GDC21D601
32-Bit RISC MCU

Preliminary

Ver 1.6

HYUNDAI
MicroElectronics

HDS-GDC21D601-9908 / 10

The information contained herein is subject to change without notice.

The information contained herein is presented only as a guide for the applications of our products. No responsibility is assumed by Hyundai for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Hyundai or others.

These Hyundai products are intended for usage in general electronic equipment (office equipment, communication equipment, measuring equipment, domestic electrification, etc.).

Please make sure that you consult with us before you use these Hyundai products in equipment which require high quality and / or reliability, and in equipment which could have major impact to the welfare of human life (atomic energy control, airplane, spaceship, traffic signal, combustion control, all types of safety devices, etc.). Hyundai cannot accept liability to any damage which may occur in case these Hyundai products were used in the mentioned equipment without prior consultation with Hyundai.

TABLE OF CONTENTS

Section 1. Overview

1. General Description

The GDC21D601 is the HME's 32bit high performance microcontroller unit (MCU). The GDC21D601 contains ARM720T, which is a general-purpose 32bit microprocessor, and extensive peripherals: 6 channel 16bit Timer, Watch Dog Timer, 2 channel UART, 2 channel SSPI, 3 channel I^2C, Programmable Priority Interrupt Controller, 10 port PIO, 2 channel DMA Controller, External Memory Controller and BUS Controller including chip select logic.

ARM720T is a 32bit Microprocessor with the CPU of the ARM7TDMI, 8KB Cache, enlarged write buffer and Memory Management Unit (MMU). The ARM720T is fully software compatible with the ARM processor family.

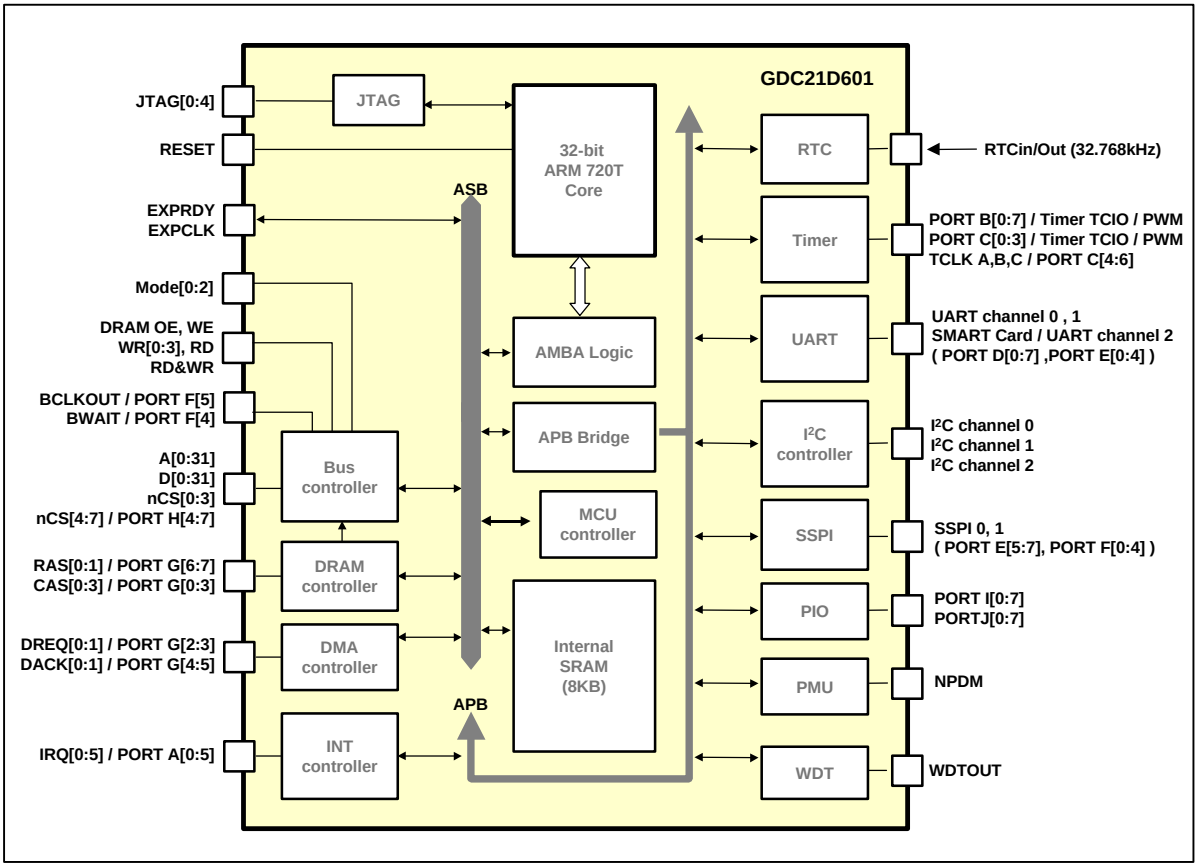


Figure. 1 GDC21D601 Block Diagram

The general descriptions of the GDC21D601 like following :

- On-Chip Modular Architecture (using AMBA)
- Utilizes the ARM720T("ARM7TDMI with 8Kbyte Cache and MMU") 32bit RISC Family
- 8Kbyte internal SRAM
- support 8bit/16bit/32bit external Data bus width
- Eight Programmable Chip Select Outputs with EXPRDY
- Support Little and Big Endian memory format
- Low Power Consumption using Power Management Unit
- Fully static operation : Max. 80MHz
- Two 32bit DMA Controllers (External request only)
- Programmable Priority Interrupt Controller (6 external sources)
- Two DRAM Banks Support
- Six 16bit Multi Function Timers / Counters for General Purpose Applications
- One 8bit Watch Dog Timer (WDT)
- Real Time Clock : 32.768 KHz
- Three UARTs (Universal Asynchronous Receiver Transmitter) compatible with 16C550 UART, one UART with Smart card interface
- Two SSPIs (Synchronous Serial Peripheral Interface) with FIFO
- Three I^2C Master/Slave Controllers
- Programmable Input/Output (8bit 10 channel)
- 208 MQFP Package

2. Feature

- ARM720T Core
  - This is an ARM7TDMI CPU core with
    . 8KB cache
    . enlarged write buffer
    . MMU(Memory Management Unit)
    . On-chip ICEbreaker debug support
    . 32-bit x 8 hardware multiplier
    . Thumb decompressor
    . High-performance 32-bit RISC architecture
    . High-density 16-bit insturction set
    Enhanced ARM software toolkit

  THUMB code is able to provide up to 65% of the code size of ARM, and 160% of the performance of an equivalent ARM processor connected to a 16-bit memory system.

  The MMU supports 4G bytes Virtual address. The allocation of virtual addresses with different task ID improves performance in task switching operations with the cache enabled.

- DMA Controller
  - Two Channels with identical function
  - Four Gigabytes of address space
  - 256 Kbytes transfers to the maximum
  - Data Transfer unit : Byte, Half-word, Word
  - Two kinds of Bus mode
    . Burst mode
    . Exception mode(Cycle steal)
  - Two kinds of address mode
    . Single address mode
    . Dual address mode
  - Two types of Transfer request source
    . External I/O request
    . Auto-request
  - Two kind of fixed priority for channels
  - Interrupted when the data transfers are complete

- DRAM Controller
  - DRAM access
  - Support Word, Half-word, and Byte transaction
  - CBR refresh in normal operation and self-refresh in power-down mode
  - Support programmable refresh rate
  - Support various DRAM access time by setting the wait count control register

- Static Memory Controller
  - Chip Select up to 8 (Each Bank is 256 MByte)
  - Exchangeable Chip Select Active High/Low (CS6 and CS7 only)
  - Little-Endian and Big-Endian Memory Support
  - Programmable wait-state (up to 16 wait-state)
  - Support External BUS Ready Strobe
  - Support various type Bus Control timing
  - Support Word, Half-word, and Byte transaction

- On-Chip SRAM
  - 8k Bytes(2048x32)
  - Asynchronous SRAM
  - Can write 8/16/32bits data, and read 32bits data

- MCU Controller
  - The Memory Map Structure Control signals
  - DRAM Power-Down Request and Powr-Down Ack signal
  - Generate the Multi Function Pin control signals
  - Device Code : $GDC601

- Power Management Unit
  - Power On Reset, WD_OF Reset, and S/W Reset
  - Status : RESET, Power Down, RUN_FAST, RUN_SLOW
  - Provide separated clock for each modules on chip
  - Provide BCLKOUT, WD_OF, Power-Down pins for external devices

- Watch Dog Timer
  - Watchdog timer mode & interval timer mode
  - Eight counter clock sources
  - Generate the Power Down reset or the Watch Dog Overflow

- Interrupt Controller
  - Asynchronous interrupt controller
  - Six external interrupt
  - Twenty internal interrupt
  - Level or edge triggered
  - Mask for each interrupt source

Request of IRQ, FIQ for each interrupt source

- Real Time Clock
  - 32bit counter clocked by a 32.768KHz clock.
  - 32bit match register

- Programmable Input Output
  - up to 80 pin (8bit 10channel)
  - Each pin can be configurable as either input or output

- Timer
  - 6 channel 16-bit up-count
  - 4-internal pre-scaleable , 4-external input clock
    . 1 interrupt per 1 channel
    . 2 inout pin per 1 channel for input capture or output compare
  - Basic function :
    . Compare match waveform output
    . Input capture
    . Match clear
    . Capture clear
  - Synchronous mode
    . Synch. clear at two or more channel
    . Synch. write at two or more channel
  - PWM waveform output mode

- Synchronous Serial Interface
  - Supports full duplex communication
  - Sends and receives data continuously, using 16 x 8 bit FIFOs
  - Built-in baud rate generator capable of generation 4 clock rate
  - Selectable clock source : either built-in buad-rate generator or external clock
  - 4 independent interrupts : transmit-end, rx-full, tx-empty and tx-full

- UART
  - 2 channel : UART only
    . Compatible with 16550
    . 16 byte each FIFO for TX / RX
    . Start, stop and parity bit can be added or deleted from/to serial data
    . 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

- SmartCard Interface
  - 1 channel : Support SmartCard Interface
    . Supports only asynchronous operation
    . Supports cards that have internal reset capability
    . Supports cards that have an active low reset input
    . Supports cards that use the internal clock
    . Generate the clock for a card expecting the external clock
    . Use the serial in/out ports for I/O
    . Use the PIO ports for other interface signals like RST, DETECT, etc

- I^2C
  - 3 channels
  - Master / Slave function
  - Programmable clock speed
  - 8bit data transfer
  - Slave clock stretch support
  - Maskable interrupt
  - Support clock rates up to 1.84MHz Baud

3. Package

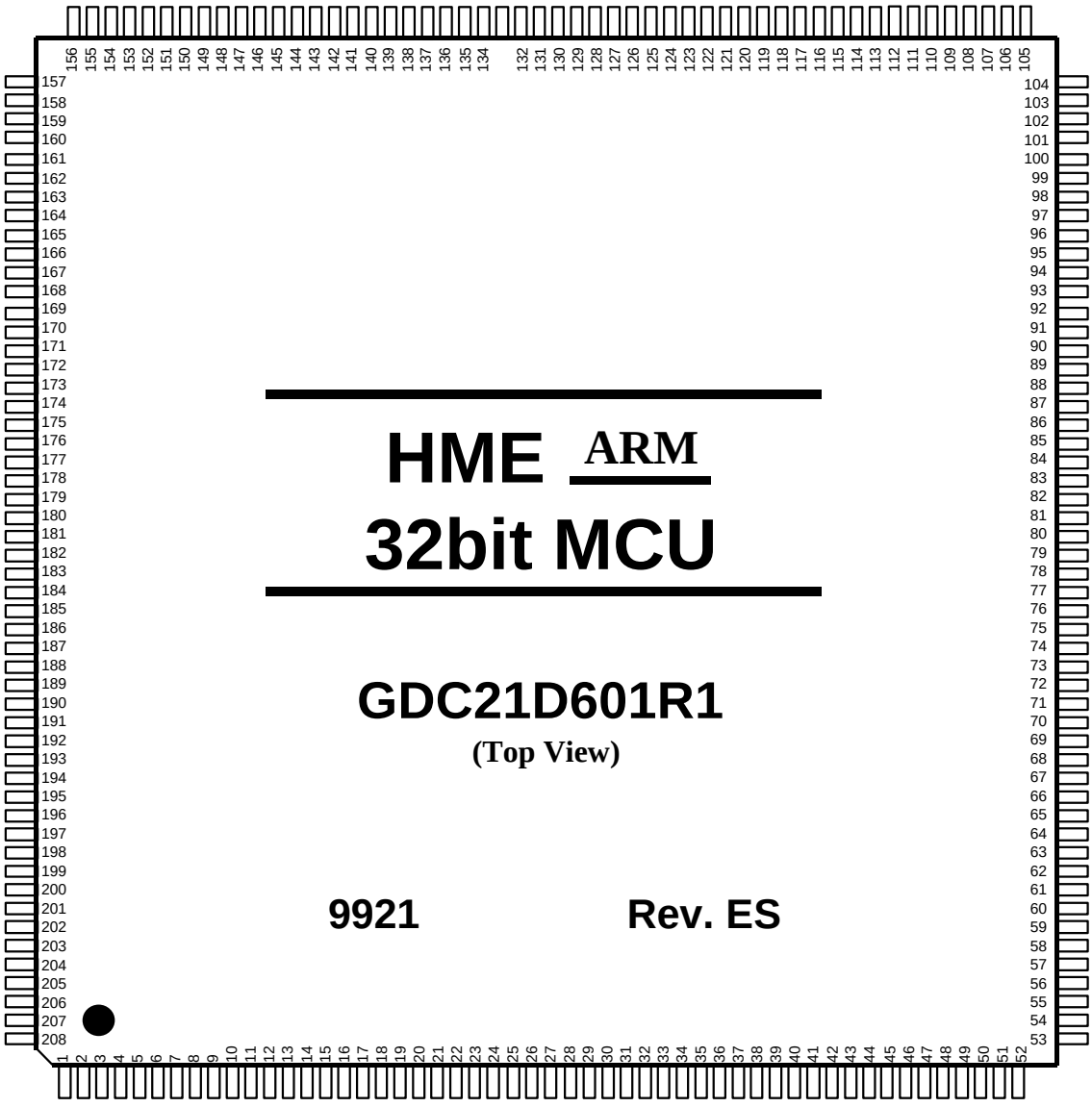


Figure 2. Package Outline

4. Pin Assignment

PIN	NAME	PIN	NAME	PIN	NAME	PIN	NAME
1	A0	45	IRQ0/PA0	89	VDD	133	NRAS0/PG6
2	A1	46	VDD	90	SMDI/PE2	134	VSS
3	VDD	47	IRQ1/PA1	91	SMDO/PE3	135	NRAS1/PG7
4	A2	48	IRQ2/PA2	92	SMCLK/PE4	136	NCAS0/PH0
5	A3	49	IRQ3/PA3	93	VSS	137	NCAS1/PH1
6	A4	50	VSS	94	SIN0/ PE5	138	NCAS2/PH2
7	VSS	51	IRQ4/PA4	95	SOUT0/PE6	139	NCAS3/PH3
8	A5	52	IRQ5/PA5	96	SCLK0/PE7	140	VSS
9	A6	53	PA6	97	VDD	141	XOUT
10	A7	54	PA7	98	SCS0/PF0/MemByte0	142	XIN
11	VDD	55	TCIOA0/PB0	99	SIN1/PF1/MemByte1	143	VDD
12	A8	56	VDD	100	SOUT1/PF2	144	NDRAMOE
13	A9	57	TCIOB0/PB1	101	VSS	145	NDRAMWE
14	A10	58	TCIOA1/PB2	102	SCLK1/PF3	146	VSS
15	VSS	59	TCIOB1/PB3	103	SCS1/PF4	147	NWR0
16	A11	60	VSS	104	BCLKOUT/PF5	148	NWR1
17	A12	61	TCIOA2/PB4	105	NFIQOUT/PF6	149	NWR2
18	VDD	62	TCIOB2/PB5	106	NIRQOUT/PF7	150	VDD
19	A13	63	TCIOA3/PB6	107	VDD	151	NWR3
20	A14	64	VDD	108	I^2CSDA0	152	NRD
21	A15	65	TCIOB3/PB7	109	I^2CSCL0	153	RDNWR
22	VSS	66	PC0/TCIOA4	110	I^2CSDA1	154	VSS
23	A16	67	PC1/TCIOB4	111	VSS	155	NEXPRDY
24	A17	68	VSS	112	I^2CSCL1	156	EXPCLK
25	A18	69	PC2/TCIOA5	113	I^2CSDA2	157	NCS0
26	VDD	70	PC3/TCIOB5	114	I^2CSCL2	158	NCS1
27	A19	71	PC4/TCLKA	115	VDD	159	NCS2
28	A20	72	VDD	116	Mode0/TREQA	160	VDD
29	A21	73	PC5/TCLKB	117	Mode1/TREQB	161	NCS3
30	VSS	74	PC6/TCLKC	118	Mode2/TACK	162	NCS4/PH4
31	A22	75	PC7/TCLKD	119	VSS	163	NCS5/PH5
32	A23	76	RXD0/PD0	120	UCLKOUT	164	VSS
33	WDTOUT	77	VSS	121	UCLKIN	165	CS6/PH6
34	NPDN	78	TXD0/PD1	122	VDD	166	CS7/PH7
35	VSS	79	RXD1/ PD2	123	TEST	167	D31/PJ7
36	RTCOSCIN	80	TXD1/PD3	124	NEXTREQ/PG0	168	VDD
37	RTCOSCOUT	81	VDD	125	NRESET	169	D30/PJ6
38	VDD	82	NCTS/ PD4	126	VSS	170	D29/PJ5
39	NTRST	83	NDSR/ PD5	127	NEXTACK/PG1	171	D28/PJ4
40	TDI	84	NDCD/ PD6	128	NDREQ0/PG2	172	VSS
41	TCK	85	VSS	129	NDACK0/PG3	173	D27/PJ3
42	VSS	86	NRI/ PD7	130	VDD	174	D26/PJ2
43	TDO	87	NDTR/PE0	131	NDREQ1/PG4	175	D25/PJ1
44	TMS	88	NRTS/PE1	132	NDACK1/PG5	176	VDD

PIN	NAME	PIN	NAME	PIN	NAME	PIN	NAME
177	D24/PJ0	185	VDD	193	VDD	201	VDD
178	D23/PI7	186	D17/PI1	194	D11	202	D5
179	D22/PI6	187	D16/PI0	195	D10	203	D4
180	D21/PI5	188	D15	196	D9	204	D3
181	VSS	189	VSS	197	VSS	205	VSS
182	D20/PI4	190	D14	198	D8	206	D2
183	D19/PI3	191	D13	199	D7	207	D1
184	D18/PI2	192	D12	200	D6	208	D0

5. Pin Descriptions

PIN NUMBER	PIN NAME	TYPE	DESCRIPTION
1~2, 4~6, 8~10, 12~14, 16~17, 19~21, 23~25, 27~29, 31~32	A[31:0]	O	Address Bus *Valid After RESET.*
33	WDTOUT	O	Watch Dog Timer Overflow Output
34	NPDN	O	Power Down Signal from PMU block *When it is LOW, MCU entered the power down mode. When HIGH, normal*
37	RTCOSCIN	I	Real Time Clock Oscillator Input 32.768kHz
36	RTCOSCOUT	O	Real Time Clock Oscillator Output
39	NRST	I	JTAG Reset
40	TDI	I	JTAG Data Input
41	TCK	I	JTAG Clock Input
43	TDO	O	JTAG Data Output
44	TMS	I	JTAG Mode Signal
45	IRQ0	I/O	External Interrupt Input 0, when PINMUX_PA[0] = 0 *Programmable I/O ports. Each pin can be mapped to specified function pin name. (External IRQ0,IRQ1,..).*
	PA0		PIO Port A[0], when PINMUX_PA[0] = 1
47	IRQ1	I/O	External Interrupt Input 1, when PINMUX_PA[1] = 0
	PA1		PIO Port A[1], when PINMUX_PA[1] = 1
48	IRQ2	I/O	External Interrupt Input 2, when PINMUX_PA[2] = 0
	PA2		PIO Port A[2], when PINMUX_PA[2] = 1
49	IRQ3	I/O	External Interrupt Input 3, when PINMUX_PA[3] = 0
	PA3		PIO Port A[3], when PINMUX_PA[3] = 1
51	IRQ4	I/O	External Interrupt Input 4, when PINMUX_PA[4] = 0
	PA4		PIO Port A[4], when PINMUX_PA[4] =1
52	IRQ5	I/O	External Interrupt Input 5, when PINMUX_PA[5] = 0
	PA5		PIO Port A[5], when PINMUX_PA[5] = 1
53	PA6	I/O	PIO Port A[6] Tbclk Clock Input for TIC test
54	PA7	I/O	PIO Port A[7] Tfclk Clock Input for TIC test
55	TCIOA0	I/O	Timer Channel 0 Input Capture A, when PINMUX_PB[0] = 0
	PB0		PIO Port B[0], when PINMUX_PB[0] = 1
57	TCIOB0	I/O	Timer Channel 0 Input Capture B, when PINMUX_PB[1] = 0
	PB1		PIO Port B[1], when PINMUX_PB[1] = 1
58	TCIOA1	I/O	Timer Channel 1 Input Capture A, when PINMUX_PB[2] = 0
	PB2		PIO Port B[2], when PINMX_PB[2] = 1
59	TCIOB1	I/O	Timer Channel 1 Input Capture B, when PINMUX_PB[3] = 0
	PB3		PIO Port B[3], when PINMUX_PB[3] = 1
61	TCIOA2	I/O	Timer Channel 2 Input Capture A, when PINMUX_PB[4] = 0
	PB4		PIO Port B[4], when PINMUX_PB[4] = 1
62	TCIOB2	I/O	Timer Channel 2 Input Capture B, when PINMUX_PB[5] = 0
	PB5		PIO Port B[5], when PINMUX_PB[5] = 1

PIN NUMBER	PIN NAME	TYPE	DESCRIPTION
63	TCIOA3	I/O	Timer Channel 3 Input Capture A, when PINMUX_PB[6] = 0
	PB6		PIO Port B[6], when PINMUX_PB[6] = 1
65	TCIOB3	I/O	Timer Channel 3 Input Capture B, when PINMUX_PB[7] = 0
	PB7		PIO Port B[7], when PINMUX_PB[7] = 1
66	PC0	I/O	PIO Port C[0], when PINMUX_PC[0] = 0
	TCIOA4		Timer Channel 4 Input Capture A, when PINMUX_PC[0] = 1
67	PC1	I/O	PIO Port C[1], when PINMUX_PC[1] = 0
	TCIOB4		Timer Channel 4 Input Capture B, when PINMUX_PC[1] = 1
69	PC2	I/O	PIO Port C[2], when PINMUX_PC[2] = 0
	TCIOA5		Timer Channel 5 Input Capture A, when PINMUX_PC[2] = 1
70	PC3	I/O	PIO Port C[3], when PINMUX_PC[3] = 0
	TCIOB5		Timer Channel 5 Input Capture B, when PINMUX_PC[3] = 1
71	PC4	I/O	PIO Port C[4], when PINMUX_PC[4] = 0
	TCLKA		External Timer Clock Source A, when PINMUX_PC[4] = 1
73	PC5	I/O	PIO Port C[5], when PINMUX_PC[5] = 0
	TCLKB		External Timer Clock Source B, when PINMUX_PC[5] = 1
74	PC6	I/O	PIO Port C[6], when PINMUX_PC[6] = 0
	TCLKC		External Timer Clock Source C, when PINMUX_PC[6] = 1
75	PC7	I/O	PIO Port C[7], when PINMUX_PC[7] = 0
	TCLKD		External Timer Clock Source D, when PINMUX_PC[7] = 1
76	RXD0	I/O	UART Channel 0 Receive Data, when PINMUX_PD[0] = 0
	PD0		PIO Port D[0], when PINMUX_PD[0] = 1
78	TXD0	I/O	UART Channel 0 Transmit Data, when PINMUX_PD[1] = 0
	PD1		PIO Port D[1], when PINMUX_PD[1] =1
79	RXD1	I/O	UART Channel 1 Receive Data, when PINMUX_PD[2] = 0
	PD2		PIO Port D[2], when PINMUX_PD[2] = 1
80	TXD1	I/O	UART Ch 1 Transmit Data, when PINMUX_PD[3] = 0
	PD3		PIO Port D[3], when PINMUX_PD[3] =1
82	NCTS	I/O	UART Ch 1 Clear to Send, when PINMUX_PD[4] = 0
	PD4		PIO Port D[4], when PINMUX_PD[4] = 1
83	NDSR	I/O	UART Ch 1 Data Set Ready, when PINMUX_PD[5] = 0
	PD5		PIO Port D[5], when PINMUX_PD[5] = 1
84	NDCD	I/O	UART Ch 1 Data Carrier Detect, when PINMUX_PD[6] = 0
	PD6		PIO Port D[6], when PINMUX_PD[6] = 1
86	NRI	I/O	UART Ch 1 Ring Indicator, when PINMUX_PD[7] = 0
	PD7		PIO Port D[7], when PINMUX_PD[7] =1
87	NDTR	I/O	UART C 1 Data Terminal Ready, when PINMUX_PE[0] = 0
	PE0		PIO Port E[0], when PINMUX_PE[0] = 1
88	NRTS	I/O	UART Ch 1 Ready to Send Data, when PINMUX_PE[1] = 0
	PE1		PIO Port E[1], when PINMUX_PE[1] = 1
90	SMDI	I/O	Smart Card Interface Data In, when PINMUX_PE[2] = 0
	PE2		PIO Port E[2], when PINMUX_PE[2] = 1
91	SMDO	I/O	Smart Card Interface Data Out, when PINMUX_PE[3] = 0
	PE3		PIO Port E[3], when PINMUX_PE[3] =1
92	SMCLK	I/O	Smart Card Interface Clock Out, when PINMUX_PE[4] = 0
	PE4		PIO Port E[4], when PINMUX_PE[4] = 1

PIN NUMBER	PIN NAME	TYPE	DESCRIPTION
94	SIN0	I/O	SSI Channel 0 Data In, when PINMUX_PE[5] = 0
	PE5		PIO Port E[5], when PINMUX_PE[5] =1
	BPROT0		AMNA BPROT[0] Signal, when PINMUX_PE[8] = 1
95	SOUT0	I/O	SSI Channel 0 Data Out, when PINMUX_PE[6] = 0
	PE6		PIO Port E[6], when PINMUX_PE[6] = 1
	BPROT1		AMBA BPROT[1] Signal, when PINMUX_PE[8] = 1
96	SCLK0	I/O	SSI Channel 0 Clock Out, when PINMUX_PE[7] = 0
	PE7		PIO Port E[7], when PINMUX_PE[7] =1
	BLOK		AMBA BLOK Signal Out, when PINMUX_PE[8] = 1
98	SCS0	I/O	SSI Channel 0 Channel Control, when PINMUX_PF[0] = 0
	PF0		PIO Port F[0], when PINMUX_PF[0] = 1
	MemByte0		MemByte[0] Signal from EBI Block, when PINMUX_PF[8] = 1
99	SIN1	I/O	SSI Channel 1 Data In, when PINMUX_PF[1] = 0
	PF1		PIO Port F[1], when PINMUX_PF[1] = 1
	MemByte1		MemByte[1] Signal from EBI Block, when PINMUX_PF[8] = 1
100	SOUT1	I/O	SSI Channel 1 Data Out, when PINMUX_PF[2] = 0
	PF2		PIO Port F[2], when PINMUX_PF[2] = 1
	BTRANS0		AMBA BTRANS[0] Signal, when PINMUX_PF[8] = 1
102	SCLK1	I/O	SSI Channel 1 Clock Out, when PINMUX_PF[3] = 0
	PF3		PIO Port F[3], when PINMUX_PF[3] = 1
	BTRANS[1]		AMBA BTRANS[1] Signal, when PINMUX_PF[8] = 1
103	SCS1	I/O	SSI Channel 1 Channel Control, when PINMUX_PF[4] = 0
	PF4		PIO Port F[4], when PINMUX_PF[4] =1
	BWAIT		AMBA BWAIT Signal, when PINMUX_PF[8] = 1
104	BCLKOUT	I/O	AMBA BCLK Signal, when PINMUX_PF[5] = 0
	PF5		PIO Port F[5], when PINMUX_PF[5] = 1
105	NFIQOUT	I/O	AMBA NFIQ Signal, when PINMUX_PF[6] = 0
	PF6		PIO Port F[6], when PINMUX_PF[6] =1
106	NIRQOUT	I/O	AMBA NIRQ Signal, when PINMUX_PF[7] = 0
	PF7		PIO Port F[7], when PINMUX_PF[7] = 1
108	I^2CSDA0	I/O	Data Signal for I^2C Channel 0 *Pins (108~110,112~114) are required to be pull-up externally.* *When bus is free, this pin goes logical "HIGH"* *After reset, SDA pins enter Idle state*
109	I^2CSCL0	I/O	Clock Signal for I^2C Channel 0
110	I^2CSDA1	I/O	Data Signal for I^2C Channel 1
112	I^2CSCL1	I/O	Clock Signal for I^2C Channel 1
113	I^2CSDA2	I/O	Data Signal for I^2C Channel 2
114	I^2CSCL2	I/O	Clock Signal for I^2C Channel 2
116	Mode0	I	Boot Mode0, when TEST pin = 0 *By default, 32-bit access* *(MCU can boot from 32- bit Memory)*
	TREQA		TREQA Signal for TIC Test, when TEST pin = 1
117	Mode1	I	Boot Mode 1
	TREQB		TREQB Signal for TIC Test

Mode[0:1] = 00 32-bit
Mode[0:1] = 01 8-bit
Mode[0:1] = 10 16-bit
Mode[0:1] = 11 Reserved

PIN NUMBER	PIN NAME	TYPE	DESCRIPTION
118	Mode 2	I/O	Boot Mode 2 (BigEndian Pin) Big-endian Selection Pin, when this pin = 1(HIGH) *Note) When this pin is HIGH, External Data will be transferred "Big-endian" format.*
	TACK		TACK Signal for TIC Test
120	UCLKIN	I	UART Clock Oscillator Clock Input *UART block dedicated clock source supported.* *(This clock source is used for UART and SMART Card Only)*
121	UCLKOUT	O	UART Clock Oscillator Clock Output
123	TEST	I	Test Input Pin, Select 116~118 pin as Boot Mode or TIC Signal
124	NEXTREQ	I/O	External Master Request Bus Mastership, when PINMUX_PG[0] = 0
	PG0		PIO Port G[0], when PINMUX_PG[0] = 1
125	NRESET	I	System Power On Reset Input *To ensure proper initialization after power is stable, assert NRESET pin for at least 20ms*
127	NEXTACK	I/O	Bus Granted Signal for External Master, when PINMUX_PG[1] = 0
	PG1		PIO Port G[1] = 1, when PINMUX_PG[1] = 1
128	NDREQ0	I/O	DMA Channel 0 Request, when PINMUX_PG[2] = 0
	PG2		PIO Port G[2], when PINMUX_PG[2] = 1
129	NDACK0	I/O	DMA Channel 0 Acknowledge, when PINMUX_PG[3] = 0
	PG3		PIO Port G[3], when PINMUX_PG[3] = 1
131	NDREQ1	I/O	DMA Channel 1 Request, when PINMUX_PG[4] = 0
	PG4		PIO Port G[4], when PINMUX_PG[4] = 1
132	NDACK1	I/O	DMA Channel 1 Acknowledge, when PINMUX_PG[5] = 0
	PG5		PIO Port G[5], when PINMUX_PG[5] = 1
133	NRAS0	I/O	DRAM Bank #0 RAS Signal, when PINMUX_PG[6] = 0
	PG6		PIO Port G[6], when PINMUX_PG[6] = 1
135	NRAS1	I/O	DRAM Bank #1 RAS Signal, when PINMUX_PG[7] = 0
	PG7		PIO Port G[7], when PINMUX_PG[7] = 1
136	NCAS0	I/O	DRAM CAS0 Signal, when PINMUX_PH[0] = 0
	PH0		PIO Port H[0], when PINMUX_PH[0] = 1
137	NCAS1	I/O	DRAM CAS1 Signal, when PINMUX_PH[1] = 0
	PH1		PIO Port H[1], when PINMUX_PH[1] = 1
138	NCAS2	I/O	DRAM CAS2 Signal, when PINMUX_PH[2] = 0
	PH2		PIO Port H[2], when PINMUX_PH[2] = 1
139	NCAS3	I/O	DRAM CAS3 Signal, when PINMUX_PH[3] = 0
	PH3		PIO Port H[3], when PINMUX_PH[3] = 1
142	XIN	I	System Clock Input (<80MHz) External TTL oscillator input
141	XOUT	O	System Clock Oscillator Output
144	NDRAMOE	O	DRAM Output Enable
145	NDRAMWE	O	DRAM Write Enable
147	NWR0	O	Write Enable 0 for Static Memory(Byte)
148	NWR1	O	Write Enable 1 for Static Memory(Byte)
149	NWR2	O	Write Enable 2 for Static Memory(Byte)
151	NWR3	O	Write Enable 3 for Static Memory(Byte)

PIN NUMBER	PIN NAME	TYPE	DESCRIPTION
152	NRD	O	Output Enable Signal for Static Memory
153	RDNWR	O	Read/Write Signal
155	EXPRDY	I	Ready Signal Input *When this pin is Low, current memory transfer extended.*
156	EXPCLK	O	Clock Output Signal *Active only during external cycles.* *Output is same phase and speed as the bus clock*
157	NCS0	O	Chip Select Signal for Bank #0 *NCS pins are required to be Pull-up for proper operation.* *All NCS pins are Active Low See Fig.1 memory Map(Section 2)*
158	NCS1	O	Chip Select Signal for Bank #1
159	NCS2	O	Chip Select Signal for Bank #2
161	NCS3	O	Chip Select Signal for Bank #3
162	NCS4	I/O	Chip Select Signal for Bank #4, when PINMUX_PH[4] = 0
	PH4		PIO Port H[4], when PINMUX_PH[4] = 1
163	NCS5	I/O	Chip Select Signal for Bank #5, when PINMUX_PH[5] = 0
	PH5		PIO Port H[5], when PINMUX_PH[5] = 1
165	CS6	I/O	Chip Select Signal for Bank #6, when PINMUX_PH[6] = 0 *CS6 pin can be programmed active HIGH/LOW*
	PH6		PIO Port H[6], when PINMUX_PH[6] = 1
166	CS7	I/O	Chip Select Signal for Bank #7, when PINMUX_PH[7] = 0 *CS7 pin can be programmed active HIGH/LOW*
	PH7		PIO Port H[7], when PINMUX_PH[7] = 1
167, 169~171, 173~175, 177~180, 182~184, 186~188, 190~192, 194~196, 198~200, 202~204, 206~208	D[31:0]	I/O	Data Bus
167, 169~171, 173~175, 177	PJ[7:0]	I/O	PIO Port J[7:0], when PINMUX_PJ[7:0] = 1
178~180, 182~184, 186~187	PI[7:0]	I/O	PIO Port I[7:0] , when PINMUX_PJ[7:0] = 1
3, 11, 18, 26, 38, 46, 56, 64, 72, 81, 89, 97, 107, 115, 122, 130, 143, 150, 160, 168, 176, 185, 193, 201	VDD	I	Power
7, 15, 22, 30, 35, 42, 50, 60, 68, 77, 85, 93, 101, 111, 119, 126, 134, 140, 146, 154, 164, 172, 181, 189, 197, 205	VSS	I	Ground

Section 2. System Architecture

1. Internal Bus Architecture

The GDC21D601 take the advantage of the AMBA(Advanced Micro-controller Bus Architecture) as the internal Bus Architecture. The AMBA specification defines an on-chip communication standard for designing high-performance embedded micocontrollers. Two distinct buses are defined within the AMBA:

- the Advanced System Bus (ASB)
- the Advanced Peripheral Bus (APB)

The AMBA ASB is for high-performance system modules. The modules connected to ASB are DRAM Controller, Static Memory Controller, DMA Controller, On-Chip SRAM, ARM720T CPU Core, Arbiter, Decoder, APB Bridge, and TIC.
The AMBA APB is for low-power peripherals. AMBA APB is optimized for minimal power consumption and reduced interface complexity to support peripheral functions. The modules connected to APB are PIO, Interrupt Controller, PMU, WDT, RTC, Timer, UART, SSPI, and I^2C.
See also AMBA Specification Rev. D (ARM IHI 0001D), and AMBA Specification Rev. 2.0 (ARM IHI 0011A) for detail.

2. Arbiter

The AMBA bus specification is a multi-master bus standard. As a result, a bus arbiter is needed to ensure that only one bus master has an access to the bus at any particular point of time. Each bus master can request the bus; the Arbiter decides which has the highest priority and issues a grant signal accordingly. The GDC21D601 can have the four bus master: ARM720T CPU Core, DMA Controller, TIC, and External Bus Master.

Every system must have a default bus master which grants the use of bus during reset, when no other bus master requires the bus. During Power On Reset, the arbiter will grant the use of bus to the default bus master and hold all other grant signals inactive. The ARM720T Core, the default bus master will grant for the use of bus under the following conditions: Reset, standby, power-down, and no other master requesting the bus

The arbiter processes the requests of the ownership of the ASB and grants one ASB master according to the arbitration scheme. The arbitration scheme of this implementation is a simple priority encoded scheme where the highest priority master requesting the ASB is granted. The priority order is as follows:

Case 1) Aripri = ‘0’
1. TIC
2. DMA
3. External BUS Master
4. ARM (default bus master)

Case 2) Aripri = ‘1’
1. TIC
2. External BUS Master
3. DMA
4. ARM (default bus master)

3. System Decoder

The decoder in an AMBA system is used to perform a centralized address decoding function, which gives two main advantages:

- It improves the portability of peripherals, by making them independent of the system memory map.
- It simplifies the design of bus slaves, by centralizing the address decoding and bus control functions.

The decoder performs three main tasks:
- address decoder
- default transfer response
- protection unit

The decoder generates a select signal for each slave on the ASB bus and, under certain circumstances, will not select any slaves and provide the transac-tion response itself.
The MCU System Memory Map is shown in Figure 1.

The decoder greatly simplifies the slave interface and removes the need for the slave to understand the different types of transfer that may occur on the bus.

4. Memory Map

The system decoder controls the memory map of the system and generates a slave select signal for each memory region.

The ReMap signal is used to provide a different memory map: ROM is required at address 0 when power on reset, and RAM also may be used at address 0 during normal operation.
The ReMap signal is typically provided by a Power Management Unit (PMU) which drives ReMap to LOW at reset. The signal is only driven to HIGH after a particular register in the PMU is accessed (See Section. 9 Power Management Unot for detail). When ReMap is HIGH and isram signal is HIGH, then Memory Map Configuration is MODE A which the internal SRAM is located at address 0x00. And When ReMap is HIGH and drambank0 signal is HIGH, then Memory Map Configuration is Mode B which the DRAM bank #0 is located at address 0x00. The isram and drambank0 signal come from MCU Controller. See Section 8. MCU Controller for detail.

Figure 2. Memory map configuration shows both the Reset (MODE R) and the Normal (MODE B and MODE A) memory map

Figure 1. shows the system memory map.

5. Memory Format

The ARM720T CPU Core supports both the Big-Endian and Little-Endian format. And the GDC21D601 can also support the Big-Endian and Little-Endian memory format. The GDC21D601 can support the Little-Endian Format by default. When using the GDC21D601 as Big-Endian format: 1) set Boot Mode 2 pin to VDD, and 2) set the ARM720T as Big-Endian mode with using Coprocessor instruction. 3) set the Big-Endian flag of the compile options when compile. The example of the coprocessor instruction is in the below. It is noted that CP15 register (CPU control register) can only be accessed with MRC and MCR instructions in a Privileged mode. See the ARM720T Data Sheet (ARM DDI 0087D) for detail. The ARM720T Data Sheet is downloadable from ARM home page (http://www.arm.com).

For example :

```
MRC   p15, 0, r3, c1, c1
ORR   r3, r3, #0x80
MCR   p15, 0, r3, c1, c1
```

Note : The GDC21D601 has a EBI (External Bus Interface) block which can copy the Byte or Half-Word of the lower position in data bus to higher data bus position, so you can use the GDC21D601 as BigEnd mode by only set the Boot Mode 2 pin to VDD and in this case you may not set the ARM720T as BigEnd Mode.

6. Boot Mode

The GDC21D601 can support 32/16/8 Bit Boot ROM. By default MCU can boot from 32 bit ROM. In this case Boot Mode[1:0] (pin number 116 and 117) are “00”. If you want use 16 bit Boot ROM, then you must set Boot Mode[1:0] are “10”. And in case of Booting from 8 bit ROM, you must set Boot Mode[1:0] are “01”. It is for reserved in case that Boot Mode[1:0] are “11” . See the Table 1. The Description of the Mode Pin.
In all case of boot mode the wait cycle of Boot area is 3 cycles. If you want to know about boot mode for detail you must see the Section 6. Static Memory Controller.

Table 1. The Description of the Mode Pin

Mode[1:0]	Bus width of Booting ROM
00	32 Bit
01	8 Bit
10	16 Bit
11	Reserved

7. Multi-Function Pin

The GDC21D601 has 80 Bit PIO pins with multiplexed by other functional pins. So you must use properly these multi-function pins by setting the PINMUX control registers in MCU controller. (See Section 8. MCU Controller for detail)

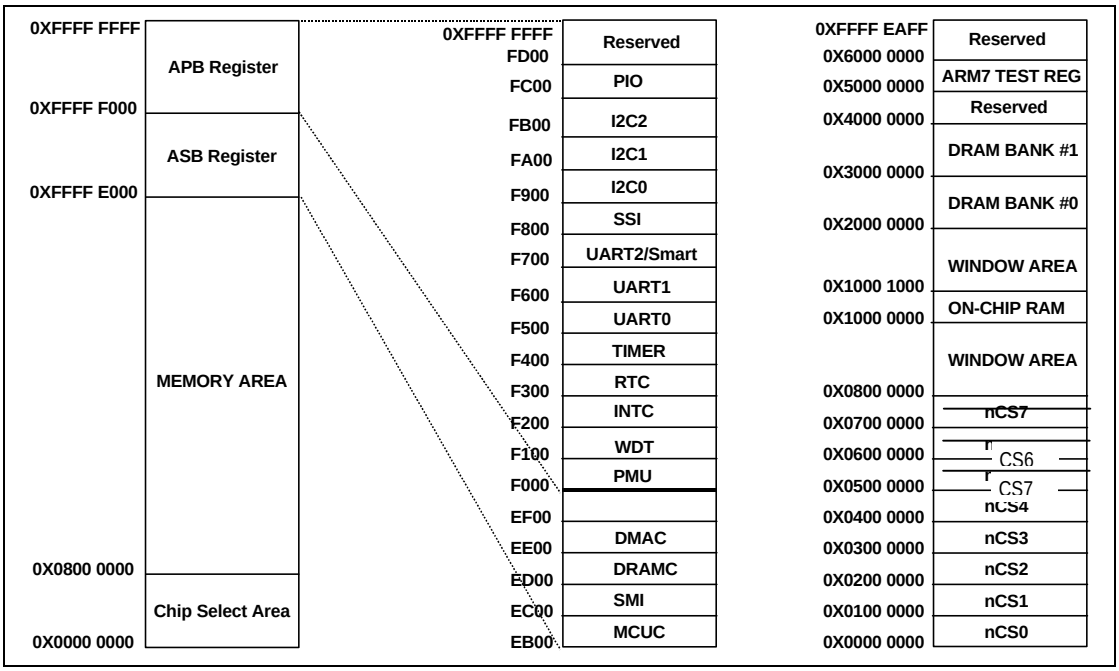
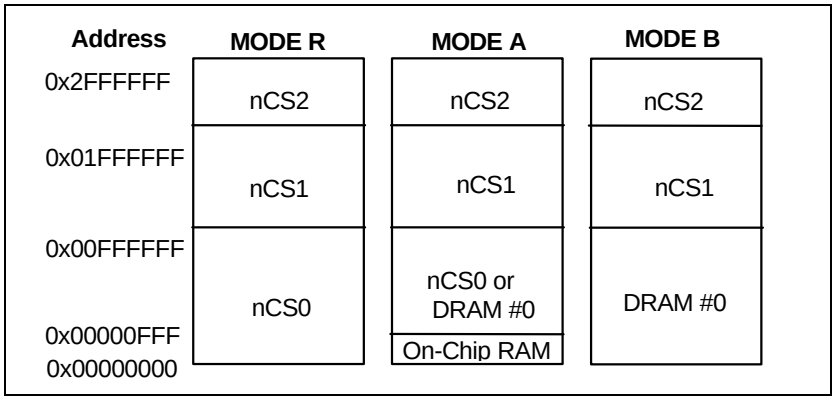


Figure 1. System Memory Map



1. MODE R : Reset Mode : default mode from power-on reset (ReMap is LOW)
2. MODE A : On-Chip SRAM in 0x0000 ~ 0x07FF range : ReMap is HIGH and isram is HIGH
3. MODE B : DRAM Bank #0 in 0x00000000 ~ 0x00FFFFFF range : Remap is HIGH and drambank0 is HIGH

Figure 2. Memory Map Configuration

Section 3. ARM720T Core

1. General Description

ARM720T is 32bit microprocessor of general purpose with 8KB cache, enlarged write buffer and Memory Management Unit (MMU), which are combined in a single chip. The CPU within ARM720T is the ARM7TDMI. The ARM720T is software compatible with the ARM processor family. The ARM7TDMI is a member of the ARM family of general purpose 32bit microprocessors, which offers high performance for very low power consumption and price. This processor employs a unique architectural strategy known as THUMB, which makes it ideally suited to high volume applications with memory restrictions or applications where code density is an issue.
The key idea behind THUMB is a super reduced instruction set. Essentially, the ARM7TDMI has two instruction sets, the standard 32bit ARM set and 16bit THUMB set. The THUMB set's 16bit instruction length allows it to approach twice the density of standard ARM code while retaining most of the ARM`s performance advantage over a traditional 16bit processor by using 16bit registers. This is possible because THUMB code operates on the same 32bit register set as ARM code.

See also ARM720T Datasheet (ARM DDI 0087D) for detail.

2. Feature

- 32bit RISC architecture
- Low power consumption
- ARM7TDMI core with;
  - On-chip ICEbreaker debug support
  - 32bit x 8 hardware multiplier
  - Thumb decompressor
- Utilizes the ARM7TDMI embedded processor
  - High performance 32 bit RISC architecture
  - High density 16 bit instruction set
- Fully static operation : 0 ~ 80MHz
- 3-stage pipeline architecture (Fetch, decode, and execution stage)
- Enhanced ARM software toolkit
- MMU, Write Buffer, 8KB I/D Cache

THUMB code is able to provide up to 65% of the code size of ARM, and 160% of the performance of an equivalent ARM processor connected to a 16-bit memory system.

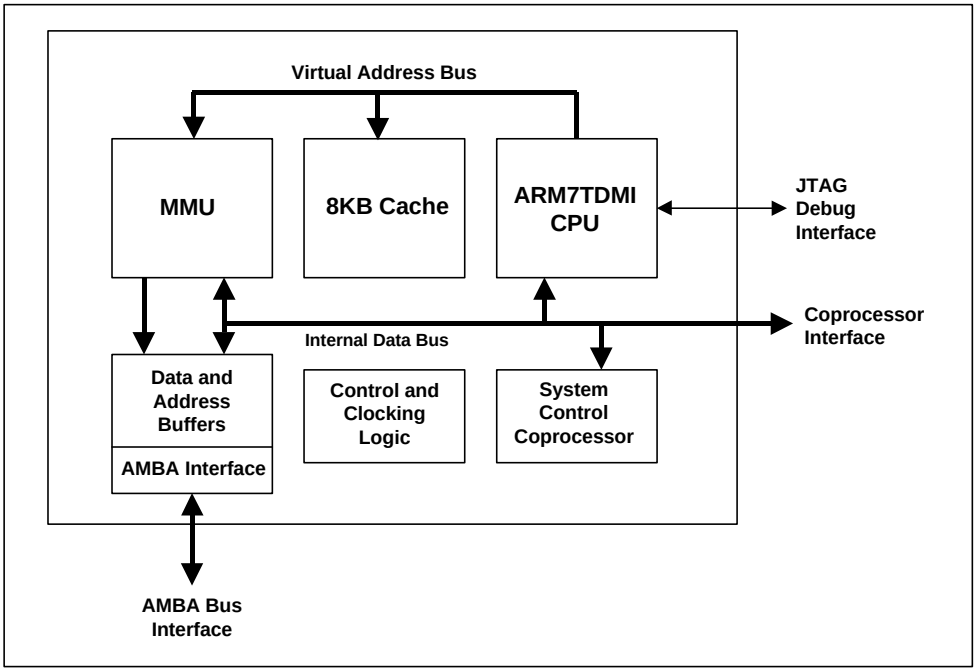


Figure 1. ARM720T Block Diagram

3. Core Block Diagram

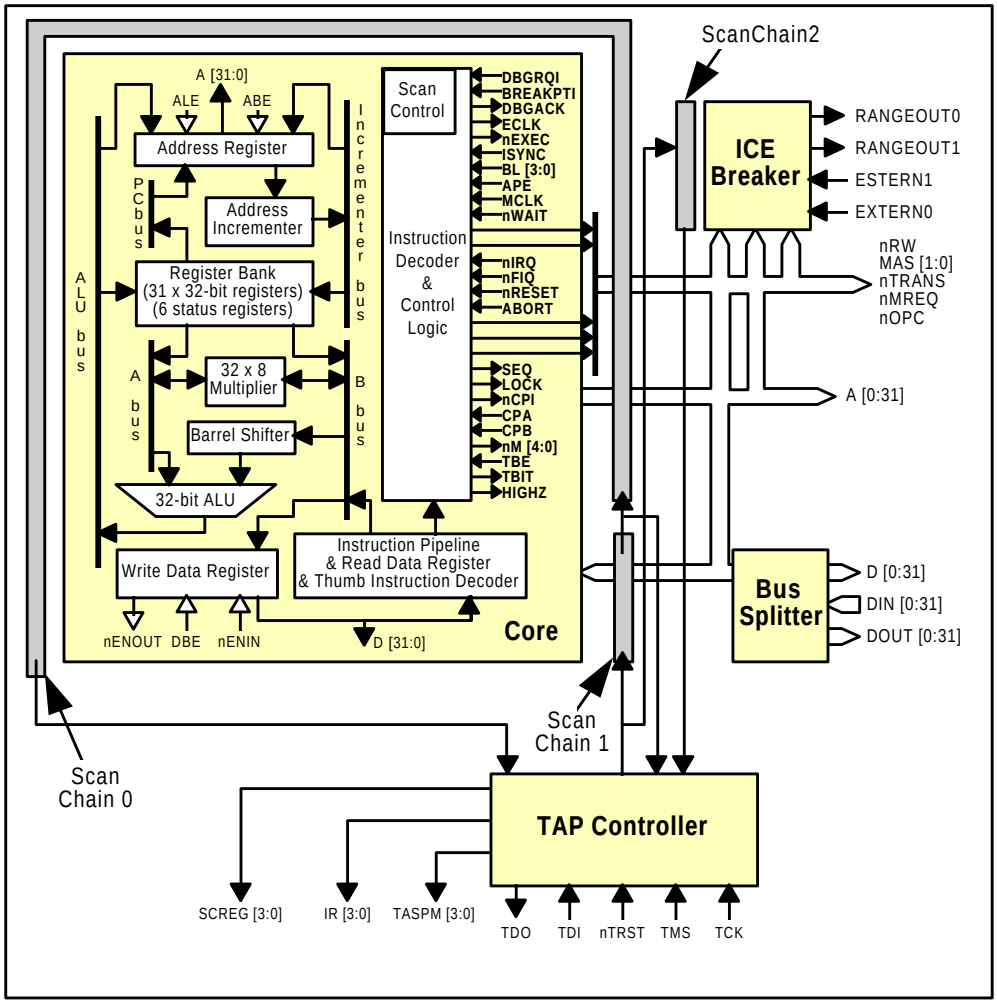


Figure 2. ARM7TDMI Core Block Diagram

Section 4. DRAM Controller

1. General Description

The DRAM controller interfaces the AMBA Advanced System Bus (ASB) to external DRAM memory banks. The DRAM controller provides the following features:

- Up to two banks of DRAM support.
- Fast page-mode sequential access support.
- EDO DRAM support
- Word, Half-word and Byte transaction support.
- Little / Big Endian Format support.
- DRAM refresh controller using CAS-before-RAS (CBR) refresh mode.
- Programmable refresh rate.
- Power-down mode where all DRAM accesses (including self-refresh) are disabled.
- Programmable DRAM timing control.
- Row/column addresses multiplexes according to DRAM capacity.

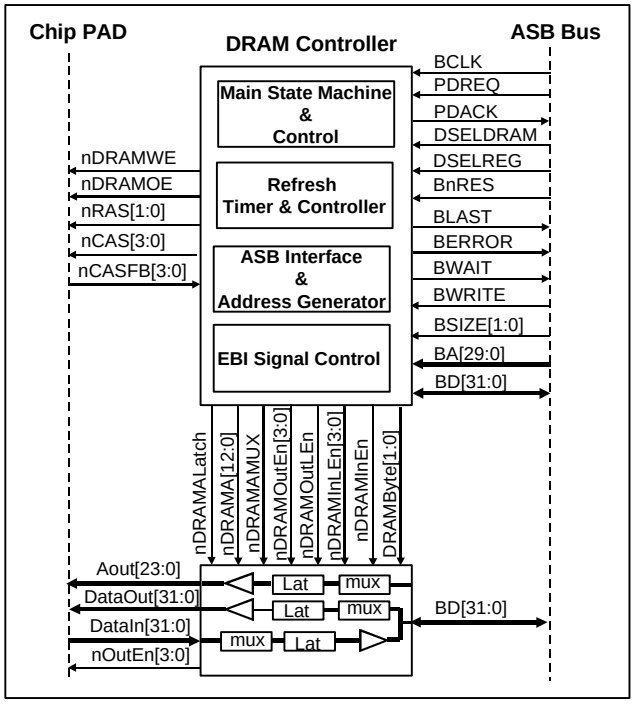


Figure 1. DRAM Controller Module Block Diagram

2. Hardware Interface and Signal Description

The DRAM Controller module is connected to the ASB bus. Table 1. DRAM interface ASB signal descriptions shows the internal bus interface signals to the DRAM controller.

Table 1. DRAM Interface ASB Signal Descriptions

NAME	DESCRIPTION
BA [27:0]	System address bus (excluding high order bits).
BCLK	The ASB clock timing all bus transfers.
BD [7:0]	Bidirectional system data bus.
BERROR	Error slave response signal. It is driven to phase 1 if the DRAM controller is selected. This signal will be asserted, when an access to the DRAM is attempted while the DRAM controller is in its *Power Down* mode.
BLAST	Last transfer of burst slave response signal. It can be driven to phase 1 if the DRAM controller is selected. It is asserted in order to indicate a 256-word boundary to force a non-sequential access.
BnRES	These signals indicate the reset status of the ASB.
BSIZE [1:0]	These signals indicate the size of the transfer that may be byte, half-word, or word.
BWAIT	Wait slave response signal. It is driven to phase 1 when the DRAM controller is selected. It is asserted while the DRAM transaction is uncompleted.
BWRITE	When this signal is HIGH, it indicates a write transfer and when LOW a read.
DSELDRAM	When this signal is HIGH, it indicates that the DRAM is selected.
DSELREG	When this signal is HIGH, it indicates that the DRAM configuration register is selected.

Table 2. DRAM interface External DRAM signal descriptions describes the DRAM controller connections to external devices of the system and to EBI (External Bus Interface) block .

Table 2. External DRAM Signal Descriptions

NAME	DESCRIPTION
nRAS[1:0]	Active LOW Row Address Strobes, one for each DRAM bank.
NCAS[3:0]	Active LOW Column Address Strobes, one for each byte.
NDRAMOE	Active LOW Output Enable.
NDRAMWE	Active LOW Write Enable.
nCASFB[3:0]	This is the nCAS[3:0] signal fed back from the output of the nCAS[3:0] pads.
PDREQ	Power Down Request. This signal indicates that the DRAM controller should enter into its low-power state, causing the DRAMs to enter into self-refresh state if refresh is enabled. When it is deasserted, the DRAM controller will exit from low power state.
PDACK	Power Down Acknowledge. This signal is asserted when the DRAM controller has successfully entered into its low-power mode. At this point BCLK may be stopped safely. It is deasserted when the DRAM controller has successfully exited from its low power state.
DRAMAMUX	DRAM Address Multiplex Select. When this signal is HIGH, it indicates to the EBI that the DRAMA[12:0] address should be used to generate DRAMA[12:0]. This signal provides the support for a shared EBI, and may not be needed in a system where the DRAM controller does not share the EBI with other memory controllers. DRAMAMUX is LOW when DRAM accesses are not performed.

NAME	DESCRIPTION
NDRAMALatch	DRAM Address Latch. When this signal is LOW, it opens the EBI address latch. This signal is HIGH when DRAM operations do not occur. This signal provides support for a shared EBI and may not be needed in a system where the DRAM controller does not share the EBI with other memory controllers.
DRAMA[12:0]	These multiplexed address lines are connected to the DRAM Address.
NDRAMInEn	DRAM Input Enable. When this signal is LOW, it enables the EBI drivers from latched XD to BD. This signal is HIGH when DRAM read operations are not performed.
NDRAMInLEn[3:0]	DRAM Input Latch Enable. When this signal is HIGH, it shuts the EBI latches on XD. This signal is LOW when DRAM read operations are not performed.
NDRAMOutEn	DRAM Output Enable. When this signal is HIGH, it disables the EBI drivers from latched BD to XD. This signal is low when DRAM write operations are not performed.
NDRAMOutLen	DRAM Output Latch Enable. When this signal is LOW, it opens the EBI latches on BD. This signal is HIGH when DRAM write operations are not performed.

Accesses to the DRAM Controller module are generated as a result of the address decode put out on the ASB address bus by the current bus master (which could be the ARM CPU or the DMA engine, for example).
The following three diagrams show the timing of the external interface for read, write and refresh cycles (Figure 2, 3, 4).

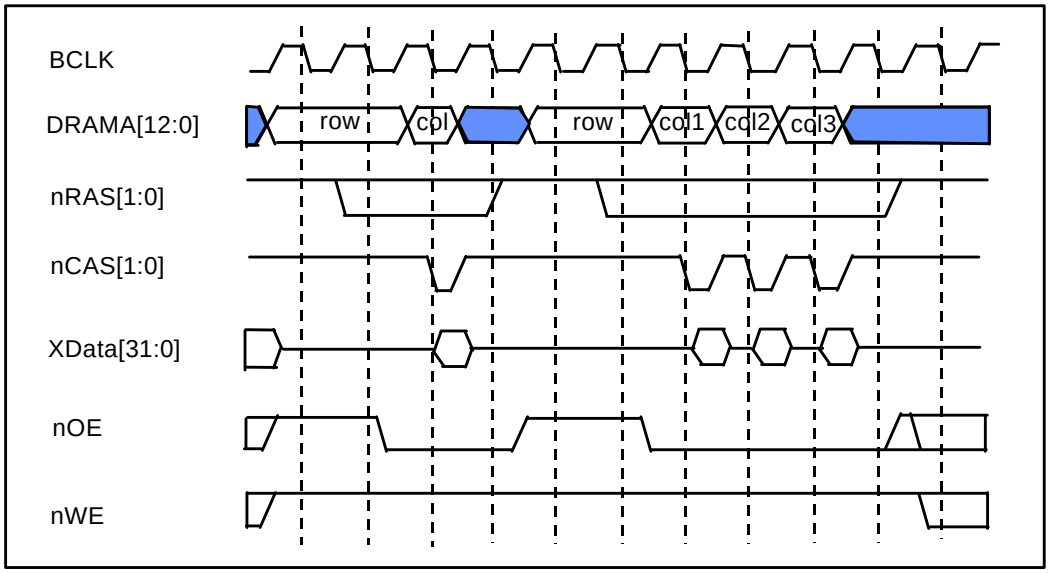


Figure 2. DRAM External Signal Timing: Read Cycles

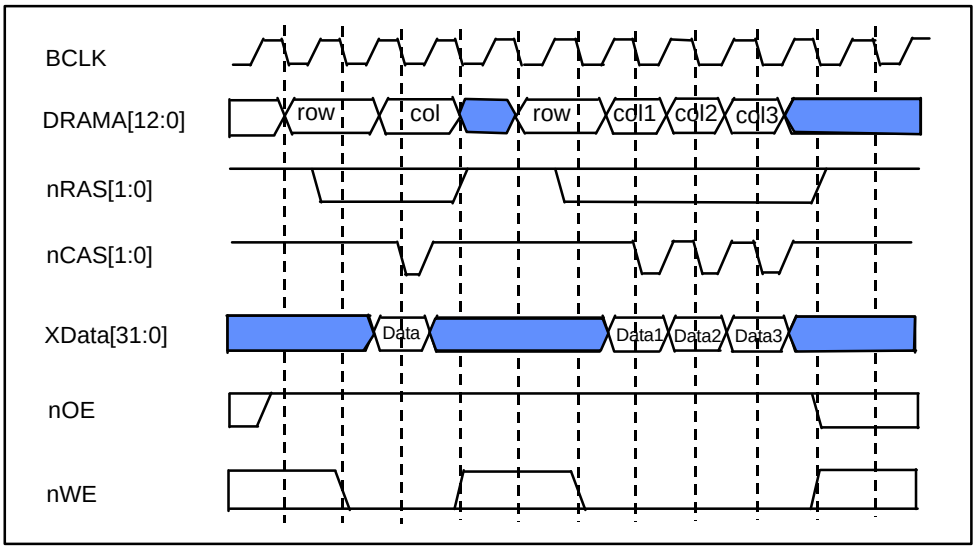


Figure 3. DRAM External Signal Timing: Write Cycles

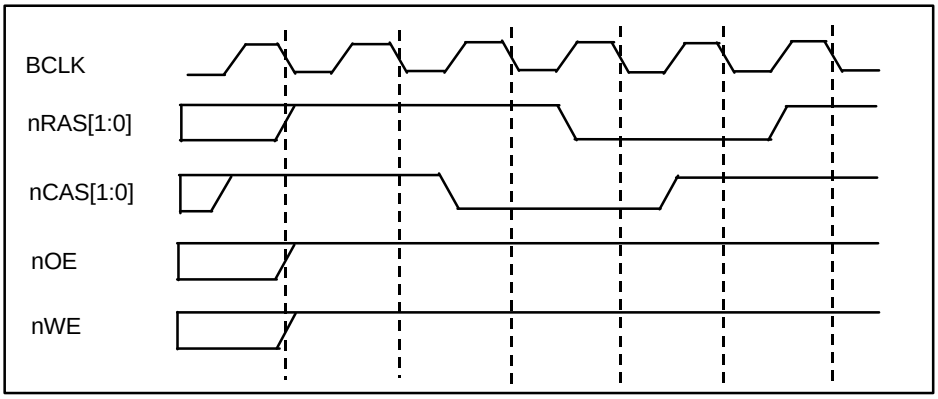


Figure 4. DRAM Controller Refresh Cycle

3. Functional Description

3.1 Introduction

The DRAM controller provides connections allowing a direct interface to up to two banks of DRAM. Each bank is 32/16/8 bits wide and up to 256MB in size. Two RAS lines are provided (one per bank) and four CAS lines (one per byte line).

3.2 Functional BreakDown

The DRAM controller consists of four main blocks: the Main State Machine & Control Block, the EBI Signal Control Block, the ASB Interface & Address Generation Block, and the Refresh Timer & Counter Block.

3.3 Main State Machine

This block contains the main DRAM timing control state machine and the decode for the external strobe signals for the DRAM interface. The state machine generates the timing for the nCAS and nRAS strobes, and the multiplexing of the DRAM row and column address lines for standard DRAM cycles and refresh cycles. The nDRAMWE and nDRAMOE signals are asserted appropriately depending on the access type. Word, Half-word, and Byte accesses are decoded from the lower bits of the BA address bus in order to assert the appropriate nCAS line(s). For word accesses all four nCAS lines are asserted. Figure 5. Descibes the Main State Machine Diagram.
Local arbitration for refresh cycles is also carried out here as refresh requests are received from the refresh timer block. The block also supports the self refresh DRAM; enter to and exit from this self refresh state are initiated by the PDRREQ signal. This is illustrated in Figure 6. DRAM signal timing: power down mode.

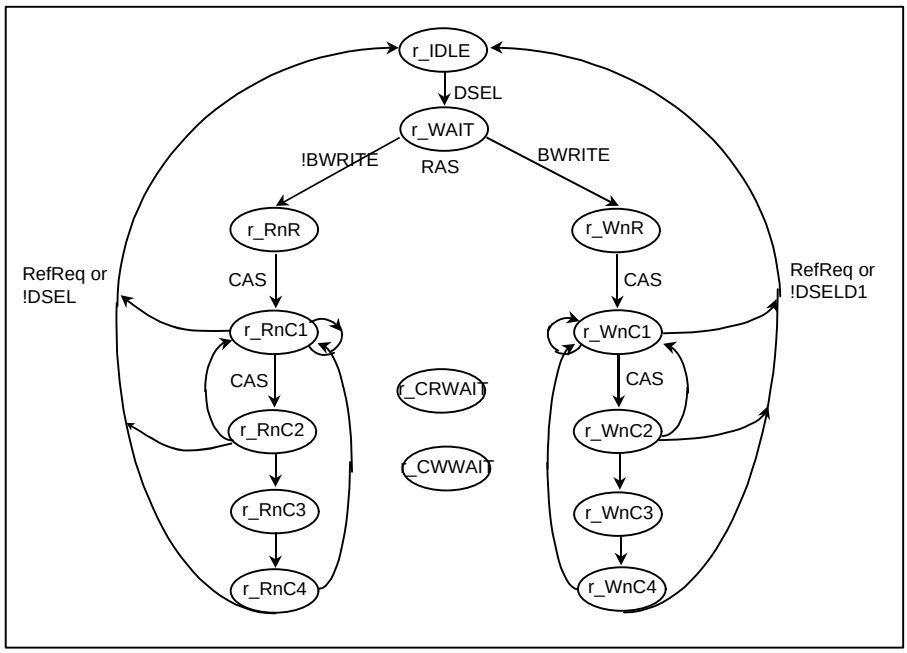


Figure 5. Main State Machine Diagram

3.4 EBI Control Block

This also generates the control signals required by the EBI(External Bus Interface). The EBI control signals are divided into three main groups; those related to the control of the Address path, the DataIn path, and the DataOut path.

Address Path Control

There are three signazls related to the address path of the EBI:

nDRAMALATCH — used to open the address latch of the EBI. This can be used to hold the external address XA while internal accesses are performed. When this signal is asserted (active LOW) the EBI address latch should be opened. When a DRAM access is not performed, the DRAM controller will de-assert this signal. In a shared EBI scheme, other memory controllers(Static Memory controller,...) must exhibit this behavior when they do not perform memory accesses.

DRAMAMUX — used to select the DRAMA[12:0] address as the address to be used on XA. This signal will be asserted (active HIGH) when a DRAM access occurs, and will be de-asserted when the transfer is completed.

DRAMA[12:0] — the multiplexed row/column address used to access the DRAM.

DataIn and DataOut Path Control

There are four signals related to the data path of the EBI:

nDRAMOUTEN — used to enable the EBIs data drivers onto XData. When this signal is de-asserted (HIGH), the EBI should disable its drive onto XData. This signal is de-asserted during read cycles and is asserted at other times. In a shared EBI scheme, other memory controllers must exhibit this behavior when they do not perform memory accesses.

nDRAMOUTLEN — used to latch the value of BD into the EBIs data output latches. When this signal is asserted (active LOW), the EBI data output latch is opened. This signal will be asserted during DRAM write transfers, and is de-asserted at other times.

nDRAMINEN — used to enable the EBI data drivers onto BD. When this signal is asserted (active LOW), the EBI should be driven onto BD. This signal is asserted during DRAM read transfers and is de-asserted at other times.

nDRAMINLEN[3:0] — used to latch the value of XData into the EBI data input latches. When this signal is de-asserted (HIGH), the EBI data input latch is shut. Four signals are provided to enable latching of byte / half-word data. nDRAMINLEN[0] is used to latch the data on M_D[7:0]. This signal is normally asserted and will be de-asserted during DRAM read transfers to latch the current data on XData.

3.5 Refresh Control Block

The refresh timer is a 7-bit timer counter which counts down and generates a refresh request when it reaches zero, at this point it is reloaded with the value in the refresh control register. This allows refresh frequencies from the Refresh Control Register and BCLK input clock.

3.6 ASB Interface Block

The ASB interface provides the interaction with the main AMBA bus. The DRAM controller will initiate a DRAM access when the DSELDRAM signal is asserted, or access the control registers when the DSELREG strobe is asserted. The timing of the ASB transfers is described in detail in the AMBA Specification rev. D. At a 256-word boundary, the BLAST signal will be asserted to indicate to the bus master that the burst sequence should be broken within the page boundary.
This block also generates the row and column addresses. During burst mode accesses, the column address is provided by a 10-bit column address incrementor to provide adequate column address timing.

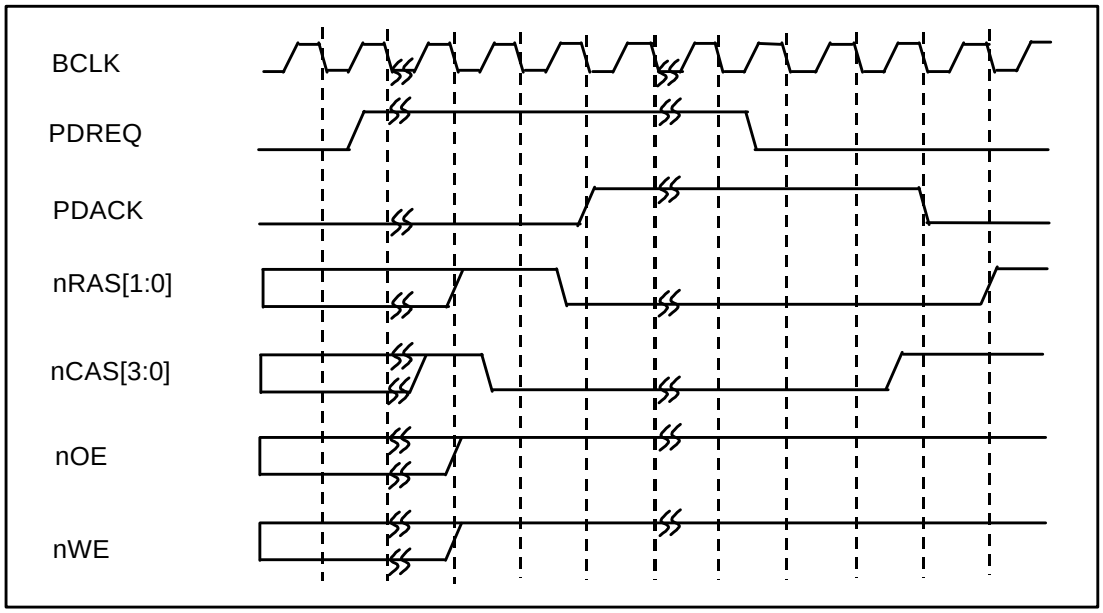


Figure 6. DRAM Signal Timing : Power Down Mode

4. Register Description

4.1 Memory Map

The base address (=DRAM REG Base) of the DRAM controller register bank is 0xFFFFED00.

Table 5. Memory Map of the Dram Controller Peripheral

ADDRESS	WRITE LOCATION	READ LOCATION	INITIAL
Base + 0x0	DRAM Refresh Control Register (RCR)	N/A	16'h0000
Base + 0x4	DRAM Control Register for CPU	DRAM Control Register for CPU	7'b0000000
Base + 0x8	DRAM Control Register for DMA	DRAM Control Register for DMA	6'b000000
Base + 0xC	DRAM Test Control Register (TCR)	N/A	4'b0000

4.2 DRAM Refresh Control Register(RCR)

The DRAM refresh period register is an 16-bit write register which enables the refresh and selects the refresh period used by the DRAM controller for its periodic CAS-before-RAS refresh. The value in the DRAM refresh period register is only cleared by a *Power On Reset* (BnRES = 0).

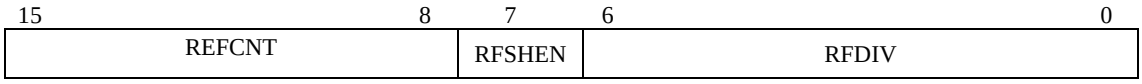


Figure 7. DRAM Controller Refresh Register

REFCNT DRAM Refresh Clock Divisor. Refresh Clock is setting by this bit field :

RefClock = BCLK/REFCNT

The REFCNT field should not be programmed with zero since this results in no initiated refresh cycles.

RFSHEN DRAM refresh enable. Setting this bit enables periodic refresh cycles to be generated by the DRAM controller at the rate set by the RFDIV field. Setting this bit also enables self-refresh mode when the DRAM controller is in the power down state.

RFDIV This 7-bit field sets the DRAM refresh rate. The refresh period is deriven from internally generated clock and is given by the following formula:

Frequency (KHz) = 2*[RefClock /(RFDIV + 1)]

or

RFDIV = (RefClock / 0.5*Refresh frequency (KHz)) - 1

The RFDIV field should not be programmed with zero since this results in no initiated refresh cycles.

4.3 DRAM Control Register for CPU (DRAMConCPU)

This Register controls the DRAM control signals when DRAM accessed by CPU. In normal condition, the DRAM access time is changed by the bus master is CPU or DMA Controller. In case of bus master is DMA Controller, the transfer timing should be properly set to the external I/O device and DRAM, so for the optimal system performance the DRAM access by the CPU is set in this DRAM Control Register for CPU (DRAMConCPU) and in case of the DRAM access by the DMA Controller DRAM control signals are controlled by the DRAM Control Register for DMA (DRAMConDMA).

15 7	6	5	4	3 2	1 0
Reserved	DMAEn	TRP	TCP	WaitCnt	BankSize

Figure 8. DRAM Control Register for CPU (DRAMConCPU)

DMAEn If DMA transfer, then the DRAM control signals are controlled by DRAM Control Register for DMA (DRAMDonDMA) by this bit setting. When this bit is '0', then the DRAM control signals are controlled by bit fields in this Control Register (DRAMConCPU) during DRAM access.

TRP Control the timing of difference between the RAS and CAS signal by this bit field setting. When this bit is '0', then DRAM access are absolutely no wait, so DRAM access time is very short, but should be considered the operating frequency of the MCU and DRAM access time.

TCP Control the timing of the Low phase of CAS signals. When this bit is '1', then the Low phase of the CAS signals are enlarged to one cycle of BCLK. When this bit is '0', then the Low phase of the CAS signals are half clock of BCLK.

WaitCnt This bit fields control the DRAM access time. The wait state is inserted in ASB BUS by the value of these WaitCnt fields. (00=0-wait, 01=1-wait, 10=2-wait, 11=3-wait)

BankSize These bits indicate the data width of the DRAM Bank. The data width of the DRAM by BankSize are shown Table 6.

Table 6. Data width of the DRAM by BankSize[1:0] fields

BankSize[1:0]	Data Width of DRAM
00	Byte
01	Half Word
10	Word
11	Reserved

4.4 DRAM Control Register for DMA

This Register controls the DRAM control signals when DRAM accessed by DMA.
Setting the register is effective only when the DMAEn bit set by DRAMConCPU(DRAM Control register from CPU).

15 6	5	4	3 2	1 0
Reserve	TRP	TCP	WaitCnt	BankSize

Figure 9. DRAM Control Register for DMA (DRAMConDMA)

4.5 DRAM Test Control(TCR)

The DRAM test control register is for test and should not be used during normal operation. It is a write-only register with the following format.

15 4	3	2	1 0
Reserved	TESTINC	FORCEADV	FORCESIZE

Figure 10. DRAM Test Control Register

TESTINC — Test increment (TESTINC). This bit puts the column address increment into a test mode. In this mode each nibble of the column address increment increments independently. Resets it to 0.

FORCEADV — Force refresh advance (FORCEREFADV). This bit forces the refresh counter to advance every BCLK. Resets it to 0.

FORCESIZE[1:0] — Force access size. These bits force the size of accesses to the DRAM bank. When this is set to 10 (default), the ASB B_SIZE is used to determine the size of the access. When this is set to 00 or 01, a byte or half-word access is forced respectively. Resets it to 10.

Section 5. On-Chip SRAM

1. General Description

The GDC21D601 has 8-kbytes of on-chip RAM. The on-chip RAM is linked to the CPU and direct memory access controller(DMAC) with 32-bit data bus. The CPU and DMA Controller can write data into the on-chip RAM in byte, half-word, or word units.

2. Signal Description

Table 1. Signal Descriptions

NAME	TYPE	DESCRIPTION
BA[31:0]	I	System address bus.
BD[31:0]	I/O	Bi-directional system data bus.
BWAIT	I/O	LOW during phase one of BCLK.
BLAST	I/O	LOW during phase one of BCLK.
BERROR	I/O	LOW during phase one of BCLK.
BWRITE	I	When this signal is HIGH, it indicates a write transfer and when LOW a read.
DSELMEM	I	When this signal is HIGH, it indicates that on-chip RAM is selected.
BnRES	I	These signals indicate the reset status of the ASB.

3. Function Description

On-Chip SRAM can read data from SRAM and can write data into SRAM in a single clock cycle through ASB bus. And SRAM is single module which have 32 bit data bus and control lines.
The data in the On-chip RAM can always be accessed in one cycle that make the RAM ideal for use as a program area, stack area, or data area, which requires high-speed access. The contents of the on-chip RAM are held in both standby and power-down modes.
Memory area 0x10000000 to 0x10001FFF is allocated to the on-chip RAM as default. When isram signal from MCU Controller is set to HIGH, memory area 0x00000000 to 0x00001FFF can be allocated to the on-chip RAM.

Section 6. Static Memory Controller

1. General Description

The Static Memory Controller interfaces the AMBA Advanced System Bus (ASB) to the External Bus Interface (EBI); controlling the external SRAM, ROM, Flash Memory or off-chip peripherals. Eight separate chip select banks are provided by this block. Each bank is 256MB in size and can be programmed individually to support:

- 8-, 16- or 32-bit wide, Little-Endian and Big-Endian Memory Format
- variable wait states (up to 16 waits)
- exchangeable active low/high chip select signal (only for CS6 and CS7)
- various type control signal timing
- bus transfers can be extended using the EXPRDY input signal. EXPRDY signal can be used by exchangeablely active HIGH or LOW in according to control register setting.

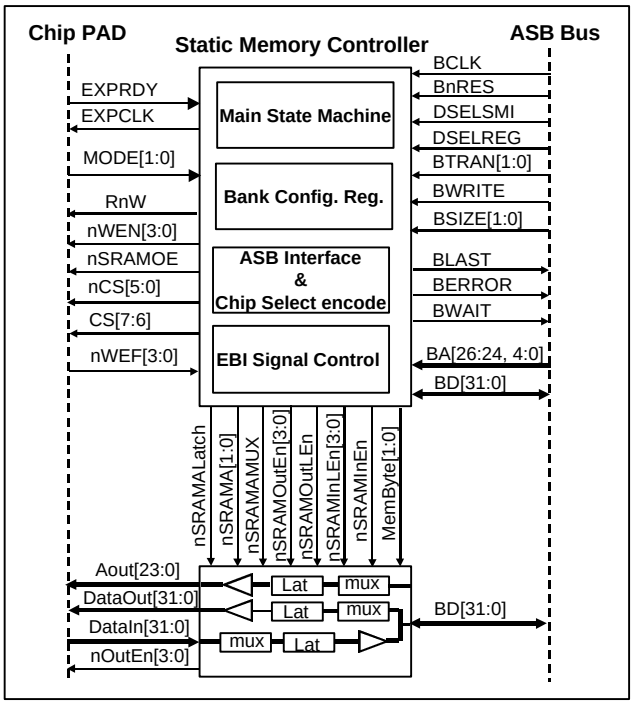


Figure 1. Static Memory Controller Block Diagram

2. Signal Description

The Static Memory Controller module is connected to the ASB bus. In Table 1. Static Memory Controller ASB signal descriptions show the internal bus interface signals(AMBA signals) to the Static Memory Controller

Table 1. Static Memory Controller ASB Signal Descriptions

NAME	TYPE	DESCRIPTION
BA[26:24, 4:0]	I	System address bus. The SRAM controller only requires seven bits of this bus to do the necessary encoding/decoding.
BCLK	I	The ASB clock.
BD[31:0]	I/O	Bi-directional system data bus. The data bus is driven by this block during read transfers from configuration registers only.
BERROR	O	LOW during phase one of BCLK when the Static Memory Controller is selected.
BLAST	O	LOW during phase one of BCLK when the Static Memory Controller is selected.
BWAIT	O	This slave response is driven during phase one of BCLK when the Static Memory Controller is selected and is used to indicate if the memory has completed its current transfer.
BnRES	I	The reset status of the ASB.
BSIZE[1:0]	I	The size of the transfer data which may be byte, half-word, or word.
BTRAN[1:0]	I	These signals are used to determine sequential and non-sequential accesses.
BWRITE	I	When this signal is HIGH, it indicates a write transfer and when LOW a read.
DSELSRAM	I	When this signal is HIGH, it indicates that the Static Memory Controller is selected.
DSELREG	I	When HIGH, this signal indicates that one of the Bank Configuration registers is selected.

Table 2. Static Memory Controller External Signal Descriptions

NAME	TYPE	DESCRIPTION
EXPRDY	I	Expansion channel ready. This signal is active LOW by default, When this signal is LOW, it will force the current memory transfer to be extended. When the RDON bit field in Configuration Register is set, then the polarity of the EXPRDY signal is reversed to active HIGH.
EXPCLK	O	Expansion clock output. Clock output at the same phase and speed as the bus clock. Active only during SRAM/ROM cycles.
nWEN[3:0]	O	These signals are active LOW write enables for each of the memory byte lanes on the external bus. For example nWEN[0] controls the writes to D[7:0].
nWEF[3:0]	I	These optional connections use PADs feedback from the external side of the nWEN[3:0] PADs. They are used to guarantee address and chip select hold time when any write enable is LOW. If not used, they should be tied to HIGH.
NSRAMOE	O	This is the active LOW output enable for devices on the external bus. This is LOW during reads from external memory and during the time that the selected bank should drive the external data bus.
nCS[5:0]	O	Active LOW Chip Select
CS[7:6]	O	Active HIGH Chip Select
SRAMA [1:0]	O	These signals form the lower two bits of the external address bus. They are used to control accesses to 16- or 8-bit memories when the AMBA bus requests an access size larger than the memory (this is handled using multiple external transfers).
nSRAMALatch	O	This signal is an active LOW transparent address latch enable. It is normally HIGH to prevent power wasting transitions on the external address bus.
MemByteSeq[1:0]	O	These signals control the data path muxes which allow 16- or 8-bit memories to read and write 32-bit values on the AMBA bus.
nSRAMOutLEn	O	Active LOW transparent latch enable for the data out path (writes).
nSRAMOutEn[1:0]	O	Active LOW byte lane data output driver enable.
nSRAMInLEn[1:0]	O	Active LOW transparent latch enable for the data in path (reads).
nSRAMInEn	O	Active LOW data input driver enable (to AMBA bus).
Mode[1:0]	I	Booting mode configuration input. If these signals are “00” during BnRES LOW then the SRAM Controller will select bank zero (nCS[0]) as 32-bit memory. If these signals are “10” then select bank zero as 16-bit memory. If these signals are “01” then bank zero as 8-bit memory.

Accesses to the Static Memory Controller module can be two basic types; control register accesses and memory area accesses. The following timing diagrams relate to the external pin timings for SRAM/ROM read and write cycles in minimum wait states.

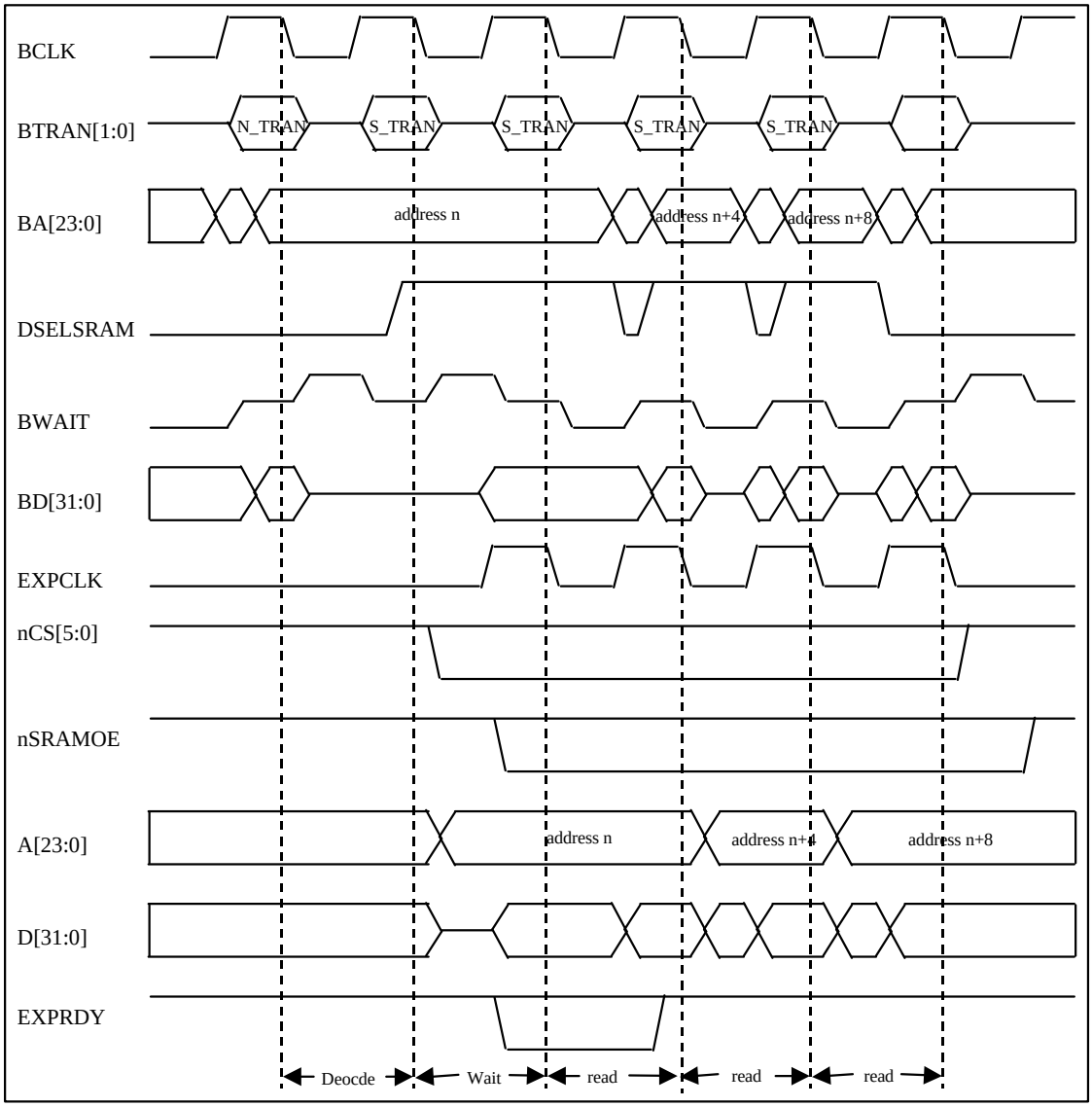


Figure 2. ROM Read Timing

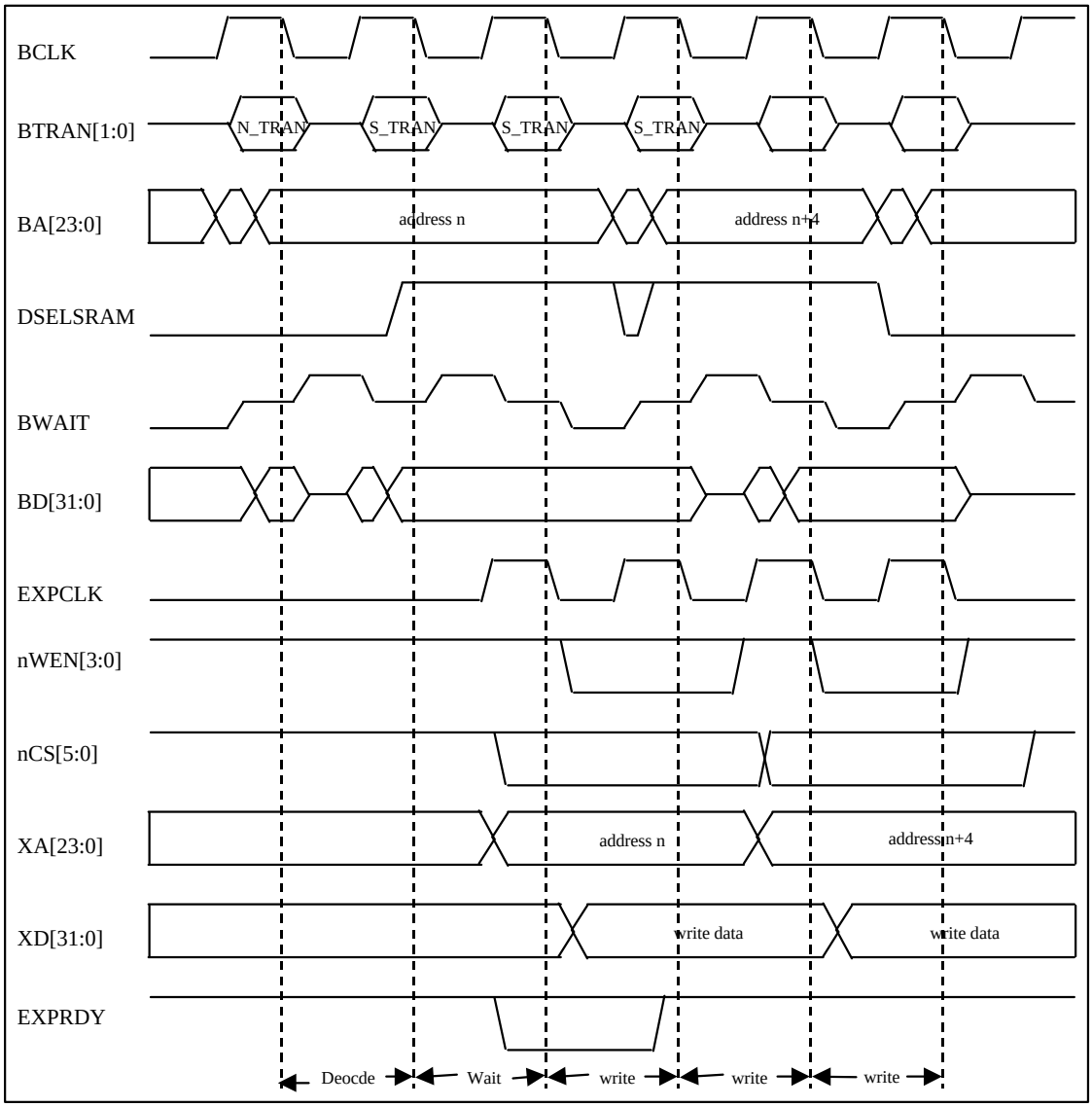


Figure 3. SRAM Write Timing

3. Functional Description

The Static Memory Controller has following functions:

- memory bank select
- off-chip expansion clock driver
- wait states generation
- byte lane write control
- burst read access
- various type control signal generation

These are described below.

3.1 Memory Bank Select

The chip select signal generation is controlled by BA[26:24]. From Table 3 static memory bank select coding is shown that these signals coded to CS[7:6] and nCS[5:0].

Table 3. Static Memory Bank Select Coding (MODE R)

DSEL	BA[26:24]	CS[7:6]	nCS[5:0]	MEMORY CONFIGURATION
1	000	00	111110	nCS0 configuration
1	001	00	111101	nCS0 configuration
1	010	00	111011	nCS2 configuration
1	011	00	110111	nCS3 configuration
1	100	00	101111	nCS4 configuration
1	101	00	011111	nCS5 configuration
1	110	01	111111	CS6 configuration
1	111	10	111111	CS7 configuration

3.2 Off-Chip Expansion Clock Driver

In the Static Memory Controller, the system clock input BCLK is passed directly to EXPCLK during memory cycles if the expansion clock enable bit of the corresponding memory bank configuration is set.

3.3 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. For example, in case that bank zero is configured as 8-bit wide memory and a 32-bit read is initiated, the ASB bus will stall while the SRAM Controller reads four consecutive bytes from the memory. During these accesses the data path is controlled (using the MemByteSeq[1:0] signals) to de-multiplex these four bytes into one 32-bit word on the ASB bus.

3.4 Wait State 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 reads from ROMs.

Note
Wait state control refers to external transfer wait states. The number of cycles where an AMBA transfer completes is controlled by two other factors; access width and external memory width. The Static Memory Controller also allows transfers to be extended indefinitely, by asserting EXPRDY to LOW. To hold the current transfer EXPRDY must be asserted on the falling edge of BCLK before the last cycle of the access. The transfer cannot be completed until EXPRDY is HIGH for at least one cycle.

3.5 Burst Read Control

This supports sequential access burst reads of up to four consecutive locations in 8-, 16- or 32-bit memories. This feature supports burst mode ROM devices and increases the bandwidth by using a reduced (configurable) access time for three sequential reads following a quad-location boundary read. (Note that quad-location boundaries occur when A[1:0]=00 for byte wide memories.)

3.6 Byte Lane Write Control

This controls nWEN[1:0] according to AMBA transfer width (indicated by BSIZE[1:0]), external memory width, BA[1:0], and the access sequencing. The following table shows the basic coding assuming 32-bit external memory:

Table 4. nWEN Coding

BSIZE[1:0]	BA[1:0]	nWEN[3:0]
10 (word)	XX	0000
01 (half-word)	1X	0011
01 (half-word)	0X	1100
00 (byte)	11	0111
00 (byte)	10	1011
00 (byte)	01	1101
00 (byte)	00	1110

4. Programmer' s Model

4.1 Memory Map

The base address for the Static Memory Controller registers is 0xFFFFEC00

Table 5. Static Memory Controller Memory Map

ADDRESS	DESCRIPTION	INITIAL VALUE
SRAMRegBase + 00000	Memory Configuration Register 1 (MEMCFG1)	32'h00000004
SRAMRegBase + 00004	Memory Configuration Register 2 (MEMCFG2)	32'h00000000
SRAMRegBase + 00008	Memory Configuration Register 3 (MEMCFG3)	32'h00000000
SRAMRegBase + 0000C	Memory Configuration Register 4 (MEMCFG4)	32'h00000000

4.2 Memory Configuration Registers

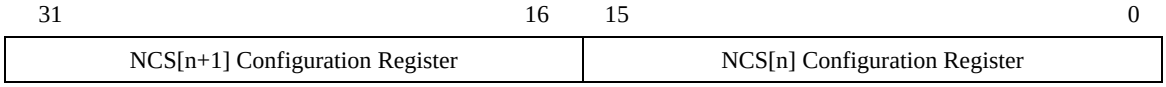


Figure 4. Memory Configuration Register

Memory configuration register (MEMCFG1, 2, 3, 4) is a 32-bit read-write register which sets the configuration of the two expansion and ROM selects. Each select is configured with a two-byte field.

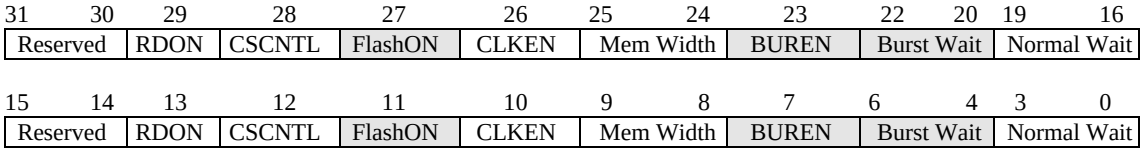


Figure 5. Two-Byte Fields in the Memory Configuration Register for CS[5:0]

(Note : Gray areas are reserved for another feature.)

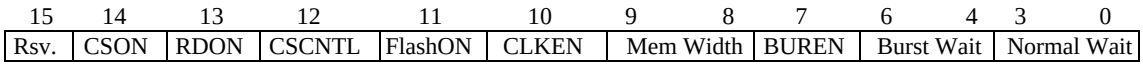


Figure 6. Two-Byte Fields in the Memory Configuration Register for CS[6]

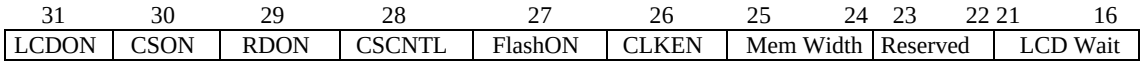


Figure 7. Two-Byte Fields in the Memory Configuration Register for CS[7]

LCDON	LCD enable. When the Bank 7 is connected to LCD panel for text display, setting this bit enables LCD wait to access directly LCD device. LCD wait bit is 6 bits therefore wait cycle is from 1 to 64.
CSON	nCS enable. Setting this bit is enables the CS6 and CS7 to be active low signal from active high signal that supports various devices.
RDON	select the polarity of EXPRDY. When this bit is set to 0, EXPRDY signal act as positive active signal. When this bit is set to 1, EXPRDY signal act as negative active signal.
CSCNTL	Make the control signals (Address, Data, CS, RnW, etc.) of external device to be similar Motorolar type CPU.
FlashON	Flash memory enable. When this bit is set to 1, memory control signals, nCS, nWEN[1:0], and nSRAMOE, are adjusted to flash memory control signal timing.
CLKEN	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.
BUREN	Burst enable. Setting this bit enables burst reads to take advantage of faster access time from ROM devices that support burst mode.

Note

Banks using EXPCLK and EXPRDY for off-chip peripheral control should not enable burst mode, and should be designed and set up to use a specific number of wait states in each access. The peripheral should time the access by counting EXPCLK cycles (there is no explicit indication of access start or end) and determine the access direction and width by using nWEN[3:0].

Table 6. Values of the Mem Width Field Define the Bus Width Field.

Table 6. Values of the Mem Width Field

MEM WIDTH FIELD	EXPANSION TRANSFER MODE
00	32-bit wide bus access
01	16-bit wide bus access
10	8-bit wide bus access
11	Reserved

Table 7. The values of the Normal Wait field define the values of the normal access wait state field. And the values of the LCD Wait define likely as Table 7.

Table 7. Values of the Normal Access Wait State Field.

VALUE	NUMBER OF WAIT STATES
0000	1
0001	2
0010	3
0011	4
0100	5
0101	6
0110	7
0111	8
1000	9
1001	10
1010	11
1011	12
1100	13
1101	14
1110	15
1111	16

Table 8. Values of the Burst Wait field define the values of the burst read wait state field.

Table 8. Values of the Burst Read Wait State Field

VALUE	NUMBER OF WAIT STATES
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Section 7. MCU Controller

1. General Description

Designing the Microcontroller unit (MCU), some control signals needed by any functional block, but not drive any other block, must be generated. So these control signals are generated in MCU Controller. The MCU Controller (MCUC) is composed of registers which are for selecting the function of multi-function pins, for defining the memory map structure, arbiter priority, MCU device code, and DRAM Power Down Req/Ack signals.

2. Signal Description

Table 1. Signal Descriptions

NAME	TYPE	DESCRIPTION
BCLK	I	System bus clock.
BnRES	I	the reset status of the ASB
BA[31:0]	I	System address bus
BD[31:0]	I/O	Bi-directional system data bus.
BWAIT	O	Low during phase one of BCLK
BLAST	O	Low during phase one of BCLK
BERROR	O	Low during phase one of BCLK
BWRITE	I	When this signal is HIGH, it indicates a write transfer and when LOW a read.
DSEL	I	When this signal is HIGH, it indicates that MCU Controller is selected.
PwrDwnAck	I	This signal indicates that DRAM is entered into self-refresh mode
PwrDwnReq	O	The request of entering the self-refresh node of DRAM
Ari_pri	O	Determine Arbiter Priority. See Section.2 System Architecture for detail
Isram	O	Allocate On-Chip SRAM address area at 0x00000000
Drambank0	O	Allocate DRAM address area at 0x00000000
PINMUX_sigs	O	These signals are for Multi-function pin

3. Register Description

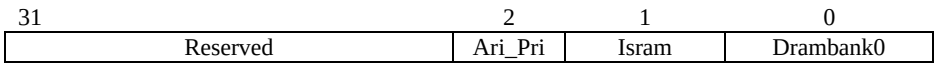
3.1 Register Memory Map

The base address of MCU control Register is 0xFFFFEB00.

Table 2. MCU Controller Memory Map

ADDRESS	R/W	INITIAL VALUE	DESCRIPTION
MCURegBase + 0x0000	R/W	0x00	MCU Control Register
MCURegBase + 0x0004	R/W	0x00	PINMUX_PA Register, Multi-function pin MUX Control signals for Port A[5:0]
MCURegBase + 0x0008	R/W	0x00	PINMUX_PB Register, Multi-function pin MUX Control signals for Port B[7:0]
MCURegBase + 0x000C	R/W	0x00	PINMUX_PC Register, Multi-function pin MUX Control signals for Port C[7:0]
MCURegBase + 0x00010	R/W	0x00	PINMUX_PD Register, Multi-function pin MUX Control signals for Port D[7:0]
MCURegBase + 0x00014	R/W	0x00	PINMUX_PE Register, Multi-function pin MUX Control signals for Port E[8:0]
MCURegBase + 0x00018	R/W	0x00	PINMUX_PF Register, Multi-function pin MUX Control signals for Port F[8:0]
MCURegBase +0x0001C	R/W	0x00	PINMUX_PG Register, Multi-function pin MUX Control signals for Port G[7:0]
MCURegBase + 0x00020	R/W	0x00	PINMUX_PH Register, Multi-function pin MUX Control signals for Port H[7:0]
MCURegBase + 0x00024	R/W	0x00	PINMUX_PI Register, Multi-function pin MUX Control signals for Port I[7:0]
MCURegBase + 0x00028	R/W	0x00	PINMUX_PJ Register, Multi-function pin MUX Control signals for Port J[7:0]
MCURegBase +0x0002C	R	$LG601	MCU Device Code Register
MCURegBase + 0x00030	R	0x0	DRAM Power Down Ack
MCURegBase + 0x00034	W	0x0	DRAM Power Down Req

3.2 MCUC_CON Register



drambank0	When this register is HIGH, DRAM memory address bank #0 area is located at 0.
Isram	When this register is HIGH, On-Chip SRAM address area is located at 0
Ari_pri	Arbiter Priority control signal. See also Section 2 System Architecture for details.

Figure 1. MCU Controller Register

3.3 PINMUX Register

Table 3. PINMUX_PA Register

BIT NO.	SIGNAL NAME	PIN FUNCTION DESCRIPTION		
		When 0	When 1	PIN No.
0	PINMUX_PA[0]	IRQ0	PA0	45
1	PINMUX_PA[1]	IRQ1	PA1	47
2	PINMUX_PA[2]	IRQ2	PA2	48
3	PINMUX_PA[3]	IRQ3	PA3	49
4	PINMUX_PA[4]	IRQ4	PA4	51
5	PINMUX_PA[5]	IRQ5	PA5	52

Table 4. PINMUX_PB Register

BIT NO.	SIGNAL NAME	PIN FUNCTION DESCRIPTION		
		When 0	When 1	PIN No.
0	PINMUX_PB[0]	TCIOA0	PB0	55
1	PINMUX_PB[1]	TCIOB0	PB1	57
2	PINMUX_PB[2]	TCIOA1	PB2	58
3	PINMUX_PB[3]	TCIOB1	PB3	59
4	PINMUX_PB[4]	TCIOA2	PB4	61
5	PINMUX_PB[5]	TCIOB2	PB5	62
6	PINMUX_PB[6]	TCIOA3	PB6	63
7	PINMUX_PB[7]	TCIOB3	PB7	65

Table 5. PINMUX_PC Register

BIT NO.	SIGNAL NAME	PIN FUNCTION DESCRIPTION		
		When 0	When 1	PIN No.
0	PINMUX_PC[0]	PC0	TCIOA4	66
1	PINMUX_PC[1]	PC1	TCIOB4	67
2	PINMUX_PC[2]	PC2	TCIOA5	69
3	PINMUX_PC[3]	PC3	TCIOB5	70
4	PINMUX_PC[4]	PC4	TCLKA	71
5	PINMUX_PC[5]	PC5	TCLKB	73
6	PINMUX_PC[6]	PC6	TCLKC	74
7	PINMUX_PC[7]	PC7	TCLKD	75

Table 6. PINMUX_PD Register

BIT NO.	SIGNAL NAME	PIN FUNCTION DESCRIPTION		
		When 0	When 1	PIN No.
0	PINMUX_PD[0]	RXD0	PD0	76
1	PINMUX_PD[1]	TXD0	PD1	78
2	PINMUX_PD[2]	RXD1	PD2	79
3	PINMUX_PD[3]	TXD1	PD3	80
4	PINMUX_PD[4]	NCTS	PD4	82
5	PINMUX_PD[5]	NDSR	PD5	83
6	PINMUX_PD[6]	NDCD	PD6	84
7	PINMUX_PD[7]	NRI	PD7	86

Table 7. PINMUX_PE Register

BIT NO.	SIGNAL NAME	PIN FUNCTION DESCRIPTION		
		When 0	When 1	PIN No.
0	PINMUX_PE[0]	NDTR	PE0	87
1	PINMUX_PE[1]	NRTS	PE1	88
2	PINMUX_PE[2]	SMDI	PE2	90
3	PINMUX_PE[3]	SMDO	PE3	91
4	PINMUX_PE[4]	SMCLK	PE4	92
5	PINMUX_PE[5]	SIN0	PE5	94
6	PINMUX_PE[6]	SOUT0	PE6	95
7	PINMUX_PE[7]	SCLK0	PE7	96

Table 8. PINMUX_PF Register

BIT NO.	SIGNAL NAME	PIN FUNCTION DESCRIPTION		
		When 0	When 1	PIN No.
0	PINMUX_PF[0]	SCS0	PF0	98
1	PINMUX_PF[1]	SIN1	PF1	99
2	PINMUX_PF[2]	SOUT1	PF2	100
3	PINMUX_PF[3]	SCLK1	PF3	102
4	PINMUX_PF[4]	SCS1	PF4	103
5	PINMUX_PF[5]	BCLKOUT	PF5	104
6	PINMUX_PF[6]	NFIQOUT	PF6	105
7	PINMUX_PF[7]	NIRQOUT	PF7	106

Table 9. PINMUX_PG Register

BIT NO.	SIGNAL NAME	PIN FUNCTION DESCRIPTION		
		When 0	When 1	PIN No.
0	PINMUX_PG[0]	EXTREQ	PG0	124
1	PINMUX_PG[1]	EXTACK	PG1	127
2	PINMUX_PG[2]	DREQ0	PG2	128
3	PINMUX_PG[3]	DACK0	PG3	129
4	PINMUX_PG[4]	DREQ1	PG4	131
5	PINMUX_PG[5]	DACK1	PG5	132
6	PINMUX_PG[6]	RAS0	PG6	133
7	PINMUX_PG[7]	RAS1	PG7	135

Table 10. PINMUX_PH Register

BIT NO.	SIGNAL NAME	PIN FUNCTION DESCRIPTION		
		When 0	When 1	PIN No.
0	PINMUX_PH[0]	CAS0	PH0	136
1	PINMUX_PH[1]	CAS1	PH1	137
2	PINMUX_PH[2]	CAS2	PH2	138
3	PINMUX_PH[3]	CAS3	PH3	139
4	PINMUX_PH[4]	CS4	PH4	161
5	PINMUX_PH[5]	CS5	PH5	162
6	PINMUX_PH[6]	CS6	PH6	165
7	PINMUX_PH[7]	CS7	PH7	166

Table 11. PINMUX_PI Register

BIT NO.	SIGNAL NAME	PIN FUNCTION DESCRIPTION		
		When 0	When 1	PIN No.
0	PINMUX_PI[0]	D16	PI0	187
1	PINMUX_PI[1]	D17	PI1	186
2	PINMUX_PI[2]	D18	PI2	184
3	PINMUX_PI[3]	D19	PI3	183
4	PINMUX_PI[4]	D20	PI4	182
5	PINMUX_PI[5]	D21	PI5	180
6	PINMUX_PI[6]	D22	PI6	179
7	PINMUX_PI[7]	D23	PI7	178

Table 12. PINMUX_PJ Register

BIT NO.	SIGNAL NAME	PIN FUNCTION DESCRIPTION		
		When 0	When 1	PIN No.
0	PINMUX_PJ[0]	D24	PJ0	177
1	PINMUX_PJ[1]	D25	PJ1	175
2	PINMUX_PJ[2]	D26	PJ2	174
3	PINMUX_PJ[3]	D27	PJ3	173
4	PINMUX_PJ[4]	D28	PJ4	171
5	PINMUX_PJ[5]	D29	PJ5	170
6	PINMUX_PJ[6]	D30	PJ6	169
7	PINMUX_PJ[7]	D31	PJ7	167

3.5 MCU Device Code Register

This Register is read only.
Device Code Value is '$LG601'
Binary Value : 0000 0100 1100 0100 0111 0110 0000 0001

3.6 DRAM Power Down Acknowledge Register

This Register is 1 bit read only register.
This register is set when DRAM is entered to power down mode.

3.7 DRAM Power Down Request Register

This Register is 1 bit read/write register.
When this register bit is HIGH, request to DRAM Controller to enter into power down mode of the DRAM.

Section 8. Power Management Unit

1. General Description

The PMU block provides:

- Clock distribution of all over system
- Reset, RUN and Power down modes control

Figure 1. shows the PMU Block Diagram.

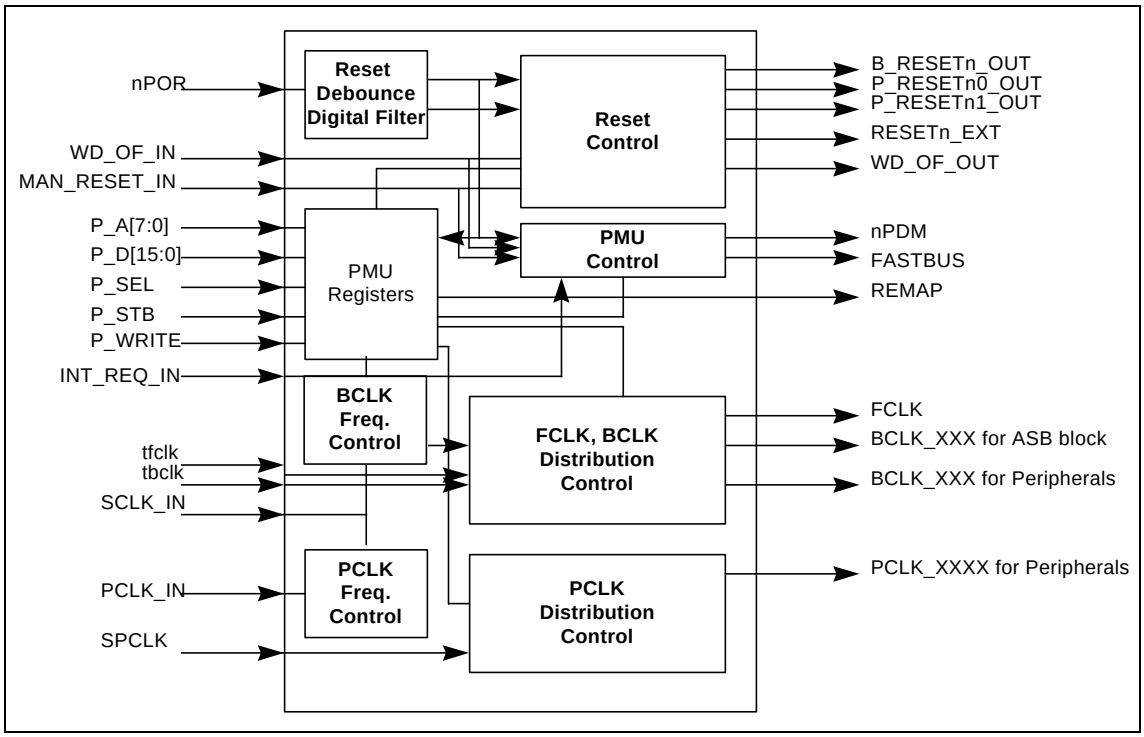


Figure 1. PMU Block Diagram

2. Hardware Interface and Signal Description

The PMU block is connected to the APB bus. Table 1. describes the APB signals and clock signals used and produced.

Table 1. PMU Signal Descriptions

NAME	TYPE	DESCRIPTION
nPOR	I	External reset input.
INT_REQ_IN	I	Interrupt request signal from the interrupt controller.
WD_OF_IN	I	Watch dog timer overflow signal.
MAN_RST_IN	I	S/W manual reset pin from watch dog timer.
P_D[15:0]	I/O	This is the bi-directional peripheral data bus. This block drives the data bus during read cycle, when P_WRITE is LOW.
P_A[7:0]	I	This is the peripheral address bus, which uses individual peripheral for decoding register accesses to that peripheral. The addresses become valid before PSTB goes to HIGH and remain valid after PSTB goes to LOW.
P_WRITE	I	This signal indicates a write to a peripheral when it is HIGH and a read from a peripheral when LOW. It has the same timing as the peripheral address bus.
P_STB	I	This strobe signal is used to time all accesses on the peripheral bus. The falling edge of PSTB is coincident with the falling edge of BCLK
P_SEL	I	When HIGH, this signal indicates that module has been selected by the APB bridge.
SCLK_IN	I	System clock input . This is the clock input from external clock circuit .
PCLK_IN	I	UART clock. This is the clock input from external UART clock module.
Tfclk	I	When it is in TIC test mode, this s the FCLK clock input signal. When TSTCR[1] is set to 1(HIGH) for entering TIC test mode.
Tbclk	I	When it is in TIC test mode, BCLK clocks signal. First set the TSTCR[1] to 1 for entering TIC test mode.
BCLK_XXX	O	System Bus clock is generated from SCLK_IN. All ASB block and some APB blocks are operated by this clock.
PCLK_XXX	O	APB Peripheral Bus Clocks. All APB blocks are operated by the clocks.
FCLK	O	FCLK pin for ARM720T. It is used in standard mode, when FASTBUS is LOW.
FASTBUS	O	ARM720T bus mode control signal. When it is LOW, it is in standard bus mode. When HIGH, fast bus mode.
NPDM	O	Indicates the PDM mode of PMU. When it is LOW, MCU entered in power down mode. When HIGH, normal operation mode.
REMAP	O	Indicates that the reset memory map is in operation.
WD_OF_OUT	O	Watch dog overflow output signal for external devices.
B_RESETn	O	Reset signal for ASB devices
P_RESETn0	O	Reset signal for APB devices
P_RESETn1	O	Same as P_RESETn0, but in manual reset mode this is not asserted.
RESETn_OUT	O	Reset signal for external devices.

3. Operation Modes

3.1 Introduction

The reset protocol guarantees that the multi-master system starts up with at most one bus driver enabled on each shared signal on the bus, and also permits a protocol reset mechanism for time-out or 'watchdog' reset support.

To improve power management, support for a power-saving mode where bus clocks may be disabled (or dropped to lower clock) is included.

The reset and power-down mechanism provides:

- Stable power-up sequence
- Hard Initialization (Power On Reset)
- Soft Initialization (S/W Manual Restart)

Additionally a system bus, once operational, benefits from well-defined modes of operation:

- RUN in the Standard BUS mode
- RUN in the FAST BUS mode
- Power-down mode

3.2 Reset and Operation Modes

A set of four useful states or modes is defined as follows:

RESET

When it is power-on, watchdog timer overflow, watchdog timer manual reset or S/W reset, the MCU is initialized

Rower on Reset

The most severe form of reset which ensures that no more than one tri-state driver is enabled on each bus and initializes all system states to ensure that the power supply can in fact rise to normal operating voltage.

This state should be forced by any on-chip power-on-reset cell or external power-on signal and maintained until bus clock is safe and stable.

The POR is forced to be in an asynchronous start-up condition and must be recognized by all master and

slave devices to disable output drives (and wait for a valid clock)

Manual Reset / Software Reset

The manual reset, which may need to apply to allow all soft resetting of the bus for a number of clock cycles. In this reset states the PMU block initializes all the ASB blocks, Bus controller, DRAM Controller, DMA Controller, ARM CPU core, and Arbiter, Decode. However some APB blocks are all valid in warm reset.

Watchdog Timer Overflow and Manual Reset

The watchdog timer can generate reset signal, when timer overflows or sets the register value. Detailed information are in the watchdog timer manual, please refer to it.

RUN - ARM720T Standard Mode.

The ARM720T works using the FCLK and BCLK. The FCLK is used for CPU operation clock, and the B_CLK is used for internal bus access, i.e. AMBA BUS. So CPU can operate very high frequency. This mode can control the clock of ASB and APB devices, so user can disable the clocks of unused devices or peripherals. It is possible to control the BCLK or PCLK mask register.

RUN -ARM720T Fast-bus Extension Mode.

The ARM720T works using only the BCLK. The CPU operation clock and AMBA bus access clock are the same. This mode can control the clock of ASB and APB devices, so user can disable the clock of devices or peripherals that are not using now. It is possible to control the BCLK mask register or PCLK mask register.

PDN – Power-Down Mode

When MCU system is in the PDN State, PMU block disables all of the blocks in the ASB and APB, so the power consumption of system is dramatically low. Although MCU is in the power down mode, user can set some blocks are working in the power down mode. It is possible control the BCLK or PCLK mask register for power-down.

Wake-up from the PDN Mode.

The Wake-up is a temporal state for wake-up from power down state through the interruption. After wake-up state, next state becomes RUN state automatically.

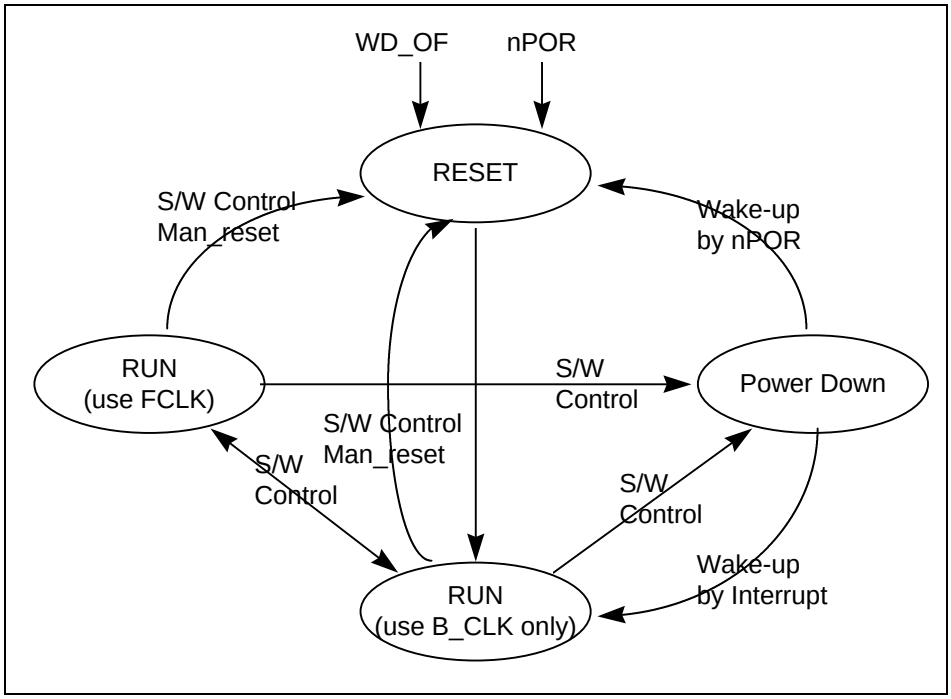


Figure 2. Reset and Power Management State Machine.

4. Register Description

The PMU supplies the clock to all of the blocks in the MCU.

4.1 PMU Control Register

This register controls the operation mode of PMU. When power on reset states, register value is initialized by Run State (00). The address of register is PMU_BASE(=0xFFFF F000) + 0x00h.

Table 2. PMUCR Bit Functions

BIT	INITIAL	NAME	FUNCTION
7~0	0x0	PMUCR	0x0 - Clear PMU Status Register. 0x03 – Entering the PD(Power down) Mode the other values - None effect.

4.2 PMU Status Register

This register holds the previous status and reset state of PMU. The address of register is PMU_BASE + 0x00h.

Table 3. PMUSR Bit Functions

BIT	INITIAL	NAME	FUNCTION
5, 4	00	PMUST[5:4] Previous Reset Status bits	00 - The Power-On reset state (nPOR). 01 - S/W Reset state using PMU. 10 - S/W Manual reset state using WDT. 11 - WD overflow reset state using WDT.
3, 2	00	PMUST[3:2] Current Status bits	00 - Running (FAST, SLOW) after nPOR. 01 - Running (FAST, SLOW) after WD_OF. 10 - Running (FAST, SLOW) after Man_reset
1, 0	00	PMUST[1:0] Previous Status bits	00 - Start (FAST, SLOW) after nPOR. 01 - Start (FAST, SLOW) after WD_OF. 10 - Start (FAST, SLOW) after Man_reset 11 - Start (FAST, SLOW) after PD Mode.

4.3 REMAP Register

The REMAP register controls re-mapping operation when the reset (POR or MAN_RST) signal is asserted or S/W is reset by RSTCR. The address is PMU_BASE + 0x10h.

Table 4. REMAP Bit Functions

BIT	INITIAL	NAME	FUNCTION
0	0	REMAP	0 – Reset operation mode map 1 – Normal operation mode map

4.4 BCLK and FCLK Control Register and BCLK Frequency Control Register

This register controls BCLK of ASB and FCLK of ARM720T. User can save the power by reduce of the clock speed. At any moment, user can change the BCLK speed but it may push the system into unstable stage, so user must change the clock speed only in BUS IDLE; this means there is no interaction between the devices used by BCLK and any other devices used by PCLK. User can control the bus mode that are standard-bus mode and fast-bus mode. The BCLK is only used in the fast-bus mode and ARM720T uses the both clock FCLK and BCLK in the fast bus mode. The address is PMU_BASE + 0x04h.

Table 5. CLKCR Bit Functions

BIT	INITIAL	NAME	FUNCTION
2 - 0	000	BCLKCR[2:0]	Control register for BCLK selection 000 - BCLK is divided SYS_CLK by 2 001 - BCLK is divided SYS_CLK by 4 010 - BCLK is divided SYS_CLK by 8 011 - BCLK is divided SYS_CLK by 16 100 - BCLK is divided SYS_CLK by 32 101 - BCLK is divided SYS_CLK by 64 110 - BCLK is divided SYS_CLK by 128 111 - BCLK is SYS_CLK.

Table 6. CLKCR Bit Functions

BIT	INITIAL	NAME	FUNCTION
3	1	BCLKCR[3]	Control register to use in FCLK mode 1 – Fast-bus mode (not use the FCLK) 0 – Standard-bus mode use the FCLK that same the SYS_CLK

4.5 BCLK Mask Register for the RUN & PD Mode.

This register is used for masking BCLK of ASB devices in the RUN and PD mode. When each control bits are written to “1 or 0”, each clock of devices is controlled by enabled or disabled clock in the RUN and PD mode. The address of the mask control register are as follows.; BCLKMSK_RUN is PMU_BASE + 0x08h, BCLKMSK_PD is PMU_BASE + 0x0Ch. When this is 1, it is enable clock. When 0, disable clock.

Table 7. BCLKMSK Bit Functions for RUN Mode

BIT	INITIAL	NAME	FUNCTION
15-13	1	BCLKMSK_RUN	Reserved bit
12	1		APB Bridge clock mask bit
11	1		BUS Controller clock mask bit
10	1		DRAM Controller clock mask bit
9	1		DMA Controller clock mask bit
8	1		TEST Controller clock mask bit
7	1		SRAM clock mask bit
6 - 1	111111		Reserved bit
0	1		B_CLK Out mask bit

Table 8. BCLKMSK Bit Functions for PD Mode

BIT	INITIAL	NAME	FUNCTION
15	0	BCLKMSK_PD	ARM7TDMI core clock mask bit
14	0		AMBA Arbiter clock mask bit
13	0		AMBA Decoder clock mask bit
12	0		APB Bridge clock mask bit
11	0		BUS Controller clock mask bit
10	0		DRAM Controller clock mask bit
9	0		DMA Controller clock mask bit
8	0		TEST Controller clock mask bit
7	0		SRAM clock mask bit
6 - 1	000000		Reserved bit
0	0		B_CLK Out mask bit

4.6 PCLK Mask Register

These registers are used for masking PCLK or BCLK of APB devices in the RUN or PD mode. The default values are all the clocks of the APB devices enabled in the RUN mode, and the clocks of the APB devices disabled in the PD mode. WDT and TIMER are APB devices but they uses the BCLK for their operation. The address of the mask control register in RUN mode is PMU_BASE + 0x18h, and that of the power-down mask control register is PMU_BASE + 0x1Ch. When this is 1, it enables clock. When 0, disables clock.

Table 9. PCLKMSK Bit Functions in the RUN Mode

BIT	INITIAL	NAME	FUNCTION
9	1	PCLKMSK_RUN	Watch dog timer clock mask bit
8	1		I^2C 2 clock mask bit
7	1		I^2C 1 clock mask bit
6	1		I^2C 0 clock mask bit
5	1		SSPI 1 clock mask bit
4	1		SSPI 0 clock mask bit
3	1		UART 2 / SMART card I/F clock mask bit
2	1		UART 1 clock mask bit
1	1		UART 0 clock mask bit
0	1		Timer clock mask bit

Table 10. PCLKMSK Bit Functions in the PD Mode

BIT	INITIAL	NAME	FUNCTION
9	0	PCLKMSK_PD	Watch dog timer clock mask bit
8	0		I^2C 2 clock mask bit
7	0		I^2C 1 clock mask bit
6	0		I^2C 0 clock mask bit
5	0		SSPI 1 clock mask bit
4	0		SSPI 0 clock mask bit
3	0		UART 2 / SMART card I/F clock mask bit
2	0		UART 1 clock mask bit
1	0		UART 0 clock mask bit
0	0		Timer clock mask bit

4.7 PCLK Frequency Control Register

This register is used to selecting the frequency of PCLK in the APB at RUN mode. Default value is 0000. The address of access the register is PMU_BASE + 0x14h

Table 11. CLKMODE Bit Functions

BIT	INITIAL	NAME	FUNCTION
2 - 0	000	PCLKCR	Select the PCLK source 000 – PCLK is external PCLK source 001 – PCLK is the SCLK divided by 2 010 – PCLK is the SCLK divided by 4 011 – PCLK is the SCLK divided by 8 100 – PCLK is the SCLK divided by 16 101 – PCLK is the SCLK divided by 32 110 – PCLK is the SCLK divided by 64 111 – PCLK is the SCLK divided by 128

4.8 Reset Control Register

This register is used for generating the S/W reset operation. The MCU is entered in reset state, when this register is set to high, it is cleared automatically at the end of manual reset procedure. The address is PMU_BASE + 0x30h.

Table 12. RSTCR Bit Functions

BIT	INITIAL	NAME	FUNCTION
0	0	RSTCR	Manual reset control bits 0 - Normal , 1 - manual reset

4.9 Test Control Register

TSTCR controls the normal mode, PMU test mode or the TIC test mode. The address is PMU_BASE + 0x40h.

Table 13. TSTCR Bit Functions

BIT	INITIAL	NAME	FUNCTION
1	0	TSTCR	0 – Normal operation mode 1 – TIC Test mode
0	0		0 – Normal operation mode 1 – PMU test mode

4.10 Test Register

This register is used to store some controls and data values for test mode. The TSTR0 is readable/writable register and the TSTR1 is a read only register. The address of TSTR0 is PMU_BASE + 0x48h and that of TSTR1 is PMU_BASE + 0x4Ch.

Table 14. TSTR0 Bit Functions

BIT	INITIAL	NAME	FUNCTION
2	0	TSTR0	Test bit for INT_REQ_IN input
1	0		Test bit for WD_OF_IN input
0	0		Test bit for MAN_RESET_IN input

Table 15. TSTR1 Bit Functions

BIT	INITIAL	NAME	FUNCTION
15	0	TSTR1	Test bit for BCLK_WDT
14	0		Test bit for PCLK_I^2C2
13	0		Test bit for PCLK_I^2C1
12	0		Test bit for PCLK_I^2C0
11	0		Test bit for PCLK_SSPI 1
10	0		Test bit for PCLK_SSPI 0
9	0		Test bit for PCLK_UART 2, SMART Card
8	0		Test bit for PCLK_UART 1
7	0		Test bit for PCLK_UART 0
6	0		Test bit for BCLK_TIMER
5	0		Test bit for B_RESETn
4	0		Test bit for P_RESETn0
3	0		Test bit for P_RESETn1
2	0		Test bit for P_RESETn
1	0		Test bit for WD_OF_OUT
0	0		Test bit for REMAP

5. Power Management Unit Register Map

The base address of the PMU(Power Management Unit) is 0xFFFF F000. May be different for any particular system implementation. However, the offset address of registers is fixed.

Table 16. Register Map of the PMU

Address	Name	Description
PMU Base + 0x00	PMUCR / PMUSR	In write operation, PMU operation mode controls register. In read operation, PMU status register shows the just previous PMU state.
PMU Base + 0x04	BCLKCR	BCLK frequency selection and BUS mode control(Standard / Fast BUS mode)
PMU Base + 0x08	BCLKMSK_RUN	BCLK Masking controls register in the RUN mode.
PMU Base + 0x0C	BCLKMSK_PD	BCLK Masking controls register in the PD mode.
PMU Base + 0x10	REMAP	REMAP register
PMU Base + 0x14	PCLKCR	PCLK control register
PMU Base + 0x18	PCLKMSK_RUN	PCLK masking controls register for the RUN mode.
PMU Base + 0x1C	PCLKMSK_PD	PCLK masking controls register for the PD mode.
PMU Base + 0x20	Reserved	Reserved.
PMU Base + 0x30	RSTCR	Reset control register
PMU Base + 0x40	TSTCR	TIC test mode and PMU test control register
PMU Base + 0x44	-	Reserved
PMU Base + 0x48	TSTR0	Test write register for external input signals
PMU Base + 0x4C	TSTR1	Test read register for clocks of ASB devices and reset signals

6. Test Mode Guide for MCU

6.1 TIC Test Mode

Step 1. Set-up TIC test environment, connect the TCLK to tbclk(=XPA[6]) pin and FCLK of fclkgen to tfclk (=XPA[7]) pin. Until the mode of the PMU is changed to the TIC test mode, sys_clk has to feed same TCLK to the clock.
Step 2. Reset the MCU using pin nPOR low.
Step 3. Then TIC becomes the bus master
Step 4. Change the mode of PMU by setting the TSTCR to “10” and set TSTCR[1] to high.
Step 5. Start TIC test using tbclk and tfclk pins.

Figure 3. shows TIC Test Environment of Internal Blocks.

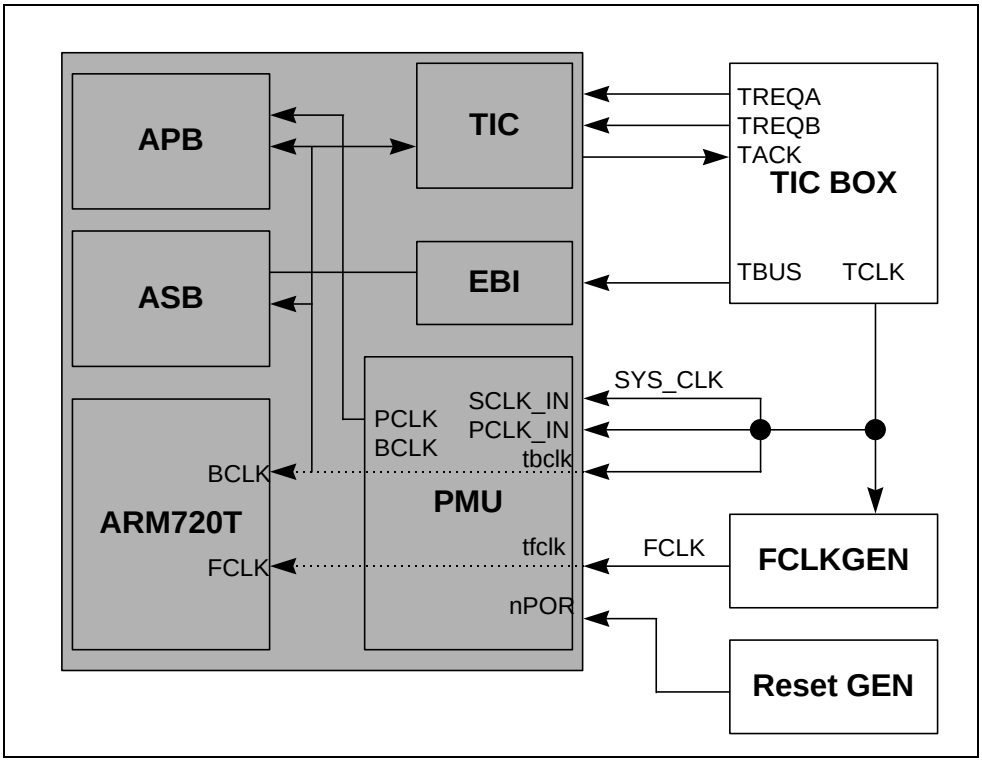


Figure 3. Internal Blocks TIC Test Environment

6.2. TIC Test for PMU Block

Step 1. Set-up TIC test environment, connect the TCLK to tbclk and FCLK of fclkgen to tfclk pin. Until the mode of the PMU is changed to the TIC test mode, sys_clk has to feed the clock. It is easier to test the sys_clk and the tbclk separately.
Step 2. Reset the MCU using pin nPOR low.
Step 3. Then TIC becomes the bus master
Step 4. Change the mode of PMU by setting the TSTCR[1:0] to “11”.
Step 5. Set value to TSTR0 for test vector, and read output value in the TSTR1 of the PMU and compare it with desired test vector.

Figure 4. shows PMU TIC Test Environment

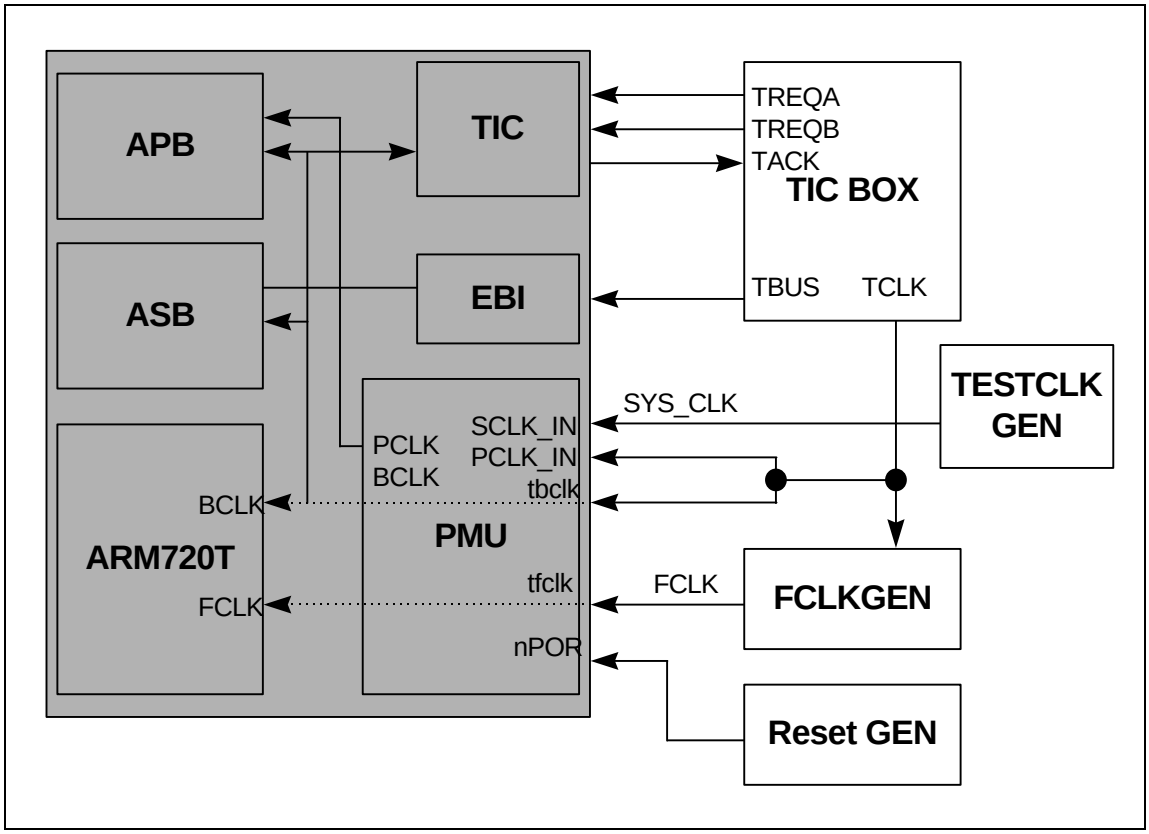


Figure 4. TIC Test Environment

7. Signal Timing Diagram

The PMU signal timing is as shown below.

7.1 Power on Reset

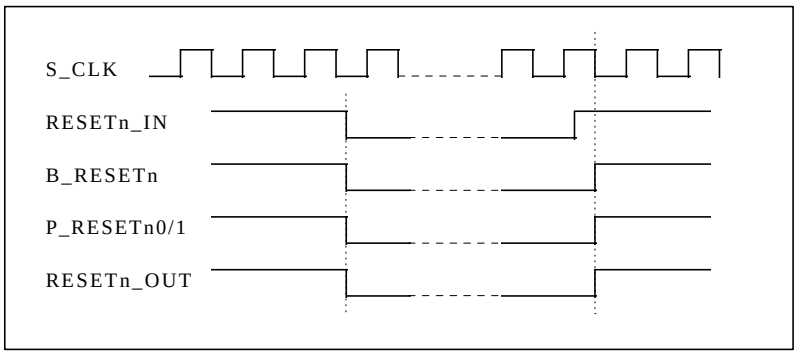


Figure 5. Power on Reset Timing Diagram

7.2 Watch Dog Timer Overflow

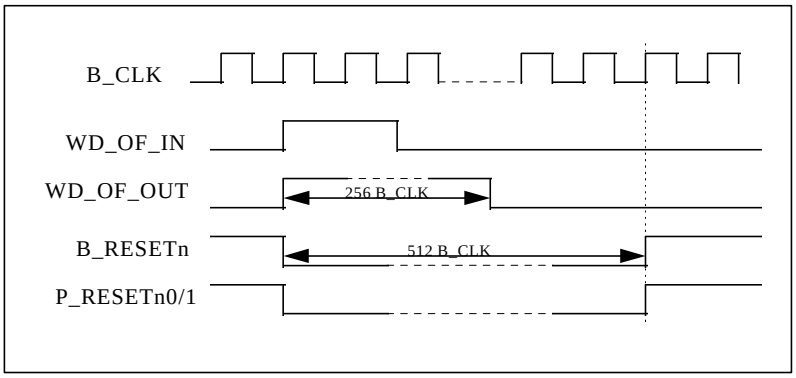


Figure 6. Watch Dog Timer Overflow Timing Diagram

7.3 Manual Reset

There are two manual reset cases. The first reset operation is switched by MAN_RST signal from WDT. Another case is called S/W reset.

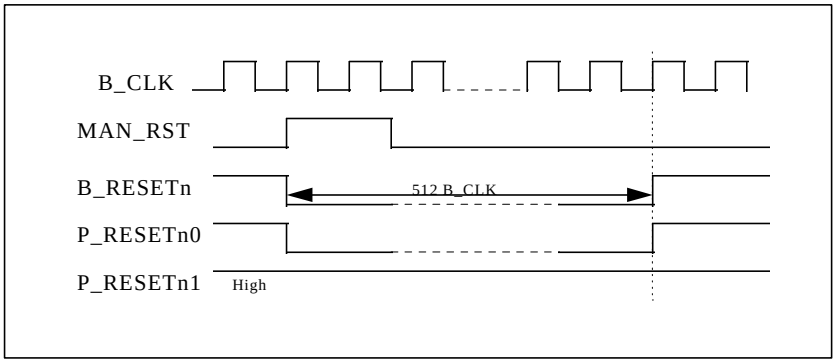


Figure 7. Manual Reset (from WDT) Timing Diagram

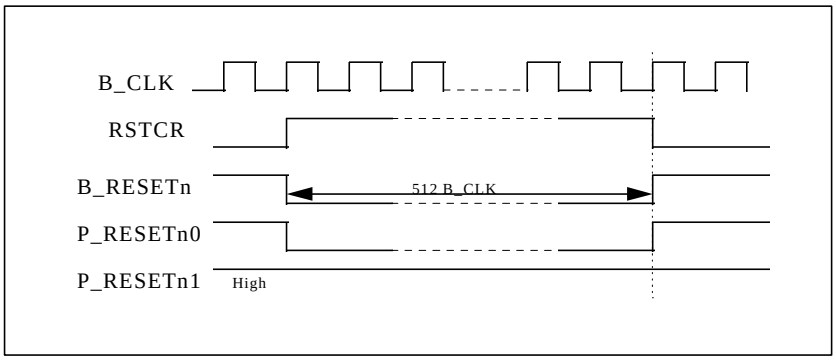


Figure 8. S/W Reset Timing Diagram

Section 9. Watchdog Timer

1. General Description

The watchdog timer has:

- watchdog timer mode and interval timer mode
- interrupt signal INT_WDT to interrupt controller in the watchdog timer mode & interval timer mode
- output signal PORESET and MNRESET to PMU(Power Management Unit)
- eight counter clock sources
- selection whether to reset the chip internally or not
- two types of reset signal : power-on reset and manual reset

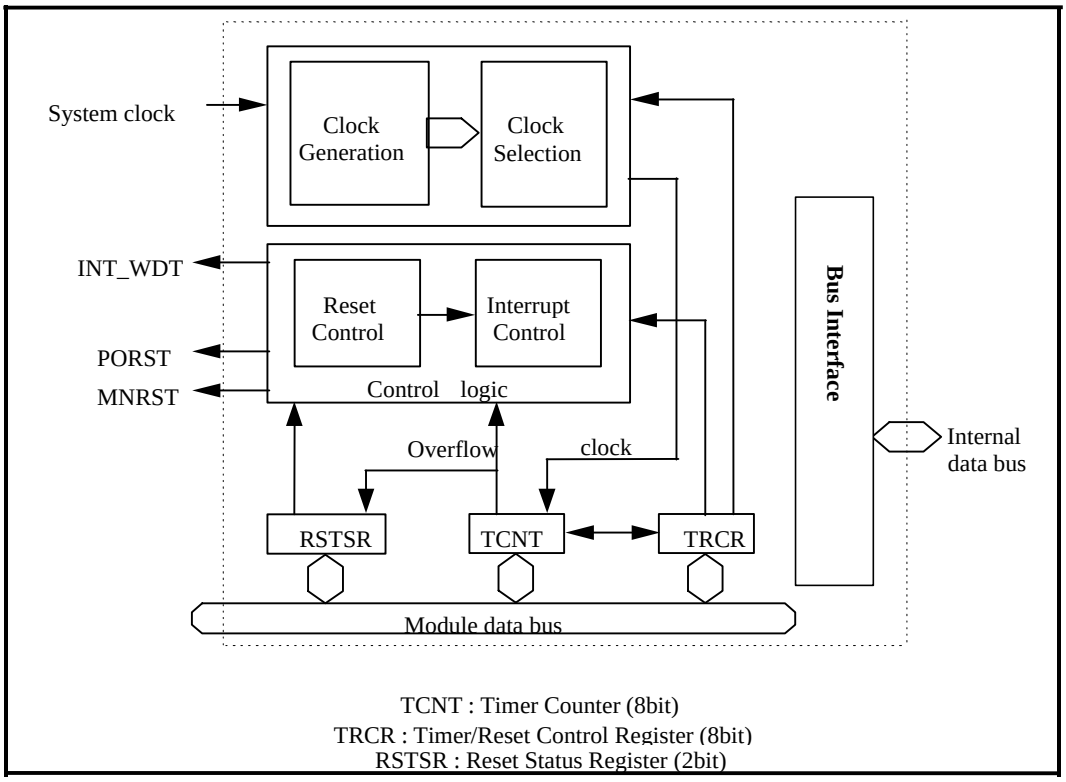


Figure 1. Watchdog Timer Module Block Diagram

2. Hardware Interface and Signal Description

The Watchdog Timer module is connected to the APB bus.

Table 1. APB Signal Descriptions

NAME	Type	SOURCE/ DESTINATION	DESCRIPTION
B_CLK	I	Clock controller	System (bus) clock. This clock times all bus transfers. The clock has two distinct phases - phase 1 when B_CLK is LOW, and phase 2 when B_CLK is HIGH.
P_A[4:2]	I	APB Bridge	This is the peripheral address bus used by an individual peripheral for decoding register accesses to that peripheral. The addresses become valid before P_STB goes to HIGH and remain valid after P_STB goes to LOW.
P_D[7:0]	I/O	APB Peripherals, B_D bus	This is the bi-directional peripheral data bus. The data bus is driven by this block during read cycles (when P_WRITE is LOW).
P_STB	I	APB Bridge	This strobe signal is used to time all accesses on the peripheral bus. The falling edge of P_STB is coincident with the falling edge of B_CLK.
P_WRITE	I	APB Bridge	When this signal is HIGH, it indicates a write to a peripheral. When LOW, it indicates a read from a peripheral. This signal has the same timing as the peripheral address bus. It becomes valid before P_STB goes to HIGH and remains valid after P_STB goes to LOW.
P_SEL	I	APB Bridge	When this signal is HIGH, it indicates that this module has been selected by the APB bridge. This selection is a decode of the system address bus (ASB). See AMBA Peripheral Bus Controller for more details.
nB_RES	I	Power Management Unit	Reset signal generated from the APB Bridge
INT_WDT	O	Interrupt Controller	When this signal is HIGH, it indicates that a system becomes uncontrolled, and the timer counter overflows without being rewritten correctly by the CPU or it overflows in the interval timer mode.
MNRST	O	Power Management Unit	When this signal is HIGH, this signal indicates that the manual reset signal has selected as the internal reset signal, and the timer counter overflows without being rewritten correctly by the CPU or it overflows in the interval timer mode..
PORST	O	Power Management Unit	When this signal is HIGH, this signal indicates that the power-on reset signal has selected as the internal reset signal.

Writes to the Watchdog Timer module are generated from the Peripheral Bus Controller module. Figure 2. Watchdog timer module APB write cycle summarizes this description.

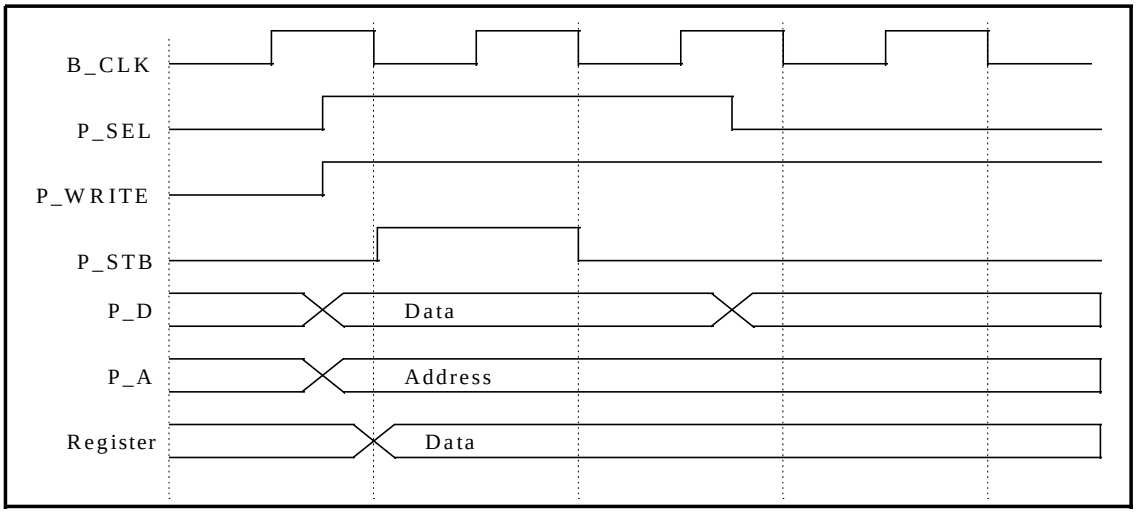


Figure 2. Watchdog Timer Module APB Write Cycle

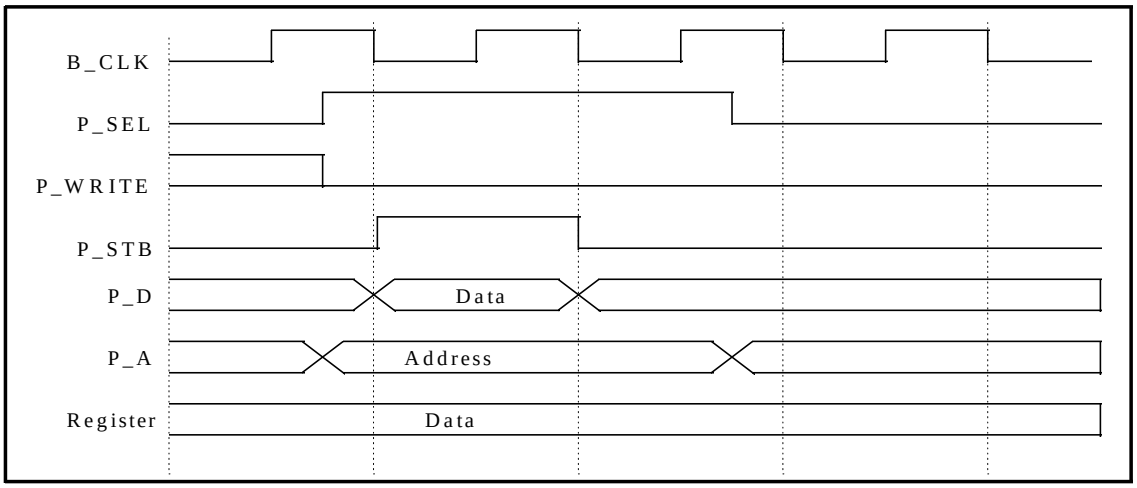


Figure 3. Watchdog Timer Module APB Read Cycle

3. Watchdog Timer Introduction

The GDC21D601 has a one-channel watchdog timer(WDT) for monitoring system operations. If a system becomes uncontrolled and the timer counter overflows without being rewritten correctly by the CPU, an 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.

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). Users can select one of eight internal clock sources for input to the TCNT by CKS2 - CKS0 in the TRCR.

Table 2. Internal Counter Clock Sources

BIT 2 - 0 (CKS2-CKS0)	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

4. Watchdog Timer Operation

The Watchdog Timer Mode

To use the WDT as a watchdog timer, set the WT/nIT and TMEN bits of the TRCR to 1. Software must prevent TCNT overflow by rewriting the TCNT value(normally by writing 0x00) before overflow occurs. If the TCNT 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 INTEN is disabled (INTEN = 0).

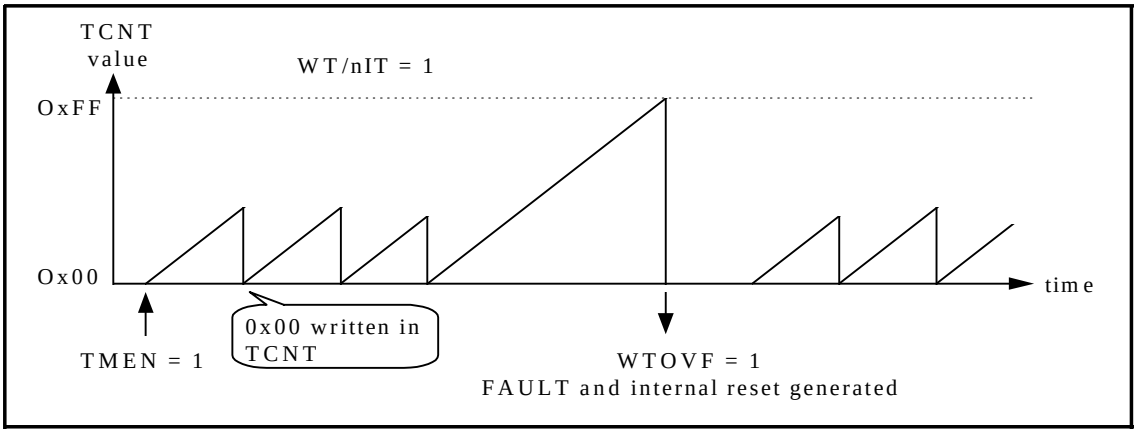


Figure 4. Operation in the Watchdog Timer Mode

If the RSTEN bit in the TRCR is set to 1, a signal to reset the chip will be generated internally when TCNT overflows. Either a power-on reset or a manual reset can be selected by the RSTSEL bit.

The Interval Timer Mode

To use the WDT as an interval timer, clear WT/nIT 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.

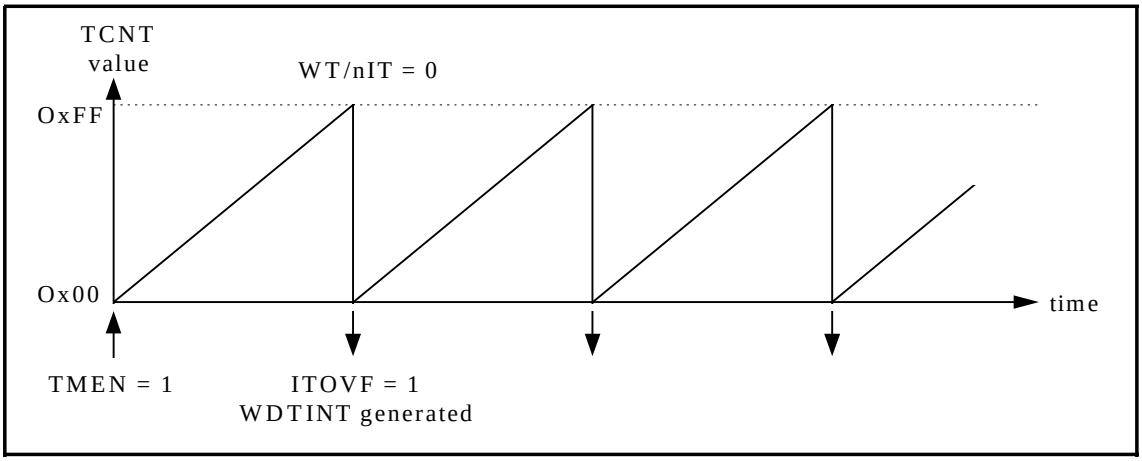


Figure 5. Operation in the Interval Timer Mode

4.1 Timing of Setting and Clearing the Overflow Flag

Timing of setting the overflow flag

In the interval timer mode when the TCNT overflows, the ITOVF flag is set to 1 and an watchdog timer interrupt (INT_WDT) is requested.

In the watchdog timer mode when the TCNT overflows, the WTOVF bit of the SR is set to 1 and a WDTOUT signal is output. When RSTEN bit is set to 1, TCNT overflow enables an internal reset signal to be generated for the entire chip.

Timing of clearing the overflow flag

When the Reset Status Register (RSTSR) is read, the overflow flag is cleared.

5. Watchdog Timer Memory Map

The WDT has five registers. They are used to select the internal clock source, switch to the WDT mode, control the reset signal, and test it. The base address of the watchdog timer is fixed to 0xFFFF F100 and the offset of any particular register from the base address is fixed.

Table 3. Memory Map of the Watchdog Timer APB Peripheral

ADDRESS	READ LOCATION	WRITE LOCATION
WdtBase + 0x00	Timer/Reset Control	Timer/Reset Control
WdtBase + 0x04	Reset Status	
WdtBase + 0x08	Timer Counter	Timer Counter
WdtBase + 0x10		Test Input
WdtBase + 0x14	Test Output	

6. Watchdog Timer Register Descriptions

The following registers are provided for watchdog timer:

Timer Counter (TCNT)

8-bit readable and writable upcounter. When the timer is enabled, the timer counter starts counting pulse of the selected clock source. When the value of the TCNT changes from 0xFF-0x00(overflows), a watchdog timer overflow signal is generated in the both timer modes. The TCNT is initialized to 0x00 by a power-reset(nB_RES).

Timer/Reset Control Register (TRCR)

8-bit readable and writable register. The following functions are provided :

Selecting the timer mode

Selecting the internal clock source

Selecting the reset mode

Setting the timer enable bit

Being enable interrupt request

Being enable reset signal occurrence

The clock signals are obtained by dividing the frequency of the system clock.

Table 4. TRCR Bit Description

BIT	INITIAL VALUE		FUNCTION
0 (clock select : CKS0)	0	000 = /2	select one of eight internal clock sources for input to the TCNT.
1 (clock select : CKS1)	0	001 = /8	
2 (clock select : CKS2)	0	010 = /32 011 = /64 100 = / 256 101 = /512 110 = /2048 111 = /8192	
3 (reset select : RSTSEL)	0	0 = poser-on reset 1 = manual reset	select the type of generated internal reset if the TCNT overflows in the watchdog timer mode.
4 (reset enable : RSTEN)	0	0 = disable 1 = enable	select whether to reset the chip internally or not if the TCNT overflows in the watchdog timer mode.
5 (timer enable : TMEN)	0	0 = disable 1 = enable	enable or disable the timer
6 (timer mode select : WT/nIT)	0	0 = interval timer mode 1 = watchdog timer mode	select whether to use the WDT as a watchdog timer or interval timer
7 (Interrupt enable : INTEN)	0	0 = disable 1 = enable	enable or disable the interrupt request

Reset Status Register (RSTSR)

Two-bit read only register. The RSTSR indicates whether TCNT is overflowed or not. The RSTSR is initialized to 0x0 by the reset signal, nB_RES.
Bit 0 (WTOVF) indicates that the TCNT has overflowed in the watchdog timer mode. Bit 1 (ITOVF) indicates that the TCNT has overflowed in the interval timer mode.

Table 5. SR Bit Description

BIT	INITIAL VALUE		FUNCTION
0 (watchdog timer overflow flag : WTOVF)	0		indicate that the TCNT has overflowed in the watchdog timer mode.
1 (interval timer overflow flag : ITOVF)	0		indicate that the TCNT has overflowed in the interval timer mode

7. Examples of Register Setting

7.1 Interval Timer Mode

TCNT = 0x00
TRCR = 0xA0

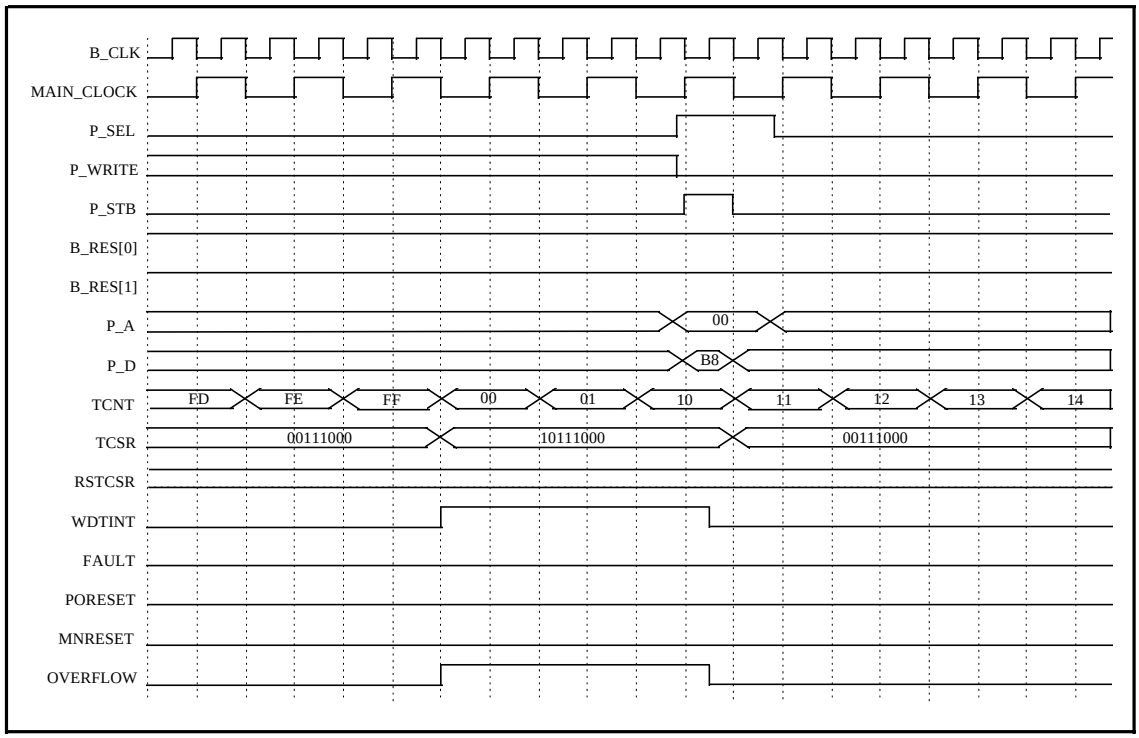


Figure 6. Interrupt Clear in the Interval Timer Mode

7.2 Watchdog Timer Mode with Internal Reset Disable

TCNT = 0x00 (normally)
TRCR = 0xE0

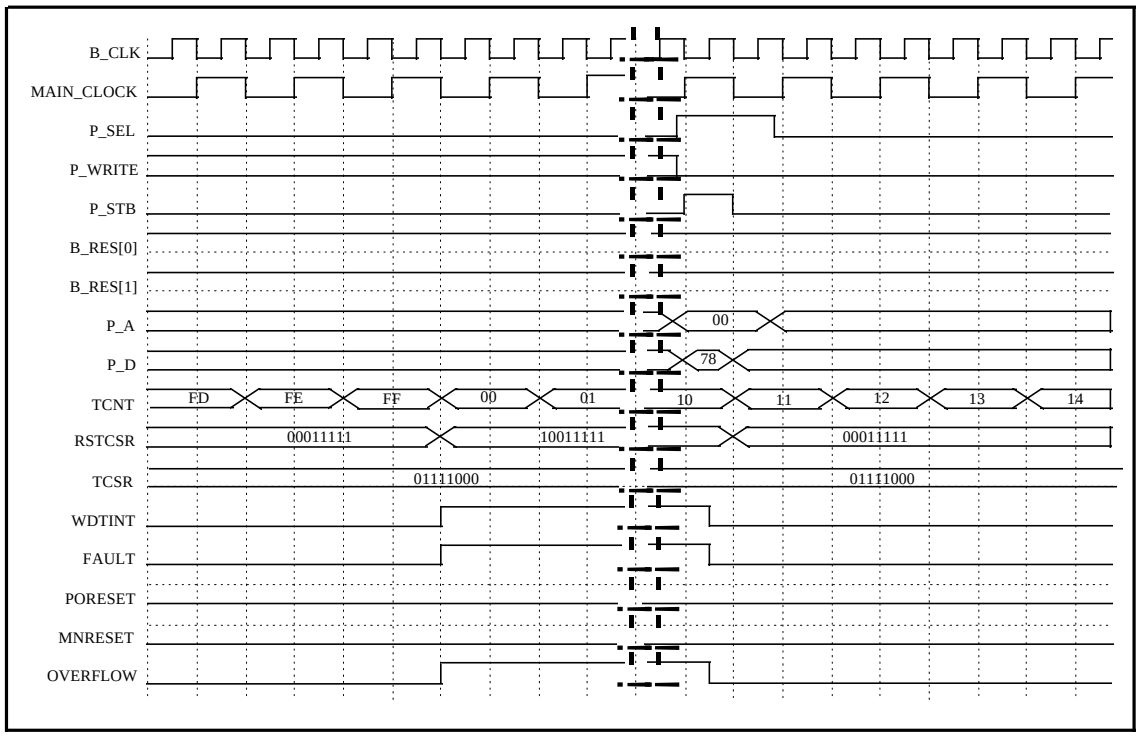


Figure 7. Interrupt Clear in the Watchdog Timer Mode with Reset Disable

7.3 Watchdog Timer Mode with Power-on Reset

TCNT = 0x00
TRCR = 0xF0

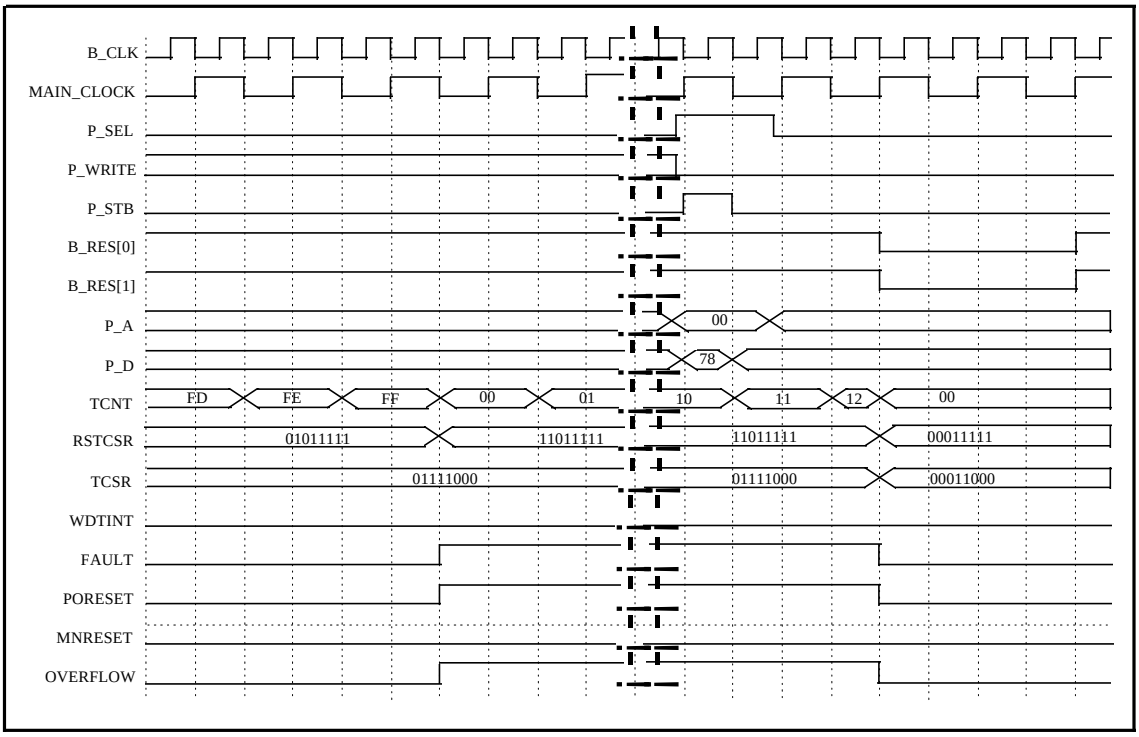


Figure 8. Interrupt Clear in the Watchdog Timer Mode with Power-on Reset

7.4 Watchdog Timer Mode with Manual Reset

TCNT = 0x00
TRCR = 0xF8

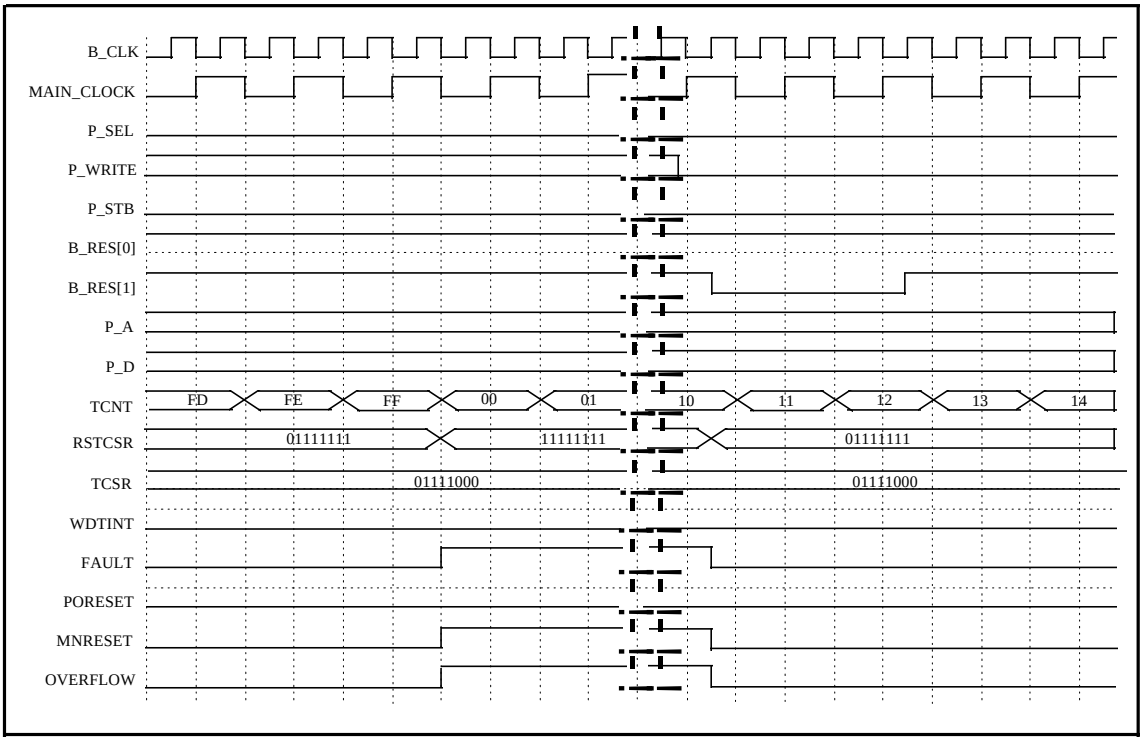


Figure 9. Interrupt Clear in the Watchdog Timer Mode with Manual Reset

Section 10. Interrupt Controller

1. General Description

The interrupt controller has the following features :

- Asynchronous interrupt controller
- Six external interrupts
- Nineteen internal interrupts
- Low interrupt latency
- Selection of the active modes of all interrupt source inputs (Level or Edge trigger)
- Maskable for each interrupt source and output signal
- Selection of the output paths (IRQ or FIQ for each interrupt source)

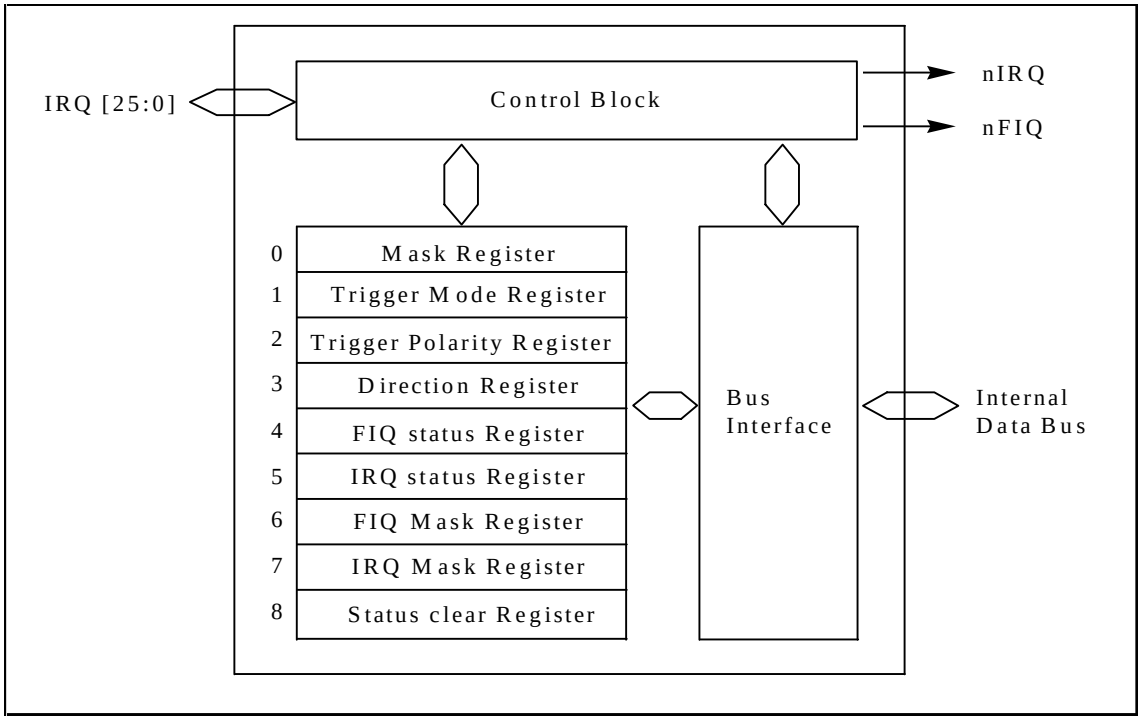


Figure 1. Interrupt Controller Module Block Diagram

2. Hardware Interface and Signal Description

The Interrupt Controller module is connected to the APB bus.

Table 1. APB Signal Descriptions

NAME	TYPE	SOURCE/ DESTINATION	DESCRIPTION
P_A[5:2]	I	APB Bridge	This is the peripheral address bus, which is used by an individual peripheral for decoding register accesses to that peripheral. The addresses become valid before P_STB goes to HIGH and remain valid after P_STB goes to LOW.
P_D[26:0]	I/O	APB Peripherals, B_D bus	This is the bidirectional peripheral data bus. The data bus is driven by this block during read cycles (when P_WRITE is LOW).
P_STB	I	APB Bridge	This strobe signal is used to time all accesses on the peripheral bus. The falling edge of P_STB is coincident with the falling edge of B_CLK.
P_WRITE	I	APB Bridge	When this signal is HIGH, it indicates a write to a peripheral. When this signal is LOW, it indicates a read from a peripheral. This signal has the same timing as the peripheral address bus. It becomes valid before P_STB goes to HIGH and remains valid after P_STB goes to LOW.
P_SEL	I	APB Bridge	When this signal is HIGH, it indicates that this module has been selected by the APB bridge. This selection is a decode of the system address bus (ASB). See AMBA Peripheral Bus Controller (ARM DDI - 0044) for more details.
INTESource[25:0]	I	APB peripherals/ external world	FIQ/IRQ interrupt signals into the Interrupt module. These active HIGH signals indicate that interrupt requests have been generated (IRQESource[25] is internally generated in the Interrupt Controller module and used to provide a software triggered IRQ).
NFIQ	O	ARM CORE	NFIQ interrupt input to the ARM core.
NIRQ	O	ARM CORE	NIRQ interrupt input to the ARM core.
BnRES	I	PMU	Reset signal generated from the Power Management Unit.

Writes to the Interrupt Controller module are generated from the Peripheral Bus Controller module. Figure 2. Interrupt control module APB write cycle summarizes this.

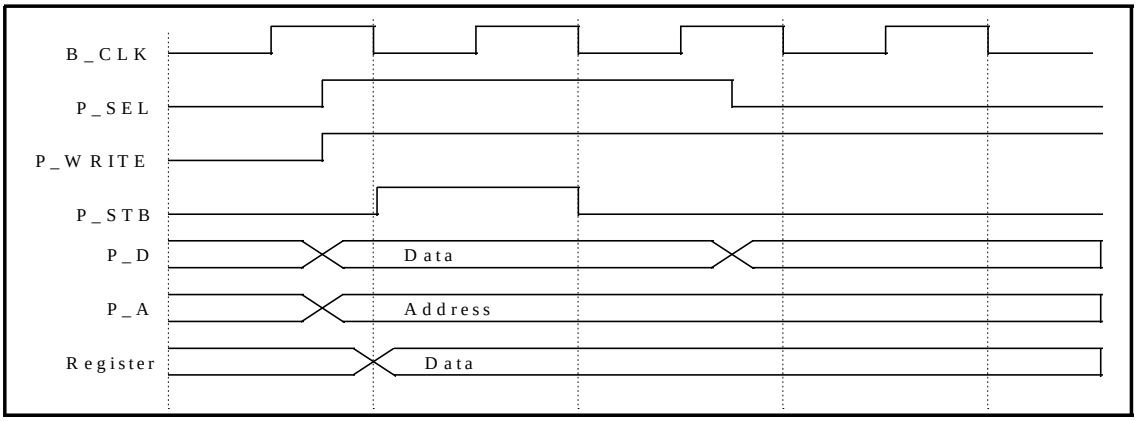


Figure 2. Interrupt Control Module APB Write Cycle

3. Interrupt Controller

3.1 Introduction

The interrupt controller provides a interface between multiple interrupt source and the processor. The interrupt controller supports internal and external interrupt sources. Internally there are 19 peripheral interrupt sources. Externally there are 6 interrupt sources. Therefore certain interrupt bits can be defined for the basic functionality required in any system, while the remaining bits are available for use by other devices in any particular implementation.

Table 2. Interrupt Controller Default Setting Value

INT #	INTERRUPT SOURCE
INT 0	EXTERNAL INT0
INT 1	EXTERNAL INT1
INT 2	EXTERNAL INT2
INT 3	EXTERNAL INT3
INT 4	EXTERNAL INT4
INT 5	EXTERNAL INT5
INT 6	COM TX
INT 7	COM RX
INT 8	DMA
INT 9	RTC
INT 10	WDT
INT 11	I^2C0
INT 12	I^2C1
INT 13	I^2C2
INT 14	UART0
INT 15	UART1
INT 16	Smart Card Interface
INT 17	SSI CHA
INT 18	SSI CHB
INT 19	TIMER CHA
INT 20	TIMER CHB
INT 21	TIMER CHC
INT 22	TIMER CHD
INT 23	TIMER CHE
INT 24	TIMER CHF
INT 25	Software Interrupt

The Users can set the active mode of all interrupt source inputs. The default mode is the falling-edge trigger mode. Any inversion or latching required to provide edge sensitivity must be provided at the generating source of the interrupt.

No hardware priority scheme or any form of interrupt vectoring is provided, but the priority can be determined using FIQ mask register and IRQ mask register under software control.

FIQ mask register and IRQ mask register are also provided to generate an interrupt under software control. Typically these registers may be used to determine either a FIQ interrupt or an IRQ interrupt.

3.2 Interrupt Control

The interrupt controller provides interrupt source status and interrupt request status. The interrupt mask registers are used to determine whether an active interrupt source should generate an interrupt request to the processor or not. A logic HIGH in the interrupt mask register indicates that the interrupt source is masked and then doesn't generate a request.

FIQ mask register and IRQ mask register indicate whether the interrupt source causes a processor interrupt or not. The interrupt mode is configured by interrupt trigger mode register and interrupt trigger polarity register. And Interrupt direction register indicates whether each interrupt source drives IRQ or FIQ.

The FIQ and IRQ status register is used to reflect the status of all channels set to produce an FIQ interrupt or IRQ interrupt. And the status registers are cleared by writing '1' to the status clear register at the edge trigger mode only.

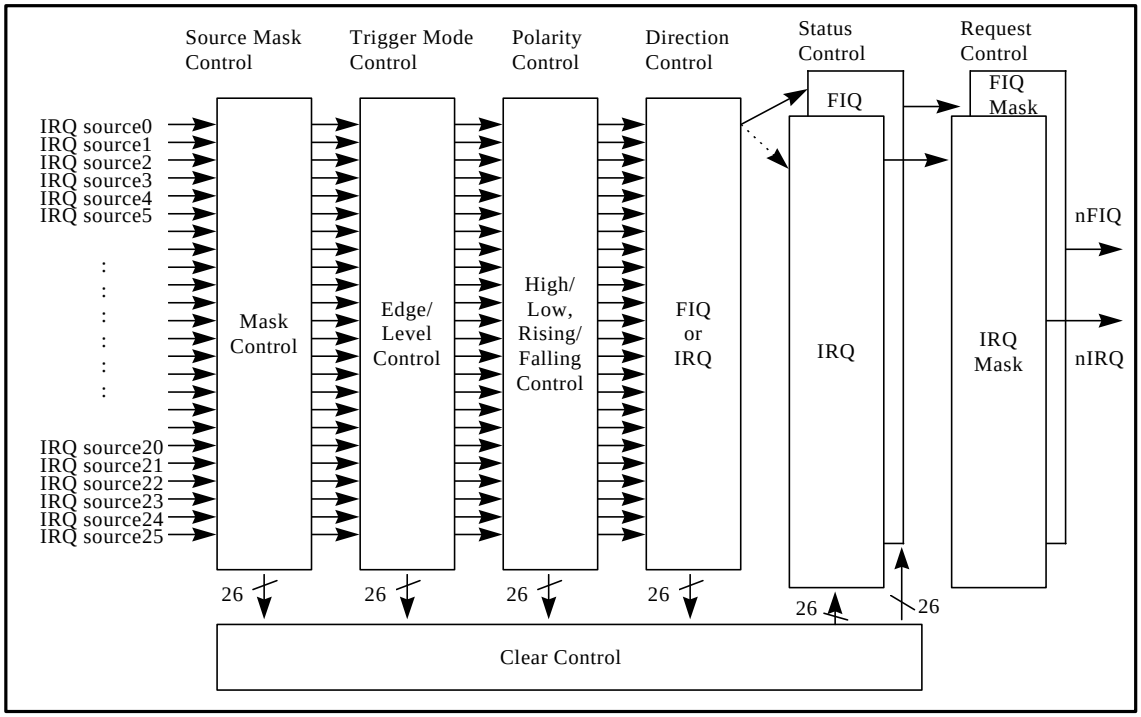


Figure 3. Interrupt Control Flow Diagram

TIC registers are used only for the production test. TIC input register is used to drive interrupt request sources by the CPU. When this register bit 26 is set, other bits of TIC input register are regarded as interrupt sources. This bit is cleared by system reset and should be cleared in normal operation.

Bit 25 is used as a software interrupt source. When source mask control register bit 25 is HIGH, an interrupt request occurs. To disable the software interrupt, Source Mask Control Register bit 25 should be Low. Software interrupt source input is fixed active HIGH and level sensitive.

4. Interrupt Controller Memory Map

The base address of the interrupt controller is 0xFFFF F200. The offset of any particular register from the base address is fixed.

Table 3. Memory Map of the Interrupt Controller APB Peripheral

ADDRESS	READ LOCATION	WRITE LOCATION
IntBase + 0x000	Mask Register	Mask Register
IntBase + 0x004	Trigger Mode Register	Trigger Mode Register
IntBase + 0x008	Trigger Polarity Register	Trigger Polarity Register
IntBase + 0x00C	Direction Register	Direction Register
IntBase + 0x010	FIQ Status Register (Read-only)	
IntBase + 0x014	IRQ Status Register (Read-only)	
IntBase + 0x018	FIQ Mask Register	FIQ Mask Register
IntBase + 0x01C	IRQ Mask Register	IRQ Mask Register
IntBase + 0x020		Status Clear Register (Write-only)
IntBase + 0x024		TicInput Register
IntBase + 0x028	TicOutputRegister	

5. Interrupt Controller Register Descriptions

The following registers are provided for both FIQ and IRQ interrupt controllers:

(1) Mask Register

Readable and Writable. The interrupt mask register is used to mask the interrupt input sources and defines which active sources will generate an interrupt request to the processor. If certain bits within the interrupt controller are not implemented, the corresponding bits in the interrupt mask register must be masked. A bit value 0 indicates that the interrupt is unmasked and will allow an interrupt request to reach the processor. A bit value 1 indicates that the interrupt is masked. Once a bit is masked, the corresponding bit in the status register is cleared. On reset, all interrupt input sources are masked.

'1' : Mask
'0' : Unmask
Initial value : 0x3FFFFFF

25																								0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

(2) Trigger Mode Register

Readable and Writable. The interrupt trigger mode register is used to configure the interrupts with the interrupt trigger polarity register. Each interrupt can be configured to level or edge triggered. A bit value 0 indicates that the interrupt is configured to edge triggered and a bit value 1 indicates that the interrupt is configured to level triggered. On reset, all interrupt input sources are configured to edge triggered.

'1' : Level Trigger Mode
'0' : Edge Trigger Mode
Initial value : 0x2000000

25																								0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(3) Trigger Polarity Register

Readable and Writable. The interrupt trigger polarity register is used to configure the interrupts with the interrupt trigger mode register. Each interrupt can be configured to rising/high or falling/low active. A bit value 0 indicates that the interrupt is configured to falling active for edge trigger mode and to low active for level trigger mode. A bit value 1 indicates that the interrupt is configured to rising active for edge trigger mode and to high active for level trigger mode. On reset, all interrupt input sources are configured to falling/low active.

'1' : Rising or High
'0' : Falling or Low
Initial value : 0x2000000

25																								0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4. Interrupt Source Trigger Mode of the Interrupt Controller

TRIGGER MODE REGISTER	TRIGGER POLARITY REGISTER	DESCRIPTION
0	0	Falling-Edge (Default)
0	1	Rising-Edge
1	0	Low-Level
1	1	High-Level

(4) Direction Register

Readable and Writable. The interrupt direction register is used to determine whether each interrupt source drives IRQ or FIQ. A bit value 0 indicates that the interrupt is driven to IRQ and a bit value 1 indicates that the interrupt is driven to FIQ. On reset, all interrupt input sources drive IRQ.

'1' : Request FIQ
'0' : Request IRQ
Initial value : 0x0000000

25																								0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(5) FIQ Status Register

Read-only. The FIQ status register is used to reflect the status of all channels set to produce an FIQ interrupt (IDR(i) = 1). When an interrupt is set for an FIQ occurring, the corresponding bit is set in FIQ status register. The interrupt handler will examine this register to determine the channel(s) that caused the FIQ interrupt. When the status clear register is written to '1', the corresponding bit is cleared if that channel is configured to edge trigger mode. A HIGH bit indicates that the interrupt is active and will generate an interrupt to the processor.

'1' : Interrupt Event Occur
'0' : No Interrupt Event
Initial value : 0x0000000

25																								0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(6) IRQ Status Register

Read-only. The IRQ status register is used to reflect the status of all channels set to produce an IRQ interrupt (IDR(i) = 0). When an interrupt is set for an IRQ occurring, the corresponding bit is set in IRQ status register. The interrupt handler will examine this register to determine the channel(s) that caused the IRQ interrupt. When the status clear register is written to '1', the corresponding bit is cleared if that channel is configured to edge trigger mode. A HIGH bit indicates that the interrupt is active and will generate an interrupt to the processor.

'1' : Interrupt Event Occur
'0' : No Interrupt Event
Initial value : 0x0000000

25																								0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(7) FIQ Mask Register

Readable and Writable. The FIQ request mask register is used to mask the request to generate an interrupt to a processor. If certain bits within the interrupt controller are not implemented, the corresponding bits in the FIQ request mask register must be masked. A bit value 0 indicates that the interrupt is unmasked and will allow an interrupt request to reach the processor. A bit value 1 indicates that the interrupt is masked. On reset, all FIQ requests are unmasked.

'1' : Request Mask
'0' : Request Unmask
Initial value : 0x0000000

25																								0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(8) IRQ Mask Register

Readable and Writable. The IRQ request mask register is used to mask the request to generate an interrupt to a processor. If certain bits within the interrupt controller are not implemented, the corresponding bits in the IRQ request mask register must be masked. A bit value 0 indicates that the interrupt is unmasked and will allow an interrupt request to reach the processor. A bit value 1 indicates that the interrupt is masked. On reset, all IRQ requests are unmasked.

'1' : Request Mask
'0' : Request Unmask
Initial value : 0x0000000

25																								0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(9) Status Clear Register

Write-only. The status clear register is used to clear bits in the status register configured to the edge trigger mode. If the channels are configured to the level trigger mode, the corresponding bits in the FIQ status register and the IRQ status register have no effect. This register is cleared when the signal, P_STB, is LOW after this register is written to '1'. When writing to this register, each data bit that is HIGH causes the corresponding bit in the status register to be cleared. Data bits that are LOW have no effect on the corresponding bit in the status register. Note that the status clear register has an effect on the status register in the edge trigger mode.

'1' : Clear the status register
'0' : Not clear
Initial value : 0x0000000

25																								0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Section 11. Real Time Clock

1. General Description

This module is a 32-bit counter clocked by a 32.768KHz clock. This clock needs to be provided by the system, since there is no oscillator inside the block. The clock is divided in the RTC core to provide a 1Hz clock used to drive a 32-bit counter which forms the Real Time Clock (RTC). It also contains a 32-bit match register which 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). The RTC has one event output which is synchronized with PCLK. RTCIRQ is to be connected to the system interrupt controller.

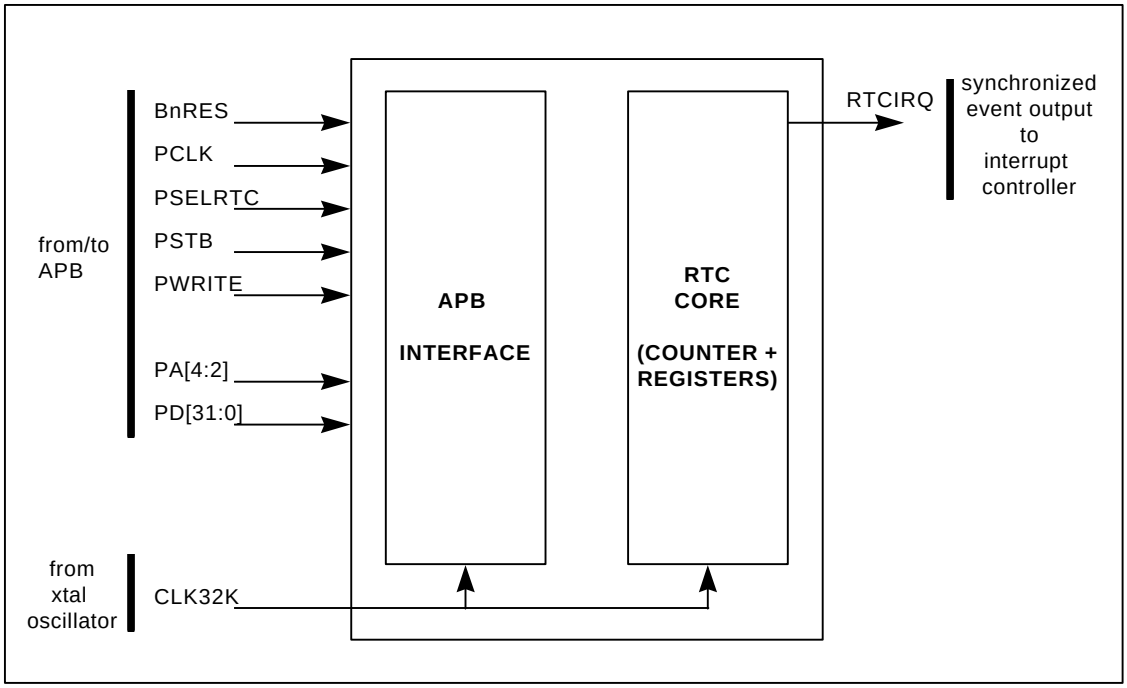


Figure 1. Real Time Clock Connections Diagram

2. Signal Description

The RTC module is connected to the APB bus. Table 1. APB signal descriptions describes the APB signals used and produced.

Table 1. APB Signal Descriptions

NAME	TYPE	SOURCE/ DESTINATION	DESCRIPTION
PCLK	I	Power Management Unit	The slow APB clock used to re-synchronize data is transferred between the 32.768KHz clock and the APB.
P_A[4:2]	I	APB Bridge	This is the peripheral address bus, which is used by an individual peripheral for decoding register accesses to this peripheral. The addresses become valid before P_STB goes to HIGH and remain valid after P_STB goes to LOW.
P_D[31:0]	I/O	APB Peripherals, B_D bus	This is the bi-directional peripheral data bus. The data bus is driven by this block during read cycles (when P_WRITE is LOW).
P_STB	I	APB Bridge	This strobe signal is used to time all accesses on the peripheral bus. The falling edge of P_STB is coincident with the falling edge of B_CLK.
P_WRITE	I	APB Bridge	When this signal is HIGH, it indicates a write to a peripheral. When this signal is LOW, it indicates a read from a peripheral. This signal has the same timing as the peripheral address bus. It becomes valid before P_STB goes to HIGH and remains valid after P_STB goes to LOW.
P_SEL	I	APB Bridge	When this signal is HIGH, it indicates that this module has been selected by the APB bridge. This selection is a decode of the system address bus (ASB). See AMBA Peripheral Bus Controller for more details.
BnRES	I	Power Management Unit	Reset signal generated from the PMU
RTCIRQ	O	Interrupt Controller	Interrupt signal to the Interrupt module. When this signal is HIGH, it indicates a valid comparison between the counter value and the match register. It also indicates 1Hz interval with enable bit in control register.

3. Hardware Interface

The APB interface is fully APB-compliant. The APB is a non-pipelined low-power interface designed to provide a simple interface to slave peripherals.

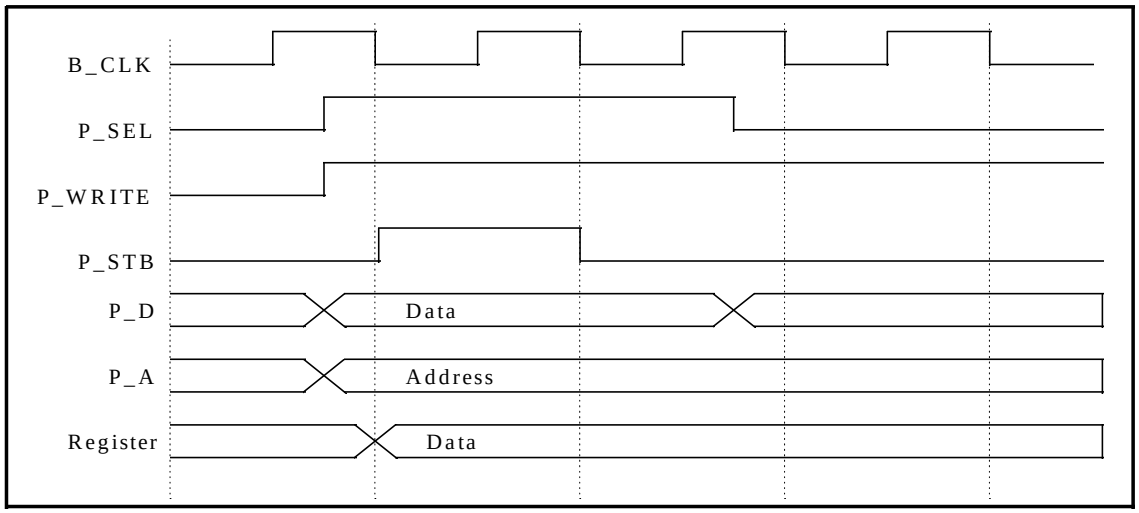


Figure 2. RTC Module APB Write Cycle

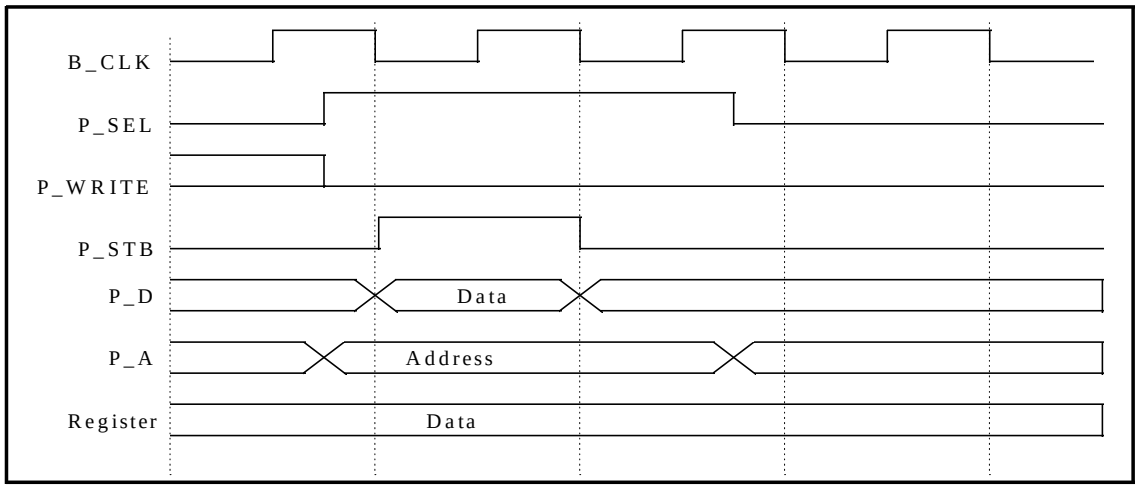


Figure 3. RTC Module APB Read Cycle

4. Functional Description

The counter is loaded by writing it to the RTC data register. The counter will count up on each rising edge of the clock and loops back with 0 when the maximum value (0xFFFFFFFF) is reached. At any moment the counter value can be obtained by reading the RTC data register.

The value of the match register can also be read at any time, and the read does not affect the counter value. The status of the interrupt signal is available in the status register. The status 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.

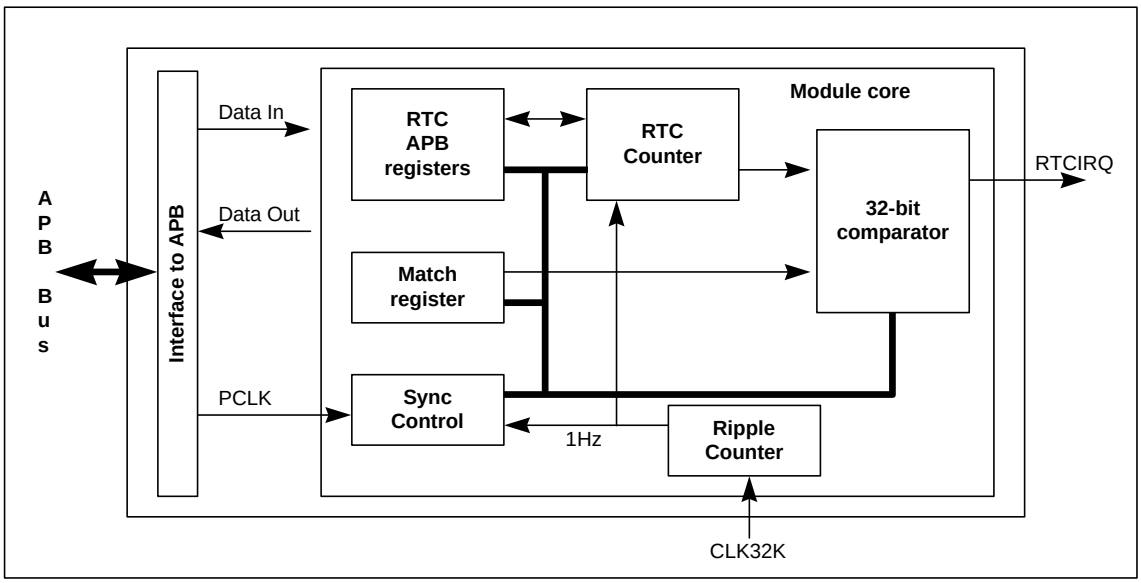


Figure 4. RTC Block Diagram

5. Real Time Clock Memory Map

The base address of the RTC is fixed as 0xFFFF F300 and the offset of any particular register from the base address is fixed.

Table 2. RTC Memory Map

ADDRESS	READ LOCATION	WRITE LOCATION
RTC Base + 0x00	RTC data register (RTCDR)	RTC data register (RTCDR)
RTC Base + 0x04	RTC match register (RTCMR)	RTC match register (RTCMR)
RTC Base + 0x08	RTC status (RTCS)	
RTC Base + 0x0C	RTC clock divider (RTCDV)	RTC clock divider (RTCDV)
RTC Base + 0x10	RTC control register (RTCCR)	RTC control register (RTCCR)
RTC Base + 0x14		RTC Tic selection register (RTCTS)
RTC Base + 0x18		TicCLK32K
RTC Base + 0x1C		TicCLKPCLK

Note The RTC clock divider register may only be written to when in test mode.

6. Real Time Clock Register Descriptions

The following user registers are provided :

RTC Data Register (RTCDR)

Read/Write. Writing to this 32-bit register will load the counter. A read will give the current value of the counter.

RTC Match Register (RTCMR)

Read/Write. Writing to this 32-bit register will load the match register. This value can also be read back.

RTC Status Register (RTCS)

Read-only. When performing a read from this location the interrupt flag will be cleared. If a match event occurs, bit[1] will be set. For a second event, bit[0] will be set. This register is affected by the control register.

RTC Clock Divider (RTCDV)

Read/Write. The reads to the register will return only four bits of the clock divider output. Bits [3:0] will return bits (14, 11, 7, 3) of the divider output. Writing zero to bit[0] clears this divider.

RTC Control Register (RTCCR)

Read/Write. This register enables the interrupt. Bit[1] enables the match event interrupt (default disable = 0). Bit[0] enables second event interrupt (default disable = 0).

RTC Tic Selection (RTCTS)

Write-only. This register is for production test purposes. Bit[0] enables TicCLK32K for 32kHz clock replacement. Bit[1] enables TicCLKPCLK for PCLK clock replacement.

TicCLK32K

Write-only. This generates 32kHz clock for production test purposes.

TicCLKPCLK

Write-only. This generates PCLK clock for production test purposes.

Section 12. General Purpose Timer Unit

1. General Description

The general-purpose timer unit has:

- Six channels with 16bit counter
- 12 different pulse outputs and 12 different pulse inputs
- Independent function with 12 general registers
- Compare match waveform output function
- Input capture function
- Counter-clearing function at compare match or input capture mode
- Synchronizing mode
- PWM mode
- 18 interrupt sources
- Selectable 4 internal clock sources and 4 external clock sources

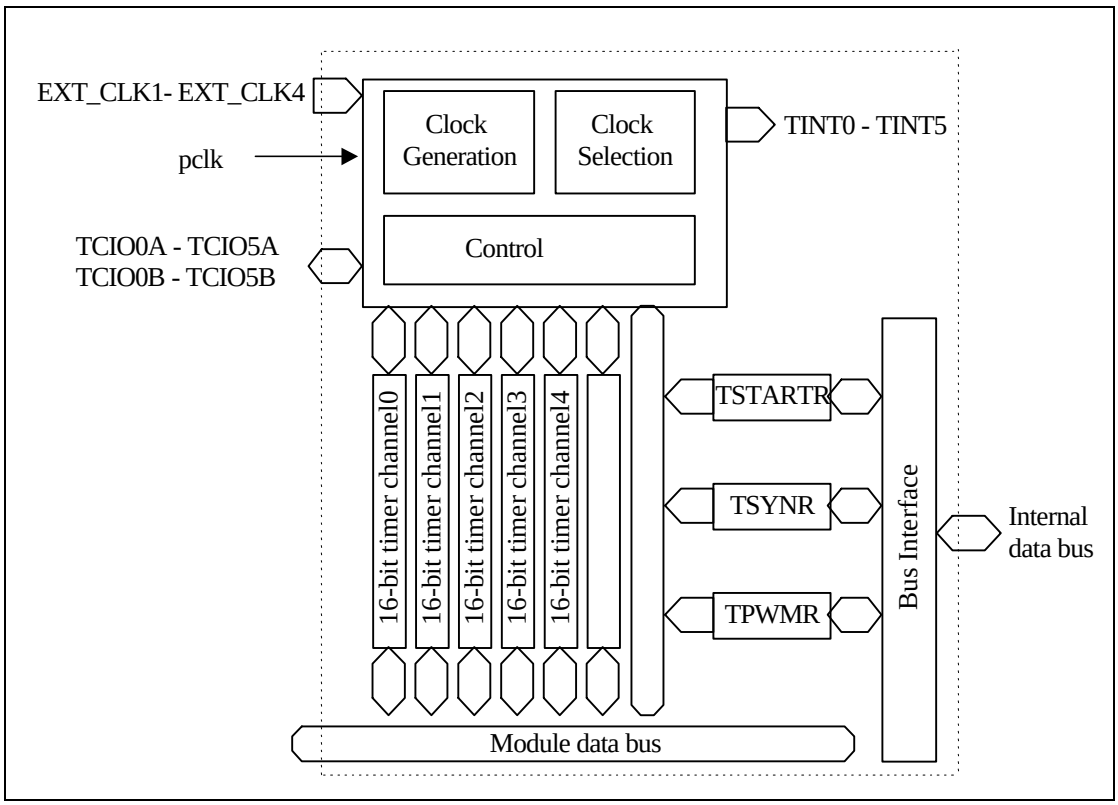


Figure 1. General-purpose Timer Unit Module Block Diagram

2. Hardware Interface and Signal Description

The General-purpose Timer Unit module is connected to the APB bus.

Table 1. APB Signal Descriptions

NAME	TYPE	SOURCE/ DESTINATION	DESCRIPTION
PCLK	I	PMU	Peripheral clock. This clock times all bus transfers.
BnRES	I	PMU	Reset signal generated from the PMU
PA[7:2]	I	APB Bridge	This is the peripheral address bus, which is used by an individual peripheral for decoding register accesses to that peripheral. The addresses become valid before PSTB goes to HIGH and remain valid after PSTB goes to LOW.
PD[31:0]	I/O	APB Peripherals, B_D bus	This is the bi-directional peripheral data bus. This block drives the data bus during read cycles (when PWRITE is LOW).
PSTB	I	APB Bridge	This strobe signal is used to time all accesses on the peripheral bus. The falling edge of PSTB is coincident with the falling edge of PCLK.
PWRITE	I	APB Bridge	When this signal is HIGH, it indicates a write to a peripheral. When this signal is LOW, it indicates a read from a peripheral. This signal has the same timing as the peripheral address bus. It becomes valid before PSTB goes to HIGH and remains valid after PSTB goes to LOW.
PSEL	I	APB Bridge	When this signal is HIGH, it indicates that the APB bridge has selected this module. This selection is a decode result of the system address bus (ASB). See AMBA Peripheral Bus Controller for more details.
EXT_CLK1	I	External	External clock1 input. This signal is selected independently from EXT_CLK2, EXT_CLK3, and EXT_CLK4.
EXT_CLK2	I	External	External clock2 input. This signal is selected independently from EXT_CLK1, EXT_CLK3, and EXT_CLK4.
EXT_CLK3	I	External	External clock3 input. This signal is selected independently from EXT_CLK1, EXT_CLK2, and EXT_CLK4
EXT_CLK4	I	External	External clock4 input. This signal is selected independently from EXT_CLK1, EXT_CLK2, and EXT_CLK3.
TCIO0A	I/O	External	This signal is used as GRA0 input in input capture mode, GRA0 output in output compare mode, and PWM output in PWM mode.
TCIO0B	I/O	External	This signal is used as GRB0 input in input capture mode, GRB0 output in output compare mode, and PWM output in PWM mode.
TCIO1A	I/O	External	This signal is used as GRA1 input in input capture mode, GRA1 output in output compare mode, and PWM output in PWM mode.
TCIO1B	I/O	External	This signal is used as GRB1 input in input capture mode, GRB1 output in output compare mode, and PWM output in PWM mode.
TCIO2A	I/O	External	This signal is used as GRA2 input in input capture mode, GRA2 output in output compare mode, and PWM output in PWM mode.
TCIO2B	I/O	External	This signal is used as GRB2 input in input capture mode, GRB2 output in output compare mode, and PWM output in PWM mode.

NAME	TYPE	SOURCE/ DESTINATION	DESCRIPTION
TCIO3A	I/O	External	This signal is used as GRA3 input in input capture mode, GRA3 output in output compare mode, and PWM output in PWM mode.
TCIO3B	I/O	External	This signal is used as GRB3 input in input capture mode, GRB3 output in output compare mode, and PWM output in PWM mode.
TCIO4A	I/O	External	This signal is used as GRA4 input in input capture mode, GRA4 output in output compare mode, and PWM output in PWM mode.
TCIO4B	I/O	External	This signal is used as GRB4 input in input capture mode, GRB4 output in output compare mode, and PWM output in PWM mode.
TCIO5A	I/O	External	This signal is used as GRA5 input in input capture mode, GRA5 output in output compare mode, and PWM output in PWM mode.
TCIO5B	I/O	External	This signal is used as GRB5 input in input capture mode, GRB5 output in output compare mode, and PWM output in PWM mode.
TINT0	O	Interrupt Controller	Interrupt signal to the Interrupt Controller module. This signal indicates that an interrupt has been generated in channel0.
TINT1	O	Interrupt Controller	Interrupt signal to the Interrupt Controller module. This signal indicates that an interrupt has been generated in channel1.
TINT2	O	Interrupt Controller	Interrupt signal to the Interrupt Controller module. This signal indicates that an interrupt has been generated in channel2.
TINT3	O	Interrupt Controller	Interrupt signal to the Interrupt Controller module. This signal indicates that an interrupt has been generated in channel3.
TINT4	O	Interrupt Controller	Interrupt signal to the Interrupt Controller module. This signal indicates that an interrupt has been generated in channel4.
TINT5	O	Interrupt Controller	Interrupt signal to the Interrupt Controller module. This signal indicates that an interrupt has been generated in channel5.

The writes to the General-purpose Timer Unit module are generated from the Peripheral Bus Controller module. Figure 2. General-purpose timer unit module APB write cycle summarizes this.

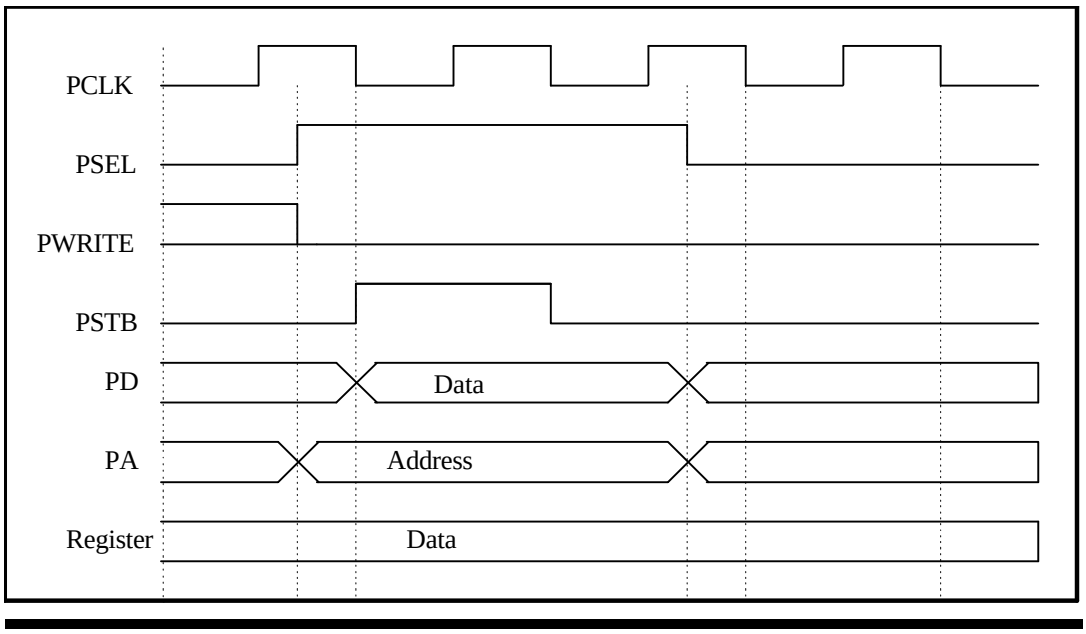


Figure 2. General-Purpose Timer Unit Module APB Read Cycle

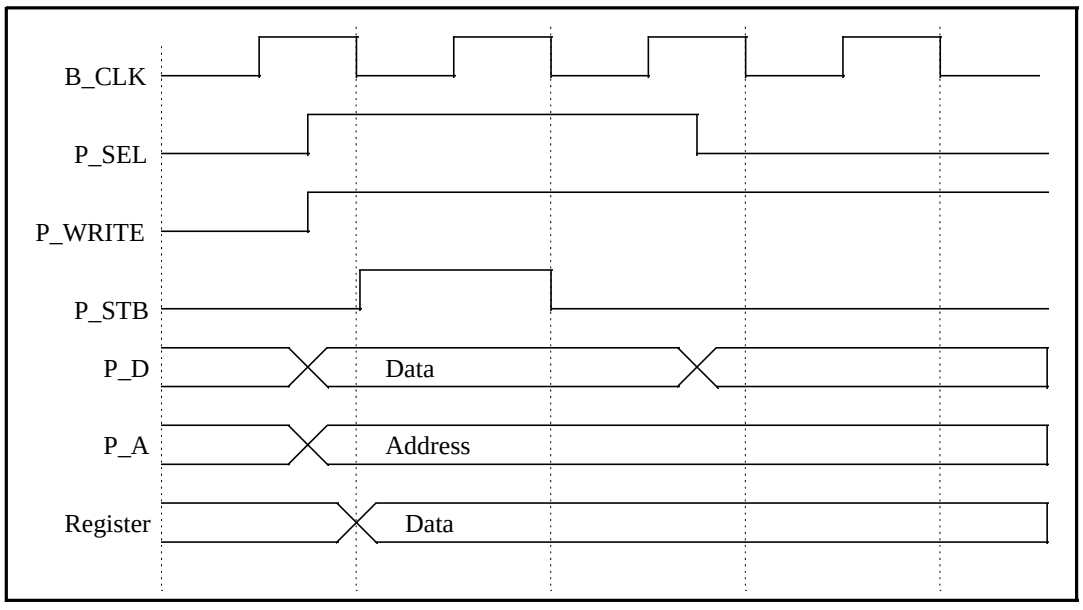


Figure 3. General-Purpose Timer Unit Module APB Write Cycle

3. General Purpose Timer Unit Introduction

The GDC21D601 has a general-purpose timer unit (GPTU) with six channels of 16-bit timer. There are two counter operation modes: a free running mode and a periodic mode. And each channel has independent operating modes. There are common functions for each channel: counter operation, input capture, compare match, PWM, and synchronized clear and write.

It is possible to select one of eight counter clock sources for all channels.

- Internal clock : counting at falling edge
  - BCLK / 2
  - BCLK / 4
  - BCLK / 16
  - BCLK / 64
- External clock: counting at rising, falling, or both edge that are user-selectable.

There are four kinds of counter clear sources which can be selected by user's setting.

- None : never clear until overflow for free running mode
- GRA match or TPA input capture
- GRB match or TPB input capture
- Synchronous clear

4. General Purpose Timer Unit Operation

The operation modes are described below.

Free Running Mode

Each channel can run from 0 to FFFF repeatedly. When it reaches FFFF, the interrupt signal is generated as user's setting.

Compare Match Mode

Each channel has 2 general registers and user can read or write from/to the registers. If user wrote some values to general register, and the counter reached that value, the channel generates interrupt and external output by user's setting. The output value can be '1', '0', or toggle value. The counter can be cleared by user's setting when the match with general register is detected.

Input Capture Mode

When set to input capture mode and rising any event at TPA or TPB, the counter value is transferred to GRA or GRB respectively. The interrupt can be generated and the external event may be rising edge, falling edge or any edge by user's setting. The counter can be cleared by user's setting when the event at TPA or TPB is detected.

Synchronized Clear & Write Mode

When some channels are set to synchronization mode, and one of them is cleared by compare match or input capture, the other channels can be cleared simultaneously by user's setting. When some channels are set to synchronization mode and user would write any value to one of them, the other channels can be written with same value simultaneously by user's setting.

PWM Mode

When a channel is set to PWM mode, the channel operates like a compare match mode and the output on compare match event is generated only at TPA. The TPA value is '1' when it is the match with GRA, and '0' when it is the match with GRB.

5. General Purpose Timer Unit Memory Map

The base address of the general-purpose timer unit is 0xFFFFF400 and the offset of any particular register from the base address is fixed.

Table 2. General Purpose Timer Unit Register Memory Map

ADDRESS	READ LOCATION	WRITE LOCATION
Gptu Base + 0x00	TSTARTR	TSTARTR
Gptu Base + 0x04	TSYNCR	TSYNCR
Gptu Base + 0x08	TPWMR	TPWMR
Gptu Base + 0x0C		TSTINR
Gptu Base + 0x10	TSTOUTR	
Gptu Base + 0x14		TSTMODER
Gptu Base + 0x18	TSTINTR	
Gptu Base + 0x20	TCONTR0	TCONTR0
Gptu Base + 0x24	TIOCR0	TIOCR0
Gptu Base + 0x28	TIER0	TIER0
Gptu Base + 0x2C	TSTATUSR0	
Gptu Base + 0x30	TCOUNT0	TCOUNT0
Gptu Base + 0x34	GRA0	GRA0
Gptu Base + 0x38	GRB0	GRB0
Gptu Base + 0x40	TCONTR1	TCONTR1
Gptu Base + 0x44	TIOCR1	TIOCR1
Gptu Base + 0x48	TIER1	TIER1
Gptu Base + 0x4C	TSTATUSR1	
Gptu Base + 0x50	TCOUNT1	TCOUNT1
Gptu Base + 0x54	GRA1	GRA1
Gptu Base + 0x58	GRB1	GRB1
Gptu Base + 0x60	TCONTR2	TCONTR2
Gptu Base + 0x64	TIOCR2	TIOCR2
Gptu Base + 0x68	TIER2	TIER2
Gptu Base + 0x6C	TSTATUSR2	
Gptu Base + 0x70	TCOUNT2	TCOUNT2
Gptu Base + 0x74	GRA2	GRA2
Gptu Base + 0x78	GRB2	GRB2
Gptu Base + 0x80	TCONTR3	TCONTR3
Gptu Base + 0x84	TIOCR3	TIOCR3
Gptu Base + 0x88	TIER3	TIER3
Gptu Base + 0x8C	TSTATUSR3	
Gptu Base + 0x90	TCOUNT3	TCOUNT3
Gptu Base + 0x94	GRA3	GRA3
Gptu Base + 0x98	GRB3	GRB3
Gptu Base + 0xA0	TCONTR4	TCONTR4
Gptu Base + 0xA4	TIOCR4	TIOCR4
Gptu Base + 0xA8	TIER4	TIER4
Gptu Base + 0xAC	TSTATUSR4	
Gptu Base + 0xB0	TCOUNT4	TCOUNT4
Gptu Base + 0xB4	GRA4	GRA4
Gptu Base + 0xB8	GRB4	GRB4

ADDRESS	READ LOCATION	WRITE LOCATION
Gptu Base + 0xD0	TCONTR5	TCONTR5
Gptu Base + 0xD4	TIOCR5	TIOCR5
Gptu Base + 0xD8	TIER5	TIER5
Gptu Base + 0xDC	TSTATUSR5	
Gptu Base + 0xE0	TCOUNT5	TCOUNT5
Gptu Base + 0xE4	GRA5	GRA5
Gptu Base + 0xE8	GRB5	GRB5

6. General Purpose Timer Unit Register Descriptions

The following registers are provided for general purpose timer unit :

Timer Start Register (TSTARTR)

Eight-bit readable and writable register that starts and stops the counter of each channel.

Table 3. TSTARTR Bit Description

BIT	INITIAL VALUE		FUNCTION
7 (reserved)	1		
6 (reserved)	1		
5 (STR5)	0		
4 (STR4)	0	1 = start counting	start and stop counting
3 (STR3)	0	0 = stop counting	
2 (STR2)	0		
1 (STR1)	0		
0 (STR0)	0		

Timer Synch. Register (TSYNCR)

Eight-bit readable and writable register that selects timer synchronizing mode for each channel.

Table 4. TSYNCR Bit Description

BIT	INITIAL VALUE		FUNCTION
7 (reserved)	1		
6 (reserved)	1		
5 (SYNC5)	0		
4 (SYNC4)	0	0 = operate independently	select the synchronizing mode
3 (SYNC3)	0	1 = operate synchronously with	
2 (SYNC2)	0	other sync. channel	
1 (SYNC1)	0		
0 (SYNC0)	0		

Timer PWM Mode Register (TPWMR)

Eight-bit readable and writable registers that select the PWM mode for each channel.

Table 5. TPWMR Bit Description

BIT	INITIAL VALUE		FUNCTION
7 (reserved)	1		
6 (reserved)	1		
5 (PWM5)	0		
4 (PWM4)	0	0 = operate normally	select the PWM mode
3 (PWM3)	0	1 = operate in PWM mode	
2 (PWM2)	0		
1 (PWM1)	0		
0 (PWM0)	0		

Timer Control Register (TCONTR)

Eight-bit readable and writable register for each channel that selects the timer counter clock source, the edges of the external clock source, and the counter clear source.

Table 6. TCONTR Bit Description

BIT	INITIAL VALUE		FUNCTION
7 (reserved)	1		
6 (CCLR1) 5 (CCLR0)	0 0	00 = not cleared - Free running mode 01 = cleared by GRA compare match or input capture - Periodic mode 10 = cleared by GRB compare match or input capture - Periodic mode 11 = cleared in synchronization with other sync. timer	Select the counter clear source
4 (reserved)	1		
3 (reserved)	1		
2 (TPSC2) 1 (TPSC1) 0 (TPSC0)	0 0 0	000 = internal clock 1 (BCLK/2) 001 = internal clock 2 (/4) 010 = internal clock 3 (/16) 011 = internal clock 4 (/64) 100 = external clock 1 (Ext_clk1) 101 = external clock 2 (Ext_clk2) 110 = external clock 3 (Ext_clk3) 111 = external clock 4 (Ext_clk4)	select the counter clock source

Timer I/O Control Register (TIOCR)

Eight-bit readable and writable register that selects the output compare or input capture function for GRA and GRB, and selects the function of the TP#A and TP#B pins. TIOCR# controls the GRs.

Table 7. TIOCR Bit Description

BIT	INITIAL VALUE		FUNCTION
7 (reserved)	1		
6 (IOB2) 5 (IOB1) 4 (IOB0)	0 0 0	000 = compare match with pin output disabled 001 = 0 output at GRB compare match 010 = 1 output at GRB compare match 011 = toggle output at GRB compare match 100 = GRB captures the rising edge of input 101 = GRB captures the falling edge of input 110 = GRB captures both edge of input	select the GRB function
3 (reserved)	1		
2 (IOA2) 1 (IOA1) 0 (IOA0)	0 0 0	000 = compare match with pin output disabled 001 = 0 output at GRA compare match 010 = 1 output at GRA compare match 011 = toggle output at GRA compare match 100 = GRA captures rising edge of input 101 = GRA captures falling edge of input 110 = GRA captures both edge of input	select the GRA function

Timer Interrupt Enable Register (TIER)

Eight-bit readable and writable register that controls the enabling/disabling of overflow interrupt request and the general register compare match/input capture interrupt requests. TIER# controls the interrupt enable/disable.

Table 8. TIER Bit Description

BIT	INITIAL VALUE		FUNCTION
7 (reserved)	1		
6 (reserved)	1		
5 (reserved)	1		
4 (reserved)	1		
3 (reserved)	1		
2 (OVFIE)	0	0 = disable interrupt requests by the OVFI 1 = enable interrupt requests from the OVFI	
1 (MCIBE)	0	0 = disable interrupt requests by the MCIB 1 = enable interrupt requests from the MCIB	
0 (MCIAE)	0	0 = disable interrupt requests by the MCIA 1 = enable interrupt requests from the MCIA	

Timer Status Register (TSTATUSR)

Eight-bit readable register contains the flags that indicate TCOUNT overflow/underflow and GRA/GRB compare match or input capture. This flags are interrupt sources.

Table 9. TIER Bit Description

BIT	INITIAL VALUE		FUNCTION
7 (reserved)	1		
6 (reserved)	1		
5 (reserved)	1		
4 (reserved)	1		
3 (reserved)	1		
2 (OVFI)	0	0 = clear condition 1 = setting condition	indicate TCOUNT overflow/underflow
1 (MCIB)	0		indicate a GRB compare match or input capture
0 (MCIA)	0		indicate a GRA compare match or input capture

Timer Counter (TCOUNT)

16-bit readable and writable counter. The clock source is selected by TCONTR of each channel. TCOUNT is cleared to 0x0000 by compare match with the corresponding GRA or GRB, or by input capture to GRA or GRB. When TCOUNT is overflow or underflow, OVFI in the TSTATUSR is set to '1'.

TCNT0 (16 bit) : upcounter

TCNT1 (16 bit) : upcounter

TCNT2 (16 bit) : upcounter

TCNT0 (16 bit) : upcounter

TCNT0 (16 bit) : upcounter

TCNT0 (16 bit) : upcounter

General Register A, B (GRA, GRB)

16-bit readable and writable register. There are 2 general registers for each channel (total 12). Each general register can function as either an output compare register or an input capture register by setting it in the TIOCR.

7. Examples of Register Setting

7.1 Six Channels

Channel 0 : In free-running counter (Compare match - 0 output at GRB and 1 output at GRA)

Channel 1 : In a periodic counter cleared by GRB (Compare match - Toggle output at GRA and GRB)

Channel 2 : In a periodic counter cleared by TPB (Input capture - TPA with both edges, TPB with the falling edge)

Channel 3 : In a periodic counter cleared by GRA (PWM mode)

Channel 4 : In a periodic counter cleared by GRA (PWM mode : duty cycle 0%)

Channel 5 : In a periodic counter cleared by GRA (PWM mode : duty cycle 100%)

Setting example

RESET

```
.
.
TSTART = 0xC0;
TCONTR0 = 0x81;      //internal clock2
TCONTR1 = 0xC2;      //internal clock3
TCONTR2 = 0xC3;      //internal clock4
TCONTR3 = 0xA5;      //external clock2 - rising edge
TCONTR4 = 0xCE;      //external clock3 - falling edge
TCONTR5 = 0xBF;      //external clock4 - both edge
TIER0 = 0xFB;        //enable interrupt requests from the MCIA, MCIB
TIER1 = 0xFA;        //enable interrupt requests from the MCIA
TIER2 = 0xFA;        //enable interrupt requests from the MCIA
TIER3 = 0xFB;        //enable interrupt requests from the MCIA, MCIB
TIER4 = 0xFB;        //enable interrupt requests from the MCIA, MCIB
TIER5 = 0xFB;        //enable interrupt requests from the MCIA, MCIB
TIOCR0 = 0x9A;
TIOCR1 = 0xBB;
TIOCR2 = 0xDF;
TCOUNT0 = 0xFFF0;
TCOUNT1 = 0xFFF0;
TCOUNT2 = 0x0000;
TCOUNT3 = 0x0000;
TCOUNT4 = 0x0000;
TCOUNT5 = 0x0000;
GRA0 = 0xFFF4;
GRA1 = 0xFFF2;
GRA3 = 0x0A;
GRA4 = 0x04;
GRA5 = 0x0A;
GRB0 = 0xFFFA;
GRB1 = 0xFFF5;
GRB3 = 0x04;
GRB4 = 0x0A;
GRB5 = 0x04;
TPWMR = 0xF8;
TSTARTR = 0xFF;
```

.

{Running...}

7.2 Free-Running Mode

Setting example

```
TSTARTR = 0xC0;
TCONTR4 = 0x80;
TIER4 = 0xFC;
TCOUNT4 = 0xFFF0;
TSTARTR = 0xD0;
```

7.3 Periodic Mode

: GRA compare match

Setting example

```
TSTARTR = 0xC0;
TCONTR4 = 0xA0;
TIER4 = 0xF9;
TCOUNT4 = 0x0000;
GRA4 = 0x0F;
TSTARTR = 0xD0;
```

7.4 Synchronizing Mode

: In a periodic mode counter cleared by GRA of channel0
In a periodic mode counter cleared by GRB of channel1
In a periodic mode counter cleared, synchronized with other sync. timer (channel2, 3, 4, 5)
Toggle output at GRA of channel0, 2, 4
Toggle output at GRB of channel1, 3, 5

Setting example

```
TSTARTR = 0xC0;
TCONTR0 = 0xA0;
TCONTR1 = 0xC1;
TCONTR2 = 0xE0;
TCONTR3 = 0xE1;
TCONTR4 = 0xE0;
TCONTR5 = 0xE1;
TIOCR0 = 0x8B;
TIOCR1 = 0xB8;
TIOCR2 = 0x8B;
TIOCR3 = 0xB8;
TIOCR4 = 0x8B;
TIOCR5 = 0xB8;
TIER0 = 0xFD;
TIER1 = 0xFE;
```

```
TIER2 = 0xFD;
TIER3 = 0xFE;
TIER4 = 0xFD;
TIER5 = 0xFE;
TCOUNT0 = 0xFF10;
TCOUNT1 = 0xFF11;
TCOUNT2 = 0xFF12;
TCOUNT3 = 0xFF13;
TCOUNT4 = 0xFF14;
TCOUNT5 = 0xFF15;
GRA0 = 0xFF1A;
GRB1 = 0xFF15;
GRA2 = 0xFF1C;
GRB3 = 0xFF1D;
GRA4 = 0xFF1E;
GRB5 = 0xFF1F;
TSYNCR = 0xFF;
TSTARTR = 0xFF;
```

Section 13. PIO

1. General Description

The PIO is an APB peripheral which provides 80 bits of programmable input /output divided into ten 8-bit ports ; port A, port B, port C, port D, port E, port F, port G, port H, port I, and port J. Each pin is configurable as either input or output. At system reset, port A, C, E, G, I set their defaults to input and port B, D, F, H, J set their defaults to output.

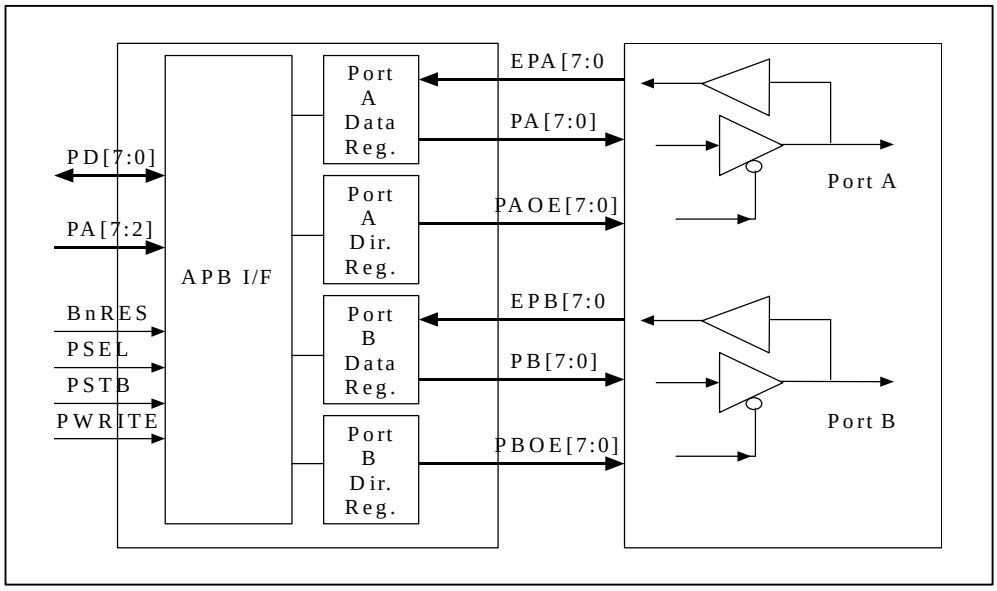


Figure 1. PIO Block Diagram and PADS Connections(Port A and Port B)

Each port has a data register and a data direction register that both are 8 bits wide. The data direction register defines whether each individual pin is an input or an output. The data register is used to read the value of the PIO pins, both input and output, as well as to set the values of pins that are configured as outputs.

2. Signal Description

The PIO module is connected to the APB bus. Table 1. Signal descriptions describe the APB signals used and produced. Table 2. Specific block signal descriptions show the non-AMBA signals from the block.

Table 1. Signal Descriptions

NAME	TYPE	SOURCE/ DESTINATION	DESCRIPTION
BnRES	I	PMU	This signal indicates a power on reset status of the bus (active LOW).
PA[7:2]	I	APB Bridge	This is the part of the peripheral address bus, which is used by the peripheral for decoding its own register accesses. The addresses become valid before PSTB goes to HIGH and remain valid after PSTB goes to LOW.
PD[7:0]	I/O	APB Peripherals, BD bus	This is the part of the bi-directional peripheral data bus. The data bus is driven by this block during read cycles (when PWRITE is LOW).
PSTB	I	APB Bridge	This strobe signal is used to time all accesses on the peripheral bus. The falling edge of PSTB is coincident with the falling edge of BCLK (ASB system clock).
PWRITE	I	APB Bridge	When this signal is HIGH, it indicates a write to a peripheral and when this signal is LOW, it indicates a read from a peripheral. This signal has the same timing as the peripheral address bus. It becomes valid before PSTB goes to HIGH and remains valid after PSTB goes to LOW.
PSEL	I	APB Bridge	When this signal is HIGH, it indicates the PIO module has been selected by the APB bridge. This selection is a decode of the system address bus (ASB). For more details, see AMBA Peripheral Bus Controller

Table 2. Specific Block Signal Descriptions

NAME	TYPE	SOURCE/ DESTINATION	DESCRIPTION
PA[7:0]	O	PADS	Port A output driver. Values written on PADR register are put onto these lines and driven out to the port A pins if the corresponding data direction bits are set to HIGH (PADDR register).
EPA[7:0]	I	PADS	Port A input driver. It reflects the external state of the port. This information is obtained when the PADR register is read.
PAOE[7:0]	O	PADS	Port A output enable (active LOW). Values written on PADDR register are put onto these lines.
PB[7:0]	O	PADS	Port B output driver. Values written on PBDR register are put onto these lines and driven out to the port A pins if the corresponding data direction bits are set to HIGH (PBDDR register).
EPB[7:0]	I	PADS	Port B input driver. It reflects the external state of the port. This information is obtained when the PBDR register is read.
PBOE[7:0]	O	PADS	Port B output enable (active LOW). Values written on PBDDR register are put onto these lines.
PC[7:0]	O	PADS	Port C output driver. Values written on PCDR register are put onto these lines and driven out to the port A pins if the corresponding data direction bits are set HIGH (PCDDR register).
EPC[7:0]	I	PADS	Port C input driver. It reflects the external state of the port. This information is obtained when the PCDR register is read.
PCOE[7:0]	O	PADS	Port C output enable (active LOW). Values written on PCDDR register are put onto these lines.
PD[7:0]	O	PADS	Port D output driver. Values written on PDDR register are put onto these lines and driven out to the port D pins if the corresponding data direction bits are set to HIGH (PDDDR register).
EPD[7:0]	I	PADS	Port D input driver. It reflects the external state of the port. This information is obtained when the PDDR register is read.
PDOE[7:0]	O	PADS	Port D output enable (active LOW). Values written on PDDDR register are put onto these lines.
PE[7:0]	O	PADS	Port E output driver. Values written on PEDR register are put onto these lines and driven out to the port E pins if the corresponding data direction bits are set to HIGH (PEDDR register).
EPE[7:0]	I	PADS	Port E input driver. It reflects the external state of the port. This information is obtained when the PEDR register is read.
PEOE[7:0]	O	PADS	Port E output enable (active LOW). Values written on PEDDR register are put onto these lines.
PF[7:0]	O	PADS	Port F output driver. Values written on PFDR register are put onto these lines and driven out to the port F pins if the corresponding data direction bits are set to HIGH (PFDDR register).
EPF[7:0]	I	PADS	Port F input driver. It reflects the external state of the port. This information is obtained when the PFDR register is read.
PFOE[7:0]	O	PADS	Port F output enable (active LOW). Values written on PFDDR register are put onto these lines.
PG[7:0]	Out	PADS	Port G output driver. Values written on PGDR register are put onto these lines and driven out to the port G pins if the corresponding data direction bits are set to HIGH (PGDDR register).

NAME	TYPE	SOURCE/ DESTINATION	DESCRIPTION
EPG[7:0]	I	PADS	Port G input driver. It reflects the external state of the port. This information is obtained when the PGDR register is read.
PGOE[7:0]	O	PADS	Port G output enable (active LOW). Values written on PGDDR register are put onto these lines.
PH[7:0]	O	PADS	Port H output driver. Values written on PHDR register are put onto these lines and driven out to the port H pins if the corresponding data direction bits are set to HIGH (PHDDR register).
EPH[7:0]	I	PADS	Port H input driver. It reflects the external state of the port. This information is obtained when the PHDR register is read.
PHOE[7:0]	O	PADS	Port H output enable (active LOW). Values written on PHDDR register are put onto these lines.
PI[7:0]	O	PADS	Port I output driver. Values written on PIDR register are put onto these lines and driven out to the port I pins if the corresponding data direction bits are set to HIGH (PIDDR register).
EPI[7:0]	I	PADS	Port I input driver. It reflects the external state of the port. This information is obtained when the PIDR register is read.
PIOE[7:0]	O	PADS	Port I output enable (active LOW). Values written on PIDDR register are put onto these lines.
PJ[7:0]	O	PADS	Port J output driver. Values written on PJDR register are put onto these lines and driven out to the port J pins if the corresponding data direction bits are set to HIGH (PJDDR register).
EPJ[7:0]	I	PADS	Port J input driver. It reflects the external state of the port. This information is obtained when the PJDR register is read.
PJOE[7:0]	O	PADS	Port J output enable (active LOW). Values written on PJDDR register are put onto these lines.

3. Hardware Interface

The APB interface is fully APB-compliant. The APB is a non-pipelined low-power interface, designed to provide a simple interface to slave peripherals.

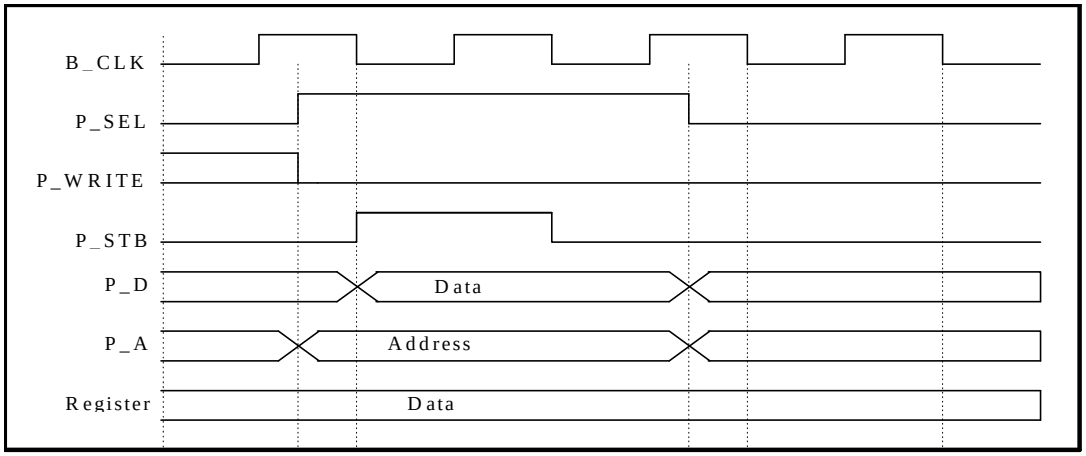


Figure 1. APB Read

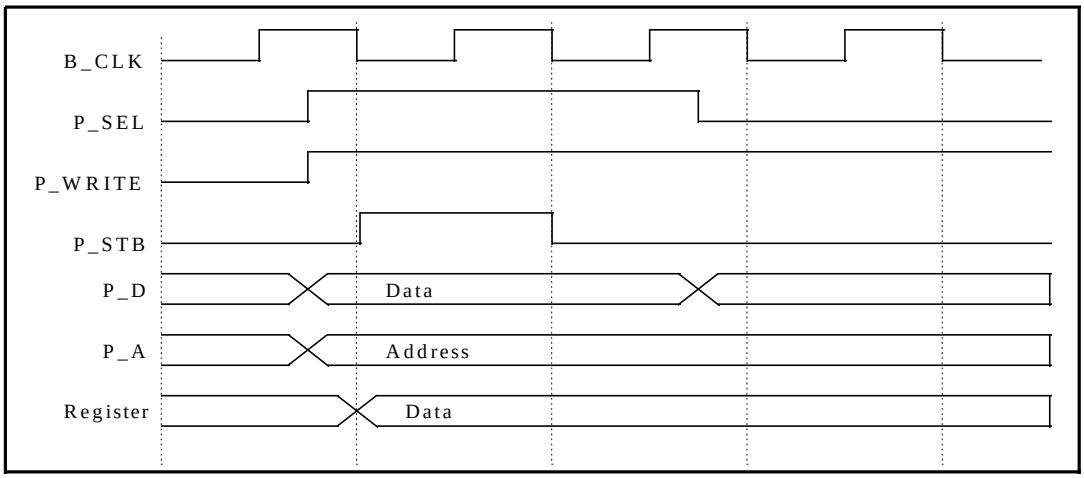


Figure 2. APB Write

4. Functional Description

All block registers are cleared during power on reset (BnRES LOW).

This disables the output drivers for port A, C, E, G and I (input as default) and enables the drivers for port B, D, F, H, and J (output as default).

For each port there is a Data Register and a Data Direction Register. 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.

The Data Direction Registers operates in a different manner on each port:

• For every port, a “0” in the data direction register indicates the port is defined as an output (default), a “1” in the data direction register indicates the port is defined as an input.

5. Programmer' s Model

5.1 PIO Registers

The following user registers are provided:

PnDR Port n Data Register. Values written to this 8-bit read/write register will be output on port A pins if the corresponding data direction bits are set to HIGH (port output). Values read from this register reflect the external states of port n, not necessarily the value should be written to it. All bits are cleared by a system reset.

PnDDR Port n Data Direction Register. Bits set in this 8-bit read/write register will select the corresponding pins in port n to become an output, clearing a bit sets the pin to input. All bits are cleared by a system reset.

n : A, B, C, D, E, F, G, H, I and J

5.2 Register Memory Map

The base address of the PIO is 0xFFFF FC00 and the offset of any particular register from the base address is determined.

Table 3. PIO Register Memory Map

ADDRESS	READ LOCATION	WRITE LOCATION
PIO Base + 0x00	PADR register	PADR register
PIO Base + 0x04	PADDR register	PADDR register
PIO Base + 0x08	PBDR register	PBDR register
PIO Base + 0x0c	PBDDR register	PBDDR register
PIO Base + 0x10	PCDR register	PCDR register
PIO Base + 0x14	PCDDR register	PCDDR register
PIO Base + 0x18	PDDR register	PDDR register
PIO Base + 0x1c	PDDDR register	PDDDR register
PIO Base + 0x20	PEDR register	PEDR register
PIO Base + 0x24	PEDDR register	PEDDR register
PIO Base + 0x28	PFDR register	PFDR register
PIO Base + 0x2c	PFDDR register	PFDDR register
PIO Base + 0x30	PGDR register	PGDR register
PIO Base + 0x34	PGDDR register	PGDDR register
PIO Base + 0x38	PHDR register	PHDR register
PIO Base + 0x3c	PHDDR register	PHDDR register
PIO Base + 0x40	PIDR register	PIDR register
PIO Base + 0x44	PIDDR register	PIDDR register
PIO Base + 0x48	PJDR register	PJDR register
PIO Base + 0x4c	PJDDR register	PJDDR register

Section 14. Synchronous Serial Peripheral Interface

1. General Description

The synchronous serial interface (SSPI) is a high-speed synchronous serial I/O system. The SSPI can be used for simple I/O expansion or for allowing several MCUs to be interconnected in a multi-master configuration. Clock polarity, clock phase, chip select polarity, and MSB /LSB first ordering are software programmable to allow direct compatibility with a large number of peripheral devices. The SSPI system can be configured as either a master or a slave.

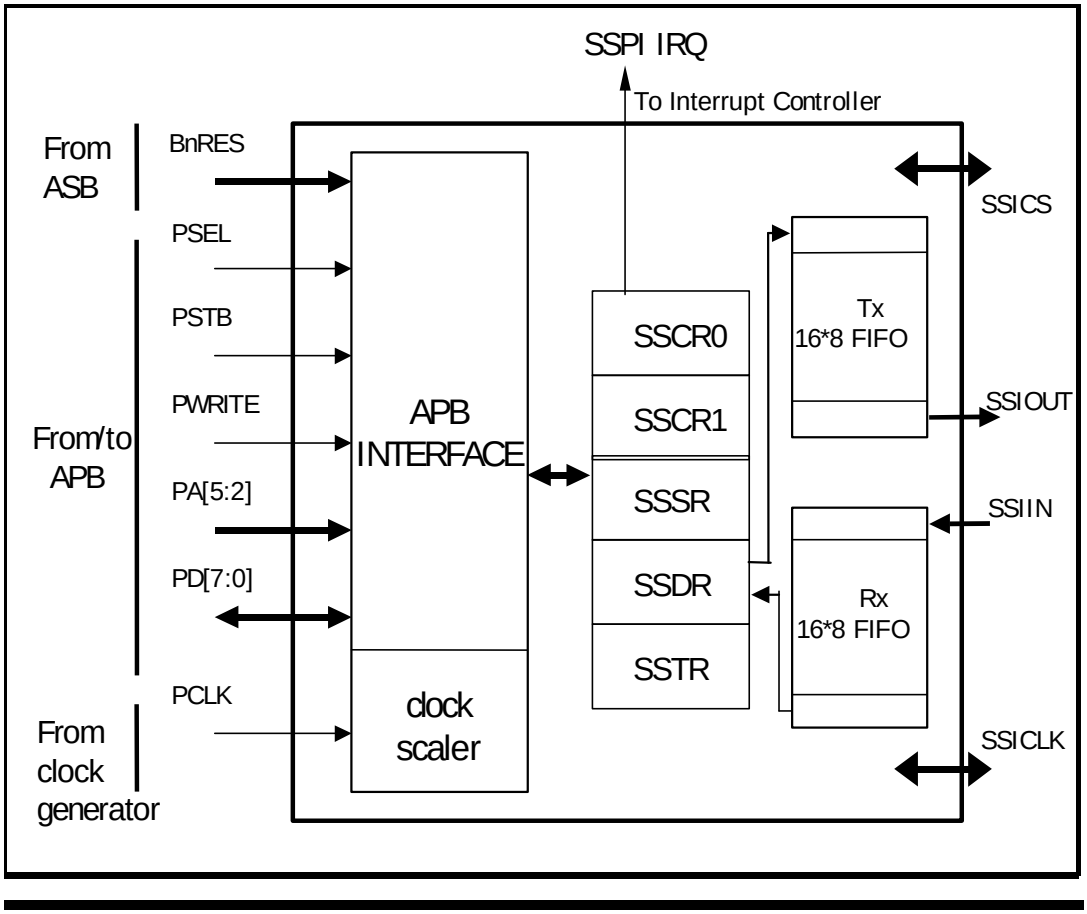


Figure 1. Signal Connections of the SSPI

An 8-bit shift register feeds the output channel, SSIOUT. During transfers, the BUSY bit in the system status register SSSR is set. Valid data can be read from a 16-bit shift register when the BUSY bit is cleared. There is also an interrupt signal, **SSIIRQ**, which is asserted at the end of data transfer. Reading data clears the interrupt signal.

2. Signal Description

The SSPI module is connected to the APB bus. Table 1. Signal descriptions describe the APB signals used and produced. Table 2. Signal descriptions show the non-AMBA signals from the block.

Table 1. Signal Descriptions

NAME	TYPE	SOURCE/ DESTINATION	DESCRIPTION
BnRES	I	PMU	ASB reset signal (active LOW).
PA[5:2]	I	APB Bridge	This is the part of the peripheral address bus, and is used by the peripheral for decoding its own register accesses. The addresses become valid before PSTB goes to HIGH and remain valid after PSTB goes to LOW.
PD[7:0]	I/O	APB Peripherals, B_D	This is the part of the bi-directional peripheral data bus. This block drives the data bus during read cycles (when PWRITE is LOW).
PSTB	I	APB Bridge	This strobe signal is used to time all accesses on the peripheral bus. The falling edge of PSTB is coincident with the falling edge of B_CLK (ASB system clock).
PWRITE	I	APB Bridge	When this signal is HIGH, it indicates a write to a peripheral, when this signal is LOW, it indicates a read from a peripheral. This signal has the same timing as the peripheral address bus. It becomes valid before PSTB goes to HIGH and remains valid after PSTB goes to LOW.
PSEL	I	APB Bridge	When this signal is HIGH, this signal indicates that the APB bridge has selected the SSPI module. This selection is a decode result of the system address bus. For more details, see AMBA Peripheral Bus Controller.

Table 2. Specific Block Signal Descriptions

NAME	TYPE	SOURCE/ DESTINATION	DESCRIPTION
CLK	I	PMU	SSPI clock input at a frequency of 3.68MHz., scaled /4, /8, /32, /64
SSIIN	I	SSIIN pad	Serial data input.
SSIOUT	O	SSIOUT pad	Serial data output.
SSICS	I/O	SSICS pad	Chip select signal.
SCLK	I/O	SCLK pad	Serial data clock to the external SSI.
SSIIRQ	O	Interrupt Controller	Active HIGH Interrupt Request.

3. Hardware Interface

The APB interface is fully APB-compliant. The APB is a non-pipelined low-power interface, designed to provide a simple interface to slave peripherals.

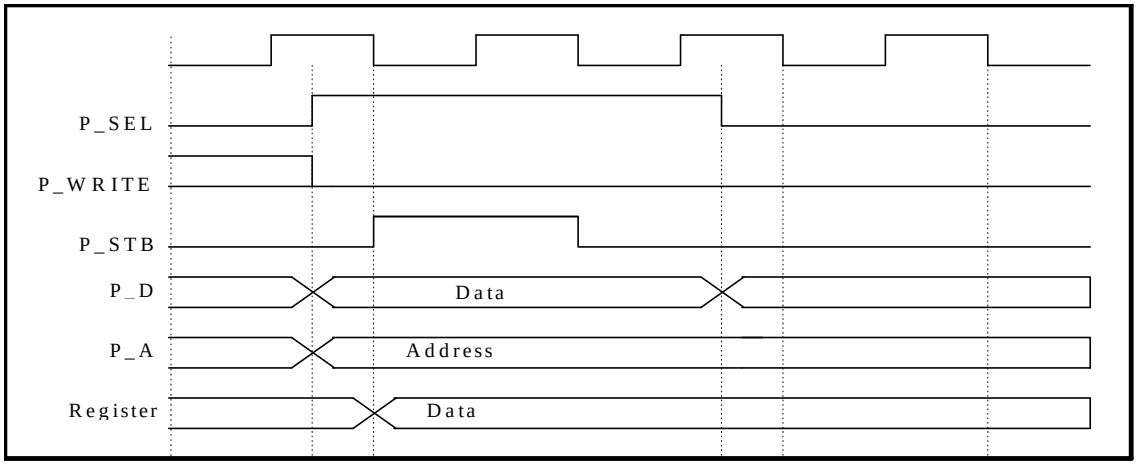


Figure 2. APB Read

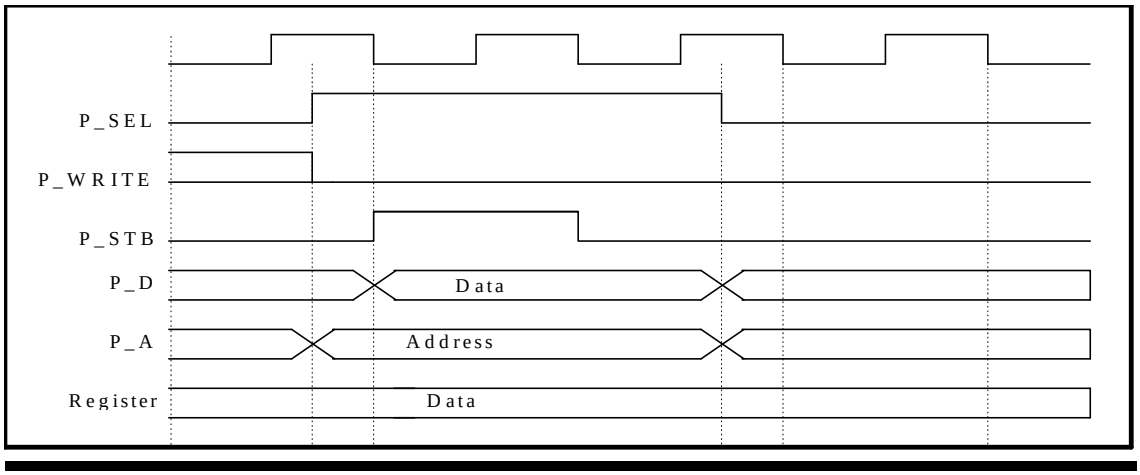


Figure 3. APB Write

4. Functional Description

The following user registers are provided:

SSCR SS Control Register. In the synchronous mode, data transfer is synchronized with a clock pulse. This mode is suitable for continuous, high-speed serial communication.

- Data length: 8 bits per character.
- Uses the built-in baud rate generator as the transmit clock.
- Support both LSB first and MSB first (receive and send).

Writing to the SSCR register controls the SSPI. A write is encoded as follows.

BIT	NAME	FUNCTION
7	Tstmode	Default 1 : normal mode, 0: test mode
6	CS_POL	Default 1 : CS Polarity(low), 0: high
5	SSI_EN	Default 1 : SSI enable signal
4	MS_MODE	Default 1 : Master Mode , 0: slave mode
3	SIN_MSB	Default 1 : MSB first sin (1: MSB First, 0: LSB First)
2	SOUT_MSB	Default 1 : MSB first sout (1: MSB First,0:LSB First)
1	CK_POL	Default 1 : Clock Polarity
0	CS_EN	Default 1 : CS enable (use only when slave mode)

SSCR0 Registers (write)

BIT	NAME	FUNCTION
7	TX end interrupt enable	Default 0 : disable 1: enable
6	TX fifo empty interrupt enable	Default 0 : disable 1: enable
5	RX fifo full interrupt enable	Default 0 : disable 1: enable
4	TX fifo full interrupt enable	Default 0 : disable 1: enable
3	Rx fifo enable	Default 0 : disable 1: enable
2	Tx fifo enable	Default 0 : disable 1: enable
1	CKSEL1	Default 0 : Clock Rate Selects
0	CKSEL0	Default 0 : Clock Rate Selects

SSCR1 Registers (write)

SSSR SS Status Register. This is automatically set when data transfer is complete between processor and external device. The flag is cleared by a read of SSSR followed by a read or write of SSDR.

SSTR SS Term Register. This is a register, which has a term between this byte and next byte by user's setting. The value can be 0 through 255. This is used only when master mode.

Table 4. SSSR Registers (Read)

BIT	NAME	FUNCTION
7	RX fifo empty	Active high
6	TX fifo empty	Active high
5	RX fifo full	Active high
4	TX fifo full	Active high
3	TX end	Active high
2	R	Reserved
1	R	Reserved
0	BUSY	when SSI transmitting and receiving

5. Register Memory Map

The base address of the SSPI interface is 0xFFFF F800 and the offset of any particular register from the base address is as followed.

ADDRESS	READ LOCATION	WRITE LOCATION
SSI Base	SSCR0	SSCR0
SSI Base + 0x04	SSCR1	SSCR1
SSI Base + 0x08	SSDR	SSDR
SSI Base + 0x0C	SSSR	
SSI Base + 0x10		SSTR
SSI Base + 0x20	SSCR0	SSCR0
SSI Base + 0x24	SSCR1	SSCR1
SSI Base + 0x28	SSDR	SSDR
SSI Base + 0x2C	SSSR	
SSI Base + 0x30		SSTR

SSI Register Memory Map

SSCR0 : Control Register0
SSCR1 : Control Register1
SSDR : Data Register
SSSR : Status Register
SSTR : Term Register

The output frequency is selected by programming the lower two bits of the SSCR1 register, SSCR1[1:0]. The following table shows the possible settings:

SSCR1[1..0]	DIV	FREQUENCY (WHEN PCLK=3.6864MHZ)
00	4	921.6 KHz
01	8	460.8 KHz
10	32	115.2 KHz
11	64	57.6 KHz

SSCR1[1:0] Encoding

6. SSPI Data Clock Timing Diagram

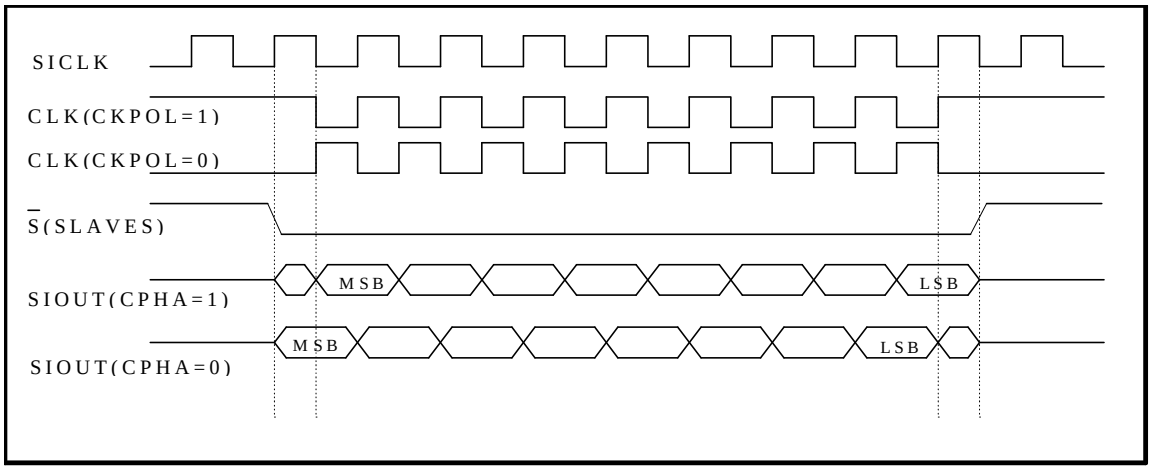


Figure 4. Timing Diagram

Section 15. UART

1. General Description

This module is an Universal Asynchronous Receiver/Transmitter(UART) with FIFOs, and is functionally identical to the 16450 on power-up (CHARACTER mode). The GM16550 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 maximize system 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 performed by the UART, as well as any error conditions(parity, overrun, framing, or break interrupt).

The UART includes a programmable baud rate generator that is capable of dividing the timing reference clock input by divisors of 1 to 65535 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.

2. Features

- Capable of running all existing 16450 software.
- After reset, all registers are identical to the 16450 register set.
- The FIFO mode transmitter and receiver are each buffered with 16 byte FIFO's to reduce the number of interrupts presented to the CPU.
- Adds or deletes standard asynchronous communication bits (start, stop, and parity) to or from the serial data.
- Hold and shift registers in the 16450 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 256k 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.

3. Signal Description

The GDC21D601 UART module is connected to the APB bus.

Table 1. Signal Descriptions

NAME	Type	SOURCE/ DESTINATION	DESCRIPTION
U_CLK	I	CPG	UART external Clock input This connects the main timing reference to the UART. 3.6864Mhz is recommendable input clock frequency.
nB_RES0	I	PMU	Reset signal generated from the APB Bridge(Master Reset) When this input is low, it clears all the registers (except the Receiver Buffer, Transmitter Holding, and Divisor Latches) and the control logic of the UART. The states of various output signals (SOUT, INT_UART, nRTS, nDTR) are affected by an active nB_RES[0] input.
P_A[2:0]	I	APB Bridge	Register select. Address signals connected to these 3 inputs select a UART register for the CPU to read from or write to during data transfer. A table of registers and their addresses is shown below. Note that the state of the Divisor Latches
P_D[7:0]	I/O	APB Bridge	Data Bus. This bus comprises eight TRI-STATE input/output lines. The bus provides bi-directional communications between the UART and the CPU. Data, control words and status information are transferred via the P_D[7:0] data bus.
P_STB	I	APB Bridge	This strobe signal is used to time all accesses on the peripheral bus. The falling edge of P_STB is coincident with the falling edge of B_CLK.(ASB System Clock)
P_WRITE	I	APB Bridge	When this signal is HIGH, it indicates a write to a peripheral. When this signal is LOW, it indicates a read from a peripheral. This signal has the same timing as the peripheral address bus. It becomes valid before P_STB goes to HIGH and remains valid after P_STB goes to LOW.
P_SEL	I	APB Bridge	When this signal is HIGH, it indicates that this module has been selected by the APB bridge. This selection is a decode of the system address bus (ASB).
INT_UART	O	INTC	Interrupt. This pin goes to high whenever any one of the following interrupt types has an active high condition and is enabled via EIR: Receiver Error Flag; Received Data Available:timeout(FIFO Mode only); Transmitter Holding Register Empty; and MODEM Status. The INT_UART signal is reset to low upon the appropriate interrupt service or a Master Reset operation.
NCTS	I	External	Clear to Send. When this signal is low, it indicates that the MODEM or data set is ready to exchange data. The NCTS 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 NCTS input has changed its state since the previous reading of the MODEM Status Register. NCTS 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.

NAME	Type	SOURCE/ DESTINATION	DESCRIPTION
SIN	I	External	Serial Input. Serial data input from the communications link (peripheral device, MODEM or data set).
NDSR	I	External	Data Set Ready. When this signal is low, it indicates that the MODEM or data set is ready to establish the communications link with the UART. The NDSR 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 nDSR signal. Bit 1(DDSR) of MODEM status Register indicates whether the nDSR input has changed its 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.
NDCD	I	External	Data Carrier Detect. When this signal is low, it 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 its state since the previous reading of the MODEM Status Register. NDCD 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.
NRI	I	External	Ring Indicator. When this signal is low, it indicates that a telephone ring signal is received by the MODEM or data set. The NRI 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 NRI signal. Bit 2 (TERI) of the MODEM Status Register indicates whether the NRI 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.
NDTR	O	External	Data Terminal Ready. When this is low, it informs the MODEM or data set that the UART is ready to establish communication link. The NDTR 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.
NRTS	O	External	Request To Send. When low, this informs the MODEM or data set that the UART is ready to exchange data. The NRTS 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.
SOUT	O	External	Serial Output. Composite serial data output to the communications link (peripheral, MODEM or data set). The SOUT signal is set to the Marking (logic 1) state upon a Master Reset operation.

Table 2. Register Address

DLAB	P_A[2]	P_A[1]	P_A[0]	REGISTER
0	0	0	0	Receiver Buffer(read), Transmitter Holding Register(write)
0	0	0	1	Interrupt Enable
x	0	1	0	Interrupt Identification(read)
x	0	1	0	FIFO Control(write)
x	0	1	1	Line Control
x	1	0	0	Modem Control
x	1	0	1	Line Status
x	1	1	0	Modem Status
x	1	1	1	Scratch
1	0	0	0	Divisor Latch(least significant byte)
1	0	0	1	Divisor Latch(most significant byte)

Table 3. UART Reset Configuration

REGISTER / SIGNAL	REGISTER CONTROL	REGISTER STATE
Interrupt Enable Register	Master Reset	0000 0000
Interrupt Identification Register	Master Reset	0000 0001
FIFO Control Register	Master Reset	0000 0000
Line Control Register	Master Reset	0000 0000
MODEM Control Register	Master Reset	0000 0000
Line Status Register	Master Reset	0110 0000
MODEM Status Register	Master Reset	xxxx 0000
SOUT	Master Reset	High
INT_UART (RCVR Errs)	Read LSR / RESET	Low
INT_UART (RCVR Data Ready)	Read RBR / RESET	Low
INT_UART (THRE)	Read IIR / Write THR / RESET	Low
INT_UART(Modem Status changes)	Read MSR / RESET	Low
NRTS	Master Reset	High
NDTR	Master Reset	High

4. Internal Block Diagram

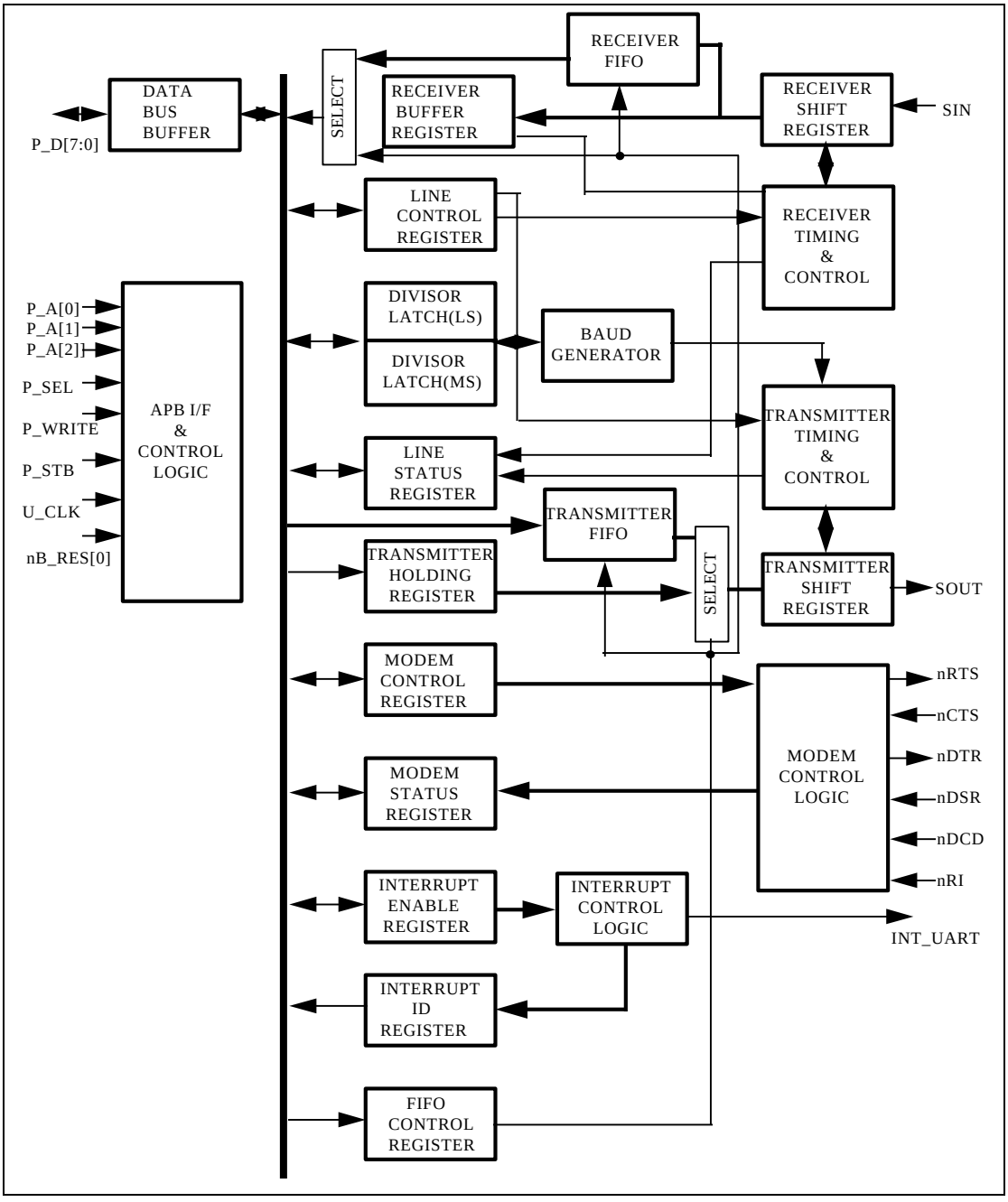


Figure 1. Internal Block Diagram

5. Registers Description

There are two URATs implemented in the design, the base addresses are 0xFFFF F500 in UART0 and 0xFFFF F600 in UART1. In Table 4. UART register address map, x can be either 0 or 1.

Table 4. UART Register Address Map

ADDRESS	NAME	DESCRIPTION
UARTxBase + 0x00	Receiver Buffer (RBR)	8-bit R/O set DLAB = 0
UARTxBase + 0x00	Transmitter Holding (THR)	8-bit W/O set DLAB = 0
UARTxBase + 0x04	Interrupt Enable (IER)	8-bit R/W
UARTxBase + 0x08	Interrupt Identification (IIR)	8-bit R/O
UARTxBase + 0x08	FIFO Control (FCR)	8-bit W/O
UARTxBase + 0x0C	Line Control (LCR)	8-bit R/W
UARTxBase + 0x10	MODEM Control (MCR)	8-bit R/W
UARTxBase + 0x14	Line Status (LSR)	8-bit R/W
UARTxBase + 0x18	MODEM Status (MSR)	8-bit R/W
UARTxBase + 0x1C	Scratch (SCR)	8-bit R/W
UARTxBase + 0x00	Divisor Latch LS (DLL)	8-bit R/W set DLAB = 1
UARTxBase + 0x04	Divisor Latch MS (DLM)	8-bit R/W set DLAB = 1

Table 5. Summary of registers gives the details of the UART registers.

Table 5. Summary of Registers

REGISTER ADDRESS												
	0 DLAB = 0	0 DLAB = 0	1 DLAB = 0	2	2	3	4	5	6	7	0 DLAB = 1	1 DLAB = 1
Bit No.	Receiver Buffer Register	Transmitter Holding Register	Interrupt Enable Register	Interrupt Ident Register	FIFO Control Register	Line Control Register	MODEM Control Register	Line Status Register	MODEM Status Register	Scratch Register	Divisor Latch (LS)	Divisor Latch (MS)
	RBR	THR	IER	IIR	FCR	LCR	MCR	LSR	MSR	SCR	DLL	DLM
0	Data Bit 0 (Note 1)	Data Bit 0	Enable received data available interrupt	0 if interrupt pending	FIFO enable	Word length select Bit 0	Data Terminal Ready (DTR)	Data Ready (DR)	Data Clear to Send (DCTS)	Bit 0	Bit 0	Bit 8
1	Data Bit 1	Data Bit 1	Enable transmitter holding register empty interrupt	Interrupt ID Bit 0	RCVR FIFO reset	Word length select Bit 1	Request to Send (RTS)	Overrun Error (OE)	Delta Data Set Ready (DDSR)	Bit 1	Bit 1	Bit 9
2	Data Bit 2	Data Bit 2	Enable receiver line status interrupt	Interrupt ID Bit 1	XMIT FIFO reset	Number of stop bit		Parity Error (PE)	Trailing Edge Ring Indicator (TERI)	Bit 2	Bit 2	Bit 10
3	Data Bit 3	Data Bit 3	Enable modem status interrupt	Interrupt ID Bit 2 (Note 2)		Parity enable		Framing Error (FE)	Delta Data Carrier Detect (DDCD)	Bit 3	Bit 3	Bit 11
4	Data Bit 4	Data Bit 4	0	0	Reserved	Even parity select	Loop	Break Interrupt (BI)	Clear to Send (CTS)	Bit 4	Bit 4	Bit 12
5	Data Bit 5	Data Bit 5	0	0	Reserved	Stick parity	0	Transmitter Holding Register Empty (THRE)	Data Set Ready (DSR)	Bit 5	Bit 5	Bit 13
6	Data Bit 6	Data Bit 6	0	FIFO enabled (Note 2)	RCVR trigger (LSB)	Set break	0	Transmitter Empty (TEMT)	Ring Indicator (RI)	Bit 6	Bit 6	Bit 14
7	Data Bit 7	Data Bit 7	0	FIFO enabled (Note 2)	RCVR trigger (MSB)	Divisor Latch Access Bit	0	Error in RCVR FIFO (Note 2)	Data Carrier Detect (DCD)	Bit 7	Bit 7	Bit 15

Note 1 : Bit 0 is the least significant bit seriously transmitted or received.
Note 2 : These bits are always 0 in the GM16C450 mode.

The system programmer may access any of the UART registers summarized in Table 5. Summary of Registers via the CPU. These registers control UART operation including transmission and reception of data. Each register bit in the table has its name and reset state as shown.

Line Control Register

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 the system programming and eliminates the need for separate storage in system memory of the line characteristics. Table 5. Summary of Registers shows the contents of the LCR. Details on each bit are :

Bit 0 and 1 : 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:

Table 6. Line Control Register Encoding

BIT 1	BIT 0	CHARACTER LENGTH
0	0	5 Bits
0	1	6 Bits
1	0	7 Bits
1	1	8 Bits

Bit 2 : This bit specifies the number of Stop bits transmitted and received in each serial character. If bit 2 is a logic 0, one Stop bit is generated in the transmitted data. If bit 2 is a 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 selected Stop bits.

Bit 3 : This bit is the Parity Enable bit. When bit 3 is a 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.)

Bit 4 : This bit is the Even Parity Select bit. When bit 3 is a logic 1 and bit 4 is a logic 0, an odd number of logic 1s is transmitted or checked in the data word bits and Parity bit. When bit 3 is a logic 1 and bit 4 is a logic 1, an even number of logic 1s is transmitted or checked.

Bit 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 a logic 0. If bits 3 and 5 are 1 and bit 4 is a logic 0, then the Parity bit is transmitted and checked as a logic 1. If bit 5 is a logic 0 Stick Parity is disabled.

Bit 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 a logic 1, the serial output (SOUT) is forced to be the Spacing (logic 0) state. The break is disabled by setting bit 6 to a 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.

Bit 7 : This bit is the Divisor Latch Access Bit (DLAB), It must be set to high (logic 1) to access the Divisor Latches of the Baud Generator during a Read or Write operation. It must be set to low (logic 0) to access the Receiver Buffer, the Transmitter Holding Register, or the Interrupt Enable Register.

Programmable Baud Generator

The UART contains a programmable Baud Generator that is capable of taking any clock input from DC to 8.0 MHz and dividing it by any divisor from 2 to 65535. 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 the proper operation of the Baud Generator. Upon loading either of the Divisor Latches, a 16-bit Baud counter is immediately loaded.

Table 7. Baud rates provide decimal divisors to use with crystal frequencies of 1.8432 MHz and 3.6864 MHz. For baud rates of 38400 and below, the error obtained is minimal. The accuracy of the desired baud rate depends on the chosen crystal frequency. Using a divisor of zero is not recommended.

Table 7. Baud Rates

1.8432 MHz			3.6864 MHz		
Desired Baud Rate	Decimal Divisor Used to Generate 16 x Clock	Percent Error Difference Between Desired and Actual	Desired Baud Rate	Decimal Divisor Used to Generate 16 x Clock	Percent Error Difference Between Desired and Actual
50	2304	-	50	4608	-
75	1536	-	-	-	-
110	1047	0.026	110	2094	0.026
134.5	857	0.058	-	-	-
150	768	-	-	-	-
300	384	-	300	768	-
600	192	-	-	-	-
1200	96	-	1200	192	-
1800	64	-	-	-	-
2000	58	0.69	-	-	-
2400	48	-	2400	96	-
3600	32	-	-	-	-
4800	24	-	4800	48	-
7200	16	-	-	-	-
9600	12	-	9600	24	-
19200	6	-	19200	12	-
38400	3	-	38400	6	-
57600	2	2.86	57600	4	-
115200	1		115200	2	-

Line Status Register

This register provides status information to the CPU concerning the data transfer. Table 5. Summary of Registers shows the contents of the Line Status Register. Details on each bit are :

Bit 0 : This bit is the receiver Data Ready (DR) indicator. Bit 0 is set to a 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 a logic 0 by reading all of the data in the Receiver Buffer Register or the FIFO.

Bit 1 : 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 a logic 1 upon the 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.

Bit 2 : 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 a logic 1 upon the detection of a parity error and is reset to a 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 where it applies to. This error is revealed to the CPU when its associated character is at the top of the FIFO.

Bit 3 : 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 a 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 where 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 resynchronize 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 it in the “data”.

Bit 4 : This bit is the Break Interrupt (BI) indicator. Bit 4 is set to a 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 where 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 through 4 are the error conditions that produce a Receiver Line Status interrupt whenever any of the corresponding conditions is detected and the interrupt is enabled.

Bit 5 : 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 to 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 by the CPU. 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.

Bit 6 : 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 a 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.

Bit 7 : In the 16450 mode, this is 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.
** Note : The Line Status Register is intended for read operations only.

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 to trigger level.

Bit 0 : Writing a 1 to FCR0 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.

Bit 1 : Writing a 1 to FCR1 resets its counter logic to 0. The shift register is not cleared. The 1 that is written to this bit position is self-cleared.

Bit 2 : Writing a 1 to FCR2 resets its counter logic to 0. The shift register is not cleared. The 1 that is written to this bit position is self-cleared.

Bit 3 : FCR3 is not used.

Bit 4,5 : FCR4 to FCR5 are reserved for future use.

Bit 6,7 : FCR6 and FCR7 are used to set the trigger level for the RCVR FIFO interrupt.

Table 8. RCVR FIFO Interrupt

FCR[7:6]	RCVR FIFO Trigger Level (Bytes)
00	01 (default)
01	04
10	08
11	14

Interrupt Identification Register

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

- Receiver Line Status
- Received Data Ready
- Transmitter Holding Register Empty
- 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 occurs, the UART records new interrupts, but does not change its current indication until the access is complete. Table 5. Summary of Registers shows the contents of the IIR. Details on each bit are :

Bit 0 : This bit can be used in a prioritized interrupt environment to indicate whether an interrupt is pending or not. When bit 0 is a logic 0, an interrupt is pending and the IIR contents may be used as a pointer for the appropriate interrupt service routine. When bit 0 is a logic 1, no interrupt is pending.

Bit 1 and 2 : These two bits of the IIR are used to identify the highest priority interrupt pending as indicated in Table 9. Interrupt control functions.

Bit 3 : In the 16450 mode this bit is 0. In the FIFO mode this bit is set along with bit 2 when a time-out interrupt is pending.

Bit 4 and 5 : These two bits of the IIR are always logic 0.

Bit 6 and 7 : These two bits are set when FCR0 = 1.

Table 9. Interrupt Control Functions

FIFO MODE ONLY	INTERRUPT IDENTIFICATION REGISTER			INTERRUPT SET AND RESET FUNCTIONS			
Bit 3	Bit 2	Bit 1	Bit 0	Priority Level	Interrupt Type	Interrupt Source	Interrupt Reset Control
0	0	0	1	-	None	None	-
0	1	1	0	Highest	Receiver Line Status	Overrun Error, Parity Error, Framing Error or Break Interrupt	Reading the Line Status Register
0	1	0	0	Second	Receiver Data Available	Receiver Data Available or Trigger Level Reached	Reading the Receiver Buffer Register or the FIFO drops below the trigger level
1	1	0	0	Second	Character Time-out Indication	No Characters have been removed from or input to the RCVR FIFO during the last 4 Char. times and there is at least 1 Char. in it during this time	Reading the Receiver Buffer Register
0	0	1	0	Third	Transmitter Holding Register Empty	Transmitter Holding Register Empty	Reading the IIR Register (if it is the source of interrupt) or writing it into the Transmitter Holding Register
0	0	0	0	Fourth	MODEM Status	Clear to Send, Data Set Ready, Ring Indicator, or Data Carrier Detect	Reading the MODEM Status Register

Interrupt Enable Register

This register enables the five types of UART interrupts. Each interrupt can individually activate the interrupt (INT_UART) output signal. It is possible to totally disable the interrupt Enable Register (IER). Similarly, setting bits of the IER register to a logic 1 enable the selected interrupt(s). Disabling an interrupt prevents it from being indicated as active in the IIR and from activating the INT_UART output signal. All other system functions operate in their normal manners, including the setting of the Line Status and MODEM Status Registers. Table 5. Summary of Registers shows the contents of the IER. Details on each bit are :

Bit 0 : This bit enables the Received Data Available Interrupt (and time-out interrupts in the FIFO mode) when it is set to logic 1.

Bit 1 : This bit enables the Transmitter Holding Register Empty Interrupt when set to logic 1.

Bit 2 : This bit enables the Receiver Line Status Interrupt when it is set to logic 1.

Bit 3 : This bit enables the MODEM Status Interrupt when it is set to logic 1.

Bit 4 through 7 :These four bits are always logic 0.

MODEM Control Register

This register controls the interface with the MODEM or data set (or a peripheral device emulating a MODEM). The contents of the MODEM Control Register are indicated in Table 5. Summary of Registers, and are described below.

Bit 0 : This bit controls the Data Terminal Ready (NDTR) output. When this bit is set to a logic 1, the NDTR output is forced to be a logic 0. When bit 0 is reset to a logic 0, the NDTR output is forced to be a 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.

Bit 1 : This bit controls the Request to Send (NRTS) output. Bit 1 affects the NRTS output in an identical manner that described above for bit 0.

Bit 2 : Not used

Bit 3 : Not used

Bit 4 : This bit provides a local loopback feature for diagnostic testing of the UART. When bit 4 is set to logic 1, 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. The two MODEM Control outputs (NDTR and NRTS) and two internal nodes (OUT1 and OUT2) are internally connected to the four MODEM Control inputs and the MODEM Control output pins are forced to be their inactive state (high). On the diagnostic mode, the transmitted data 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.

Bit 5 through 7 :These bits are permanently set to logic 0.

MODEM Status Register

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 a logic 1 whenever a control input from the MODEM changes its state. They are reset to logic 0 whenever the CPU reads the MODEM Status Register.

The contents of the MODEM Status Register are indicated in Table 5. Summary of Registers, and are described below.

Bit 0 : This bit is the Delta Clear to Send (DCTS) indicator. Bit 0 indicates that the NCTS input to the chip has changed its state since the last time it was read by the CPU.

Bit 1 : This bit is the Delta Data Set Ready (DDSR) indicator. Bit 1 indicates that the NDSR input to the chip has changed its state since the last time it was read by the CPU.

Bit 2 : This bit is the Trailing Edge of Ring Indicator (TERI) detector. Bit 2 indicates that the NRI input to the chip has changed from a low to a high state.

Bit 3 : This bit is the Delta Data Carrier Detect (DDCD) indicator. Bit 3 indicates that the NDCD input to the chip has changed its 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.

Bit 4 : This bit is the complement of the Clear to Send (NCTS) input. If bit 4 (loop) of the MCR is set to a 1, this bit is equivalent to RTS in the MCR.

Bit 5 : This bit is the complement of the Data Set Ready (NDSR) input. If bit 4 of the MCR is set to 1, this bit is equivalent to DTR in the MCR.

Bit 6 : This bit is the complement of the Ring Indicator (NRI) input. If bit 4 of the MCR is set to 1, this bit is equivalent to OUT1 in the MCR.

Bit 7 : This bit is the complement of the Data Carrier Detect (NDCD) input. If bit 4 of the MCR is set to 1, this bit is equivalent to OUT2 in the MCR.

Scratch Register

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

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 timeout interrupts occurs as follows :

1. A FIFO timeout interrupt occurs in the following conditions :
   - at least one character is in the FIFO
   - the latest serial character received was longer than 4 continuous character times (if 2 stop bits are programmed, the second one is included in this time delay).
   - the latest CPU read of the FIFO was longer than 4 continuous character times.

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 for a clock signal (this makes the delay proportional to the baud rate).

3. When a timeout interrupt has occurred, it is cleared and the timer is reset when the CPU reads one character from the RCVR FIFO.

4. When a timeout interrupt has not occurred the timeout 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 occur as follows :

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 this interrupt is serviced or the IIR is read.

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 FCR 0 will be immediate if it is enabled.

Character timeout 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.

FIFO Polled Mode Operation

When FCR 0 = 1 resetting, IER 0, IER 1, IER 2, IER3 or all to zero puts the UART in the FIFO Polled Mode. Since the RCVR and XMITTER are controlled separately, either one or both can be in the polled mode of operation.

Section 16. Smart Card Interface

1. General Description

The smart card interface block is basically a general-purpose serial interface block that has the smart card interface features additionally. And some ports of GPIO are necessary to provide complete interfaces to the smart card. As a general-purpose serial interface, it has the UART(16550) compatible register sets and bit definitions although it doesn't have the modem control pins.
As a smart card interface, it has the following general features.

- Supports only asynchronous operation.
- Supports cards that have internal reset capability.
- Supports cards that have an active low reset input.
- Supports cards that use the internal clock.
- Generate the clock for a card expecting the external clock.
- Use the serial in/out ports for I/O.
- Use the GPIO ports for other interface signals like RST, DETECT, etc.

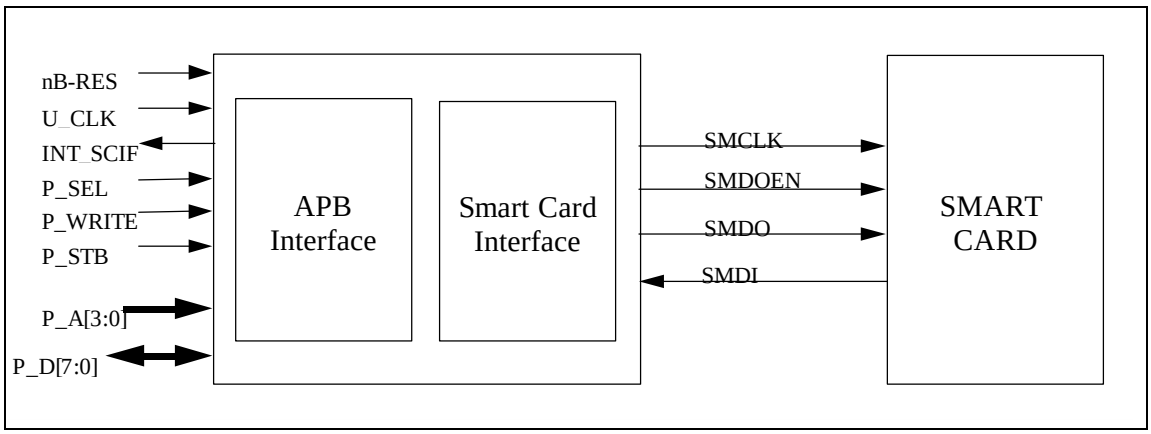


Figure 1. Signal Connections of the SMART CARD Interface

2. Signal Description

The SMART CARD Interface module is connected to the APB. Table 1. Signal descriptions describes the APB signals used and produced. Table 2. Signal descriptions shows the non-AMBA signals from the block.

Table 1. Signal Descriptions

Name	Type	Source/ Destination	Description
nB_RES	I	Reset Controller	This signal indicates system reset status of the bus (active LOW)
P_A[3:0]	I	APB Bridge	This is part of the peripheral address bus, which is used by the peripheral for decoding its own register accesses. The addresses become valid before P_STB goes HIGH and remain valid after P_STB goes LOW.
P_D[7:0]	I/O	APB, B_D	This is part of the bidirectional peripheral data bus. The data bus is driven by this block during read cycles (when P_WRITE is LOW).
P_STB	I	APB Bridge	This strobe signal is used to time all accesses on the peripheral bus. The falling edge of P_STB is coincident with the falling edge of B_CLK (ASB system clock).
P_WRITE	I	APB Bridge	When HIGH, this signal indicates a write to a peripheral, and when LOW, a read from a peripheral. This signal has the same timing as the peripheral address bus. It becomes valid before P_STB goes HIGH and remains valid after P_STB goes LOW.
P_SEL	I	APB Bridge	When HIGH, this signal indicates the SMART CARD interface module has been selected by the APB bridge. This selection is a decode of the system address bus (ASB). For more details see AMBA Peripheral Bus Controller (ARM DDI 0044).

Table 2. Specific Block Signal Descriptions

Name	Type	Source/ Destination	Description
SMDI	I	SMART CARD	Serial data input
SMDO	O	SMART CARD	Serial data output
SMCLK	O	SMART CARD	Clock output for smart card that expects the external clock
SMDOEN	O	SMART CARD	Tri-State buffer enable signal for SMDO output. For normal uart operation, it is always ‘1’. For smart card interface mode, it’s used to prevent the serial I/O bus conflict because the serial I/O for smart card interface is bidirectional
INT_SCIF	Out	Interrupt Controller	SMART CARD Interrupt.

3. Hardware Interface

The APB interface is fully APB-compliant. The APB is a nonpipelined low-power interface, designed to provide a simple interfacing to slave peripherals.

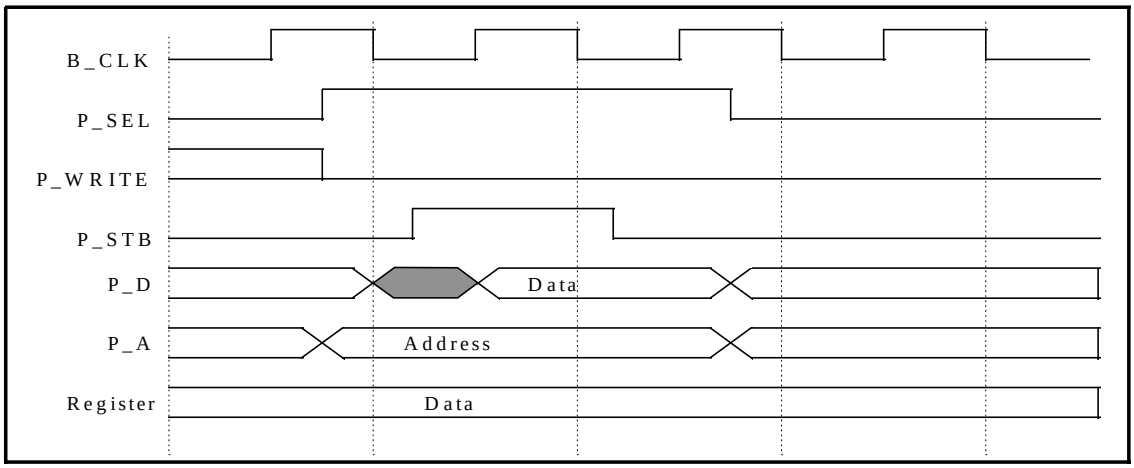


Figure 2. APB Read

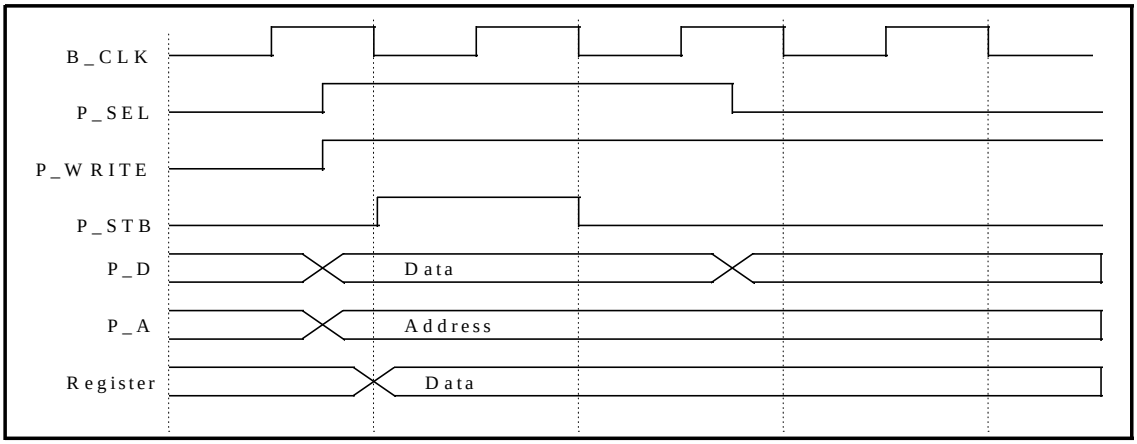


Figure 3. APB Write

4. Functional Description

4.1 Reset and Detection of the Card

All the interface signals except the I/O and the CLK are connected to the GPIO ports. Usually, RST and the card detect signal are connected to the GPIO. (The number of the interface signals needed and their functionality can vary according to the physical application.)
The ISO/IEC 7816 standard defined the timing requirements for RST and that can be controlled by software. The implementation of the detection of the card is not defined in the standard, but usual implementation uses the external circuit to generate the card-detect signal that can indicates the presence of the card. The smart card interface of the GDC27D601 also expects the “card detect signal” from the outside through the GPIO port.
Optionally, GPIO can also provide the voltage control signals that are needed to support the various kinds of smart cards that use the different Vcc and Vpp voltages.

4.2 Transmitting the Data

Transmitting the data to the smart card is performed with the following procedure.

- Program the baud rate generator to the appropriate value.
- Software writes the data to the transmit buffer.
- After transmission, card sends an error signal if there is a parity error.
- Smart card interface detects the parity error and sets the error status for CPU to read the status.
- The CPU read s the error status and determine if re-transmission is necessary.
- If there is an error, write the same data again to the transmit buffer and re-transmit.
- Guard time is controlled by software.

4.3 Receiving the Data

Receiving the data from the smart card is done with the following procedure.

- Program the baud rate generator to the appropriate value.
- Receive the data.
- If there is an parity error, smart card interface sets the internal error status and send the error signal back to the card through the I/O pin.
- Software checks the error status and wait for data if there is an error.
- Guard time is controlled by software.

5. Programmer' s Model

Smart Card Interface Mode Control Register (SMCR)

SMCR is a 3-bit readable/writable register, in which is used to control the pure smart card interface part except uart and APB interface. All the bits in this register are initialized to 0 at reset.

Table 3. SMCR Bit Functions

BIT	NAME	FUNCTION	
0	MDSEL	Mode Select	0 : UART (initial value) 1 : Smart card I/F
1	SMCLKEN	Smart card Clock Enable	0 : Disable(initial value) 1 : Enable
2	SMPEDEN	Parity Error Detect Enable	0 : Disable(initial value) 1 : Enable

Smart Card Clock Devisior Latch (SMDLL)

Smart Card Clock Devisior Latch (SMDLM)

The smart card interface has a programmable clock generator for smart card that uses the external clock. SMDLL and SMDLM are two 8-bit latchs, in which is used to store in a 16-bit binary format. All the bits in this registers are initialized to 0 at reset.

Table 4. SMDLL Bit Functions

BIT	NAME	FUNCTION
7:0	devisior (LS)	Divisor for SMCLK

Table 5. SMDLM Bit Functions

BIT	NAME	FUNCTION
7:0	devisior (MS)	Divisor for SMCLK

Smart Card Interface Status Register (SMSR)

SMSR is a 1-bit read-only register which is used to indicate whether partty error is detected or not. It is cleared after reading it.

Table 6. SMSR Bit Functions

BIT	NAME	FUNCTION
0	SMPEDI	Parity error 0 : No interrupt occurrenrce (initial value) 1 : Interrupt occurrerce

Test Register for Input (TIR)

TIR is a 3-bit write-only register defined for test purpose. This register allows simulation of input signals to the block, as well as the generation of a special test clock signal aimed for production test vectors.

Table 7. TIR Bit Functions

BIT	NAME	FUNCTION
2	TNMODE	Mode select bit 0 : Normal operation mode 1 : Test mode
1	TICUCLK	Programmable serial clock for test
0	TICSMDI	Programmable serial data input for test

Test Register for Output (TOR)

TOR is a 3-bit read-only register defined for test purpose. This register allows simulation of output signals from the block.

Table 8. TOR Bit Functions

BIT	NAME	FUNCTION
2	TSMDO	Serial data output line
1	TSMDOEN	Serial outputl data enable line
0	TSMCLK	Serial clock line for smart card

The following regitsers are same ones in the UART module. Details on each register see the data sheet of GDC21D601 UART.

Receiver Buffer Register (RBR)

Transmitter Holding Register (THR)

Interrupt Enable Register (IER)

Interrupt Identification Register (IIR)

FIFO Control Register (FCR)

Line Control Register (LCR)

Line Status Register (LSR)

Scratch Register (SCR)

Divisor Latch (DLL)

Divisor Latch (DLM)

5.2 Register Memory Map

The base address of the SMART CARD interface is 0xFFFFF700 and the offset of any particular register from the base address is determined.

Table 9. SMART CARD Interface Register Memory Map

ADDRESS	REGISTER	READ LOCATION	WRITE LOCATION
SMART CARD I/F Base	RBR/THR (DLAB = 0) DLL (DLAB = 1)	Receiver Buffer Divisor Latch (LS)	Transmitter Holding Divisor Latch (MS)
SMART CARD I/F Base + 0b0001	IER (DLAB = 0) DLM (DLAB = 1)	Interrupt Enable Divisor Latch (MS)	Interrupt Enable Divisor Latch (MS)
SMART CARD I/F Base + 0b0010	IIR/FCR	Interrupt Ident	FIF Control
SMART CARD I/F Base + 0b0011	LCR	Line Control	Line Control
SMART CARD I/F Base + 0b0101	LSR	Line Status	
SMART CARD I/F Base + 0b0111	SCR	Scratch	Scratch
SMART CARD I/F Base + 0b1000	SMCR	Smart card control	Smart card control
SMART CARD I/F Base + 0b1001	SMDLL	Divisor Latch (LS) for SMCLK	Divisor Latch (LS) for SMCLK
SMART CARD I/F Base + 0b1010	SMDLM	Divisor Latch (MS) for SMCLK	Divisor Latch (MS) for SMCLK
SMART CARD I/F Base + 0b1011	SMSR	Status for SMPED Error	
SMART CARD I/F Base + 0b1100	TIR		Test Register for Input
SMART CARD I/F Base + 0b1101	TOR	Test Register for Output	

Section 17. I²C Controller

1. General Description

The I²C controller allows the GDC21D601 to exchange data with a number of other I²C devices such as micro controller, EEPROMs, real-time clock devices, A/D converters, LCD displays, NTSC/PAL encoder, and etc.

The I²C is a synchronous bus that is used to connect several ICs on a board. The I²C bus uses two wires, serial data (SDA), and serial clock (SCL) to carry information between the ICs connected to the bus.
The I²C controller consists of transmitter and receiver sections, an independent baud rate generator, and a control unit. The transmitter and receiver sections use the same clock, which is derived from the I²C controller baud rate generator in master mode. Refer to Figure 1. for the I²C controller block diagram.

The GDC21D601 I²C bit7(MSB) is shifted out first.

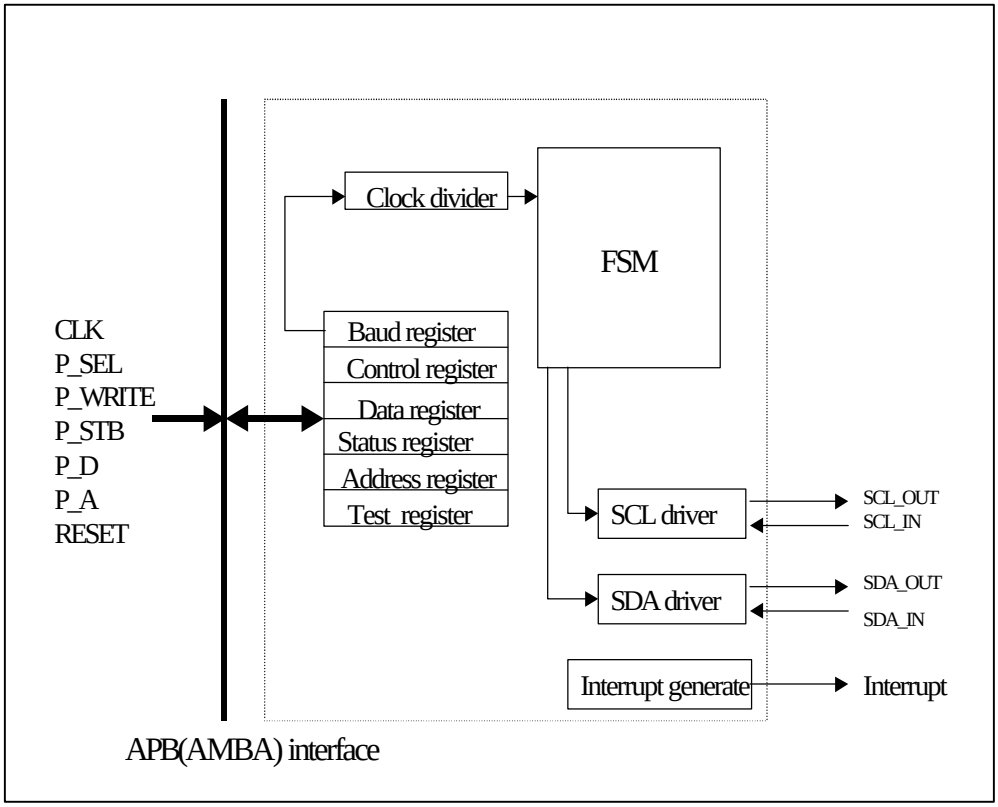


Figure 1. I²C Block Diagram

2. I²C Controller Key Features

The I²C controller contains the following key features:

- Two-Wire Interfaces (SDA and SCL)
- Both Master and Slave functions
- Supports Clock Rates up to 400khz in Master Mode .
- Independent Programmable Baud Rate Generator
- Local Loopback Capability for Testing
- Slave clock stretching support

3. I²C Controller Clocking and Pin Functions

The I²C controller can be configured as a master or slave for the serial channel.

When the I²C controller is a master, the I²C controller baud rate generator is used to generate the I²C controller transmit and receive clocks. The I²C baud rate generator takes its input from the block clock input.

Both serial data (SDA) and serial clock (SCL) are bi-directional pins. These pins are connected to a positive supply voltage via an external pull up resister. When the bus is free, both lines are high.

When the I²C controller is working as a master, SCL is the clock output signal that shifts the received data in and shifts the transmit data out from/to the SDA pin.

When the I²C controller functions as a slave, its internal clock is synchronized by the incoming clock from SCL line.

4. I²C Master Mode Transmit / Receive Process

When the I²C controller functions in master mode, the I²C master initiates a transaction by transmitting a message to the peripheral (I²C slave) as a transmitter mode. The message specifies a read or write operation. If a read operation is specified, the direction of the transfer is changed at the moment of the first acknowledge, and the called slave receiver becomes a slave transmitter. Otherwise, the master functions as master transmitter continuously.

Before the data exchange, Core must check if the bus is used by other masters by reading the status register(stat_r). If the bus is not used, the address of the slave with which you want to communicate should be written to transmit register(tx_r), and configure the control register to start the cycle. After interrupt happens, confirming the bus is acquired and called slave is responded. When the slave is not responded, the sequence must be ended by stop condition by configuring control register(ctrl_r). If bus is lost and called address is not master itself, I²C block is gone to initial state. But if bus winner calls this master, master is gone into slave mode. If all these processes are okay, then the data transfer follows. Data transfer cycle are started by writing transmit register(tx_r) and configuring control register. When the I²C controller functions as a receiver, tx_r is written with 0xFF. Detailed sequence is described below.

4.1 Master Transmitter Sequence

- Step 1 : read status register. Check if bbusy is cleared
- Step 2 : write slave address to data register
  write 5'b10111 to control register
- Step 3 : wait for interrupt
  write 5'bx0011 to control register
- Step 4 : read status register. Check blost, ack_rpy
  if blost is 1, go to step 1.
  else if ack_rpy is 1 go to step 6.
- Step 5 : write data to transmit register
  if this data is the last, go to step 6.
  wait for interrupt. Go to step 4.
- Step 6 : write 5'b11011 to control register

4.2 Master Receiver Sequence

- Step 1 : read status register.
  Check if bbusy is cleared
- Step 2 : write slave address to data register
  write 5'b10111 to control register
- Step 3 : wait for interrupt
  read status register. Check blost, ack_rpy
  if blost is 1, go to step 1.
  else if ack_rpy is 1, go to step 5.
  write 5'b10001 to control register
  write 0xff to transmit register
- Step 4 : if this data is the last, go to step 5.
  wait for interrupt
  read data from data register
  read status register. Check ack_rpy
  if ack_rpy is 0 go to step 4.
- Step 5 : write 5'b11011 to control register

5. I²C Restart Capability (Combined Mode)

The I²C controller can restart without going to STOP condition. If a I²C master wants to restart after sending/receiving 1 byte data , just keep the start_ctrl and ack_ctrl bit in ctrl_r, and if a master wants to restart after sending/receiving several bytes data, set start_ctrl and ack_ctrl bit in ctrl_r just before sending/receiving last data.

The followings shows the restart operation after transfer 1 byte. It sets start_ctrl and ack_ctrl to 1 at first interrupt.

1. read status register. Check if bbusy is cleared
2. write slave address to data register
   write 5'b10111 to control register
3. wait for interrupt
   read status register. Check ack_rpy
   if ack_rpy is 1, go to step 5.
   else write 5'b10111 to control register
   to restart
4. wait for interrupt.
   Write slave address to data register.
   Write 5'b10001 to control register.
5. wait for interrupt
   if this data is the last, go to step 6.
   read status register. Check ack_rpy
   if ack_rpy is 0 go to step 5.
6. write 5'b11011 to control register

If above data transfer is transmitter, write data to transmit register else if recevier write 8'hff to transmit register.

6. I²C Controller Programming Model

The following paragraphs describe the registers in the I²C controller.

6.1 I²C Control Register (ctrl_r).

ctrl_r is a write-only register that controls the I²C operation mode. Ctrl_r is cleared by reset excluding ack_ctrl bit.

7	6	5	4	3	2	1	0
Reserved	Reserved	Reserved	include0	stop_ctrl	start_ctrl	ack_ctrl	int_en

BIT	NAME	FUNCTION
7:5	Reserved	
4	include0	Indicates whether addr_r[0] bit should be used or not in comparing addresses in slave mode Cleared by reset
3	stop_ctrl	Indicates the transfer cycle should be ended by generating stop condition in master mode Cleared by reset
2	start_ctrl	Indicates the transfer cycle should be started by generating start condition in master mode Cleared by reset
1	ack_ctrl	This bit value directly goes to the SDA pin in acknowledge phase in both master and slave mode Set by reset
0	int_en	Interrupt enable bit Cleared by reset

6.2 I²C Status Register (stat_r).

The stat_r Register is an 8-bit memory mapped and read-only register. The stat_r register shows the state of I²C block.

7	6	5	4	3	2	1	0
Reserved	Reserved	Reserved	Intr	bbusy	blost	ack_rpy	slave

BIT	NAME	FUNCTION
7:5	Reserved	
4	intr	Indicates that interrupt is generated. Used when interrupt polling is used
3	bbusy	Indicates that I²C bus is used
2	blost	Indicates that bus is lost during bus arbitration
1	ack_rpy	Indicates the real state of SDA line in acknowledge phase
0	slave	Indicates that I²C block was called. This bit is set when I²C block was called by other master

6.3 I²C Address Register (addr_r).

The addr_r is an 8-bit write-only register that is used to be accessed by other master.

7	6	5	4	3	2	1	0
addr_r							

BIT	NAME	FUNCTION
7:0	addr_r	Indicates the slave address of the I²C controller. This address is used in comparing the incoming addresses in slave mode

6.4 I²C Baud-Rate Register (baud_r).

I²C block uses the clock generated by dividing system clock which is set in baud_r. Additionally this clock is used to generate the SCL clock which is eight-divided by internal clock. The default value is 4. Assuming system clock is 29Mhz, SCL clock is 300kHz and maximum SCL frequency is 1.8Mhz

7	6	5	4	3	2	1	0
baud_r							

BIT	NAME	FUNCTION
7:0	baud_r	indicates the clock dividing value whose clock signal is used in internal operation internal clock frequency is (baud_r + 2) *2 and when it is set as 255, the internal clock is half-divided .

6.5 I²C Data Register (data_r).

Data register is the 8bit read/write register which is used to send or receive data. Internally this register consists of two registers, transmit register and receive register respectively. Written data is transferred to transmit register, and read data from receive register. In every phase, SDA line is driven by transmit register and SDA line is read by receive register.

7	6	5	4	3	2	1	0
data_r							

BIT	NAME	FUNCTION
7:0	data_r	Written data is serially transmitted through the SDA line, and SDA line data is written In receive register

7. I²C Module Signal Description

The I²C module is connected to the APB bus. Table 1. Signal descriptions describes the APB signals used and produced, Table 1. Signal descriptions shows the non-AMBA signals from the block.

Table 1. Signal Descriptions

NAME	TYPE	SOURCE/ DESTINATION	DESCRIPTION
BnRES	I	Reset Controller	ASB soft reset signal (active LOW).
PA[3:2]	I	APB Bridge	This is part of the peripheral address bus, and is used by the peripheral for decoding its own register accesses. The addresses become valid before PSTB goes to HIGH and remain valid after PSTB goes to LOW.
PD[7:0]	I/O	APB Peripherals	This is the part of the bi-directional peripheral data bus. The data bus is driven by this block during read cycles (when PWRITE is LOW).
PSTB	I	APB Bridge	This strobe signal is used to time all accesses on the peripheral bus. The falling edge of PSTB is coincident with the falling edge of BCLK (ASB system clock).
PWRITE	I	APB Bridge	When this signal is HIGH, it indicates a write to a peripheral, when this signal is LOW, it indicates a read from a peripheral. This signal has the same timing as the peripheral address bus. It becomes valid before PSTB goes to HIGH and remains valid after PSTB goes to LOW.
PSEL	I	APB Bridge	When this signal is HIGH, it indicates the SSPI module has been selected by the APB bridge. This selection is a decode of the system address bus.

Table 2. Specific Block Signal Descriptions

Name	Type	Source / Destination	Description
SDAin	I	PAD	Serial data input.
SDAout	O	PAD	Serial data output.
SCLin	I	PAD	Serial clock in
SCLout	O	PAD	Serial clock output
I²CIRQ	O	Interrupt Controller	Active HIGH Interrupt Request

8. Hardware Interface

The APB interface is fully APB-compliant. The APB is a non pipelined low-power interface, designed to provide a simple interfacing to slave peripherals.

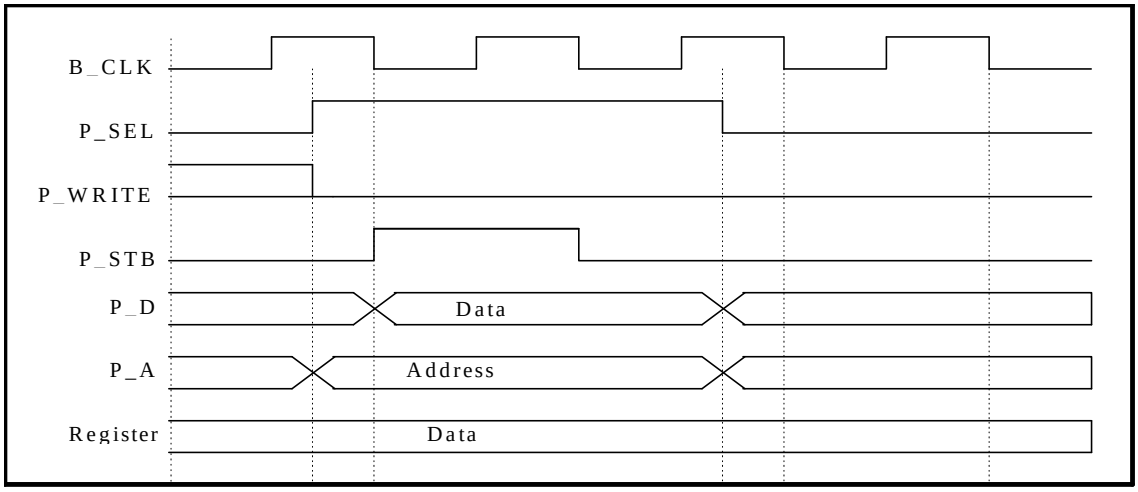


Figure 2. APB Read

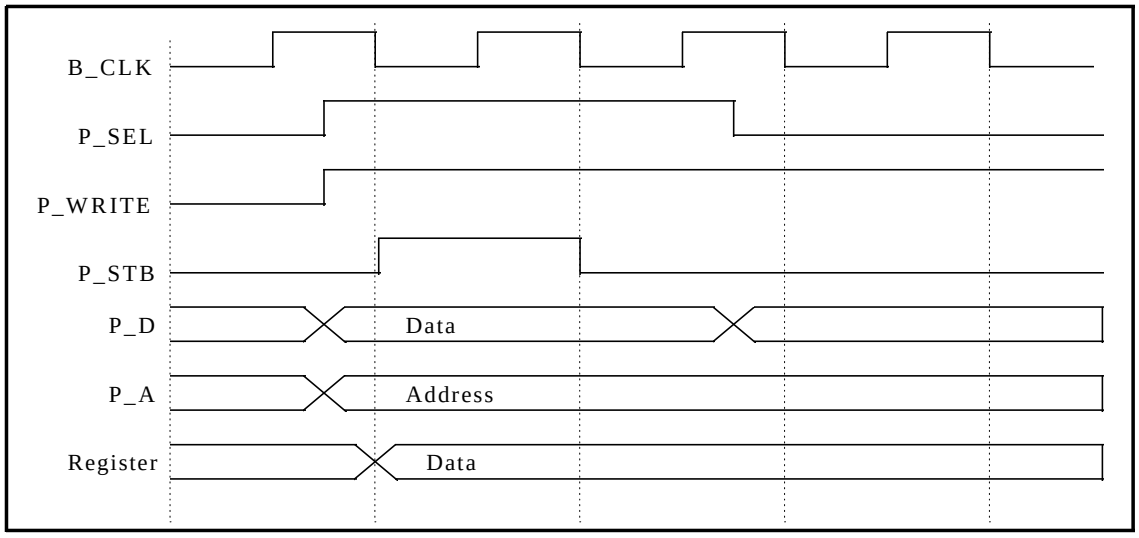


Figure 3. APB Write

9. Register Memory Map

The base address of the I^2C interface is 0xFFFF F900(channel0), 0xFFFF FA000(channel1), and 0xFFFF FB00(channel2), the offset of any particular register from the base address is determined.

ADDRESS	READ LOCATION	WRITE LOCATION
I^2C Base		Baud_r
I^2C Base + 0x04		Ctrl_r
I^2C Base + 0x08	Data_r	Data_r
I^2C Base + 0x0C	Stat_r	
I^2C Base + 0x10		Addr_r
I^2C Base + 0x14	Test_r	Test_r

I^2C Register Memory Map

Section 18. Direct Memory Access Controller

1. General Description

This chip includes a 2-channel direct memory access controller (DMAC). The DMAC can be used in place of the CPU to perform high speed data transfers among external devices equipped;, external memories, memory-mapped external devices. Using the DMAC reduces the burden on the CPU and increases operating efficiency of the MCU.

The features of the DMA Controller are listed below :

- Two Channels with identical function
- Four Gigabytes of address space
- Max. 256 Kbytes transfer
- Data Transfer unit : Byte, Half-word, Word
- Bus mode : Burst mode, Exception mode (Cycle steal mode)
- Two kinds of address modes
  - Single address mode
  - Dual address mode
- Two types of Transfer request sources
  - External I/O request
  - Auto request
- Two kinds of fixed priorities for channels
  - Channel 1's priority > Channel 0's priority
  - Channel 0's priority > Channel 1's priority
- CPU can be interrupted when the specified number of data transfers are completed.

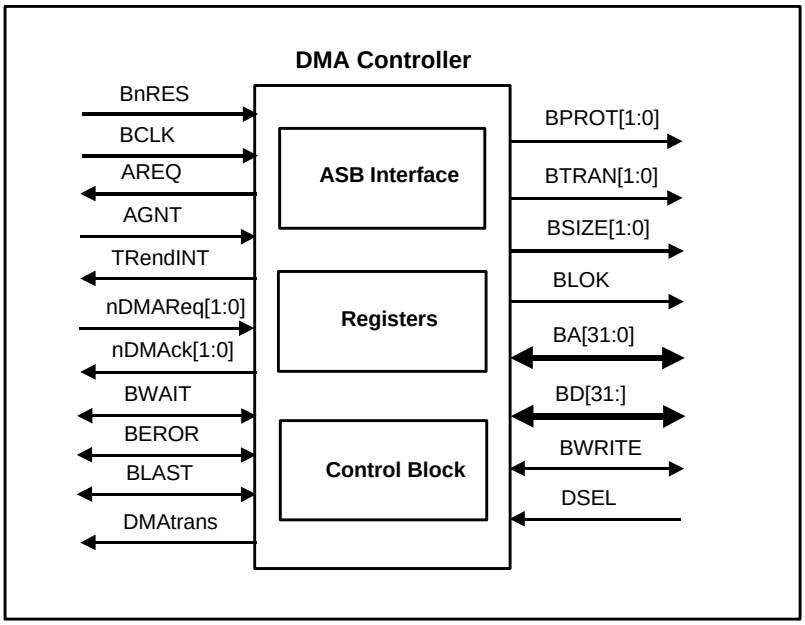


Figure 1. DMAC Top Block Diagram

2. Signal Description

Table 1. DMA Controller Signal Descriptions

NAME	TYPE	DESCRIPTION
BCLK	I	AMBA System bus clock. This clock times all bus transfers. The clock has two distinct phases - phase 1 where BCLK is LOW, and phase 2 where BCLK is HIGH.
BnRES	I	This signal indicates the reset status of the bus.
BA[31:0]	I/O	ASB address. Output for DMAC operation. Input for Register access.
BD[31:0]	I/O	This is the part of Bi-directional system data bus.
AREQ	O	Request signal for ASB Bus mastership.
AGNT	I	Bus Grant signal from ASB arbiter.
BERROR	I/O	ASB error signal.
BLAST	I/O	ASB break burst signal from DRAM Controller.
BLOCK	O	ASB locked transfer signal
BPROT[1:0]	O	ASB master protection information.
BSIZE[1:0]	O	ASB transaction size signal.
BTRAN[1:0]	O	ASB transaction type signal.
BWAIT	I/O	ASB wait transfer signal. Input for DMA cycle stretch. Out for Register access.
BWRITE	I/O	ASB transfer direction signal
DSEL	I	Register select signal
nDMAReq[1:0]	I	DMA transfer request signal from the external I/O device. These are connected to nDMAReq[1:0] pins
nDMAck[1:0]	I	DMA transfer acknowledge signal to the external I/O device. These are connected to nDMAAck[1:0] pins.
DMAtrans	O	Indicate the DRAM access during DMA transfer. This signal connected to DRAM Controller.
TRendINT	O	DMA transfer end interrupt signal to CPU

3. Programmer' s Model

3.1 Memory Map

The base addresses for the DMAC's registers are not fixed and may be different for any particular system implementation. The base address of the DMAC's register is 0xFFFFEE00. However, the offset of any particular DMAC's register from the base address is fixed.

Table 2. External Signal Descriptions

ABBREVIATION	ADDR. OFFSET	NAME	R/W	INITIAL VALUE
SAR0	H'00	Source Address Register for Channel 0	R/W	H'00000000
DAR0	H'04	Destination Address Register for Channel 0	R/W	H'00000000
TNR0	H'08	Transfer Number Register for Channel 0	R/W	H'00001111
CCR0	H'0C	Channel Control Register for Channel 0	R/W	H'00000000
SAR1	H'10	Source Address Register for Channel 1	R/W	H'00000000
DAR	H'14	Destination Address Register for Channel 1	R/W	H'00000000
TNR1	H'18	Transfer Number Register for Channel 1	R/W	H'00001111
CCR1	H'1C	Channel Control Register for channel 1	R/W	H'00000000
TSTR0	H'20	Test Register 0	R/W	H'00000000
TSTR1	H'24	Test Register 1	R	H'00000000
TSTR2	H'28	Test Register 2	R	H'00000000
DMAOR	H'2C	DMA Operation Register	R/W	H'00000000

3.2 Source Address Register 0,1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SAB31	SAB30	SAB29	SAB28	SAB27	SAB26	SAB25	SAB24	SAB23	SAB22	SAB21	SAB20	SAB19	SAB18	SAB17	SAB16

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SAB15	SAB14	SAB13	SAB12	SAB11	SAB10	SAB9	SAB8	SAB7	SAB6	SAB5	SAB4	SAB3	SAB2	SAB1	SAB0

Figure 2. Source Address Register

3.3 Destination Address Register 0, 1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DAB31	DAB30	DAB29	DAB28	DAB27	DAB26	DAB25	DAB24	DAB23	DAB22	DAB21	DAB20	DAB19	DAB18	DAB17	DAB16

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DAB15	DAB14	DAB13	DAB12	DAB11	DAB10	DAB9	DAB8	DAB7	DAB6	DAB5	DAB4	DAB3	DAB2	DAB1	DAB0

Figure 3. Destination Address Register

3.4 Transfer Number Register 0, 1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TNB15	TNB14	TNB13	TNB12	TNB11	TNB10	TNB9	TNB8	TNB7	TNB6	TNB5	TNB4	TNB3	TNB2	TNB1	TNB0

Figure 4. Transfer Number Register

3.5 Channel Control Register 0, 1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Bit	15	14~12	11	10	9	8	7	6 - 5	4	3	2	1	0
	DRAMAcc	ACKLEN	DADRM	SADRM	TSIZE1	TSIZE0	REV.	RTYPE[1:0]	AREQ	TBUSM	TENDFL	INTREN	CHEN

Figure 5. Channel Control Register

Table 3. Channel Control Register

BIT	INIT. VALUE	NAME	DESCRIPTION
15	0	DRAMAcc	Indicate to the DRAM Controller during DMA transfer 0 : not DRAM access 1 : DRAM access
14 ~ 12	000	ACKLEN	Enlarge the LOW phase of the DMAck signal for single address transfer
11	0	DADRM	Destination addressing mode 0 : fixed addressing 1 : incremental addressing
10	0	SADRM	Source addressing mode 0 : fixed addressing 1 : incremental addressing
9 ~ 8	00	TSIZE[1:0]	Transfer size 00 : Byte 01 : Half-word 10 : Word 11 : Reserved
6 ~ 5	00	RTYPE[1:0]	DMA request resource selecting 00 : Memory space to Memory space 01 : Memory space to External IO Device 10 : External IO Device to Memory space Others : Reserved
4	0	ATREQ	Auto request enable
3	0	TBUSM	Transfer bus mode 0 : Burst 1 : cycle-steal mode
2	0	TENDFL	Transfer end flag 0 : incomplete transfer 1 : complete transfer Write 1 to clear this flag bit
1	0	INTREN	DMA transfer complete interrupt enable 0 : interrupt disable 1 : interrupt enable
0	0	CHEN	Channel mode 0 : channel disable 1 : channel enable
Others	0		Reserved

3.6 Test Register 0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	--	--	--	--	--	--	--	--	--	--	--	TCIN1	TCIN0	TREQ1	TREQ0	TMEN

Figure 6. Test Register 0

Table 4. Test Register 0

BIT	INIT. VALUE	NAME	DESCRIPTION
4	0	TCIN1	Carry-in bit of channel 1 counter 0 : carry-in is not occurred 1 : carry-in is occurred
3	0	TCIN0	Carry-in bit of channel 0 counter 0 : carry-in is not occurred 1 : carry-in is occurred
2	0	TREQ1	DMAC request bit of channel 1 0 : request is not occurred 1 : request is occurred
1	0	TREQ0	DMAC request bit of channel 0 0 : request is not occurred 1 : request is occurred
0	0	TMEN	Test mode enable bit 0 : test mode is not enable 1 : test-mode is enable
Others	0		Reserved

3.7 Test Register 1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TA31	TA30	TA29	TA28	TA27	TA26	TA25	TA24	TA23	TA22	TA21	TA20	TA19	TA18	TA17	TA16

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TA15	TA14	TA13	TA12	TA11	TA10	TA9	TA8	TA7	TA6	TA5	TA4	TA3	TA2	TA1	TA0

Figure 7. Test Register 1

Table 5. Test Register 1

BIT	INIT. VALUE	NAME	DESCRIPTION
31 ~ 0	0	TA[31:0]	Latches BA[31:0] signal when TMEN bit of Test Register 0 is high

3.8 Test Register 2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	--	--	--	--	--	--	--	--	--	--	BTRAN1	BTRAN0	BSIZE1	BSIZE0	AREQ	BWRITE

Figure 8. Test Register 2

Table 6. Test Register 2

BIT	INIT. VALUE	NAME	DESCRIPTION
5 ~ 4	00	BTRAN[1:0]	Latches BTRAN signal
3 ~ 2	00	BSIZE[1:0]	Latches BSIZE signal
1	0	AREQ	Latches AREQ signal
0	0	BWRITE	Latches BWRITE signal
Others	0		Reserved

3.9 DMA Operation Register

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	--	--	--	--	--	--	--	--	--	--	--	--	--	--	PRIOM	DMAEN

Figure 9. DMA Operation Register

Table 7. DMA Operation Register

BIT	INIT. VALUE	NAME	DESCRIPTION
1	0	PRIOM	Channel priority level selection bit 0 : Ch0 > Ch1 1 : Ch1 > Ch0
0	0	DMAEN	DMAC operation enable bit 0 : DMAC is not enabled 1 : DMAC is enabled
Others	0		Reserved

4. Address Modes

4.1 Dual Address Mode

The dual address mode of DMAC are described in the Figure. 10. The source and destination area can be external memory, internal SRAM, and external memory mapped I/O device. In dual address mode, both the transfer source and the destination are accessed by an address. DMA Controller read the data source device and store temporarily in temp register in DMAC. And then transfer this data destination device. So read and write transactions are performed in one data transfer. Figure 12. shows the DMA transfer timing in the dual address mode.

The example of the register setting for dual address mode is following that : 1) set Source Address Register (SAR). In this register you write the source area address. 2) set Destination Address Register (DAR). In this register you write the destination area address. 3) set Transfer Number Register (TNR). In this register you can write the transfer number value. 4) set Channel Control Register (CCR). In this register you should write properly the control value. RTYPE[1:0] field value are should be “00”, and DARAM, SADRM fields should be “1”. You should set the TSIZE[1:0] field properly by the transferred data width. And you must set the CHEN field to “1”. INTREN field are set by your need. 5) finally set DMAEN field to “1” in DMA Operation Register (DMAOR). If you set AutoReq field in CCR, as soon as set DMAEN field, DMA transfer the data.

Figure 11. show the timing diagram of DMA transfer when CCR value is 0x0D13.

Note) In the end of DMA transfer, DMA Controller is initialized by clearing the TENDFL field in CCR. This can be performed by writing “1” value in this field.

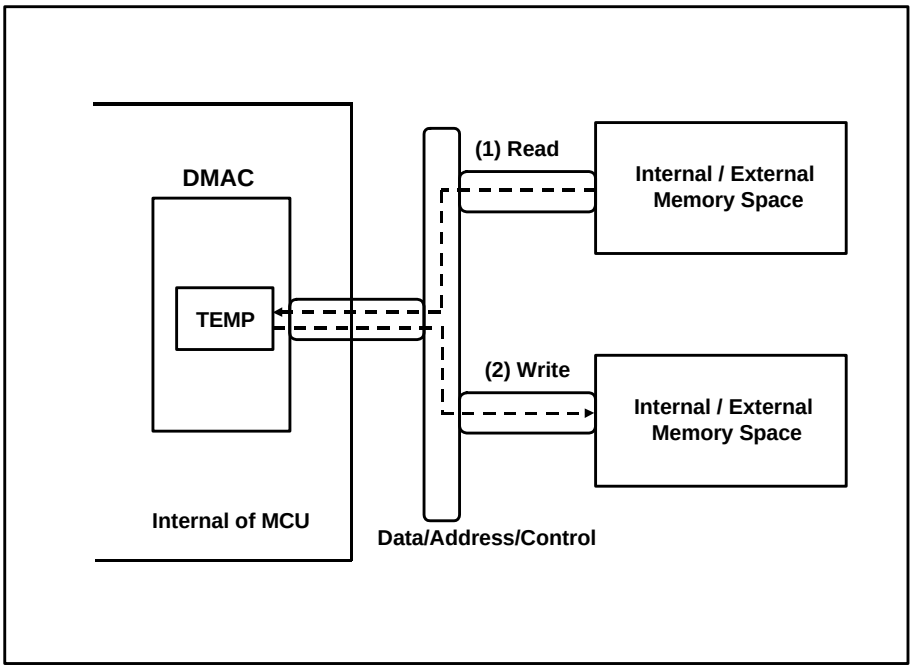


Figure 10. DMAC Dual Address Mode Block Diagram

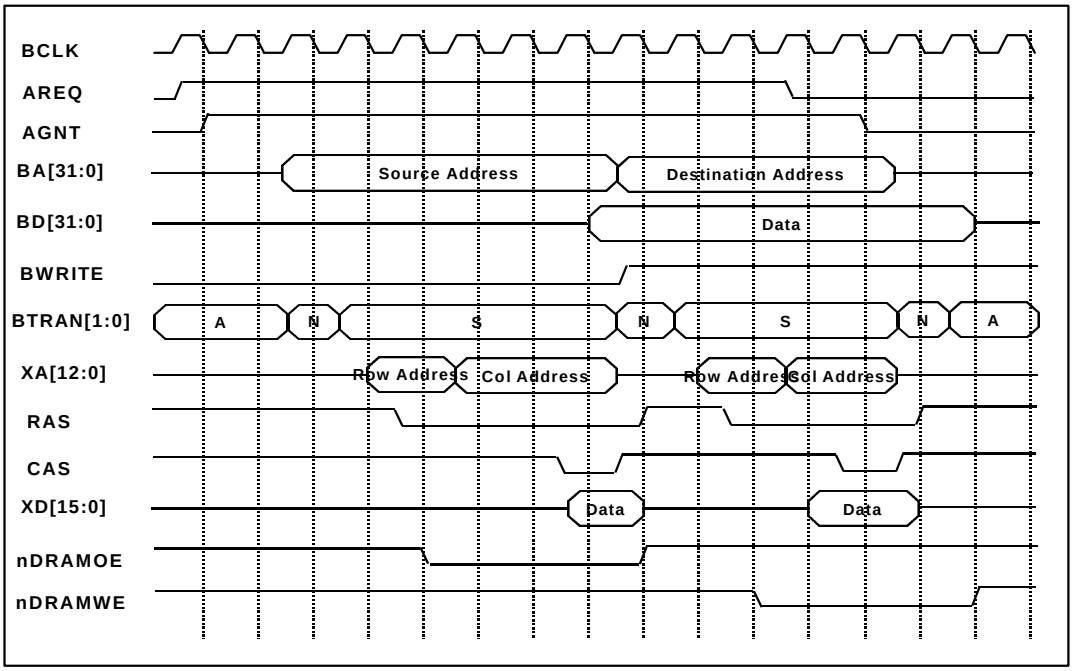


Figure 11. Transfer Timing in Dual Address Mode (Memory Space to Memory Space)

4.2 Single Address Mode

The single address mode of DMAC are described in the Figure 12. In single address mode, there are two types of transfer, one is memory-to-I/O device that source is memory area and destination is external I/O device, and the other is external I/O device–to-memory that source is I/O device and destination is memory area. Single address transfer mode is composed of only one data transaction, so it is very fast to transfer the data comparing with dual addres mode in which composed of 2 transactions : read transaction and write transactio. The I/O device is acessed by nDREQ and nDACK signals. nDACK signals are controlled by the ACKLEN field in CCR. And nDACK signals are coincident with the RAS and CAS signals from DRAM Controller. When memory is DRAM, you must set properly the control flags in the control register in DRAM Controller and DMA Controller.

The example of the register setting for memory-to-I/O device transfer in single address mode is following that : 1) set Source Address Register (SAR). In this register you write the source area address. 2) set Destination Address Register (DAR). In this register you write the destination area address. 3) set Transfer Number Register (TNR). In this register you write the transfer number value. 4) set Channel Control Register (CCR). In this register you should write properly the control value. When data transfer is the memory space to external I/O devices, RTYPE[1:0] field value are should be “01”, and DARAM field is should be “0”, and SADRM fields should be “1”. You should set the TSIZE[1:0] field properly by the transferred data width. And you must set the CHEN field to “1”. INTREN field are set by your need. 5) finally set DMAEN field to “1” in DMA Operation Register (DMAOR). If you set AutoReq field in CCR, as soon as set DMAEN field, DMA transfer the data.

Figure 13 shows the timing diagram of the memory-to-I/O device transfer when CCR value is 0x0523. Figure 14 shows the timing diagram of the timing diagram of the I/O device-to-memory transfer when CCR value is 0x0943. In these case the contents of the control register for DRAM in DRAM Controller is 0x69. In the end of DMA transfer, DMA Controller is initialized by clearing the TENDFL field in CCR. This can be performed by writing “1” value in this field.

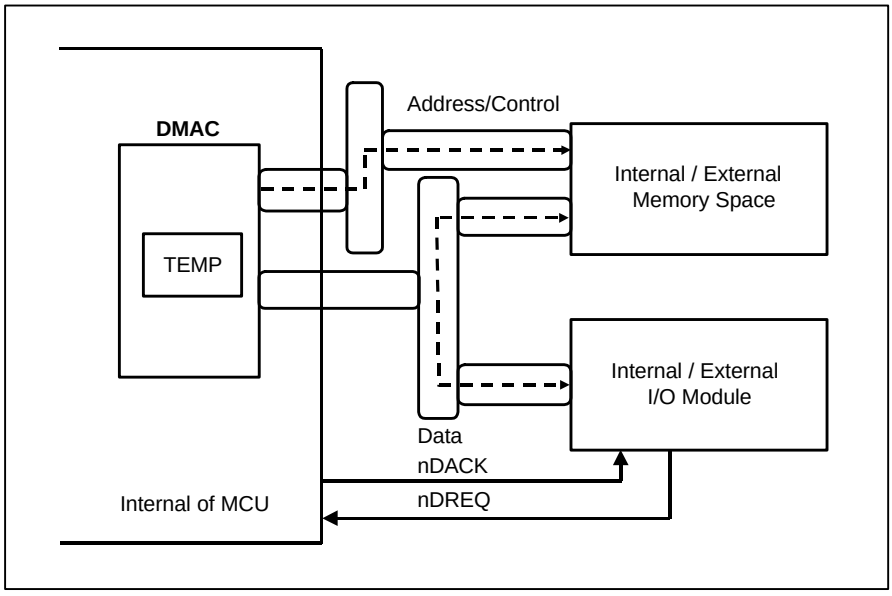


Figure 12. DMAC Single Address Mode Block Diagram

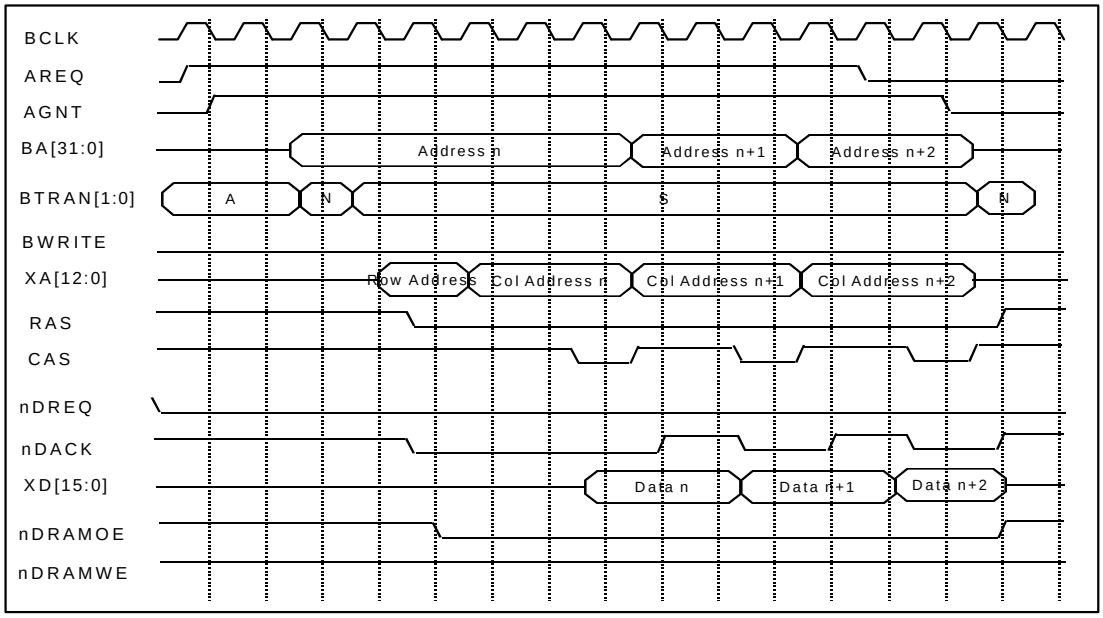


Figure 13. Transfer Timing in Single Burst Address Mode (Memory Space to I/O Device with nDMACK)

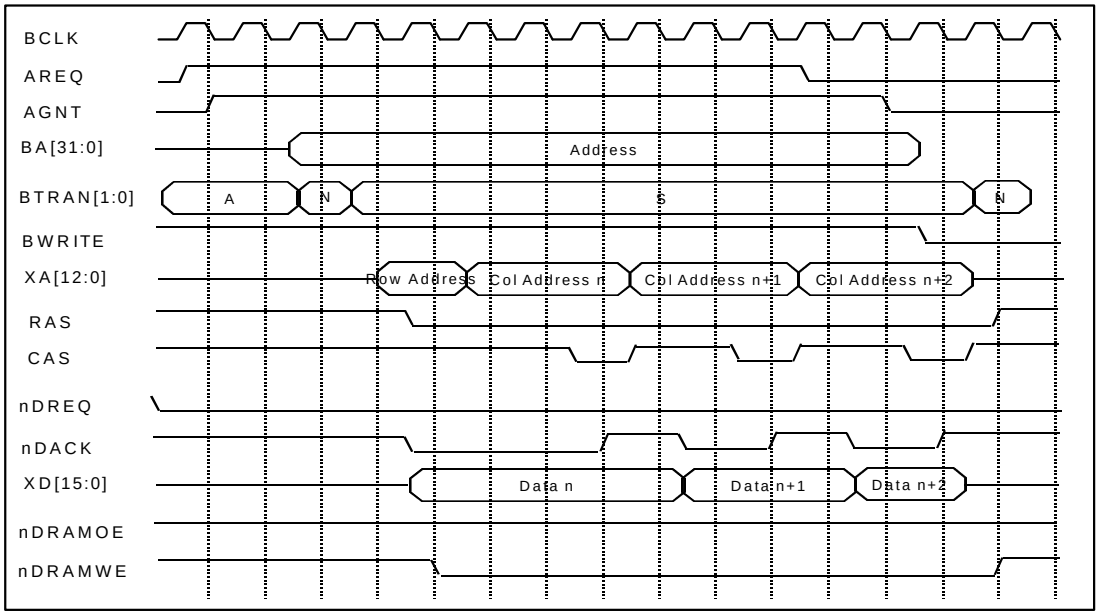


Figure 14. Transfer Timing in Single Burst Address Mode (I/O Device with nDMACK to Memory Space)

Section 19. Debug and Test Interface

1. General Description

The GDC21D601 has built-in features which enable debug and test in a number of different contexts. Firstly, there are circuit structures to help with software development.

Secondly, the device contains boundary scan cells for circuit board test. Finally, the device contains some special test modes which enable the generation production patterns for the device itself.

2. Software Development Debug and Test Interface

The ARM720T Core and numerous peripherals implemented inside GDC21D601 contain hardware extensions for advanced debugging features. These are intended to ease user development and debugging of application software, operating systems, and the hardware itself. Full details of the debug interfaces and their programming can be found in ARM720T Data Sheet (ARM DDI-0087). The MultiICE product enables the ARM720T to be debugged in one environment. Refer to Guide to MultiICE (ARM DUI-0048).

3. Test Access Port and Boundary Scan

GDC21D601 contains full boundary scan on its inputs and outputs to help with circuit board test. This supports both INTEST and EXTEST, allowing patterns to be applied serially to the GDC21D601 when fixed in a board and for full circuit board connection respectively.

The boundary-scan interface conforms to the IEEE Std. 1149.1- 1990, Standard Test Access Port and Boundary-Scan Architecture. (Please refer to this standard for an explanation of the terms used in this section and for a description of the TAP controller states.) The boundary-scan interface provides a means of testing the core of the device when it is fitted to a circuit board, and a means of driving and sampling all the external pins of the device irrespective of the core state. This latter function permits testing of both the device' selectrical connections to the circuit board, and (in conjunction with other devices on the circuit board having a similar interface) testing the integrity of the circuit board connections between devices. The interface intercepts all external connections within the device, and each such " ell?is then connected together to form a serial register (the boundary scan register). The whole interface is controlled via 5 dedicated pins: **TDI**, **TMS**, **TCK**, **nTRST** and **TDO**.

Full details of the debug interfaces and their programming can be found in ARM720T Data Sheet (ARM DDI-0087E, Section 8 Debug Architecture).

3.1 Reset

The boundary-scan interface includes a state-machine controller (the TAP controller).
A pulldown resistor is included in the **nTRST** pad which holds the TAP controller state machine in a safe state after power up. In order to use the boundary scan interface, **nTRST** should be driven HIGH to take the TAP state machine out of reset.

The action of reset (either a pulse or a DC level) is as follows:

* System mode is selected (i.e. the boundary scan chain does NOT intercept any of the signals passing between the pads and the core).
* IDcode mode is selected. If **TCK** is pulsed, the contents of the ID register will be clocked out of **TDO**.

Note : The TAP controller inside ARM7201 contains a scan chip register which is reset to the value b0011 thus selecting the boundary scan chain. If this register is programmed to any value other than b0011, then it must be reprogrammed with b0011 or a reset applied before boundary scan operation can be attempted.

3.2 Pullup Resistors

The IEEE 1149.1 standard requires pullup resistors in the input pins. However, to ensure safe operation an internal pulldown is present in the **nTRST** pin and therefore will have to be driven HIGH when using this interface.

Table 1. Internal Resistors for Input Pins

PIN	INTERNAL RESISTOR
TCLK	Pullup
nTRST	Pulldown
TMS	Pullup
TDI	Pullup

Section 20. Electrical Ratings

1. Absolute Maximum Ratings

2. Thermal Characteristics

3. D.C Electrical Characteristics

APENDIX A. Register Map

The following diagram shows System Memory Map of GDC21D601.

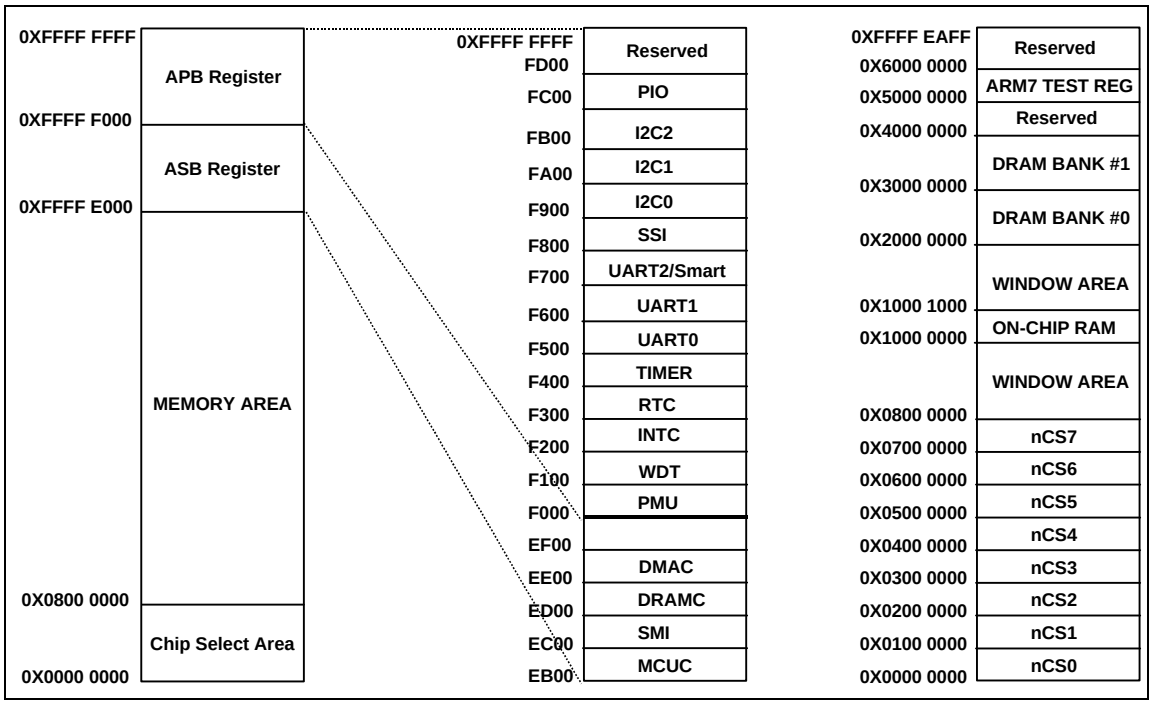


Figure 1. System Memory Map

DMA Controller Registers(@0xFFFF EE00)

ABBREVIATION	ADDRESS	DESCRIPTIONS	R/W	INITIAL VALUE
SAR0	0xFFFFEE00	Source Address Register for Channel 0	R/W	32h'00000000
DAR0	0xFFFFEE04	Destination Address Register for Channel 0	R/W	32h'00000000
TNR0	0xFFFFEE08	Transfer Number Register for Channel 0	R/W	32h'00001111
CCR0	0xFFFFEE0C	Channel Control Register for Channel 0 Bit 15 : DRAM Access indicator 0: Not DRAM 1: DRAM transfer Bit 14-12 : DMAck Low length Bit 11 : Destination Addressing mode 0: Fixed 1: Incremental addressing Bit 10 : Source Addressing mode 0: Fixed 1: Incremental addressing Bit 9-8 : Transfer size 00:Byte 01:Half-word 10:Word 11:Reserved Bit 7 : Reserved Bit 6-5 : DMA Request mode 00 : Mem. space to Mem. space 01 : Mem.External IO Device 10 : Ext. IO Device to Mem. 11 : Reserved Bit 4 : Auto Request Enable Bit 3 : Transfer bus mode 0:Burst 1:Cycle-steal mode Bit 2 : Transfer End flag 0:incomplete 1: complete Bit 1 : Transfer complete interrupt Enable 0: disabled 1: enabled Bit 0 : Channel control 0: Ch 0 disabled 1: Ch 0 enabled	R/W	32h'00000000
SAR1	0xFFFFEE10	Source Address Register for Channel 1	R/W	32h'00000000
DAR	0xFFFFEE14	Destination Address Register for Channel 1	R/W	32h'00000000
TNR1	0xFFFFEE18	Transfer Number Register for Channel 1	R/W	32h'00001111
CCR1	0xFFFFEE1C	Channel Control Register for channel 1	R/W	32h'00000000
TSTR0	0xFFFFEE20	Test Register 0	R/W	32h'00000000
TSTR1	0xFFFFEE24	Test Register 1	R	32h'00000000
TSTR2	0xFFFFEE28	Test Register 2	R	32h'00000000
DMAOR	0xFFFFEE2C	DMA Operation Register Bit 7-2 : Reserved Bit 1: Channel priority level select 0: Ch0 > Ch1 1: Ch0 < Ch1 Bit 0 : DMAC operation control 0: enabled 1: disabled	R/W	32h'00000000

DRAM Controller Registers(@0xFFFFED00)

ABBREVIATION	ADDRESS	DESCRIPTIONS	R/W	INITIAL VALUE
DRAMRCR	0xFFFFED00	DRAM Refresh Control Register Bit 15-8: REFCNT (DRAM Refresh Clock Divisor) RefClock = BCLK / REFCNT Bit 7 : DRAM Refresh clock control 0 / 1 = Disabled / Enabled Bit 6-0 : REFDIV(DRAM Refresh rate) Frequency (KHz) = 2*[RefClock /(RFDIV + 1)]	W	16h'0000
DRAMconCPU	0xFFFFED04	DRAM Controller for CPU Bit 15-7: Reserved Bit 6: DMAEn(DMA Control signal) 0 / 1 = DMA Disabled / Enabled Bit 5: TRP(t_{rp} = |RAS-CAS|) Bit 4 : TCP(t_{cp}= |Low phase of CAS|) Bit 3-2: Wait count 00: 0-wait 01: 1-wait 10: 2-wait 11: 3-wait Bit 1-0: Bank size 00 : byte 01 : Half-word 10 : Word 11 : Reserved	R/W	7b'0000000
DRAMCDMA	0xFFFFED08	DRAM Controller for DMA Bit 15-6: Reserved Bit 5: TRP(t_{rp} = |RAS-CAS|) Bit 4 : TCP(t_{cp}= |Low phase of CAS|) Bit 3-2: Wait count 00: 0-wait 01: 1-wait 10: 2-wait 11: 3-wait Bit 1-0: Bank size 00 : byte 01 : Half-word 10 : Word 11 : Reserved	R/W	6b'000000

DRAM Controller Registers(@0xFFFFED00) -- Continued

ABBREVIATION	ADDRESS	DESCRIPTIONS	R/W	INITIAL VALUE
DRAMTCR	0xFFFFED0C	DRAM Test Control Register Bit 15-4: Reserved Bit 3: TESTINC(Test Increment) Column address auto increment 0 / 1 = Disabled / Enabled Bit 2: FORCEADV Forces refresh counter by BCLK 0 / 1 = Disabled / Enabled Bit 1-0: Force size 00 : byte 01 : Half-word 10 : Word (Default) 11 : Reserved	W	4b'0010

Static Memory Controller Registers(@0xFFFF F000)

ABBREVIATION	ADDRESS	DESCRIPTIONS	R/W	INITIAL VALUE
MEMCFG1	0xFFFFEC00	Memory Configuration Register 1 **nCS1 area control** Bit 31-30: Reserved Bit 29: EXPRDY polarity (0:Active High) Bit 28: Control signal type 0/1=ARM type / Motorola type: Bit 27: FlashON –Reserved Bit 26: Expansion clock Enable Bit 25-24: Mem width 00: 32-bit memory 01: 16-bit memory 10: 8-bit memory 11: Reserved Bit 23: Burst mode Enable – Reserved Bit 22-20: Burst wait cycle –Reserved Bit 19-16: Access wait cycle(1~16cycles) 0000(1 cycle) ~ 1111(16 cycles)	R/W	32'h0000000 4 0 : Default 1 cycle
		nCS0 area control Bit 15-14: Reserved Bit 13: EXPRDY polarity (0:Active High) Bit 12: Control signal type 0/1=ARM type / Motorola type: Bit 11: FlashON –Reserved Bit 10 : Expansion clock Enable Bit 9-8: Mem width 00: 32-bit memory 01: 16-bit memory 10: 8-bit memory 11: Reserved Bit 7: Burst mode Enable – Reserved Bit 6-4: Burst wait cycle – Reserved Bit 3-0: Access wait cycle(1~16cycles) 0000(1 cycle) ~ 1111(16 cycles)		 4 : Default 5 cycles

Static Memory Controller Registers(@0xFFFF F000) -- Continued

ABBREVIATION	ADDRESS	DESCRIPTIONS	R/W	INITIAL VALUE
MEMCFG2	0xFFFFEC04	Memory Configuration Register 2 NCS3 area control Bit 31-30: Reserved Bit 29: EXPRDY polarity (0:Active High) Bit 28: Control signal type 0/1=ARM type / Motorola type Bit 27: FlashON –Reserved Bit 26: Expansion clock Enable Bit 25-24: Mem width 00: 32-bit memory 01: 16-bit memory 10: 8-bit memory 11: Reserved Bit 23: Burst mode Enable – Reserved Bit 22-20: Burst wait cycle –Reserved Bit 19-16: Access wait cycle(1~16cycles) 0000(1 cycle) ~ 1111(16 cycles)	R/W	32'h0000000 0 Default 1 cycle
		NCS2 area control Bit 15-14: Reserved Bit 13: EXPRDY polarity (0:Active High) Bit 12: Control signal type 0/1=ARM type / Motorola type: Bit 11: FlashON –Reserved Bit 10 : Expansion clock Enable Bit 9-8: Mem width 00: 32-bit memory 01: 16-bit memory 10: 8-bit memory 11: Reserved Bit 7: Burst mode Enable – Reserved Bit 6-4: Burst wait cycle –Reserved Bit 3-0: Access wait cycle(1~16cycles) 0000(1 cycle) ~ 1111(16 cycles)		Default 1 cycle
MEMCFG3	0xFFFFEC08	Memory Configuration Register 3 NCS4,3 area controls	R/W	32'h0000000 0
MEMCFG4	0xFFFFEC0C	Memory Configuration Register 4 NCS6,5 area controls	R/W	32'h0000000 0

MCU Controller Registers(@0xFFFF EC00)

ABBREVIATION	ADDRESS	DESCRIPTIONS	R/W	INITIAL VALUE
MCUC	0xFFFFEB00	MCU Control Register	R/W	8h'00
PINMUX_PA	0xFFFFEB04	Port A MUX Register (Port A[5:0])	R/W	8h'00
PINMUX_PB	0xFFFFEB08	Port B MUX Register (Port B[7:0])	R/W	8h'00
PINMUX_PC	0xFFFFEB0C	Port C MUX Register (Port C[7:0])	R/W	8h'00
PINMUX_PD	0xFFFFEB10	Port D MUX Register (Port D[7:0])	R/W	8h'00
PINMUX_PE	0xFFFFEB14	Port E MUX Register (Port E[7:0])	R/W	8h'00
PINMUX_PF	0xFFFFEB18	Port F MUX Register (Port F[7:0])	R/W	8h'00
PINMUX_PG	0xFFFFEB1C	Port G MUX Register (Port G[7:0])	R/W	8h'00
PINMUX_PH	0xFFFFEB20	Port H MUX Register (Port H[7:0])	R/W	8h'00
PINMUX_PI	0xFFFFEB24	Port I MUX Register (Port I[7:0])	R/W	8h'00
PINMUX_PJ	0xFFFFEB28	Port J MUX Register (Port J[7:0])	R/W	8h'00
MCUDC	0xFFFFEB2C	MCU Device Code Register	R	24h'LG601
DRAMPDACK	0xFFFFEB30	DRAM Power Down Ack	R	8h'00
DRAMPDREQ	0xFFFFEB34	DRAM Power Down Request	R/W	8h'00

PMU Registers (@0xFFFF F000)

ABBREVIATION	ADDRESS	DESCRIPTIONS	R/W	INITIAL VALUE
PMUCR	0xFFFFF000	PMU Control Register Only following values are effective. 0x00 := Clear PMU status register 0x03 := Enters the Power Down mode PMU Status Register Bit 7-6: Reserved Bit 5-4: Previous Reset Status 00 : Power-On Reset status 01 : S/W Reset state by PMU 10 : S/W Manual Reset by WDT 11 : WD overflow Reset state by WDT Bit 3-2: Current status flag bits 00 : Running state after nPOR 01 : Running state after WD_OF 10 : Running state after Man_Reset 11 : Reserved Bit 1-0: Previous status flag bits 00 : Start state after nPOR 01 : Start state after WD_OF 10 : Start state after Man_Reset 11 : Start state after PD mode	W R	8h'00
BCLKCR	0xFFFFF004	BCLK frequency selection and BUS mode control(Standard / Fast BUS mode) Bit 7-4: Reserved Bit 3: FCLK control bit 0 – Fast-bus mode (not use the FCLK) 1 – Standard-bus mode Use FCLK as SYS_CLK Bit 2-0: BCLK selection bits 000 : BCLK = SYS_CLK / 2 001 : BCLK = SYS_CLK / 4 010 : BCLK = SYS_CLK / 8 011 : BCLK = SYS_CLK / 16 100 : BCLK = SYS_CLK / 32 101 : BCLK = SYS_CLK / 64 110 : BCLK = SYS_CLK / 128 111 : BCLK = SYS_CLK.	R/W	8h'00

PMU Registers (@0xFFFF F000) -- Continued

ABBREVIATION	ADDRESS	DESCRIPTIONS	R/W	INITIAL VALUE
BCLKMSK_RUN	0xFFFFF008	BCLK Masking controls register in the RUN mode. Enable / Disable clock : 1/0 Bit 15-13 : Reserved Bit 12 : APB Bridge clock control Bit 11 : BUS Controller clock control Bit 10 : DRAM Controller clock control Bit 9 : DMA Controller clock mask bit Bit 8 : TEST Controller clock mask bit Bit 7 : SRAM clock mask bit Bit 6-1 : Reserved Bit 0 : B_CLK Out mask bit	R/W	16h'FFFF
BCLKMSK_PD	0xFFFFF00C	BCLK controls register (PD mode.) Enable / Disable clock : 1/0 Bit 15 : ARM7TDMI Core clock control Bit 14 : AMBA Arbiter clock control Bit 13 : AMBA Decoder clock control Bit 12 : APB Bridge clock control Bit 11 : BUS Controller clock control Bit 10 : DRAM Controller clock control Bit 9 : DMA Controller clock control Bit 8 : TEST Controller clock control Bit 7 : SRAM clock control Bit 6-1 : Reserved Bit 0 : B_CLK Out control	R/W	16h'0000
REMAP	0xFFFFF010	REMAP register	R/W	8h'00
PCLKCR	0xFFFFF014	PCLK control register Bit 7-3 : Reserved Bit 2-0: PCLK selection 000 : PCLK = external PCLK source 001 – PCLK = SCLK / 2 010 – PCLK = SCLK / 4 011 – PCLK = SCLK / by 8 100 – PCLK = SCLK / 16 101 – PCLK = SCLK / 32 110 – PCLK = SCLK / 64 111 – PCLK = SCLK / 128	R/W	8h'00

PMU Registers (@0xFFFF F000) -- Continued

ABBREVIATION	ADDRESS	DESCRIPTIONS	R/W	INITIAL VALUE
PCLKMSK_RUN	0xFFFFF018	PCLK control register in RUN mode. Enable / Disable clock : 1/0 Bit 9 : Watchdog Timer clock control Bit 8 : I^2C 2 clock control Bit 7 : I^2C 1 clock control Bit 6 : I^2C 0 clock control Bit 5 : SSPI 1 clock control Bit 4 : SSPI 0 clock control Bit 3 : UART 2/ SMIC clock control Bit 2 : UART 1 clock control Bit 1 : UART 0 clock control Bit 0 : Timer clock control	R/W	10h'11FF
PCLKMSK_PD	0xFFFFF01C	PCLK control register in PD mode. Enable / Disable clock : 1/0 Bit 9 : Watchdog Timer clock control Bit 8 : I^2C 2 clock control Bit 7 : I^2C 1 clock control Bit 6 : I^2C 0 clock control Bit 5 : SSPI 1 clock control Bit 4 : SSPI 0 clock control Bit 3 : UART 2/ SMIC clock control Bit 2 : UART 1 clock control Bit 1 : UART 0 clock control Bit 0 : Timer clock control	R/W	8h'00
Reserved	0xFFFFF020	Reserved	R/W	8h'00
RSTCR	0xFFFFF030	Reset control register Manual / Normal Reset : 1 / 0 Bit 7-1 : Reserved Bit 0 : Manual Reset control	R/W	8h'00
TSTCR	0xFFFFF040	TIC test mode and PMU test control register Normal Operation / TIC-PMU test : 1 / 0 Bit 7-2 : Reserved Bit 1 : TIC test mode when set to 1 Bit 0 : PMU test mode when set to 1	R/W	8h'00
Device Code	0xFFFFF044	Reserved	R	$LG601
TSTR0	0xFFFFF048	Test write register for external input signals Bit 7-3 : Reserved Bit 2 : Test bit for INT_REQ_IN input Bit 1 : Test bit for WD_OF_IN input Bit 0 : Test bit for MAN_RESET input	R	8h'00

PMU Registers (@0xFFFF F000) -- Continued

ABBREVIATION	ADDRESS	DESCRIPTIONS	R/W	INITIAL VALUE
TSTR1	0xFFFFF04C	Test read register for clocks of ASB devices and reset signals Bit 15 : Test bit for BCLK_WDT Bit 14 : Test bit for PCLK_I^{2}C 2 Bit 13 : Test bit for PCLK_I^{2}C 1 Bit 12 : Test bit for PCLK_I^{2}C 0 Bit 11 : Test bit for PCLK_SSPI 1 Bit 10 : Test bit for PCLK_SSPI 0 Bit 9 : Test bit for PCLK_UART 2 Bit 8 : Test bit for PCLK_UART 1 Bit 7 : Test bit for PCLK_UART 0 Bit 6 : Test bit for BCLK_TIMER Bit 5 : Test bit for B_RESETn Bit 4 : Test bit for P_RESETn0 Bit 3 : Test bit for P_RESETn1 Bit 2 : Test bit for P_RESETn Bit 1 : Test bit for WD_OF_OUT Bit 0 : Test bit for REMAP	R	16h'0000

Watchdog Timer Register (@0xFFFF F100)

ABBREVIATION	ADDRESS	DESCRIPTIONS	R/W	INITIAL VALUE
WDTC	0xFFFFF100	Watchdog Timer Control Register Bit 7: Interrupt Request 0 / 1 = Interrupt Disable / Enable Bit 6: Timer Mode Select 0 / 1 = Interval / Watchdog Bit 5: Timer Control 0 / 1 = Disable / Enable Bit 4: WD Reset Control 0 / 1 = Disable / Enable Bit 3: Reset Select (if WD overflowed) 0 / 1 = POR / Manual Reset Bit 2-0: Clock Select 000 = CKS0 / 2 001 = CKS1 / 8 010 = CKS2 / 32 011 = CKS2 / 64 100 = CKS2 / 256 101 = CKS2 / 512 110 = CKS2 / 2048 111 = CKS2 / 8192	R/W	8b'00000000
WDTS	0xFFFFF104	Watchdog Timer Reset Status Register Bit 1: ITOVF(internal timer overflowed) Bit 0: WTOVF(WD timer overflowed)	R	2b'00
WDTCNT	0xFFFFF108	Watchdog Timer Counter Register	R/W	8h'00
WDTEST	0xFFFFF10C	Watchdog Timer Test Input Register	R/W	8h'00
WDTESTO	0xFFFFF114	Watchdog Timer Test Output Register	R/W	8h'00

RTC Control Register (@0xFFFF F300)

ABBREVIATION	ADDRESS	DESCRIPTIONS	R/W	INITIAL VALUE
RTCDR	0xFFFFF300	RTC Data Register	R/W	0x00
RTCMR	0xFFFFF304	RTC Match Register	R/W	0x00
RTCS	0xFFFFF308	RTC status	R	0x00
RTCDV	0xFFFFF30C	RTC clock divider	R/W	0x00
RTCCR	0xFFFFF310	RTC control register	R/W	0x00
RTCTS	0xFFFFF314	RTC TIC Selection Register	W	0x00
RTCTIC32K	0xFFFFF318	TicCLK32K	W	0x00
RTCTICPCLK	0xFFFFF31C	TicCLKPCLK	W	0x00

Interrupt Controller Register (@0xFFFF F200)

<table>
<tr><th>ABBREVIATION</th><th>ADDRESS</th><th>DESCRIPTIONS</th><th>R/W</th><th>INITIAL VALUE</th></tr>
<tr><td>INTC</td><td>0xFFFFF200</td><td>Interrupt Control/Mask Register
INT25 : SWI(Software Interrupt)
INT24 : Timer 5
INT23 : Timer 4
INT22 : Timer 3
INT21 : Timer 2
INT20 : Timer 1
INT19 : Timer 0
INT18 : SSPI 1
INT17 : SSPI 0
INT16 : Smart Card Interface
INT15 : UART 1
INT14 : UART 0
INT13 : I^{2}C 2
INT12 : I^{2}C 1
INT11 : I^{2}C 0
INT10: WDT(Watchdog Timer)
INT9 : RTC(Real Time Clock)
INT8 : DMA(Direct Memory Access)
INT7 : COM RX
INT6 : COM TX
INT5 : External Interrupt 5
INT4 : External Interrupt 4
INT3 : External Interrupt 3
INT2 : External Interrupt 2
INT1 : External Interrupt 1
INT0 : External Interrupt 0</td><td>R/W</td><td>26h’3FFFFFF</td></tr>
<tr><td>INTMD</td><td>0xFFFFF204</td><td>Interrupt Trigger Mode control</td><td>R/W</td><td>26h’2000000</td></tr>
<tr><td>INTPOL</td><td>0xFFFFF208</td><td>Interrupt Trigger Polarity control Register.
<table><tr><th>Mode</th><th>Polarity</th><th>Triggered by</th></tr><tr><td>0</td><td>0</td><td>Falling Edge</td></tr><tr><td>0</td><td>1</td><td>Rising Edge</td></tr><tr><td>1</td><td>0</td><td>Low Level</td></tr><tr><td>1</td><td>1</td><td>High Level</td></tr></table></td><td>R/W</td><td>26h’2000000</td></tr>
<tr><td>INTDIR</td><td>0xFFFFF20C</td><td>Interrupt direction control Register
(0 / 1 = Request IRQ / FIQ)</td><td>R/W</td><td>26h’0000000</td></tr>
<tr><td>INTFIQS</td><td>0xFFFFF210</td><td>FIQ Status Flag Register (0/1 = Clear/Set)</td><td>R</td><td>26h’0000000</td></tr>
<tr><td>INTIRQS</td><td>0xFFFFF214</td><td>IRQ Status Flag Register(0/1 = Clear/Set)</td><td>R</td><td>26h’0000000</td></tr>
<tr><td>INTFIQMSK</td><td>0xFFFFF218</td><td>FIQ Mask Register(0/1 = Clear/Masked)</td><td>R/W</td><td>26h’0000000</td></tr>
<tr><td>INTIRQMSK</td><td>0xFFFFF21C</td><td>IRQ Mask Register(0/1 = Clear/Masked)</td><td>R/W</td><td>26h’0000000</td></tr>
<tr><td>INTSCL</td><td>0xFFFFF220</td><td>Interrupt Status Clear Register
0 / 1 = Clear / Set</td><td>W</td><td>26h’0000000</td></tr>
<tr><td>INTTICIN</td><td>0xFFFFF240</td><td>TIC Input Register</td><td>R/W</td><td>0x00</td></tr>
<tr><td>INTTICOUT</td><td>0xFFFFF280</td><td>TIC Output Register</td><td>R/W</td><td>0x00</td></tr>
</table>

General Purpose Timer Unit Control Register(@0xFFFFF400)

ABBREVIATION	ADDRESS	DESCRIPTIONS	R/W	INITIAL VALUE
TSTARTR	0xFFFFF400	Timer Start Register Bit 7-6 : Reserved Bit 5-0 : 1:start counting, 0:stop counting	R/W	8b'11000000
TSYNCR	0xFFFFF404	Timer Sync. Register Bit 7-6 : Reserved Bit 5-0 : 1:sync. mode, 0:normal mode	R/W	8b'11000000
TPWMR	0xFFFFF408	Timer PWM Register(1=PWM,0=normal) Bit 7-6 : Reserved Bit 5-0 : 1: PWM mode, 0:normal mode	R/W	8b'11000000
TSTINR	0xFFFFF40C		W	
TSTOUTR	0xFFFFF410		R	
TSTMODER	0xFFFFF414		W	
TSTINTR	0xFFFFF418		R	
TCONTR0	0xFFFFF420	Timer 0 Control Register 8b'1xx11yyy xx : mode selection 00 : not cleared-Free running mode 01 : cleared by GRA(periodic mode) 10 : cleared by GRB(periodic mode) 11 : cleared in sync w/ other sync. timer yyy : timer clock prescaler selection 000 : timer clock = BCLK/2 001 : timer clock = BCLK/4 010 : timer clock = BCLK/16 011 : timer clock = BCLK/64 100 : timer clock = EXT_CLK/2 101 : timer clock = EXT_CLK/4 110 : timer clock = EXT_CLK/64 111 : timer clock = EXT_CLK/2	R/W	8b'10011000
TIOCR0	0xFFFFF424	Timer 0 I/O Control Register 8b'1xxx1yyy xxx : GRB function select 000 : compare match with no output 001 : output 0 when matched 010 : output 1 when matched 011 : output toggle when matched 100 : GRB input captured at rising edge 101 : GRB input captured at falling edge 110 : GRB input captured at both edge yyy : GRA function select 000 : compare match with no output 001 : output 0 when matched 010 : output 1 when matched 011 : output toggle when matched 100 : GRA input captured at rising edge 101 : GRA input captured at falling edge 110 : GRA input captured at both edge	R/W	8b'10001000

General Purpose Timer Unit Control Register(@0xFFFFF400) -- Continued

ABBREVIATION	ADDRESS	DESCRIPTIONS	R/W	INITIAL VALUE
TIER0	0xFFFFF428	Timer 0 Interrupt Enable Register 8b'11111xyz x,y,z = 0 : disable interrupt x,y,z = 1 : enable interrupt x = 1 : interrupt by Overflow y = 1 : interrupt by MCIB z = 1 : interrupt by MCIA	R/W	8b'11111000
TSTATUSR0	0xFFFFF42C	Timer 0 Status Register 8b'11111xyz x (OVFI)=1 Timer Counter Overflow/underflow occurred y(MCIB) =1 GRB input capture/compare match occurred z(MCIA) =1 GRA input capture/compare match occurred	R	8b'11111000
TCOUNT0	0xFFFFF430	Timer 0 Counter Register	R/W	16h'0000
GRA0	0xFFFFF434	Timer 0 General data A Register	R/W	16h'0000
GRB0	0xFFFFF438	Timer 0 General data B Register	R/W	16h'0000
TCONTR1	0xFFFFF440	Timer 1 Control Register	R/W	8b'10011000
TIOCR1	0xFFFFF444	Timer 1 I/O Control Register	R/W	8b'10001000
TIER1	0xFFFFF448	Timer 1 Interrupt Enable Register	R/W	8b'11111000
TSTATUSR1	0xFFFFF44C	Timer 1 Status Register	R	8b'11111000
TCOUNT1	0xFFFFF450	Timer 1 Counter Register	R/W	16h'0000
GRA1	0xFFFFF454	Timer 1 Data A Register	R/W	16h'0000
GRB1	0xFFFFF458	Timer 1 Data B Register	R/W	16h'0000
TCONTR2	0xFFFFF460	Timer 2 Control Register	R/W	8b'10011000
TIOCR2	0xFFFFF464	Timer 2 I/O Control Register	R/W	8b'10001000
TIER2	0xFFFFF468	Timer 2 Interrupt Enable Register	R/W	8b'11111000
TSTATUSR2	0xFFFFF46C	Timer 2 Status Register	R	8b'11111000
TCOUNT2	0xFFFFF470	Timer 2 Counter Register	R/W	16h'0000
GRA2	0xFFFFF474	Timer 2 Data A Register	R/W	16h'0000
GRB2	0xFFFFF478	Timer 2 Data B Register	R/W	16h'0000
TCONTR3	0xFFFFF480	Timer 3 Control Register	R/W	8b'10011000
TIOCR3	0xFFFFF484	Timer 3 I/O Control Register	R/W	8b'10001000
TIER3	0xFFFFF488	Timer 3 Interrupt Enable Register	R/W	8b'11111000
TSTATUSR3	0xFFFFF48C	Timer 3 Status Register	R	8b'11111000
TCOUNT3	0xFFFFF490	Timer 3 Counter Register	R/W	16h'0000
GRA3	0xFFFFF494	Timer 3 Data A Register	R/W	16h'0000
GRB3	0xFFFFF498	Timer 3 Data B Register	R/W	16h'0000
TCONTR4	0xFFFFF4A0	Timer 4 Control Register	R/W	8b'10011000
TIOCR4	0xFFFFF4A4	Timer 4 I/O Control Register	R/W	8b'10001000
TIER4	0xFFFFF4A8	Timer 4 Interrupt Enable Register	R/W	8b'11111000
TSTATUSR4	0xFFFFF4AC	Timer 4 Status Register	R	8b'11111000
TCOUNT4	0xFFFFF4B0	Timer 4 Counter Register	R/W	16h'0000
GRA4	0xFFFFF4B4	Timer 4 Data A Register	R/W	16h'0000
GRB4	0xFFFFF4B8	Timer 4 Data B Register	R/W	16h'0000

PIO Register (@0xFFFF FC00)

ABBREVIATION	ADDRESS	DESCRIPTIONS	R/W	INITIAL VALUE
PADR	0xFFFFFC00	Port A Data Register *At Power-On default, set as IRQ pins.* IRQ[5:0] when PINMUX_PA[5:0] = 0 PIOA[5:0] when PINMUX_PA[5:0] = 1 PA[7:6] is always set to PIO	R/W	8b'00000000
PADDR	0xFFFFFC04	Port A Direction control Register 1: Input Port 0: Output Port	R/W	8b'00000000
PBDR	0xFFFFFC08	Port B Data Register *At Power-On default, set as TimerOut pins.* Timer Out when PINMUX_PB[7:0] = 0 PIOB[7:0] when PINMUX_PB[7:0] = 1	R/W	8b'00000000
PBDDR	0xFFFFFC0C	Port B Direction control Register 1: Input Port 0: Output Port	R/W	8b'00000000
PCDR	0xFFFFFC10	Port C Data Register *At Power-On default, set as TimerIn pins.* Timer In when PINMUX_PC[7:0] = 0 PIOC[7:0] when PINMUX_PC[7:0] = 1	R/W	8b'00000000
PCDDR	0xFFFFFC14	Port C Direction control Register 1: Input Port 0: Output Port	R/W	8b'00000000
PDDR	0xFFFFFC18	Port D Data Register *At Power-On default, set as UART pins* NRI when PINMUX_PD[7] = 0 NDCD when PINMUX_PD[6] = 0 NDSR when PINMUX_PD[5] = 0 NCTS when PINMUX_PD[4] = 0 TXD1 when PINMUX_PD[3] = 0 RXD1 when PINMUX_PD[2] = 0 TXD0 when PINMUX_PD[1] = 0 RXD0 when PINMUX_PD[0] = 0 PIOD[7:0] when PINMUX_PD[7:0] = 1	R/W	8b'00000000
PDDDR	0xFFFFFC1C	Port D Direction control Register 1: Input Port 0: Output Port	R/W	8b'00000000
PEDR	0xFFFFFC20	Port E Data Register *At Power-On default, set as SM/UART pins* SCLK0 when PINMUX_PE[7] = 0 SOUT0 when PINMUX_PE[6] = 0 SIN0 when PINMUX_PE[5] = 0 SMCLK when PINMUX_PE[4] = 0 SMDO when PINMUX_PE[3] = 0 SMDI when PINMUX_PE[2] = 0 NRTS when PINMUX_PE[1] = 0 NDTR when PINMUX_PE[0] = 0 PIOE[7:0] when PINMUX_PE[7:0] = 1 PIOE[7:0] when PINMUX_PE[7:0] = 1	R/W	8b'00000000
PEDDR	0xFFFFFC24	Port E Direction control Register 1: Input Port 0: Output Port	R/W	8b'00000000

PIO Register (@0xFFFF FC00) -- Continued

ABBREVIATION	ADDRESS	DESCRIPTIONS	R/W	INITIAL VALUE
PFDR	0xFFFFFC28	Port F Data Register *At Power-On default, set as SSPI pins* NIRQOUT when PINMUX_PF[7] = 0 NFIQOUT when PINMUX_PF[6] = 0 BCLKOUT when PINMUX_PF[5] = 0 SCS1 when PINMUX_PF[4] = 0 SCLK1 when PINMUX_PF[3] = 0 SOUT1 when PINMUX_PF[2] = 0 SIN1 when PINMUX_PF[1] = 0 SCS0 when PINMUX_PF[0] = 0 PIOE[7:0] when PINMUX_PF[7:0] = 1 PIOE[7:0] when PINMUX_PF[7:0] = 1 PIOF[7:0] when PINMUX_PF[7:0] = 1	R/W	8b'00000000
PFDDR	0xFFFFFC2C	Port F Direction control Register 1: Input Port 0: Output Port	R/W	8b'00000000
PGDR	0xFFFFFC30	Port G Data Register *At Power-On default, set as DMA pins* RAS1 when PINMUX_PG[7] = 0 RAS0 when PINMUX_PG[6] = 0 DACK1 when PINMUX_PG[5] = 0 DREQ0 when PINMUX_PG[4] = 0 SMDO when PINMUX_PG[3] = 0 SMDI when PINMUX_PG[2] = 0 NRTS when PINMUX_PG[1] = 0 NDTR when PINMUX_PG[0] = 0 PIOG[7:0] when PINMUX_PG[7:0] = 1	R/W	8b'00000000
PGDDR	0xFFFFFC34	Port G Direction control Register 1: Input Port 0: Output Port	R/W	8b'00000000
PHDR	0xFFFFFC38	Port H Data Register *At Power-On default, set as DRAM pins* CAS[3:0] when PINMUX_PH[3:0] = 0 PIOH[3:0] when PINMUX_PH[3:0] = 1 CS[7:4] when PINMUX_PH[7:4] = 0 PIOH[7:4] when PINMUX_PH[7:4] = 1	R/W	8b'00000000
PHDDR	0xFFFFFC3C	Port H Direction control Register 1: Input Port 0: Output Port	R/W	8b'00000000
PIDR	0xFFFFFC40	Port I Data Register *At Power-On default, set as DBUS pins* D[23:16] when PINMUX_PI[7:0] = 0 PIOI[7:0] when PINMUX_PI[7:0] = 1	R/W	8b'00000000
PIDDR	0xFFFFFC44	Port I Direction control Register 1: Input Port 0: Output Port	R/W	8b'00000000
PJDR	0xFFFFFC48	Port J Data Register *At Power-On default, set as DBUS pins* D[31:24] when PINMUX_PJ[7:0] = 0 PIOJ[7:0] when PINMUX_PJ[7:0] = 1	R/W	8b'00000000
PJDDR	0xFFFFFC4C	Port J Direction control Register 1: Input Port 0: Output Port	R/W	8b'00000000

SSPI Register (@ 0xFFFF F800)

ABBREVIATION	ADDRESS	DESCRIPTIONS	R/W	INITIAL VALUE
SSCR0A	0xFFFFF800	SSPI 0 Control Register A Bit 7 : Enters Test mode when set to 0 Bit 6 : CS is Active High when set to 0 Bit 5 : SSPI 0 Disabled when set to 0 Bit 4 : Slave mode when set to 0 Bit 3 : LSB input first when set to 0 Bit 2 : LSB out first when set to 0 Bit 1 : Rising edge clock when set to 0 Bit 0 : Slave: CS disabled when set to 0	R/W	8b'11111111 1: Normal 1: Active Low 1: Enabled 1: Master 1: MSB in 1: MSB out 1: Falling 1: CS enabled
SSCR0B	0xFFFFF804	SSPI 0 Control Register B Bit 7 : Tx interrupt enabled when set to 1 Bit 6 : Tx FIFO empty interrupt enable Bit 5 : Rx FIFO full interrupt enable Bit 4 : Tx FIFO full interrupt enable Bit 3 : Rx FIFO enabled when set to 1 Bit 2 : Tx FIFO enabled when set to 1 Bit 1 : Rising edge clock when set to 0 Bit 0 : Slave: CS disabled when set to 0	R/W	8b'00000000 0: Disable 0: Disable 0: Enabled 0: Master 0: MSB in 0: MSB out 0: Falling 0: CS enabled
SSDR0	0xFFFFF808	SSPI 0 Data Register	R/W	8b'11111111
SSSR0	0xFFFFF80C	SSPI 0 Status Register Bit 7 : Rx FIFO Empty Bit 6 : Tx FIFO Empty Bit 5 : Rx FIFO FULL Bit 4 : Tx FIFO FULL Bit 3 : Tx END Bit 2 : Reserved Bit 1 : Reserved Bit 0 : SSPI BUSY	R	8b'00000000
SSTR0	0xFFFFF810	SSPI 0 Term Register	W	8b'11111111
SSCR1A	0xFFFFF820	SSPI 1 Control Register A Bit 7 : Enters Test mode when set to 0 Bit 6 : CS is Active High when set to 0 Bit 5 : SSPI 0 Disabled when set to 0 Bit 4 : Slave mode when set to 0 Bit 3 : LSB input first when set to 0 Bit 2 : LSB out first when set to 0 Bit 1 : Rising edge clock when set to 0 Bit 0 : Slave: CS disabled when set to 0	R/W	8b'11111111 1: Normal 1: Active Low 1: Enabled 1: Master 1: MSB in 1: MSB out 1: Falling 1: CS enabled
SSCR1B	0xFFFFF824	SSPI 1 Control Register B Bit 7 : Tx interrupt enabled when set to 1 Bit 6 : Tx FIFO empty interrupt enable Bit 5 : Rx FIFO full interrupt enable Bit 4 : Tx FIFO full interrupt enable Bit 3 : Rx FIFO enabled when set to 1 Bit 2 : Tx FIFO enabled when set to 1 Bit 1 : Rising edge clock when set to 0 Bit 0 : Slave: CS disabled when set to 0	R/W	8b'00000000 0: Disable 0: Disable 0: Enabled 0: Master 0: MSB in 0: MSB out 0: Falling 0: CS enabled

SSPI Register (@ 0xFFFF F800) -- Continued

ABBREVIATION	ADDRESS	DESCRIPTIONS	R/W	INITIAL VALUE
SSDR1	0xFFFFF828	SSPI 1 Data Register	R/W	8b'11111111
SSSR1	0xFFFFF82C	SSPI 1 Status Register Bit 7 : Rx FIFO Empty Bit 6 : Tx FIFO Empty Bit 5 : Rx FIFO FULL Bit 4 : Tx FIFO FULL Bit 3 : Tx END Bit 2 : Reserved Bit 1 : Reserved Bit 0 : SSPI BUSY	R	8b'00000000
SSTR1	0xFFFFF830	SSPI 0 Term Register	W	8b'11111111

UART Register (@0xFFFF F500)

ABBREVIATION	ADDRESS	DESCRIPTIONS	R/W	INITIAL VALUE
RBR0	0xFFFFF500	Receiver Buffer Register (DLAB=0)	R	8b'10011000
THR		Transmitter Holding Register (DLAB=0)	W	8b'10011000
DLL		Divisor Latch LS (DLAB=1)	R/W	8b'10011000
DLM		Divisor Latch MS (DLAB=1)	R/W	8b'10011000
IER	0xFFFFF504	Interrupt Enable Register Bit 7-4 : Reserved Bit 3 : Modem status interrupt Bit 2 : Receiver line status interrupt Bit 1 : THR empty interrupt Bit 0 : Rx data available interrupt	R/W	8b'00000000
IIR / FCR	0xFFFFF508	Interrupt Identification Register FIFO Control Register Bit 7 : RCVR Trigger to MSB Bit 6 : RCVR Trigger to LSB Bit 5-3 : Reserved Bit 2 : XMIT FIFO Reset Bit 1 : RCVR FIFO Reset Bit 0 : FIFO Enable	R W	8b'00000001
LCR	0xFFFFF50C	Line Control Register Bit 7 : DLAB (Divisor Latch Access Bit) Bit 6 : Break control bit Bit 5 : Stick parity bit Bit 4 : Even parity control Bit 3 : Parity Control (0: Disabled) Bit 2 : Stop bit(s) (0: Disabled) Bit 1,0 : character bits 00(5-bit),01(6-bit), 10(7-bit), 11(8-bit)	R/W	8b'00000000

UART Register (@0xFFFF F500) -- Continued

ABBREVIATION	ADDRESS	DESCRIPTIONS	R/W	INITIAL VALUE
MCR	0xFFFFF510	MODEM Control Register Bit 7-5 : Reserved Bit 4 : Loop control Bit 3,2 : Reserved Bit 1 : RTS Bit 0 : DTR	R/W	8b'00000000
LSR	0xFFFFF514	Line Status Register Bit 7 : Rx FIFO Error Bit 6 : Transmitter Empty Bit 5 : THR Empty Bit 4 : Break Interrupt Bit 3 : Framming Error Bit 2 : Parity Error Bit 1 : Overrun Error Bit 0 : Data Ready	R/W	8b'01100000
MSR	0xFFFFF518	MODEM Status Register Bit 7 : DCD(Data Carrier Detect) Bit 6 : RI(Ring Indicator) Bit 5 : DSR(Data Set Ready) Bit 4 : CTS(Clear To Send) Bit 3 : DDCD(Delta Data Carrier Detec) Bit 2 : TERI(Trailing Edge Ring Indi.) Bit 1 : DDSR(Delta Data Set Ready) Bit 0 : DCTS(Data Clear To Send)	R/W	8b'xxxx0000
SCR	0xFFFFF51C	Scratch Register	R/W	8b'10011000

I²C Registers (@ 0xFFFF F900 : channel 0)

ABBREVIATION	ADDRESS	DESCRIPTIONS	R/W	INITIAL VALUE
Baud_r0	0xFFFFF900	I²C Baud rate register for channel 0	W	8b'00000100
CTRL0	0xFFFFF904	I²C Control Register for channel 0 Bit 7-5 : Reserved Bit 4 : Include address bit Bit 3 : Transfer END Bit 2 : Transfer START Bit 1 : SDA pin set/clear Bit 0 : Interrupt Enable (0: Disabled)	W	8b'xxx00010
Data_r0	0xFFFFF908	I²C Data Register for channel 0	R/W	8b'xxxxxxxx
Stat_r0	0xFFFFF90C	I²C Status Register for channel 0 Bit 7-5 : Reserved Bit 4 : Interrupt Flag bit Bit 3 : Bus Busy Flag bit Bit 2 : Bus Lost Flag bit Bit 1 : SDA pin Ack bit Bit 0 : Slave Called Flag bit	R	8b'xxx00000
Addr_r0	0xFFFFF910	I²C Address Register for channel 0	W	8b'00000000
Test_r0	0xFFFFF914	Test Register for channel 0	R/W	8b'00000000

I²C Registers (@0xFFFF FA000 : channel 1)

ABBREVIATION	ADDRESS	DESCRIPTIONS	R/W	INITIAL VALUE
Baud_r1	0xFFFFFA00	I²C Baud rate register for channel 1	W	8b'10011000
CTRL1	0xFFFFFA04	I²C Control Register for channel 1	W	8b'10011000
Data_r1	0xFFFFFA08	I²C Data Register for channel 1	R/W	8b'10011000
Stat_r1	0xFFFFFA0C	I²C Status Register for channel 1	R	8b'10011000
Addr_r1	0xFFFFFA10	I²C Address Register for channel 1	W	8b'10011000
Test_r1	0xFFFFFA14	Test Register for channel 1	R/W	8b'10011000

I²C Registers (@ 0xFFFF FB00 : channel 2)

ABBREVIATION	ADDRESS	DESCRIPTIONS	R/W	INITIAL VALUE
Baud_r2	0xFFFFFB00	I²C Baud rate register for channel 2	W	8b'10011000
CTRL2	0xFFFFFB04	I²C Control Register for channel 2	W	8b'10011000
Data_r2	0xFFFFFB08	I²C Data Register for channel 2	R/W	8b'10011000
Stat_r2	0xFFFFFB0C	I²C Status Register for channel 2	R	8b'10011000
Addr_r2	0xFFFFFB10	I²C Address Register for channel 2	W	8b'10011000
Test_r2	0xFFFFFB14	Test Register for channel 2	R/W	8b'10011000