing a software reset.
Receive mode: The flag is always cleared. The value of the data that is obtained in an underflow condition is not guaranteed.

Bit 3: **NOTFULLFLG FIFO Not Full Interrupt Flag**

Indicates whether the FIFO has ever been in the not full state.

Write 0: n/a
Write 1: Attempts to clear the flag.
Read 0: This flag is not enabled or the FIFO has never been in the not-full state.
Read 1: The FIFO has been in the not full state at least once.

Whether writing "1" to this bit clears the flag depends on the state of the FIFO.

When the FIFO is full: The flag is cleared.
When the FIFO is not full: The flag is not cleared.

Bit 2: **NOTEMPTYFLG FIFO Not Empty Interrupt Flag**

Indicates whether the FIFO has ever been in the not empty state.

Write 0: n/a
Write 1: Attempts to clear the flag.
Read 0: This flag is not enabled or the FIFO has never been in the not empty state.
Read 1: The FIFO has been in the not empty state at least once.

Whether writing "1" to this bit clears the flag depends on the state of the FIFO.

When the FIFO is empty: The flag is cleared.
When the FIFO is not empty: The flag is not cleared.

Bit 1: **FULLFLG FIFO Full Interrupt Flag**

Indicates whether the FIFO has ever been in the full state.

Write 0: n/a
Write 1: Attempts to clear the flag.
Read 0: This flag is not enabled or the FIFO has never been in the full state.
Read 1: The FIFO has been in the full state at least once.

Whether writing "1" to this bit clears the flag depends on the state of the FIFO.

When the FIFO is full: The flag is not cleared.
When the FIFO is not full: The flag is cleared.

Bit 0: **EMPTYFLG FIFO Empty Interrupt Flag**

Indicates whether the FIFO has ever been in the empty state.

Write 0: n/a
Write 1: Attempts to clear the flag.
Read 0: This flag is not enabled or the FIFO has never been in the empty state.
Read 1: The FIFO has been in the empty state at least once.

Whether writing "1" to this bit clears the flag depends on the state of the FIFO.

When the FIFO is empty: The flag is not cleared.
When the FIFO is not empty: The flag is cleared.

I^2S[1:0] Raw Interrupt Status Register

I^2S0[0x14], I^2S1[0x54] Default value = 0x0009 Read Only

n/a							
15	14	13	12	11	10	9	8
n/a		RAW OVERFLOWFLG	RAW UNDERFLOWFLG	RAW NOTFULLFLG	RAW NOTEMPTYFLG	RAWFULLFLG	RAW EMPTYFLG
7	6	5	4	3	2	1	0

Bit 5: **RAWOVERFLOWFLG Raw FIFO Overflow Interrupt Flag**

Indicates whether the FIFO has ever overflowed.

0: The FIFO has never overflowed.
1: The FIFO has overflowed at least once.

Bit 4: **RAWUNDERFLOWFLG Raw FIFO Underflow Interrupt Flag**

Indicates whether the FIFO has ever underflowed.

0: The FIFO has never underflowed.
1: The FIFO has underflowed at least once.

Bit 3: **RAWNOTFULLFLG Raw FIFO Not Full Interrupt Flag**

Indicates whether 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 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 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 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.

I^2S[1:0] Interrupt Enable Register

I^2S0[0x18], I^2S1[0x58] Default value = 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

Bit 5: **OVERFLOWIRQEN FIFO Overflow Interrupt Enable**

Specifies whether an interrupt should occur when the FIFO has overflowed at least once.

0: Does not generate an interrupt.
1: Generates an interrupt.

Bit 4: **UNDERFLOWIRQEN FIFO Underflow Interrupt Enable**

Specifies whether an interrupt should occur when the FIFO has undeflowed at least once.

0: Does not generate an interrupt.
1: Generates an interrupt.

Bit 3: **NOTFULLIRQEN FIFO Not Full Interrupt Enable**

Specifies whether an interrupt should occur when the FIFO has been in the not full state at least once.

0: Does not generate an interrupt.
1: Generates an interrupt.

Bit 2: **NOTEMPTYIRQEN FIFO Not Empty Interrupt Enable**

Specifies whether an interrupt should occur when the FIFO has been in the not empty state at least once.

0: Does not generate an interrupt.
1: Generates an interrupt.

Bit 1: **FULLIRQEN FIFO Full Interrupt Enable**

Specifies whether an interrupt should occur when the FIFO has been in the full state at least once.

0: Does not generate an interrupt.
1: Generates an interrupt.

Bit 0: **EMPTYIRQEN FIFO Empty Interrupt Enable**

Specifies whether an interrupt should occur when the FIFO has been in the empty state at least once.

0: Does not generate an interrupt.
1: Generates an interrupt.

I²S[1:0] Current Status Register

I²S0[0x1C], I²S1[0x5C] Default value = 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

Bit [15:12]: **FIFOWPNTR [3:0] FIFO Write Pointer**

Indicates the current write pointer of the FIFO. (0x0 to 0xF)

Bit [11:8]: **FIFORPNTR [3:0] FIFO Read Pointer**

Indicates the current read pointer of the FIFO. (0x0 to 0xF)

The real address of the FIFO varies depending on the data size and the type (stereo or monaural).
The lower one bit of the pointer: 16-bit stereo, 14-bit stereo
The lower two bits of the pointer: 16-bit monaural, 14-bit monaural, 8-bit monaural
The lower three bits of the pointer: 8-bit monaural

Bit 7: **DMASTS DMA Status**

Indicates whether a DMA request is currently issued.

0: No DMA request is currently issued.
1: A DMA request is currently issued.

Bit 3: **NOTFULLSTS FIFO Not Full Current Status**

Indicates whether the FIFO is currently in the not full state.

0: The FIFO is currently full.
1: The FIFO is not currently full.

Bit 2: **NOTEMPTYSTS FIFO Not Empty Current Status**

Indicates whether the FIFO is currently in the not empty state.

0: The FIFO is currently empty.
1: The FIFO is not currently empty.

Bit 1: **FULLSTS FIFO Full Current Status**

Indicates whether the FIFO is currently in the full state.

0: The FIFO is not currently full.
1: The FIFO is currently full.

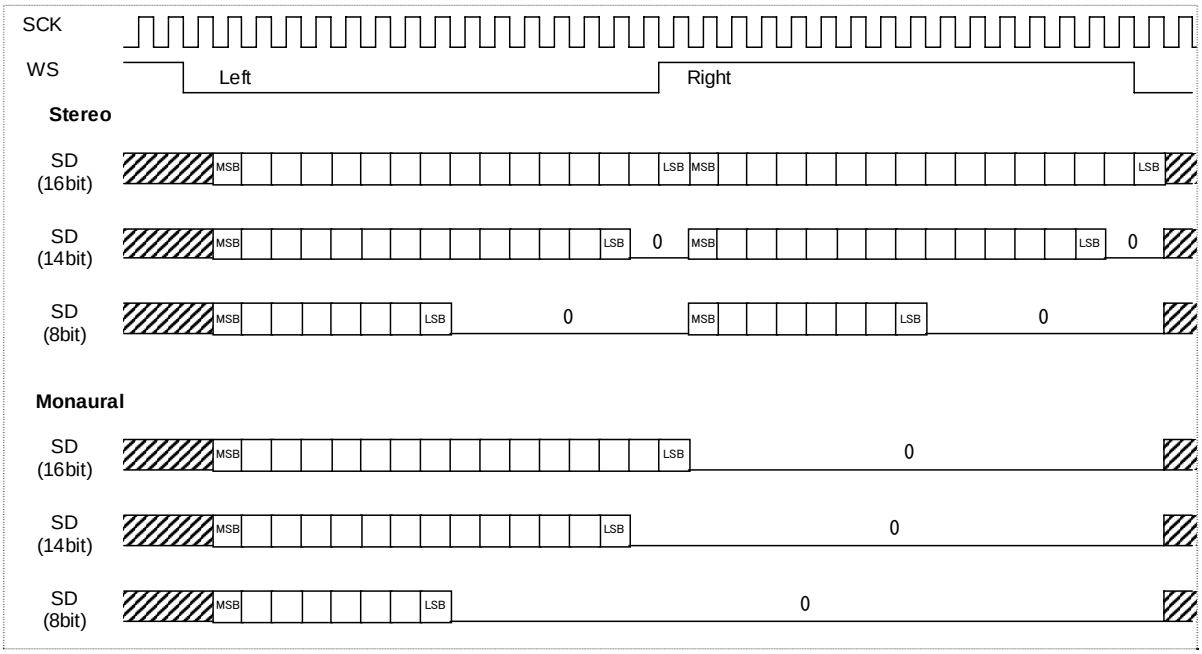
Bit 0: **EMPTYSTS FIFO Empty Current Status**

Indicates whether the FIFO is currently in the empty state.

0: The FIFO is not empty.
1: The FIFO is empty.

18.5 Functional description

18.5.1 I²S Timing Chart (32fs)



18.5.2 Data Width and Number of FIFO Stages

The amount of data that can be read or write at one time differs depending on the data width setting and the stereo/monaural setting.

16-bit stereo:
- TXD/RXD[31:16] Right data
- TXD/RXD[15:0] Left data

14-bit stereo:
- TXD/RXD[31:30] 0 data
- TXD/RXD[29:16] Right data
- TXD/RXD[15:14] 0 data
- TXD/RXD[13:0] Left data

8-bit stereo:
- TXD/RXD[31:16] Invalid data
- TXD/RXD[15:8] Right data
- TXD/RXD[7:0] Left 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] 0 data
- TXD/RXD[13:0] Monaural data

8-bit monaural:
- TXD/RXD[31:8] Invalid data
- TXD/RXD[7:0] Monaural data

Note: A whole piece of sampling data must be read or written by one register access because the transmit/receive port register handles data on a sampling basis. For example, if you try to read 16-bit stereo data by 16-bit access, only the left data may be read and the right data may be overwritten by the next data. So, in this case, you should perform 32-bit access to read both the left data and the right data at one time.

The number of FIFO stages varies depending on the data width and the stereo/monaural setting.
Two-stage FIFO: 16-bit stereo, 14-bit stereo
Four-stage FIFO: 16-bit monaural, 14-bit monaural, 8-bit stereo
Eight-stage FIFO: 8-bit monaural

18.5.3 DMA Transfer

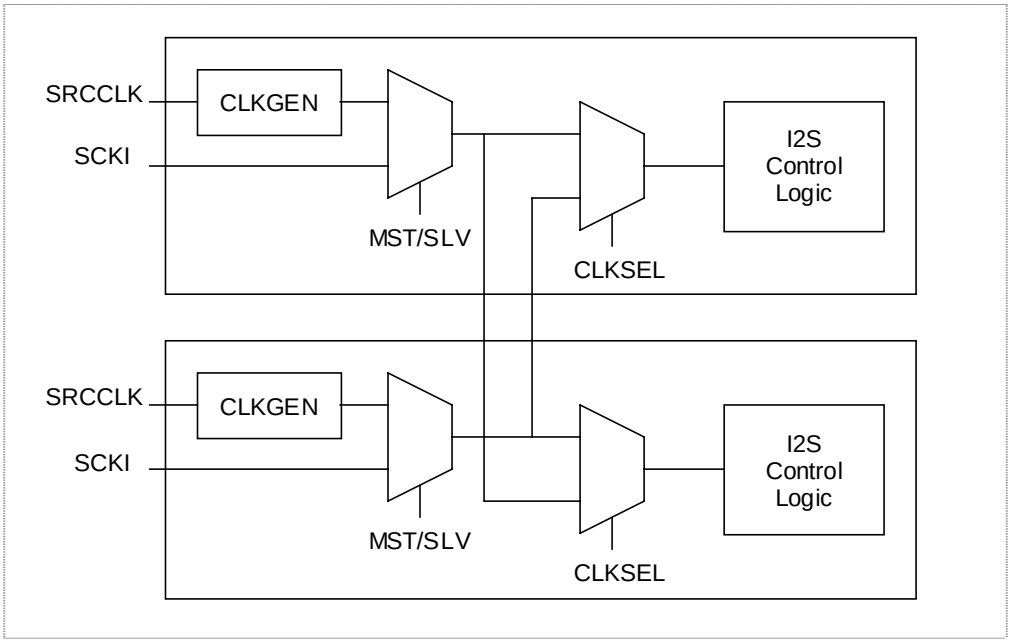
I^2S supports DMA transfer using the DMA controller 1. However, available channels will be 2/3. In transmit mode, a DMA request is asserted when the FIFO is not full; in receive mode, when the FIFO is not empty. A DMA request is negated at the point when the FIFO overflows. This is done for the I^2S in order not to perform unnecessary operations in an abnormal condition of FIFO overflow.
DMA transfer is enabled by setting Bit 3 of I2S0[0x00]/I2S1[0x40] to "1".

18.5.4 Interrupt to INTC

The I^2S, which has a two-channel configuration, transmits an interrupt to INTC (IRQ31) by ORing the two channels. To determine the interrupt cause, you should check both of the I^2S[1:0] interrupt status registers (I2S0[0x10], I2S1[0x50]).

18.5.5 Clock Select (Clock Sharing)

The clock to use can be shared by using the clock select function provided by Bit 8 of I2S0[0x00]/I2S1[0x40]. The following shows the configuration of sharing circuits:



The channel that starts to share the clock as a result of clock input from the other channel is no longer able to output clock in master mode. Therefore, the channel should be used in slave mode by setting Bit 1 of I2S0[0x00]/I2S1[0x40] to "1". The channel that provides clock can be used either in master mode or in slave mode.
Use this clock select function if you want to use the same clock to transmit/receive data by the I^2S simultaneously by using four signal pins.

18.5.6 Monaural-to-Stereo Conversion Function

Monaural data can be converted to stereo data before outputting it to a stereo type audio device. This is done by copying the L data of the monaural data, which only has L data, to R data.
Data is output as stereo data by setting Bit 12 of I2S0[0x00]/I2S1[0x40] to "1" in transmit mode.
Received stereo data cannot be converted to monaural data.

18.6 Setup example

The following is an I2S register setup example where an audio chip with transfer/receive functions is connected by using SCK, WS, SDI, and SDO to receive clock and word select (LR clock) from the audio chip. If the I/O of the I^2S and other I/Os are in sharing state, you must first configure the GPIO registers so that the I/O of the I^2S is enabled, before configuring this setup.

Setup Conditions for Channel 0
- Slave mode
- Transmit mode
- DMA enable
- 16-bit stereo
- 32fs

Setup Conditions for Channel 1
- Slave mode
- Receive mode
- DMA enable
- 16-bit stereo
- 32fs
- Clock sharing

Setup Procedure

I2S0[0x00] = 0x00000005 Slave mode, transmit mode, 16-bit stereo, 32fs
I2S0[0x18] = 0x00000030 Overflow and underflow interrupts enable
I2S0[0x00] = 0x00000085 Software reset
I2S0[0x00] = 0x0000000D DMA enable
I2S1[0x40] = 0x00000101 Slave mode, receive mode, 16-bit stereo, 32fs, clock sharing
I2S1[0x58] = 0x00000030 Overflow and underflow interrupts enable
I2S1[0x40] = 0x00000181 Software reset
I2S1[0x40] = 0x00000109 DMA enable

REVISION HISTORY

Appendix-1

Rev. No.	Date	Page	Type	Description of revisions (including contents of previous revisions) and reasons
Rev 1.0	2009/08/	All pages	New	Newly established
Rev 1.2	2010/4/5	-	Update	Revision of misentry
Rev 1.3	2010/10/12	P29, P31, P33	Update	DMA control register channel 0/1/2/3 initial value 0x0000_0000 -> 0x00XX_XXXX
		P48, P70	Update	JPEG line buffer current status register initial value 0x0009 -> 0xX009
		P190, P197	Update	Line statues register initial value 0x00 -> 0x60
		P190, P199	Update	Modem status register initial value 0x00 -> 0xEX
		P190, P202	Update	Test statue register 1 initial value 0x00 -> 0X01
		P190, P202	Update	Test statue register 2 initial value 0x00 -> 0X0F
		P190, P203	Update	Test statue register 3 initial value 0x00 -> 0X02
		P208, P214	Update	I2C I/O control register 0x11 -> 0x00

EPSON

International Sales Operations

AMERICA

EPSON ELECTRONICS AMERICA, INC.
214 Devcon Drive,
San Jose, CA 95112, USA
Phone: +1-800-228-3964 FAX: +1-408-922-0238

EUROPE

EPSON EUROPE ELECTRONICS GmbH
Riesstrasse 15, 80992 Munich,
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: +86-10-8522-1199 FAX: +86-10-8522-1125

SHANGHAI BRANCH
7F, Block B, Hi-Tech Bldg., 900 Yishan Road,
Shanghai 200233, CHINA
Phone: +86-21-5423-5577 FAX: +86-21-5423-4677

SHENZHEN BRANCH
12F, Dawning Mansion, Keji South 12th Road,
Hi-Tech Park, Shenzhen 518057, CHINA
Phone: +86-755-2699-3828 FAX: +86-755-2699-3838

EPSON HONG KONG LTD.
Unit 715-723, 7/F Trade Square, 681 Cheung Sha Wan Road,
Kowloon, Hong Kong.
Phone: +852-2585-4600 FAX: +852-2827-4346
Telex: 65542 EPSCO HX

EPSON TAIWAN TECHNOLOGY & TRADING LTD.
14F, No. 7, Song Ren Road,
Taipei 110, TAIWAN
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 CORP.
KOREA OFFICE
5F, KLI 63 Bldg., 60 Yoido-dong,
Youngdeungpo-Ku, Seoul 150-763, KOREA
Phone: +82-2-784-6027 FAX: +82-2-767-3677

SEIKO EPSON CORP.
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: 411697903
First Issue September 2009 (H)
Revised Oct 2011 in JAPAN