HMS30C7210

ARM Based 32-Bit Microprocessor

DATASHEET

(7210 DS-07)

MagnaChip Semiconductor Ltd.
#1, Hyangjeong-dong, Heungduk-gu, Cheongju-si, Chungcheongbuk-do, Republic of Korea

Homepage: www.magnachip.com
Technical Support Homepage: www.softonchip.com

H.Q. of MagnaChip Semiconductor Ltd.
Telephone: 82-(0)43-270-4070
Facsimile: 82-(0)43-270-4099

Marketing Site Sales in Korea
Telephone: 82-(0)43-270-4085
Facsimile: 82-(0)43-270-4099

Sales in Korea
Telephone: 82-(0)2-3459-3738
Facsimile: 82-(0)2-3459-3945

World Wide Sales Network

U.S.A.
Telephone: 1-408-232-8757
Facsimile: 1-408-232-8135

Taiwan
Telephone: 886-(0)2-2500-8357
Facsimile: 886-(0)2-2509-8977

Hong Kong
Telephone: 852-2971-1640
Facsimile: 852-2971-1622

Proprietary Notice

Change Log

Issue	Date	By	Change
A-01	2004/02/23	Injae Koo	The First Draft
A-02	2004/07/05	Hyerim Chung	ADC / LCD / RTC / SCI / SDRAM / SSI / TIMER / UART / USB Add ELECTRICAL CHARACTERISTICS
A-03	2004/10/04	Hyerim Chung	SMI / UART / SCI / KEYBOARD / PIN DESCRIPTION / ELECTRICAL CHARACTERISTICS
A-04	2004/12/14	Hyerim Chung	Matrix Keyboard Interface Controller
A-05	2005/01/13	Hyerim Chung	UART p82 Interrupt Identification Register Table
A-06	2005/01/25	Hyerim Chung	Timer & PWM / Matrix Keyboard Interface Controller
DS-07	2005/03/22	Hyun-il Kim	Change DataSheet Style and Adding contents

OVERVIEW

The HMS30C7210 is a highly integrated low power microprocessor for card reader system, and other applications described below. The device incorporates an ARM720T CPU and system interface logic to interface with various types of devices. HMS30C7210 is a highly modular design based on the AMBA bus architecture between CPU and internal modules.

The on-chip peripherals include LCD controller with DMA support for internal SRAM and external SDRAM memory, analog functions such as ADC and PLL. Intelligent interrupt controller and internal 8Kbytes SRAM can support an efficient interrupt service execution. The HMS30C7210 also supports a touch panel interface. UART and USB provide serial communication channels for external systems. The power management features result in very low power consumption. The HMS30C7210 provides an excellent solution for card reader system.

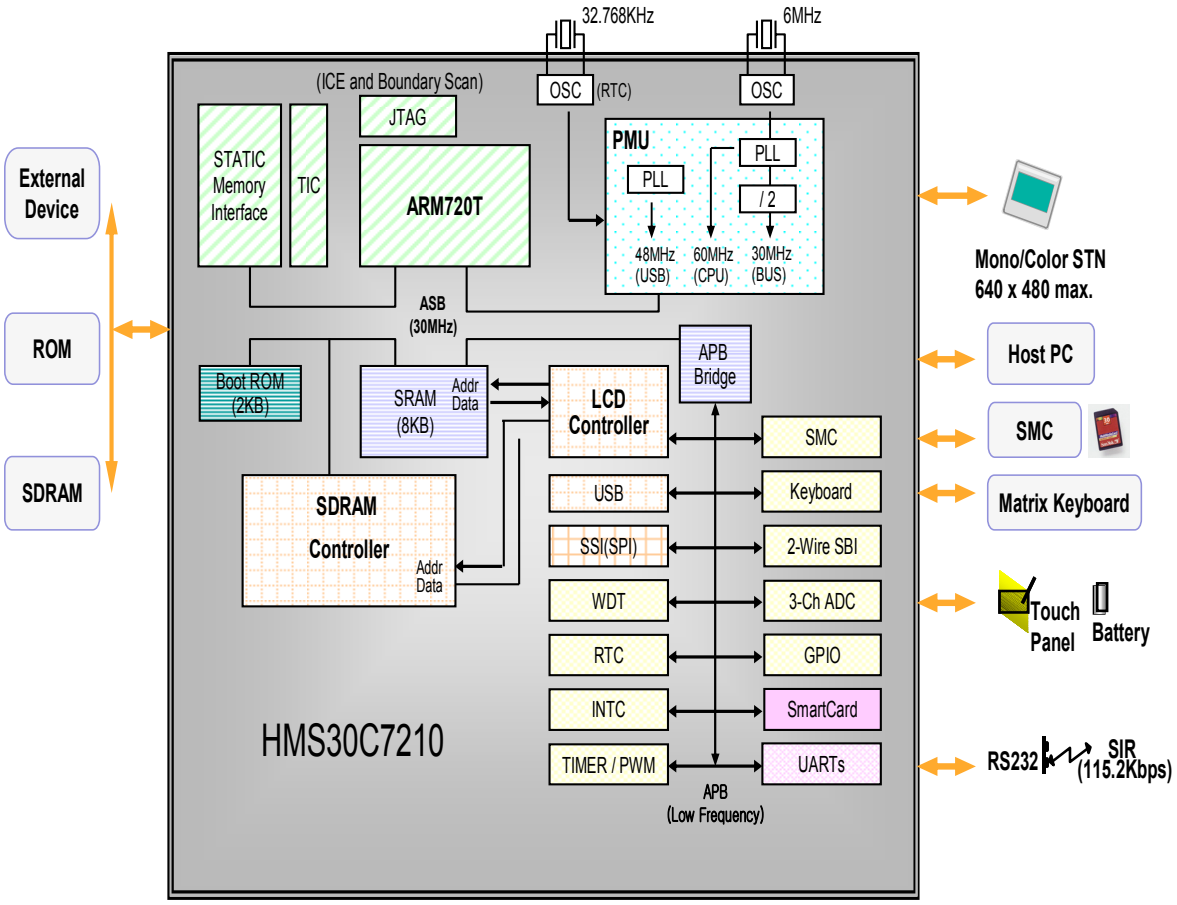
FEATURES

32-bit ARM7TDMI RISC static CMOS CPU core (Running up to 60 MHz)
8Kbytes combined instruction/data cache
Memory management unit
Supports Little-endian operating system
8Kbytes SRAM for internal buffer memory
2Kbytes Boot ROM
On-chip peripherals with individual power-down:

- Memory controller for ROM(x8,16), Flash(x8,16), SRAM(x8,16), SDRAM(x16)
- 5-State Power management unit (Sofrware selectable Clock Frequency)
- Interrupt Controller
- LCD Controller for color and mono STN
- USB 1.1(slave)
- Two Smart Card Interface (UART 0,1)
- Two UART (UART 2,3)
- One SIR support UART (UART4)
- One Modem support UART (UART5)
- Four 16-bit Timer Channels (with Output Port)
- Two 16-bit PWM Channels (with Output Port)
- Programmable WatchDog Timer with On-chip Oscillator
- Real-time clock (32.768kHz oscillator) with separated Vcc
- Matrix Keyboard control interface (6x6)
- 97 Programmable GPIO
- One 2-Wire Serial Bus Interface
- 2-Channel Master/Slave SSI (SPI)
- SMC Card Interface
- On-chip 3-Channel 10-bit ADC

JTAG debug interface and boundary scan
0.35um CMOS Process
3.3V supply voltage
208-pin LQFP / CABGA package
Low power consumption

HMS30C7210 System Overview



LIST OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 ARCHITECTURAL OVERVIEW

1.1 Processor

The ARM720T core incorporates an 8KB unified write-through cache, and an 8 data entry, 4-address entry write buffer. It also incorporates an MMU with a 64 entry TLB.

1.2 Video

The integrated LCD controller can control color and monochrome STN displays, up to 640x480 (VGA) resolution. 1, 2, 4, and 8 bit-per-pixel is supported and a patented gray scaler can directly generate 16 gray scales.

1.3 Memory

The 16-bit external data path interfaces to ROM or Flash devices. Burst mode ROMs are supported, for increased performance, allowing operating system code to be executed directly from ROM.

1.4 Internal Bus Structure

The HMS30C7210 internal bus organization is based upon the AMBA standard, but with some minor modifications to the peripheral buses (the APBs). There are two main buses in the HMS30C7210:

- The main system bus (the ASB) to which the CPU and memory controllers are connected
- The APB to peripherals are connected

1.4.1 ASB

The ASB is designed to allow the ARM continuous access to both, the ROM and the SDRAM interface. The SDRAM controller straddles both the ASB and the video DMA bus so the LCD can access the SDRAM controller simultaneously with activity on the ASB. This means that the ARM can read code from ROM, or access a peripheral, without being interrupted by video DMA.
The HMS30C7210 uses a modified arbiter to control mastership on the main ASB bus. The arbiter only arbitrates on quad-word boundaries, or when the bus is idle. This is to get the best performance with the ARM720T, which uses a quad-word cache line, and also to get the best performance from the SDRAM, which uses a burst size of eight half-words per access. By arbitrating only when the bus is idle or on quad-word boundaries (A[3:2] = 11), it ensures that cache line fills are not broken up, hence SDRAM bursts are not broken up.
The SDRAM controller controls video ASB arbitration. This is explained in 6.5 Arbitration.

1.4.2 Video bus

The video bus connects the LCD controller with the internal SRAM and the SDRAM controller. Data transfers are DMA controlled. The video bus consists of an address bus, data bus and control signals to/from the internal SRAM and the SDRAM controller. The LCD registers are programmed through the fast APB. The SDRAM

controller arbitrates between ASB, Video access requests. Video always has higher priority than ASB access requests. The splitting ASB/video bus allows slow ASB device accesses internal SRAM and SDRAM without blocking video DMA.

1.4.3 APB

The most APB peripherals do not support DMA transfers. This arrangement of running most of the peripherals at a slower clock, and reducing the load on the faster bus, results in significantly reduced power consumption. The APB bus connects to the main ASB bus via bridges. The APB Bridge takes care of all resynchronization, handing over data and control signals between the ASB and UART clock domains in a safe and reliable manner. USB, LCD Controller, SMC and SPI are operated at the speed of the ASB. Theses are high performance peripherals.

1.5 SDRAM Controller

The SDRAM controller is a key part of the HMS30C7210 architecture. The SDRAM controller has two data ports - one for video DMA and one for the main ASB - and interfaces to 16-bit wide SDRAMs. One to four 16, 64, 128, or 256 Mbits x 16-bit devices are supported, giving a memory size ranging from 2 to 64 Mbytes.
The main ASB and video DMA buses are independent, and operate concurrently. The video bus has always higher priority than the main bus. The video interface consists of address, data and control signals. The video access burst size is fixed to 16 words. The address is non-incrementing for words within a burst (as the SDRAM controller only makes use of the first address for each burst request).

1.6 Peripherals

Universal Serial Bus (USB) device controller

The USB device controller is used to transfer data from/to host system like PC in full-speed (12Mbits/s) mode. No external USB transceiver is necessary.

Universal Asynchronous Receiver and Transmitter (UART)

Six UART ports are implemented. UART0,1 supports Smart Card Interface signals.

IrDA / Modem

IrDA uses UART4 for its SIR transfer in 115 Kbit/s speed. UART5 supports full modem interface signals.

Pulse-Width-Modulated (PWM) Interface

Two PWM output signals are generated. The pins are used as GPIO when not used for PWM.

Matrix Keyboard Interface

Matrix keyboard interface supports 6x6. The pins are used as GPIO when not used for matrix keyboards.

ADC

3 channel ADC is implemented for touch panel, monitoring of battery voltage or general purpose.

PLL

CPU, video and USB clocks are generated by two PLLs with 6 MHz input clock.

1.7 Power management

The HMS30C7210 incorporates advanced power management functions, allowing the whole device to be put into a standby mode, when only the real time clock runs. The SDRAM is put into low-power self-refresh mode to preserve its contents. The HMS30C7210 may be forced out of this state by either a real-time clock wake-up interrupt, a user wake-up event (which would generally be a user pressing the “on” key) or by the UART ring-indicate input. The power management unit (PMU) controls the safe exit from standby mode to operational mode, ensuring that SDRAM contents are preserved. In addition, halt and slow modes allow the processor to be halted or run at reduced speed to reduce power consumption. The processor can be quickly brought out of the halted state by a peripheral interrupt. The advanced power management unit controls all this functionality. In addition, individual devices and peripherals may be powered down when they are not in use. The HMS30C7210 is designed for battery-powered portable applications and incorporates innovative design features in the bus structure and the PMU to reduce power consumption. The APB bus allows peripherals to be clocked slowly hence reducing power consumption. The use of two buses reduces the number of nodes that are toggled during a data access, and thereby further reducing power consumption. In addition, clocks to peripherals that are not active can also be gated.

1.7.1 Clock gating

The high performance peripherals, such as the SDRAM controller and the LCD controller, run most of the time at high frequencies and careful design, including the use of clock gating, has minimized their power consumption. Any peripherals can be powered down completely when not in use.

1.7.2 PMU

The Power Management Unit (PMU) is used to control the overall state the system is in. The system can be in one of five states:

RUN

The system is running normally. All clocks are running (except where gated locally). The SDRAM controller is performing normal refresh.

SLOW

The CPU is switched into FastBus mode, and hence runs at the BCLK rate (half the FCLK rate). This is the default mode after exiting DEEP SLEEP mode or system power on.

IDLE

In this mode, the PMU becomes the bus master until there is either a fast or normal interrupt for the CPU.
This will cause the clocks in the CPU to stop when it attempts an ASB access. The HMS30C7210 can enter this mode by writing 0x2 to the bits [2:0] of the PMUMR when in RUN or SLOW mode, or by WakeUp signal activation while in SLEEP or DEEP SLEEP mode.

SLEEP

In this mode, the SDRAM is put into self-refresh mode, and internal clocks are gated off. This mode can only be entered from IDLE mode (the PMU bus master must have the mastership of the ASB before this mode can be entered). The PMU must be the bus master to ensure that the system is stopped in a safe state, and is not half way through a SDRAM write (for example). Both the Video and Communication clocks (VCLK and CCLK) should be disabled before entering this state.
Usually the CPU would only drop in at this mode on the way to the DEEP SLEEP mode.

DEEP SLEEP

In the DEEP SLEEP mode, the crystal oscillator for the 6-MHz PLL input clock and the PLLs are disabled. This is the lowest power state available. Only the 32-KHz RTC oscillator runs and provides clocks for the RTC logic and the debouncing logic of the PMU, which runs at the 4-KHz frequency (i.e. the RTC clock frequency divided by 8). Everything else is powered down, and SDRAM is in self refresh mode. This is the normal system "off" mode.
The HMS30C7210 can get out of the SLEEP and DEEP SLEEP modes either by a user wake-up event (generally pressing the "On" key), by an RTC wake-up alarm, or by a modem ring indicate event. These wake-up sources go directly to the PMU.

1.8 Test and debug

The HMS30C7210 incorporates the ARM standard test interface controller (TIC) allowing 32-bit parallel test vectors to be passed onto the internal bus. This allows access to the ARM720T macro-cell core, and also to memory mapped devices and peripherals within the HMS30C7210. In addition, the ARM720T includes support for the ARM debug architecture (Embedded ICE), which makes use of a JTAG boundary scan port to support debug of code on the embedded processor. The same boundary scan port is also used to support a normal pad-ring boundary scan for board level test applications.

2 SIGNAL DESCRIPTION

2.1 208-Pin Diagram

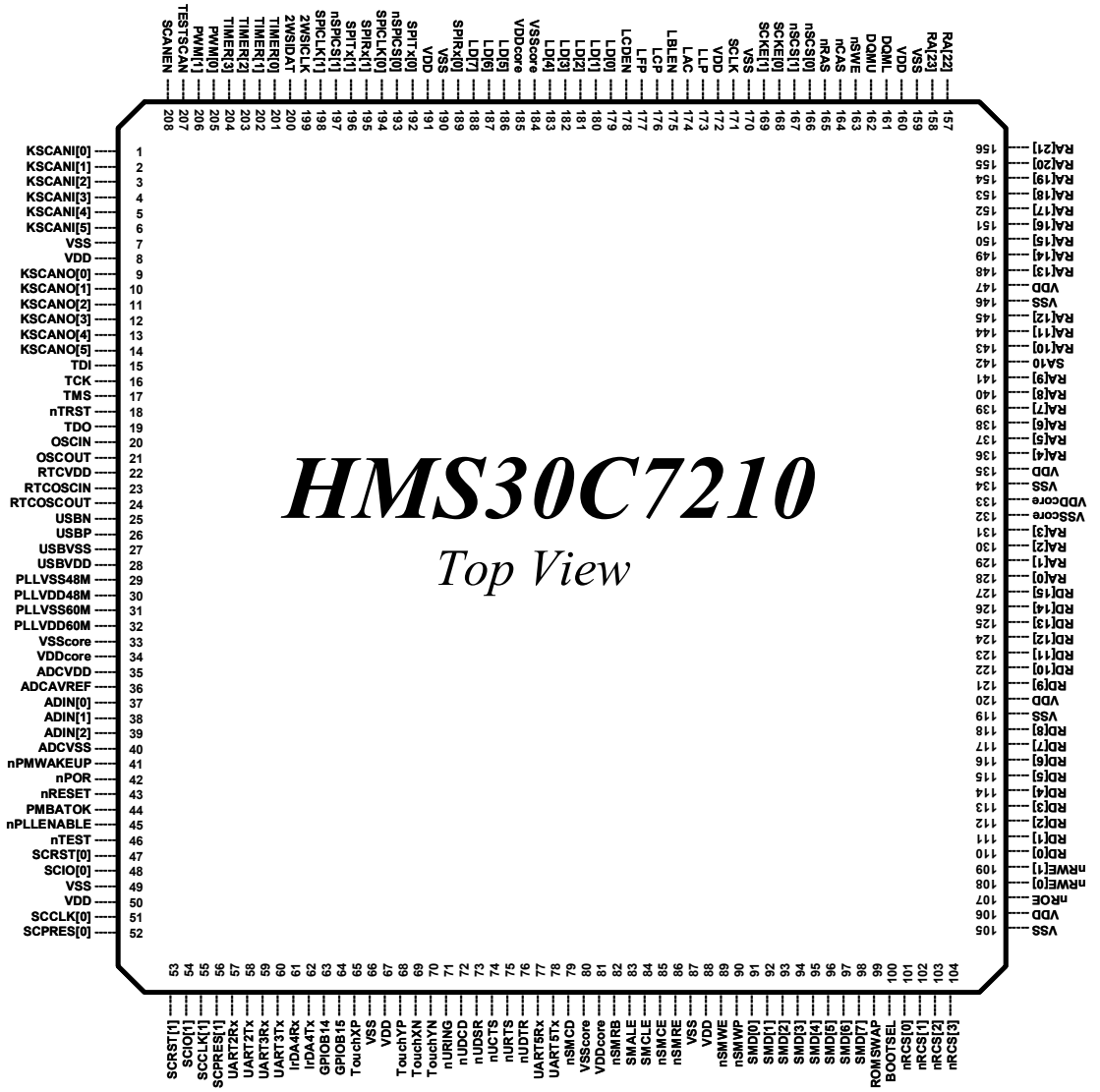


Figure 2-1. 208 Pin diagram

2.2 208 Pin / Ball Name

Pin No.	Ball No.	PAD Name	Pin No.	Ball No.	PAD Name	Pin No.	Ball No.	PAD Name	Pin No.	Ball No.	PAD Name
1	A1	KSCANI[0]	53	U1	SCRST[1]	105	U17	VSS	157	A17	RA[22]
2	B2	KSCANI[1]	54	T2	SCIO[1]	106	T16	VDD	158	B16	RA[23]
3	B1	KSCANI[2]	55	U2	SCCLK[1]	107	T17	nROE	159	A16	VSS
4	C1	KSCANI[3]	56	U3	SCPRES[1]	108	R17	nRWE[0]	160	A15	VDD
5	D3	KSCANI[4]	57	R4	UART2Rx	109	P15	nRWE[1]	161	C14	DQML
6	C2	KSCANI[5]	58	T3	UART2Tx	110	R16	RD[0]	162	B15	DQMU
7	D1	VSS	59	U4	UART3Rx	111	P17	RD[1]	163	A14	nSWE
8	E4	VDD	60	P5	UART3Tx	112	N14	RD[2]	164	D13	nCAS
9	E3	KSCANO[0]	61	R5	IrDA4Rx	113	N15	RD[3]	165	C13	nRAS
10	D2	KSCANO[1]	62	T4	IrDA4Tx	114	P16	RD[4]	166	B14	nSCS[0]
11	E2	KSCANO[2]	63	T5	GPIOB14	115	N16	RD[5]	167	B13	nSCS[1]
12	F3	KSCANO[3]	64	R6	GPIOB15	116	M15	RD[6]	168	C12	SCKE[0]
13	F4	KSCANO[4]	65	P6	TouchXP	117	M14	RD[7]	169	D12	SCKE[1]
14	E1	KSCANO[5]	66	U5	VSS	118	N17	RD[8]	170	A13	VSS
15	F2	TDI	67	T6	VDD	119	M16	VSS	171	B12	SCLK
16	G3	TCK	68	R7	TouchYP	120	L15	VDD	172	C11	VDD
17	G4	TMS	69	P7	TouchXN	121	L14	RD[9]	173	D11	LLP
18	F1	nTRST	70	U6	TouchYN	122	M17	RD[10]	174	A12	LAC
19	G2	TDO	71	T7	nURING	123	L16	RD[11]	175	B11	LBLEN
20	H3	OSCIN	72	R8	nUDCD	124	K15	RD[12]	176	C10	LCP
21	H4	OSCOUT	73	P8	nUDSR	125	K14	RD[13]	177	D10	LFP
22	G1	RTCVDD	74	U7	nUCTS	126	L17	RD[14]	178	A11	LCDEN
23	H2	RTCOSCIN	75	T8	nURTS	127	K16	RD[15]	179	B10	LD[0]
24	J3	RTCOSCOUT	76	R9	nUDTR	128	J15	RA[0]	180	C9	LD[1]
25	J4	USBN	77	P9	UART5Rx	129	J14	RA[1]	181	D9	LD[2]
26	H1	USBP	78	U8	UART5Tx	130	K17	RA[2]	182	A10	LD[3]
27	J2	USBVSS	79	T9	nSMCD	131	J16	RA[3]	183	B9	LD[4]
28	K3	USBVDD	80	R10	VSScore	132	H15	VSScore	184	C8	VSScore
29	K4	PLLVSS48M	81	P10	VDDcore	133	H14	VDDcore	185	D8	VDDcore
30	J1	PLLVDD48M	82	U9	nSMRB	134	J17	VSS	186	A9	LD[5]
31	K2	PLLVSS60M	83	T10	SMALE	135	H16	VDD	187	B8	LD[6]
32	L3	PLLVDD60M	84	R11	SMCLE	136	G15	RA[4]	188	C7	LD[7]
33	L4	VSScore	85	P11	nSMCE	137	G14	RA[5]	189	D7	SPIRx[0]
34	K1	VDDcore	86	U10	nSMRE	138	H17	RA[6]	190	A8	VSS
35	L2	ADCVDD	87	T11	VSS	139	G16	RA[7]	191	B7	VDD
36	M3	ADCVREF	88	R12	VDD	140	F15	RA[8]	192	C6	SPITx[0]
37	M4	ADIN[0]	89	P12	nSMWE	141	F14	RA[9]	193	D6	nSPICS[0]
38	L1	ADIN[1]	90	U11	nSMWP	142	G17	SA10	194	A7	SPICLK[0]
39	M2	ADIN[2]	91	T12	SMD[0]	143	F16	RA[10]	195	B6	SPIRx[1]
40	N3	ADCVSS	92	R13	SMD[1]	144	E15	RA[11]	196	C5	SPITx[1]
41	N4	nPMWAKEUP	93	P13	SMD[2]	145	E14	RA[12]	197	D5	nSPICS[1]
42	M1	nPOR	94	U12	SMD[3]	146	F17	VSS	198	A6	SPICLK[1]
43	N2	nRESET	95	T13	SMD[4]	147	E16	VDD	199	B5	2WSICLK
44	P3	PMBATOK	96	R14	SMD[5]	148	D15	RA[13]	200	C4	2WSIDAT
45	P4	nPLLENABLE	97	P14	SMD[6]	149	D14	RA[14]	201	D4	TIMER[0]
46	N1	nTEST	98	U13	SMD[7]	150	E17	RA[15]	202	A5	TIMER[1]
47	R3	SCRST[0]	99	R15	ROMSWAP	151	C15	RA[16]	203	C3	TIMER[2]
48	P2	SCIO[0]	100	T14	BOOTSEL	152	D16	RA[17]	204	B4	TIMER[3]
49	P1	VSS	101	U14	nRCS[0]	153	D17	RA[18]	205	A4	PWM[0]
50	R1	VDD	102	U15	nRCS[1]	154	C17	RA[19]	206	A3	PWM[1]
51	R2	SCCLK[0]	103	T15	nRCS[2]	155	C16	RA[20]	207	B3	TESTSCAN
52	T1	SCPRES[0]	104	U16	nRCS[3]	156	B17	RA[21]	208	A2	SCANEN

2.2.1 LQFP Type Dimensions

- All dimensions in mm.

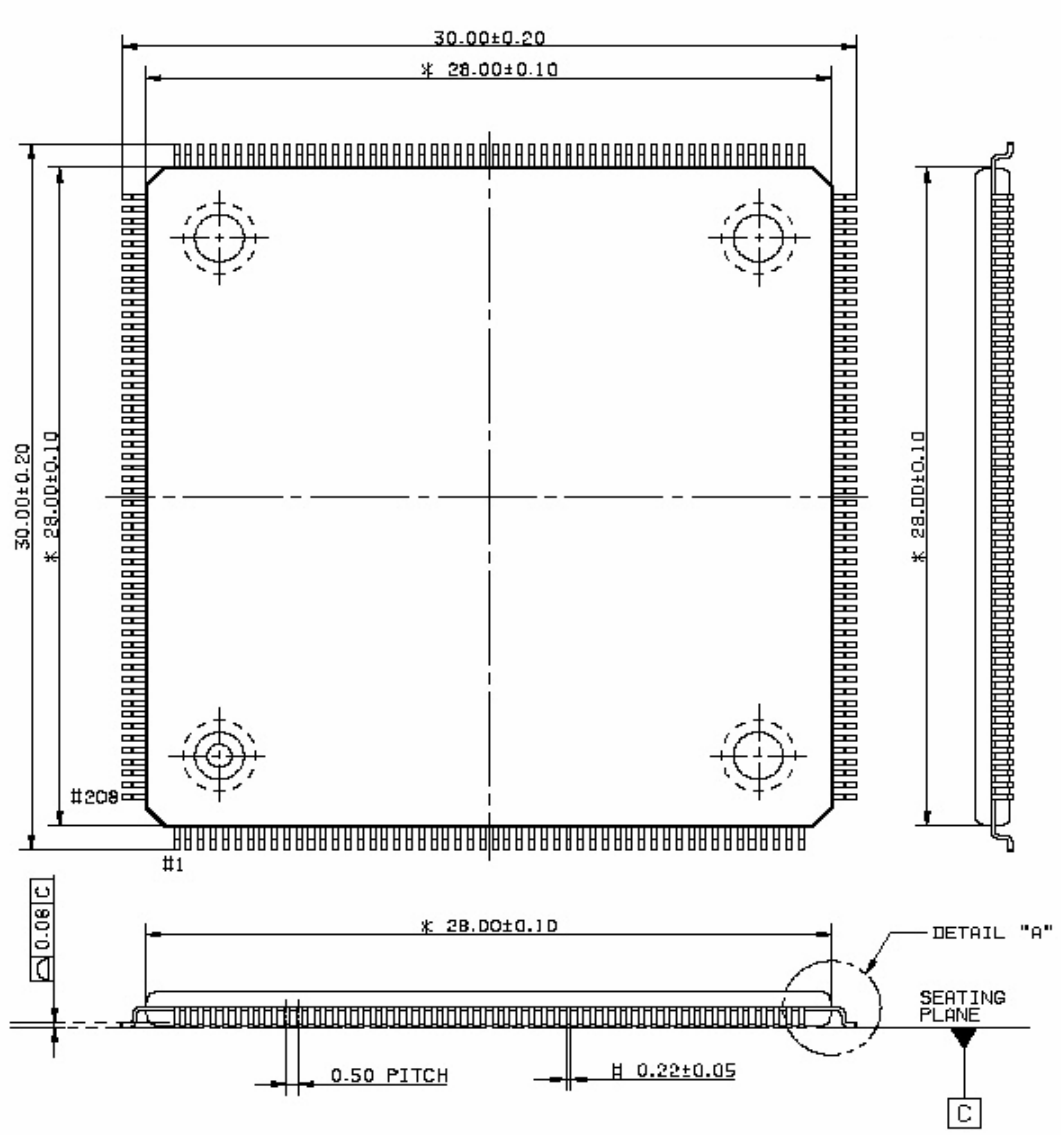


Figure 2-2. 208 LQFP Dimensions-1

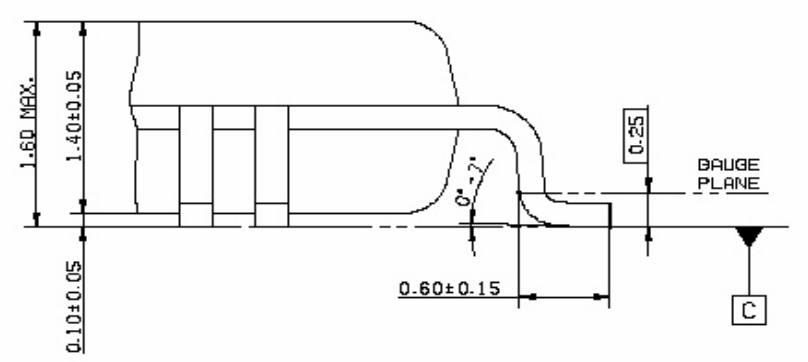


Figure 2-3. 208 LQFP Dimensions-2 < Detail "A" (Scale 1/30) >

2.2.2 CABGA Type Dimensions

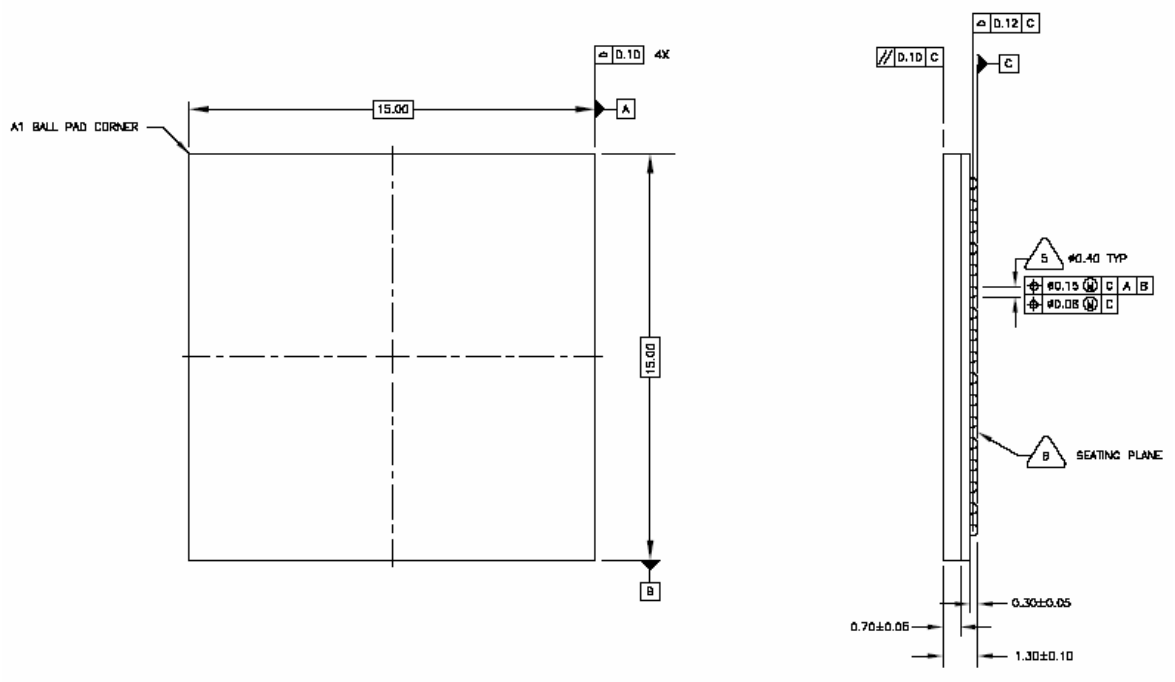


Figure 2-4. 208 CABGA Top and Side view

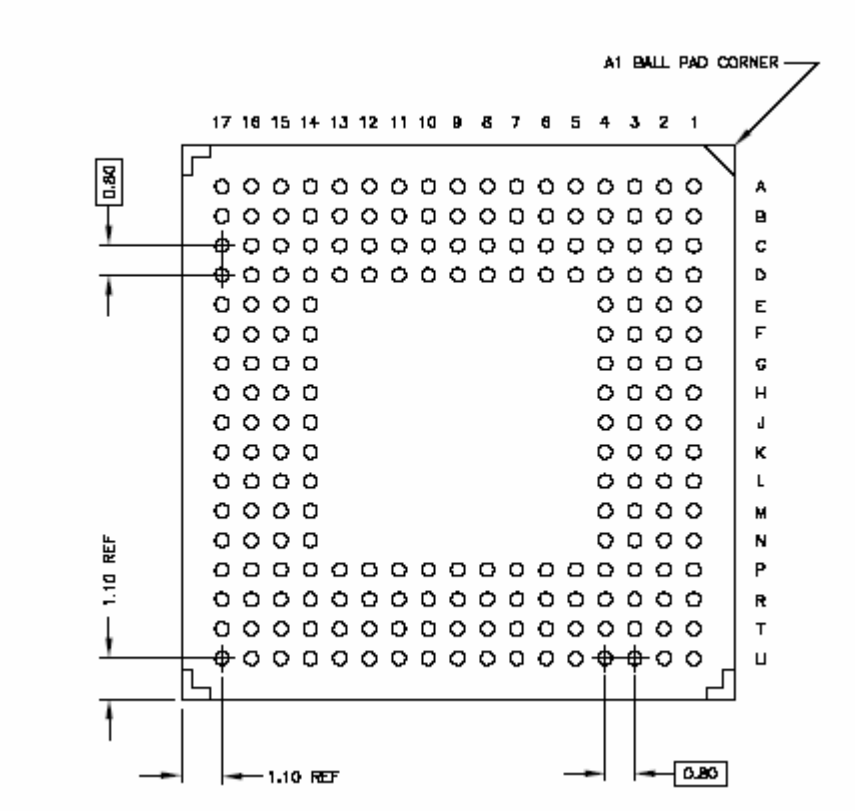


Figure 2-5. 208 CABGA Bottom view

2.3 Pin Descriptions

Table 2-2 describes the function of all the external signals to the HMS30C7210.

Type	Description	Type	Description
O	Output	AO	Analog Output
I	Input	AI	Analog Input
IO	Input/Output	AIO	Analog Input/Output
IS	Input with Schmitt level input threshold	P	Power input
U	Suffix to indicate integral pull-up	D	Suffix to indicate integral pull-down

Table 2-1 Pin Signal Type Definition

2.3.1 External Signal Functions

Function	Signal Name	Signal Type	LQFP Pin Number	Description
LCD	LD[7:0]	O	179~183 186~188	LCD data bus LD[3:0] for 4bit bus LCD and LD[7:0] for 8-bit bus LCD
	LCP	O	176	LCD clock pulse
	LLP	O	173	LCD line pulse
	LFP	O	177	LCD frame pulse
	LAC	O	174	LCD AC bias
	LCDEN	O	178	Display enable signal for LCD Enables high voltage to LCD
	LBLEN	O	175	LCD backlight enable
SMI (Static Memory Interface)	RA[24:0]	O	128~131 136~141 143~145 148~158	ROM address bus
	RD[15:0]	IO	110~118 121~127	ROM data bus
	nRCS[3:0]	O	101~104	ROM chip select outputs
	nROE	O	107	ROM output enable signal
	nRWE[1:0]	O	108~109	ROM write enable signals
SDRAM Interface	RA[14:11],[9:0]	O	128~131 136~141 144~145 148~149	SDRAM address bus
	SA10	O	142	SDRAM address bus (for PRECHARGE command)
	RD[15:0]	IO	110~118 121~127	SDRAM data bus
	SCLK	O	171	SDRAM clock output
	SCKE[1:0]	O	168~169	SDRAM clock enable outputs
	nRAS	O	165	SDRAM row address select output
	nCAS	O	164	SDRAM column address select output
	nSWE	O	163	SDRAM write enable output
	nSCS[1:0]	O	166~167	SDRAM chip select outputs
	DQML	O	161	SDRAM lower data byte enable
	DQMU	O	162	SDRAM upper data byte enable
Smart Card Interface (UART 0,1)	SCIO[1:0]	IO	48,54	SmartCard data I/O (UART 0,1 Tx)
	SCRST[1:0]	IO	47,53	SmartCard reset outputs (UART 0,1 Rx)
	SCPRES[1:0]	I	52,56	SmartCard presence detection (not used at UART mode)
	SCCLK[1:0]	O	51,55	SmartCard clock outputs (not used at UART mode)
UART 2	UART2Tx	O	58	UART2 serial data output
	UART2Rx	I	57	UART2 serial data input

Function	Signal Name	Signal Type	LQFP Pin Number	Description
UART 3	UART3Tx	O	60	UART3 serial data output
	UART3Rx	I	59	UART3 serial data input
IrDA (UART 4)	IrDA4Tx	O	62	IrDA serial data output (UART4 Tx)
	IrDA4Rx	I	61	IrDA serial data input (UART4 Rx)
UART 5 (For a Modem device application)	UART5Tx	O	78	UART5 serial data output
	UART5Rx	I	77	UART5 serial data input
	nUDCD	I	72	UART5 data carrier detect input
	nUDSR	I	73	UART5 data set ready input
	nUCTS	I	74	UART5 clear to send input
	nUDTR	O	76	UART5 data terminal ready
	nURTS	O	75	UART5 request to send
	nURING	I	71	UART5 ring input signal
SSI (SPI)	SPITx[1:0]	O	192,196	SPI data output
	SPIRx[1:0]	I	189,195	SPI data input
	nSPICS[1:0]	O	193,197	SPI chip select signal
	SPICLK[1:0]	O	194,198	SPI clock output
2WSI	2WSICLK	IO	199	2WSI clock input/output
	2WSIDAT	IO	200	2WSI data input/output
USB	USBP	AIO	26	USB positive signal
	USBN	AIO	25	USB negative signal
	USBVDD	P	28	USB analog Vdd
	USBVSS	P	27	USB analog Vss
TIMER, PWM	TIMER[3:0]	O	201~204	Timer data output
	PWM[1:0]	O	205~206	Pulse Width Modulator data output
Matrix Keyboard	KSCANO[5:0]	O	9~14	Matrix keyboard scan output
	KSCANI[5:0]	I	1~6	Matrix keyboard scan input
SMC (SmartMedia Card)	SMD[7:0]	IO	91~98	SMC bi-directional data signal
	nSMWP	O	90	SMC write protect
	nSMWE	O	89	SMC write enable
	SMALE	O	83	SMC address latch enable
	SMCLE	O	84	SMC command latch enable
	nSMCD	I	79	SMC card detection signal
	nSMCE	O	85	SMC chip enable
	nSMRE	O	86	SMC read enable
	nSMRB	I	82	SMC ready/busy signal
ADC	TouchXP	IO	65	Touch screen switch X-positive drive
	TouchXN	O	69	Touch screen switch X-negative drive
	TouchYP	IO	68	Touch screen switch Y-positive drive
	TouchYN	O	70	Touch screen switch Y-negative drive
	ADIN[2:0]	AI	37~39	ADC input for battery, touch
	RTCVDD	I	22	RTC Vdd
	ADCVDD	P	35	ADC analog Vdd
	ADCVSS	P	40	ADC analog Vss
	ADCVREF	AI	36	ADC reference voltage
PLL	PLLVDD48M	P	30	PLL 48MHz analog Vdd
	PLLVSS48M	P	29	PLL 48MHz analog Vss
	PLLVDD60M	P	32	PLL 60MHz analog Vdd
	PLLVSS60M	P	31	PLL 60MHz analog Vss
GPIO	GPIOA[11:0]	IO	1~6,9~14	General purpose input/output signals
	GPIOB[27:0]	IO	47,48,51~65,68 ~78	General purpose input/output signals
	GPIOC[15:0]	IO	79,82~86, 89~98	General purpose input/output signals

Function	Signal Name	Signal Type	LQFP Pin Number	Description
	GPIOD[24:0]	IO	103,104,161~169,173~183,186~188	General purpose input/output signals
	GPIOE[15:0]	IO	189,192~206	General purpose input/output signals
Boot	ROMSWAP	I	99	Swap internal ROM area / external Flash ROM area
	BOOTSEL	I	100	Select boot bus width and direction (SMC/MMC)
System	nPOR	IS	42	Power on reset input. Schmitt level input with pull-up
	nPMWAKEUP	IS	41	Wake-up “on-key” input.
	nRESET	IO	43	Reset input
	PMBATOK	I	44	Main battery ok
Oscillator	RTCOSCIN	I	23	RTC oscillator input
	RTCOSCOUT	O	24	RTC oscillator output
	OSCIN	I	20	Main oscillator input
	OSCOUT	O	21	Main oscillator output
Digital Power / Ground	VDDCore	P	34,81,133,185	Core Vdd supply (3.3V)
	VSSCore	P	33,80,132,184	Core Vss supply
	VDD	P	8,50,67,88 106,120,135,147,160,172,191	IO Vdd supply (3.3V)
	VSS	P	7,49,66,87 105,119,134,146,159,170,190	IO Vss supply
JTAG	TCK	IU	16	JTAG boundary scan and debug test clock
	nTRST	ID	18	JTAG boundary scan and debug test reset
	TMS	IU	17	JTAG boundary scan and debug test mode select
	TDI	IU	15	JTAG boundary scan and debug test data input
	TDO	O	19	JTAG boundary scan and debug test data output
Test	nPLLENABLE	I	45	PLL enable input
	TESTSCAN	ID	207	Scan test mode enable
	SCANEN	ID	208	Scan chain pass enable
	nTEST	IU	46	Test mode select input

Table 2-2 External Signal Functions

2.3.2 Pin Specific Description

Key to PAD types : O (Output), I (Input), IO (Input / Output), A (Analog), C (Crystal Oscillator), OD (Output Open Drain), S (Input Schmitt level), D (Input Pull-Down), U (Input Pull-Up), 1x (CMOS PAD 0.8mA), 8mA (TTL PAD)

Pin	nTEST=1 && nPLLENABLE=0			PAD Direction	PAD Type	Drive Strength	Function
	Primary	GPIO En	Muxed Func.				
1	KSCANI[0]	GPIOA[0]		IO		1x	Matrix Keyboard Scan Bus Input
2	KSCANI[1]	GPIOA[1]		IO		1x	Matrix Keyboard Scan Bus Input
3	KSCANI[2]	GPIOA[2]		IO		1x	Matrix Keyboard Scan Bus Input
4	KSCANI[3]	GPIOA[3]		IO		1x	Matrix Keyboard Scan Bus Input
5	KSCANI[4]	GPIOA[4]		IO		1x	Matrix Keyboard Scan Bus Input
6	KSCANI[5]	GPIOA[5]		IO		1x	Matrix Keyboard Scan Bus Input
7	VSS						
8	VDD						
9	KSCANO[0]	GPIOA[6]		IO	OD	1x	Matrix Keyboard Scan Bus Output
10	KSCANO[1]	GPIOA[7]		IO	OD	1x	Matrix Keyboard Scan Bus Output
11	KSCANO[2]	GPIOA[8]		IO	OD	1x	Matrix Keyboard Scan Bus Output
12	KSCANO[3]	GPIOA[9]		IO	OD	1x	Matrix Keyboard Scan Bus Output
13	KSCANO[4]	GPIOA[10]		IO	OD	1x	Matrix Keyboard Scan Bus Output
14	KSCANO[5]	GPIOA[11]		IO	OD	1x	Matrix Keyboard Scan Bus Output
15	TDI			I	U		JTAG Data Input
16	TCK			I	U		JTAG Clock Input
17	TMS			I	U		JTAG Mode Sel.
18	nTRST			I	D		JTAG Reset
19	TDO			O		1x	JTAG Data Output
20	OSCIN			C			Main Oscillator In
21	OSCOUT						Main Oscillator Out
22	RTCVDD						RTC VDD
23	RTCOSCIN			A			RTC Oscillator In
24	RTCOSCOUT			A			RTC Oscillator Out
25	USBN			A			USB Transceiver Neg. Data I/O
26	USBP			A			USB Transceiver Pos. Data I/O
27	USBVSS						
28	USBVDD						
29	PLLVSS48M						
30	PLLVDD48M						
31	PLLVSS60M						
32	PLLVDD60M						
33	VSScore			A			Core VSS
34	VDDcore			A			Core VDD
35	ADCVDD						
36	ADCVREF			A			ADC Ref. Voltage
37	ADIN[0]			A			ADC Data Input
38	ADIN[1]			A			ADC Data Input
39	ADIN[2]			A			ADC Data Input
40	ADCVSS						
41	nPMWAKEUP			I	SU		Wake-up "On-Key" Input
42	nPOR			I	SU		Power On Reset Input
43	nRESET			IO	U	1x	Reset Input
44	PMBATOK			I	U		Main Battery OK
45	nPLLENABLE			I			PLL Enable Input
46	nTEST			I	U		Test Mode Sel. In
47	SCRST[0]	GPIOB[0]	UART0Rx	IO	OD	1x	SmartCard Reset Output (UART0 Rx)
48	SCIO[0]	GPIOB[1]	UART0Tx	IO	OD	1x	SmartCard Data I/O (UART0 Tx)
49	VSS						

Pin	nTEST=1 && nPLLENABLE=0			PAD Direction	PAD Type	Drive Strength	Function
	Primary	GPIO En	Muxed Func.				
50	VDD						
51	SCCLK[0]	GPIOB[2]		IO	OD	1x	SmartCard Clock Output
52	SCPRES[0]	GPIOB[3]		IO		1x	SmartCard Detect Input
53	SCRST[1]	GPIOB[4]	UART1Rx	IO	OD	1x	SmartCard Reset Output (UART1 Rx)
54	SCIO[1]	GPIOB[5]	UART1Tx	IO	OD	1x	SmartCard Data I/O (UART1 Tx)
55	SCCLK[1]	GPIOB[6]		IO	OD	1x	SmartCard Clock Output
56	SCPRES[1]	GPIOB[7]		IO		1x	SmartCard Detect Input
57	UART2Rx	GPIOB[8]		IO		1x	UART2 Serial Data Input
58	UART2Tx	GPIOB[9]		IO		1x	UART2 Serial Data Output
59	UART3Rx	GPIOB[10]		IO		1x	UART3 Serial Data Input
60	UART3Tx	GPIOB[11]		IO		1x	UART3 Serial Data Output
61	IrDA4Rx	GPIOB[12]	UART4Rx	IO		1x	IrDA Serial Data Input (UART4 Rx)
62	IrDA4Tx	GPIOB[13]	UART4Tx	IO		1x	IrDA Serial Data Output (UART4 Tx)
63	GPIOB14	GPIOB[14]		IO		1x	General Purpose I/O (To Deep Sleep source)
64	GPIOB15	GPIOB[15]		IO		1x	General Purpose I/O (HotSync wake-up source)
65	TouchXP	GPIOB[16]		IO		1x	Touch Screen Switch X-Pos. Out
66	VSS						
67	VDD						
68	TouchYP	GPIOB[17]		IO		1x	Touch Screen Switch Y-Pos. Out
69	TouchXN	GPIOB[18]		IO		1x	Touch Screen Switch X-Neg. Out
70	TouchYN	GPIOB[19]		IO		1x	Touch Screen Switch N-Neg. Out
71	nURING	GPIOB[20]		IO		1x	UART5 Ring Input (Wakeup to PMU)
72	nUDCD	GPIOB[21]		IO		1x	UART5 Data Carrier Detect In
73	nUDSR	GPIOB[22]		IO		1x	UART5 Data Set Ready Input
74	nUCTS	GPIOB[23]		IO		1x	UART5 Clear To Send Input
75	nURTS	GPIOB[24]		IO		1x	UART5 Request To Send Output
76	nUDTR	GPIOB[25]		IO		1x	UART5 Data terminal ready out
77	UART5Rx	GPIOB[26]		IO		1x	UART5 Serial Data Input
78	UART5Tx	GPIOB[27]		IO		1x	UART5 Serial Data Output
79	nSMCD	GPIOC[0]		IO		1x	SMC card detect In
80	VSScore						
81	VDDcore						
82	nSMRB	GPIOC[1]		IO		1x	SMC ready/busy in
83	SMALE	GPIOC[2]		IO		1x	SMC address latch enable Output
84	SMCLE	GPIOC[3]		IO		1x	SMC command latch enable Output
85	nSMCE	GPIOC[4]		IO		1x	SMC chip En Out
86	nSMRE	GPIOC[5]		IO		1x	SMC read En Out
87	VSS						
88	VDD						
89	nSMWE	GPIOC[6]		IO		1x	SMC write En Out
90	nSMWP	GPIOC[7]		IO		1x	SMC write Protect Output
91	SMD[0]	GPIOC[8]		IO		1x	SMC Bidir. Data I/O
92	SMD[1]	GPIOC[9]		IO		1x	SMC Bidir. Data I/O
93	SMD[2]	GPIOC[10]		IO		1x	SMC Bidir. Data I/O
94	SMD[3]	GPIOC[11]		IO		1x	SMC Bidir. Data I/O
95	SMD[4]	GPIOC[12]		IO		1x	SMC Bidir. Data I/O
96	SMD[5]	GPIOC[13]		IO		1x	SMC Bidir. Data I/O
97	SMD[6]	GPIOC[14]		IO		1x	SMC Bidir. Data I/O
98	SMD[7]	GPIOC[15]		IO		1x	SMC Bidir. Data I/O
99	ROMSWAP			I			Swap Internal ROM / External FlashROM
100	BOOTSEL			I			Select BootBus Width and Direction (SMC/MMC)
101	nRCS[0]			O		3x	ROM Chip Sel. Out
102	nRCS[1]			O		3x	ROM Chip Sel. Out

Pin	nTEST=1 && nPLLENABLE=0			PAD Direction	PAD Type	Drive Strength	Function
	Primary	GPIO En	Muxed Func.				
103	nRCS[2]	GPIOD[0]		IO		3x	ROM Chip Sel. Out
104	nRCS[3]	GPIOD[1]		IO		3x	ROM Chip Sel. Out
105	VSS						
106	VDD						
107	nROE			O		3x	ROM Out En Out
108	nRWE[0]			O		3x	ROM Write En Out
109	nRWE[1]			O		3x	ROM Write En Out
110	RD[0]		SD[0]	IO		8mA	ROM Bidir. Data I/O
111	RD[1]		SD[1]	IO		8mA	ROM Bidir. Data I/O
112	RD[2]		SD[2]	IO		8mA	ROM Bidir. Data I/O
113	RD[3]		SD[3]	IO		8mA	ROM Bidir. Data I/O
114	RD[4]		SD[4]	IO		8mA	ROM Bidir. Data I/O
115	RD[5]		SD[5]	IO		8mA	ROM Bidir. Data I/O
116	RD[6]		SD[6]	IO		8mA	ROM Bidir. Data I/O
117	RD[7]		SD[7]	IO		8mA	ROM Bidir. Data I/O
118	RD[8]		SD[8]	IO		8mA	ROM Bidir. Data I/O
119	VSS						
120	VDD						
121	RD[9]		SD[9]	IO		8mA	ROM Bidir. Data I/O
122	RD[10]		SD[10]	IO		8mA	ROM Bidir. Data I/O
123	RD[11]		SD[11]	IO		8mA	ROM Bidir. Data I/O
124	RD[12]		SD[12]	IO		8mA	ROM Bidir. Data I/O
125	RD[13]		SD[13]	IO		8mA	ROM Bidir. Data I/O
126	RD[14]		SD[14]	IO		8mA	ROM Bidir. Data I/O
127	RD[15]		SD[15]	IO		8mA	ROM Bidir. Data I/O
128	RA[0]		SA[0]	IO		8mA	ROM Address Out
129	RA[1]		SA[1]	IO		8mA	ROM Address Out
130	RA[2]		SA[2]	IO		8mA	ROM Address Out
131	RA[3]		SA[3]	IO		8mA	ROM Address Out
132	VSScore						
133	VDDcore						
134	VSS						
135	VDD						
136	RA[4]		SA[4]	IO		8mA	ROM Address Out
137	RA[5]		SA[5]	IO		8mA	ROM Address Out
138	RA[6]		SA[6]	IO		8mA	ROM Address Out
139	RA[7]		SA[7]	IO		8mA	ROM Address Out
140	RA[8]		SA[8]	IO		8mA	ROM Address Out
141	RA[9]		SA[9]	IO		8mA	ROM Address Out
142	SA10		SA[10]	O		8mA	ROM Address Out
143	RA[10]			IO		8mA	ROM Address Out
144	RA[11]		SA[11]	IO		8mA	ROM Address Out
145	RA[12]		SA[12]	IO		8mA	ROM Address Out
146	VSS						
147	VDD						
148	RA[13]		SA[13]	IO		8mA	ROM Address Out
149	RA[14]		SA[14]	IO		8mA	ROM Address Out
150	RA[15]			IO		8mA	ROM Address Out
151	RA[16]			O		8mA	ROM Address Out
152	RA[17]			O		8mA	ROM Address Out
153	RA[18]			O		8mA	ROM Address Out
154	RA[19]			O		8mA	ROM Address Out
155	RA[20]			O		8mA	ROM Address Out

Pin	nTEST=1 && nPLLENABLE=0			PAD Direction	PAD Type	Drive Strength	Function
	Primary	GPIO En	Muxed Func.				
156	RA[21]			O		8mA	ROM Address Out
157	RA[22]			O		8mA	ROM Address Out
158	RA[23]			O		8mA	ROM Address Out
159	VSS						
160	VDD						
161	DQML	GPIOD[2]		IO		8mA	SDRAM Lower Data Mask Output
162	DQMU	GPIOD[3]		IO		8mA	SDRAM Upper Data Mask Output
163	nSWE	GPIOD[4]		IO		8mA	SDRAM Write Enable Output
164	nCAS	GPIOD[5]		IO		8mA	SDRAM Column Address Select Out
165	nRAS	GPIOD[6]		IO		8mA	SDRAM Row Address Select Out
166	nSCS[0]	GPIOD[7]		IO		8mA	SDRAM Chip Select Output
167	nSCS[1]	GPIOD[8]		IO		8mA	SDRAM Chip Select Output
168	SCKE[0]	GPIOD[9]		IO		8mA	SDRAM Clock Enable Output
169	SCKE[1]	GPIOD[10]		IO		8mA	SDRAM Clock Enable Output
170	VSS						
171	SCLK			IO		8mA	SDRAM Clock I/O (For FBCLK)
172	VDD						
173	LLP	GPIOD[11]		IO		1x	LCD Line Pulse
174	LAC	GPIOD[12]		IO		1x	LCD AC Bias
175	LBLEN	GPIOD[13]		IO		1x	LCD Back-Light En
176	LCP	GPIOD[14]		IO		1x	LCD Clock Pulse
177	LFP	GPIOD[15]		IO		1x	LCD Frame Pulse
178	LCDEN	GPIOD[16]		IO		1x	LCD Display En
179	LD[0]	GPIOD[17]		IO		1x	LCD Data Bus
180	LD[1]	GPIOD[18]		IO		1x	LCD Data Bus
181	LD[2]	GPIOD[19]		IO		1x	LCD Data Bus
182	LD[3]	GPIOD[20]		IO		1x	LCD Data Bus
183	LD[4]	GPIOD[21]		IO		1x	LCD Data Bus
184	VSScore						
185	VDDcore						
186	LD[5]	GPIOD[22]		IO		1x	LCD Data Bus
187	LD[6]	GPIOD[23]		IO		1x	LCD Data Bus
188	LD[7]	GPIOD[24]		IO		1x	LCD Data Bus
189	SPIRx[0]	GPIOE[0]		IO		1x	SPI Data In
190	VSS						
191	VDD						
192	SPITx[0]	GPIOE[1]		IO		1x	SPI Data Output
193	nSPICS[0]	GPIOE[2]		IO		1x	SPI Chip Select
194	SPICLK[0]	GPIOE[3]		IO		1x	SPI Clock Output
195	SPIRx[1]	GPIOE[4]		IO		1x	SPI Data In
196	SPITx[1]	GPIOE[5]		IO		1x	SPI Data Output
197	nSPICS[1]	GPIOE[6]		IO		1x	SPI Chip Select
198	SPICLK[1]	GPIOE[7]		IO		1x	SPI Clock Output
199	2WSICLK	GPIOE[8]		IO	OD	1x	2WSI Clock I/O
200	2WSIDAT	GPIOE[9]		IO	OD	1x	2WSI Data I/O
201	TIMER[0]	GPIOE[10]		IO		1x	Timer Data Output
202	TIMER[1]	GPIOE[11]		IO		1x	Timer Data Output
203	TIMER[2]	GPIOE[12]		IO		1x	Timer Data Output
204	TIMER[3]	GPIOE[13]		IO		1x	Timer Data Output
205	PWM[0]	GPIOE[14]		IO		1x	PWM Data Output
206	PWM[1]	GPIOE[15]		IO		1x	PWM Data Output
207	TESTSCAN			I	D		TEST Signal Input
208	SCANEN			I	D		TEST Signal Input

3 ARM720T MACROCELL

3.1 ARM720T Macrocell

For details of the ARM720T, please refer to the **ARM720T Data Sheet** (DDI 0087).

4 MEMORY MAP

There are five main memory map divisions, outlined in Table 4-1 Top-level address map

Function	Base Address (Hex)	Size	Description
Internal Boot ROM / External Static Memory (ROMSWAP = 0)	0x0000.0000	2 Kbytes	**Internal Boot ROM**
	0x0000 0800	--	Reserved
	0x0100 0000	16 Mbytes	External Static Memory chip select 1
	0x0200 0000	16 Mbytes	External Static Memory chip select 2
	0x0300 0000	16 Mbytes	External Static Memory chip select 3
	0x0400 0000	--	Reserved
	0x1000.0000	16 Mbytes	External Static Memory **chip select 0**
	0x1100 0000	--	Reserved
Internal Boot ROM / External Static Memory (ROMSWAP = 1)	0x0000.0000	16 Mbytes	External Static Memory **chip select 0**
	0x0100 0000	16 Mbytes	External Static Memory chip select 1
	0x0200 0000	16 Mbytes	External Static Memory chip select 2
	0x0300 0000	16 Mbytes	External Static Memory chip select 3
	0x0400 0000	--	Reserved
	0x1000.0000	2 Kbytes	**Internal Boot ROM**
	0x1000 0800	--	Reserved
Internal SRAM	0x3000 0000	-	Reserved
	0x3FFF.E000	8 Kbytes	Internal SRAM
External SDRAM	0x4000.0000	32 Mbytes	SDRAM chip select 0
	0x4200.0000	32 Mbytes	SDRAM chip select 1
	0x4400.0000	--	SDRAM mode register chip 0
	0x4600.0000	--	SDRAM mode register chip 1
	0x4800 0000	--	Reserved
Peripherals	0x8000.0000	--	ASB, APB Peripherals
	0x8006 3000	--	Reserved

Table 4-1 Top-level address map

When a ROMSWAP pin is set low, if a BOOTSEL pin is set high, the SMC(Nand Flash) can be used by connecting to EBI, and if a BOOTSEL pin is set low, the MMC can be used by connecting to SSI 0.

When a ROMSWAP pin is set high, if a BOOTSEL pin is set high, support External 16-bit Memory, and if a BOORSEL pin is set low, support External 8-bit Memory booting,

The external Static Memory has an address space of 64Mbytes that is split equally between four external Static Memory chip select. Actual address range for each chip select is 16Mbytes with 24 external address signals.

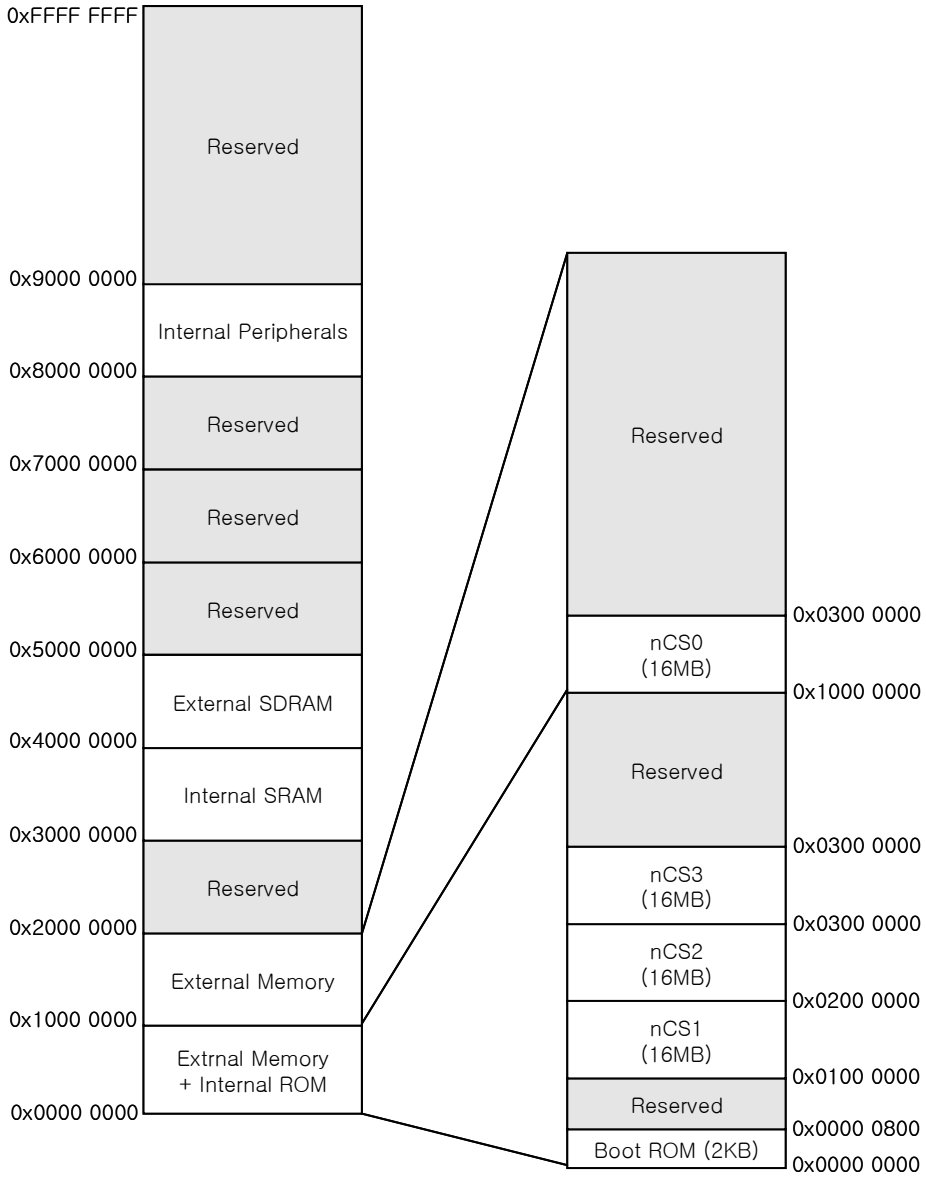


Figure 4-1. Internal Boot ROM / External Static Memory Map (ROMSWAP=0)

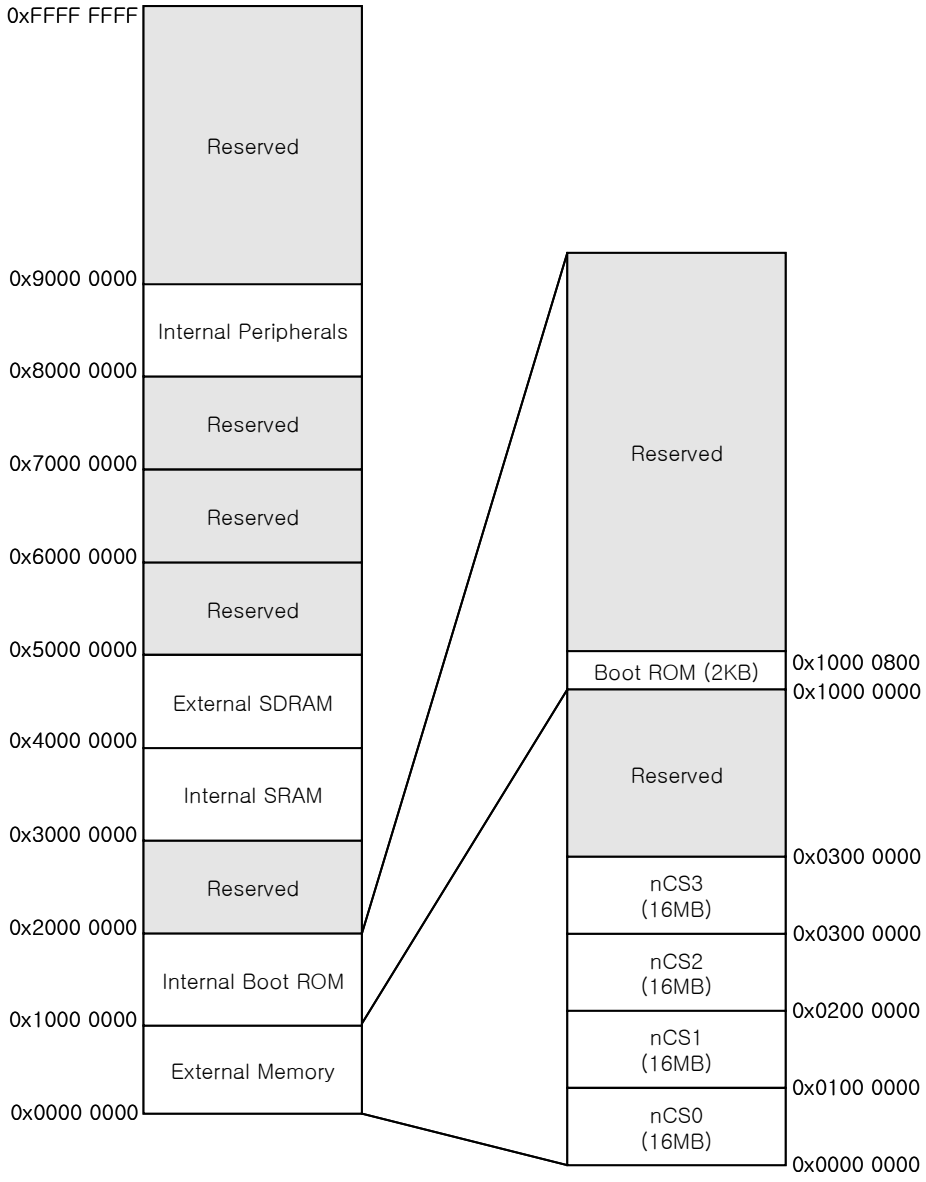


Figure 4-2. Internal Boot ROM / External Static Memory Map (ROMSWAP=1)

There is a maximum of 64Mbytes of SDRAM space. The mode registers (in the SDRAM) are programmed by reading from 64Mbyte address space immediately above the SDRAM (over 0x4400.0000).

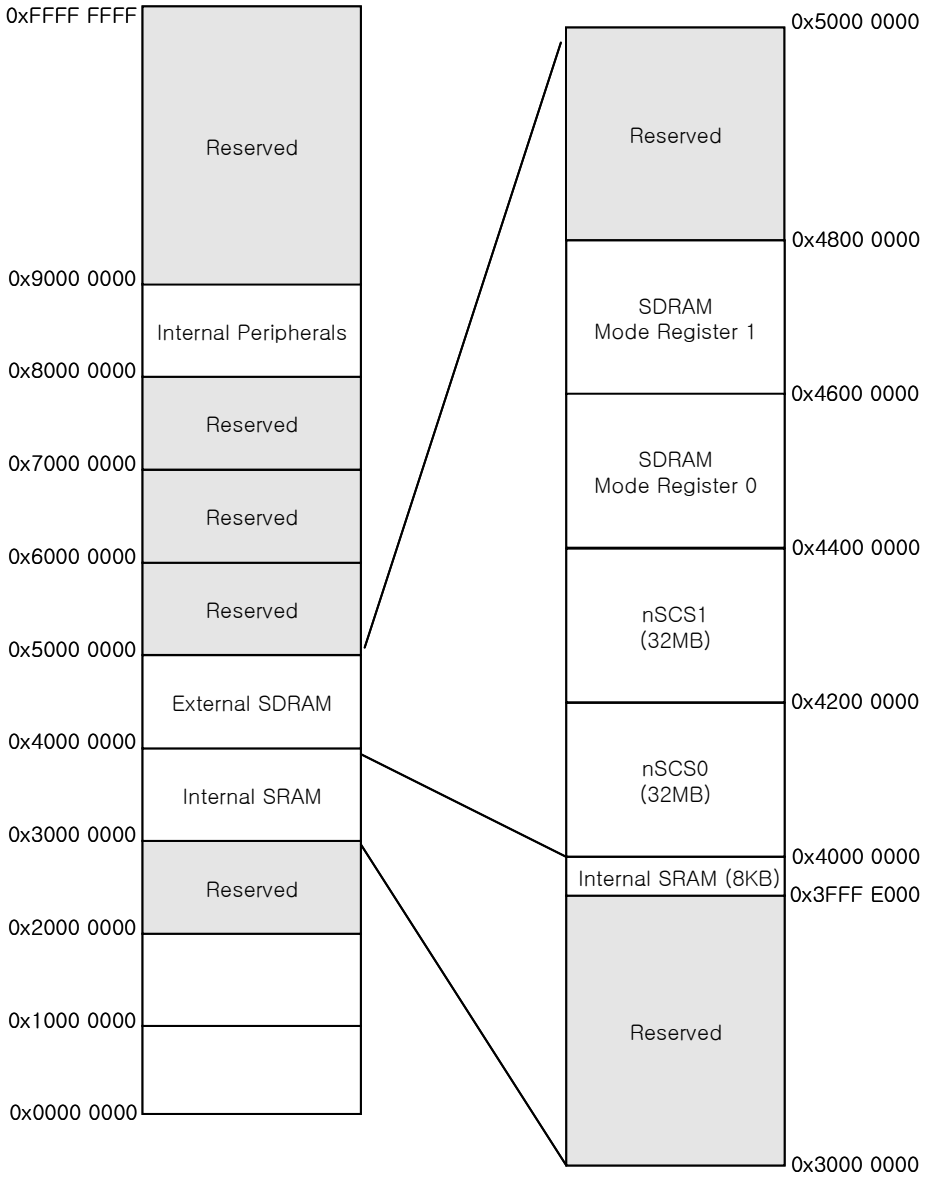


Figure 4-3. Internal SRAM / External SDRAM Memory Map

The peripheral address space is subdivided into two main areas: those on the ASB, the APB. The base address for the peripherals is given in Table 4-2: Peripherals base addresses.

Function	Base Address (Hex)	Name	Description
ASB Peripherals	0x8000.0000	SDRAMC Base	SDRAM Controller
	0x8001.0000	PMU Base	PMU
	0x8002.0000	ExtFLASHC Base	External Bus Interface
	0x8003.0000	Reserved	
	0x8004.0000	ARMTest Base	To ARM CPU
APB Peripherals	0x8005.0000	INTC Base	Interrupt Controller
	0x8005.1000	USB Base	USB Controller
	0x8005.2000	LCD Base	LCD Controller
	0x8005.3000	ADC Base	ADC Interface
	0x8005.4000	UART0 Base	UART0 (SCI0)
	0x8005.5000	UART1 Base	UART1 (SCI1)
	0x8005.6000	UART2 Base	UART2
	0x8005.7000	UART3 Base	UART3
	0x8005.8000	UART4 Base	UART4 (SIR)
	0x8005.9000	UART5 Base	UART5 (Modem)
	0x8005.A000	SSI0 Base	SSI 0
	0x8005.B000	SSI1 Base	SSI 1
	0x8005.C000	SMC Base	SMC
	0x8005.D000	TIM Base	Timerx4 / PWMx2
	0x8005.E000	WDT Base	WDT
	0x8005.F000	RTC Base	RTC
	0x8006.0000	2WSBI Base	2WSBI
	0x8006.1000	KBD Base	Matrix Keyboard
	0x8006.2000	GPIO Base	GPIO

Table 4-2 Peripherals Base Addresses

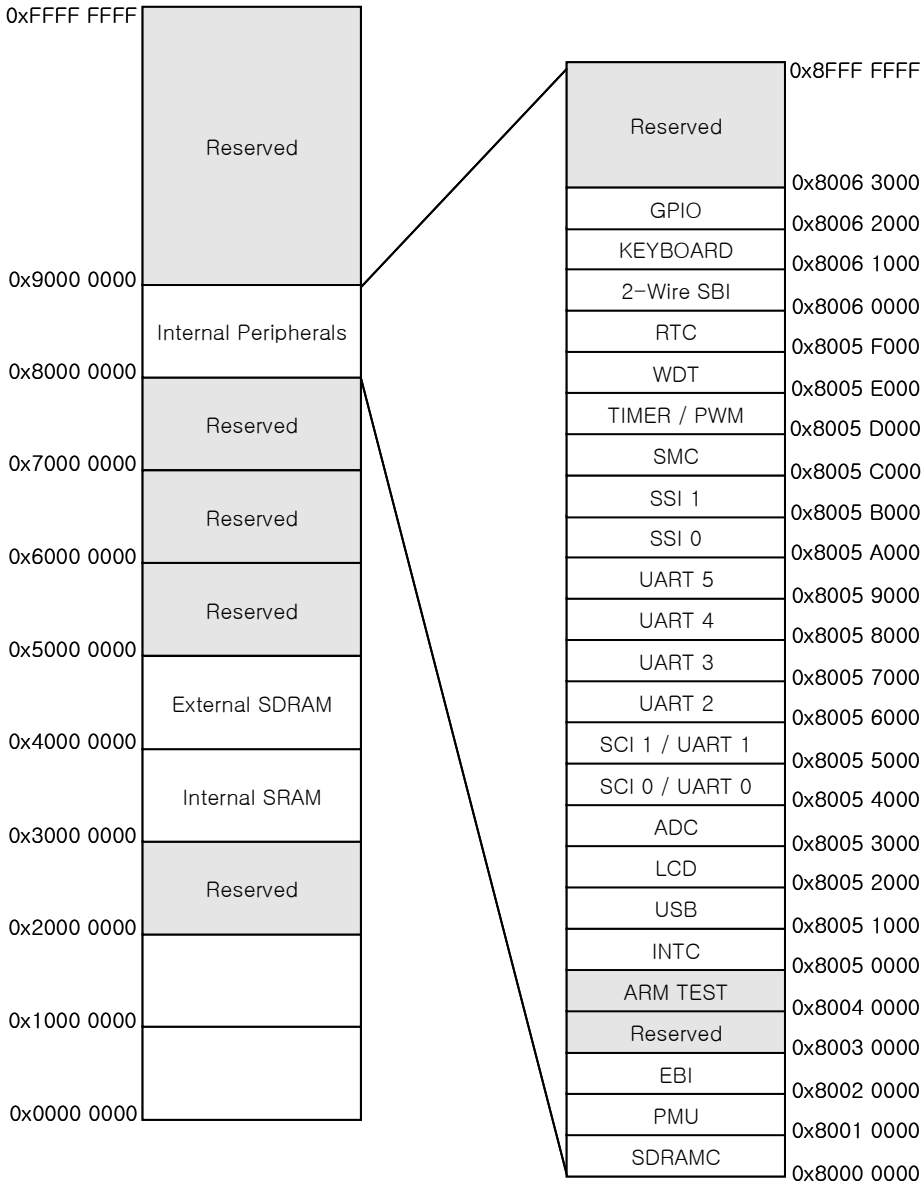


Figure 4-4. Peripherals Address Map

5 Internal Boot ROM

HMS30C7210 has Internal boot ROM. Boot ROM's role load user's image code from external EBI NAND Flash / MMC connected SSI to Internal SRAM (8kbytes) and jumps to user's image code at Internal SRAM.

Like previous explanation, HMS30C7210 has two internal booting modes [NAND / MMC]. Each mode setting is decided as two external pin [ROMSWAP (99), BOOTSEL (100)] states.

Initially contents of copied image code from NAND / MMC to Internal SRAM are SDRAM initialization routine, copy routine from boot loader or executive binary image to SDRAM and jump to SDRAM starting address.

5.1 Hardware Setting

HMS30C7210 can boot internal Boot ROM [ROMSWAP=0] and external memory [ROMSWAP=1]. If it is set to internal Boot ROM, it can be NAND [BOOTSEL=1]/MMC [BOOTSEL=0] boot mode setting.

ROMSWAP	BOOTSEL	BOOT MODE
LOW (=0)	LOW (=0)	MMC
	HIGH (=1)	NAND
HIGH (=1)	LOW (=0)	8 BIT
	HIGH (=1)	16 BIT

Table 5-1. Pin Configuration

5.2 Software Setting

If it is decided on internal booting mode at H/W, you can program code to copy from NAND/MMC to Internal SRAM. Internal SRAM has 8Kbytes. So size of code, data and stack don't have over 8Kbytes. Presently code and data size is Max. 7.5Kbytes, stack size can use 0.5Kbytes. Following figure show S/W flows.

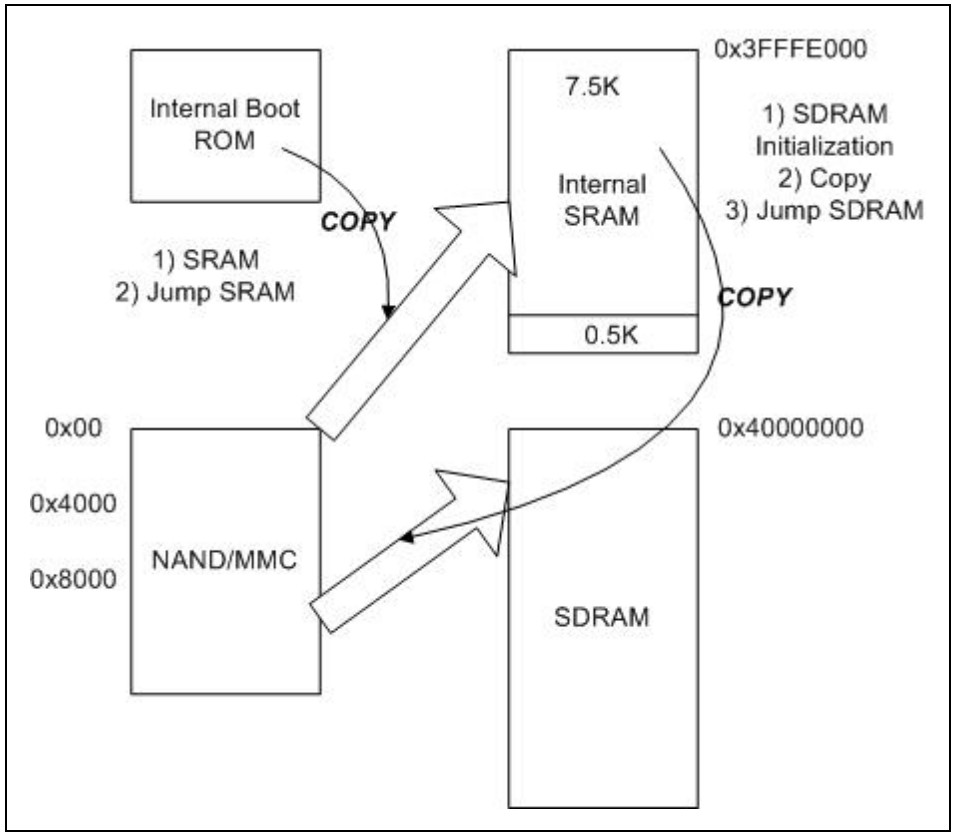


Figure 5-1. Software Boot Flows

- After power on, internal boot ROM copies executive code (NAND/MMC address 0x00) from NAND/MMC to internal SRAM.
- Internal SRAM code from NAND/MMC has SDRAM controller initialization routine, copy other executive binary code from NAND/MMC to SDRAM and jump to SDRAM start address.

Used NAND/MMC map is following table.

Address	Discription
0x0000 0000 ~ 0x0000 3FFF	Boot 0 (no change) Iram2dram.axf
0x0000 4000 ~ 0x0000 7FFF	Boot 1 (no change) Iram2dram.axf
0x0000 8000 ~	Binary image (SDRAM no initialization)

Table 5-2. NAND / MMC Map

There are Boot0, Boot1 area in NAND/MMC. If Boot0 don't operate correctly, boot ROM uses Boot1 area in internal Boot Program. Also user's binary program don't initialize SDRAM init routine.

6 PMU & PLL

The HMS30C7210 is designed primarily for smart card reader and other portable computing applications. Therefore there are 4 operating modes to reduce power consumption and extend battery life.

- RUN - normal operation (typically used for CPU-intensive tasks).
- SLOW - half-speed operation used in the application demanding low computing power.
- IDLE - where the CPU operation is halted but peripherals continue their operations (such as screen refresh, or serial communications).
- SLEEP & DEEP SLEEP - This mode will be perceived as `off' by the user, but the SDRAM contents are preserved and only the real-time clock is running.

The transition between these modes is controlled by the PMU (see also Section 6.4 Power management). The PMU is an ASB slave unit to allow the CPU to access (read/write) its control registers, and is an ASB master unit to provide the mechanism for stopping the ARM core's internal clock.

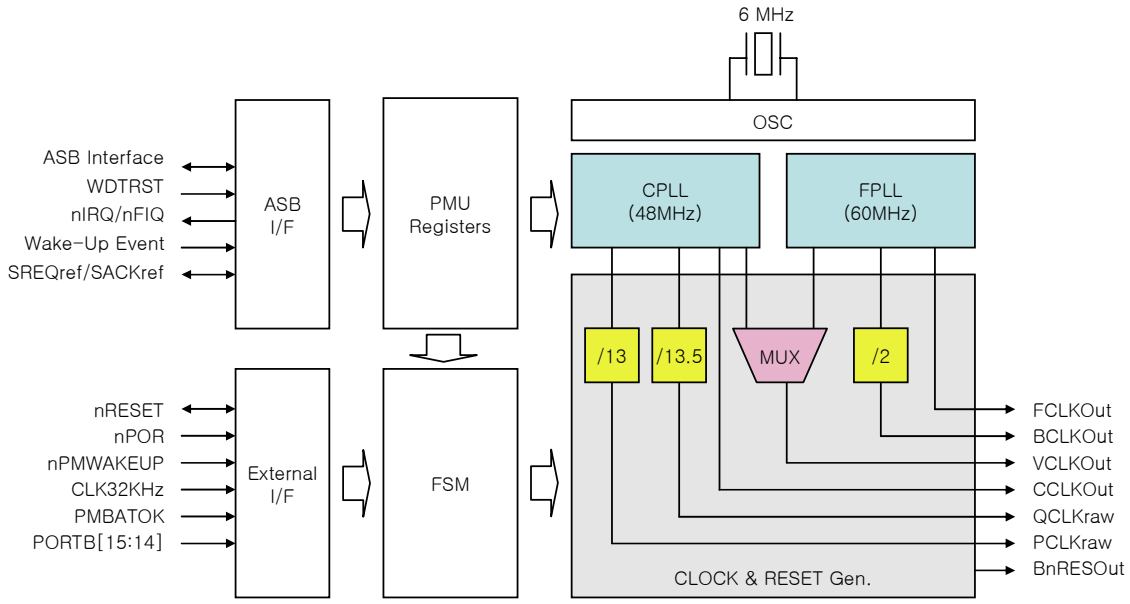


Figure 6-1. PMU Block Diagram

6.1 External Signals

Pin Name	Type	Description
nPOR	IS	Power on reset input. Schmitt level input with pull-up
nPMWAKEUP	IS	Wake-up “on-key” input.
nRESET	I/O	Reset input
PMBATOK	I	Main battery ok
GPIOB[14]	I	To-deep-sleep input from GPIOB[14]
GPIOB[15]	I	HotSync request from PORTB[15]

Refer to Figure 2-1. 208 Pin diagram.

6.2 Registers

Address	Name	Width	Default	Description
0x8001.0000	PMUMR	4	0x0	PMU Mode Register
0x8001.0010	PMUIDR	32	0x0072100	PMU ID Register
0x8001.0020	PMURSR	27	-	PMU Reset/Status Register
0x8001.0028	PMUCCR	16	0x2F	PMU Clock Control Register
0x8001.0030	PMUDCTR	18	-	PMU Debounce Counter Test Register
0x8001.0038	PMUTR	8	0x0	PMU Test Register

Table 6-1. PMU Register Summary

6.2.1 PMU Mode Register (PMUMR)

This read/write register is to change from RUN mode or SLOW mode into a different mode. The PMU mode encoding is shown below. The register can only be accessed in RUN mode or SLOW mode (these are the only modes in which the processor is active). Therefore, the processor will never be able to read values for modes other than mode 0x00 and mode 0x01. A test controller may read other values as long as clocks are enabled in every PMU mode by the bit 8 of the PMU Debounce Counter Test Register (PMUDCTR). For more information, please refer to 6.2.5 .

0x80010000

31	...	3	2	1	0
Reserved	Reserved	WAKEUP CTRL	MODESEL[2:0]		

Bits	Type	Function
31:4	-	Reserved
3	R/W	Wake-up Control 0 = Prevent the PMU from exiting the DEEP SLEEP mode when the pin PMBATOK is inactive. 1 = Allow the PMU to exit the DEEP SLEEP mode even if the pin PMBATOK is inactive.
2:0	R/W	MODE Selection In reads, the read value is the current PMU mode. In writes, the write value is the target mode at which the PMU will arrive eventually.

Value	PMU mode encoding
0x04	Initialization mode
0x01	RUN mode
0x00	SLOW mode
0x02	IDLE mode
0x03	SLEEP mode
0x07	DEEP SLEEP mode

Note All other values in the above table are undefined.

6.2.2 PMU ID Register (PMUID)

This read-only register returns a unique chip revision ID. Revision 0 of the HMS30C7210 device (the first revision) will return the constant value 0x00721000.

0x80010010

31	...	0
0x00721000		

6.2.3 PMU Reset/Status Register (PMURSR)

This read/write register provides information on power-on reset and the PLL status as well as wakeup and interrupt events. The PMURSR also provides software-initiated warm reset, and wakeup and interrupt masking The allocation is shown in the following two tables: PMURSR Bits. The event bits in this register are `sticky' bits. For a definition of a sticky bit, please refer to 5.2.3 Wake-up Debounce and Interrupt. Generally, this register will be read each time the ARM exits from reset mode, so that the ARM can identify what event has caused it to exit from reset mode.

0x80010020

31	30	29	28	27	26	25	24
Reserved	Reserved	Reserved	Reserved	Reserved	WARM RESET	HOTSYNC DBEN	WARM RST DBEN
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PFAIL DBEN	MRING DBEN	ONKEY DBEN	HOTSYNC WAKEN	WARM RST WAKEN	RTC WAKEN	MRING WAKEN	ONKEY WAKEN
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
HOTSYNC INTREN	PFAIL INTREN	RTC INTREN	MRING INTREN	ONKEY INTREN	HOTSYNC EVT	WDT RST EVT	WARM RST EVT
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PFAIL EVT	RTC EVT	MRING EVT	ONKEY EVT	FPLL Un-LOCK	CPLL Un-LOCK	DEEP EVT	POR EVT

Bits	Type	Function
31:27	-	Reserved
26	W	Software Warm Reset. Writing a `1' to this bit causes nRESET and the ASB system reset to be asserted. Writing a `0' to this bit has no effect.
25	R/W	Debounce Enable of Hot Sync Event. 0 = Disable debouncing of hot sync event. 1 = Enable debouncing of hot sync event (default).
24	R/W	Debounce Enable of Warm Reset Event. 0 = Disable debouncing of warm reset event. 1 = Enable debouncing of warm reset event (default).
23	R/W	Debounce Enable of Power Fail Event. 0 = Disable debouncing of power fail event. 1 = Enable debouncing of power fail event (default).
22	R/W	Debounce Enable of Modem Ring Indicator Event. 0 = Disable debouncing of modem ring indicator event. 1 = Enable debouncing of modem ring indicator event (default).
21	R/W	Debounce Enable of On Key Event. 0 = Disable debouncing of on key event. 1 = Enable debouncing of on key event (default).
20	R/W	Wake-up Enable of Hot Sync Event. 0 = Disable CPU wake-up due to hot sync event (default). 1 = Enable CPU wake-up due to hot sync event.
19	R/W	Wake-up Enable of External Warm Reset Event. 0 = Disable CPU wake-up due to external warm reset event (default). 1 = Enable CPU wake-up due to external warm reset event.
18	R/W	Wake-up Enable of RTC Alarm Event 0 = Disable CPU wake-up due to RTC alarm event (default). 1 = Enable CPU wake-up due to RTC alarm event.
17	R/W	Wake-up Enable of Modem Ring Indicator Event 0 = Disable CPU wake-up due to modem ring indicator event (default). 1 = Enable CPU wake-up due to modem ring indicator event.
16	R/W	Wake-up Enable of On Key Event. 0 = Disable CPU wake-up due to on key event (default). 1 = Enable CPU wake-up due to on key event.

15	R/W	Interrupt Mask of Hot Sync Event. 0 = Disable PMU interrupt due to hot sync event (default). 1 = Enable PMU interrupt due to hot sync event.
14	R/W	Interrupt Mask of Power Fail Event. 0 = Disable PMU interrupt due to power fail event (default). 1 = Enable PMU interrupt due to power fail event.
13	R/W	Interrupt Mask of RTC Alarm Event 0 = Disable PMU interrupt due to RTC alarm event (default). 1 = Enable PMU interrupt due to RTC alarm event.
12	R/W	Interrupt Mask of Modem Ring Indicator Event 0 = Disable PMU interrupt due to modem ring indicator event (default). 1 = Enable PMU interrupt due to modem ring indicator event.
11	R/W	Interrupt Mask of On Key Event 0 = Disable PMU interrupt due to on key event (default). 1 = Enable PMU interrupt due to on key event.
10	R/W	Hot Sync Event (IRQ from GPIOB[15]) In reads, 0 = No hot sync event has occurred since last cleared; 1 = Hot sync event has occurred since last cleared. In writes, writing a `1' to this bit causes it to be cleared. When set, a PMU interrupt is generated if PMURSR[15] (HOTSYNC INTREN) is also set.
9	R/W	Watch Dog Timer Reset Event (a kind of warm reset) In reads, 0 = No watch dog timer reset event has occurred since last cleared; 1 = watch dog timer reset event has occurred since last cleared. In writes, writing a `1' to this bit causes it to be cleared.
8	R/W	Warm (External or Software) Reset Event In reads, 0 = No warm reset event has occurred since last cleared; 1 = Warm reset event has occurred since last cleared. In writes, writing a `1' to this bit causes it to be cleared.
7	R/W	Power Fail Event (Adaptor Not OK, Low PMBATOK) In reads, 0 = No power fail event has occurred since last cleared; 1 = Power fail event has occurred since last cleared. In writes, writing a `1' to this bit causes it to be cleared. When set, a PMU interrupt is generated if PMURSR[14] (PFAIL INTREN) is also set.
6	R/W	RTC (Real Time Clock) Alarm Event In reads, 0 = No RTC alarm event has occurred since last cleared; 1 = RTC alarm event has occurred since last cleared. In writes, writing a `1' to this bit causes it to be cleared. When set, a PMU interrupt is generated if PMURSR[13] (RTC INTREN) is also set.
5	R/W	Modem Ring Indicator Event (Low nMRING) In reads, 0 = No modem ring indicator event has occurred since last cleared; 1 = Modem ring indicator event has occurred since last cleared. In writes, writing a `1' to this bit causes it to be cleared. When set, a PMU interrupt is generated if PMURSR[12] (MRING INTREN) is also set.
4	R/W	On Key Event (Low nPMWAKEUP) In reads, 0 = No on key event has occurred since last cleared; 1 = On key event has occurred since last cleared. In writes, writing a `1' to this bit causes it to be cleared. When set, a PMU interrupt is generated if PMURSR[11] (ONKEY INTREN) is also set
3	R/W	FCLK PLL Un-Lock Event In reads, 0 = FCLK PLL has been locked since last cleared; 1 = FCLK PLL has fallen out of lock since last cleared. In writes, writing a `1' to this bit causes it to be cleared.
2	R/W	CCLK PLL Un-Lock Event In reads,

		0 = CCLK PLL has been locked since last cleared; 1 = CCLK PLL has fallen out of lock since last cleared. In writes, writing a `1' to this bit causes it to be cleared.
1	R/W	DEEP SLEEP Event In reads, 0 = PMU has not entered the DEEP SLEEP mode since last cleared; 1 = PMU has entered the DEEP SLEEP mode since last cleared. In writes, writing a `1' to this bit causes it to be cleared.
0	R/W	Power-on Reset Event In reads, 0 = No power-on reset event has occurred since last cleared; 1 = Power-on reset event has occurred since last cleared. In writes, writing a `1' to this bit causes it to be cleared.

6.2.4 PMU Clock Control Register (PMUCCR)

This register is used to control the two PLLs (FCLK and CCLK PLLs) and three main clocks (FCLK, CCLK and VCLK). The six bits of the PMUCCR are used to compose the input pins of the FCLK PLL for frequency selection and thus define the frequency of the FCLK. The default value (after power-on reset) for this register is 0x2F.

0x80010028

15	14	13	12	11	10	9	8
CCLK ENABLE	VCLK ENABLE	VCLK SEL	Reserved	Reserved	Reserved	Reserved	Reserved
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FCLK MUTE CTRL	FFREQ UPDATE CTRL	FCLK PLL FREQ[5:0]					

Bits	Type	Function
31:16	-	Reserved
15	R/W	CCLK Enable 0 = The CCLK is disabled. 1 = The CCLK is enabled.
14	R/W	VCLK Enable 0 = The VCLK is disabled. 1 = The VCLK is enabled.
13	R/W	VCLK Select 0 = The VCLK uses the clock source of the FCLK as its clock source. 1 = The VCLK uses the clock source of the CCLK as its clock source.
12:8	R/W	Reserved
7	R/W	FCLK Mute Control 0 = The FCLK is muted when the FCLK PLL is out of lock. 1 = The FCLK is only muted during power-on reset. Subsequent unlock condition does not mute the FCLK. Allows dynamic changes to the clock frequency without halting execution. Care: this only will be legal if FCLK PLL is under-damped (i.e. will not exhibit overshoot in its lock behavior).
6	R/W	FCLK PLL Frequency Update Control 0 = The written value to the bits[5:0] of the PMUCCR is transferred to a 6-bit temporary register, not the PMUCCR[5:0]. After that, if the CPU enters the DEEP SLEEP mode, the value in the temporary register is transferred to the bits[5:0] of the PMUCCR and thus the frequency control of the FCLK PLL is updated. And then, the FCLK PLL comes to life with the new frequency when the CPU exits the DEEP SLEEP mode. 1 = The PMUCCR[5:0] and the frequency of the FCLK PLL is updated immediately after writing to the PMUCCR[5:0].
5:0	R/W	FCLK PLL Frequency Control **Value** **Frequency** Bit[5]= 0 0x0C 21 MHz 0x0D 22.5 MHz 0x0E 24 MHz 0x0F 25.5 MHz 0x10 27 MHz 0x11 28.5 MHz 0x12 30 MHz 0x13 31.5 MHz 0x14 33 MHz 0x15 34.5 MHz 0x16 36 MHz 0x17 37.5 MHz 0x18 39 MHz 0x19 40.5 MHz 0x1A 42 MHz

0x1B	43.5 MHz
0x1C	45 MHz
0x1D	46.5 MHz
0x1E	48 MHz

UNPREDICTABLE otherwise

Value	**Frequency**
Bit[5] = 1	
0x25	21 MHz
0x26	24 MHz
0x27	27 MHz
0x28	30 MHz
0x29	33 MHz
0x2A	36 MHz
0x2B	39 MHz
0x2C	42 MHz
0x2D	45 MHz
0x2E	48 MHz
0x2F	51 MHz (default)
0x30	54 MHz
0x31	57 MHz
0x32	60 MHz
0x33	63 MHz
0x34	66 MHz
0x35	69 MHz
0x36	72 MHz
0x37	75 MHz
0x38	78 MHz
0x39	81 MHz
0x3A	84 MHz
0x3B	87 MHz
0x3C	90 MHz
0x3D	93 MHz
0x3E	96 MHz

UNPREDICTABLE otherwise

IF BIT 6 (FCLK Frequency Update Control) is `0'

When the CPU core writes to bits[5:0] of this register, these bits are stored in a temporary buffer, which is not transferred to the input pins of the FCLK PLL until the next time the CPU enters the DEEP SLEEP mode. This means that for a new value to take effect, it is necessary for the device to enter the DEEP SLEEP mode first.

IF BIT 6 (FCLK Frequency Update Control) is `1'

The first effect that writing a new value to bits [5:0] will have is that the FCLK PLL will go out of lock, and the clock control circuit will immediately inhibit FCLK and BCLK, without first verifying that SDRAM operations have completed.

6.2.5 PMU Debounce Counter Test Register (PMUDCTR)

0x80010030

23	22	21	20	19	18	17	16
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	RST DB CTRL	CLK32K EXTSEL
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DBGPIOA	DBSEL[3:0]				CLK15	CLK31	CLK62
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CLK125	CLK500	CLK1K	CLK2K	CLK4K	DBCNT[2:0]		

Bits	Type	Function Read Write
31:18	-	Reserved
17	R	Warm Reset Debounce Time Control This is set to the same value of the pin TDI (input with a pull-up resistor) during power-on reset. 0 = Debouncing time of warm reset (or power on reset) is short since the debounce counter uses 16-KHz clock. 1 = Debouncing time of warm reset (or power on reset) is long since the debounce counter uses 15.625-Hz clock (default).
16	R/W	External CLK32K Select 0 = Use the RTC clock as the 32-KHz input clock. 1 = Use the external clock (from the TBFCLK pin) as the 32-KHz input clock in the TIC test mode (nTEST = '0') to test the frequency-division circuit making the debounce clock.
15	R/W	GPIOA Debounce Counter Select 0 = Select a debounce counter other than GPIOA debounce counters as the bits[2:0] of PMUDCTR in read. 1 = Select one among GPIOA debounce counters as the bits[2:0] of the PMUDCTR in read.
14:11	R/W	Debounce Counter Select When DBGPIOA (PMUDCTR[15]) is reset **Value** **Function** 0x0 On key event 0x1 Modem ring indicator event 0x2 Power fail event 0x3 Warm reset event 0x4 Hot sync event (GPIOB[15]) 0x5 ToDeepSleep event (GPIOB[14]) UNPREDICTABLE otherwise. When DBGPIOA (PMUDCTR[15]) is set, **Value** **Function** 0x0 GPIOA[0] 0x1 GPIOA[1] 0x2 GPIOA[2] 0x3 GPIOA[3] 0x4 GPIOA[4] 0x5 GPIOA[5] 0x6 GPIOA[6] 0x7 GPIOA[7] 0x8 GPIOA[8] 0x9 GPIOA[9] 0xA GPIOA[10] 0xB GPIOA[11]
10	R	15.625-Hz CLK 15.625-Hz debouncing CLK derived from the RTC clock. This clock is used to debounce on key, warm reset, hot sync, Todeepsleep events and GPIOA values.
9	R	31.25-Hz CLK 31.25-Hz CLK derived from the RTC clock.

8	R	62.5-Hz CLK 62.5-Hz CLK derived from the RTC clock.
7	R	125-Hz CLK 125-Hz CLK derived from the RTC clock.
6	R	500-Hz CLK 500-Hz CLK derived from the RTC clock.
5	R	1-KHz CLK 1-KHz CLK derived from the RTC clock.
4	R	2-KHz CLK 2-KHz CLK derived from the RTC clock.
3	R	4-KHz CLK 4-KHz CLK derived from the RTC clock.
2:0	R	Selected Debounce Counter Debounce counter selected by the bits[15:11] of the PMUDCTR.

In order that the debounce counters (which would normally be clocked at 250 Hz or 15.625 Hz) may be independently exercised and observed, the counters may be triggered and observed using the above registers. **This register is for the test purpose only and not required in normal use.**

6.2.6 PMU Test Register (PMUTR)

This register is used to control the PMU operation in the TIC test mode. **This register is for test purpose only and not required in normal use.**

0x80010038

7	6	5	4	3	2	1	0
CLK BYPASS	NPLLEN[1:0]		PQCLK BYPASS CTRL	CCLK BYPASS SELECT	BCLK BYPASS	CLK ENFORCE	PMUTEST

Bits	Type	Function
31:8	-	Reserved
7	R	Clock Bypass Enable Read value is the same value of the input pin nPLLENABLE If this value is '1', the clocks (of the system, USB, LCD, etc.) are provided using external bypass clocks from the pins. Normal, this value is '0', and the clocks are made using PLL output clocks.
6:5	R	Intermediate PLL Enable. When the bit[7] and bit[6] (NPLLEN[1]) of this register are both zero, the PLLs (FPLL and CPLL) are enabled.
4	R/W	PCLK/QCLK Bypass Control 0 = When nPLLENABLE is '1', the PCLK and QCLK are directly bypassed from pin pads. 1 = When nPLLENABLE is '1', the PCLK and QCLK are provided by a frequency divider that uses the bypass clock of the CCLK as its source clock.
3	R/W	Bypass Clock Select for the CCLK 0 = CCLK uses TBQFCLK, as bypass clocks used when the nPLLENABLE pin is reset. 1 = CCLK uses TBCCLK, as bypass clocks used when the nPLLENABLE pin is reset, in the TIC test mode.
2	R/W	BCLK Bypass Enable 0 = BCLK is derived from the FCLKQ (clock lagging behind the FCLK by 90 degrees). 1 = BCLK is derived from the bypass clock TBBCLK in the TIC test mode.
1	R/W	Clock Enforce 0 = The FCLK, BCLK, VCLK and CCLK are disabled in the DEEP SLEEP mode (normal). 1 = The FCLK, BCLK, VCLK and CCLK are enabled regardless of the PMU states (even in the DEEP SLEEP mode) in the TIC test mode.
0	R/W	PMUTEST 0 = The PMU has lower priority than the TIC controller in the ASB ownership arbitration. 1 = The PMU has higher priority than the TIC controller in the ASB ownership arbitration in the TIC test mode (for the purpose of TIC-testing the PMU).

6.3 PMU Functions

Clock Generator

The Clock generator in the PMU is responsible for controlling the PLLs and masking clocks by AND-gating while the PLL outputs are unstable, and ensures that clocks are available during test modes and during RESET sequences.

FCLK (ARM Processor and SDRAM controller clock)

This clock is derived from the FCLK PLL (FPLL) whose frequency is programmable between 21 MHz and 96 MHz using the LSB 6 bits of the PMUCCR (PMU Clock Control Register). Its default frequency is 51 MHz.
There are two methods for updating frequency, depending upon the state of the bit 6 of the PMUCCR (see PMUCCR register on Section 5.3.4). If the bit 6 is set, then any data written to the bits [5:0] of the PMUCCR are immediately transferred to the pins of FPLL, thus causing the loop to unlock and to mute FCLK. This is only a safe mode of operation if FPLL frequency and mark-space ratio is guaranteed to be within limits immediately after lock time. If the bit 6 is not set, then the HMS30C7210 must enter DEEP SLEEP mode before the written bits [5:0] of the PMUCCR register are transferred to the FPLL.

To switch between the two frequencies when the bit 6 is not set:

- Software writes the new value into the PMUCCR register.
- Set a Real Time Clock (RTC) Alarm to wake up the HMS30C7210 in 2 seconds.
- Enter DEEP SLEEP mode by writing 0x7 to the bits [2:0] of the PMU Mode Register (PMUMR).
- The HMS30C7210 will resurrect with FPLL running at the new frequency by the preset RTC Alarm.

To switch between the two frequencies when the bit 6 is set and bit 7 is not set:

- Software writes the new value into the PMUCCR register.
- Changes to the clock frequency with program halting execution.
- After FPLL state is stable, program is executed. (So you don't need to check FPLL LOCK bit state)

To switch between the two frequencies when the bit 6 is set and bit 7 is set:

- Software writes the new value into the PMUCCR register.
- Changes to the clock frequency without program halting execution.

For final switch methode has unstable state(program is not stopped). If you want to check FPLL stable state, Write '1' bit in FPLL LOCK bit (it to be cleared) And read FPLL LOCK bit. If FPLL LOCK bit is '1', state is unstable. If FPLL LOCK bit is '0', state is stable.

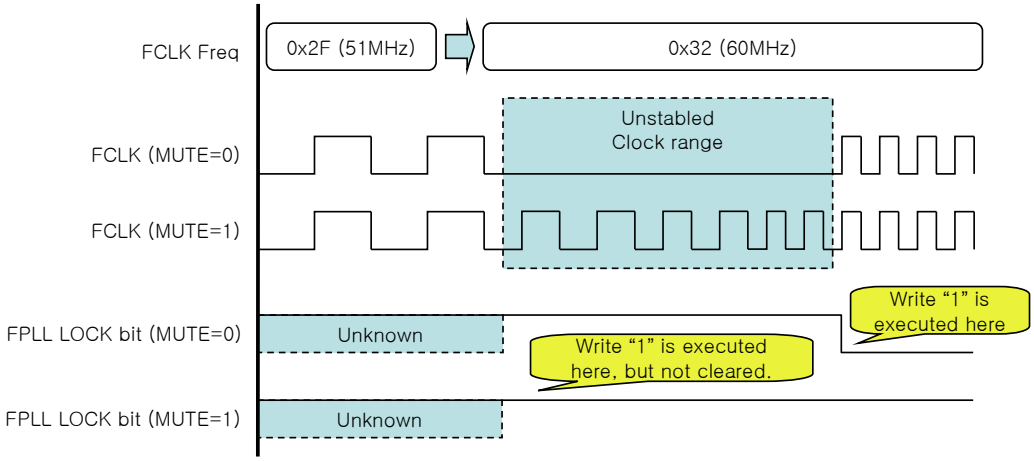


Figure 6-2. FCLK Frequency Update When the bit 6 is set

BCLK

This clock is ASB system bus clock generated by the PMU through dividing the FCLK frequency by 2 and 1/4 phase shift.

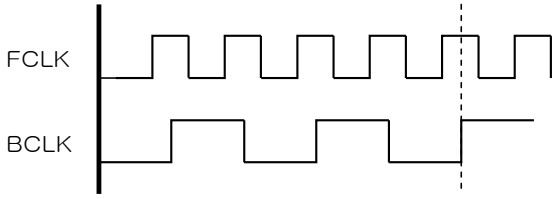


Figure 6-3. FCLK / BCLK relation

CCLK

The CCLK is generated by the CPLL and the frequency is fixed 48MHz. This clock is only used for the USB. The CCLK is disabled when BnRES (system reset) is active or when the PMU is put into DEEP SLEEP mode. On exit from either of these conditions, the CCLK must be re-enabled by software.

VCLK

The VCLK is selected between the FPLL and CPLL clock outputs using the bit 13 of the PMUCCR (the VCLK uses the FPLL output by default), and clocks the LCD controller. The VCLK is disabled when BnRES is active or when the PMU is put into DEEP SLEEP mode. On exit from either of these conditions, the VCLK must be re-enabled by software.

Changing Clock (PLL) Selection:

- Software must first disable the VCLK, by writing `0' to the bit 14 of the PMUCCR register.
- Modify the bit 13 of the PMUCCR.
- Re-enable the VCLK by writing ‘1’ to the bit 14 of the PMUCCR register.

PCLK

The PCLK is generated the CPLL divied by 13 (CPLL / 13 = 3.692308MHz). This clock is used for APB Block Function (UART, WDT, Timer etc).

QCLK

The QCLK is generated the CPLL divied by 13.5 (CPLL / 13.5 = 3.555556MHz). This clock is only used for the SmartCard Interface.

PMU state machine

The state machine handles the transition between the power management states described below. The CPU can write to the PMU mode registers (which is what would typically happens when a user switches off the device) and the state machine will proceed to the commanded state.

6.4 Power Management

6.4.1 State Diagram

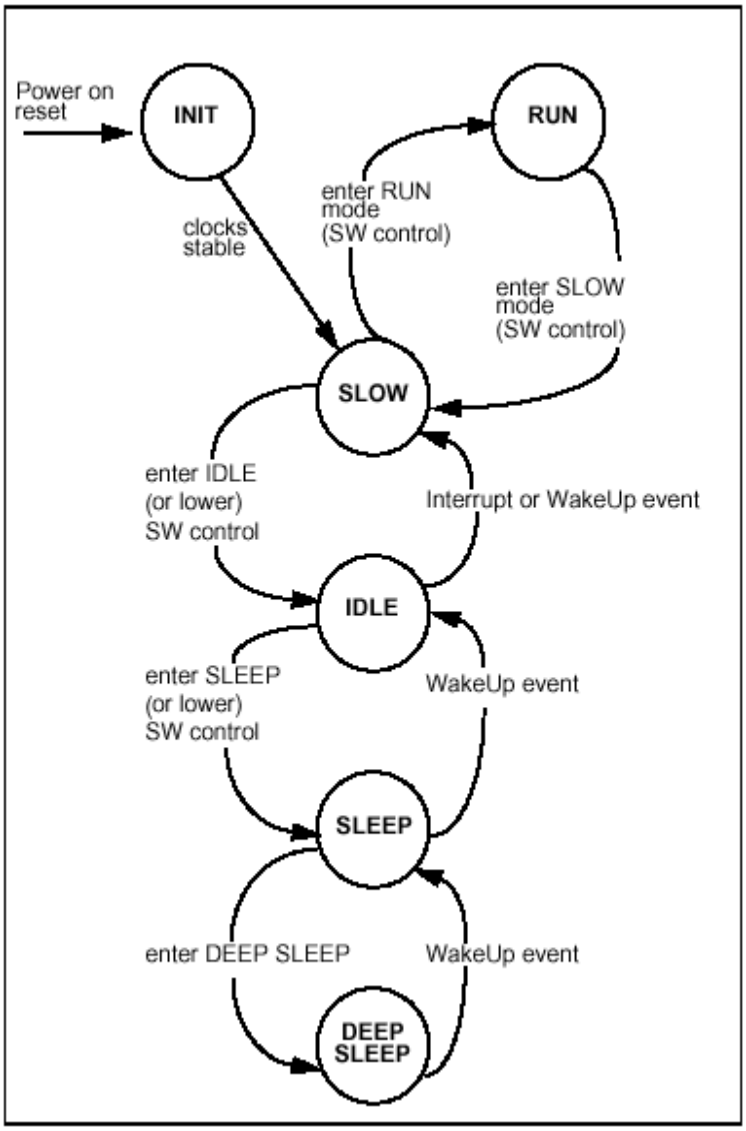


Figure 6-4. PMU Power Management State Diagram

6.4.2 Power management States

RUN

The system is running normally. All clocks are running (except where gated locally). The SDRAM controller is performing normal refresh.

SLOW

The CPU is switched into FastBus mode (please refer to the ARM720T DataSheet - DDI 0087), and hence runs at the BCLK rate (half the FCLK rate). This is the default mode after exiting DEEP SLEEP mode or system power on.

IDLE

In this mode, the PMU becomes the bus master until there is either a fast or normal interrupt for the CPU.
This will cause the clocks in the CPU to stop when it attempts an ASB access. The HMS30C7210 can enter this mode by writing 0x2 to the bits [2:0] of the PMUMR when in RUN or SLOW mode, or by WakeUp signal activation while in SLEEP or DEEP SLEEP mode.

NOTE: When the CPU sets IDLE mode into the PMU Mode Register, it must read non-chachable area for enter IDLE state.

SLEEP

In this mode, the SDRAM is put into self-refresh mode, and internal clocks are gated off. This mode can only be entered from IDLE mode (the PMU bus master must have the mastership of the ASB before this mode can be entered). The PMU must be the bus master to ensure that the system is stopped in a safe state, and is not half way through a SDRAM write (for example). Both the Video and Communication clocks (VCLK and CCLK) should be disabled before entering this state.
Usually the CPU would only drop in at this mode on the way to the DEEP SLEEP mode.

DEEP SLEEP

In the DEEP SLEEP mode, the crystal oscillator for the 6-MHz PLL input clock and the PLLs are disabled. This is the lowest power state available. Only the 32.768-KHz RTC oscillator runs and provides clocks for the RTC logic and the debouncing logic of the PMU. Everything else is powered down, and SDRAM is in self refresh mode. This is the normal system "off" mode.
The HMS30C7210 can get out of the SLEEP and DEEP SLEEP modes either by a user wake-up event (generally pressing the "On" key), by an RTC wake-up alarm, or by a modem ring indicate event. These wake-up sources go directly to the PMU.

6.4.3 Wake-up Debounce and Interrupt

The Wake-up events are debounced as follows:
Each of the event signals which are liable to noise (**nRESET, RTC, nPMWAKEUP, and Modem Ring Indicator, Power Adapter Condition**) is re-timed to a 15.625- or 250-Hz clock derived from the 32.768-KHz RTC clock. After being filtered to a quarter of the frequency of debouncing clock, each event has an associated `sticky' register bit. nPMWAKEUP (active low) is an external input, which may be typically connected to an "ON" key.

A `sticky' bit is a register bit that is set by the incoming event, but is only reset by the CPU. Thus should the FCLK PLL drop out of lock momentarily (for example) the CPU will be informed of the event, even if the PLL has regained lock by the time the CPU can read its associated register bit.

The **nPMWAKEUP, Modem, Real Time Clock, HotSync and Power Adapter condition** inputs are combined to form the PMU Interrupt. Each of these four interrupt sources (except Power Adapter condition) can wake up the CPU form the DEEP SLEEP mode, and then the CPU can be informed of each interrupt event. All of wakeup and interrupt sources may be individually enabled.

To make use of the nPMWAKEUP interrupt, (for example) controlling software will need to complete the following tasks:

- Enable the nPMWAKEUP interrupt, by writing ‘1’ to bit 11 of the PMU Reset / Status Register (PMURSR).
- Once an interrupt has occurred, read the PMURSR register to identify the source(s) of interrupt. In the case of a nPMWAKEUP event, the register will return 0x10.
- Clear the appropriate `sticky' bit by writing a ‘1’ to the appropriate bit location (in the nPMWAKEUP case, this will be the bit 4.).

PORTB[15] (HotSync) Wake-up Sequence

The PORTB[15] (HotSync) interrupt is OR-gated with nPMWAKEUP to support additional wake up sources.

PORTB[15] (HotSync) input signal can be used as a wake up source; it is also enabled an interrupt source using the Interrupt Enable Register of the Interrupt Controller. After wake up, software should program the GPIO PORTB interrupt mask bit of the Interrupt Enable Register and/or the HotSync interrupt mask bit of the PMURSR register.

One possible application is to use the **nDCD** signal, from the UART interface, as a wake up source, by connecting nDCD to a PORTB[15] input. In the DEEP SLEEP mode, nDCD can wake up the system by generating a PORTB[15] interrupt request to the PMU block. The PMU state machine then returns the system to the operational mode.

6.5 Reset Sequences

6.5.1 Power On Reset (Cold Reset))

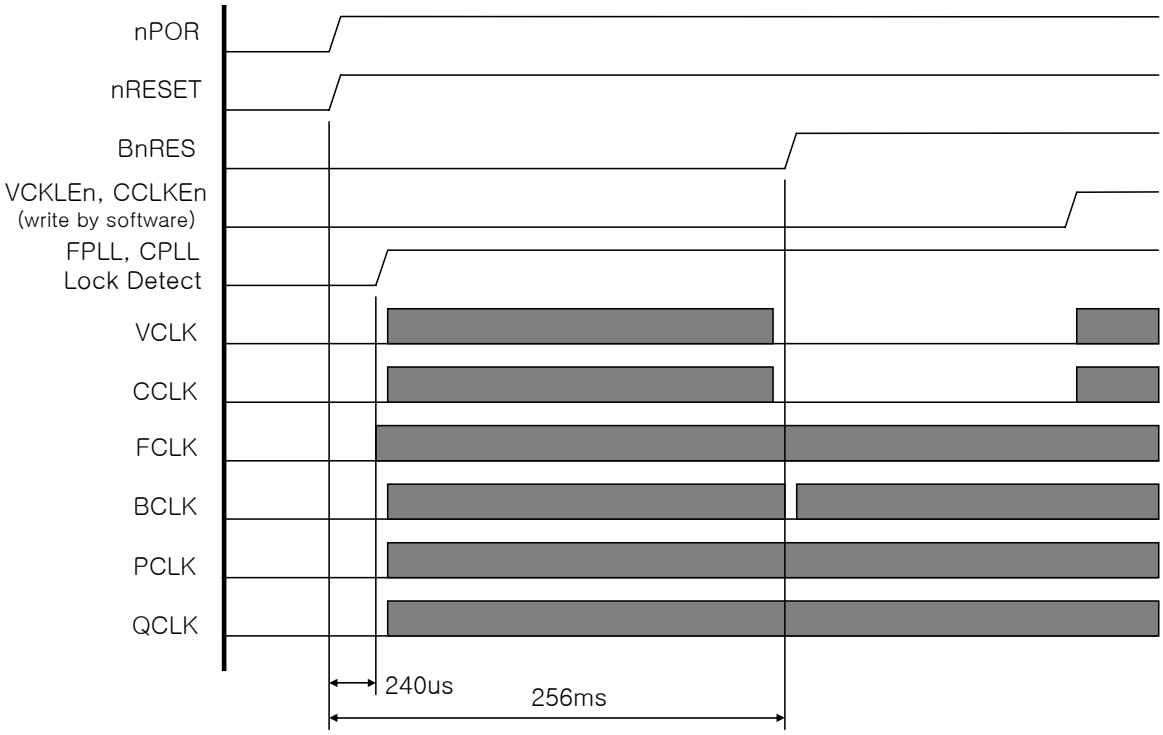


Figure 6-5. A Cold Reset Event

In the removal and re-application of all power to the HMS30C7210, the following sequence may be typical:

- nPOR input is active. All internal registers are reset to their default values. The PMU drives nRESETout LOW to reset any off-chip peripheral devices.
- BnRES becomes active on exit from nPOR condition. Clocks are enabled temporarily to allow synchronus resets to operate.
- The default frequency of FCLK on exit from nPOR will be 51MHz.
- When FCLK is stable, the CPU clock is released. If the CPU were to read the Reset / Status register (PMURSR) at this time, It will return 0x03E0_000D.
- The CPU may write 0x03E0_000D to the PMURSR to clear these flag bits.

Bit	Meaning
bit 3 set:	FCLK PLL has been 'unlocked'
bit 2 set:	CCLK PLL has been 'unlocked'
bit 0 set:	Power On Reset event has occuerred

Table 6-2. Bit Settings for a Cold RESET Event within PMURSR register

- The CPU writes 0x0032 to the Clock Control register (PMUCCR), which will set a FCLK speed of 60MHz. The new clock frequency, however, is not adopted until the PMU has entered and left DEEP SLEEP mode.

- The CPU sets a RTC timer alarm to expire in approximately 2 seconds.
- The CPU sets DEEP SLEEP into the PMU Mode Register
- The PMU state machine will enter DEEP SLEEP mode (via the intermediate states shown in Figure 6-4. PMU Power Management State Diagram.
- When the RTC timer alarm is activated, the PMU automatically wakes up into SLOW mode, but with the new FCLK frequency of 60MHz.
- The CPU may write 0xC032 to the Clock Control register, which enable CCLK and VCLK, and retains the new FCLK frequency.

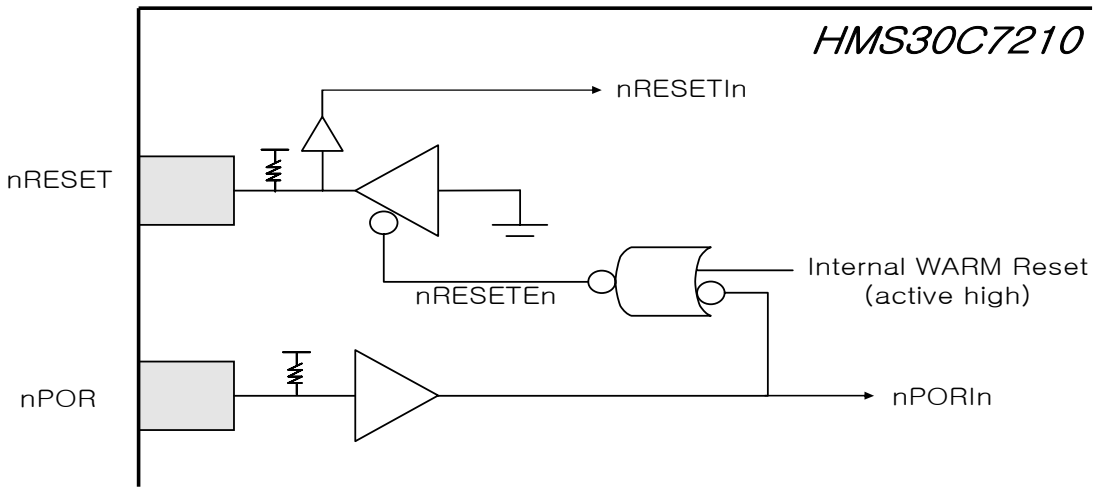


Figure 6-6. nPOR / nRESET / SoftwareReset Function

6.5.2 Software Generated Warm Reset

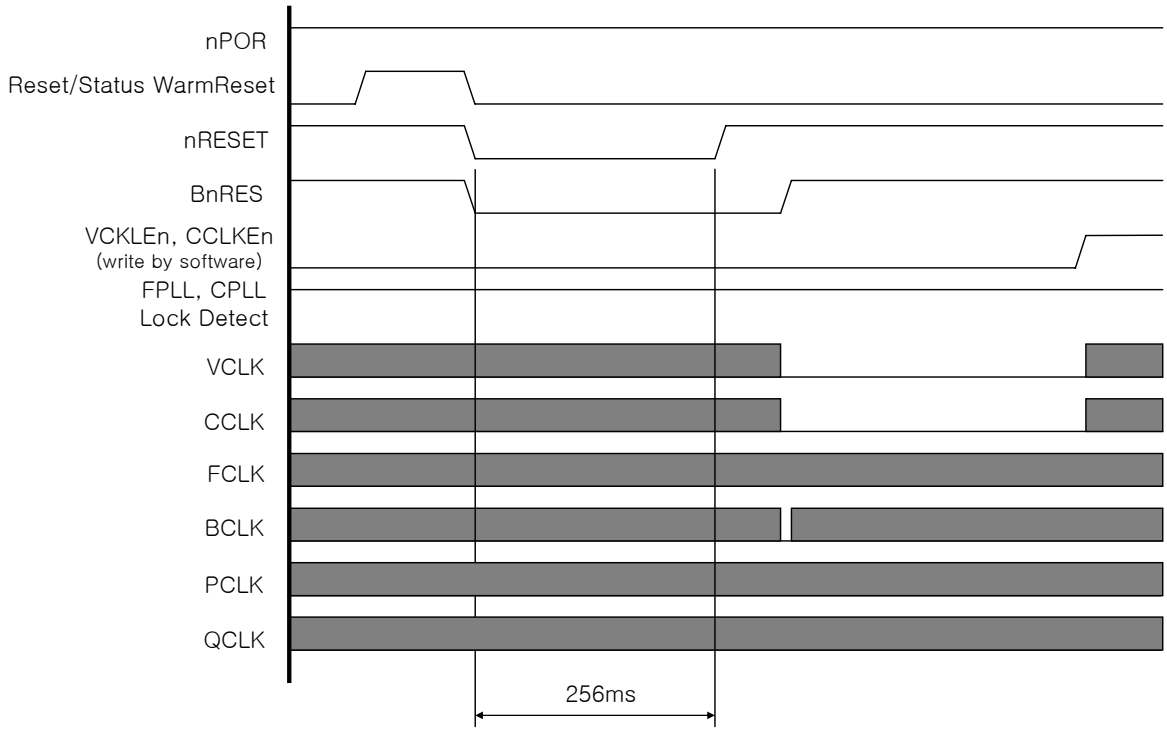


Figure 6-7. Software Generated Warm Reset

- The CPU writes '1' to the WarmReset bit of RESET / Status register. The PMU drives **nRESET** low. The internal chip reset, BnRES is drive low. The PMU detects that the bidirectional nRESET pin is low. nRESET is filtered by a de-bounce circuit. Note that this means that nRESET will remain low for a mininum of 256ms (15.625Hz Pulse x 4). BnRES becomes active once the de-bounced nRESET goes high once more, whihc disables VCLK and CCLK. The CPU may read the RESET / Status register, which will return 0x03E0_010C.

Bit	Meaning
bit 8 set:	WARM Reset event has occurred.
bit 3 set:	FCLK PLL has been 'unlocked'
bit 2 set:	CCLK PLL has been 'unlocked'

Table 6-3. Bit Settings for a Software generated Warm Reset within Reset / Status register

6.5.3 An Externally Generated Warm Rese

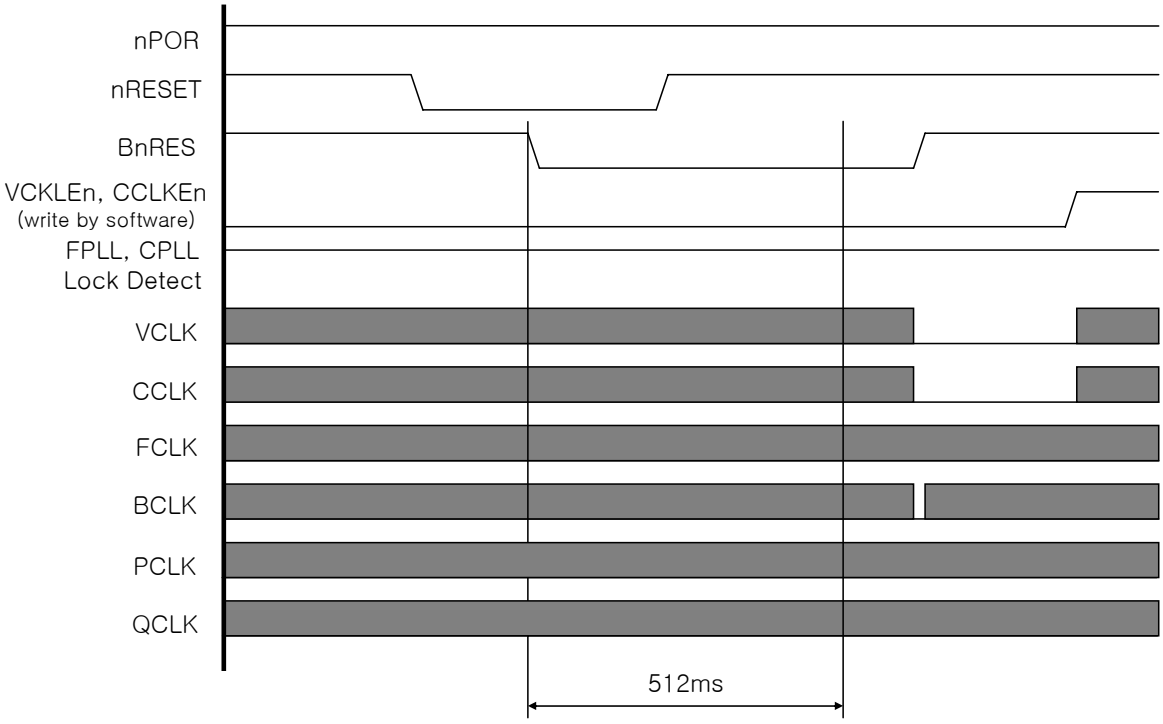


Figure 6-8. An Externally Generated Warm Reset

- nRESET is driven to '0' by external hardware. The nRESET input is filtered by a de-bounce circuit. Note that this means that nRESET must remain low for a minimum of 512ms. BnRES (the on-chip reset signal) becomes active as soon as nRESET is low, and high once the de-bounced nRESET goes high once. BnRES disables VCLK and CCLK. The CPU may read the RESET / Status register, which will return 0x03E0_010C.

Bit	Meaning
bit 8 set:	WARM Reset event has occurred.
bit 3 set:	FCLK PLL has been 'unlocked'
bit 2 set:	CCLK PLL has been 'unlocked'

Table 6-4. Bit Settings for a Warm Reset within Reset / Status register

Note. The internal chip reset, BnRES remains active for 256ms after an externally generated nRESET. External devices should not assume that the HMS30C7210 is in an active state during this period.

7 SDRAM CONTROLLER

The SDRAM controller operates at the full CPU core frequency (FCLK) and is connected to the core via the ASB bus. Internally the SDRAM controller arbitrates between access requests from the main AMBA bus, and the LCD bus.

It can control up to two SDRAMs of 256Mbit (x16) density maximum. To reduce the system power consumption it can power down these individually using the Clock Enable (CKE). When the MCU is in standby mode the SDRAMs are powered down into self-refresh mode.

SDRAMs achieve the highest throughput when accessed sequentially – like LCD data. However accesses from the core are less regular. The SDRAM controller uses access predictability to maximize the memory interface bandwidth by having access to the LCD address buses. LCD accesses to the SDRAM occur in fixed-burst lengths of 16 words. ARM accesses occur in a fixed-burst length of four words. If the requested accesses are shorter than four words, then the extra data is ignored.

FEATURES

- 16 Bits wide external bus interface (two access requires for each word)
- Supports 16/64/128/256Mbit device
- Supports 2~64 Mbytes in up to two devices (the size of each memory device may be different)
- Programmable CAS latency
- Supports 2/4 banks with page lengths of 256 or 512 half words
- Programmable Auto Refresh Timer
- Support low power mode when IDLE (each device's CKE is disable individually).

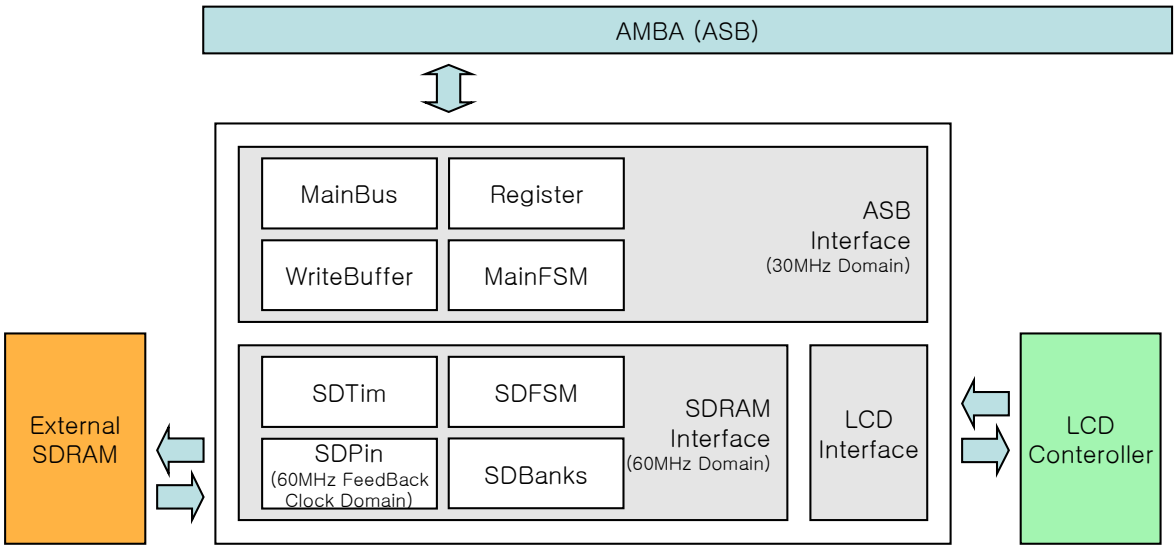


Figure 7-1. SDRAM Controller Block Diagram

7.1 Supported Memory Devices

2-64MBytes of SDRAM are supported with any combination of one or two 16/64/128/256Mbit devices. Each device is mapped to a 32MByte address space. The MMU (memory management unit) maps different device combinations (e.g. 16- and 64Mbit devices) into a continuous address space for the ARM core.

Total Memory	16Mbit devices	64Mbit devices	128Mbit devices	256Mbit devices
2Mbyte	1	-	-	-
4Mbyte	2	-	-	-
8Mbyte	-	1	-	-
16Mbyte	-	2	1	-
32Mbyte	-	-	2	1
64Mbyte	-	-	-	2

Note The HMS30C7210 can use any mixture of 16-, 64-, 128- or 256Mbit SDRAMs. It is the responsibility of software to determine the actual external memory configuration, and to program the memory management unit appropriately.
The SDRAM controller allows up to four memory banks to be open simultaneously. The open banks may exist in different physical SDRAM devices.

7.2 External Signals

Pin Name	Type	Description
RA [14:11] SA10 RA [9:0]	O	SDRAM address bus
RD [15:0]	I/O	SDRAM data bus
SCLK	O	SDRAM clock output
SCKE [1:0]	O	SDRAM clock enable outputs
nRAS	O	SDRAM row address select output
nCAS	O	SDRAM column address select output
nSWE	O	SDRAM write enable output
nSCS [1:0]	O	SDRAM chip select outputs
DQML	O	SDRAM lower data byte enable
DQMU	O	SDRAM upper data byte enable

Refer to Figure 2-1. 208 Pin diagram.

7.3 Registers

The SDRAM controller has three registers: the configuration, refresh timer and the Write Buffer Flush timer. The configuration register's main function is to specify the number of SDRAMs connected, and whether they are 2- or 4-bank devices. The refresh timer gives the number of BCLK ticks that need to be counted in-between each refresh period. The Write Buffer Flush timer is used to set the number of BCLK ticks since the last write operation, before the write buffer's contents are transferred to SDRAM.

Address	Name	Width	Default	Description
0x8000.0000	SDCON	32	0x0070 0000	Configuration register
0x8000.0004	SDREF	16	0x0000 0080	Refresh timer
0x8000.0008	SDWBF	3	0x0000 0000	Write back buffer flush timer

Table 7-1 SDRAM Controller Register Summary

In addition to the SDRAM control registers, the ARM may access the SDRAM mode registers by writing to a 64MByte address space referenced from the SDRAM mode register base address. Writing to the SDRAM mode registers is discussed further..

7.3.1 SDRAM Controller Configuration Register (SDCON)

0x8000.0000

31	30	29	28	27	26	25	24
S1	S0	-	-	-	-	-	-
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
R	A	C1	C0	D	C	B	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
-	-	-	-	-	-	-	-
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
E1	B1	-	-	E0	B0	-	-

Bits	Type	Function
31:30	R	SDRAM controller Status, read-only S[1:0] = 11, Reserved S[1:0] = 10, Self refresh S[1:0] = 01, Busy S[1:0] = 00, Idle
23	R/W	Normal SDRAM controller refresh enable 1 = the SDRAM controller provides refresh control 0 = the SDRAM controller does not provide refresh
22	R/W	Auto pre-charge on ASB accesses 1 = auto pre-charge (default) 0 = no auto pre-charge
21:20	R/W	CAS Latency Control C[1:0] = 11, CAS latency 3 C[1:0] = 10, CAS latency 2 C[1:0] = 01, CAS latency 1 C[1:0] = 00, Reserved
19	R/W	SDRAM bus tri-state control 0 = the controller drives the last data onto the SDRAM data bus (default) 1 = the SDRAM bus is tri-stated except during writes This bit should be cleared before the IC enters a low power mode. Driving the data lines avoids floating inputs that could increase device power consumption. During normal operation the D bit should be set, to avoid data bus drive conflicts with SDRAM.
18	R/W	SDRAM clock enable control 0 = the clock of IDLE devices are disabled to save power (default) 1 = all clock enables are driven HIGH continuously
17	R/W	Write buffer enable Value = 1 if the write buffer is enabled Value = 0 if the write buffer is disabled
7	R/W	Device enable – indicates that there is a physical SDRAM present in each of the two slots in the address map. This bit is used to determine whether an auto-refresh command should be issued to a particular memory device. 1 = a device is present at address range 32-64Mbyte (SLOT 1) 0 = a device is not present at address range 32-64Mbyte
6	R/W	Indicates whether the SDRAM in the SLOT is a 2- or 4-bank device 1 = the SDRAM is a four-bank device 0 = the SDRAM is a two-bank device
3	R/W	Device enable – indicates that there is a physical SDRAM present in each of the two slots in the address map. This bit is used to determine whether an auto-refresh command should be issued to a particular memory device. 1 = a device is present at address range 0-32MByte (SLOT 0) 0 = a device is not present at address range 0-32Mbyte
2	R/W	Indicates whether the SDRAM in the SLOT is a 2- or 4-bank device 1 = the SDRAM is a four-bank device 0 = the SDRAM is a two-bank device

The SDRAM controller configuration register is a 32-bit wide split read/write register, such that bits [23:0] should be configured by the ARM, and bits [31:24] provide status information that read-only. All locations containing "-"are for future expansion, and should always be programmed with the binary value 0. Writes to bits [31:24] are always ignored. During power-up initialization, it is important that the E[1:0] and the R bits are set in the correct sequence.
The SDRAM controller powers-up with E[1:0]=00 and R=0.

This indicates that the memory interface is IDLE. Next, the software should set at least one E bit to 1 with the R bit 0. This will cause both devices to be precharged (if present).

The next operation in the initialization sequence is to auto-refresh the SDRAMs. Note that the number of refresh operations required is device-dependent. Set R=1 and E[1:0]=00 to start the auto-refresh process. Software will have to ensure that the prescribed number of refresh cycles is completed before moving on to the next step.

The final step in the sequence is to set R=1 and to set the E bits corresponding to the populated slots. This will put the SDRAM controller (and the SDRAMs) in their normal operational mode.

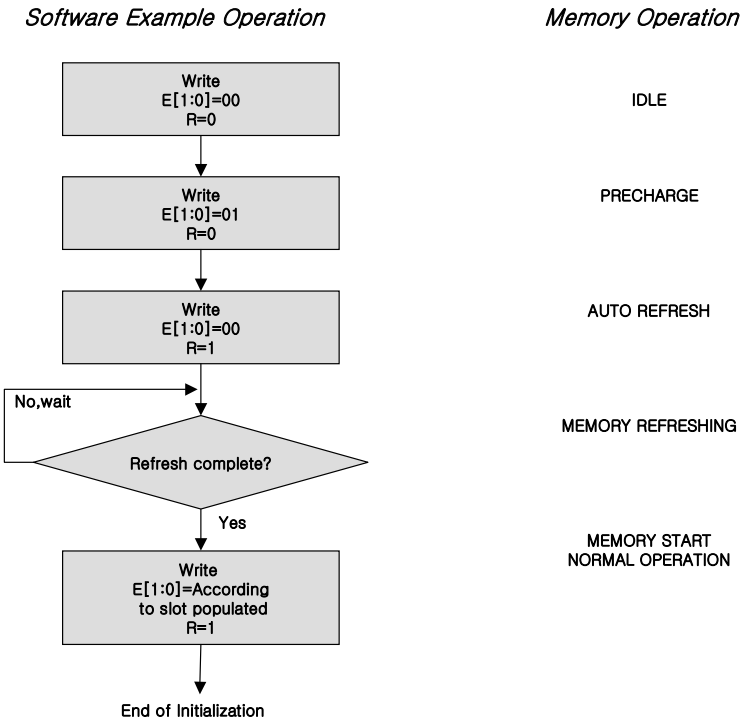


Figure 7-2. SDRAM Controller Software Example and Memory Operation Diagram

7.3.2 SDRAM Controller Refresh Timer Register (SDREF)

0x8000.0004

-	15 – 0
Reserved	SDREF

Bits	Type	Function
15:0	R/W	A 16-bit read/write register that is programmed with the number of BCLK ticks that should be counted between SDRAM refresh cycles. For example, for the common refresh period of 16us (16x10E-6), and a BCLK frequency of 30MHz (30x10E6), the following value should be programmed into it: (16x10E-6) x (30x10E6) = 480 The refresh timer defaults to a value of 128, which for a 16us refresh period assumes a worst case (i.e. slowest) clock rate of: 128 / (16x10E-6) = 8 MHz The refresh register should be programmed as early as possible in the system start-up procedure, and in the first few cycles if the system clock is less than 8MHz.

7.3.3 SDRAM Controller Write buffer flush timer Register (SDWBF)

0x8000.0008

-	2 – 0
Reserved	SDWBF

Bits	Type	Function
2:0	R/W	A 3-bit read/write register that sets the time-out value for flushing the quad word merging write buffer. The times are given in the following table.

Timer value	BCLK ticks between time-outs
111	128
110	64
101	32
100	16
011	8
010	4
001	2
000	Time-out disabled

7.4 Power-up Initialization of the SDRAMs

The SDRAMs are initialized by applying power, waiting a prescribed amount of settling time (typically 100us), performing at least 2 auto-refresh cycles and then writing to the SDRAM mode register. The exact sequence is SDRAM device-dependent.

The settling time is referenced from when the SDRAM CLK starts. The processor should wait for the settling time before enabling the SDRAM controller refreshes, by setting the R bit in the SDRAM control register. The SDRAM controller automatically provides an auto refresh cycle for every refresh period programmed into the Refresh Timer when the R bit is set. The processor must wait for sufficient time to allow the manufacturer's specified number of auto-refresh cycles before writing to the SDRAM's mode register.

The SDRAM's mode register is written to via its address pins (A[14:0]). Hence, when the processor wishes to write to the mode register, it should read from the binary address (AMBA address bits [24:9]), which gives the binary pattern on A[14:0] which is to be written. The mode register of each of the SDRAMs may be written to by reading from a 64Mbyte address space from the SDRAM mode register base address. The correspondence between the AMBA address bits and the SDRAM address lines (A[14:0]) is given in the Row address mapping of Table 7-2 SDRAM Row/Column Address Map. Bits [25] of the AMBA address bus select the device to be initialized.

The SDRAM must be initialized to have the same CAS latency as is programmed into C[1:0] bits of the SDRAM control register, and always to have a burst length of 8.

7.5 SDRAM Memory Map

The SDRAM controller can interface with up to two SDRAMs. Four SDRAM sizes are supported -- 16, 64, 128 and 256Mbits -- which may be organized in either two or four banks but which must have a 16-bit data bus. A maximum of 64Mbytes of memory may be addressed by the SDRAM controller, which subdivided into two 32Mbyte blocks, one for each of the external SDRAMs.

The mapping of the AMBA address bus to the SDRAM row and column addresses is given in Table 7-2 SDRAM Row/Column Address Map. The first row of the diagram indicates the SDRAM address bit (A[14:0]); the remaining numbers indicate the AMBA address bits BA[24:1]. Note that for 16Mbit device, pins A[11,9] on thee SDRAM should be connected to pins RA[13,12] on the HMS30C7210, and the pins RA[11,9] should not be connected.

SDRAM ADDR	14	13 (BS0)	12 (BS1)	11	10	9	8	7	6	5	4	3	2	1	0
Row 16Mbit	24	10*	9*	Note 1	20*	Note 1	19*	18*	17*	16*	15*	14*	13*	12*	11*
Col 16Mbit	24	10	10	Note 1	20	Note 1	23	8*	7*	6*	5*	4*	3*	2*	Note 2
Row 64Mbit	24	10*	9*	22*	20*	21*	19*	18*	17*	16*	15*	14*	13*	12*	11*
Col 64Mbit	24	10	10	22	20	21	23	8*	7*	6*	5*	4*	3*	2*	Note 2
Row 128Mbit	24	10*	9*	22*	20*	21*	19*	18*	18*	16*	15*	14*	13*	12*	11*
Col 128Mbit	24	10	10	22	20	21	23*	8*	7*	6*	5*	4*	3*	2*	Note 2
Row 256Mbit	24*	10*	9*	22*	20*	21*	19*	18*	18*	16*	15*	14*	13*	12*	11*
Col 256Mbit	24	10	10	22	20	21	23*	8*	7*	6*	5*	4*	3*	2*	Note 2
Mode Write	24*	10*	9*	22*	20*	21*	19*	18*	17*	16*	15*	14*	13*	12*	11*
Summary	24	10	9	22	20	21	19/23	18/8	17/7	16/6	15/5	14/4	13/3	12/2	11*

Table 7-2 SDRAM Row/Column Address Map

Notes (1) For the 16Mbit device, SDRAM address line A11 should be connected to the HMS30C7210 pin RA[13](BS0), and the SDRAM address line A9 should be connected to the HMS30C7210 pin RA[12](BS1). The HMS30C7210 address lines RA[11] and RA[9] should not be connected.
(2) Since all burst accesses commence on a word boundary, and SDRAM addresses are non-incrementing (the address incremented is internal to the device), column address zero will always be driven to logic `0'.
* An asterisk denotes the address lines that are used by the SDRAM.

The start address of each SDRAM is fixed to a 32Mbyte boundary. The memory management unit will be used to map the actual banks that exist into contiguous memory as seen by the ARM. Bits [25] of the AMBA address bus select the device to be initialized, as described in Table 7-3.

BA25	Device selected
0	Device 0
1	Device 1

Table 7-3 SDRAM Device Selection

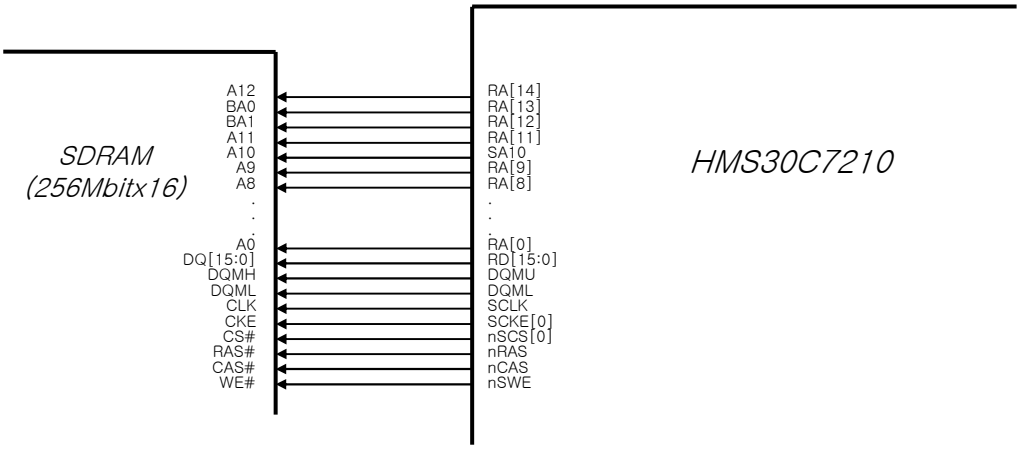


Figure 7-3. 256Mbitx16 (4Banks) Device Connection

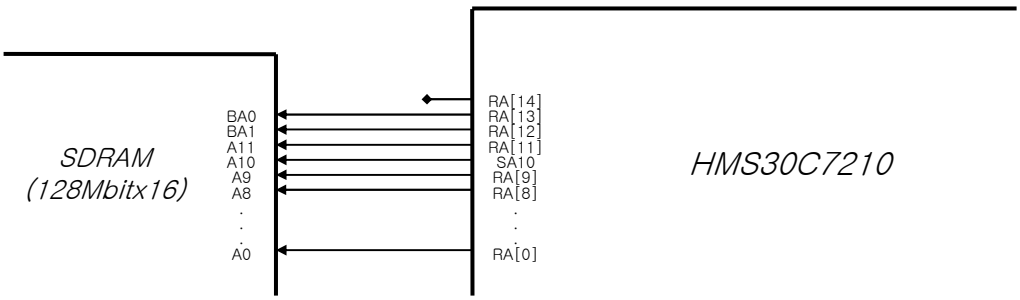


Figure 7-4. 128Mbitx16 (4Banks) Device Connection

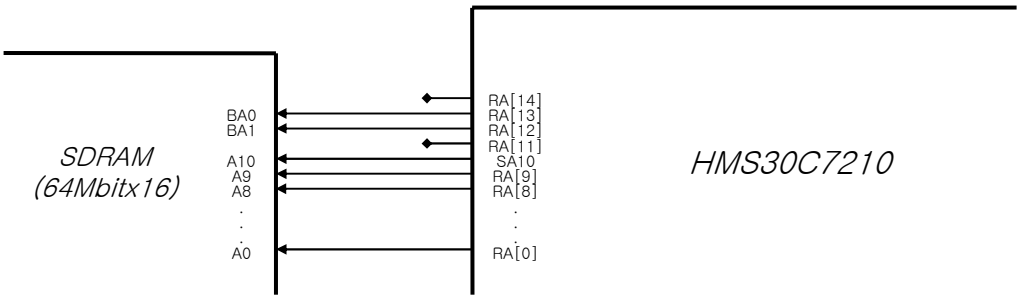


Figure 7-5. 64Mbitx16 (4Banks) Device Connection

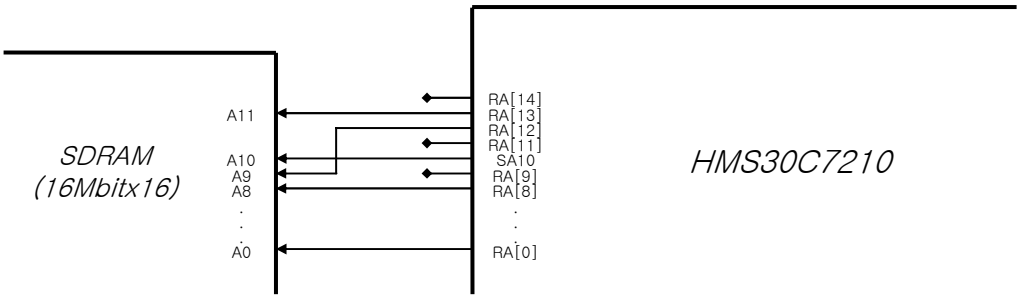


Figure 7-6. 16Mbitx16 (2Banks) Device Connection

7.6 AMBA Accesses and Arbitration

The SDRAM controller bridges both the AMBA Main and Video buses. On the Main bus, the SDRAM appears as a normal slave device. On the LCD DMA bus, the SDRAM controller integrates the functions of the bus arbiter and address decoder. Writes from the main bus may be merged in the quad word merging write buffer. A Main/LCD arbiter according to the following sequence arbitrates access requests from either the Main or LCD buses:

- Highest Priority: LCD
- Middle Priority: Refresh request
- Lowest Priority: Main bus peripheral (PMU, ARM)--order determined by Main bus arbiter.

LCD SDRAM accesses always occur in bursts of 16 words. Once a burst has started, the SDRAM controller provides data without wait states. LCD data is only read from SDRAM, no write path is supported.

If a refresh cycle is requested, then it will have lower priority than the Video bus, but will be higher than any other accesses from the Main bus. Assuming a worst-case BCLK frequency of 8MHz, the maximum, worst-case latency that the arbitration scheme enforces is 11.5us before a refresh cycle can take place. This is comfortably within the 16us limit. Note that the 2 external SDRAM devices are refreshed on 2 consecutive clock cycles to reduce the peak current demand on the power source.

The arbitration of the Main bus is left to the Main bus arbiter. Data transfers requested from the Main bus always occur as a burst of eight half-word accesses to SDRAM. The Main bus arbiter cannot break into access requests from the Main bus. In the case where fewer than four words are actually requested by the Main bus peripheral, the excess data from the SDRAM is ignored by the SDRAM controller in the case of read operations, or masked in the case of writes.

In the case where more than four words are actually requested by the Main bus peripheral, the SDRAM controller asserts BLAST to force the ASB decoder to break the burst.

In the case of word/half-word/byte misalignment to a quad word boundary (when any of address bits [3:0] are non-zero at the start of the transfer), BLAST is asserted at the next quad word boundary to force the ASB decoder to break the burst. Sequential half word (or byte) reads are supported and the controller asserting BLAST at quad word boundary. In the case of byte or half word reads, data is replicated across the whole of the ASB data bus.

Data bus for word access:

31 **23**
15 **7** **0**
d31 d30 d29 d28 d27 d26 d25 d24 d23 d22 d21 d20 d19 d18 d17 d16 d15 d14 d13 d12 d11 d10 d9 d8 d7 d6 d5 d4 d3 d2 d1 d0

Data bus for half word access:

31 **23** **15**
7 **0**
d15 d14 d13 d12 d11 d10 d9 d8 d7 d6 d5 d4 d3 d2 d1 d0 d15 d14 d13 d12 d11 d10 d9 d8 d7 d6 d5 d4 d3 d2 d1 d0

Data bus for byte access:

31 **23** **15**
7 **0**
d7 d6 d5 d4 d3 d2 d1 d0 d7 d6 d5 d4 d3 d2 d1 d0 d7 d6 d5 d4 d3 d2 d1 d0 d7 d6 d5 d4 d3 d2 d1 d0

7.7 Merging Write Buffer

An eight word merging Write-Buffer is implemented in the SDRAM controller to improve write performance. The write buffer can be disabled, but its operation is completely transparent to the programmer. The eight words of the buffer are split into two quad words, the same size as all data transactions to the SDRAMs. The split into two quad words allows one quad word to be written to at the same time as the contents of the other are being transferred to SDRAM. The quad word buffer currently being written to may be accessed with non-contiguous word, half word or byte writes, which will be merged into a single quad word. The buffered quad word will be transferred to the SDRAM when:

- There is a write to an SDRAM address outside the current quad word being merged into
- There is a read to the address of the quad word being merged into
- There is a time-out on the write back timer

The two quad-words that make up the write buffer operate in "ping-pong" fashion, whereby one is initially designated the buffer for writes to go into, and the other is the buffer for write backs. When one of the three events that can cause a write-back occurs, the functions of the two buffers are swapped. Thus the buffer containing data to be written back becomes the buffer that is currently writing back, and the buffer that was the write-back buffer becomes the buffer being written to.

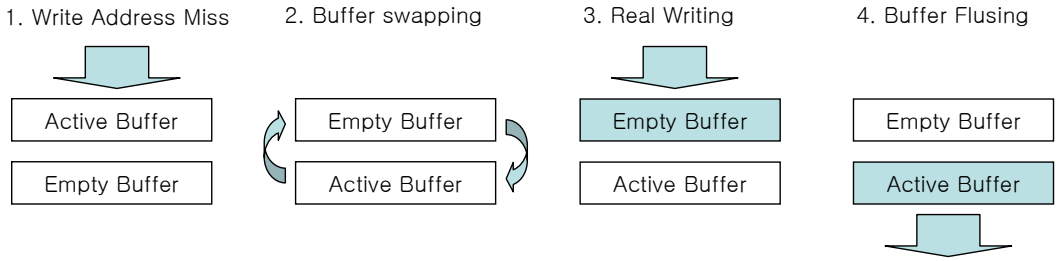


Figure 7-7. Write Miss Flusing

In the case of a write-back initiated by a read from the same address as the data in the merge buffer, the quad word in the buffer is written to SDRAM, and then the read occurs from SDRAM. The write before read is essential, because not all of the quad word in the buffer may have been updated, so its contents need to be merged with the SDRAM contents to fill any gaps where the buffer was not updated.

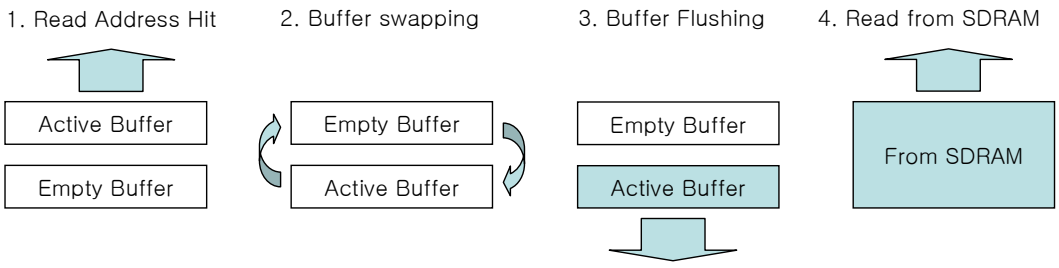


Figure 7-8. Read Hit Flusing

The write buffer flush timer forces a write back to occur after a programmable amount of time. Every time a write into the buffer occurs, the counter is re-loaded with the programmed time-out value, and starts to counts down. If a time-out occurs, then data in the write buffer is written to SDRAM.

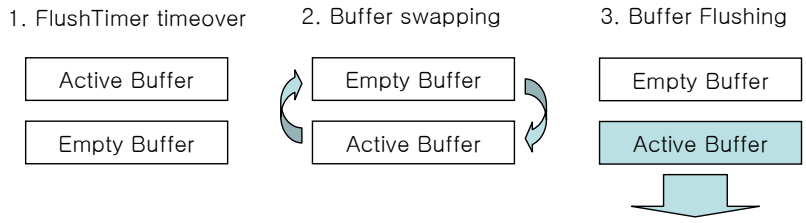


Figure 7-9. Timer timeover Flusing

8 STATIC MEMORY INTERFACE

The Static Memory Controller interfaces the AMBA Advanced System Bus (ASB) to External Memory Systems e,g, SRAM, FLASH, ROM. It can be programmed to use EBI(External Bus Interface) or not. It provides four separate memory or expansion banks. Each bank is 16MB in size and can be programmed individually to support:

FEATURES

- Unified External Bus Interface with SDRAM Address and Data pins
- 8- or 16-bit wide, little-endian memory
- Alignment Error Checking
- Burst read access support
- Variable wait states (up to 15 for READ, up to 16 for Write) :: Unable to Write with Zero wait state
- SMC (Nand Flash Memory) access support (See SMC controller, section 9.8.3 SMC access using EBI interface)

In addition, Burst mode access allows fast sequential read access by the System Bus Commands. This can significantly improve bus bandwidth in reading from memory (that must support at least four word burst reads).

8.1 External Signals

Pin Name	Type	Description
nRWE[1:0]	O	These signals are active LOW write enables for each of the memory byte lanes on the external bus.
nROE	O	Active LOW Output enable
nRCS[3:0]	O	Active LOW chip selects.
RA [23:0]	O	Address Bus
RD [15:0]	I/O	Data Bus

Refer to Figure 2-1. 208 Pin diagram.

8.2 Registers

Address	Name	Width	Default	Description
0x8002.0004	BANK0_REG	13	0x0041	Memory Configuration Register 0
0x8002.0008	BANK1_REG	13	0x0041	Memory Configuration Register 1
0x8002.000C	BANK2_REG	13	0x0041	Memory Configuration Register 2
0x8002.0010	BANK3_REG	13	0x0041	Memory Configuration Register 3

Table 8-1 Static Memory Controller Register Summary

8.2.1 MEM Configuration Register

			12	11	10	9	8	7	6	5	4	3	2	1	0
			BT Dne	BUR EN	BURST READ WAIT STATE				NORMAL ACCESS WAIT STATE				-	MEM WIDTH	

Bits	Type	Function
31:13		Reserved
12	R/W	Boot Done This controller can have the boot bits which defines the Memory Size for the Booting. And in the Boot Mood, All external Memory Bank Memory Size is determined only by the boot bits signal. So, when the booting is done, attached external memory size should be properly set by the host software. *** MEM WIDTH field can only be set when this bit is logic 1. So, after booting is done, the host software should set this bit to logic 1 for properly setting the attached memory size.
11	R/W	Burst Enable Setting this bit enables burst reads to take advantage of faster access times from memory devices that support burst mode.
10:7	R/W	BURST Read Wait State Value Number of Burst Read Wait State :: same as the bit number 0000 0 0001 1 …… 1111 15 default wait is not set
6:3	R/W	NORMAL Access Wait State Value Number of Normal Access Wait State 0000 0(read mode), 1(write mode) 0001 1(read mode), 2(write mode) …… 1111 15(read mode), 16(write mode) default is 1000 (8, read mode :: 9, write mode) :: In case of read operation, the asserted wait numbers are equal to the value of this field. But, in write operation, the asserted wait number should add 1 to this field value. So, write operation to external memory can't be done in zero wait
2	-	-
1:0	R/W	Memory Width 00 :: 8bit-wide Memory 01 :: 16bit-wide Memory 10 :: Reserved 11 :: Reserved for future Use

8.3 Functional Description

The Static Memory Controller (SMI) has six main functions:

- Memory bank select
- Access sequencing
- Wait states generation
- Burst read control
- Byte lane write control these are described below

8.3.1 Memory bank select

Internally, The Static Memory Controller can support up to four External Memory Bank and for this purpose, it's equipped with four bank controller registers. But externally, only one chip Select pin is assigned. So, only Bank0 Can be used for External Memory Access.

Case I. ROMSWAP is '1' address mapping (Means that external booting)

Start Address	Address (Hex)	Size	Description
(256M +0M)Byte	0x0000.0000	16Mbytes	ROM chip select 0
(256M+ 16M)Byte	0x0100.0000	16Mbytes	ROM chip select 1
(256M + 32M)Byte	0x0200.0000	16Mbytes	ROM chip select 2
(256M + 64M)Byte	0x0300.0000	16Mbytes	ROM chip select 3

Case II. ROMSWAP is '0' address mapping (Means that internal booting)

Start Address	Address (Hex)	Size	Description
(256M +0M)Byte	0x1000.0000	16Mbytes	ROM chip select 0
(256M+ 16M)Byte	0x0100.0000	16Mbytes	ROM chip select 1
(256M + 32M)Byte	0x0200.0000	16Mbytes	ROM chip select 2
(256M + 64M)Byte	0x0300.0000	16Mbytes	ROM chip select 3

Refer to Figure 4-1, Figure 4-2.

8.3.2 Access sequencing

Bank configuration also determines the width of the external memory devices. When the external memory bus is narrower than the transfer initiated from the current master, the internal transfer will take several external bus transfers to complete. And in addition, the access to External memory should always meet the Alignment Condition. When there is an access which does not meet the Alignment, this controller generates bus error condition which may be used for abort condition.

8.3.3 Wait states generation

The Static Memory Controller supports various wait states for read and write accesses. This is configurable between zero and 15 wait states for standard memory access (write operation to external memory can't be done in 0 wait).

8.3.4 Burst read control

This supports sequential access burst reads in 8- or 16-bit memories according to the ABMA Bus signal.

8.3.5 Byte lane write control

This controls nRWE[1:0] according to transfer width, BA[1:0] and the access sequencing. The table below shows nRWE[1:0] coding case by little endian accessing to 16, 8-bit external memory bus.

CASE 1. ACCESS : Write, 16-bit external bus

BSIZE [1:0]	BA [1:0]	IA [1:0]*note1	nRWE [1:0]
10 (WORD)	XX	1X	00
	XX	0X	00
01 (HALF)	1X	1X	00
	0X	0X	00
00 (BYTE)	11	1X	01
	10	1X	10
	01	0X	01
	00	0X	10

CASE 1. ACCESS : Write, 8-bit external bus

BSIZE [1:0]	BA [1:0]	IA [1:0]*note1	nRWE [1:0]
10 (WORD)	XX	11	10
	XX	10	10
	XX	01	10
	XX	00	10
01 (HALF)	1X	11	10
	1X	10	10
	0X	01	10
	0X	00	10
00 (BYTE)	11	11	10
	10	10	10
	01	01	10
	00	00	10

Note1 : IA[1:0] (internal SMI Address)

The Write Operation can be attempted with 8 or 16Bit Wide regardless of the attached External Memory Size. The translation is done internally in this controller. Internally, this controller support 2bit wide Write Enable strobe, each for individual Byte. But there exist only one external Write Enable Strobe(nRWE[1:0]).

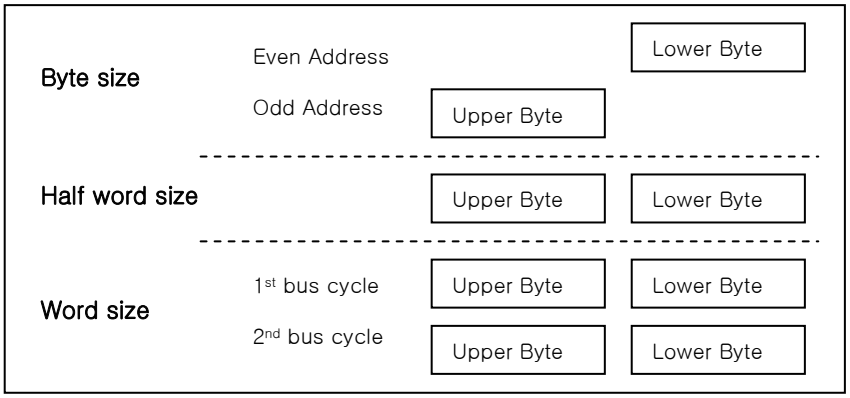


Figure 8-1. Data flow at 16-bit width memory

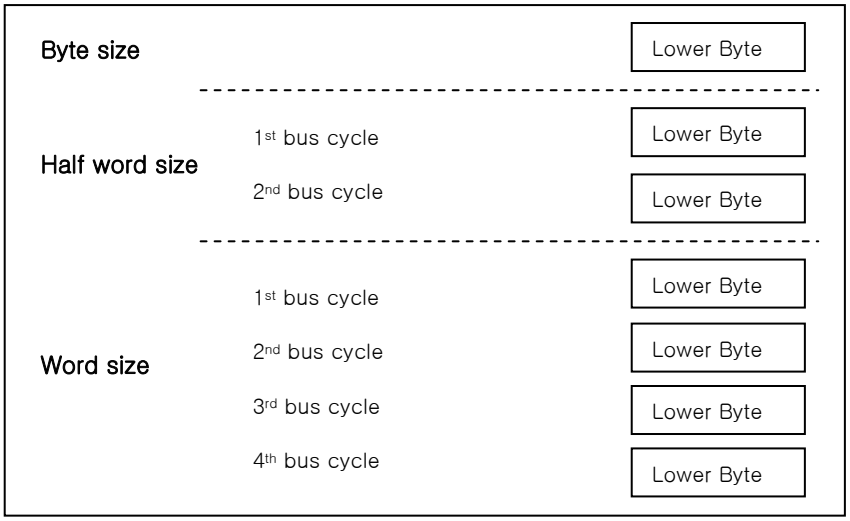


Figure 8-2. Data flow at 8-bit width memory

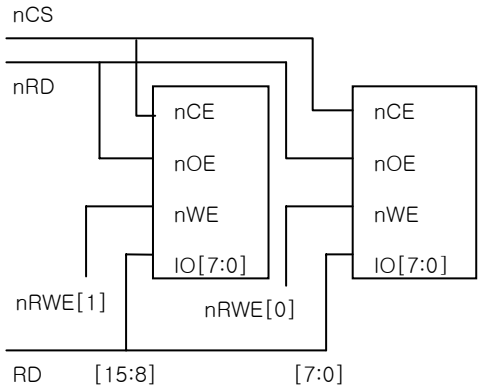


Figure 8-3. 16-bit bank configuration with 8-bit width memory

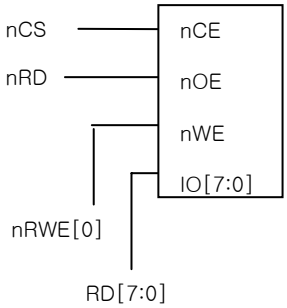


Figure 8-4. 8-bit bank configuration with 8-bit width memory

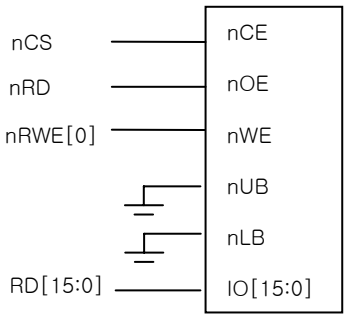


Figure 8-5. 16-bit bank configuration with 16-bit width memory

8.4 Read, Write Timing Diagram for External Memory

8.4.1 Read Access Timing (Single mode)

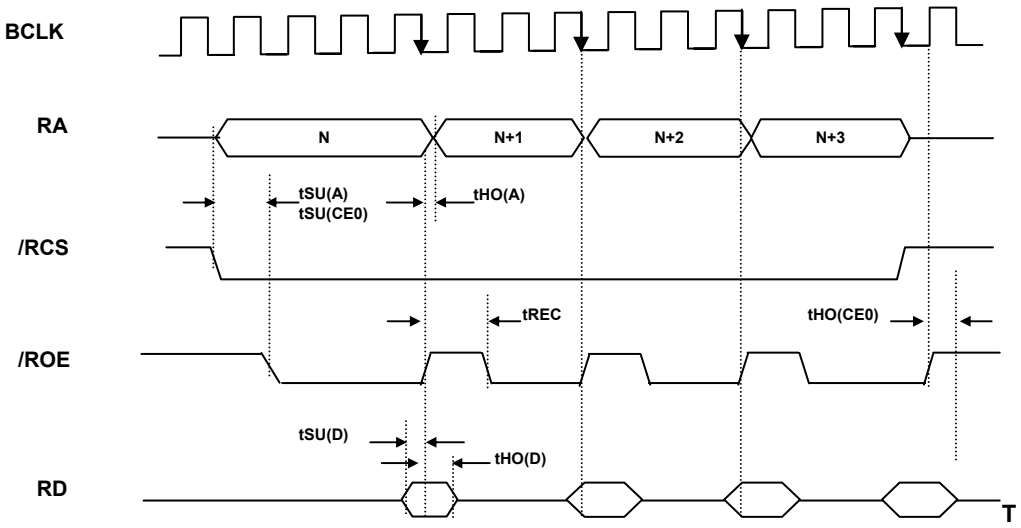


Figure 8-1 Read Access Timing (Single Mode)

Name	Description	Min	Typical	Unit	Note
tSU(A)	Address to /ROE falling-edge setup time		30	ns	
tHO(A)	/ROE rising-edge to Address hold time		0		
tSU(CE0)	/RCS falling-edge to /ROE falling-edge setup time		30		
tHO(CE0)	/ROE rising-edge to /RCS rising-edge setup time		-15		
tREC	/ROE negate to start of next cycle	30			
tSU(D)	Data setup time before latch	5			
tHO(D)	Data hold time after latch	0			

Table 8-2. Timing values for read access in single mode data transfer (BCLK=33MHz)

8.4.2 Read Access Timing (Burst mode)

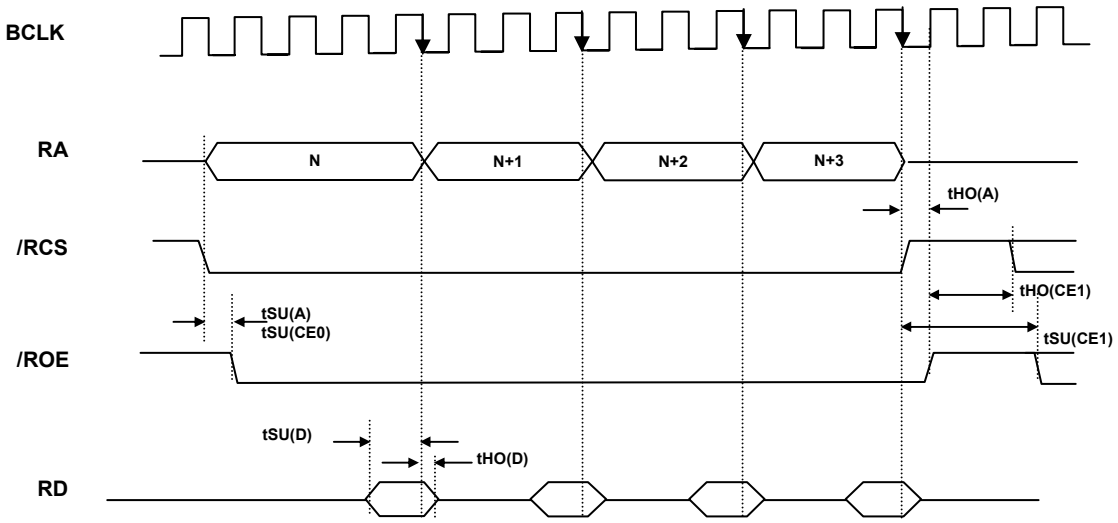


Figure 8-2 Read Access Timing (Burst Mode)

Name	Description	Min	Typical	Unit	Note
tSU(A)	Address to /ROE falling-edge setup time		15	ns	
tHO(A)	/ROE rising-edge to Address hold time		-15		
tSU(CE0)	/RCS falling-edge to /ROE falling-edge setup time		15		
tHO(CE0)	/ROE rising-edge to /RCS rising-edge setup time		-15		
tHO(CE1)	/ROE or /RWE rising-edge to /RCS falling-edge hold time	45			
tSU(CE1)	/RCE rising-edge to /ROE or /RWE falling-edge setup time	75			
tSU(D)	Data setup time before latch	5			
tHO(D)	Data hold time after latch	0			

Table 8-3. Timing values for read access in burst mode data transfer (BCLK=33MHz)

8.4.3 Write Access Timing

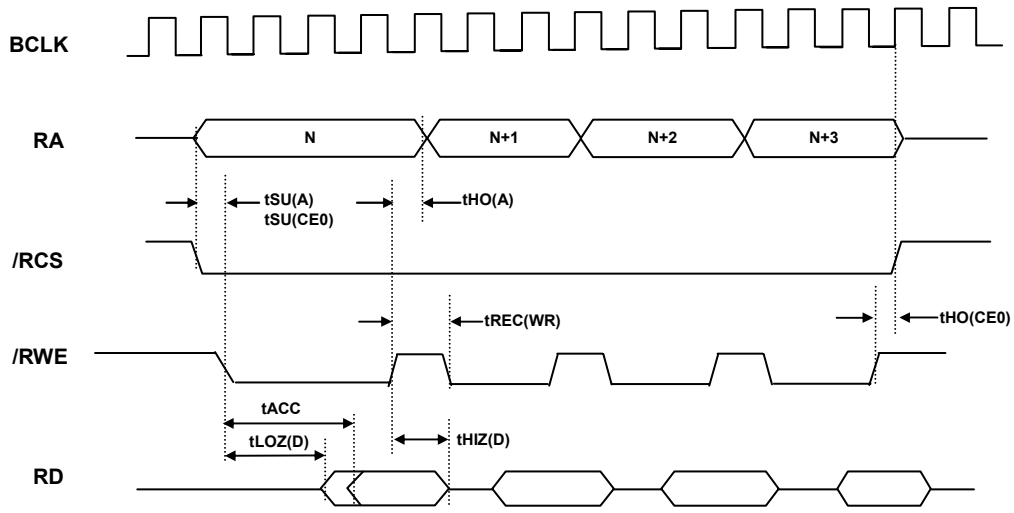


Figure 8-3 Write Access Timing

Name	Description	Min	Typical	Unit	Note
tSU(A)	Address to /RWE falling-edge setup time		15	ns	
tHO(A)	/RWE rising-edge to Address hold time		15		
tSU(CE0)	/RCS falling-edge to /RWE falling-edge setup time		15		
tHO(CE0)	/RWE rising-edge to /RCS rising-edge setup time		15		
tREC(WR)	/RWE negate to start of next cycle	30			
tHIZ(D)	/RWE rising edge to D Hi-Z delay		30		
tLOZ(D)	/RWE falling-edge to D driven	0			

Table 8-4. Timing values for write access (BCLK=33MHz)

9 AMBA PERIPHERALS

This chapter describes the peripherals that are connected to the 3.692308MHz internal peripheral bus; these are peripherals that need relatively low data rates on the internal bus. (call APB)

9.1 LCD CONTROLLER

FEATURES

- Single panel color and monochrome STN displays
- Resolution programmable up to 640x480
- Single panel STN displays with either 4- or 8-bit interfaces
- 8 and 12 bits per pixel for color display
- 1, 2, and 4 bits per pixel for monochrome display
- Big and little endian pixel order in a byte.
- Palette for 256 colors and 15 gray-level monochrome
- Programmable timing for various display panels
- Patented grayscale algorithm
- Relocatable frame buffer for Internal SRAM and SDRAM

Note. The controller does not support dual panel STN displays. There is no hardware cursor support, since WinCE does not use a cursor.

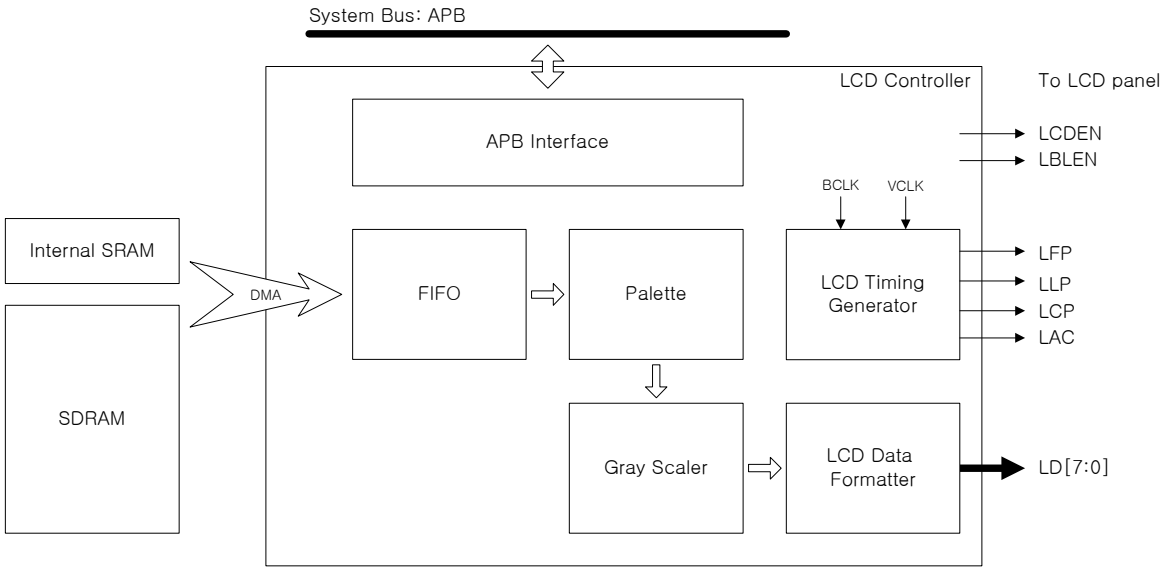


Figure 9-1. Block digram of LCD controller

9.1.1 External Signals

Pin Name	Type	Description	
LCDEN	O	Power on/off signal for a LCD panel	
LBLEN	O	Backlight enable signal for a LCD panel	
LFP	O	LCD frame pulse	(corresponds to FRAME pin of a LCD panel)
LLP	O	LCD line pulse	(corresponds to CL1 pin of a LCD panel)
LCP	O	LCD clock pulse	(corresponds to CL2 pin of a LCD panel)
LAC	O	LCD AC bias	
LD[7:0]	O	LCD data bus	

Refer to Figure 2-1. 208 Pin diagram.

9.1.2 Registers

Address	Name	Width	Default	Description
0x8005.2000	LcdControl	16	0000.0000	LCD Control Register
0x8005.2004	LcdStatus	4	0000.0000	LCD Status Register
0x8005.2008	LcdStatusM	4	0000.0000	LCD Status Mask Register
0x8005.200C	LcdInterrupt	4	0000.0000	LCD Interrupt Register
0x8005.2010	LcdDBAR	32	0000.0000	LCD DMA Channel Base Address Register
0x8005.2014	LcdDCAR	32	0000.0000	LCD DMA Channel Current Address Register
0x8005.2020	LcdTiming0	32	0000.0000	LCD Timing 0 Register
0x8005.2024	LcdTiming1	32	0000.0000	LCD Timing 1 Register
0x8005.2028	LcdTiming2	32	0000.0000	LCD Timing 2 Register
0x8005.2030	LcdPaletteR	32	7654.3210	LCD Palette for Red Color or LSP
0x8005.2034	LcdPaletteG	32	FEDC.BA98	LCD Palette for Green Color or MSP
0x8005.2038	LcdPaletteB	16	0000.FA50	LCD Palette for Blue Color

Table 9-1. LCD Controller Register Summary

9.1.2.1 LCD Control Register (LcdControl)

0x80052000

		13	12		10	9	8
		VCOMP			LEP	BPP	
	6	5	4		2	1	0
	BGR	LDW	BW		BLEN	PWREN	LCDEN

Bits	Type	Function
31:14	-	Reserved
13:12	R/W	VCMODE (Vertical Compare Mode) Generate interrupt at: 00 - start of VSYNC 01 - start of BACK PORCH 10 - start of ACTIVE VIDEO 11 - start of FRONT PORCH
11	-	Reserved
10	R/W	LEP (Little Endian Pixel) 0 - big endian pixel order in a byte 1 - little endian pixel order in a byte
9:8	R/W	BPP (Bits Per Pixel) 00 - 1bpp 01 - 2bpp 10 - 4bpp 11 - 8bpp (for color display only)
7	-	Reserved
6	R/W	BGR (Blue-Green-Red mode for color mode) 0 - RGB normal video output for LCD 1 - BGR red and blue swapped for LCD
5	R/W	LDW (LCD Data bus Width for monochrome mode) 0 - 4-bit data width LCD module 1 - 8-bit data width LCD module
4	R/W	BW (monochrome or color display mode) 0 - Color operation enabled 1 - Monochrome operation enabled
3	-	Reserved
2	R/W	BLEN (LCD Backlight Enable) This drives "0" or "1" out to the LCD backlight enable pin
1	R/W	PWREN (LCD Power Enable) 0 - LCD is off 1 - LCD is on when LcdEn=1
0	R/W	LCDEN (LCD Controller Enable) 0 - LCD controller disabled 1 - LCD controller enabled

9.1.2.2 LCD Controller Status/Mask and Interrupt Registers (LcdStatus, LcdStatusM, and LcdInterrupt)

0x80052004 ~ 0x8005200C

				3	2	1	0
				LDONE	VCOMP	LNEXT	LFUF

Bits	Type	Function
31:4	-	Reserved
3	R	LDONE (LCD Done frame status/mask/interrupt bit) The LCD Frame Done (Done) is a read-only status bit that is set after the LCD has been disabled (LcdEn = 0) and the frame that is current active finishes being output to the LCD's data pins. It is cleared by writing the base address (LcdDBAR) or enabling the LCD, or, by writing "1" to the LDone bit of the Status Register. When the LCD is disabled by clearing the LCD enable bit (LcdEn=0) in LcdControl, the LCD allows the current frame to complete before it is disabled. After the last set of pixels is clocked out onto the LCD's data pins by the pixel clock, the LCD is disabled and Done is set.
2	R/W	VCOMP (Vertical Compare status/mask/interrupt bit) This bit is set when the LCD timing generator reaches the vertical region, VCOMP, programmed in the Video Control Register. This bit is "sticky", meaning it remains set until it is cleared by writing a "1" to this bit
1	R	LNEXT (LCD Next base address update status/mask/interrupt bit) The LCD Next Frame (LNext) is a read-only status bit that is set after the contents of the LCD DMA base address register are transferred to the LCD DMA current address register at the start of frame, and it is cleared when the LCD DMA base address register is written.
0	R/W	LFUF (FIFO Underflow status/mask/interrupt bit) The LCD FIFO underflow (LFUF) status bit is set when the LCD FIFO under-runs. The status bit is "sticky", meaning it remains set after the FIFO is no longer underrunning. The status bit is cleared by writing a `1' to this bit.

9.1.2.3 LCD DMA Base Address Register (LcdDBAR)

0x80052010

31	30	29	28	27	26	25	24	23	22	21	20	29	28	17	16
0	LcdDBAR														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LcdDBAR (continued)										0	0	0	0	0	0

Bits	Type	Function
31	-	Reserved. Keep these bits zero
30:6	R/W	LCD DMA Channel Base Address Pointer 16-word aligned base address of the frame buffer (SDRAM or Internal SRAM)
5:0	-	Reserved. Keep these bits zero

9.1.2.4 LCD DMA Channel Current Address Register (LcdDCAR)

0x80052014

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

Bits	Type	Function
31	-	Read as zero
30:6	R	LCD DMA Channel Current Address Pointer 16-word aligned current address pointer to data in frame buffer currently being displayed
5:0	-	Read as zero

9.1.2.5 LCD Timing 0 Register (LcdTiming0)

0x80052020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HBP								HFP							
15	14	13	12	11	10	9	8		6	5	4	3	2	1	0
HSW									PPL						

Bits	Type	Function
31:24	R/W	HBP (Horizontal Back Porch) The 8-bit HBP field is used to specify the number of pixel clock periods to insert at the beginning of each line or row of pixels. After the line clock for the previous line has been negated, the value in HBP is used to count the number of pixel clocks to wait before starting to output the first set of pixels in the next line. HBP generates a wait period ranging from 1-256 pixel clock cycles (Number of LCLK clock periods to add to the beginning of a line transmission before the first set of pixels is output to the display minus 1). HBP = # of LCLK – 1
23:16	R/W	HFP (Horizontal Front Porch) The 8-bit HFP field is used to specify the number of pixel clock periods to insert at the end of each line or row of pixels before pulsing the line clock pin. Once a complete line of pixels is transmitted to the LCD driver, the value in HFP is used to count the number of pixel clocks to wait before pulsing the line clock. HFP generates a wait period ranging from 1-256 pixel clock cycles. (Program to value required minus 1). HFP = # of LCLK – 1
15:8	R/W	HSW (Horizontal Sync Pulse Width) The 8-bit HSW field is used to specify the pulse width of the line clock. Number of LCLK clock periods to pulse the line clock at the end of each line minus 1 HSW = # of LCLK – 1
7	-	Reserved
6:0	R/W	PPL (Pixels Per Line) PPL is used to specify the number of pixels in each line or row on the screen. PPL is a 7-bit value that represents between 16-2048 pixels per line. PPL is used to count the correct number of pixel clocks that must occur before the line clock can be pulsed. Program the value required divided by 16, minus 1. PPL = (actual_pixels_per_line / 16) – 1

9.1.2.6 LCD Timing 1 Register (LcdTiming1)

0x80052024

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

Bits	Type	Function
31:16	R/W	Reserved. Keep these bits zero
15:10	R/W	VSW (Vertical Sync Pulse Width) The 6-bit VSW field is used to add extra dummy line clock delays between frames. The value should be small for STN LCD, but should be long enough to re-program the video palette under interrupt control, without writing the video palette at the same time as video is being displayed. The register is programmed with the number of lines of VSync minus 1. VSW = # of lines – 1
9:0	R/W	LPS (Lines Per Screen) The LPS bit-field is used to specify the number of lines or rows per LCD panel being controlled. LPS is a 10-bit value that represents 1-1024 Lines Per Screen. The register is programmed with the number of lines per screen minus 1. LPS = actual_lines_per_screen – 1

9.1.2.7 LCD Timing 2 Register (LcdTiming2)

0x80052028

31	30	29	28				24	23	22	21	20			17	16
IAC	ICP	ILP	IFP				ACB							CPL	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CPL (continued)								LCS	CSD		PCD				

Bits	Type	Function
31	R/W	IAC (Invert LAC pin) The IAC bit is used to invert the polarity of the LAC signal. 0 - LAC pin is active HIGH and inactive LOW 1 - LAC pin is active LOW and inactive HIGH
30	R/W	ICP (Invert LCP pin) The ICP bit is used to select which edge of the pixel clock pixel data is driven out onto the LCD's data lines. When IPC=0, data is driven onto the LCD's data lines on the rising-edge of LCP. When IPC=1, data is driven onto the LCD's data lines on the falling-edge of LCP. 0 - Data is driven on the LCD's data lines on the rising-edge of LCP. 1 - Data is driven on the LCD's data lines on the falling-edge of LCP.
29	R/W	ILP (Invert LLP pin) The ILP bit is used to invert the polarity of the LLP signal. 0 - LLP pin is active HIGH and inactive LOW. 1 - LLP pin is active LOW and inactive HIGH.
28	R/W	IFP (Invert LFP pin) The IFP bit is used to invert the polarity of the LFP signal. 0 - LFP pin is active HIGH and inactive LOW. 1 - LFP pin is active LOW and inactive HIGH.
27:25	-	Reserved
24:20	R/W	ACB (AC Bias pin frequency) The 5-bit ACB field is used to specify the number of line clock periods to count between each toggle of the AC-bias pin (LAC). This pin is used to periodically invert the polarity of the power supply to prevent DC charge build-up within the display. The value programmed is the number of lines between transitions, minus 1. ACB = # of lines – 1
19:18	-	Reserved
17:8	R/W	CPL (Clocks Per Line) This is the actual number of clocks output to the LCD panel each line, minus 1. This must be programmed, in addition to the PPL field in the LCD Timing 0 Register. The number of clocks per line is the number of pixels per line divided by 4, 8 or two-and-two-thirds for mono 4-bit mode, mono 8-bit, or color STN mode ($2\frac{2}{3}$) respectively. CPL = actual_clocks_per_line – 1
7	R/W	LCS (LCD Clock Source selection) 0 - System bus clock (BCLK) 1 - Video clock from PMU (VCLK)
6:5	R/W	CSD (LCD Clock Source Divisor) The selected clock by LCS bit is divided by LCD pre-divider. The divided source clock becomes the fundamental clock of LCD controller, LCLK. 00 – no division 01 – clock is divided by 4 10 – clock is divided by 16 11 – reserved
4:0	R/W	PCD (Pixel Clock Divisor) PCD is used to specify the frequency of LCP signal based on LCLK frequency. Pixel clock frequency can range from LCLK/2 to LCLK/33, where LCLK is the clock divided by CSD. $f_{LCP} = f_{LCLK} / (PCD + 2)$ Note that f_{LCP} is not the frequency of some nominal clock rate that individual pixels are output to the LCD. In normal mono mode (4-bit interface), four pixels are output per LCP cycle, so the PixelClock is one quarter the nominal pixel rate. In the case of 8-bit interface, PixelClock is one-eighth the nominal pixel rate, since 8 pixels are output per LCP cycle. In the case of color, PixelClock is 0.375 times the nominal pixel rate, because $2\frac{2}{3}$ pixels are output per LCP cycle.

9.1.2.8 LCD Palette Registers (LcdPaletteR, LcdPaletteG, LcdPaletteB, LcdPaletteLSP, and LcdPaletteMSP)

0x80052030 LcdPaletteR (LcdPaletteLSP for monochrome display)

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

0x80052034 LcdPaletteG (LcdPaletteMSP for monochrome display)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Palette value for pixel value 7 or pixel value 15 for mono disp				Palette value for pixel value 6 or pixel value 14 for mono disp				Palette value for pixel value 5 or pixel value 13 for mono disp				Palette value for pixel value 4 or pixel value 12 for mono disp			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Palette value for pixel value 3 or pixel value 11 for mono disp				Palette value for pixel value 2 or pixel value 10 for mono disp				Palette value for pixel value 1 or pixel value 9 for mono disp				Palette value for pixel value 0 or pixel value 8 for mono disp			

0x80052038 LcdPaletteB

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Palette value for pixel value 3				Palette value for pixel value 2				Palette value for pixel value 1				Palette value for pixel value 0			

9.1.3 LCD controller datapath

User can use both internal SRAM and SDRAM for storage of LCD frame data. The base address of frame data (LcdDBAR) can be located in the internal SRAM as well as SDRAM. If the size of frame data is larger than that of the internal SRAM, the rest of data must be stored in the head of SDRAM. However, user does not have to care about it, because the head of SDRAM is seamlessly connected to the tail of the internal SRAM (refer to Memory Map). DMA of LCD controller will switch between both areas and get proper frame data from them.

FIFO is designed to store 32 words. If user chooses 1 bpp mode for pixel data width, FIFO can store 1024 pixel data at a time. One DMA operation will fill FIFO with 16 words of frame data. The frame data coming out from FIFO will be divided into each pixel or each color component for color mode. Then it is translated by palette registers. The translated pixel or color value has 4-bit width, no matter which bpp mode user chooses. Gray scaler block convert these 4 bit gun data in a single bit per gun, using a patented time/space dither algorithm.

The output of the gray scaler is fed to the LCD data formatter, which formats the pixels in the correct order for the LCD panel type in use: 4 or 8 mono pixels per clock for mono panels, or 2 ⅔ pixels per clock for color data. The output of the formatter in color mode is bursty, due to the 2 ⅔ pixels per clock that are output, so the formatter output goes to a small FIFO, which smoothes out this burstiness, before data is output to the LCD panel at a constant rate.

9.1.4 Color/Grayscale dithering

Entries selected from the look-up palette are sent to the color/grayscale space/time base dither generator. Each 4-bit value is used to select one of 15 intensity levels.
Note that two of the 16 dither values are identical. The table below assumes that a pixel data input to the LCD panel is active HIGH. That is, a '1' in the pixel data stream will turn the pixel on, and a '0' will turn it off. If this is not the case, the intensity order will be reversed, with "0000" being the most intense color. This polarity is LCD panel dependent.

The gray/color intensity is controlled by turning individual pixels on and off at varying periodic rates. More intense grays/colors are produced by making the average time that the pixel is off longer than the average time that it is on. The proprietary dither algorithm is optimized to provide a range of intensity values that match the eye's visual perception of color/gray gradations, with smaller changes in intensity nearer to the mid-gray level, and greater nearer the black and the white levels. In color mode, red, green and blue components are gray-scaled simultaneously as if they were mono pixels. The duty cycle and resultant intensity level for all 15 color/grayscale levels is summarized in Table 9-1: Color/grayscale intensities and modulation rates.

Dither Value (4 bit value from palette)	Intensity (0% is white)	Modulation Rate (ration of ON to ON+OFF pixels)
1111	100.0	1
1110	100.0	1
1101	88.9	8/9
1100	80.0	4/5
1011	73.3	11/15
1010	66.6	6/9
1001	60.0	3/5
1000	55.6	5/9
0111	50.0	1/2
0110	44.4	4/9
0101	40.0	2/5
0100	33.3	3/9
0011	26.7	4/15
0010	20.0	1/5
0001	11.1	1/9
0000	0.0	0

Table 9-2. LCD Color/Grayscale Intensities and Modulation Rates

9.1.5 LCD panel dependent settings

These registers need to be set carefully according to a LCD panel specification.

- BW : Monochrome or color display
- BGR : RGB or BGR for color display
- LDW : LCD Data bus width
- IFP, ILP, ICP, IAC : Signal polarity
- PPL, CPL, LPS : Resolution
- LCS, CSD, PCD : Fundamental clock
- VFP, VBP, VSW, HFP, HBP, HSW, ACB : Control timing
- PWREN, BLE : LCD panel on/off control

If a LCD panel is monochome, set **BW** as 1. For a color LCD panel, set **BW** as 0.

In the case of a color LCD panel, the sequence of color components in a pixel can differ by product. Most panels have red as the first color components of a pixel and blue as the last one. In this case, set **BGR** as 0. If **BGR** is set as 1, LCD controller displays a blue component in the first and green and red in a row. Hence, you can display a image without changing the original data to a LCD panel with the different color sequence.

LCD controller supports 8-bit data bus for a color LCD panel. However, for a mono LCD panel, 4-bit and 8-bit data bus are possible. If you set **LDW** as 0, LCD controller displays pixels through LD[3:0]. Set **LDW** as 1 to display though LD[7:0]. The pixel display sequence is depicted in Figure 9-2. The first pixel is output to the MSB of LD. In the color display mode, the first color component is displayed in the first.

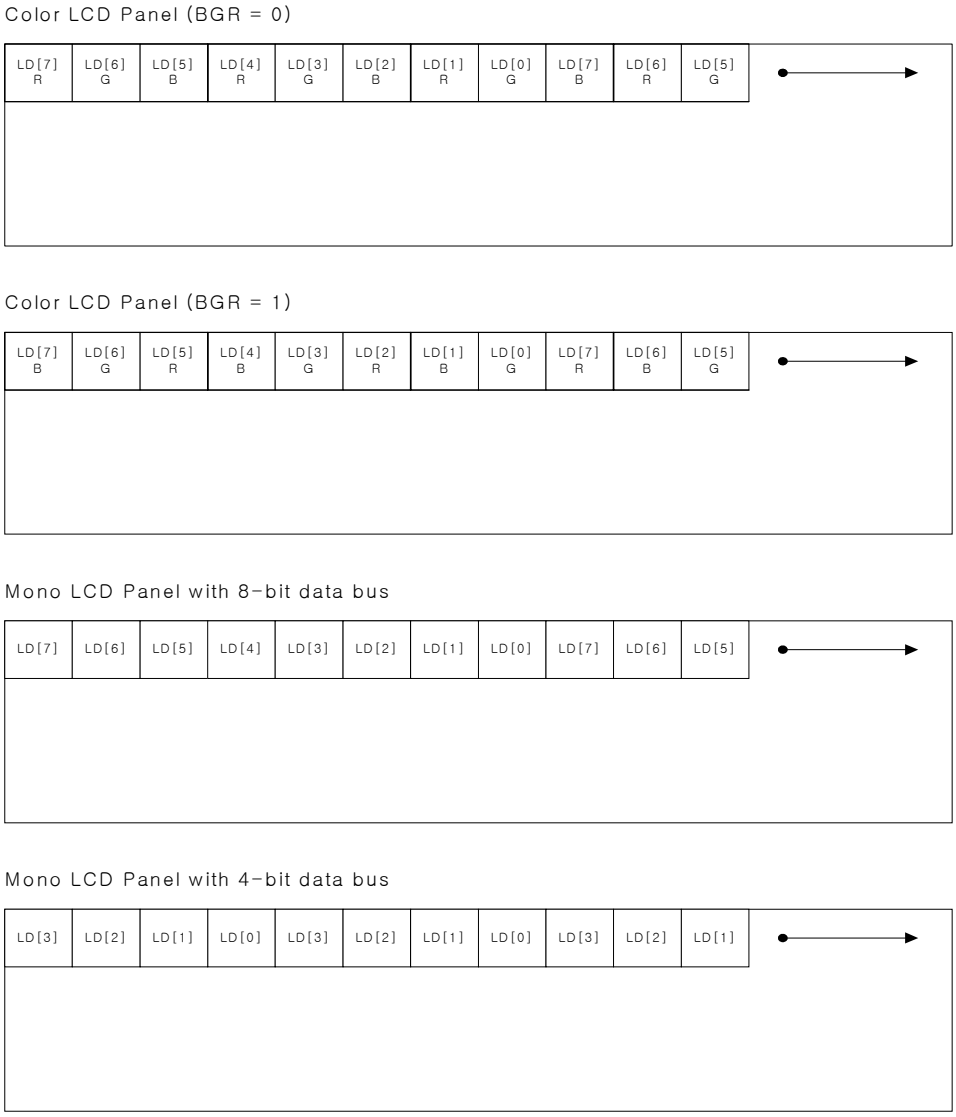


Figure 9-2. Pixel display sequence of LD bus

The LCD panel signals, LFP, LLP, LCP, LAC, LD, LCDEN, and LBLEN, are active high. Hence, the timing diagrams for the signals are shown as active high signals. However, some LCD panels have active low signals. To display images in such panels without any glue logics, LCD controller can program a polarity of each signal. If set **IFP** as 1, LFP pin becomes active low and it is driven low at the start of a new frame. **ILP**, **ICP**, and **IAC** work likewise. It is depicted in Figure 9-3.

ICP can be used to adjust timing of the LCD panel signals. LCD controller drives the signals at the rising edge of LCP when **ICP = 0**. It is to stable the signals at the falling edge of LCP because of most LCD panels read the signals at that time. However, if the timing of LCD panel signals is changed by glue logics such as a voltage level shifter or a LCD panel read the signals at the rising edge of LCP, you can use **ICP** to ensure the timing margin for such cases. The LCD panel signals are driven at the falling edge of LCP when set **ICP** as 1.

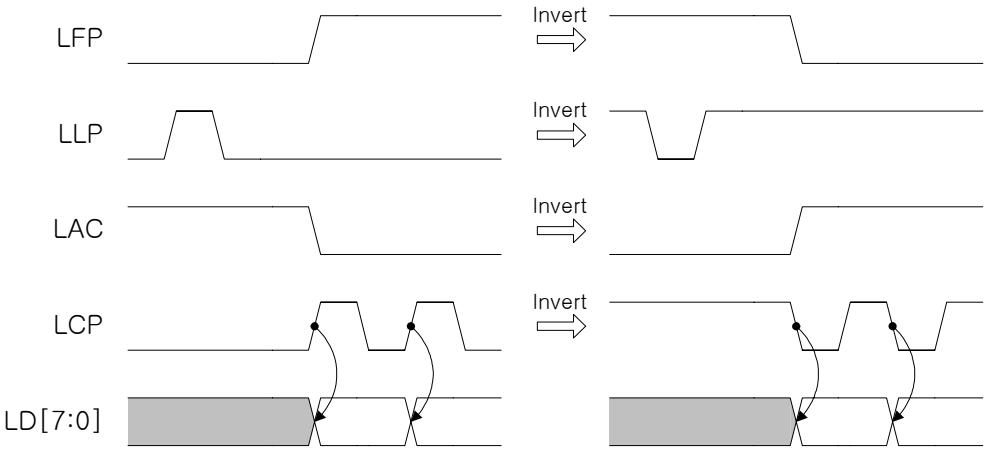


Figure 9-3. Changing polarity of LCD panel signals

PPL is to set the number of pixels in each line.
PPL = (actual_pixels_per_line / 16) - 1

CPL is to set the number of clocks in each line. It is different to **PPL** because STN LCD panels display several pixels for a clock. **CPL** can be calculated as follows:

	BW = 0 (Mono)	**BW = 1 (Color)**
LDW = 0 (4-bit data bus)	(actual_pixels_per_line / 4) - 1	-
LDW = 1 (8-bit data bus)	(actual_pixels_per_line / 8) - 1	(actual_pixels_per_line x 3 / 8) - 1

LPS is to set the numer of lines per screen.
LPS = actual_lines_per_screen - 1

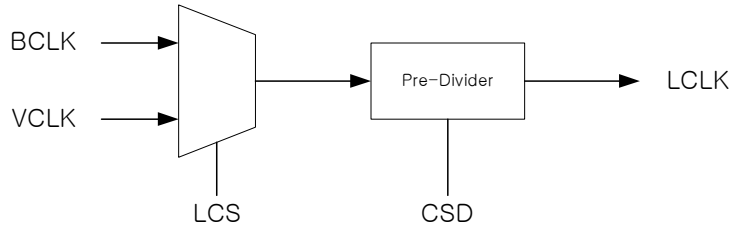


Figure 9-4. Block diagram of clock source generation

LCD controller can have two different clock sources to generate LCD panel signals. If you want to use system bus clock, BCLK, set **LCS** as 0. VCLK also can be used which is generated In PMU. In this case, set **LCD** as 1.
Before the selected clock source is used in LCD controller, it is divided by **CSD**. After the division, LCLK is the fundamental clock that is used to generated LCD panel signals. The frequency of LCLK is as follows:

	LCS = 0 (BCLK)	**LCS = 1 (VCLK)**
CPD = 00 (No division)	f_{BCLK}	f_{VCLK}
CPD = 01 (1/4 division)	f_{BCLK} / 4	f_{VCLK} / 4
CPD = 10 (1/16 division)	f_{BCLK} / 16	f_{VCLK} / 16
CPD = 11 (Reserved)	Unknown	Unknown

PCD is to set the frequence of LCP.
$f_{LCP} = f_{LCLK} / (\mathbf{PCD} + 2)$
Hence, the period of LCP is (**PCD** + 2) times to the period of LCLK.
$t_{LCP} = t_{LCLK} \times (\mathbf{PCD} + 2)$
To ensure proper operation of LCD controller, there is lower bound value of PCD.

	BW = 0 (Mono)	**BW = 1 (Color)**
LDW = 0 (4-bit data bus)	PCD >= 2	-
LDW = 1 (8-bit data bus)	PCD >= 6	PCD >= 2

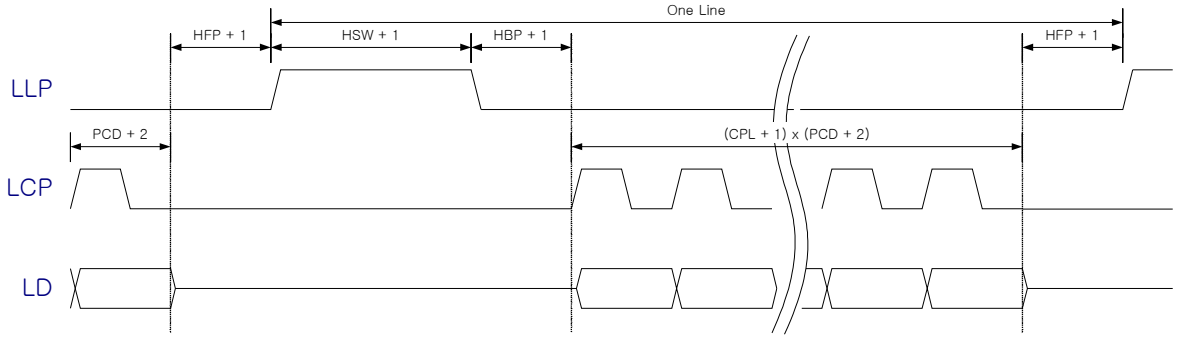


Figure 9-5. Timing diagram of a line with LLP, LCP, and LD signals

Figure 9-5 shows the timing diagram of a line displayed by LCD controller. The unit of dimension is the period of LCLK. **PCD** controls LCP signal as explained above. And **HFP**, **HSW**, and **HBP** control LLP signal.
The period and frequence of a line can be calculated:

$t_{line} = (t_{LCP} \times (\mathbf{CPL} + 1)) + (t_{LCLK} \times (\mathbf{HFP} + 1 + \mathbf{HSW} + 1 + \mathbf{HBP} + 1))$
$= t_{LCLK} \times ((\mathbf{CPL} + 1) \times (\mathbf{PCD} + 2) + (\mathbf{HFP} + 1 + \mathbf{HSW} + 1 + \mathbf{HBP} + 1))$
$f_{line} = f_{LCLK} / ((\mathbf{CPL} + 1) \times (\mathbf{PCD} + 2) + (\mathbf{HFP} + 1 + \mathbf{HSW} + 1 + \mathbf{HBP} + 1))$

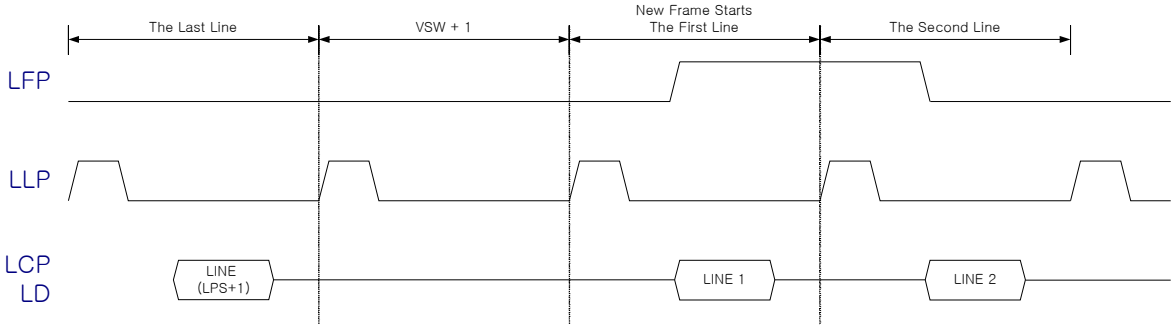


Figure 9-6. Timing diagram of LFP signal

Figure 9-6 shows the timing diagram of LFP signals that is controlled by **VSW**. The unit of dimension is the period of a line. LCP and LD signal are drawn as simplified. Every new frame starts with active LFP.
The period and frequence of a frame are:

t_{frame} = t_{line} x (**LPS** + 1 + **VSW** + 1)
= t_{LCLK} x { ((**CPL** + 1) x (**PCD** + 2) + (**HFP** + 1 + **HSW** + 1 + **HBP** + 1))
x (**LPS** + 1 + **VSW** + 1) }
f_{frame} = f_{line} / (**LPS** + 1 + **VSW** + 1)
= f_{LCLK} / { ((**CPL** + 1) x (**PCD** + 2) + (**HFP** + 1 + **HSW** + 1 + **HBP** + 1))
x (**LPS** + 1 + **VSW** + 1) }

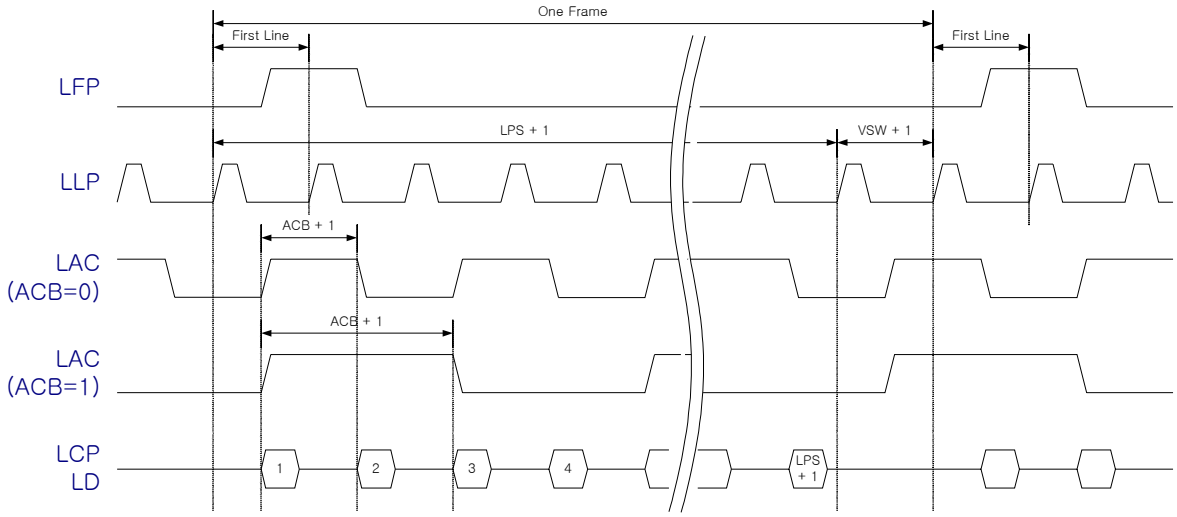


Figure 9-7. Timing diagram of a frame be different by the differ

Figure 9-7 depicts the complete waveform of a frame. The unit of dimension is the period of a line.
You can choose **ACB** to toggle the bias level of a LCD panel. If a LCD panel uses LAC pin, the value must be carefully determined to ensure the average bias level of LAC as 0. If the average bias is not 0, the LCD panel may suffer long-term damage. To avoid this, the total line number, (**LPS** + 1 + **VSW** + 1), should not be the integer multiple propotion of 2 x (**ACB** + 1).

9.1.6 Frame data dependent settings

- LcdDBAR : Frame memory address
- BPP : Bits per pixel
- LEP : Endian mode in a byte
- LcdPaletteR, LcdPaletteG, LcdPaletteB, LcdPaletteMSP, LcdPaletteLSP : Palette data

The LCD DMA base address register (**LcdDBAR**) is a read/write register used to specify the base address of the off-chip frame buffer for the LCD. Addresses programmed in the base address register must be aligned on sixteen-word boundaries, thus the least significant six bits (**LcdDBAR** [5:0]) must always be written with zeros. 31 bits of the register, including the LS 6 bits which must be zero, are valid, because LCD DMA is allowed from SDRAM and the internal SRAM. The most significant bit of LcdDBAR is assumed as '0'.

User must initialize the base address register before enabling the LCD, and may also write a new value to it while the LCD is enabled to allow a new frame buffer to be used for the next frame. The user can change the state of **LcdDBAR** while the LCD controller is active, after the next frame status bit (**LNEXT**) is set within the LCD's status register that generates an interrupt request. This status bit indicates that the value in the base address pointer has been transferred to the current address pointer register and that it is safe to write a new base address value. This allows double-buffered video to be implemented if required.

The LCD palette registers are a set of two word and one half-word registers that allow the LCD to be programmed. These registers are used for both color and monochrome display. The format of the palette data is shown below.
In the color display mode, **LcdPaletteR** register translates pixel values for red color component. **LcdPaletteG** and **LcdPaletteB** translate for green and blue color component, respectively. For 8 bpp pixel data, each color component will be unpacked from one byte, as shown below. For 1, 2, and 4 bpp, color components will not be distinguished and whole pixel data will be translated by each palette register.

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
Red			Green			Blue	

In the monochrome display mode, **LcdPaletteR**(**LcdPaletteLSP**) is used for 8 least significant pixel values and **LcdPaletteG**(**LcdPaletteMSP**) for 8 most significant pixel values. It is because maximum 16 palette values are required to translate pixel values in a mono 4 bpp mode. For an example, if a pixel represents 11 in the 4 bpp mode, it will be translated to the value of **LcdPaletteMSP**[15:12]. For 1 and 2 bpp pixel data, **LcdPaletteMSP** and a part of **LcdPaletteLSP** which have no correspondences will be ignored. In the monochrome display mode, **LcdPaletteB** does nothing.

HMS30C7210 is basically little endian. LCD frame data also follows little endian. However, user can choose the alignment of pixel data in a byte. Figure 9-8 shows display order against the pixel alignment chosen by **LEP**. For 1 and 4 bpp mode, the pixel alignment also follows the same manner as depicted in Figure 9-8.

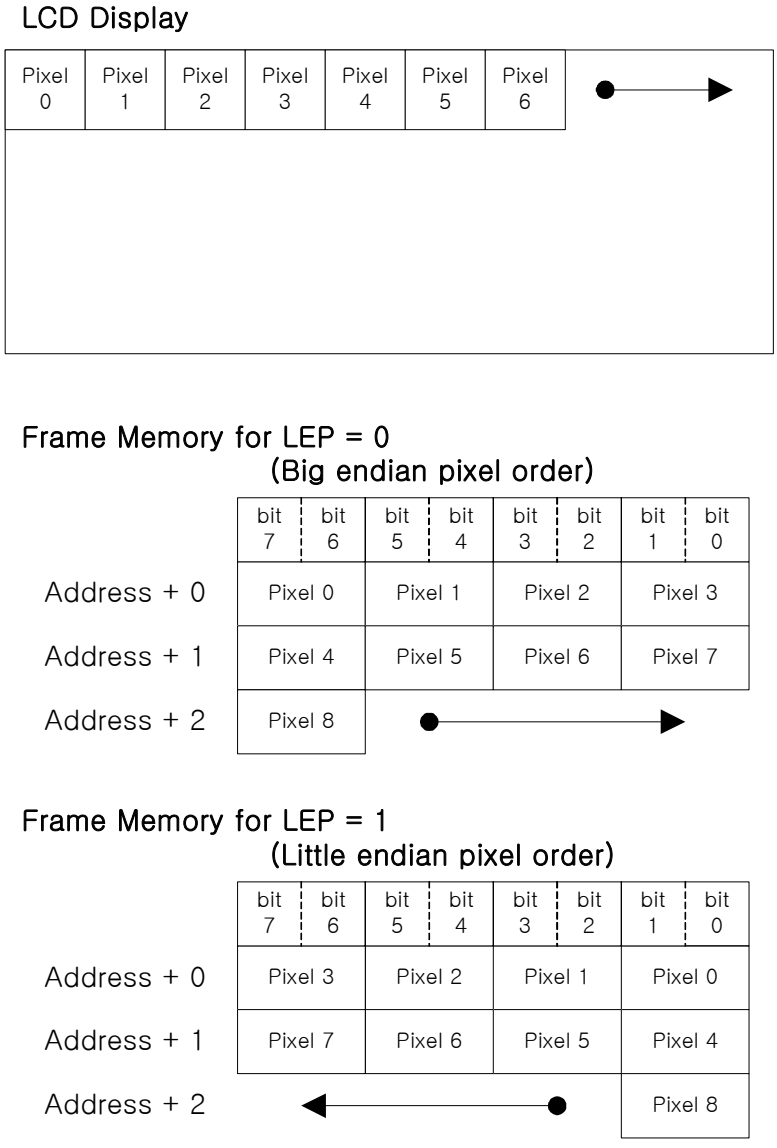


Figure 9-8. Pixel Display Order for Big and Little-endian Pixel Alignment in 2-bpp Mode

9.1.7 Other settings

- BLEN, PWREN, LCDEN : Enable sequence
- LcdStatus, LcdStatusM, LcdInterrupt : Interrupt mode

LCD panels require that the LCD controller is running before power is applied. For this reason, the LCD's power on control is not set to "1" unless both **LcdEn** and **PWREN** are set to "1". Note that most LCD displays require the **LcdEn** must be set to "1" approximately 20ms before **PWREN** is set to "1" for powering up. Likewise, **PWREN** is set to "0" 20ms before **LcdEn** is set to "0" for powering down.

To change the value of this register, LCD controller must be disabled. Otherwise, LCD may display improperly. Right after disabling the controller by setting **LcdEn** to “0”, however, it may still operate until the end of displaying the current active frame. User has to refer **LDONE** bit in **LcdStatus** register to ensure that LCD controller stops the operation.

The LCD controller status, **LcdStatus**, mask, **LcdStatusM**, and interrupt registers, **LcdInterrupt**, all have the same format. Each bit of the status register is a status bit that may generate an interrupt. The corresponding bits in the mask register mask the interrupt. The interrupt register is the logical AND of the status and mask registers, and the interrupt output from the LCD controller is the logical OR of the bits within the interrupt register.

The LCD controller status register contains bits that signal an under-run error for the FIFO, the DMA next base update ready status, and the DMA done status. Each of these hardware-detected events can generate an interrupt request to the interrupt controller.

9.2 Interrupt Controller

The interrupt controller has the following features

- A status register
- Selection of the output path (IRQ or FIQ for each input)
- Enabling the interrupt

The interrupt controller provides a simple software interface to the interrupt system. In an ARM system, two levels of interrupt are available:

- FIQ (Fast Interrupt Request) for fast, low-latency interrupt handling
- IRQ (Interrupt Request) for more general interrupts

Ideally, in an ARM system, only a single FIQ source would be in use at any particular time. This provides a true low-latency interrupt, because a single source ensures that the interrupt service routine may be executed directly without the need to determine the source of the interrupt. It also reduces the interrupt latency because the extra-banked registers, which are available for FIQ interrupts, may be used to maximum efficiency by preventing the need for a context save.

The interrupt controller provides a bit position for each different interrupt source. Bit positions are defined for a software-programmed interrupt. Any interrupt source can be programmed as a source to FIQ or IRQ interrupt. All interrupt source inputs must be active HIGH and level sensitive. Neither hardware priority scheme nor any form of interrupt vectoring is provided, because these functions can be provided in software. Any interrupt source may be masked.

9.2.1 Registers

Address	Name	Width	Default	Description
0x8005.0000	ENABLE	29	0x0000000	Interrupt Enable Register
0x8005.0004	DIR	29	0x0000000	Interrupt Direction Register
0x8005.0008	STATUS	29	0x0000000	Interrupt Status Register
0x8005.000C	-	0	0x000000	Reserved for Test Only : Do not write
0x8005.0010		0	0x000000	Reserved for Test Only : Do not write
0x8005.0020	IRQFIQ	2	0x3	IRQ/FIQ Status Register

Table 9-3. Interrupt Controller Register Summary

9.2.1.1 *Enable Register: enable each interrupt source*

0x80050000

31	30	29	28	27	26	25	24
Reserved			TICK	GPIOB[15]	GPIOB[14]	GPIOE	GPIOD
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
GPIOC	GPIOB	GPIOA	KBD	2WSI	RTC	WDT	TIMER3
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TIMER2	TIMER1	TIMER0	SMC	SPI1	SPI0	UART5	UART4
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
UART3	UART2	UART1	UART0	ADC	LCD	USB	PMU

Bits	Type	Function
0:29	R/W	Each bit of this register enables/disables corresponding interrupt sources.

Bit	Interrupt Name	Description
28	TICK	RTC TICK
27	GPIOB[15]	HotSync
26	GPIOB[14]	To the Deep-sleep
25	GPIOE	GPIOE
24	GPIOD	GPIOD
23	GPIOC	GPIOC
22	GPIOB	GPIOB
21	GPIOA	GPIOA
20	KBD	Keyboard Controller
19	2WSI	2WSI
18	RTC	Real Time Clock Controller
17	WDT	Watch Dog Timer
16	TIMER3	TIMER3
15	TIMER2	TIMER2
14	TIMER1	TIMER1
13	TIMER0	TIMER0
12	SMC	SMC
11	SPI1	SPI1
10	SPI0	SPI0
9	UART5	UART5
8	UART4	UART4
7	UART3	UART3
6	UART2	UART2
5	UART1	UART1
4	UART0	UART0
3	ADC	ADC
2	LCD	LCD Controller
1	USB	USB Controller
0	PMU	Power Management Unit

0 = disable interrupt (default)
1 = enable interrupt

9.2.1.2 *Direction Register: interrupt source will trigger nIRQ or nFIQ*

0x80050004

31	30	29	28	27	26	25	24
Reserved			TICK	GPIOB[15]	GPIOB[14]	GPIOE	GPIOD
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
GPIOC	GPIOB	GPIOA	KBD	2WSI	RTC	WDT	TIMER3
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TIMER2	TIMER1	TIMER0	SMC	SPI1	SPI0	UART5	UART4
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
UART3	UART2	UART1	UART0	ADC	LCD	USB	PMU

Bits	Type	Function
0:29	R/W	Each bit of this register indicates whether it is IRQ or FIQ for corresponding interrupt sources. 0 = IRQ (default) 1 = FIQ

9.2.1.3 *Status Register: current interrupt request status (read-only)*

0x80050008

31	30	29	28	27	26	25	24
Reserved			TICK	GPIOB[15]	GPIOB[14]	GPIOE	GPIOD
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
GPIOC	GPIOB	GPIOA	KBD	2WSI	RTC	WDT	TIMER3
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TIMER2	TIMER1	TIMER0	SMC	SPI1	SPI0	UART5	UART4
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
UART3	UART2	UART1	UART0	ADC	LCD	USB	PMU

Bits	Type	Function
0:29	R	Each bit of this register indicates whether IRQ(or FIQ) is generated or not. Masked bit by Enable Register shows always '0'. 0 = No interrupt request (default) 1 = Interrupt pending

9.2.1.4 *IRQFIQ Register: current IRQ/FIQ status (read-only)*

0x80050020

7	6	5	4	3	2	1	0
Reserved						IRQ	FIQ

Bits	Type	Function
0:1	R	Bit 0 indicates current status of nFIQ. Bit 1 indicates current status of nIRQ. 0 = Request pending 1 = No request (default)

9.2.2 Interrupt Control

The interrupt controller provides interrupt request status, interrupt enable and interrupt direction selection registers. The enable register is used to determine whether or not an active interrupt source should generate an interrupt request to the processor. All bits are cleared by system reset.

The interrupt request status indicates whether or not the interrupt source is causing a processor interrupt.

The direction register is used to determine which interrupt request is generated to the CPU. If the bit is set, FIQ request is activated. All bits are cleared by system reset. TIC registers are used only for the production test. TIC Input Select Register is used to drive interrupt request sources by CPU. When this register is set, TIC register bits are regarded as interrupt sources. This bit is cleared by system reset and should be cleared in normal operation.

9.3 USB Slave Interface

This section describes the implementation-specific options of USB protocol for a device controller. It is assumed that the user has knowledge of the USB standard. This USB Device Controller (USBD) is chapter 9 (of USB specification) compliant, and supports standard device requests issued by the host. The user should refer to the Universal Serial Bus Specification revision 1.1 for a full understanding of the USB protocol and its operation. (The USB specification 1.1 can be accessed via the World Wide Web at: http://www.usb.org). The USBD is a universal serial bus device controller (slave, not hub or host controller) which supports three endpoints and can operate half-duplex at a baud rate of 12 Mbps. Endpoint 0,by default is only used to communicate control transactions to configure the USBD after it is reset or physically connected to an active USB host or hub. Endpoint 0's responsibilities include connection, address assignment, endpoint configuration and bus numeration.

The connected host that can get a device descriptor stored in USBD's internal ROM via endpoint 0 configures the USBD. The USBD uses two separate 32 x 8 bit FIFO to buffer receiving and transmitting data to/from the host. The CPU can access the USBD using Interrupt controller, by setting the control register appropriately. This section also defines the interface of USBD and CPU.

FEATURES

- Full universal serial bus specification 1.1 compliant.
- Receiver and Transceiver have 32 bytes FIFO individually (this supports maximum data packet size of bulk transfer).
- Internal automatic FIFO control logic. (According to FIFO status, the USBD generates Interrupt service request signals to the CPU)
- Supports high-speed USB transfer (12Mbps).
- There are two endpoints of transmitter and receiver respectively, totally three endpoints including endpoint 0 that has responsibility of the device configuration.
- CPU can access the internal USB configuration ROM storing the device descriptor for Hand-held PC (HPC) by setting the predefined control register bit.
- USB protocol and device enumeration is performed by internal state-machine in the USBD.
- The USBD only supports bulk transfer of 4-transfer type supported by USB for data transfer.
- Endpoint FIFO (Tx, Rx) has the control logic preventing FIFO overrun and under run error.

Note Product ID: 7210 Vendor ID: 05b4 * can be modified

9.3.1 Block Diagram

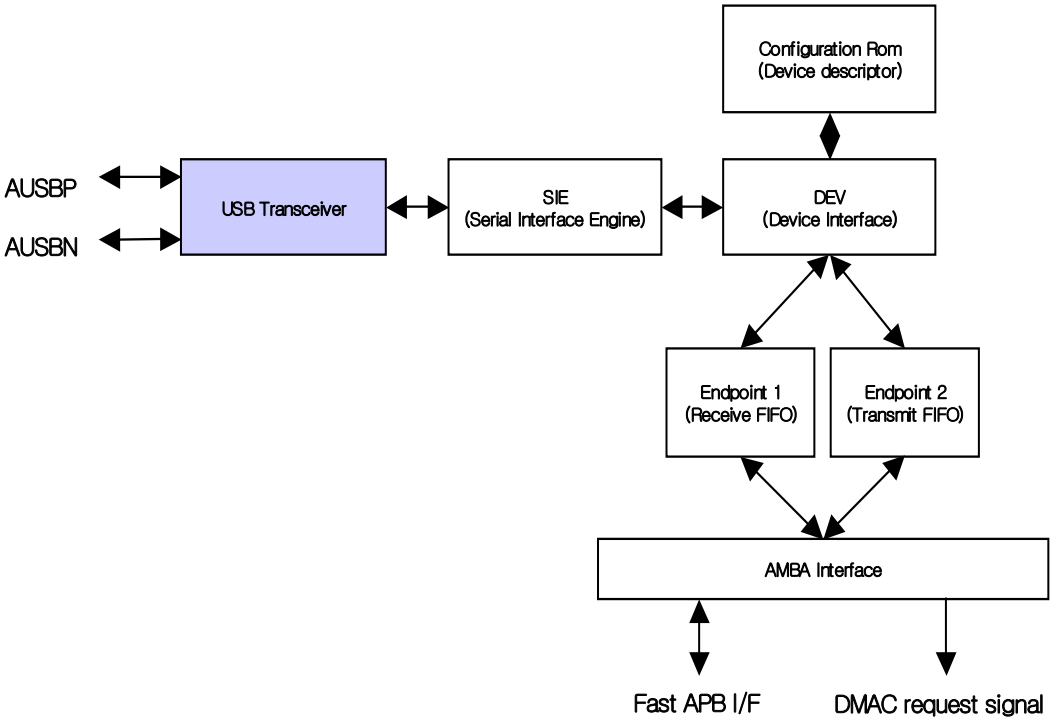


Figure 9-9. USB Block Diagram

The USB, Figure 9-9. USB Block Diagram comprises the Serial Interface Engine (SIE) and Device Interface (DEV). The SIE connects to the USB through a bus transceiver, and performs NRZI conversion, bit un-stuffing, CRC checking, packet decoding and serial to parallel conversion of the incoming data stream. In outgoing data, it does the reverse, that is, parallel to serial of outgoing data stream and packetizing the data, CRC generation, bit stuffing and NRZI generation.

The DEV provides the interface between the SIE and the device's endpoint FIFO, ROM storing the device descriptor. The DEV handles the USB protocol, interpreting the incoming tokens and packets and collecting and sending the outgoing data packets and handshakes. The endpoints FIFO (RX, TX) give the information of their status (full/ empty) to the AMBA interface and AMBA I/F enable the CPU to access the FIFO's status register and the device descriptor stored in ROM. The AMBA interface generates a FIFO read/write strobe without FIFO's errors, based on APB signal timing. In case of data transmitting through TX FIFO (when USB generates an OUT token, AMBA I/F generates Interrupt to CPU), the user should set the transmitting enable bit in the control register. If the error of FIFO (Rx: overrun, TX: under-run) occurs, the AMBA I/F cannot generate FIFO read/ write.

9.3.2 External Signals

Pin Name	Type	Description
USBP	I/O	USB transceiver signal for P+
USBN	I/O	USB transceiver signal for N+

Refer to Figure 2-1. 208 Pin diagram.

9.3.3 Registers

Address	Name	Width	Default	Description
0x8005.1000	GCTRL	4	0x0	USB Global Configuration Register
0x8005.1004	EPCTRL	21	0x0	Endpoint Control Register
0x8005.1008	INTMASK	10	0x3ff	Interrupt Mask Register
0x8005.100C	INTSTAT	20	0x0	Interrupt Status Register
0x8005.1018	DEVID	32	0x721005b4	Device ID Register
0x8005.101C	DEVCLASS	32	0xffffff	Device Class Register
0x8005.1020	INTCLASS	32	0xffffff	Interface Class Register
0x8005.1024	SETUP0	32	-	SETUP Device Request Lower Address
0x8005.1028	SETUP1	32	-	SETUP Device Request Upper Address
0x8005.102C	ENDP0RD	32	-	ENDPOINT0 Read Address
0x8005.1030	ENDP0WT	32	-	ENDPOINT0 WRITE Address
0x8005.1034	ENDP1RD	32	-	ENDPOINT1 READ Address
0x8005.1038	ENDP2WT	32	-	ENDPOINT2 WRITE Address

Table 9-4 USB Slave interface Register Summary

9.3.3.1 GCTRL

0x8005.1000

31 4	3	2	1	0
Reserved	TRANSel	WBack	Resume	DMADis

Bits	Type	Function
3	R/W	Forced SUSPEND mode setting '1' : Forced SUSPEND enable '0' : Foced SUSPEND disable. And, normal operation or normal SUSPEND enable.
2	R/W	writeback mode for Interrupt status register. '1' : writeback erase enable. '0' : writeback erase disable.
1	R/W	This Enables Remote Resume Capabilities. When This Bit Set, USB Drives remote resume signaling. Should be cleared to stop resume
0	R	DMA Disable bit. HMS30C7210 does not support DMA, so value of this bit (logic 1) is not changeable

9.3.3.2 EPCTRL

0x8005.1004

31 21	20	19	18	17 16	15	14	13	12
Reserved	CLR2	CLR1	CLR0	E2TXB	E2SND	E2NK	E2ST	E2En

11	10	9	8	7 4	3	2	1	0
E1RCV	E1NK	E1ST	E1En	E0TXB	E0NK	E0ST	E0TR	E0En

Bits	Type	Function
21	R/W	Read Ready Signal control for Endpoint 2 '1' : read ready signal operation disabled. (always not-ready) '0' : read ready signal operation enabled.
20	R/W	Clear Endpoint2 FIFO Pointer(Auto cleared by Hardware).
19	R/W	Clear Endpoint1 FIFO Pointer(Auto cleared by Hardware).
18	R/W	Clear Endpoint0 FIFO Pointer(Auto cleared by Hardware).
17~16	R/W	USB Can Transmit NON Maximum sized Packet. This Field contains the residue byte which should be transmitted.
15	R/W	This Bit enables NON Maximum sized Packet transfer. After NON maximum sized packet transfer, this bit is auto cleared and return to Maximum Packet size transfer mode.
14	R/W	When This Bit is Set, and Endpoint2 is not enabled, USB should send NAK Handshake
13	R/W	When This Bit is Set, and Endpoint2 is not enabled, USB should send STALL Handshake
12	R/W	Enable Endpoint2 as IN Endpoint
11	R/W	This bit must be zero. So only maximum packet size RX transfer mode is supported. This means RX (HOST OUT) data packet size is fixed to 32 bytes only.
10	R/W	When This Bit is Set, and Endpoint1 is not enabled, USB should send NAK Handshake
9	R/W	When This Bit is Set, and Endpoint1 is not enabled, USB should send STALL Handshake
8	R/W	Enable Endpoint1 as OUT Endpoint
7~4	R/W	This Bit Stores the Byte Count which should be transmitted to HOST when IN token is received (Exception :: When This bit is 0, 8 Byte are transferred)
3	R/W	When This Bit is Set, and Endpoint0 is not enabled, USB should send NAK Handshake
2	R/W	When This Bit is Set, and Endpoint0 is not enabled, USB should send STALL Handshake
1	R/W	When this Bit1, Endpoint0 is configured to IN endpoint. (others OUT endpoint)
0	R/W	Enable Endpoint0

9.3.3.3 INTMASK

0x8005.1008

31 10	9	8	7	6	5	4	3	2	1	0
Reserved	E0STL	SUS	RESET	E2EM	E1OV	E1FU	E0EM	E0OV	E0FU	SET

Bits	Type	Function
9	R/W	Mask Endpoint0 Stall Interrupt
8	R/W	Mask SUSPEND Interrupt
7	R/W	Mask USB Cable RESET Interrupt
6	R/W	Mask Endpoint2 Empty Interrupt
5	R/W	Mask Endpoint1 Overrun Interrupt (May not be used)
4	R/W	Mask Endpoint1 Full Interrupt
3	R/W	Mask Endpoint0 Empty Interrupt
2	R/W	Mask Endpoint0 Overrun Interrupt (May not be used)
1	R/W	Mask Endpoint0 Full Interrupt
0	R/W	Mask Endpoint0 Setup Token Received Interrupt

9.3.3.4 INTSTAT

0x8005.100C

31 20	19 14	13 0
Reserved	EP1RXBYTE	EP0RXBYTE

9	8	7	6	5	4	3	2	1	0
E0STL	SUS	RESET	E2EM	E1OV	E1FU	E0EM	E0OV	E0FU	SET

Bits	Type	Function
19~14	R/W	Currently Remained Byte In Endpoint1 Receive FIFO which should be read by HOST
13~0	R/W	Currently Remained Byte in Endpoint0 Receive FIFO which should be read by HOST
9	R/W	Endpoint0 Stall Interrupt
8	R/W	SUSPEND Interrupt
7	R/W	USB Cable RESET Interrupt
6	R/W	Endpoint2 Empty Interrupt
5	R/W	Endpoint1 Overrun Interrupt (May not be used)
4	R/W	Endpoint1 Full Interrupt
3	R/W	Endpoint0 Empty Interrupt
2	R/W	Endpoint0 Overrun Interrupt (May not be used)
1	R/W	Endpoint0 Full Interrupt
0	R/W	Endpoint0 Setup Token Received Interrupt

9.3.3.5 DEVID

0x8005.1018

Bits	Type	Function
31:0	R/W	USB Core Can Change Device ID Field by writing Appropriate Device ID Value to This Register

9.3.3.6 DEVCLASS

0x8005.101C

Bits	Type	Function
23:0	R/W	USB Core Can Change Device Class Field by writing Appropriate Device ID Value to This Register

9.3.3.7 INTCLASS

0x8005.1020

Bits	Type	Function
23:0	R/W	USB Core Can Change Interface Class Field by writing Appropriate Device ID Value to This Register

While USB device configuration process, HOST requests Descriptors.
This USB block has a hard-wired descriptor ROM, but there are 3 fields (whole 10 bytes size) user adjustable.

[DEVICE DESCRIPTOR]

* see USB spec. 1.1 (9.6 Standard USB Descriptor Definitions) for more detail

OFFSET (BYTE)	INITIAL VALUE	DESCRIPTION	ADJUSTABLE
h00	h12	length	
h01	h01	DEVICE	
h02	h00	spec version 1.00	
h03	h01	spec version	
h04	hFF	device class	YES
h05	hFF	device sub-class	YES
h06	hFF	vendor specific protocol	YES
h07	h08	max packet size	
h08	hB4	vendor id	YES
h09	h05	vendor id (05b4) for HME	YES
h0a	h02	product id	YES
h0b	h72	product id (7210) for HME7210	YES
h0c	h01	device release #	
h0d	h00	device release #	
h0e	h00	manufacturer index string	
h0f	h00	product index string	
h10	h00	serial number index string	
h11	h01	number of configurations	

* DEVID register has 32-bit width and it covers vendor id to product id (offset from h08 to h0b): DEVID [31:24] – h0b, DEVID [23:16] – h0a, DEVID [15:8] – h09, DEVID [7:0] – h08
* DEVCLASS register has 24-bit width and it covers device class to vendor specific protocol (offset from h04 to h06): DEVCLASS [23:16] – h06, DEVCLASS [15:8] – h05, DEVCLASS [7:0] – h04

[CONFIGURATION DESCRIPTOR]

OFFSET (BYTE)	INITIAL VALUE	DESCRIPTION	ADJUSTABLE
h00	h09	Length of this descriptor	
h01	h02	CONFIGURATION (2)	
h02	h20	Total length includes endpoint descriptors	
h03	h00	Total length high byte	
h04	h01	Number of interfaces	
h05	h01	Configuration value for this one	
h06	h00	Configuration - string	
h07	h80	Attributes - bus powered, no wakeup	
h08	h32	Max power - 100 ma is 50 (32 hex)	
h09	h09	Length of the interface descriptor	
h0a	h04	INTERFACE (4)	
h0b	h00	Zero based index 0f this interface	
h0c	h00	Alternate setting value (?)	
h0d	h02	Number of endpoints (not counting 0)	
h0e	hFF	Interface class, ff is vendor specific	YES
h0f	hFF	Interface sub-class	YES
h10	hFF	Interface protocol	YES
h11	h00	Index to string descriptor for this interface	
h12	h07	Length of this endpoint descriptor	
h13	h05	ENDPOINT (5)	
h14	h01	Endpoint direction (00 is out) and address	
h15	h02	Transfer type – h02 = BULK	
h16	h20	Max packet size - low : 32 byte	
h17	h00	Max packet size – high	
h18	h00	Polling interval in milliseconds (1 for iso)	
h19	h07	Length of this endpoint descriptor	
h1a	h05	ENDPOINT (5)	
h1b	h82	Endpoint direction (80 is in) and address	
h1c	h02	Transfer type – h02 = BULK	
h1d	h20	Max packet size - low : 32 byte	
h1e	h00	Max packet size – high	
h1f	h00	Polling interval in milliseconds (1 for iso)	

* see USB spec. 1.1 (9.6 Standard USB Descriptor Definitions) for more detail
* The descriptor has 4 parts : Configuration, Interface, Endpoint1, Endpoint2 (doubled lines)

[STRING DESCRIPTOR]

OFFSET	INITIAL VALUE	DESCRIPTION	ADJUSTABLE
h0	h02	size in bytes	
h1	h03	STRING type (3)	

* This index zero string descriptor means a kind of look up table. As there is no other string descriptor and as there is no further information in this descriptor, USB block does not support strings. (All string index fields are filled with zero)

9.3.3.8 SETUP0 / SETUP1

0x8005.1024 / 0x8005.1028

Bits	Type	Function
31:0	R/W	USB Core can accept vendor specific protocol command using Endpoint0. This Register contains previously received Setup Device Request Value (64-bit Wide, half in each Register)

Below is Request format from HOST when configuration.

[Standard Device Request Format]

bmRequestType	bRequest	wValue	wIndex	wLength
Byte 0	Byte 1	Byte 2 / Byte 3	Byte 4 / Byte 5	Byte 6 / Byte 7

When HOST sends request to USB device, this USB block handles a few requests by SIE (Serial Interface Engine).

This is the condition of requests which this USB SIE can handle.

- Request Type must be Standard (b00): see USB spec. 9.3 Table 9-2 'Format of Setup Data' for more detail. Offset 0 (bmRequestType field) D[6:5] (Type) ; 00 – Standard, 01 Class, 10 – Vendor, 11 – reserved.
- Request must be one of these: GET_DESCRIPTOR, SET_ADDRESS, SET_INTERFACE, SET_CONFIGURATION, GET_INTERFACE, GET_CONFIGURATION and GET_STATUS.

So for requests other than above, HMS30C7210 USB sets 9.2.5.4 INTSTAT [0] and it means HOST sent Setup Request that USB SIE cannot handle by itself and these 9.5.5.8 SETUP0 and SETUP1 resister hold Device Request Data (8 bytes : 64 bit described above). This function is to handle standard requests that SIE cannot handle and to handle vendor specific requests.

* Note: 9.2.5.4 INTSTAT [0] bit will not go 'high' in case of Setup request if SIE can handle that request by itself.

9.3.3.9 ENDP0RD

0x8005.102C

Bits	Type	Function
31:0	R/W	Each Endpoint 0 FIFO Read

9.3.3.10 ENDP0WT

0x8005.1030

Bits	Type	Function
31:0	R/W	Each Endpoint 0 FIFO Write

9.3.3.11 ENDP1RD

0x8005.1034

Bits	Type	Function
31:0	R/W	Each Endpoint 1 FIFO Read

9.3.3.12 ENDP2WT

0x8005.1038

Bits	Type	Function
31:0	R/W	Each Endpoint 2 FIFO Write

9.3.4 Theory of Operation

The MagnaChip USB Core enables a designer to connect virtually any device requiring incoming or outgoing PC data to the Universal Serial Bus. As illustrated in Figure 9-1: USB Block Diagram, the USB core comprises two parts, the SIE and DEV. The SIE connects to the Universal Serial Bus via a bus transceiver. The interface between the SIE and the DEV is a byte-oriented interface that exchanges various types of data packets between two blocks.

Serial Interface Engine

The SIE converts the bit-serial, NRZI encoded and bit-stuffed data stream of the USB into a byte and packet oriented data stream required by the DEV. As shown in Figure 9-2: USB Serial Interface Engine, it comprises seven blocks: Digital Phase Lock Loop, Input NRZI decode and bit-unstuff, Packet Decoder, Packet Encoder, Output bit stuff and NRZI encode, Counters, and the CRC Generation & Checking block. Each of the blocks is described in the following sections.

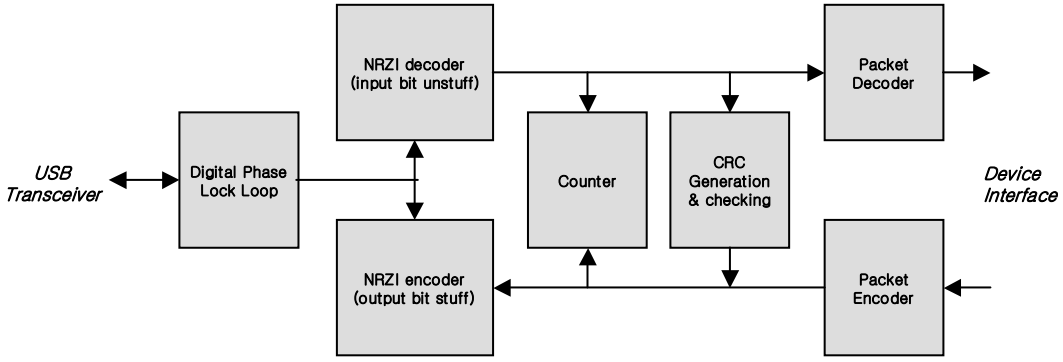


Figure 9-10. USB Serial Interface Engine

Digital Phase Lock Loop

The Digital Phase Lock Loop module takes the incoming data signals from the USB, synchronizes them to the 48MHz input clock, and then looks for USB data transitions. Based on these transitions, the module creates a divide-by-4 clock called the usbclock. Data is then output from this module synchronous to the usbclock.

Input NRZI decode and bit-unstuff

The Input NRZI decodes and bit-unstuff module extracts the NRZI encoded data from the incoming USB data. Transitions on the input serial stream indicate a 0, while no transition indicates a 1. Six ones in a row cause the transmitter to insert a 0 to force a transition, therefore any detected zero bit that occurs after six ones is thrown out.

Packet Decoder

The Packet Decoder module receives incoming data bits and decodes them to detect packet information. It checks that the PID (Packet ID) is valid and was sent without error.

After decoding the PID, the remainder of the packet is split into the address, endpoint, and CRC5 fields, if present. The CRC Checker is notified to verify the data using the incoming CRC5 field. If the packet is a data packet, the data is collected into bytes and passed on with an associated valid bit. Table 9-1: Supported PID Types shows the PID Types that are decoded (marked as either Receive or Both). At the end of the packet, either the packetok or packetnotok signal is asserted. Packetnotok is asserted if any error condition arose (bad valid bit, bit-stuff, bad PID, wrong length of a field, CRC error, etc.).

PID Type	Value	Send/Receive	PID Type	Value	Send/Receive
OUT	4'b0001	Receive	DATA1	4'b1011	Both
IN	4'b1001	Receive	ACK	4'b0010	Both
SOF	4'b1101	Receive	NAK	4'b1010	Send
SETUP	4'b0000	Receive	STALL	4'b1110	Send
DATA0	4'b0011	Both	PRE	4'b1100	Receive

Table 9-5. USB Supported PID Types

Packet Encoder

The Packet Encoder creates outgoing packets based on signals from the DEV. Table 9-1: Supported PID Types shows the PID Types that can be encoded (marked as Send or Both). For each packet type, if the associated signal sends type is received from the DEV, the packet is created and sent. Upon completion of the packet, packettypesent is asserted to inform the DEV of the successful transmission. The Packet Encoder creates the outgoing PID, grabs the data from the DEV a byte at a time, signals the CRC Generator to create the CRC16 across the data field, and then sends the CRC16 data. The serial bits are sent to the Output bit stuff and NRZI encoder.

Output bit stuff and NRZI encoder

The Output bit stuff and NRZI encoder takes the outgoing serial stream from the Packet Encoder, inserts stuff bits (a zero is inserted after six consecutive ones), and then encodes the data using the NRZI encoding scheme (zeroes cause a transition, ones leave the output unchanged).

Counter block

The Counter block tracks the incoming data stream in order to detect the following conditions: reset, suspend, and turnaround. It also signals to the transmit logic (Output NRZI and bit stuff) when the bus is idle so transmission can begin.

Generation and Checking block

The Generation and Checking block checks incoming CRC5 and CRC16 data fields, and generates CRC16 across outgoing data fields. It uses the CRC polynomial and remainder specified in the USB Specification Version 1.1.

Device Interface

The DEV shown in Figure 9-3: Device Interface works at the packet and byte level to connect a number of endpoints to the SIE. It understands the USB protocol for incoming and outgoing packets, so it knows when to grab data and how to correctly respond to incoming packets. A large portion of the DEV is devoted to the setup, configuration, and control features of the USB. As shown in Figure 9-3: Device Interface the DEV is divided into three blocks: Device Controller, Device ROM, and Start of Frame. The three blocks are described in the following sections.

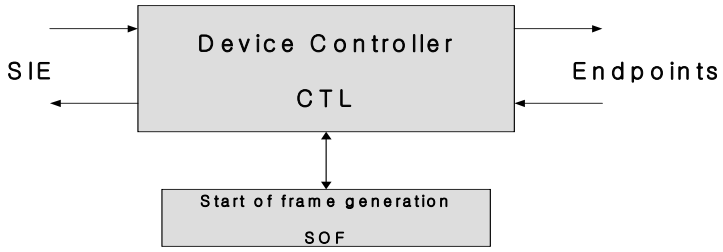


Figure 9-1 USB Device Interface Device Controller

Device Controller

The Device Controller contains a state machine that understands the USB protocol. The (SIE) provides the Device Controller with the type of packet, address value, endpoint value, and data stream for each incoming packet. The Device Controller then checks to see if the packet is targeted to the device by comparing the address/endpoint values with internal registers that were loaded with address and endpoint values during the USB enumeration process. Assuming the address/endpoint is a match, the Device Controller then interprets the packet. Data is passed on to the endpoint for all packets except SETUP packets, which are handled specially. Data toggle bits (DATA0 and DATA1 as defined by the USB spec) are maintained by the Device Controller. For IN data packets (device to host) the Device Controller sends either the maximum number of bytes in a packet or the number of bytes available from the endpoint. All packets are acknowledged as per the spec. For SETUP packets, the incoming data is extracted into the relevant internal fields, and then the appropriate action is carried out. Table 9-2: Supported Setup Requests lists the types of setup operations that are supported.

Setup Request	Value	Supported	Setup Request	Value	Supported
Get Status	0	Device, Interface, Endpoint	Get Configuration	8	Device
Clear Feature	1	Not supported	Set Configuration	9	Device
Set Feature	3	Not supported	Get Interface	10	Device
Set Address	5	Device	Set Interface	11	Device
Get Descriptor	6	Device	Synch Frame	12	Not supported
Set Descriptor	7	Not supported			

Table 9-6 USB Supported Setup Requests

Start of Frame

The Start of Frame logic generates a pulse whenever either the incoming Start of Frame (SOF) packet arrives or approximately 1 ms after it the last one arrived. This allows an isochronous endpoint to stay in sync even if the SOF packet has been

garbled.

9.3.5 Endpoint FIFOs (Rx, Tx)

Each endpoint FIFO has the specific number of FIFO depth according to data transfer rate. In case of maximum packet size for bulk transfer is 32 bytes that is supported in USBD. Each FIFO generates data ready signals (means FIFO not full or FIFO not empty) to AMBA IF. It contains the control logic for transferring 4 bytes at a read/write strobe generated by AMBA to obtain better efficiency of AMBA bus.

9.4 ADC Interface Controller

HMS30C7210 has internal ADC and ADC interface logic for analog applications of touch panel interface and general purpose. If user doesn't need these applications or want to use for other functions, there's a direct ADC control register available.

All channels can be used for general purpose application. ADC operating clock is "ACLK" called as "PCLK" in AMBA Peripherals. ADC sampling clock is "OCLK". It is about 8KHz.

FEATURES

- 3-channel 10-bit ADC.
- 8-sample data per one sampling point of touch panel (channel 0,1)
- 4-sample data per one sampling point of general purpose channel(channel 2)
- Manual and Auto ADC power down mode
- ADC input range : ADCVSS ~ ADCVREF
- Conversion time : 4.33usec (@ 3.6923MHz))

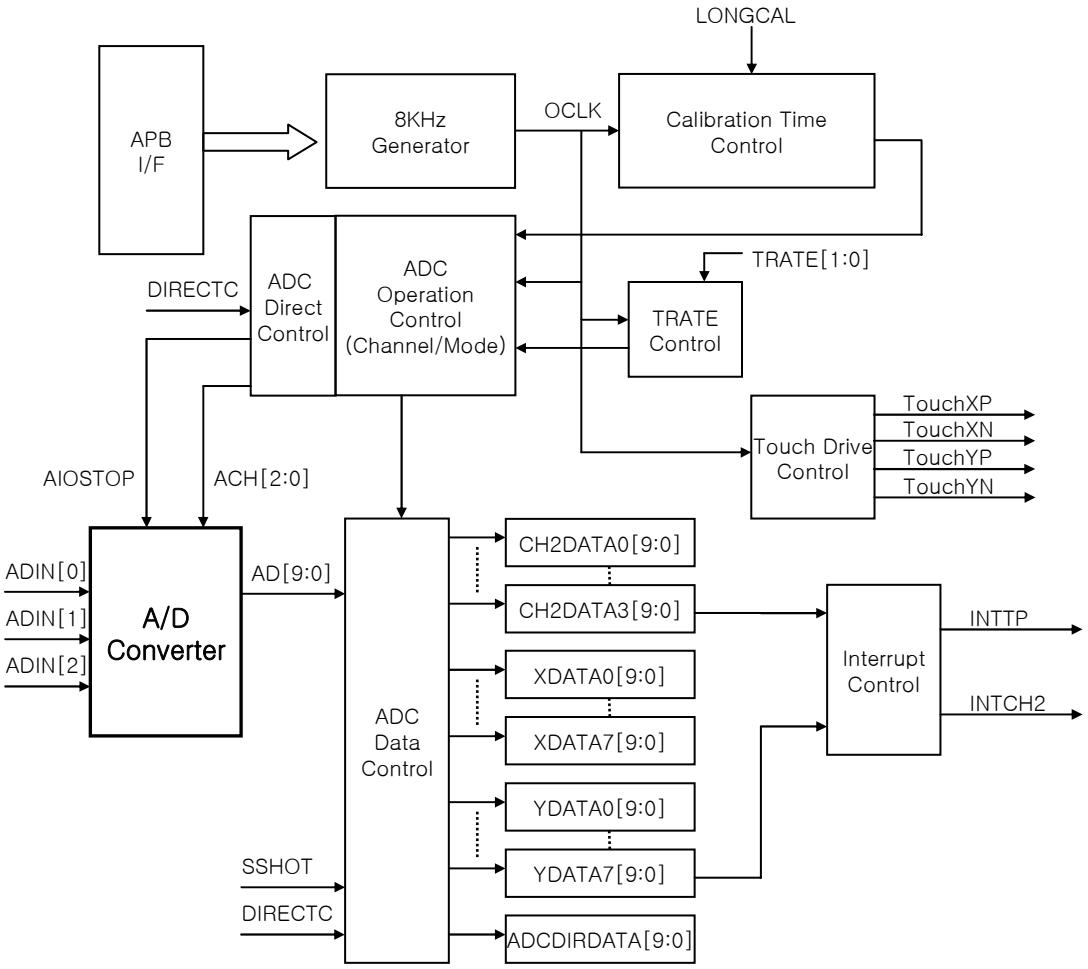


Figure 9-11. Block diagram of ADC, ADC I/F

9.4.1 External Signals

Pin Name*	Type**	Description
ADIN[0]	AI	ADC input. Touch Panel X-axis signal input or general purpose input
ADIN[1]	AI	ADC input. Touch Panel Y-axis signal input or general purpose input
ADIN[2]	AI	ADC input. CH2 value input.
ADCVDD	P	ADC analog VDD
ADCVSS	P	ADC analog VSS
ADCVREF	AI	ADC Reference voltage.
TouchXP	O	Touch screen switch X-positive drive
TouchXN	O	Touch screen switch X-negative drive
TouchYP	O	Touch screen switch Y-positive drive
TouchYN	O	Touch screen switch Y-negative drive

Refer to Figure 2-1. 208 Pin diagram.

9.4.2 Registers

Address	Name	Width	Default	Description
0x8005.3000	ADCCR	8	0x80	ADC Control Register
0x8005.3004	ADCTPCR	8	0x0	Touch panel Control register
0x8005.3008	ADCBACR	8	0x0	CH2 Control Register
0x8005.3010	ADCISR	8	0x0	ADC Interrupt Status Register
0x8005.3020	ADCDIRCR	8	0x0	ADC Direct Control Register
0x8005.3024	ADCDIRDATA	10	0x0	ADC Direct Data read register
0x8005.3030	ADCTPXDR0	32	0x0	Touch Panel X Data register 0
0x8005.3034	ADCTPXDR1	32	0x0	Touch Panel X Data register 1
0x8005.3038	ADCTPYDR0	32	0x0	Touch Panel Y Data register 0
0x8005.303C	ADCTPYDR1	32	0x0	Touch Panel Y Data register 1
0x8005.3040	ADCTPXDR2	32	0x0	Touch Panel X Data register 2
0x8005.3044	ADCTPXDR3	32	0x0	Touch Panel X Data register 3
0x8005.3048	ADCTPYDR2	32	0x0	Touch Panel Y Data register 2
0x8005.304C	ADCTPYDR3	32	0x0	Touch Panel Y Data register 3
0x8005.3050	ADCMBDATA0	32	0x0	CH2 Data Register0
0x8005.3054	ADCMBDATA1	32	0x0	CH2 Data Register1

Table 9-7. ADC Controller Register Summary

9.4.2.1 ADC Control Register (ADCCR)

0x8005.3000

7	6	5	4	3	2	1	0
ADCPD	DIRECTC			WAIT[3:2]		SOP	LONGCAL

Bits	Type	Function
7	R/W	ADC power down bit User can set ADCPD to save power consumption by ADC. This bit blocks the clock to ADC and ADC I/F, so they consumes no power when this bit is set to "1". But after writing this bit to "0", ADC need about 10ms calibration time to normal operation. 0: normal mode 1: power down mode
6	R/W	ADC Direct access control DIRECTC bit can be used for direct access from CPU to ADC without interface function logic. All direct control signals are describe in ADCDIRCR register field. If this bit is set to "1", CPU directly access ADC through ADCDIRCR and directly read ADC result value through ADCDIRDATA register. In this mode, ADCCR register except ADCPD don't affect to ADC and ADC I/F 0: No direct access mode 1: Direct access mode
5:4	-	Reserved
3:2	R/W	ADC conversion wait time Basically ADC core converts Analog data to Digital data continuously in every 16 ADC operation-clocks called as "ACLK". WAIT bit field select data loading time of ADC I/F logic because in certain case ADC I/F logic can read wrong or unstable value from ADC. "No Wait" informs that ADC data loading clock period is equal to ADC conversion clock period. "2, 4 clock wait" informs that ADC data loading clock period is longer than ADC conversion clock period.
1	R/W	Self Operating Power down bit It means that power down mode of ADC –not ADC I/F- is controlled by TPEN and CH2EN in addition to ADCPD. SOP bit can be used for one-shot operation to save power. When this bit is set to "1" and all ADC functions aren't enabled, so ADC goes to power down mode. 0: No SOP mode 1: SOP mode
0	R/W	Long calibration time. LONGCAL selects self-calibration time. Initially this bit is set to "0" .It means short calibration time (about 12 ms). But if first a couple of data were wrong value, user should select long calibration time (about 48 ms) by writing this bit set to "1". The default ADC calibration time is 12 ms. But when it is needed, ADC can be calibrated during 48ms with this bit. - Calibration time T_{SCAL} = 96 / F_{OCLK} = 12msec or T_{LCAL} = 384 / F_{OCLK} = 48msec 0: Short calibration time (96 cycles of OCLK*) 1: Long calibration time (384 cycles of OCLK)

WAIT[1:0]	wait time	A period of loading data clock
00	No wait	Equal to a period of ADC conversion clock
01	2 clock wait	More 2cycles of ACLK
10	4 clock wait	More 4cycles of ACLK
11	Reserved	Reserved

*ADC conversion clock is 16 cycles of ACLK

9.4.2.2 ADC Touch Panel Control Register (ADCTPCR)

0x8005.3004

7	6	5	4	3	2	1	0
TPEN	TINTMSK		SWINVT		SSHOT	TRATE[1:0]	

Bits	Type	Function
7	R/W	Touch panel read enable bit. If this bit is set to “1”, Touch panel function is enabled. 0: Touch panel read disable 1: Touch panel read enable
6	R/W	Touch panel read interrupt mask bit. Writing this bit to “1” enables generating of interrupt signal from Touch panel data receiver. Refer to ADCISR[2] 0: Interrupt disable 1: Interrupt enable
5	-	Reserved
4	R/W	Touch panel drive signal inversion bit for flexibility. TouchXM and TouchYM output initial value is “0”. While Touch panel X mode in progress, TouchXM value is “1”. Also while Touch panel Y mode in progress, TouchYM value is “1”. Always TouchXP and TouchYP output value opposite to TouchXM and TouchYM respectively. Writing this bit to “1” inverts the above. For example, TouchXM output initial value change to “1”. Also while Touch panel X mode in progress, TouchXM value is “0”. 0: No inversion 1: Inversion
3	-	Reserved
2	R/W	Single touch panel read operation. Normally, touch panel data read twice per 4-sample. But this bit is set to “1”, touch panel data read just once per 4-sample and saving power to read touch panel. 0: data read twice. Touch panel data is loaded into 1st and 2nd Touch Panel data registers. 1: data read once. Touch panel data is loaded into just 1st Touch Panel data registers.
1:0	R/W	Touch panel data sampling rate. It depends on OCLK of ADC interface.

TRATE[1:0]	samples / sec	description
00	50 samples / sec	One sample per 160 cycles of OCLK
01	100 samples / sec	One sample per 80 cycles of OCLK
10	200 samples / sec	One sample per 40 cycles of OCLK
11	400 samples / sec	One sample per 20 cycles of OCLK

9.4.2.3 ADC CH2 Control Register (ADCCH2CR)

This register controls CH2 channel check operation.

0x8005.3008

7	6	5	4	3	2	1	0
						CH2INTMSK	CH2EN

Bits	Type	Function
7:2	-	Reserved Should be set to “0”
1	R/W	CH2 channel interrupt mask bit Writing this bit to “1” enables generating interrupt signal from CH2 channel data register. Refer to ADCISR[1]. 0: Interrupt disable 1: Interrupt enable
0	R/W	CH2 enable If this bit is set to “1”, CH2 function is enabled. 0: CH2 channel disable 1: CH2 channel enable

9.4.2.4 ADC Interrupt Status Register (ADCISR)

0x8005.3010

7	6	5	4	3	2	1	0
					TP_INT	CH2_INT	

Bits	Type	Function
7:3	-	Reserved
2	R	Touch panel data interrupt flag. Interrupt signal is generated at the end of CH2_MODE after 4-sampling. Read only valid and writing this bit to “1” clear this flag. 0: Interrupt was not generated or was cleared. 1: Interrupt was generated.
1	R	CH2 channel interrupt flag. Interrupt signal is generated at the end of TPY_MODE after 4-samling or 8-samlping. If SSHOT is set to “0”, TP_INT is generated at the end of 2nd TPY_MODE after 8-sampling of TPX and TPY respectively. But if SSHOT is set to “1”, TP_INT is generated at the end of 1st TPY_MODE after 4-sampling of TPX and TPY respectively. Read only valid and writing this bit to “1” clear this flag. 0: Interrupt was not generated or was cleared 1: Interrupt was generated.
0	-	Reserved

9.4.2.5 ADC Direct Control Register (ADCDIRCR)

0x8005.3020

7	6	5	4	3	2	1	0
DIR_AIOSTOP			DIR_ACH[2:0]				

Bits	Type	Function
7	R/W	Direct AIOSTOP When DIRECTC(ADCCR[6]) bit is set to "1", ADC power down mode is controlled by DIR_AIOSTOP, not ADCPD(ADCCR[7]). But if DIRECTC bit is "0", DIR_AIOSTOP doesn't affected to ADC power down mode. 0: normal mode in the direct access mode 1: power down mode in the direct access mode
6:3	-	Reserved Should be set to "0"
2:0	R/W	Direct ADC channel When DIRECTC(ADCCR[6]) bit is set to "1", ADC channel is controlled by DIR_ACH.

DIR_ACH[2:0]	channel	description
001	channel 0	touch panel X
010	channel 1	touch panel Y
100	channel 2	general purpose

9.4.2.6 ADC Direct Data Read Register (ADCDIRDATA)

Register can be used to read data from ADC.

0x8005.3024

9	8	7	6	5	4	3	2	1	0
DIR_AD[9:0]									

Bits	Type	Function
9:0	R	10-bit AD conversion data

9.4.2.7 ADC 1ST Touch Panel Data register

0x8005.3030 – 0x8005.303C

25	24	23	22	21	20	19	18	17	16
XDATA1: ADCTPXDR0[25:16], XDATA3: ADCTPXDR1[25:16] YDATA1: ADCTPYDR0[25:16], YDATA3: ADCTPYDR1[25:16]									
9	8	7	6	5	4	3	2	1	0
XDATA0: ADCTPXDR0[9:0], XDATA2: ADCTPXDR1[9:0] YDATA0: ADCTPYDR0[9:0], YDATA2: ADCTPYDR1[9:0]									

ADCTPXDR0: 0x8005.3030

Bits	Type	Function
31:26	-	Reserved
25:16	R	Touch panel X data 10-bit, 2/4 of the first sample cycle (XDATA1)
15:10	-	Reserved
9:0	R	Touch panel X data 10-bit, 1/4 of the first sample cycle (XDATA0)

ADCTPXDR1: 0x8005.3034

Bits	Type	Function
31:26	-	Reserved
25:16	R	Touch panel X data 10-bit, 4/4 of the first sample cycle (XDATA3)
15:10	-	Reserved
9:0	R	Touch panel X data 10-bit, 3/4 of the first sample cycle (XDATA2)

ADCTPYDR0: 0x8005.3038

Bits	Type	Function
31:26	-	Reserved
25:16	R	Touch panel Y data 10-bit, 2/4 of the first sample cycle (YDATA1)
15:10	-	Reserved
9:0	R	Touch panel Y data 10-bit, 1/4 of the first sample cycle (YDATA0)

ADCTPYDR1: 0x8005.303C

Bits	Type	Function
31:26	-	Reserved
25:16	R	Touch panel Y data 10-bit, 4/4 of the first sample cycle (YDATA3)
15:10	-	Reserved
9:0	R	Touch panel Y data 10-bit, 3/4 of the first sample cycle (YDATA2)

9.4.2.8 ADC 2ND Touch Panel Data Register

0x8005.3040 – 0x8005.304C

25	24	23	22	21	20	19	18	17	16
XDATA5: ADCTPXDR2[25:16], XDATA7: ADCTPXDR3[25:16] YDATA5: ADCTPYDR2[25:16], YDATA7: ADCTPYDR3[25:16]									
9	8	7	6	5	4	3	2	1	0
XDATA5: ADCTPXDR2[9:0], XDATA6: ADCTPXDR3[9:0] YDATA5: ADCTPYDR2[9:0], YDATA6: ADCTPYDR3[9:0]									

ADCTPXDR2: 0x8005.3040

Bits	Type	Function
31:26	-	Reserved
25:16	R	Touch panel X data 10-bit, 2/4 of the second sample cycle (XDATA5)
15:10	-	Reserved
9:0	R	Touch panel X data 10-bit, 1/4 of the second sample cycle (XDATA4)

ADCTPXDR3: 0x8005.3044

Bits	Type	Function
31:26	-	Reserved
25:16	R	Touch panel X data 10-bit, 4/4 of the second sample cycle (XDATA7)
15:10	-	Reserved
9:0	R	Touch panel X data 10-bit, 3/4 of the second sample cycle (XDATA6)

ADCTPYDR2: 0x8005.3038

Bits	Type	Function
31:26	-	Reserved
25:16	R	Touch panel Y data 10-bit, 2/4 of the second sample cycle (YDATA5)
15:10	-	Reserved
9:0	R	Touch panel Y data 10-bit, 1/4 of the second sample cycle (YDATA4)

ADCTPYDR3: 0x8005.303C

Bits	Type	Function
31:26	-	Reserved
25:16	R	Touch panel Y data 10-bit, 4/4 of the second sample cycle (YDATA7)
15:10	-	Reserved
9:0	R	Touch panel Y data 10-bit, 3/4 of the second sample cycle (YDATA6)

9.4.2.9 ADC CH2 Data Register (ADCCH2DATA)

0x8005.3050 – 0x8005.3054

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
						CH2DATA1: ADCCH2DATA0, CH2DATA3: ADCCH2DATA1									
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
						CH2DATA0: ADCCH2DATA0, CH2DATA2: ADCCH2DATA1									

ADCMBDATA0: 0x8005.3050

Bits	Type	Function
31:26	-	Reserved
25:16	R	CH2 channel data 10-bit, 2/4 of the CH2 sample cycle (CH2DATA1)
15:10	-	Reserved
9:0	R	CH2 channel data 10-bit, 1/4 of the CH2 sample cycle (CH2DATA0)

ADCMBDATA1: 0x8005.3054

Bits	Type	Function
31:26	-	Reserved
25:16	R	CH2 channel data 10-bit, 4/4 of the CH2 sample cycle (CH2DATA3)
15:10	-	Reserved
9:0	R	CH2 channel data 10-bit, 3/4 of the CH2 sample cycle (CH2DATA2)

9.4.3 Operation

9.4.3.1 Clock & power down mode

The clock source of ADC is the peripheral clock PCLK. This is called the ACLK and is controlled by the ADCPD bit in ADCCR register. Writing “0” to the ADCPD bit is that the PCLK is connected to ACLK. On the contrary, writing “1” to this bit means that ADC mode is power down mode. In this mode, the ACLK is always “0”. The data sampling clock of ADC Interface controller is the OCLK.
This clock has a frequency of F_{PCLK} / 461.

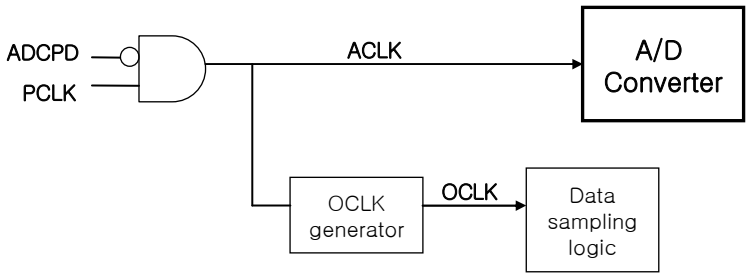


Figure 9-12. ADC Clock & Data sampling clock

9.4.3.2 Operating stop condition & power down mode

The ADC can go to power down mode by blocking ACLK and controlling AIOSTOP. The AIOSTOP is an enable signal of the ADC. When this signal is low, the ADC starts normal operation. By writing to “0” to the ADCPD bit, AIOSTOP is set to “0”. But if the SOP bit in ADCCR register, the TPEN in ADCTPCR register or CH2EN in ADCCH2CR register should be set to “1” in addition to the ADCPD low.

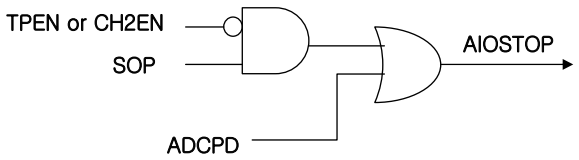


Figure 9-13. ADC operating stop condition

9.4.3.3 *Calibration time*

The ADC needs calibration time for ADC conversion start. The calibration time for the ADC is about 10msec. When the LONGCAL in ADCCR register is low, the calibration time is about 12msec. If the first a couple of data were wrong value, the ADC is not stable yet. In this case, user should set “1” to the LONGCAL bit. Long calibration time is about 48msec.

9.4.3.4 *Data sampling & loading time*

The data sampling frequency of the ADCIF is OCLK. ADC data is loaded into ADCTPXDR, ADCTPYDR, ADCCH2DR registers four times per one period of OCLK. When OCLK is high, data loading is started after 90 cycles of ACLK. The conversion clock of the ADC is $F_{ACLK}/16$. User can select data loading cycle. The WAIT bits in ADCCR register determine a period of loading data. When the WAIT bits are ‘0’, a period of loading data is equal to a period of ADC conversion clock. When the WAIT bits are ‘1’ or ‘2’, a period of it is more 2 or 4 cycles of ACLK.

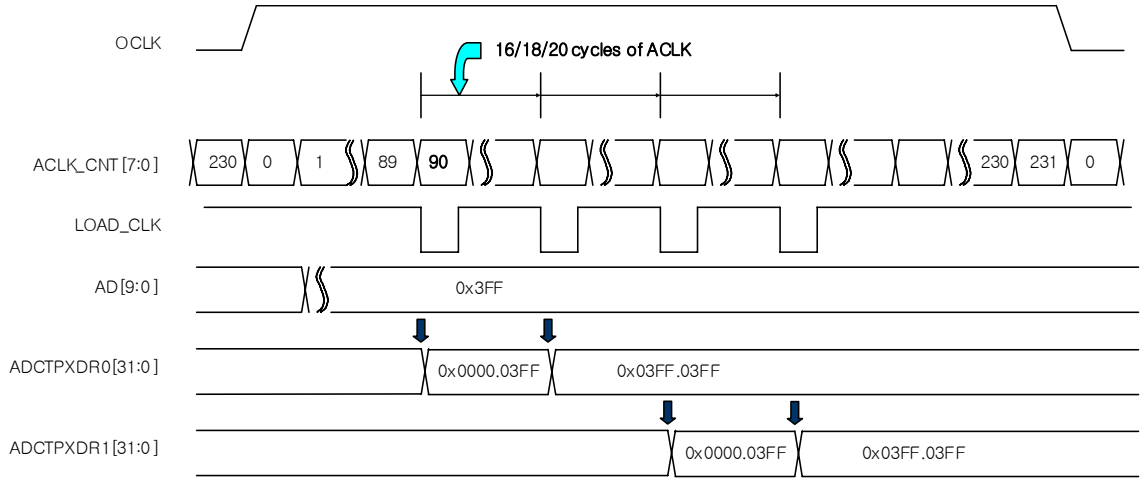


Figure 9-14. Data loading timing

9.4.3.5 Data sampling sequence

One sampling cycle is consisted of OCLK 20 cycles.

Mode, Channel operation

Normally mode & channel of CH2 are generated once per sampling cycle. Mode & channel of touch panel are generated twice per sampling cycle. But in this case, touch panel is dependent on SSHOT or TRATE.

SSHOT operation

Normally touch panel data register is loaded twice. So touch panel data is loaded into 1st and 2nd Touch Panel data registers. If the SSHOT bit in ADCTPCR register is set high, touch panel data register is loaded just once for a point and saving power to read touch panel. So touch panel data is loaded into just 1st Touch Panel data register.

TRATE [1:0] operation

These bits are in ADCTPCR register. If the TRATE bits are 2'b11, Touch Panel data registers are updated every sampling cycle. If the TRATE bits are 2'b10, Touch Panel data registers are updated once per 2 sampling cycles. If the TRATE bits are 2'b01, Touch Panel data registers are updated once per 4 sampling cycles. If the TRATE bits are 2'b10, Touch Panel data registers are updated once per 8 sampling cycles.

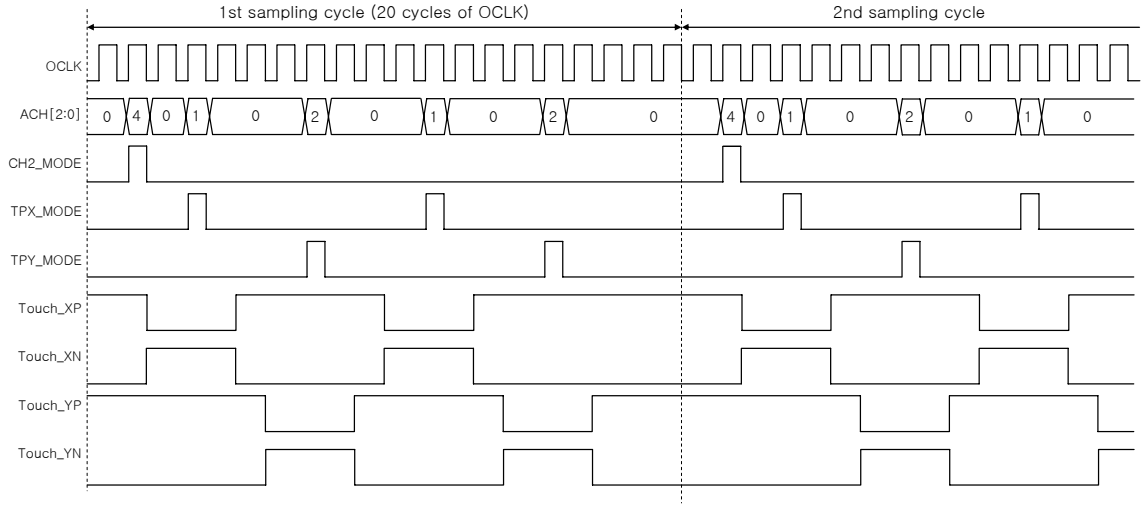


Figure 9-15. Data sampling sequence – TRATE is 2'b11 / SSHOT is 1'b0 / SWINVT is 1'b0

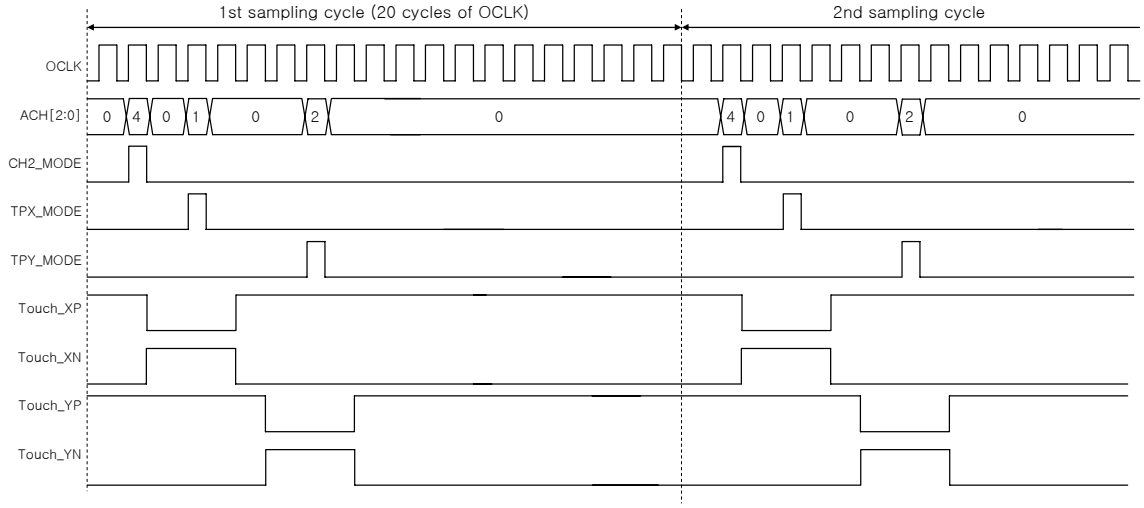


Figure 9-16. Data sampling sequence – TRATE is 2'b11 / SSHOT is 1'b1 / SWINVT is 1'b0

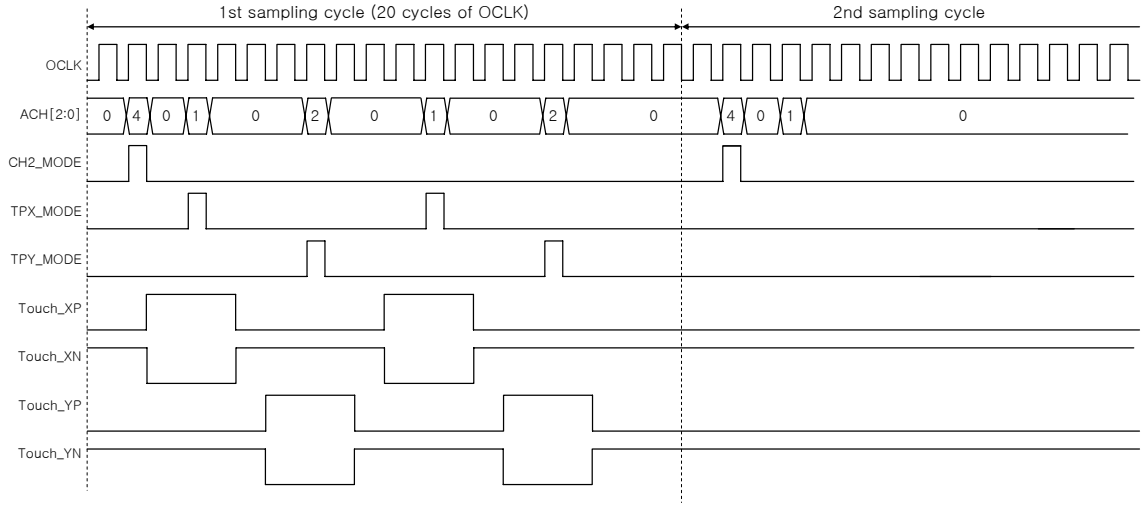


Figure 9-17. Data sampling sequence – TRATE is 2'b10 / SSHOT is 1'b0 / SWINVT is 1'b1

9.4.3.6 Interrupt control

Interrupt signal is generated at the end of CH2_MODE, TPY_MODE.

For generating interrupt signal, the TINTMSK bit in ADCTPCR register and the CH2INTMSK bit in ADCCH2CR register are set high. If the SSHOT bit in ADCTPCR register is low, TP_INT is generated at the end of 2nd TPY_MODE. As soon as ADC 2nd Touch Panel Data Registers is updated, TP_INT is generated. But if the SSHOT bit in ADCTPCR register is high, TP_INT is generated at the end of 1st TPY_MODE. As soon as ADC 1st Touch Panel Data Registers is updated, TP_INT is generated.
In case of CH2_MODE, CH2_INT is always generated at the end MB_MODE.

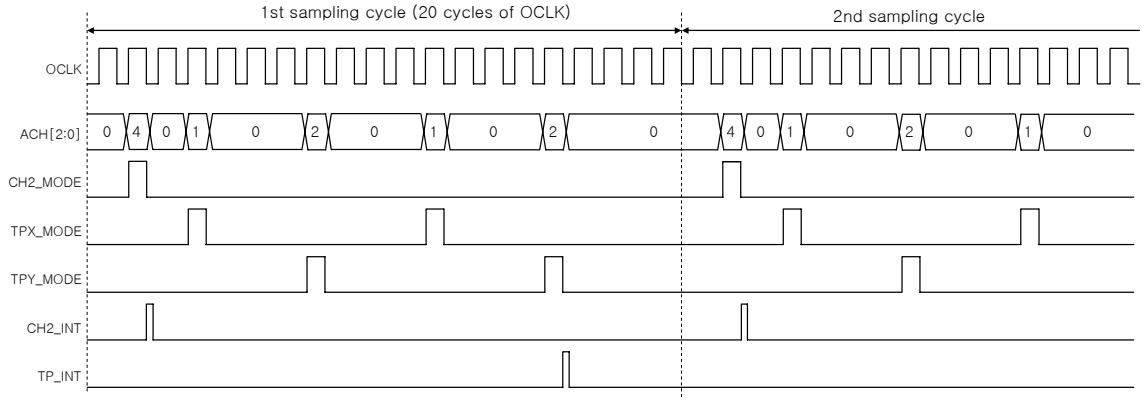


Figure 9-18. Interrupt generating timing – TRATE is 2'b11 / SSHOT is 1'b0

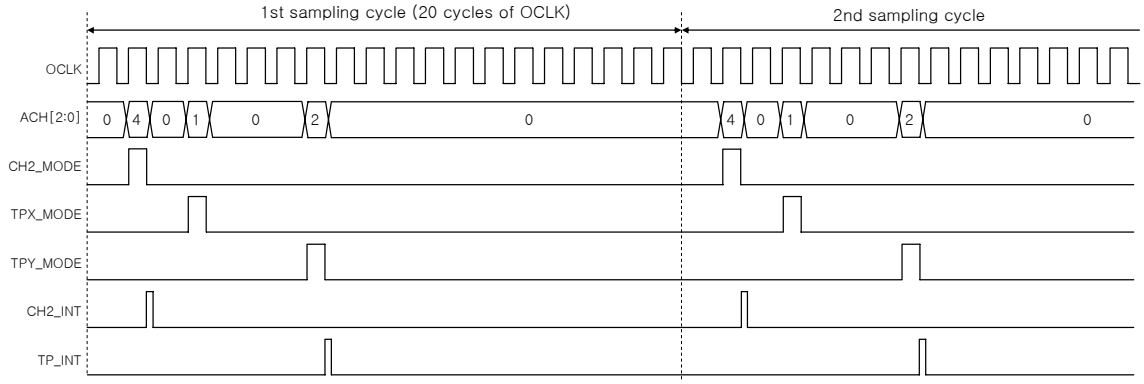


Figure 9-19. Interrupt generating timing – TRATE is 2'b11 / SSHOT is 1'b1

9.4.3.7 *Direct access mode*

The CPU can directly access the ADC. When the DIRECTC bit in ADCCR register is high, the direct control logic is enabled and the ADC is directly connected by using ADCDIRCR register. ADC conversion data is loaded into ADCDIRDATA register. The ADCPD bit in ADCCR register should be set low to start this mode.

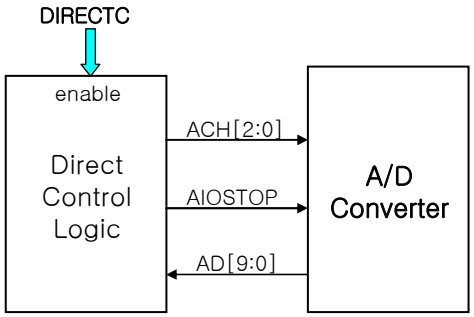


Figure 9-20. ADC direct access mode

9.4.3.8 *Operation setup flow*

Touch panel mode

- Select SWINTV, SSHOT, TRATE[1:0] in ADCTPCR register.
- Set TPEN, TINTMSK in ADCTPCR register.
- Select WAIT[3:2], SOP, LONGCAL in ADCCR register.
- Set ADCPD to low in ADCCR register for starting.
- Check TP_INT in ADCISR register.

CH2 mode

- Set CH2EN, CH2INTMSK in ADCCH2CR register.
- Select WAIT[3:2], SOP, LONGCAL in ADCCR register.
- Set ADCPD to low in ADCCR register for starting.
- Check CH2_INT in ADCISR register.

Direct access mode

- Set DIRECTC in ADCCR register.
- Select DIR_ACH[2:0] in ADCDIRCR register.
- Set DIR_AIOSTOP to low in ADCDIRCR register.
- Set ADCPD to low in ADCCR register for starting.
- Check DIR_AD[9:0] in ADCDIRDATA register

9.4.3.9 *About Touch Panel board setup*

ADCTPCR register control functions related with touch panel interface. HMS30C7210 supports only external drive for touch panel (TouchXP/TouchXN/TouchYP/TouchYN), so prudent setting of this register is needed. For more information about touch panel setup, refer to “HMS30C7210 H/W Reference Development Kit Reference board ver0.1” in www.magnachip.com web site.

9.4.4 A/D Converter

H35AD33S is a CMOS(0.35㎛, 1-poly, 3-metal) 10-bit successive approximation A/D Converter which has high speed, low power consumption. The ADC has multiplexed 8 input channels. The serial output is configured to interface with standard shift registers. The differential analog voltage input allows for common-mode rejection or offset of the analog zero input voltage value. The voltage reference input can be adjusted to allow encoding any smaller analog voltage span to the full 10 bits of resolution

FEATURES

- Power supply: 3.3v
- Resolution: 10 bits
- Signal-to-noise ratio (SNR): 54dB
- 3 channels
- Conversion speed: 230KHz (@ 3.6923Mhz)
- Main clock: 3.6923 MHz
- Power-down mode
- Analog input range: AVSS~ avref
- Cell Size: 1000㎛ x 1000 ㎛

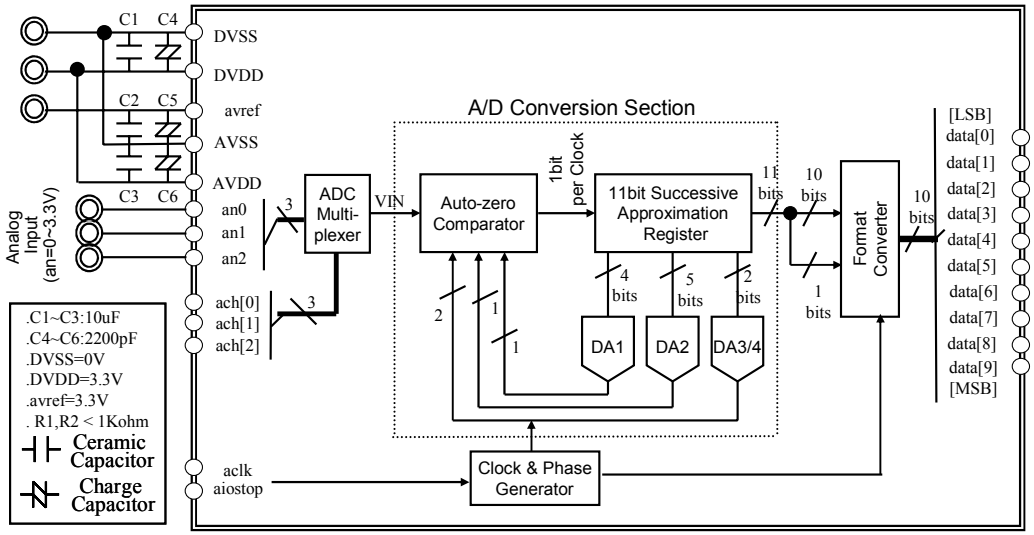


Figure 9-21. Block diagram of A/D Converter

9.4.4.1 Functional description

This SAR-type ADC contains a SAR register, an auto-zero comparator, three internal DAC, MUX(3x1), a format converter, a clock & phase generator, and a reference ladder & calibrator. The conversion rate ranges up to 1MHz. These blocks contained in ADC can be described as follows:

SAR register

This block is a successive approximation register which latches the output of comparator and generates the input of the internal DAC.

Auto-zero comparator

This comparator is able to reduce the offset error periodically and senses the difference between analog input and DAC output.

Internal DAC

These DAC generate analog reference voltage according to SAR register output.

Multiplexer

One of the eight channel can be selected by the control pins (ach[0] ~ ach[2])

Format converter

This format converter is to latch the 11-bit SAR output data stream and convert it to a standard 10-bit binary format.

Clock & Phase generator

The outputs generated in clock & phase generator control SAR-type ADC conversion operation.

Reference ladder & calibration

This reference ladder generates the analog reference voltage used by the internal DAC. The reference ladder taps are adjusted by using an auto-calibration technique.

9.4.4.2 *Timing diagram*

ADC starts data conversion after calibration time.

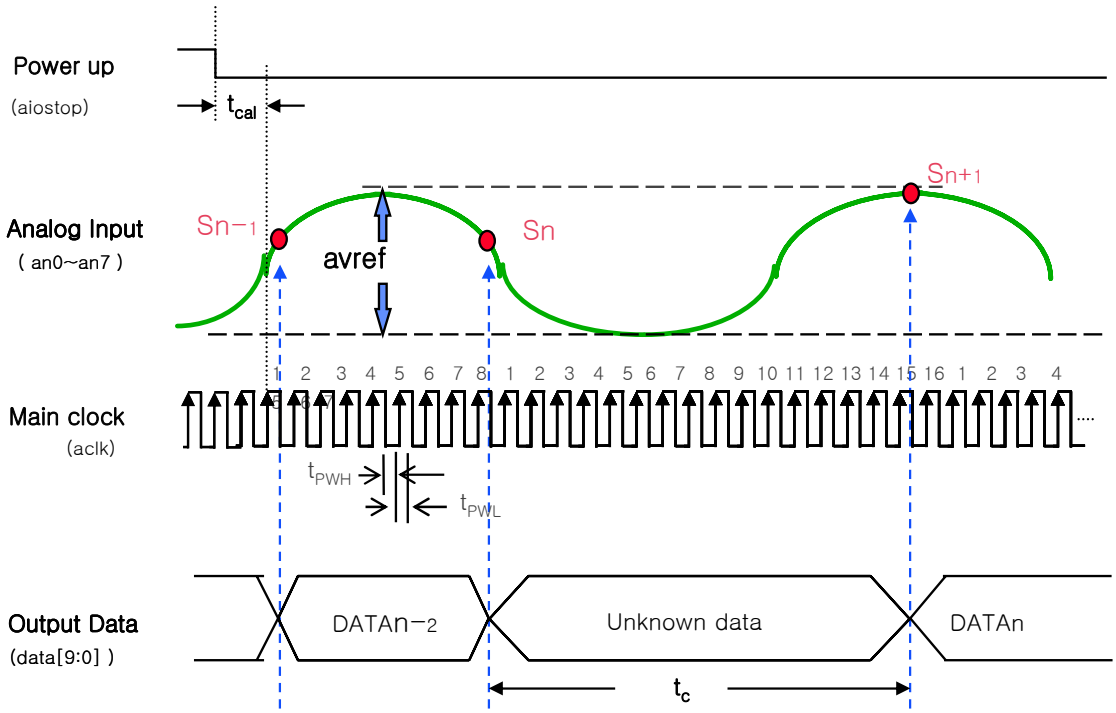


Figure 9-22. Timing diagram of A/D Converter

9.4.4.3 *Electrical characteristics*

Refer to ‘chapter 11.3 A/D Converter Electrical Characteristics’

9.5 UART/SIR

UART (Universal Asynchronous Receiver/Transmitter) of HMS30C7210 is functionally identical to the 16C550. On power-up, UART is set to CHARACTER mode(Non-FIFO Mode) and has a single Tx/Rx buffer. This UART can be put into an alternate mode (FIFO mode) to relieve the CPU of excessive software overhead. In the FIFO mode internal FIFOs are activated - RECEIVE FIFO (16 bytes plus 3 bit of error data per byte) stores the received data and the error information of individual received data and TRANSMIT FIFO(16 Bytes) stores the data to be trainsmitted. All the logic is on the chip to minimize the system overhead and to maximize efficiency.

The UART performs serial-to-parallel conversion on data characters received from a peripheral device or a MODEM, and parallel-to-serial conversion on data characters received from the CPU. The CPU can read the complete status of the UART at any time during the functional operation. Status information reported includes the type and condition of the transfer operations being performed by the UART, as well as any error conditions (parity, overrun, framing, or break interrupt).

The UART includes a programmable baud rate generator capable of dividing the timing reference clock input by divisors of 1 to 2^{16}-1, and producing a 16x clock for driving the internal transmitter logic. Provisions are also included to use this 16x clock to drive the receiver logic.

The UART has complete MODEM-control capability, and a processor-interrupt system. Interrupts can be programmed to the user's requirements, minimizing the computing required to handle the communications link.

FEATURES

- Capable of running all existing 16C550 software (Except UART0, UART1).
- After reset, all registers are identical to the 16C550 register set. (Except UART0, UART1).
- The FIFO mode transmitter and receiver are each buffered with 16 byte FIFOs to reduce the number of interrupts presented to the CPU.
- Add or delete standard asynchronous communication bits (start, stop and parity) to or from the serial data.
- Holding and shift registers in the 16C450 mode eliminate the need for precise synchronization between the CPU and serial data.
- Independently controlled transmit, receive, line status and data set interrupts.
- Programmable baud generator divides any input clock by 1 to 65535 and generates 16x clock
- Independent receiver clock input.
- MODEM control functions (CTS, RTS, DSR, DTR, RI and DCD) (UART5 Only).
- Fully programmable serial-interface characteristics:
- 5-, 6-, 7- or 8-bit characters
- Even, odd or no-parity bit generation and detection
- 1-, 1.5- or 2-stop bit generation and detection
- Baud generation (DC to 230k baud)
- False start bit detection.
- Complete status-reporting capabilities.
- Line breaks generation and detection.
- Internal diagnostic capabilities:
- Loopback controls for communications link fault isolation
- Full prioritized interrupt system controls.

9.5.1 External Signals

These uart pin names are same as HMS30C7210 Top pin names.
To get the information about pin number of UART signal at Chip, refer to “Table 2-3 Detail Pin Description”.

Pin Name	Type	Description
SCRST [0]	I	UART 0 serial data inputs. Serial data input from the communications link (peripheral device, MODEM or data set).
SCIO [0]	O	UART 0 serial data outputs. Composite serial data output to the communications link (peripheral, MODEM or data set). The USOUT signal is set to the Marking (logic 1) state upon a Master Reset operation.
SCRST [1]	I	UART 1 serial data inputs
SCIO [1]	O	UART 1 serial data outputs
UART2Rx	I	UART 2 serial data inputs
UART2Tx	O	UART 2 serial data outputs
UART3Rx	I	UART 3 serial data inputs
UART3Tx	O	UART 3 serial data outputs
IrDA4Rx	I	UART 4 serial data inputs
IrDA4Tx	O	UART 4 serial data outputs
UART5Rx	I	UART 5 serial data inputs
UART5Tx	O	UART 5 serial data outputs
nURING	I	UART 5 ring input signal (wake-up signal to PMU). When LOW, this indicates that the MODEM or data set has received a telephone ring signal. The nURING signal is a MODEM status input whose condition can be tested by the CPU reading bit 6 (RI) of the MODEM Status Register. Bit 6 is the complement of the nURING signal. Bit 2 (TERI) of the MODEM Status Register indicates whether the nURING input signal has changed from a LOW to a HIGH state since the previous reading of the MODEM Status Register.
nUDTR	O	UART 5 data terminal ready. When LOW, this informs the MODEM or data set that the UART is ready to establish communication link. The nUDTR output signal can be set to an active LOW by programming bit 0 (DTR) of the MODEM Control Register to HIGH level.
nUCTS	I	UART 5 clear to send input. When LOW, this indicates that the MODEM or data set is ready to exchange data. The nUCTS signal is a MODEM status input whose conditions can be tested by the CPU reading bit 4 (CTS) of the MODEM Status Register. Bit 4 is the complement of the nURING signal. Bit0 (DCTS) indicates whether the nUCTS input has changed state since the previous reading of the MODEM Status Register. nUCTS has no effect on the Transmitter.
nURTS	O	UART 5 request to send. When LOW, this informs the MODEM or data set that the UART is ready to exchange data. The nURTS output signal can be set to an active LOW by programming bit 1 (RTS) of the MODEM Control Register.
nUDSR	I	UART 5 data set ready input. When LOW, this indicates that the MODEM or data set is ready to establish the communications link with the UART. The nUDSR signal is a MODEM status input whose conditions can be tested by the CPU reading bit 5 (DSR) of the MODEM Status Register. Bit 5 is the complement of the nUDSR signal. Bit 1(DDSR) of MODEM Status Register indicates whether the nUDSR input has changed state since the previous reading of the MODEM status register.
nUDCD	I	UART 5 data carrier detect input. When LOW, indicates that the data carrier has been detected by the MODEM data set. The signal is a MODEM status input whose condition can be tested by the CPU reading bit 7 (DCD) of the MODEM Status Register. Bit 7 is the complement of the signal. Bit 3 (DDCD) of the MODEM Status Register indicates whether the input has changed state since the previous reading of the MODEM Status Register. nUDCD has no effect on the receiver.

Refer to Figure 2-1. 208 Pin diagram.

9.5.2 Registers

Address	Name	Width	Default	Description
0x8005.4000	U0Base	-	-	UART 0 Base
0x8005.5000	U1Base	-	-	UART 1 Base
0x8005.6000	U2Base	-	-	UART 2 Base
0x8005.7000	U3Base	-	-	UART 3 Base
0x8005.8000	U4Base	-	-	UART 4 Base
0x8005.9000	U5Base	-	-	UART 5 Base
UxBase+0x00	RBR	8	0x00	Receiver Buffer Register (DLAB = 0, Read Only)
	THR	8	0x00	Transmitter Holding Register (DLAB = 0, Write Only)
	DLL	8	0x00	Divisor Latch Least Significant Byte (DLAB = 1, Read/Write)
UxBase+0x04	IER	8	0x00	Interrupt Enable Register (DLAB = 0, Read/Write)
	DLM	8	0x00	Divisor Latch Most Significant Byte (DLAB = 1, Read/Write)
UxBase+0x08	IIR	8	0x01	Interrupt Identification Register (Read Only)
	FCR	8	0x00	FIFO Control Register (Write Only)
UxBase+0x0C	LCR	8	0x00	Line Control Register (Read/Write)
UxBase+0x10	MCR	3	0x00	Modem Control Register (Read/Write)
UxBase+0x14	LSR	8	0x60	Line Status Register (Read/Write)
UxBase+0x18	MSR	8	0x00	Modem Status Register (Read/Write)
UxBase+0x1C	SCR	8	0x00	Scratch Register (Read/Write)
UxBase+0x30	UCR	6	0x00	UART Configuration Register (Read/Write)

Table 9-8 UART/SIR Register Summary

9.5.2.1 RBR

RBR is the Receive Buffer Register and stores the data from serial input. This register is read-only and can be accessed when DLAB(Bit7 of Line Control Register) is set to 0.

UxBase+0x00

7	6	5	4	3	2	1	0
Receive Data Bit 7 ~ Receive Data Bit 0							

Bits	Type	Function
7:0	R	Receive Byte that is received from Serial input.

9.5.2.2 THR

THR is the Transmit Buffer Register and stores the data to be transmitted through serial output. This register is write-only and can be accessed when DLAB(Bit7 of Line Control Register) is set to 0.

UxBase+0x00

7	6	5	4	3	2	1	0
Transmit Data Bit 7 ~ Transmit Data Bit 0							

Bits	Type	Function
7:0	W	Transmit Byte that is transmitted through Serial output.

9.5.2.3 DLL

DLL is the Divisor Latch Least Significant Byte Register and used to set the lower 8-bit of 16-bit Baud-Rate divisor value.

UxBase+0x00

7	6	5	4	3	2	1	0
Baud-Rate divisor Bit 7 ~ Baud-Rate divisor Bit 0							

Bits	Type	Function
7:0	R/W	Lower 8-bit of 16-bit Baud-Rate divisor.

9.5.2.4 IER

IER is the Interrupt Enable Reigster and enables the five types of UART interrupts. Each interrupt can individually activate the interrupt (INTUART) output signal. It is possible to totally disable the interrupt Enable Register (IER). Similarly, setting bits of the IER register to logic 1 enables the selected interrupt(s). Disabling an interrupt prevents it from being indicated as active in the IIR and from activating the INTUART output signal. All other system functions operate in their normal manner, including the setting of the Line Status and MODEM Status Registers. Table 13-6: Summary of registers on page 13-10 shows the contents of the IER. Details on each bit follow.

UxBase+0x04

7	6	5	4	3	2	1	0
0	0	0	0	MS INTR	LS INTR	TX EMPTY INTR	DATA RDY INTR

Bits	Type	Function
7	R/W	0
6	R/W	0
5	R/W	0
4	R/W	0
3	R/W	Enables the MODEM Status Interrupt when set to logic 1.
2	R/W	Enables the Receiver Line Status Interrupt when set to logic 1.
1	R/W	Enables the Transmitter Holding Register Empty Interrupt when set to logic 1.
0	R/W	Enables the Received Data Available Interrupt (and time-out interrupts in the FIFO mode) when set to logic 1.

9.5.2.5 DLM

DLM is the Divisor Latch Most Significant Byte Register and used to set the Upper 8-bit of 16-bit Baud-Rate divisor value.

UxBase+0x00

7	6	5	4	3	2	1	0
Baud-Rate divisor Bit 15 ~ Baud-Rate divisor Bit 8							

Bits	Type	Function
7:0	R/W	Upper 8-bit of 16-bit Baud-Rate divisor.

9.5.2.6 IIR

In order to provide minimum software overhead during data character transfers, the UART prioritizes interrupts into four levels and records these in the Interrupt Identification Register. The four levels of interrupt conditions are, in order of priority

- Receiver Line Status
- Received Data Ready
- Transmitter Holding Register Empty
- MODEM Status

Bit3~Bit0 of the IIR are used to identify the highest priority interrupt that is pending.
Bit0 represents whether the interrupt is pending or not – If Bit0 is 1, no interrupt occurs now and if Bit0 is 0, an interrupt is pending and the IIR contents may be used as a pointer to the appropriate interrupt service routine. If two interrupts occurs simultaneously, Bit3~Bit0 of IIR represents the Higher priority number between these two interrupts. These bits represent the lower priority interrupt after CPU clears the higher priority interrupt.
When the CPU accesses the IIR, the UART freezes all interrupts and indicates the highest priority pending interrupt to the CPU. While this CPU access is occurring, the UART records new interrupts, but does not change its current indication until the access is complete.
Bit7~Bit6 of IIR are set to 1, when Bit0 of FCR(FIFO Control Register) is 1, otherwise these two bits are set to 0.

UxBase+0x08

7	6	5	4	3	2	1	0
FIFO EN		0	0	INTR ID			INTR PEND

Bits	Type	Function				
		Value	Priority Level	Interrupt Type	Interrupt Source	Interrupt Reset Condition
3:0	R	0001	-	None	None	-
		0110	Highest	Receiver Line Status	Overrun Error or Parity Error or Framing Error or Break Interrupt Reading the Line Status Register	Reading the Line Status Register
		0100	Second	Received Data Available	Receiver Data Available or Trigger Level Reached	Reading the Receiver Buffer Register or the FIFO drops below the trigger level
		1100	Second	Character Time-out Indication	No Characters have been removed from or input to the RCVR FIFO during the last 4 Character times and there is at least 1 Character in it during this time	Reading the Receiver Buffer Register
		0010	Third	Transmitter Holding Register Empty	Transmitter Holding Register Empty	Reading the IIR Register (if source of interrupt) or writing into the Transmitter Holding Register
		0000	Fourth	MODEM Status	Clear to Send or Data Set Ready or Ring Indicator or Data Carrier Detect	Reading the MODEM Status Register

9.5.2.7 FCR

This is a write-only register at the same location as the IIR (the IIR is a read-only register). This register is used to enable the FIFOs, clear the FIFOs and set the RCVR FIFO trigger level.

UxBase+0x08

7	6	5	4	3	2	1	0
RCVR TRIG LEVEL		-	-	-	XMIT RESET	RCVR RESET	FIFO EN

Bits	Type	Function
7:6	W	These two bits sets the trigger level for the RCVR FIFO interrupt **Value** **RCVR FIFO Trigger Level (Bytes)** 00 01 01 04 10 08 11 14
5:3	-	Reserved
2	W	Writing 1 resets the transmitter FIFO counter logic to 0. The shift register is not cleared. The 1 that is written to this bit position is self-clearing
1	W	Writing 1 resets the receiver FIFO counter logic to 0. The shift register is not cleared. The 1 that is written to this bit position is self-clearing
0	W	Writing 1 enables both the XMIT and RCVR FIFOs. Resetting FCR0 will clear all bytes in both FIFOs. When changing from FIFO Mode to 16C450 Mode and vice versa, data is automatically cleared from the FIFOs. This bit must be a 1 when other FCR bits are written to or they will not be programmed

9.5.2.8 LCR

The system programmer specifies the format of the asynchronous data communications exchange and set the Divisor Latch Access bit via the Line Control Register (LCR). The programmer can also read the contents of the Line Control Register. The read capability simplifies system programming and eliminates the need for separate storage in system memory of the line characteristics.

UxBase+0x0C

7	6	5	4	3	2	1	0
DLAB	SET BREAK	STICK PARITY	EVEN PARITY	PARITY ENABLE	STOPBIT NUMBER	WORD LENGTH SELECT	

Bits	Type	Function
7		This bit is the Divisor Latch Access Bit (DLAB). It must be set HIGH (logic 1) to access the Divisor Latches of the Baud Generator during a Read or Write operation. It must be set LOW (logic 0) to access the Receiver Buffer, the Transmitter Holding Register or the Interrupt Enable Register
6		This bit is the Break Control bit. It causes a break condition to be transmitted to the receiving UART. When it is set to logic 1, the serial output (SOUT) is forced to the Spacing (logic 0) state. The break is disabled by setting logic 0. The Break Control bit acts only on SOUT and has no effect on the transmitter logic. Note: This feature enables the CPU to alert a terminal in a computer communications system. If the following sequence is followed, no erroneous or extraneous characters will be transmitted because of the break.
5		This bit is the Stick Parity bit. When bits 3, 4 and 5 are logic 1 the Parity bit is transmitted and checked as logic 0. If bits 3 and 5 are 1 and bit 4 is logic 0 then the Parity bit is transmitted and checked as logic 1. If bit 5 is a logic 0 Stick Parity is disabled.
4		This bit is the Even Parity Select bit. When bit 3 is logic 1 and bit 4 is logic 0, an odd number of logic 1s is transmitted or checked in the data word bits and Parity bit. When bit 3 is logic 1 and bit 4 is logic 1, an even number of logic 1s is transmitted or checked.
3		This bit is the Parity Enable bit. When bit 3 is logic 1, a Parity bit is generated (transmit data) or checked (receive data) between the last data word bit and Stop bit of the serial data. (The Parity bit is used to produce an even or odd number of 1s when the data word bits and the Parity bit are summed).
2		This bit specifies the number of Stop bits transmitted and received in each serial character. If bit 2 is logic 0, one Stop bit is generated in the transmitted data. If bit 2 is logic 1 when a 5-bit word length is selected via bits 0 and 1, one and a half Stop bits are generated. If bit 2 is a logic 1 when either a 6-, 7- or 8-bit word length is selected, two Stop bits are generated. The Receiver checks the first Stop-bit only, regardless of the number of Stop bits selected.
1:0	R/W	These two bits specify the number of bits in each transmitted and received serial character. The encoding of bits 0 and 1 is as follows: **Value** **Character Length** 00 5 Bits 01 6 Bits 10 7 Bits 11 8 Bits

Programmable Baud Generator

HMS30C7210 UART can use only 3.692308MHz (PCLK) that is made from 48MHz (CCLK) clock at PMU. In addition, UART0 / UART1 can select 3.555556MHz (QCLK) that is also made at PMU for Smart Card operation and the selection between 3.692308MHz and 3.555556MHz is performed by setting CLOCKSEL (bit4 of UCR). The output frequency of the Baud Generator is 16 x the Baud [divisor # = (frequency input) / (baud rate x 16)]. Two 8-bit latches store the divisor in a 16-bit binary format. These Divisor Latches must be loaded during initialization to ensure proper operation of the Baud Generator. Upon loading either of the Divisor Latches, a 16-bit Baud counter is immediately loaded.
Baud rate table below provides decimal divisors to use with a frequency of 3.692308MHz. For baud rates of 38400 and below, the error obtained is minimal. The accuracy of the desired baud rate is dependent on the crystal frequency chosen. Using a divisor of zero is not recommended.

Desired Baud Rate	Decimal Divisor (Used to generate 16 x Clock)	Percent Error Difference Between Desired and Actual
50	4608	-
110	2094	0.026
300	768	-
1200	192	-
2400	96	-
4800	48	-
9600	24	-
19200	12	-
38400	6	-
57600	4	
115200	2	

Table 9-9 Baud Rate with Decimal Divisor at 3.92308MHz Clock Input

9.5.2.9 MCR(Uart5 Only)

This register controls the interface with the MODEM or data set (or a peripheral device emulating a MODEM) and is valid at only UART5 because the onlu UART5 has the external modem pins.
In addtion, MCR should not be accessed at UART0 and UART1, because UART0 and UART1 use this address for address of SMR(Smart Card Mode Register).

UxBase+0x10

7	6	5	4	3	2	1	0
0	0	0	LOOP	-	-	RTS UART5 Only	DTR UART5 Only

Bits	Type	Function
7:5	R	These bits are permanently set to logic 0
4	R/W	This bit provides a local loop back feature for diagnostic testing of the UART. When bit 4 is set to logic 1, the following occur: the transmitter Serial Output (SOUT) is set to the Marking (logic 1) state; the receiver Serial Input (SIN) is disconnected; the output of the Transmitter Shift Register is "looped back" into the Receiver Shift Register input; the four MODEM Control inputs (NCTS, NDSR, NDCD and NRI) are disconnected; and the two MODEM Control outputs (NDTR and NRTS) are internally connected to the four MODEM Control inputs, and the MODEM Control output pins are forced to their inactive state (HIGH). On the diagnostic mode, data that is transmitted is immediately received. This feature allows the processor to verify the transmit- and received-data paths of the UART. In the diagnostic mode, the receiver and transmitter interrupts are fully operational. Their sources are external to the part. The MODEM Control interrupts are also operational, but the interrupts sources are now the lower four bits of the MODEM Control Register instead of the four MODEM Control inputs. The interrupts are still controlled by the Interrupt Enable Register.
3:2	-	Reserved
1	R/W	This bit controls the Request to Send (nURTS) output. Bit 1 affects the NRTS output in a manner identical to that described above for bit 0.
0	R/W	This bit controls the Data Terminal Ready (nUDTR) output. When bit is set to logic 1, the NDTR output is forced to logic 0. When bit 0 is reset to logic 0, the NDTR output is forced to logic 1. **Note**: The NDTR output of the UART may be applied to an EIA inverting line driver (such as the DS1488) to obtain the proper polarity input at the succeeding MODEM or data set.

9.5.2.10 LSR

This register provides status information to the CPU concerning the data transfer.

UxBase+0x14

7	6	5	4	3	2	1	0
FIFO ERR	TEMT	THRE	BI	FE	PE	OE	DR

Bits	Type	Function
7	R	In the 16C450 mode this is always 0. In the FIFO mode LSR7 is set when there is at least one parity error, framing error or break indication in the FIFO. LSR7 is cleared when the CPU reads the LSR, if there are no subsequent errors in the FIFO.
6	R	This bit is the Transmitter Empty (TEMT) indicator. Bit 6 is set to a logic 1 whenever the Transmitter Holding Register (THR) and the Transmitter Shift Register (TSR) are both empty. It is reset to logic 0 whenever either the THR or TSR contains a data character. In the FIFO mode this bit is set to one whenever the transmitter FIFO and register are both empty.
5	R	This bit is the Transmitter Holding Register Empty (THRE) indicator. Bit 5 indicates that the UART is ready to accept a new character for transmission. In addition, this bit causes the UART to issue an interrupt to the CPU when the Transmit Holding Register Empty Interrupt enable is set HIGH. The THRE bit is set to a logic 1 when a character is transferred from the Transmitter Holding Register into the Transmitter Shift Register. The bit is reset to logic 0 concurrently with the loading of the Transmitter Holding Register. In the FIFO mode this bit is set when the XMIT FIFO is empty; it is cleared when at least 1 byte is written to the XMIT FIFO. It may cause transmit error to write transmit FIFOs after polling this bit. If you want to use the transmit idle status to decides when to write the transmit FIFOs in polling mode, you had better to check the TEMP(bit6 of this register) rather than this bit. But, this bit can be used to check the timing to write transmit data in polling mode when FIFO is disabled. This bit can also be used in the interrupt mode.
4	R	This bit is the Break Interrupt (BI) indicator. Bit 4 is set to logic 1 whenever the received data input is held in the Spacing (logic 0) state for longer than a full word transmission time (that is, the total time of Start bit + data bits + Parity + Stop bits). The BI indicator is reset whenever the CPU reads the contents of the Line Status Register. In the FIFO mode this error is associated with the particular character in the FIFO it applies to. This error is revealed to the CPU when its associated character is at the top of the FIFO. When break occurs, only one zero character is loaded into the FIFO. The next character transfer is enabled after SIN goes to the marking state and receives the next valid start bit. **Note**: Bits 1--4 are the error conditions that produce a Receiver Line Status interrupt whenever any of the corresponding conditions are detected and the interrupt is enabled.
3	R	This bit is the Framing Error (FE) indicator. Bit 3 indicates that the received character did not have a valid stop bit. Bit 3 is set to logic 1 whenever the Stop bit following the last data bit or parity bit is detected as a logic 0 bit (Spacing level). The FE indicator is reset whenever the CPU reads the contents of the Line Status Register. In the FIFO mode this error is associated with the particular character in the FIFO it applies to. This error is revealed to the CPU when its associated character is at the top of the FIFO. The UART will try to re-synchronize after a framing error. To do this it assumes that the framing error was due to the next start bit, so it samples this "start" bit twice and then takes in the "data".
2	R	This bit is the Parity Error (PE) indicator. Bit 2 indicates that the received data character does not have the correct even or odd parity, as selected by the even-parity-select bit. The PE bit is set to logic 1 upon detection of a parity error and is reset to logic 0 whenever the CPU reads the contents of the Line Status Register. In the FIFO mode, this error is associated with the particular character in the FIFO it applies to. This error is revealed to the CPU when its associated character is at the top of the FIFO.
1	R	This bit is the Overrun Error (OE) indicator. Bit 1 indicates that data in the Receiver Buffer Register was not read by the CPU before the next character was transferred into the Receiver Buffer Register, thereby destroying the previous character. The OE indicator is set to logic 1 upon detection of an overrun condition and reset whenever the CPU reads the contents of the Line Status Register. If the FIFO mode data continues to fill the FIFO beyond the trigger level, an overrun error will occur only after the FIFO is full and the next character has been completely received in the shift register. OE is indicated to the CPU as soon as it happens. The character in the shift register is overwritten, but it is not transferred to the FIFO.
0	R	This bit is the receiver Data Ready (DR) indicator. Bit 0 is set to logic 1 whenever a complete incoming character has been received and transferred into the Receiver Buffer Register or the FIFO. Bit 0 is reset to logic 0 by reading all of the data in the Receiver Buffer Register or the FIFO.

Some bits in LSR are automatically cleared when CPU reads the LSR register, so

interrupt handling routine should be written that if once reads LSR, then keep the value through entire the routine because second reading LSR returns just reset value.

9.5.2.11 MSR (Uart5 Only)

This register provides the current state of the control lines from the MODEM (or peripheral device) to the CPU. In addition to this current-state information, four bits of the MODEM Status Register provide change information. These bits are set to logic 1 whenever a control input from the MODEM change state. They are reset to logic 0 whenever the CPU reads the MODEM Status Register.

UxBase+0x18

7	6	5	4	3	2	1	0
DCD	RI	DSR	CTS	DDCD	TERI	DDSR	DCTS

Bits	Type	Function
7	R/O	This bit is the complement of the Data Carrier Detect (nUDCD) input. If bit 4 of the MCR is set to a 1, this bit is equivalent to OUT2 in the MCR. Note: Whenever this bit changes its state, an interrupt is generated if the MODEM Status Interrupt is enabled.
6	R/O	This bit is the complement of the Ring Indicator (nURING) input. If bit 4 of the MCR is set to a 1, this bit is equivalent to OUT1 in the MCR. Note: Whenever this bit changes its state from a HIGH to a LOW state, an interrupt is generated if the MODEM Status Interrupt is enabled.
5	R/O	This bit is the complement of the Data Set Ready (nUDSR) input. If bit 4 of the MCR is set to a 1, this bit is equivalent to DTR in the MCR. Note: Whenever this bit changes its state, an interrupt is generated if the MODEM Status Interrupt is enabled.
4	R/O	This bit is the complement of the Clear to Send (nUCTS) input. If bit 4 (loop) of the MCR is set to a 1, this bit is equivalent to RTS in the MCR. Note: Whenever this bit changes its state, an interrupt is generated if the MODEM Status Interrupt is enabled.
3	R/O	This bit is the Delta Data Carrier Detect (nUDCD) indicator. Bit 3 indicates that the nUDCD input to the chip has changed state since the last time it was read by the CPU. Note: Whenever bit 0, 1, 2 or 3 is set to logic 1, a MODEM Status Interrupt is generated.
2	R/O	This bit is the Trailing Edge of Ring Indicator (TERI) detector. Bit 2 indicates that the nURING input to the chip has changed from a LOW to a HIGH state.
1	R/O	This bit is the Delta Data Set Ready (nUDSR) indicator. Bit 1 indicates that the nUDSR input to the chip has changed state since the last time it was read by the CPU.
0	R/O	This bit is the Delta Clear to Send (nUCTS) indicator. Bit 0 indicates that the nUCTS input to the chip has changed state since the last time it was read by the CPU.

9.5.2.12 SCR

This 8-bit Read/Write Register does not control the UART in any way. It is intended as a scratchpad register to be used by the programmer to hold data temporarily.

UxBase+0x1C

7	6	5	4	3	2	1	0
DATA							

Bits	Type	Function
7:0	R/W	Temporary data storage

9.5.2.13 UCR (Uart Configuration Register)

To make the Smart Card Interface mode set, SMCARDEN and UARTEN are set to '1' at the same time.
If you use SIR function, you must set SIREn and UART En bit at the same time.

UxBase+0x30

7	6	5	4	3	2	1	0
-	-	SMCARDEN Uart0/1 only	CLOCKSEL Uart0/1 only	SIR Loop Back Uart4 only	Full Duplex Force Uart4 only	SIREN Uart4 only	UARTEN

Bits	Type	Function
7:6	-	Reserved
5	R/W	Smart Card Interface mode set 0 = Smart Card interface disable 1 = Smart Card interface enable
4	R/W	Clock Select 0 = 3.6864MHz 1 = 3.5712MHz
3	R/W	SIR Loop-back Test (Uart1 only) 0 = SIR Loop-back Test disable 1 = SIR Loop-back Test enable.
2	R/W	SIR Full-duplex Force (Uart1 only) 0 = Half Duplex. 1 = Full Duplex.
1	R/W	SIR Enable (Uart1 only) 0 = SIR Mode disable 1 = SIR Mode enable
0	R/W	UART Enable. 0 = UART disable (Power-Down), UART Clock stop. 1 = UART enable.

9.5.3 FIFO Interrupt Mode Operation

When the RCVR FIFO and receiver interrupts are enabled (FCR 0 = 1, IER 0 = 1) RCVR interrupts occur as follows:

- The received data available interrupt will be issued to the CPU when the FIFO has reached its programmed trigger level. It will be cleared as soon as the FIFO drops below its programmed trigger level.
- The IIR receive data available indication also occurs when the FIFO trigger level is reached, and like the interrupt, it is cleared when the FIFO drops below the trigger level.
- The receiver line status interrupt (IIR-06), as before, has higher priority than the received data available (IIR-04) interrupt.
- The data ready bit (LSR 0) is set as soon as a character is transferred from the shift register to the RCVR FIFO. It is reset when the FIFO is empty.

When RCVR FIFO and receiver interrupts are enabled, RCVR FIFO time-out interrupts occurs as follows:

- A FIFO time-out interrupt occurs if the following conditions exist: at least one character is in the FIFO
- the most recent serial character received was longer than four continuous character times ago (if two stop bits are programmed, the second one is included in this time delay)
- the most recent CPU read of the FIFO was longer than four continuous character times ago This will cause a maximum character received to interrupt issued delay of 160 ms at 300 baud with a 12-bit character.
- Character times are calculated by using the RCLK input, which is the internal signal of UART for a clock signal (this makes the delay proportional to the baud rate).
- When a time-out interrupt has occurred, it is cleared and the timer is reset when the CPU reads one character from the RCVR FIFO.
- When a time-out interrupt has not occurred the time-out timer is reset after a new character is received or after the CPU reads the RCVR FIFO.

When the XMIT FIFO and transmitter interrupts are enabled (FCR 0 = 1, IER 1 = 1), XMIT interrupts occurs as follows:

- The transmitter holding register interrupt (02) occurs when the XMIT FIFO is empty. It is cleared as soon as the transmitter holding register is written to (1 to 16 characters may be written to the XMIT FIFO while servicing this interrupt) or the IIR is read.
- The transmitter FIFO empty indications will be delayed 1 character time minus the last stop bit time whenever the following occurs: THRE = 1 and there has not been at least two bytes at the same time in the transmit FIFO since the last THRE = 1. The first transmitter interrupt affect changing FCR0 will be immediate if it is enabled.

Character time-out and RCVR FIFO trigger level interrupts have the same priority as the current received data available interrupt; XMIT FIFO empty has the same priority as the current transmitter holding register empty interrupt.

9.5.4 FIFO Polling Mode Operation

When FCR is set to 1 and all bits of IER are clear to '0', UART is put to the FIFO polled mode of operation. In this mode, user program will check Receive and Transmit status via Line Status Register. CPU should do appropriate operation at each case of Line Status Register.:

- LSR0 will be set as long as there is one byte in the Receive FIFO.
- LSR1~LSR4 will specify which error has occurred. Character error status is handled the same way when in the interrupt mode, the IIR is not affected since IER2 is '0'.
- LSR5 will indicate when the Transmit FIFO is empty.
- LSR6 will indicate that both the Transmit FIFO and shift register are empty.
- LSR7 will indicate whether there are any errors in the Receive FIFO
- There are no trigger level reached or timeout condition indicated in the FIFO Polled Mode.

9.6 SMART Card Interface

A Smart Card interface is an extension of UART0/UART1 functions and supports the ISO7816-3 standard.

The switchover between normal UART function and Smart Card interface function is controlled by setting a UART Configuration register (UCR) appropriately.
If the UARTEN bit and SMCARDEN bit of UCR are set simultaneously, the UART0 and UART1 are changed from normal UART mode to Smart Card Interface mode.

FEATURES

- Card detect function(support the detection of case that card's present and absent both)
- Execute automatic contact activation and deactivation sequence.
- Programmable clock cycle number setting of Reset transition.
- Built-in baud generator allows any bit rate to be selected.
- Supports the asynchronous Smart Card communication.
- Half-duplex data communication
- 8-bit data length
- Support direct convention and indirect convention both
- Parity bit generation and check
- Transmit error signal (parity error) in receive mode
- Error signal detection and automatic retransmission in transmission mode
- Programmable extra guard time in transmission mode
- Programmable waiting time cycle number.
- Clock is enabled or disabled by register setting.

9.6.1 External Signals

UART0 and UART1 have Smart Card Interface extension. At setting the Smart Card Interface. Enable bit of UCR, the signals table below are enabled at each UART / Smart Card Interface.

These Smart Card Interface pin names are same as HMS30C7210 Top pin names.
To get the information about pin number of Smart Card Interface signal at Chip, refer to "Table 2-3 Detail Pin Description".

Pin Name	Type	Description
SCPRES[1:0]	I	Card Present signal This signal indicates that Smart Card is present(if this signal is logic '1') or not(if this signal is logic '0') in the slot. The Card detect interrupt is generated at the rising edge(Card is inserted) and falling edge(Card is removed) both if the Card detect interrupt is enable in IER
SCIO[1:0]	I/O	Data in/out signal from/to external Smart Card. This signal shall be fixed to logic '0' at idle state. This signal is set in receive mode except transmitting data or parity error flag after contact activation starts
SCRST[1:0]	O	Smart Card reset signal. This signal is fixed to logic '0' at idle state. On starting of contact activation sequence, this signal remains to logic '0' waiting ATR until the number of clock cycle set in the RTR. If the ATR is not received until that number of clock cycle, CRST is set to logic '1' and waits for ATR during the number of clock cycle set in the RTR once more. If There is no ATR and the clock cycle elapses(the initialization of Smart Card fails) ,the contact deactivation start and the CRST is et to logic '0'
SCCLK[1:0]	O	Smart Card Clock signal. This clock starts when contact activation sequence starts(If CardInit and CLKEn are set to '1' in the SMR). During the data transfer, 1-bit period is configured to the any number of CCLK cycle as configured by divider value of DLL/DLM and BaudSel of SMR if CLKEn of SMR is set to '1'. If the BaudSel is set to '1', 1-bit period is "31 X divider-value". If the BaudSel is set to '0', 1-bit period is "16 X divider value". The CCLK can be disabled by setting CLKEn of SMR to '0'. In this case, CCLK is fixed to '0' if CLKPol is '0' and CCLK is fixed to '0' if CCLK is fixed to '1'

Refer to Figure 2-1. 208 Pin diagram.

9.6.2 Registers

After UART0 and UART1 is set to the Smart Card Interface mode, the register set is changed from the normal UART registers to Smart Card Interface(SCI) registers as blow.

Address	Name	Width	Default	Description
0x8005.4000	SCI0Base	-	-	Smart Card Interface 0 Base
0x8005.5000	SCI1Base	-	-	Smart Card Interface 1 Base
SCIxBase+0x00	RBR	8	0x00	Receiver Buffer Register (DLAB = 0, Read Only)
	THR	8	0x00	Transmitter Holding Register (DLAB = 0, Write Only)
	DLL	8	0x00	Divisor Latch Least Significant Byte (DLAB = 1, Read/Write)
SCIxBase+0x04	IER	8	0x00	Interrupt Enable Register (DLAB = 0, Read/Write)
	DLM	8	0x00	Divisor Latch Most Significant Byte (DLAB = 1, Read/Write)
SCIxBase+0x08	IIR	8	0x01	Interrupt Identification Register (Read Only)
	FCR	8	0x00	FIFO Control Register (Write Only)
SCIxBase+0x0C	LCR	8	0x00	Line Control Register(Read/Write)
SCIxBase+0x10	SMR	12	0x00	Smart Card Mode Register(Read/Write)
SCIxBase+0x14	LSR	8	0x60	Line Status Register(Read Only)
SCIxBase+0x18	SSR	8	0xX0	Smart Card Status Register(Read Only)
SCIxBase+0x1C	SCR	8	0x00	Scratch Register(Read/Write)
SCIxBase+0x20	RTR	16	0x0190	Reset Timing Register(Read/Write)
SCIxBase+0x24	RNR	8	0x00	Retransmit number Register(Read/Write)
SCIxBase+0x28	WTR	24	0x2580	Waiting Time Register(Read/Write)
SCIxBase+0x2C	EGR	8	0x00	Smart Card Interface Extra-Guard Time Register(Read/Write).
SCIxBase+0x30	UCR	6	0x00	UART Configuration Register(Read/Write)

Table 9-10 Smart Card Interface Register Summary

9.6.2.1 RBR

RBR is the Receive Buffer Register and stores the data from serial input. This register is read-only and can be accessed when DLAB(Bit7 of Line Control Register) is set to 0.

SCIxBase+0x00

7	6	5	4	3	2	1	0
Receive Data Bit 7 ~ Receive Data Bit 0							

Bits	Type	Function
7:0	R	Receive Byte that is received from Serial input.

9.6.2.2 THR

THR is the Transmit Buffer Register and stores the data to be transmitted through serial output. This register is write-only and can be accessed when DLAB(Bit7 of Line Control Register) is set to 0.

SCIxBase+0x00

7	6	5	4	3	2	1	0
Transmit Data Bit 7 ~ Transmit Data Bit 0							

Bits	Type	Function
7:0	W	Transmit Byte that is transmitted through Serial output.

9.6.2.3 DLL

DLL is the Divisor Latch Least Significant Byte Register and used to set the lower 8-bit of 16-bit Baud-Rate divisor value.

SCIxBase+0x00

7	6	5	4	3	2	1	0
Baud-Rate divisor Bit 7 ~ Baud-Rate divisor Bit 0							

Bits	Type	Function
7:0	R/W	Lower 8-bit of 16-bit Baud-Rate divisor.

9.6.2.4 IER/DLM

This register enables the five types of Smart Card Interface interrupts. Each interrupt can individually activate the interrupt (INTUART) output signal. It is possible to totally disable the interrupt Enable Register (IER). Similarly, setting bits of the IER register to logic 1 enables the selected interrupt(s). Disabling an interrupt prevents it from being indicated as active in the IIR and from activating the INTUART output signal. All other system functions operate in their normal manner, including the setting of the Line Status and Smart Card Status Registers. Table 13-6: Summary of registers on page 13-10 shows the contents of the IER. Details on each bit follow.

SCIxBase+0x04

7	6	5	4	3	2	1	0
0	0	CARD DET INTR	WAIT TIME INTR	TX LS INTR	RX LS INTR	TX EMPTY INTR	DATA RDY INTR

Bits	Type	Function
7	R/W	0
6	R/W	0
5	R/W	Enable the Card Detect (Card insertion or removal) interrupt
4	R/W	Enables the Initialization Fail (ATR is not received) Interrupt or Waiting Time Out interrupt
3	R/W	Enables the Transmitter Line Status(Parity error) Interrupt when set to logic 1.
2	R/W	Enables the Receiver Line Status (Overrun/Parity error) Interrupt when set to logic 1.
1	R/W	Enables the Transmitter Holding Register Empty Interrupt when set to logic 1.
0	R/W	Enables the Received Data Available Interrupt (and time-out interrupts in the FIFO mode) when set to logic 1.

9.6.2.5 DLM

DLM is the Divisor Latch Most Significant Byte Register and used to set the Upper 8-bit of 16-bit Baud-Rate divisor value.

SCIxBase+0x00

7	6	5	4	3	2	1	0
Baud-Rate divisor Bit 15 ~ Baud-Rate divisor Bit 8							

Bits	Type	Function
7:0	R/W	Upper 8-bit of 16-bit Baud-Rate divisor.

9.6.2.6 IIR

In order to provide minimum software overhead during data character transfers, the Smart Card Interface prioritizes interrupts into five levels and records these in the Interrupt Identification Register. The five levels of interrupt conditions are, in order of priority

- Card Detect (Card insert or removal)
- Receiver Line Status / Transmitter Line Line Status
- Received Data Ready
- Transmitter Holding Register Empty
- Card Initialize Fail / Waiting Time Out

Bit4~Bit0 of the IIR are used to identify the highest priority interrupt that is pending.
Bit0 represents whether the interrupt is pending or not – If Bit0 is 1, no interrupt occurs now and if Bit0 is 0, an interrupt is pending and the IIR contents may be used as a pointer to the appropriate interrupt service routine. If two interrupts occurs simultaneously, Bit4~Bit0 of IIR represents the Higher priority number between these two interrupts. These bits represent the lower priority interrupt after CPU clears the higher priority interrupt.
When the CPU accesses the IIR, the UART freezes all interrupts and indicates the highest priority pending interrupt to the CPU. While this CPU access is occurring, the UART records new interrupts, but does not change its current indication until the access is complete.
Bit7~Bit6 of IIR are set to 1, when Bit0 of FCR(FIFO Control Register) is 1, otherwise these two bits are set to 0.

SCIxBase+0x08

7	6	5	4	3	2	1	0
FIFO EN		0	INTR ID				INTR PEND

Bits	Type	Function				
		Value	Priority Level	Interrupt Type	Interrupt Source	Interrupt Reset Condition
4:0	R	00001	-	None	None	-
		01000	Highest	Card Detect Status	Card insert or removal from/to slot	Reading the Smart Card Status Register
		00110	Second	Receiver Line Status	Overrun Error or Parity Error	Reading the Line Status Register
		10110	Second	Transmitter Line Status	Transmit Parity Error	Reading the Line Status Register
		00100	Third	Receiver Data Avaliable	Receiver Data Available or Trigger Level Reached	Reading the Receiver Buffer Register or the FIFO drops below the trigger level
		10100	Third	Character Time-out Indication	No Characters have been removed from or input to the RCVR FIFO during the last 4 Character times and there is at least 1 Character in it during this time	Reading the Receiver Buffer Register
		00010	Fourth	Transmitter Holding Register Empty	Transmitter Holding Register Empty	Reading the IIR Register (if source of interrupt) or writing into the Transmitter Holding Register
		00000	Fifth	Wating Timeout	Receive serial data waiting time is elapsed	Reading the Smart Card Status Register

	10000	Fifth	Card Initialization Fail	The External Smart Card dose not give the ATR during the initialization cycle	Reading the Smart Card Status Register

9.6.2.7 FCR

This is a write-only register at the same location as the IIR (the IIR is a read-only register). This register is used to enable the FIFOs, clear the FIFOs and set the RCVR FIFO trigger level.

SCIxBase+0x08

7	6	5	4	3	2	1	0
RCVR TRIG LEVEL		-	-	-	XMIT RESET	RCVR RESET	FIFO EN

Bits	Type	Function
7:6	W	These two bits sets the trigger level for the RCVR FIFO interrupt Value RCVR FIFO Trigger Level (Bytes) 00 01 01 04 10 08 11 14
5:3	-	Reserved
2	W	Writing 1 resets the transmitter FIFO counter logic to 0. The shift register is not cleared. The 1 that is written to this bit position is self-clearing
1	W	Writing 1 resets the receiver FIFO counter logic to 0. The shift register is not cleared. The 1 that is written to this bit position is self-clearing
0	W	Writing 1 enables both the XMIT and RCVR FIFOs. Resetting FCR0 will clear all bytes in both FIFOs. When changing from FIFO Mode to 16C450 Mode and vice versa, data is automatically cleared from the FIFOs. This bit must be a 1 when other FCR bits are written to or they will not be programmed

9.6.2.8 LCR

The system programmer specifies the format of the asynchronous data communications exchange and set the Divisor Latch Access bit via the Line Control Register (LCR). The programmer can also read the contents of the Line Control Register. The read capability simplifies system programming and eliminates the need for separate storage in system memory of the line characteristics.

SCIxBase+0x0C

7	6	5	4	3	2	1	0
DLAB	SET BREAK	STICK PARITY	EVEN PARITY	PARITY ENABLE	STOPBIT NUMBER	WORD LENGTH SELECT	

Bits	Type	Function
7		This bit is the Divisor Latch Access Bit (DLAB). It must be set HIGH (logic 1) to access the Divisor Latches of the Baud Generator during a Read or Write operation. It must be set LOW (logic 0) to access the Receiver Buffer, the Transmitter Holding Register or the Interrupt Enable Register
6		This bit is the Break Control bit. This bit must be set to '0' at the Smart Card Interface mode It causes a break condition to be transmitted to the receiving UART. When it is set to logic 1, the serial output (SOUT) is forced to the Spacing (logic 0) state. The break is disabled by setting logic 0. The Break Control bit acts only on SOUT and has no effect on the transmitter logic. Note: This feature enables the CPU to alert a terminal in a computer communications system. If the following sequence is followed, no erroneous or extraneous characters will be transmitted because of the break.
5		This bit is the Stick Parity bit. This bit must be set to '0' at the Smart Card Interface mode When bits 3, 4 and 5 are logic 1 the Parity bit is transmitted and checked as logic 0. If bits 3 and 5 are 1 and bit 4 is logic 0 then the Parity bit is transmitted and checked as logic 1. If bit 5 is a logic 0 Stick Parity is disabled.
4		This bit is the Even Parity Select bit. This bit must be set to '1' at the Smart Card Interface direct convention mode This bit must be set to '0' at the Smart Card Interface indirect convention mode When bit 3 is logic 1 and bit 4 is logic 0, an odd number of logic 1s is transmitted or checked in the data word bits and Parity bit. When bit 3 is logic 1 and bit 4 is logic 1, an even number of logic 1s is transmitted or checked.
3		This bit is the Parity Enable bit. This bit must be set to '1' at the Smart Card Interface mode When bit 3 is logic 1, a Parity bit is generated (transmit data) or checked (receive data) between the last data word bit and Stop bit of the serial data. (The Parity bit is used to produce an even or odd number of 1s when the data word bits and the Parity bit are summed).
2		This bit specifies the number of Stop bits transmitted and received in each serial character. This bit must be set to '1' at the Smart Card Interface mode. If bit 2 is logic 0, one Stop bit is generated in the transmitted data. If bit 2 is logic 1 when a 5-bit word length is selected via bits 0 and 1, one and a half Stop bits are generated. If bit 2 is a logic 1 when either a 6-, 7- or 8-bit word length is selected, two Stop bits are generated. The Receiver checks the first Stop-bit only, regardless of the number of Stop bits selected.
1:0	R/W	These two bits specify the number of bits in each transmitted and received serial character. These two bits must be '11'(8-bit) at Smart Card Interface mode The encoding of bits 0 and 1 is as follows: Value Character Length 00 5 Bits 01 6 Bits 10 7 Bits 11 8 Bits

Programmable Baud Generator

The UART0,1 / Smart Card Interface contains a programmable Baud Generator that is capable of taking clock input (3.692308MHz or 3.555556MHz) and dividing it by any divisor from 2 to 2^{16}-1. UART0,1/ Smart Card Interface can select between 3.692308MHz and 3.555556HMz is performed by setting CLOCKSEL (bit4 of UCR).These Divisor Latches must be loaded during initialization to ensure proper operation of the Baud Generator. Upon loading either of the Divisor Latches, a 16-bit Baud counter is immediately loaded.The output frequency of the Baud Generator is 16 x the Baud [divisor # = (frequency input) / (baud rate x 16)], if BaudSel of SMR is logic '0' or 31 x the Baud [divisor # = (frequency input) / (baud rate x 31)], if BaudSel of SMR is logic '1'. The selection of BaudSel depends on FI bits of ATR in the Smart Card Initialization process. If the forth bit of FI is logic '0', the BaudSel shall be set to '1'. If the forth bit of FI is logic '1', the BaudSel shall be set to '0'. Two 8-bit latches store the divisor in a 16-bit binary format.
Baud rate table below provides decimal divisors to use with a frequency of 3.555556MHz and BaudSel is logic '1' or '0'. Using a divisor of zero is not recommended.

Desired Baud Rate	Decimal Divisor (Used to generate 16 x Clock)	Percent Error Difference Between Desired and Actual
9600	12 (BaudSel = 1, FI = 0001)	-
6400	18 (BaudSel = 1, FI = 0010)	-
4800	24 (BaudSel = 1, FI = 0011)	-
3200	36 (BaudSel = 1, FI = 0100)	-
2400	48 (BaudSel = 1, FI = 0101)	-
1920	60 (BaudSel = 1, FI = 0110)	-
6975	32 (BaudSel = 0, FI = 1001)	-
4650	48 (BaudSel = 0, FI = 1010)	-
3487	64 (BaudSel = 0, FI = 1011)	-
2325	96 (BaudSel = 0, FI = 1100)	
1744	128 (BaudSel = 0, FI = 1101)	

Table 9-11 Baud Rate with Decimal Divisor at 3.55556MHz Clock Input

9.6.2.9 SMR (Smart Card Mode Register)

This register controls the configuration when Smart Card interface mode is enabled.
SCIxBase+0x10

				11	10	9	8
				DISINIT	DIRCTLEN	RSTVAL	IOVAL
7	6	5	4	3	2	1	0
CARDINIT	RETRANEN	DATAPOL	-	DATADIR	CLKVAL	CLKEN	BAUDSEL

Bits	Type	Function
11	R/W	Smart Card Initialization Sequence disable bit Before data is transferred between Smart Card and SCI, Smart Card Contact must be activated and ATR must be transferred from Smart Card to SCI. If this bit is reset to '0', the above initialization sequence is performed as soon as CARDINIT (bit7 of this register) is set to '1'. Otherwise, the smart Card skip the initialization sequence and ready for data transfer, as soon as CARDINIT is set to '1'.
10	R/W	Direct control of CIO/CRST enable bit If this bit is set to '1', CIO and CRST pin are controlled directly by setting IOVAL(bit8 of this register) or RSTVAL(bit9 of this register) in spite of current state of initialization sequence. If this bit is reset to '0', CRST and CIO pin's levels are controlled only by state of initialization sequence. For example, CRST is changed from '0' to '1' when SCI is in Smart Card Contact Activation state, CRST is fixed to '1' when SCI is in the data transfer state.
9	R/W	Reset bit(CRST signal) level select bit when Direct control of CIO/CRST is enabled This bit is used to indicate state of the CRST pin when Direct control of CIO/CRST is enabled (when DIRCTLEN is set to '1'). If this bit is reset to '0' and DIRCTLEN(bit10 of this register) is '1' , the CRST pin is fixed to logic '0' state. Otherwise, the CCLK pin is fixed to logic '1' state
8	R/W	Data bit(CIO signal) level select bit when Direct control of CIO/CRST is enabled This bit is used to indicate state of the CIO pin when Direct control of CIO/CRST is enabled (when DIRCTLEN is set to '1'). If this bit is reset to '0' and DIRCTLEN(bit10 of this register) is '1' , the CIO pin is fixed to logic '0' state. Otherwise, the CCLK pin is fixed to logic '1' state
7	R/W	Smart Card Initialization bit. The contact initialization sequence starts when this bit and CARDEN bit of UCR is set to '1' After Card Initialization sequence is successfully finished, the Smart Card interface can exchange the data with the external card. This bit shall be reset to '0' to make the contact deactivation sequence start at the end of data transfer with the external card This bit is also reset to '0' automatically in the case that the external card does not give the ATR and initialization is failed. At this case, the contact deactivation sequence starts automatically
6	R/W	Retransmit Enable bit This bit is set to enable the retransmission of parity-errored data at transmitter operation and the transmission of error flag at receiver operation If this bit is reset to '0', the function of error flag transmission and data retransmission is disable
5	R/W	Data bit(CIO signal) polarity bit If this bit is reset to '0', the logic 1 level of CIO corresponds to state Z and the logic 0 level to state A. Otherwise, the logic 1 level corresponds to state A and the logic 0 level to state Z This bit shall be reset to '1' at direct convention and this bit shall be set to '1' at indirect convention
4	-	Reserved for normal UART function.
3	R/W	Data bit(CIO signal) direction select bit When this bit is reset to '0', the data frame transfer is performed in LSB-first order. Otherwise, the data frame is performed in MSB-first order. This bit shall be reset to '1' at direct convention and this bit shall be set to '1' at indirect convention
2	R/W	CCLK level select bit when CCLK is disabled This bit is used to indicate state of the CCLK pin when CCLK is not enabled (when CLKEN is reset to '0'). If this bit is reset to '0' and CLKEN(bit1 of this register) is '0' , the CCLK pin is fixed to logic '0' state. Otherwise, the CCLK pin is fixed to logic '1' state
1	R/W	CCLK enable bit This bit is used to enable or disable the CCLK pin. If this bit is reset to '0', CCLK pin is disabled and fixed to logic level as indicated to CLKVALI(bit2 of this register) Otherwise, CCLK pin is enabled and Clock signal is transferred

		to CCLK pin
0	R/W	Baud Select bit The output frequency of the Baud Generator is 16 x the Baud [divisor # = (frequency input) / (baud rate x 16)], this bit is logic '0'. 31 x the Baud [divisor # = (frequency input) / (baud rate x 31)], if this bit is logic '1'.

9.6.2.10 LSR

This register provides status information to the CPU concerning the data transfer.
SCIxBase+0x14

7	6	5	4	3	2	1	0
FIFO ERR	TEMT	THRE	TXPE	-	PE	OE	DR

Bits	Type	Function
7	R	In the 16C450 mode this is always 0. In the FIFO mode LSR7 is set when there is at least one parity error in the FIFO. LSR7 is cleared when the CPU reads the LSR, if there are no subsequent errors in the FIFO.
6	R	This bit is the Transmitter Empty (TEMT) indicator. Bit 6 is set to a logic 1 whenever the Transmitter Holding Register (THR) and the Transmitter Shift Register (TSR) are both empty. It is reset to logic 0 whenever either the THR or TSR contains a data character. In the FIFO mode this bit is set to one whenever the transmitter FIFO and register are both empty.
5	R	This bit is the Transmitter Holding Register Empty (THRE) indicator. Bit 5 indicates that the UART/Smart Card interface is ready to accept a new character for transmission. In addition, this bit causes the UART/Smart Card Interface to issue an interrupt to the CPU when the Transmit Holding Register Empty Interrupt enable is set HIGH. The THRE bit is set to a logic 1 when a character is transferred from the Transmitter Holding Register into the Transmitter Shift Register. The bit is reset to logic 0 concurrently with the loading of the Transmitter Holding Register. In the FIFO mode this bit is set when the XMIT FIFO is empty; it is cleared when at least 1 byte is written to the XMIT FIFO.
4	R-	This bit is transmitter parity error (TXPE) indicator. If RETRANEN bit of SMR is set to '1', this bit is set to logic 1 in the case that the external card transmits the parity error flag of received data and the interface device re-transmit the errored data frame for the times of the RNR value, but parity will not be removed and the external card transmit error flag also If RETRANEN bit of SMR is set to '0', this bit is set to '1' as soon as the parity error flag is received. This bit is reset whenever the CPU reads the contents of the Line Status Register. In the FIFO mode this error is associated with the particular character in the FIFO it applies to. This error is revealed to the CPU when its associated character is at the top of the FIFO. Note: Bits 4 is the error conditions that produce a Transmitter Line Status interrupt whenever any of the corresponding conditions are detected and the interrupt is enabled.
3	-	This bit is reserved at Smart Card Interface mode
2	R	This bit is the Receive Parity Error (PE) indicator. If RETRANEN bit of SMR is set to '0', this bit is set to '1' upon detection of a parity error. If RETRANEN bit of SMR is set to '1', the interface detects the received parity error and transmit the error flag and the external card retransmits the data. If the number of receiving re-transferred data from the external card is same to the value saved in the RNR but error is not corrected, this PE bit is set to logic 1. This bit is reset to logic 0 whenever the CPU reads the contents of the Line Status Register. In the FIFO mode, this error is associated with the particular character in the FIFO it applies to. This error is revealed to the CPU when its associated character is at the top of the FIFO. Note: Bits 2-1 is the error conditions that produce a Receiver Line Status interrupt whenever any of the corresponding conditions are detected and the interrupt is enabled.
1	R	This bit is the Overrun Error (OE) indicator. Bit 1 indicates that data in the Receiver Buffer Register was not read by the CPU before the next character was transferred into the Receiver Buffer Register, thereby destroying the previous character. The OE indicator is set to logic 1 upon detection of an overrun condition and reset whenever the CPU reads the contents of the Line Status Register. If the FIFO mode data continues to fill the FIFO beyond the trigger level, an overrun error will occur only after the FIFO is full and the next character has been completely received in the shift register. OE is indicated to the CPU as soon as it happens. The character in the shift register is overwritten, but it is not transferred to the FIFO.
0	R	This bit is the receiver Data Ready (DR) indicator. Bit 0 is set to logic 1 whenever a complete incoming character has been received and transferred into the Receiver Buffer Register or the FIFO. Bit 0 is reset to logic 0 by reading all of the data in the Receiver Buffer Register or the FIFO.

Some bits in LSR are automatically cleared when CPU reads the LSR register, so interrupt handling routine should be written that if once reads LSR, then keep the value through entire the routine because second reading LSR returns just reset value.

9.6.2.11 SSR (Smart Card Status Register)

This register provides the additional state of the Smart Card interface to the CPU. In addition to this current-state information, three bits of the Smart Card Status Register provide interrupt information except Tx /Rx data interrupt (these information is in the LSR). These bits are set to logic 1 whenever a interrupt condition occurs e. They are reset to logic 0 whenever the CPU reads the MODEM Status Register.
SCIxBase+0x18

7	6	5	4	3	2	1	0
-	-	-	-	RETRANS_TO	WAITTIMEOUT	INITFAIL	CARDPRE

Bits	Type	Function
7		This bit is reserved at Smart Card Interface mode
6		This bit is reserved at Smart Card Interface mode
5		This bit is reserved at Smart Card Interface mode
4		This bit is reserved at Smart Card Interface mode
3		This bit indicate the retransmit of error data is timeout when RETRANEN(bit 6 of SMR) is set to 1. This bit is set to '1' in the case that the interval between start leading edge of the retransmitted data frame sent by the external card and the start leading edge of previous error data frame (sent by the card but parity error is detected by SCI) exceeds the waiting time value of WTR register. This bit is reset to '0' whenever the CPU reads the contents of the Smart Card Status Register
2		This bit indicates that the waiting time out is occurs. This bit is set to '1' in the case that the interval between start leading edge of the data frame sent by the external card and the start leading edge of previous data frame (sent either by the card or by the interface device) exceeds the waiting time value of WTR register. This bit is reset to '0' whenever the CPU reads the contents of the Smart Card Status Register
1		This bit is set to '1' when the initialization sequence is fail and the ATR from the external card is not received. As soon as this bit is set to '1' and card initialization is failed, the interface device starts the contact deactivation sequence and the CARDINIT bit of SMR is reset to '0' automatically. This bit is reset to '0' whenever the CPU reads the contents of the Smart Card Status Register
0	-	This bit is set to '1' when the external card is inserted and CardPresent pin is logic '1' This bit is reset to '1' when the external card is removed and CardPresent pin is logic '0' The change of this bit or CardPresent pin triggers the CARDDET interrupt

9.6.2.12 SCR

This 8-bit Read/Write Register does not control the UART/Smart Card Interface in any way. It is intended as a scratchpad register to be used by the programmer to hold data temporarily.
SCIxBase+0x1C

7	6	5	4	3	2	1	0
DATA							

Bits	Type	Function
7:0	R/W	Temporary data storage

9.6.2.13 RTR (Reset Timing Register)

On starting of contact activation sequence, the CRST remain to logic '0' waiting ATR until the number of clock cycle set in the RTR register. If the ATR is not received until that number of the clock cycle, CRST is set to logic '1' and waits for ATR during the number of clock cycle set in the RTR once more. If There is no ATR and the clock cycle elapses(the initialization of Smart Card fails) ,the contact deactivation start and the CRST is set to logic '0' The minimum value of this register is 200,so this register must be set greater than 200.

SCIxBase+0x20

16	15	...				1	0
Clock Cycle Number							

Bits	Type	Function
15:0	R/W	The clock(CCLK) cycle number that is used to count the clock number during which the interface device waits for ATR.

9.6.2.14 RNR (Retransmit Number Register)

This register value identifies the number of retransmission before Tx/Rx Line Status interrupt is activated and Line Status error occurs. The Tx/Rx Line Status interrupt occurs if the line status error is not cleared after the re-transmission of the times that is saved in this register.

If the value of this register is set to '0', no error flag is transmitted even though the Smart Card interface receives the error-ed data frame and Rx Line error status interrupt occurs immediately. If the interface device is transmit mode and receives the error flag, the interface device does not re-transmit the error-ed data frame and activates the Tx Line Status error interrupt immediately.

SCIxBase+0x24

7	6	5	4	3	2	1	0
Re-transmission Number							

Bits	Type	Function
7:0	R/W	Retransmission number of errored data, before Tx/Rx Line Status interrupt occurs.

9.6.2.15 WTR (Waiting Time Register)

In the case that the interval between start leading edge of the data frame sent by the external card and the start leading edge of previous data frame (sent either by the card or by the interface device) exceeds the waiting time value of WTR register, the Waiting Timeout interrupt occurs

SCIxBase+0x28

23	22	…				1	0
The number of data bit period							

Bits	Type	Function
23:0	R/W	Waiting Timeout value that is number of 1-bit data period

9.6.2.16 EGR (Extra Guard-Time Register)

This register value set the number of bit –period that follows the 12-bit data frame, and from 0 to 254. If EGR value is 255, the minimum delay between the start edges of two consecutive data frame is reduce to 11-bit period.

SCIxBase+0x2C

23	22	…				1	0
The number of data bit period							

Bits	Type	Function
23:0	R/W	Extra Guard-Time that follow the 12-bit character data frame

9.6.2.17 UCR (UART Configuration Register)

To make the Smart Card Interface mode set, SMCARDEN and UARTEN are set to '1' at the same time.

UxBase+0x30

7	6	5	4	3	2	1	0
-	-	SMCARDEN	CLOCKSEL	SIR Loop Back Uart4 only	Full Duplex Force Uart4 only	SIREN Uart4 only	UARTEN

Bits	Type	Function
7:6	-	Reserved
5	R/W	Smart Card Interface mode set 0 = Smart Card interface disable 1 = Smart Card interface enable (If you use Smart Card Interface function, you must set this bit with UARTEn bit at the same time).
4	R/W	Clock Select 0 = 3.6864MHz 1 = 3.5712MHz
3	R/W	SIR Loop-back Test (Uart1 only) 0 = SIR Loop-back Test disable 1 = SIR Loop-back Test enable.
2	R/W	SIR Full-duplex Force (Uart1 only) 0 = Half Duplex. 1 = Full Duplex.
1	R/W	SIR Enable (Uart1 only) 0 = SIR Mode disable 1 = SIR Mode enable (If you use SIR function, you must set this bit with UARTEn bit at the same time).
0	R/W	UART Enable. 0 = UART disable (Power-Down), UART Clock stop. 1 = UART enable.

9.6.3 Smart Card Interface Operation Flow Chart

Before transmitting or receiving data, the smart card interface and Smart Card must be initialized as described in figure 9-4, after performing Contact initialization and ATR receiving, the configuration of Smart Card Interface must be change to meet the condition of ATR as describe in figure 9-5.

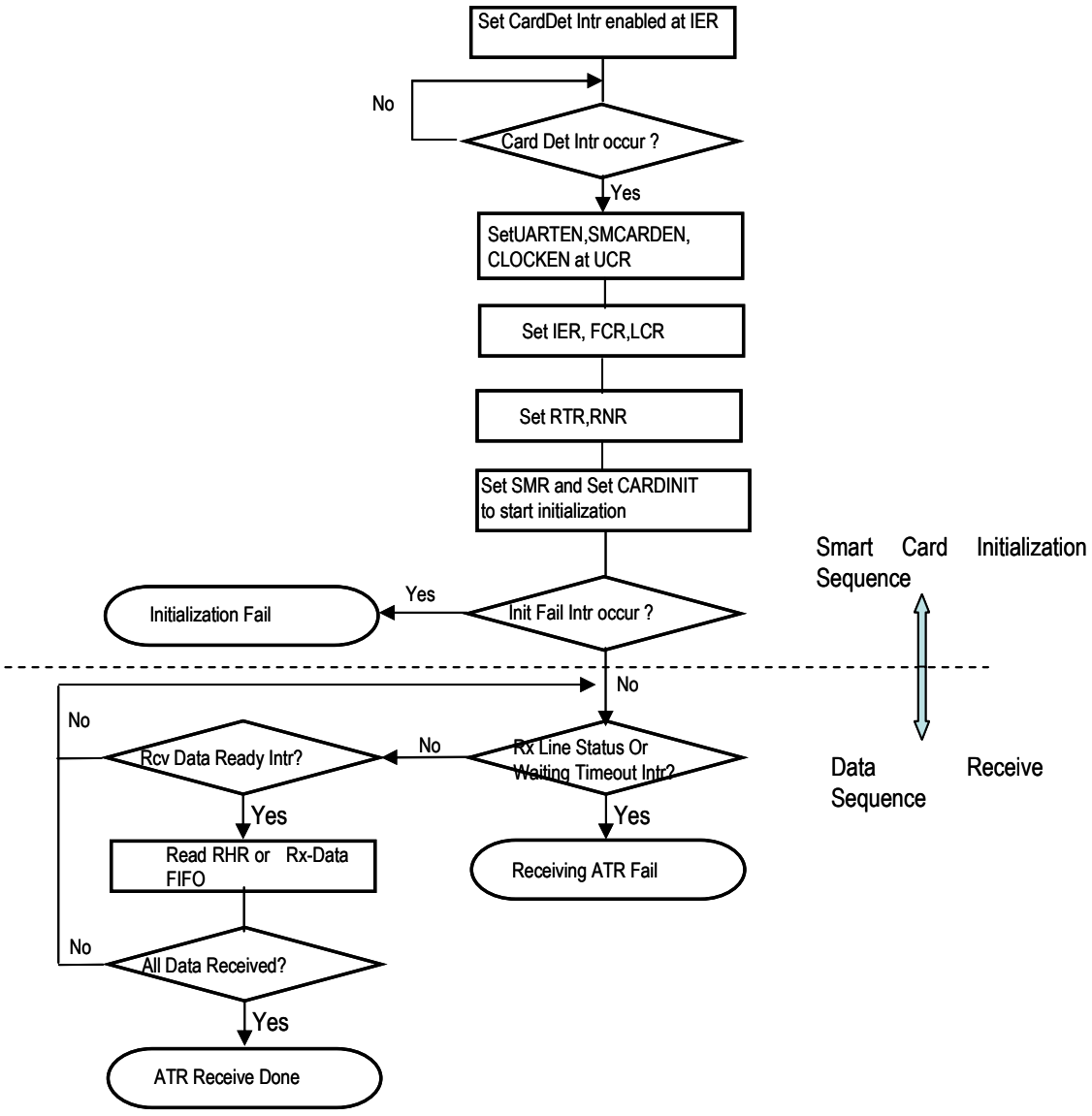


Figure 9-2 Card Initialization and Receiving ATR Flow Chart

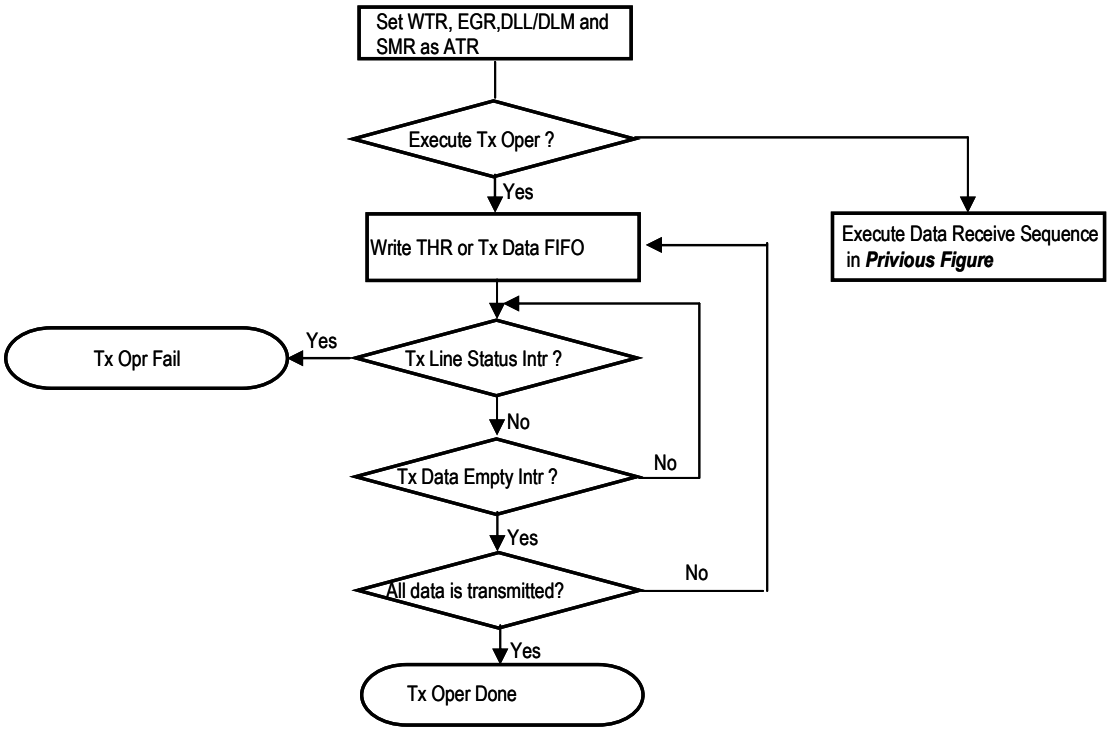


Figure 9-3 Data Transmission and Reception Flow Chart

9.7 Synchronous Serial Interface (SSI)

The HMS30C7210 includes two SSIs (Synchronous Serial Interface) that are AMBA slave blocks connecting to the APB. The SSI is a master or slave interface that enables synchronous serial communication with an external slave or master peripheral. The SSI only supports a Motorola SPI-compatible interface that features full-duplex, three-wire synchronous transfers and programmable clock polarity and phase. In both master and slave configurations, the SSI performs parallel-to-serial conversion on data written to a 8-bit wide, 8-location deep transmit FIFO and serial-to-parallel conversion on received data, buffering it in a 8-bit wide, 8-location deep receive FIFO. Figure 9-23 shows a block diagram of the SSI.

FEATURES

- Master or slave operation
- Motorola SPI-compatible synchronous serial interface
- Programmable transfer clock bit rate, clock polarity and phase
- Separate transmit and receive FIFO buffers, 8 bits wide, 8 locations deep
- 8-bit data frame size
- Full-duplex, 3-wire synchronous transfers
- Independent masking of transmit FIFO, receive FIFO and receive overrun interrupts
- Internal loop-back test mode available

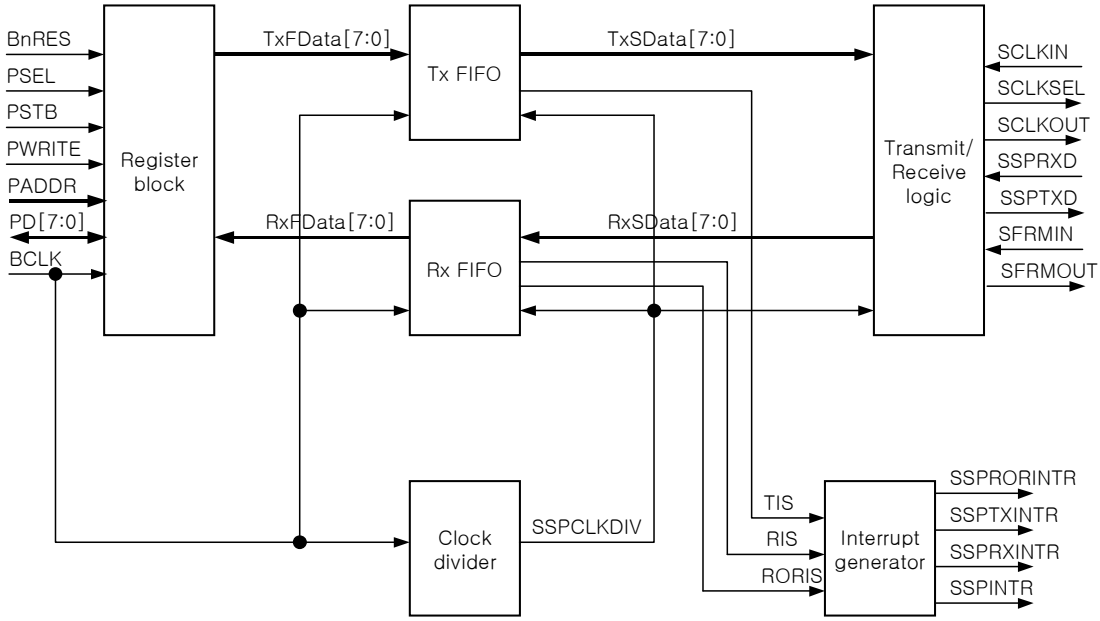


Figure 9-23. SSI Block Diagram

9.7.1 Register description

The SSIBASE is 0x8005A000 for SSI0 and 0x8005B000 for SSI1.

Note Marked '-' bits in the following tables are reserved bits and return zeros on reads.

9.7.1.1 SSPCR0 (control register 0)

SSIBASE + 0x00 (initial value 8'bxxx0_0000)

7	6	5	4	3	2	1	0
-	-	-	GSEL	SDIR	SPH	SPO	MS

Bits	Type	Function
4	R/W	nSFRMIN/OUT select from GPIO 0 : nSFRMIN = 0 (SDIR=0), GPIO pin =nSFRMOUT (SDIR=1) 1 : nSFRMIN = GPIO pin (SDIR=0), GPIO pin = nSFRMOUT (SDIR=1)
3	R/W	SCLKIN/OUT nSFRMIN/OUT direction The reset value of SDIR bit is zero (input). If MS bit is used to indicate the direction instead of SDIR, bus conflicts may occur. 0 = input (SCLKIN, nSFRMIN input) 1 = output (SCLKOUT, nSFRMOUT output)
2	R/W	SCLKIN input phase (MS=1) and/or SCLKOUT output phase (MS=0) 0 = SCLKIN/OUT starts toggling at the middle of the data transfer. 1 = SCLKIN/OUT start toggling at the beginning of the data transfer.
1	R/W	SCLKIN input polarity (MS=1) and/or SCLKOUT output polarity (MS=0) 0 = The inactive state of SCLKIN/OUT is LOW. 1 = The inactive state of SCLKIN/OUT is HIGH.
0	R/W	Master or Slave select 0 = Configured as a master 1 = Configured as a slave

9.7.1.2 SSPCR1 (Control Register 1)

SSIBASE + 0x04 (initial value 8'bxxxx_0000)

7	6	5	4	3	2	1	0
-	-	-	-	SSE	RORIE	TIE	RIE

Bits	Type	Function
3	R/W	SSE : SSI Enable 0 = SSI disabled 1 = SSI enabled
2	R/W	RORIE : Rx FIFO Over-Run Interrupt Enable 0 = Receive over-run interrupt disabled Writing '0' to this bit will also clear RORIS bit in SSPICR 1 = Receive over-run interrupt enabled
1	R/W	TIE : Tx FIFO Interrupt Enable 0 = Tx FIFO interrupt disabled 1 = Tx FIFO interrupt enabled
0	R/W	RIE : Rx FIFO Interrupt Enable 0 = Rx FIFO interrupt disabled 1 = Rx FIFO interrupt enabled

9.7.1.3 SSPDR (Data Register)

SSPDR is the data register and is 8-bit wide. When SSPDR is read, the entry in the receive FIFO pointed to by the current FIFO read pointer is accessed. When SSPDR is written to, the entry in the transmit FIFO pointed to by the write pointer is written to.

SSIBASE + 0x08 (initial value 8'bxxxx_xxxx)

7	6	5	4	3	2	1	0
FIFO7	FIFO6	FIFO5	FIFO4	FIFO3	FIFO2	FIFO1	FIFO0

Bits	Type	Function
7:0	R/W	Transmit/Receive FIFO Read – Receive FIFO Write – Transmit FIFO

9.7.1.4 SSPSR (Status Register)

SSIBASE + 0x0c (Read-only register)

7	6	5	4	3	2	1	0
-	-	-	BSY	RFF	RNE	TNF	TFE

Bits	Type	Function
4	R	BSY : SSI Busy 0 = SSI is idle or is transferring MSB (FIFO[7]) 1 = SSI is transferring frame FIFO[6:0], not MSB
3	R	RFF : Receive FIFO Full 0 = Rx FIFO is not full 1 = Rx FIFO is full
2	R	RNE : Receive FIFO Not Empty 0 = Rx FIFO is empty 1 = Rx FIFO is not empty
1	R	TNF : Transmit FIFO Not Full 0 = Tx FIFO is full 1 = Tx FIFO is not full
0	R	TFE : Transmit FIFO Empty 0 = Tx FIFO is not empty 1 = Tx FIFO is empty

9.7.1.5 SSPCSR (Clock Scale Register)

SSPCSR specifies the division factor by which the input BCLK should be internally divided to make SCLKOUT.

SSIBASE + 0x10 (initial value 8'bxxxx_xxx0)

7	6	5	4	3	2	1	0
CSR7	CSR6	CSR5	CSR4	CSR3	CSR2	CSR1	CSR0

Bits	Type	Function
7:0	R/W	Clock divisor scale Should be an even number from 2 to 254 on writes. The least significant bit always returns zero on reads.

9.7.1.6 SSPIIR/SSPICR (Interrupt Status/Clear Register)

SSIBASE+0x14 (initial value 8'bxxxx_x000)

7	6	5	4	3	2	1	0
-	-	-	-	-	RORIS	TIS	RIS

Bits	Type	Function
2	R/W	RORIS : Rx over-run interrupt status/clear register Write 0 – No effect Write 1 – Clears this bit Read 0 – No Rx over-run interrupt state Read 1 – Rx over-run interrupt state Writing 0 to RORIE bit will also clear RORIS bit
1	R/W	TIS : Tx interrupt status/clear register Write 0 – No effect Write 1 – Clears this bit Read 0 – No Tx interrupt state Read 1 – Tx interrupt state
0	R/W	RIS : Rx interrupt status/clear register Write 0 – No effect Write 1 – Clears this bit Read 0 – No Rx interrupt state Read 1 – Rx interrupt state

9.7.1.7 SSPFENT (FIFO Entry number)

SSIBASE + 0x18 (initial value 8'b0000_0000)

7	6	5	4	3	2	1	0
TXENT3	TXENT2	TXENT1	TXENT0	RXENT3	RXENT2	RXENT1	RXENT0

Bits	Type	Function
7:4	R	The number of valid entries in transmit FIFO
3:0	R	The number of valid entries in receive FIFO

9.7.1.8 SSPIENT (FIFO Entry Interrupt number)

SSIBASE + 0x1c (initial value 8'b0100_0100)

7	6	5	4	3	2	1	0
TXIENT3	TXIENT2	TXIENT1	TXIENT0	RXIENT3	RXIENT2	RXIENT1	RXIENT0

Bits	Type	Function
7:4	R/W	This register is reset to 0x4 and enables programmers to specify the number at which TIS is set. 0xf – 0x9 : TIS is never set. 0x8 : TIS is always set 0x7 – 0x0 : TIS is set when TXENT <= TXIENT TIS is not set when TXENT > TXIENT
3:0	R/W	This register is reset to 0x4 and enables programmers to specify the number at which RIS is set. 0xf – 0x9 : RIS is never set. 0x8 – 0x1 : RIS is set when RXENT >= RXIENT RIS is not set when RXENT < RXIENT 0x0 : RIS is always set

9.7.1.9 SSPTCER (Test Clock Enable Register)

SSIBASE + 0x40-0x7c (initial value 8'b0000_0000)

7	6	5	4	3	2	1	0
TCE7	TCE6	TCE5	TCE4	TCE3	TCE2	TCE1	TCE0

Bits	Type	Function
7:0	R/W	Test Clock Enable. Actually 0-bit register Write : When in registered clock mode, a test clock enable is produced only when this register is accessed Read : When in registered clock mode, a test clock enable is produced only when this register is accessed Returned value is always 8'b0000_0000

SSPTCER has a multiple word space in the register address map to allow for the generation of multiple test clock enable pulses.

9.7.1.10 SSPTCR (Test Control Register)

SSIBASE + 0x80 (initial value 8'bxxx0_0000)

7	6	5	4	3	2	1	0
-	-	-	TINPSEL	TRESET	REGCLK	TCLKEN	TESTEN

Bits	Type	Function
4	R/W	TINPSEL : Test Input Select 0 = Normal input is selected 1 = Values from SSPTISR is multiplexed into input
3		TRESET : Test Reset 0 = No test reset 1 = nSSPRST is asserted throughout the SSI except for test registers
2		REGCLK : Registered mode clock See table below.
1	R/W	TCLKEN : Test Clock Enable See table below.
0	R/W	TESTEN : Test Mode Enable 0 = Normal operating mode is selected 1 = Test mode is selected See table below.

REGCLK	TCLKEN	TESTEN	SCLKIN/OUT	BCLK
1	1	1	Registered clock	Registered clock
1	0	1	Registered clock	BCLK
0	1	1	Divided clock	Registered clock
0	0	1	Strobe clock	BCLK
X	X	0	Divided clock	BCLK

Registered clock : generates a test clock enable on an APB access only to the SSPTCER
Strobe clock : generates a test clock enable on every AMBA APB access to the block
Divided clock : generates a normal mode SCLKIN/OUT by dividing BCLK

9.7.1.11 SSPTMR (Test Mode Register)

SSIBASE + 0x84 (initial value 8'bxxxx_xx00)

7	6	5	4	3	2	1	0
-	-	-	-	-	-	NIBMODE	LBM

Bits	Type	Function
1	R/W	Nibble Mode Counter 0 = Normal CSR counter (CSC) mode 1 = 7-bit CSR counter is partitioned into two nibbles (3-bit, 4-bit) and decrements by 0x11 on successive clocks
0	R/W	Loop Back Mode 0 = Normal serial port operation 1 = Output of transmit serial shifter is connected to input of receive serial shifter internally

9.7.1.12 SSPTISR (Test Input Stimulus Register)

SSPTISR provides test mode stimulus for the SCLKIN and SCLKIN input to the SSI. When TINPSEL bit in the SSPTCR register is 1, the values in the SSPTISR are routed to the internal lines.

SSIBASE + 0x88 (initial value 8'bxxxx_xxxx)

7	6	5	4	3	2	1	0
-	-	-	-	-	nTSFRMIN	TSCLKIN	TSSPRXD

Bits	Type	Function
2	R/W	Test nSFRMIN input for nSFRMIN pin
1	R/W	Test SCLKIN input for SCLKIN pin
0	R/W	Test SSPRXD input for SSPRXD pin

9.7.1.13 SSPTOCR (Test Output Capture Register)

SSIBASE + 0x8C (initial value 8'bx000_0010)

7	6	5	4	3	2	1	0
-	RORINTR	TXINTR	RXINTR	INTR	SSPTXD	nSFRMOUT	SCLKOUT

Bits	Type	Function
6	R	RORINTR : returns the status of SSPRORINTR SSPRORINTR is generated by RORIS ANDed with RORIE 0 = SSPRORINTR pin is driven to logic 0 1 = SSPRORINTR pin is driven to logic 1
5	R	TXINTR : returns the status of SSPTXINTR SSPTXINTR is generated by TIS ANDed with TIE 0 = SSPTXINTR pin is driven to logic 0 1 = SSPTXINTR pin is driven to logic 1
4	R	RXINTR : returns the status of SSPRXINTR SSPRXINTR is generated by RIS ANDed with RIE 0 = SSPRXINTR pin is driven to logic 0 1 = SSPRXINTR pin is driven to logic 1
3	R	INTR : returns the status of SSPINTR 0 = SSPINTR pin is driven to logic 0 1 = SSPINTR pin is driven to logic 1
2	R	SSPTXD : returns the status of SSPTXD 0 = SSPTXD pin is driven to logic 0 1 = SSPTXD pin is driven to logic 1
1	R	nSFRMOUT : returns the status of nSFRMOUT 0 = nSFRMOUT pin is driven to logic 0 1 = nSFRMOUT pin is driven to logic 1
0	R	SCLKOUT : returns the status of SCLKOUT 0 = SCLKOUT pin is driven to logic 0 1 = SCLKOUT pin is driven to logic 1

9.7.1.14 SSPTCCR (Test Clock Counter Register)

This register provides observation for the clock scale counter. The counter is 7-bit, free-running, down counter that operates on BCLK, in normal mode of operation. It can be configured as two nibbles and decremented by test clocks in test mode through SSPTMR and SSPTCR registers. The seven most significant bits programmed in the 8-bit SSPCSR register form the reload value for this counter. The counter reloads when it reaches 0x01.

SSIBASE + 0x90 (initial value 8'bx000_0001)

7	6	5	4	3	2	1	0
-	CSC6	CSC5	CSC4	CSC3	CSC2	CSC1	CSC0

Bits	Type	Function
6:0	R	This bits return the current count of the clock scale counter

9.7.2 Overview

The SSI performs parallel-to-serial conversion on data to transmit to an external device and serial-to-parallel conversion on data to receive from an external device. The transmit and receive paths are buffered with internal FIFO memories allowing up to eight 8-bit values to be stored independently.

The SSI includes a programmable bit rate clock divider to generate the serial output clock SCLKOUT from the bus clock BCLK when configured as a master. The frequency of BCLK is 30MHz (FCLK/2) and it is divided, through the SSPCSR register, by a factor of from 2 to 254 in steps of two. When configured as a slave, the SCLKIN clock is provided by an external master and used to time its transmission and reception sequences.

There are four interrupts generated by the SSI and three of these are individual, maskable, active HIGH interrupts:

SSPTXINTR : active when the number of valid entries in the transmit FIFO is equal to or less than the predetermined number specified by RXIENT.
SSPRXINTR : active when the number of valid entries in the receive FIFO is equal to or more than the predetermined number specified by TXIENT.
SSPRORINTR : active when the receive FIFO is already full and an additional data frame is received.

Above three individual interrupts are also combined into a single output interrupt signal (SSPINTR). The combined SSPINTR is asserted if any of the three individual interrupts are asserted and enabled.

There are registers and logic for functional block verification, and manufacturing or production test using TIC vectors. Test registers should not be read or written to during normal use.

9.7.3 Operational Description

The SSI is reset by nSSPRST and it is generated by the global reset signal BnRES or the test reset signal in SSI test mode. An external reset controller must use BnRES to reset the whole SSI including test logic. The test reset signal resets SSI registers except for test mode registers.

Following the reset, the SSI is disabled and should be configured in this state. Control register SSPCR0 need to be programmed to decide several operation parameters. GSEL bit determines whether nSFRMIN signal from the GPIO is used in slave mode. If GSEL bit is cleared, the SSI regards nSFRMIN signal as zero and transfers are synchronized only with SCLKIN clock signal. If GSEL bit is set, nSFRMIN signal from a GPIO pin is used to indicate valid SCLKIN period and transfers are synchronized with SCLKIN when nSFRMIN is zero. In master mode, GSEL bit has no effects and nSFRMOUT signal to a GPIO pin is always valid. SDIR bit is used to determine the direction of nSFRMIN/OUT and SCLKIN/OUT pins in the GPIO. When SDIR bit is set, the direction is output and nSFRMOUT and SCLKOUT signals go out through GPIO pins. MS bit configures the SSI as a master or slave and SPH and SPO bits determine clock phase and polarity respectively.

When master, the bit rate requires the programming of the clock scale register SSPCSR. The SSPCR1 has SSI enable (SSE) and interrupts enable bits. When disabled in master mode, SCLKOUT is forced to LOW (SPO=0) or HIGH (SPO=1), nSFRMOUT to HIGH, and SSPTXD to LOW. When disabled in slave mode, SCLKIN, nSFRMIN and SSPRXD has no meanings and SSPTXD is set to LOW. Once enabled, transmission and reception of data begins on transmit (SSPTXD) and receive (SSPRXD) pins.

NOTE : When nSFRMIN/OUT signal from/to a GPIO pin is not connected, SDIR and SPO bits in a master should be configured before a slave is enable. Otherwise, the transition of SCLKOUT generated by setting CDIR and/or SPO in the master may cause the slave into malfunctioning. In this case, the recommended sequence of register setup is following. SSPCR0 register in a master should be configured first. Then SSPCR0 in a slave is set and a slave SSI is enabled. The master is enabled last.

Once the bottom entry of the transmit FIFO in a master contains data, nSFRMOUT is active to LOW to indicate valid data frame and the MSB of the 8-bit data frame is shifted out onto the SSPTXD pin. Then, SCLKOUT pin starts running and the serial data bit through SSPRXD is captured in the receive FIFO. After the LSB of the current data frame is shifted out, if there is no more valid entry in the transmit FIFO, SCLKOUT stops toggling and nSFRMOUT is inactive to indicate the completion of the transfer. Otherwise, any valid entries in the transmit FIFO enables another data frame transfer to be continued without delay. Figure 9-7. shows the frame format for a single frame and Figure 9-8. shows the timing diagram when back to back frames are transmitted.

If the receive FIFO is already full and the transmit FIFO is not empty in master mode, a transfer will start but this transfer will cause receive overrun interrupt condition. In this case, a transmit data frame is read from the transmit FIFO and transferred, and a received data frame is overwritten in the receive serial shift buffer normally. But, data in the receive serial buffer will not be stored in the receive FIFO, if the receive FIFO is still full until this transfer finishes. If RORIE bit is set for the receive overrun condition, SSPRORINTR will signal and further data frame will not start until RORIS bit is cleared. In case of slave mode, the operation is the same except that a data frame starts with SCLKIN from external device.

If the transmit FIFO is already empty and another data frame is request in slave mode, a transmit FIFO underrun condition occurs. The receive FIFO operates normally but

transmit FIFO transfers the same data frame as in the previous transfer. This condition cannot occur in master mode. In this version of SSI, there is not an assigned interrupt for this case.

If CPU writes data to the transmit FIFO that is already full, the valid entries (from the oldest entry that was written) in the FIFO can be overwritten. To detect this erroneous state, TXENT bits can be read. If TXENT[3:0] is in the range of from 0x9 to 0xf, the number of lost entries is TXENT - 0x8.

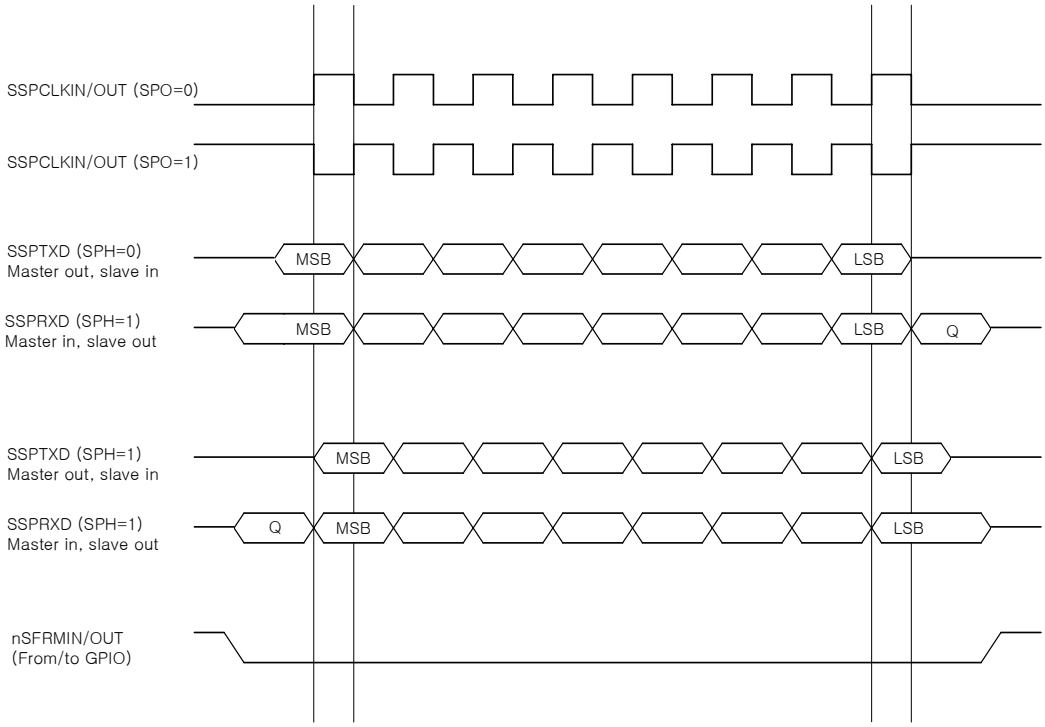


Figure 9-24. Transfer Format (Single Transfer)

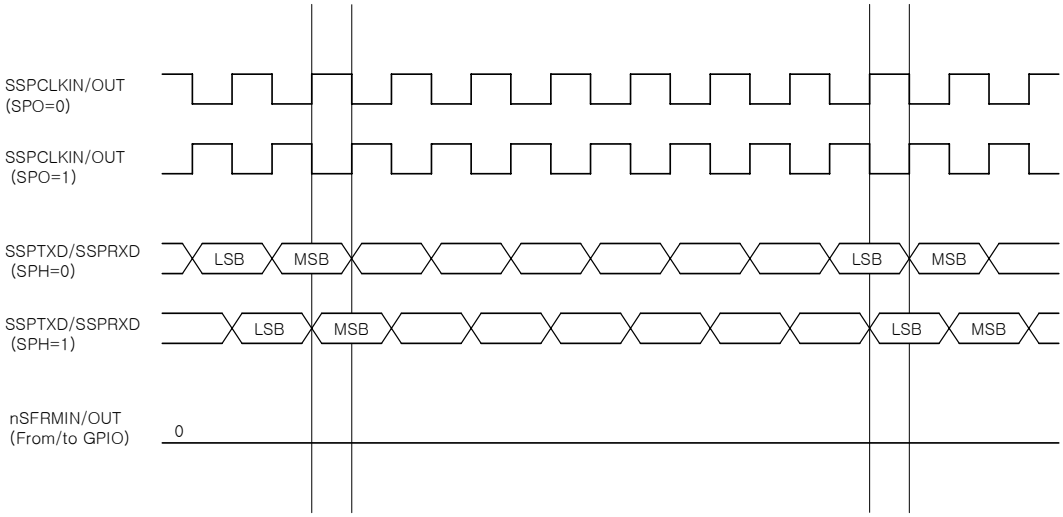
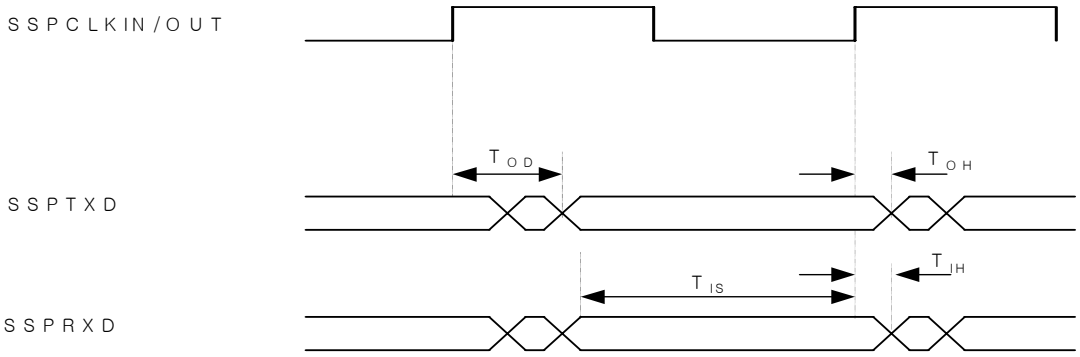


Figure 9-25. Transfer Format (Back to Back Transfer)

If CPU reads data from the receive FIFO that is already empty, invalid entries in the receive FIFO can be read. By reading RXENT bits, this erroneous state can be detected. If RXENT[3:0] is in the range of from 0x9 to 0xf, the number of entries that has been read by mistake is 0x10 – RXENT.

Note
This version of the SSI supports neither multi-master nor multi-slave configurations.

9.7.4 SSI AC Timming



Symbol	Description	Min.	Max
TOD	Output Delay from clock to TXD	-	3ns
TOH	Output Hold time from clock to TXD	1ns	-
TIS	RXD Input Setup Time	3ns	-
TIH	RXD Input Hold Time	0.5ns	-

Figure 9-4 SSI AC Timing

9.8 SMC Controller

This SmartMedia™ Card Controller is an Advanced Microcontroller Bus Architecture (AMBA) compliant System-on-a-Chip peripheral providing an interface to industry-standard SmartMedia™ Flash Memory Card. A channel has 8 control signal outputs and 8 bits of bi-directional data ports.

FEATURES

- One 3.3V SmartMedia support
- 4MB to 128MB media (both Flash and Mask ROM type)
- Interrupt mode support when erase/write operation is finished
- Unique ID SmartMedia support
- Multi-page (up to 32 pages) access (read/write)
- Hardware 3Byte ECC generation & check (software correctable).
- Marginal timing operation settable.

9.8.1 External Signals

Pin Name	Type	Description
SMD [7:0]	I/O	Smart Media Card (SSFDC) 8bit data signals
nSMWP	O	Smart Media Card (SSFDC) write protect
nSMWE	O	Smart Media Card (SSFDC) write enable
SMALE	O	Smart Media Card (SSFDC) address latch enable
SMCLE	O	Smart Media Card (SSFDC) command latch enable
nSMCD	I	Smart Media Card (SSFDC) card detection signal
nSMCE	O	Smart Media Card (SSFDC) chip enable
nSMRE	O	Smart Media Card (SSFDC) read enable
nSMRB	I	Smart Media Card (SSFDC) READY/nBUSY signal. This is open-drain output so it requires a pull-up resistor.

Refer to Figure 2-1. 208 Pin diagram.

9.8.2 Registers

Address	Name	Width	Default	Description
0x8005.C000	SMCCMD	32	0x0	SmartMedia Card Command register
0x8005.C004	SMCADR	27	0x0	SmartMedia Card Address register
0x8005.C008	SMCDATW	32	0x0	Data written to SmartMedia Card
0x8005.C00C	SMCDATR	32	0x0	Data received from SmartMedia Card
0x8005.C010	SMCCONF	8	0x0	SmartMedia Card controller configuration register
0x8005.C014	SMCTIME	20	0x0	Timing parameter register
0x8005.C01C	SMCSTAT	32	0x0	SmartMedia Card controller status register
0x8005.C024	SMCECC1	24	0x0	ECC register for first half page data
0x8005.C028	SMCECC2	24	0x0	ECC register for second half page data
0x8005.C02C	SMCMRW	12	0x0	Multi-page read/write configuration register
0x8005.C030	SMCMSTAT	12	0x0	Multi-page read/write status register
0x8005.C034	**SMCEBICON**	**3**	**0x0**	**SMC control register using EBI interface**

Table 9-12 SmartMedia Controller Register Summary

9.8.2.1 SMC Command Register (SMCCMD)

0x8005.C000

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

Bits	Type	Function
31:24	R/W	Hidden Command 0. This Unique ID feature will be available to 128Mb NAND Flash and upward density products to prevent illegal copy of music files. Unique ID is put into redundant block of SmartMedia. Use this hidden command to access redundant block that cannot be accessed with open command, This byte filed is ignored when user block is accessed. For more information, refer to SmartMedia Maker's datasheet.
23:16	R/W	Hidden Command 1. Read ID command returns whether the SmartMedia card supports unique ID or not. Hidden 2 step command for Samsung is 30h-65h and for Toshiba is 5Ah-B5h. To return back to user block after accessing redundant block area, Reset command (FFh) should be carried out.
15:8	R/W	There are 9 commands to operate SmartMedia card. This controller supports only parts of them (bold type). Set 1ST command into this byte field except writing to SmartMedia. For write operation, set this byte field to Serial Data Input (80h) and set Second Command byte field to Page Program (10h).

Function	1ST cycle	2ND cycle	Function	1ST cycle	2ND cycle
Serial Data Input	80h		Page Program	10h	
Read 0	00h		Block Erase	60h	D0h
Read 1	01h		Status Read	70h	
Read 2	50h		ID Read	90h	
Reset	FFh				

7:0	R/W	Set 2ND command here

9.8.2.2 SMC Address Register (SMCADR)

0x8005.C004

26	25	24	23	22	21	20	19	18	17	16
SMCADR26 ~ SMCADR16										

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMCADR15 ~ SMCADR0															

Bits	Type	Function
26:0	R/W	SMC Address. SMC controller begins to operate after writing an address to SMCADR. Hence a valid command must be set to SMCCMD before writing to SMCADR. However, reset and status read commands activate SMC controller after writing to SMCCMD because they do not require an address. Following table shows valid address range according to SmartMedia card size. MODEL VALID PAGE ADDRESS 4 MB SMCADR0 ~ SMCADR21 8 MB SMCADR0 ~ SMCADR22 16 MB SMCADR0 ~ SMCADR23 32 MB SMCADR0 ~ SMCADR24 64 MB SMCADR0 ~ SMCADR25 128 MB SMCADR0 ~ SMCADR26

9.8.2.3 SMC Data Write Register (SMCDATW)

0x8005.C008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
N * (SMCADR + 3)'s Byte Data								N * (SMCADR + 2)'s Byte Data							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
N * (SMCADR + 1)'s Byte Data								N * SMCADR's Byte Data							

Bits	Type	Function
31:0	R/W	Four byte data written to this register will be sent to SmartMedia. SMC controller receives a 32bit data from host controller. Then It starts to transmit from least significant byte to most significant byte, one byte at a time. This SMC controller writes a whole page at a single write transaction, so it requires 132 times consecutive writing (528 = 512+16 bytes). A page program process is as follows: Set SMCCMD to xxxx8010h (Sequential Data Input + Page Program), SMCADR to desired target page address space, and then write first 4 byte data onto SMCDATW. In normal mode, interrupt will be generated every 4 bytes write. At the end of sequential data input, SmartMedia goes into page program mode by transmitting the second command to SmartMedia. Usually page program takes long time, no polling status register is recommended. SMC controller automatically generates write finish interrupt when SmartMedia comes back to ready mode.

9.8.2.4 SMC Data Read Register (SMCDATR)

0x8005.C00C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
N * (SMCADR + 3)'s Byte Data								N * (SMCADR + 2)'s Byte Data							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
N * (SMCADR + 1)'s Byte Data								N * SMCADR's Byte Data							

Bits	Type	Function
31:0	R	Four byte data read from SmartMedia is stored in this register. SMC controller receives a byte data from SmartMedia and stores it into 4 byte internal buffer to create 32bit data. First read byte data is stored at least significant byte and fourth byte data is stored at most significant byte of buffer. Host controller reads this register to get 4 byte data at a time. This SMC controller reads a whole page at a single read transaction, so it requires 132 times consecutive reading. A page reading process is as follows: Set SMCCMD to xxxx00yyh (xxxx can be unique ID if redundant area accessed, yy is don't care. Only 00h command is valid. No 01h or 50h command supported) and then set SMCADR to target page address. SMC controller will access SmartMedia with given command and address. Interrupt will be generated after first four byte read. Like writing process, reading process reads a whole 528 byte in a page at a single transaction, so interrupt will be 132 times. Against to write operation, there is no read finish interrupt because we can count the number of read transfers in software or can get the total access word size from BYTE COUNT of SMCSTAT.

9.8.2.5 SMC Configuration Register (SMCCONF)

0x8005.C010

31	30	29	28	27	26	25	24
POWER ENABLE	-	-	-	-	-	-	-
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
-	-	-	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
-	-	-	-	-	MULTI-PAGE WRITE ENALBE	MULTI-PAGE READ ENALBE	WRITE ECC ENABLE
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Read ECC ENABLE	SAFE MARGIN	SMC ENABLE	-	INTR EN	-	UNIQUE ID EN	BIG CARD ENABLE

Bits	Type	Function
31	R/W	Power on bit. To activate SMC controller, set this bit. Reset will fall the controller into the deep sleep mode.
30:11	-	Reserved. Keep these bits to zero.
10	R/W	Multi-page write enable bit. When this bit set, data can be stored in SMC continuously up to 32 pages. While the single page write requires write command and address for each operation, it does not necessary write command and address for each page.
9	R/W	Multi-page read enable bit. When this bit set, data stored in SMC can be read continuously up to 32 pages. While the single page read requires read command and address for each operation, it does not necessary read command and address for each page.
8	R/W	ECC write enable bit. When this bit set, 3 Byte ECC code (specified in SSFDC standard) is generated in ECC block and written to SmartMedia.
7	R/W	ECC read & check enable bit. When this bit set, 3 Byte ECC code is read out from SmartMedia and compared with regenerated ECC code, for which the data read out from SmartMedia is used. The result is returned to a host when a host reads ECC area in redundant area.
6	R/W	Safe margin enable bit. In normal mode, chip select signal changes simultaneously with read enable and write enable signals. But when this bit set, the duration of read and write enable signal applied to SmartMedia is reduced by 1 automatically. By enabling this, the rising edge of read and write enable signal will be earlier than the rising edge of chip enable, which guarantees latching data safely.
5	R/W	SMC controller enable bit. Reset this bit will make SMC controller stay in standby mode. No interrupt generated, no action occurred.
4	-	Reserved. Keep these bits to zero.
3	R/W	Interrupt enable. After reading a word or before writing a word, the interrupt bit of SMCSTAT will be set and interrupt will occur if INTR EN is enabled. If this bit is disabled, software must poll the interrupt flag of SMCSTAT to know the occurrence of an interrupt. After writing a whole page (or pages when CONT PAGE EN is enabled) to SmartMedia, write finish interrupt will also be generated to notice that the SmartMedia complete the write operation successfully.
2	-	Reserved. Keep these bits to zero.
1	R/W	Redundant page enable. When use SmartMedia with unique ID and want to access redundant page area, set high. This bit cannot be cleared automatically, so in order to read open page area clear this bit and set a reset command to SMCCMD.
0	R/W	Larger than 32MB SmartMedia support enable. When using 64MB or 128MB SmartMedia, set this bit high.

9.8.2.6 SMC Timing Parameter Register (SMCTIME)

0x8005.C014

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

Bits	Type	Function
31:28	-	Reserved. Keep these bits to zero.
27:24	R/W	Wait counter maximum limit value. Waiting time delay between address latch and write data in page program mode or between address latch and read data in read ID mode and read status register is determined by this register. 0000 = 1 BCLK width 0001 = 2 BCLK width … 1111 = 16 BCLK width
23	-	Reserved
22:16	R/W	Should set these bits as 0x7F to access full 512 bytes page at one access command (read or program).
15:10	-	Reserved
9:8	R/W	High pulse width value of read enable and write enable signal. The width must satisfy the AC characteristics of SmartMedia to guarantee correct transfer of data. With Safety Margin enable, width will be decreased by one. 00 = 1 BCLK width (0 BCLK with safety margin enable. Don't make this case) 01 = 2 BCLK width (1 BCLK with safety margin enable) 10 = 3 BCLK width (2 BCLK with safety margin enable) 11 = 4 BCLK width (3 BCLK with safety margin enable)
7:3	-	Reserved
2:0	R/W	Low pulse width value of read enable and write enable signal. The width must satisfy the AC characteristics of SmartMedia to guarantee correct transfer of data. With Safety Margin enable, width will be decreased by one. 000 = 1 BCLK width (0 BCLK with safety margin enable, Don't make this case) 001 = 2 BCLK width (1 BCLK with safety margin enable) … 111 = 8 BCLK width (7 BCLK with safety margin enable)

9.8.2.7 SMC Status Register (SMCSTAT)

0x8005.C01C

31	30	29	28	27	26	25	24
CD INTR	nSMCE	SMCLE	SMALE	nSMWE	nSMRE	nSMWP	SMR/B
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CURRENT COMMAND/CARD DETECT NOTIFICATION							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
EXTRA AREA	BYTE COUNT						
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INTERNAL STATE				CARD DETECT	IRQ	-	BUSY

Bits	Type	Function
31	R	Card Detect Interrupt. When card inserted or removed, card detect interrupt will be generated. In the interrupt service routine, look at this bit to identify interrupt type.
30:24	R	Current status of output signals.
23:16	R	Current active command. If in card detect interrupt, this byte shows 0xCD.
15	R	Set when extra area of a page is accessed.
14:8	R	Current address of a page in word units.
7:4	R	Shows internal state machine's state.
3	R	Set when SMC enable and SMC card inserted. It will be zero when card removed.
2	R	Interrupt flag
1	-	Reserved
0	R	Reset shows SMC is in idle mode. Set means SMC in working mode.

9.8.2.8 SMC first half page ECC Register (SMCECC1)

It contains generated ECC value of 0~255th byte in an page. Especially, it is used to calculate the error position with ECC data stored in SMC in reading operation.
0x8005.C024

23	22	21	20	19	18	17	16
P4	P4'	P2	P2'	P1	P1'	1	1
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
P1024	P1024'	P512	P512'	P256	P256'	P128	P128'
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
P64	P64'	P32	P32'	P16	P16'	P8	P8'

Bits	Type	Function
31:24	R	Reserved.
23,21,19	R	Bit position vector. It is used to calculate the bit position in the byte having error.
22,20,18	R	Complementary value of Bit position vector.
17	R	Reserved.
16	R	Reserved.
15,13,11,9, 7,5,3,1	R	Byte position vector. It is used to calculated the byte position in the first half page having error.
14,12,10,8, 6,4,2,0	R	Complementary value of Byte position vector.

9.8.2.9 SMC second half page ECC Register (SMCECC2)

It contains generated ECC value of 256~511st byte in an page. Especially, it is used to calculate the error position with ECC data stored in SMC in reading operation.
0x8005.C028

23	22	21	20	19	18	17	16
P4	P4'	P2	P2'	P1	P1'	1	1
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
P1024	P1024'	P512	P512'	P256	P256'	P128	P128'
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
P64	P64'	P32	P32'	P16	P16'	P8	P8'

Bits	Type	Function
31:24	R	Reserved.
23,21,19	R	Bit position vector. It is used to calculate the bit position in the byte having error.
22,20,18	R	Complementary value of Bit position vector.
17	R	Reserved.
16	R	Reserved.
15,13,11,9, 7,5,3,1	R	Byte position vector. It is used to calculated the byte position in the second half page having error.
14,12,10,8, 6,4,2,0	R	Complementary value of Byte position vector.

9.8.2.10 SMC Multi-page Read/Write Configuration Register (SMCMRW)

0x8005.C02C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	PAGE SIZE for WRITE						PAGE SIZE for READ					

Bits	Type	Function
31:12	-	Reserved. Keep these bits to zero.
11:6	R/W	Multi-page WRITE size bit. Maximum 32 pages can be written to SMC with single command and start address. 000000 = no writing. 000001 = 1 pages. 000010 = 2 pages. … 011111 = 31 pages. 100000 = 32 pages
5:0	R/W	Multi-page READ size bit. Maximum 32 pages can be read from SMC with single command and start address. 000000 = no reading. 000001 = 1 pages. 000010 = 2 pages. … 011111 = 31 pages. 100000 = 32 pages.

9.8.2.11 SMC Multi-page Read/Write Status Register (SMCSTAT)

0x8005.C030

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	WRITE Page Count						READ Page Count					

Bits	Type	Function
31:12	-	Reserved. Keep these bits to zero.
11:6	R/W	Current page count in multi-page writing operation. During a page write operation, it is equal to (current page count -1). After full one page (528byte) writing, it becomes ‘current page count’.
5:0	R/W	Current page count in multi-page reading operation. During a page read operation, it is equal to (current page count -1). After full one page (528byte) reading, it becomes ‘current page count’.

9.8.2.12 SMC Control Register using EBI interface (SMCEBICON)

0x8005.C034

7	6	5	4	3	2	1	0
-	-	-	-	-	SMC access select	nSMWP	nSMCE

Bits	Type	Function
31:3	-	Reserved
2	W	SMC access mode select. When this bit set (=1), EBI interface controls SMC. When this bit unset (=0), SMC controller controls SMC.
1	W	nSMWP control for SMC control using EBI interface. When bit [2] is used to set nSMWP of SMC.
0	W	nSMCE control for SMC control using EBI interface. When bit [2] is used to set nSMCE of SMC.

9.8.3 SMC access using EBI interface

HMS30C7210 provides 2 methods to access SMC memory. One is the SMC controller and the other is the SMI controller.

SMC access scheme of the SMC controller in HMS30C7210 is different than that of the SMI controller. If an user want to access the SMC like as the SRAM, SMI controller must be used with the register 'SMCEBICON' (address 0x8005.C034).

The figure below shows the scheme in the HMS30C7210 for the SMC access using the EBI interface (ECC is not supported at this method). The following represents the SMC access method using the EBI interface.

- The bit 2 of the register 'SMCEBICON' must be set to '1'.
- The memory address, which enables the nRCS[3], must be used.
- When the 2 least significant bits of the memory address is equal to '01' (RA[1:0]='01'), the signal SMCLE is set.
- When the 2 least significant bits of the memory address is equal to '10' (RA[1:0]='01'), the signal SMALE is set.
- The bit 1 of the reigster 'SMCEBICON' set the signal nSMWP.
- The bit 0 of the reigster 'SMCEBICON' set the signal nSMCE.

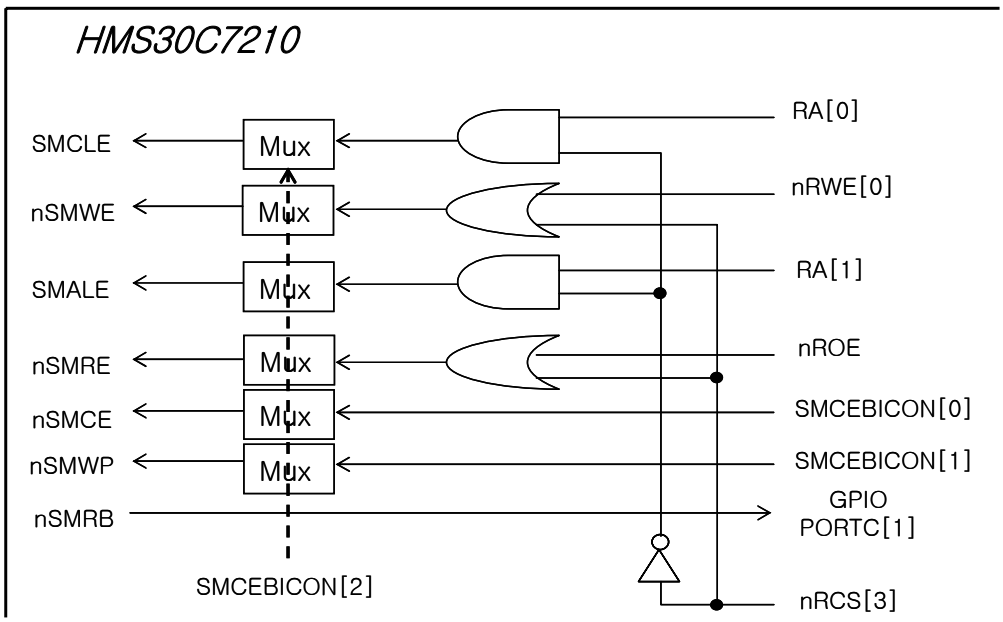


Figure 9-26. SMC access using the EBI Interface

9.9 TIMER & PWM

This module is a 16-bit counter clocked by PCLK. The frequency of PCLK is approximately 3.6923MHz when F_{CCLK} is 48MHz and obtained by the formula $F_{PCLK} = F_{CCLK} / 13$, where F_{PCLK} is the frequency of PCLK and F_{CCLK} is the frequency of CCLK. TIMER/PWM is an AMBA slave module that connects to the Advanced Peripheral Bus (APB). For more information about AMBA, please refer to the AMBA Specification (ARM IHI 0001).

The main features of timer module are :

- 8/16-bit up counter
- Auto repeat mode
- Count enable/disable
- Interrupt enable/disable
- 4-timer channel and 4 timer outputs

The main features of PWM modules are :

- 16-bit up counter
- Count enable/disable
- 2-PWM channel and 2 PWM outputs
- Adjustable PWM output period and duty ratio

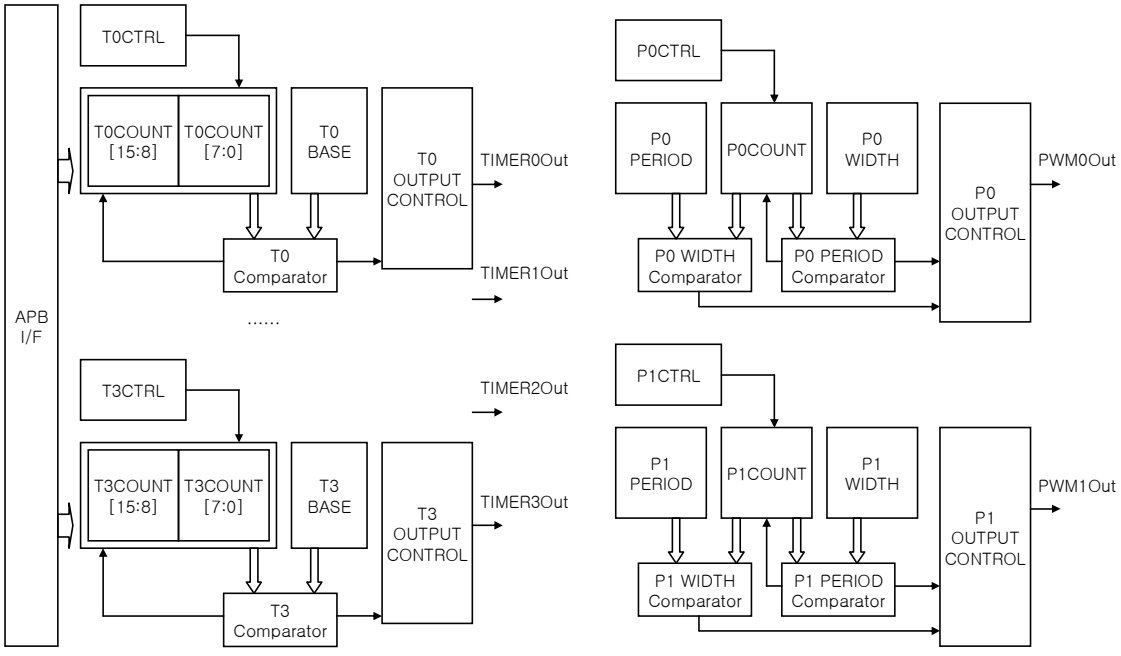


Figure 9-27. Block Diagram of TIMER/PWM

9.9.1 External Signals

Pin Name*	Type	Description
PWM [1:0]	O	The outputs of 2 PWM channels
TIMER[3:0]	O	The outputs of 4 TIMER channels

Refer to Figure 2-1. 208 Pin diagram.

9.9.2 Registers

Address	Name	Width	Default	Description
0x8005.D000	T0BASE	16	0xFFFF	Timer0 Base Register
0x8005.D008	T0COUNT	16	0x0	Timer0 Counter Register
0x8005 D00C	T0STAT	1	0x0	Timer0 Status Register
0x8005.D010	T0CTRL	8	0x0	Timer0 Control Register
0x8005.D020	T1BASE	16	0xFFFF	Timer1 Base Register
0x8005.D028	T1COUNT	16	0x0	Timer1 Counter Register
0x8005 D02C	T1STAT	1	0x0	Timer1 Status Register
0x8005.D030	T1CTRL	8	0x00	Timer1 Control Register
0x8005.D040	T2BASE	16	0xFFFF	Timer2 Base Register
0x8005.D048	T2COUNT	16	0x0	Timer2 Counter Register
0x8005 D04C	T2STAT	1	0x0	Timer2 Status Register
0x8005.D050	T2CTRL	8	0x0	Timer2 Control Register
0x8005.D060	T3BASE	16	0xFFFF	Timer3 Base Register
0x8005 D068	T3COUNT	16	0x0	Timer3 Counter Register
0x8005 D06C	T3STAT	1	0x0	Timer3 Status Register
0x8005 D070	T3CTRL	8	0x0	Timer3 Control Register
0x8005 D080	TOPCTRL	10	0x0	Top-level Control Register
0x8005.D084	TOPSTAT	4	0x0	Top-level Status Register
0x8005.D0A0	P0COUNT	16	0x0	PWM channel 0 count register
0x8005.D0A4	P0WIDTH	16	0xFFFF	PWM channel 0 width register
0x8005.D0A8	P0PERIOD	16	0xFFFF	PWM channel 0 period register
0x8005.D0AC	P0CTRL	8	0x0	PWM channel 0 control register
0x8005.D0C0	P1COUNT	16	0x0	PWM channel 1 count register
0x8005.D0C4	P1WIDTH	16	0xFFFF	PWM channel 1 width register
0x8005.D0C8	P1PERIOD	16	0xFFFF	PWM channel 1 period register
0x8005.D0CC	P1CTRL	8	0x0	PWM channel 1 control register

Table 9-13. Timer Register Summary

9.9.2.1 Timer Top-level Control Register (TOPCTRL)

0x8005.D080

15	14	13	12	11	10	9	8
-	-	-	-	-	-	TIMER3 OUTEN	TIMER2 OUTEN
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TIMER1 OUTEN	TIMER0 OUTEN	TIMER3 CLKSEL	nPOWER DOWN	TIMER3 INTEN	TIMER2 INTEN	TIMER1 INTEN	TIMER0 INTEN

Bits	Type	Function
9	R/W	Timer channel 3 Output Enable Setting this bit enables the output of timer channel 3 to propagate through pin TIMER[3]. Whenever T3COUNT reaches T3BASE, the output of timer channel 3(TIMER[3]) toggles. If a system reset or SOFTRESET in T3CTRL register occurs, the output is reset to '0'. 0 = Output of timer channel 3 is blocked. (default) 1 = Output of timer channel 3 appears on pin TIMER[3].
8	R/W	Timer channel 2 Output Enable Setting this bit enables the output of timer channel 2 to propagate through pin TIMER[2]. Whenever T2COUNT reaches T2BASE, the output of timer channel 2(TIMER[2]) toggles. If a system reset or SOFTRESET in T2CTRL register occurs, the output is reset to '0'. 0 = Output of timer channel 2 is blocked. (default) 1 = Output of timer channel 2 appears on pin TIMER[2].
7	R/W	Timer channel 1 Output Enable Setting this bit enables the output of timer channel 1 to propagate through pin TIMER[1]. Whenever T1COUNT reaches T1BASE, the output of timer channel 1(TIMER[1]) toggles. If a system reset or SOFTRESET in T1CTRL register occurs, the output is reset to '0'. 0 = Output of timer channel 1 is blocked. (default) 1 = Output of timer channel 1 appears on pin TIMER[1].
6	R/W	Timer channel 0 Output Enable Setting this bit enables the output of timer channel 0 to propagate through pin TIMER[0]. Whenever T0COUNT reaches T0BASE, the output of timer channel 0(TIMER[0]) toggles. If a system reset or SOFTRESET in T0CTRL register occurs, the output is reset to '0'. 0 = Output of timer channel 0 is blocked. (default) 1 = Output of timer channel 0 appears on pin TIMER[0].
5	R/W	Timer channel 3 Clock source All counters in timer channel 0,1,2,3 operate in PCLK domain. But timer channel 3 Select the clock source of 16bit Timer 3. (For details, see operation section) 0 = T3COUNT is clocked by PCLK. (default) 1 = T3COUNT is clocked when T2COUNT reaches T2BASE.
4	R/W	Power down mode (Active low) Activates TIMER/PWM module by supplying PCLK. 0 = Indicates power down mode and clock signal(PCLK) is always '0'. (default) 1 = Supply PCLK to TIMER/PWM module (Normal operation mode).
3	R/W	Timer channel 3 interrupt Enable Setting this bit enables generation of interrupt signal from timer channel 3. 0 = No interrupt is requested from timer channel 3. (default) 1 = Interrupt is generated when T3COUNT reaches T3BASE.
2	R/W	Timer channel 2 Interrupt Enable Setting this bit enables generation of interrupt signal from timer channel 2. 0 = No interrupt is requested from timer channel 2. (default) 1 = Interrupt is generated when T2COUNT reaches T2BASE.
1	R/W	Timer channel 1 Interrupt Enable Setting this bit enables generation of interrupt signal from timer channel 1. 0 = No interrupt is requested from timer channel 1. (default) 1 = Interrupt is generated when T1COUNT reaches T1BASE.
0	R/W	Timer channel 0 Interrupt Enable Setting this bit enables generation of interrupt signal from timer channel 0.

	0 = No interrupt is requested from timer channel 0. (default) 1 = Interrupt is generated when T0COUNT reaches T0BASE.

9.9.2.2 Timer Status Register (TOPSTAT)

0x8005.D084

7	6	5	4	3	2	1	0
-	-	-	-	TIMER3 MATCH	TIMER2 MATCH	TIMER1 MATCH	TIMER0 MATCH

Bits	Type	Function
7:4	-	Reserved
3	R	This bit reflect the status of ST bit in T3STAT 0 = MATCH bit in T3STAT is cleared. 1 = MATCH bit In T3STAT is set.
2	R	This bit reflect the status of ST bit in T2STAT 0 = MATCH bit in T2STAT is cleared. 1 = MATCH bit In T2STAT is set.
1	R	This bit reflect the status of ST bit in T1STAT 0 = MATCH bit in T1STAT is cleared. 1 = MATCH bit In T1STAT is set.
0	R	This bit reflect the status of ST bit in T0STAT 0 = MATCH bit in T0STAT is cleared. 1 = MATCH bit In T0STAT is set.

9.9.2.3 Timer [0,1,2,3] Base Register (T[0,1,2,3]BASE)

0x8005.D000 / 0x8005.D020 / 0x8005.D040 / 0x8005 D060

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T[0,1,2,3]BASE [15:0]															

Bits	Type	Function
15:0	R/W	Timer 0 (Timer 1, Timer 2, Timer3) Base Register This register is used to limit the upper boundary of TnCOUNT(n = 0,1,2,3). When TnCOUNT reaches TnBASE, the TnCOUNT is cleared and each timer channel may generate an interrupt. And also the output of each timer channel may toggle. The initial value of TnBASE is 0xFFFF.

9.9.2.4 Timer [0,1,2,3] Count Register (T[0,1,2,3]COUNT)

0x8005.D008 / 0x8005.D028 / 0x8005.D048 / 0x8005 D068

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T[0,1,2,3]COUNT [15:0]															

Bits	Type	Function
15:0	R/W	Timer 0 (Timer1, Timer2, Timer3) Up Counter The clock source of this count is controlled by PRESCALER in TnCTRL(n = 0,1,2,3). TnCOUNT is not loadable. The initial value of TnCOUNT is 0x0000.

9.9.2.5 Timer [0,1,2,3] Control Register (T[0,1,2,3]CTRL)

0x8005.D010 / 0x8005.D030 / 0x8005.D050 / 0x8005 D070

7	6	5	4	3	2	1	0
PRESCALER				BYTE MODE	SOFT RESET	REPEAT MODE	COUNT ENABLE

Bits	Type	Function
7:4	R/W	Counter clock prescaler TnCOUNT is clocked by (PRESCALER + 1)th CLK(n = 0,1,2,3). The symbol CLK represents normally PCLK or the moment when T2COUNT equals T2BASE.
		PRESCALER — Clock source 0000 — CLK (default) 0001 — CLK/2 0010 — CLK/3 0011 — CLK/4 … — … 1110 — CLK/15 1111 — CLK/16
3	R/W	Byte mode. If BYTEMODE is set, each TnCOUNT operates as 8-bit counter and the upper limit of TnCOUNT is 0xFF. 0 = TnCOUNT operates as normal 16-bit counter. (default) 1 = TnCOUNT operates as 8-bit counter and is cleared when it reaches 0xFF.
2	R/W	Software reset command This bit resets TnCOUNT and the output of each timer channel. This bit is not auto-cleared so user should clear this bit after issuing SOFTRESET command. 0 = Normal operation. (default) 1 = Resets TnCOUNT and output of timer channel.
1	R/W	When this bit is set, TnCOUNT repeats the following actions until REPEATMODE is cleared : TnCOUNT increments → reaches TnBASE → clears → increments → … 0 = TnCOUNT stops counting when TnCOUNT reaches TnBASE. (default) 1 = TnCOUNT increments repeatedly while COUNTENABLE in TnCTRL is set.
0	R/W	Counter enable Setting this bit enables TnCOUNT to increment and this bit will be cleared automatically when TnCOUNT reaches TnBASE if REPEATMODE is '0'. 0 = Stops counting. (default) 1 = Starts counting.

9.9.2.6 Timer [0,1,2,3] Status Register (T[0,1,2,3]STAT)

0x8005.D00C / 0x8005.D02C / 0x8005.D04C / 0x8005 D06C

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	MATCH

Bits	Type	Function
7:1	-	Reserved
0	R	TnCOUNT match MATCH bit is set when TnCOUNT equals TnBASE. Writing any value to TnSTAT clears MATCH bit and disables interrupt request when interrupt is pending. 0 = TnSTAT is cleared or TnCOUNT not equals TnBASE. (default) 1 = TnCOUNT reached TnBASE.

9.9.2.7 PWM Channel [0,1] Count Register (P[0,1]COUNT)

0x8005.D0A0 / 0x8005.D0C0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P[0,1]COUNT															

Bits	Type	Function
15:0	R	PWM 0 (PWM 1) up counter The clock source of this count is controlled by PRESCALER in PnCTRL(n = 0,1). PnCOUNT is not loadable. The initial value of PnCOUNT is 0x0000.

9.9.2.8 PWM Channel [0,1] Width Register (P[0,1]WIDTH)

0x8005.D0A4 / 0x8005.D0C4

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P[0,1]WIDTH															

Bits	Type	Function
15:0	R/W	PWM 0 (PWM 1) width register When OUTPUTINVERT in PnCTRL is '0', the value written in this register represents the duration of PWM output's HIGH level. When OUTPUTINVERT in PnCTRL is '1', the value written in this register represents the duration of PWM output's LOW level.

9.9.2.9 PWM Channel [0,1] Period Register (P[0,1]PERIOD)

0x8005.D0A8 / 0x8005.D0C8

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P[0,1]PERIOD															

Bits	Type	Function
15:0	R/W	PWM 0 (PWM 1) period register This register is used to define 1 period of PWM output. When PnCOUNT reaches PnPERIOD, the counter resets to 0x0000 and starts counting again.

9.9.2.10 PWM Channel [0,1] Control Register (P[0,1]CTRL)

0x8005.D0AC / 0x8005.D0CC

7	6	5	4	3	2	1	0
PRESCALER				OUTPUT INVERT	OUTPUT ENABLE	SOFT RESET	PWM ENABLE

Bits	Type	Function	
7:4	R/W	Counter clock prescaler PnCOUNT is clocked by (PRESCALER + 1)th PCLK(n = 0,1).	
		PRESCALER	Clock source
		0000	PCLK (default)
		0001	PCLK/2
		0010	PCLK/3
		0011	PCLK/4
		…	…
		1110	PCLK/15
		1111	PCLK/16
3	R/W	PWM output waveform inverting Normally the PWM output is LOW when PnCOUNT reaches PnWIDTH and HIGH when PnCOUNT reaches PnPERIOD. Setting this bit makes the polarity of PWM output to be inverted. If this bit is set, the PWM output is HIGH when PnCOUNT reaches PnWIDTH and LOW when PnCOUNT reaches PnPERIOD. The initial value of PWM output is HIGH regardless of OUTPUTINVERT in PnCTRL. 0 = PWM output is not inverted. (default) 1 = PWM output is inverted.	
2	R/W	PWM output enable Setting this bit enables the output of each PWM channel to propagate through pin PWM[0] or PWM[1]. If a system reset or SOFTRESET in PnCTRL register occurs, the output is reset to '0'. 0 = Output propagation is disabled. (default) 1 = Output propagation is enabled.	
1	R/W	Software reset command This bit resets PnCOUNT and the output of each PWM channel. This bit is not auto-cleared so user should clear this bit after issuing SOFTRESET command. 0 = Normal operation. (default) 1 = Resets PnCOUNT and output of PWM channel.	
0	R/W	Counter enable. Setting this bit enables PnCOUNT to increment. 0 = Stops counting. (default) 1 = Starts counting.	

9.9.3 Operation

9.9.3.1 Timer Counter Clock Sources

The counter of each timer channel is clocked by the peripheral clock PCLK. The clock source is selected by the clock select logic which is controlled by the PRESCALER bits in TnCTRL.

The counter can be clocked directly by the PCLK by setting the PRESCALER “0000”. This provides the fastest operation, with a maximum clock frequency equal to the PCLK frequency(F_{PCLK}). Alternatively, one of 15 taps from the prescaler can be used as a clock source. The prescaled clock has a frequency of either F_{PCLK} /2, F_{PCLK} /3, F_{PCLK} /4, …, F_{PCLK} /14, or F_{PCLK} /16.

The prescaler operates when PRESCALER in TnCTRL is non-zero value, and each counter logic has it’s own clock select logic. The counter starts to counting upward after COUNTENABLE in TnCTRL is set.

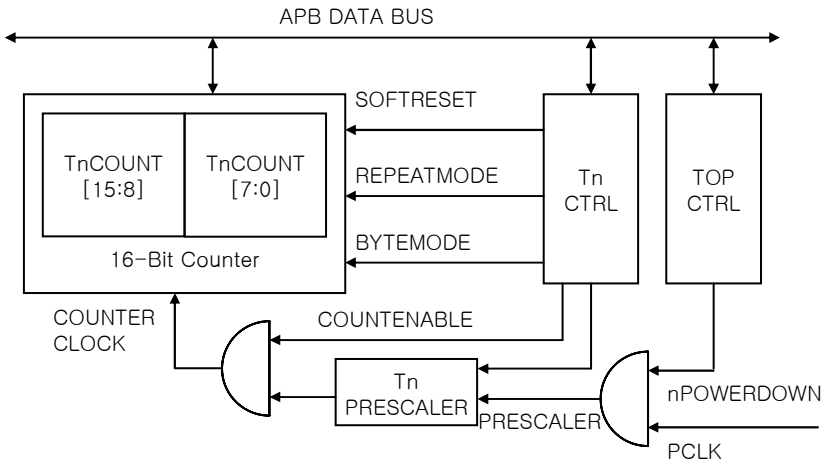


Figure 9-28. Clock select logic

9.9.3.2 *Repeat and non-repeat mode of timer channel*

There are two operation modes in each counter module which are non-repeat mode and repeat mode.

In non-repeat mode, the counter stops when TnCOUNT reaches TnBASE and an interrupt can be triggered if TIMERnINTEN bit is set. Also, the output of timer channel is toggled.

In repeat mode, the counter is free-running until COUNTENABLE is cleared. Whenever TnCOUNT reaches TnBASE, timer channel's output toggles and an interrupt can be triggered. At the moment TnCOUNT equals to TnBASE, the counter is cleared and starts counting from initial value(0x0000) while COUNTENABLE is high.

To operate timer in non-repeat mode, follow the steps below :

Non-repeat mode

- Activate clock source by setting nPOWERDOWN '1' and determine whether to propagate the output of timer channel or not. Also determine whether interrupt is enabled or not. (TOPCTRL)
- Set the target value. (TnBASE)
- Select clock frequency and non-repeat mode. (TnCTRL)
- Start counting. (TnCTRL)

Through out this chapter, the following symbols are used.

PCLK : peripheral clock PCLK (CCLK/13)
CountClk : clock source of counter which is PCLK or It's prescaled clock.
TIMERnOut : output of each timer channel that can be propagated through TIMER[n].
TIMERnInterrupt : interrupt source of each timer channel

The following figure is an example of non-repeat mode operation.

In this figure, see that CountClk is stopped when TnCOUNT equals to TnBASE and the LSB of TnCTRL is cleared. These are the characteristics of non-repeat mode operation of timer. The output of timer channel changes and interrupt can be triggered when TnCOUNT equals to TnBASE.

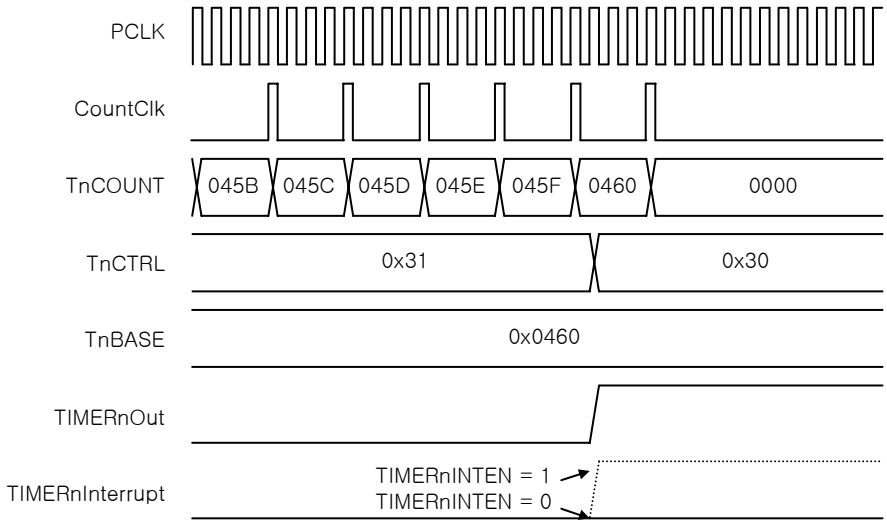


Figure 9-29. Non-repeat mode operation

To operate timer in repeat mode, follow the steps below :

Repeat mode

- Activate clock source by setting nPOWERDOWN '1' and determine whether to propagate the output of timer channel or not. Also determine whether interrupt is enabled or not. (TOPCTRL)
- Set the target value. (TnBASE)
- Select clock frequency and repeat mode. (TnCTRL)
- Start counting. (TnCTRL)

The following figure is an example of repeat mode operation.

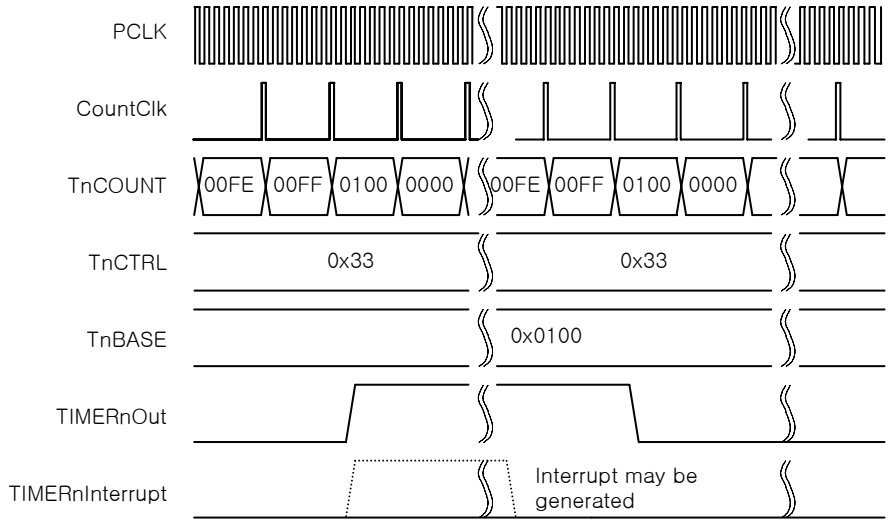


Figure 9-30. Repeat mode operation

As it can be seen in the above figure, CountClk is not stopped while COUNTENABLE is high. And TIMERnOut changes it's value at the moment TnCOUNT equals to TnBASE.

9.9.3.3 8-bit timer operation

Normally TnCOUNT is 16-bit up counter. And if TnBASE is at it's reset value, TnCOUNT increments up to 0xFFFF and then overflows(overflow interrupt is not supported). But TnCOUNT can also used as 8-bit counter by setting BYTEMODE

To operate timer in repeatmode, follow the steps below :

Byte mode

- Activate clock source by setting nPOWERDOWN '1' and determine whether to propagate the output of timer channel or not. Also determine whether interrupt is enabled or not. (TOPCTRL)
- Set the target value. (TnBASE)
- Select clock frequency and determiner repeat or non-repeat mode. (TnCTRL)
- Select byte mode and start counting. (TnCTRL)

The following figure is an example of byte counter in non-repeat mode operation. Note that the timing and operation is the same as normal 16-bit counter in non-repeat mode when TnBASE is less than or equal to "0xFF".

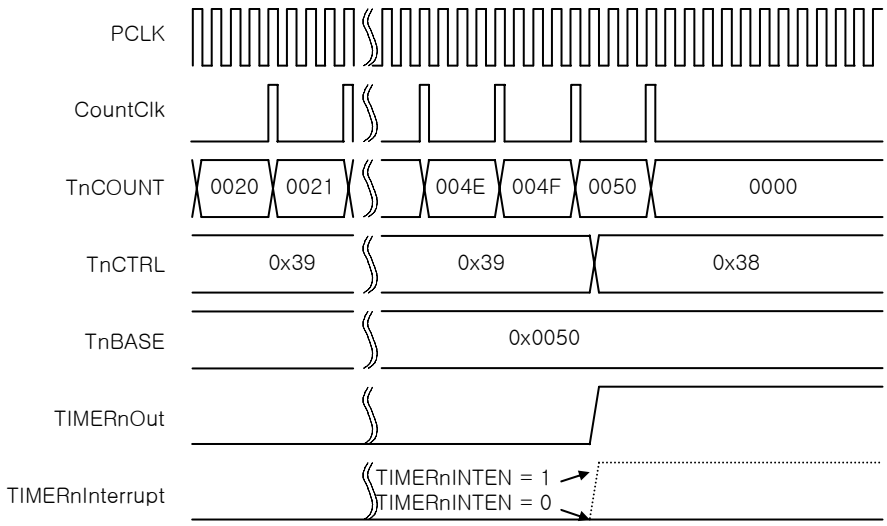


Figure 9-31. Byte counter operation in non-repeat mode

The following figure is an example of byte counter in repeat mode operation.

Note that TnBASE is out of the range of 8-bit counter, so TnCOUNT never reaches TnBASE therefore no interrupt is triggered and output of timer maintain previous value.

Except that it is the same as normal 16-bit counter in repeat mode operation.

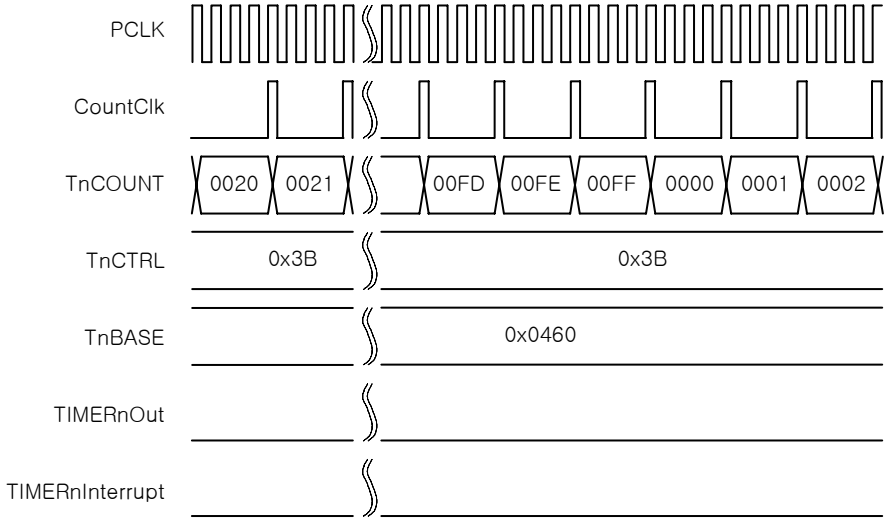


Figure 9-32. Byte counter operation in repeat mode

9.9.3.4 *Timer channel 3 clock source change*

Counters of all timer channel are clocked by PCLK or it's prescaled clock. But counter of timer channel 3 has additional clock source.

When TIMER3CLKSEL in TOPCTRL is set, T3COUNT is clocked when T2COUNT equals to T2BASE(T2MATCH event). Even if T3COUNT is clocked by T2MATCH event, the prescaler of timer channel 3 works.

In the following figure, the PRESCALER value of timer channel 3 is '0', so at each T2MATCH event T3COUNT is clocked.

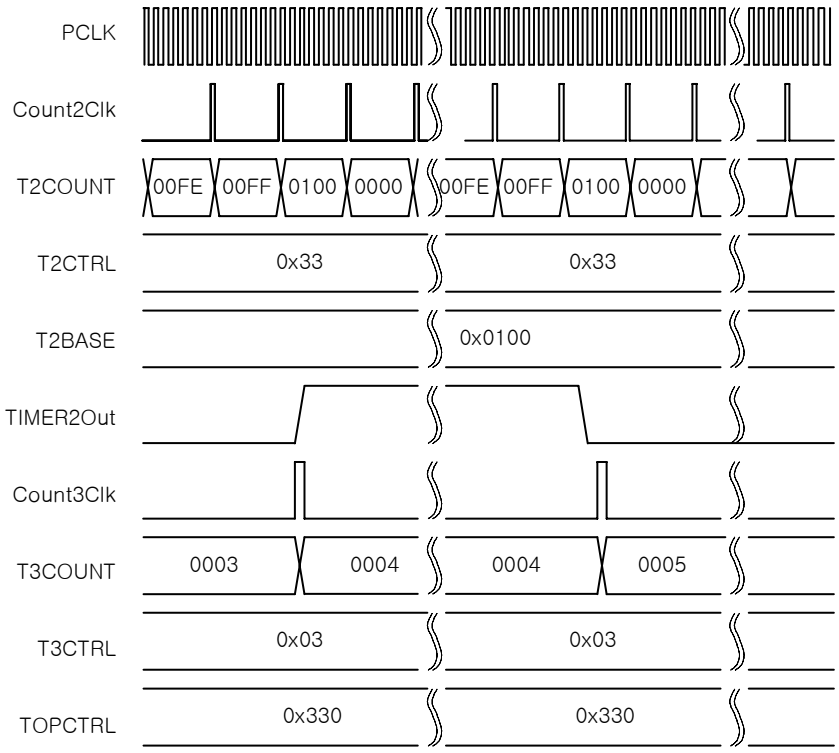


Figure 9-33. Clock source of T3COUNT is T2MATCH event

9.9.3.5 *Timer soft reset*

When SOFTRESET in TnCTRL is set, counter and output of timer channel n is cleared.

Note that SOFTRESET bit is not auto-cleared, so TnCTRL must be re-written to start counting again. SOFTRESET is an asynchronous reset input to counter module, so while SOFTRESET is HIGH, the counter and output are in their reset state.

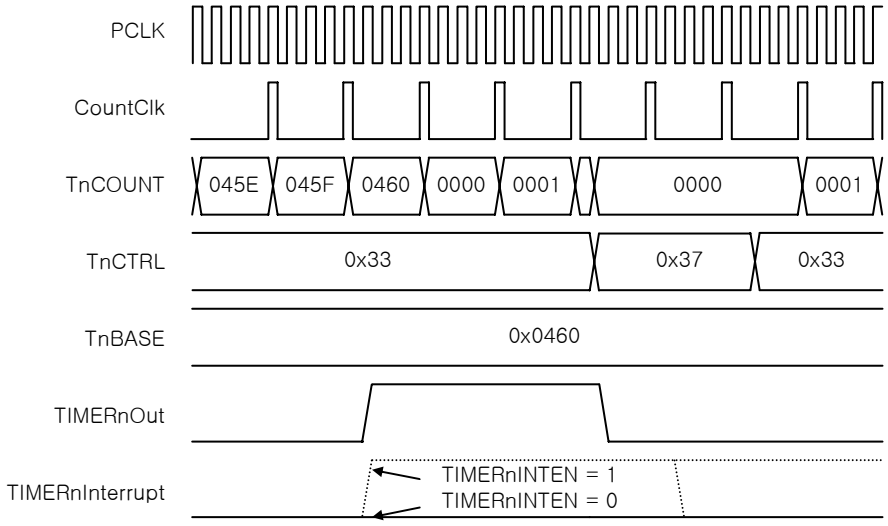


Figure 9-34. Software issued reset command

9.9.3.6 *Timer output and interrupt generation*

There is only one interrupt condition in each timer channel and it's match event of TnCOUNT.

As seen below, TnCOUNT increments after COUNTENABLE is set. When TnCOUNT reaches TnBASE (match condition), the counter is cleared and timer output is toggled, and if interrupt generation is enabled by TIMERnINTEN timer interrupt is also requested. If timer operates in repeat mode, the counter continues to increment from 0x0000. If match condition occurs, the MATCH bit in TnSTAT is set.

The following figure is an example of counter in repeat mode operation.

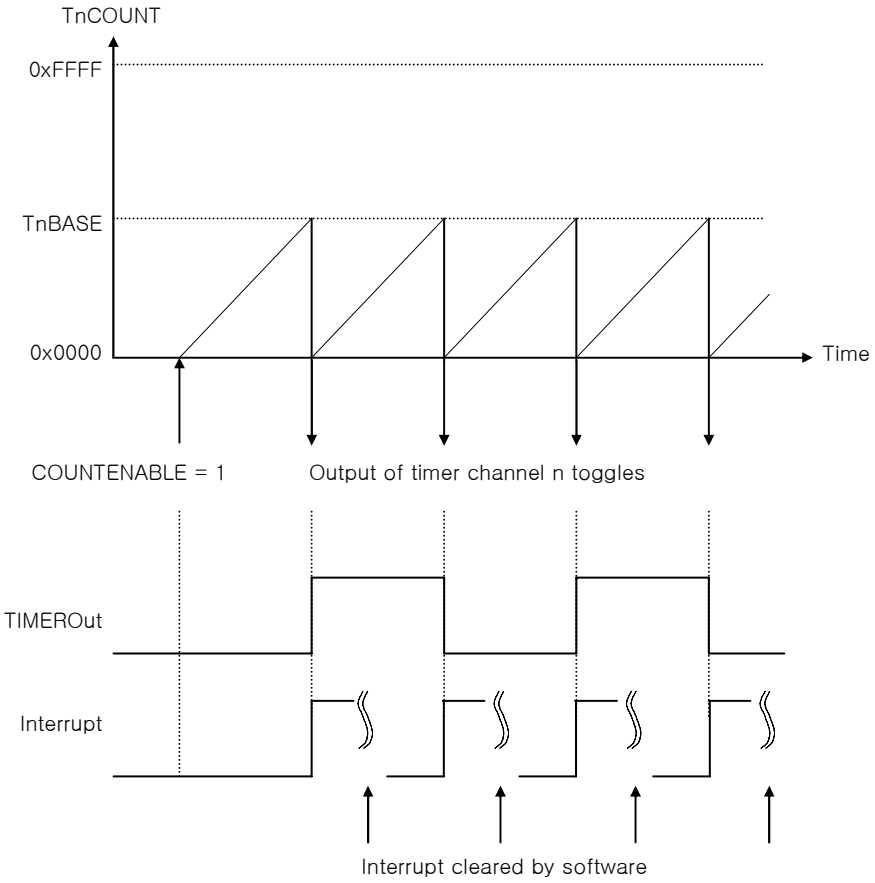


Figure 9-35. Output and interrupt generation in repeat mode

The following figure is an example of counter in non-repeat mode operation. All is the same as above but when match condition occurs the counter stops.

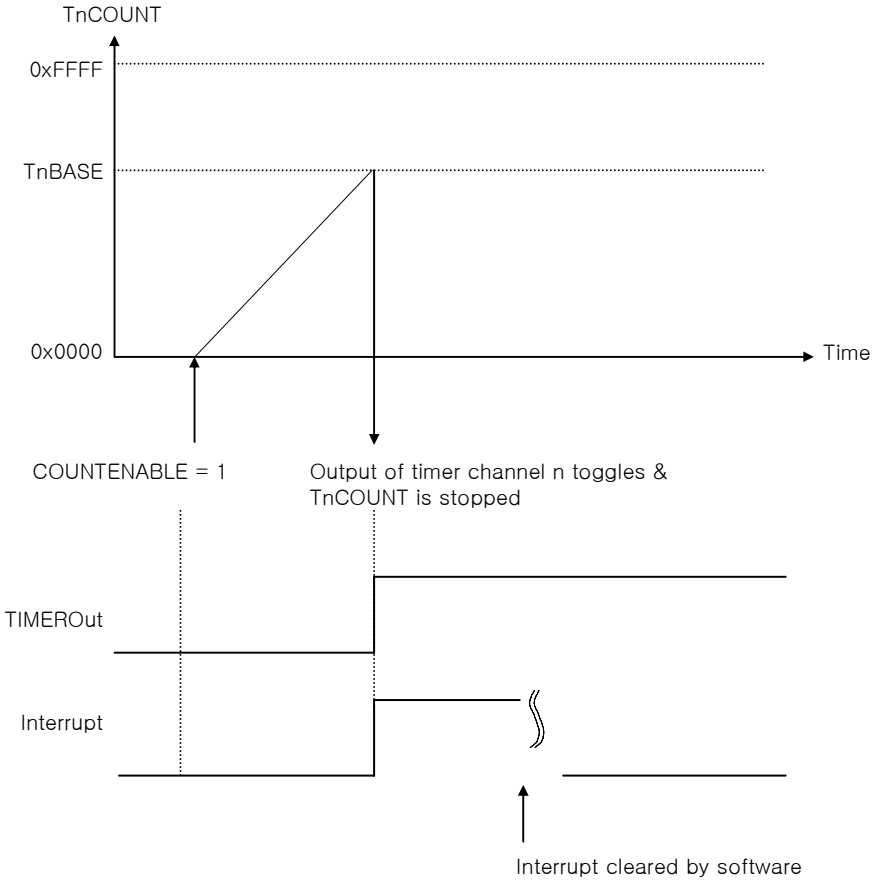


Figure 9-36. Output and interrupt generation in non-repeat mode

9.9.3.7 PWM Counter Clock Sources

The counter of each PWM channel is clocked by the peripheral clock PCLK. The clock source is selected by the clock select logic which is controlled by the PRESCALER bits in PnCTRL.

The PWM counter can be clocked directly by the PCLK by setting the PRESCALER "0000". This provides the fastest operation, with a maximum clock frequency equal to the PCLK frequency(F_{PCLK}). Alternatively, one of 15 taps from the prescaler can be used as a clock source. The prescaled clock has a frequency of either F_{PCLK} /2, F_{PCLK} /3, F_{PCLK} /4, …, F_{PCLK} /14, or F_{PCLK} /16.

The prescaler operates when PRESCALER in PnCTRL is non-zero value, and each counter logic has it's own clock select logic.

The counter starts to counting upward after PWMENABLE in PnCTRL is set.

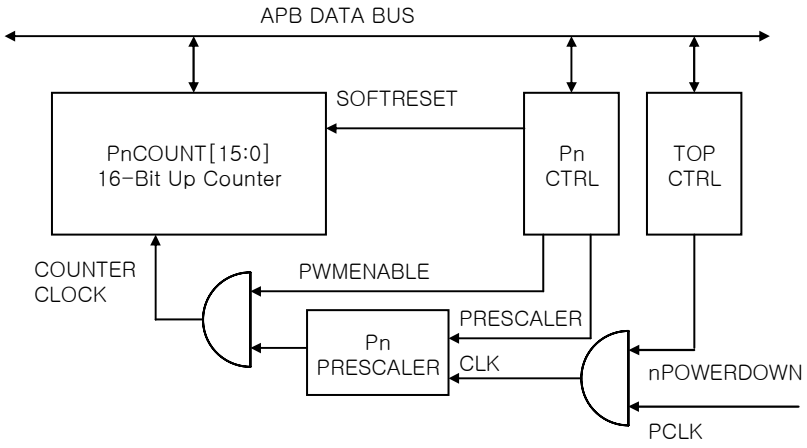


Figure 9-37. Clock select logic

- Activate clock source by setting nPOWERDOWN '1'. (TOPCTRL)
- Set the PWM period and duration. (PnPERIOD, PnWIDTH)
- Determine whether to propagate the output of PWM channel or not. (PnCTRL)
- Select clock frequency and start counting. (PnCTRL)

9.9.3.8 *PWM output generation*

PWM output's duty and period is controlled by the registers PnWIDTH and PnPERIOD.

When OUTPUTINVERT is '0' :

PWM output goes LOW when PnCOUNT reaches PnWIDTH and continues to increment. When PnCOUNT reaches PnPERIOD, PWM output goes HIGH and PnCOUNT is cleared. This is repeated until PWMENABLE is high. In this setting PnWIDTH is the duration of HIGH level of PWM output. The following figure shows the example of PWM waveform when OUTPUTINVERT is '0'. At reset, PWMOut is HIGH.

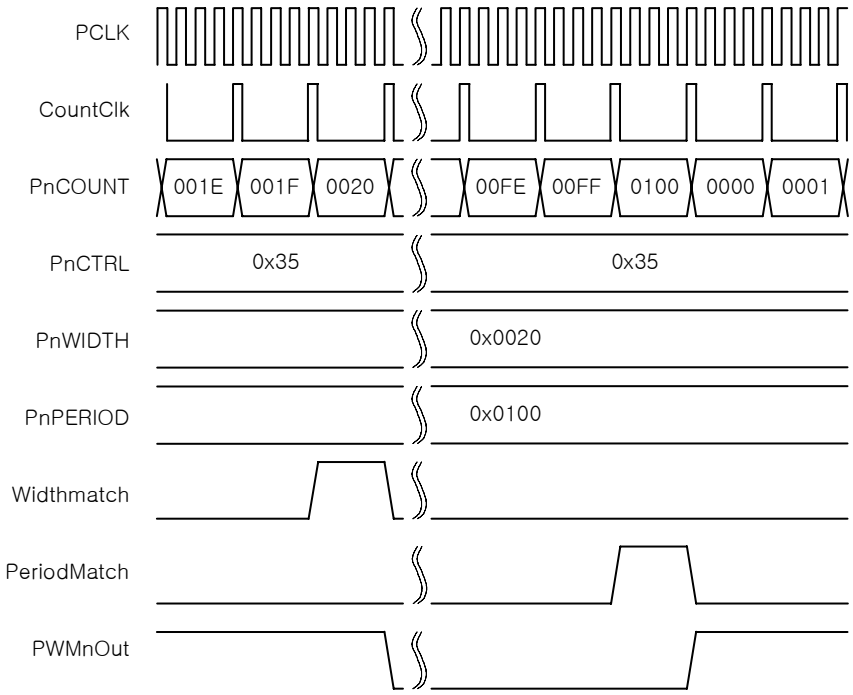


Figure 9-38. Timing diagram of PWM channel when OUTPUTINVERT = 0

When OUTPUTINVERT is ‘1’ :
PWM output goes HIGH when PnCOUNT reaches PnWIDTH and continues to increment. When PnCOUNT reaches PnPERIOD, PWM output goes LOW and PnCOUNT is cleared. This is repeated until PWMENABLE is high. In this setting PnWIDTH is the duration of LOW level of PWM output. The following figure shows the example of PWM waveform when OUTPUTINVERT is ‘1’.

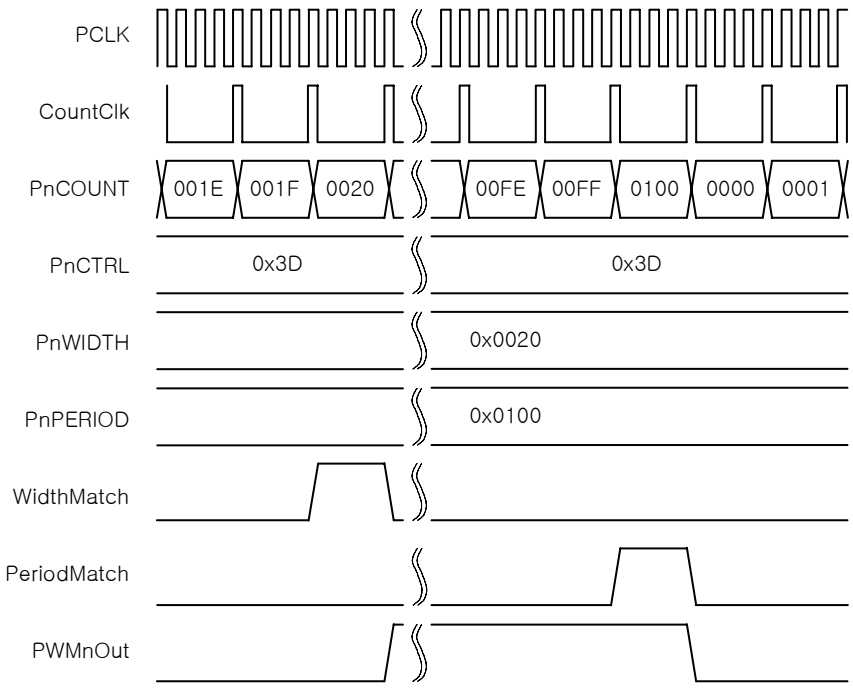


Figure 9-39. Timing diagram of PWM channel when OUTPUTINVERT = 1

9.9.3.9 *PWM duty control*

When OUTPUTINVERT is '0' :

In this setting PnWIDTH is the duration of HIGH level of PWM output. Below 2 figures show the waveform of PWM output for 30% and 80% duty ratio when OUTPUTINVERT is '0'.

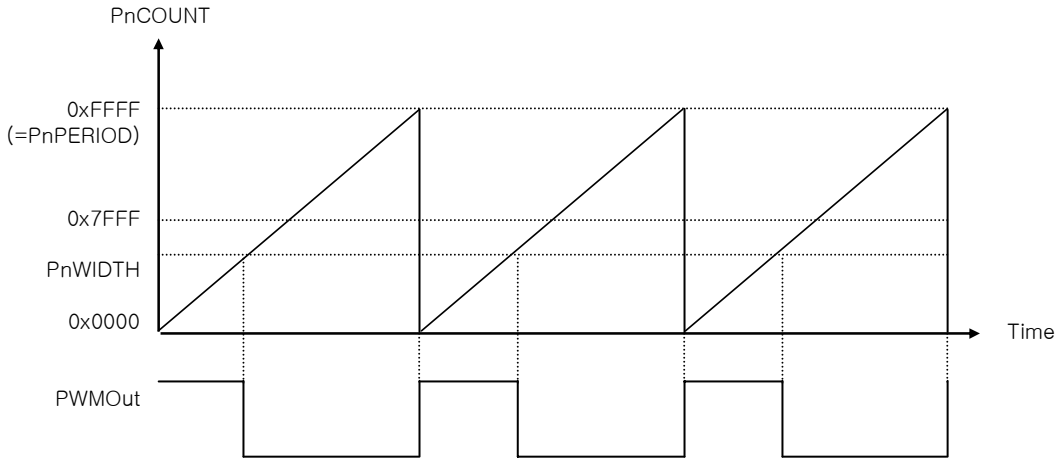


Figure 9-40. PWM waveform when OUTPUTINVET = 0, duty = 30%

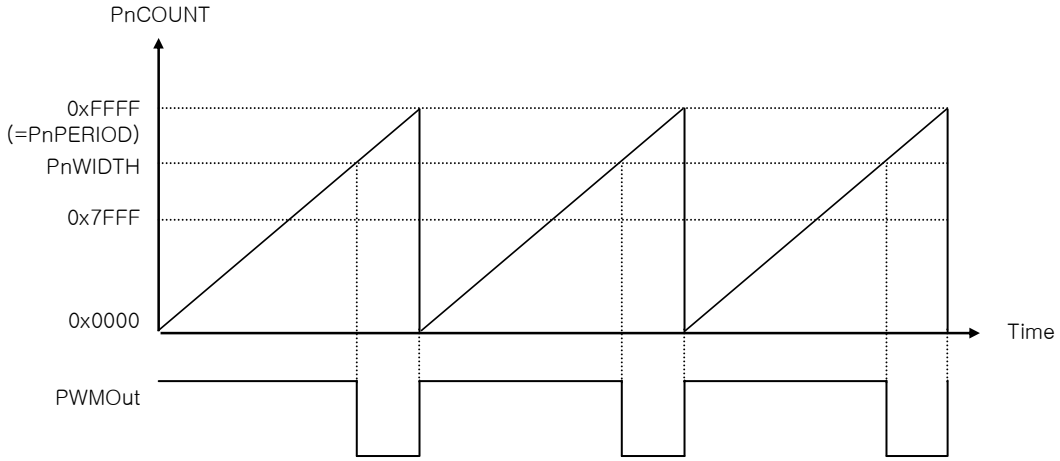


Figure 9-41. PWM waveform when OUTPUTINVET = 0, duty = 80%

The frequency of PWM output is calculated by the equation F_{PWM} = $F_{CountClk}$ / PnPERIOD where $F_{CountClk}$ is the frequency of clock source of PWM counter. Hence the value of PnPERIOD affects the period of PWM output.
If the value PnPERIOD is fixed, changing the value of PnWIDTH extends or shrinks the length of HIGH level of PWM output with period fixed. Hence the value of

PnWIDTH affects the duty ratio of PWM output.
If PnWIDTH is greater than PnPERIOD, the PWM output is always HIGH. 50% duty ratio is achieved by setting PnWIDTH half of PnPERIOD.
At reset, PWMOut is HIGH.

When OUTPUTINVERT is ‘1’ :

In this setting PnWIDTH is the duration of LOW level of PWM output. Below 2 figures show the waveform of PWM output for 30% and 80% duty ratio when OUTPUTINVERT is ‘1’.

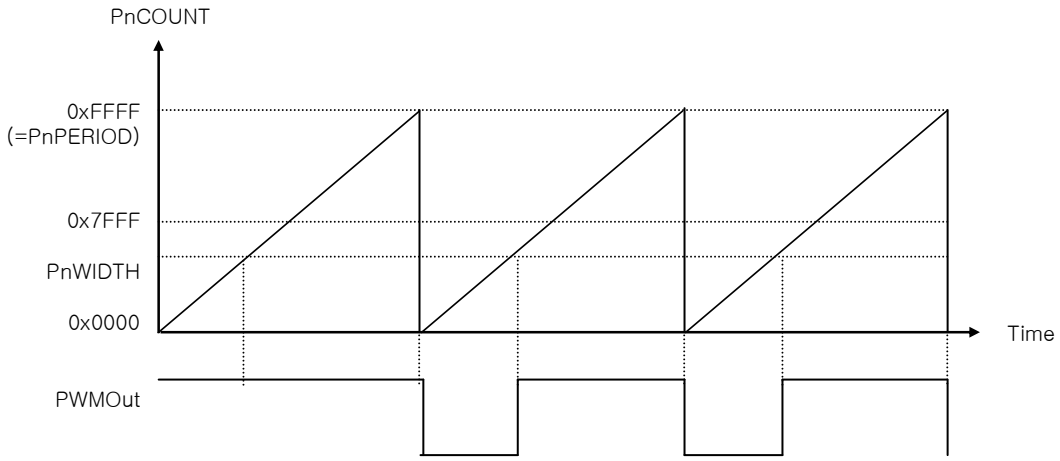


Figure 9-42. PWM waveform when OUTPUTINVET = 1, duty = 30%

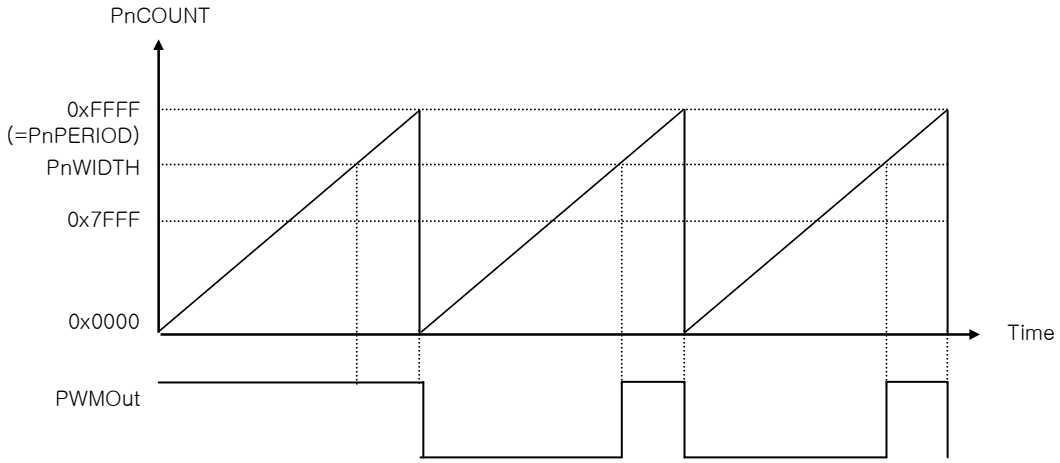


Figure 9-43. PWM waveform when OUTPUTINVET = 1, duty = 80%

Note that the initial value of PWMOut is HIGH, so the 1st period of PWM output is always HIGH when OUTPUTINVERT is ‘1’.

9.9.3.10 Timer soft reset

Like timer module, when SOFTRESET in PnCTRL is set, counter and output of PWM channel n is cleared.

Note that SOFTRESET bit is not auto-cleared, so PnCTRL must be re-written to start counting again. SOFTRESET is an asynchronous reset input to counter module, so while SOFTRESET is HIGH, the counter and output are in their reset state.

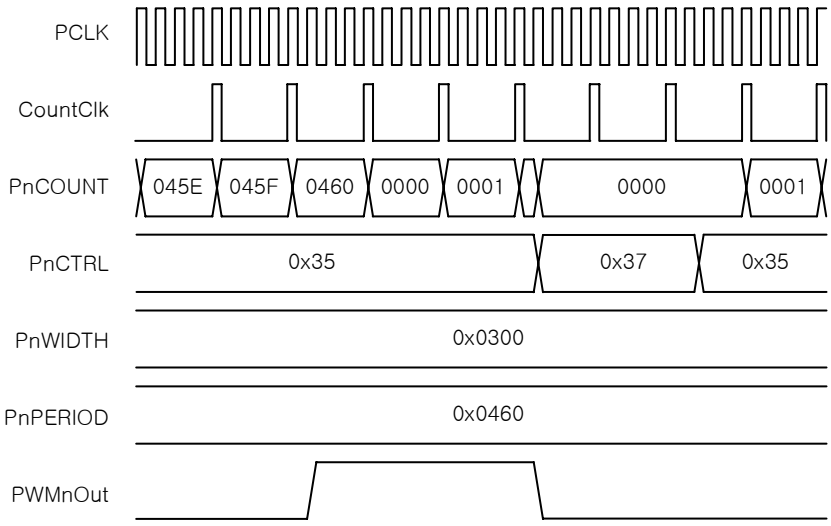


Figure 9-44. Software issued reset command

9.10 Watchdog Timer

The watchdog timer (WDT) has an one-channel for monitoring system operation. If a system becomes uncontrolled and the timer counter overflows without being rewritten correctly by the CPU, a reset signal is output to PMU. When this watchdog function is not needed, the WDT can be used as an interval timer. In the interval timer operation, an interval timer interrupt is generated at each counter overflow.

FEATURES

- Watchdog timer mode and interval timer mode
- Interrupt signal INTWDT to interrupt controller in the watchdog timer mode & interval timer mode
- Output signal MNRESET to PMU (Power Management Unit)
- Eight counter clock sources
- Selection whether to reset the chip internally or not
- Reset signal type: manual reset
- Clock source is 32.768KHz

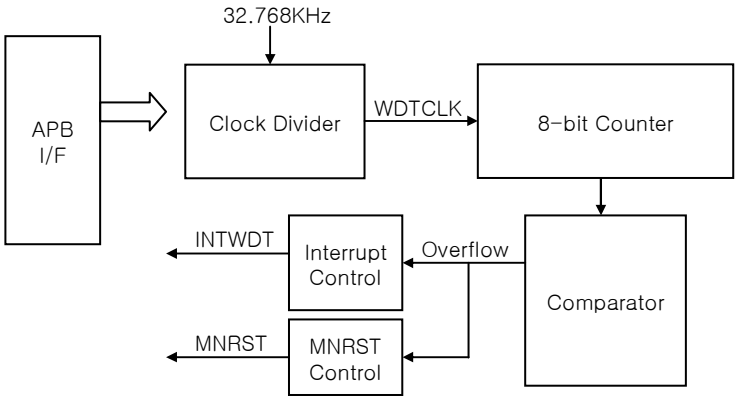


Figure 9-4 WDT block diagram

9.10.1 Registers

Address	Name	Width	Default	Description
0x8005.E000	WDTCTRL	8	0x0	Timer/Reset Control
0x8005.E004	WDTSTAT	2	0x0	Reset Status
0x8005.E008	WDTCNT	8	0x0	Timer Counter

Table 9-14. Watchdog Timer Register Summary

9.10.1.1 WDT Control Register (WDTCTRL)

0x8005.E000

7	6	5	4	3	2	1	0
INTEN	MODESEL	TMEN	MNRSTEN [4:3]		CLKSEL [2:0]		

Bits	Type	Function
7	R/W	The interrupt request enable When the value of WDTCNT register matches to 256 decimal value, an interrupt signal is generated. 0 = disable 1 = enable
6	R/W	Timer mode select Select whether to use the WDT as a watchdog timer or interval timer. 0 = interval timer mode 1 = watchdog timer mode
5	R/W	Enable the WDT timer When this bit is set to “0”, user can load data in WDTCNT register. 0 = disable 1 = enable
4:3	R/W	MNRST output enable Select whether to reset the chip internally or not if the TCNT overflows in the watchdog timer mode. 00 = disable 11 = MNRST output enable
2:0	R/W	Clock select The WDT has a clock generator which products eight counter clock sources. The clock signals are obtained by dividing the clock source. The clock Source is 32.768KHz.

CLKSEL[2:0]	Divide value	Divided clock	Max. overflow interval
000	2	16384 Hz	15.6 ms
001	8	4096 Hz	62.5 ms
010	32	1024 Hz	0.25 s
011	64	512 Hz	0.5 s
100	256	128 Hz	2 s
101	512	64 Hz	4 s
110	2048	16 Hz	16 s
111	8192	4 Hz	64 s

9.10.1.2 WDT Status Register (WDTSTAT)

0x8005.E004

7	6	5	4	3	2	1	0
-	-	-	-	-	-	ITOVF	WTOVF

Bits	Type	Function
7:2	-	Reserved
1	R	Interval timer interrupt flag This bit will be set to '1' when WDTCNT has overflowed in the interval timer mode. This bit is reset to '0' whenever the CPU reads the contents of this Register. 0: Interrupt was not generated or was cleared. 1: Interrupt was generated.
0	R	Watchdog timer interrupt flag This bit will be set to '1' when WDTCNT has overflowed in the watchdog timer mode. This bit is reset to '0' whenever the CPU reads the contents of this Register. 0: Interrupt was not generated or was cleared. 1: Interrupt was generated.

9.10.1.3 WDT Counter (WDTCNT)

0x8005.E008

7	6	5	4	3	2	1	0
WDTCNT							

Bits	Type	Function
7:0	R	8-bit up counter. When the timer is enabled, the timer counter starts counting pulse of the selected clock source. When the value of the WDTCNT changes from 0xFF-0x00(overflows), a watchdog timer overflow signal is generated in the both timer modes. The WDTCNT is initialized to 0x00 by a power-reset.

9.10.2 Watchdog Timer Operation

9.10.2.1 The Watchdog Timer Mode

To use the WDT as a watchdog timer, set the MODESEL and TMEN bits of the WDTCTRL register to '1'. Software must prevent WDTCNT overflow by rewriting the WDTCNT value (normally by writing 0x00) before overflow occurs. If the WDTCNT fails to be rewritten and overflow due to a system crash or the like, INTWDT signal and MNRST signal are output. The INTWDT signal is not output if the INTEN bit of WDTCTRL register is disabled (INTEN = 0). The MNRSTEN bits of WDTCTRL register should be set to '11' for MNRST output.

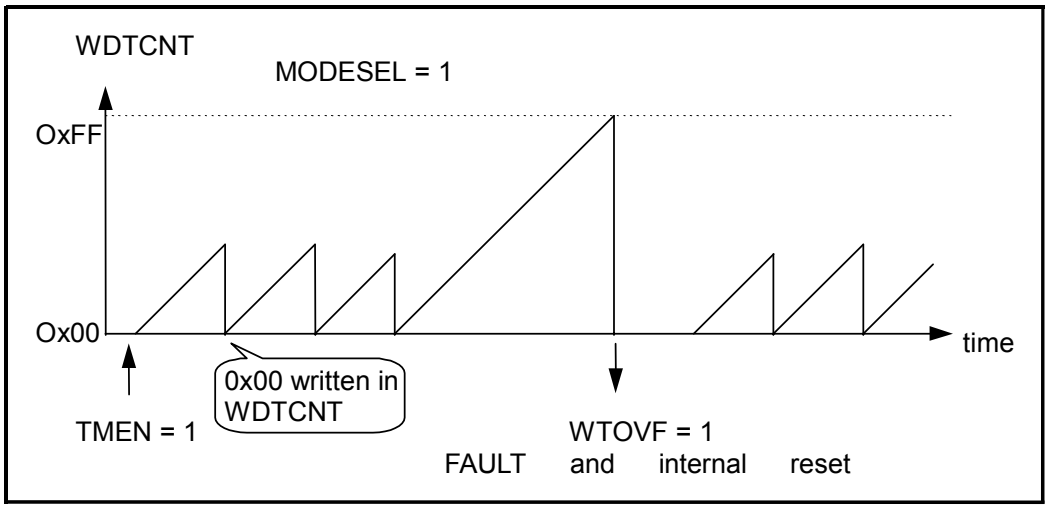


Figure 9-5 WDT Operation in the Watchdog Timer mode

9.10.2.2 *The Interval Timer Mode*

To use the WDT as an interval timer, clear MODESEL in WDTCTRL register to '0' and set TMEN to '1'. A interval timer interrupt (INTWDT) is generated each time the timer counter overflows. This function can be used to generate interval timer interrupts at regular intervals. The MNRSTEN bits of WDTCTRL register should be set to '00' .

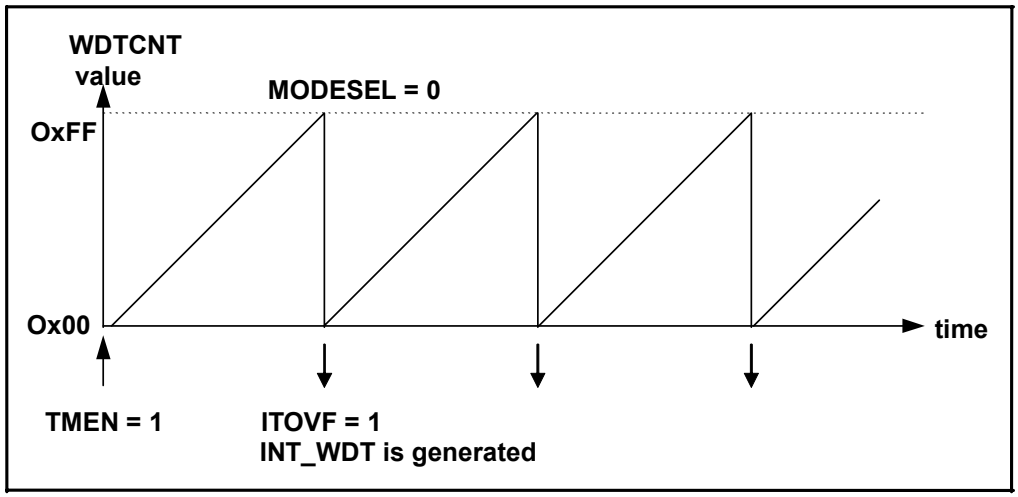


Figure 9-6 WDT Operation in the Interval Timer mode

9.10.2.3 *Timing of setting the overflow flag*

In the interval timer mode when the WDTCNT overflows, the ITOVF flag is set to 1 and an watchdog timer interrupt (INTWDT) is requested.
In the watchdog timer mode when the WDTCNT overflows, the WTOVF bit of the WDTSTAT is set to 1 and a WDTOUT signal is output. When RSTEN bit is set to 1, WDTCNT overflow enables an internal reset signal to be generated for the entire chip.

9.10.2.4 *Timing of clearing the overflow flag*

When the WDT Status Register (WDTSTAT) is read, the overflow flag is cleared.

9.10.2.5 Examples of Register setting

Interval Timer Mode (WDTCNT = 0x00 WDTCTRL = 0xA0)

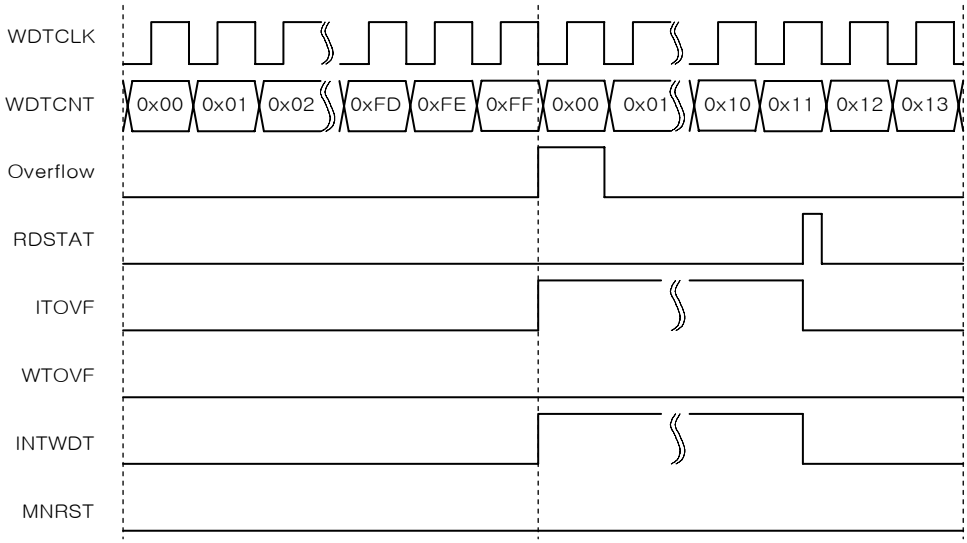


Figure 9-7 Interrupt clear in the interval timer mode

Watchdog Timer Mode with Internal Reset Disable (WDTCNT = 0x00 (normally) WDTCTRL = 0xE0)

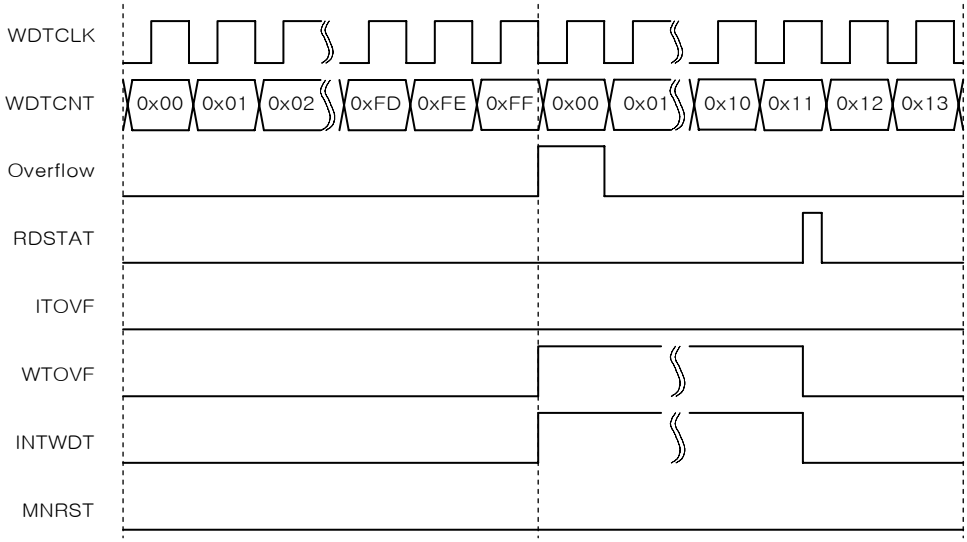


Figure 9-8 Interrupt Clear in the watchdog timer mode with MNRST disable

Watchdog Timer Mode with Manual Reset (WDTCNT = 0x00 WDTCTRL = 0xF8)

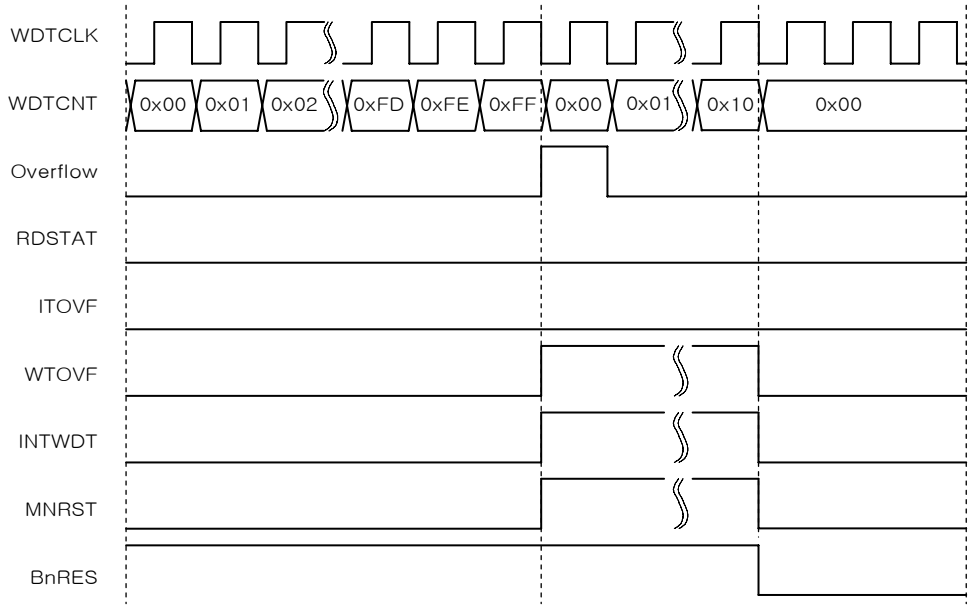


Figure 9-9 System reset generate in the watchdog timer mode with MSRST enable

9.10.2.6 WDT Setup Flow

Watchdog timer flow

- Set low to the TMEN bit in WDTCTRL register
- Load the wished data in WDTCNT reigster (default is 8'b00)
- Select the CLKSEL, INTEN bits in WDTCTRL register
- Set high to the MODESEL bit in WDTCTRL register
- Set '11' to the MNRSTEN bits in WDTCTRL register
- Set high to the TMEN bit in WDTCTRL register

Interval timer flow

- Set low to the TMEN bit in WDTCTRL register
- Load the wished data in WDTCNT reigster (default is 8'b00)
- Select the CLKSEL, INTEN bits in WDTCTRL register
- Set low to the MODESEL bit in WDTCTRL register
- Set '00' to the MNRSTEN bits in WDTCTRL register
- Set high to the TMEN bit in WDTCTRL register

9.11 RTC

The RTC works with an external 32768Hz crystal oscillator. It comprises second-counter to year-counter clock and calendar circuits that feature automatic leap-year adjustment up to year 2099, alarm and tick-timer interrupt functions. Also it can be operated by the backup battery while the system power down.
The RTC has two event outputs, one which is synchronized to PCLK, RTCIRQ, and the second, PWKUP synchronized to the 32768Hz clock. RTCIRQ is connected to the system interrupt controller, and PWKUP is used by the PMU to provide a system alarm Wake up.

FEATURES

- RTC count second, minute, hour, day, day of week, month and year with leap-year compensation valid up to 2099
- Alarm interrupt or wake-up signal from power-down mode
- Tick timer interrupt
- Independent power pin
- Write protection function

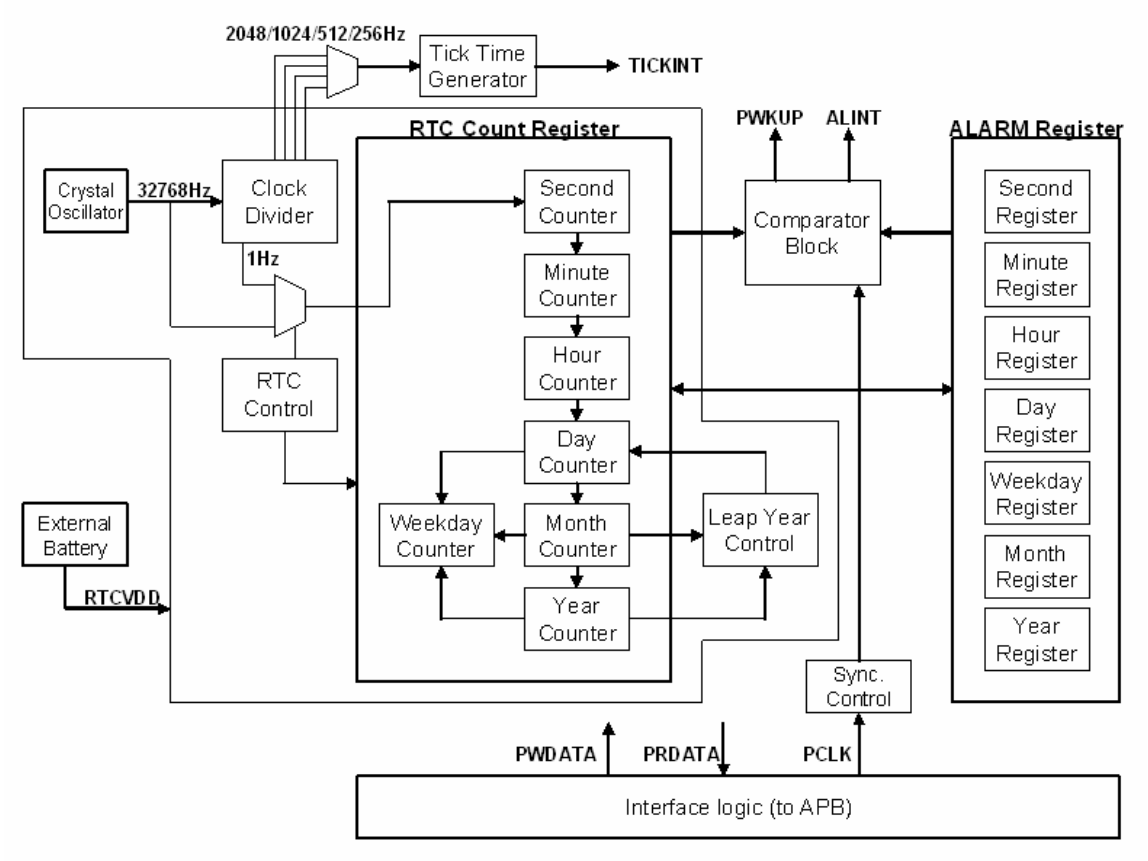


Figure 9-45. RTC Block Diagram

As shown in Figure 9-16, RTC module is connected to the APB. APB signals are refer to AMBA APB spec, and following table shows the non-AMBA signals from the RTC core block. The following table shows non-AMBA signals within RTC core block for more information about APB signals refer to the AMBA APB spec.

NAME	Source/Destination	Description
CLK32KHZ	Clock generator	32768HZ clock input. This is the signal that clocks the counter during normal operation.
RTCIRQ	APB(Interrupt controller) ASB(PMU)	When HIGH, this signal indicates a valid comparison between the counter value and the alarm register. It also indicates 1HZ interval with enable bit in control register. This signal is used to interrupt controller. Also it is used to wake up the HMS30C7210 when it is in deep sleep mode.
TICKIRQ	APB(Interrupt controller)	TICK Timer interrupt signal. It is generated when TCNT value meets TBASE value.

Table 9-15 Non-AMBA Signals within RTC Core Block

9.11.1 External Signals

Pin Name	Type	Description
RTCOSCIN	I	RTC oscillator input. 32.768KHz
RTCOSCOUT	O	RTC oscillator output. 32.768KHz

Refer to Figure 2-1. 208 Pin diagram.

9.11.2 Registers

Address	Name	Width	Default	Description	Write Protect
0x8005.F000	RTCTRL	6	0x1	RTC Control Register	Y
0x8005.F004	RTCSTAT	3	0x0	RTC Status Register	-
0x8005.F008	RTCSEC	7	0x0	RTC Second Register	Y
0x8005.F00C	RTCMIN	7	0x0	RTC Minute Register	Y
0x8005.F010	RTCHOR	6	0x0	RTC Hour Register	Y
0x8005.F014	RTCDAY	6	0x1	RTC Day Register	Y
0x8005.F018	RTCMON	5	0x1	RTC Month Register	Y
0x8005.F01C	RTCYER	8	0x0	RTC Year Register	Y
0x8005.F020	RTCWEK	3	0x0	RTC Week Register	Y
0x8005.F024	ALCTRL	8	0x0	ALARM Control Register	Y
0x8005.F028	ALSEC	7	0x0	ALARM Second Register	Y
0x8005.F02C	ALMIN	7	0x0	ALARM Minute Register	Y
0x8005.F030	ALHOR	6	0x0	ALARM Hour Register	Y
0x8005.F034	ALDAY	6	0x1	ALARM Day Register	Y
0x8005.F038	ALMON	5	0x1	ALARM Month Register	Y
0x8005.F03C	ALYER	8	0x0	ALARM Year Register	Y
0x8005.F040	ALWEK	3	0x0	ALARM Week Register	Y
0x8005.F044	TICTRL	8	0x0	TICK Control Register	Y
0x8005.F048	TICNT	8	0x0	TICK Count Register	Y
0x8005.F04C	TIBASE	8	0xFF	TICK Base Register	Y
0x8005.F060	PROTCTRL	1	0x1	Write Protection Control Register	-
0x8005.F07C	PROTECT1	8	-	Write Protection Register 1 (w/o)	-
0x8005.F064	PROTECT2	8	-	Write Protection Register 2 (w/o)	-
0x8005.F078	PROTECT3	8	-	Write Protection Register 3 (w/o)	-
0x8005.F06C	PROTECTLAST	8	-	Write Protection Register Last (w/o)	-
0x8005.F068	RTCTRLRESET	1	0x0	Control Register Reset Register	-

9.11.2.1 RTC Reset Register (RTCTRLRESET)

0x8005.F068

7	6	5	4	3	2	1	0
CTRLRESET							

Bits	Type	Function
7:1	-	Reserved
0	W	Reset bit to initialize the RTC Control Register If you use the RTC for the first time, you should set this bit first of all. If this bit is set to “1”, the RTC Control Register will be cleared. Notice: Only INTEN & EVTEN bits in the RTC control register are initialized by the system reset signal. 1: reset

9.11.2.2 RTC Protection Enable Register (PROTCTRL)

0x8005.F064

7	6	5	4	3	2	1	0
PROTECTEN							

Bits	Type	Function
7:1	-	Reserved
0	R/W	Write protection enable When this bit set to “1” , wirte protection setup flow is started. To release write protection, user should write some fixed value into RTC protection data registers sequentially. 0: No write avaliable to another register 1: wirte protection enable

Note the specific description is in chapter 9.11.3.6 Write operation

9.11.2.3 RTC Protection 1,2,3,LAST Register (PROTECT 1,2,3,LAST)

0x8005.F07C / 0x8005.F064 / 0x8005.F078 / 0x8005.F06C

7	6	5	4	3	2	1	0
Protect data [7:0]							

Bits	Type	Function
7:0	W	Protect data

Note the specific description is in chapter 9.11.3.6 Write operation

9.11.2.4 RTC Control Register (RTCTRL)

0x8005.F000

7	6	5	4	3	2	1	0
	EVTEN		INTEN				CLKSEL
RESET		RTC Stop					

Bits	Type	Default	Function
7:6	-	-	Reserved
5	R/W	0	RTC Event Enable. If this bit is set high, the event signal could be sent to PMU for using as a wake-up signal. There is no need for the event signal to set the INTEN bit 0: RTC event disable 1: RTC event enalbe
4	R/W	0	RTC Interrupt Enable When RTC Count register value meets RTC Alarm register's, alarm interrupt is generated. 0: Interrupt disable 1: Interrupt enable
3	-	-	Reserved
2	R/W	0	RTC Clock Select If this bit set high, RTC clock source will be connneted to 32768KHz only for test 0: RTC clock is 1Hz 1: RTC clock is 32768Hz
1	R/W	0	RTC Counter Register Reset If this bit is set high, RTC Counter Register will be cleared. Although this bit is 1, the RTC clock still alive. 0: no reset 1: RTC CNT register reset
0	R/W	1	RTC start / stop If this bit is set high, a 32KHz clock isn't supplied to the Clock Divider, and then a CLK1Hz isn't made. 0: RTC Start 1: RTC Stop

9.11.2.5 RTC Status Register (RTCSTAT)

0x8005.F004

7	6	5	4	3	2	1	0
TICK FLAG		READ FLAG	ALM FLAG				

Bits	Type	Function
7:3	-	Reserved
2	R	TICK Interrupt Status Flag Interrupt signal is generated when TICNT register value meets TIBASE register value. Read only valid and writing this bit to “1” clears this flag. 0: Interrupt was not generated or was cleared. 1: Interrupt was generated.
1	R	Read Status Flag If this bit is set, RTC CNT Register value is copied into the COPY reigster internally. The system could read the wished value in the COPY register. Read only valid and writing this bit to “1” clears this flag. 0: Read flag was not generated or was cleared. 1: Read flag was generated.
0	R	Alarm Interrupt Status Flag Alarm event interrupt flag is set when the RTC Register values equal to the contents of the Alarm Register. Read only valid and writing this bit to “1” clears this flag. An interrupt is continued for one second. After generating an interrupt, you have to clear ALARM enable bit in ALCTRL register before clearing the status bit. If the status bit is just only set low before going by 1 second, an interrupt is made again. 0: Interrupt was not generated or was cleared. 1: Interrupt was generated.

9.11.2.6 RTC Second Register (RTCSEC)

Data in RTC counter registers is interpreted in BCD format. For example, if the second register contains 0101[6:4] 1001[3:0], then the contents are interpreted as the value 59 seconds.

0x8005.F008

7	6	5	4	3	2	1	0
RTCSEC10 [6:4]						RTCSEC1 [3:0]	

Bits	Type	Function
7	-	Reserved
6:4	R/W	Value for 10 Seconds Unit from 0 to 5
3:0	R/W	Value for Second Unit from 0 to 9

9.11.2.7 RTC Minute Register (RTCMIN)

0x8005.F00C

7	6	5	4	3	2	1	0
RTCMIN10 [6:4]					RTCMIN1 [3:0]		

Bits	Type	Function
7	-	Reserved
6:4	R/W	Value for 10 Minutes Unit from 0 to 5
3:0	R/W	Value for Minute Unit from 0 to 9

9.11.2.8 RTC Hour Register (RTCHOR)

Hour register contents are values expressed in 24 hour mode.

0x8005.F010

7	6	5	4	3	2	1	0
		RTCHOR10 [5:4]			RTCHOR1 [3:0]		

Bits	Type	Function
7:6	-	Reserved
5:4	R/W	Value for 10 Hours Unit from 0 to 2
3:0	R/W	Value for Hour Unit from 0 to 9

9.11.2.9 RTC DAY Register (RTCDAY)

0x8005.F014

7	6	5	4	3	2	1	0
RTCDAY10 [5:4]			RTCDAY1 [3:0]				

Bits	Type	Function
7:6	-	Reserved
5:4	R/W	Value for 10 Days Unit from 0 to 3
3:0	R/W	Value for Day Unit from 0 to 9

9.11.2.10 RTC Month Register (RTCMON)

0x8005.F018

7	6	5	4	3	2	1	0
RTCMON10	RTCMON1 [3:0]						

Bits	Type	Function
7:5	-	Reserved
4	R/W	Value for 10 Months Unit from 0 to 1
3:0	R/W	Value for Month Unit from 0 to 9

9.11.2.11 RTC Year Register (RTCYER)

Leap-year adjustment is automatic for year 2000 to 2099

0x8005.F01C

7	6	5	4	3	2	1	0
RTCYER10							[7:4]
RTCYER1 [3:0]							

Bits	Type	Function
7:4	R/W	Value for 10 Years Unit from 0 to 9
3:0	R/W	Value for Year Unit from 0 to 9

9.11.2.12 RTC Day of Week Register (RTCWEK)

The day-of-week register contains values representing the day of week as shown in the following table.

0x8005.F020

7	6	5	4	3	2	1	0
RTCWEK [2:0]							

Bits	Type	Function
7:3	-	Reserved
2:0	R	Value for Weekday Unit from Saturday to Friday

Bit 2	Bit 1	Bit 0	Day of week
0	0	0	Saturday
0	0	1	Sunday
0	1	0	Monday
0	1	1	Tuesday
1	0	0	Wednesday
1	0	1	Thursday
1	1	0	Friday

9.11.2.13 RTC Alarm Control Register (ALCTRL)

0x8005.F024

7	6	5	4	3	2	1	0
ALEN		ALWEKEN		ALYEREN		ALMONEN	ALDAYEN
ALHOREM	ALMINEN		ALSECEN				

Bits	Type	Function
7	R/W	Alarm Enable An interrupt is continued for one second. After generating an interrupt, you have to clear ALARM enable bit before clearing the status bit. If the status bit is just only set low before going by 1 second, an interrupt is made again. 0: alarm function disable 1: alarm function enable
6	R/W	Alarm Day of Week Enable 0: disable 1: enable
5	R/W	Alarm Year Enable 0: disable 1: enable
4	R/W	Alarm Month Enable 0: disable 1: enable
3	R/W	Alarm Day Enable 0: disable 1: enable
2	R/W	Alarm Hour Enable 0: disable 1: enable
1	R/W	Alarm Minute Enable 0: disable 1: enable
0	R/W	Alarm Second Enable If this bit is set to “0”, the alarm interrupt is generated at “00” second. 0: disable 1: enable

9.11.2.14 RTC Alarm Second Register (ALSEC)

0x8005.F028

7	6	5	4	3	2	1	0
ALSEC10 [6:4]					ALSEC1 [3:0]		

Bits	Type	Function
7	-	Reserved
6:4	R/W	Value for 10 Seconds Unit from 0 to 5
3:0	R/W	Value for Second Unit from 0 to 9

9.11.2.15 RTC Alarm Minute Register (ALMIN)

0x8005.F02C

7	6	5	4	3	2	1	0
ALMIN10 [6:4]					ALMIN1 [3:0]		

Bits	Type	Function
7	-	Reserved
6:4	R/W	Value for 10 Minutes Unit from 0 to 5
3:0	R/W	Value for Minute Unit from 0 to 9

9.11.2.16 RTC Alarm Hour Register (ALHOR)

0x8005.F030

7	6	5	4	3	2	1	0
		ALHOR10 [5:4]			ALHOR1 [3:0]		

Bits	Type	Function
7:6	-	Reserved
5:4	R/W	Value for 10 Hours Unit from 0 to 2
3:0	R/W	Value for Hour Unit from 0 to 9

9.11.2.17 RTC Alarm Day Register (ALDAY)

0x8005.F034

7	6	5	4	3	2	1	0
ALDAY10 [5:4]				ALDAY1 [3:0]			

Bits	Type	Function
7:6	-	Reserved
5:4	R/W	Value for 10 Days Unit from 0 to 3
3:0	R/W	Value for Day Unit from 0 to 9

9.11.2.18 RTC Alarm Month Register (ALMON)

0x8005.F038

7	6	5	4	3	2	1	0
				ALMON10	ALMON1 [3:0]		

Bits	Type	Function
7:5	-	Reserved
4	R/W	Value for 10 Months Unit from 0 to 1
3:0	R/W	Value for Month Unit from 0 to 9

9.11.2.19 RTC Alarm Year Register (ALYER)

0x8005.F03C

7	6	5	4	3	2	1	0
ALYER10							[7:4]
ALYER1 [3:0]							

Bits	Type	Function
7:4	R/W	Value for 10 Years Unit from 0 to 9
3:0	R/W	Value for Year Unit from 0 to 9

9.11.2.20 RTC Alarm Day of Week Register (ALWEK)

The day-of-week register contains values representing the day of week as shown in the following table.

0x8005.F040

7	6	5	4	3	2	1	0
					ALWEK [2:0]		

Bits	Type	Function
7:3	-	Reserved
2:0	R/W	Value for Weekday Unit from Saturday to Friday

Bit 2	Bit 1	Bit 0	Day of week
0	0	0	Saturday
0	0	1	Sunday
0	1	0	Monday
0	1	1	Tuesday
1	0	0	Wednesday
1	0	1	Thursday
1	1	0	Friday

9.11.2.21 RTC Tick Timer Control Register (TICTRL)

0x8005.F044

7	6	5	4	3	2	1	0
TINTEN		CLKSEL [5:4]				nPWDN	CNTReset
CNTRepeat	CNTEN						

Bits	Type	Function
7	-	Reserved
6	R/W	Tick Timer Interrupt Enable 0: Interrupt disable 1: Interrupt enable
5:4	R/W	Tick Timer Source Clock Select 00: 256Hz 01: 512Hz 10: 1024Hz 11: 2048Hz
3	R/W	Tick Timer Power Down mode If this bit is set to “1”, source clock is not connected into TICK Timer. 0: normal mode 1: power down mode
2	R/W	Tick Timer Count Register Reset 0: No reset 1: Counter Register Reset
1	R/W	Tick Timer Repeat Mode
0	R/W	Tick Timer Count Enable 0: Stop Count 1: Start Count

9.11.2.22 RTC Tick Timer Count Register (TICNT)

0x8005.F048

7	6	5	4	3	2	1	0
TICNT [7:0]							

Bits	Type	Function
7:0	R	Tick Time Count Value from 0 to 255

9.11.2.23 RTC Tick Timer Base Register (TIBASE)

0x8005.F04C

7	6	5	4	3	2	1	0
TIBASE [7:0]							

Bits	Type	Function
7:0	R/W	Tick Time Base Value from 0 to 255

9.11.3 Operation

9.11.3.1 Read/Write Operation

To read and write the register in RTC, Bit 0 of the RTCTRL register must be set.
To display calendar and present time, you (or CPU) should read the data in RTCSEC, RTCMIN, RTCHOR, RTCDAY, RTCWEK, RTCMON, RTCYER registers respectively.

9.11.3.2 Leap Year Generator

This block can determine whether the last date of each month is 28,29,30,31.
It is based on data from RTCDAY, RTCMON and RTCYER registers.

9.11.3.3 Alarm Function

Alarm can be set for year, month, day, weekday, hour, minute, and second.
Alarm Function is operated in normal mode or power down mode.
In power down (deep sleep) mode, the RTC generates wake-up signal (PWAKUP) for activating CPU when Alarm data is same with RTC data.
The RTC Alarm Control Register (ALCTRL) determines the alarm enable and the condition of the alarm time setting.
An interrupt is continued for one second. After generating an interrupt, you have to clear ALARM enable bit in ALCTRL register before clearing the status bit in RTCSTAT register. If the status bit is just only set low before going by 1 second, and interrupt is made again

9.11.3.4 Backup Battery Operation

When the system down, the RTC must be divided on the CPU. After that the RTC operates by using the backup battery.

9.11.3.5 Tick Time Interrupt

For interrupt request, the RTC includes Tick Time Counter Block,
Tick Time Counter can count value up to 255 (tick input frequency is optional. 2048/1024/512/256Hz)
The Tick timer also offers interrupt capability including a periodic interval timer.

9.11.3.6 *Write operation: Protection*

RTC write operation flow

- Protection write enable
- Set high to the CTRLRESET bit in RTCTRLRESET register
- Set low to the CTRLRESET bit in RTCTRLRESET register
- Set high to the RESET bit in RTCTRL register
- Set low to the RESET bit in RTCTRL register
- RTC register setup
- Set low to the RTCStop bit in RTCTRL register for starting
- Protection write disable

*Write Enable: PROTCTRL “high” → PROTECT1 “8’hAA” → PROTECT2 “8’h48” → PROTECT3 “8’h61” → PROTECTLAST “8’h99” → write enable

*Write Disable: PROTCTRL “low” → write disable

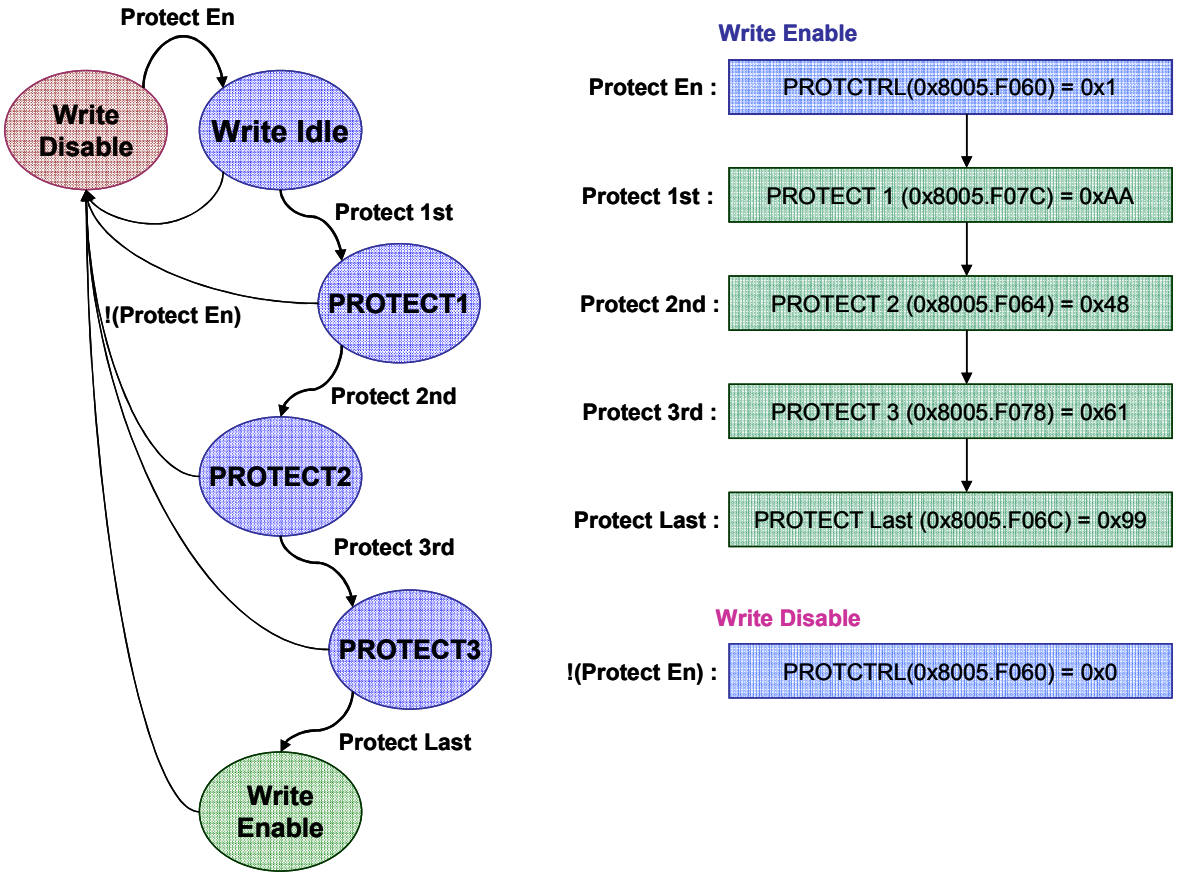


Figure 9-10 Write Protection Diagram

9.11.3.7 Read operation

Read Register: one of them - RTCSEC, RTCMIN, RTCHOR, RTCDAY, RTCMON, RTCYER, RTCWEK

- RTCSTAT[1] “high”
- After reading RTC register (SEC~WEK), RTCSTAT[1] should be cleared.
- If you read RTC register without clearing it, you will be given old values.

9.11.3.8 RTC Setup Flow

RTC initialization flow

- Protection write enable (refer to chapter 9.11.3.6 write operation)
- Set high to the CTRLRESET bit in RTCTRLRESET register
- Set low to the CTRLRESET bit in RTCTRLRESET register
- Set high to the RESET bit in RTCTRL register
- Set low to the RESET bit in RTCTRL register
- Protection write disable (refer to chapter 9.11.3.6 write operation)

RTC operation flow

- Protection write enable (refer to chapter 9.11.3.6 write operation)
- Set high the RTCStop bit in RTCTRL Register for RTC stop
- Set RTC count registers – RTCSEC/MIN/HOR/DAY/MON/YER
- Set Alarm control register
- Set Alarm time registers to wished value – ALSEC/MIN/HOR/DAY/MON/YER/WEK
- Select the EVTEN, INTEN, CLKSEL bits in RTCTRL register
- Set low the RTCStop bit in RTCTRL Register for starting RTC
- Protection write disable (refer to chapter 9.11.3.6 write operation)

TICK timer operation flow

- Protection write enable (refer to chapter 9.11.3.6 write operation)
- Set low the RTCStop bit in RTCTRL Register
- Set TIBASE reigster to wished value
- Select the TINTEN, CLKSEL[1:0], CNTRepeat bits in TICTRL register
- Set the CNTEN bit in TICTRL register for starting TICK timer
- Protection write disable (refer to chapter 9.11.3.6 write operation)

9.12 2-Wire Serial Bus Interface

The 2-Wire Serial Bus Interface (2-Wire SBI) is used to communicate external 2-Wire SBI compliant devices such as serial ROM or serial display device, etc. It supports both master and slave operation. The 2-Wire SBI protocol allows the systems designer to interconnect up to 128 different devices using only two bi-directional bus lines, one for clock (SCL) and one for data (SDA). The only external hardware needed to implement the bus is a single pull-up resistor for each of the 2-Wire SBI lines. All devices connected to the bus have individual addresses, and mechanisms for resolving bus contention are inherent in the 2-Wire SBI protocol.

The main features of 2-Wire SBI are :

- Only 2 lines needed to communicate
- Master and Slave operation
- Programmable transfer bit rate at master mode (Up to 400 KHz data transfer speed)
- Independently programmable mask of interrupts
- Multi-master capability
- Device can operate as transmitter or receiver
- Only 7-bit addressing is available

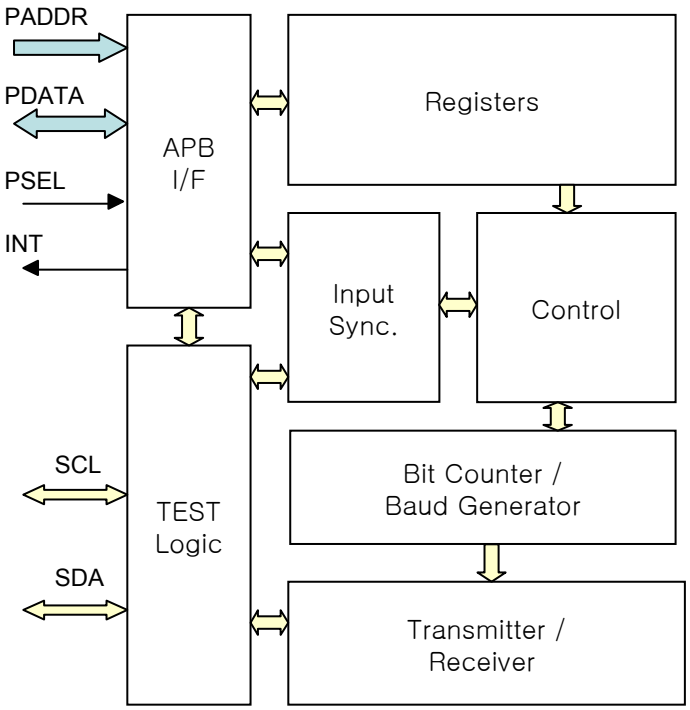


Figure 9-46. Block diagram of 2-Wire SBI

9.12.1 External Signals

Pin Name	Type	Description
SCL	I/O	Serial clock line SCL Serial clock signal pin. Pull-up this pin (open-drain)
SDA	I/O	Serial data line SDA Serial data signal pin. Pull-up this pin (open-drain)

Refer to Figure 2-1. 208 Pin diagram.

9.12.2 Registers

Address	Name	Width	Default	Description
0x8006.0000	DATAREG	8	0x0	2-Wire SBI Data Register
0x0806.0004	TARGETREG	8	0x0	2-Wire SBI Target Slave Address Register
0x8006.0008	STATUSREG	16	0x0	2-Wire SBI Status Register
0x8006.000C	SLAVEREG	7	0x0	2-Wire SBI Slave Mode Address Register
0x8006.0010	INTMASKREG	8	0x0	2-Wire SBI Interrupt Mask Register
0x8006.0014	CONFIGREG	8	0x0	2-Wire SBI Configuration Register
0x8006.0018	BAUDREG	8	0xf	2-Wire SBI Baud Rate Control Register

Table 9-16. 2-Wire SBI's Register Summary

9.12.2.1 2-Wire SBI Data Register (DATAREG)

0x8006.0000

7	6	5	4	3	2	1	0
DATA[7:0]							

Bits	Type	Function
7:0	R/W	Data to be transferred In transmit mode, DATAREG contains the next byte to be transmitted. In receive mode, the DATAREG contains the last byte received. It is writable while the 2-Wire SBI is not in the process of shifting a byte. This occurs when the 2-Wire SBI interrupt flag (bit that can be interrupt source in CONFIGREG) is set by hardware. The data in DATAREG remains stable as long as interrupt is set.

9.12.2.2 2-Wire SBI Target Slave Register (TARGETREG)

0x8006.0004

7	6	5	4	3	2	1	0
TARGET ADDR[6: 0]							R/W

Bits	Type	Function
7:1	R/W	Target slave's address These bits are the 1st data to be transmitted in the master mode and not needed in the slave mode. These bits are slave device's address.
0	R/W	Read or write This bit specifies transfer direction. When this value is '1', master request slave to transmit data (master Rx) and when '0', master transmits data to slave device(master Tx). 0 = Master is operating as transmitter. (default) 1 = Master is operating as receiver.

9.12.2.3 2-Wire SBI Status Register (STATUSREG)

0x8006.0008

15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	TRANSMITTER
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TRANSREQ	STOPREQ	EOTREQ	DATAREQ	BUSLOST	BUSBUSY	ACK RECEIVE	MASTER

Bits	Type	Function
15-9	R	Reserved.
8	R	2-Wire SBI is transmitter If this bit is set, it indicates that 2-Wire SBI operates as a transmitter. 0 = Operates as a receiver unit. (default) 1 = Operates as a transmitter unit.
7	R	Serial transfer requested This bit is set when serial communication is started by other external master and the 2-Wire SBI of HMS30C7210 is addressed by the master. This bit can be an interrupt source and is cleared by writing any value to STATUSREG. 0 = Status is cleared or 2-Wire SBI is not a slave module. (default) 1 = 2-Wire SBI is addressed by winning master.
6	R	Stop condition This bit is set when abnormal stop condition is detected during data transmission and can be an interrupt source. This bit is cleared by writing any value to STATUSREG. 0 = Status is cleared or normal stop condition is detected. (default) 1 = Serial communication is terminated abnormally.
5	R	End of transmission condition This bit is set when serial communication is ended by normal stop condition and can be an interrupt source. This bit is cleared by writing any value to STATUSREG. 0 = Status is cleared or serial communication is in progress. (default) 1 = Serial communication is terminated normally.
4	R	Data request After one byte of data is transferred on serial data line (SDA), this bit is set for 2-Wire SBI to prepare another byte of data (2-Wire SBI is a transmitter) or to read the received data (2-Wire SBI is a receiver). This bit can be an interrupt source and is cleared by writing any value to STATUSREG. 0 = Status is cleared or serial communication is over. (default) 1 = Prepare another byte of data or read the received data in DATAREG.
3	R	Bus lost event generated This bit is set when 2-Wire SBI lost mastership during arbitration (master mode) or there is no slave device addressed by 2-Wire SBI. This bit can be an interrupt source and is cleared by writing any value to STATUSREG. 0 = Status is cleared or 2-Wire SBI grant the ownership of serial bus lines. (default) 1 = Bus losing condition is generated.
2	R	Bus is busy now This bit is set while serial communication is going on. 0 = Serial bus is idle, in this case any bus master can issue a start condition. (default) 1 = Serial bus is used by 2-Wire SBI unit now.
1	R	ACK status This bit is set if the SDA line is pulled low by addressed slave device after ADDRESS cycle, or by a receiver acknowledges after DATA cycle. 0 = No ACK is received. (default) 1 = ACK is received.
0	R	2-Wire SBI is master Indicates whether 2-Wire SBI is configured as master or slave. 0 = 2-Wire SBI is a slave or the serial bus is idle. (default) 1 = 2-Wire SBI is a master.

The TRANSREQ, DATAREQ, STOPREQ, EOTREQ and BUSLOST bits in this

register are the source of 2-Wire SBI unit's interrupt. When 2-Wire SBI requests an interrupt, the handler reads or writes data according to the TRANSMITTER bit and clear the interrupt by writing STATUSREG. Or in receiver mode, the 2-Wire SBI can terminate serial communication by giving no ACK signal at ACK cycle. This can be done by writing SINGLEBYTE bit before last data packet. Serial communication via 2-Wire serial bus is over when BUSLOST, STOPREQ or EOTREQ bit is set. In this case, the handler must read the STATUSREG after writing STATUSREG.

9.12.2.4 2-Wire SBI Slave Mode Address Register (SLAVEREG)

0x8006.000C

7	6	5	4	3	2	1	0
-	Slave Address[6:0]						

Bits	Type	Function
7	-	Reserved
6:0	R/W	Slave address of 2-Wire SBI itself When 2-Wire SBI is configured as slave device, this register contains the slave address of 2-Wire SBI itself.

9.12.2.5 2-Wire SBI Interrupt Mask Register (INTMASKREG)

0x8006.0010

7	6	5	4	3	2	1	0
-	-	-	TRANSREQ MASK	STOPREQ MASK	EOTREQ MASK	DATAREQ MASK	BUSLOST MASK

Bits	Type	Function
7-5	-	Reserved
4	R/W	TRANSREQ interrupt mask If this bit is set, TRANREQ interrupt is masked, so no interrupt is requested. 0 = TRANSREQ interrupt is enabled. (default) 1 = TRANSREQ interrupt is disabled.
3	R/W	STOPREQ interrupt mask If this bit is set, STOPREQ interrupt is masked, so no interrupt is requested. 0 = STOPREQ interrupt is enabled. (default) 1 = STOPREQ interrupt is disabled.
2	R/W	EOTREQ interrupt mask If this bit is set, EOTREQ interrupt is masked, so no interrupt is requested. 0 = EOTREQ interrupt is enabled. (default) 1 = EOTREQ interrupt is disabled.
1	R/W	DATAREQ interrupt mask If this bit is set, DATAREQ interrupt is masked, so no interrupt is requested. 0 = DATAREQ interrupt is enabled. (default) 1 = DATAREQ interrupt is disabled.
0	R/W	BUSLOST interrupt mask If this bit is set, BUSLOST interrupt is masked, so no interrupt is requested. 0 = BUSLOST interrupt is enabled. (default) 1 = BUSLOST interrupt is disabled.

9.12.2.6 2-Wire SBI Configuration Register (CONFIGREG)

0x8006.0014

7	6	5	4	3	2	1	0
RESTART	-	SOFT RESET	SINGLE BYTE	-	MULTI BYTE	FORCE STOP	START

Bits	Type	Function
7	R/W	RESTART condition (master only) When 2-Wire SBI is configured as master, setting this bit transmits a RESTART condition. 0 = No action is done. (default) 1 = RESTART condition is generated.
6	-	Reserved
5	R/W	Software reset command Setting this bit resets 2-Wire SBI module and this bit is auto-cleared. 0 = Normal operation. (default) 1 = Software reset command is issued.
4	R/W	Single byte is remained This bit is used in 2 cases. I) If only one byte of data is to be transferred, setting this bit with START bit completes serial communication. II) If more than one byte of data(n bytes) are to be transferred, set this bit after (n-1) bytes are transferred to terminate serial communication. 0 = Indicates more than one byte of data are remained when MULTIBYTE bit is set. (default) 1 = Serial communication is terminated after next DATA cycle.
3	-	Reserved
2	R/W	Multiple bytes transfer (master only) When more than one bytes are to be transferred, set this bit. 0 = Only one byte of data is to be transferred. (default) 1 = Multiple bytes are to be transferred.
1	R/W	Forces STOP condition If this bit is set, the STOP condition is transmitted during DATA cycle. That is data transfer is terminated abnormally. 0 = No action is done. (default) 1 = STOP condition is generated during DATA cycle.
0	R/W	START condition (master only) 2-Wire SBI is a master device and initiates a serial communication. 0 = 2-Wire SBI is a slave device or no action is done. (default) 1 = START condition is generated.

9.12.2.7 2-Wire SBI Baud Rate Control Register (BAUDREG)

0x8006.0018

7	6	5	4	3	2	1	0
BAUDRATE[7:0]							

Bits	Type	Function
7:0	R/W	Baud rate control The serial clock (SCL) rate is determined as F_{PCLK}/(2*(BAUDRATE+1)), where F_{PCLK} is the frequency of peripheral clock, PCLK. To operate correctly, the BAUDRATE value should be greater than 3.

Baud rate (decimal)	Divider value	SCL rate
03	F_{PCLK} /8	460 KHz
04	F_{PCLK} /10	369 KHz
10	F_{PCLK} /22	168 KHz
17	F_{PCLK} /36	102 KHz

9.12.3 Operation

Both SDA and SCL are bi-directional lines and connected to the positive supply voltage through pull-up resistors. The bus drivers of all 2-Wire SBI-compliant devices are open-drain or open-collector. This implements a wired-AND function which is essential to the operation of the interface. A low level on a 2-Wire SBI bus line is generated when one or more devices output a zero. A high level is output when all 2-Wire SBI devices release bus line, allowing the pull-up resistors to pull the line high. Below figure depicts general form of connecting more than two devices to the serial bus.

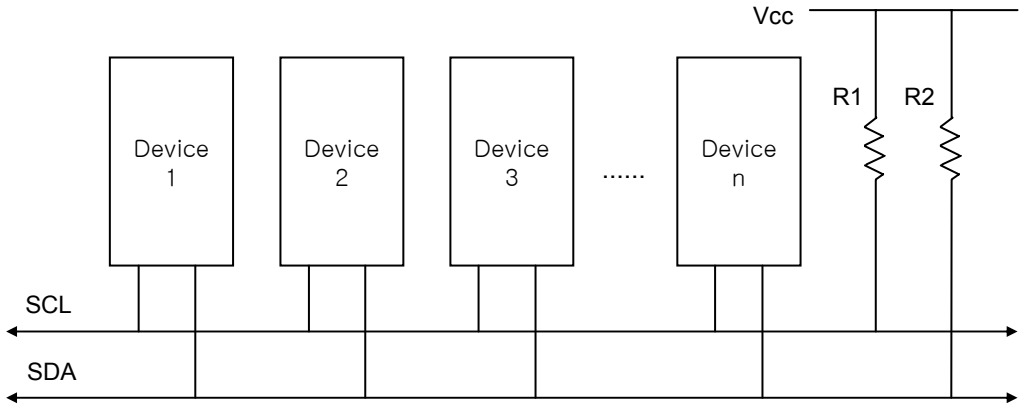


Figure 9-47. Connection of devices to the 2-Wire serial bus

9.12.3.1 *Transferring Bits on 2-Wire Serial Bus*

Each data bit transferred on 2-Wire serial bus is accompanied by a pulse on the clock line, SCL. The level of the data line must be stable when the clock line is high. The only exception to this rule is for generating start and stop conditions.

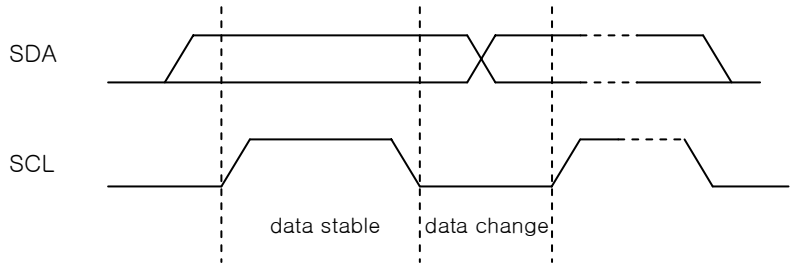


Figure 9-48. Data validity

9.12.3.2 *START and STOP Conditions of 2-Wire SBI*

The master initiates and terminates a data transfer. The serial communication is initiated when the master issues a START condition on the bus, and it is terminated when the master issues a STOP condition. Between a START and a STOP condition, the bus is considered busy, and no other master is allowed to try to gain the ownership of the bus.

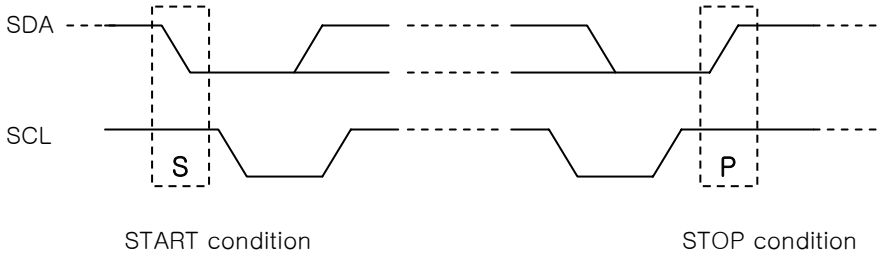


Figure 9-49. START and STOP conditions

Before STOP condition is detected, a master device can issue a RESTART condition which is identical to START condition and is symbolized as Sr.

9.12.3.3 *Multi-master bus systems, arbitration and synchronization*

The 2-Wire SBI protocol allows bus systems with several masters. Special concerns have been taken in order to ensure that transmissions will proceed as normal, even if more than two masters initiate transmission at the same time. In that case, two problems arise in multi-master bus systems :

- An algorithm must be implemented allowing only one of the masters to complete the transmission. All other masters must stop transmission when they know that they have lost the bus ownership. This process is called arbitration. When a contending master finds out that it has lost the arbitration process, it must immediately switch to slave mode to check whether it is being addressed by the winning master. The fact that multiple masters have started transmission at the same time should not be detectable to the slaves, i.e., the data being transferred on the bus must not be corrupted.
- Different masters may use different SCL frequencies. A scheme must be devised to synchronize the serial clocks from all masters, in order to let the transmission proceed.

The wired-ANDing of the bus lines is used to solve both these problems. The serial clocks from all masters will be wired-ANDed, yielding a combined clock with a high period equal to the one from the master with the shortest high period. The low period of the combined clock is equal to the low period of the master with the longest low period. Note that all masters checks the SCL line, effectively starting to count their SCL high and low time-out periods when the combined SCL line goes high or low, respectively.

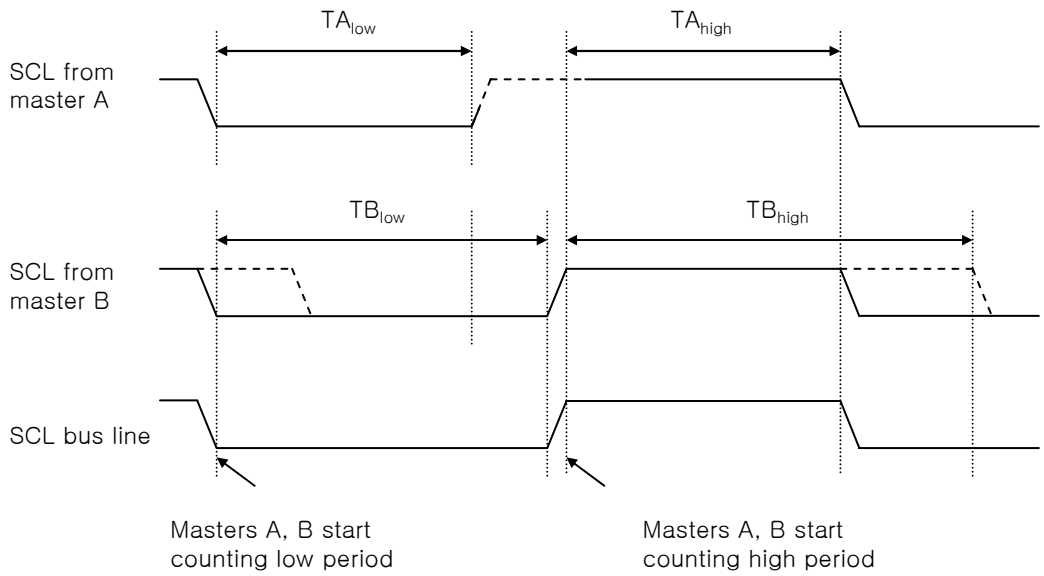


Figure 9-50. SCL synchronization between multiple masters

Arbitration is carried out by all masters continuously monitoring the SDA line after outputting data. If the value read from the SDA line does not match the value the master had output, it has lost the arbitration. Note that a master can only lose arbitration when it outputs a high SDA value while another master outputs a low value. The losing master must immediately go to slave mode, checking if it is being

addressed by the winning master. The SDA line should be left high, but losing masters are allowed to generate a clock signal until the end of the current data or address packet. Arbitration will continue until only one master remains, and this may take many bits. If several masters are trying to address the same slave, arbitration will continue into the data packet.

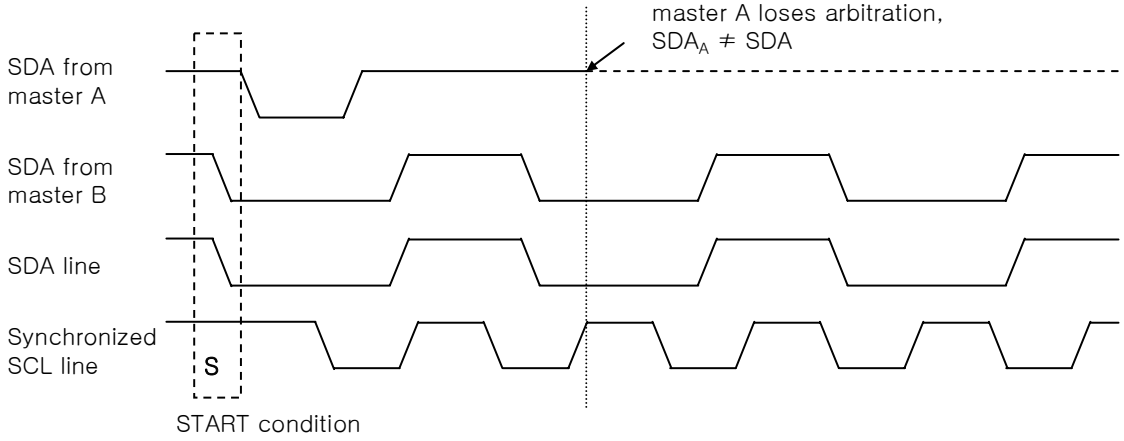


Figure 9-51. Arbitration between two masters

During serial communication, the arbitration procedure is still in progress at the moment when a RESTART condition or a STOP condition is transmitted to the serial bus. If it's possible for such a situation to occur, the masters involved must send this RESTART condition or STOP condition at the same position in the format frame. In other words, arbitration is not allowed between :

- A RESTART condition and a data bit
- A STOP condition and a data bit
- A RESTART condition and a STOP condition.

Slaves are not involved in the arbitration procedure.

9.12.3.4 Serial communication

The data transfer on 2-Wire serial bus is performed as depicted in the following figure. First, the master device examines the serial bus lines are available. When the serial bus is not busy, the master transmits a START condition and the first data packet which are composed of 7 address bits and, one READ/WRITE control bit. And then, the slave device addressed by the master device acknowledges by pulling SDA line low in the ninth SCL cycle (ACK cycle). If the addressed slave device does not exist or is busy doing other tasks, the serial communication is terminated and the SDA line is left high in the ACK cycle.

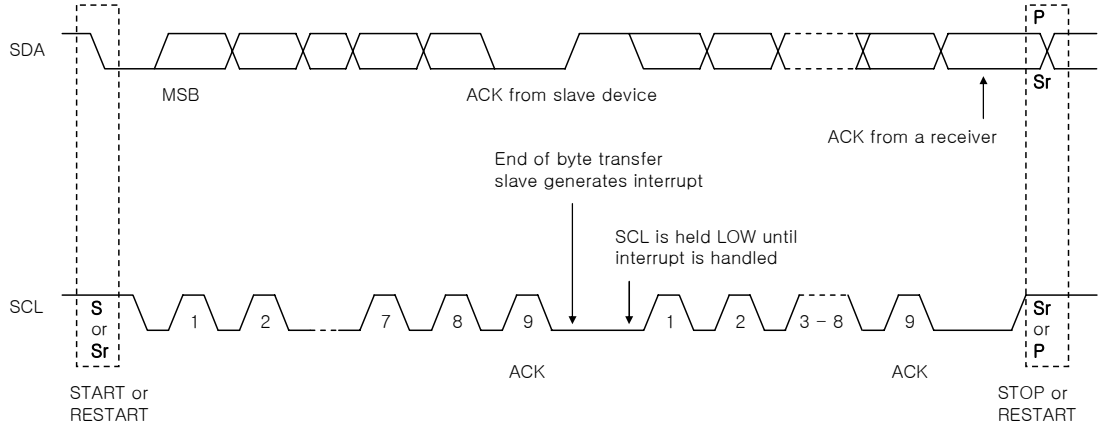


Figure 9-52. Address and data packet of 2-Wire SBI

After data transfer is ended, the master transmits a STOP condition or RESTART condition. Note that between a START and a STOP condition, all data packet is composed of 8 bits data and one ACK bit. The first data packet after a START or RESTART condition is an address packet which is composed of 7 address bits and one R/W control bit. And all address and data packets are transmitted MSB first.

A transfer is basically consists of a START condition, a address packet, one or more data packets and a STOP condition. ADDRESS cycle is the cycle while address packet is transferred and DATA cycle is the cycle while data packet is transferred. And address packet is the 1st 9-bit data after START condition, and data packets are 9-bit data consisting of 8-bit data byte and one bit ACK

.

Either in ADDRESS or DATA cycle, the master generates the clock and START and STOP conditions, while the receiver is responsible for acknowledging the reception. An Acknowledge, ACK is signaled by the receiver pulling the SDA line low during the ninth SCL cycle. If the receiver leaves the SDA line high, a NACK is signaled. When the receiver has received the last byte, or for some reason cannot receive any more bytes, it should inform the transmitter by sending a NACK after the final byte.

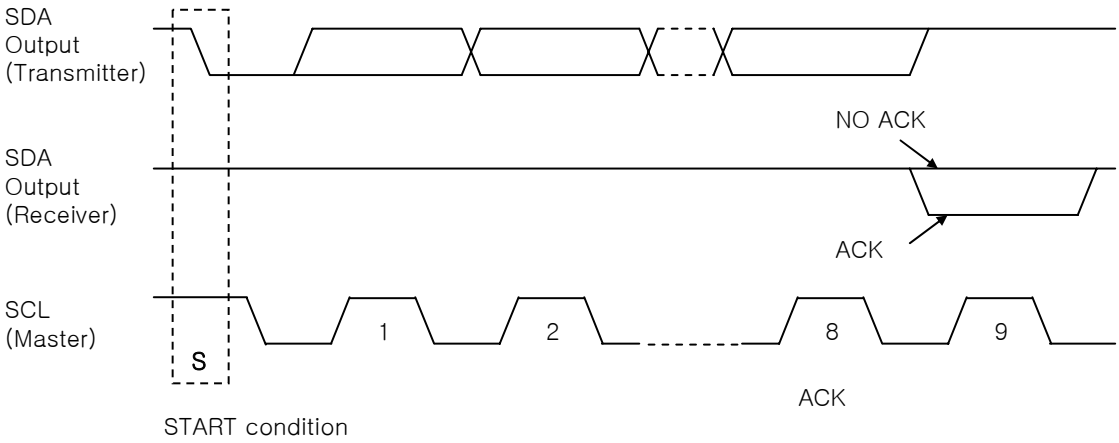


Figure 9-53. ACK signal generation

In HMS30C7210, the address packet comes from TARGETREG when configured as master mode. And there are 4 operating modes internally according to transfer direction and bus mastership. The individual operating sequence is stated below. All cases are stated assuming interrupt mode operation.

Master transmitter

- Decide target slave device to which 2-Wire SBI wants to transmit data and write 7-bit address and 1-bit R/W control bit to TARGETREG. The value written in TARGETREG is the 1st 8-bit data (address packet) to be transmitted.
- Configure BAUDREG to select SCL frequency.
- Write 2nd 8-bit data (1st data packet) to transmit to DATAREG.
- Enable interrupt sources by writing INTMASKREG. Assume all interrupt sources are enabled through out this sequence.
- Generate START condition. This is done by setting both START bit and MULTIBYTE bit in CONFIGREG. If only one byte of data is need to be transmitted, set both the START bit and SINGLEBYTE bit in CONFIGREG. In this case, an EOT interrupt is requested after 1st data packet and the below steps are needless.
- Wait ACK from addressed slave after transmitting address packet consisting of 7-bit address and 1-bit R/W control bit. Step 6 is done by 2-Wire SBI unit not by software.
- If 2-Wire SBI receives an ACK for address packet, the 1st data packet is transmitted and DATAREQ interrupt is requested. The interrupt handler prepares next data to transmit and write STATUS register to clear interrupt and proceed to DATA cycle. And then wait next DATAREQ interrupt. If no ACK is signaled for address packet, serial communication is terminated and BUSLOST interrupt is requested. In this case, read STATUSREG to release serial bus after writing STATUSREG.
- If 2-Wire SBI receives an ACK for data packet, DATAREQ interrupt is requested. The interrupt handler writes next data to transmit into DATAREG and clears interrupt by writing STATUSREG. Termination of serial communication is done in 2 ways. One method is by software decision. If there are n bytes of data packet to transmit, software sets SINGLEBYTE bit in CONFIGREG after (n-1)th data packets are transmitted. While changing CONFIGREG, START bit must preserve previous value and MULTIBYTE bit must be cleared simultaneously. The other method is based on ACK signal. If no ACK for data packet is received, serial communication is terminated and an EOT interrupt is requested. In both cases, read STATUSREG to release serial bus after writing STATUSREG. Repeat step 8 until serial communication is over.

The above steps are normally used when 2-Wire SBI is configures as master transmitter. Even if ACK for a data packet is received, serial communication can be terminated by setting STOP bit in CONFIGREG. The next figure depicts above steps.

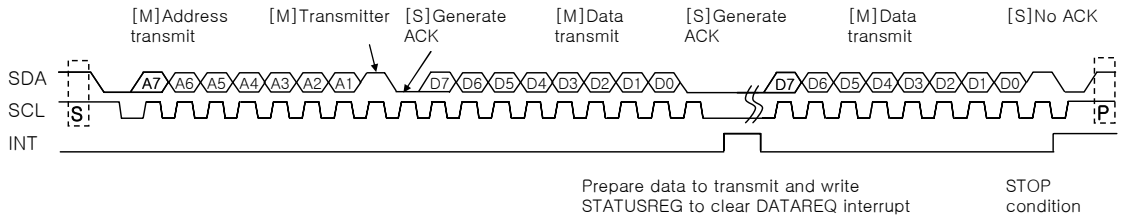


Figure 9-54. Waveform when 2-Wire SBI is master transmitter

In the above figure, the symbol INT represents an interrupt from 2-Wire SBI and [M] represents signal generated from master, [S] represents signal generated by slave.

Master receiver

- Decide target slave device from which 2-Wire SBI wants to receive data and write 7-bit address and 1-bit R/W control bit to TARGETREG. The value written in TARGETREG is the 1st 8-bit data (address packet) to be transmitted.
- Configure BAUDREG to select SCL frequency.
- Enable interrupt sources by writing INTMASKREG. Assume all interrupt sources are enabled through out this sequence.
- Generate START condition. This is done setting both START bit and MULTIBYTE bit in CONFIGREG. If only one byte of data is need to be received, set both the SIGNLEBYTE bit and START bit in CONFIGREG. In this case, an EOT interrupt is requested after 1st data packet and the below steps are needless.
- Wait ACK from addressed slave after transmitting address packet consisting of 7-bit address and 1-bit R/W control bit.
- If 2-Wire SBI receives an ACK for address packet, the 1st data packet is received and DATAREQ interrupt is requested. The interrupt handler prepares next data to transmit and write any value STATUSREG to clear interrupt and proceed to DATA cycle. And then wait next DATAREQ interrupt. If no ACK is signaled for address packet, serial communication is terminated and BUSLOST interrupt is requested. In this case, read STATUSREG to release serial bus after writing STATUSREG.
- When DATAREQ interrupt is requested, 2-Wire SBI reads the DATAREG which contains the recently received 8-bit data packet. If there're more data to be received from the slave, the interrupt handler need only to clear interrupt by writing the STATUSREG. But if next data packet is the last data packet or 2-Wire SBI can't receive more than one data for some reason, 2-Wire SBI signals no ACK at the next data packet by setting the SINGLEBYTE bit in CONFIGREG. While changing CONFIGREG, START bit must preserve previous value and MULTIBYTE bit must be cleared simultaneously. When 2-Wire SBI compliant transmitter does not receive an ACK at ACK cycle, the serial communication ends automatically and the 2-Wire SBI of HMS30C7210 request an EOT interrupt. In this case, read STATUSREG to free serial bus after writing STATUSREG. Repeat step 7 until serial communication is over.

The above steps are normally used when 2-Wire SBI is configures as master receiver. The next figure depicts above steps.

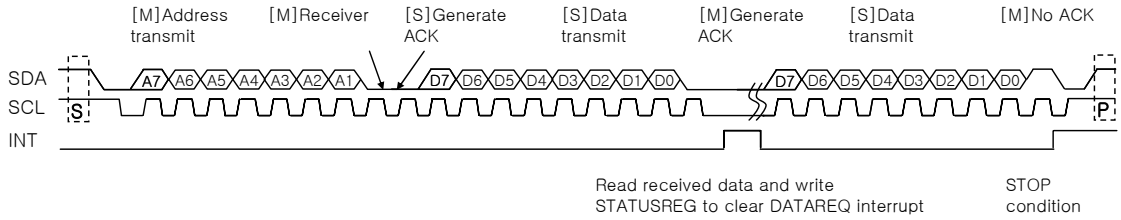


Figure 9-55. Waveform when 2-Wire SBI is master receiver

In the above figure, the symbol INT represents an interrupt from 2-Wire SBI and [M] represents signal generated from master, [S] represents signal generated by slave.

Slave transmitter

- Enable interrupt sources by writing INTMASKREG. Assume all interrupt sources are enabled through out this sequence. By default, TRANSREQ interrupt must be enabled to use interrupt mode.
- Wait for START condition from external master device.
- If 7-bits address of address packet matches SLAVEREG, ACK signal for address packet is transmitted and TRANSREQ interrupt is requested. When TRANSREQ interrupt is requested, read the STATUSREG and verify that the master requires data from 2-Wire SBI by checking the TRANSMITTER bit in STATUSREG.
- Write the 1st data into DATAREG and clear interrupt by writing STATUSREG.
- After transmitting data packet, 2-Wire SBI checks the ACK signal at ACK cycle. If no ACK is received, the serial communication ends and an EOT interrupt is requested. In this case, read STATUSREG to release serial bus after writing STATUSREG. If ACK signal is received, 2-Wire SBI requests an DATAREQ interrupt. In this case, write the next data to transmit into DATAREG and clear interrupt by writing the STATUSREG. Repeat this step until serial communication is over.

The above steps are normally used when 2-Wire SBI is configures as slave transmitter. The next figure depicts above steps.

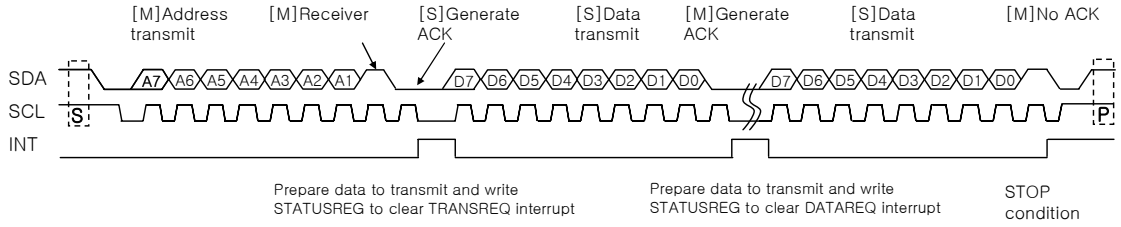


Figure 9-56. Waveform when 2-Wire SBI is slave transmitter

In the above figure, the symbol INT represents an interrupt from 2-Wire SBI and [M] represents signal generated from master, [S] represents signal generated by slave.

Slave receiver

- Enable interrupt sources by writing INTMASKREG. Assume all interrupt sources are enabled through out this sequence. By default, TRANSREQ interrupt must be enabled to use interrupt mode.
- Wait for START condition from external master device.
- If 7-bits address of address packet matches SLAVEREG, ACK signal for address packet is transmitted and TRANSREQ interrupt is requested. When TRANSREQ interrupt is requested, read the STATUSREG and verify that the master wants to transmit data to 2-Wire SBI by checking the TRANSMITTER bit in STATUSREG.
- If no more than one data is acceptable, set the SINGLEBYTE bit in CONFIGREG to transmit no ACK at next data packet, then an EOT interrupt is requested and serial communication is over. In this case, step 5 is needless and read the STATUSREG to release the serial bus after clearing the interrupt by writing the STATUSREG. Or 2-Wire SBI is capable of more than one data packet, just clear interrupt by writing STATUSREG and receive 1st data.
- When DATAREQ interrupt is requested, 2-Wire SBI reads the DATAREG which contains the recently received 8-bit data packet. If there're more data to be received from the master, the interrupt handler need only to clear interrupt by writing the STATUSREG. But if next data packet is the last data packet or 2-Wire SBI can't receive more than one data for some reason, 2-Wire SBI signals no ACK at the next data packet by setting the SINGLEBYTE bit in CONFIGREG. When 2-Wire SBI compliant transmitter does not receive an ACK at ACK cycle, the serial communication ends automatically and the 2-Wire SBI of HMS30C7210 request an EOT interrupt. In this case, read STATUSREG to release serial bus after writing STATUSREG. Repeat step 5 until serial communication is over.

The above steps are normally used when 2-Wire SBI is configures as slave receiver. The next figure depicts above steps.

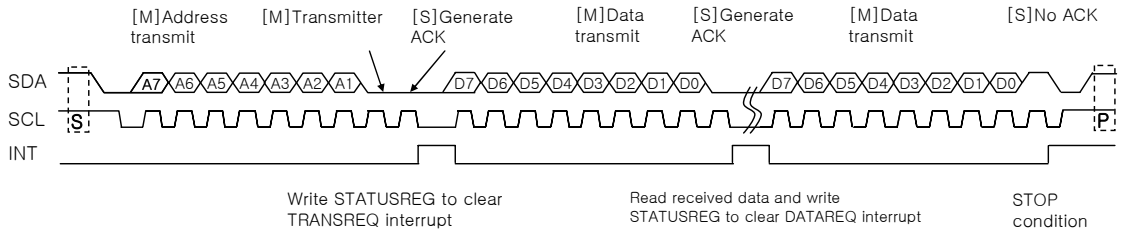


Figure 9-57. Waveform when 2-Wire SBI is slave receiver

In the above figure, the symbol INT represents an interrupt from 2-Wire SBI and [M] represents signal generated from master, [S] represents signal generated by slave.

9.13 Matrix Keyboard Interface Controller

The Matrix keyboard interface controller is an AMBA slave module that connects to the Advanced Peripheral Bus (APB). For more information about AMBA, please refer to the AMBA Specification (ARM IHI 0001).
The interface controller is designed to communicate with the external keyboard matrix. The keyboard interface uses the pins KSCANI [5:0] and KSCANO [5:0]. It is possible to select one of three scan clock frequencies.

The main features of keyboard controller are :

- Controllable scanning frequency
- Maximum 6x6 keyboard matrix is supported
- Key value is stored in KBVR0/1

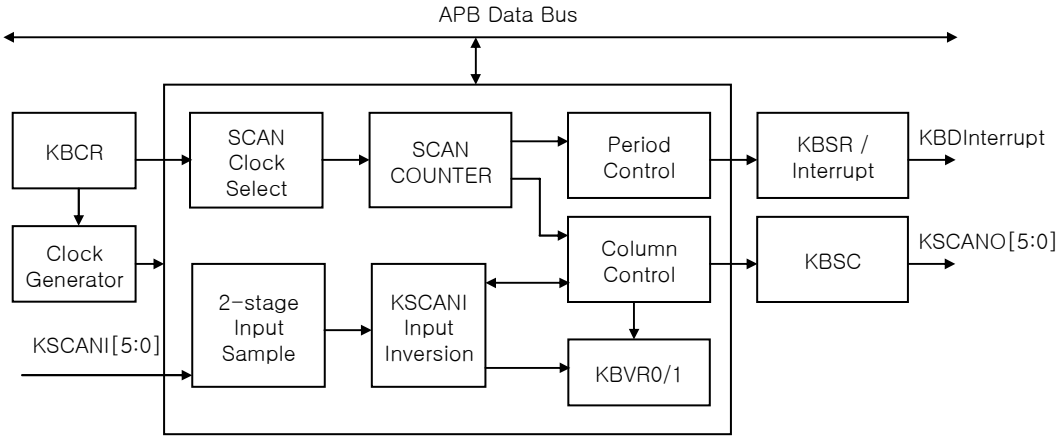


Figure 9-58. Block diagram of keyboard controller

9.13.1 External Signals

Pin Name	Type	Description
KSCANO [5:0]	O	Column enable signals to keyboard matrix Key input is valid only when KSCANO pin is LOW. The outputs of KSCANO pins act like ring counter so as to cover all columns of keyboard matrix. If pins are used for keyboard function, pull-up resistors need to be connected.
KSCANI [5:0]	I	Row inputs from keyboard matrix If pins are used for keyboard function, pull-up resistors need to be connected. Normally each KSCANI line maintains HIGH level because of pull-up resistor so, LOW input is detected as “key pressed”.

Refer to Figure 2-1. 208 Pin diagram.

9.13.2 Registers

Address	Name	Width	Default	Description
0x8006.1000	KBCR	8	0x0	Keyboard Configuration Register
0x8006.1004	KBSC	6	0x0	Keyboard Scan Out Register
0x8006.100C	KBVR0	32	0x0	Keyboard Value Register 0
0x8006.1010	KBVR1	16	0x0	Keyboard Value Register 1
0x8006.1018	KBSR	2	0x0	Keyboard Status Register

Table 9-17. Matrix Keyboard Interface Controller Register Summary

9.13.2.1 Keyboard Configuration Register (KBCR)

0x8006.1000

7					2	1	0
SCAN ENABLE					nPOWER DOWN	CLKSEL	

Bits	Type	Function
7	R/W	Key input scanning enable Setting this bit enables key input scanning coming from KSCANI pins. Note that both SCANENABLE and nPOWERDOWN bits must be set to start key input scanning. It is recommended that both SCANENABLE and nPOWERDOWN bits are cleared to stop key input scanning. It is software's responsibility to de-bounce the key pressed information. Keyboard interrupt is generated in all PMU states except deep sleep. 0 = Stops key input scanning. 1 = Starts key input scanning.
6:3	-	Reserved. Keep these bits to zero.
2	R/W	Power down mode (Active low) Activates keyboard controller module by supplying PCLK. 0 = Indicates power down mode and internal operating clock signal is always '0'. (default) 1 = Clock generator unit supplies incoming PCLK to keyboard controller module.
1:0	R/W	Scan clock select bits This controls the operating clock of scanning matrix keyboard.

Value	Scan clock source	Scan Rate
00	Reserved	Not available
01	PCLK / 128 (28KHz)	138 times / sec
10	PCLK / 256 (14KHz)	69 times / sec
11	PCLK / 512 (7KHz)	34 times / sec

9.13.2.2 Keyboard Scan Out Register(KBSC)

0x8006.1004

		5	4	3	2	1	0
		SCANOUT					

Bits	R/W	Function
5	R	Indicates that 1st column is being scanned. When low, the pressed KSCANI inputs are stored in KBVR0[29:24]. This bit is directly connected to KSCANO[5]. 0 = 1st column is being scanned. (default) 1 = 1st column is not being scanned.
4	R	Indicates that 2nd column is being scanned. When low, the pressed KSCANI inputs are stored in KBVR0[21:16]. This bit is directly connected to KSCANO[4]. 0 = 2nd line will be scanned. (default) 1 = 2nd column is not being scanned.
3	R	Indicates that 3rd column is being scanned. When low, the pressed KSCANI inputs are stored in KBVR0[13:8]. This bit is directly connected to KSCANO[3]. 0 = 3rd line will be scanned. (default) 1 = 3rd column is not being scanned.
2	R	Indicates that 4th column is being scanned. When low, the pressed KSCANI inputs are stored in KBVR0[5:0]. This bit is directly connected to KSCANO[2]. 0 = 4th line will be scanned. (default) 1 = 4th column is not being scanned.
1	R	Indicates that 5th column is being scanned. When low, the pressed KSCANI inputs are stored in KBVR1[13:8]. This bit is directly connected to KSCANO[1]. 0 = 5th line will be scanned. (default) 1 = 5th column is not being scanned.
0	R	Indicates that 6th column is being scanned. When low, the pressed KSCANI inputs are stored in KBVR1[5:0]. This bit is directly connected to KSCANO[0]. 0 = 6th line will be scanned. (default) 1 = 6th column is not being scanned.

9.13.2.3 Keyboard Value Register (KBVR0)

0x8006.100C

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

Bits	Type	Function
31:30	R	Reserved
29:24	R	The pressed KSCANI during KSCANO[5] is LOW. KSCANI[5:0] maps to KBVR0[29:26]. If any pin of KSCANI[5:0] is LOW, the corresponding bit position in KBVR0[29:26] becomes '1' . 0 = KSCANI input is pressed while KSCANO[5] is HIGH or no KSCANI input is pressed while KSCANO[5] is LOW. 1 = The corresponding KSCANI input is pressed.
23:22	R	Reserved
21:16	R	The pressed KSCANI during KSCANO[4] is LOW. KSCANI[5:0] maps to KBVR0[21:16]. If any pin of KSCANI[5:0] is LOW, the corresponding bit position in KBVR0[21:16] becomes '1' . 0 = KSCANI input is pressed while KSCANO[4] is HIGH or no KSCANI input is pressed while KSCANO[4] is LOW. 1 = The corresponding KSCANI input is pressed.
15:14	R	Reserved
13:8	R	The pressed KSCANI during KSCANO[3] is LOW. KSCANI[5:0] maps to KBVR0[13:8]. If any pin of KSCANI[5:0] is LOW, the corresponding bit position in KBVR0[13:8] becomes '1' . 0 = KSCANI input is pressed while KSCANO[3] is HIGH or no KSCANI input is pressed while KSCANO[3] is LOW. 1 = The corresponding KSCANI input is pressed.
7:6	R	Reserved
5:0	R	The pressed KSCANI during KSCANO[2] is LOW. KSCANI[5:0] maps to KBVR0[5:0]. If any pin of KSCANI[5:0] is LOW, the corresponding bit position in KBVR0[5:0] becomes '1' . 0 = KSCANI input is pressed while KSCANO[2] is HIGH or no KSCANI input is pressed while KSCANO[2] is LOW. 1 = The corresponding KSCANI input is pressed.

9.13.2.4 Keyboard Value Register (KBVR1)

0x8006.1010

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		5th column KSCANI [5:0]								6th column KSCANI [5:0]					

Bits	Type	Function
15:14	R	Reserved
13:8	R	The pressed KSCANI during KSCANO[1] is LOW. KSCANI[5:0] maps to KBVR1[13:8]. If any pin of KSCANI[5:0] is LOW, the corresponding bit position in KBVR0[13:8] becomes '1' . 0 = KSCANI input is pressed while KSCANO[1] is HIGH or no KSCANI input is pressed while KSCANO[1] is LOW. 1 = The corresponding KSCANI input is pressed.
7:6	R	Reserved
5:0	R	The pressed KSCANI during KSCANO[0] is LOW. KSCANI[5:0] maps to KBVR1[5:0]. If any pin of KSCANI[5:0] is LOW, the corresponding bit position in KBVR0[5:0] becomes '1' . 0 = KSCANI input is pressed while KSCANO[0] is HIGH or no KSCANI input is pressed while KSCANO[0] is LOW. 1 = The corresponding KSCANI input is pressed.

9.13.2.5 Keyboard Status Register (KBSR)

0x8006.1018

						1	0
						WAKEUP	KEYINTR

Bits	Type	Function
7:2	-	Reserved
1	R	Wake up status This bit is set if any key is pressed when SCANENABLE in KBCR is LOW. This bit is a source of keyboard interrupt, which is generated in all PMU states except deep sleep mode. This bit is cleared when non-zero value is written in this register. 0 = Key scanning is enabled or no key is pressed when SCANENABLE is LOW. (default) 1 = There is at least one point pressed at matrix keyboard when SCANENABLE is LOW.
0	R	End of one scan period If one scan period is over, this flag is set and KBVR0/1 contains all the pressed points of matrix keyboard. When this bit is set, a keyboard interrupt is requested. This bit is cleared when non-zero value is written in this register. 0 = KBSR is cleared or key scanning is going on. (default) 1 = Indicates that KBVR0/1 are loaded with the value of keys pressed and software should read KBVR0/1 registers.

9.13.3 Operation

9.13.3.1 Conceptual configuration of keyboard matrix

Keyboards use a matrix with the rows and columns made up of wires. Each key acts like a switch. When a key is pressed, a column wire(called KSCANO) makes contact with a row wire(called KSCANI) and completes a circuit. The keyboard controller detects this closed circuit and registers it as a key press. Here is a simple keyboard matrix. The symbol KSCANO and KSCANI are same as those of HMS30C7210. At reset or when key scanning is not enabled, all KSCANO lines of HMS30C7210 are LOW to generate WAKEUP event in KBSR.

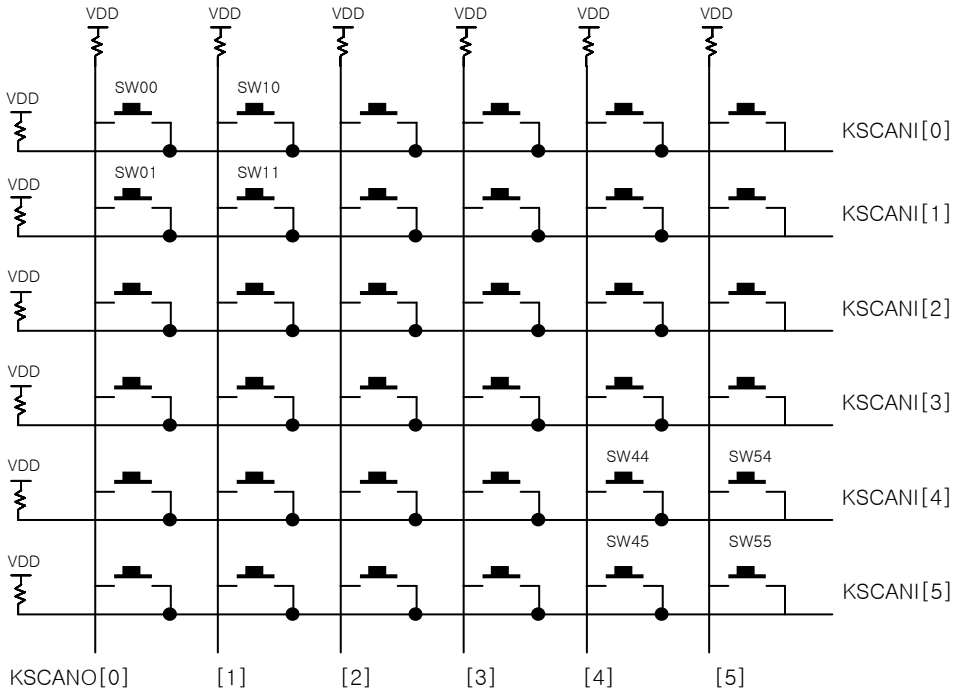


Figure 9-59. Keyboard matrix configuration

The above keyboard matrix works ‘cause only one of KSCANO lines are LOW while key scanning is enabled. If a key SW00 is pressed when KSCANO[0] is LOW, the keyboard controller detects that KSCANI[0] input is active. Similarly If two keys SW10, SW11 are pressed when KSCANO[1] is LOW, the controller detects that KSCANI[1:0] inputs are active. Note that pull-up resistors are connected to KSCANI and KSCANO lines. If no switch is pressed, KSCANI maintain HIGH level and the controller knows that there’s no key input. If any switch is pressed, the corresponding KSCANI line is changed to LOW level and the controller knows that there are some keys pressed and stores the position of KSCANI to KBVR0/1 register. The pressed key position is stored as ‘1’ in KBVR0/1.

9.13.3.2 KSCANO output timing

When SCANENABLE is set, the outputs of keyboard controller, KSCANO[5:0], acts like ring counter. In other words, during 1 scan period only one of KSCANO lines is LOW at one time(Column period). This enables KSCANI[n] is detected as unique switch during 1 scan period.
The following figure shows the output waveform of KSCANO lines. Once SCANENABLE in KBCR is set according to scan rate which is controlled by CLKSEL, KSCANO[5] is LOW at 1st column period and then KSCANO[4], KSCANO[3], KSCANO[2], KSCANO[1], KSCANO[0] are LOW periodically. In 6x6 matrix configuration, only 6 column periods are needed in one scan period. But there are 8 column periods and this makes no problem using keyboard matrix.

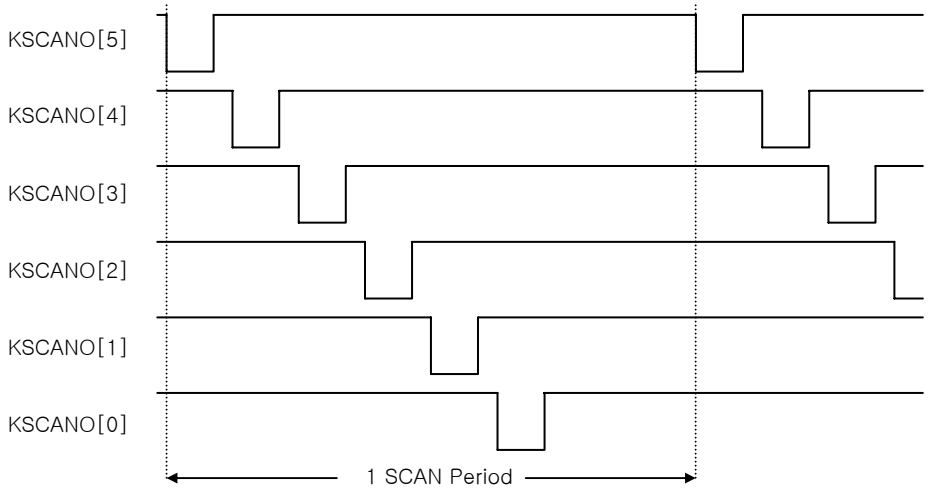


Figure 9-60. KSCANO output timing

9.13.3.3 Scanning rate selection and clock divider

The scan rate is controlled by CLKSEL in KBCR.

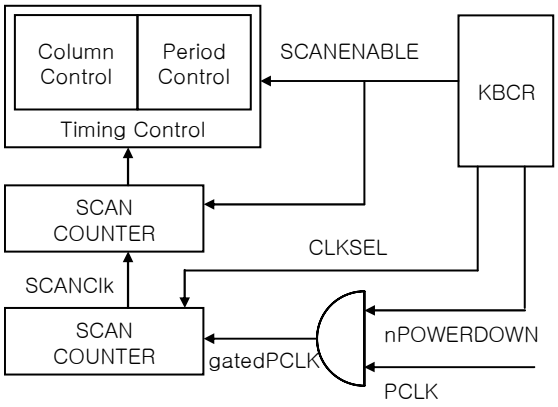


Figure 9-61. Clock divider of keyboard controller

Like other slow APB peripherals, keyboard controller is clocked by PCLK. Key scanning is much like mechanic process and PCLK is very fast for that purpose. So the main clock of SCAN COUNTER unit which is used to control column and scan period is SCANClk controlled by CLKSEL bits.

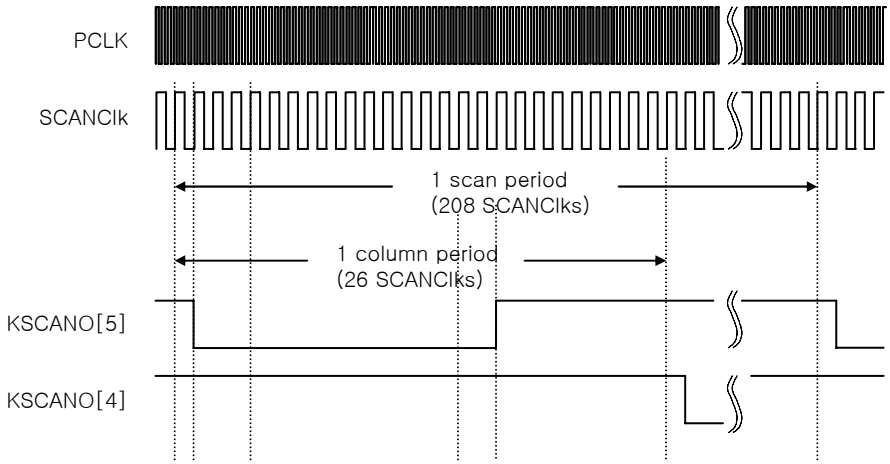


Figure 9-62. Key scan period and column period

SCANClk is achieved from output of flip-flops(SCANCOUNTER) which are clocked by PCLK. These flip-flops are asynchronously cleared when SCANENABLE is '0', and increments by one when SCANENABLE is '1'. The SCANCOUNTER is 9-bit(8 to 0) counter and the output is the source of SCANClk.
1 column is composed of 26 SCANClks and 1 scan period is composed of 8 columns, therefore 1 scan period is composed of 208 SCANClks.
The following table shows how the scan rate is calculated from CLKSEL. For example, CLKSEL is "01", the output of 7th flip-flop of SCANCOUNTER counter is the source of

SCANClk, so DIVIDER value becomes “128” because output of 7th flip-flop makes 1 clock pulse after 128 PCLKs.

CLKSEL	DIVIDER	$F_{SCANClk} = F_{PCLK}$ / DIVIDER	SCAN RATE
01	128	Approximately 28 KHz	138 times / sec
10	256	Approximately 14 KHz	69 times / sec
11	512	Approximately 7KHz	34 times / sec

Table 9-18. Scan rate calculation from CLKSEL

9.13.3.4 Scanning sequence of key inputs

As stated previously KSCANI lines are detected pressed only when corresponding KSCANO line is LOW.

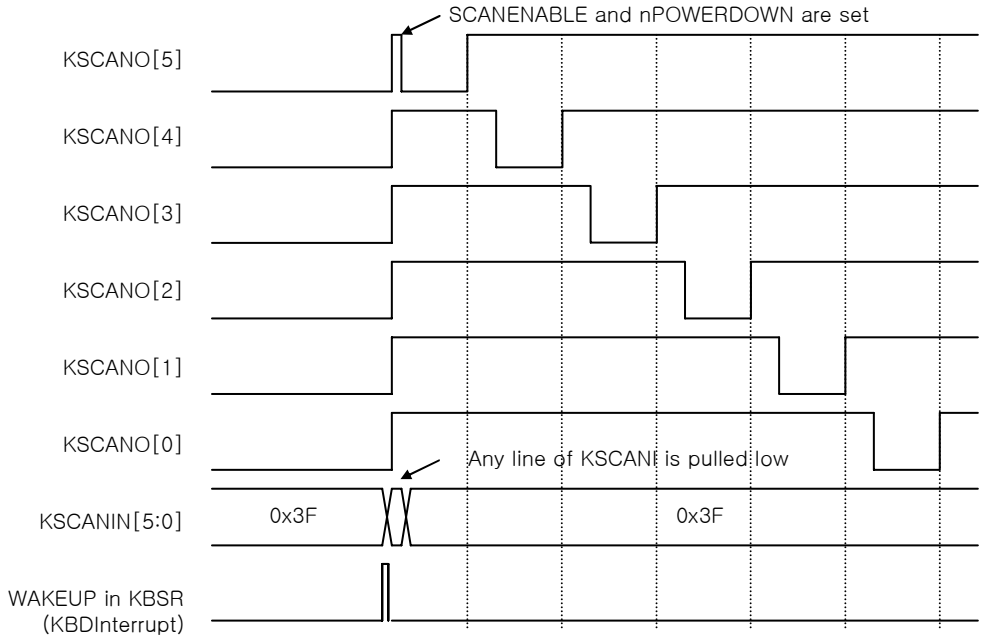


Figure 9-63. Wakeup interrupt & Key scanning enabled

At reset or when key scanning is not enabled, all KSCANO lines are LOW. If any switch is pressed WAKEUP in KBSR is set and interrupt is requested. The keyboard interrupt handler usually enables key scanning by setting both SCANENABLE and nPOWERDOWN in KBCR. Simultaneously KSCANO lines start making column period as in the previous figure.

The following figure shows example of interrupt handler routine related to keyboard interrupt.

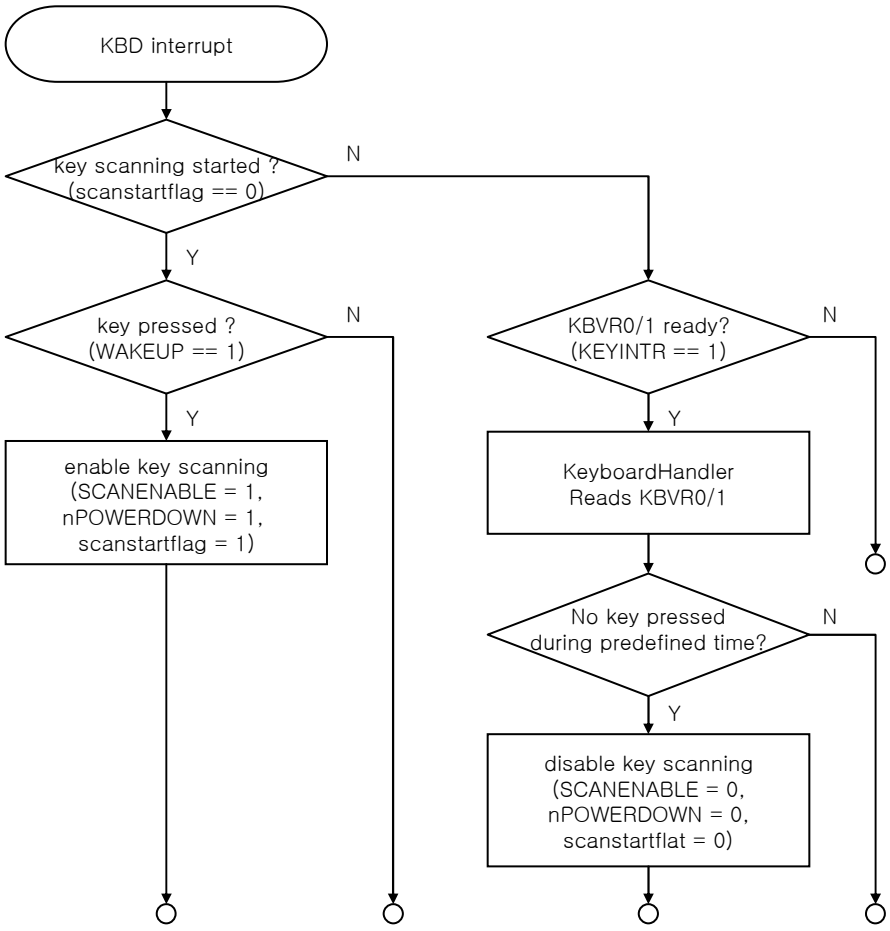


Figure 9-64. A flow chart of setting keyboard controller

The above flow chart can be summarized as follows :

- See if key scanning is started already by checking scanstartflag. If scanstartflag is not set, go to step 4.
- Check WAKEUP in KBSR. (interrupt)
- Set SCANENABLE, nPOWERDOWN and scanstartflag to enable key scanning and exit handler routine. (KBCR)
- Check KEYINTR in KBSR. (interrupt)
- Read KBVR0/1.
- If no key is pressed for predefined time, disable key scanning and exit handler routine.
- To continue key scanning, just exit handler routine and wait next keyboard interrupt.

The following figure shows internal timing diagram of keyboard controller.

Note that LOW value of KSCANI is detected as “key pressed” and stored in KBVR0/1 as binary ‘1’. The KSCANI lines are sampled 2 times during LOW phase of each KSCANO line and if 2 sampled values are different, the KSCANI line is considered as not pressed. 2 times sampling is simplified de-bouncing for input pin KSCANI lines.

When one scan period is over, the KEYINTR bit in KBSR is set and an interrupt is requested. Because the timing of KSCANO is periodic after SCANENABLE is set, Software must handle the requested interrupt by reading KBVR0/1 before KSCANO[5] of next scan period makes an rising edge. This time limit is symbolized as t_{INT} In the following figure. As 26 SCANClks makes one column period, t_{INT} is approximately 3.7ms when CLKSEL is “01”.

CLKSEL	$F_{SCANClk}$	t_{INT}
01	28 KHz	Approximately 0.9 ms
10	14 KHz	Approximately 1.8 ms
11	7 KHz	Approximately 3.7 ms

Table 9-19. Estimated t_{INTR} according to CLKSEL

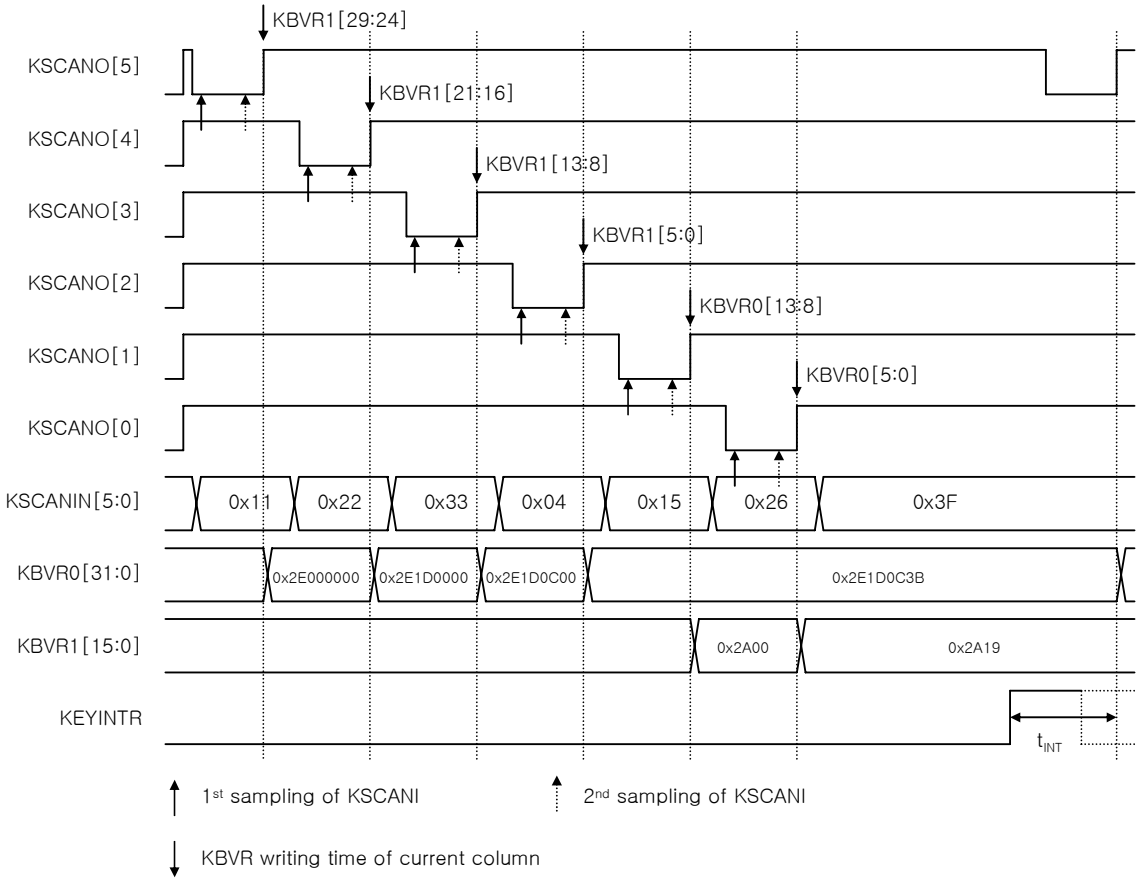


Figure 9-65. KBVR0/1 write timing

9.13.3.5 *Usage and restrictions*

The maximum size of keyboard matrix that can be used is 6x6(KSCANI x KSCANO). But there are some restrictions for KSCANI pins. The restrictions result in minimum matrix size of 4x1.

Before using keyboard matrix, pull-up resistors must be connected to KSCANI and KSCANO lines that are used for keyboard function.

Using some of KSCANO :
Even if keyboard matrix is connected to HMS30C7210, each KSCANO line can be configured for GPIO. The below table shows possible configuration for KSCANO pins. The 'O' means KSCANO[n] can be used for that function (Keyboard or GPIO) in the table where n is 0,1,2,3,4 or 5.

	KSCANO[0]	KSCANO[1]	KSCANO[2]	KSCANO[3]	KSCANO[4]	KSCANO[5]
Keyboard	O (pull-up)	O (pull-up)	O (pull-up)	O (pull-up)	O (pull-up)	O (pull-up)
GPIO	O	O	O	O	O	O

Table 9-20. Possible configuration of KSCANO pins when keyboard matrix is connected

In the following figure, KSCANO[3:2] are used for GPIOs and only KSCANO[5:4, 1:0] are used for keyboard function. This means that switches sw3x and sw2x(see keyboard matrix configuration figure) are ignored and not stored in KBVR0/1 when pressed. Note that KSCANO[3:2] are always HIGH in the figure but these pins can change level 'cause these pins are not connected to keyboard matrix.

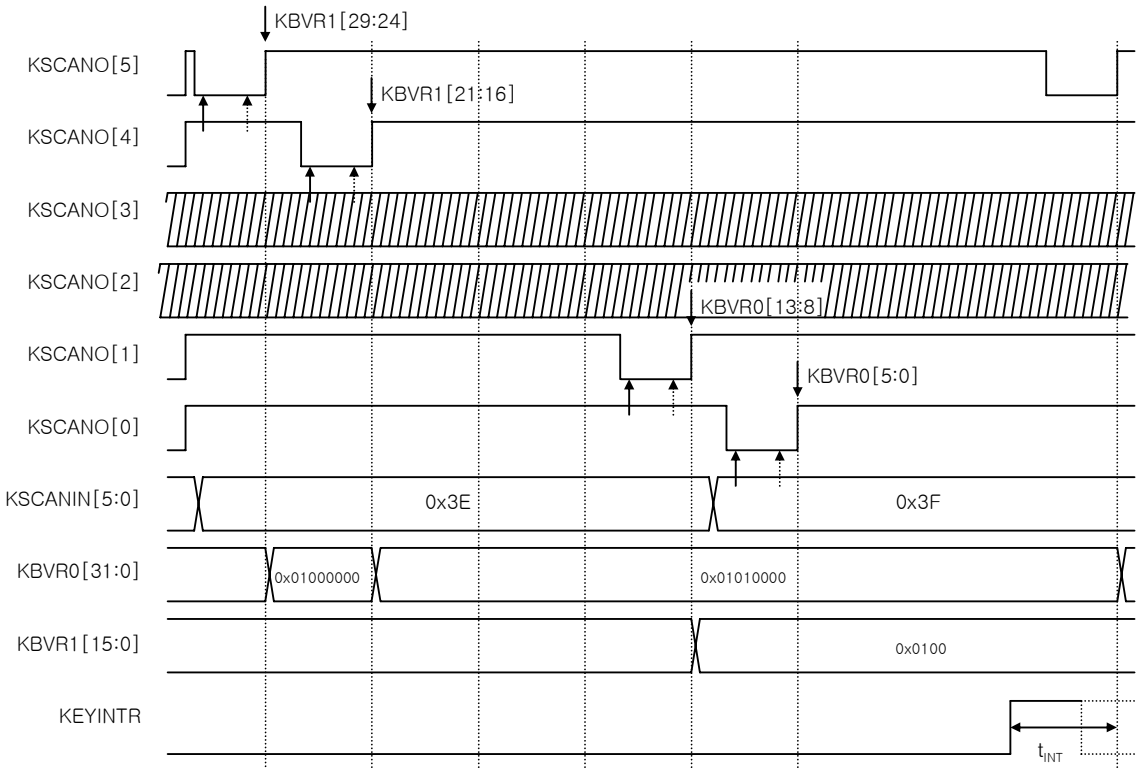


Figure 9-66. KSCANO[3:2] are configured for GPIO

Using some of KSCANI :
Not like KSCANO, some KSCANI pins must be configured for keyboard function to use keyboard matrix. That is, KSCANI[3:0] must be configured for keyboard function. But KSCANI[5:4] can be configured for GPIO or keyboard function.

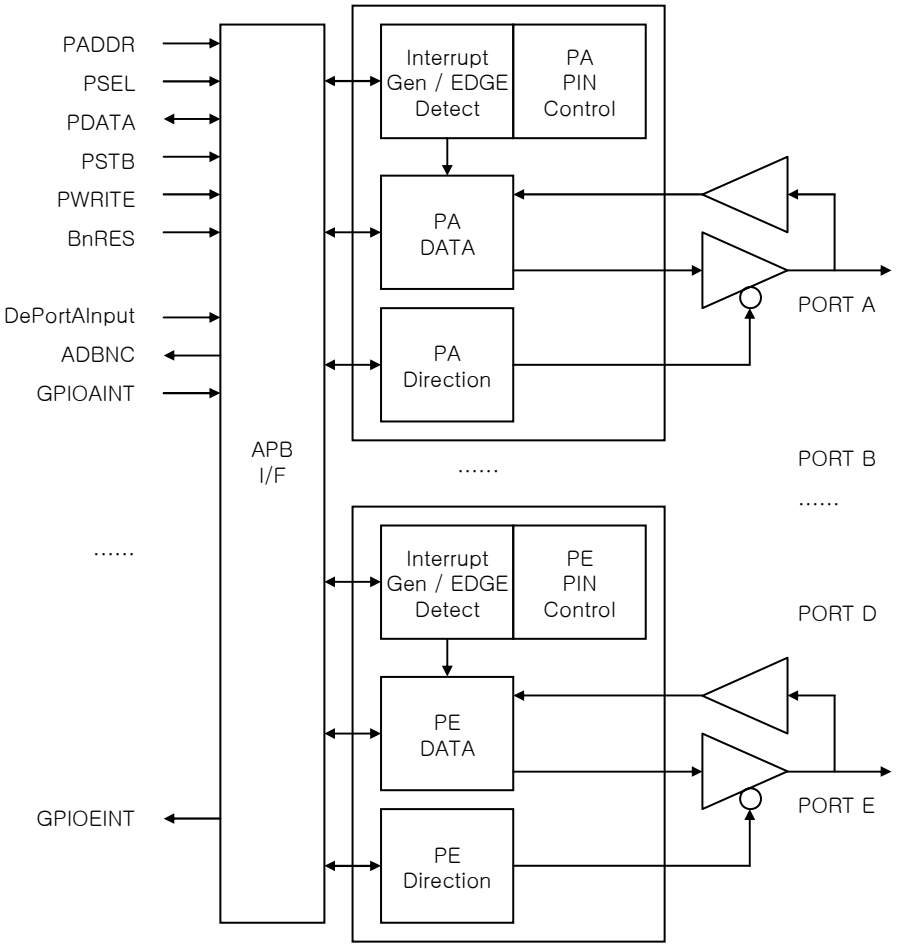
The below table shows possible configuration for KSCANI pins. In the table below, the 'O' means KSCANO[n] can be used for that function (Keyboard or GPIO) and 'X' means that KSCANO[n] cannot be used for GPIO when keyboard function is enabled where n is 0,1,2,3,4 or 5.

	KSCANI[0]	KSCANI[1]	KSCANI[2]	KSCANI[3]	KSCANI[4]	KSCANI[5]
Keyboard	O (pull-up)	O (pull-up)	O (pull-up)	O (pull-up)	O (pull-up)	O (pull-up)
GPIO	X	X	X	X	O	O

Table 9-21. Possible configuration of KSCANI pins when keyboard matrix is connected

9.14 GPIO

This document describes the Programmable Input /Output module (PIO). This is an AMBA slave module that connects to the Advanced Peripheral Bus (APB). For more information about AMBA, please refer to the AMBA Specification (ARM IHI 0001).
Most port pins are multiplexed with alternate functions for the peripheral features on the device. How each alternate function interferes with the port pin is described in “Operation” section. Refer to the individual module sections for a full description of the alternate function. The I/O status of each port is not changed during “SLEEP” or “DEEPSLEEP” mode of PMU.



DePortAInput : Port A inputs de-bounced by PMU unit

ADBNC : Port A de-bounce enable

Figure 9-67. Block diagram of GPIO

9.14.1 External Signals

Pin Name	Type	Description
KSCANI [5:0]	I/O	General Port A [5:0]
KSCANO [5:0]	I/O	General Port A [11:6]
UART5Tx	I/O	General Port B [27]
UART5Rx	I/O	General Port B [26]
nUDCD	I/O	General Port B [25]
nUDSR	I/O	General Port B [24]
nURTS	I/O	General Port B [23]
nUCTS	I/O	General Port B [22]
nUDTR	I/O	General Port B [21]
nURING	I/O	General Port B [20]
TouchYN	I/O	General Port B [19]
TouchXN	I/O	General Port B [18]
TouchYP	I/O	General Port B [17]
TouchXP	I/O	General Port B [16]
GPIOB15	I/O	General Port B [15]
GPIOB14	I/O	General Port B [14]
IrDA4Tx	I/O	General Port B [13]
IrDA4Rx	I/O	General Port B [12]
UART3Tx	I/O	General Port B [11]
UART3Rx	I/O	General Port B [10]
UART2Tx	I/O	General Port B [9]
UART2Rx	I/O	General Port B [8]
SCPRES[1]	I/O	General Port B [7]
SCCLK[1]	I/O	General Port B [6]
SCIO[1]	I/O	General Port B [5]
SCRST[1]	I/O	General Port B [4]
SCPRES[0]	I/O	General Port B [3]
SCCLK[0]	I/O	General Port B [2]
SCIO[0]	I/O	General Port B [1]
SCRST[0]	I/O	General Port B [0]
SMD[7:0]	I/O	General Port C [15:8]
nSMWP	I/O	General Port C [7]
nSMWE	I/O	General Port C [6]
nSMRE	I/O	General Port C [5]
nSMCE	I/O	General Port C [4]
SMCLE	I/O	General Port C [3]
SMALE	I/O	General Port C [2]
nSMRB	I/O	General Port C [1]
nSMCD	I/O	General Port C [0]
LD[7:0]	I/O	General Port D [24:17]
LCDEN	I/O	General Port D [16]
LFP	I/O	General Port D [15]
LCP	I/O	General Port D [14]
LBLEN	I/O	General Port D [13]
LAC	I/O	General Port D [12]
LLP	I/O	General Port D [11]
SCKE[1]	I/O	General Port D [10]
SCKE[0]	I/O	General Port D [9]
nSCS[1]	I/O	General Port D [8]
nSCS[0]	I/O	General Port D [7]
nRAS	I/O	General Port D [6]
nCAS	I/O	General Port D [5]
nSWE	I/O	General Port D [4]
DQMU	I/O	General Port D [3]

DQML	I/O	General Port D [2]
nRCS[3]	I/O	General Port D [1]
nRCS[2]	I/O	General Port D [0]
PWM[1:0]	I/O	General Port E [15:14]
TIMER[3:0]	I/O	General Port E [13:10]
SDA	I/O	General Port E [9]
SCL	I/O	General Port E [8]
SPICLK[1]	I/O	General Port E [7]
nSSICS[1]	I/O	General Port E [6]
SSITx[1]	I/O	General Port E [5]
SSIRx[1]	I/O	General Port E [4]
SSICLK[0]	I/O	General Port E [3]
nSSICS[0]	I/O	General Port E [2]
SSITx[0]	I/O	General Port E [1]
SSIRx[0]	I/O	General Port E [0]

Refer to Figure 2-1. 208 Pin diagram.

9.14.2 Registers

Address	Name	Width	Default	Description
0x8006.2000	ADATA	12	0x000	Port A Data Register
0x8006.2004	ADIR	12	0xFFF	Port A Data Direction Register
0x8006.2008	AIE	12	0x000	Port A Interrupt Enable Register
0x8006.200C	ASTAT	12	0x000	Port A Interrupt Status Register
0x8006.2010	AEDGE	12	0x000	Port A Edge Interrupt Register
0x8006.2014	ACLR	12	0x000	Port A Interrupt Clear Register
0x8006.2018	APOL	12	0x000	Port A Interrupt Polarity Register
0x8006.201C	AEN	12	0x000	Port A Enable Register
0x8006.2020	BDATA	28	0x00000000	Port B Data Register
0x8006.2024	BDIR	28	0x1FFFFFFF	Port B Data Direction Register
0x8006.2028	BIE	28	0x00000000	Port B Interrupt Enable Register
0x8006.202C	BSTAT	28	0x00000000	Port B Interrupt Status Register
0x8006.2030	BEDGE	28	0x00000000	Port B Edge Interrupt Register
0x8006.2034	BCLR	28	0x00000000	Port B Interrupt Clear Register
0x8006.2038	BPOL	28	0x00000000	Port B Interrupt Polarity Register
0x8006.203C	BEN	28	0x00000000	Port B Enable Register
0x8006.2040	CDATA	16	0x0000	Port C Data Register
0x8006.2044	CADIR	16	0xFFFF	Port C Data Direction Register
0x8006.2048	CIE	16	0x0000	Port C Interrupt Enable Register
0x8006.204C	CSTAT	16	0x0000	Port C Interrupt Status Register
0x8006.2050	CEDGE	16	0x0000	Port C Edge Interrupt Register
0x8006.2054	CCLR	16	0x0000	Port C Interrupt Clear Register
0x8006.2058	CPOL	16	0x0000	Port C Interrupt Polarity Register
0x8006.205C	CEN	16	0x0000	Port C Enable Register
0x8006.2060	DDATA	25	0x0000000	Port D Data Register
0x8006.2064	DDIR	25	0x1FFFFFF	Port D Data Direction Register
0x8006.2068	DIE	25	0x0000000	Port D Interrupt Enable Register
0x8006.206C	DSTAT	25	0x0000000	Port D Interrupt Status Register
0x8006.2070	DEDGE	25	0x0000000	Port D Edge Interrupt Register
0x8006.2074	DCLR	25	0x0000000	Port D Interrupt Clear Register
0x8006.2078	DPOL	25	0x0000000	Port D Interrupt Polarity Register
0x8006.207C	DEN	25	0x0000000	Port D Enable Register
0x8006.2080	EDATA	16	0x00000	Port E Data Register
0x8006.2084	EDIR	16	0x1FFFF	Port E Data Direction Register
0x8006.2088	EIE	16	0x00000	Port E Interrupt Enable Register
0x8006.208C	ESTAT	16	0x00000	Port E Interrupt Status Register
0x8006.2090	EEDGE	16	0x00000	Port E Edge Interrupt Register
0x8006.2094	ECLR	16	0x00000	Port E Interrupt Clear Register
0x8006.2098	EPOL	16	0x00000	Port E Interrupt Polarity Register
0x8006.209C	EEN	16	0x00000	Port E Enable Register
0x8006.20A4	ADEBE	12	0x000	Port A De-bounce Enable Register
0x8006.20A8	BDEBE	1	0x0	Port B De-bounce Enable Register

9.14.2.1 Port A Data Register (ADATA)

0x8006.2000

11	10	...	1	0
DATA, DIR, INTEN, STAT, EDGE, CLR, POL, ENABLE [7:0]				

Bits	Type	Function
12	R/W	Port A output data Values written to this register will be output on port A pins if the corresponding bits of port A direction register are zeros (port pin is configured as output). Values read from the address of this register reflect the external state of port A not the value written to this register. All bits are cleared by a system reset. When the port pin is configured as input, this input can be an interrupt source with appropriate register setting. When DIR[n] bit in ADIR register is 0, 0 = Drives port A[n] pin LOW. (default) 1 = Drives port A[n] pin HIGH. When DIR[n] bit in ADIR register is 1, 0 = The read value on port A[n] is '0'. (default) 1 = The read value on port A[n] is '1'.

9.14.2.2 Port A Direction Register (ADIR)

0x8006.2004

11	10	...	1	0
DIR [7:0]				

Bits	Type	Function
12	R/W	Port A direction Bits set in this register will select the corresponding pin of port A to configured as an input. All bits are set by a system reset. 0 = Port A[n] is configured as an output. 1 = Port A[n] is configured as an input. (default)

9.14.2.3 Port A Interrupt Enable Register (AIE)

0x8006.2008

11	10	...	1	0
INTEN [11:0]				

Bits	Type	Function
12	R/W	Port A interrupt enable Bits set in this register make the corresponding pins of port A to become an external interrupt source. All bits are cleared by a system reset. 0 = Disable interrupt. (default) 1 = Enable interrupt

9.14.2.4 Port A Interrupt Status Register (ASTAT)

0x8006.200C

11	10	...	1	0
STAT [11:0]				

Bits	Type	Function
12	R	Port A interrupt status All PIO signals can be used as interrupt sources according to the settings. Each port has the following registers and interrupt signals to interrupt controller. The interrupt controller unit of HMS30C7210 receives active HIGH level interrupt sources only. But GPIO block can receive not only active HIGH or active LOW level, but also rising or falling edge signals. Then interprets and sends interrupt request to the interrupt controller. All bits can be controlled separately. Values in this read-only register represents that the interrupt requests are pending on corresponding pins. All bits are cleared by a system reset. 0 = Interrupt is cleared or no interrupt is requested. (default) 1 = Interrupt pending.

9.14.2.5 Port A Edge Interrupt Register (AEDGE)

0x8006.2010

11	10	...	1	0
EDGE [11:0]				

Bits	Type	Function
12	R/W	Port A interrupts are edge triggered All pins of port A can be an external interrupt source. And the external interrupts can be triggered by detecting an edge or a level. Bits set in this register makes the corresponding pins of port A to be edge triggered interrupt source. All bits are cleared by a system reset. 0 = External interrupt is triggered by level. (default) 1 = External interrupt is triggered by edge.

9.14.2.6 Port A Interrupt Clear Register (ACLR)

0x8006.2014

11	10	...	1	0
CLR [11:0]				

Bits	Type	Function
12	W	Port A interrupt clear If a edge triggered interrupt is used, the status register (ASTAT) and interrupt pending are cleared by writing '1' in the corresponding bit position of this register. All bits are automatically cleared after written. This register is write only. 0 = No action is done. (default) 1 = Clear edge triggered interrupt request and interrupt status register (ASTAT).

9.14.2.7 Port A Interrupt Polarity Register (APOL)

0x8006.2018

11	10	...	1	0
POL [11:0]				

Bits	Type	Function
12	R/W	Port A interrupt polarity If level triggered interrupts are used, bits set in this register activate the interrupts when the level of corresponding pins of port A is low. If edge triggered interrupts are used, bits set in this register activate the interrupts when the corresponding pins of port A make an falling edge. All bits are cleared by a system reset. When interrupt is level sensitive (EDGE[n] in AEDGE register is 0), 0 = External interrupt is triggered by a high level. (default) 1 = External interrupt is triggered by a low level. When interrupt is edge triggered (EDGE[n] in AEDGE register is 1), 0 = External interrupt is triggered by a rising edge. (default) 1 = External interrupt is triggered by a falling edge.

9.14.2.8 Port A Enable Register (AEN)

0x8006.201C

11	10	...	1	0
ENABLE [11:0]				

Bits	Type	Function
11	R/W	Port A[11] Enable Setting this bit makes the pin KSCANO[5] to be used as general digital I/O pin. 0 = Port A[11] is used as KSCANO[5]. (default) 1 = Port A[11] is used as general I/O pin.
10	R/W	Port A[10] Enable 0 = Port A[10] is used as KSCANO[4]. (default) 1 = Port A[10] is used as general I/O pin.
9	R/W	Port A[9] Enable 0 = Port A[9] is used as KSCANO[3]. (default) 1 = Port A[9] is used as general I/O pin.
8	R/W	Port A[8] Enable 0 = Port A[8] is used as KSCANO[2]. (default) 1 = Port A[8] is used as general I/O pin.
7	R/W	Port A[7] Enable 0 = Port A[7] is used as KSCANO[1]. (default) 1 = Port A[7] is used as general I/O pin.
6	R/W	Port A[6] Enable 0 = Port A[6] is used as KSCANO[0]. (default) 1 = Port A[6] is used as general I/O pin.
5	R/W	Port A[5] Enable 0 = Port A[5] is used as KSCANI[5]. (default) 1 = Port A[5] is used as general I/O pin.
4	R/W	Port A[4] Enable 0 = Port A[4] is used as KSCANI[4]. (default) 1 = Port A[4] is used as general I/O pin.
3	R/W	Port A[3] Enable 0 = Port A[3] is used as KSCANI[3]. (default) 1 = Port A[3] is used as general I/O pin.
2	R/W	Port A[2] Enable 0 = Port A[2] is used as KSCANI[2]. (default) 1 = Port A[2] is used as general I/O pin.
1	R/W	Port A[1] Enable 0 = Port A[1] is used as KSCANI[1]. (default) 1 = Port A[1] is used as general I/O pin.
0	R/W	Port A[0] Enable 0 = Port A[0] is used as KSCANI[0]. (default) 1 = Port A[0] is used as general I/O pin.

9.14.2.9 Port B Data Register (BDATA)

0x8006.2020

27	26	...	1	0
DATA, DIR, INTEN, STAT, EDGE, CLR, POL, ENABLE [27:0]				

9.14.2.10 Port B Direction Register (BDIR)

0x8006.2024

27	26	...	1	0
DIR [27:0]				

9.14.2.11 Port B Interrupt Enable Register (BIE)

0x8006.2028

27	26	...	1	0
INTEN [27:0]				

9.14.2.12 Port B Interrupt Status Register (BSTAT)

0x8006.202C

27	26	...	1	0
STAT [27:0]				

9.14.2.13 Port B Edge Interrupt Register (BEDGE)

0x8006.2030

27	26	...	1	0
EDGE [27:0]				

9.14.2.14 Port B Interrupt Clear Register (BCLR)

0x8006.2034

27	26	...	1	0
CLR [27:0]				

9.14.2.15 Port B Interrupt Polarity Register (BPOL)

0x8006.2038

27	26	...	1	0
POL [27:0]				

9.14.2.16 Port B Enable Register (BEN)

0x8006.203C

27	26	...	1	0
ENABLE [27:0]				

Bits	Type	Function
27	R/W	Port B[27] Enable Setting this bit makes the pin UART5Tx to be used as general digital I/O pin. 0 = Port B[27] is used as UART5Tx. (default) 1 = Port B[27] is used as general I/O pin.
26	R/W	Port B[26] Enable 0 = Port B[26] is used as UART5Rx. (default) 1 = Port B[26] is used as general I/O pin.
25	R/W	Port B[25] Enable 0 = Port B[25] is used as nUDCD. (default) 1 = Port B[25] is used as general I/O pin.
24	R/W	Port B[24] Enable 0 = Port B[24] is used as nUDSR. (default) 1 = Port B[24] is used as general I/O pin.
23	R/W	Port B[23] Enable 0 = Port B[23] is used as nURTS. (default) 1 = Port B[23] is used as general I/O pin.
22	R/W	Port B[22] Enable 0 = Port B[22] is used as nUCTS. (default) 1 = Port B[22] is used as general I/O pin.
21	R/W	Port B[21] Enable 0 = Port B[21] is used as nUDTR. (default) 1 = Port B[21] is used as general I/O pin.
20	R/W	Port B[20] Enable 0 = Port B[20] is used as nURING. (default) 1 = Port B[20] is used as general I/O pin.
19	R/W	Port B[19] Enable 0 = Port B[19] is used as TouchYN. (default) 1 = Port B[19] is used as general I/O pin.
18	R/W	Port B[18] Enable 0 = Port B[18] is used as TouchXN. (default) 1 = Port B[18] is used as general I/O pin.
17	R/W	Port B[17] Enable 0 = Port B[17] is used as TouchYP. (default) 1 = Port B[17] is used as general I/O pin.
16	R/W	Port B[16] Enable 0 = Port B[16] is used as TouchXP. (default) 1 = Port B[16] is used as general I/O pin.
15	R/W	Port B[15] Enable 0 = Port B[15] is used as HotSync input to PMU unit. (default) 1 = Port B[15] is used as general I/O pin.
14	R/W	Port B[14] Enable 0 = Port B[14] is used as ToDeepSleep input to PMU unit. (default) 1 = Port B[14] is used as general I/O pin
13	R/W	Port B[13] Enable 0 = Port B[13] is used as IrDATx. (default) 1 = Port B[13] is used as general I/O pin.
12	R/W	Port B[12] Enable 0 = Port B[12] is used as IrDARx. (default) 1 = Port B[12] is used as general I/O pin.
11	R/W	Port B[11] Enable 0 = Port B[11 is used as UART3Tx. (default) 1 = Port B[11] is used as general I/O pin.
10	R/W	Port B[10] Enable 0 = Port B[10] is used as UART3Rx. (default) 1 = Port B[10] is used as general I/O pin.
9	R/W	Port B[9] Enable 0 = Port B[9] is used as UART2Tx. (default) 1 = Port B[9] is used as general I/O pin.
8	R/W	Port B[8] Enable 0 = Port B[8] is used as UART2Rx. (default) 1 = Port B[8] is used as general I/O pin.

7	R/W	Port B[7] Enable 0 = Port B[7] is used as SCPRES[1]. (default) 1 = Port B[7] is used as general I/O pin.
6	R/W	Port B[6] Enable 0 = Port B[6] is used as SCCLK[1]. (default) 1 = Port B[6] is used as general I/O pin.
5	R/W	Port B[5] Enable 0 = Port B[5] is used as SCIO[1]. (default) 1 = Port B[5] is used as general I/O pin.
4	R/W	Port B[4] Enable 0 = Port B[4] is used as SCRST[1]. (default) 1 = Port B[4] is used as general I/O pin.
3	R/W	Port B[3] Enable 0 = Port B[3] is used as SCPRES[0]. (default) 1 = Port B[3] is used as general I/O pin.
2	R/W	Port B[2] Enable 0 = Port B[2] is used as SCCLK[0]. (default) 1 = Port B[2] is used as general I/O pin.
1	R/W	Port B[1] Enable 0 = Port B[1] is used as SCIO[0]. (default) 1 = Port B[1] is used as general I/O pin.
0	R/W	Port B[0] Enable 0 = Port B[0] is used as SCRST[0]. (default) 1 = Port B[0] is used as general I/O pin.

9.14.2.17 Port C Data Register (CDATA)

0x8006.2040

15	14	...	1	0
DATA, DIR, INTEN, STAT, EDGE, CLR, POL, ENABLE[15:0]				

9.14.2.18 Port C Direction Register (CDIR)

0x8006.2044

15	14	...	1	0
DIR [15:0]				

9.14.2.19 Port C Interrupt Enable Register (CIE)

0x8006.2048

15	14	...	1	0
INTEN [15:0]				

9.14.2.20 Port C Interrupt Status Register (CSTAT)

0x8006.204C

15	14	...	1	0
STAT [15:0]				

9.14.2.21 Port C Edge Interrupt Register (CEDGE)

0x8006.2050

15	14	...	1	0
EDGE [15:0]				

9.14.2.22 Port C Interrupt Clear Register (CCLR)

0x8006.2054

15	14	...	1	0
CLR [15:0]				

9.14.2.23 Port C Interrupt Polarity Register (CPOL)

0x8006.2058

15	14	...	1	0
POL [15:0]				

9.14.2.24 Port C Enable Register (CEN)

0x8006.205C

15	14	...	1	0
ENABLE [15:0]				

Bits	Type	Function
15	R/W	Port C[15] Enable Setting this bit makes the pin SMD[7] to be used as general digital I/O pin. 0 = Port C[15] is used as SMD[7]. (default) 1 = Port C[15] is used as general I/O pin.
14	R/W	Port C[14] Enable 0 = Port C[14] is used as SMD[6]. (default) 1 = Port C[14] is used as general I/O pin
13	R/W	Port C[13] Enable 0 = Port C[13] is used as SMD[5]. (default) 1 = Port C[13] is used as general I/O pin.
12	R/W	Port C[12] Enable 0 = Port C[12] is used as SMD[4]. (default) 1 = Port C[12] is used as general I/O pin.
11	R/W	Port C[11] Enable 0 = Port C[11 is used as SMD[3]. (default) 1 = Port C[11] is used as general I/O pin.
10	R/W	Port C[10] Enable 0 = Port C[10] is used as SMD[2]. (default) 1 = Port C[10] is used as general I/O pin.
9	R/W	Port C[9] Enable 0 = Port C[9] is used as SMD[1]. (default) 1 = Port C[9] is used as general I/O pin.
8	R/W	Port C[8] Enable 0 = Port C[8] is used as SMD[0]. (default) 1 = Port C[8] is used as general I/O pin.
7	R/W	Port C[7] Enable 0 = Port C[7] is used as nSMWP. (default) 1 = Port C[7] is used as general I/O pin.
6	R/W	Port C[6] Enable 0 = Port C[6] is used as nSMWE. (default) 1 = Port C[6] is used as general I/O pin.
5	R/W	Port C[5] Enable 0 = Port C[5] is used as nSMRE. (default) 1 = Port C[5] is used as general I/O pin.
4	R/W	Port C[4] Enable 0 = Port C[4] is used as nSMCE. (default) 1 = Port C[4] is used as general I/O pin.
3	R/W	Port C[3] Enable 0 = Port C[3] is used as SMCLE. (default) 1 = Port C[3] is used as general I/O pin.
2	R/W	Port C[2] Enable 0 = Port C[2] is used as SMALE. (default) 1 = Port C[2] is used as general I/O pin.
1	R/W	Port C[1] Enable 0 = Port C[1] is used as nSMRB. (default) 1 = Port C[1] is used as general I/O pin.
0	R/W	Port C[0] Enable 0 = Port C[0] is used as nSMCD. (default) 1 = Port C[0] is used as general I/O pin.

9.14.2.25 Port D Data Register (DDATA)

0x8006.2060

24	23	...	1	0
DATA, DIR, MASK, STAT, EDGE, CLR, POL, ENABLE[24:0]				

9.14.2.26 Port D Direction Register (DDIR)

0x8006.2064

24	23	...	1	0
DIR [24:0]				

9.14.2.27 Port D Interrupt Enable Register (DIE)

0x8006.2068

24	23	...	1	0
INTEN [24:0]				

9.14.2.28 Port D Interrupt Status Register (DSTAT)

0x8006.206C

24	23	...	1	0
STAT [24:0]				

9.14.2.29 Port D Edge Interrupt Register (DEDGE)

0x8006.2070

24	23	...	1	0
EDGE [24:0]				

9.14.2.30 Port D Interrupt Clear Register (DCLR)

0x8006.2074

24	23	...	1	0
CLR [24:0]				

9.14.2.31 Port D Interrupt Polarity Register (DPOL)

0x8006.2078

24	23	...	1	0
POL [24:0]				

9.14.2.32 Port D Enable Register (DEN)

0x8006.207C

24	23	...	1	0
ENABLE [24:0]				

Bits	Type	Function
24	R/W	Port D[24] Enable Setting this bit makes the pin LD[7] to be used as general digital I/O pin. 0 = Port D[24] is used as LD[7]. (default) 1 = Port D[24] is used as general I/O pin.
23	R/W	Port D[23] Enable 0 = Port D[23] is used as LD[6]. (default) 1 = Port D[23] is used as general I/O pin.
22	R/W	Port D[22] Enable 0 = Port D[22] is used as LD[5]. (default) 1 = Port D[22] is used as general I/O pin.
21	R/W	Port D[21] Enable 0 = Port D[21] is used as LD[4]. (default) 1 = Port D[21] is used as general I/O pin.
20	R/W	Port D[20] Enable 0 = Port D[20] is used as LD[3]. (default) 1 = Port D[20] is used as general I/O pin.
19	R/W	Port D[19] Enable 0 = Port D[19] is used as LD[2]. (default) 1 = Port D[19] is used as general I/O pin.
18	R/W	Port D[18] Enable 0 = Port D[18] is used as LD[1]. (default) 1 = Port D[18] is used as general I/O pin.
17	R/W	Port D[17] Enable 0 = Port D[17] is used as LD[0]. (default) 1 = Port D[17] is used as general I/O pin.
16	R/W	Port D[16] Enable 0 = Port D[16] is used as LCDEN. (default) 1 = Port D[16] is used as general I/O pin.
15	R/W	Port D[15] Enable 0 = Port D[15] is used as LFP. (default) 1 = Port D[15] is used as general I/O pin.
14	R/W	Port D[14] Enable 0 = Port D[14] is used as LCP. (default) 1 = Port D[14] is used as general I/O pin
13	R/W	Port D[13] Enable 0 = Port D[13] is used as LBLEN. (default) 1 = Port D[13] is used as general I/O pin.
12	R/W	Port D[12] Enable 0 = Port D[12] is used as LAC. (default) 1 = Port D[12] is used as general I/O pin.
11	R/W	Port D[11] Enable 0 = Port D[11 is used as LLP. (default) 1 = Port D[11] is used as general I/O pin.
10	R/W	Port D[10] Enable 0 = Port D[10] is used as SCKE[1]. (default) 1 = Port D[10] is used as general I/O pin.
9	R/W	Port D[9] Enable 0 = Port D[9] is used as SCKE[0]. (default) 1 = Port D[9] is used as general I/O pin.
8	R/W	Port D[8] Enable 0 = Port D[8] is used as nSCS[1]. (default) 1 = Port D[8] is used as general I/O pin.
7	R/W	Port D[7] Enable 0 = Port D[7] is used as nSCS[0]. (default) 1 = Port D[7] is used as general I/O pin.
6	R/W	Port D[6] Enable 0 = Port D[6] is used as nRAS. (default) 1 = Port D[6] is used as general I/O pin.
5	R/W	Port D[5] Enable 0 = Port D[5] is used as nCAS. (default) 1 = Port D[5] is used as general I/O pin.

4	R/W	Port D[4] Enable 0 = Port D[4] is used as SWE. (default) 1 = Port D[4] is used as general I/O pin.
3	R/W	Port D[3] Enable 0 = Port D[3] is used as DQMU. (default) 1 = Port D[3] is used as general I/O pin.
2	R/W	Port D[2] Enable 0 = Port D[2] is used as DQML. (default) 1 = Port D[2] is used as general I/O pin.
1	R/W	Port D[1] Enable 0 = Port D[1] is used as nRCS[3]. (default) 1 = Port D[1] is used as general I/O pin.
0	R/W	Port D[0] Enable 0 = Port D[0] is used as nRCS[2]. (default) 1 = Port D[0] is used as general I/O pin.

9.14.2.33 Port E Data Register (EDATA)

0x8006.2080

15	14	...	1	0
DATA, DIR, INTEN, STAT, EDGE, CLR, POL, ENABLE [15:0]				

9.14.2.34 Port E Direction Register (EDIR)

0x8006.2084

15	14	...	1	0
DIR [15:0]				

9.14.2.35 Port E Interrupt Enable Register (EIE)

0x8006.2088

15	14	...	1	0
INTEN [15:0]				

9.14.2.36 Port E Interrupt Status Register (ESTAT)

0x8006.208C

15	14	...	1	0
STAT [15:0]				

9.14.2.37 Port E Edge Interrupt Register (EEDGE)

0x8006.2090

15	14	...	1	0
EDGE [15:0]				

9.14.2.38 Port E Interrupt Clear Register (ECLR)

0x8006.2094

15	14	...	1	0
CLR [15:0]				

9.14.2.39 Port E Interrupt Polarity Register (EPOL)

0x8006.2098

15	14	...	1	0
POL [15:0]				

9.14.2.40 Port E Enable Register (EEN)

0x8006 209C

15	14	...	1	0
ENABLE [15:0]				

Bits	Type	Function
15	R/W	Port E[15] Enable Setting this bit makes the pin PWM[1] to be used as general digital I/O pin. 0 = Port E[15] is used as PWM[1]. (default) 1 = Port E[15] is used as general I/O pin.
14	R/W	Port E[14] Enable 0 = Port E[14] is used as PWM[0]. (default) 1 = Port E[14] is used as general I/O pin
13	R/W	Port E[13] Enable 0 = Port E[13] is used as TIEMR[3]. (default) 1 = Port E[13] is used as general I/O pin.
12	R/W	Port E[12] Enable 0 = Port E[12] is used as TIMER[2]. (default) 1 = Port E[12] is used as general I/O pin.
11	R/W	Port E[11] Enable 0 = Port E[11 is used as TIMER[1]. (default) 1 = Port E[11] is used as general I/O pin.
10	R/W	Port E[10] Enable 0 = Port E[10] is used as TIMER[0]. (default) 1 = Port E[10] is used as general I/O pin.
9	R/W	Port E[9] Enable 0 = Port E[9] is used as SDA. (default) 1 = Port E[9] is used as general I/O pin.
8	R/W	Port E[8] Enable 0 = Port E[8] is used as SCL. (default) 1 = Port E[8] is used as general I/O pin.
7	R/W	Port E[7] Enable 0 = Port E[7] is used as SPICLK[1]. (default) 1 = Port E[7] is used as general I/O pin.
6	R/W	Port E[6] Enable 0 = Port E[6] is used as nSPICS[1]. (default) 1 = Port E[6] is used as general I/O pin.
5	R/W	Port E[5] Enable 0 = Port E[5] is used as SPITx[1]. (default) 1 = Port E[5] is used as general I/O pin.
4	R/W	Port E[4] Enable 0 = Port E[4] is used as SPIRx[1]. (default) 1 = Port E[4] is used as general I/O pin.
3	R/W	Port E[3] Enable 0 = Port E[3] is used as SPICLK[0]. (default) 1 = Port E[3] is used as general I/O pin.
2	R/W	Port E[2] Enable 0 = Port E[2] is used as nSPICS[0]. (default) 1 = Port E[2] is used as general I/O pin.
1	R/W	Port E[1] Enable 0 = Port E[1] is used as SPITx[0]. (default) 1 = Port E[1] is used as general I/O pin.
0	R/W	Port E[0] Enable 0 = Port E[0] is used as SPIRx[0]. (default) 1 = Port E[0] is used as general I/O pin.

9.14.2.41 Port A De-Bounce Enable Register (ADEBE)

0x8006 20A4

11	10	...	1	0
ADBNC[11:0]				

Bits	Type	Function
11	R/W	Port A[11] input de-bounce enable The input signal of port A[11] can be de-bounced by setting this bit to remove mechanical jitter. If this bit is cleared, input signal of port A[11] reflects the status of pin KSCANO[5] immediately. 0 = Port A[11] input is used directly. (default) 1 = Port A[11] input is used after de-bouncing.
10	R/W	Port A[10] input de-bounce enable 0 = Port A[10] input is used directly. (default) 1 = Port A[10] input is used after de-bouncing.
9	R/W	Port A[9] input de-bounce enable 0 = Port A[9] input is used directly. (default) 1 = Port A[9] input is used after de-bouncing.
8	R/W	Port A[8] input de-bounce enable 0 = Port A[8] input is used directly. (default) 1 = Port A[8] input is used after de-bouncing
7	R/W	Port A[7] input de-bounce enable 0 = Port A[7] input is used directly. (default) 1 = Port A[7] input is used after de-bouncing
6	R/W	Port A[6] input de-bounce enable 0 = Port A[6] input is used directly. (default) 1 = Port A[6] input is used after de-bouncing.
5	R/W	Port A[5] input de-bounce enable 0 = Port A[5] input is used directly. (default) 1 = Port A[5] input is used after de-bouncing.
4	R/W	Port A[4] input de-bounce enable 0 = Port A[4] input is used directly. (default) 1 = Port A[4] input is used after de-bouncing.
3	R/W	Port A[3] input de-bounce enable 0 = Port A[3] input is used directly. (default) 1 = Port A[3] input is used after de-bouncing.
2	R/W	Port A[2] input de-bounce enable 0 = Port A[2] input is used directly. (default) 1 = Port A[2] input is used after de-bouncing.
1	R/W	Port A[1] input de-bounce enable 0 = Port A[1] input is used directly. (default) 1 = Port A[1] input is used after de-bouncing.
0	R/W	Port A[0] input de-bounce enable 0 = Port A[0] input is used directly. (default) 1 = Port A[0] input is used after de-bouncing.

9.14.2.42 Port B De-Bounce Enable Register (BDEBE)

0x8006.20A8

							0
							BDBNC14

Bits	Type	Function
0	R/W	Port B[14] input de-bounce enable The input signal of port B[14] can be de-bounced by setting this bit to remove mechanical jitter. If this bit is cleared, input signal of port B[14] reflects the status of pin GPIOB14 immediately. 0 = Port B[14] input is used directly. (default) 1 = Port B[14] input is used after de-bouncing.

9.14.3 Operations

Throughout the operation description of each port, port A is used as an example port. All is same to other ports.

9.14.3.1 Configuring the pin

The DIR[n] bit in the ADIR register selects the direction of this pin. If DIR[n] is written logic one, port A[n] is configured as an input pin. If DIR[n] is written logic zero, port A[n] is configured as an output pin. Note that port A[n] can be used as an input or output pin only when ENABLE[n] bit in the AEN register is written logic one. Otherwise, port A[n] is used as an primary function pin.

9.14.3.2 Writing the pin value

Values written to ADATA register will be output on port A pins if the corresponding bits of port A direction register are zeros.
The pin of port A[n] is driven high when the DATA[n] bit in ADATA register is written logic one. And the pin of port A[n] is driven low when the DATA[n] is written logic zero.

9.14.3.3 Reading the pin value

Independent of the setting of data direction bit DIR[n], the port pin can be read through the ADATA register bit. In that case, ENABLE[n] bit in the AEN register must be written logic one to read the pin value. If ENABLE[n] bit is written logic zero, the pin value will be read as zero.

9.14.3.4 *Alternate port functions*

All port pins have alternate functions in addition to being general digital I/Os. The alternate function can be selected by clearing ENABLE[n] bit in each port enable register. If ENABLE[n] bit in the AEN register is written logic one, port A[n] is configured as an general digital I/O and if ENABLE[n] is written zero, port A[n] is used by alternate function block. For example if AEN[11:0] is written value 0xF00, port A[7:0] are used for keyboard function, and port A[11:8] are used as general I/Os.

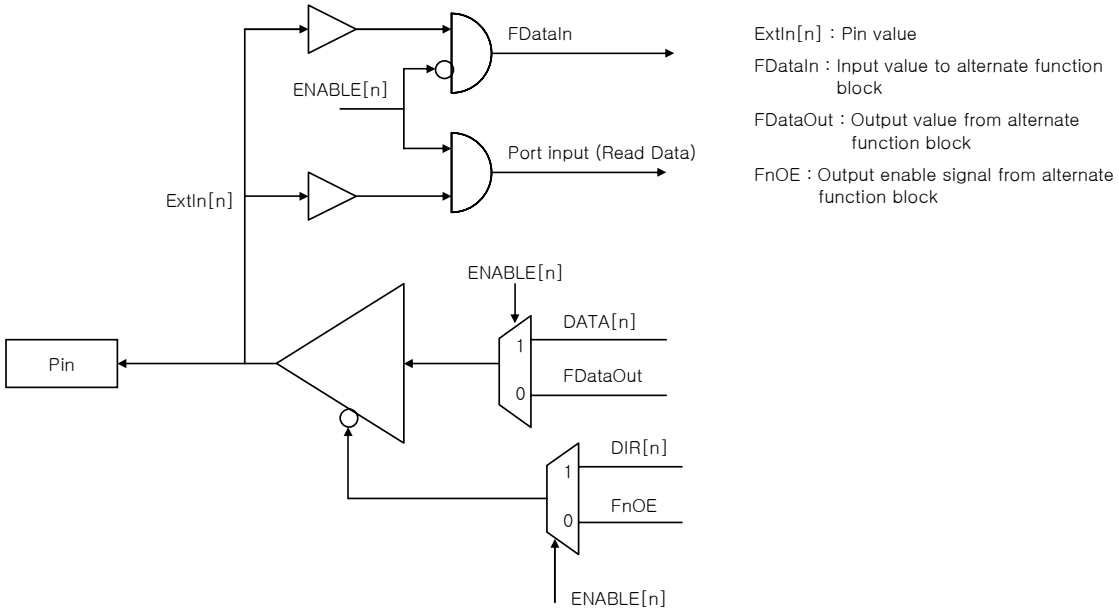


Figure 9-68. Alternate port functions

9.14.3.5 External interrupt request

GPIO has 7 interrupt sources. Each port can be configured as 1 interrupt source except port B. That is, if any pin of port A makes an interrupt condition, an interrupt is requested form port A. In order to use a port A as an interrupt source, specify EDGE[n] bits in AEDGE register and POL[n] bits in APOL register according to interrupt type. And then set the INTEN[n] bits in AIE register to enable interrupt request. The usage of port C, D and E is same as port A.

Unlike other ports, port B has 3 interrupt sources.

- The first interrupt source comes from port B[27:16] or port B[13:0], and these port pins are used as normal external interrupt sources like other port pins.
- The second interrupt source is port B[15] (GPIOB[15]). GPIOB[15] is used to detect HotSync. When PMU is in DEEPSLEEP or SLEEP modes, the interrupt of port B[15] makes the PMU wake-up.
- And the third interrupt source is port B[14] (GPIOB[14]). GPIOB[14] is required to make the operating mode of PMU unit go to DEEPSLEEP mode. Changing the operation mode of PMU unit is software's responsibility. That is, when GPIOB[14] triggers an interrupt, the interrupt handler forces the PMU to enter DEEPSLEEP mode.

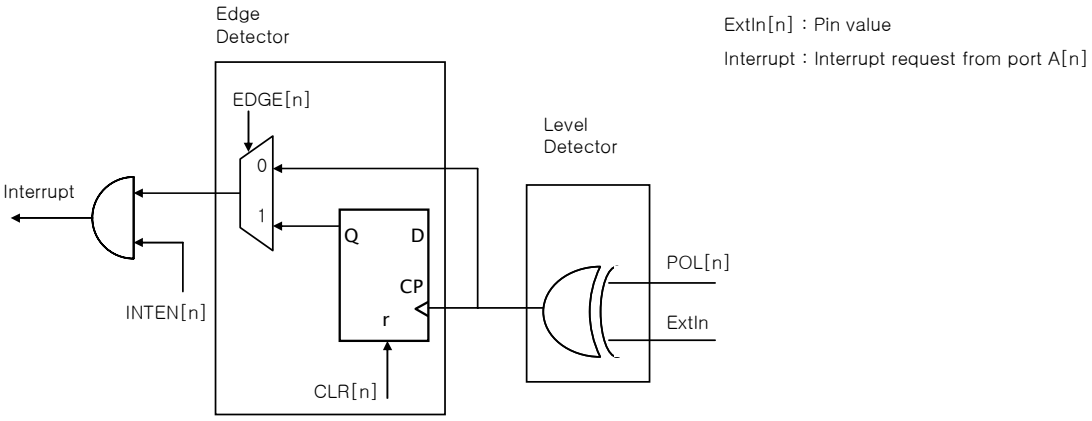


Figure 9-69. Interrupt request

There are 4 cases to trigger an interrupt, and the sequence to trigger an interrupt is shown below. Port A is used to be an interrupt source. Note that interrupt clear methods are different according to the triggering condition.

A high level of port pin

- Decide port pins to be interrupt sources.
- Write zeros to the selected bits in APOL register.
- Write zeros to the selected bits in AEDGE register.
- Enable interrupts by writing ones to the selected bits in AIE register.
- If interrupt is requested, the external port pin must be changed to low level to clear interrupt request.

A low level of port pin

- Decide port pins to be interrupt sources.
- Write ones to the selected bits in APOL register.
- Write zeros to the selected bits in AEDGE register.
- Enable interrupts by writing ones to the selected bits in AIE register.
- If interrupt is requested, the external port pin must be changed to high level to clear interrupt request.

A rising edge of port pin

- Decide port pins to be interrupt sources.
- Write zeros to the selected bits in APOL register.
- Write ones to the selected bits in AEDGE register.
- Enable interrupts by writing ones to the selected bits in AIE register.
- If interrupt is requested, the handler writes one to ACLR register (corresponding bit position).

A falling edge of port pin

- Decide port pins to be interrupt sources.
- Write ones to the selected bits in APOL register.
- Write ones to the selected bits in AEDGE register.
- Enable interrupts by writing ones to the selected bits in AIE register.
- If interrupt is requested, the handler writes one to ACLR register (corresponding bit position).

Interrupt Name	Configurable Bits
GPIOAINT	Port A[11:0]
GPIOBINT	Port B[27:0]
GPIOCINT	Port C[15:0]
GPIODINT	Port D[24:0]
GPIOEINT	Port E[15:0]
GPIOB14INT	Port B[14], Deep Sleep interrupt
GPIOB15INT	Port B[15], Hotsync interrupt

Table 9-22. Interrupt sources of I/Os (to interrupt controller unit)

9.14.3.6 *De-bouncing port A and port B[14]*

All pins of port A and GPIOB[14] can be de-bounced before being used as input signals. If ADBNC[n] bit in ADEBE register is written logic one, the input signal of port A[n] is de-bounced by a slow clock, and the de-bounced signal is used in alternate function block or interrupt source of port A[n]. Also, the read value of port A[n] is de-bounced signal.

In port B, only GPIOB[14] can be de-bounced.

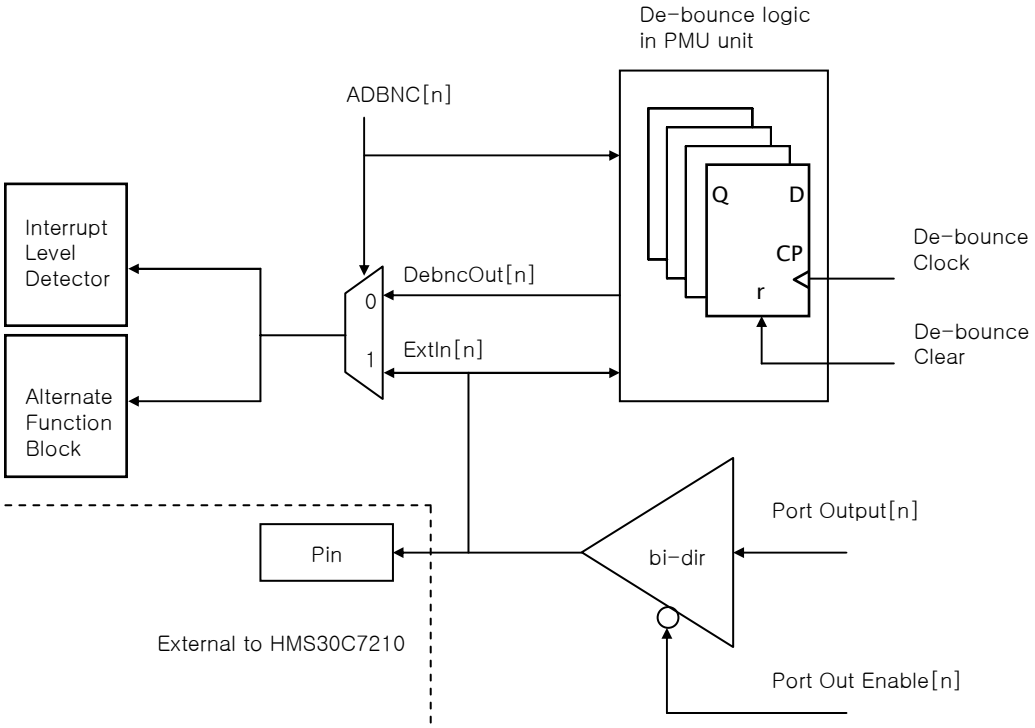


Figure 9-70. De-bouncing of port A

9.14.4 GPIO Rise and Fall Time

This sections describes the rise and fall time of each pad pin.
The pad library cells used in HMS30C7210 are symbolized as PC3B01, PC3B03 and PT3B03. PC3B01 and PC3B03 cells are three state CMOS input/output pads with AC drive capability of 1x and 3x. PT3B03 cells are three state TTL input/output pads with DC drive capability of 8mA.

The following 2 figures depicts pad organization and waveform respectively. And these figures are used to explain timing symbols.
The symbol tCMOS or tTTL mean the propagation delay from I to PAD of CMOS or TTL pad and tOEN means the propagation delay from OEN to PAD of each pad cell.

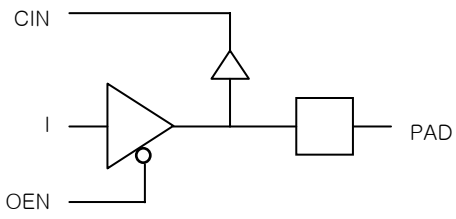


Figure 9-71. Pad organization

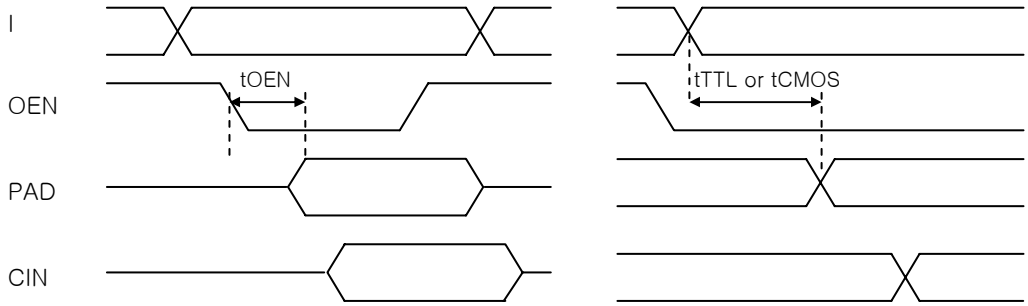


Figure 9-72. Timing diagram of bi-directional pad (CMOS or TTL)

The propagation delay listed in the following table is rounded off to three decimal places.

Port Name		50pF		100pF		150pF	
		Rise(ns)	Fall(ns)	Rise(ns)	Fall(ns)	Rise(ns)	Fall(ns)
PC3B01	tCMOS	5.60	4.94	9.80	8.29	14.01	11.64
	tOEN	5.92	4.25	10.10	7.63	14.29	11.02
PC3B03	tCMOS	3.60	3.74	5.71	5.39	7.82	7.04
	tOEN	3.84	2.53	5.93	4.21	8.02	5.90
PT3B03	tTTL	2.71	2.74	4.17	3.87	5.63	5.00
	tOEN	3.28	1.96	4.72	3.12	6.15	4.27

Table 9-23. Propagation delays (ns) for sample pad loads

10 DEBUG AND TEST INTERFACE

10.1 Overview

The HMS30C7210 has built-in features that enable debug and test in a number of different contexts. Firstly, there are circuit structures to help with software development. Secondly, the device contains boundary scan cells for circuit board test. Finally, the device contains some special test modes that enable the generation production patterns for the device itself.

10.2 Software Development Debug and Test Interface

The ARM720T processors incorporated inside HMS30C7210 contain hardware extensions for advanced debugging features. These are intended to ease user development and debugging of application software, operating systems, and the hardware itself.

Full details of the debug interfaces and their programming can be found in ARM720T Data Sheet (ARM DDI-0087). The MultiICE product enables the ARM720T macrocells to be debugged in one environment. Refer to Guide to MultiICE (ARM DUI-0048).

10.3 Test Access Port and Boundary-Scan

HMS30C7210 contains full boundary scan on its inputs and outputs to help with circuit board test. This supports both INTEST and EXTEST, allowing patterns to be applied serially to the HMS30C7202 when fixed in a board and for full circuit board connection respectively. The boundary-scan interface conforms to the IEEE Std. 1149.1- 1990, Standard Test Access Port and Boundary-Scan Architecture. (Please refer to this standard for an explanation of the terms used in this section and for a description of the TAP controller states.) The boundary-scan interface provides a means of testing the core of the device when it is fitted to a circuit board, and a means of driving and sampling all the external pins of the device irrespective of the core state. This latter function permits testing of both the device's electrical connections to the circuit board, and (in conjunction with other devices on the circuit board having a similar interface) testing the integrity of the circuit board connections between devices. The interface intercepts all external connections within the device, and each such “cell” is then connected together to form a serial register (the boundary scan register). The whole interface is controlled via 5 dedicated pins: **TDI**, **TMS**, **TCK**, **nTRST** and **TDO**. ***Figure 11-1: Test Access Port (TAP) Controller State Transitions*** shows the state transitions that occur in the TAP controller.

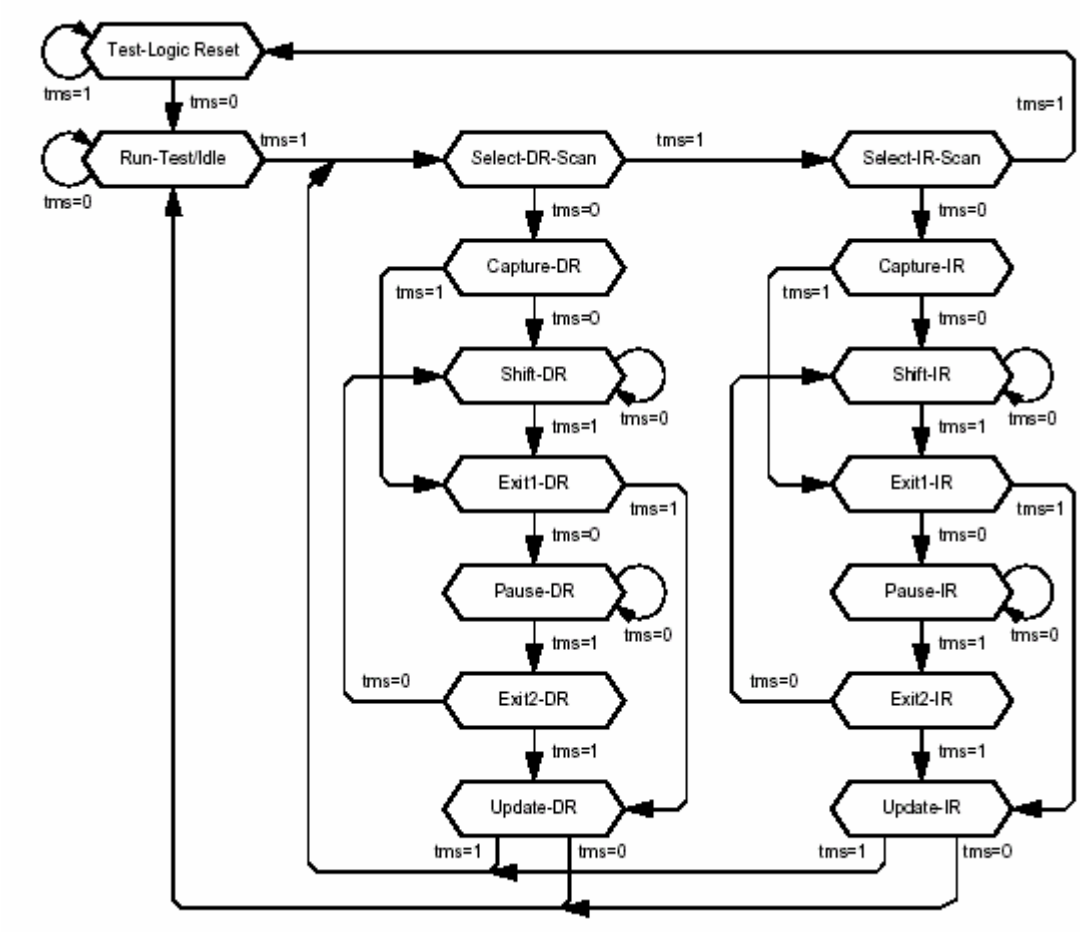


Figure 10-1. Test Access Port(TAP) Controller State Transitions

10.3.1 Reset

The boundary-scan interface includes a state-machine controller (the TAP controller). A pulldown resistor is included in the **nTRST** pad which holds the TAP controller state machine in a safe state after power up. In order to use the boundary scan interface, **nTRST** should be driven HIGH to take the TAP state machine out of reset.
The action of reset (either a pulse or a DC level) is as follows:
• System mode is selected (i.e. the boundary scan chain does NOT intercept any of the signals passing between the pads and the core).
• IDcode mode is selected. If **TCK** is pulsed, the contents of the ID register will be clocked out of **TDO**.

Note The TAP controller inside HMS30C7210 contains a scan chip register which is reset to the value b0011 thus selecting the boundary scan chain. If this register is programmed to any value other than b0011, then it must be reprogrammed with b0011 or a reset applied before boundary scan operation can be attempted.

10.3.2 Pull-up Register

The IEEE 1149.1 standard requires pullup resistors in the input pins. However, to ensure safe operation an internal pulldown is present in the **nTRST** pin and therefore will have to be driven HIGH when using this interface.

Pin Name	Internal Resistor
TCLK	Pull-up
nTRST	Pull-down
TMS	Pull-up
TDI	Pull-up

10.3.3 Instruction Register

The instruction register is 4 bits in length.
There is no parity bit. The fixed value loaded into the instruction register during the CAPTURE-IR controller state is: 0001.

10.3.4 Public Instructions

The following public instructions are supported:

Instruction	Binary Code
EXTEST	0000
SAMPLE/PRELOAD	0011
CLAMP	0101
HIGHZ	0111
CLAMPZ	1001
INTEST	1100
IDCODE	1110
BYPASS	1111

In the descriptions that follow, **TDI** and **TMS** are sampled on the rising edge of **TCK** and all output transitions on **TDO** occur as a result of the falling edge of **TC**K.

EXTEST (0000)

The BS (boundary-scan) register is placed in test mode by the EXTEST instruction.The EXTEST instruction connects the BS register between **TDI** and **TDO**.When the instruction register is loaded with the EXTEST instruction, all the boundary-scan cells are placed in their test mode of operation.
In the CAPTURE-DR state, inputs from the system pins and outputs from the boundary-scan output cells to the system pins are captured by the boundary-scan cells. In the SHIFT-DR state, the previously captured test data is shifted out of the BS register via the **TDO** pin, whilst new test data is shifted in via the **TDI** pin to the BS register parallel input latch. In the UPDATE-DR state, the new test data is transferred into the BS register parallel output latch. Note that this data is applied immediately to the system logic and system pins. The first EXTEST vector should be clocked into the boundary-scan register, using the SAMPLE/PRELOAD instruction, prior to selecting EXTEST to ensure that known data is applied to the system logic.

SAMPLE/PRELOAD (0011)

The BS (boundary-scan) register is placed in normal (system) mode by the SAMPLE/PRELOAD instruction.
The SAMPLE/PRELOAD instruction connects the BS register between **TDI** and **TDO**. When the instruction register is loaded with the SAMPLE/PRELOAD instruction, all the boundary-scan cells are placed in their normal system mode of operation.
In the CAPTURE-DR state, a snapshot of the signals at the boundary-scan cells is taken on the rising edge of **TCK**. Normal system operation is unaffected. In the SHIFT-DR state, the sampled test data is shifted out of the BS register via the **TDO** pin, whilst new data is shifted in via the **TDI** pin to preload the BS register parallel input latch. In the UPDATE-DR state, the preloaded data is transferred into the BS register parallel output latch. Note that this data is not applied to the system logic or system pins while the SAMPLE/PRELOAD instruction is active. This instruction should be used to preload the boundary-scan register with known data prior to selecting the INTEST or EXTEST instructions.

CLAMP (0101)

The CLAMP instruction connects a 1 bit shift register (the BYPASS register) between **TDI** and **TDO**. When the CLAMP instruction is loaded into the instruction register, the state of all output signals is defined by the values previously loaded into the boundary-scan register. A guarding pattern should be pre-loaded into the boundary-scan register using the SAMPLE/PRELOAD instruction prior to selecting the CLAMP instruction. In the CAPTURE-DR state, a logic 0 is captured by the bypass register. In the SHIFT-DR state, test data is shifted into the bypass register via **TDI** and out via **TDO** after a delay of one **TCK** cycle. Note that the first bit shifted out will be a zero. The bypass register is not affected in the UPDATE-DR state.

HIGHZ (0111)

The HIGHZ instruction connects a 1 bit shift register (the BYPASS register) between **TDI** and **TDO**. When the HIGHZ instruction is loaded into the instruction register, all outputs are placed in an inactive drive state. In the CAPTURE-DR state, a logic 0 is captured by the bypass register. In the SHIFT-DR state, test data is shifted into the bypass register via **TDI** and out via **TDO** after a delay of one **TCK** cycle. Note that the first bit shifted out will be a zero. The bypass register is not affected in the UPDATE-DR state.

CLAMPZ (1001)

The CLAMPZ instruction connects a 1 bit shift register (the BYPASS register) between **TDI** and **TDO**. When the CLAMPZ instruction is loaded into the instruction register, all outputs are placed in an inactive drive state, but the data supplied to the disabled output drivers is derived from the boundary-scan cells. The purpose of this instruction is to ensure, during production testing, that each output driver can be disabled when its data input is either a 0 or a 1. A guarding pattern (specified for this device at the end of this section) should be pre-loaded into the boundary-scan register using the SAMPLE/PRELOAD instruction prior to selecting the CLAMPZ instruction. In the CAPTURE-DR state, a logic 0 is captured by the bypass register. In the SHIFT-DR state, test data is shifted into the bypass register via **TDI** and out via **TDO** after a delay of one **TCK** cycle.
Note that the first bit shifted out will be a zero. The bypass register is not affected in the UPDATE-DR state.

INTEST (1100)

The BS (boundary-scan) register is placed in test mode by the INTEST instruction. The INTEST instruction connects the BS register between **TDI** and **TD**O. When the instruction register is loaded with the INTEST instruction, all the boundary-scan cells are placed in their test mode of operation. In the CAPTURE-DR state, the complement of the data supplied to the core logic from input boundary-scan cells is captured, while the true value of the data that is output from the core logic to output boundary- scan cells is captured. Note that CAPTURE-DR captures the complemented value of the input cells for testability reasons. In the SHIFT-DR state, the previously captured test data is shifted out of the BS register via the **TDO** pin, whilst new test data is shifted in via the **TDI** pin to the BS register parallel input latch. In the UPDATE-DR state, the new test data is transferred into the BS register parallel output latch. Note that this data is applied immediately to the system logic and system pins. The first INTEST vector should be clocked into the boundary-scan register, using the SAMPLE/PRELOAD instruction, prior to selecting INTEST to ensure that known data is applied to the system logic. Single-step operation is possible using the INTEST instruction.

IDCODE (1110)

The IDCODE instruction connects the device identification register (or ID register) between **TDI** and **TD**O. The ID register is a 32-bit register that allows the manufacturer, part number and version of a component to be determined through the TAP. The IDCODE returned will be that for the ARM720T core. When the instruction register is loaded with the IDCODE instruction, all the boundary-scan cells are placed in their normal (system) mode of operation. In the CAPTURE-DR state, the device identification code (specified at the end of this section) is captured by the ID register.
In the SHIFT-DR state, the previously captured device identification code is shifted out of the ID register via the **TDO** pin, whilst data is shifted in via the **TDI** pin into the ID register. In the UPDATE-DR state, the ID register is unaffected.

BYPASS (1111)

The BYPASS instruction connects a 1 bit shift register (the BYPASS register) between**TDI** and **TD**O. When the BYPASS instruction is loaded into the instruction register, all the boundary-scan cells are placed in their normal (system) mode of operation. This instruction has no effect on the system pins. In the CAPTURE-DR state, a logic 0 is captured by the bypass register. In the SHIFT-DR state, test data is shifted into the bypass register via **TDI** and out via **TDO** after a delay of one **TCK** cycle. Note that the first bit shifted out will be a zero. The bypass register is not affected in the UPDATE-DR state.

10.3.5 Test Data Register

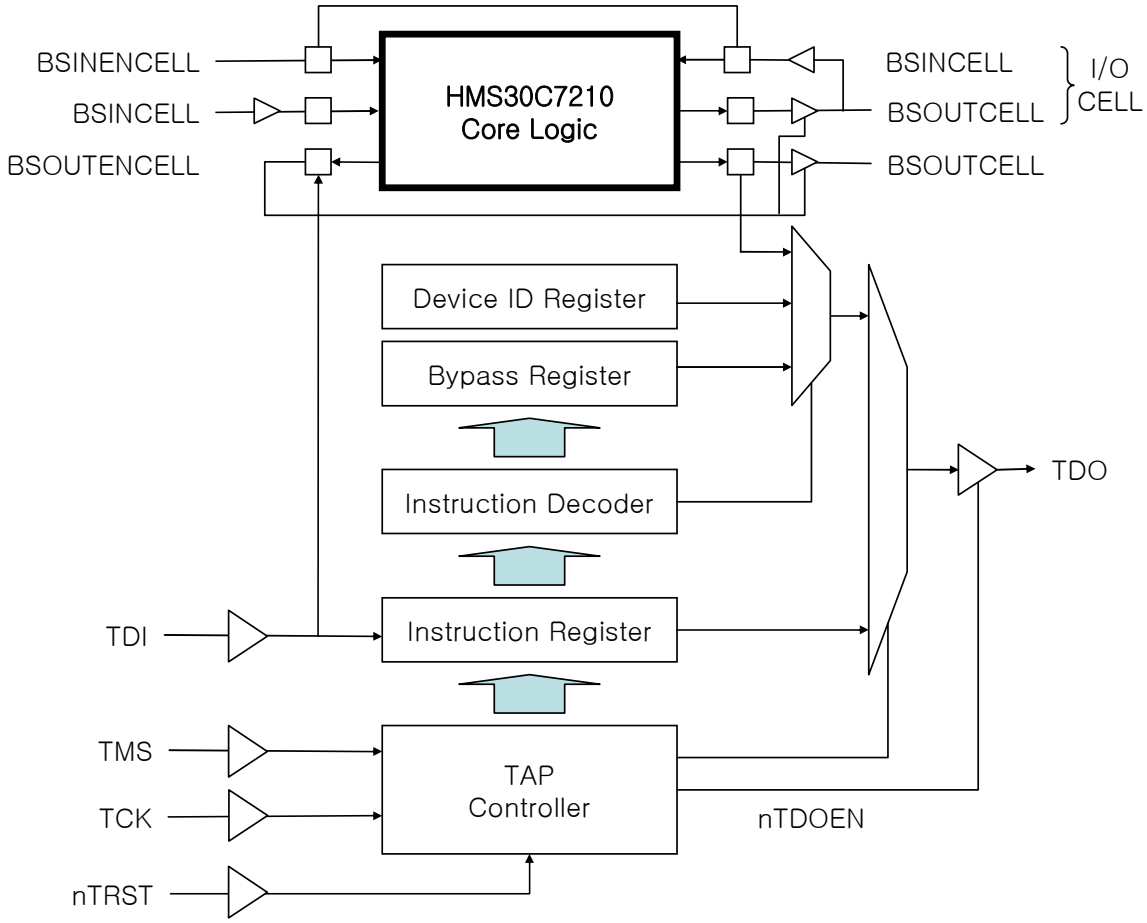


Figure 10-2. Boundary Scan Block Diagram

Bypass Register

Purpose: This is a single bit register which can be selected as the path between **TDI** and **TDO** to allow the device to be bypassed during boundary-scan testing.
Length: 1 bit
Operating Mode: When the BYPASS instruction is the current instruction in the instruction register, serial data is transferred from **TDI** to **TDO** in the SHIFT-DR state with a delay of one **TCK** cycle.
There is no parallel output from the bypass register.
A logic 0 is loaded from the parallel input of the bypass register in the CAPTURE-DR state.

Boundary Scan (BS) Register

Purpose: The BS register consists of a serially connected set of cells around the periphery of the device, at the interface between the core logic and the system input/output pads. This register can be used to isolate the core logic from the pins and then apply tests to the core logic, or conversely to isolate the pins from the core logic and then drive or monitor the system pins. Operating modes: The BS register is selected as the register to be connected between **TDI** and **TDO** only during the SAMPLE/PRELOAD, EXTEST and INTEST instructions. Values in the BS register are used, but are not changed, during the CLAMP and CLAMPZ instructions. In the normal (system) mode of operation, straight-through connections between the core logic and pins are maintained and normal system operation is unaffected. In TEST mode (i.e. when either EXTEST or INTEST is the currently selected instruction), values can be applied to the core logic or output pins independently of the actual values on the input pins and core logic outputs respectively. On the HMS30C7202 all of the boundary scan cells include an update register and thus all of the pins can be controlled in the above manner.
Additional boundary-scan cells are interposed in the scan chain in order to control the enabling of tristateable buses. The values stored in the BS register after power-up are not defined. Similarly, the values previously clocked into the BS register are not guaranteed to be maintained across a Boundary Scan reset (from forcing **nTRST** LOW or entering the Test Logic Reset state).

Single-step Operation

HMS30C7210 is a static design and there is no minimum clock speed. It can therefore be single-stepped while the INTEST instruction is selected and the PLLs are bypassed.
This can be achieved by serializing a parallel stimulus and clocking the resulting serial vectors into the boundary-scan register. When the boundary-scan register is updated, new test stimuli are applied to the core logic inputs; the effect of these stimuli can then be observed on the core logic outputs by capturing them in the boundary-scan register.

10.3.6 Boundary Scan Interface Signals

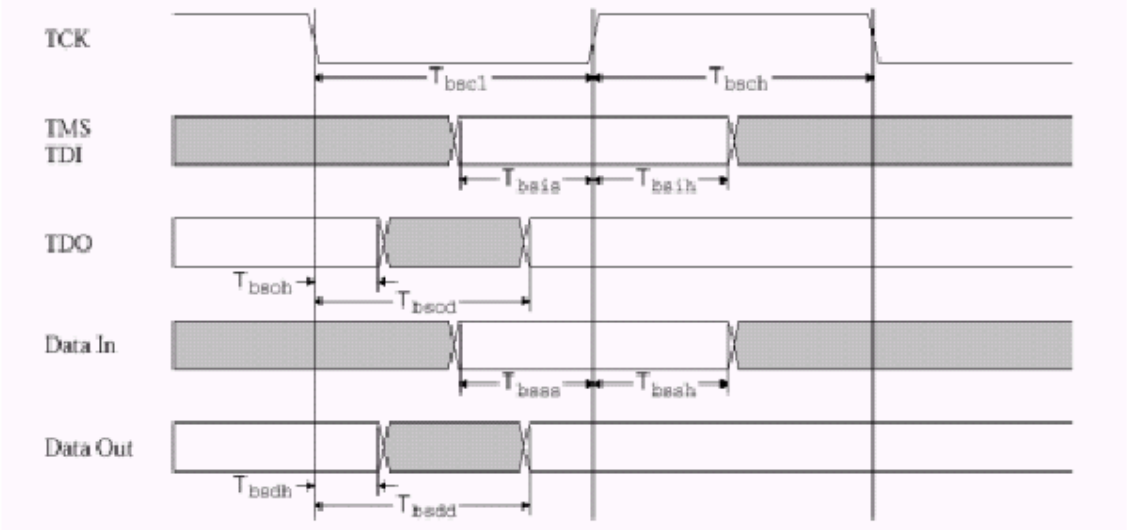


Figure 10-3. Boundary Scan General Timing

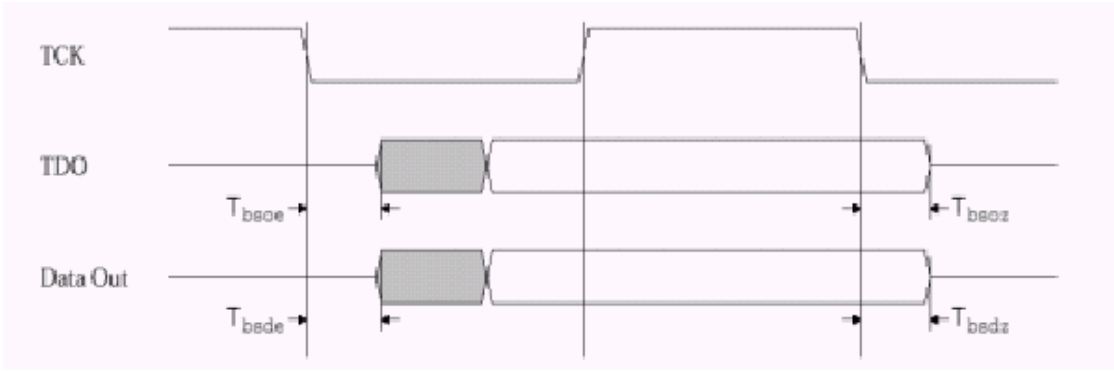


Figure 10-4. Boundary Scan Tri-state Timing

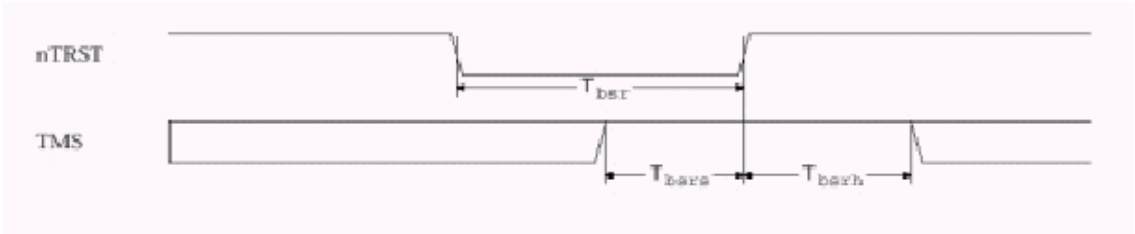


Figure 10-5. Boundary Scan Reset Timing

Symbol	Parameter	Min	Max
Tbscl	TCK low period	50	-
Tbsch	TCK high period	50	-
Tbsis	TMS, TDI setup to TCKr	0	-
Tbsih	TMS, TDI hold from TCKr	2	-
Tbsoh	TDO output hold from TCKf	3	-
Tbsod	TDO output delay from TCKf	-	20
Tbsss	Test mode Data in setup to TCKr	2	-
Tbssh	Test mode Data in hold from TCKf	5	-
Tbsdh	Test mode Data out hold from TCKf	3	-
Tbsdd	Test mode Data out delay from TCKf	-	20
Tbsoe	TDO output enable delay from TCKf	2	15
Tbsoz	Test mode Data enable delay from TCKf	2	15
Tbsde	TDO output disable delay from TCKf	2	15
Tbsdz	Test mode Data disable delay from TCKf	2	15
Tbsr	NTRST minimun pulse width	25	-
Tbsrs	TMS setup to nTRSTr	20	-
Tbsrh	TMS hold from nTRSTr	20	-

The AC parameters are based on simulation results using 0.0pf circuit signal loads. Delays should be calculated using manufacturers output derating values for the actual circuit capacitance loading.

The correspondence between boundary-scan cells and system pins, system direction controls and system output enables is shown below. The cells are listed in the order in which they are connected in the boundary-scan register, starting with the cell closest to TDI. All outputs are three-state outputs. All boundary-scan register cells at input pins can apply tests to the on-chip system logic.
EXTEST/CLAMP guard values specified in the table below should be clocked into the boundary-scan register (using the SAMPLE/PRELOAD instruction) before the EXTEST, CLAMP or CLAMPZ instructions are selected to ensure that known data is applied to the system logic during the test. The INTEST guard values shown in the table below should be clocked into the boundary-scan register (using the SAMPLE/PRELOAD instruction) before the INTEST instruction is selected to ensure that all outputs are disabled. An asterisk in the guard value column indicates that any value can be submitted (as test requires), but ones and zeros should always be placed as shown.

11 ELECTRICAL CHARACTERISTICS

11.1 Absolute Maximum Ratings

Symbol	Parameter	Typical	Units	VDD Condition
P_{RUN}	RUN Mode Power	391	mW	@ 3.3V
P_{SLOW}	SLOW Mode Power	355	mW	@ 3.3V
P_{IDLE}	IDLE Mode Power	276	mW	@ 3.3V
P_{PD}	Deep-Sleep Mode Power	3.3	uW	@ 3.3V
P_{RTC}	RTC Power	36	uW	@ 3.0V

Table 11-1. Maximum Ratings

- Core / IO / Analog VDD are 3.3V
- Operating frequency is 60MHz.
- In RUN/SLOW Mode CPU generated image pattern (on SDRAM) and displayed to 640x480 Color STN LCD (8bpp). In Slow Mode CPU runs with “half clock speed” (Bus Clock).
- IDLE Mode went to IDLE state from LCD SDRAM loop.
- RTC Power is independent. RTC can be operated in system power off mode. At this time RTC power is connected to a battery (3.0V).
- RUN / SLOW / IDLE / DEEPSLEEP Power consumption is estimated without RTC power dissipation.

Recommended Operating Range

Symbol	Parameter	Min	Max	Units
VDD (3.3V)	DC Power Supply Voltage (3.3V) → use for I/O	3.0	3.6	V
VDD (3.3V)	DC Power Supply Voltage (3.3V) → use for a Core	3.0	3.6	V
T_{OPR}	Operating Temperature (Industrial Temperature)	-40	85	℃

Table 11-2. Operating Range

11.2 DC characteristics

All characteristics are specified at Vdd=3.0 to 3.6 volts ans Vss=0 volts, over an operating temperature range of 0 to 100 °C

CMOS Pins

		0℃	100℃	Conditions	
Symbol	**Parameter**	**Min**	**Max**	**VDD**	
V_{IL}	Low-level Input Voltage	-0.5V	0.3xVDD	2.7V to 3.6V	Guaranteed Input Low Voltage
V_{IH}	High-level Input Voltage	0.7xVDD	VDD+0.5V	2.7V to 3.6V	Guaranteed Input High Voltage
V_{OL}	Low-level Output Voltage	-	VSS+0.1V	2.7V	IOL = 0.8mA
V_{OH}	High-level Output Voltage	VDD–0.1V	-	2.7V	IOH = 0.8mA
I_I	Input Current at maximum voltage	-	1mA	2.7V to 3.6V	Input = 5.5V

Table 11-3. CMOS signal pin characteristics

TTL Compatible Pin

		0℃	100℃	Conditions	
Symbol	**Parameter**	**Min**	**Max**	**VDD**	
V_{IL}	Low-level Input Voltage	-0.5V	0.8V	2.7V to 3.6V	Guaranteed Input Low Voltage
V_{IH}	High-level Input Voltage	2.0V	VDD+0.5V	2.7V to 3.6V	Guaranteed Input High Voltage
V_{OL}	Low-level Output Voltage	-	0.4V	2.7V	IOL,2 to 0.8mA Depending on Cell
V_{OH}	High-level Output Voltage	2.4V	-	2.7V	IOH,2 to 0.8mA Depending on Cell
I_I	Input Current at maximum voltage	-	1mA	2.7V to 3.6V	Input = 5.5V

Table 11-4. TTL signal pin characteristics

I/O Circuit Pull-up Pin

The following current values are used for I/Os with internal pull-up devices.

	Min Current (at pad = 0V)	Max Current (at pad = 0V)
3.3V Pull-up	-30uA	-146uA
Equivalent resistance	88.3k Ohms	22.6k Ohms

I/O Circuit Pull-down Pin

The following current values are used for I/Os with internal pull-down devices.

	Min Current (at pad = 2.65V)	Max Current (at pad = 3.6V)
3.3V Pull-down	31uA	159uA
Equivalent resistance	85.5k Ohms	22.6k Ohms

11.3 A/D Converter Electrical Characteristics

Symbol	Paramter	Test Condition	Minimum	Typical	Maximum	Unit
I_{dd}	Normal	aclk = 3.704MHz Input = avref Fin = 4kHz ramp		4.0		mA
	Power Down	aclk = 3.704MHz			40	uA
an*	Analog Input Voltage		AVSS		avref	V
Accuracy	Resolution				10	Bits
INL	Integral Non-linearity	aclk = 3.704MHz Input = 0-avref(V) (Fin = 4kHz ramp)			±2.0	LSB
DNL	Differential Non-linearity	aclk = 3.704MHz Input = 0-avref(V) (Fin = 4kHz ramp)			±1.0	LSB
SNR	Signal-to-Noise Ratio	Fsample = 231.5ksps Fin = 4KHz	50	54		dB
SNDR	Signal-to-Noise Distortion Ratio		48	52		dB
aclk				3.704		MHz
t_c	Conversion Time		1	4	8	us
avref*	Analog Reference Voltage				AVDD	V
T_{cal}	Power-up Time	Calibration time	1.2			ms
THD	Total Harmonic Distortion		50	54		dB
AVDD*	Analog Power		3.0	3.3	3.6	V
DVDD	Digital Power		3.0	3.3	3.6	V
Fin	Analog Input Frequency				60	KHz

Table 11-5. A/D converter characteristics

- AVSS ≤an0~3 ≤avref ≤AVDD

12 APPENDIX

The Method of clearing Status register

Peri. Name	Register Name	Width	Address	Method of clearing
PMU	PMURSR	27	0x8001.0020	Write “1”
*LCD	LcdStatus	4	0x8005.2004	Write “1”
INTC	STATUS	29	0x8005.0008	Clear interrupt source
USB	INTSTAT	20	0x8005.100C	Write “1”
ADCIF	ADCISR	3	0x8005.3010	Write “1”
**UART(Smart Card)	LSR	8	UxBase+0x14	Read register
	MSR	8	UxBase+0x18	
***SSI	SSPSR	5	SSIBase+0x0c	Write “1”
SMC	SMCSTAT	32	0x8005.C01C	Write “0”
TIMER	TOPSTAT	4	0x8005.D084	Write “1”
	T(0/1/2/3)STAT	1	0x8005.D0(0/2/4/6)C	
Watchdog Timer	WDTSTAT	2	0x8005.E004	Read register
RTC	RTCSTAT	3	0x8005.F004	Write “1”
2WSI	STATUSREG	16	0x8006.0008	Write “1”
Matrix KBD	KBSR	1	0x8006.1018	Write “1”

*** LCD (Method of clearing) – reference : 9.1.2.2 LCD Controller Status/Mask and Interrupt Registers (LcdStatus, LcdStatusM, and LcdInterrupt)**

- LcdStatus[3] : Write anything at LcdDBAR register or Enable LcdEn signal at Lcd control register[0]
- LcdStatus[2] : Write “1”
- LcdStatus[1] : Write anything at LcdDBAR register
- LcdStatus[0] : Write “1”

**** UART (Address)**

- UxBase : 0x8005.4000 (UART0), 0x8005.5000 (UART1), 0x8005.6000 (UART2), 0x8005.7000 (UART3),
- 0x8005.8000 (UART4), 0x8005.9000 (UART5)

***** SSI (Address)**

- SSIBase: 0x8005.A000 (SSI0), 0x8005.B000(SSI1)