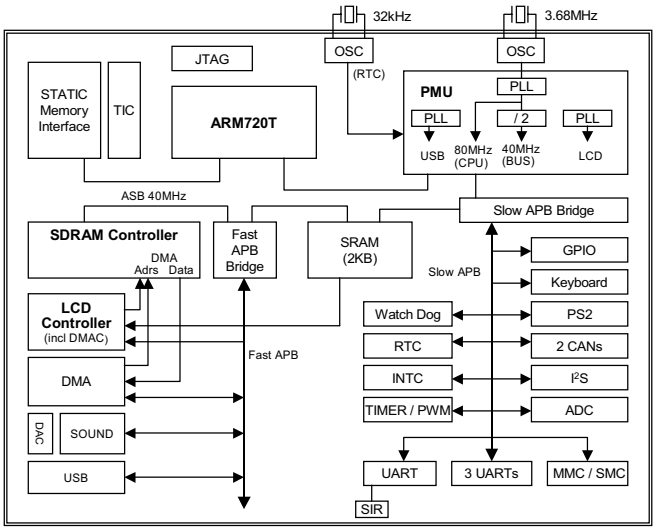
FEATURES

- **32-bit ARM7TDMI RISC static CMOS CPU core**
- **8Kbytes combined instruction/data cache**
- **Memory management unit for WindowsCE**
- **Supports Little Endian operating system**
- **2Kbytes SRAM for internal buffer memory**
- **On-chip peripherals with individual power-down:**
  - Multi-channel DMA
  - 4 Timer Channels with Watch Dog Timer
  - Intelligent Interrupt Controller
  - Memory controller for ROM, Flash, SRAM, SDRAM
  - Power management unit
  - LCD Controller for mono/color STN and TFT LCD
  - Real-time clock (32.768kHz oscillator)
  - Infrared communications (SIR support)
  - 4 UARTs (16C550 compatible)
  - PS/2 External Keyboard / Mouse interface
  - 2 Pulse-Width-Modulated (PWM) interface
  - Matrix Keyboard control interface (8*8)
  - GPIO
  - MMC / SMC Card interface
  - USB (target)
  - On-chip ADC and interface module (Battery Check, Audio In, Touch Panel)
  - On-chip DAC and interface module (8 Bit Stereo Audio Output)
  - PLL



Functional Block Diagram

- **JTAG debug interface and boundary scan**
- **0.20um Low Power CMOS Process**
- **1.8V internal / 3.3V IO supply voltage**
- **256-pin QFP / BGA package**
- **100MHz operation frequency**
- **Low power consumption**

OVERVIEW

The HMS30C7202 is a highly integrated low power microprocessor for personal digital assistants, and other applications described below. The device incorporates an ARM720T CPU and system interface logic to interface with various types of

devices. HMS30C7202 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 external SDRAM memory, analog functions like ADC, DAC, PLLs. Intelligent interrupt controller and internal 2Kbytes SRAM can support very efficient interrupt service execution. The HMS30C7202 also supports voice recording, sound playback and a touch panel interface. UART, USB, PS2 and CAN provide serial communication channels for external systems. I2S interface enables to use external DAC for high quality audio output. The power management features result in very low power consumption. The HMS30C7202 provides an excellent solution for personal digital assistants (PDAs), and data terminal running the Microsoft Windows CE operating system. Other applications include smart phones, Internet appliances, car navigation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 ARCHITECTURAL OVERVIEW

1.1 Processor

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

1.2 Video

The HMS30C7202 has direct support for mono and color passive LCD displays, as well as color TFT LCD displays, with resolution programmable up to 640x480 VGA resolution.

1.3 Memory

HMS30C7202 incorporates two separate memory interfaces. A high-speed 16-bit wide interface connects directly to one to two 16, 64,128 or 256MBit SDRAM devices, supporting DRAM memory sizes in the range 2 to 64MB. In addition, a separate lower speed 32- bit 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. Since the ROM and SDRAM interfaces are separate, the ARM processor core can access O/S code in ROM simultaneously with video DMA access to the SDRAM, thus increasing total effective memory bandwidth, and hence overall performance.

1.4 Internal Bus Structure

The GMS30C7202 internal bus organization is based upon the AMBA standard, but with some minor modifications to the peripheral buses (the APBs). There are three main buses in the GMS30C7202:

1. The main system bus (the ASB) to which the CPU and memory controllers are connected
2. The fast APB to which high-bandwidth peripherals are connected
3. The slow APB (to which timers, the UART and other low-bandwidth peripherals are connected)

There is also a separate video DMA bus.

1.4.1 ASB

The ASB is designed to allow the ARM to have continuous access to both the ROM interface and the SDRAM. 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 GMS30C7202 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 x.x.x Arbitration on page xxx.

1.4.2 Video bus

The video bus hosts the LCD controller DMA. The video bus consists of separate address inputs, a request / acknowledge to / from the SDRAM controller, a data bus. The LCD registers are programmed from the fast APB. The SDRAM controller arbitrates between ASB, VGA access requests. Video always has higher priority than ASB access requests. The splitting ASB/video bus allows slow ASB device accesses SDRAM without blocking video DMA.

1.4.3 APB

There are two APB buses. These are the fast and slow APB buses. The fast APB bus operates at the speed of the ASB (100 MHz), and hosts the USB interface, the sound output interface, the LCD registers, etc. These are the high performance peripherals, which are generally DMA targets. The slow APB peripherals generally operate at the UART crystal clock frequency of 3.6864MHz, though register access via the APB is at ASB speed.
The slow APB peripherals do not support DMA transfers. This arrangement of operating most of the peripherals from a slower clock, and reducing the load on the faster bus, results in significantly reduced power

consumption. Both APB buses connect to the main ASB bus via specially modified bridges. The slow 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.
The fast APB Bridge is modified from the normal AMBA Bridge, to allow DMA access to fast APB peripherals. Additional signals from the DMA controller to the APB bridge request select and acknowledge DMA transfers to and from DMA-aware peripherals.

1.5 SDRAM Controller

The SDRAM controller is a key part of the GMS30C7202 architecture. The SDRAM controller has two data ports - one for video DMA and one for the main ASB - and interfaces to a single 16-bit wide SDRAM. One to four 16, 64 or 128Mbit x16-bit devices are supported, giving a memory size ranging from 2 to 64Mbytes.
The main ASB and video DMA buses are independent, and operate concurrently. The video bus is always higher priority than the main bus.
The video interface consists of address, data and request inputs to the SDRAM controller. 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 Peripheral DMA

1.6.1 Overview

GMS30C7202 incorporates a three-channel, general-purpose DMA controller that operates on the ASB. The DMA controller is an AMBA compliant ASB bus master with a higher arbitration priority than either the ARM or Piccolo DSP coprocessor, to ensure low DMA latency. Since, however, the main ASB bus always has lower priority access to the SDRAM controller than the video bus, it will always get lower priority access to SDRAM than the LCD and VGA.

1.6.2 Transfer sizes

The devices that make use of the peripheral DMA are:

1. USB
2. Fast/Medium IR
3. Sound output

The USB and FIR are bi-directional but half-duplex, so only one DMA channel is required at a time. The data rate for the USB is 12Mbit/sec, which translates to 1.5Mbyte/sec. The data rate for the FIR is a maximum of 4Mbit/sec, which translates to 0.5Mbyte/sec. The sound output data rate is 88.2KB/sec. To ensure reasonable usage of SDRAM, APB and ASB bandwidth, the transfer sizes to these device are:

USB	Quad-word
FIR	Word
Sound	Word

The SDRAM controller will do a complete quad-word access for every SDRAM access. With the transfer sizes above, the approximate SDRAM bandwidth taken by the devices is:

USB	3%
FIR	4%
Sound	0.75%

The maximum total of SDRAM bandwidth taken by all three devices running concurrently is 7.75%.
DMA accesses to FIR and Sound blocks are fully AMBA compliant, meaning that a word transfer takes a minimum of two bus cycles to complete. The APB protocol however, for USB DMA accesses, has been slightly modified to allow burst accesses.

1.6.3 Fly-by

The DMA controller is tightly coupled to the fast APB Bridge. In order for the DMA Controller to start a transfer, it must first receive a DMA data request from one of the peripherals; it will then request mastership of the ASB. Once granted, the DMA Controller will retain mastership of the ASB until the requested DMA transaction is completed, which ensures correct data in the DMA peripherals (that is data in the DMA peripherals cannot be modified by the ARM processor while a DMA transfer is in progress).
The DMA transfer request is monitored by the Fast APB bridge, who will perform the correspondent APB transfer by inverting the read/write line with respect to the ASB, to generate a PWRITE signal on the APB. The DMA transfer is acknowledged on the APB by asserting a PSELDMA signal for the given peripheral. The data is timed by PSTB as on a normal APB transfer. The APB address PA is not used for DMA transfers.

The APB bridge receives two signals from the DMA controller called CHAN [1:0], which tells it which DMA channel (peripheral) the DMA access is for. All other information comes from monitoring the ASB bus signals. For example, the direction of transfer comes from BWRITE (the sense is inverted to get the APB signal), and when the SDRAM transfer completes, comes from the bridge monitoring the BWAIT ASB signal.

1.6.4 Timing

This is detailed in Chapter xxx, Fast AMBA Peripherals.

1.6.5 Slow APB peripherals

Since the DMA controller is not coupled with the slow APB Bridge, it is not possible to use DMA with devices on the slow APB bus. However, since devices on the slow APB bus are inherently low performance, this is not a serious restriction. Devices on the slow APB bus must use the ARM acting under interrupt control to simulate DMA. The highest data rate peripheral on the slow APB bus is the modem CODEC interface, at a maximum of 48KB/sec. The ARM FIQ is used to transfer data to the CODEC.

1.6.6 Sound output

In the GMS30C7202, the sound peripheral is located on the fast APB bus, and is supported by the DMA controller. (Note that this is compatible with some operating systems, which require DMA-support sound hardware.)

1.7 Peripherals and communications

Universal Serial Bus (USB) device controller
The USB device controller is used to transfer data from/to host system like PC in high-speed (12Mbits/s) mode. No external USB transceiver is necessary.
PS/2 Interface
The one PS/2 port is can be used with keyboard, mice or PS/2 compliant device. The PS/2 pins behave like open-drain I/Os in PS/w mode. When not used for PS/2, the pins used as GPIO.
Universal Asynchronous Receiver and Transmitter (UART)
The four UART ports are used. One of them supports full modem interface signals. Some pins are used as GPIO or matrix keyboard pins when not used for UART.
IrDA
IrDA uses one UART channel for its SIR transfer in 115Kbits/s speed. The pins are used as GPIO or matrix keyboard pins when not used for IrDA.
Controller Area Network (CAN)
The two CAN ports are used. The pins are used as GPIO when not used for CAN.
Multimedia Card (MMC), Solid State Floppy Disk Card (SSFDC)
MMC or SSFDC memory card can be used as storage device. The pins are used as GPIO when not used for MMC or SSFDC.
I2S
I2S transfers serialized audio data to external device using I2S format for high quality sound. The pins are used as GPIO when not used for I2S.
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 up to 64 keys. The pins are used as GPIO when not used for matrix keyboards.
General Purpose DMA Channel
One DMA channel is provided for external device that needs DMA access. The pins are used as GPIO when not used for DMA.
DAC
On chip DAC provides 8-bit audio stereo sound.
ADC
On chip ADC receives 5 analog input signals for touch panel, audio input and two battery levels. No external transistor switch is necessary for touch panel operation.
PLL
CPU, video and USB clocks are generated by three PLL with 3.6864 MHz input clock.

1.8 Power management

The HMS30C7202 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 HMS30C7202 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 more slowly than usual, 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 HMS30C7202 is designed for battery-powered portable applications and incorporates innovative design features in the bus structure and the PMU to reduce power consumption. The slow APB bus allows peripherals to be clocked slowly hence reducing power consumption. The use of three 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.

1.8.1 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.

1.8.2 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), and the SDRAM controller is performing normal refresh.

Slow
The system operates normally, except the ARM is placed into Fast Bus mode, and hence is clocked at half its normal rate.

Idle
In this mode, the PMU becomes the bus master until there is an interrupt for the CPU, or the peripheral DMA controller requests mastership of the bus.

Sleep
The SDRAM is placed into self-refresh mode, and internal clocks are gated off. This mode can only be entered from Idle mode (that is, the PMU must be ASB master before this mode can be entered). The PMU must get bus mastership to ensure that the system is stopped in a safe state and not, for example, halfway through an SDRAM write. Usually this state is only to be entered briefly, on the way to entering deep sleep mode.

Deep Sleep
In deep sleep mode, the 3.6864MHz oscillator and the PLLs are disabled. This is the lowest power state available. Only the 32kHz oscillator runs. The real time clock and wakeup sections of the PMU are operated from this clock. Everything else is powered down, and SDRAM is in self-refresh mode. This is the normal system “off” mode. Sleep and Deep Sleep modes are exited either by a user wake-up event (generally pressing the “On” key), an RTC wake-up alarm, a device reset request, or by a modem ring indicate event. These interrupt sources go directly to the PMU. In addition, the modem ring indicate signal also goes to the normal interrupt controller to signal an interrupt if there is a ring indicate event in a non-sleep mode.

1.9 Test and debug

The HMS30C7202 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 HMS30C7202. 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 PIN DESCRIPTION

2.1 256-Pin PQFP Pin Diagram

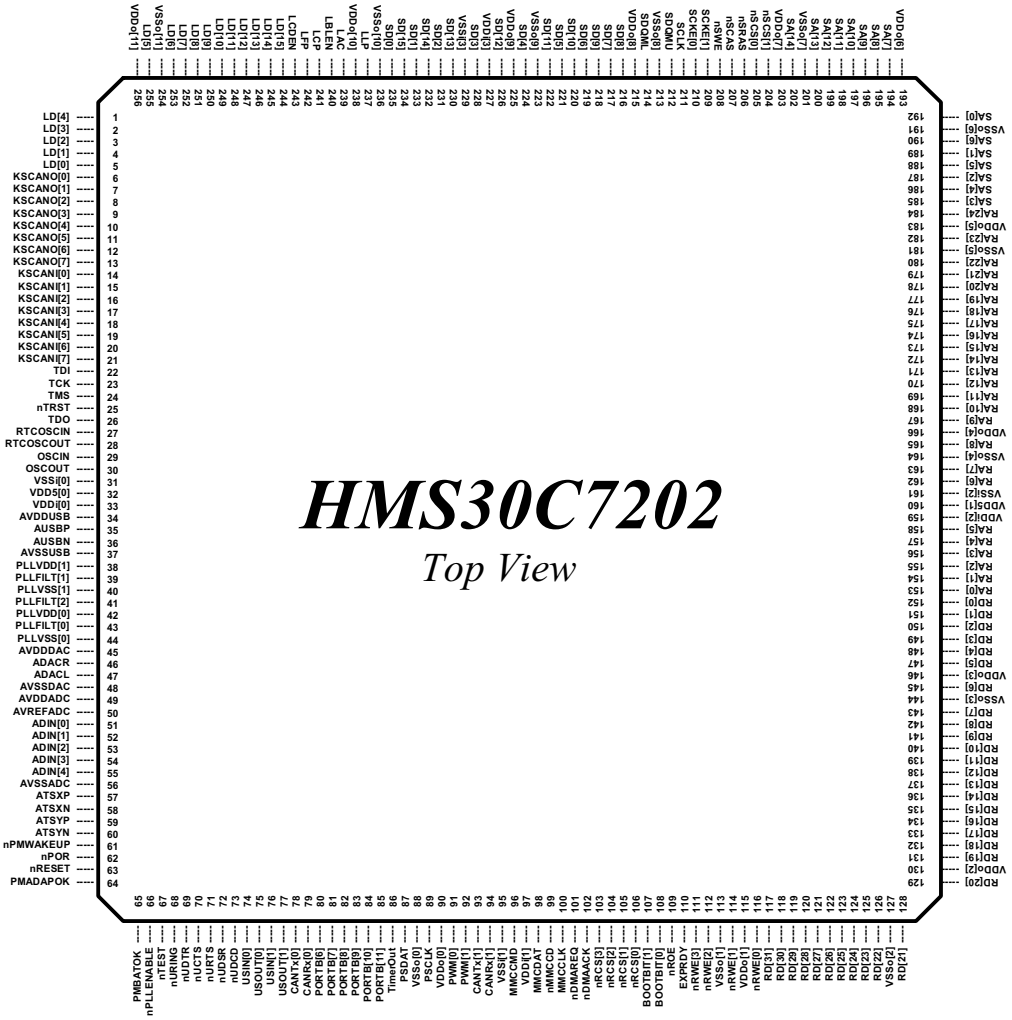


Figure 2-1 Pin Location and Signal Cross-References (Top View)

Pin No.	PAD Name	Pin No.	PAD Name	Pin No.	PAD Name	Pin No.	PAD Name
1	LD[4]	65	PMBATOK	129	RD[20]	193	uVDDo6
2	LD[3]	66	nPLLENABLE	130	uVDDo2	194	SA[7]
3	LD[2]	67	nTEST	131	RD[19]	195	SA[8]
4	LD[1]	68	nURING	132	RD[18]	196	SA[9]
5	LD[0]	69	nUDTR	133	RD[17]	197	SA[10]
6	KSCANO[0]	70	nUCTS	134	RD[16]	198	SA[11]
7	KSCANO[1]	71	nURTS	135	RD[15]	199	SA[12]
8	KSCANO[2]	72	nUDSR	136	RD[14]	200	SA[13]
9	KSCANO[3]	73	nUDCD	137	RD[13]	201	uVSSo7
10	KSCANO[4]	74	USIN[0]	138	RD[12]	202	SA[14]
11	KSCANO[5]	75	USOUT[0]	139	RD[11]	203	uVDDo7
12	KSCANO[6]	76	USIN[1]	140	RD[10]	204	nSCS[1]

13	KSCANO[7]	77	USOUT[1]	141	RD[9]	205	nSCS[0]
14	KSCANI[0]	78	CANTx[0]	142	RD[8]	206	nSRAS
15	KSCANI[1]	79	CANRx[0]	143	RD[7]	207	nSCAS
16	KSCANI[2]	80	PORTB[6]	144	uVSSo3	208	nSWE
17	KSCANI[3]	81	PORTB[7]	145	RD[6]	209	SCKE[1]
18	KSCANI[4]	82	PORTB[8]	146	uVDDo3	210	SCKE[0]
19	KSCANI[5]	83	PORTB[9]	147	RD[5]	211	SCLK
20	KSCANI[6]	84	PORTB[10]	148	RD[4]	212	SDQMU
21	KSCANI[7]	85	PORTB[11]	149	RD[3]	213	uVSSo8
22	TDI	86	TimerOut	150	RD[2]	214	SDQML
23	TCK	87	PSDAT	151	RD[1]	215	uVDDo8
24	TMS	88	uVSSo0	152	RD[0]	216	SD[8]
25	nTRST	89	PSCLK	153	RA[0]	217	SD[7]
26	TDO	90	uVDDo0	154	RA[1]	218	SD[9]
27	RTCOSCIN	91	PWM[0]	155	RA[2]	219	SD[6]
28	RTCOSCOUT	92	PWM[1]	156	RA[3]	220	SD[10]
29	OSCIN	93	CANTx[1]	157	RA[4]	221	SD[5]
30	OSCOUT	94	CANRx[1]	158	RA[5]	222	SD[11]
31	uVSSi0	95	uVSSi1	159	uVDDi2	223	uVSSo9
32	uVDD5_0	96	MMCCMD	160	uVDD5_1	224	SD[4]
33	uVDDi0	97	uVDDi1	161	uVSSi2	225	uVDDo9
34	AVDDUSB	98	MMCDAT	162	RA[6]	226	SD[12]
35	AUSBP	99	nMMCCD	163	RA[7]	227	uVDDi3
36	AUSBN	100	MMCCLK	164	uVSSo4	228	SD[3]
37	AVSSUSB	101	nDMAREQ	165	RA[8]	229	uVSSi3
38	PLLVDD[1]	102	nDMAACK	166	uVDDo4	230	SD[13]
39	PLLFILT[1]	103	nRCS[3]	167	RA[9]	231	SD[2]
40	PLLVSS[1]	104	nRCS[2]	168	RA[10]	232	SD[14]
41	PLLFILT[2]	105	nRCS[1]	169	RA[11]	233	SD[1]
42	PLLVDD[0]	106	nRCS[0]	170	RA[12]	234	SD[15]
43	PLLFILT[0]	107	BOOTBIT[1]	171	RA[13]	235	SD[0]
44	PLLVSS[0]	108	BOOTBIT[0]	172	RA[14]	236	uVSSo10
45	AVDDDAC	109	nROE	173	RA[15]	237	LLP
46	ADACR	110	EXPRDY	174	RA[16]	238	uVDDo10
47	ADACL	111	nRWE[3]	175	RA[17]	239	LAC
48	AVSSDAC	112	nRWE[2]	176	RA[18]	240	LBLEN
49	AVDDADC	113	uVSSo1	177	RA[19]	241	LCP
50	AVREFADC	114	nRWE[1]	178	RA[20]	242	LFP
51	ADIN[0]	115	uVDDo1	179	RA[21]	243	LCDEN
52	ADIN[1]	116	nRWE[0]	180	RA[22]	244	LD[15]
53	ADIN[2]	117	RD[31]	181	uVSSo5	245	LD[14]
54	ADIN[3]	118	RD[30]	182	RA[23]	246	LD[13]
55	ADIN[4]	119	RD[29]	183	uVDDo5	247	LD[12]
56	AVSSADC	120	RD[28]	184	RA[24]	248	LD[11]
57	ATSXP	121	RD[27]	185	SA[3]	249	LD[10]
58	ATSXN	122	RD[26]	186	SA[4]	250	LD[9]
59	ATSYP	123	RD[25]	187	SA[2]	251	LD[8]
60	ATSYN	124	RD[24]	188	SA[5]	252	LD[7]
61	nPMWAKEUP	125	RD[23]	189	SA[1]	253	LD[6]
62	nPOR	126	RD[22]	190	SA[6]	254	uVSSo11
63	nRESET	127	uVSSo2	191	uVSSo6	255	LD[5]
64	PMADAPOK	128	RD[21]	192	SA[0]	256	uVDDo11

2.2 Pin Descriptions

Table 2-2 describes the function of all the external signals to the GMS30C7202.

Type	Description	Type	Description
O	Output	OA	Analog Output
I	Input	IA	Analog Input
IO	Input/Output	IOA	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
m	Suffix to multiple function pin		

Table 2-1 Pin Signal Type Definition

2.2.1 External Signal Functions

Function	Signal Name	Signal Type	Description
LCD	LD[15:0]	Om	LCD data bus. Allow 5:6:5 TFT, color (using [7:0]) or mono, using [3:0] or [7:0]
	LCP	O	LCD clock pulse
	LLP	O	LCD line pulse (Hsync for TFT)
	LFP	O	LCD frame pulse (Vsync for TFT)
	LAC	O	LCD AC bias (clock enable for TFT)
	LCDEN	O	Display enable signal for LCD. Enables high voltage to LCD
	LBLEN	Om	LCD backlight enable
Static Memory Interface	RA[24:0]	O	ROM address bus
	RD[31:0]	IOm	ROM data bus
	nRCS[3:0]	Om	ROM chip select outputs
	nROE	O	ROM output enable signal
	nRWE[3:0]	Om	ROM write enable signals
	EXPRDY	I	Wait from external I/O
	BOOTBIT[1:0]	I	8/16/32 bit ROM selection
SDRAM Interface	SCLK	O	SDRAM clock output
	SCKE[1:0]	O	SDRAM clock enable output
	nSRAS	O	SDRAM RAS output
	nSCAS	O	SDRAM CAS output
	nSWE	O	SDRAM write enable output
	nSCS[1:0]	O	SDRAM chip select outputs
	SDQML	O	SDRAM lower data byte enable
	SDQMU	O	SDRAM upper data byte enable
	SD[15:0]	IO	SDRAM data bus
	SA[14:0]	O	SDRAM address bus
DMA Interface	DMAREQ	Im	DMA request input
	DMAACK	Om	DMA acknowledge output
UART	nUDCD0	Im	UART data carrier detect input
	nUDSR0	Im	UART data set ready input
	nUCTS0	Im	UART clear to send input
	USIN[3:0]	Im	UART serial data inputs
	USOUT[3:0]	Om	UART serial data outputs
	nUDTR0	Om	UART data terminal ready
	nURTS0	Om	UART request to send
	nURING0	Im	UART ring input signal (wake-up signal to PMU)
IrDA	IRDIN1	Im	IrDA infra-red data input
	IRDOUT1	Om	IrDA infra-red data output
USB	AUSBP	AIO	USB positive signal
	AUSBN	AIO	USB negative signal
	AVDDUSB	P	USB analog Vdd
	AVSSUSB	P	USB analog Vss

Function	Signal Name	Signal Type	Description
PWM	PWM[1:0]	Om	Pulse width modulation output
I^2S	ISD	Om	I2S data output
	ISCLK	Om	I2S clock output
	ISWS	Om	I2S word select output
	ISE_CLK	Im	I2S external clock input
CAN	CANTX[1:0]	Om	Controlled Area Network data output
	CANRX[1:0]	Im	Controlled Area Network data input
Matirix Keyboard	KSCANO[7:0]	ODm	Matrix keyboard scan outputs
	KSCANI[7:0]	Im	Matrix keyboard scan inputs
PS/2 Interface	PS2D	ODm	PS2 data signal
	PS2CK	ODm	PS2 clock signal
MMC	SSDO	Om	MMC card controller data output
	SSDI	Im	MMC card controller data input
	SSCLK	Om	MMC card controller clock output
	nSSCS	Om	MMC card controller chip select
SSFDC (SmartCard)	SMD[7:0]	IOm	Smart Media Card (SSFDC) data signals
	nSMWP	Om	Smart Media Card (SSFDC) write protect
	nSMWE	Om	Smart Media Card (SSFDC) write enable
	SMALE	Om	Smart Media Card (SSFDC) address latch enable
	SMCLE	Om	Smart Media Card (SSFDC) command latch enable
	nSMCD	Im	Smart Media Card (SSFDC) card detection signal
	nSMCE	Om	Smart Media Card (SSFDC) chip enable
	nSMRE	Om	Smart Media Card (SSFDC) read enable
	nSMRB	Im	Smart Media Card (SSFDC) READY/nBUSY signal
ADC	ATSXP	IO	Touch screen switch X high drive
	ATSXN	O	Touch screen switch X low drive
	ATSYP	IO	Touch screen switch Y high drive
	ATSYN	O	Touch screen switch Y low drive
	ADIN[4:0]	AI	ADC inputs for MIC, battery, touch
	AVDDADC	P	ADC analog Vdd
	AVSSADC	P	ADC analog Vss
	AVREFADC	AI	ADC reference voltage
DAC	AVDDDAC	P	DAC analog Vdd
	AVSSDAC	P	DAC analog Vss
	ADACR	AO	Sound DAC output (Right channel)
	ADACL	AO	Sound DAC output (Left channel)
PLL	PLLVDD[1:0]	P	PLL analog Vdd
	PLLVSS[1:0]	P	PLL analog Vss
	PLLFILT[2:0]	AI	External PLL loop filter input pins (1 per PLL)
GPIO	PORTA[15:0]	IOm	General purpose input/output signals
	PORTB[11:0]	IOm	General purpose input/output signals
	PORTC[10:0]	IOm	General purpose input/output signals
	PORTD[8:0]	IOm	General purpose input/output signals
	PORTE[24:0]	IOm	General purpose input/output signals
System	nPOR	IS	Power on reset input. Schmitt level input with pullup
	nPMWAKEUP	IS	Wake-up “on-key” input. Low causes PMU to exit standby state.
	nRESET	IO	Reset input (also driven out in POR, until the PLL is locked)
	PMADAPOK	I	Adapter power OK
	PMBATOK	I	Main battery OK
Oscillator	RTCOSCIN	I	RTC oscillator input
	RTCOSCOUT	O	RTC oscillator output
	OSCIN	I	Main oscillator input
	OSCOUT	O	Main oscillator output
Digital Power/ Ground	VDDCore[3:0]	P	Core Vdd supply (1.8V)

Function	Signal Name	Signal Type	Description
	VSSCore[3:0]	P	Core Vss supply
	VDD[11:0]	P	IO Vdd supply (3.3V)
	VSS[11:0]	P	IO Vss supply
	VDD5[1:0]	P	IO Vdd supply for 5V input tolerant (supply 3.3V if there is no 5V input signal)
JTAG	TCK	Iu	JTAG boundary scan and debug test clock
	nTRST	Id	JTAG boundary scan and debug test reset
	TMS	Iu	JTAG boundary scan and debug test mode select
	TDI	Iu	JTAG boundary scan and debug test data input
	TDO	O	JTAG boundary scan and debug test data output
Test	nPLLENABLE	Id	Low to enable PLL. High to bypass PLL with clock from OSCIN
	nTEST	Iu	Test mode select

Table 2-2 External Signal Functions

2.2.2 Multiple Function Pins

Pin No.	Default Function	Function 1	Function 2
6	KSCANO0		PORTA0
7	KSCANO1		PORTA1
8	KSCANO2		PORTA2
9	KSCANO3	ISD	PORTA3
10	KSCANO4	ISCLK	PORTA4
11	KSCANO5	USIN2	PORTA5
12	KSCANO6	USOUT2	PORTA6
13	KSCANO7	IRDOUT1	PORTA7
14	KSCANI0		PORTA8
15	KSCANI1		PORTA9
16	KSCANI2		PORTA10
17	KSCANI3	ISWS	PORTA11
18	KSCANI4	ISE_CLK	PORTA12
19	KSCANI5	USIN3	PORTA13
20	KSCANI6	USOUT3	PORTA14
21	KSCANI7	IRDIN1	PORTA15
68	nURING		PORTB0
69	nUDTR0		PORTB1
70	nUCTS0		PORTB2
71	nURTS0		PORTB3
72	nUDSR0		PORTB4
73	nUDCD0		PORTB5
86	TimerOut		PORTC0
78	CANTX0		PORTC1
79	CANRX0		PORTC2
87	PS2D		PORTC3
89	PS2CK		PORTC4
91	PWM0		PORTC5
92	PWM1		PORTC6
101	DMAREQ		PORTC7
102	DMAACK		PORTC8
104	nRCS2		PORTC9
103	nRCS3		PORTC10
251	LD8		PORTD0
250	LD9		PORTD1
249	LD10		PORTD2
248	LD11		PORTD3
247	LD12		PORTD4
246	LD13		PORTD5

Pin No.	Default Function	Function 1		Function 2
245	LD14			PORTD6
244	LD15			PORTD7
243	LBLEN			PORTD8
134	RD16			PORTE0
133	RD17			PORTE1
132	RD18			PORTE2
131	RD19			PORTE3
129	RD20			PORTE4
128	RD21			PORTE5
126	RD22			PORTE6
125	RD23	SMD7		PORTE7
124	RD24	SMD6		PORTE8
123	RD25	SMD5		PORTE9
122	RD26	SMD4		PORTE10
121	RD27	SMD3		PORTE11
120	RD28	SMD2		PORTE12
119	RD29	SMD1		PORTE13
118	RD30	SMD0		PORTE14
117	RD31	nSMWP		PORTE15
112	nRWE2	nSMWE		PORTE16
111	nRWE3	SMALE		PORTE17
96	MMCCMD(SSDI)	nSMRE	SSDI	PORTE18
98	MMCDAT(SSDO)	nSMCE	SSDO	PORTE19
99	nMMCCD(nSSCS)	nSMCD	NSSCS	PORTE20
100	MMCCLK(SSCLK)	SMCLE	SSCLK	PORTE21
94	CANRX1	nSMRB		PORTE22
93	CANTX1			PORTE23
184	RA24			PORTE24

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

Base Address (Byte)	Base Address (Hex)	Size	Description
0 Mbyte	0x0000.0000	32Mbytes	ROM chip select 0
64 Mbytes	0x0400.0000	32Mbytes	ROM chip select 1
128 Mbytes	0x0800.0000	32Mbytes	ROM chip select 2
192 Mbytes	0x0C00.0000	32Mbytes	ROM chip select 3
256 Mbytes	0x1000.0000	256Mbytes	Reserved
512 Mbytes	0x2000.0000	512Mbytes	Reserved
1024 Mbytes	0x4000.0000	32Mbytes	SDRAM chip select 0
1056 Mbytes	0x4200.0000	32Mbytes	SDRAM chip select 1
1088 Mbytes	0x4400.0000	32Mbytes	SDRAM mode register chip 0
1120 Mbytes	0x4600.0000	32Mbytes	SDRAM mode register chip 1
1152 Mbytes	0x4800.0000	896Mbytes	Reserved
2048 Mbytes	0x8000.0000	336Kbytes	Peripherals

Table 4-1 Top-level address map

The ROM has an address space of 256Mbytes that is split equally between four external ROM chip select. Actual address range for each chip select is 32Mbytes with 25 external address signals.

There is a maximum of 64Mbytes of SDRAM space. The mode registers in the SDRAM are programmed by reading from the 64Mbytes address space immediately above the SDRAM address space.

The peripheral address space is subdivided into three main areas: those on the ASB, the fast APB and the slow APB. The base address for the peripherals is given in Table 3-2: Peripherals base addresses.

Function	Base Address (Hex)	Name	Description
ASB Peripherals	0x8000.0000	SDRAMC Base	SDRAM Controller
	0x8000.1000	PMU Base	PMU/PLL
	0x8000.2000	Reserved	
	0x8000.3000	BUSC Base	Bus controller
	0x8000.4000	DMAC Base	DMAC
	0x8000.5000	Reserved	~0x8000.FFFF
Fast APB Peripherals	0x8001.0000	LCD	LCD
	0x8001.1000	Reserved	
	0x8001.2000	USB Base	USB
	0x8001.3000	Sound Base	SOUND
	0x8001.4000	I2S Base	I2S
	0x8001.5000	MMC Base	MMC/ SPI
	0x8001.6000	SMC Base	SMC
	0x8001.7000	Reserved	~0x8001.FFFF
DMA	2Gbyte + 0x51000	Reserved	
	2Gbyte + 0x52000	USB Base	USB- ARM accesses as DMA bus master
	2Gbyte + 0x53000	Sound Base	SOUND – ARM accesses as DMA bus master
	2Gbyte + 0x54000	MMC Base	MMC, SMC- ARM accesses as DMA bus master
	2Gbyte + 0x55000 ~ 4Gbyte-1	Reserved	
Slow APB Peripherals	0x8002.0000	U1 Base	UART 1
	0x8002.1000	U2 Base	UART 2
	0x8002.2000	KBD Base	KBD
	0x8002.3000	GPIO Base	GPIO
	0x8002.4000	INTC Base	INTC
	0x8002.5000	Timer Base	TIMER
	0x8002.6000	Reserved	~0x8002.7FFF

Function	Base Address (Hex)	Name	Description
	0x8002.8000	RTC Base	RTC
	0x8002.9000	ADC Base	ADC
	0x8002.A000	Reserved	
	0x8002.B000	WDT Base	WDT
	0x8002.C000	PS2 Base	PS2
	0x8002.D000	UART2 Base	UART2
	0x8002.E000	UART3 Base	UART3
	0x8002.F000	CAN0 Base	CAN0
	0x8003.0000	CAN1 Base	CAN1
	0x8003.1000	Reserved	~0x8004.FFFF

Table 4-2 Peripherals Base Addresses

5 PMU & PLL

The HMS30C7202 is designed primarily for HPC and other portable computing applications. Therefore there are 4 operating modes to reduce power consumption and extend battery life.

- RUN - normal operation (typically used when soft modem is in operation and other CPU-intensive tasks)
- SLOW - half-speed operation used when the application interacts with a user (e.g. word processing)
- IDLE - where the CPU operation is halted but peripherals operation continue (such as screen refresh, or serial communications)
- SLEEP & DEEP SLEEP - This mode will be perceived as `off' by the user, but the contents of SDRAM are maintained and only the real-time clock is running.
- There are a number of Power management states, (see 7.3 Power management states on page 7-5) as described above, and the transition between states is controlled by the PMU. The PMU is an ASB slave unit to allow the CPU to write to its control registers, and is an ASB master unit to provide the mechanism for stopping the ARM core's internal clock).

5.1 Block Functions

CLOCK generator
The CLOCK generator module is responsible for controlling the PLL's and gating clocks while the outputs of the PLLs are uncertain and to ensure that clocks are available during test modes and during RESET sequences.

FCLK (ARM Processor and SDRAM controller clock)
Derived from PLL3 whose Frequency is controllable between 49.7664 MHz and 82.944 MHz. Frequency of operation is set using a 6-bit register.
There are two methods for updating frequency, depending upon the state of bit 6 of the Clock Control register ClkCtl (see ClkCtl register on page 7-11). If bit 6 is set, then any data written to bits [5:0] of the ClkCtl register are immediately transferred to the pins of PLL3, thus causing the loop to unlock and to mute FCLK. This is only a safe mode of operation if PLL3 frequency and mark-space ratio is guaranteed to be within limits immediately after the Lock Detect signal has become active. If bit 6 is NOT set, then the HMS30C7202 must enter DEEP sleep mode before bits [5:0] of the Clock Control register are transferred to PLL3.

To switch between the two frequencies when bit 6 is not set:
- Software writes the new value into the ClkCtl register
- Set a Real Time Clock Alarm to wake the HMS30C7202 in 2 seconds
- Enter DEEP SLEEP Mode by writing to the PMUMode Register
- The HMS30C7202 will power up with PLL3 running at the new frequency

BCLK
Bus Clock is generated by the PMU by dividing FCLK by 2.

VCLK
Clock for the LCD controller. Frequency is selectable between 31.5MHz or 40MHz. The VCLK PLL is disabled when on BnRES is active or when the PMU is put into DEEP SLEEP mode. On exit from either of these conditions, the VCLK PLL must be re-enabled by software.

Changing Frequency:
1. Software must first disable the VCLK pll, by writing a `0' to the PLL1Enable bit of the ClkCtl register.
2. Write the new value to the PLL1Freq bit.
3. Re-enable the VCLK pll by writing 1 to the PLL1Enable bit.

CCLK
Clock for the IR comms and the USB. Nominally 48MHz. The CCLK PLL is disabled when BnRES active or when the PMU is put into DEEP SLEEP mode. On exit from either of these conditions, the CCLK PLL must be re-enabled by software.

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 happen when a user switches off the device) and the state machine will proceed to the commanded state.

5.2 Power management

5.2.1 State Diagram

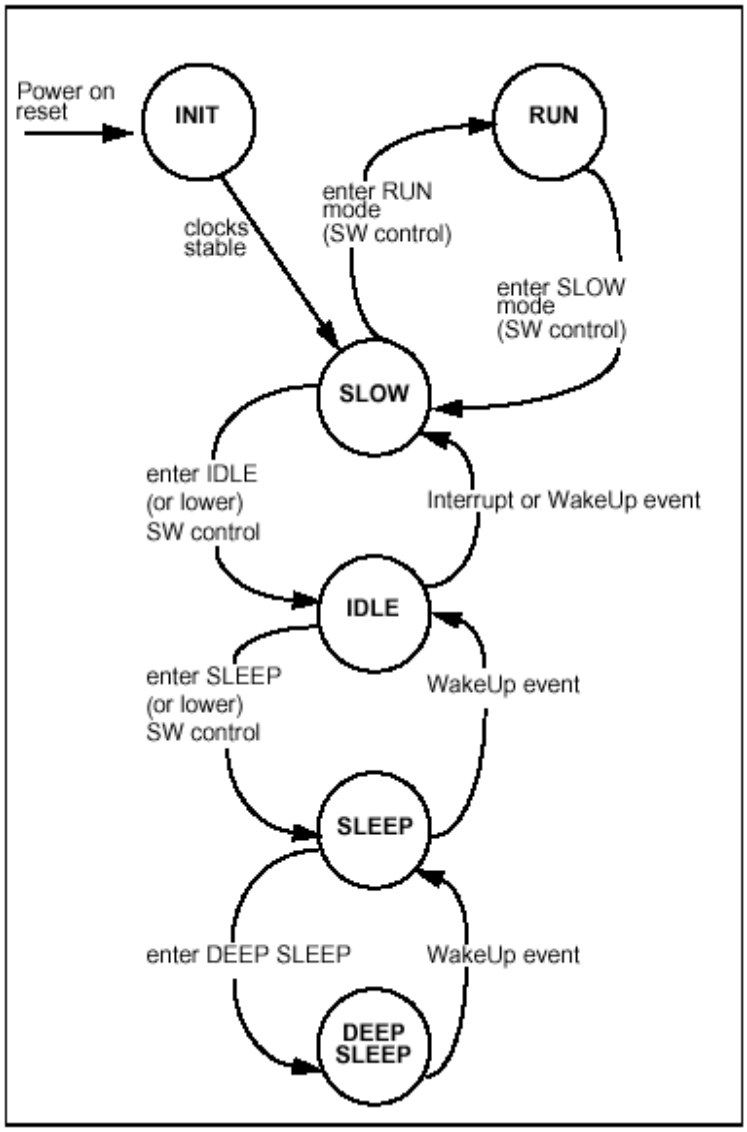


Figure 5-1 PMU Power Management State Diagram

5.2.2 Power management states

Run
The system is running normally. All Clocks 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 SLEEP Mode.

IDLE
In this mode, the PMU becomes the bus master until there is either a fast or normal interrupt for the CPU, or the peripheral DMA controller requests master-ship of the bus.
This will cause the clocks in the CPU to stop when it attempts an ASB access. Entry to this mode can be caused by the CPU writing the PMU_IDLE value to the PMU Mode Register when in RUN or SLOW modes, or a WakeUp signal becoming active when the PMU is SLEEP or DEEP SLEEP modes

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 mastership of the ASB before this mode can be entered). The PMU must be bus master to ensure that the system is stopped in a safe state, and is not half way through an SDRAM write (for example). Both the Video and Communication clocks should be disabled before entering this state.
Usually this state would only be entered briefly, on the way to entering DEEP SLEEP mode.

DEEP SLEEP
In DEEP SLEEP mode, the 3.6864MHz oscillator and the PLL are disabled. This is the lowest power state available. Only the 32KHz oscillator runs. The real time clock and the PMU are clocked from this clock. Clocked circuitry in the PMU runs from 4kHz (ie the RTC clock divided by 8). Everything else is powered down, and SDRAM is in selfrefresh mode. This is the normal system "off" mode.
SLEEP and DEEP SLEEP modes are exited either by a user wake-up event (generally pressing the "On" key), or by an RTC wake-up alarm, or by a modem ring indicate event. These interrupt sources go directly to the PMU.

5.2.3 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 250Hz clock derived from the low power (4kHz) clock. After filtering to a quarter of 250 Hz, each event has an associated `sticky' register bit. nPMWAKEUP 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 a 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 and Power Adapter condition inputs are combined to form the PMU Interrupt. Each of these three 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 bit, by writing 0 to bit 9 of the ResetStatus register.
- Once an interrupt has occurred, read the RESET / Status 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 location (in the nPMWAKEUP case, this will be 0x10.).

PORTA Wake-up Sequence
The PORTA interrupt is OR gated with nPMWAKEUP to support additional wake up sources.
Each PORTA input signal can be used as a wake up source; they are enabled using the Interrupt MASK Register. After wake up, s/w should program the PORTA Interrupt Mask Register and/or the PMU ResetStatus Register.
One possible application is to use the nDCD signal, from the a UART interface, as a wake up source, by connecting nDCD to a PORTA input. In Deep Sleep mode, nDCD can wake up the system by generating a PORTA interrupt request to the PMU block. The PMU state machine then returns the system to the operational mode.

5.3 Registers

Address	Name	Width	Default	Description
0x8000.1000	PMUMODE	4		PMU Mode Register

0x8000.1010	PMUID	32		PMU ID Register
0x8000.1018	PMUBRT	2		PMU Bus Retract Register
0x8000.1020	PMUSTAT	17		PMU Reset/PLL Status Register
0x8000.1028	PMUCLK	16	0x1B	PMU Clock Control Register
0x8000.1030	PMUDBCT	9		PMU Debounce Test Register

Table 5-1 PMU Register Summary

5.3.1 PMU Mode Register (PMUMODE)

This read/write register is written to by the CPU to change mode from RUN mode or SLOW mode into a different mode. The encoding is shown below, in Table 7-2: Mode entry encoding. Obviously the register can only be read and written to in RUN mode or SLOW mode, since these are the only modes in which the processor can access these registers. Therefore, the processor will never be able to read values for modes other than mode 0x00 and mode 0x 01. Other values may be read by a test controller so long as clocks are enabled with bit 8 of the DbCtr register. See Table 7-12: DbCtr Register Bit 8 on page 7-14.

31	…	3	2	1	0
		WAKEUP	MODE SEL		

Bits	Type	Function
31:4	-	Reserved
3	R/W	Writing a `1' to this bit allows PMU to exit DEEP SLEEP mode when pins PMBATOK and PMADAPOK are both low. Writing a `0' to this bit prevents the PMU from leaving DEEP SLEEP mode when PMBATOK and PMADAPOK are both low
2:0	R/W	**Value** **PMU Mode** 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.

5.3.2 PMU ID Register (PMUID)

This read-only register returns a unique chip revision ID. Revision 0 of the HMS30C7202 device (the first revision) will return the constant value 0x00720200.

31	…	0
	0x00720200	

5.3.3 PMU Bus Retract Register (PMUBR)

31	…	1	0

						BRDelay	BRENABLE

Bits	Type	Function
31:2	-	Reserved
1	R/W	Enables bus retracts
0	R/W	0: bus retracts after 8 cycles 1: bus retracts after 12 cycles

BRENABLE enables correct DMA operation when slow peripherals are connected to the external bus. When enabled, bus retracts occur when either nPCAWAIT, nPCBWAIT or EXPRDY are held active by a slow external peripheral for more than the number of clocks specified by BRDelay. The bus retract ensures the DMA is not stalled for the duration of the slow peripheral bus access

5.3.4 PMU Reset /PLL Status Register (PMUSTAT)

This read/write register provides status information on power on reset and the PLL status. The allocation is a shown in following two tables: ResetStatus Register Bits. The bits in this register are `sticky' bits. For a definition of a sticky bit please refer to Wake-up Debounce and Interrupt on page 7-7. Generally, this register will be read each time the ARM exits reset mode, so that the ARM can identify what event has caused it to exit from reset mode.

							16
							WARM RESET
15	14	13	12	11	10	9	8
HOTSYNC INTR	ADAPTOR INTR	RTC INTR	MRING INTR	WAKEUP INTR	HOTSYNC STATUS	WDT RST	WARMRST STATUS
7	6	5	4	3	2	1	0
ADAPTOR STATUS	RTC STATUS	MRING STATUS	WAKEUP STATUS	PLL3 LOCK	PLL2 LOCK	PLL1 LOCK	POR STATUS

Bits	Type	Function
31:17	-	Reserved
16	W	Warm RESET. Writing a `1' causes nRESET to be asserted. Writing `0' has no effect.
15	R/W	HOTSYNC interrupt Mask. When reads, 0 = Disable Hotsync interrupt from External pin. 1 = Enable Hotsync interrupt from External pin.
14	R/W	No External Power Interrupt Mask. When reads, 0 = Disable PMU interrupt from PMADAPOK LOW. 1 = Enable PMU interrupt from PMADAPOK LOW.
13	R/W	RTCEvt Interrupt Mask. When reads, 0 = Disable PMU interrupt from RTC 1 = Enable PMU interrupt from RTC
12	R/W	RIEvt Interrupt MASK PMU Interrupt Request / Clear When reads, 0 = Disable PMU interrupt from MRING 1 = Enable PMU interrupt from MRING
11	R/W	OnEvt Interrupt MASK PMU Interrupt Enable When reads, 0 = Disable PMU interrupt from nPMWAKEUP 1 = Enable PMU interrupt from nPMWAKEUP
10	R/w	HOTSYNC Event When reads, 0 = Not Hot Sync state; 1 = Hot Sync status When writes, HotSync Interrupt Clear. Writing a `1' to this bit clears the event bit
9	R/w	WDTEvt: Watch Dog Reset (Warm reset) When reads, 0 = No Watch dog Timer event occured 1 = A Watch dog timer event has ocurred since last cleared When writes, Watch dog Reset Clear. Writing a `1' to this bit clears the event bit
8	R/w	RESETEvt: Warm RESET Event (debounced)

When writes to these bits, PMU Interrupts will be enable. `1' enables interrupts to the CPU, `0' masks such activity. Should the enable bit be set to one when one of the debounced event signals is set, then an interrupt WILL be generated (i.e. the interrupt is level sensitive, not edge sensitive).

		When reads, 0 = No Warm RESET event has occurred 1 = A Warm RESET event has ocurred since last cleared When writes, Warm Reset Clear. Writing a `1' to this bit clears the event bit.
7	R/w	PowerFailEvt: ADPATOR NOT OK (debounced) When reads, 0 = No PowerFail event since last cleared 1 = A PowerFail event has occurred since last cleared When writes, Power Fail Interrupt Clear. Writing a `1' to this bit clears a pending interrupt bit.
6	R/w	RTCEvt When reads, 0 = No Real Time Clock (RTC) calendar wake-up event since last cleared 1 = Real Time Clock (RTC) calendar wake-up event since last cleared When writes, RTC Interrupt Clear. Writing a `1' to this bit clears a pending interrupt bit.
5	R/w	RIEvt (debounced) When reads, 0 = No Modem Ring Indicate wake-up event since last cleared 1 = Modem Ring Indicate wake-up event since last cleared When writes, RI Interrupt Clear. Writing a `1' to this bit clears a pending interrupt bit.
4	R/w	OnEvt (debounced) When reads, 0 = No On key event since last cleared; 1 = On key event since last cleared When writes, OnEvt Interrupt Clear. Writing a `1' to this bit clears a pending interrupt bit.
3	R/w	PLLLock3 When reads, 0 = System PLL has been locked since last cleared 1 = System PLL has fallen out of lock since last cleared When writes, writing a `1' to this bit causes the PLL3 Unlock event flag to be cleared.
2	R/w	PLLLock2 When reads, 0 = Comms PLL has been locked since last cleared 1 = Comms PLL has fallen out of lock since last cleared When writes, writing a `1' to this bit causes the PLL2 Unlock event flag to be cleared.
1	R/w	PLLLock1 When reads, 0= LCD PLL has been locked since last cleared 1= LCD PLL has fallen out of lock since last cleared When writes, writing a `1' to this bit causes the PLL1 Unlock event flag to be cleared.
0	R/w	PORStatus When reads, 0 = No POR since last cleared; 1 = POR since last cleared When writes, writing a `1' to this bit causes the nPOR event flag to be cleared.

5.3.5 PMU Clock Control Register (PMUCLK)

This register is used to control the frequency of PLL3, the system clock PLL and PLL1, the LCD clock. Six bits are defined which control the frequency of FCLK, and a further bit is used to control the frequency of PLL1, the LCD clock. The Default (Power on Reset) value for this register is 0x1b.

15	14	13	12	11	10	9	8
PLL2 ENABLE	PLL1 ENABLE	PLL1 FREQ					
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PLL3 MUTE	PLL3 FREQ UPDATE	PLL3 FREQ					

Bits	Type	Function
31:16	-	Reserved
15	R/W	Set for PLL2 enable. Output will be gated until PLL2 Lock Detect (LD) is received. Reset for disable PLL2
14	R/W	Set for PLL1 enable. Output will be gated until PLL1 Lock Detect (LD) is received. Reset for disable PLL1

13:8	R/W	Same with bit [5:0].
7	R/W	Reset: PLL3 is muted when Lock detect = 0 (default) Set: PLL3 only muted after nPOR or nRESET. Subsequent unlock condition does not mute the clock. Allows dynamic changes to the clock frequency without halting execution. Care: this only will be legal if PLL3 is under-damped (i.e. will not exhibit overshoot in its lock behavior).
6	R/W	Reset: PLL3 frequency control frequency is only updated when PMU exits DEEP SLEEP mode (default) Set: PLL3 frequency control frequency is updated instantaneously
5:0	R/W	**Value** **Frequency** **Value** **Frequency** 0x1B 49.7664 MHz 0x25 68.1984 MHz 0x1C 51.6096 MHz 0x26 70.0416 MHz 0x1D 53.4528 MHz 0x27 71.8848 MHz 0x1E 55.2960 MHz 0x28 73.7280 MHz 0x1F 57.1392 MHz 0x29 75.5712 MHz 0x20 58.9824 MHz 0x2A 77.4144 MHz 0x21 60.8256 MHz 0x2B 79.2576 MHz 0x22 62.6688 MHz 0x2c 81.1008 MHz 0x23 64.5120 MHz 0x2D 82.9440 MHz 0x24 66.3552 MHz Other values

	Reserved

IF BIT 6 is `0'
When the CPU writes to bits 5:0 of this register, these bits are stored in a temporary buffer, which is not transferred to the PLL until the next time the PLL lock signal becomes inactive. This means that for a new value to take effect, it is necessary for the device to enter DEEP SLEEP mode first.

IF BIT 6 is `1'
The first effect that writing a new value to bits [5:0] will have is that PLL3 will go out of lock, and the Clock control circuit will immediately inhibit FCLK and BCLK, without first verifying that SDRAM operations have completed.

5.3.6 PMU Debounce Counter Test Register (PMUDBCT)

Bits	Type	Function	
		Read	**Write**
31:9	-	Reserved	
8	W	Reset: Normal operation Set: Forces FCLK and BLCK to be active in all PMU states (test purposes only)	
7:6	-	Reserved	
5	R	Selected debounce counter bits	Reserved
4	R/W		Reset: normal operation Set: disables Bus Request from the PMU to allow CPU to read state machine for test purposes during PMU IDLE state.
3	R/W	Prescaler bits	Reset: nTEST takes value from input pin Set: forces local test mode
2:0	R/W		Select Debounce counter for **Value** **Function** 0x0 nPMWAKEUP 0x1 RING event 0x3 Power Adaptor event 0x4 Warm Reset

In order that the debounce counters (which would normally be clocked from 4kHz) may be independently exercised and observed, the counters may be triggered and observed using the above registers. These registers are for testing only and are not required in normal use.

5.4 Timings

5.4.1 Reset Sequences of Power On Reset

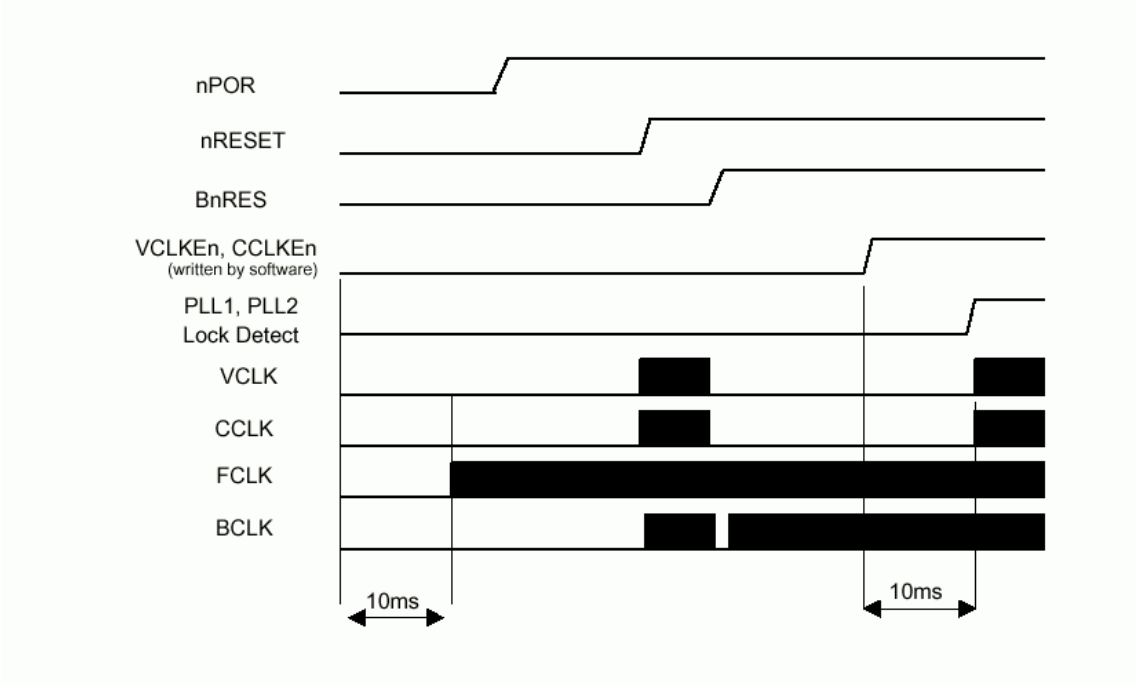


Figure 5-2 PMU Cold Reset Event

In the event of removal and re-application of all power to the HMS30C7202, 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 the nPOR condition. Clocks are enabled temporarily to allow synchronous resets to operate.
- The default frequency of FCLK on exit from nPOR will be 49.7664MHz.
- When FCLK is stable, the CPU clock is released. If the CPU were to read the RESET/Status register at this time, it will return 0x10f:
- The CPU may write 0x10f to the RESET register to clear these flag bits.
- The CPU writes 0x20 to the clock control register, which will set a FCLK speed of 58.9824MHz. 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 7-2: Power Management State Diagram on page 7-6).
- When the RTC timer alarm is activated, the PMU automatically wakes up into SLOW mode, but with the new FCLK frequency of 58.9824Mhz.
- The CPU may write 0x620 to the Clock Control register, which enables CCLK and VCLK, and retains the new FCLK frequency.

Bit	Meaning
Bit 0 set:	Power On Reset event has occurred
Bit 1 set:	PLL1 has been `unlocked'
Bit 2 set:	PLL2 has been `unlocked'
Bit 3 set:	PLL3 has been `unlocked'

Table 5-2 PMU Bit Settings for a cold Reset Event within PMUSTAT Register

5.4.2 Software Generated Warm Reset

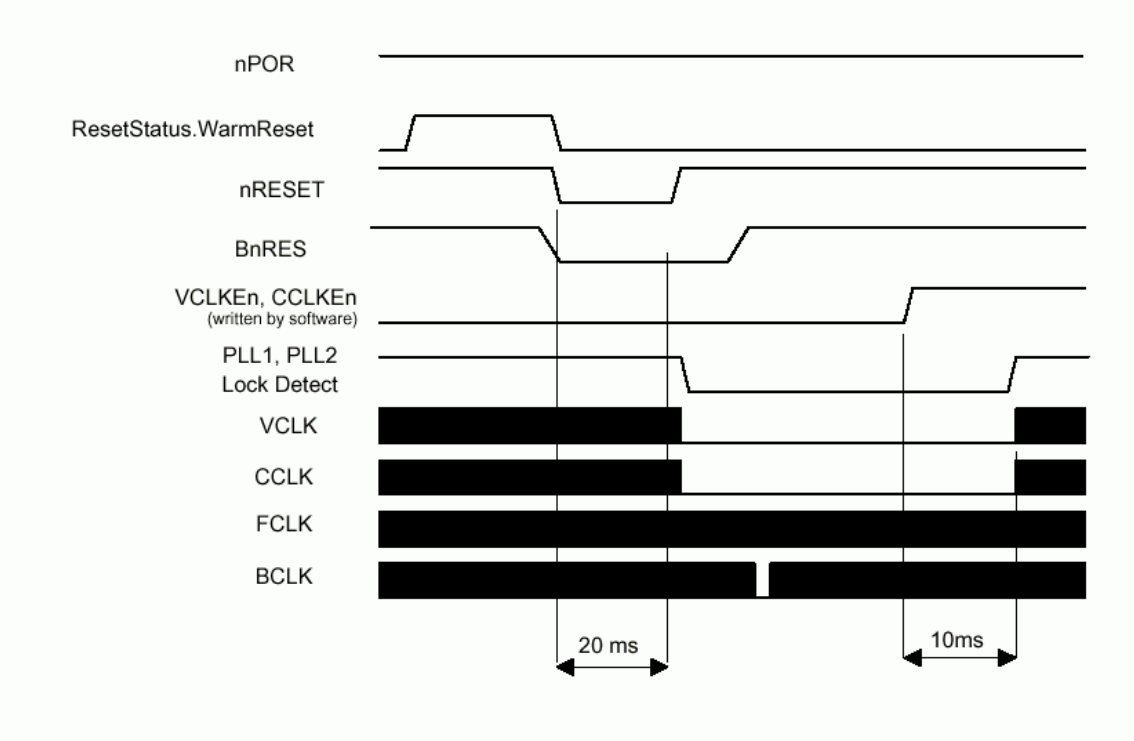


Figure 5-3 PMU Software Generated Warm Reset

The CPU writes `1' to the WarmReset bit of PMUSTAT register. The PMU drives nRESET low. The internal chip reset, BnRES is driven low. The PMU detects that the bi-directional nRESET pin is low. nRESET is filtered by a de-bounce circuit. Note that this means that nRESET will remain low for a minimum of 16ms. BnRES becomes active once the de-bounced nRESET goes high once more, which disables PLL1 and PLL2. The CPU may read the PMUSTAT register, which will return 0x106:

Bit	Meaning
Bit 1 set:	PLL1 has been `unlocked'
Bit 2 set:	PLL2 has been `unlocked'
Bit 8 set:	A RESET event has occurred.

Table 5-3 PMU Bit Settings for a Software Generated Warm Reset within PMUSTAT Register

5.4.3 An Externally generated Warm Reset

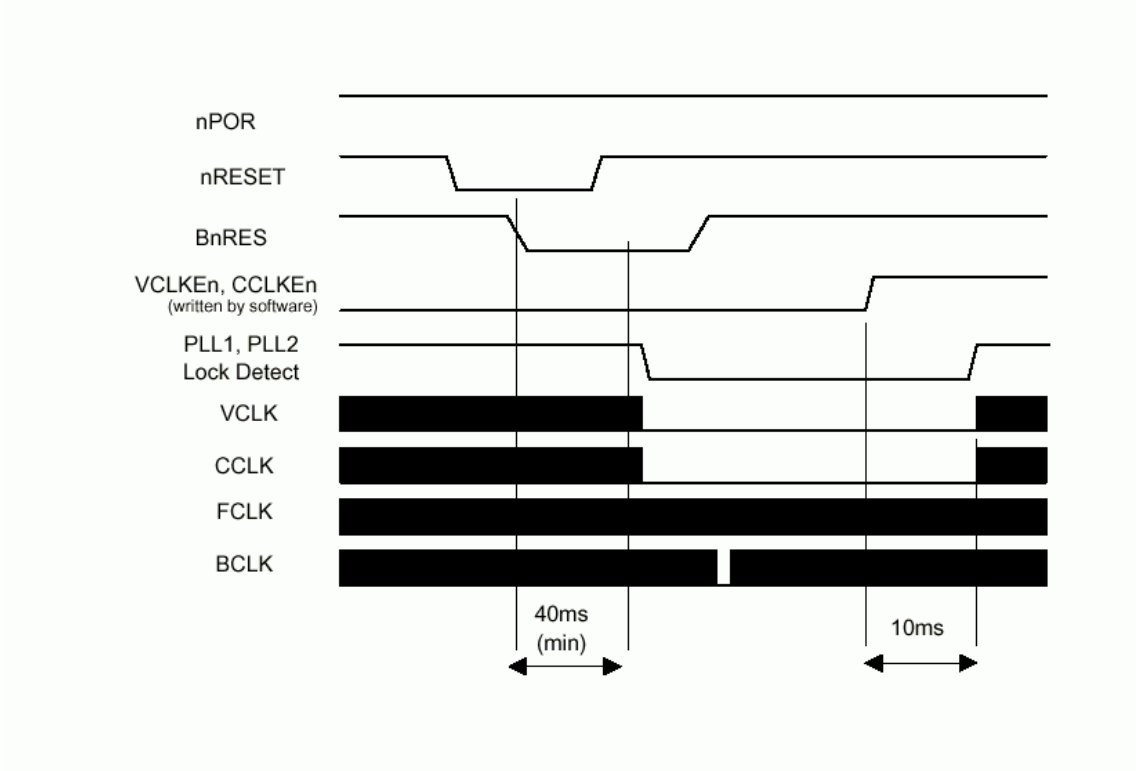


Figure 5-4 PMU An Externally Generated Warm Reset

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 40ms. BnRES (the on-chip reset signal) becomes active as soon as nRESET is low, and high once the de-bounced nRESET goes high once more. BnRES disables PLL1 and PLL2. The CPU may read the RESET register, which will return 0x106:

Bit	Meaning
Bit 1 set:	PLL1 has been `unlocked'
Bit 2 set:	PLL2 has been `unlocked'
Bit 8 set:	A RESET event has occurred.

Table 5-4 PMU Bit Settings for a Warm Reset within PMUSTAT Register

Note
The internal chip reset, BnRES, remains active for 20ms after an externally generated nRESET. External devices should not assume that the HMS30C7202 is in an active state during this period.

6 SDRAM CONTROLLER

The system RAM resource is provided by SDRAM, on an interface that is run at the HMS30C7201's core clock frequency. Between 2 and 64Mbytes of external SDRAM are supported by one to four external devices. To reduce power consumption, each SDRAM device has its own Clock Enable (CKE), so each device may individually be placed in low power mode when idle. The SDRAMs are powered down into self-refresh mode when the whole system is placed in standby mode.

Internal to the GMS30C7201, the SDRAM controller arbitrates between access requests from the Main AMBA bus, and a custom Video bus.

The best use of an SDRAM is made when data is streamed in sequence, and future access requests can be predicted. It is in the nature of video data to be accessed in sequence at regular intervals; however, SDRAM accesses from the ARM are a lot less predictable. The SDRAM controller makes use of access predictability to maximize the use of memory interface bandwidth by having simultaneous access to both the LCD and VGA address buses. Video accesses to the SDRAM occur in fixed-burst lengths of 16 words, ARM and DMA controller 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

- 80MHz clock speed
- 16 Bits wide external bus interface (Two access requires for each word)
- 16/64/128/256Mbit device supports
- 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).
- Support External Device interface with DMA channel 2.

6.1 Supported Memory Devices

From 2-64Mbytes of SDRAM are supported with any mixture of up to two of 16/64/128/256Mbit devices. Each of the two external devices is mapped to a 32Mbyte boundary, and relies on the memory management unit to map different mixtures of devices (for example, 16- and 64Mbit devices) into a continuous address space for the ARM. Note that 16Mbit devices appear eight times, and the 64Mbit devices appear twice in the memory map.

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 HMS30C7202 can use any mixture of 16/64/128/256Mbit 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 banks of memory to be open at once. The open banks may exist in different physical SDRAM devices.

6.2 Registers

The SDRAM controller has four registers: the configuration, refresh timer, the Write Buffer Flush timer and wait driver. 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. The wait driver is used to set wait delay for external slow device.

Address	Name	Width	Default	Description
0x8000.0000	SDCON	32	0x00700000	Configuration register
0x8000.0004	SDREF	16	0x0080	Refresh timer
0x8000.0008	SDWBF	3	0x1	Write back buffer flush timer
0x8000.000C	SDWAIT	4	0x1	Wait driver register

Table 6-1 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 in 6.3 Power-up Initialization of the SDRAMs

6.2.1 SDRAM Controller Configuration Register (SDCON)

31	30	...	24	23	22	21	20	19	18	17	...	7	6	...	3	2	...
S1	S0	-	W	R	A	C1	C0	D	C	B	-	E1	B1	-	E0	B0	-

Bits	Type	Function
31:30	R	SDRAM controller Status 11:Reserved 10:Self refresh 01:Busy 00:Idle
24	R/W	*Wait driver enable bit for test purpose*
23	R/W	Normal SDRAM controller refresh enable Value = 1 if the SDRAM controller provides refresh control Value = 0 if the SDRAM controller does not provide refresh
22	R/W	Auto pre-charge on ASB accesses A = 1 auto pre-charge (default) A = 0 no auto pre-charge
21:20	R/W	11:CAS latency3 10:CAS latency2 01:CAS latency1 00:Reserved
19	R/W	SDRAM bus tri-state control D = 0 the controller drives the last data onto the SDRAM data bus (default) D = 1 the SDRAM bus is tri-stated except during writes This bit should be cleared before the IC is programmed into 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 C = 0 the clock enable of all IDLE devices are de-asserted to save power (default) C = 1 all clock enables are driven HIGH continuously During power-up initialization, it is important that the E[3:0] and the R bits are set in the correct sequence.
17	R/W	Write buffer enable Value = 1 if the write buffer is enabled Value = 0 if the write buffer is disabled
7,3	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 Slot 0 - address range 0-32MByte Slot 1 - address range 32-64MByte Value = 1 if a device is present Value = 0 if a device is not present
6,2	R/W	Indicates whether the SDRAM in the slot is a 2- or 4-bank device Value = 1 if a four-bank device Value = 0 if a two-bank device

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 all 2 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 the R bit to 1 and all the E bits to 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 the R bit to 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.

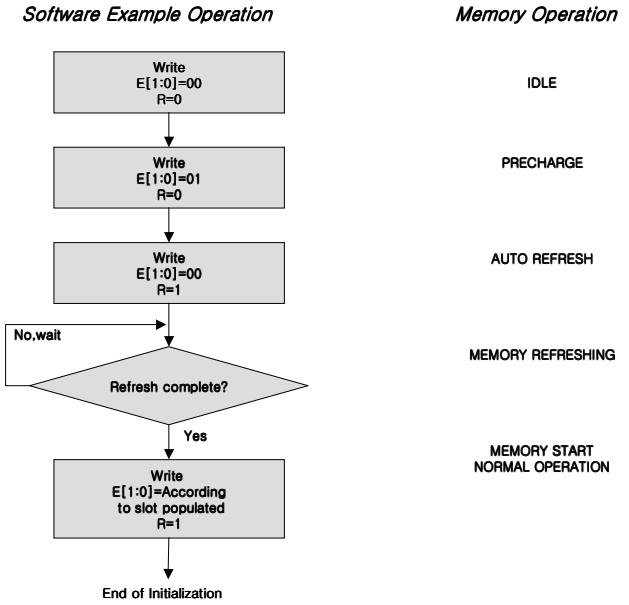


Figure 6-1 SDRAM Controller Software Example and Memory Operation Diagram

6.2.2 SDRAM Controller Refresh Timer Register (SDREF)

-	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, and a BCLK frequency of 50MHz, the following value should be programmed into it: *16x10-6 * 50x106 = 800* 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 written to as early as possible in the system start-up procedure, and in the first few cycles if the system clock is less than 8MHz.

6.2.3 SDRAM Controller Write buffer flush timer Register (SDWBF)

-	2 - 0
Reserved	SDWBF

Bits	Type	Function
2:0	R/W	A 3-bit read/write register that is selects 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

6.2.4 SDRAM Controller Wait Driver Register (SDWAIT)

-	3 – 0
Reserved	SDWAIT

Bits	Type	Function
3;0	R/W	SDWAIT programmable driven BWAIT of system bus signal (AMBA ASB). Wait driver register is operated only external device by DMA operation. It is automatically enable at DMA channel-2 operation but other channel is not. It's default and minimum value is 1. Wait function is enable only External device interface with granted DMA, disable during Internal DMA access. At this time, Write-Back buffer is always enable even if SDCON's W bit is set zero.

6.3 Power-up Initialization of the SDRAMs

The SDRAMs are initialized by applying power, waiting a prescribed amount of settling time (typically 100us), performing some auto-refresh cycles (minimum 2) 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 6-2 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.

6.4 SDRAM Memory Map

The SDRAM controller can interface with up to four SDRAMs. Three 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 is 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 6-2 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 MBA[24:1]. Note that for 16Mbit devices, pins A[11,9] on the SDRAM should be connected to pins A[13,12] on the HMS30C7202, and the pins A[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*	Note1	20*	Note 1	19*	18*	17*	16*	15*	14*	13*	12*	11*
Col 16Mbit	24	10	10	Note1	20	Note 1	23	8*	7*	6*	5*	4*	3*	2*	Note2
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*	Note2
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*	Note2
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*	Note2
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 6-2 SDRAM Row/Column Address Map

Notes
(1) For the 16Mbit device, SDRAM address line A11 should be connected to the HMS30C7202 pin SA[13](BS0), and the SDRAM address line A9 should be connected to the HMS30C7202 pin SA[12](BS1). The HMS30C7202 address lines SA[11] and SA[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 6-3.

A25	Device selected
0	Device 0
1	Device 1

Table 6-3 SDRAM Device Selection

6.5 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 Video 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/Video arbiter according to the following sequence arbitrates access requests from either the Main or

Video buses:

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

Video SDRAM accesses always occur in bursts of 16 words. Once a burst has started, the SDRAM controller without wait states presents data. Video 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 (bits 3, 2, 1 and 0 properly set 1 for each type) 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

6.6 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.
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. 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.

7 STATIC MEMORY INTERFACE

The Static Memory Controller interfaces the AMBA Advanced System Bus (ASB) to the External Bus Interface (EBI). It provides six separate memory or expansion banks. Each bank is 64MB in size and can be programmed individually to support:

- 8-, 16- or 32-bit wide, little-endian memory
- Burst mode read access support
- Variable wait states (up to 16)

In addition, bus transfers can be extended using the EXPRDY input signal. Burst mode access allows fast sequential access within quad word boundaries. This can significantly improve bus bandwidth in reading from memory (that must support at least four word burst reads).

7.1 External Signals

Pin Name	Type	Description
EXPRDY	I	Expansion channel ready. When LOW, during phase one this signal will force the current memory transfer to be extended.
nRWE [3:0]	O	These signals are active LOW write enables for each of the memory byte lanes on the external bus.
nROE	O	This is the active LOW output enable for devices on the external bus.
nRCS [3:0]	O	Active LOW chip selects.
RA [24:0]	O	ROM Address Bus
RD [31:0]	I/O	ROM Data Bus
BOOTSBIT [1:0]	I	Configuration input. 00 - Select bank 0 as 32-bit memory 01 - Select bank 0 as 16-bit memory 10 - Select bank 0 as 8-bit memory 11 - Reserved

7.2 Functional Description

The Static Memory Controller has six main functions:

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

7.2.1 Memory bank select

Start Address	Address (Hex)	Size	Description
0 Mbytes	0x0000.0000	32Mbytes	ROM chip select 0
64 Mbytes	0x0400.0000	32Mbytes	ROM chip select 1
128 Mbytes	0x0800.0000	32Mbytes	ROM chip select 2
192 Mbytes	0x0C00.0000	32Mbytes	ROM chip select 3

7.2.2 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.

7.2.3 Wait states generation

The Static Memory Controller supports wait states for read and write accesses. This is configurable between one and 16 wait states for standard memory access, and zero and 15 wait states for burst mode.

The Static Memory Controller also allows transfers to be extended indefinitely. This is done by asserting EXPRDY LOW. To hold the current transfer, EXPRDY must be asserted on the falling edge of BCLK before the last cycle of the accesses. The transfer cannot complete until EXPRDY is HIGH for at least one cycle.

7.2.4 Burst read control

This supports sequential access burst reads of up to four consecutive locations in 8-, 16- or 32-bit memories.

7.2.5 Byte lane write control

This controls nRWE [3:0] according to transfer width, RA [1:0] and the access sequencing. Table below shows nRWE coding assuming 32-bit external memory.

Type	RA [1:0]	nRWE [3:0]
Word access mode	XX	0000
Half word access mode	1X	0011
	0X	1100
Byte access mode	11	0111
	10	1011
	01	1101
	00	1110

7.3 Registers

Address	Name	Width	Default	Description
0x8000.4000	MEMCFG0			Memory Configuration Register 0
0x8000.4004	MEMCFG1			Memory Configuration Register 1
0x8000.4008	MEMCFG2			Memory Configuration Register 2
0x8000.400C	MEMCFG3			Memory Configuration Register 3

Table 7-1 Static Memory Controller Register Summary

7.3.1 MEM Configuration Register

				11	10	9	8	7	6	5	4	3	2	1	0
				BUR EN	BURST READ WAIT STATE				MORMAL ACCESS WAIT STATE				CLK EN	MEM WIDTH	

Bits	Type	Function
31:12	-	Reserved
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	Value Number of Burst Read Wait State 1111 0 1110 1 … … 0001 14 0000 15

6:3	R/W	Value Number of Normal Access Wait State 1111 1 1110 2 … … 0001 15 0000 16
2	R/W	Expansion clock enable. Setting this bit enables the EXPCLK to be active during accesses to the specified bank. This provides a timing reference for devices that need to extend bus cycles using the EXPRDY input. Back-to-back sequential accesses result in a continuous clock.
1:0	R/W	Value Memory Width 11 Reserved 10 8 bit memory access 01 16 bit memory access 00 32 bit memory access

8 LCD CONTROLLER

FEATURES

- Single panel color and monochrome STN displays
- TFT color displays
- Resolution programmable up to 640x480
- Single panel mono STN displays with either 4- or 8-bit interfaces
- 15 gray-level mono support, 3375 color STN support
- 4bpp mono, 4 or 8bpp palettized color displays
- 12bpp color `true-color' non-palettized color displays
- Programmable timing for different display panels
- 3 x 256 entry, 4-bit palette RAM
- Patented grayscale algorithm
- Little-endian operation

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

8.1 Video operation

A block diagram of the video system is shown in Figure 8-1: Video System Block Diagram. The video system has a data path for STN LCD and for TFT LCDs.

8.1.1 LCD datapath

The LCD data path is similar to the VGA data path, but it has a few additions. In TFT mode, it is similar to VGA, except that the digital RGB data is output directly to the pins of the chip, without going via a video DAC. However, in STN mode, the data must be gray scaled, and then formatted for the LCD panel. The grayscaler block converts the 4 bit per color gun data into a single bit per gun, using a patented time/space dither algorithm. In mono mode, only the B gun data is used. The output of the grayscaler is fed to the 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 2/3pixels per clock for color data.) The output of the formatter in color mode is bursty, due to the 2 2/3pixels 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.

8.1.2 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 8-1: 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 8-1 LCD Colorgrayscale intensities and modulation rates

8.1.3 TFT mode

When TFT display mode is enabled, the timing of the pixel, line and frame clocks as well as the AC-bias pin change. The pixel clock transitions continuously in this mode as long as the LCD is enabled. The AC-bias pin functions as an output enable. When it is HIGH, the display latches data from the LCD's pins using the pixel clock. The line clock pin is used as the horizontal synchronization signal (HSYNC), and the frame clock is used as the vertical synchronization signal (VSync). Pixel data is output one pixel per clock, rather than 4, 8 or 22/3pixels per clock, as it is in the passive LCD modes.

8.2 External Signals

8.3 Registers

Address	Name	Width	Default	Description
0x8001.0000	LcdControl			LCD Control Register
0x8001.0004	LcdStatus			LCD Status Register
0x8001.0008	LcdStatusM			LCD Status Mask Register
0x8001.000C	LcdInterrupt			LCD Interrupt Register
0x8001.0010	LcdDBAR			LCD DMA Channel Base Address Register
0x8001.0014	LcdDCAR			LCD DMA Channel Current Address Register
0x8001.0020	LcdTiming0			LCD Timing 0 Register
0x8001.0024	LcdTiming1			LCD Timing 1 Register
0x8001.0028	LcdTiming2			LCD Timing 2 Register
0x8001.0040	LcdTest			LCD Test register
0x8001.0044	GSFState			Grayscaler production test register
0x8001.0048	GSRState			Grayscaler production test register
0x8001.004C	GSCState			Grayscaler production test register
0x8001.0400~ 0x8001.07FC	LCDPalette			LCD Palette programming registers

Table 8-2 LCD Controller Register Summary

8.3.1 LCD Power Control

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

							24
							LDbusEn
23	**22**	**21**		**19**	**18**		
LcdBLE	LcdPwr	LcdMono8		LcdVComp			

			12	11			
			BGR	ShareDMA			
			4	3	2	1	0
			LcdTFT	LcdBW	LcdBpp		LcdEn

Bits	Type	Function
31:25	-	Reserved
24	R/W	LD data bus Enable 0 – LD data bus disable (initial value) 1 – LD data bus Enable
23	R/W	Lcd Backlight enable This drives "0" or "1" out to the Lcd backlight enable pin
22	R/W	Lcd power enable 0 - Lcd is off 1 - Lcd is on when LcdEn=1
21	R/W	Lcd monochrome data width 0 - 4 bits Lcd module 1 - 8 bits Lcd module
20	-	Reserved
19:18	R/W	Generate interrupt at: 00 - start of VSync 01 - start of BACK PORCH 10 - start of ACTIVE VIDEO 11 - start of FRONT PORCH
17:13	-	Reserved
12	R/W	0 - RGB normal video output for LCD 1 - BGR red and blue swapped for LCD
11	R/W	Share DMA Data If this bit is set, the DMA data streams for LCD and VGA are shared. The request is only generated when there is space for data in both FIFOs (both FIFO requests should be programmed to 8 words). The LCD timing generator should be slaved off the VGA timing generator, with the clock source set as the VGA clock
10:5	-	Reserved
4	R/W	LCD TFT 0 - Passive or STN display operation enabled 1 - Active or TFT display operation enabled
3	R/W	LCD Monochrome 0 - Color operation enabled 1 - Monochrome operation only enabled
2:1	R/W	LCD Bits Per Pixel 00 - 4bpp 01 - 8bpp 10 - 16bpp 11 – Reserved
0	R/W	LCD Controller Enable 0 - LCD controller disabled 1 - LCD controller enabled

8.3.2 LCD Controller Status/Mask and Interrupt Registers

The LCD controller status, mask and interrupt registers 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.

				3	2	1	0

				LDone	VComp	LNext	LFUF

Bits	Type	Function
31:4	-	Reserved
3	R	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	Vertical compare interrupt This bit is set when the Lcd timing generator reaches the vertical region 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	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	FIFO underflow status/mask/interrupt bit The LCD FIFO underflow status (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.

8.3.3 LCD DMA Base Address Register

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. Only 26 bits of the register are valid (including the LS 6 bits which must be zero), because LCD DMA is only allowed from SDRAM.
The 26 bits address range allows the LCD DMA to access any address within the SDRAM. The upper address lines are not needed, because these are the address lines used to select which device is accessed, but the LCD always accesses SDRAM. The 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 (Next) status bit 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.

Bits	Type	Function
31:26	-	Reserved. Keep these bits zero
25:6	R/W	LcdDBAR: LCD DMA Channel Base Address Pointer 16-word aligned base address in SDRAM of the frame buffer within off-chip memory.
5:0	-	Reserved. Keep these bits zero

8.3.4 LCD DMA Channel Current Address Register

This read-only register allows the processor to read the current value of the LCD DMA channel current address register. This is not something that would normally be done, but it allows additional test observability. Its value cannot be expected to be exact, it could change at an moment. However, its contents can be read to determine the approximate line that the LCD controller is currently displaying and driving out to the display

Bits	Type	Function
31:26	-	Reserved. Keep these bits zero
25:6	R/W	LcdDCAR: LCD DMA Channel Current Address Pointer 16-word aligned current address pointer to data in SDRAM frame buffer currently being

		displayed
5:0	-	Reserved. Keep these bits zero

8.3.5 LCD Timing 0 Register

LCD Timing 0 Register (LcdTiming0) controls horizontal LCD timing. See 8.6.2 Pixel Clock Divider (PCD) on page 8-13 for a description of the terms "PixelClock" and "LcdClk"

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	7	6	5	4	3	2		
HSW								PPL							

Bits	Type	Function
31:24	R/W	Horizontal Back Porch The 8-bit Horizontal Back Porch (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 LcdClk clock periods to add to the beginning of a line transmission before the first set of pixels is output to the display minus 1).
23:16	R/W	HFP Horizontal Front Porch The 8-bit Horizontal Front Porch (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 one).
15:8	R/W	Horizontal Sync Pulse Width The 6-bit horizontal sync pulse width (HSW) field is used to specify the pulse width of the line clock in passive mode, or horizontal synchronization pulse in active mode. Number of LcdClk clock periods to pulse the line clock at the end of each line minus 1
7:2	R/W	The pixels-per-line (PPL) bit-field is used to specify the number of pixels in each line or row on the screen. PPL is a 6-bit value that represents between 16-1024 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.
1:0	-	Reserved

8.3.6 LCD Timing 1 Register

LCD Timing 1 Register (LcdTiming1) controls LCD vertical timing parameters.

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

Bits	Type	Function
31:24	R/W	Vertical Back Porch The 8-bit Vertical Back Porch (VBP) field is used to specify the number of line clocks to insert at the beginning of each frame, i.e. number of inactive lines at the start of a frame, after VSync period. The VBP count starts just after the VSync signal for the previous frame has been negated for active mode, or the extra line clocks have been inserted as specified by the VSW bit-field in passive mode. After this has occurred, the value in VBP is used to count the number of line clock periods to insert before starting to output pixels in the next frame. VBP generates from 0-255 extra line clock cycles. This should be programmed to zero in passive mode, unless sensing LCD to VGA to share DMA data
23:16	R/W	Vertical Front Porch The 8-bit Vertical Front Porch (VFP) field is used to specify the number of line clocks to insert at the end of each frame, i.e. number of inactive lines at the end of frame, before VSync

		period. Once a complete frame of pixels is transmitted to the LCD display, the value in VFP is used to count the number of line clock periods to wait. After the count has elapsed the VSync (LcdFP) signal is pulsed in active mode, or extra line clocks are inserted as specified by the VSW bit-field in passive mode. VFP generates from 0-255 line clock cycles. This should be zero for passive display modes, unless synchronizing to the VGA to share data.
15:10	R/W	Vertical Sync Pulse Width The 6-bit vertical sync pulse width (VSW) field is used to specify the pulse width of the vertical synchronization pulse in active mode, or is used to add extra dummy line clock delays between frames in passive mode. Should be small for passive 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 one.
9:0	R/W	Lines Per Screen The Lines Per Screen (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.

8.3.7 LCD Timing 2 Register

LCD Timing 2 Register (LcdTiming2) controls various functions associated with the timing of the LCD controller.

				27	26	25	24	23	22	21	20	19	18	17	16
				Skip4	BCD	CPL									
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SLV	IEO	IPC	IHS	IVS	ACB					PCS	PCD				

Bits	Type	Function
31:28	-	Reserved
27	R/W	Set this bit to "1" when running a color passive LCD with slave mode. This produces an irregular clock to the LCD, where every fourth clock pulse is suppressed (the clock stays LOW for one clock period). This is necessary because two-and-two-third pixels per clock, which are sent to the LCD, is not an integer multiple. This means that three clocks will be output every four-clock period. If PCD is zero, then eight pixels will be output every eight LcdClk periods, since the LCD CP clock will be half the frequency of LcdClk.
26	R/W	Bypass Pixel Clock Divider. Setting this bit allows an undivided LCD clock to be output on LCD. This bit should only be set for TFT mode.
25:16	R/W	Clocks Per Line This is the actual number of clocks output to the LCD panel each line, minus one. 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 1, 4, 8 or two-and-two-thirds for TFT mode, mono 4-bit mode, mono 8-bit, or color STN mode (22/3) respectively.
15	R/W	Slave mode Slave (or genlock) LCD to VGA video. The HSync and VSync are locked to the VGA timing generator. The LCD horizontal timing must be carefully programmed if sharing DMA data
14	R/W	Invert Output Enable The Invert Output Enable (IEO) bit is used to select the active and inactive state of the output enable signal in active display mode. In this mode, the AC-bias pin is used as an enable that signals the off-chip device when data is actively being driven out using the pixel clock. When IEO=0, the LcdAC pin is active HIGH. When IEO=1, the LcdAC pin is active LOW. In active display mode, data is driven onto the LCD's data lines on the programmed edge of LcdCP when LcdAC is in its active state. 0 - LcdAC pin is active HIGH in TFT mode 1 - LcdAC pin is active LOW in TFT mode
13	R/W	Invert Pixel Clock The Invert Pixel Clock (IPC) 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 LcdCP. When IPC=1, data is driven onto the LCD's data lines on the falling-edge of LcdCP.

		0 - Data is driven on the LCD's data lines on the rising-edge of LcdCP. 1 - Data is driven on the LCD's data lines on the falling-edge of LcdCP.
12	R/W	Invert Hsync The Invert HSync (IHS) bit is used to invert the polarity of the LcdLP signal. 0 - LcdLP pin is active HIGH and inactive LOW. 1 - LcdLP pin is active LOW and inactive HIGH.
11	R/W	Invert Vsync The Invert VSync (IVS) bit is used to invert the polarity of the LcdFP signal. 0 - LcdFP pin is active HIGH and inactive LOW. 1 - LcdFP pin is active LOW and inactive HIGH.
10:6	R/W	AC Bias Pin Frequency The 5-bit AC-bias frequency (ACB) field is used to specify the number of line clock periods to count between each toggle of the AC-bias pin (LcdAC). 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. Note The ACB bit field had no effect on LcdAC in active mode. The pixel clock transitions continuously in active mode and the AC Bias line is used as an output enable signal
5	R/W	Pixel Clock Source This bit controls the source of the pixel clock. It can either be the LCD bus clock, or it can be the LCD clock. Selecting the video bus clock means that the LCD clock can be used for the 48MHz clock for USB when using only the LCD. 0 - LCD DMA bus clock 1 - LCD clock
4:0	R/W	Pixel Clock Divisor Used to specify the frequency of the pixel clock based on the LCD clock (LcdCLK) frequency. Pixel clock frequency can range from LcdCLK/2 to LcdCLK/33, where LcdClk is the clock selected by LCS. Pixel Clock Frequency = LcdCLK/(PCD+2). Note that in the case of the LCD, the pixel clock is not the frequency of some nominal clock rate that individual pixels are output to the LCD. It is the frequency of the LcdCP signal. In normal mono mode (4-bit interface), four pixels are output per LcdCP cycle, so the PixelClock is one quarter the nominal pixel rate. In the case of 8-bit interface mono, PixelClock is one-eighth the nominal pixel rate, since 8 pixels are output per LcdCP cycle. In the case of color, PixelClock is 0.375 times the nominal pixel rate, because 22/3 pixels are output per LcdCP cycle. If the LCD and VGA are operating concurrently, and sharing DMA data, then in color mode the pixel clock should normally be 3/8 the VGA clock. To achieve this, PCD should be programmed to the value 0 and the skip4 bit set to "1". The skip4 bit produces a null clock cycle (no high phase) every fourth clock cycle.

8.3.8 LCD Test Register

The LCD test register contains bits that allow certain LCD signals to be output on the LCD pins for test purposes. This register should not normally be used. The register is reset to all zero, and this will result in normal operation.

							8
							TCOUNT
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TCC	TLC	TCR	TLR	TCF	TRF	TLDATA	TEST MODE

Bits	Type	Function
31:9	-	Reserved
8	R/W	Separates the 10-bit counter into nibbles for the test purpose
7	R/W	For production test of grayscaler, never write a "1" to these registers in normal use.
6	R/W	For production test of grayscaler, never write a "1" to these registers in normal use.
5	R/W	For production test of grayscaler, never write a "1" to these registers in normal use.
4	R/W	For production test of grayscaler, never write a "1" to these registers in normal use.
3	R/W	For production test of grayscaler, never write a "1" to these registers in normal use.

2	R/W	For production test of grayscaler, never write a "1" to these registers in normal use.
1	R/W	Walking one's pattern used in place of SDRAM data for the LCD controller
0	R/W	Test mode bit for grey-scaler

8.3.9 Grayscaler Test Registers

The registers GSFrame State, GSRow State and GS Column State are used for the purpose of production test and **must not** be written to or read from in normal use.

8.3.10LCD Palette registers

The LCD palette registers are a set of 256 word-aligned registers that allow the LCD to be programmed. The format of the palette data is shown below. At the TFT mode, the palette RAM bit width will be increased as Table 8-12(2).

								23	22	21	20				
								B							
15	14	13	12					7	6	5	4				
G								R							

Figure 8-1 LCD Palette Word Bit Field for STN mode

								23	22	21	20	19			
								B							
15	14	13	12	11	10			7	6	5	4	3			
G								R							

Figure 8-2 LCD Palette Word Bit Field for TFT mode

8.4 Timings

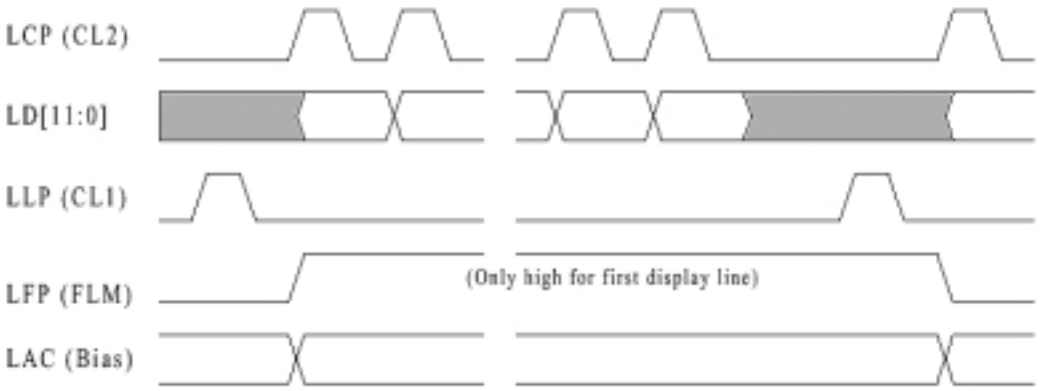


Figure 8-3 Example Mono STN LCD Panel Signal Waveforms

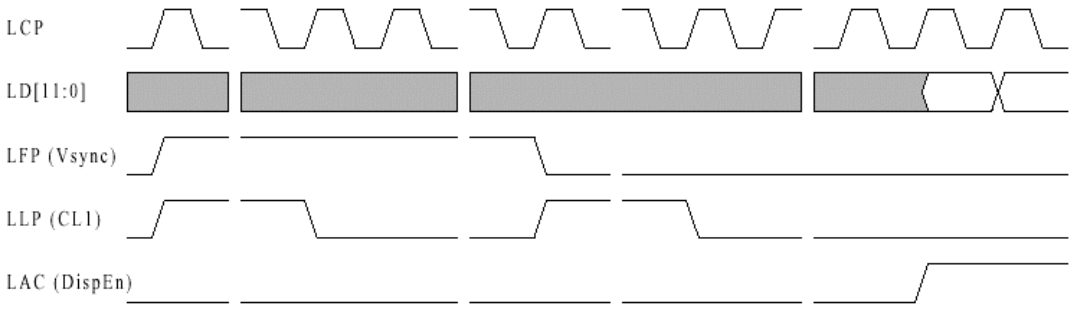


Figure 8-4 Example TFT Signal Waveforms, Start of Frame

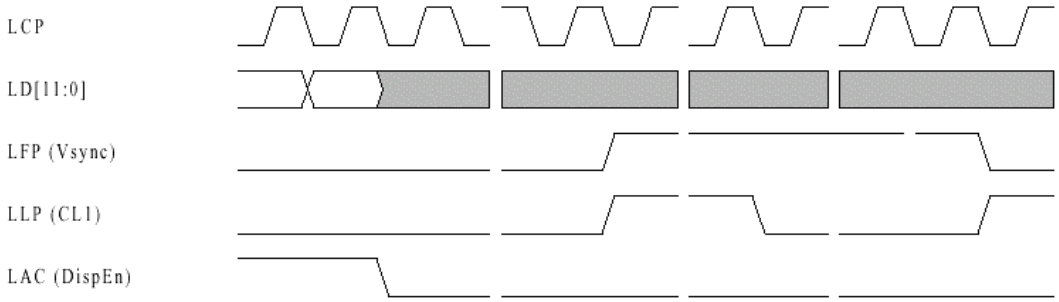


Figure 8-5 Example TFT Signal Waveforms, End of Last Line

9 FAST AMBA PERIPHERALS

9.1 DMA Controller

This chip includes a four-channel direct memory access controller (DMAC). The DMAC can be used in place of the CPU to perform high-speed transfers between peripheral devices and memory space. Note the DMA controller can only transfer data to and from SDRAM. Transfers to addresses other than SDRAM will produce unpredictable results.

Features

- Four Channels
- 4 GB of address space on the architecture
- Max Transfer rate: 910MB/s
- Max Transfer number: 16384
- Address mode: Single address is supported.
- Channel function: Transfer mode is different in each channel.
  - i. **Channel 0:** This channel has a source address reload function, which is used by sound interface controller. The memory space of sound I/O device consists of double buffer. The sound interface uses exception bus mode and word access. Exception bus mode: when the request is active, DMAC serves only one time operation.
  - ii. **Channel 1:** This channel is used by MMC interface block. The channel uses exception bus mode and word access.
  - iii. **Channel 2:** This channel is used by External IO device. The channel uses exception and burst bus mode and word access.
  - iv. **Channel 3:** This channel is used by USB. The channel uses burst bus mode and word access.
- Channel priority level: Selectable fixed mode
- Interrupt request: An interrupt request can be generated to the CPU after transfers end by the specified counts.

9.1.1 External Signals

Pin Name	Type	Description
nDMAREQ	I	DMA request input signal from External device.
nDMAACK	O	DMA acknowledge output signal to External Device.

9.1.2 Registers

Address	Name	Width	Default	Description
0x8000.4000	ADR0	32	0x0	Ch0 start address (buffer0)
0x8000.4004	ASR	32	0x0	Ch0 start address (buffer1)
0x8000.4008	TNR0	14	0x3FFF	Ch0 transfer number (buffer0)
0x8000.400C	TSR	14	0x3FFF	Ch0 transfer number (buffer1)
0x8000.4010	CCR0	4	0x0	Ch0 control
0x8000.4014	ADR1	32	0x0	Ch1 start address
0x8000.4018	TNR1	14	0x3FFF	Ch1 transfer number
0x8000.401C	CCR1	3	0x0	Ch1 control
0x8000.4020	ADR2	32	0x0	Ch2 start address
0x8000.4024	TNR2	14	0x3FFF	Ch2 transfer number
0x8000.4028	CCR2	8	0x0	Ch2 control
0x8000.402C	ADR3	32	0x0	Ch3 start address
0x8000.4030	TNR3	14	0x3FFF	Ch3 transfer number
0x8000.4034	CCR3	3	0x0	Ch3 control
0x8000.4038~ 0x8000.4040	-	-	-	Reserved
0x8000.4044	FLAGR	5	0x0	Interrupt flag
0x8000.4048	TESTR0	14	0x0	Test register0

0x8000.404C	TESTR1	32	0x0	Test register1
0x8000.4050	TESTR2	9	0x0	Test register2
0x8000.4054	DMAOR	3	0x0	DMA operation

Table 9-1 DMA Controller Register Summary

9.1.2.1 ADR0

31	30	29	...	2	1	0
ADR0						

Bits	Type	Function
31:0	R/W	DMAC 0 Source Address (Buffer 0 Address) This channel transfers data from External Memory to the Sound Interface block

9.1.2.2 ASR

31	30	29	...	2	1	0
ASR						

Bits	Type	Function
31:0	R/W	DMAC 0 Sound Address (Buffer 1 Address) This channel transfers data from External Memory to Sound Interface block. *This value will be automatically reloaded.*

9.1.2.3 TNR0

-	13	12	...	1	0
Reserved			TNR0		

Bits	Type	Function
13:0	R/W	DMAC0 Transfer Number (Maximum Transfer Number of Buffer0 is 0x4000 word)

9.1.2.4 TSR

-	13	12	...	1	0
Reserved			TSR		

Bits	Type	Function
13:0	R/W	DMAC0 Sound Transfer Number (Maximum Transfer Number of Buffer1 is 0x4000 word) *This value will be automatically reloaded.*

9.1.2.5 CCR0

-	3	2	1	0
Reserved	I2SEN	MASK01	MASK00	DMEN0

Bits	Type	Function
3	R/W	Select Sound (1'b0) or I2S (1'b1)
2	R/W	Buffer 1 transfer end interrupt mask bit Interrupt request is generated if data transfer end by the specified count (1'b1)
1	R/W	Buffer 0 transfer end interrupt mask bit Interrupt request is not generated even if data transfer end by the specified count (1'b0)
0	R/W	Channel 0 enable (1'b1)

9.1.2.6 ADR1

31	30	29	...	2	1	0
ADR1						

Bits	Type	Function
31:0	R/W	This value is start address of RX buffer.

9.1.2.7 TNR1

-	13	12	...	1	0
Reserved			TNR1		

Bits	Type	Function
13:0	R/W	Maximum Transfer Number of DMAC1 is 0x4000 word

9.1.2.8 CCR1

-	2	1	0
Reserved	MODE1	MASK1	DMEN1

Bits	Type	Function
2	R/W	The mask bit of transfer end interrupt
1	R/W	When LOW, transfer from memory to I/O. When HIGH, transfer from I/O to memory.
0	R/W	Channel 1 enable bit

9.1.2.9 ADR2

31	30	29	...	2	1	0
ADR2						

Bits	Type	Function
31:0	R/W	This value is the start address of DMA Channel for External IO device

9.1.2.10TNR2

-	13	12	...	1	0
Reserved			TNR2		

Bits	Type	Function
13:0	R/W	Maximum Transfer Number of DMAC2 is 0x4000 word

9.1.2.11CCR2

-	7	6	5	4	3	2	1	0
Reserved	BURST		TYPE	SIZE		MASK2	MODE2	DMEN2

Bits	Type	Function
7:6	R/W	Burst Length 11: 32 word 10: 16 word 01: 8 word 00: 4 word
5	R/W	Transfer Type 0: cycle-steal mode 1: burst mode
4:3	R/W	Transfer Size 11: reserved 10: word 01: half word 00: byte
2	R/W	The mask bit of transfer end interrupt
1	R/W	When LOW, transfer from memory to I/O. When HIGH, transfer from I/O to memory.
0	R/W	Channel 2 enable bit

9.1.2.12ADR3

31	30	29	...	2	1	0
ADR3						

Bits	Type	Function
31:0	R/W	This value is the start address of DMA Channel for USB Controller

9.1.2.13TNR3

-	13	12	...	1	0
Reserved	TNR3				

Bits	Type	Function
13:0	R/W	Maximum Transfer Number of DMAC3 is 0x4000 word

9.1.2.14CCR3

-	2	1	0
Reserved	MODE3	MASK3	DMEN3

Bits	Type	Function
2	R/W	The mask bit of transfer end interrupt
1	R/W	When LOW, transfer from memory to I/O. When HIGH, transfer from I/O to memory.
0	R/W	Channel 3 enable bit

9.1.2.15FLAGR

-	4	3	2	1	0
Reserved	FLAG3	FLAG2	FLAG1	FLAG01	FLAG00

Bits	Type	Function
4	R/W	Transfer end interrupt flag bit for channel 3
3	R/W	Transfer end interrupt flag bit for channel 2
2	R/W	Transfer end interrupt flag bit for channel 1
1	R/W	Buffer 1 transfer end interrupt flag bit for channel 0
0	R/W	Buffer 0 transfer end interrupt flag bit for channel 0

Note: Each flag bits is cleared by writing '1'.

9.1.2.16TESTR0

-	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TTC3	TTC2	TTC1	TTC0	TAC3	TAC2	TAC1	TAC0	TR3	TR2	TR1	TR0	TEX	TIN

Bits	Type	Function
13	R/W	Carry-In bit for testing the transfer counter of channel 3
12	R/W	Carry-In bit for testing the transfer counter of channel 2
11	R/W	Carry-In bit for testing the transfer counter of channel 1
10	R/W	Carry-In bit for testing the transfer counter of channel 0
9	R/W	Carry-In bit for testing the address counter of channel 3
8	R/W	Carry-In bit for testing the address counter of channel 2
7	R/W	Carry-In bit for testing the address counter of channel 1
6	R/W	Carry-In bit for testing the address counter of channel 0
5	R/W	Test request of channel 3
4	R/W	Test request of channel 2

3	R/W	Test request of channel 1
2	R/W	Test request of channel 0
1	R/W	Test external mode
0	R/W	Test internal mode

9.1.2.17TESTR1

31	30	29	...	2	1	0
TESTR1						

Bits	Type	Function
31:0	R	When in test mode, this register latches the address of DMAC.

9.1.2.18TESTR2

-	8	7	6	5	4	3	2	1	0
Reserved	TCHSEL		TINT	TTRAN		TSIZE		TREQ	TWR

Bits	Type	Function
5	R	Channel select
4	R	Transfer end interrupt of DMA
3	R	BTRAN[1:0]
2	R	BSIZE[1:0]
1	R	DMAC request signal
0	R	BWRITE

9.1.2.19DMAOR

-	2	1	0
Reserved	PRMD		DMAEN

Bits	Type	Function
2;1	R/W	Select the priority level between channels when there are transfer requests for multiple channels simultaneously 11: ch0 > ch1 > ch2 > ch3 10: ch2 > ch3 > ch1 > ch0 01: ch3 > ch1 > ch0 > ch2 00: ch1 > ch2 > ch3 > ch0 (initial value)
0	R/W	Enable DMA transfer on all channels (1'b1)

9.1.3 DMAC operation

After the DMA address register (ADR, ASR), DMA transfer number register (TNR), DMA channel control register (CCR), and DMA operation register (DMAOR) is set, the DMAC transfers data according to the following procedure.

- See if the DMEN bit of CCR and the DMAEN of DMAOR are enabled.
- When a transfer request comes and transfer condition is enabled, the DMAC transfers data according to bus size, address mode and bus mode.
- When the specified number of transfer has been competed (TNR = count value), the transfer ends normally. If the MASK bit of the CCR is set to 1 at this time, the DMA transfer end interrupt is sent to the CPU.

DMA Channel Priority
When the DMAC receives simultaneous transfer requests, it selects a channel according or a predetermined priority order. Priority order bits, PRMD in the DMA operation register, select each channel's priority.

DMA bus mode

Burst mode
Once the bus mastership is obtained, the transfer is performed continuously until the transfer end condition is satisfied. However, when the nDMREQ pin is driven high, the bus passes to the other bus master after current cycle ends. DMA request is nDMREQ level detection.

nDMREQ
CPU
CPU
DMA
DMA
DMA
DMA
CPU

Exception mode (Cycle-steal mode)
In the exception mode, the bus mastership is given to another bus master after a one-transfer-unit

DMA transfer.
The DMA request should be disabled by I/O device module. DMA request is nDMREQ level detection.
nDMREQ
CPU
DMA
CPU
CPU
DMA
CPU
CPU

9.2 I²S Transmitter

I²S interface is a serial sound data interface for sound applications. HMS30C7202 has I2S support transmitter block to connect with sound codec chip or other appliances. Originally it can just support 16-bit stereo sound data but also has capability to treat more than 16-bit data anyway.
This block has 8-word depth FIFO and support DMA transfer through DMA channel 1 (It shares DMA channel 1 with sound DAC interface block so user can not use both function at the same time).
In some transfer mode, there's some restriction to set registers and functions. For example user shouldn't set CLK32 bit in control register in more than 16-bit data mode unless only front 16-bit data are transferred.

FEATURES

- 32-bit AMBA APB bus interface
- Support 48, 44.1, 32, 24, 22.05, 16, 12, 11.025, 8 kHz sample frequency
- Can use external clock (through pin mux)
- 8-depth word width FIFO with valid tag bit
- 16, 18, 20, 22, 24, and 32-bit sound data transfer with MSB justified format
- Only support I²S interface

9.2.1 External Signals

Pin Name	Type	Description
ISD	O	Serial sound data output (muxed with KSCANO3)
ISCLK	O	I²S clock output (muxed with KSCANO4)
ISWS	O	Word select signal output (muxed with KSCANI3)
ISE_CLK	I	External clock input (optional, muxed with KSCANI4)

9.2.2 Registers

Address	Name	Width	Default	Description
0x8001.4000	I2SCR	8	0x0	I2S Control Register
0x8001.4004	I2SDMACR	8	0x0	I2S DMA Control Register
0x8001.4008	BBCR	8	0x0	Bit Clock Control Register
0x8001.400C	I2SFCR	8	0x0	I2S FIFO Control Register
0x8001.4010	I2SSR	8	0x0	I2S Status Register
0x8001.4014	I2SISR	8	0x0	I2S Interrupt Status Register
0x8001.4040	I2SFIFO	32	0x0	I2S Data FIFO
0x8001.4080	I2STSTCR	8	0x0	I2S Test Control Register
0x8001.4084	I2STSTCK	8	0x0	I2S Test Clock
0x8001.4088	I2STSTR1	8	0x0	I2S Test Register 1
0x8001.408C	I2STSTR2	8	0x0	I2S Test Register 2

Table 9-2 I²S Transmitter Register Summary

9.2.2.1 I²S Control Register (I2SCR)

I²SCR control I²S transmitter operation, interrupt masking, dummy cycle insertion and test mode enable. CHSEL bit to indicate the first sending bit is right data and WS start with high state (right), but in 16-bit sound data mode upper and lower half word data will be swapped and WS start with low state.

7	6	5	4	3	2	1	0
I2SE	I2STE	EOTMSK	FUDMSK	ED	DMMY	CHSEL	I2STST

Bits	Type	Function
7	R/W	I²S transmitter enable Write "1" to enable I2S module, unless "0" bit mask operation clock to save power.
6	R/W	I²S transmit start. After fill FIFO with some data, write "1" to start I2S data transmission.
5	R/W	End of Data interrupt Mask

4	R/W	I²S interrupt mask. Write “1” to enable interrupt signal assertion.
3	R/W	End of Data When this bit is set, FIFO under-run interrupt will not strobe at FIFO underrun situation and automatically clear I2STE bit, then the transmission will be completed.
2	R/W	Dummy WS cycle enable If this bit was set with “1”, the first one cycle of WS signal has No sound data. To give some time to audio codec for sync.
1	R/W	Channel select This module always transmit lower 16-bit first in 16bit-data mode. User can indicate the first data is right or left 0: lower 16-bit is left channel data, WS starts with “low” state 1: lower 16-bit is right channel data, WS starts with “high”
0	R/W	I²S block test enable. Set to zero always

9.2.2.2 I²S DMA Control Register (I2SDMACR)

7	6						
DMAEN	REQMSK						

Bits	Type	Function
7	R/W	I²S DMA operation enable Write “1” to enable DMA operation
6	R/W	I²S DMA Request mask. Write “1” to enable DMA Request
5:0	-	

9.2.2.3 I²S Bit Clock Control Register (BCCR)

BCCR controls main clock and transmission frequency. If external clock enabled, there’s no meaning SELWS [3:0]. External clock only will be divided by SECLK [1:0]. Use CLK32 bit to use 32 SCLK cycles for each WS duration (each left, right channel has 16 SCLK cycles), default number of SCLK cycles are 64(cover up to 32-bit transfer). But in some transfer rate, there are more than 96 SCLKs.

7	6	5	4	3	2	1	0
ECLKEN	SELCLK		CLK32	SELWS			

Bits	Type	Function
7	R/W	External clock enable Set “1” to use external clock.
6:5	R/W	Select SCLK divider 00: bypass (to use 256x clock) 01: divide by 2 10: divide by 4 (to use 64x clock) 11: divide by 8 (to use 32x clock)
4	R/W	Use x32 Internal SCLK (instead of x64 clock) User can set “1” this bit with 16-bit sound data, but if data length was over 16-bit, user should use x64 clock.
3:0	R/W	Select transfer clock (WS frequency, effective only internal clock mode) SELWS [3]: 0 = 2.8224MHz base (44.1, 22.05..) 1 = 3.072MHz base (48, 32, 24..) SELWS [2:0]: 000 = bypass (base clock) 001 = 2/3 base clock (32k) 010 = 1/2 base clock 011 = 1/3 base clock 100 = 1/4 base clock 101 = 1/6 base clock

111 = Not available

Examples
Value
Frequency
Value
Frequency

0000
44.1 KHz
0010
22.05 KHz

0100
11.025 KHz

1000
48 KHz
1001
32 KHz

1010
24 KHz
1011
16 KHz

1100
12 KHz
1101
8 KHz

9.2.2.4 I²S FIFO Control register (I2SFCR) – read/write register, initial value is 0x00

DWS indicate valid data width of LSB aligned sound data. If MSB justified sound data stored in a memory location, user can use 32-bit mode to bypass align function. All DWS bit are “0”, assume the lower half word is left data, if not, user can swap with CHSEL bit of I2SCR.

	6	5	4	3	2	1	0
	DWS			FIDT			

Bits	Type	Function
7	-	Reserved
6:4	R/W	Select transmitted data width 000: 16-bit data 001: 18-bit data 010: 20-bit data 011: 22-bit data 100: 24-bit data 101: 32-bit data 111: Not available
3:0	R/W	FIFO interrupt depth threshold (8-word depth) Select threshold value for data request. Recommended value is above 3. This value must be corresponded with DMA transfer size.

9.2.2.5 I²S Status register (I2SSR)

			4	3	2	1	0

			DCNT

Bits	Type	Function
7:5	-	Reserved
4:0	R	Number of transmitted data (transmit count value)

9.2.2.6 I²S Interrupt Status register (I2SISR)

This register contains the information of interrupt source and keeps the current status until it is cleared.

7	6	5				1	0
FUDR	DREQ	EOT				FEMPTY	TCH

Bits	Type	Function
7	R/W	FIFO underrun occurred (can use as interrupt) Write “1” to clear this status bit
6	R/W	Data Request interrupt Status Write “1” to clear this status bit
5	R/W	End of Transmission Write “1” to clear this status bit
4:2	-	Reserved
1	R	FIFO Empty Status (read only)
0	R	Transmit channel (left or right : WS)

9.2.2.7 I²S FIFO Register (I2SFIFO)

I²S block has 32-bit (word) size 8 depths FIFO. User can write 8 times with a word size data but cannot read directly from this FIFO.

Bits	Type	Function
31:0	W	32 bits 8 depths FIFO Data Register.

9.2.3 I²S Operation

9.2.3.1 Serial Data

Serial data is transmitted in two’s complement with the MSB first. The MSB is transmitted first because the transmitter and receiver may have different word lengths. It isn’t necessary for the transmitter to know how many bits the receiver can handle, nor does the receiver need to know how many bits are being transmitted.
When the system word length is greater than the transmitter word length, the word is truncated (least significant data bits are set to ‘0’) for data transmission. If the receiver is sent more bits than its word length, the bits after the LSB are ignored. On the other hand, if the receiver is sent fewer bits than its word length, the missing bits are set to zero internally. And so, the MSB has a fixed position, whereas the position of the LSB depends on the word length. The transmitter always sends the MSB of the next word one clock period after the WS changes.
Serial data sent by the transmitter may be synchronized with either the trailing (HIGH-to-LOW) or the leading (LOW-to-HIGH) edge of the clock signal. However, the serial data must be latched into the receiver on the leading edge of the serial clock signal, and so there are some restrictions when transmitting data that is synchronized with the leading edge.

9.2.3.2 Word Select

The word select line indicates the channel being transmitted:

- ISWS = 0: channel 1 (left)
- ISWS = 1: channel 2 (right)

ISWS may change either on a trailing or leading edge of the serial clock, but it doesn’t need to be symmetrical.

In the slave, this signal is latched on the leading edge of the clock signal. The ISWS line changes one clock period before the MSB is transmitted. This allows the slave transmitter to derive synchronous timing of the serial data that will be set up for transmission. Furthermore, it enables the receiver to store the previous word and clear the input for the next word

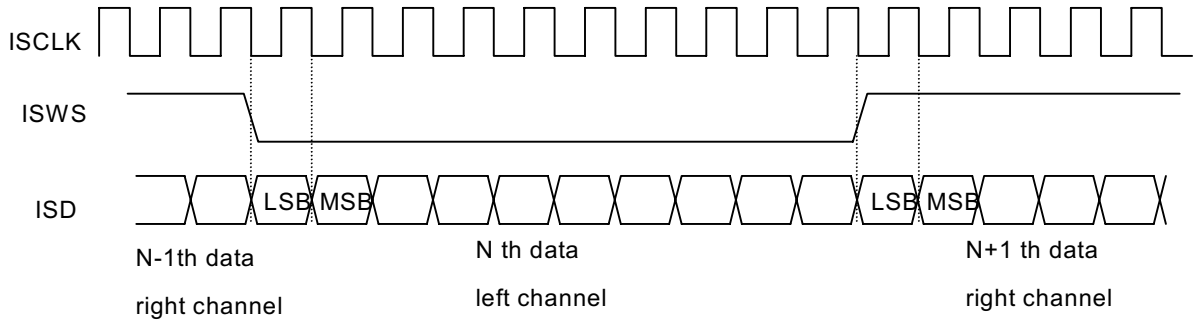


Figure 9-1 I^2S Timing Diagram

9.3 MMC/ SPI Controller

The SPI is a high-speed synchronous serial port for communicating to external devices. The SPI in this document is for MMC.
SPI-MMC is byte-orientated and every command, response and data block is built with a byte (8-bit). SPI-MMC messages are built from command, response and data-block tokens. All communication between CP and MMC is controlled by the CP (master).
Serial data transmission through SPI starts when the chip-select (CS) is asserted (i.e. when the CS goes to LOW) and ends when the chip-select is released (i.e. when the CS goes to HIGH).
Every MMC token transferred on the data signal is protected by CRC bits. But MMC offers a non-protected mode that enables a system built with reliable data links to exclude the hardware or firmware required for implementing the CRC generation and verification functions.
In the non-protected mode, the CRC bits of the command, response and data tokens are still required in the tokens; they are, however, defined as "don't care" for the transmitters and are ignored by the receivers.
MMC is initialized in the non-protected mode. The CP can turn this option on and off using the CRCONOFF command (CMD39). We assume that CRC is processed by software.

9.3.1 External Signals

Pin Name	Type	Description
SSDO	O	MMC card controller data output
SSDI	I	MMC card controller data input
SSCLK	O	MMC card controller clock output
nSSCS	O	MMC card controller chip select

9.3.2 Registers

Address	Name	Width	Default	Description
0x8001.4000	SPICR		0x20	SPI control register
0x8001.4004	SPISR		0x0	SPI status register
0x8001.4008	XCHCNT		0x0	Number of exchange data
0x8001.400C	TXBUFF		0x0	TX data buffer (8*8 bits)
0x8001.4010	RXBUFF		0x0	RX data buffer (8*8 bits)
0x8001.401C	ResetReg		0x0	SPI reset register

Table 9-3 SPIMMC Controller Register Summary

9.3.2.1 SPIMMC Control Register (SPICR)

	6	5	4	3	2	1	0
	DataRate	CS	XCHMode	TestMode	LOOP	SPIEN	XCH

Bits	Type	Function
7	-	Reserved
6	R/W	These bits select the baud rate of the SPICLK based on divisions of the system clock. The master clock for the SPIMMC is PCLK. The bits are encoded as: 0 = Bypass 1 = Divide by 2
5	R/W	This bit is Chip select signal. In order to communicate external device (MMC), CP asserts 0 in this bit. 0 = when CP can exchange data with external device (MMC) 1 = when CP cannot exchange data with external device (MMC)
4	R/W	This bit determines the direction of transfer 0 = when CP have valid data to send to MMC (send mode) 1 = when CP have valid data to receive from MMC (receive mode)
3	R/W	When TestMode bit is set, SPI-MMC block is in TIC mode. When Tic mode, the operation of the SPI-MMC is same in normal mode except that Clock source is not PCLK but TCLK that is made in the block. 0 = Normal operation 1 = The SPI-MMC block is in TIC mode

2	R/W	When set, this bit selects the local loopback operation. The transmitter output is internally connected to the receiver input. When in loopback mode, the operation of SPI-MMC block is same in normal mode except MISO is internally connected MOSI. 0 = Normal operation 1 = The SPI-MMC block is in loopback mode
1	R/W	This bit enables the SPIMMC. The enable should be asserted before initiating an exchange and should be negated after the exchange is complete. When the SPIEN bit is cleared, consumes minimal power. 0 = SPI master disable 1 = SPI master enable
0	R/W	This bit triggers the state machine to generate clocks at the selected bit rate. 1 = Initiate exchange 0 = No exchange occurs

9.3.2.2 SPIMMC Status Register (SPISR)

7	6	5					
TXET	XCHDONE	RXFULL					

Bits	Type	Function
7	R	This bit is set when TX data buffer is empty. If TX empty goes HIGH, a serial peripheral interrupt is generated. Clearing the TX empty bit is accomplished by reading the SPISR.
6	R	This bit is set when exchange is completed between CP and MMC. If XCHDONE bit goes HIGH, a serial peripheral interrupt is generated. Clearing the XCHDONE bit is accomplished by reading the SPISR.
5	R	This bit is set when RX data buffer is full. If RX full bit goes HIGH, a serial peripheral interrupt is generated. Clearing the RX full bit is accomplished by reading the SPISR
4:0	-	Reserved

9.3.2.3 SPIMMC XCH Counter Register (XCHCNT)

9	8	7	6	5	4	3	2	1	0
XCH COUNTER									

Bits	Type	Function
9:0	R/W	Number of bytes to be exchanged between CP and SPI

9.3.2.4 SPIMMC TX Data Buffer Register (TXBUFF)

This 8-bit register is an entry point of the TX FIFO. When CP writes an 8-bit data to this register, the SPI-MMC block shifts the content of the TX FIFO and appends the new data to the FIFO.

7	6	5	4	3	2	1	0
TX FIFO ENTRY POINT							

Bits	Type	Function
7:0	W	TX FIFO's Entry Point

9.3.2.5 SPIMMC RX Data Buffer Register (RXBUFF)

This register is the access point of the RX FIFO. When CP reads one data item from this register, the SPI-MMC block shifts the RX FIFO so that the next data item becomes available at this location.

7	6	5	4	3	2	1	0
RX FIFO ACCESS POINT							

Bits	Type	Function

7:0	R	RX FIFO's Access Point

9.3.2.6 SPIMMC Reset Register (ResetReg)

							0
							RESET

Bits	Type	Function
7:1	-	Reserved
7:0	R/W	When CP writes 0 to this location, all registers and counters of the SPI-MMC block are cleared.

9.3.3 Timings

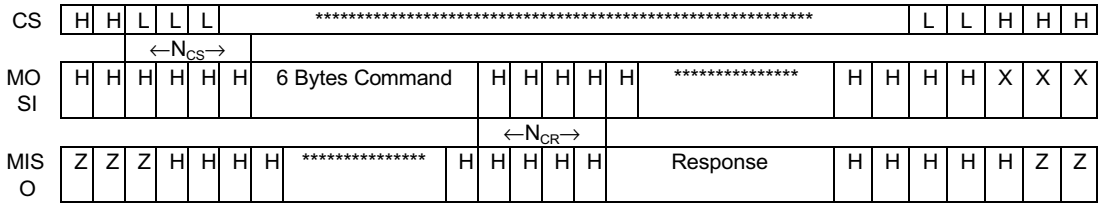
All timing diagrams use the following schematics and abbreviations.

Name	Description	Name	Description
H	Signal is HIGH (logic 1)	Busy	Busy token
L	Signal is LOW (logic 0)	Command	Command token
X	Don't care	Response	Response token
Z	High Impedance State	DataBlk	Data token
*	Repeater		

All timing values are defined as outlined below.

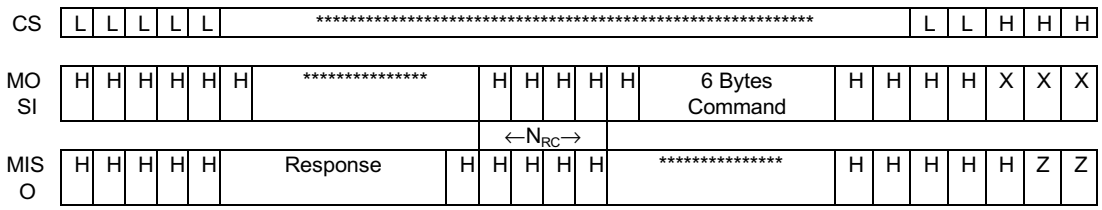
Command/Response

Host command to card response: card is ready

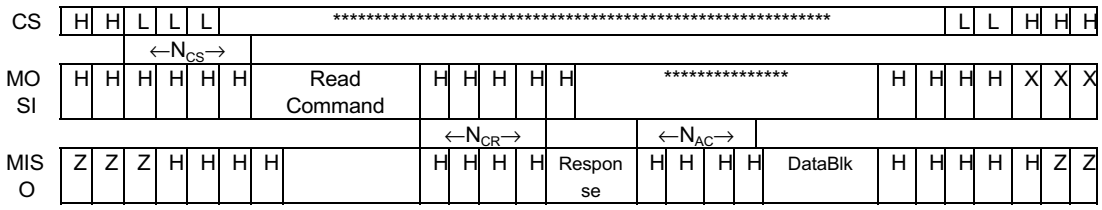


Host command to card response: card is busy

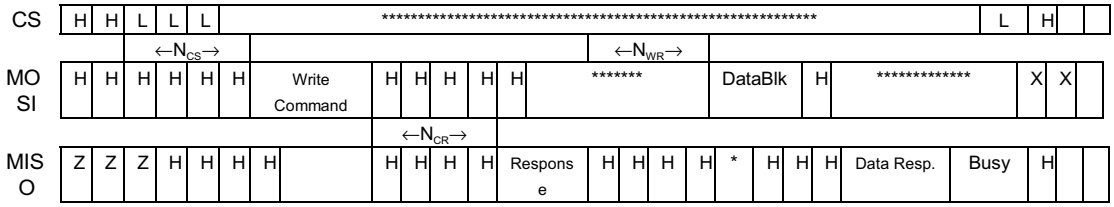
Card response to host command:



Data read



Data write



Timing constants definitions

Name	Minimum	Maximum	Unit
N_{CS}	0	-	8 Clock Cycles
N_{CR}	1	2	8 Clock Cycles
N_{RC}	1	-	8 Clock Cycles
N_{AC}	1	-	8 Clock Cycles
N_{WR}	1	-	8 Clock Cycles

9.3.4 MMC/ SPI Operation

After CP writes a sequence of data to the TX FIFO, the content of the FIFO is loaded into the TX shift register and is shifted out serially one byte at a time. When all elements in the TX FIFO are transferred to the TX shift register, the SPI-MMC issues an interrupt to CP, which may fill the TX FIFO for further data transfer.
Serial input data is shifted into the RX shift register. After 8 bits are shifted in, the content of the RX shift register is copied into the RX FIFO. When the RX FIFO is full, the SPI-MMC issues an interrupt to CP through the SPIIRQ signal. CP reads the content of the RX FIFO in an interrupt service routine.
The timing and control block produces all necessary control signals of the SPI-MMC block including SPICLK. The frequency of SPICLK signal is programmable.
SPI-MMC transfer's protocol is command and response. Whenever CP sends a command to MMC (via SPI), MMC sends CP (via SPI) a response. The response is variable length for command--for example, there is 1-, 6-, and 17-byte. There is only 6 bytes in command.

Consider the sequence of operations that occur in a read transfer.

1. CP sends a reset signal to the SPI-MMC block. In other word, CP writes "0" to bit in the ResetReg register. The signal is used to clear counters inside the block. Before new exchange begins and the content of XCHCOUNTER is changed, and transmit mode is changed (XCHMODE BIT in the SPICR), CP must send a reset signal to the SPI-MMC block.
2. First, CP set up the SPICR register. In this example, XCHMODE is send mode.
3. CP writes number to send into XCHCOUNTER register.
4. CP writes "Data read command (CMD17)" into the TX FIFO.
5. CP asserts CS signal. In other words, CP write 0 to CS bit in the SPICR.
6. CP sends a start signal to SPI-MMC. In other word, CP set XCH bit in the SPICR.
7. The SPI-MMC block sends out 6 bytes of command data from TX FIFO through TX shift register.
8. The SPI-MMC block issues the interrupt after it send all data in TX FIFO.
9. The CP reads the SPISR register in The SPI-MMC block and disable start signal (reset XCH bit). In other words, CP writes the SPICR register.
10. CP sends a reset signal to the SPI-MMC block. In other word, CP writes 0 to bit in the ResetReg

register. The signal is used to clear counters inside the block. Before new exchange begins and the content of XCHCOUNTER is changed, and transmit mode is changed (XCHMODE BIT in the SPICR), CP must send a reset signal to the SPI-MMC block.

11. CP changes transmit mode (XCHMODE is receive mode).
12. The CP writes number to be received into XCHCOUNTER register.
13. CP sends a start signal to SPI-MMC (set XCH bit).
14. Then SPI-MMC controller receives response from MMC.
15. After SPI-MMC receives 1 byte (for CMD17 command), it sets XCH DONE status bits and it issues an interrupt to a CP.
16. The CP reads the SPISR register in the SPI-MMC block and disable start signal (reset XCH bit). In other words, CP writes the SPICR register.
17. The CP reads data RX FIFO.
18. After CP takes this response data and examine it, CP act as response data. If there is no error indication in response, CP informs SPI-MMC block that MMC sends data to it.
19. CP sends a reset signal to the SPI-MMC block. In other words, CP writes 0 to bit in the Reset register. The signal is used to clear counters inside the block. Before new exchange begins and the content of XCHCOUNTER is changed, and transmit mode is changed (XCHMODE BIT in the SPICR), CP must send a reset signal to the SPI-MMC block.
20. The CP writes number to be received into XCHCOUNTER register.
21. CP sends a start signal to SPI-MMC (set XCH bit).
22. The SPI-MMC block receives data from MMC (for example, data length is from 4 byte to 515 byte).
23. If SPI-MMC receives data like RX FIFO size, SPI-MMC block sets the "RX FIFO full" status bit and issues an interrupt to CP. At this time SPICLK disable start signal for prevention of RX FIFO overrun. If CP takes all data in RX FIFO, CP sends a start signal and receives response to remain. Repeat it.
24. After SPI-MMC block receive all data from MMC, it sets the XCH DONE status bit and issues an interrupt to CP.
25. The CP reads the SPISR register in the SPI-MMC block and disable start signal (reset XCH bit). In other words, CP writes the SPICR register.
26. After CP takes last data from RX FIFO, CP de-asserts CS signal.

9.4 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 DMA access
- Marginal timing operation settable.

9.4.1 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.

9.4.2 Registers

Address	Name	Width	Default	Description
0x8001.6000	SMCCMD	32	0x0	SmartMedia Card Command register
0x8001.6004	SMCADR	27	0x0	SmartMedia Card Address register
0x8001.6008	SMCDATW	32	0x0	Data written to SmartMedia Card
0x8001.600C	SMCDATR	32	0x0	Data received from SmartMedia Card
0x8001.6010	SMCCONF	8	0x0	SmartMedia Card controller configuration register
0x8001.6014	SMCTIME	20	0x0	Timing parameter register
0x8001.601C	SMCSTAT	32	0x0	SmartMedia Card controller status register

Table 9-4 SmartMedia Controller Register Summary

9.4.2.1 SMC Command Register (SMCCMD)

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

9.4.2.2 SMC Address Register (SMCADR)

					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
S<CADR15 ~ SMCADR0															

Bits	Type	Function
26:0	R/W	SMC 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

9.4.2.3 SMC Data Write Register (SMCDATW)

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 or DMA 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: 1. 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. If DMA mode enabled, DMA interrupt will be repeated until it writes 528 byte data to SmartMedia. In normal mode, interrupt will be generated every 4 bytes write. 2. At the end of sequential data input, SmartMedia goes into page program mode by inputting second command in SMCCMD. Usually page program takes long time, no polling status register is recommended. SMC controller automatically generates job finish interrupt after SmartMedia comes back to ready mode.

9.4.2.4 SMC Data Read Register (SMCDATR)

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 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 or DMA controller read 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: 1. 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.
2. SMC controller will access SmartMedia with given command and address.
3. Interrupt (or DMA interrupt according to interrupt mode setting) 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.

Total access byte size can be known in SMCSTAT status register (for read/write).

9.4.2.5 SMC Configuration Register (SMCCONF)

31	6	5	4	3	2	1	0
POWER ENABLE	SAFE MARGIN	SMC ENABLE	CONT PAGE EN	INTR EN	DMA 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.
30:7	-	Reserved. Keep these bits to zero.
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	R/W	Continuous page read enable. If this bit set, then multi-page can be accessed in a single command and address setting. Usually DMA controller accesses multiple pages with only a start address and its predefined size. Set DMA access size in SMCTIME then enable this bit will automatically read or write SmartMedia with DMA mode and only return one job finish interrupt.
3	R/W	Interrupt enable. If reset, software must poll the status register of SMC controller. After each read or write word job ends, interrupt bit of SMCSTAT will be set. SMC controller never stop before read or write a whole page in a transaction so every time interrupt bit set software should feed or get data. To minimize CPU burden and to maximize BUS utilization, enabling both interrupt and DMA mode together is recommended.
2	R/W	DMA enable. If set, all interrupt during read or write data will be sent to DMA controller. Write finish interrupt, however, will be normal interrupt instead of DMA one.
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	Bigger than 64MB SmartMedia support enable. When use 128MB SmartMedia, set this bit high.

9.4.2.6 SMC Timing Parameter Register (SMCTIME)

31	30	29	28	27	26	25	24		22	21	20	19	18	17	16
DMA SIZE				WAIT COUNTER					BYTE COUNTER						
						9	8						2	1	0
						HIGH COUNTER							LOW COUNTER		

Bits	Type	Function
31:28	R/W	Multi-page DMA size bit. Maximum 15 pages are accessible at a time. 0000 = not defined. 0001 = 1 page 0010 = 2 pages … 1111 = 15 pages
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. 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. 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)

9.4.2.7 SMC Status Register (SMCSTAT)

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	DRQ	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 accessed byte address of a page.
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	R	DMA interrupt flag
0	R	Reset shows SMC is in idle mode. Set means SMC in working mode.

9.5 Sound Interface

The Sound Control Unit (SOC) is an interface block to transfer sound data to external speakers.
The SOC is an interface block used to send data to the external speaker through the internal 8-bit DA converter. It can process 44.1/22.05/11.025/8KHz sampled 8-bit mono or 16-bit stereo sound data.
This unit has a 32-bit register to receive sound data from the CPU through DMA or interrupt mode. This unit requests the DMA or interrupt controller every 32-bit processing time, which depends on the sampling frequency. It has two separate signals for DAC that indicate the direction of data for the stereo sound. Either higher or lower byte of 16-bit stereo sound data can be played through the left or right speaker by programming the control register. During mono playback, this unit sends the same data for the left and right channels.
There are two test registers. Both these registers should be cleared during normal operation. TICCLK port is also assigned for production test only.

Features

- Sound playback
- Supports programmable sampling rate
- 32-bit internal data register for DMA
- Auto DMA request
- 8-bit resolution DAC control
- Supports non-overlapping left/right signal for DAC
- Supports test mode

9.5.1 External Signals

Pin Name	Type	Description
ADACR	O	Sound DAC output for Right
ADACL	O	Sound DAC output for Left

9.5.2 Registers

Address	Name	Width	Default	Description
0x8001.3000	SCONT	8	0x0	Control register
0x8001.3004	SDADR	32	0x0	Data register
0x8001.3008	STOR	18	0x0	Test output register
0x8001.300C	STIR	15	0x0	Test input register
0x8001.3010	TICCLK	0	-	TIC clock register

Table 9-5 Sound Controller Register Summary

9.5.2.1 SCONT

-	7	6	5	4	3	2	1	0
Reserved	MONO	DMA	POR	DAC	RL	SAMP		INT

Bits	Type	Function
7	R/W	0 – stereo 1 – mono
6*	R/W	DMA request masking bit 0 - masking 1 – unmasking
5	R/W	This bit should be cleared to minimize power consumption when not in use. 0 - power down mode 1 - normal mode
4	R/W	DAC operation enable/disable. During disabled, DAC is in power save mode. 0 - DAC disable 1 - DAC enable
3	R/W	When cleared, lower byte data goes to left speaker. (ADACL pin)

		0 - lower byte data goes to ADACL pin 1 - lower byte data goes to ADACR pin
2:1	R/W	Programmable sampling rate 00 - 11.025KHz 01 - 22.05KHz 10 - 44.1KHz 11 - 8KHz
0*	R/W	0 Interrupt request masking bit 0 - masking 1 – unmasking

Note *Those bits marked with an asterisk should not be enabled simultaneously during normal operation. (The programmer can select only one--either Interrupt or DMA mode.)*

9.5.2.2 SDADR

This register can be programmed after setting Bit 5 of the SCONT register.

31	30	29	...	2	1	0
SDADR						

Bits	Type	Function
32	R/W	Sound Data This register receives data by DMA Controller or CPU. This unit processes the lower 16-bit data followed by the higher 16-bit data. After the lower 16-bit is processed, this unit is ready to receive new data and sends a request signal to DMA Controller or CPU. In mono mode, the lower byte is processed first followed by the higher byte.

9.5.2.3 STOR (test output register)

This register is used for the operation of DAC. This register should only be used for DAC test purposes, and should not be accessed during normal operation.

-	17	16	15	14	13	12	11	10	9	8	7-0
Reserved	TCK	TMOD	SCLK	DAPUL	TDAC	TINT	TDMA	TOP	TL	TR	TSD

Bits	Type	Function
17	R/W	When set, TICCLK is used as clock source instead of normal clock input during test: 0 - normal mode 1 - TICCLK mode
16	R/W	Only if set, the values of bit 15 and bit 14 replace the original Soundclk and Dapulse signal. Used only for test purposes. 0 - normal mode 1 - test mode
15	R/W	Soundclk signal input - see Note 1. When bit 16 is set, this bit is meaningful. The Soundclk signal is changed by the value of this bit.
14	R/W	Dapulse signal input - see Note 2. When bit 16 is set, this bit is meaningful. The Dapulse signal is changed by the value of this bit.
13	R/W	TICdac mode for DAC test. In this mode, PCLK is changed by TICCLK register: 0 - Normal mode 1 - TICdac mode
12	R/W	When only TICdac mode, this bit can be programmed: 0 - interrupt not request 1 - interrupt request
11	R/W	When only TICdac mode, this bit can be programmed: 0 - DMA not request 1 - DMA request
10	R/W	When only TICdac mode, this bit can be programmed: 0 - DAC operation run

		1 - DAC operation stop
9	R/W	When only TICdac mode, this bit can be programmed: 0 - DLEFT LOW 1 - DLEFT HIGH
8	R/W	When only TICdac mode, this bit can be programmed: 0 - DRIGHT LOW 1 - DRIGHT HIGH
7:0	R/W	SD signal. When only TICdac mode, this bit can be programmed.

Notes

(1) Soundclk: this is an internal signal used as a reference clock source to play the sound data.
(2) Dapulse: this is an internal signal used to make the DAC channel select signal.

9.5.2.4 STIR (test input register)

This register is for monitoring the unit status and the signals to DAC in both normal and TICdac mode.

-	7	6	5	4	3	2	1	0
Reserved	MONO	DMA	POR	DAC	RL	SAMP		INT

Bits	Type	Function
14	R	Soundclk state. This bit indicates the state of Soundclksignal
13	R	Dapulse state. This bit indicates the state of Dapulsesignal.
12	R	This bit is set by INT signal in Ticdac and normal mode: 0 - interrupt not request 1 - interrupt request
11	R	This bit is set by DRQ signal in Ticdac and normalmode. 0 - DMA not request 1 - DMA request
10	R	This bit is set by IOSTOP signal in Ticdac and normalmode. 0 - DAC operation run 1 - DAC operation stop
9	R	This bit is set by DLEFT signal in Ticdac and normalmode. 0 - DLEFT LOW 1 - DLEFT HIGH
8	R	This bit is set by DRIGHT signal in Ticdac and normalmode. 0 - DRIGHT LOW 1 - DRIGHT HIGH
7:0	R	SD signal. This bit is set by SD signal in Ticdac and nor-mal mode.

9.5.2.5 TICCLK (test clock register)

Whenever this register is accessed, TICCLK is generated. When STOR[17] bit is set, TICCLK is used as clock source instead of normal clock input.

9.6 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 1.0 for a full understanding of the USB protocol and its operation. (The USB specification 1.0 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 FIFO can be accessed by the DMAC (Direct Memory Controller), with service requests being signaled when either FIFO is full/empty. The external pins dedicated to this interface are UVPO, UVP, UVMO, UVM, URCVIN, nUSBOE and USUSPEND. These signals should be connected to USB transceiver such as PDIUSBP11 provided by Philip Semiconductor. Refer to data sheet PDIUSBP11). The interface of the USBD and the CPU uses DMAC to reduce CPU load of transferring data from external memory to USBD and from USBD to external memory. The CPU can also access the USBD using Interrupt controller, by setting the control register appropriately. This section also defines the interface of USBD and CPU. The USBD uses one dedicated DMA channel for receiving and transmitting data, so the DMAC should be programmed into receiving channel initially for both data transferring. If transferring data to USB host occurs (setting the control register bit), that is, USB host issue IN Token, and then DMAC should be programmed to transmitting channel. After transmitting data, DMAC should be programmed to the receiving channel again.

FEATURES

- Full universal serial bus specification 1.0 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 DMA service request signals to DMAC or Interrupt service request signals to the CPU)
- Supports high-speed USB transfer (12Mbps).
- There are two endpoint 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: 7202 Vendor ID: 05b4*

9.6.1 Block Diagram

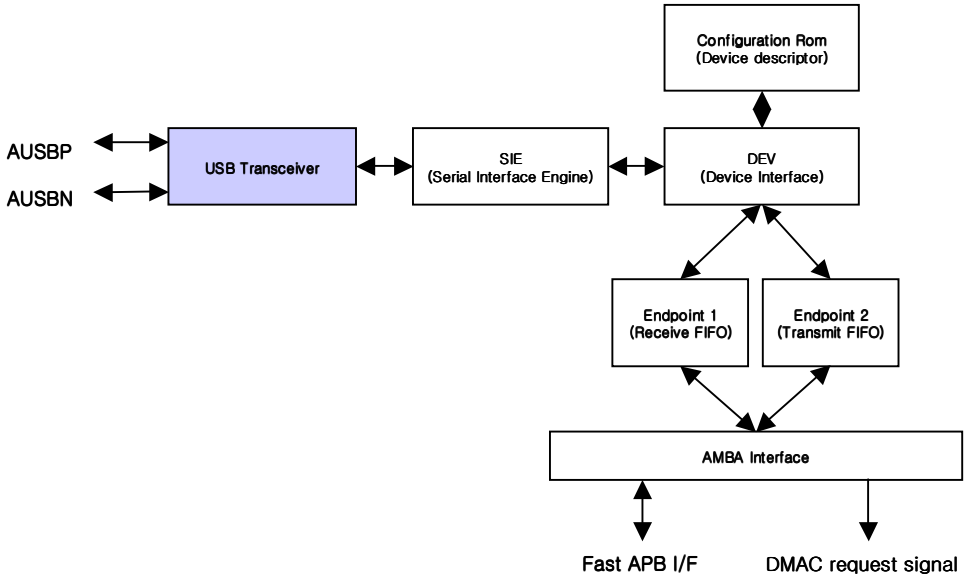


Figure 9-2 USB Block Diagram

The USB, Figure 12-16: USBD 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 to generate DMA request signal 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. Automatically it requests the DMA data handling when RX FIFO is full. In case of data transmitting through TX FIFO (when USB generates an OUT token, AMBA I/F generates Interrupt to CPU), the user should program the DMAC to transmitting channel, 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 signals and DMA service request signals.

9.6.2 Theory of Operation

The LGS 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 12-16: USBD Block Diagram on page 12-54, 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 12-17: LGS 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.

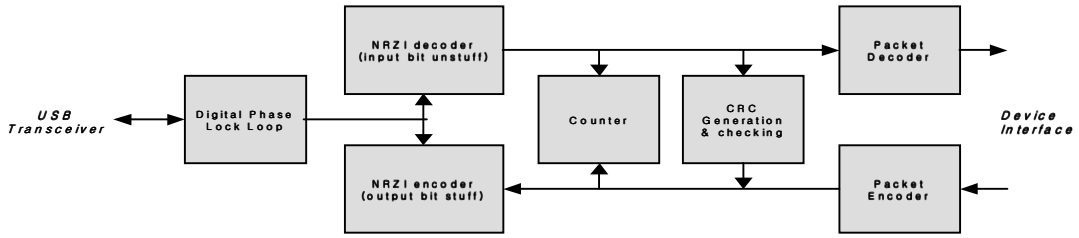


Figure 9-3 USB Serial Interface Engine

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 12-35: 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 9-6 USB Supported PID Types

Packet Encoder
The Packet Encoder creates outgoing packets based on signals from the DEV. Table 12-35: 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 1.0.

Device Interface
The DEV shown in Figure 12-18: 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 12-18: 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.

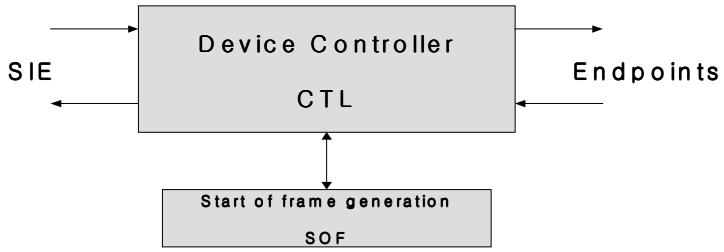


Figure 9-4 USB Device Interface Device Controller

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 12-36: 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	Endpoints Only	Set Configuration	9	Device
Set Feature	3	Not supported	Get Interface	10	Not supported
Set Address	5	Device	Set Interface	11	Not supported
Get Descriptor	6	Device	Synch Frame	12	Not supported
Set Descriptor	7	Not supported			

Table 9-7 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.

9.6.3 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 I/F and causes AMBA I/F to produce DMAC request signals. It contains the control logic for transferring 4 bytes at a read/write strobe generated by AMBA to obtain better efficiency of AMBA bus.

9.6.4 External Signals

Pin Name	Type	Description
AUSBP	I/O	USB transceiver signal for P+
AUSBN	I/O	USB transceiver signal for N+

9.6.5 Registers

Address	Name	Width	Default	Description
0x8001.2000	CONT0	1	0x0	USB I/F control register 0
0x8001.2004	CONT1	2	0x0	USB I/F control register 1
	RXDATA	32	-	Receive data register
	TXDATA	32	-	Transmit data register
0x8001.2008	STATUS	2	-	Status register
0x8001.200C	TicRXDATA	32	-	Receive data register for TIC mode
0x8001.2010	TicTXDATA	32	-	Transmit data for TIC mode
0x8001.2014	TicSEL	2	0x0	TIC mode select register
0x8001.2018	TicREG	3	0x0	TIC Input register
0x8001.201C	TicRESULT	3	0x0	TIC output register
0x8001.2020	SWRESET	1	0x0	Generate software reset to USBD
0x8001.2024	DRQMASK	2	0x0	DMA request masking register

Table 9-8 USB Slave interface Register Summary

9.6.5.1 CONT0

31	...	0
Reserved		TXEN

Bits	Type	Function
0	R/W	When 1, the function of USB is transmitting data from external memory to USB host PC. When `0', receiving data from USB host PC to external memory; default = 0. Before setting this bit 1, the user should program the DMAC to transmitting channel. If it is 1, AMBA I/F generate DMAC request signal for data transmitting after checking if Tx FIFO's status is empty.

9.6.5.2 CONT1

31	...	1	0
Reserved		MFULL	MEMPTY

Bits	Type	Function
1	R/W	Mask FIFO full interrupt bit. When it is `1' the FIFO full interrupt is masked.
0	R/W	Mask FIFO empty interrupt bit. When it is `1', the FIFO empty interrupt is masked.

9.6.5.3 RXDATA

31	...	24	23	...	16	15	...	8	7	...	0
LSB	N+3	MSB	LSB	N+2	MSB	LSB	N+1	MSB	LSB	N	MSB

Bits	Type	Function
31:0	-	32-bit receive data register. DMAC reads 4 bytes at data read strobe. Each byte is pre-sent lsb first. N is represented as Rx FIFO address.

9.6.5.4 TXDATA

31	...	24	23	...	16	15	...	8	7	...	0
LSB	N+3	MSB	LSB	N+2	MSB	LSB	N+1	MSB	LSB	N	MSB

Bits	Type	Function
31:0	-	32-bit transmit data register. DMAC writes 4 bytes at data write strobe. Each byte is presented lsb first. N is represented as Tx FIFO address.

9.6.5.5 STATUS

31	...	1	0
Reserved		EMPTY	FULL

Bits	Type	Function
1	R/W	This bit indicates that Tx FIFO is empty when the CONT0 TXEN bit is set to 1.
0	R/W	This bit indicates that Rx FIFO is filled. When it is 1, automatically, AMBA I/F generates a DMAC request signal. DMAC can read the RxDATA register according to read strobe.

9.6.5.6 TicRXDATA

31	...	24	23	...	16	15	...	8	7	...	0
LSB	N+3	MSB	LSB	N+2	MSB	LSB	N+1	MSB	LSB	N	MSB

Bits	Type	Function
31:0	R/W	32-bit test receive data register for TIC mode. TIC reads 4 bytes at data read strobe. each byte is presented lsb first. N is represented as Rx FIFO address.

9.6.5.7 TicTXDATA

31	...	24	23	...	16	15	...	8	7	...	0
LSB	N+3	MSB	LSB	N+2	MSB	LSB	N+1	MSB	LSB	N	MSB

Bits	Type	Function
31:0	R/W	32-bit transmit data register for TIC mode. TIC writes 4 bytes at data write strobe. Each byte is presented lsb first. N is represented as Tx FIFO address.

9.6.5.8 TicSEL

31	...	1	0
Reserved		CPUMODE	TICEN

Bits	Type	Function
1	R/W	CPU access mode enable bit. When CPUMODE = 1, the USBD is allowed to enter the CPU access mode. For DMAC mode, this bit should be 0.
0	R/W	TIC mode enable bit. When TICEN =1, USBD enters the TIC mode.

9.6.5.9 TicREG

31	...	2	1	0

Reserved	Tvmin	Tvpin	Trcvin

Bits	Type	Function
2	R/W	Test input for vmin
1	R/W	Test input for vpin
0	R/W	Test input for rcvin

9.6.5.10TicRESULT

31	...	2	1	0
Reserved		Tusboen	Tvmo	Tvpo

Bits	Type	Function
2	R/W	Test output for usboen
1	R/W	Test output for vmo
0	R/W	Test output for vpo

9.6.5.11SWRESET

31	...	0
Reserved		ESWRES

Bits	Type	Function
0	R/W	This bit should be set to `1' after PMU generated 48 MHz USBD clock. If this bit is `1', the dpll (digital pll) in the USBD is initialized and so USBD operates successfully. After data transfer has finished, the bit should be set to `0'.

9.6.5.12DRQMASK

31	...	1	0
Reserved		MRX	MTX

Bits	Type	Function
1	R/W	Mask FIFO full DMAC request signal. When it is `1', FIFO full DMAC request signal is masked.
0	R/W	Mask FIFO empty DMAC request signal When it is `1', FIFO empty DMAC request signal is masked.

Note *In CPU access mode, the DMAC request signal should be masked.*

DMAC I/F

This field describes the interface of the DMAC and the USBD. The USBD (Rx/Tx buffer) transmits and receives data from and to the DMAC based on the signal of the fast APB.

That is, after generating the DMA request signal, USBD expects the DMAC to produce PSELdmausb, PD, PSTB and PWRITE that are the fast APB signals. With these signals, the FIFOs in Rx buffer put data to PD and the FIFOs in Tx buffer get data from PD through the Quad Word Access.

10 SLOW AMBA PERIPHERALS

10.1 ADC Interface Controller

HMS30C7202 has internal ADC and ADC interface logic for analog applications (for example, touch panel, sound input, battery check, etc.). And also has other functions for direct interface with touch panel. It can be connected with passive (resistive type) touch panel directly, and built in tip down interrupt check function.

FEATURES

- 5-channel 10-bit ADC interface
- Internal tip down interrupt drive for touch panel interface
- Touch panel direct drive capability (do not need external circuit to connect touch panel)
- 4-sample data per one sampling point of touch panel (use 2 channels, X and Y)
- Main and backup battery check function (use 2 channels)
- Eight 32-byte sound data buffer (8-word buffer)
- Manual and Auto ADC power down mode

10.1.1 External Signals

Pin Name	Type	Description
	I/O	
	I/O	
	I/O	

10.1.2 Registers

Address	Name	Width	Default	Description
0x8002.9000	ADCCR		0x0	ADC Control Register
0x8002.9004	ADCTPCR		0x0	Touch panel control register
0x8002.9008	ADCBACR		0x0	Battery check Control Register
0x8002.900C	ADCSDCR		0x0	Sound Data Control Register
0x8002.9010	ADCISR		0x0	ADC Interrupt Status Register
0x8002.901C	ADCTDCSR		0x0X	Tip Down Control/Status Register
0x8002.9020	ADCDIRCR			ADC Direct Control Register
0x8002.9024	ADCDIRDATA			ADC Direct Data read register
0x8002.9030	ADCTPXDR0			Touch Panel X Data register 0
0x8002.9034	ADCTPXDR1			Touch Panel X Data register 1
0x8002.9038	ADCTPYDR0			Touch Panel Y Data register 0
0x8002.903C	ADCTPYDR1			Touch Panel Y Data register 1
0x8002.9040	ADCTPXDR2			Touch Panel X Data register 2
0x8002.9044	ADCTPXDR3			Touch Panel X Data register 3
0x8002.9048	ADCTPYDR2			Touch Panel Y Data register 2
0x8002.904C	ADCTPYDR3			Touch Panel Y Data register 3
0x8002.9050	ADCMBDATA			Main Battery check Data Register
0x8002.9054	ADCBBDATA			Backup Battery check Data Register
0x8002.9060	ADCSDATA0			Sound Data Register
0x8002.9064	ADCSDATA1			Sound Data Register
0x8002.9068	ADCSDATA2			Sound Data Register
0x8002.906C	ADCSDATA3			Sound Data Register
0x8002.9070	ADCSDATA4			Sound Data Register
0x8002.9074	ADCSDATA5			Sound Data Register
0x8002.9078	ADCSDATA6			Sound Data Register
0x8002.907C	ADCSDATA7			Sound Data Register
0x8002.90A0	ADCTSTCR			ADC interface Test Control Register
0x8002.90A4	ADCTSTCK			ADC interface Test Clock Reigster

0x8002.90A8	ADCTSTR1	ADC interface Test register 1
0x8002.90AC	ADCTSTR2	ADC interface Test register 2
0x8002.90B0	ADCTSTR3	ADC interface Test register 3
0x8002.90C0	ADCEXTEST	ADC Test with External signal

Table 10-1 ADC Controller Register Summary

10.1.2.1 ADC Control Register (ADCCR)

User can set ADCPD to save power consumption by ADC. But ADC needs 10-40 ms to self calibrate for normal operation. DIRECTC bit can be used for direct accessing from CPU to ADC without interface function logic. All direct control signals are describe in ADCDIRCR register field. Basically ADC core converts Analog data to Digital data continuously in every 16 ADC operation-clocks. WAIT bit field select conversion time of ADC because in certain case interface logic can read wrong or unstable value from ADC. SOP bit can be used for one-shot operation to save power. When this bit is set and all ADC functions are disable then interface logic strobe “power down” signal to ADC core. LONGCAL signal selects self-calibration time. Initially this bit set as “0” it means short calibration time (about 10 ms). But if first a couple of data were wrong value, user should select long calibration time (about 40 ms).

7	6			3	2	1	0
ADCPD	DIRECTC			WAIT		SOP	LONGCAL

Bits	Type	Function
7	R/W	ADC power down bit. Write “1” to go ADC power save mode. This bit blocks the clock to ADC, so ADC consumes no power when this bit is set. But after release this bit, ADC need 10 ~ 40 ms calibration time to normal operation.
6	R/W	If this bit was set, CPU access directly ADC through DIRCR and directly read ADC result value through DIRDATA register.
5:4	-	Reserved
3:2	R/W	Select ADC conversion wait time. 00: no wait (read after 16 cycles, default wait time) 01: 2 clock wait (read after 18 cycles) 10: 4 clock wait (read after 20 cycles) 11: read ADC data on negative edge of ADEND signal
1	R/W	Self Operate Power down bit. When this bit is set, AIOSTOP bit will strobe high when no ADC functions are enabled.
0	R/W	Long calibration time. The default ADC calibration time is 10 ms but when needed ADC can be calibrated during 40ms with this bit. Short calibration time need 96 cycles of 8 kHz OCLK or 128 of 11 kHz OCLK and the long time need 384 cycles of 8 kHz or 512 cycles of 11 kHz OCLK. OCLK is determined from SRATE bit of ADCSDCR.

10.1.2.2ADC Touch Panel Control Register (ADCTPCR)

This register control functions related with touch panel interface.

7	6	5	4	3	2	1	0
TPEN	TINTMSK	SWBYPSS	SWINVT	INTTDEN	SSHOT	TRATE	

Bits	Type	Function
7	R/W	Touch panel read enable bit. Write “1” to enable touch panel function.
6	R/W	Touch panel read interrupt mask bit. Write “1” to enable touch panel interrupt.
5	R/W	Touch panel drive signal bypass bit. When external touch panel drive circuit used, this bit disable internal drive circuit and bypass switching signal to external pins SW_XP, SW_XN, SW_YP and SW_YN.
4	R/W	Touch panel drive signal inversion bit. for flexibility
3	R/W	Internal tip-down detection logic enable bit. Write “1” to enable this function

2	R/W	Single touch panel read operation. Normally, touch panel date read twice. But this bit is set, touch panel data read once for a point and save power to read touch panel.
1:0	R/W	Select touch panel date sampling rate. It depends on basic operation clock of ADC interface(sound sampling rate). 11: 400 or 550 samples / sec 10: 200 or 275 samples / sec 01: 100 or 138 samples / sec 00: 50 or 69 samples / sec

10.1.2.3ADC Battery check Control Register (ADCBACR)

This register controls battery check operation.

7	6			3	2		
MBEN	MINTMSK			BBEN	BINTMSK		

Bits	Type	Function
7	R/W	Main battery check enable Write “1” to enable Four 8-bit battery check data recorded in ADCMBDATA register
6	R/W	Main battery check interrupt mask bit Write “1” to enable
5:4	-	Reserved
3	R/W	Backup battery check enable Write “1” to enable Four 8-bit battery check data recorded in ADCBBDATA register
2	R/W	Backup battery check interrupt mask bit Write “1” to enable
1:0	-	Reserved

10.1.2.4ADC Sound Control Register (ADCSDCR)

This register controls sound sampling function. SRATE bit control base clock of ADC interface logic.

7	6						0
SNDEN	SINTMSK						SRATE

Bits	Type	Function
7	R/W	Sound date capture enable bit Write “1” to enable
6	R/W	Sound date interrupt mask bit Write “1” to enable
5:1	-	Reserved
0	R/W	Sound date sampling rate selection bit. This bit affects to all sampling rates of touch panel and battery operations. 0: 8 kHz sound sampling 1: 11.025 kHz sound sampling

10.1.2.5ADC Interrupt Status Register (ADCISR)

Read only valid but write “1” to clear all interrupt value

7	6	5	4			1	0
INTTP	INTMB	INTBB	INTSD			INTTD	INTTU

Bits	Type	Function
7	R/W	Touch panel data interrupt. Write “1” here to clear this interrupt.
6	R/W	Main battery check interrupt. Write “1” here to clear this interrupt.

5	R/W	Backup battery check interrupt. Write “1” to clear this interrupt.
4	R/W	Sound data interrupt. Write “1” here to clear this interrupt.
3:2	-	Reserved
1	R/W	Tip Down interrupt. Write “1” here to clear this interrupt.
0	R/W	Tip Up interrupt. Write “1” here to clear this interrupt.

10.1.2.6 ADC Tip Down Control Status Register (ADCTDCSR)

7	6	5	4	3		1	0
TDEN	TDMSK	TUEN	TUMSK	TPSEL		TP_X	TP_Y

Bits	Type	Function
7	R/W	Touch panel tip-down detection logic enable Write “1” to enable this function
6	R/W	Touch panel tip-down interrupt mask bit Write “1” to enable interrupt
5	R/W	Touch panel tip-up detection enable. When this bit is set, once in every 20 OCLK cycles, monitor touch panel status periodically.
4	R/W	Touch panel tip-up interrupt mask bit.
3	R/W	Select Tip Down/Up monitoring channel (0:X, 1:Y)
2	-	Reserved
1	R/W	X axis Tip status monitor bit (read only bit)
0	R/W	Y axis Tip status monitor bit (read only bit)

10.1.2.7ADC Direct Control Register (ADCDIRCR)

7	6	5	4	3	2	1	0
AIOSTOP			ACH				

Bits	Type	Function
7	R/W	AIOSTOP bit value to access ADC directly
6:5	-	Reserved
4:0	R/W	ADC channel selection bits to control ADC directly 00001: select channel 0 (touch panel X) 00010: select channel 1 (touch panel Y) 00100: select channel 2 (Main battery) 01000: select channel 3 (Backup battery) 10000: select channel 4 (Sound input)

10.1.2.8ADC Direct Data Read Register (ADCDIRDATA)

Register can be used to read data from ADC.

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

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

10.1.2.9ADC 1ST Touch Panel Data register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
XDATA1: ADCTPXDR0, XDATA3: ADCTPXDR1 YDATA1: ADCTPYDR0, YDATA3: ADCTPYDR1															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
XDATA0: ADCTPXDR0, XDATA2: ADCTPXDR1 YDATA0: ADCTPYDR0, YDATA2: ADCTPYDR1															

ADCTPXDR0:

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:

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:

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:

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)

10.1.2.10 ADC 2ND Touch Panel Data Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
XDATA5: ADCTPXDR2, XDATA7: ADCTPXDR3 YDATA5: ADCTPYDR2, YDATA7: ADCTPYDR3															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
XDATA5: ADCTPXDR2, XDATA6: ADCTPXDR3 YDATA5: ADCTPYDR2, YDATA6: ADCTPYDR3															

ADCTPXDR2:

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:

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:

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:

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)

10.1.2.11 ADC Main Battery Data Register (ADCMBDATA)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MBDATA3								MBDATA2							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MBDATA1								MBDATA0							

Bits	Type	Function
31:24	R/W	Forth main battery check data
23:16	R/W	Third main battery check data
15:8	R/W	Second main battery check data
7:0	R/W	First main battery check data

10.1.2.12 ADC Backup Battery Data Register (ADCBBDATA)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BBDATA3								BBDATA2							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BBDATA1								BBDATA0							

Bits	Type	Function
31:24	R/W	Forth backup battery check data
23:16	R/W	Third backup battery check data
15:8	R/W	Second backup battery check data
7:0	R/W	First backup battery check data

10.1.2.13 ADC Sound Data Register (ADCSDATA0 – ADCSDATA7)

HMS30C7202 has 8-word size sound register so it can contain 32 8-bit sound data.

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

Bits	Type	Function
31:24	R/W	$(n+3)^{TH}$ Sound Data. (n = ADCSDATAn)
23:16	R/W	$(n+2)^{TH}$ Sound Data. (n = ADCSDATAn)
15:8	R/W	$(n+1)^{TH}$ Sound Data. (n = ADCSDATAn)

7:0	R/W	$(n)^{TH}$ Sound Data. (n = ADCSDATAn)

10.2 CAN Interface

Confidential.

10.3 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).

10.3.1 External Signals

Pin Name	Type	Description
KBDI [7:0]	I/O	GPIO PORTA [15:8]
KBDO [7:0]	I/O	GPIO PORTA [7:0]
PORTB [11:6]	I/O	GPIO PORTB [11:6]
nUDCD	I/O	GPIO PORTB [5]
nUDSR	I/O	GPIO PORTB [4]
nURTS	I/O	GPIO PORTB [3]
nUCTS	I/O	GPIO PORTB [2]
nUDTR	I/O	GPIO PORTB [1]
nURING	I/O	GPIO PORTB [0]
nRCS3	I/O	GPIO PORTC [10]
nRCS2	I/O	GPIO PORTC [9]
DMAA	I/O	GPIO PORTC [8]
DMAR	I/O	GPIO PORTC [7]
PWM1	I/O	GPIO PORTC [6]
PWM0	I/O	GPIO PORTC [5]
PS2CK	I/O	GPIO PORTC [4]
PS2D	I/O	GPIO PORTC [3]
CANRx0	I/O	GPIO PORTC [2]
CANTx0	I/O	GPIO PORTC [1]
TimerOut	I/O	GPIO PORTC [0]
LBLEN	I/O	GPIO PORTD [8]
LD [15:8]	I/O	GPIO PORTD [7:0]
RA [24]	I/O	GPIO PORTE [24]
CANTx1	I/O	GPIO PORTE [23]
CANRx1	I/O	GPIO PORTE [22]
MMCCLK	I/O	GPIO PORTE [21]
MMCCD	I/O	GPIO PORTE [20]
MMCDAT	I/O	GPIO PORTE [19]
MMCCMD	I/O	GPIO PORTE [18]
NRW3	I/O	GPIO PORTE [17]
NRW2	I/O	GPIO PORTE [16]
RD [31:16]	I/O	GPIO PORTE [15:0]

10.3.2 Registers

Address	Name	Width	Default	Description
0x8002.3000	ADATA	16	0x0000	GPIO PORTA Data register
0x8002.3004	ADIR	16	0xFFFF	GPIO PORTA Data Direction register
0x8002.3008	AMASK	16		GPIO PORTA Interrupt Mask register
0x8002.300C	ASTAT	16		GPIO PORTA Interrupt Status register
0x8002.3010	AEDGE	16		GPIO PORTA Edge Mode register
0x8002.3014	ACLR	16		GPIO PORTA Clear register
0x8002.3018	APOL	16		GPIO PORTA Polarity register
0x8002.301C	AMUXCTL	16		GPIO PORTA Multi-function Pin Select register
0x8002.3020	BDATA	12		GPIO PORTB Data register
0x8002.3024	BDIR	12		GPIO PORTB Data Direction register
0x8002.3028	BMASK	12		GPIO PORTB Interrupt Mask register
0x8002.302C	BSTAT	12		GPIO PORTB Interrupt Status register
0x8002.3030	BEDGE	12		GPIO PORTB Edge Moderegister

0x8002.3034	BCLR	12	GPIO PORTB Clear register
0x8002.3038	BPOL	12	GPIO PORTB Polarity register
0x8002.303C	BMUXCTL	12	GPIO PORTB Multi-function Pin Select register
0x8002.3040	CDATA	11	GPIO PORTC Data register
0x8002.3044	CADIR	11	GPIO PORTC Data Direction register
0x8002.3048	CMASK	11	GPIO PORTC Interrupt Mask register
0x8002.304C	CSTAT	11	GPIO PORTC Interrupt Status register
0x8002.3050	CEDGE	11	GPIO PORTC Edge Mode register
0x8002.3054	CCLR	11	GPIO PORTC Clear register
0x8002.3058	CPOL	11	GPIO PORTC Polarity register
0x8002.305C	CMUXCTL	11	GPIO PORTC Multi-function Pin Select register
0x8002.3060	DDATA	9	GPIO PORTD Data register
0x8002.3064	DDIR	9	GPIO PORTD Data Direction register
0x8002.3068	DMASK	9	GPIO PORTD Interrupt Mask register
0x8002.306C	DSTAT	9	GPIO PORTD Interrupt Status register
0x8002.3070	DEDGE	9	GPIO PORTD Edge Mode register
0x8002.3074	DCLR	9	GPIO PORTD Clear register
0x8002.3078	DPOL	9	GPIO PORTD Polarity register
0x8002.307C	DMUXCTL	9	GPIO PORTD Multi-function Pin Select register
0x8002.3080	EDATA	25	GPIO PORTE Data register
0x8002.3084	EDIR	25	GPIO PORTE Data Direction register
0x8002.3088	EMASK	25	GPIO PORTE Interrupt Mask register
0x8002.308C	ESTAT	25	GPIO PORTE Interrupt Status register
0x8002.3090	EEDGE	25	GPIO PORTE Edge Mode register
0x8002.3094	ECLR	25	GPIO PORTE Clear register
0x8002.3098	EPOL	25	GPIO PORTE Polarity register
0x8002.309C	EMUXCTL	25	GPIO PORTE Multi-function Pin Select register

10.3.2.1[A,B,C,D,E]DATA

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AMUXCTL [31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADATA, ADIR, AMASK, ASTAT, AEDGE, ACLR, APOL, AMUXCTL [15:0]															

11	10	9	8	7	6	5	4	3	2	1	0
BDATA, BDIR, BMASK, BSTAT, BEDGE, BCLR, BPOL, BMUXCTL											

10	9	8	7	6	5	4	3	2	1	0
CDATA, CDIR, CMASK, CSTAT, CEDGE, CCLR, CPOL, CMUXCTL										

8	7	6	5	4	3	2	1	0
DDATA, DDIR, DMASK, DSTAT, DEDGE, DCLR, DPOL, DMUXCTL								

							24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EDATA, EDIR, EMASK, ESTAT, EEDGE, ECLR, EPOL, E MUXCTL [24:0]															

Bits	Type	Function
16/ 12/ 11/ 9/ 25	R/W	Values written to this register will be output on port [A,B,C,D,E] pins if the corresponding data direction bits are set Low (port output). Values read from this register reflect the external state of port [A,B,C,D,E] not necessarily the value written to it. All bits are cleared by a system reset. When the PIO pin is defined as input, this input can be an interrupt source with register setting. On reads, the Data Register contains the current status of correspondent port pins, whether they are configured as input or output. Writing to a Data Register only affects the pins that are configured as outputs. All PIO input pins can be used as interrupt source with enabled

	interrupt mask register bit. These interrupt sources can be selected as active HIGH/LOW, EDGE/LEVEL trigger mode.

10.3.2.2[A,B,C,D,E]DIR

Bits	Type	Function
16/ 12/ 11/ 9/ 25	R/W	Bits set in this register will select the corresponding pin in port [A,B,C,D,E] to become an input, clearing a bit sets the pin to output. All bits are set by a system reset.

10.3.2.3[A,B,C,D,E]MASK

Bits	Type	Function
16/ 12/ 11/ 9/ 25	R/W	Bits set in this register will select the corresponding pin to become an interrupt source. All bits are cleared by a system reset. 0 = disable interrupt (default) 1 = enable interrupt

10.3.2.4[A,B,C,D,E]STAT

Bits	Type	Function
16/ 12/ 11/ 9/ 25	R/W	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. Interrupt controller receives active HIGH, level mode interrupt sources only. But PIO block can receive not only active HIGH or active LOW, but also level or edge mode signals. Then interprets and sends interrupt request to the interrupt controller. All bits can be controlled separately. Values in this 8-bit read-only register represents that the interrupt requests are pending on corresponding pins. All bits are cleared by a system reset. 0 = no interrupt request 1 = interrupt pending (masked interrupt is always 0)

10.3.2.5[A,B,C,D,E]EDGE

Bits	Type	Function
16/ 12/ 11/ 9/ 25	R/W	Bits set in this 8-bit read/write register will select the corresponding pin to become an edge mode interrupt source. All bits are cleared by a system reset. 0 = level mode (default) = edge mode

10.3.2.6[A,B,C,D,E]CLR

Bits	Type	Function
16/ 12/ 11/ 9/ 25	R/W	Bits set in this 8-bit write-only register will clear the stored interrupt request of corresponding bit in edge mode. All bits are automatically cleared after written. 0 = no action (default) 1 = clear interrupt source (self reset)

10.3.2.7[A,B,C,D,E]POL

Bits	Type	Function
16/ 12/ 11/	R/W	Bits set in this 8-bit read/write register will select the corresponding pin to become an active LOW mode interrupt source. All bits are cleared by a system reset. After accessing this register, the Edge Mode register should be cleared with the Clear register.

9/ 25	0 = active HIGH mode 1 = active LOW mode

10.3.2.8[A,B,C,D,E]MUXCTL

Bits	Type	Function
32/ 12/ 11/ 9/ 25	R/W	Bits set in this register will select the corresponding pin to become a PIO pin except PORTA. PORTA is configured as following table **n** **AMUXCTL [n] = 0** **AMUXCTL [n+16] = 0** **AMUXCTL [n] = 0** **AMUXCTL [n+16] = 1** **AMUXCTL [n] = 1** **AMUXCTL [n+16] = x** 15 KBDI [7] IRDI [1] GPIOA [15] 14 KBDI [6] USO [3] GPIOA [14] 13 KBDI [5] USI [3] GPIOA [13] 12 KBDI [4] ISECK GPIOA [12] 11 KBDI [3] ISWS GPIOA [11] 10 KBDI [2] KBDI [2] GPIOA [10] 9 KBDI [1] KBDI [1] GPIOA [9] 8 KBDI [0] KBDI [0] GPIOA [8] 7 KBDO [7] IRDO [1]

GPIOA [7]

6
KBDO [6]
USO [2]
GPIOA [6]

5
KBDO [5]
USI [2]
GPIOA [5]

4
KBDO [4]
ISCLK
GPIOA [4]

3
KBDO [3]
ISD
GPIOA [3]

2
KBDO [2]
KBDO [2]
GPIOA [2]

1
KBDO [1]
KBDO [1]
GPIOA [1]

0
KBDO [0]
KBDO [0]
GPIOA [0]

PORTE has following configuration.

n
EMUXCTL [n] = 0
BOOTBIT [1:0] = 00
EMUXCTL [n] = 0
BOOTBIT [1:0] ≠ 00
EMUXCTL [n] = 1

SPIMMC enable
SPIMMC disable

24
RA [24]
0

GPIOE [24]

23
CANTx [1]
0

GPIOE [23]

22
CANRx [1]
nSMRB

GPIOE [22]

21
MMCCLK
SSCLK
SMCLE
GPIOE [21]

20
MMCCD
nSSCS
nSMCD
GPIOE [20]

19
MMCDAT
SSDO
nSMCE
GPIOE [19]

18
MMCCMD
SSDI
nSMRE
GPIOE [18]

17
nRW [3]
SMALE

GPIOE [17]

16
nRW [2]
SMWE

GPIOE [16]

15
RD [31]
SMWP

GPIOE [15]

14
RD [30]
SMD [0]

GPIOE [14]

13
RD [29]
SMD [1]

GPIOE [13]

12
RD [28]
SMD [2]

GPIOE [12]

11
RD [27]
SMD [3]

GPIOE [11]

10
RD [26]
SMD [4]

GPIOE [10]

9
RD [25]
SMD [5]

GPIOE [9]

8
RD [24]
SMD [6]

GPIOE [8]

7
RD [23]
SMD [7]

GPIOE [7]

6
RD [22]
UVP

GPIOE [6]

5
RD [21]
UVM

GPIOE [5]

4
RD [20]
URCVIN

GPIOE [4]

3
RD [19]
USUSPEND

GPIOE [3]

2

RD [18]
UVMO

GPIOE [2]

1
RD [17]
UVPO

GPIOE [1]

0
RD [16]
nUSBOE

GPIOE [0]

10.4 Interrupt Controller

The GMS30C7202 has a fully programmable priority, individually maskable, vectored interrupt controller. This feature reduces the software overhead in handling interrupts. The Interrupt controller provides the ARM720T processor with the Fast interrupt request (NFIQ) and the standard interrupt request (NIRQ) inputs. The NFIQ and the NIRQ are selected out of all interrupt sources, which can be asserted by the interrupt generated by the on-chip peripherals and the external GPIO lines. The fully programmable priority encoder allows the user to define the full priority between the different NIRQ interrupt sources. Internal interrupt sources should be high level sensitive. External interrupt sources can be positive or negative edge triggered or high or low level sensitive, which are controlled by GPIO Interface.

ID Code	Interrupt Source	ID Code	Interrupt Source
00	PMU	10	Timer1 or Timer2 or Timer3
01	DMA	11	Watchdog
02	LCD	12	CAN0
03	Sound	13	CAN1
04	I2S	14	GPIOB0 (GPIOB [10])
05	USB	15	GPIOB1 (GPIOB [11])
06	MMC or SmartMedia Card	16	GPIOA
07	RTC	17	GPIOB
08	UART0	18	GPIOC
09	UART1	19	GPIOD
0A	UART2	1A	GPIOE
0B	UART3	1B	ARM core (COMMRX debug only)
0C	KBD (KeyBoard Interface)	1C	ARM core (COMMTX debug only)
0D	PS2	1D	Reserved
0E	AIC	1E	Software (auto generation by CPU register set)
0F	Timer0		

Table 10-2 Interrupt controller Configuration

Note *Software Interrupt is always set high (active). If you want to use the Software Interrupt, you should set the corresponding bit of IER (IRQ Enable Register).*

10.4.1 Block diagram

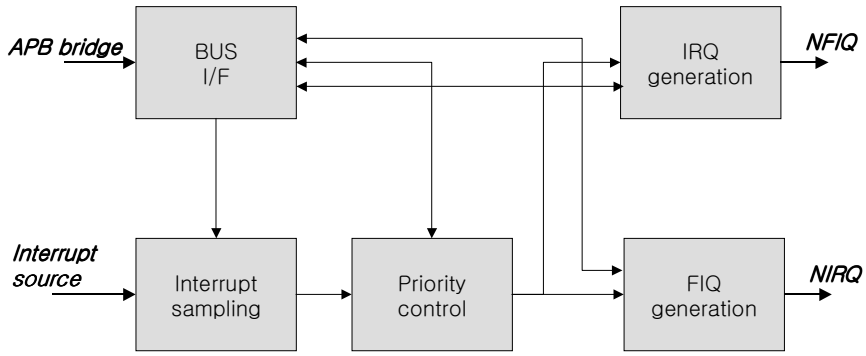


Figure 10-1 Interrupt controller block diagram

10.4.2 Registers

Address	Name	Width	Default	Description
0x8002.4000	IER	31	0x00000000	Interrupt enable register
0x8002.4004	ISR	31	0x00000000	Interrupt status register
0x8002.4008	IVR	32	0x00000000	IRQ vector register

0x8002.4010	SVR0	32	0x00000000	Source vector register 0
0x8002.4014	SVR1	32	0x00000000	Source vector register 1
0x8002.4018	SVR2	32	0x00000000	Source vector register 2
0x8002.401C	SVR3	32	0x00000000	Source vector register 3
0x8002.4020	SVR4	32	0x00000000	Source vector register 4
0x8002.4024	SVR5	32	0x00000000	Source vector register 5
0x8002.4028	SVR6	32	0x00000000	Source vector register 6
0x8002.402C	SVR7	32	0x00000000	Source vector register 7
0x8002.4030	SVR8	32	0x00000000	Source vector register 8
0x8002.4034	SVR9	32	0x00000000	Source vector register 9
0x8002.4038	SVR10	32	0x00000000	Source vector register 10
0x8002.403C	SVR11	32	0x00000000	Source vector register 11
0x8002.4040	SVR12	32	0x00000000	Source vector register 12
0x8002.4044	SVR13	32	0x00000000	Source vector register 13
0x8002.4048	SVR14	32	0x00000000	Source vector register 14
0x8002.404C	SVR15	32	0x00000000	Source vector register 15
0x8002.4050	SVR16	32	0x00000000	Source vector register 16
0x8002.4054	SVR17	32	0x00000000	Source vector register 17
0x8002.4058	SVR18	32	0x00000000	Source vector register 18
0x8002.405C	SVR19	32	0x00000000	Source vector register 19
0x8002.4060	SVR20	32	0x00000000	Source vector register 20
0x8002.4064	SVR21	32	0x00000000	Source vector register 21
0x8002.4068	SVR22	32	0x00000000	Source vector register 22
0x8002.406C	SVR23	32	0x00000000	Source vector register 23
0x8002.4070	SVR24	32	0x00000000	Source vector register 24
0x8002.4074	SVR25	32	0x00000000	Source vector register 25
0x8002.4078	SVR26	32	0x00000000	Source vector register 26
0x8002.407C	SVR27	32	0x00000000	Source vector register 27
0x8002.4080	SVR28	32	0x00000000	Source vector register 28
0x8002.4084	SVR29	32	0x00000000	Source vector register 29
0x8002.4088	SVR30	32	0x00000000	Source vector register 30
0x8002.4090	IDR	32	0x00001F1F	Interrupt ID register
0x8002.4094	PSR0	32	0x03020100	Priority set register 0
0x8002.4098	PSR1	32	0x07060504	Priority set register 1
0x8002.409C	PSR2	32	0x0B0A0908	Priority set register 2
0x8002.40A0	PSR3	32	0x0F0E0D0C	Priority set register 3
0x8002.40A4	PSR4	32	0x13121110	Priority set register 4
0x8002.40A8	PSR5	32	0x17161514	Priority set register 5
0x8002.40AC	PSR6	32	0x1B1A1918	Priority set register 6
0x8002.40B0	PSR7	32	0x001E1D1C	Priority set register 7
0x8002.40B4	TER	1	0x0	Test enable register
0x8002.40B8	TIR	31	0X00000000	Test interrupt register

Table 10-3 Interrupt controller Register Summary

10.4.2.1Interrupt Enable Register (IER)

The enable register is used to determine whether or not an active interrupt source should generate an interrupt request to the processor.

Bits	Type	Function
31	R	Reserved
30	R/W	Software Interrupt
29	R	Reserved
28	R/W	ARM core (COMMTX: debug only)
27	R/W	ARM core (COMMRX: debug only)
26	R/W	GPIO port E
25	R/W	GPIO port D
24	R/W	GPIO port C

23	R/W	GPIO port B
22	R/W	GPIO port A
21	R/W	External Interrupt1 (GPIOB[11])
20	R/W	External Interrupt0 (GPIOB[10])
19	R/W	CAN1
18	R/W	CAN0
17	R/W	Watchdog timer
16	R/W	Timer1 or Timer2 or Timer3
15	R/W	Timer0
14	R/W	AIC
13	R/W	PS2
12	R/W	KBD (keyboard interface)
11	R/W	UART3
10	R/W	UART2
9	R/W	UART1
8	R/W	UART0
7	R/W	RTC
6	R/W	MMC or Smart Media Card
5	R/W	USB
4	R/W	I2S
3	R/W	Sound
2	R/W	LCD
1	R/W	DMA
0	R/W	PMU

Note
0: Disable interrupt
1: Enable interrupt

10.4.2.2Interrupt Status Register (ISR)

The IRQ Status register indicates whether or not the interrupt source is causing a processor IRQ interrupt.

Bits	Type	Function
31	R	Reserved
30	R/W	Software Interrupt
29	R	Reserved
28	R/W	ARM core (COMMTX: debug only)
27	R/W	ARM core (COMMRX: debug only)
26	R/W	GPIO port E
25	R/W	GPIO port D
24	R/W	GPIO port C
23	R/W	GPIO port B
22	R/W	GPIO port A
21	R/W	External Interrupt1 (GPIOB[11])
20	R/W	External Interrupt0 (GPIOB[10])
19	R/W	CAN1
18	R/W	CAN0
17	R/W	Watchdog timer
16	R/W	Timer1 or Timer2 or Timer3
15	R/W	Timer0
14	R/W	AIC
13	R/W	PS2
12	R/W	KBD (keyboard interface)
11	R/W	UART3
10	R/W	UART2
9	R/W	UART1
8	R/W	UART0
7	R/W	RTC
6	R/W	MMC or Smart Media Card
5	R/W	USB

4	R/W	I2S
3	R/W	Sound
2	R/W	LCD
1	R/W	DMA
0	R/W	PMU

Note

0: No interrupt request

1: Interrupt pending (If interrupt source don't enable, the bit always "0")

10.4.2.3IRQ Vector Register (IVR)

31	...	0
IVR		

Bits	Type	Function
31:0	R	The IRQ Vectored Register contains the vector programmed by the user in the Source Vector Register corresponding to the current interrupt. The Source Vector Register (0 to 31) is indexed using the ID number in the current interrupt ID register when the IRQ Vector Register is read. When there is no IRQ status, the IRQ Vector Register is set to 0.

10.4.2.4Source Vector Register (SVR0 to SVR30)

31	...	0
IVR		

Bits	Type	Function
31:0	R	The user may store in these registers the address of the corresponding handler for each interrupt source. This interrupt controller has 31-Source Vector Registers, which are corresponded to ID code. For example the Source Vector Register of the Interrupt by RTC is the SVR7 (Source Vector Register 7)

10.4.2.5Interrupt ID Register (IDR)

The Interrupt ID Register returns the current FIQ and IRQ interrupt source number.

31 – 13	12 - 8	7 – 5	4 - 0
Reserved	FIQID	Reserved	IRQID

Bits	Type	Function
31:13	R	Reserved
12:8	R	FIQID
7:5	R	Reserved
4:0	R	IRQID

10.4.2.6Priority Set Register (PSR0 to PSR7)

The Priority Set Registers consist of 8 registers. The 8 bits are assigned to each interrupt and its value is ID code of corresponding interrupt source (See table 1-1). The FIQ interrupt source is defined by the ID code number in PSR1[7:0] of Priority Set Register 1. If an user sets bit[7:0] of PSR0 to be 0x09, FIQ interrupt is the UART 1 interrupt . If an user sets bit[15:8] of PSR3 to be 0x04, IRQ interrupt is the I2S interrupt and IVR(IRQ Vector Register) is the SVR4(Source Vector Register 4).

31 – 24	23 – 16	15 – 8	7 – 0
IRQ priority *	IRQ priority *	IRQ priority *	IRQ priority *

Register	Bits	Type	Initial ID value	Function
PSR7	31:24	R	0x00	Reserved

	12:8	R/W	0x1E	IRQ priority 1E
	7:5	R/W	0x1D	IRQ priority 1D (reserved)
	4:0	R/W	0x1C	IRQ priority 1C
PSR6	31:24	R/W	0x1B	IRQ priority 1B
	12:8	R/W	0x1A	IRQ priority 1A
	7:5	R/W	0x19	IRQ priority 19
	4:0	R/W	0x18	IRQ priority 18
PSR5	31:24	R/W	0x17	IRQ priority 17
	12:8	R/W	0x16	IRQ priority 16
	7:5	R/W	0x15	IRQ priority 15
	4:0	R/W	0x14	IRQ priority 14
PSR4	31:24	R/W	0x13	IRQ priority 13
	12:8	R/W	0x12	IRQ priority 12
	7:5	R/W	0x11	IRQ priority 11
	4:0	R/W	0x10	IRQ priority 10
PSR3	31:24	R/W	0x0F	IRQ priority F
	12:8	R/W	0x0E	IRQ priority E
	7:5	R/W	0x0D	IRQ priority D
	4:0	R/W	0x0C	IRQ priority C
PSR2	31:24	R/W	0x0B	IRQ priority B
	12:8	R/W	0x0A	IRQ priority A
	7:5	R/W	0x09	IRQ priority 9
	4:0	R/W	0x08	IRQ priority 8
PSR1	31:24	R/W	0x07	IRQ priority 7
	12:8	R/W	0x06	IRQ priority 6
	7:5	R/W	0x05	IRQ priority 5
	4:0	R/W	0x04	IRQ priority 4
PSR0	31:24	R/W	0x03	IRQ priority 3
	12:8	R/W	0x02	IRQ priority 2
	7:5	R/W	0x01	IRQ priority 1
	4:0	R/W	0x00	***FIQ source***

Note

The Priority Level is to be defined as follows.

IRQ Priority 1 > IRQ Priority 2 > . . . > IRQ Priority 1D> IRQ Priority 1E

10.4.2.7Test Enable register (TER)

-	0
Reserved	TER

Bits	Type	Function
0	R/W	Two extra registers are defined to facilitate testing of the interrupt controller module using the AMBA test methodology: This register can control the interrupt controller in Test Interface mode. The Bit 0 of this register controls whether can enable interrupts sources of Test Interrupt Register or not. If the bit 0 is enabled to “1”, Interrupt sources can be selected by Test Interrupt Register.

10.4.2.8Test interrupt register (TIR)

This function of the interrupt controller is available for auto-test or software debug purpose. All interrupt sources which are programmed to be high or low can be individually set or cleared by respectively writing to the register TIR.

Bits	Type	Function
31	R	Reserved
30	R/W	Software Interrupt
29	R	Reserved
28	R/W	ARM core (COMMTX: debug only)
27	R/W	ARM core (COMMRX: debug only)

26	R/W	GPIO port E
25	R/W	GPIO port D
24	R/W	GPIO port C
23	R/W	GPIO port B
22	R/W	GPIO port A
21	R/W	External Interrupt1 (GPIOB[11])
20	R/W	External Interrupt0 (GPIOB[10])
19	R/W	CAN1
18	R/W	CAN0
17	R/W	Watchdog timer
16	R/W	Timer1 or Timer2 or Timer3
15	R/W	Timer0
14	R/W	AIC
13	R/W	PS2
12	R/W	KBD (keyboard interface)
11	R/W	UART3
10	R/W	UART2
9	R/W	UART1
8	R/W	UART0
7	R/W	RTC
6	R/W	MMC or Smart Media Card
5	R/W	USB
4	R/W	I2S
3	R/W	Sound
2	R/W	LCD
1	R/W	DMA
0	R/W	PMU

10.5 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. The keyboard interface uses the pins KSCANI [7:0], KSCANO [7:0]. It is possible to select one of four scan clock modes.

FEATURES

- Four scanning modes
- 8x8 Matrix
- Byte key buffers

10.5.1 External Signals

Pin Name	Type	Description
KSCANO [7:0]	O	This assigns the x-axis' scan line. The value is changed periodically so as to cover every key matrix. During one keyboard scan, KSCANO [7:0] can have 8 different values. Active LOW signal.
KSCANI [7:0]	I	This indicates which key is pressed in the assigned scan line. Active LOW signal

10.5.2 Registers

Address	Name	Width	Default	Description
0x8002.2000	KBCR	8	0x0	Keyboard Configuration Register
0x8002.200C	KBVR0	32	0x0	Keyboard value register 0
0x8002.2010	KBVR1	32	0x0	Keyboard value register 1
0x8002.2018	KBSR	1	0x0	Keyboard status register

Table 10-4 Matrix Keyboard Interface Controller Register Summary

10.5.2.1 Keyboard Configuration Register (KBCR)

7					2	1	0
SCAN ENABLE					POWER DOWN	CLK SEL	

Bits	Type	Function
7	R/W	SCANENABLE bit. This starts or stops matrix keyboard scanning. To start keyboard input scanning, set the SCANENABLE bit and POWERDOWN bit of KBCR (Keyboard Configuration Register) and the CLK SEL bit of the KBCR. The key scan control signal is generated. Periodically, column scan code is saved in the 8byte key buffer. After the 8th column key data is stored, keyboard interrupt is generated to make the CPU read 8 scan values. The SCANENABLE bit and POWERDOWN bit are usually set or reset simultaneously When all the column of keyboard has been scanned, an interrupt is generated, and, by interrogating the KBVR registers, software can determine which keys have been pressed. It is software's responsibility to debounce the key pressed information. Keyboard key press interrupts are generated in all PMU states except deep sleep. Start and stop scanning 0 = stop 1 = start
6:3	-	Reserved. Keep these bits to zero.
2	R/W	POWERDOWN bit. In the power down mode, no clock is inputted to this controller logic. 0 = power down mode, where clock is not operating 1 = normal mode, where clock is operating
1:0	R/W	CLKSEL bit. This controls the operating clock of scanning matrix keyboard. Base Scanning clock is generated using PCLK (3.6864MHz). **Value**

Base Scanning Clock Rate
Scan Rate (8byte column buffer)

00
PCLK/2 (1.84MHz, test mode only)
8861 times/sec

01
PCLK/128 (28KHz)
138 times/sec

10
PCLK/256 (14KHz)
69 times/sec

11
PCLK/512 (7KHz)
34 times/sec

10.5.2.2Keyboard Value Register (KVR0)

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

Bits	Type	Function
31:24	R	1st column matrix keyboard scan input data. For example, if the value of KBVR0[32:24] is 00001100, the 5th and 6th keys are pressed and the others are released in 1st column.
23:16	R	2nd column matrix keyboard scan input data
15:8	R	3rd column matrix keyboard scan input data
7:0	R	4th column matrix keyboard scan input data

10.5.2.3Keyboard Value Register (KVR1)

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

Bits	Type	Function
31:24	R	5th column matrix keyboard scan input data
23:16	R	6th column matrix keyboard scan input data
15:8	R	7th column matrix keyboard scan input data
7:0	R	8th column matrix keyboard scan input data

10.5.2.4Keyboard Status Register (KBSR)

						1	0
						WAKEUP	INTR

Bits	Type	Function
7:2	-	Reserved
1	R	The interrupt and the KBSR bit are cleared after the CPU reads KBSR. The KBSR bit is set when the key buffer is full, or when the key is pressed in powerdown mode (keyboard disabled). Wake up state:

		0 = no key pressed in powerdown mode 1 = key pressed in powerdown mode
0	R	Key bufferstate: 0 = key buffer is not full 1 = key buffer is full

10.6 PS/2 Interface Controller

This PS/2 Controller is an Advanced Microcontroller Bus Architecture (AMBA) compliant System-on-a-Chip peripheral providing industry-standard PS/2 data transfer channel. A channel has two bi-directional signals that serve as direct interfaces to an external keyboard, mouse or any other PS/2-compatible pointing device. 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).

FEATURES

- AMBA compliant
- PS/2 compatible interface
- Half-duplex bi-directional synchronous serial interface using open-drain outputs for clock and data
- Enable/Disable channel
- Operation in polled or interrupt-driven mode
- Hardware support for PS/2 auxiliary device protocol
- Maskable transmit and receive interrupts
- Automatic odd parity generation and checking
- Optional software based PS/2 implementation
- Test Interface Controller compatible test registers and test modes

10.6.1 External Signals

Pin Name	Type	Description
PSCLK	I/O	PS/2 compatible clock signal pin. Pull-up this pad output (open-drain pad used.)
PSDAT	I/O	PS/2 compatible data signal pin. Also pull-up this pad (open-drain).

10.6.2 Registers

Address	Name	Width	Default	Description
0x8002.C000	PSDATA	8	00h	Transmit/Receive data register
0x8002.C004	PSSTAT	7	00h	Internal status register
0x8002.C008	PSCONF	6	00h	Configuration register
0x8002.C00C	PSINTR	5	00h	Interrupt/Error status and Interrupt ACK register
0x8002.C010	PSTDLO	8	00h	Timing parameter register
0x8002.C014	PSTPRI	8	00h	Timing parameter register
0x8002.C018	PSTXMT	8	00h	Timing parameter register
0x8002.C01C	-	-	-	Reserved for test purpose
0x8002.C020	PSTREC	8	00h	Timing parameter register
0x8002.C024	-	-	-	Reserved for test purpose
0x8002.C03C	PSPWDN	1	00h	Power-down configuration register

Table 10-5 PS/2 Controller Register Summary

NOTE: The initial value of registers may be not correct with the condition of testing environment. Above values are based on TIC test environment. With external model, some registers may have different value.

10.6.2.1 PSDATA

7	6	5	4	3	2	1	0
Transmit / Receive Data							

Bits	Type	Function
7:0	R/W	After wake up, PS/2 interface waits for one of two events: 1. If data is written to the PSDATA register, a transmit sequence is initiated and the data is transmitted serially. 2. If data signal is pulled low by the external devices and clock signal's negative edge is detected, a receive sequence begins and data is clocked into PSDATA register. At the end of transmission, transmit interrupt will occur. By reading PSSTAT status register will reveal the data is transmitted properly. Reading PSSTAT also de-asserts transmit interrupt request.

	PS/2 controller usually remains in receive data mode if no data is transmitting. The controller automatically receives data from external device and generates receive interrupt. By just reading PSDATA register the data will be acquired and the receive interrupt will be cleared.

10.6.2.2PSSTAT

	6	5	4	3	2	1	0
	PARITY	DATA IN	CLK IN	RX BUSY	RX FULL	TX BUSY	TX EMPTY

Bits	Type	Function
7	-	Reserved. Always Zero
6	R/O	The parity bit of the last received data byte
5	R/O	Double synchronized value of the current PSDAT being received/transmitted
4	R/O	Double synchronized value of the current PSCLK being received/transmitted
3	R/O	This bit indicates that the PS/2 controller is currently receiving data or not
2	R/O	This bit indicates that the a data is received and ready to be read
1	R/O	This bit indicates that the PS/2 controller is currently transmitting data or not
0	R/O	This bit indicates that the transmit register is empty and ready to transmit

10.6.2.3PSCONF

	6	5	4	3	2		0
	LCE	FORCE DAT LOW	FORCE CLK LOW	RX INTREN	TX INTREN		ENABLE

Bits	Type	Function
7	-	Reserved
6	R/W	*L*ine *C*ontrol detection *E*nable bit. If set, PS/2 controller checks the line control bit from external device following by STOP bit. Otherwise PS/2 controller skips checking line control bit and proceeds to next operation. Default value is zero. Most PS/2 compatible device supports line control bit mechanism. But there are some devices that don't support line control bit. To handle such device, PS/2 controller can skip line control bit detection by resetting this bit.
5	R/W	When set, PSDAT output is forced LOW regardless of the current state of the PS/2 control logic. This mode can be used as manual communication with external device.
4	R/W	When set, PSCLK output is forced LOW regardless of the current state of the PS/2 control logic.
3	R/W	Enable receiver interrupt. To set means enable interrupt. Receiver interrupt is generated whenever PS/2 controller finishes receiving a byte data from external device. Except when transmit data, PS/2 controller goes in receive mode automatically. If receiver interrupt is disabled, PS/2 controller doesn't notify a data received. So polling PSINTR interrupt register is needed.
2	R/W	Enable transmitter interrupt. To set means enable interrupt. Transmitter interrupt is generated whenever PS/2 controller completes to transmit a byte data to external device. If transmitter interrupt is disabled then poll status register to know that the transmitting transaction is completed or poll interrupt register transmitter interrupt is generated.
1	-	Reserved
0	R/W	When reset, PS/2 controller is disabled and gets into deep sleep mode. When set, enabled. To activate PS/2 controller,, first set proper parameters of timing registers and then set this bit. As soon as this bit is enabled, PS/2 controller goes into receive mode by default.

10.6.2.4PSINTR

			4	3	2	1	0
			TRANSMIT TIMEOUT	RECEIVE TIMIEOUT	PARITY ERROR	RX INTR	TX INTR

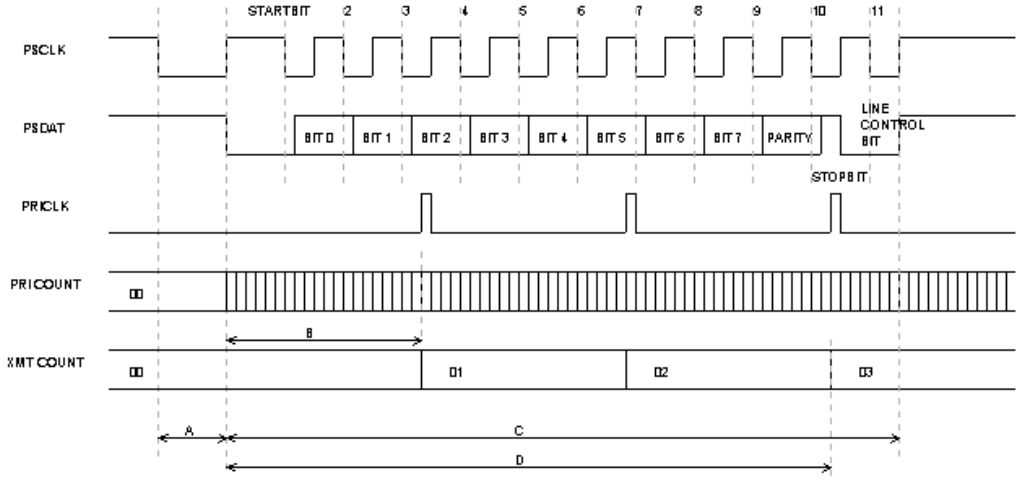
Bits	Type	Function
7:5	-	Reserved

4	R/O	Set when PS/2 controller fails to send a complete byte data to external device in a given time. The time limit is defined in PSTXMT register. PS/2 controller doesn't try to re-transmit the data. Reset when PSSTAT register is read.
3	R/O	Set when a byte data was not constructed in a certain predefined time limit due to no more bit received or bit-rate is too slow. The time limit is defined in PSTREC register. PSDATA shows the incomplete data that has been received by that time. Reset as soon as the next byte data is arrived.
2	R/O	Set when the last received data has parity error. Cleared when the very next byte data is arrived.
1	R/O	Set when PS/2 controller receives a byte data from external device. Cleared when PSDATA register is read. When PSCONF.RXINTREN is reset, the only way to know that receiver interrupt is generated is to read this bit.
0	R/O	Set when PS/2 controller completes to transmit a byte data to external device. Cleared when PSSTAT register is read. When PSCONF.TXINTREN is reset, poll this bit to confirm that the transmission is completed.

10.6.2.5PSTDLO

7	6	5	4	3	2	1	0
PSTDLO							

Bits	Type	Function
7:0	R/W	t_{PSTDLO} means the period that defines PSCLK low period before initiates transmission (A in Figure 10-1 PS/2 Controller Transmitting Data Timing Diagram). Usually the value is 64us. To meet this condition, user must set this timing register properly. INT(64us/(PSCLK speed) – 1) is appropriate value for this register.



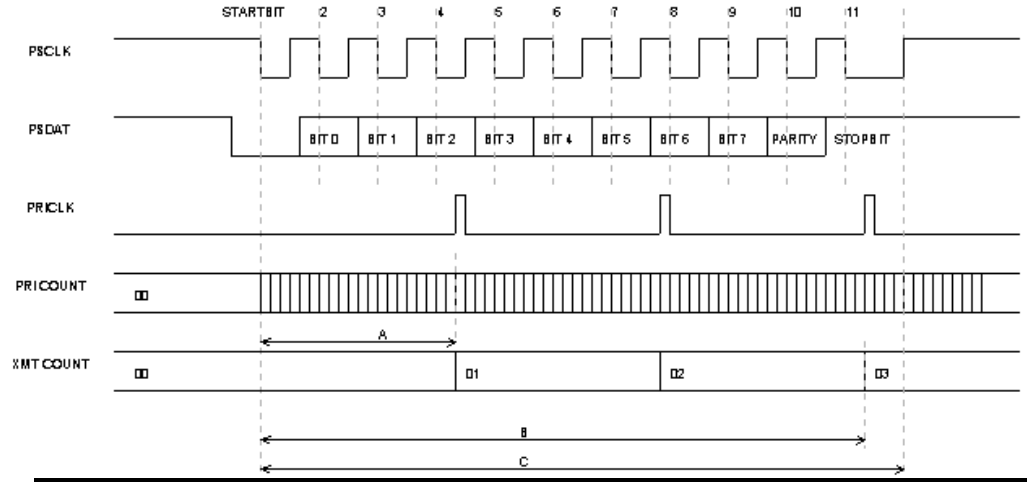
A: t_{PSTDLO}, B: t_{PSTPRI}, C: t_{XMT}, D: t_{PSTXMT}

Figure 10-1 PS/2 Controller Transmitting Data Timing Diagram

10.6.2.6PSTPRI

7	6	5	4	3	2	1	0
PSTPRI							

Bits	Type	Function
7:0	R/W	Every timer in PS/2 controller is clocked by PRICLK except PRI COUNTER that generates



PRICLK itself. The reason why uses PRICLK instead of PCLK is that PCLK is too fast so timing check counter requires more bits than slower clock rate. The period of PRICLK is determined by (PSTPRI+1) * PCLK.

10.6.2.7PSTXMT

7	6	5	4	3	2	1	0
PSTXMT							

Bits	Type	Function
7:0	R/W	This parameter determines the maximum transmission time. It is calculated as t_{PSTXMT} (D in Figure 10-1 PS/2 Controller Transmitting Data Timing Diagram) = (PSTXMT+1)*t_{PSTRI} (B in Figure 10-1 PS/2 Controller Transmitting Data Timing Diagram). Error condition is when t_{XMT}(total transmission time, C in Figure 10-1 PS/2 Controller Transmitting Data Timing Diagram) exceeds t_{PSTXMT}. Typical value of max. t_{XMT} is 15ms. So adjust t_{PSTPRI} and t_{PSTXMT} to meet the condition.

10.6.2.8PSTREC

7	6	5	4	3	2	1	0
PSTREC							

Bits	Type	Function
7:0	R/W	**This parameter determines the maximum data receiving time. It is calculated as t_{PSTREC} (B in Figure 10-2 PS/2 Controller Receiving Data Timing Diagram) = (PSTREC+1)*t_{PSTRI} (A in Figure 10-2 PS/2 Controller Receiving Data Timing Diagram). Error condition is when t_{REC}(total receiving time, C in Figure 10-2 PS/2 Controller Receiving Data Timing Diagram**) exceeds t_{PSTREC}. Typical value of max. t_{REC} is 15ms. So adjust t_{PSTPRI} and t_{PSTREC} to meet the condition.

A: t_{PSTPRI}, B: t_{PSTREC}, C: t_{REC}

Figure 10-2 PS/2 Controller Receiving Data Timing Diagram

10.6.2.9PSPWDN

							0
							PSPWDN

Bits	Type	Function
7:1	-	Reserved
0	R/W	Power Down disable. The initial value of power on reset is zero that means the PS/2 controller is in power down mode. To wake up PS/2 controller, set other timing registers then set this bit at last. User can put the PS/2 controller into power down mode by resetting this register at any time.

10.6.3Application Notes

Use pull up resistors at the PSCLK and PSDAT pad output.

10.7 RTC

This module is a 32-bit counter clocked by a 32768Hz clock. This clock needs to be provided by the system, as there is no crystal inside the block. It also contains a 32-bit match register that can be programmed to generate an interrupt signal when the time in the RTC matches the specific value written to this register (alarm function - RTC event).

FEATURES

- Two type of Alarm function

10.7.1 External Signals

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

10.7.2 Registers

Address	Name	Width	Default	Description
0x8002.8000	RTCDR	32	0x0	RTC Data Register
0x8002.8004	RTCMR	32	0x0	RTC Match Register
0x8002.8008	RTCS	2	0x0	RTC Status Register
0x8002.8010	RTCCR	2	0x0	RTC Control Register

Table 10-6 RTC Register Summary

10.7.2.1 RTC Data Register (RTCDR)

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

Bits	Type	Function
31:0	R/W	RTC Data register. Writing to this 32-bit register will load the counter. A read will give the current value of the counter. The counter is loaded by writing to the RTC data register. The counter will count up on each rising edge of the clock and loops back to 0 when the maximum value (0xFFFFFFFF) is reached. At any moment the counter value can be obtained by reading the RTC data register.

10.7.2.2 RTC Match Register (RTCMR)

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

Bits	Type	Function
31:0	R/W	RTC Match register. If this register's value is matched with current counter, an interrupt will be generated to implement alarm function. Writing to this 32-bit register will load the match register. This value can also be read back.

10.7.2.3 RTC Status Register (RTCS)

						1	0
						MATCH FLAG	1 SEC FLAG

Bits	Type	Function
7:2	-	Reserved
1	R	Match event interrupt flag bit is set if a comparator match event has occurred or 1 second has elapsed. Reading from the status register will clear the status register.
0	R	When performing a read from this register the interrupt flag will be cleared. If 1 second has elapsed, this bit will be set.

10.7.2.4RTC Control Register (RTCCR)

						1	0
						MATCH INTR EN	1 SEC INTR EN

Bits	Type	Function
7:2	-	Reserved
1	R/W	Set this bit enables match event interrupt.
0	R/W	Set this bit enables 1 second event interrupt.

10.8 TIMER

Timer 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).

FEATURES

- 32-bit up ripple counter
- Auto repeat mode
- Count enable/disable
- Interrupt enable/disable
- 3-timer channel

10.8.1 External Signals

Pin Name	Type	Description
PWM [1:0]	O	PWM Output
TimerOut	O	Timer 1 output divided by 2

10.8.2 Registers

Address	Name	Width	Default	Description
0x8002.5000	T0BASE	32	0xFFFFFFFF	Timer0 Base Register
0x8002.5008	T0COUNT	32	0x0	Timer0 Counter Register
0x8002.5010	T0CTRL	3	0x0	Timer0 Control Register
0x8002.5020	T1BASE	32	0xFFFFFFFF	Timer1 Base Register
0x8002.5028	T1COUNT	32	0x0	Timer1 Counter Register
0x8002.5030	T1CTRL	3	0x00	Timer1 Control Register
0x8002.5040	T2BASE	32	0xFFFFFFFF	Timer2 Base Register
0x8002.5048	T2COUNT	32	0x0	Timer2 Counter Register
0x8002.5050	T2CTRL	3	0x0	Timer2 Control Register
0x8002.5060	TOPCTRL	32	0x9	Top-level Control Register
0x8002.5064	TOPSTAT	3	0x0	Top-level Status Register
0x8002.5080	T64LOW	32	0x0	Lower 32-bit of 64-bit counter
0x8002.5084	T64HIGH	32	0x0	Upper 32-bit of 64-bit counter
0x8002.5088	T64CTRL	2	0x0	64-bit Timer Control Register
0x8002.50A0	P0COUNT	16	0x0	PWM channel 0 count register
0x8002.50A4	P0WIDTH	16	0xFFFF	PWM channel 0 width register
0x8002.50A8	P0PERIOD	16	0xFFFF	PWM channel 0 period register
0x8002.50AC	P0CTRL	5	0x0	PWM channel 0 control register
0x8002.50C0	P1COUNT	16	0x0	PWM channel 1 count register
0x8002.50C4	P1WIDTH	16	0xFFFF	PWM channel 1 width register
0x8002.50C8	P1PERIOD	16	0xFFFF	PWM channel 1 period register
0x8002.50CC	P1CTRL	5	0x0	PWM channel 1 control register

Table 10-7 Timer Register Summary

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

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

Bits	Type	Function
31:0	R/W	Timer 0 (Timer 1, Timer 2) Base Register. 32-bit target count value (interval) is stored in here. The interrupt interval in repeat mode is (Base Register value + 1) clock periods. For example, if the Base Register is set to 0x3333, then the timer generates an interrupt request every 0x3333 + 1 clock cycles.

10.8.2.2Timer [0,1,2] Count Register (T[0,1,2]COUNT)

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

Bits	Type	Function
31:0	R/W	32bit up counter

10.8.2.3Timer [0,1,2] Control Register (T[0,1,2]CTRL)

					2	1	0
					RESET	REPEAT MODE	COUNT ENABLE

Bits	Type	Function
7:3	-	Reserved
2	R/W	Set for reset counter register
1	R/W	Set for count repeat mode
0	R/W	Set to start count and reset to stop. For Timer 0 and Timer 1 in non-repeat mode, This bit will be cleared automatically whenever the counter reaches the target value.

10.8.2.4Timer Top-level Control Register (TOPCTRL)

	6	5	4	3	2	1	0
	TIMER OUT EN	TIMER 64 INTR EN	TIMER 64 ENABLE	POWER DOWN	TIMER 2 INTR EN	TIMER 1 INTR EN	TIMER 0 INTR EN

Bits	Type	Function
7	-	Reserved
6	R/W	Timer 1 Output Enable. The interval of this output is 2 times of interrupt interval of Timer 1. 0 = disable, 1 = enable
5	R/W	64bit Timer Counter Overflow Interrupt Enable 0 = disable, 1 = enable
4	R/W	64bit Timer Enable. 0 = disable, 1 = enable
3	R/W	Timer Controller POWER DOWN. 0 = Power Down mode, 1 = enable
2	R/W	Timer 2 Interrupt Enable 0 = disable, 1 = enable
1	R/W	Timer 1 Interrupt Enable 0 = disable, 1 = enable
0	R/W	Timer 0 Interrupt Enable. If reset, no interrupt is generated at Timer 0. 0 = disable, 1 = enable

10.8.2.5Timer Status Register (TOPSTAT)

				3	2	1	0
				TIMER 64 INTR	TIMER 2 INTR	TIMER 1 INTR	TIMER 0 INTR

Bits	Type	Function
7:4	-	Reserved
3	R	Timer 64 Interrupt Status Flag
2	R	Timer 2 Interrupt Status Flag

1	R	Timer 1 Interrupt Status Flag
0	R	Timer 0 Interrupt Status Flag

10.8.2.6Timer Lower 32-bit Base Register of 64-bit Counter (T64LOW)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
T64LOW [31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T64LOW [15:0]															

Bits	Type	Function
31:0	R/W	Lower 32bit base value of 64bit Timer

10.8.2.7Timer Upper 32-bit Base Register of 64-bit Counter (T64HIGH)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
T64HIGH [31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T64HIGH [15:0]															

Bits	Type	Function
31:0	R/W	Upper 32bit base value of 64bit Timer

10.8.2.8Timer 64-bit Counter Control Register (T64CTRL)

						1	0
						RESET	COUNT ENABLE

Bits	Type	Function
7:2	-	Reserved
1	R/W	Reset Timer 64. 0 = Keep Counting, 1 = Reset the counter register
0	R/W	Timer 64 Enable. 0 = Stop Counter, 1 = Start Counter

10.8.2.9PWM Channel [0,1] Count Register (P[0,1)COUNT)

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 [0,1] Count Register

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

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 [0,1] Width Register. Actual width of output is (P[0,1]WIDTH + 1) x PCLK.

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

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 [0,1] Period Register. Actual Period of output is (P[0,1]WIDTH + 1) x PCLK.

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

			4	3	2	1	0
			CLK SEL	OUTPUT INVERT	OUTPUT ENABLE	RESET	PWM[0,1] ENABLE

Bits	Type	Function
7:5	-	Reserved
4	R/W	PWM [0,1] Source Clock Selection 0 = 3.6864MHz, 1 = 1.8432MHz
3	R/W	PWM [0,1] Output Waveform Inverting 0 = non inverting, 1 = inverting
2	R/W	PWM [0,1] Output Enable 0 = disable output driver, 1 = enable output driver
1	R/W	PWM [0,1] Counter Reset 0 = keep count, 1 = reset counter register
0	R/W	PWM [0,1] Counter Enable. 0 = stop counter, 1 = start counter

10.9 UART/SIR

The 16C550 is a Universal Asynchronous Receiver/Transmitter (UART), with FIFOs, and is functionally identical to the 16C450 on power-up (CHARACTER mode). The 16550 can be put into an alternate mode (FIFO mode) to relieve the CPU of excessive software overhead. In this mode internal FIFOs are activated, allowing 16 bytes plus 3 bit of error data per byte in the RCVR FIFO, to be stored in both receive and transmit modes. 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 216-1, 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 16C450 software.
- After reset, all registers are identical to the 16C450 register set.
- 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).
- 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 break generation and detection.
- Internal diagnostic capabilities:
  - Loopback controls for communications link fault isolation
- Full prioritized interrupt system controls.

10.9.1 External Signals

Pin Name	Type	Description
nURING	I	UART 0 ring input signal (wake-up signal to PMU). When LOW, this indicates that a telephone ring signal has been received by the MODEM or data set. 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. **Note**: Whenever the RI bit of the MODEM Status Register changes from a HIGH to a LOW state, an interrupt is generated if the MODEM Status Interrupt is enabled. The nURING input from the external PAD is not provided. To use this signal, you should set up the UART control register of the AFE interface. For further information, refer to 13.9 Analog Front End, AFE (CODEC Interface) on

		page 13-56.
nUDTR	O	UART 0 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. A Master Reset operation sets this signal to its inactive (HIGH) state. Loop mode operation holds this signal in its inactive state.
nUCTS	I	UART 0 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 indicates whether the nUCTS input has changed state since the previous reading of the MODEM Status Register. nUCTS has no effect on the Transmitter. **Note**: Whenever the CTS bit of the MODEM Status Register changes its state, an interrupt is generated if the MODEM Status Interrupt is enabled.
nURTS	O	UART 0 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. A Master Reset operation sets this signal to its inactive (HIGH) state. Loop mode operation holds this signal in its inactive state.
nUDSR	I	UART 0 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. **Note**: Whenever the DSR bit of the MODEM Status Register changes its state, an interrupt is generated if the MODEM Status Interrupt is enabled.
nUDCD	I	UART 0 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. **Note**: Whenever the DCD bit of the MODEM Status Register changes its state, an interrupt is generated if the MODEM Status Interrupt is enabled.
USIN [0]	I	UART 0 serial data inputs. Serial data input from the communications link (peripheral device, MODEM or data set).
USOUT [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.
USIN [1]	I	UART 1 serial data inputs
USOUT [1]	O	UART 1 serial data outputs
USIN [2]	I	UART 2 serial data inputs (muxed with KSCANO5)
USOUT [2]	O	UART 2 serial data outputs (muxed with KSCANO6)
USIN [3]	I	UART 3 serial data inputs (muxed with KSCANI5)
USOUT [3]	O	UART 3 serial data outputs (muxed with KSCANI6)

10.9.2 Registers

Address	Name	Width	Default	Description
0x8002.0000	U0Base	-	-	UART 0 Base
0x8002.1000	U1Base	-	-	UART 1 Base
0x8002.D000	U2Base	-	-	UART 2 Base
0x8002.E000	U3Base	-	-	UART 3 Base
UxBase+0x00	RBR THR DLL	8	0x0	Receiver Buffer Register (DLAB = 0, Read) Transmitter Holding Register (DLAB = 0, Write) Divisor Latch Least Significant Byte (DLAB = 1)

UxBase+0x04	IER DLM	8	0x0	Interrupt Enable Register (DLAB = 0) Divisor Latch Most Significant Byte (DLAB = 1)
UxBase+0x08	IIR FCR	8	0x1 0x0	Interrupt Identification Register (Read) FIFO Control Register (Write)
UxBase+0x0C	LCR	8	0x0	Line Control Register
UxBase+0x10	MCR	3	0x0	Modem Control Register
UxBase+0x14	LSR	8	0x60	Line Status Register
UxBase+0x18	MSR	8	0xX0	Modem Status Register
UxBase+0x1C	SCR	8	0x0	Scratch Register
UxBase+0x20	UartEN	1	0x0	UART Enable Register

Table 10-8 UART/SIR Register Summary

10.9.2.1RBR/THR/DLL

7	6	5	4	3	2	1	0
Data Bit 7 ~ Data Bit 0 (RBR, THR; DLAB = 0)							
Bit 7 ~ Bit 0 (DLL; DLAB = 1)							

Bits	Type	Function
7:0	R/W	When DLAB = 0, read this register represents RBR while writes does THR. When DLAB = 1, DLL will be read or written.

10.9.2.2IER/DLM

This register 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.

7	6	5	4	3	2	1	0
0	0	0	0	MS INTR	LS INTR	TX EMPTY INTR	DATA RDY INTR
Bit 7 ~ Bit 0 DLM; (DLAB = 1)							

Bits	Type	Function	
		IER	DLM
7	R/W	0	Most significant byte of Divisor Latch
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.	

10.9.2.3IIR/FCR

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

RCVR TRIG LEVEL	-	-	-	XMIT RESET	RCVR RESET	FIFO EN

Interrupt Identification Register

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

1. Receiver Line Status
2. Received Data Ready
3. Transmitter Holding Register Empty
4. MODEM Status

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.

Bits	Type	Function
7:6	R	These two bits are set when FCR [0] = 1.
5:4	R	These two bits of the IIR are always logic 0
3:1	R	These two bits of the IIR are used to identify the highest priority interrupt pending. In the 16C450 mode, IIR [3] is 0. In the FIFO mode, IIR [3] is set along with IIR [2] when a time-out interrupt is pending **IIR [3:1]** **Interrupt Set and Reset Function** **Priority Level** **Interrupt Type** **Interrupt Source** **Interrupt Reset Control** 000 - None None - 011 Highest Receiver Line Status Overrun Error or Parity Error or Framing Error or Break Interrupt Reading the Line Status Register 010 Second Receiver Data Available Receiver Data Available or Trigger Level Reached Reading the Receiver Buffer Register or the FIFO drops below the trigger level 110 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 001 Third Transmitter Holding Register Empty

		Transmitter Holding Register Empty Reading the IIR Register (if source of interrupt) or writing into the Transmitter Holding Register 000 Fourth MODEM Status Clear to Send or Data Set Ready or Ring Indicator or Data Carrier Detect Reading the MODEM Status Register
0	R	This bit can be used in a prioritized interrupt environment to indicate whether an interrupt is pending. When bit 0 is logic 0, an interrupt is pending and the IIR contents may be used as a pointer to the appropriate interrupt service routine. When bit 0 is logic 1, no interrupt is pending

FIFO Control Register

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.

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

10.9.2.4LCR

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.

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

The UART contains a programmable Baud Generator that is capable of taking any clock input from DC to 8.0MHz and dividing it by any divisor from 2 to 216-1. 4MHz is the highest input clock frequency recommended when the divisor=1. 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 crystal frequency of 3.6864MHz. 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 10-9 Baud Rate with Decimal Divisor at 3.6864MHz Crystal Frequency

10.9.2.5MCR

This register controls the interface with the MODEM or data set (or a peripheral device emulating a MODEM).

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

Bits	Type	Function
7:5	R	These bits are permanently set to logic 0
4		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		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.

10.9.2.6LSR

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

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.
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.

10.9.2.7MSR

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.

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

Bits	Type	Function
7		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.

6		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.
5		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.
4		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.
3		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		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		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/W	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.

10.9.2.8SCR

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.

7	6	5	4	3	2	1	0
DATA							

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

10.9.2.9UartEn

							0
							UARTEN

Bits	Type	Function
7:1	-	Reserved
0	R/W	UART Enable. 0 = UART disable (Power-Down), UART Clock stop. 1 = UART enable.

10.9.3 FIFO Interrupt Mode Operation

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

1. 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.
2. 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.
3. The receiver line status interrupt (IIR-06), as before, has higher priority than the received data available (IIR-04) interrupt.
4. 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:

1. 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.

2. 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).
3. 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.
4. 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:

1. 1 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.
2. 2 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.

10.10 Watchdog Timer

The watchdog timer (WDT) has a one-channel for monitoring system operations. 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 INT_WDT 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

10.10.1 Watchdog Timer Operation

10.10.1.1 The Watchdog Timer Mode

To use the WDT as a watchdog timer, set the MODESEL and TMEN bits of the WDTCTRL 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, INT_WDT signal and PORESET/MNRESET signal are output. The INT_WDT signal is not output if INTREN is disabled (INTREN = 0).

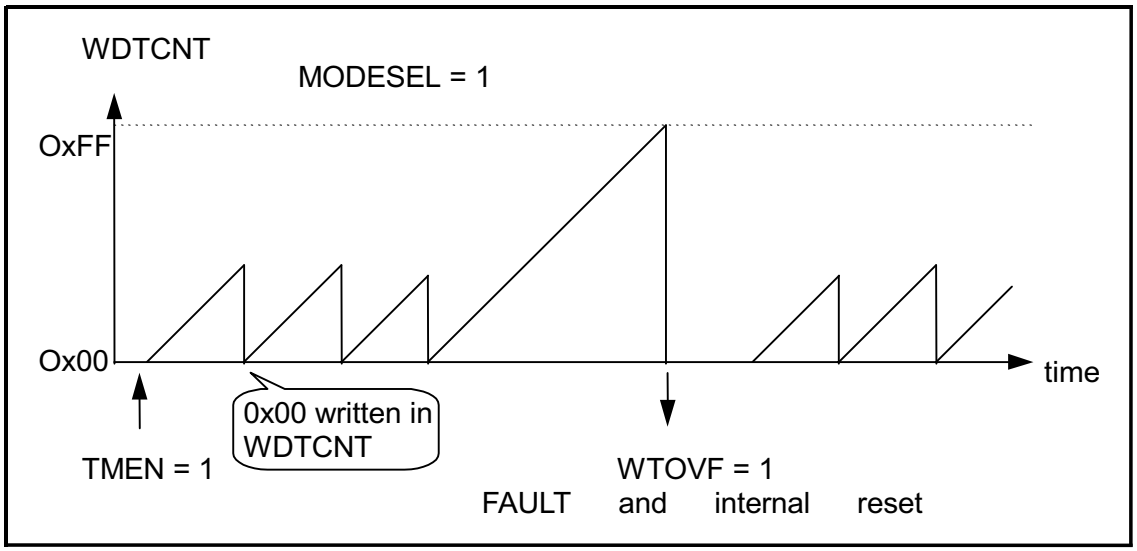


Figure 10-3 WDT Operation in the Watchdog Timer mode

If the RSTEN bit in the WDTCTRL is set to 1, a signal to reset the chip will be generated internally when WDTCNT overflows.

10.10.1.2 The Interval Timer Mode

To use the WDT as an interval timer, clear MODESEL in WDTCTRL to 0 and set TMEN to 1. A watchdog timer interrupt (INT_WDT) is generated each time the timer counter overflows. This function can be used to generate interval timer interrupts at regular intervals.

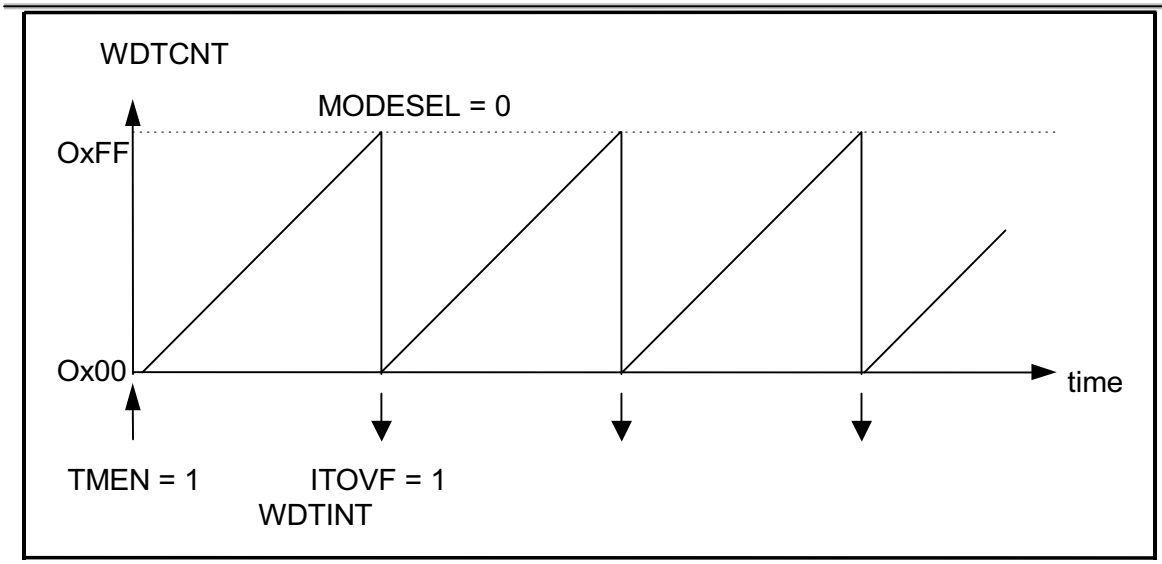


Figure 10-4 WDT Operation in the Interval Timer mode

10.10.1.3 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 (INT_WDT) 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.

10.10.1.4 Timing of clearing the overflow flag

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

10.10.2 Registers

Address	Name	Width	Default	Description
0x8002.B000	WDTCTRL	8	0x0	Timer/Reset Control
0x8002.B004	WDTSTAT	2	0x0	Reset Status
0x8002.B008	WDTCNT			Timer Counter

Table 10-10 Watchdog Timer Register Summary

10.10.2.1 WDT Control Register (WDTCTRL)

7	6	5	4	3	2	1	0
INTREN	MODESEL	TMEN	RSTEN	RSTSEL	CLK SOURCE SEL		

Bits	Type	Function
7	R/W	Enable or disable the interrupt request. 0 = disable 1 = enable
6	R/W	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 or disable the timer. 0 = disable 1 = enable
4	R/W	Select whether to reset the chip internally or not if the TCNT overflows in the watchdog timer mode.

		0 = disable 1 = enable
3	R/W	Select the type of generated internal reset if the TCNT overflows in the watchdog timer mode. 1 = manual reset enable
2:0	R/W	The WDT has a clock generator which products eight counter clock sources. The clock signals are obtained by dividing the frequency of the system clock (B_CLK). **VALUE** **CLOCK SOURCE (SYSTEM CLOCK = 40 MHz)** **OVERFLOW INTERVAL** 000 The system clock is divided by 2 12.8 us 001 The system clock is divided by 8 51.2 us 010 The system clock is divided by 32 204.8 us 011 The system clock is divided by 64 409.6 us 100 The system clock is divided by 256 1.64 ms 101 The system clock is divided by 512 3.28 ms 110 The system clock is divided by 2048 13.11 ms 111 The system clock is divided by 8192 52.43 ms

10.10.2.2 WDT Status Register (WDTSTAT)

						1	0
						ITOVF	WTOVF

Bits	Type	Function
7:2	-	Reserved
1	R	Set when WDTCNT has overflowed in the interval timer mode.
0	R	Set when WDTCNT has overflowed in the watchdog timer mode.

10.10.2.3 WDT Counter (WDTCNT)

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.

10.10.3 Examples of Register Setting

10.10.3.1 Interval Timer Mode

TCNT = 0x00
TRCR = 0xA0

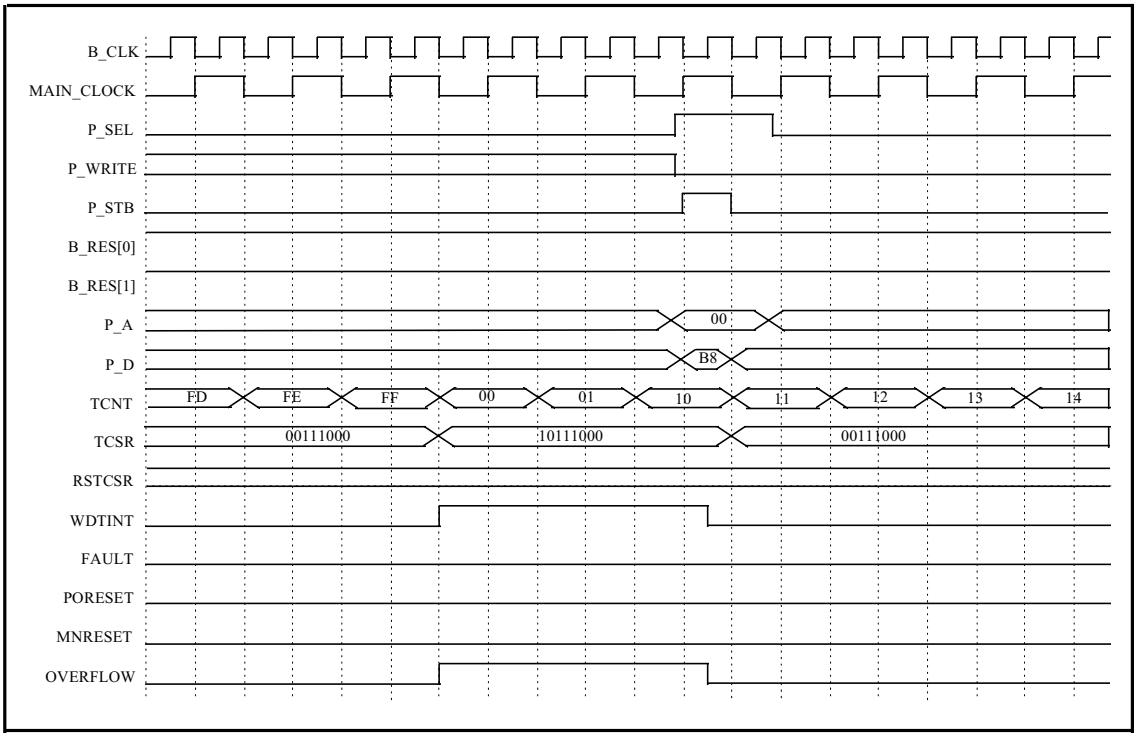


Figure 10-5 Interrupt Clear in the interval timer mode

10.10.3.2 Watchdog Timer Mode with Internal Reset Disable

TCNT = 0x00 (normally)
TRCR = 0xE0

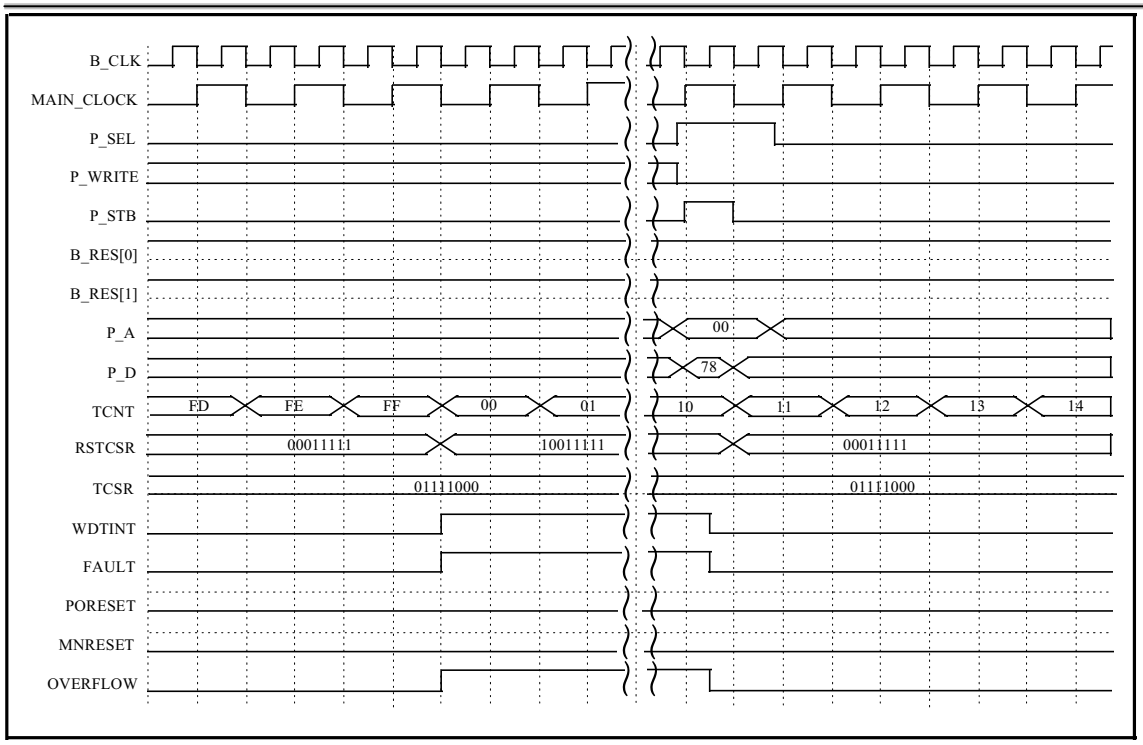


Figure 10-6 Interrupt Clear in the watchdog timer mode with reset disable

10.10.3.3 Watchdog Timer Mode with Manual Reset

TCNT = 0x00
TRCR = 0xF8

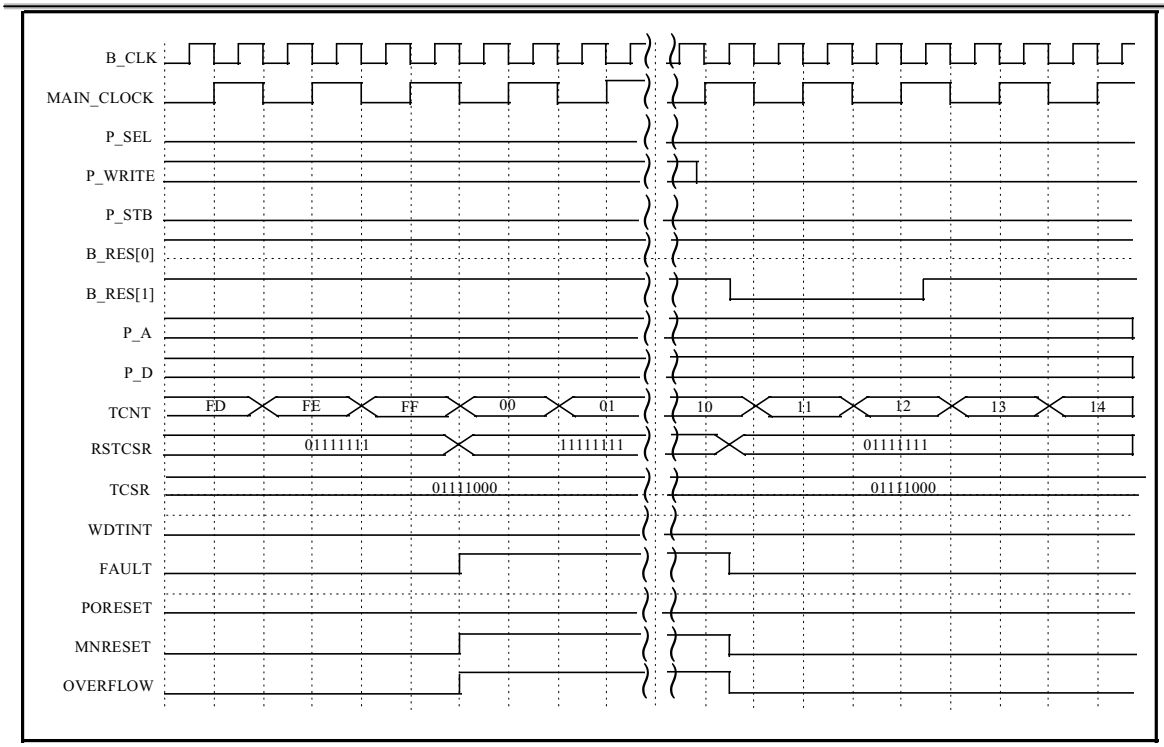


Figure 10-7 Interrupt Clear in the watchdog timer mode with manual reset

11 DEBUG AND TEST INTERFACE

12 ELECTRICAL CHARACTERISTICS