

---

# **HMS39C7092**

---

# **Embedded Flash MCU**

**Specification Ver 1.0**

**System IC SBU, SP BU**

**MCU Business Division,**

**Flash Team**



Released : February. 2001

ARM® is trademark of Advanced RISC Machine Ltd.  
ARM7TDMI is designed by ARM Ltd.

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 Hynix 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 Hynix or others.

These Hynix 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 Hynix 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.). Hynix cannot accept liability to any damage which may occur in case these Hynix products were used in the mentioned equipment without prior consultation with Hynix.

Copyright 2000 Hynix Semiconductor, Inc.  
All Rights Reserved

## Contents

Chapter 1.....	13
Introduction.....	13
1.1 General Description.....	14
1.2 Feature.....	15
1.3 Pin Descriptions .....	16
1.4 Operation Mode description .....	21
1.5 Memory Map.....	25
Chapter 2.....	27
ARM7TDMI Core .....	27
2.1 General Description.....	28
2.2 Feature.....	28
2.3 Core Block Diagram .....	29
2.4 Instruction Set .....	30
2.4.1 ARM Instruction .....	30
2.4.2 THUMB Instruction.....	33
2.4.3 The Program Status Registers .....	36
2.4.3.1 The condition code flags .....	37
2.4.3.2 The control bits.....	37
2.4.4 ARM pseudo-instructions .....	39
2.4.5 THUMB pseudo-instructions .....	43
Chapter 3.....	47
BUS Controller .....	47
3.1 Overview .....	48
3.1.1 Features .....	48
3.1.2 Pin Configuration.....	49
3.2 Bus Controller Registers .....	50
3.2.1 Configuration Registers .....	51
3.3 Operation .....	52
3.3.1 Area Division .....	52
3.3.2 Area Division .....	53
3.3.3 Chip Select Signals .....	53
3.4 Basic Bus Interface .....	54
3.4.1 Overview .....	54
3.4.2 Byte Lane Write Control .....	54
3.4.3 Basic Bus Control Signal Timing.....	56
3.4.4 Wait Control.....	62
3.4.5 Bus Arbiter.....	63
Chapter 4.....	65
MCU Controller .....	65
4.1 General Description.....	66
4.2 Pin Function Description.....	66
4.3 Register Description .....	67
4.3.1 Register Memory Map .....	67
4.3.2 PINMUX Register .....	68
4.3.3 MCU Device Code Register (0x0900_002C Read Only) .....	72

Chapter 5.....	73
Power Management Unit.....	73
5.1    General Description.....	74
5.2    Operation Modes.....	75
5.2.1  Introduction.....	75
5.2.2  Reset and Operation Modes .....	75
5.3    Power Management Unit Register Map.....	77
5.4    Register Description .....	78
5.5    Signal Timing Diagram.....	81
5.5.1  Power on Reset .....	81
5.5.2  Watch Dog Timer Overflow .....	81
5.5.3  Soft-Reset .....	82
Chapter 6.....	83
The Interrupt Controller.....	83
6.1    About the Interrupt controller .....	84
6.1.1  Interrupt sources .....	85
6.1.2  Interrupt Control .....	85
6.2    Interrupt Controller Registers .....	87
Chapter 7.....	91
Watchdog Timer.....	91
7.1    General Description.....	92
7.2    Watchdog Timer Introduction.....	93
7.3    Watchdog Timer Operation .....	94
7.3.1  Timing of Setting and Clearing the Overflow Flag .....	95
7.4    Watchdog Timer Memory Map.....	96
7.5    Watchdog Timer Register Descriptions .....	97
7.6    Examples of Register Setting .....	100
7.6.1  Interval Timer Mode.....	100
7.6.2  Watchdog Timer Mode with Internal Reset Disable .....	101
7.6.3  Watchdog Timer Mode with Power-on Reset .....	102
7.6.4  Watchdog Timer Mode with Manual Reset .....	103
Chapter 8.....	105
The General Purpose Timer.....	105
8.1    About the General Purpose Timer Unit.....	106
8.1.1  General Purpose Timer Unit Introduction .....	107
8.2    General Purpose Timer Unit Memory Map .....	108
8.2.1  Register Assignment .....	108
8.2.2  General Purpose Timer Unit Register Descriptions .....	109
8.2.2.1  Timer Global Control Registers .....	109
8.2.2.2  Timer Channel Control Registers .....	110
8.3    General Purpose Timer Unit Operation .....	114
8.3.1  Free Running Mode.....	115
8.3.2  Compare Match Mode .....	117
8.3.3  Input Capture Mode.....	119
8.3.4  Synchronized Clear and Write Mode.....	120
8.3.5  PWM Mode.....	121
8.3.5.1  PWM Mode Operation .....	121
Chapter 9.....	125
UART (Universal Asynchronous Receiver/Transmitter).....	125

## Flash MCU(HMS39C7092)

---

9.1	General Description.....	126
9.2	Features .....	127
9.3	Signal Description .....	127
9.4	Internal Block Diagram .....	128
9.5	Registers Description .....	129
9.6	UART Operations .....	140
9.6.1	FIFO Interrupt Mode Operation .....	140
9.6.2	FIFO Polled Mode Operation .....	141
9.7	Register Summary.....	142
Chapter 10	.....	143
	GPIO (General Purpose Input Output) .....	143
10.1	General Description.....	144
10.2	GPIO Registers .....	145
10.2.1	Register Memory Map.....	145
10.3.1	Register Description .....	146
10.3	Functional Description .....	147
Chapter 11	.....	149
	On-Chip SRAM .....	149
11.1	General Description.....	150
11.2	Function Description.....	150
Chapter 12	.....	151
	On-chip Flash Memory .....	151
12.1	General Description.....	152
12.2	Features .....	152
12.3	Block Diagram .....	154
12.4	Flash Memory Register Description .....	156
12.5	On-Board Programming Mode .....	161
12.5.1	Boot Mode .....	161
12.5.2	User Program Mode .....	164
12.6	Flash Memory Programming/Erasing .....	166
12.6.1	Program & Program-Verify Mode .....	166
12.6.2	Pre-program & Pre-program Verify Mode .....	168
12.6.3	Erase & Erase Verify Mode .....	170
12.6.4	Erase Algorithm .....	172
12.7	Flash Memory PROM Mode .....	173
12.7.1	PROM Mode Setting .....	173
12.7.2	Memory Map .....	174
12.7.3	PROM Mode Operation.....	174
12.7.4	Timing Diagram and AC/DC Characteristics .....	175
Chapter 13	.....	179
	A/D Converter.....	179
13.1	Overview .....	180
13.1.1	Features .....	180
13.1.2	Pin Configuration .....	181
13.2	A/D Converter Registers .....	182
13.2.1	Register Descriptions .....	182
13.3	Operation .....	185
13.4	Interrupts.....	186
13.5	Usage Notes .....	187

13.6	Example .....	190
Chapter 14	.....	191
Electrical Characteristics	.....	191
14.1	Absolute Maximum Ratings.....	192
14.2	Recommended Operating Conditions: .....	192
14.3	DC Characteristics .....	193
14.4	AC Characteristics.....	194
14.4	AD Conversion characteristics (Preliminary).....	196
14.5	Operational Timing .....	197
14.5.1	Clock Timing .....	197
14.5.2	Reset Timing .....	197
14.5.3	Bus Timing .....	198
Appendix		
A-1	Peripheral Setting & Flash memory control Examples	
A-2	Package Dimension	

## Figures

Figure 1.1	Package Outline.....	14
Figure 1.2	HMS39C7092 Block Diagram .....	15
Figure 1.3	HMS39C7092 Memory Map.....	25
Figure 1.4	Memory Map of Mode 3.....	25
Figure 1.5	Memory Map of when Mode 4 and Mode 5 .....	26
Figure 1.6	Memory Map of Mode 6 and Mode 7 .....	26
Figure 2.1	ARM7TDMI Core Block Diagram .....	29
Figure 2.2	ARM instruction set formats .....	30
Figure 2.3	Register Organization in ARM state.....	32
Figure 2.4	THUMB instruction set formats.....	33
Figure 2.5	Register Organization in THUMB state .....	35
Figure 2.6	Mapping of THUMB state registers onto ARM state registers. ....	35
Figure 2.7	Program status register format .....	36
Figure 3.1	Block Diagram of the Bus Controller .....	48
Figure 3.2	Access Area Map for Each Operating Mode.....	52
Figure 3.3	Access Size and Data Alignment Control (8-Bit Access Area) .....	54
Figure 3.4	Access Size and Data Alignment Control (16-Bit Access Area) .....	55
Figure 3.5	Bus Control Signal Write Timing for 16-Bit, 1-Wait (Word Access) .....	56
Figure 3.6	Bus Control Signal Read Timing for 16-Bit, 1-Wait (Word Access).....	56
Figure 3.7	Bus Control Signal Write Timing for 16-Bit, 1-Wait (Half-word Access).....	57
Figure 3.8	Bus Control Signal Read Timing for 16-Bit, 1-Wait (Half-word Access).....	57
Figure 3.9	Bus Control Signal Write Timing for 16-Bit, 1-Wait (Byte Access).....	58
Figure 3.10	Bus Control Signal Read Timing for 16-Bit, 1-Wait (Byte Access).....	58
Figure 3.11	Bus Control Signal Write Timing for 16-Bit, 2-Wait (Word Access).....	59
Figure 3.12	Bus Control Signal Read Timing for 16-Bit, 2-Wait (Word Access) .....	59
Figure 3.13	Bus Control Signal Write Timing for 16-Bit, 2-Wait (Half-Word Access).....	60
Figure 3.14	Bus Control Signal Read Timing for 16-Bit, 2-Wait (Half-Word Access) .....	60
Figure 3.15	Bus Control Signal Write Timing for 16-Bit, 2-Wait (Byte Access).....	61
Figure 3.16	Bus Control Signal Read Timing for 16-Bit, 2-Wait (Byte Access).....	61
Figure 3.17	Example of Wait State Insertion Timing. ....	62
Figure 3.18	Example of External Bus Master Operation.....	64
Figure 5.1	PMU Block Diagram .....	74
Figure 5.2	Reset and Power Management State Machine. ....	76
Figure 5.3	Power on Reset Timing Diagram .....	81
Figure 5.4	Watch Dog Timer Overflow Timing Diagram .....	81
Figure 5.5	Soft Reset (from WDT) Timing Diagram .....	82
Figure 5.6	Soft Reset (from PMU) Timing Diagram .....	82
Figure 6.1	Interrupt Control Flow Diagram .....	84
Figure 7.1	Watchdog Timer Module Block Diagram .....	92
Figure 7.2	Operation in the Watchdog Timer Mode .....	94
Figure 7.3	Operation in the Interval Timer Mode .....	95
Figure 7.4	Interrupt Clear in the Interval Timer Mode .....	100
Figure 7.5	Interrupt Clear in the Watchdog Timer Mode with Reset Disable.....	101
Figure 7.6	Interrupt Clear in the Watchdog Timer Mode with Power-on Reset.....	102
Figure 7.7	Interrupt Clear in the Watchdog Timer Mode with Manual Reset .....	103
Figure 8.1	General-purpose Timer Unit Module Block Diagram .....	106

Figure 8.2	Free-Running Counter Operation .....	115
Figure 8.3	Periodic Counter Operation.....	116
Figure 8.4	Example of 0 Output/1 Output .....	117
Figure 8.5	Example of Toggle Output.....	118
Figure 8.6	Compare Match Signal Output Timing .....	118
Figure 8.7	Input Capture Operation .....	119
Figure 8.8	Synchronized Operation Example .....	120
Figure 8.9	PWM Mode Operation Example 1 .....	121
Figure 8.10	PWM Mode Operation Example 2.....	122
Figure 8.11	Reset-Synchronized PWM Mode Operation Example.....	123
Figure 9.1	TOP BLOCK Diagram .....	126
Figure 9.2	Internal UART Diagram .....	128
Figure 10.1	GPIO Block Diagram and PADS Connections(example for Port A and Port B) .....	144
Figure 12.1	Block Diagram of Flash Memory .....	154
Figure 12.2	System Configuration When Using On-Board Boot Mode.....	161
Figure 12.3	Boot Mode Execution Procedure .....	162
Figure 12.4	User Mode Execution Procedure.....	164
Figure 12.5	Flash Program & Program Verify Sequence.....	167
Figure 12.6	Flash Pre-program & Pre-program Verify Sequence .....	169
Figure 12.7	Flash Erase & Erase Verify Sequence .....	171
Figure 12.8	Flash Erase Algorithm .....	172
Figure 12.9	Timing Diagram of Read.....	175
Figure 12.10	Timing Diagram of Pre-Program/Program .....	176
Figure 12.11	Timing Diagram of Erase.....	176
Figure 12.12	Timing Diagram of Pre-Program/Program Verify .....	177
Figure 12.13	Timing Diagram of Erase Verify .....	177
Figure 13.1	Block Diagram of A/D Converter.....	180
Figure 13.2	A/D converter Operation .....	185
Figure 13.3	Example of Analog Input Circuit .....	188
Figure 13.4	A/D Converter Accuracy Definitions (1).....	188
Figure 13.5	A/D Converter Accuracy Definitions (2).....	189
Figure 14.1	The settling time of the crystal oscillator.....	197
Figure 14.2	Reset Input Timing .....	197
Figure 14.3	The Write Timing Diagram of the Bus Controller .....	198
Figure 14.4	The Read Timing Diagram of the Bus Controller .....	198
Figure 14.5	Basic Bus Cycle with External Wait State.....	199
Figure 14.6	Bus Release Mode Timing .....	199



## Tables

Table 1.1	Pin Descriptions .....	16
Table 1.1	Pin Descriptions (Continued) .....	17
Table 1.1	Pin Descriptions (Continued) .....	18
Table 1.1	Pin Descriptions (Continued) .....	19
Table 1.1	Pin Descriptions (Continued) .....	20
Table 1.2	HMS39C7092 Operation modes .....	21
Table 1.3	Pin assignment by mode.....	22
Table 1.3	Pin assignment by mode (continued) .....	23
Table 1.3	Pin assignment by mode (continued) .....	24
Table 2.1	The ARM Instruction set.....	31
Table 2.2	THUMB instruction set opcodes .....	34
Table 2.3	Condition code summary .....	36
Table 2.4	PSR mode bit values .....	38
Table 3.1	Bus Controller Pins .....	49
Table 3.2	BUS Controller Register Map.....	50
Table 3.3	Byte Lane condition by XA[0].....	55
Table 4.1	Pin Function Descriptions .....	66
Table 4.2	Memory map of the MCU Controller .....	67
Table 4.3	MCU Controller Initial values in each mode .....	67
Table 5.1	Register Map of the PMU .....	77
Table 6.1	Interrupt Controller Default Setting Value.....	85
Table 6.2	Memory Map of the Interrupt Controller .....	87
Table 6.3	Interrupt Source Trigger Mode.....	88
Table 7.1	Memory Map of the Watchdog Timer APB Peripheral .....	96
Table 7.2	Internal Counter Clock Sources (SYSCLK = 40 MHz).....	98
Table 8.1	Timer Global Control Register Map .....	108
Table 8.2	Timer Channel Control Register Map .....	108
Table 8.3	Timer Channel Starting Address .....	108
Table 9.1	Signal Descriptions .....	127
Table 9.2	UART Register Address Map (0x1500 in UART1) .....	129
Table 9.3	UART Register Reset Values .....	129
Table 9.4a	Divisor Values for each Baud rate (CLK=1.8432MHz).....	133
Table 9.4b	Divisor Values for each Baud rate (CLK=3.6864MHz).....	133
Table 9.5	Interrupt Control Functions .....	138
Table 9.6	Summary of Registers.....	142
Table 10.1	GPIO Register Memory Map.....	145
Table 12.1	Operating mode.....	153
Table 12.2	Signal description of Figure 12.1(BUS Interface) .....	155
Table 12.3	Flash Memory Registers.....	156
Table 12.4	Control Register.....	158
Table 12.5	Erase Block Register.....	159
Table 12.6	Status & Power Register .....	160
Table 12.7	FR_SEL Value for access to internal Register .....	173
Table 12.8	Setting for Register read/write.....	173
Table 12.9	Erase Block Register.....	174
Table 12.10	Setting for Flash PROM read/write .....	175

Table 12.11	DC Characteristics .....	178
Table 12.12	AC Characteristics .....	178
Table 13.1	A/D Converter Pins .....	181
Table 13.2	Summarizes the A/D converter' s registers.....	182
Table 14.1	Absolute Maximum Ratings	- Preliminary -..... 192
Table 14.2	Recommended Operating Conditions	- Preliminary- ..... 192
Table 14.3	DC Characteristics	- Preliminary- ..... 193
Table 14.4	IO Circuits with pull-ups	- Preliminary- ..... 193
Table 14.5	IO Circuits with pull-downs	- Preliminary- ..... 193
Table 14.6	Clock Timing	- Preliminary-..... 194
Table 14.7	Control Signal Timing	- Preliminary- ..... 194
Table 14.8	Bus Timing	- Preliminary- ..... 195
Table 14.9	Operating Conditions of the AD Conversion	- Preliminary-..... 196
Table 14.10	Electrical characteristics of the AD converter	- Preliminary-..... 196

## Flash MCU(HMS39C7092)

---



Chapter 1  
**Introduction**

1.1 General Description

The 16bit MCU with embedded flash memory for optical storage is the first member of Hynix Micro Electronics 16/32bit MCU Family of high performance microcontroller units (MCUs). This family includes a series of peripherals from which numerous MCUs are assembled. This MCU contains extensive peripherals : 192Kbytes flash memory, 4K bytes SRAM, 6 channel 16bit Timer, Watch Dog Timer, 2 channel UART, Programmable Priority Interrupt Controller, 81bits PIO, BUS Controller including Chip select logic, which is On-Chip Modular Architecture (using AMBA).

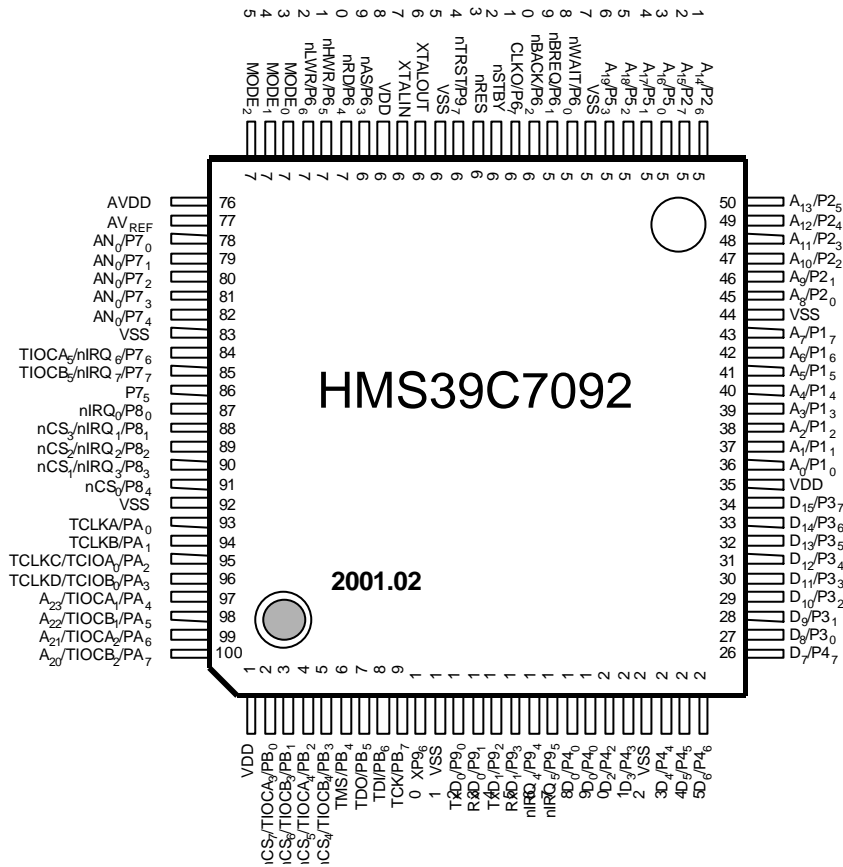


Figure 1.1 Package Outline

## 1.2 Feature

- On-Chip Modular Architecture (using AMBA)
- Utilizes the ARM7TDMI 32/16bit RISC Family
- 192Kbyte flash memory
- 4Kbyte internal SRAM
- 8/16-bit external Data Bus
- Eight Programmable Chip Select Outputs with external wait input
- Low Power Consumption using Power Management Unit
- Fully static operation : Max. 50MHz
- Programmable Priority Interrupt Controller (8 external sources)
- Six 16bit Multi Function Timers/Counters for General Purpose Applications
- One 8bit Watch Dog Timer (WDT)
- Two UARTs (Universal Asynchronous Receiver Transmitter) compatible with 16C550 UART
- Programmable Input/Output ports (81-bits)
- 100 TQFP Package

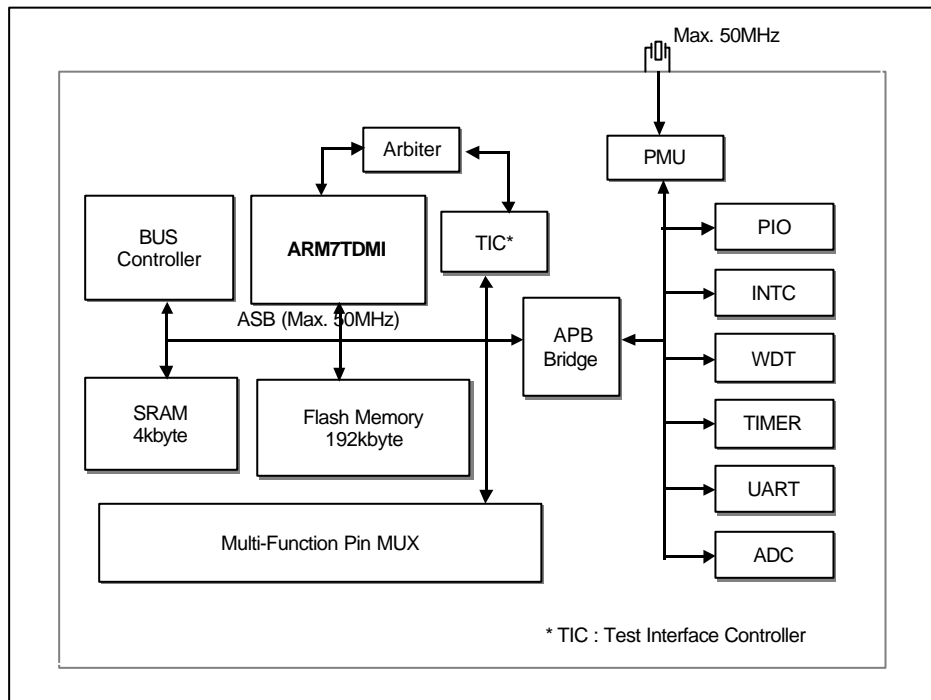


Figure 1.2 HMS39C7092 Block Diagram

### 1.3 Pin Descriptions

**Table 1.1 Pin Descriptions**

PIN	SYMBOL	DIR	DESCRIPTION
1	VDD	-	Power Supply 3.3V
	nCS <sub>7</sub>	O	External Chip Selection Number 7
2	TCIOA <sub>3</sub>	I/O	PWM output, Compare match output of Reg.A and signal capture input of Timer Ch3
	PB <sub>0</sub>	I/O	General purpose input output of port B bit0
3	nCS <sub>6</sub>	O	External Chip Selection Number 6
	TCIOB <sub>3</sub>	I/O	PWM output, Compare match output of Reg.B and signal capture input of Timer Ch3
	PB <sub>1</sub>	I/O	General purpose input output of port B bit 1
4	nCS <sub>5</sub>	O	External Chip Selection Number 5
	TIOCA <sub>4</sub>	I/O	PWM output, Compare match output of Reg.A and signal capture input of Timer Ch4
	PB <sub>2</sub>	I/O	General purpose input output of port B bit2
5	nCS <sub>4</sub>	O	External Chip Selection Number 4
	TIOCB <sub>4</sub>	I/O	PWM output, Compare match output of Reg.B and signal capture input of Timer Ch4
	PB <sub>3</sub>	I/O	General purpose input output of port B bit3
6	TMS	I	JTAG Test Mode Selection
	PB <sub>4</sub>	I/O	General purpose input output of port B bit4
7	TDO	O	JTAG Test Data Output
	PB <sub>5</sub>	I/O	General purpose input output of port B bit5
8	TDI	I	JTAG Test Data Input
	PB <sub>6</sub>	I/O	General purpose input output of port B bit6
9	TCK	I	JTAG Test Clock
	PB <sub>7</sub>	I/O	General purpose input output of port B bit7
10	TVPPD	I	5Vinput for the use of Programming and Erasing of the Flash Memory
11	VSS	-	Power ground
12	TxD <sub>0</sub>	O	Transmit Data of UART Ch0
	P9 <sub>0</sub>	I/O	General purpose input output of port 9 bit 0
13	RxD <sub>0</sub>	O	Receive Data of UART Ch0
	P9 <sub>1</sub>	I/O	General purpose input output of port 9 bit 1
14	TxD <sub>1</sub>	O	Transmit Data of UART Ch1
	P9 <sub>2</sub>	I/O	General purpose input output of port 9 bit 2
15	RxD <sub>1</sub>	O	Receive Data of UART Ch1
	P9 <sub>3</sub>	I/O	General purpose input output of port 9 bit 3
16	nIRQ <sub>4</sub>	I	External Interrupt Request number 4
	P9 <sub>4</sub>	I/O	General purpose input output of port 9 bit 4
17	nIRQ <sub>5</sub>	I	External Interrupt Request number 5
	P9 <sub>5</sub>	I/O	General purpose input output of port 9 bit 5
18	D <sub>0</sub>	I/O	External Data Bus bit 0
	P4 <sub>0</sub>	I/O	General purpose input output or port 4 bit 0
19	D <sub>1</sub>	I/O	External Data Bus bit 1
	P4 <sub>1</sub>	I/O	General purpose input output or port 4 bit 1
20	D <sub>2</sub>	I/O	External Data Bus bit 2
	P4 <sub>2</sub>	I/O	General purpose input output or port 4 bit 2
21	D <sub>3</sub>	I/O	External Data Bus bit 3
	P4 <sub>3</sub>	I/O	General purpose input output or port 4 bit 3



Table 1.1 Pin Descriptions (Continued)

PIN	SYMBOL	DIR	DESCRIPTION
22	VSS	-	Power ground
23	D <sub>4</sub>	I/O	External Data Bus bit 4
	P4 <sub>4</sub>	I/O	General purpose input output or port 4 bit 4
24	D <sub>5</sub>	I/O	External Data Bus bit 5
	P4 <sub>5</sub>	I/O	General purpose input output or port 4 bit 5
25	D <sub>6</sub>	I/O	External Data Bus bit 6
	P4 <sub>6</sub>	I/O	General purpose input output or port 4 bit 6
26	D <sub>7</sub>	I/O	External Data Bus bit 7
	P4 <sub>7</sub>	I/O	General purpose input output or port 4 bit 7
27	D <sub>8</sub>	I/O	External Data Bus bit 8
	P3 <sub>0</sub>	I/O	General purpose input output or port 3 bit 0
28	D <sub>9</sub>	I/O	External Data Bus bit 9
	P3 <sub>1</sub>	I/O	General purpose input output or port 3 bit 1
29	D <sub>10</sub>	I/O	External Data Bus bit 10
	P3 <sub>2</sub>	I/O	General purpose input output or port 3 bit 2
30	D <sub>11</sub>	I/O	External Data Bus bit 11
	P3 <sub>3</sub>	I/O	General purpose input output or port 3 bit 3
31	D <sub>12</sub>	I/O	External Data Bus bit 12
	P3 <sub>4</sub>	I/O	General purpose input output or port 3 bit 4
32	D <sub>13</sub>	I/O	External Data Bus bit 13
	P3 <sub>5</sub>	I/O	General purpose input output or port 3 bit 5
33	D <sub>14</sub>	I/O	External Data Bus bit 14
	P3 <sub>6</sub>	I/O	General purpose input output or port 3 bit 6
34	D <sub>15</sub>	I/O	External Data Bus bit 15
	P3 <sub>7</sub>	I/O	General purpose input output or port 3 bit 7
35	VDD	-	Power Supply 3.3V
36	A <sub>0</sub>	O	External Address Bus bit 0
	P1 <sub>0</sub>	I/O	General purpose input output or port 1 bit 0
37	A <sub>1</sub>	O	External Address Bus bit 1
	P1 <sub>1</sub>	I/O	General purpose input output or port 1 bit 1
38	A <sub>2</sub>	O	External Address Bus bit 2
	P1 <sub>2</sub>	I/O	General purpose input output or port 1 bit 2
39	A <sub>3</sub>	O	External Address Bus bit 3
	P1 <sub>3</sub>	I/O	General purpose input output or port 1 bit 3
40	A <sub>4</sub>	O	External Address Bus bit 4
	P1 <sub>4</sub>	I/O	General purpose input output or port 1 bit 4
41	A <sub>5</sub>	O	External Address Bus bit 5
	P1 <sub>5</sub>	I/O	General purpose input output or port 1 bit 5
42	A <sub>6</sub>	O	External Address Bus bit 6
	P1 <sub>6</sub>	I/O	General purpose input output or port 1 bit 6
43	A <sub>7</sub>	O	External Address Bus bit 7
	P1 <sub>7</sub>	I/O	General purpose input output or port 1 bit 7
44	VSS	-	Power ground
45	A <sub>8</sub>	O	External Address Bus bit 8
	P2 <sub>0</sub>	I/O	General purpose input output or port 2 bit 0
46	A <sub>9</sub>	O	External Address Bus bit 9
	P2 <sub>1</sub>	I/O	General purpose input output or port 2 bit 1
47	A <sub>10</sub>	O	External Address Bus bit 10
	P2 <sub>2</sub>	I/O	General purpose input output or port 2 bit 2

Table 1.1 Pin Descriptions (Continued)

PIN	SYMBOL	DIR	DESCRIPTION
48	A <sub>11</sub>	O	External Address Bus bit 11
	P <sub>23</sub>	I/O	General purpose input output or port 2 bit 3
49	A <sub>12</sub>	O	External Address Bus bit 12
	P <sub>24</sub>	I/O	General purpose input output or port 2 bit 4
50	A <sub>13</sub>	O	External Address Bus bit 13
	P <sub>25</sub>	I/O	General purpose input output or port 2 bit 5
51	A <sub>14</sub>	O	External Address Bus bit 14
	P <sub>26</sub>	I/O	General purpose input output or port 2 bit 6
52	A <sub>15</sub>	O	External Address Bus bit 15
	P <sub>27</sub>	I/O	General purpose input output or port 2 bit 7
53	A <sub>16</sub>	O	External Address Bus bit 16
	P <sub>50</sub>	I/O	General purpose input output of port 5 bit 0
54	A <sub>17</sub>	O	External Address Bus bit 17
	P <sub>51</sub>	I/O	General purpose input output of port 5 bit 1
55	A <sub>18</sub>	I	External Address Bus bit 18
	P <sub>52</sub>	I/O	General purpose input output of port 5 bit 2
56	A <sub>19</sub>	O	External Address Bus bit 19
	P <sub>53</sub>	I/O	General purpose input output of port 5 bit 3
57	VSS	-	Power ground
58	nWAIT	I	External BUS cycle wait signal
	P <sub>60</sub>	I/O	General purpose input output of port 6 bit 0
59	nBREQ	I	External BUS Request
	P <sub>61</sub>	I/O	General purpose input output of port 6 bit 1
60	nBACK	I	External BUS Acknowledge
	P <sub>62</sub>	I/O	General purpose input output of port 6 bit 2
61	CLKO	O	BUS Clock Output
	P <sub>67</sub>	I/O	General purpose input output of port 6 bit 7
62	nSTBY	O	Standby mode signal. Power Down mode indicating
63	nRES	I	External Reset input
64	nTRST	I	JTAG Test Reset input
	P <sub>97</sub>	I/O	General purpose input output of port 9 bit 7
65	VSS	-	Power ground
66	XTALOUT	O	Crystal feedback output
67	XTALIN	I	Crystal or External Oscillator input
68	VDD	-	Power Supply 3.3V
69	nAS	O	External Address Bus strobe
	P <sub>63</sub>	I/O	General purpose input output of port 6 bit 3
70	nRD	O	External Bus Read
	P <sub>64</sub>	I/O	General purpose input output of port 6 bit 4
71	nHWR	O	External upper 8 bit data bus write
	P <sub>65</sub>	I/O	General purpose input output of port 6 bit 5
72	nLWR	O	External lower 8 bit data bus write
	P <sub>66</sub>	I/O	General purpose input output of port 6 bit 6
73	MODE <sub>0</sub>	I	MODE bit 0
74	MODE <sub>1</sub>	I	MODE bit 1
75	MODE <sub>2</sub>	I	MODE bit 2
76	AVDD	-	Analog Power Supply 3.3V
77	AVREF	-	ADC Reference Voltage

Table 1.1 Pin Descriptions (Continued)

PIN	SYMBOL	DIR	DESCRIPTION
78	P7 <sub>0</sub>	O	General purpose output of port 7 bit 0
	AN <sub>0</sub>	I	ADC Channel 0 input
79	P7 <sub>1</sub>	O	General purpose output of port 7 bit 1
	AN <sub>1</sub>	I	ADC Channel 1 input
80	P7 <sub>2</sub>	O	General purpose output of port 7 bit 2
	AN <sub>2</sub>	I	ADC Channel 2 input
81	P7 <sub>3</sub>	O	General purpose output of port 7 bit 3
	AN <sub>3</sub>	I	ADC Channel 3 input
82	P7 <sub>4</sub>	O	General purpose output of port 7 bit 4
	AN <sub>4</sub>	I	ADC Channel 4 input
83	VSS	-	Power ground
84	TIOCA <sub>5</sub>	I/O	PWM output, Compare match output of Reg.A and signal capture input of Timer Ch5
	nIRQ <sub>6</sub>	I	External Interrupt Request number 6
	P7 <sub>6</sub>	I/O	General purpose input output of port 7 bit 6
85	TIOCB <sub>5</sub>	I/O	PWM output, Compare match output of Reg.B and signal capture input of Timer Ch5
	nIRQ <sub>7</sub>	I	External Interrupt Request number 7
	P7 <sub>7</sub>	I/O	General purpose input output of port 7 bit 7
86	P7 <sub>5</sub>	I/O	General purpose input output of port 7 bit 5
87	nIRQ <sub>0</sub>	I	External Interrupt Request number 0
	P8 <sub>0</sub>	I/O	General purpose input output of port 8 bit 0
88	nCS <sub>3</sub>	O	External Chip Selection Number 3
	nIRQ <sub>1</sub>	I	External Interrupt Request number 1
	P8 <sub>1</sub>	I/O	General purpose input output of port 8 bit 1
89	nCS <sub>2</sub>	O	External Chip Selection Number 2
	nIRQ <sub>2</sub>	I	External Interrupt Request number 2
	P8 <sub>2</sub>	I/O	General purpose input output of port 8 bit 2
90	nCS <sub>1</sub>	O	External Chip Selection Number 1
	nIRQ <sub>3</sub>	I	External Interrupt Request number 3
	P8 <sub>3</sub>	I/O	General purpose input output of port 8 bit 3
91	nCS <sub>0</sub>	O	External Chip Selection Number 0
	P8 <sub>4</sub>	I/O	General purpose input output of port 8 bit 4
92	VSS	-	Power ground
93	TCLKA	I	External timer input clock A
	PA <sub>0</sub>	I/O	General purpose input output of port A bit 0
94	TCLKB	I	External timer input clock B
	PA <sub>1</sub>	I/O	General purpose input output of port A bit 1
95	TCLKC	I	External timer input clock C
	TIOCA <sub>0</sub>	I/O	PWM output, Compare match output of Reg.A and signal capture input of Timer Ch0
	PA <sub>2</sub>	I/O	General purpose input output of port A bit 2
96	TCLKD	I	External timer input clock D
	TIOCB <sub>0</sub>	I/O	PWM output, Compare match output of Reg.B and signal capture input of Timer Ch0
	PA <sub>3</sub>	I/O	General purpose input output of port A bit 3
97	A <sub>23</sub>	O	External Address Bus bit 23
	TIOCA <sub>1</sub>	I/O	PWM output, Compare match output of Reg.A and signal capture input of Timer Ch1
	PA <sub>4</sub>	I/O	General purpose input output of port A bit 4

**Table 1.1 Pin Descriptions (Continued)**

PIN	SYMBOL	DIR	DESCRIPTION
98	A <sub>22</sub>	O	External Address Bus bit 22
	TIOCB <sub>1</sub>	I/O	PWM output, Compare match output of Reg.B and signal capture input of Timer Ch1
	PA <sub>5</sub>	I/O	General purpose input output of port A bit 5
99	A <sub>21</sub>	O	External Address Bus bit 21
	TIOCA <sub>2</sub>	I/O	PWM output, Compare match output of Reg.A and signal capture input of Timer Ch2
	PA <sub>6</sub>	I/O	General purpose input output of port A bit 6
100	A <sub>20</sub>	O	External Address Bus bit 20
	TIOCB <sub>2</sub>	I/O	PWM output, Compare match output of Reg.B and signal capture input of Timer Ch2
	PA <sub>7</sub>	I/O	General purpose input output of port A bit 7

## 1.4 Operation Mode description

HMS39C7092 is Flash Memory-embedded ARM microcontroller. It has six-operation modes shown in **Table 1.2**. HMS39C7092 External pin function is changed by setting external MODE pin or configuring the PIN MUX registers. The pin assignment by mode is shown in **Table 1.3**. Especially changing mode causes memory remap for appropriate mode. **Figure 1.3** shows default memory map and the memory maps of respective modes are shown in **Figure 1.4**, **Figure 1.5** and **Figure 1.6**. The Mode definition is listed as follows:

**Table 1.2 HMS39C7092 Operation modes**

MODE	MODE DESCRIPTION
0,1	Reserved for Test
2	External 8-bit data bus with 16MBytes of Address Range
3	External 16-bit data bus with 16MBytes of Address Range
4	Flash-boot mode with 16-bit data bus
5	Flash-boot mode (micro-computer mode)
6	UART-boot mode with 16-bit data bus
7	UART-boot mode (micro-computer mode)

Table 1.3 Pin assignment by mode

PIN	MODE 2 External 8bit BUS	MODE 3 External 16bit BUS	MODE 4 Flash boot mode with 16bit BUS	MODE 6 UART boot mode with 16bit BUS	MODE 5 Flash boot mode (MICOM mode)	MODE 7 UART boot mode (MICOM mode)
1	VDD	←	←	←	←	←
2	nCS7	←	←	←	TIOCA3	←
3	nCS6	←	←	←	TIOCB3	←
4	nCS5	←	←	←	TIOCA4	←
5	nCS4	←	←	←	TIOCB4	←
6	TMS	←	←	←	←	←
7	TDO	←	←	←	←	←
8	TDI	←	←	←	←	←
9	TCK	←	←	←	←	←
10	TVPPD	←	←	←	←	←
11	VSS	←	←	←	←	←
12	TxD0	←	←	←	←	←
13	RxD0	←	←	←	←	←
14	TxD1	←	←	←	←	←
15	RxD1	←	←	←	←	←
16	nIRQ4	←	←	←	←	←
17	nIRQ5	←	←	←	←	←
18	D0	←	←	←	P40	←
19	D1	←	←	←	P41	←
20	D2	←	←	←	P42	←
21	D3	←	←	←	P43	←
22	VSS	←	←	←	←	←
23	D4	←	←	←	P44	←
24	D5	←	←	←	P45	←
25	D6	←	←	←	P46	←
26	D7	←	←	←	P47	←
27	P30	D8	←	←	P30	←
28	P31	D9	←	←	P31	←
29	P32	D10	←	←	P32	←
30	P33	D11	←	←	P33	←
31	P34	D12	←	←	P34	←
32	P35	D13	←	←	P35	←
33	P36	D14	←	←	P36	←
34	P37	D15	←	←	P37	←
35	VDD	←	←	←	←	←
36	A0	←	←	←	P10	←
37	A1	←	←	←	P11	←
38	A2	←	←	←	P12	←
39	A3	←	←	←	P13	←
40	A4	←	←	←	P14	←

Table 1.3 Pin assignment by mode (continued)

PIN No.	MODE2 External 8bit BUS	MODE3 External 16bit BUS	MODE4 Flash boot mode with 16bit BUS	MODE6 UART boot mode with 16bit BUS	MODE5 Flash boot mode (MICOM mode)	MODE7 UART boot mode (MICOM mode)
41	A5	←	←	←	P15	←
42	A6	←	←	←	P16	←
43	A7	←	←	←	P17	←
44	VSS	←	←	←	←	←
45	A8	←	←	←	P20	←
46	A9	←	←	←	P21	←
47	A10	←	←	←	P22	←
48	A11	←	←	←	P23	←
49	A12	←	←	←	P24	←
50	A13	←	←	←	P25	←
51	A14	←	←	←	P26	←
52	A15	←	←	←	P27	←
53	A16	←	←	←	P50	←
54	A17	←	←	←	P51	←
55	A18	←	←	←	P52	←
56	A19	←	←	←	P53	←
57	VSS	←	←	←	←	←
58	nWAIT	←	←	←	P60	←
59	nBREQ	←	←	←	P61	←
60	nBACK	←	←	←	P62	←
61	CLKO	←	←	←	P67	←
62	nSTBY	←	←	←	nSTBY	←
63	nRES	←	←	←	nRES	←
64	nTRST	←	←	←	nTRST	←
65	VSS	←	←	←	←	←
66	XTALOUT	←	←	←	←	←
67	XTALIN	←	←	←	←	←
68	VDD	←	←	←	←	←
69	nAS	←	←	←	P63	←
70	nRD	←	←	←	P64	←
71	nHWR	←	←	←	P65	←
72	nLWR	←	←	←	P66	←
73	MODE0	←	←	←	←	←
73	MODE1	←	←	←	←	←
75	MODE2	←	←	←	←	←
76	AVDD	←	←	←	←	←
77	AVREF	←	←	←	←	←
78	AN0	←	←	←	←	←
79	AN1	←	←	←	←	←
80	AN2	←	←	←	←	←

Table 1.3 Pin assignment by mode (continued)

PIN No.	MODE2 External 8bit BUS	MODE3 External 16bit BUS	MODE4 Flash boot mode with 16bit BUS	MODE6 UART boot mode with 16bit BUS	MODE5 Flash boot mode (MICOM mode)	MODE7 UART boot mode (MICOM mode)
81	AN3	←	←	←	←	←
82	AN4	←	←	←	←	←
83	VSS	←	←	←	←	←
84	TIOCA5	←	←	←	←	←
85	TIOCB5	←	←	←	←	←
86	P75	←	←	←	←	←
87	nIRQ0	←	←	←	←	←
88	nCS3	←	←	←	P81	←
89	nCS2	←	←	←	P82	←
90	nCS1	←	←	←	P83	←
91	nCS0	←	←	←	P84	←
92	VSS	←	←	←	←	←
93	TCLKA	←	←	←	←	←
94	TCLKB	←	←	←	←	←
95	TCLKC	←	←	←	←	←
96	TCLKD	←	←	←	←	←
97	A23	←	←	TIOCA1	←	←
98	A22	←	←	TIOCB1	←	←
99	A21	←	←	TIOCA2	←	←
100	A20	←	←	TIOCB2	←	←



1.5 Memory Map

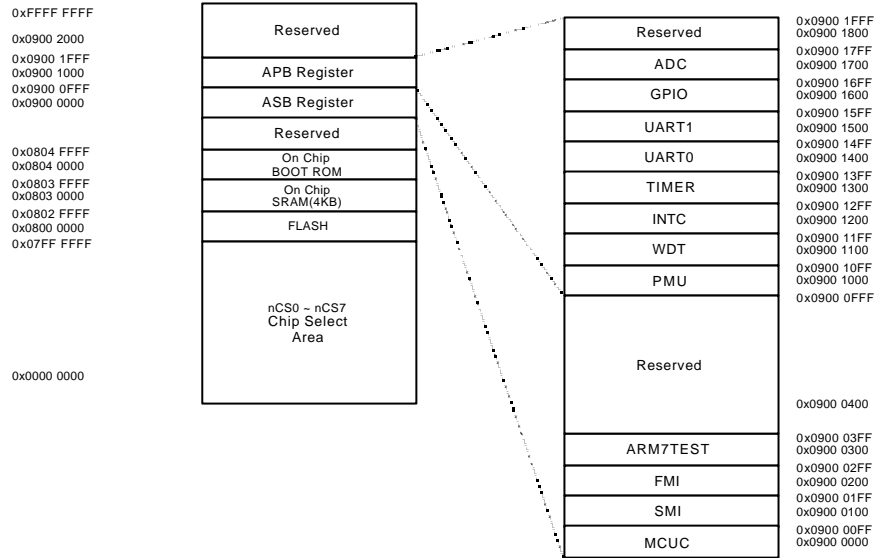


Figure 1.3 HMS39C7092 Memory Map

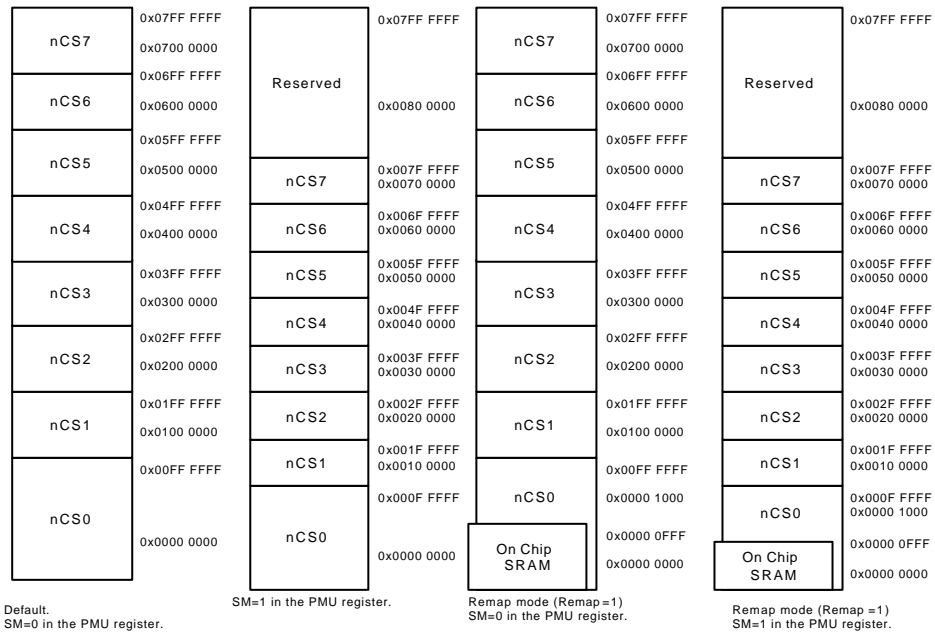
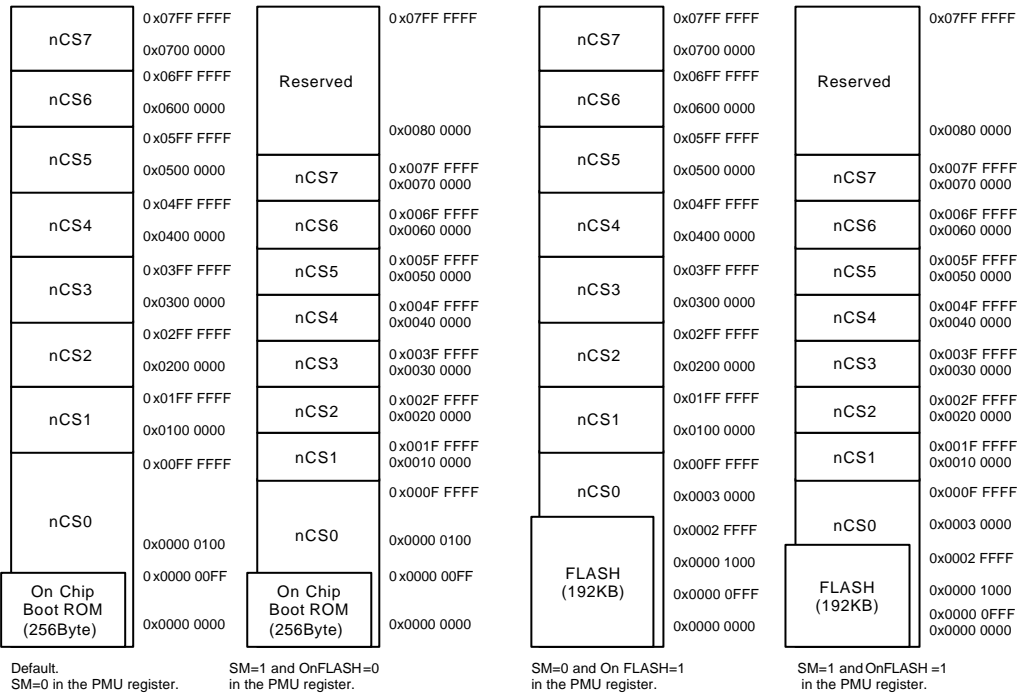
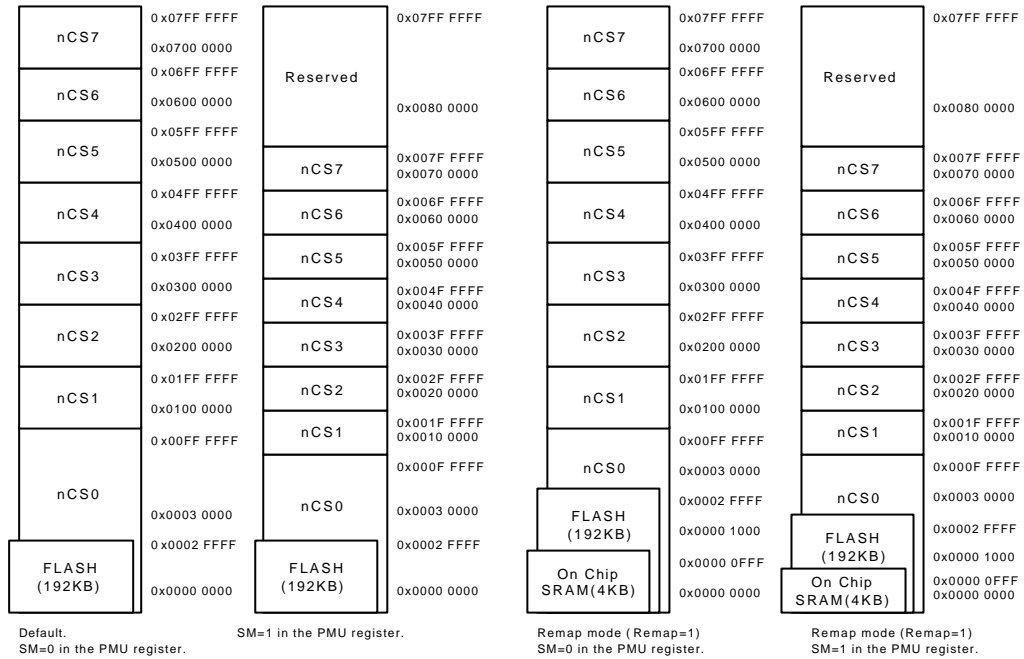


Figure 1.4 Memory Map of Mode 3



Chapter 2  
**ARM7TDMI Core**

## 2.1 General Description

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 ARM7TDMI Datasheet (ARM DDI 0029E) for detail.

## 2.2 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 (THUMB code)
- Fully static operation : 0 ~ 80MHz
- 3-stage pipeline architecture (Fetch, decode, and execution stage)
- 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.

2.3 Core Block Diagram

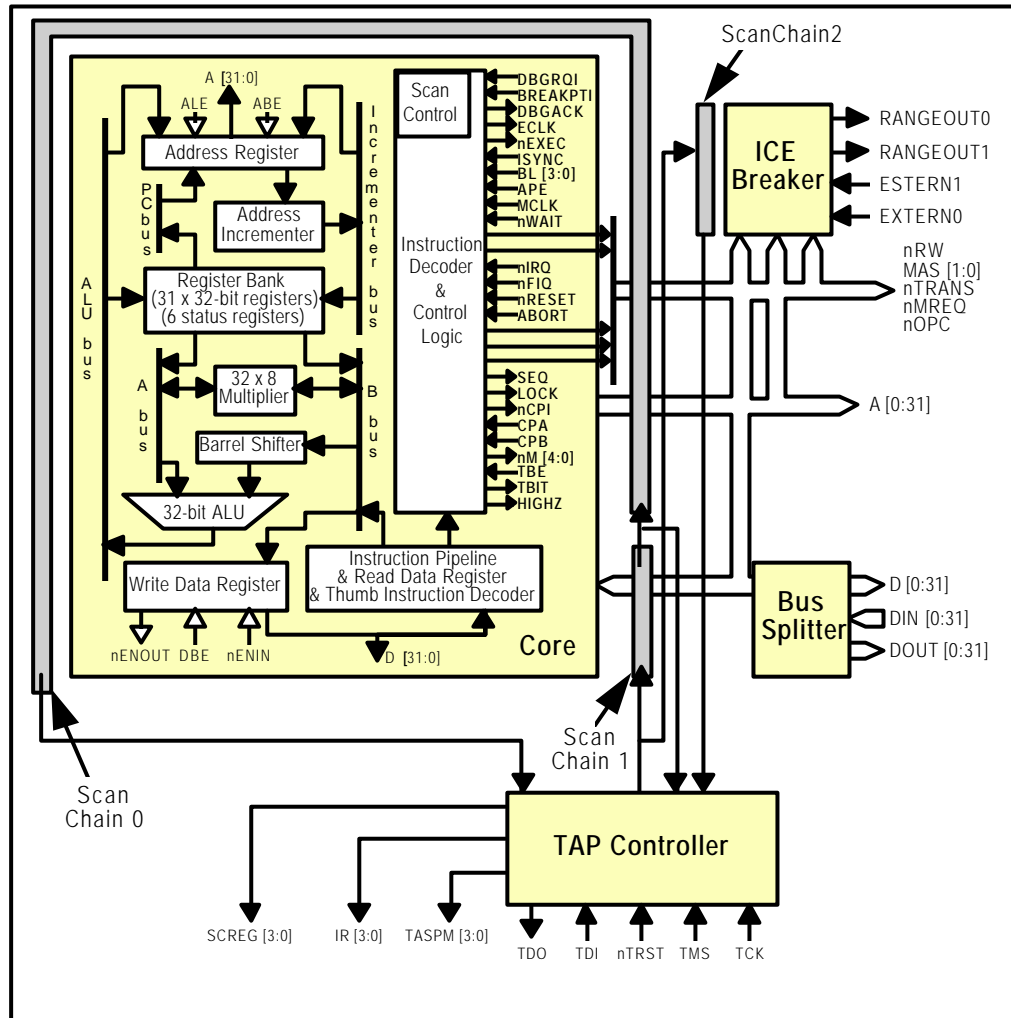


Figure 2.1 ARM7TDMI Core Block Diagram

2.4 Instruction Set

2.4.1 ARM Instruction

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Cond	0	0	1	Opcode				S	Rn	Rd	Operand										<i>Data Processing / PSR Transfer</i>											
Cond	0	0	0	0	0	0	0	A	S	Rd	Rn	Rs	1	0	0	1	Rm	<i>Multiply</i>														
Cond	0	0	0	0	1	U	A	S	RdHi	RdLo	Rn	1	0	0	1	Rm	<i>Multiply Long</i>															
Cond	0	0	0	1	0	B	0	0	Rn	Rd	0	0	0	0	1	0	0	1	Rm	<i>Single Data Swap</i>												
Cond	0	0	0	1	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	Rn	<i>Branch and Exchange</i>							
Cond	0	0	0	P	U	0	W	L	Rn	Rd	0	0	0	0	1	S	H	1	Rm	<i>Halfword Data Transfer: register offset</i>												
Cond	0	0	0	P	U	1	W	L	Rn	Rd	Offset				1	S	H	1	Offset	<i>Halfword Data Transfer: immediate offset</i>												
Cond	0	1	I	P	U	B	W	L	Rn	Rd	Offset						<i>Single Data Transfer</i>															
Cond	0	1	1														1		<i>Undefined</i>													
Cond	1	0	0	P	U	S	W	L	Rn	Register List								<i>Block Data Transfer</i>														
Cond	1	0	1	L	Offset												<i>Branch</i>															
Cond	1	1	0	P	U	N	W	L	Rn	CRd	CP#	Offset					<i>Coprocessor Data Transfer</i>															
Cond	1	1	1	0	CP	Opc	CRn	CRd	CP#	CP	0	CRm	<i>Coprocessor Data Operation</i>																			
Cond	1	1	1	0	CP	Opc	L	CRn	Rd	CP#	CP	1	CRm	<i>Coprocessor Register Transfer</i>																		
Cond	1	1	1	1	Ignored by processor												<i>Software Interrupt</i>															
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Figure 2.2 ARM instruction set formats

Table 2.1 The ARM Instruction set

Mnemonic	Instruction	Action
ADC	Add with carry	$Rd := Rn + Op2 + Carry$
ADD	Add	$Rd := Rn + Op2$
AND	AND	$Rd := Rn \text{ AND } Op2$
B	Branch	$R15 := \text{address}$
BIC	Bit Clear	$Rd := Rn \text{ AND NOT } Op2$
BL	Branch with Link	$R14 := R15, R15 := \text{address}$
BX	Branch and Exchange	$R15 := Rn, T \text{ bit} := Rn[0]$
CDP	Coprocessor Data Processing	(Coprocessor-specific)
CMN	Compare Negative	$CPSR \text{ flags} := Rn + Op2$
CMP	Compare	$CPSR \text{ flags} := Rn - Op2$
EOR	Exclusive OR	$Rd := (Rn \text{ AND NOT } Op2) \text{ OR } (Op2 \text{ AND NOT } Rn)$
LDC	Load coprocessor from memory	Coprocessor load
LDM	Load multiple registers	Stack manipulation (Pop)
LDR	Load register from memory	$Rd := (\text{address})$
MCR	Move CPU register to coprocessor register	$cRn := rRn \{<op>cRm\}$
MLA	Multiply Accumulate	$Rd := (Rm * Rs) + Rn$
MOV	Move register or constant	$Rd := Op2$
MRC	Move from coprocessor register to CPU register	$Rn := cRn \{<op>cRm\}$
MRS	Move PSR status/flags to register	$Rn := PSR$
MSR	Move register to PSR status/flags	$PSR := Rm$
MUL	Multiply	$Rd := Rm * Rs$
MVN	Move negative register	$Rd := 0xFFFFFFFF \text{ EOR } Op2$
ORR	OR	$Rd := Rn \text{ OR } Op2$
RSB	Reverse Subtract	$Rd := Op2 - Rn$
RSC	Reverse Subtract with Carry	$Rd := Op2 - Rn - 1 + Carry$
SBC	Subtract with Carry	$Rd := Rn - Op2 - 1 + Carry$
STC	Store coprocessor register to memory	$\text{address} := CRn$
STM	Store Multiple	Stack manipulation (Push)
STR	Store register to memory	$<\text{address}> := Rd$
SUB	Subtract	$Rd := Rn - Op2$
SWI	Software Interrupt	OS call
SWP	Swap register with memory	$Rd := [Rn], [Rn] := Rm$
TEQ	Test bitwise equality	$CPSR \text{ flags} := Rn \text{ EOR } Op2$
TST	Test bits	$CPSR \text{ flags} := Rn \text{ AND } Op2$

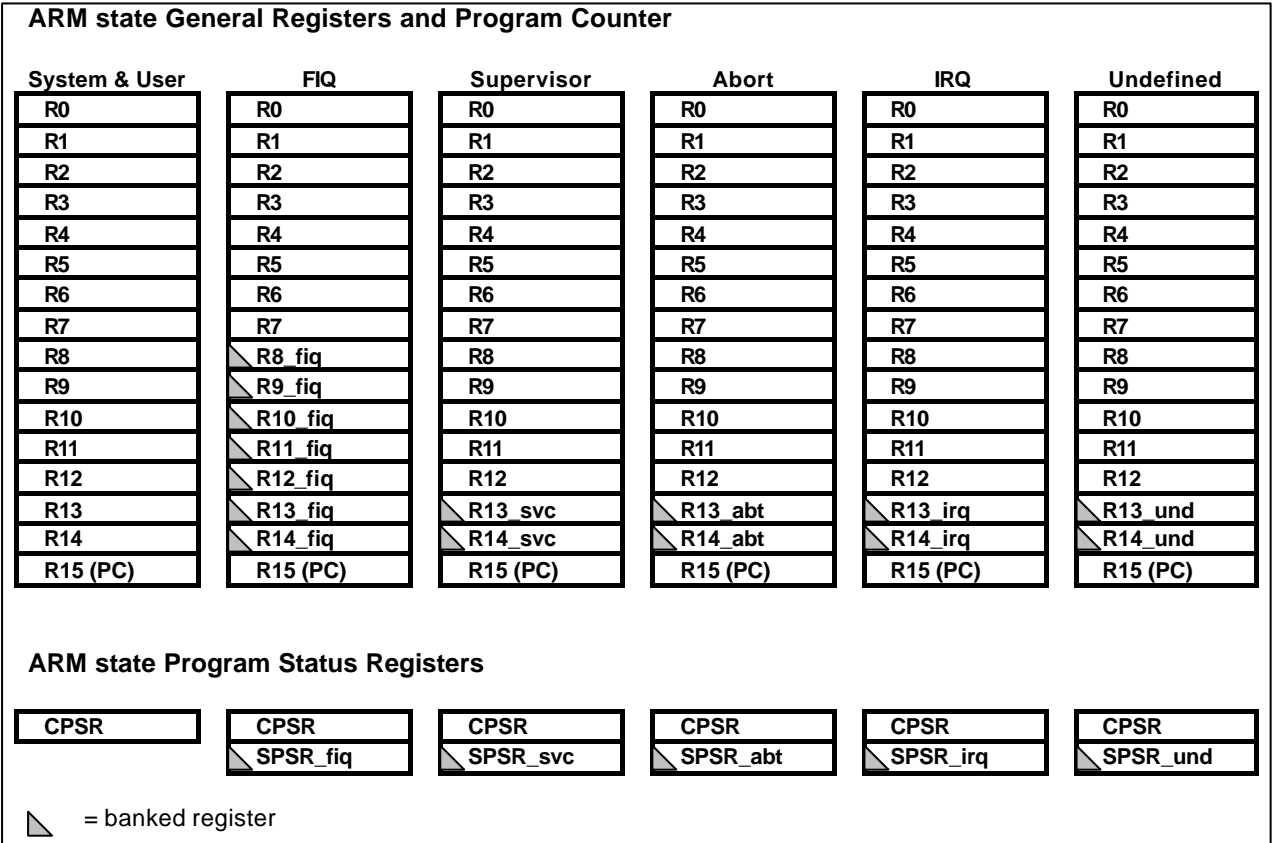


Figure 2.3 Register Organization in ARM state



## 2.4.2 THUMB Instruction

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
1	0	0	0			Offset5					Rs						Rd	<i>Move shifted register</i>	
2	0	0	0	1	1	I	Op	Rn/offset3					Rs				Rd	<i>Add/subtract</i>	
3	0	0	1		Op					Rd	Offset8							<i>Move/compare/add/subtract immediate</i>	
4	0	1	0	0	0	0				Op							Rs	Rd	<i>ALU operation</i>
5	0	1	0	0	0	1		Op	H1	H2							Rs/Hs	Rd/Hd	<i>Hi register operations/branch exchange</i>
6	0	1	0	0	1					Rd	Word8							<i>PC-relative load</i>	
7	0	1	0	1	L	B	0			Ro							Rb	Rd	<i>Load/store with register Offset</i>
8	0	1	0	1	H	S	1			Ro							Rb	Rd	<i>Load/store sign-extended byte/halfword</i>
9	0	1	1	B	L					Offset5							Rb	Rd	<i>Load/store with immediate</i>
10	1	0	0	0	L					Offset5							Rb	Rd	<i>Load/store halfword</i>
11	1	0	0	1	L					Rd	Word8							<i>SP-relative load/store</i>	
12	1	0	1	0	SP					Rd	Word8							<i>Load address</i>	
13	1	0	1	1	0	0	0	0	S		SWord7							<i>Add offset to stack pointer</i>	
14	1	0	1	1	L	1	0	R			Rlist							<i>Push/pop registers</i>	
15	1	1	0	0	L					Rb	Rlist							<i>Multiple load/store</i>	
16	1	1	0	1						Cond	Soffset8							<i>Conditional branch</i>	
17	1	1	0	1	1	1	1	1			Value8							<i>Software Interrupt</i>	
18	1	1	1	0	0						Offset11							<i>Unconditional branch</i>	
19	1	1	1	1	H						Offset11							<i>Long branch with link</i>	

Figure 2.4 THUMB instruction set formats

Table 2.2 THUMB instruction set opcodes

Mnemonic	Instruction	Lo reg. oper.	Hi reg. oper	Condition code set
ADC	Add with Carry	V		V
ADD	Add	V	V	V
AND	AND	V		V
ASR	Arithmetic Shift Right	V		V
B	Unconditional branch	V		
B xx	Conditional branch	V		
BIC	Bit Clear	V		V
BL	Branch and Link			
BX	Branch and Exchange	V		V
CMN	Compare Negative	V		V
CMP	Compare	V	V	V
EOR	EOR	V		V
LDMIA	Load multiple	V		
LDR	Load word	V		
LDRB	Load byte	V		
LDRH	Load halfword	V		
LSL	Logical Shift Left	V		V
LDSB	Load sign-extended byte	V		
LDSH	Load sign-extended Halfword	V		
LSR	Logical Shift Right	V		V
MOV	Move register	V	V	V
MUL	Multiply	V		V
MVN	Move Negative register	V		V
NEG	Negate	V		V
ORR	OR	V		V
POP	Pop registers	V		
PUSH	Push registers	V		
ROR	Rotate Right	V		V
SBC	Subtract with Carry	V		V
STMIA	Store Multiple	V		
STR	Store word	V		
STRB	Store byte	V		
STRH	Store halfword	V		
SWI	Software Interrupt			
SUB	Subtract	V		V
TST	Test bits	V		V

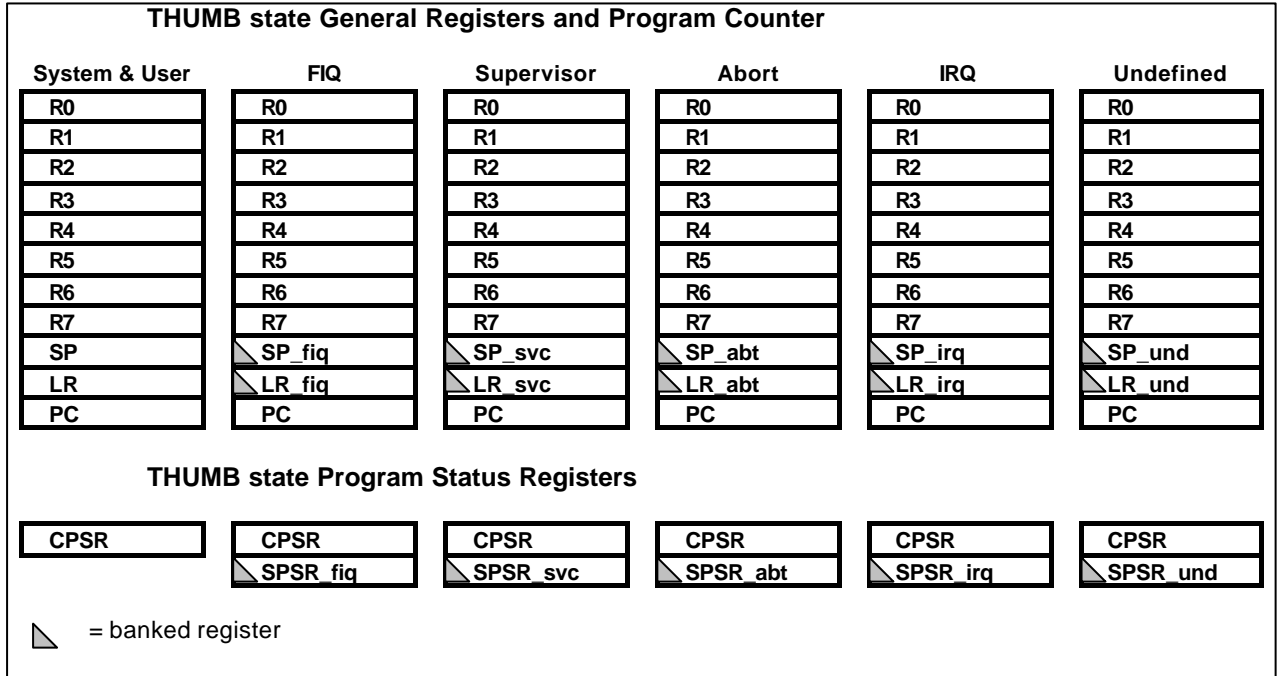


Figure 2.5 Register Organization in THUMB state

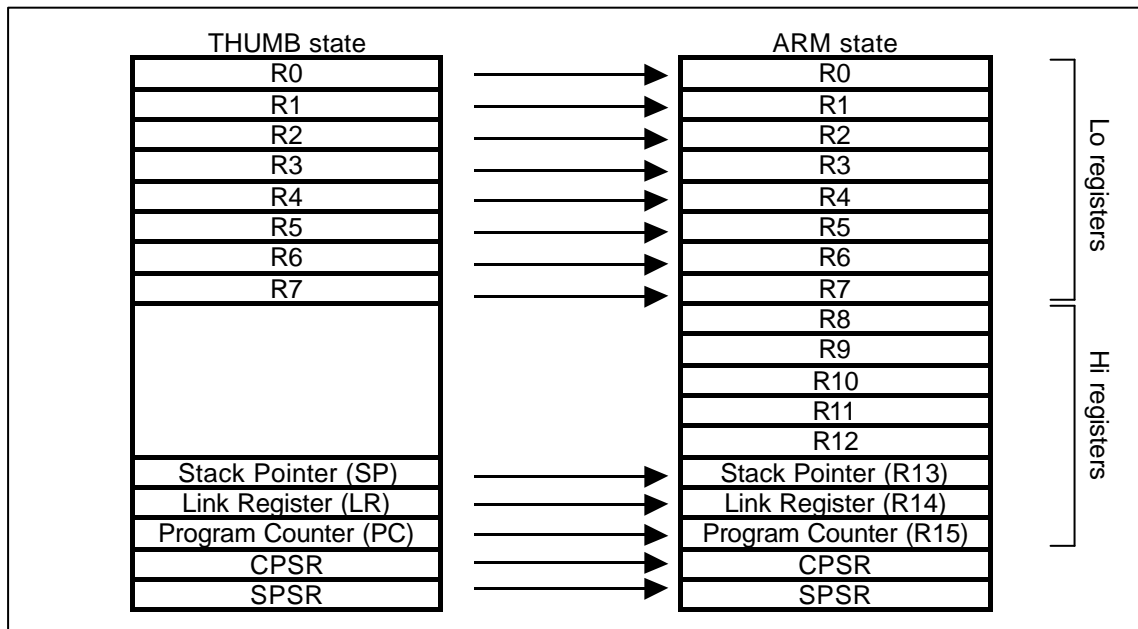


Figure 2.6 Mapping of THUMB state registers onto ARM state registers.

Table 2.3 Condition code summary

Code	Suffix	Flags	Meaning
0000	EQ	Z	set equal
0001	NE	Z	clear not equal
0010	CS	C	set unsigned higher or same
0011	CC	C	clear unsigned lower
0100	MI	N	set negative
0101	PL	N	clear positive or zero
0110	VS	V	set overflow
0111	VC	V	clear no overflow
1000	HI	C	set and Z clear unsigned higher
1001	LS	C	clear or Z set unsigned lower or same
1010	GE	N	equals V greater or equal
1011	LT	N	not equal to V less than
1100	GT	Z	clear AND (N equals V) greater than
1101	LE	Z	set OR (N not equal to V) less than or equal
1110	AL	(Ignored)	always

2.4.3 The Program Status Registers

The ARM7TDMI contains Current Program Status Register (CPSR), plus five Saved Program Status Register (SPSRs) for use by exception handlers. These registers hold information about the most recently performed ALU operation control the enabling and disabling of interrupts set the processor operating mode

The arrangement of bits is shown in Fig. 2.7 Program status register format.

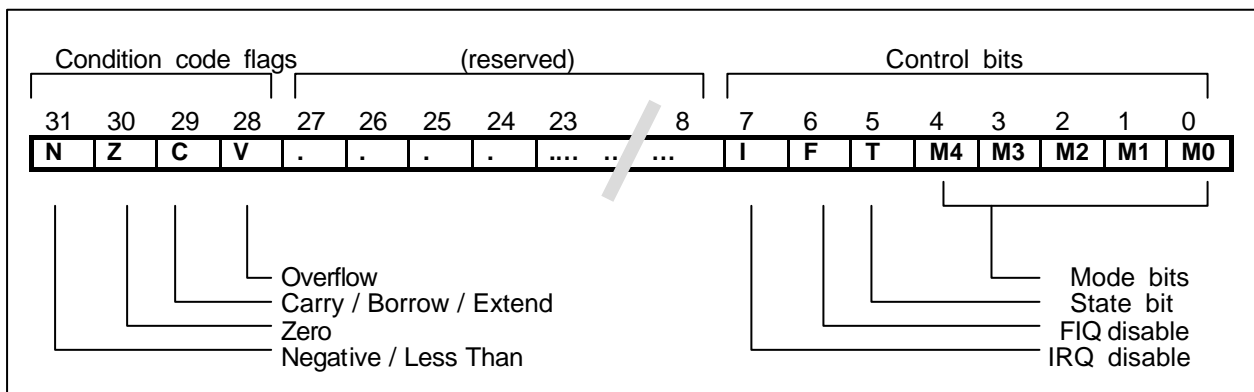


Figure 2.7 Program status register format

### 2.4.3.1 The condition code flags

The **N,Z,C** and **V** bits are the condition code flags. These may be changed as a result of arithmetic and logical operations, and may be tested to determine whether an instruction should be executed.

In ARM state, all instructions may be executed conditionally : see table 2.3 in chapter 2.4.2.

In THUMB state, only the Branch instruction is capable of conditional execution

### 2.4.3.2 The control bits

The bottom 8 bits of a PSR(incorporating I,F,T and M[4:0]) are known collectively as the control bits. These will change when an exception arises. If the processor is operating in a privileged mode, they can also be manipulated by software.

#### *The T bit*

This reflects the operating states. When this bit is set, the processor is executing in THUMB state, otherwise it is executing in ARM state.

Note that the software must never change the state of the **TBIT** in the CPSR. If this happens, the processor will enter an unpredictable state.

#### *Interrupt disable bits*

The **I** and **F** bits are the interrupt disable bits. When set, these disable the IRQ and FIQ interrupts respectively.

#### *The mode bits*

The **M4, M3, M2, M1** and **M0** bits (**M[4:0]**) are the mode bits. These determine the processor's operating mode, as shown in following table 2.4. Not all combinations of the mode bits define a valid processor mode. Only those explicitly described shall be used. The user should be aware that if any illegal value is programmed into the mode bits, **M{4:0}**, then the processor will enter an unrecoverable state. If this occurs, reset should be applied.

#### *Reserved bits*

The remaining bits in the PSRs are reserved. When changing a PSR's flag or control bits, you must ensure these unused bits are not altered. Also, your program should not rely on them containing specific values, since in future processors they may read as one or zero.

Table 2.4 PSR mode bit values

M[4:0]	Mode	Visible THUMB state registers	Visible ARM state registers
10000	User	R7..R0, LR, SP, PC, CPSR	R14..R0, PC, CPSR
10001	FIQ	R7..R0, LR_fiq, SP_fiq, PC, CPSR, SPSR_fiq	R7..R0, R14_fiq...R8_fiq, PC, CPSR, SPSR_fiq
10010	IRQ	R7..R0, LR_irq, SP_irq, PC, CPSR, SPSR_irq	R12..R0, R14_irq, R13_irq, PC, CPSR, SPSR_irq
10011	Supervisor	R7..R0, LR_svc, SP_svc, PC, CPSR, SPSR_svc	R12..R0, R14_svc, R13_svc, PC, CPSR, SPSR_svc
10111	Abort	R7..R0, LR_abt, SP_abt, PC, CPSR, SPSR_abt	R12..R0, R14_abt, R13_abt, PC, CPSR, SPSR_abt
11011	Undefined	R7..R0, LR_und, SP_und, PC, CPSR, SPSR_und	R12..R0, R14_und, R13_und, PC, CPSR
11111	System	R7..R0, LR, SP, PC, CPSR	R14..R0, PC, CPSR

#### 2.4.4 ARM pseudo-instructions

##### ADR

The ADR pseudo-instruction loads a program-relative or register-relative address into a register.

##### Syntax

The syntax of ADR is:

*ADR*{ condition} *register*, *expression*

where:

*register* is the register to load.

*expression* is a program-relative or register-relative expression that evaluates to:

- a non word-aligned address within 255 bytes
- a word-aligned address within 1020 bytes.

The address can be either before or after the address of the instruction or the base register.

##### Usage

ADR always assembles to one instruction. The assembler attempts to produce a single ADD or SUB instruction to load the address. If the address cannot be constructed in a single instruction, an error is generated and the assembly fails.

Use the ADRL pseudo-instruction to assemble a wider range of effective addresses.

If expression is program-relative, it must evaluate to an address in the same code area as the ADR pseudo-instruction. Otherwise the address may be out of range after linking.

##### Example

```
start    MOV r0,#10
         ADR r4,start      ; => SUB r4,pc,#0xc
```

## ADRL

The ADRL pseudo-instruction loads a program-relative or register-relative address into a register. It is similar to the ADR pseudo-instruction. ADRL can load a wider range of addresses than ADR because it generates two data processing instructions.

### Syntax

The syntax of ADRL is:

```
ADRL{ condition} register, expression
```

where:

*register* is the register to load.

*expression* is a register-relative or program-relative expression that evaluates to:

- a non word-aligned address within 64KB
- a word-aligned address within 256KB.

The address can be either before or after the address of the instruction or the base register.

### Usage

ADRL always assembles to two instructions. Even if the address can be reached in a single instruction, a second, redundant instruction is produced.

If the assembler cannot construct the address in two instructions, it generates an error message and the assembly fails. See *LDR ARM pseudo-instruction* for information on loading a wider range of addresses. See also Chapter 5 *Basic Assembly Language Programming* in the *ARM Software Development Toolkit User Guide*.

If expression is program-relative, it must evaluate to an address in the same code area as the ADRL pseudo-instruction. Otherwise the address may be out of range after linking.

### Note

ADRL is not available when assembling Thumb instructions. Use it only in ARM code.

### Example

```
start    MOV r0,#10
         ADRL r4,start + 60000      ; => ADD r4,pc,#0xe800
                                         ; ADD r4,r4,#0x254
```



**LDR**

The LDR pseudo-instruction loads a register with either:

- a 32-bit constant value
- an address.

**Note**

This section describes the LDR *pseudo*-instruction only. Refer to the *ARM Architectural Reference Manual* for information on the LDR *instruction*.

**Syntax**

The syntax of LDR is:

```
LDR{ condition} register, =[ expression | label-expression]
```

where:

*condition* is an optional condition code.

*register* is the register to be loaded.

*expression* evaluates to a numeric constant:

- If the value of expression is within range of a MOV or MVN instruction, the assembler generates the appropriate instruction.
- If the value of expression is *not* within range of a MOV or MVN instruction, the assembler places the constant in a literal pool and generates a program-relative LDR instruction that reads the constant from the literal pool.

The offset from the pc to the constant must be less than 4KB. You are responsible for ensuring that there is a literal pool within range. See *LTORG directive* for more information.

*label-expression* is a program-relative or external expression. The assembler places the value of label-expression in a literal pool and generates a program-relative LDR instruction that loads the value from the literal pool.

The offset from the pc to the value in the literal pool must be less than 4KB. You are responsible for ensuring that there is a literal pool within range. See *LTORG directive* for more information.

If label-expression is an external expression, or is not contained in the current area, the assembler places a linker relocation directive in the object file. The linker ensures that the correct address is generated at link time.

**Usage**

The LDR pseudo-instruction is used for two main purposes:

- to generate literal constants when an immediate value cannot be moved into a register because it is out of range of the MOV and MVN instructions.
- to load a program-relative or external address into a register. The address remains valid regardless of where the linker places the AOF area containing the LDR. Refer to Chapter 5 *Basic Assembly Language Programming* in the *ARM Software Development Toolkit User Guide* for a more detailed explanation of how to use LDR, and for more information on MOV and MVN.

**Example**

```
LDR r1,=0xfff      ; loads 0xfff into r1
                   ;
LDR r2,=place      ; loads the address of
                   ; place into r2
```

**NOP**

NOP generates the preferred ARM no-operation code. This is:  
MOV r0,r0

**Syntax**

The syntax of NOP is:

NOP

**Usage**

NOP cannot be used conditionally. Not executing a no-operation is the same as executing it, so conditional execution is not required. Condition codes are unaltered by NOP.

### 2.4.5 THUMB pseudo-instructions

#### ADR

The thumb ADR pseudo-instruction loads a program-relative or register-relative address into a register.

#### Syntax

The syntax of ADR is:

ADR register, expression

where:

register is the register to load.

Expression is a register-relative or program-relative expression that evaluates to a word-aligned address within the range +4 to +1020 bytes. Expression must be defined locally, it cannot be imported.

Refer to *^ or MAP directive* for more information on register-relative expressions.

#### Usage

In Thumb state, ADR can generate word-aligned addresses only. Use the ALIGN directive to ensure that expression is aligned.

If expression is program-relative, it must evaluate to an address in the same code area as the ADR pseudo-instruction. There is no guarantee that the address will be within range after linking if it resides in another AOF area.

#### Example

```
ADR r4, txampl      ; => ADD r4,pc,#nn
; code
ALIGN
txampl DCW 0,0,0,0
```

**LDR**

The thumb LDR pseudo-instruction loads a low register with either:

- a 32-bit constant value
- an address.

**Note**

This section describes the LDR *pseudo*-instruction only. Refer to the *ARM Architectural Reference Manual* for information on the LDR *instruction*.

**Syntax**

The syntax of LDR is:

```
LDR register, =[ expression | label-expression]
```

where:

*register* is the register to be loaded. LDR can access the low registers (r0-r7) only. *expression* evaluates to a numeric constant:

- If the value of *expression* is within range of a MOV instruction, the assembler generates the instruction.
- If the value of *expression* is *not* within range of a MOV instruction, the assembler places the constant in a literal pool and generates a program-relative LDR instruction that reads the constant from the literal pool.

The offset from the pc to the constant must be positive and less than 1KB. You are responsible for ensuring that there is a literal pool within range. See *LORG directive* for more information.

*label-expression* is a program-relative or external expression. The assembler places the value of *label-expression* in a literal pool and generates a program-relative LDR instruction that loads the value from the literal pool.

The offset from the pc to the value in the literal pool must be positive and less than 1KB. You are responsible for ensuring that there is a literal pool within range. See *LORG directive* for more information.

If *label-expression* is an external expression, or is not contained in the current area, the assembler places a linker relocation directive in the object file. The linker ensures that the correct address is generated at link time.

**Usage**

The LDR pseudo-instruction is used for two main purposes:

- to generate literal constants when an immediate value cannot be moved into a register because it is out of range of the MOV instruction.
- to load a program-relative or external address into a register. The address remains valid regardless of where the linker places the AOF area containing the LDR.

Refer to Chapter 5 *Basic Assembly Language Programming* in the *ARM Software Development Toolkit User Guide* for a more detailed explanation of how to use LDR, and for more information on MOV.

**Example**

```
LDR r1, =0xfff      ; loads 0xfff into r1
                   ;
LDR r2, =labelname ; loads the address of
                   ; labelname into r2
```

**MOV**

The Thumb MOV *pseudo*-instruction moves the value of a low register to another low register (r0-r7).

The Thumb MOV *instruction* cannot move values from one low register to another.

**Note**

The ADD immediate instruction generated by the assembler has the side-effect of updating the condition codes.

**Syntax**

The syntax of MOV is:

MOV *Rd*, *Rs*

where:

*Rd* is the destination register.

*Rs* is the source register.

**Usage**

The MOV pseudo-instruction uses an ADD immediate instruction with a zero immediate value. Refer to the *ARM Architectural Reference Manual* for more information on the Thumb MOV instruction.

**Example**

MOV *Rd*, *Rs* ; generates the opcode for ADD *Rd*, *Rs*, #0

**NOP**

NOP generates the preferred Thumb no-operation instruction. This is:  
MOV r8,r8

**Syntax**

The syntax for NOP is:

NOP

**Usage**

Condition codes are unaltered by NOP

Chapter 3  
**BUS Controller**

3.1 Overview

The HMS39C7092 has an on-chip bus controller that manages the external address space divided into eight areas, which can attach SRAM, ROM, Flash-memory or off-chip peripheral devices. The bus specifications, such as bus width and number of access states, can be set independently for each area, enabling multiple memories to be connected easily.

3.1.1 Features

The features of the bus controller are listed below.

- 8-bit access or 16-bit access can be selected for each area (In THUMB mode, only 16-bit accessing of external code memory is allowed)
- Active low chip select signals ( $nCS_0$  to  $nCS_7$ ) can be output for area 0 to 7
- Bus specifications can be set independently for each area
- Support Little-Endian Memory Format
- Variable wait states (up to 16 waits)
- Bus transfers can be extended using the  $nWAIT$  signal. The  $nWAIT$  signal is active LOW
- Each area is 16MB(when SM='0' in PMU), or 1MB(when SM='1' in PMU) in Size and can be programmed individually.

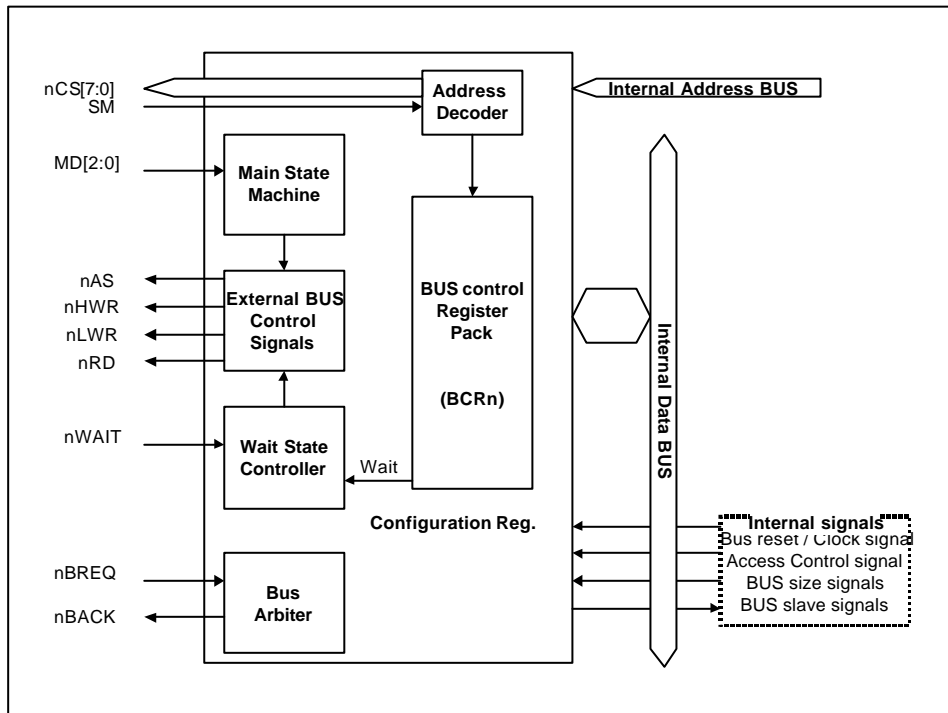


Figure 3.1 Block Diagram of the Bus Controller



## 3.1.2 Pin Configuration

*Table 3.1* summarizes the input/output pins of the bus controller.

**Table 3.1 Bus Controller Pins**

Name	I/O	Function
nCS <sub>n</sub>	O	Strobe signals selecting areas 0 to 7
nAS	O	Strobe signal indicating valid address output on the address bus
nRD	O	Strobe signal indicating reading from the external address space
nHWR	O	Strobe signal indicating writing to the external address space, with valid data on the upper data bus (D <sub>15</sub> to D <sub>8</sub> )
nLWR	O	Strobe signal indicating writing to the external address space, with valid data on the lower data bus (D <sub>7</sub> to D <sub>0</sub> )
nWAIT	I	Wait request signal
nBREQ	I	Request signal for releasing the bus to an external device
nBACK	O	Acknowledge signal indicating release of the bus to an external device

### 3.2 Bus Controller Registers

The base address for the BUS Controller's registers is **0x0900\_0100**. Each configuration registers (BCR0~7) are assigned to chip selected area, CS0~CS7.

**Table 3.2 BUS Controller Register Map**

Reg.	I/O Offset	Dir.	Description	Initial Value
<b>BCR0</b>	0x0100	R/W	CS0 Bus Configuration Register	0x00F*
<b>BCR1</b>	0x0104	R/W	CS1 Bus Configuration Register	0x0
<b>BCR2</b>	0x0108	R/W	CS2 Bus Configuration Register	0x0
<b>BCR3</b>	0x010C	R/W	CS3 Bus Configuration Register	0x0
<b>BCR4</b>	0x0110	R/W	CS4 Bus Configuration Register	0x0
<b>BCR5</b>	0x0114	R/W	CS5 Bus Configuration Register	0x0
<b>BCR6</b>	0x0118	R/W	CS6 Bus Configuration Register	0x0
<b>BCR7</b>	0x011C	R/W	CS7 Bus Configuration Register	0x0

Notes : 1) In mode 2, the initial value of BCR0 is 0x010F.  
 2) In mode 3, the initial value of BCR0 is 0x000F.  
 3) The initial value of the other control registers are 0x0000.

### 3.2.1 Configuration Registers

The configuration register (BCR0~7) is a 16-bit read-write register.

#### BCR0~7 Bus Configuration Register (0x0900\_0100 to 0x0900\_011C R/W)

	B15 - b9	b8	b7	B6	B5	b4	b3	b2	b1	b0
<b>BCRn</b>	Reserved	MemWidth	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Reset	0000000	1	0	0	0	0	1	1	1	1
	Initial value : 0x010F (BCR0 at Mode2)									
	0x000F (BCR0 at Mode3)									
	0x0000 (BCR1~7)									

**MemWidth** Select the size of the external bus width. When this bit is 0, means that the MCU interface with 8bit external bus. When 1, the external bus of the MCU is 16 bit width bus.

**NormWait** Select the values of the normal access wait state

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

3.3 Operation

3.3.1 Area Division

The external address space is divided into area 0 to 7. Each area has a size of 16-Mbyte modes, or 1-Mbyte modes. Figure 3.2 shows a general view of the memory map.

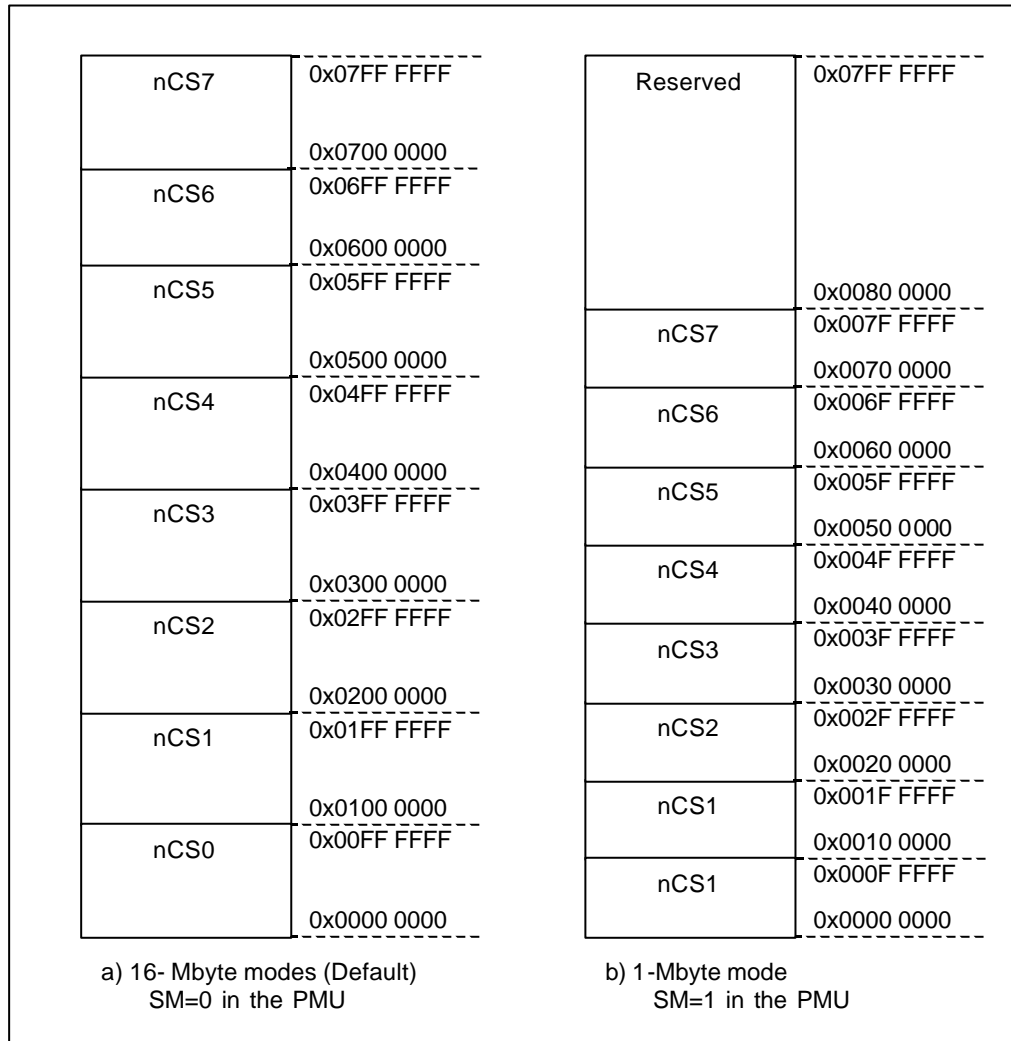


Figure 3.2 Access Area Map for Each Operating Mode

Chip select signals (nCS<sub>0</sub> to nCS<sub>7</sub>) can be output for area 0 to 7. The bus specifications for each area are selected in BCR<sub>0</sub> to BCR<sub>7</sub>.

### 3.3.2 Area Division

The external space bus specifications consist of two elements: (1) bus width, (2) number of wait states.

The bus width and number of access states for on-chip memory and registers are fixed, and are not affected by the bus controller.

**Bus Width:** A bus width of 8 or 16 bits can be selected with MemWidth bit-field in BCR0 to 7. An area for which an 8-bit bus is selected functions as an 8-bit access space, and an area for which a 16-bit bus is selected functions as a 16-bit access space.

If all areas are designed for 8-bit access, 8-bit bus mode is set; if any area is designed for 16-bit access, 16-bit bus mode is set.

**Number of Wait States:** One to 16 wait states can be selected with NormalWait bit-field in BCR0 to 7. When using **nWAIT** signal, then wait state is the minimum over two-states.

### 3.3.3 Chip Select Signals

For each of areas 0 to 7, the HMS39C7092 can output a chip select signal (**nCS<sub>0</sub>** to **nCS<sub>7</sub>**) that goes low when the corresponding area is selected in expanded mode. From **Figure 3.3** to **Figure 3.15** shows the output timing of **nCS<sub>0-7</sub>** signal.

**Output of nCS<sub>0</sub> to nCS<sub>7</sub>:** Output of **nCS<sub>0</sub>** to **nCS<sub>7</sub>** is enabled or disabled in the data direction register of the corresponding port.

**3.4 Basic Bus Interface**

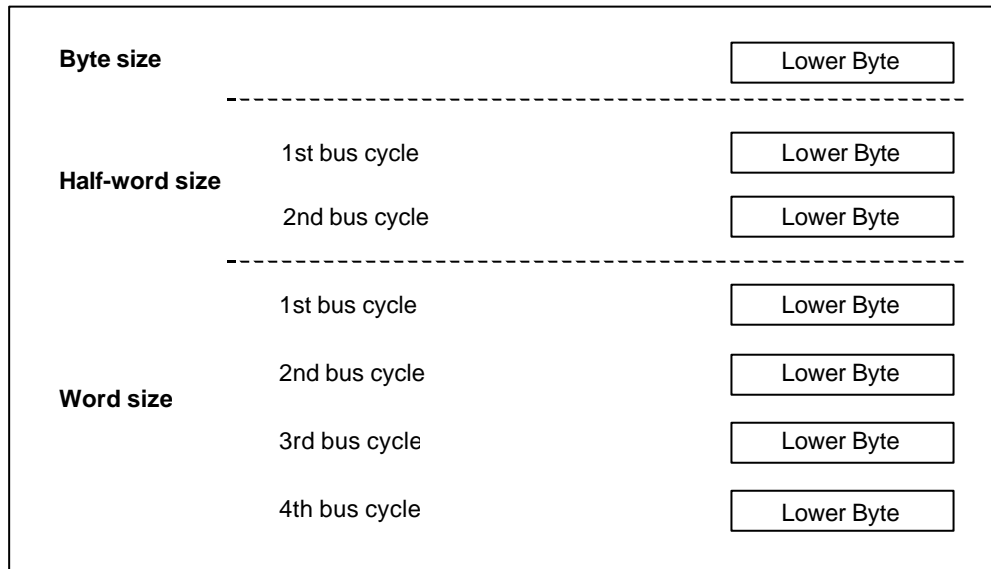
**3.4.1 Overview**

The HMS39C7092 has only a basic interface that allows direct connection of ROM, SRAM, off-chip peripheral devices and so on.

**3.4.2 Byte Lane Write Control**

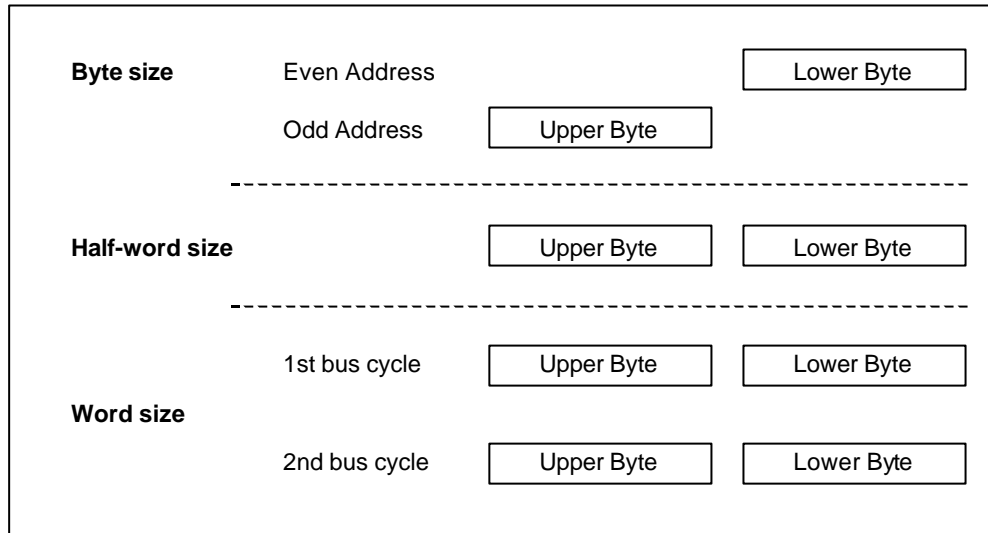
Data size for the CPU and other internal masters are byte(8-bit), half-word(16-bit), word(32-bit). The bus controller has a data alignment function, and when accessing external space, controls whether the upper data bus (**D<sub>15</sub> to D<sub>8</sub>**) or lower data bus (**D<sub>7</sub> to D<sub>0</sub>**) is used according to the bus specifications for the area being accessed (8-bit access area or 16-bit access area) and the data size.

**8-Bit Access Areas:** *Figure 3.3* shows data alignment control for 8-bit access space. With 8-bit access space, the lower data bus (**D<sub>7</sub> to D<sub>0</sub>**) is always used for accesses. The amount of data that can be accessed at one time is one byte: a half-word access is performed as two byte accesses, and a word access, as four byte accesses.



**Figure 3.3 Access Size and Data Alignment Control (8-Bit Access Area)**

**16-Bit Access Areas:** *Figure 3.4* shows data alignment control for 16-bit access areas. With 16-bit access areas, the lower data bus (**D<sub>7</sub>** to **D<sub>0</sub>**) and higher data bus (**D<sub>15</sub>** to **D<sub>8</sub>**) are used for accesses. The amount of data that can be accessed at one time is one byte or one half-word, and a word access is executed as two half-word accesses.



*Figure 3.4 Access Size and Data Alignment Control (16-Bit Access Area)*

**nHWR**, **nLWR** signals are generated according to the memory transfer width, external memory width, **A0**, and the access sequencing. The following table shows the basic coding example assuming 16-bit external memory:

*Table 3.3 Byte Lane condition by XA[0]*

CPU access Size	A0	nHWR	nLWR	Number of Access
Word (32bit)	X	Low	Low	2
Half-word (16-bit)	X	Low	Low	1
Byte (8bit)	0	High	Low	1
Byte (8bit)	1	Low	High	1

3.4.3 Basic Bus Control Signal Timing

**16-Bit 1-Wait-Access Areas:** *Figure 3.5* shows the write timing of bus control signals for a 16-Bit 1-wait-access area (in case of 32-bit word access). *Figure 3.6* shows the read timing of bus control signals for a 16-Bit 1-wait-access area (In case of 32-bit word access). In this case the NormWait value in **BCR** of this area is '0'.

Note: Sequential read access keeps nRD signal to LOW state.

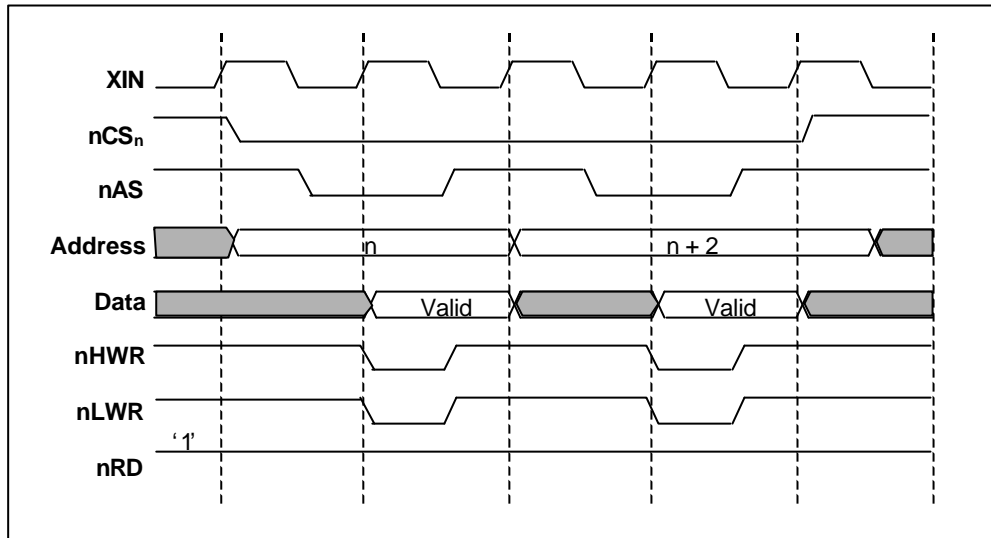


Figure 3.5 Bus Control Signal Write Timing for 16-Bit, 1-Wait (Word Access)

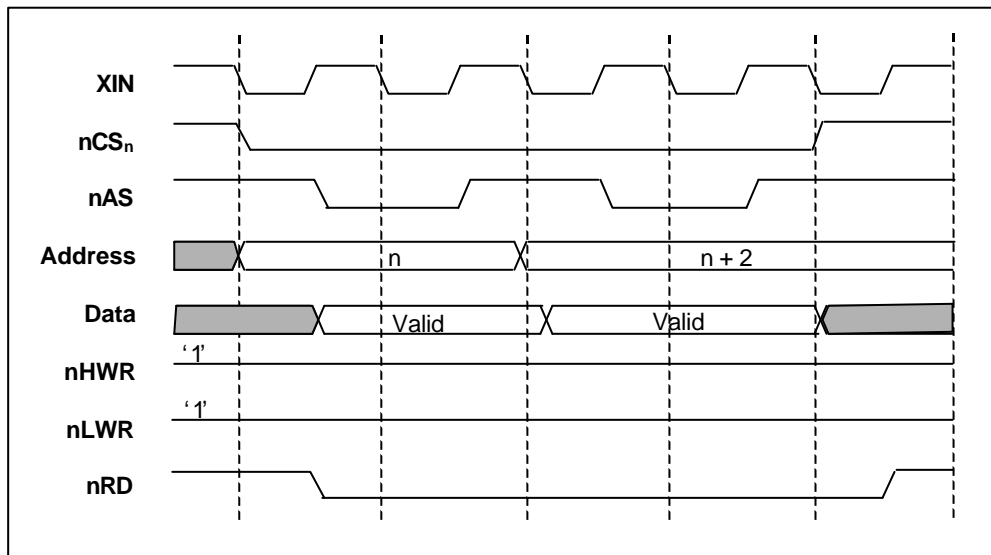


Figure 3.6 Bus Control Signal Read Timing for 16-Bit, 1-Wait (Word Access)



Figure 3.7 shows the write timing of bus control signals for a 16-Bit 1-wait-access area (In case of half-word access). Figure 3.8 shows the read timing of bus control signals for a 16-Bit 1-wait-access area (In case of half-word access).

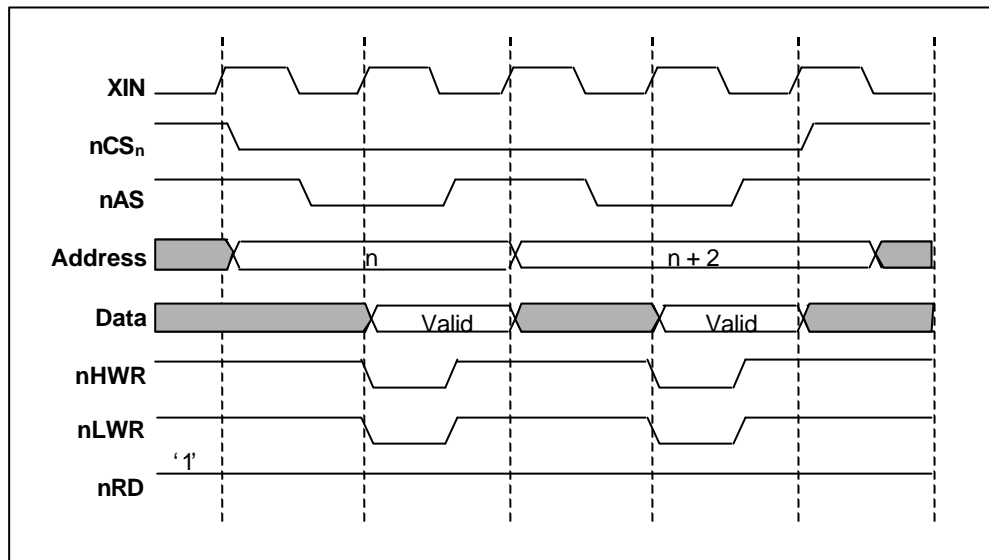


Figure 3.7 Bus Control Signal Write Timing for 16-Bit, 1-Wait (Half-word Access)

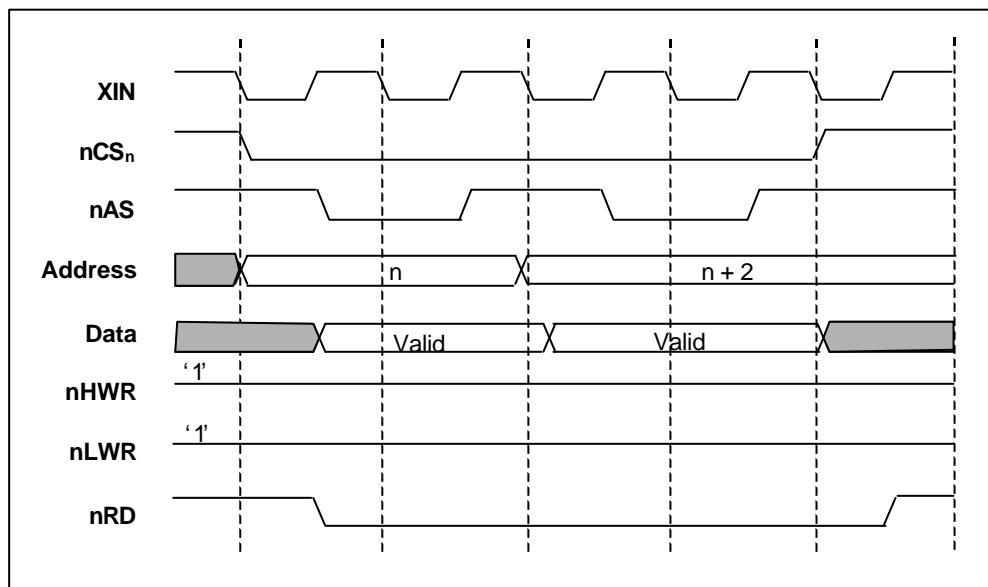


Figure 3.8 Bus Control Signal Read Timing for 16-Bit, 1-Wait (Half-word Access)

Figure 3.9 shows the write timing of bus control signals for a 16-Bit 1-wait-access area (In case of byte access). Figure 3.10 shows the read timing of bus control signals for a 16-Bit 1-wait-access area (In case of byte access).

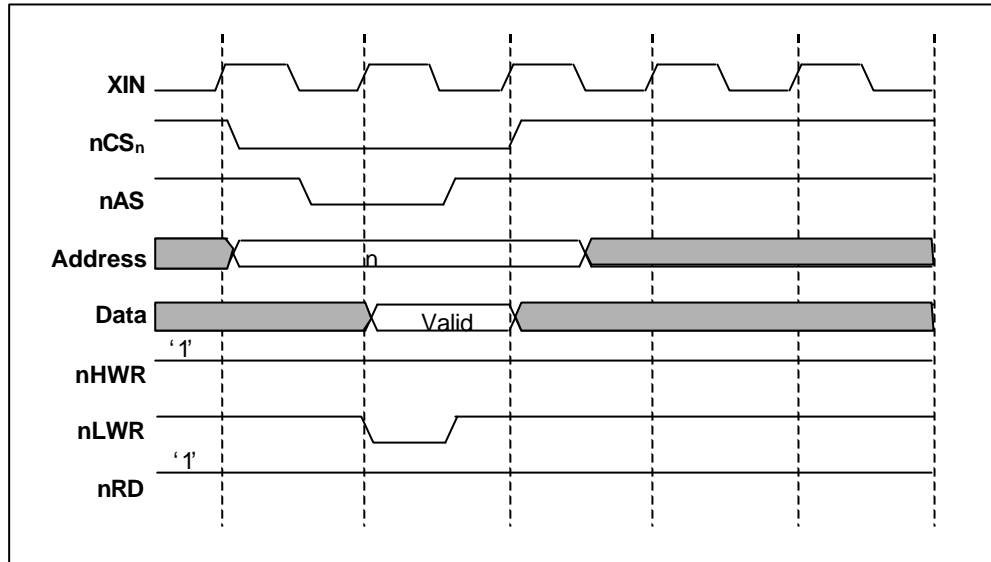


Figure 3.9 Bus Control Signal Write Timing for 16-Bit, 1-Wait (Byte Access)

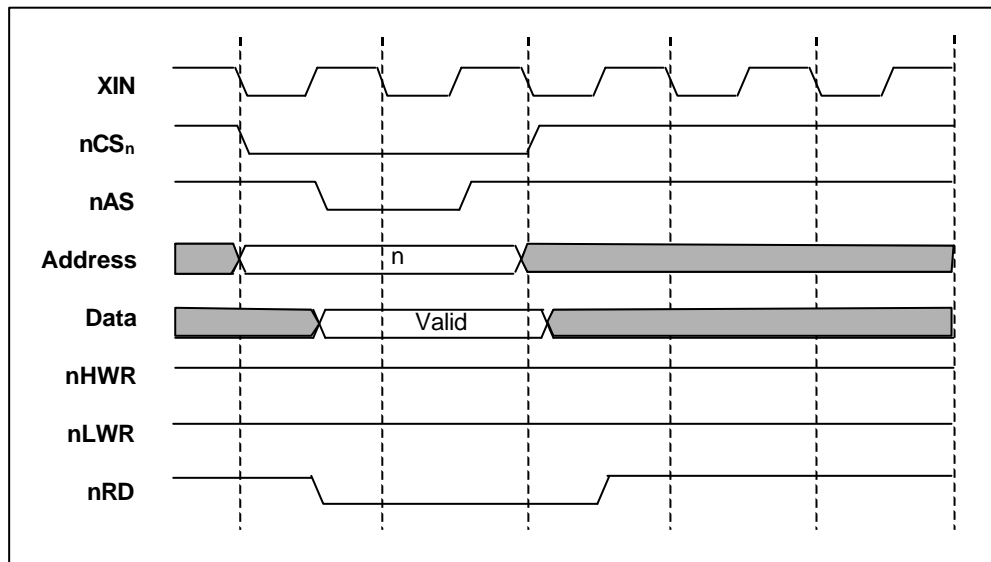
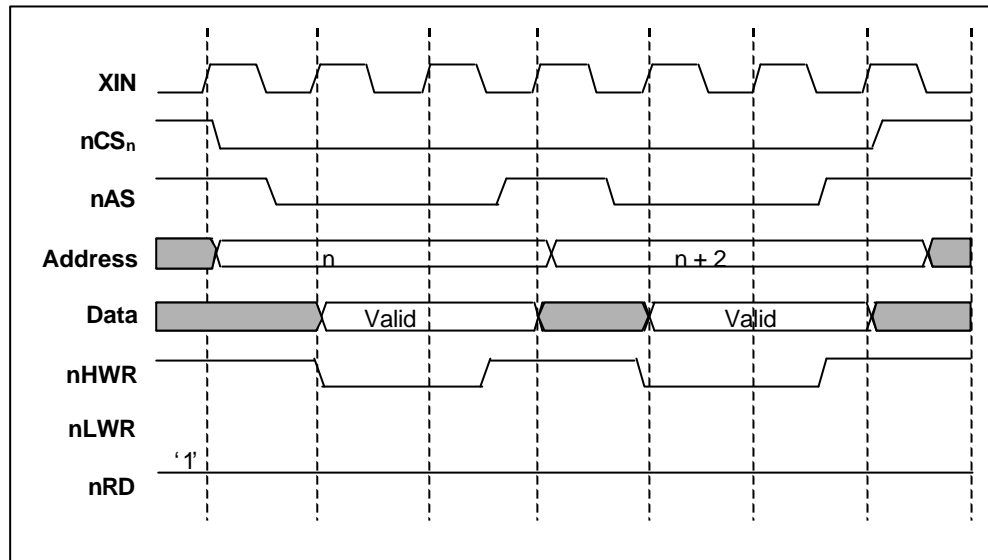
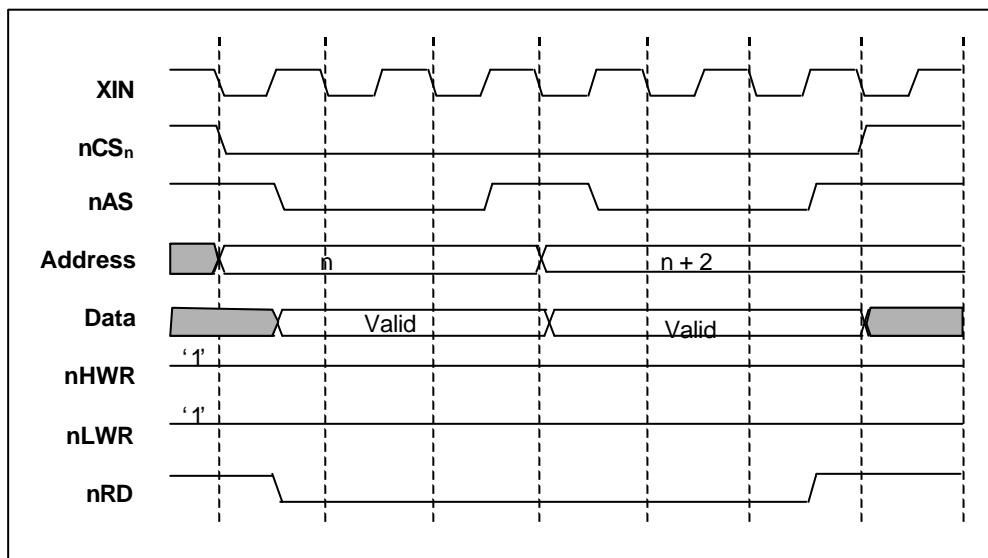


Figure 3.10 Bus Control Signal Read Timing for 16-Bit, 1-Wait (Byte Access)

**Figure 3.11** shows the write timing of bus control signals for a 16-Bit 2-wait-access area (In case of word access). **Figure 3.12** shows the read timing of bus control signals for a 16-Bit 2-wait-access area (In case of word access).



**Figure 3.11** Bus Control Signal Write Timing for 16-Bit, 2-Wait (Word Access)



**Figure 3.12** Bus Control Signal Read Timing for 16-Bit, 2-Wait (Word Access)

Figure 3.13 shows the write timing of bus control signals for a 16-Bit 2-wait-access area (In case of half-word access). Figure 3.14 shows the read timing of bus control signals for a 16-Bit 2-wait-access area (In case of half-word access).

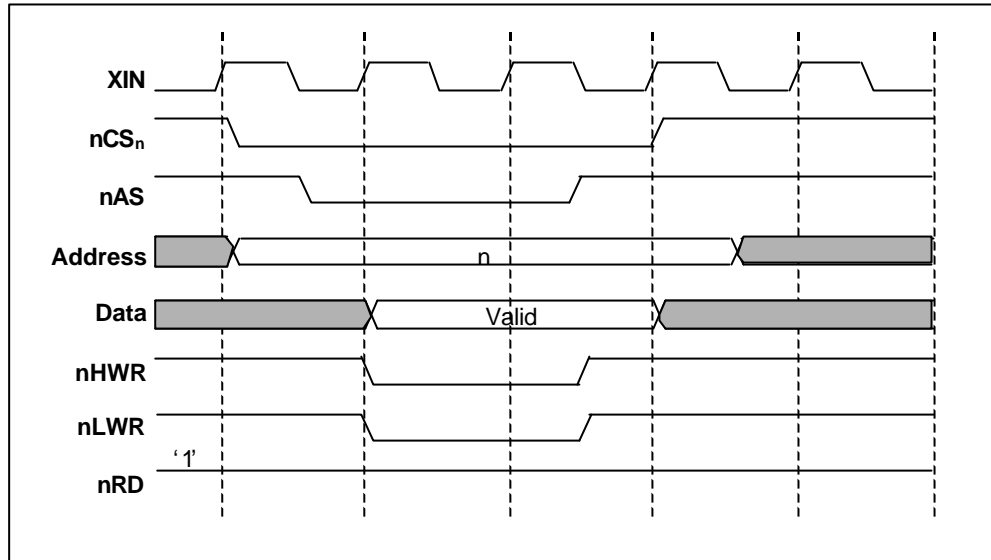


Figure 3.13 Bus Control Signal Write Timing for 16-Bit, 2-Wait (Half-Word Access)

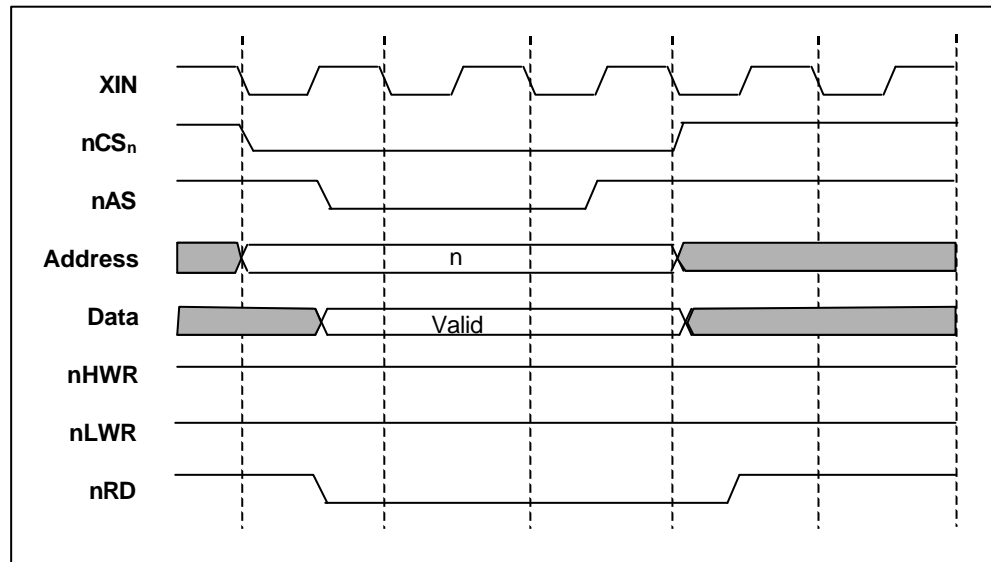
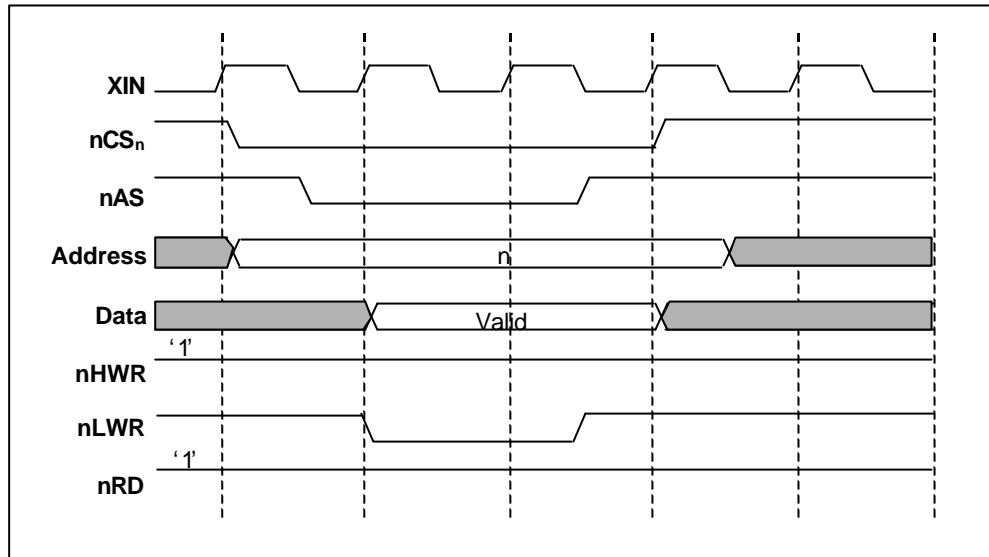
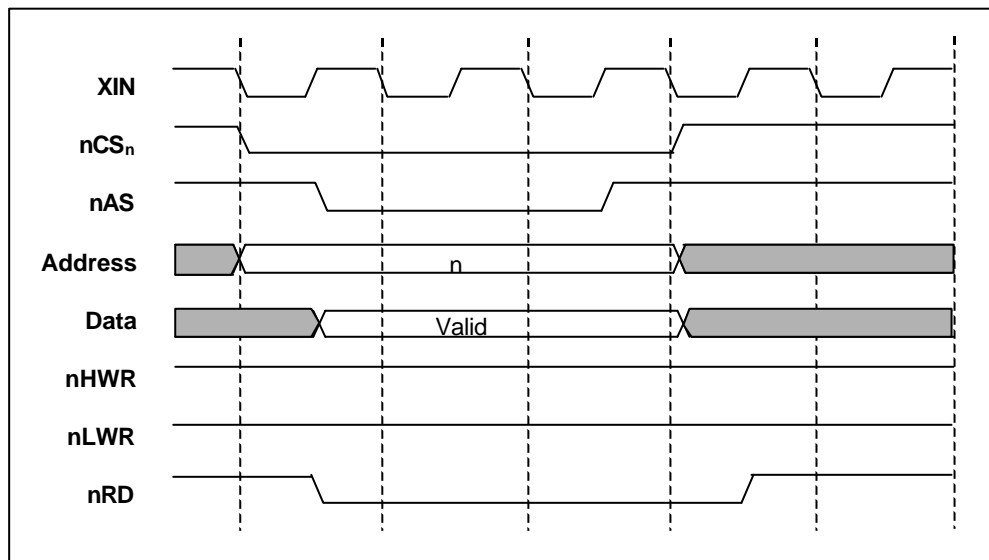


Figure 3.14 Bus Control Signal Read Timing for 16-Bit, 2-Wait (Half-Word Access)

**Figure 3.15** shows the write timing of bus control signals for a 16-Bit 2-wait-access area (In case of byte access). **Figure 3.16** shows the read timing of bus control signals for a 16-Bit 2-wait-access area (In case of byte access).



**Figure 3.15** Bus Control Signal Write Timing for 16-Bit, 2-Wait (Byte Access)



**Figure 3.16** Bus Control Signal Read Timing for 16-Bit, 2-Wait (Byte Access)

3.4.4 Wait Control

When accessing external space, the HMS39C7092 can extend the bus cycle by inserting wait states ( $T_w$ ). There are two ways of inserting wait states: (1) program wait insertion and (2) pin wait insertion using the **nWAIT** pin.

**Program Wait Insertion:** From 1 to 16 wait states can be inserted automatically between the T2 state and T3 state on an individual basis in each access space, according to the settings of NormWait bit fields in **BCR0-7**.

**Pin Wait Insertion:** When external space is accessed in this state, a program wait is first inserted. If the **nWAIT** pin is low at the falling edge of XIN in the last T2 or  $T_w$  state, another  $T_w$  state is inserted. If the **nWAIT** pin is held low,  $T_w$  states are inserted until it goes high.

**Figure 3.17** shows an example of the timing for insertion of one program wait state in 3-wait-state space.

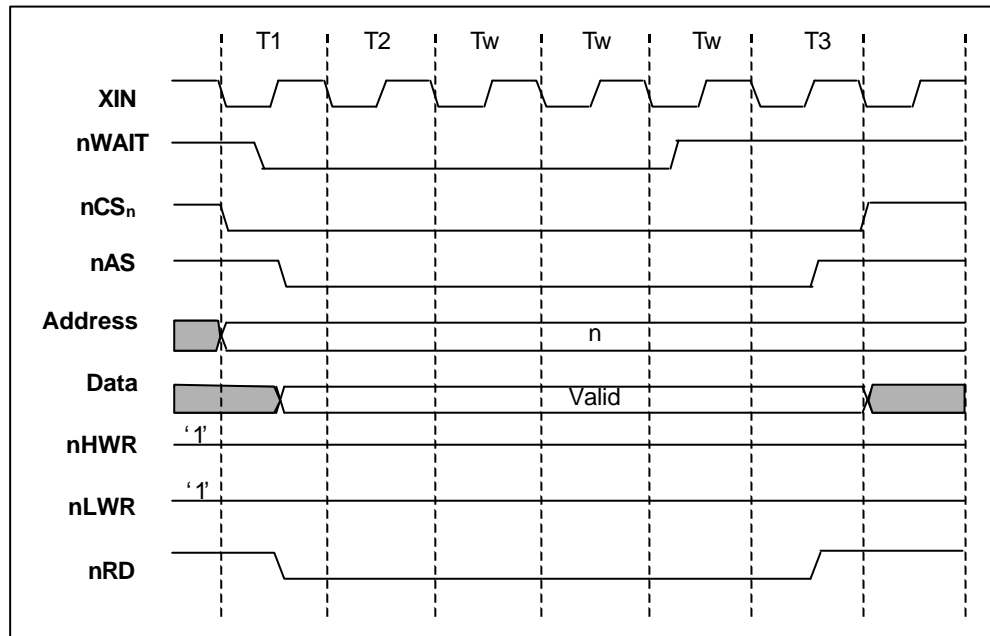


Figure 3.17 Example of Wait State Insertion Timing.

### 3.4.5 Bus Arbiter

The bus controller has a built-in bus arbiter that arbitrates between different bus masters. The bus master can be either the CPU or an external bus master. When a bus master has the bus right it can carry out read and write operations. Each bus master uses a bus request signal to request the bus right. At fixed times the bus arbiter determines priority and uses a bus acknowledge signal to grant the bus to a bus master, which can operate using the bus.

The bus arbiter checks whether the bus request signal from a bus master is active or inactive, and returns an acknowledge signal to the bus master. When two or more bus masters request the bus, the highest-priority bus master receives an acknowledge signal and can continue to use the bus until the acknowledge signal is deactivated.

The bus master priority order is:

**(High)**                      **External bus master > ARM CPU**                      **(Low)**

The bus arbiter samples the bus request signals and determines priority at all times, but it does not always grant the bus immediately, even when it receives a bus request from a bus master with higher priority than the current bus master. Each bus master has certain times at which it can release the bus to a higher-priority bus master.

**ARM CPU:** The ARM CPU is the lowest-priority bus master. If an external bus master requests the bus while the CPU has the right, the bus arbiter transfers the bus right to the bus master that requested it. The bus right is transferred at the following times:

- The bus right is transferred at the boundary of a bus cycle. If word data is accessed by two consecutive byte accesses, however, the bus right is not transferred between the two byte accesses.
- If another bus master requests the bus while the CPU is performing internal operations, such as executing a multiply or divide instruction, the bus right is transferred immediately. The CPU continues its internal operations.
- If another bus master requests the bus while the CPU is in power down mode, the bus right is transferred immediately.

**External Bus Master:** The HMS39C7092 can be always released to an external bus master. The external bus master has highest priority, and requests the bus right from the bus arbiter driving the **nBREQ** signal low. Once the external bus master acquires the bus, it keeps the bus until the **nBREQ** signal goes to high. While the bus is released to an external bus master, the HMS39C7092 chip holds the address bus, data bus, bus control signals (**nAS**, **nRD**, **nHWR**, and **nLWR**), and chip select signals (**nCS0** to **7**), and holds the **nBACK** pin in the low output state.

The bus arbiter samples the **nBREQ** pin at the rise of the system clock (**XIN**). If **nBREQ** is low, the bus is released to the external bus master at the appropriate opportunity. The **nBREQ** signal should be held low until the **nBACK** goes low.

When the **nBREQ** pin is high in two consecutive samples, the **nBACK** pin is driven high to end the bus-release cycle.

Figure 3.18 shows the timing when the bus right is requested by an external bus master during a read cycle in a 1-wait-state access area. There is a minimum interval of three states from when the **nBREQ** signal goes low until the bus is released.

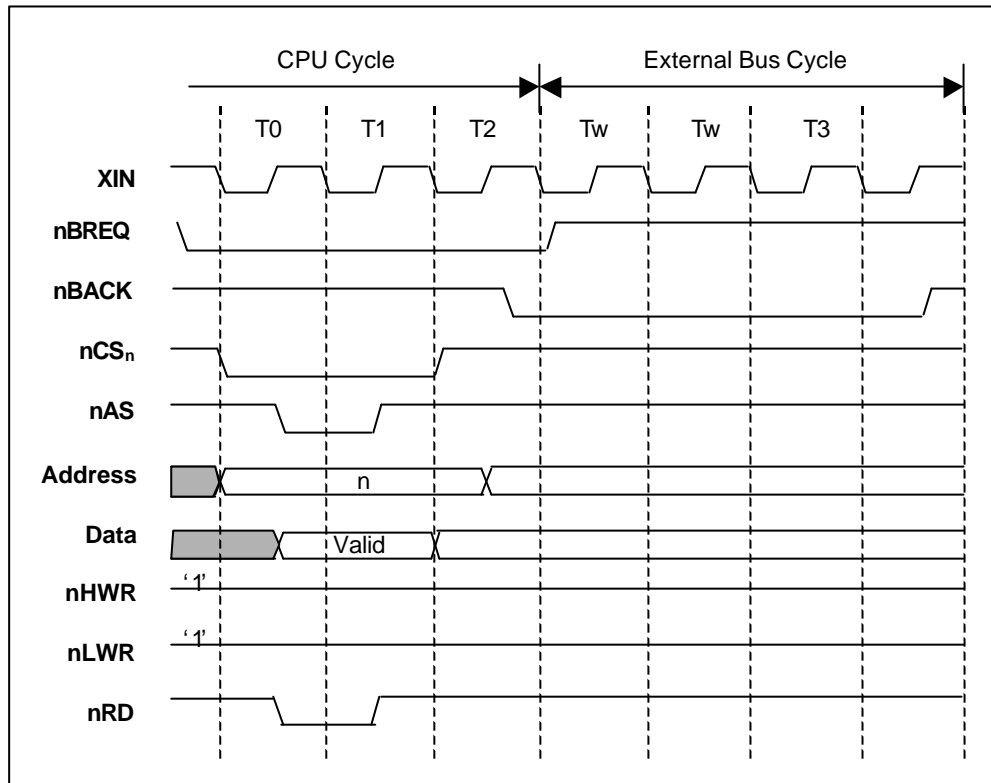


Figure 3.18 Example of External Bus Master Operation



Chapter 4  
**MCU Controller**

## 4.1 General Description

The MCU Controller (MCUC) is composed of 11 multi-function pin multiplex control signal registers and device code register.

## 4.2 Pin Function Description

Table 4.1 shows Pin function description.

**Table 4.1 Pin Function Descriptions**

NAME	Port No.	Multiplexed functions	NAME	Port No.	Multiplexed functions
Port A	PA0	TCLKA	Port 4	P40	D0
	PA1	TCLKB		P41	D1
	PA2	TCLKC, TIOCA0		P42	D2
	PA3	TCLKD, TIOCB0		P43	D3
	PA4	A23, TIOCA1		P44	D4
	PA5	A22, TIOCB1		P45	D5
	PA6	A21, TIOCA2		P46	D6
Port B	PA7	A20, TIOCB2	P47	D7	
	PB0	nCS7, TIOCA3	Port 5	P50	A16
	PB1	nCS6, TIOCB3	P51	A17	
	PB2	nCS5, TIOCA4	P52	A18	
	PB3	nCS4, TIOCB4	P53	A19	
	PB4	TMS	Port 6	P60	nWAIT
	PB5	TDO		P61	nBREQ
PB6	TDI	P62		nBACK	
PB7	TCK	P63		nAS	
Port 1	P10	A0	P64	nRD	
	P11	A1	P65	nHWR	
	P12	A2	P66	nLWR	
	P13	A3	P67	CLKO	
	P14	A4	Port 7	P70	AN0
	P15	A5		P71	AN1
	P16	A6		P72	AN2
P17	A7	P73		AN3	
Port 2	P20	A8	P74	AN4	
	P21	A9	P75	P75	
	P22	A10	P76	TIOCA5, nIRQ6	
	P23	A11	P77	TIOCB5, nIRQ7	
	P24	A12	Port 8	P80	nIRQ0
	P25	A13		P81	nCS3, nIRQ1
	P26	A14		P82	nCS2, nIRQ2
P27	A15	P83		nCS1, nIRQ3	
Port 3	P30	D8	P84	nCS0	
	P31	D9	Port 9	P90	TxD0
	P32	D10		P91	RxD0
	P33	D11		P92	TxD1
	P34	D12		P93	RxD1
	P35	D13		XP96*	XFVPPD*
	P36	D14		P97	nTRST
	P37	D15			

\*XP96/XFVPPD pin is package bonding options

Note: Each port functions are changed by Mode-setting or user definition.  
Default functions are showed in 4.3.2 PINMUX Register

### 4.3 Register Description

#### 4.3.1 Register Memory Map

**Table 4.2** is the memory map of the MCU Controller. The base address of MCU control Register is **0x0900\_0000**. **Table 4.3** shows the initial value in each mode. The initial values are different by operation mode.

**Table 4.2 Memory map of the MCU Controller**

Reg.	I/O OFFSET	Dir.	Description
PAMR	0x0000	R/W	Pin MUX Control Register for Port A
PBMR	0x0004	R/W	Pin MUX Control Register for Port B
P1MR	0x0008	R/W	Pin MUX Control Register for Port 1
P2MR	0x000C	R/W	Pin MUX Control Register for Port 2
P3MR	0x0010	R/W	Pin MUX Control Register for Port 3
P4MR	0x0014	R/W	Pin MUX Control Register for Port 4
P5MR	0x0018	R/W	Pin MUX Control Register for Port 5
P6MR	0x001C	R/W	Pin MUX Control Register for Port 6
P7MR	0x0020	R/W	Pin MUX Control Register for Port 7
P8MR	0x0024	R/W	Pin MUX Control Register for Port 8
P9MR	0x0028	R/W	Pin MUX Control Register for Port 9
DCR	0x002C	R	MCU Device Code Register

**Table 4.3 MCU Controller Initial values in each mode**

Reg.	Mode 2	Mode 3,4	MODE 5,7	MODE 6
PAMR	0x0000	0x0000	0x1540	0x1540
PBMR	0x0000	0x0000	0x0055	0x0000
P1MR	0x0000	0x0000	0x00FF	0x0000
P2MR	0x0000	0x0000	0x00FF	0x0000
P3MR	0x00FF	0x0000	0x00FF	0x0000
P4MR	0x0000	0x0000	0x00FF	0x0000
P5MR	0x0000	0x0000	0x000F	0x0000
P6MR	0x0000	0x0000	0x03FF	0x0000
P7MR	0x0000	0x0000	0x0000	0x0000
P8MR	0x0000	0x0000	0x00D4	0x0000
P9MR	0x0000	0x0000	0x0000	0x0000
DCR	0x39437092	0x39437092	0x39437092	0x39437092

4.3.2 PINMUX Register

PAMR Port A Multiplex Register (0x0900\_0000 RW)

	b31	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PAMR	Reserved		PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0						
Initial value : depend on operating mode (refer to <b>Table 4.3</b> )																
								PA7	00 : A20 1x : PA7				01	: TIOCB2		
								PA6	00 : A21 1x : PA6				01	: TIOCA2		
								PA5	00 : A22 1x : PA5				01	: TIOCB1		
								PA4	00 : A23 1x : PA4				01	: TIOCA1		
								PA3	00 : TCLKD 1x : PA3				01	: TIOCB0		
								PA2	00 : TCLKC 1x : PA2				01	: TIOCA0		
								PA1	0 : TCLKB				1	: PA1		
								PA0	0 : TCLKA				1	: PA2		

PBMR Port B Multiplex Register (0x0900\_0004 RW)

	b31	b14	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	
PBMR	Reserved			PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0				
Initial value : depend on operating mode (refer to <b>Table 4.3</b> )															
								PB7	0 : TCK			1	: PB7		
								PB6	0 : TDI			1	: PB6		
								PB5	0 : TDO			1	: PB5		
								PB4	0 : TMS			1	: PB4		
								PB3	00 : /CS4 1x : PB3			01	: TIOCB4		
								PB2	00 : /CS5 1x : PB2			01	: TIOCA4		
								PB1	00 : /CS6 1x : PB1			01	: TIOCB3		
								PB0	00 : /CS7 1x : PB0			01	: TIOCA3		

**P1MR Port 1 Multiplex Register (0x0900\_0008 RW)**

	b31	b8	b7	b6	b5	b4	b3	b2	b1	b0
P1MR	Reserved		P17	P16	P15	P14	P13	P12	P11	P10
Initial value : depend on operating mode (refer to <b>Table 4.3</b> )										
		P17	0	: A7			1	: P17		
		P16	0	: A6			1	: P16		
		P15	0	: A5			1	: P15		
		P14	0	: A4			1	: P14		
		P13	0	: A3			1	: P13		
		P12	0	: A2			1	: P12		
		P11	0	: A1			1	: P11		
		P10	0	: A0			1	: P10		

**P2MR Port B Multiplex Register (0x0900\_000C RW)**

	b31	b8	b7	b6	b5	b4	b3	b2	b1	b0
P2MR	Reserved		P27	P26	P25	P24	P23	P22	P21	P20
Initial value : depend on operating mode (refer to <b>Table 4.3</b> )										
		P27	0	: A15			1	: P27		
		P26	0	: A14			1	: P26		
		P25	0	: A13			1	: P25		
		P24	0	: A12			1	: P24		
		P23	0	: A11			1	: P23		
		P22	0	: A10			1	: P22		
		P21	0	: A9			1	: P21		
		P20	0	: A8			1	: P20		

**P3MR Port 3 Multiplex Register (0x0900\_0010 RW)**

	b31	b8	b7	b6	b5	b4	b3	b2	b1	b0
P3MR	Reserved		P37	P36	P35	P34	P33	P32	P31	P30
Initial value : depend on operating mode (refer to <b>Table 4.3</b> )										
		P37	0	: D8			1	: P37		
		P36	0	: D9			1	: P36		
		P35	0	: D10			1	: P35		
		P34	0	: D11			1	: P34		
		P33	0	: D12			1	: P33		
		P32	0	: D13			1	: P32		
		P31	0	: D14			1	: P31		
		P30	0	: D15			1	: P30		



**P7MR Port 7 Multiplex Register (0x0900\_0020 R/W)**

	b31	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
<b>P7MR</b>	Reserved			P77	P76	P74	P73	P72	P71	P70	
	Initial value : depend on operating mode (refer to <b>Table 4.3</b> )										
		P77	00	: TIOCB5			01	: /IRQ7			
			1x	: P77							
		P76	00	: TIOCA5			01	: /IRQ6			
			1x	: P76							
		P74	0	: AN4			1	: P74			
		P73	0	: AN3			1	: P73			
		P72	0	: AN2			1	: P72			
		P71	0	: AN1			1	: P71			
		P70	0	: AN0			1	: P70			

**P8MR Port 8 Multiplex Register (0x0900\_0024 R/W)**

	b31	b8	b7	b6	b5	b4	b3	b2	b1	b0	
<b>P8MR</b>	Reserved			P84	P83	P82	P81	P80			
	Initial value : depend on operating mode (refer to <b>Table 4.3</b> )										
		P84	0	: /CS0			1	: P84			
		P83	00	: /CS1			01	: /IRQ3			
			1x	: P83							
		P82	00	: /CS2			01	: /IRQ2			
			1x	: P82							
		P81	00	: /CS3			01	: /IRQ1			
			1x	: P81							
		P80	0	: /IRQ0			1	: P80			

**P9MR Port 9 Multiplex Register (0x0900\_0028 R/W)**

	b31	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
<b>P9MR</b>	Reserved			P97	P95	P94	P93	P92	P91	P90			
	Initial value : depend on operating mode (refer to <b>Table 4.3</b> )												
		P97	0	: /TRST			1	: P97					
		P95	00	: /IRQ5			01	: P95					
			1x	: Reserved									
		P94	00	: /IRQ4			01	: P94					
			1x	: Reserved									
		P93	00	: /RxD1			01	: P93					
			1x	: Reserved									
		P92	00	: /TxD1			01	: P92					
			1x	: Reserved									
		P91	0	: RxD0			1	: P91					
		P90	0	: TxD0			1	: P90					

**4.3.3 MCU Device Code Register (0x0900\_002C Read Only)**

This Register is read only. Device Code Value is ' **0x3943\_7092**'



Chapter 5  
**Power Management Unit**

5.1 General Description

The PMU block provides:

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

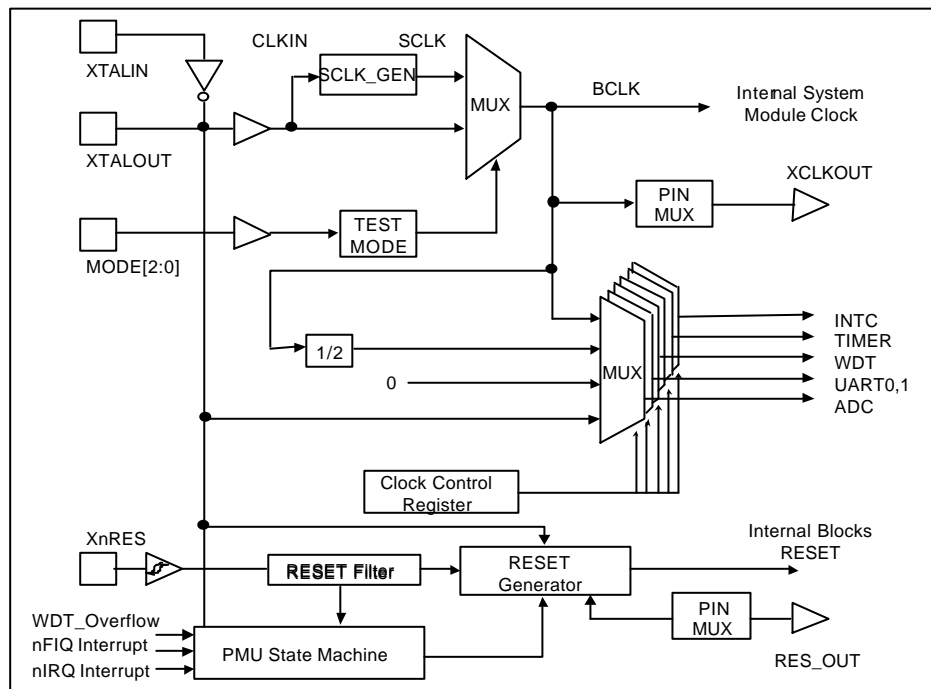


Figure 5.1 PMU Block Diagram

## 5.2 Operation Modes

### 5.2.1 Introduction

The PMU is consisted of clock controller and reset controller. User can control internal clocks those are embedded peripherals and main clock of MCU by setting the registers of PMU. The MCU has three reset sources those are external power-on reset, soft-reset of PMU, soft-reset of WDT and overflow reset of WDT. And PMU has status registers that old reset value and PMU status.

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
- Power On Reset
- Soft Reset

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

- RUN
- Power-down mode

### 5.2.2 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 soft-reset or PMU soft-reset, the MCU is initialized

#### ***Power on Reset***

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). The MCU is running after 32 clocks end of reset timing that rising edge of reset signal.

#### ***Soft- Reset of PMU***

The soft-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, Decoder.

**Overflow and Soft-Reset of Watchdog timer**

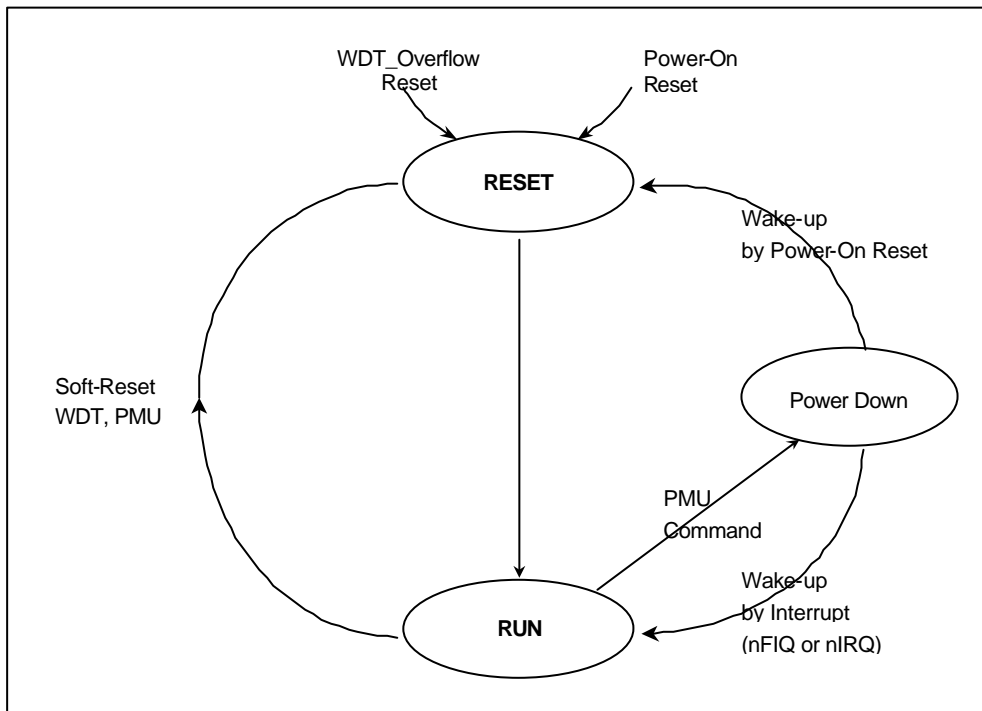
The watchdog timer can generate reset signal, when timer overflows or sets the register value. Detailed information are in the watchdog timer manual, please refer to it.

**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 interrupt controller block working in the power down mode.

**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 5.2 Reset and Power Management State Machine.**

### 5.3 Power Management Unit Register Map

The start address of the PMU(Power Management Unit) is **0x0900\_1000**.

**Table 5.1 Register Map of the PMU**

Name	I/O Offset	DIR	Description
<b>PMUCR</b>	0x1000	W	PMU operation mode controls register.
<b>PMUSR</b>	0x1000	R	PMU status register shows the just previous PMU state.
<b>PCLKCR</b>	0x1008	R/W	Peripheral clock control register.
<b>MEMSR</b>	0x100C	R	Memory remap status register.
<b>MEMCR</b>	0x1010	W	Memory remap control register
<b>RSTCR</b>	0x1030	W	Soft-Reset control register

5.4 Register Description

The PMU supplies the clock to all of the blocks in the MCU. The start address of register is **0x0900\_1000**.

**PMUCR PMU Control Register (0x0900\_1000 Write-Only)**

	b31 - b8	b7	b6	b5	b4	b3	b2	b1	b0
<b>PMUCR</b>	Reserved							PD	
Reset	-	0	0	0	0	0	0	0	0

Initial value : 0x-00

**PD** 11 : Entering the Power down Mode  
00 : Clear PMU Status Register

This register controls the operation mode of PMU. When power on reset states, register value is initialized by Run State (00). If PMUCR is 3, device enters the PD(Power Down) mode. The other values don't effect. The address of register is 0x0900\_1000.

**PMUST PMU Status Register (0x0900\_1000 Read-Only)**

	b31 - b8	b5	b4	b3	b2	b1	b0
<b>PMUST</b>	Reserved	PMUST		Reserved	PMUST		
Reset	-	0	0	-	0	0	0

Initial value : 0x-00

This register holds the previous status and reset state of PMU. The address of register is 0x0900\_1000.

**PMUST** (Previous Reset Status bits)  
00 – The Power-On reset state (nPOR)  
01 – PMU Soft-reset state  
10 – WDT Soft-reset state  
11 – WDT Overflow-reset state

**PMUST** (PMU Status bits)  
00 – Start after Power-On reset  
01 – reserved state  
10 – reserved state  
11 – Start after Power-Down mode

**PCLKCR**      **Clock Control Register (0x0900\_1008 R/W)**

	b31 - b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	B5	b4	b3	b2	b1	b0
<b>PCLKCR</b>	Reserved	WU_SEL	INTC_CC	WDT_CC			UART_Clk		UART_CC		TIMER_CC			ADC_CC			
Reset	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Initial value : 0x00000000

The address of register is 0x0900\_1008.

**WU\_SEL** : Wake-up source interrupt select register

- 0 - MCU wake-up when nFIQ interrupt occur
- 1 - MCU wake-up when nIRQ interrupt occur

**INTC\_CC** : Interrupt controller clock control register

- 0 - Interrupt controller use the XIN clock. XIN is not killed at any mode
- 1 - Interrupt controller use the BCLK of internal Bus clock. The Bus clock is killed when Power-down mode

**WDT\_CC** : Clock control register of WDT

- 000 - BCLK
- 001 - BCLK/2
- 010 ~ 111 – Reserved

**UART\_Clk** : UART0,1 clocks on-off control register.

- 00 - UART0,1 clocks ON
- 01 - UART1 clock ON, UART0 clock OFF
- 10 - UART1 clock OFF, UART0 clock ON
- 11 – UART0,1 clocks OFF

**UART\_CC** : Clocks Control register of UART.

- 000 - BCLK
- 001 - BCLK/2
- 010 ~ 111 – Reserved

**TIMER\_CC** : Clocks Control register of TIMER.

- 000 - BCLK
- 001 - BCLK/2
- 010 ~ 111 – Reserved

**ADC\_CC** : Clocks Control register of ADC.

Values are same as WDT\_CC

**MEMCR** Memory map Control Register (0x0900\_1010 Write-Only)  
**MEMSR** Memory map Status Register (0x0900\_100C Read-Only)

	b31 - b3	b2	b1	b0
<b>MEMCR</b>	Reserved	SM	On-Flash	REMAP
<b>MEMSR</b>	-	0	0	0
Reset	-	0	0	0

Initial value : 0x-0

In write operation, the address of register is 0x0900\_1010 and in read operation, the address of register is 0x0900\_100C.

**SM** : External bus controller mapping change.  
 0 - Each nCS0 ~ nCS7 of address space is 16MB size  
 1 - Each nCS0 ~ nCS7 of address space is 1MB size

**On-Flash** : Re-mapping of Flash start address to 0x0 in MODE 6 and 7.  
 0 - Default value.  
 1 - Re-mapping of Flash start address to 0x0 in the memory map. It is valid at MODE 6 and 7.

**REMAP** : Re-map internal SRAM address location.  
 0 - Default value.  
 1 - Re-mapping of internal SRAM start address to 0x0 in the memory map. It is used at MODE 2,3,4,5,6 and 7.

**RSTCR** Soft-Reset Control Register (0x0900\_1030 R/W)

	b31 - b1	b0
<b>RSTCR</b>	Reserved	RSTCR
Reset	-	0

Initial value : 0x-0

**RSTCR** 1 : Normal reset  
 0 : Normal

This register is used for generating the Soft-reset operation. The MCU is entered in reset state, when this register is set to high, it is cleared automatically at the end of Soft-Reset procedure. The address is 0x0900\_1030.



## 5.5 Signal Timing Diagram

The PMU signal timing is as shown below.

### 5.5.1 Power on Reset

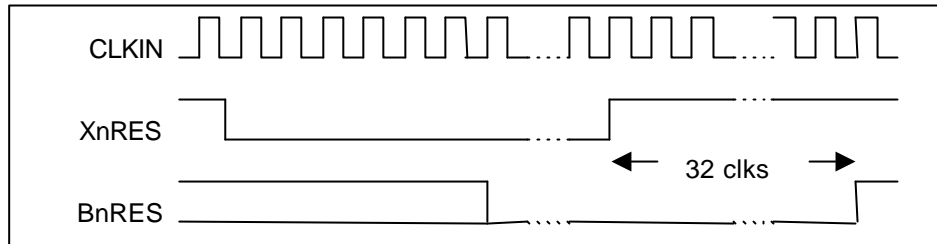


Figure 5.3 Power on Reset Timing Diagram

### 5.5.2 Watch Dog Timer Overflow

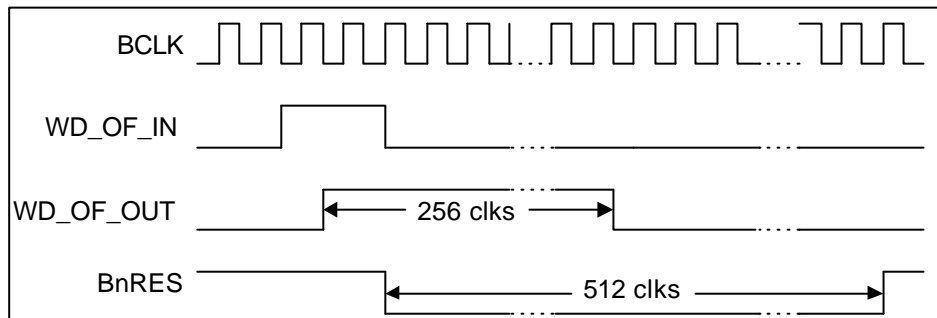


Figure 5.4 Watch Dog Timer Overflow Timing Diagram

5.5.3 Soft-Reset

There are two Soft-Reset cases. The first Soft-Reset operation is switched by MAN\_RST signal from WDT. Another case is from PMU reset control register.

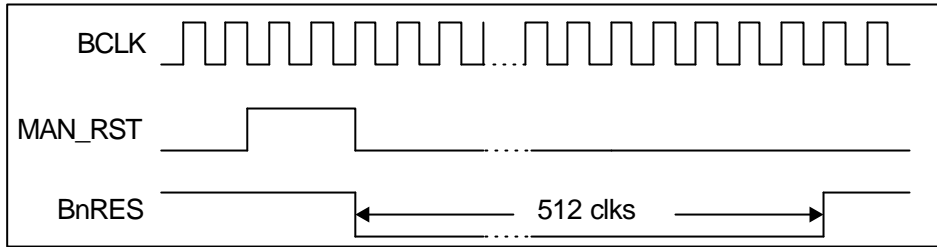


Figure 5.5 Soft Reset (from WDT) Timing Diagram

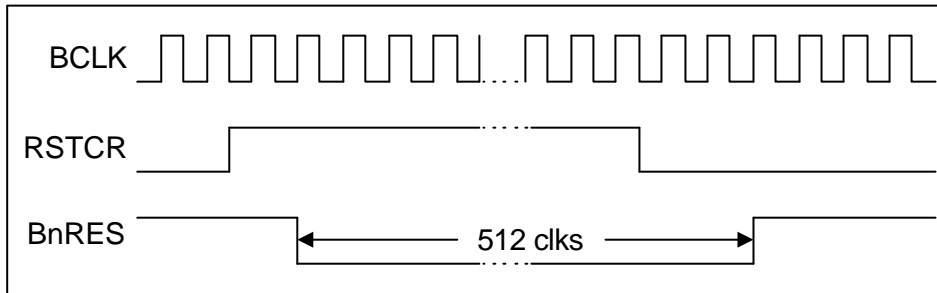


Figure 5.6 Soft Reset (from PMU) Timing Diagram

Chapter 6  
**The Interrupt Controller**

6.1 About the Interrupt controller

The interrupt controller has the following features :

- Asynchronous interrupt controller
- 8 external interrupt sources
- 13 internal interrupt sources
- Low interrupts latency
- Selection of the active modes of all interrupts source inputs  
(Level or Edge trigger)
- Mask-able for each interrupt source and output signal
- Selection of the output paths (IRQ or FIQ for each interrupt source)

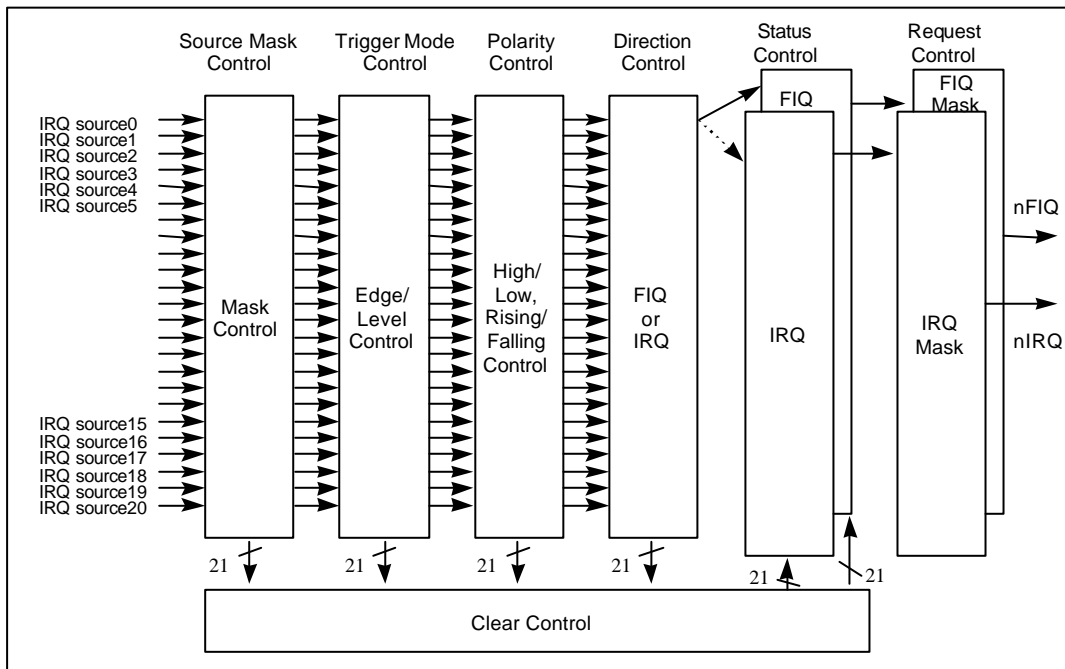


Figure 6.1 Interrupt Control Flow Diagram

### 6.1.1 Interrupt sources

The interrupt controller provides interface between multiple interrupt sources and the processor. The interrupt controller supports internal and external interrupt sources. Internally there are 11 peripheral interrupt sources. Externally there are 8 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 6.1 Interrupt Controller Default Setting Value**

Interrupt No.	INTERRUPT SOURCES
INT0	External Interrupt 0
INT1	External Interrupt 1
INT2	External Interrupt 2
INT3	External Interrupt 3
INT4	External Interrupt 4
INT5	External Interrupt 5
INT6	External Interrupt 6
INT7	External Interrupt 7
INT8	reserved
INT9	reserved
INT10	WDT
INT11	UART0
INT12	UART1
INT13	ADC
INT14	Timer 0
INT15	Timer 1
INT16	Timer 2
INT17	Timer 3
INT18	Timer 4
INT19	Timer 5
INT20	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 providing 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.

### 6.1.2 Interrupt Control

The interrupt controller provides the interrupt source status and the 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-level 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 modes are configurable 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 in the edge trigger mode only.

Bit 20 is used as a software interrupt source. When source mask control register bit 20 is HIGH, an interrupt request occurs. To disable the software interrupt, Source Mask Control Register bit 20 should be Low. Software interrupt source input is fixed active HIGH and level sensitive.

## 6.2 Interrupt Controller Registers

The start address of the interrupt controller is **0x0900\_1200**. The offset of any particular register from the start address is fixed. The following registers are provided for both FIQ and IRQ interrupt controllers:

**Table 6.2 Memory Map of the Interrupt Controller**

REG.	I/O OFFSET	Dir	Description
<b>GMR</b>	0x1200	R/W	Global Mask Register
<b>TMR</b>	0x1204	R/W	Trigger Mode Register
<b>TPR</b>	0x1208	R/W	Trigger Polarity Register
<b>IDR</b>	0x120C	R/W	Interrupt Direction Register
<b>FSR</b>	0x1210	R	FIQ Status Register
<b>ISR</b>	0x1214	R	IRQ Status Register
<b>FMR</b>	0x1218	R/W	FIQ Mask Register
<b>IMR</b>	0x121C	R/W	IRQ Mask Register
<b>ISCR</b>	0x1220	W	Interrupt Status Clear Register

### GMR Global Mask Register (0x0900\_1200 R/W)

		b31	-	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
<b>GMR</b>	Reserved	I24	I23	I22	I21	I20	I19	I18	I17	I16	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0			
Reset	0000000	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	1	1	1
		Initial value : 0x01FFFFFF										Bit field I0-I24						1 : Mask 0 : Unmask											

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.

### TMR Trigger Mode Register (0x0900\_1204 R/W)

		b31	-	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
<b>TMR</b>	Reserved	I24	I23	I22	I21	I20	I19	I18	I17	I16	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0			
Reset	0000000	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	0	0	0
		Initial value : 0x0100000										Bit field I0-I24						1 : Level Trigger Mode 0 : Edge Trigger Mode											

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.

**TPR Trigger Polarity Register (0x0900\_1208 R/W)**

b31 - b25 b24b23b22b21b20b19b18b17b16b15b14b13b12b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0

<b>TPR</b>	Reserved	I24	I23	I22	I21	I20	I19	I18	I17	I16	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
Reset	0000000	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

Initial value : 0x01000000      Bit field I0-I24      1 : High/Rising Edge  
 0 : Low/Falling Edge

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.

**Table 6.3 Interrupt Source Trigger Mode**

DETECTION MODE	TMR	TPR
<i>Falling-Edge (Default)</i>	0	0
Rising-Edge	0	1
Low-Level	1	0
High-Level	1	1

**IDR Interrupt Direction Register (0x0900\_120C R/W)**

b31 - b25 b24b23b22b21b20b19b18b17b16b15b14b13b12b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0

<b>IDR</b>	Reserved	I24	I23	I22	I21	I20	I19	I18	I17	I16	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
Reset	0000000	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

Initial value : 0x00000000      Bit field I0-I24      1 : Direction to FIQ  
 0 : Direction to IRQ

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.



**FSR**                      **FIQ Status Register (0x0900\_1210 Read Only)**

	b31	-	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
<b>FSR</b>	Reserved	I24	I23	I22	I21	I20	I19	I18	I17	I16	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0		
Reset	0000000	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	0

Initial value : 0x00000000                      Bit field I0-I24                      1 : FIQ Pending  
0 : FIQ Idle

The FIQ status register is used to reflect the status of all channels set to produce an FIQ interrupt ( $IDR_n = 1$ ). 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.

**ISR**                      **IRQ Status Register (0x0900\_1214 Read Only)**

	b31	-	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
<b>ISR</b>	Reserved	I24	I23	I22	I21	I20	I19	I18	I17	I16	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0		
Reset	0000000	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	0

Initial value : 0x00000000                      Bit field I0-I24                      1 : IRQ Pending  
0 : IRQ Idle

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.

**FMR**                      **FIQ Mask Register (0x0900\_1218 R/W)**

	b31	-	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
<b>FMR</b>	Reserved	I24	I23	I22	I21	I20	I19	I18	I17	I16	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0		
Reset	0000000	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	0

Initial value : 0x00000000                      Bit field I0-I24                      1 : Disable FIQ  
0 : Enable FIQ

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.



Chapter 7  
**Watchdog Timer**

7.1 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

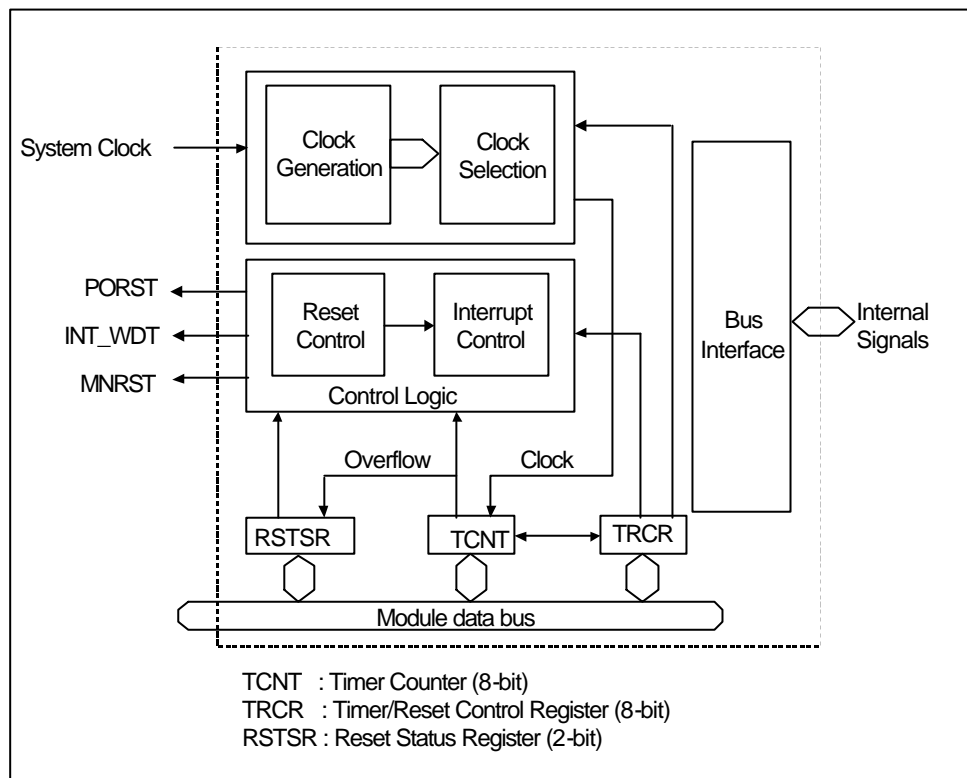


Figure 7.1 Watchdog Timer Module Block Diagram

## 7.2 Watchdog Timer Introduction

The HMS39C7092 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 produces eight counter clock sources. The clock signals are obtained by dividing the frequency of the system clock. Users can select one of eight internal clock sources for input to the **WTCNT** by CKS2 - CKS0 in the **WTCR**.

### 7.3 Watchdog Timer Operation

#### The Watchdog Timer Mode

To use the WDT as a watchdog timer, set the WT/nIT and TMEN bits of the WTCR to 1. Software must prevent WTCNT overflow by rewriting the TCNT value (normally by writing 0x00) before overflow occurs. If the WTCNT fails to be rewritten and overflow occurs 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).

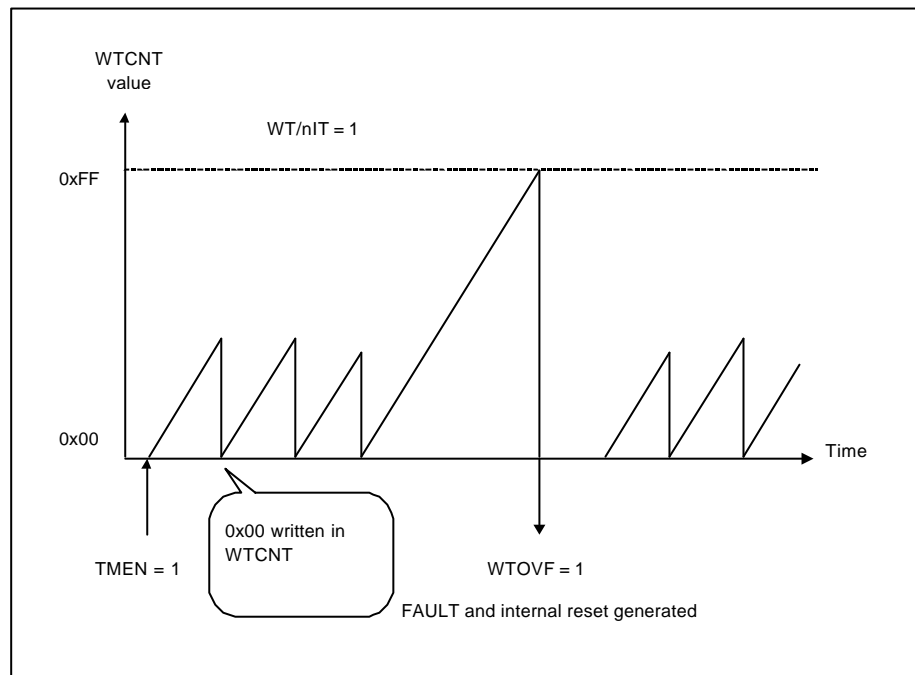
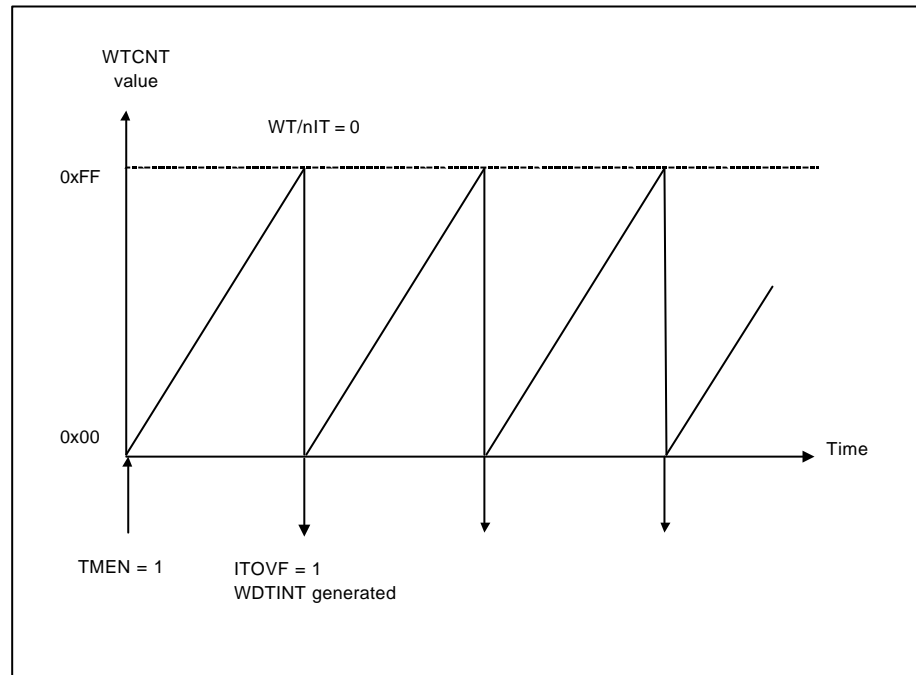


Figure 7.2 Operation in the Watchdog Timer Mode

If the RSTEN bit in the WTCR 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 7.3** Operation in the Interval Timer Mode

### 7.3.1 Timing of Setting and Clearing the Overflow Flag

#### Timing of setting the overflow flag

In the interval timer mode when the WTCNT overflows, the ITOVF flag is set to 1 and watchdog timer interrupt (**INT\_WDT**) is requested.

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

#### Timing of clearing the overflow flag

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

7.4 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 start address of the watchdog timer is fixed to **0x0900\_1100** and the offset of any particular register from the base address is fixed.

**Table 7.1 Memory Map of the Watchdog Timer APB Peripheral**

Name	I/O Offset	DIR	Description
<b>WTCR</b>	0x1100	R/W	WDT control. (8-bit)
<b>WRSR</b>	0x1104	R	WDT Reset status reg. (2-bit)
<b>WTCNT</b>	0x1108	R/W	WDT Timer counter. (8-bit)



## 7.5 Watchdog Timer Register Descriptions

The following registers are provided for watchdog timer:

WTCR		Watchdog Timer Control Register ( 0x0900_1100 R/W )								
		b31 - b8	b7	b6	b5	b4	B3	b2	b1	b0
<b>WTCR</b>		Reserved	INTEN	WT/nIT	TMEN	RSTEN	RSTSEL	CKSEL		
Reset		-	0	0	0	0	0	0	0	0

Initial value : 0x-00

**CKSEL** : Clock select. Select one of eight internal clock sources for input to the **WTCNT**.

- 000 – BCLK / 2
- 001 – BCLK / 8
- 010 – BCLK / 32
- 011 – BCLK / 64
- 100 – BCLK / 256
- 101 – BCLK / 512
- 110 – BCLK / 2048
- 111 – BCLK / 8192

**RSTSEL** : Reset select register. Select the type of generated internal reset if the **WTCNT** overflows in the watchdog timer mode.

- 0 - Power-on reset
- 1 - Manual reset

**RSTEN** : Reset enable register. Select whether to reset the chip internally or not if the **WTCNT** overflows in the watchdog timer mode.

- 0 - Disable
- 1 - Enable

**TMEN** : Timer enable register. Enable or disable the timer.

- 0 - Disable
- 1 - Enable.

**WT/nIT** : Timer mode select register. Select whether to use the WDT as a watchdog timer or interval timer.

- 0 - Interval timer mode
- 1 - Watchdog timer mode

**INTEN** : Interrupt enable register. Enable or disable the interrupt request, INT\_WDT.

- 0 - Disable
- 1 - Enable

8-bit readable and writable register. The start address of register is 0x0900\_1100. The following functions are provided :

- Selecting the timer mode
- Selecting the internal clock source

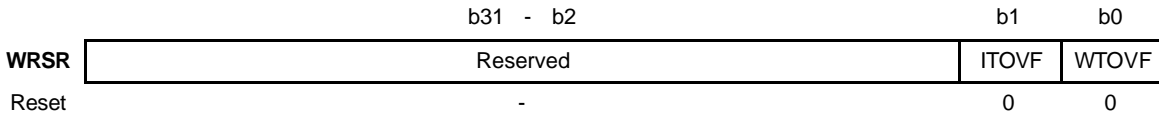
- Selecting the reset mode
- Setting the timer enable bit
- Being enable interrupt request
- Being enable reset signal occurrence

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

**Table 7.2 Internal Counter Clock Sources (SYSCLK = 40 MHz)**

CKSEL	CLOCK SOURCE	OVERFLOW INTERVAL
000	SYSCLK / 2	12.8 us
001	SYSCLK / 8	51.2 us
010	SYSCLK / 32	204.8 us
011	SYSCLK / 64	409.6 us
100	SYSCLK / 256	1.64 ms
101	SYSCLK / 512	3.28 ms
110	SYSCLK / 2048	13.11 ms
111	SYSCLK / 8192	52.43 ms

**WRSR Reset Status Register ( 0x0900\_1104 Read-Only )**



Initial value : 0x-0

1 : Overflowed  
0 : Normal

**WTOVF** : Watchdog timer overflow flag. Indicates that the **WTCNT** has overflowed in the watchdog timer mode.

**ITOVF** : Interval timer overflow flag. Indicates that the **WTCNT** has overflowed in the interval timer mode.

Two-bit read only register. The **WRSR** indicates whether **WTCNT** is overflowed or not. The **WRSR** is initialized to 0x0 by the reset signal, **nB\_RES**.

Bit 0 (**WTOVF**) indicates that the **WTCNT** has overflowed in the watchdog timer mode. Bit 1 (**ITOVF**) indicates that the **WTCNT** has overflowed in the interval timer mode.

**WTCNT**      **Watchdog Timer Counter ( 0x0900\_1108 R/W )**

	b31 - b8	b7	b6	b5	b4	b3	b2	b1	b0
<b>WTCNT</b>	Reserved	17	16	15	14	13	12	11	10
Reset	-	0	0	0	0	0	0	0	0
	Initial value : 0x-00			Bit field 10-17					

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 **WTCNT** changes from 0xFF-0x00(overflows), a watchdog timer overflow signal is generated in the both timer modes. The **WTCNT** is initialized to 0x00 by a power-reset (**nB\_RES**).

7.6 Examples of Register Setting

7.6.1 Interval Timer Mode

WTCNT = 0x00  
 WTCR = 0xA0

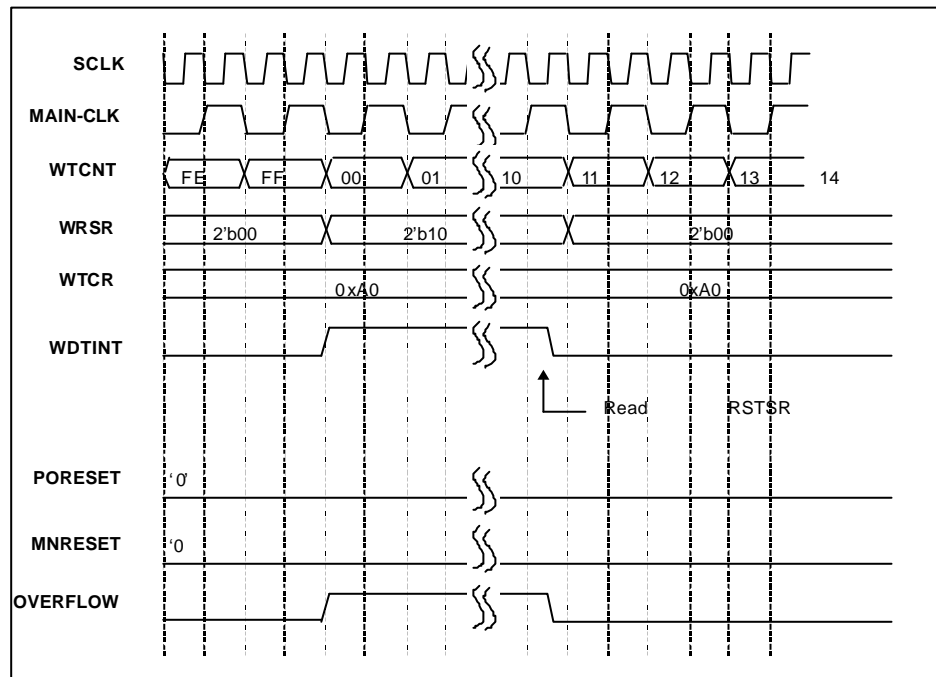


Figure 7.4 Interrupt Clear in the Interval Timer Mode

## 7.6.2 Watchdog Timer Mode with Internal Reset Disable

WTCNT = 0x00 (normally)  
WTCR = 0xE0

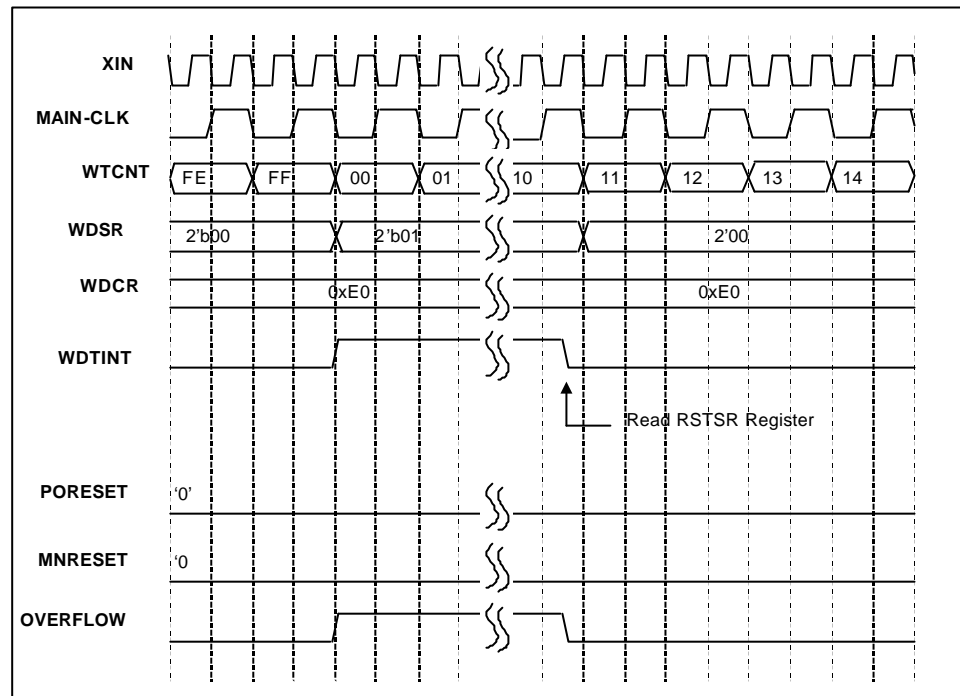


Figure 7.5 Interrupt Clear in the Watchdog Timer Mode with Reset Disable

7.6.3 Watchdog Timer Mode with Power-on Reset

WTCNT = 0x00  
 WTCR = 0xF0

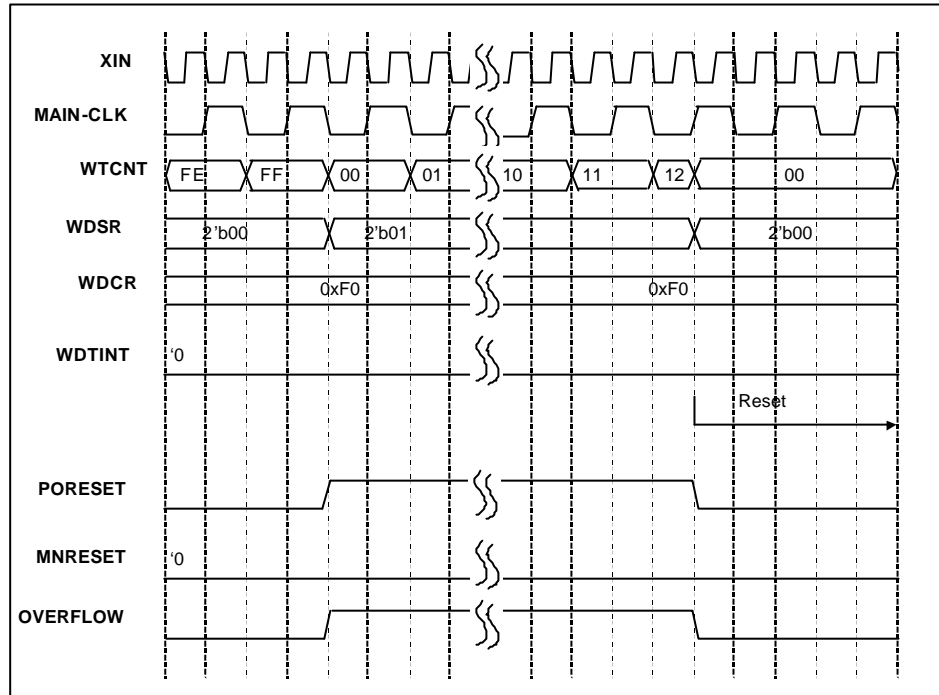


Figure 7.6 Interrupt Clear in the Watchdog Timer Mode with Power-on Reset

## 7.6.4 Watchdog Timer Mode with Manual Reset

WTCNT = 0x00  
WTCR = 0xF8

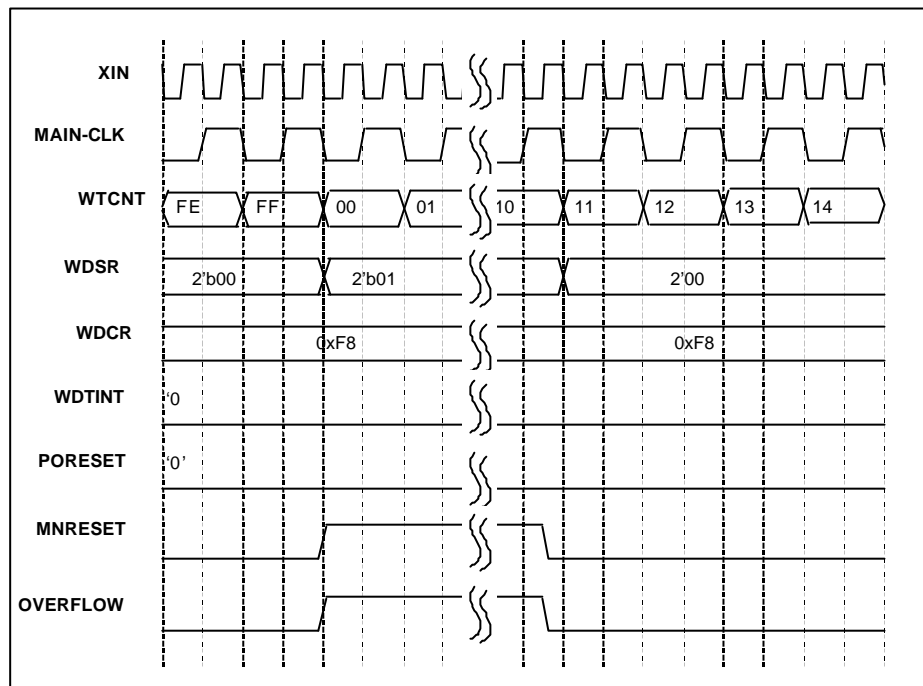


Figure 7.7 Interrupt Clear in the Watchdog Timer Mode with Manual Reset





Chapter 8

**The General Purpose Timer**

8.1 About the General Purpose Timer Unit

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

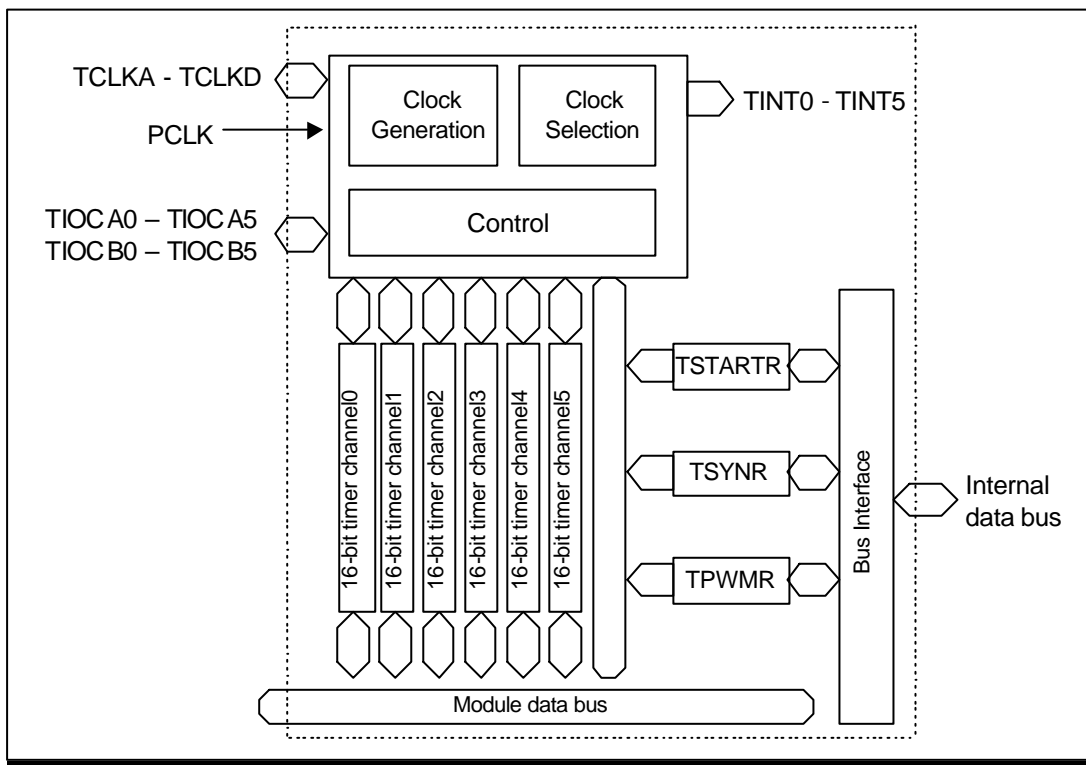


Figure 8.1 General-purpose Timer Unit Module Block Diagram

### 8.1.1 General Purpose Timer Unit Introduction

The HMS39C7092 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 falling edge.

There are five particular operation mode which can be configured respectively. The operation modes are described below.

- Free Running Mode
- Compare Match Mode
- Input Capture Mode
- Synchronized Clear and Write Mode
- PWM(Pulse Width Modulation) Mode

And 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

8.2 General Purpose Timer Unit Memory Map

8.2.1 Register Assignment

The base address of the general-purpose timer unit is **0x0900\_1300** and the offset of any particular register from the base address is fixed.

**Table 8.1 Timer Global Control Register Map**

REG.	I/O OFFSET	DIR.	DESCRIPTION
TSTARTR	0x1300	R/W	Timer Start Register
TSYNCR	0x1304	R/W	Timer Sync. Register
TPWMR	0x1308	R/W	Timer PWM Mode Register
-	0x130C	W	(test only)
-	0x1310	R	(test only)
-	0x1314	W	(test only)
-	0x1318	R	(test only)

**Table 8.2 Timer Channel Control Register Map**

REG.	I/O OFFSET	DIR.	DESCRIPTION
TCR0	0x1320	R/W	Timer 0 Control Register
TIOCR0	0x1324	R/W	Timer 0 I/O Control Register
TIER0	0x1328	R/W	Timer 0 Interrupt Enable Register
TSR0	0x132C	R	Timer 0 Interrupt Status Register
TCNT0	0x1330	R/W	Timer 0 Counter Register
GRA0	0x1334	R/W	Timer 0 General Register A
GRB0	0x1338	R/W	Timer 0 General Register B

GP Timer Unit has consists of six unit timer channels and each address starts at following address

**Table 8.3 Timer Channel Starting Address**

Timer No.	Starting Offsets
Timer 0	0x1320
Timer 1	0x1340
Timer 2	0x1360
Timer 3	0x1380
Timer 4	0x13A0
Timer 5	0x13D0

### 8.2.2 General Purpose Timer Unit Register Descriptions

The base address of the general-purpose timer unit is **0x0900\_1300**. The following registers are provided for general purpose timer unit :

#### 8.2.2.1 Timer Global Control Registers

##### TSTARTR Timer Start Register (0x0900\_1300 R/W)

	b31	b8	b7	b6	b5	b4	b3	b2	b1	b0
TSTARTR	Reserved	res	res	STR5	STR4	STR3	STR2	STR1	STR0	
Reset	-	1	1	0	0	0	0	0	0	0

Initial value : 0xXXXXXXC0

**STR<sub>n</sub>** 1 : Start Timer Channel n  
0 : Stop Timer Channel n

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

##### TSYNCR Timer Sync. Register (0x0900\_1304 R/W)

	b31	b8	b7	b6	b5	b4	b3	b2	b1	b0
TSYNCR	Reserved	res	res	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0	
Reset	-	1	1	0	0	0	0	0	0	0

Initial value : 0xXXXXXXC0

**SYNC<sub>n</sub>** 1 : Operate Synchronously with other sync. channel  
0 : Independent Counting

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

##### TPWMR Timer PWM Mode Register (0x0900\_1308 R/W)

	b31	b8	b7	b6	b5	b4	b3	b2	b1	b0
TPWMR	Reserved	res	res	PWM5	PWM4	PWM3	PWM2	PWM1	PWM0	
Reset	-	1	1	0	0	0	0	0	0	0

Initial value : 0xXXXXXXC0

**PWM<sub>n</sub>** 1 : PWM Mode  
0 : Counter Mode

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

8.2.2.2 Timer Channel Control Registers

TCR0

**Timer 0 Control Register (0x0900\_1320 R/W)**

*0x1340 for Timer 1, 0x1360 for Timer 2, 0x1380 for Timer 3, 0x13A0 for Timer 4, 0x13D0 for Timer 5*

	b31	b8	b7	b6	b5	b4	b3	b2	b1	b0
<b>TCR0</b>	Reserved		res	CCLR		res	res	TPSC		
Reset	-		1	00		1	1	000		

Initial value : 0xXXXXXX98

**CCLR** Select the Counter clear condition  
 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

**TPSC** Select the Count clock Source  
 000 : BCLK/2  
 001 : BCLK/4  
 010 : BCLK/16  
 011 : BCLK/64  
 100 : Ext ClkA  
 101 : Ext ClkB  
 110 : Ext ClkC  
 111 : Ext ClkD

8-bit readable and writable register for each channel that selects the timer counter clock source and the counter clear source.

**TIOCR0****Timer 0 I/O Control Register (0x0900\_1324 R/W)***0x1344 for Timer 1, 0x1364 for Timer 2, 0x1384 for Timer 3, 0x13A4 for Timer 4, 0x13D4 for Timer 5*

	b31	b8	b7	b6	b5	b4	b3	b2	b1	b0
<b>TIOCR0</b>	Reserved		res	IOB			res	IOA		
Reset	-	1		00		1	1		000	

Initial value : 0xXXXXXX88

**IOB** Select GRB Function

000 : compare match with pin output disable  
 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  
 111 : Don' t care

**IOA** Select GRA Function

000 : compare match with pin output disable  
 001 : 0 output at GRA compare match  
 010 : 1 output at GRA compare match  
 011 : toggle output at GRA compare match  
 100 : GRA captures the rising edge of input  
 101 : GRA captures the falling edge of input  
 110 : GRA captures both edge of input  
 111 : Don' t care

8-bit readable and writable register that selects the output compare or input capture function for GRA and GRB, and selects the function of the **TIOCA<sub>n</sub>** and **TIOCB<sub>n</sub>** pins. **TIOCR<sub>n</sub>** controls the GRA and GRB.

**TIER0**

**Timer 0 Interrupt Enable Register (0x0900\_1328 R/W)**

*0x1348 for Timer 1, 0x1368 for Timer 2, 0x1388 for Timer 3, 0x13A8 for Timer 4, 0x13D8 for Timer 5*

	b31	b8	b7	b6	b5	b4	b3	b2	b1	b0
<b>TIER0</b>	Reserved	Res	res	res	res	res	res	OVFIE	MCIBE	MCIAE
Reset	-	1	1	1	1	1	1	0	0	0

Initial value : 0xXXXXXXF8

- OVFIE** 0 : Disable Overflow Interrupt  
1 : Enable Overflow Interrupt
- MCIBE** 0 : Disable GRB Match or GRB capture Interrupt  
1 : Enable GRB Match or GRB capture Interrupt
- MCIAE** 0 : Disable GRA Match or GRA capture Interrupt  
1 : Enable GRA Match or GRA capture Interrupt

8-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 $n$  controls the interrupt enable/disable.

**TSR0**

**Timer 0 Status Register (0x0900\_132C Read Only)**

*0x134C for Timer 1, 0x136C for Timer 2, 0x138C for Timer 3, 0x13AC for Timer 4, 0x13DC for Timer 5*

	b31	b8	b7	b6	b5	b4	b3	b2	b1	b0
<b>TSR0</b>	Reserved	res	res	res	res	res	res	OVFI	MCIB	MCIA
Reset	-	1	1	1	1	1	1	0	0	0

Initial value : 0xXXXXXXF8

- OVFI** 0 : no overflow occurs  
1 : Overflow occurs
- MCIB** 0 : no GRB Match or Capture occurs  
1 : GRB Match or Capture occurs
- MCIA** 0 : no GRA Match or Capture occurs  
1 : GRA Match or Capture occurs

8-bit readable register contains the flags that indicate TCNT overflow and GRA/GRB compare match or input capture. These flags are interrupt sources.



**TCNT0****Timer 0 Counter (0x0900\_1330 R/W)***0x1350 for Timer 1, 0x1370 for Timer 2, 0x1390 for Timer 3, 0x13B0 for Timer 4, 0x13E0 for Timer 5*

	b31	b16	B15	b14	b13	b12	b11	b10	b9	b8	B7	b6	b5	b4	b3	b2	b1	b0
<b>TCNT0</b>	Reserved		TCNT															
Reset	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Initial value : 0xXXXX0000

**TCNT**    16bit Counter Value

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

**General Register A,B**

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.

**GRA0****General Register A (0x0900\_1334 R/W)***0x1354 for Timer 1, 0x1374 for Timer 2, 0x1394 for Timer 3, 0x13B4 for Timer 4, 0x13E4 for Timer 5*

	b31	b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
<b>GRA0</b>	Reserved		GRA															
Reset	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Initial value : 0xXXXX0000

**GRA**    16bit Compare Match A Value

**GRB0****General Register B (0x0900\_1338 R/W)***0x1358 for Timer 1, 0x1378 for Timer 2, 0x1398 for Timer 3, 0x13B8 for Timer 4, 0x13E8 for Timer 5*

	b31	b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
<b>GRB0</b>	Reserved		GRB															
Reset	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Initial value : 0xXXXX0000

**GRB**    16bit Compare Match B Value

**8.3 General Purpose Timer Unit Operation**

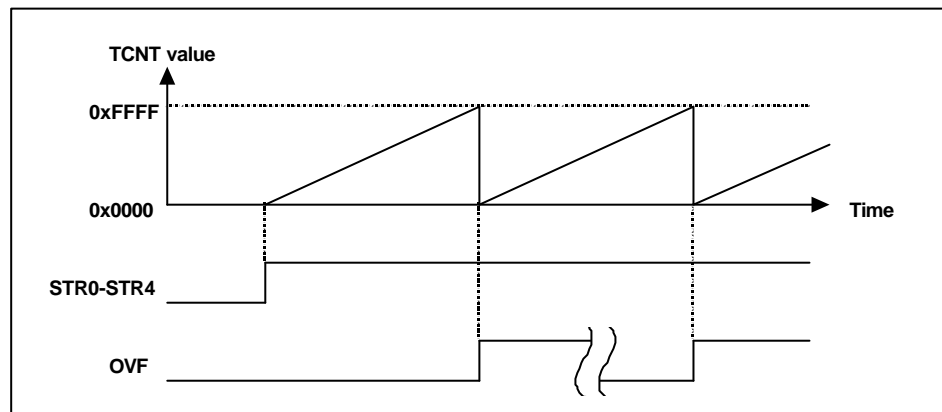
There are five particular operation mode which can be configured respectively. The operation modes are described below.

- Free Running Mode
- Compare Match Mode
- Input Capture Mode
- Synchronized Clear and Write Mode
- PWM(Pulse Width Modulation) Mode

### 8.3.1 Free Running Mode

A reset of the counters for channels 0 - 5 leaves them all in the free-running mode. When a corresponding bit in the TSR is set to 1, the corresponding timer counter operates as a free-running counter and begins to increment. When the count wraps round from 0xFFFF - 0x0000, the overflow flag (OVFI) in the timer status register (TSR) is set to 1. If the OVFI bit in the timer's corresponding interrupt enable register (TIER) is set to 1, the CPU will be asked for an interrupt. After the TCNT overflows, counting continues from 0x0000. **Figure 8.2** shows an example of free-running counting.

Periodic counter operation is obtained for a given channel's TCNT by selecting compare match as a TCNT clear source. (Set the GRA or GRB for period setting to output compare register and select counter clear upon compare match using the CCLR1 and CCLR0 bits of the timer control register (TCR). After setting, the TCNT begins incrementing as a periodic counter when the corresponding bit of TSTARTR is set to 1. When the count matches GRA or GRB, the MCIA/MCIB bit in the TSR is set to 1 and the counter is automatically cleared to 0x0000. If the MCIAE/MCIBE bit of the corresponding TIER is set to 1 at this point, the CPU will be asked for an interrupt. After the compare match, TCNT continues counting from 0x0000. **Figure 8.3** shows an example of periodic counting.



**Figure 8.2 Free-Running Counter Operation**

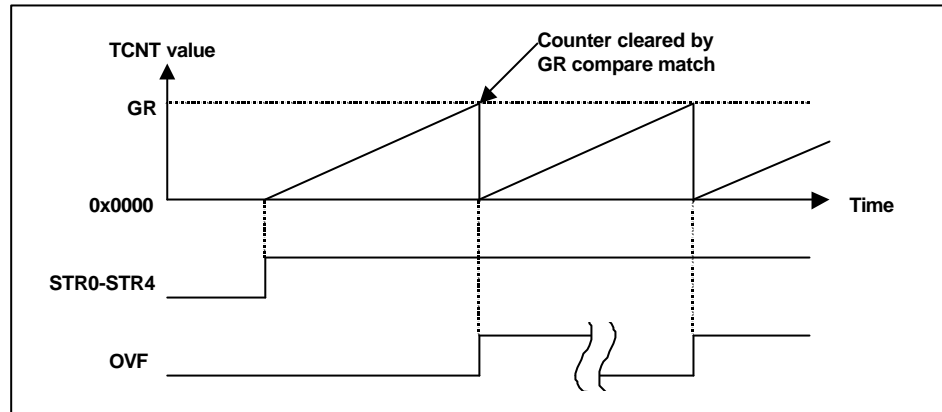


Figure 8.3 Periodic Counter Operation

### 8.3.2 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.

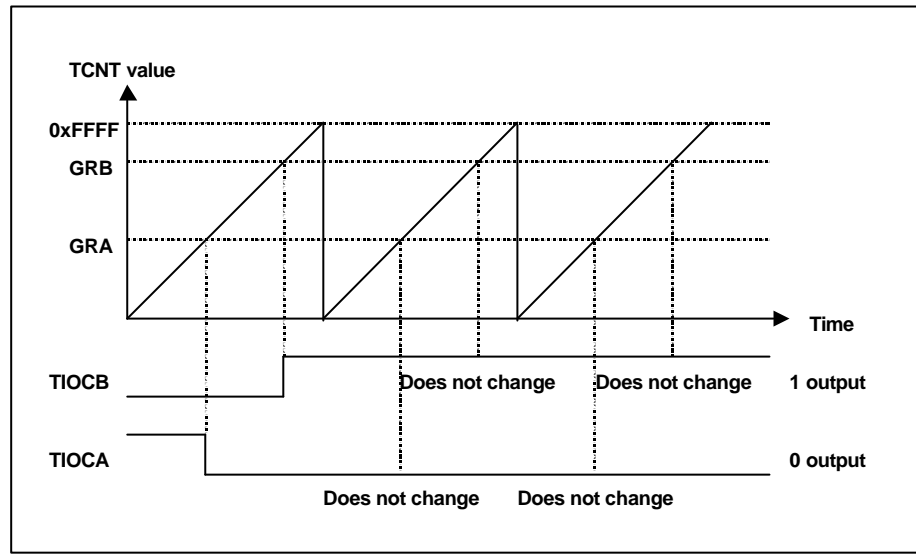


Figure 8.4 Example of 0 Output/1 Output

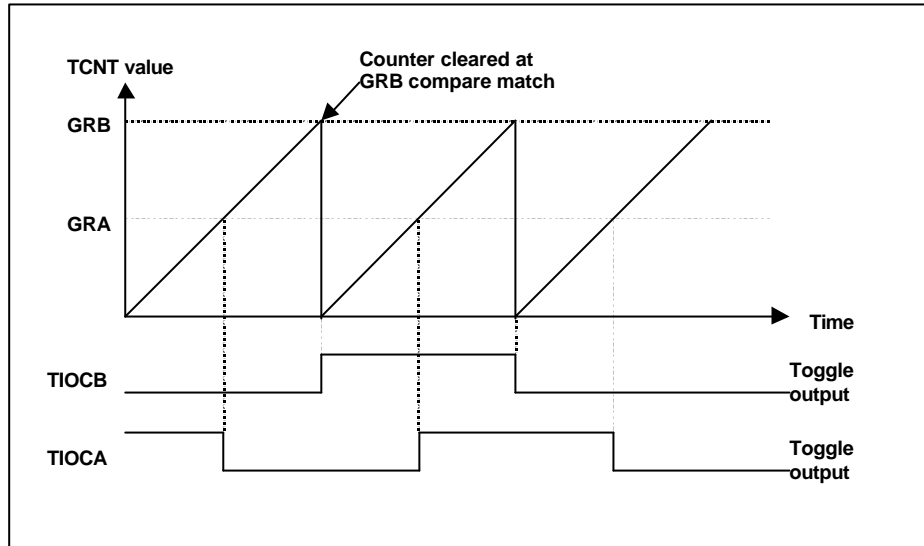


Figure 8.5 Example of Toggle Output

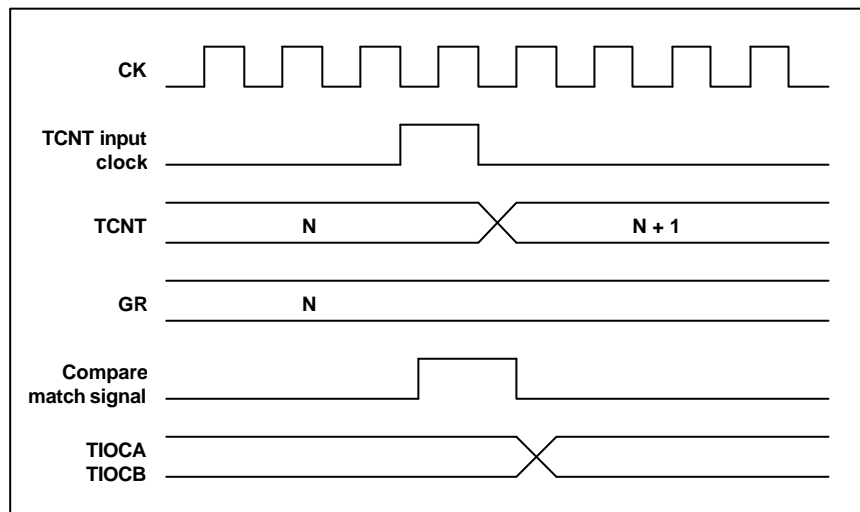


Figure 8.6 Compare Match Signal Output Timing

### 8.3.3 Input Capture Mode

When set to input capture mode, At the rising/falling edge of either capture input TIOCA or TIOCB, the counter value is transferred to GRA or GRB respectively. Also setting the MCIAE or MCIBE in TIER the interrupt can be generated by the external capture event . If CCR field in TCR is appropriately set, The counter can be cleared when the edge of TIOCA or TIOCA is detected.

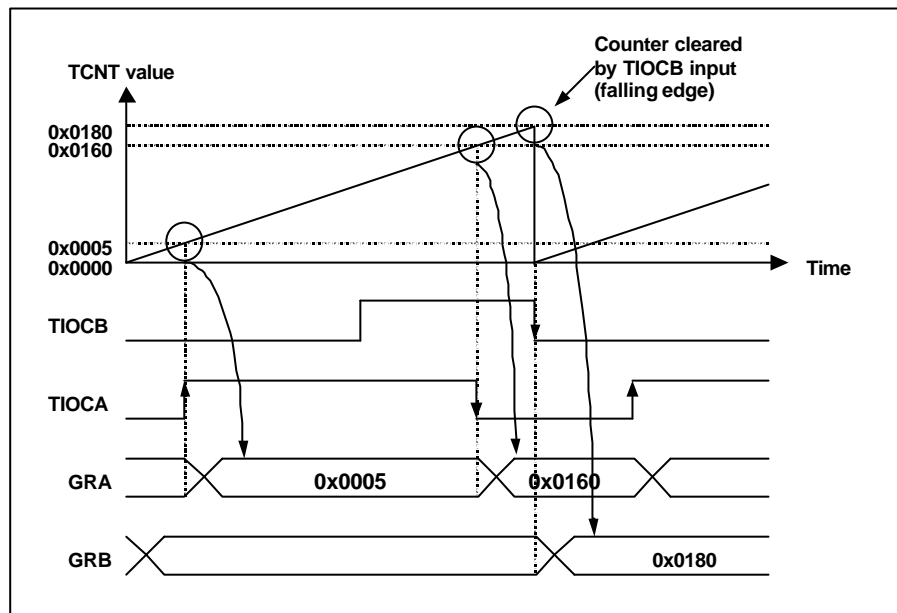


Figure 8.7 Input Capture Operation

8.3.4 Synchronized Clear and 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. 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.

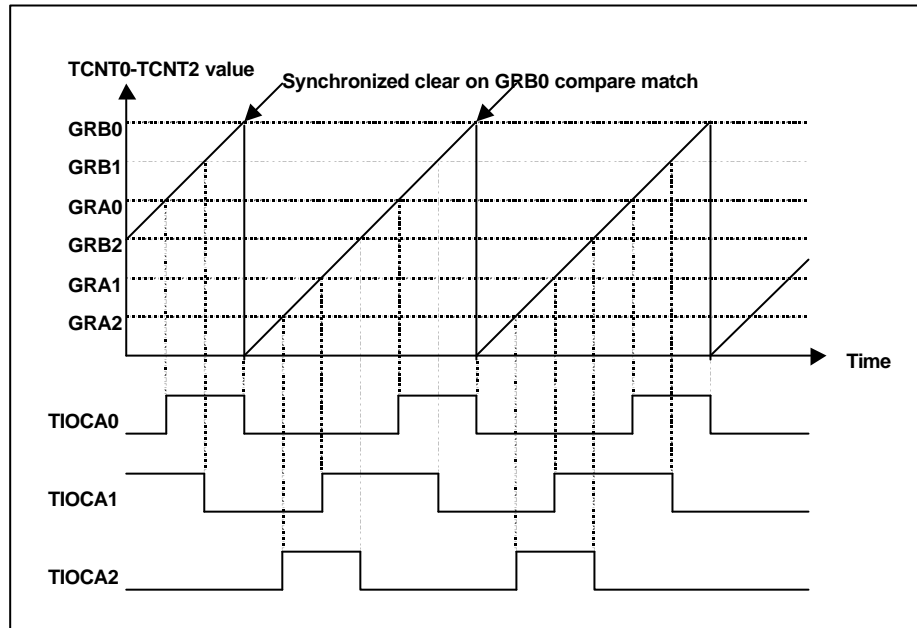


Figure 8.8 Synchronized Operation Example



### 8.3.5 PWM Mode

The PWM mode is controlled using both the GRA and GRB in pairs. The PWM waveform is output from the TIOCA output pin. The PWM waveform's 1 output timing is set in GRA and the 0 output timing is set in GRB. A PWM waveform with duty cycle between 0% and 100% can be output from the TIOCA pin by having either compare match GRA or GRB be the counter clear source for the timer counter. All five channels can be set to PWM mode.

#### 8.3.5.1 PWM Mode Operation

**Figure 8.9** illustrates PWM mode operations. When the PWM mode is set, the TIOCA pin becomes the output pin. Output is 1 when the TCNT matches the GRA, and 0 when TCNT matches the GRB. The TCNT can be cleared by compare match with either GRA or GRB. This can be used in both free-running and synchronized operation.

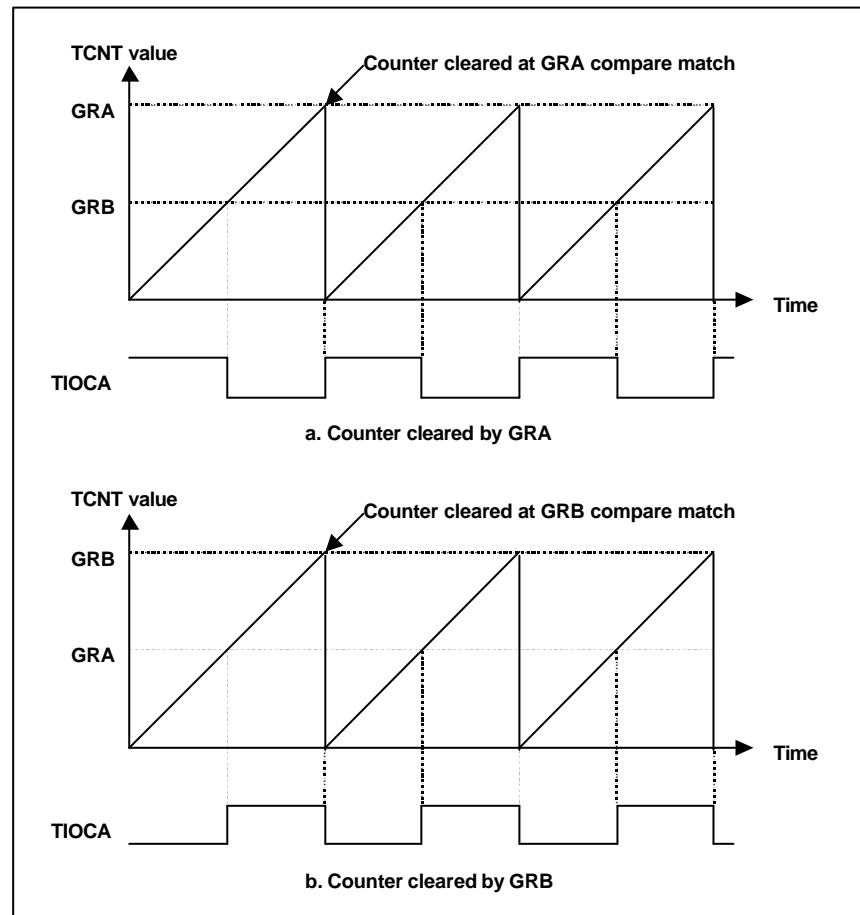
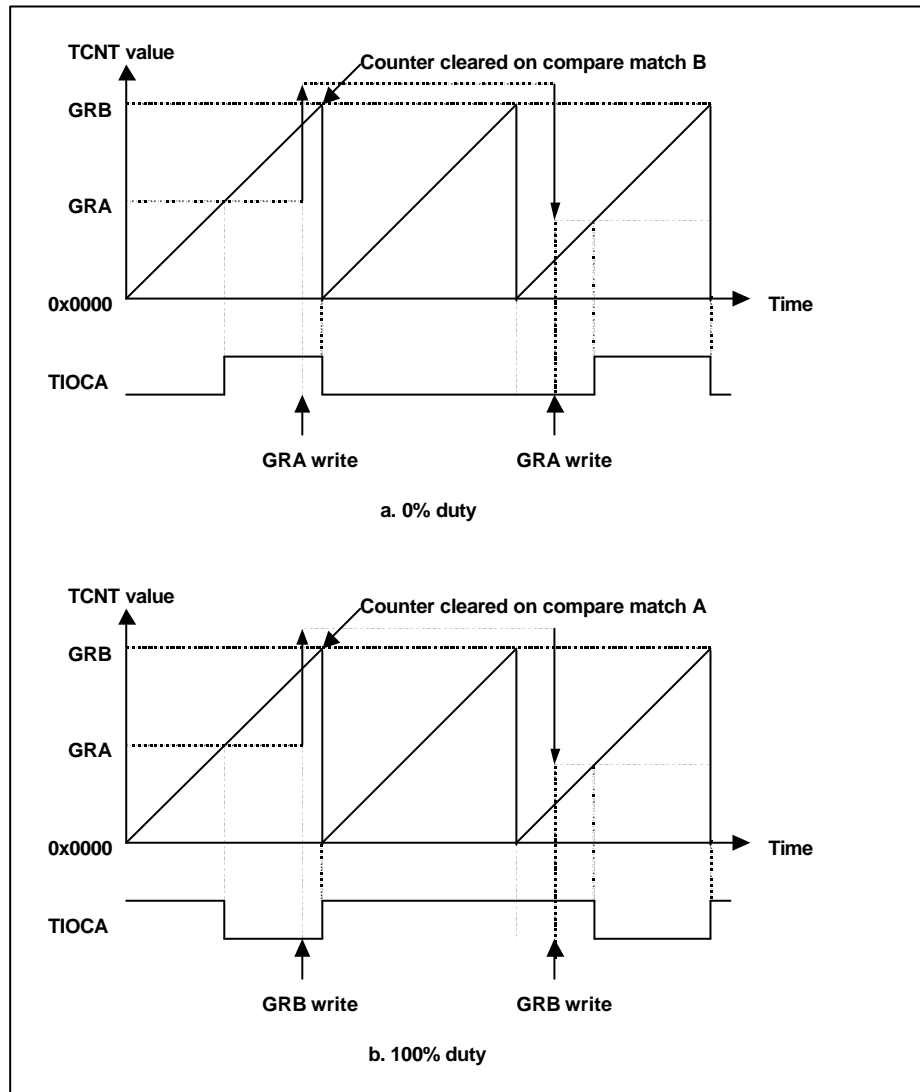
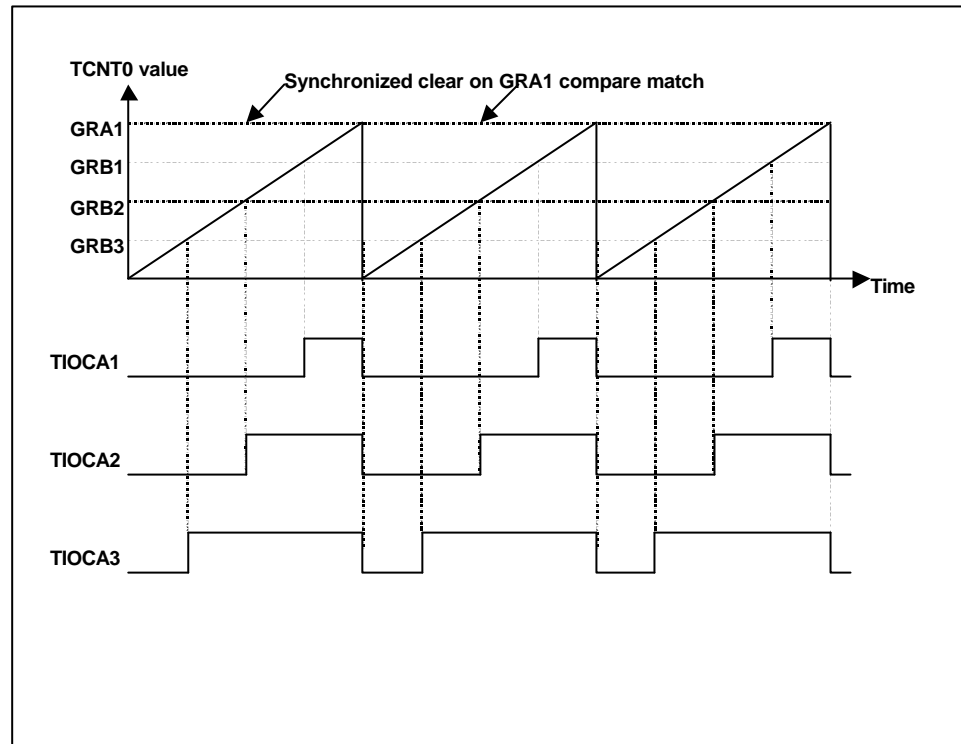


Figure 8.9 PWM Mode Operation Example 1

**Figure 8.10** shows examples of PWM waveforms output with 0% and 100% duty cycles. A 0% duty waveform can be obtained by setting the counter clear source to GRB and then setting GRA to a larger value than GRB. A 100% duty waveform can be obtained by setting the counter clear source to GRA and then setting GRB to a larger value than GRA



**Figure 8.10 PWM Mode Operation Example 2**



**Figure 8.11** Reset-Synchronized PWM Mode Operation Example

**Reset-Synchronized PWM Mode Operation:**

**Figure 8.11** shows an example of operation in the reset-synchronized PWM mode. TCNT1 operates as an upcounter that is cleared to 0x0000 at compare match with GRA1. TCNT2 runs independently and is isolated from GRA2 and GRB2. The PWM waveform outputs toggle at each compare match (GRB1, GRB2, and GRB3 with TCNT1) and when the counter cleared.



Chapter 9

## **UART (Universal Asynchronous Receiver/Transmitter)**

9.1 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 16550 can be put into an alternate mode (FIFO mode) to relieve the CPU of excessive software overhead.

In this mode internal FIFOs are activated allowing 16 bytes plus 3 bit of error data per byte in the RCVR FIFO, to be stored in both receive and transmit modes. All the logic is on the chip to minimize the system overhead and maximize system efficiency.

The UART performs serial-to-parallel conversion on data characters received from a peripheral device 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 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. In addition to baud rate generate, the UART also include clock divider which divide the input system clock by setting the 8-bit divider register.

The UART has a processor-interrupt system. Interrupts can be programmed to the user's requirements, minimizing the computing required to handle the communications link. The general 16450/16550 has MODEM control signals, and thus this module also has a MODEM signals internally, but these signals are concealed.

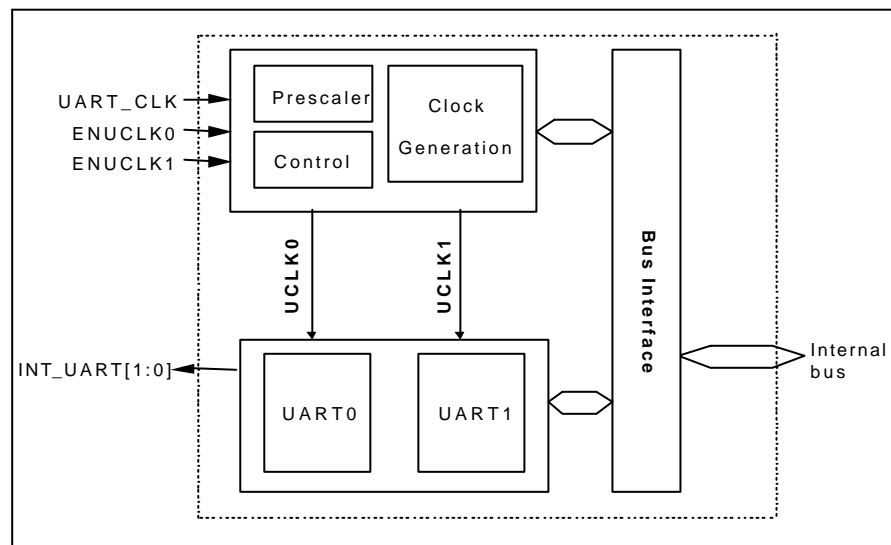


Figure 9.1 TOP BLOCK Diagram

## 9.2 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
- Input clock divider by setting 8-bit divider register.
- Independent receiver clock input.
- 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.

## 9.3 Signal Description

**Table 9.1 Signal Descriptions**

Name	Type	Description
RXD0	I	Serial Input. Serial data input from the communications link (peripheral device, MODEM or data set).
RXD1	I	Serial Input. Serial data input from the communications link (peripheral device, MODEM or data set).
TXD0	O	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.
TXD1	O	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.

9.4 Internal Block Diagram

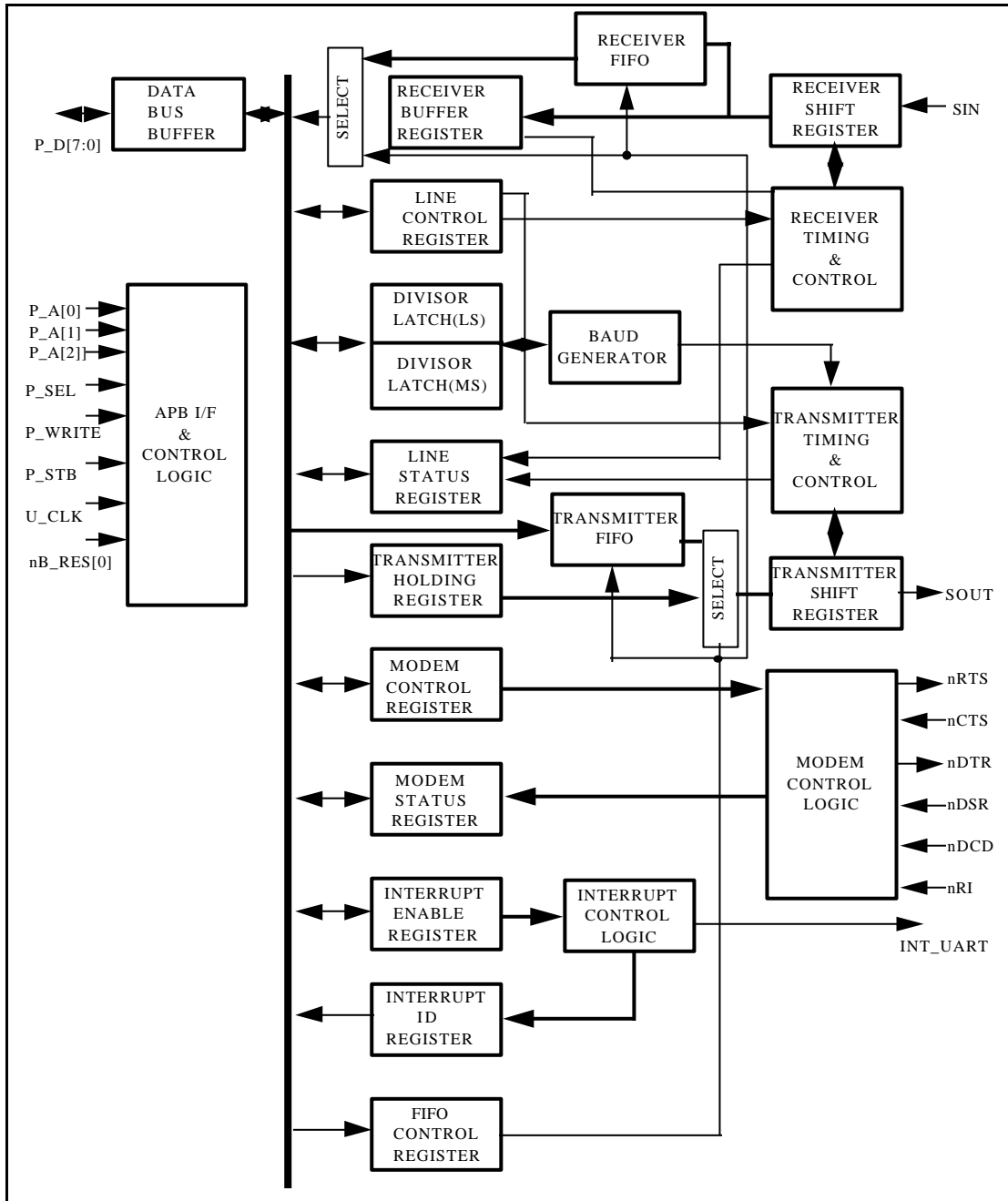


Figure 9.2 Internal UART Diagram



## 9.5 Registers Description

There are two UARTs implemented in the design, the base addresses are 0x0900\_1400 in UART0 and 0x0900\_1500 in UART1.

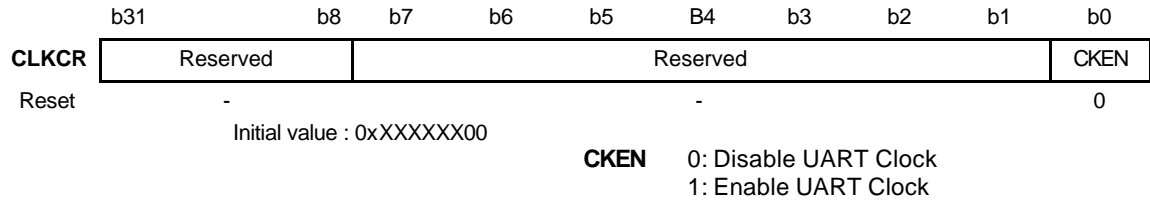
**Table 9.2 UART Register Address Map (0x1500 in UART1)**

Reg. Name	I/O Offset	Dir,	Description
RBR	0x1400	R	Receiver Buffer (DLAB = 0)
THR	0x1400	W	Transmitter Holding (DLAB = 0)
IER	0x1404	R/W	Interrupt Enable
IIR	0x1408	R	Interrupt Identification
FCR	0x1408	W	FIFO Control
LCR	0x140C	R/W	Line Control
LTR	0x1410	R/W	Loop Test Control
LSR	0x1414	R/W	Line Status
-	0x1418	-	Reserved
SCR	0x141C	R/W	Scratch Register
DLL	0x1400	R/W	Divisor Latch LSB (DLAB = 1)
DLM	0x1404	R/W	Divisor Latch MSB (DLAB = 1)
CLKCR	0x1420	R/W	Clock Control
CLKDR	0x1424	R/W	Clock Divisor

**Table 9.3 UART Register Reset Values**

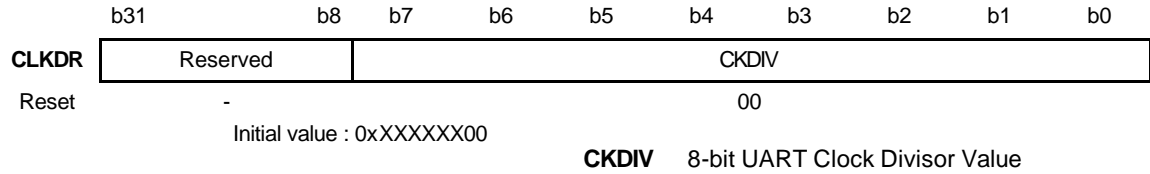
Reg.	Reset Values
IER	0x00
IIR	0x01
FCR	0x00
LCR	0x00
LTR	0x00
LSR	0x60
TxD	'1'

**CLKCR Clock Control Register (0x1420 R/W)**



The system programmer starts and stops the UART clock generator the Clock Control Register (CLKCR). The programmer can also read the contents of the Clock Control Register. The CKEN bit is the start clock bit. When this bit is logic 1, the UART clock is generated from the on-chip clock generator. If this bit is a logic 0, then the clock generator stop to operate.

**CLKDR Clock Divisor Register (0x1424 R/W)**



The UART contains a programmable Clock Generator that is capable of taking any clock input and dividing it by any divisor from 0 to 255. One 8-bit register stores the divisor in a 8-bit binary format. This Divisor Register must be loaded before setting the Clock Control Register to ensure the proper operation of the UART Clock Generator.

## LCR Line Control Register (0x1400 ReadOnly)

	b31	b8	b7	b6	b5	b4	b3	b2	b1	b0
LCR	Reserved		DLAB	BREAK	STICKP	PARITY	PEN	STOPBIT	DLEN	
Reset	-		0	0	0	0	0	0	00	

Initial value : 0xXXXXXX00

- DLEN** 00: 5-bit Data  
01: 6-bit Data  
10: 7-bit Data  
11: 8-bit Data
- STOPBIT** 0: 1 stop bit  
1: 1.5/2 stop bits
- PEN** 0: Disable Parity  
1: Enable Parity
- PARITY** 0: Odd Parity  
1: Even Parity
- STICKP** 0: Stick Parity as ' 0'  
1: Stick Parity as ' 1'
- BREAK** 0: Nomal Transmission  
1: Send Break
- DLAB** 0: Nomal state  
1: Divisor Latch Access Mode

The system programmer specifies the format of the asynchronous data communications exchange and set the Divisor Latch Access bit (DLAB) 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 9.8 Summary of Registers shows the contents of the LCR. Details on each bit are :

- DLEN** These two bits specify the number of bits in each transmitted and received serial character.
- STOPBIT** 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.
- PEN** 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.)
- PARITY** 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.
- STICKP** 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

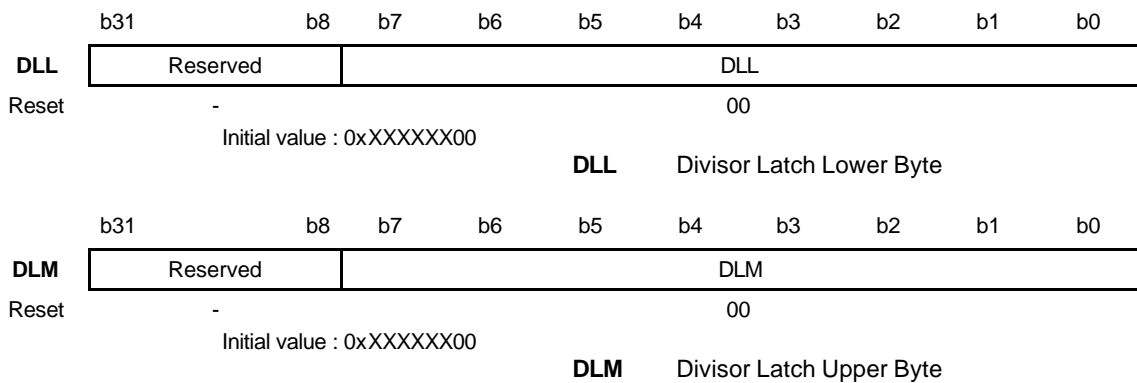
is a logic 0 Stick Parity is disabled.

**BREAK** 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 (TxD) 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 TxD 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.*

**DLAB** This bit is the Divisor Latch Access Bit. 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.

**DLL/DLM Divisor Latch Register (0x1400/0x1404 R/W)**



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 12-5** 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 9.4a Divisor Values for each Baud rate (CLK=1.8432MHz)**

1.8432 MHz		
Desired Baud Rate	Decimal Divisor Value	Error Percentage
50	2304	-
75	1536	-
110	1047	0.026
150	768	-
300	384	-
600	192	-
1200	96	-
1800	64	-
2000	58	-
2400	48	-
3600	32	-
4800	24	-
7200	16	-
9600	12	-
19200	6	-
38400	3	-
57600	2	-
115200	1	-

**Table 9.4b Divisor Values for each Baud rate (CLK=3.6864MHz)**

3.6864 MHz		
Desired Baud Rate	Decimal Divisor Value	Error Percentage
50	4608	-
110	2094	0.026
300	768	-
1200	192	-
2400	96	-
4800	48	-
9600	24	-
19200	12	-
38400	6	-
57600	4	-
115200	2	-

**LSR Line Status Register (0x1414 ReadOnly)**

	b31	b8	b7	b6	b5	b4	b3	b2	b1	b0
<b>LSR</b>	Reserved	FIFOE	TEMT	THRE	BI	FE	PE	OE	DR	
Reset	-	0	1	1	0	0	0	0	0	0

Initial value : 0xXXXXXX60

- DR** 0: No data received  
1: Received Data Ready
- OE** 0: No overrun error  
1: Overrun Error
- PE** 0: No parity error  
1: Parity Error
- FE** 0: No framing error  
1: Framing Error
- BI** 0: No break detected  
1: Break Interrupted
- THRE** 0: THR not empty  
1: THR Empty
- TEMT** 0: Transmitter not empty  
1: Transmitter Empty
- FIFOE** 0: FIFO is valid  
1: FIFO has Invalid data

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 :

- DR** This bit is the receiver Data Ready 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.
- OE** This bit is the Overrun Error 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.
- PE** This bit is the Parity Error 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.
- FE** This bit is the Framing Error 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”.

**BI** This bit is the Break Interrupt 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.*

**THRE** This bit is the Transmitter Holding Register Empty 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.

**TEMT** This bit is the Transmitter Empty 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.

**FIFOE** In the 16450 mode, this is 0. In the FIFO mode, FIFOE 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.*

**FCR FIFO Control Register (0x1408 WriteOnly)**

	b31	b8	b7	b6	b5	b4	b3	b2	b1	b0
<b>FCR</b>	Reserved		FIFODEPTH		Res	Res	Res	FCR2	FCR1	FIFOEN
Reset	-	00		0	0	0	0	0	0	0

Initial value : 0xXXXXXX00

- FIFOEN** 0: Disable FIFO  
1: Enable Both Tx/Rx FIFO
- FCR1** 0: shift register not cleared  
1: Self-clear shift register
- FCR2** 0: shift register not cleared  
1: Self-clear shift register
- FIFODEPTH** Receive FIFO Trigger Level  
00: 1 Byte  
01: 4 Bytes  
10: 8 Bytes  
11: 14 Bytes

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.

- FIFOEN** 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.
- FCR1** 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.
- FCR2** 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.
- FIFODEPTH** These bits are used to set the trigger level for the RCVR FIFO interrupt.



**IIR**                      **Interrupt Identification Register (0x1408 ReadOnly)**

	b31	b8	b7	b6	b5	b4	b3	b2	b1	b0
<b>IIR</b>	Reserved		FIFO		Res	Res	FID	IID		IPEN
Reset	-	00		0	0	0	00		1	

Initial value : 0xXXXXXX01

- IPEN**    0: Interrupt Pending  
           1: No Interrupt pending
- IID/FID**    Interrupt Identification Value  
               (refer Table 9.5)
- FIFO**        Indicate FIFO mode  
               00: None-FIFO mode  
               11: FIFO mode

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 three levels of interrupt conditions are as follows in order of priority:

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

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 9.6 Summary of Registers** shows the contents of the IIR. Details on each bit are :

- IPEN**        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.
- IID**         These two bits of the IIR are used to identify the highest priority interrupt pending as indicated in **Table 9.5 Interrupt control functions**.
- FID**         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.
- FIFO**        These two bits are set when FIFOEN = 1.

Table 9.5 Interrupt Control Functions

Priority Level	FIFO Mode Only	Interrupt Identification Register				Interrupt Set and Reset Functions		
	Bit 3	Bit 2	Bit 1	Bit 0	Interrupt Type	Interrupt Source	Interrupt Reset Control	
-	0	0	0	1	None	None	-	
Highest	0	1	1	0	Receiver Line Status	Overrun Error, Parity Error, Framing Error or Break Interrupt	Reading the Line Status Register	
Second	0	1	0	0	Receiver Data Available	Receiver Data Available or Trigger Level Reached	Reading the Receiver Buffer Register or the FIFO drops below the trigger level	
Second	1	1	0	0	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	
Third	0	0	1	0	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	
Fourth	0	0	0	0	MODEM Status	Clear to Send, Data Set Ready, Ring Indicator, or Data Carrier Detect	Reading the MODEM Status Register	

IEN Interrupt Enable Register (0x1404 R/W)

	b31	b8	b7	b6	b5	b4	b3	b2	b1	b0	
IEN	Reserved		Res				RLSIE	THREIE	DRIE		
Reset	-		0	0	0	0	0	0	0	0	

Initial value : 0xXXXXXX00

- DRIE** 0: No data received  
1: Received Data Ready
- THREIE** 0: No overrun error  
1: Overrun Error
- RLSIE** 0: No parity error  
1: Parity Error

This register enables the five types of UART interrupts. Each interrupt can individually activate the interrupt 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 UART interrupt output signal. All other system functions operate in their normal manners, including the setting of the Line Status Registers. **Table 9.6 the Summary of Registers** shows the contents of the IER. Details on each bit are :

- DRIE** This bit enables the Received Data Available Interrupt (and time-out interrupts in the FIFO mode) when it is set to logic 1.
- THREIE** This bit enables the Transmitter Holding Register Empty Interrupt when set

to logic 1.

**RLSIE** This bit enables the Receiver Line Status Interrupt when it is set to logic 1.

#### LTR Loop Test Control Register (0x1410 R/W)

	b31	b8	b7	B6	b5	b4	b3	b2	b1	b0
<b>LTR</b>	Reserved		Res			LBON	Res			
Reset	-	0	0	0	0	0	0	0	0	0

Initial value : 0xXXXXXX00

**LBON** 0: Normal Transmission  
1: Loopback mode

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 9.6 Summary of Registers**, and are described below.

**LBON** 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.

#### SCR Scratch Register (0x141C R/W)

	b31	b8	b7	B6	b5	B4	b3	b2	b1	b0
<b>SCR</b>	Reserved		SCR							
Reset	-	00								

Initial value : 0xXXXXXX00

**SCR** 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.

## 9.6 UART Operations

### 9.6.1 FIFO Interrupt Mode Operation

**When the RCVR FIFO and receiver interrupts are enabled (FIFOEN = 1, DRIE = 1), RCVR interrupts occur as follows :**

The received data available interrupt will be issued to the CPU when the FIFO has reached its programmed trigger level; it will be cleared as soon as the FIFO drops below its programmed trigger level.

The IIR receive data available indication also occurs when the FIFO trigger level is reached, and like the interrupt it is cleared when the FIFO drops below the trigger level.

The receiver line status interrupt (IIR=0x06), as before, has higher priority than the received data available (IIR=0x04) interrupt.

The data ready bit (DR) 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 (FIFOEN = 1, THREIE = 1), XMIT interrupts occur as follows :**

1. The transmitter holding register interrupt (IIR=0x02) 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 RFOEN 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.

### 9.6.2 FIFO Polled Mode Operation

When FIFOEN = 1 resetting, DRIE, THREIE, RLSIE 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.

9.7 Register Summary

Table 9.6 Summary of Registers

Reg. Name	Offset	Dir.	Bit Field								cf.	
			Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		
RBR	0x00	R	RBR								DLAB=0	
THR	0x00	W	THR								DLAB=0	
IER	0x04	R/W					RLSIE	THREIE	DRIE			
IIR	0x08	R					FID	IID		IPEN		
FCR	0x08	W	FIFODEPTH					FCR2	FCR1	FIFOEN		
LCR	0x0C	R/W	DLAB	BREAK	STICKP	PARITY	PEN	STOPBIT	DLEN			
LTR	0x10	R/W					LBON					
LSR	0x14	R	FIFOE	TEMT	THRE	BI	FE	PE	OE	DR		
-	0x18	-										
SCR	0x1C	R/W	SCR									
DLL	0x00	R/W	DLL								DLAB=1	
DLM	0x01	R/W	DLM								DLAB=1	
CLKCR	0x20	R/W								CKEN		
CLKDR	0x24	R/W	CLKDIV									

Chapter 10.

## **GPIO (General Purpose Input Output)**

10.1 General Description

The GPIO is an APB peripheral which provides 79 bits of programmable input /output divided into 11 ports ; port A, port B, port 1, port 2, port 3, port 4, port 5, port 6, port 7, port 8 and port 9. Each pin is configurable as either input or output except port7 [4:0]. At system reset, port A, 1, 3, 5, 8, 9 set their defaults to input and port B, 2, 4, 6, 7 set their defaults to output.

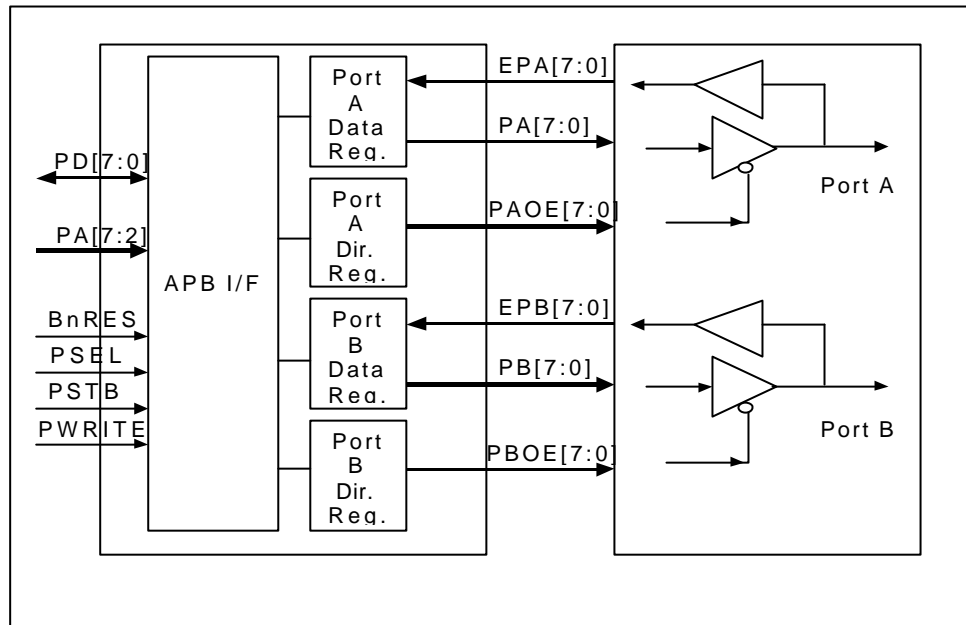


Figure 10.1 GPIO Block Diagram and PADS Connections(example for Port A and Port B)

Each port has a data register and a data direction register. 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 GPIO pins, both input and output, as well as to set the values of pins that are configured as outputs.



## 10.2 GPIO Registers

The following user registers are provided:

**P<sub>n</sub>DR\*** Port n Data Register. Values written to this read/write register will be input on port A pins if the corresponding data direction bits are set to HIGH (port input). 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.

**P<sub>n</sub>DDR\*** Port n Data Direction Register. Bits set in this read/write register will select the corresponding pins in port n to become an input, clearing a bit sets the pin to output. All bits are cleared by a system reset.

\*n is: A, B, 1, 2, 3, 4, 5, 6, 7, 8 or 9

### 10.2.1 Register Memory Map

The start address of the GPIO is **0x0900\_1600** and the offset of any particular register from the base address is determined.

**Table 10.1 GPIO Register Memory Map**

REG.	I/O OFFSET	DIR	DESCRIPTION
PADR	0x1600	R/W	8-bit Port A Data register
PADDR	0X1604	R/W	Port A Data Direction register
PBDR	0X1608	R/W	8-bit Port B Data register
PBDDR	0X160C	R/W	Port B Data Direction register
P1DR	0X1610	R/W	8-bit Port 1 Data register
P1DDR	0X1614	R/W	Port 1 Data Direction register
P2DR	0X1618	R/W	8-bit Port 2 Data register
P2DDR	0X161C	R/W	Port 2 Data Direction register
P3DR	0X1620	R/W	8-bit Port 3 Data register
P3DDR	0X1624	R/W	Port 3 Data Direction register
P4DR	0X1628	R/W	8-bit Port 4 Data register
P4DDR	0X162C	R/W	Port 4 Data Direction register
P5DR	0X1630	R/W	4-bit Port 5 Data register
P5DDR	0X1634	R/W	Port 5 Data Direction register
P6DR	0X1638	R/W	8-bit Port 6 Data register
P6DDR	0X163C	R/W	Port 6 Data Direction register
P7DR	0X1640	R/W	8-bit Port 7 Data register
P7DDR	0X1644	R/W	Port 7 Data Direction register
P8DR	0X1648	R/W	5-bit Port 8 Data register
P8DDR	0X164C	R/W	Port 8 Data Direction register
P9DR	0X1650	R/W	8-bit Port 9 Data register
P9DDR	0X1654	R/W	Port 9 Data Direction register

### 10.3.1 Register Description

Each GPIO port have their own Data register and Data Direction register. All those ports are not 8-bit register.

#### **PnDR** Port *n* Data Register (R/W, *n* is A,B,1,2,3,4,5,6,7,8 or 9)

	b31 - b8	b7	b6	b5	b4	b3	b2	b1	b0
<b>PnDR</b>	Reserved	D7	D6	D5	D4	D3	D2	D1	D0
Reset	-	0	0	0	0	0	0	0	0
	Initial value : 0x-00			Bit field D0-D7		1 : On 0 : OFF			

#### **PnDDR** Port *n* Data Direction Register (R/W, *n* is A,B,1,2,3,4,5,6,7,8 or 9)

	b31 - b8	b7	b6	b5	b4	b3	b2	b1	b0
<b>PnDDR</b>	Reserved	I7	I6	I5	I4	I3	I2	I1	I0
Reset	-	1	1	1	1	1	1	1	1
	Initial value : 0x-FF			Bit field I0-I7		1 : Assign the port as Input 0 : Assign the port as Output			

### 10.3 Functional Description

All block registers are cleared during power on reset.

This disables the output drivers for port A, 1, 3, 5, 8 and 9 (input as default) and enables the drivers for port B, 2, 4, 6 and 7 (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.



Chapter 11  
**On-Chip SRAM**

### 11.1 General Description

The HMS39C7092 has 4-kbytes of high speed static RAM on-chip. The RAM is connected to the CPU by a 32-bit ASB (Advanced System Bus) bus. The CPU accesses byte data, half-word data, and word data in one cycle, making the RAM useful for rapid data transfer.

### 11.2 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.

Since the on-chip RAM is connected to the CPU by an internal 32-bit data bus, it can be written and read by word access. It can also be written and read by half-word or byte access.

Memory area **0x0803\_0000** to **0x0803\_FFFF** is allocated to the on-chip SRAM as default. The memory area of the on-chip SRAM is allocated **0x0000\_0000** to **0x0000\_0FFF** in Remap mode. This Remap mode is entered by setting the REMAP flag in MEM\_CR of PMU(see **Chapter 5 Power Management Unit** for detail).

Chapter 12

## On-chip Flash Memory

## 12.1 General Description

The **HMS39C7092** has 192-Kbytes of on-chip flash memory. The flash memory is connected to the CPU by a 16-bit data bus. The CPU accesses both half-word and word data in several states depending on the wait register value.

The on-chip flash memory booting option is enabled and disabled by setting the mode pins (MD<sub>2</sub> to MD<sub>0</sub>) as shown in **Table 12.1**.

## 12.2 Features

The features of the flash memory are summarized below.

- Memory organization : 96K x 16 (1.5Mbit)
- Operating Voltage : dual power 3.0V ~ 3.6V(V<sub>cc</sub>), 4.5~5.5V(FTVPPD)
- Random access time : 90nsec
- Program time : typ. 100usec/word
- Erase block size : 32KB x 4, 24KB x 2, 8KB x 2
- Block erase time : typ. 1.5sec/32KB (pre program + erase)
- Multiple block erase command support (maximum 4 blocks)
- Endurance : Min. 100 cycles
- Both on-chip (user/boot mode) and on-board (PROM mode) program/erase support
- Bi-directional Data IO
- Operation current : Standby mode : 10uA
- Read/Program/Erase mode : max. 20mA



Table 12.1 Operating mode

MD <sub>2</sub>	MD <sub>1</sub>	MD <sub>0</sub>	Mode	Description
0	1	0	mode 2	External 8bit data bus and 16-Mbyte address mode
0	1	1	mode 3	External 16-bit data bus and 16-Mbyte address mode
1	0	0	mode 4	Flash memory boot mode with external 16-bit data bus mode
1	0	1	mode 5	Flash memory boot mode(micorocomputer mode)
1	1	0	mode 6	UART booting mode with external 16-bit data bus
1	1	1	mode 7	UART booting mode with microcomputer mode

12.3 Block Diagram

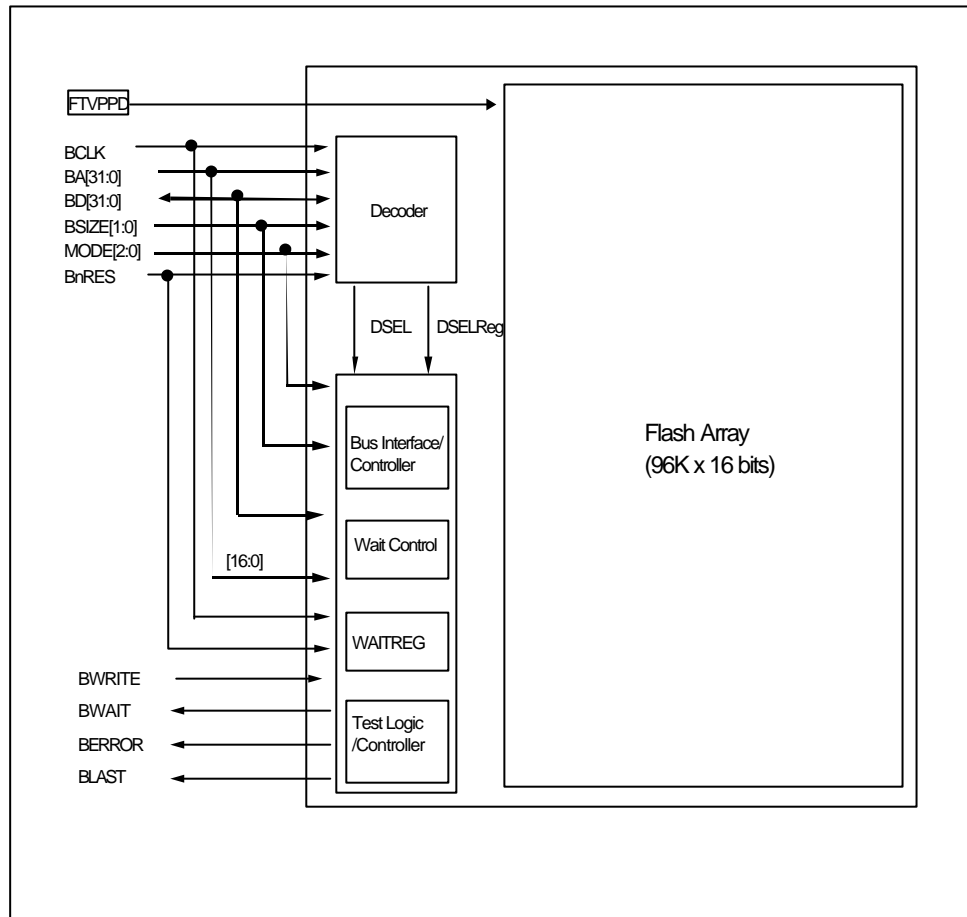


Figure 12.1 Block Diagram of Flash Memory

Table 12.2 Signal description of Figure 12.1(BUS Interface)

Name	I/O	Function
BnRES	I	These signal indicate the reset status of the ASB
BCLK	I	The ASB clock timing all bus transfers
DSELREG	I	When this signal is HIGH, it indicates that the Flash Memory configuration Internal registers are selected. (When BA[31:0] is set to FMI Region, 0x09000200~0x090002ff, of Memory Map)
DSEL	I	When this signal is HIGH, it indicates that the Flash Memory Array address is selected. (When BA[31:0] is set to Flash Memory Region of Memory Map.)
BWRITE	I	When this signal is HIGH, it indicates a write transfer and when LOW a read.
BSIZE[1:0]	I	These signals indicate the size of the transfer that may be byte, half-word, or word.
BA[31:0]	I	System address bus. BA[[31:17] is used for selection between Internal Register Block and Flash Memory. BA[16:0] is used for selection of Specific Internal Register or Flash Memory Address.
BD[31:0]	I/O	Bi-directional system data bus.
BWAIT	O	Wait slave response signal. It is driven to phase 1 when Flash Memory Read operation is selected. It is asserted while the Flash Memory Read operation is uncompleted.
BERROR	O	When BEEOR is HIGH, a transfer error has occurred. When BERROR is LOW, then the transfer is successful
BLAST	O	When BLAST is HIGH, the system decoder must allow sufficient time for address decoding. When BLAST is LOW, the next transfer may continue a burst sequence.
MODE [2:0]	I	These signals are directly connected to external pins(MD <sub>2</sub> ~ MD <sub>0</sub> ), and represent operating mode at table12.1
FTVPPD	In	It is external power that is used to flash memory program and directly Connected to external pin at all operating mode.

12.4 Flash Memory Register Description

The registers used to control the on-chip flash memory when enabled are shown in **Table 12.3**. The base address of the flash memory register(FMU\_base) is 0x0900\_0200.

**Table 12.3 Flash Memory Registers**

Reg.	I/O Offset	Dir.	Description	Initial Value
WAITREG	0x0200	R/W	Wait Register	0x000F
ADDREG	0x0204	R/W	Address Register	0x0000
DATAREG	0x0208	R/W	Data Register	0xFFFF
CONTREG	0x020C	R/W	Control Register	0x0000
EBR	0x0210	R/W	Erase Block Select Register	0x0000
STATPWR	0x0214	R/W	Status & Power Register	0x0000
TESTR	0x0218	R/W	Test Register	0x0400

**WAITREG**

**Wait Control Register (WAITREG)**

Bit	7	6	5	4	3	2	1	0
	W7	W6	W5	W4	W3	W2	W1	W0
Initial Value	0	0	0	0	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**WAITREG** is an 8-bit register used for bus access wait time control. The initial value is 0xF. Once Read command is executed, the next command execution is delayed for the number of bus clocks(BCLK) that the WAITREG is set. So, Flash Memory Read Operation is performed during the time of WAITREG value \* period of BCLK. For the successful execution of Flash Memory Read Operation without interruption, this time must be longer than a Flash Memory Access time(TACC). The typical value of WAITREG using 33 MHz-clock input is 03H, which means that the delay time is 90nsec.

**ADDREG Address Register**

Bit	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
Init. Val.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RD/WR	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address of 16-bit word to be programmed or verified is stored in this register in the Program mode(including Pre-Program) and the Verify mode (program/erase verify). In Normal Read Mode, FA[16:0] is passed directly in address decoding block without passing through ADDREG.

In Erase Mode, selecting block in 'block select register' causes the specified Flash Memory block to be erased.

Users can write this register directly at mode1(PROM Mode) by setting FR\_SEL signal, and usable address range are 96K x16 bits, 0x00000 ~ 0x17FFF. In this Mode, if FR\_SEL[2:0] is '001' and FWEB is rising\_edge, FA[16:0] is passed into ADDREG. If FR\_SEL[2:0] is '001', FWEB is '1' and FOEB=0, users are able to read 16-bit of ADDREG[16:1] through FD[15:0] in this Mode.

In other mode except mode1, ADDRREG are written via decoded value from BA[16:0] of the Flash Memory address write command (not the ADDRREG write) and read directly through ADDRREG read.

**DATREG Data Register**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Init. Val.	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
RD/WR	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Register for storing data that is programmed to Flash Memory Address of ADDREG value in Program mode. Each bit is corresponded to each cell one by one and if it is 0, cell can be programmed, else not programmed. Flash Memory of HMS39C7092 can be programmed 16 bits at one time.

After reset, Data register output value is all reset to 0xFFFF and the other registers are reset to '0'.

Users can write this register directly at mode1(PROM Mode). In this Mode, if FR\_SEL[2:0] = '001' and FWEB = rising\_edge, FD[15:0] is passed into DATAREG. If FR\_SEL[2:0] = '010', FWEB = '1' and FOEB=0, users are able to read 16-bit of DATAREG[15:0] through FD[15:0].

In other mode except mode1, DATAREG are written via BD[16:0] of the Flash Memory address write command (not the DATAREG write) and read directly using DATAREG read.

**CONTREG Control Register**

Bit	7	6	5	4	3	2	1	0
	-	-	ER_VFY	PGM_VFY	ERSE	PGM	ER_PWR	PGM_PWR
Initial Value	-	-	0	0	0	0	0	0
Read/Write	-	-	R/W	R/W	R/W	R/W	R/W	R/W

**CONTREG** is an 8-bit register used for flash memory operating mode control. It can be read and written in all modes. It controls transition to state of flash memory array read, program and erase operation, and charge pump operation mode. Table 12.4 shows the function of each bit of Control register. Program process has one modes(PGM) and Verify process has two modes (PGM\_VFY,ER\_VFY)

**Table 12.4 Control Register**

Name	Function
<b>PGM_PWR</b>	Program Power Setup (Drain/Positive Gate Pump Enable)
<b>ER_PWR</b>	Erase Power Setup (Negative/Positive Gate Pump Enable)
<b>PGM</b>	Program Start bit. Program Pulse Supply to Addressed Cell (Drain/Gate Pulse)
<b>ERSE</b>	Erase Start bit. Erase Pulse Supply to Addressed Block (Gate/Bulk Pulse)
<b>PGM_VFY</b>	Program Verify Read Enable (Positive Gate Pump Enable)
<b>ER_VFY</b>	Erase Verify Read Enable (Positive Gate Pump Enable)

Controlling control bit of Control register in table 12.4 can perform Program/Erase /Verify process. PGM\_PWR/ER\_PWR sets up the pump before Program and Erase. To make high voltage necessary to program or erase Flash memory address, set the PGM\_PWR for program or ER\_PWR for erase and wait for the pump setup time. After setting up the pump, set the PGM for program or ERSE for erase to start program or erase. In verify mode, without setting bit0 and bit1, setting one bit corresponds to each verify modes (PGM\_VFY,ER\_VFY) set up the pump and perform verify read. In Mode1, if FR\_SEL[2:0] is '011' and FWEB is rising\_edge, FD[7:0] is passed into CONTREG. If FR\_SEL[2:0] is '011', FWEB is '1' & FOEB=0, users are able to read 8-bit of CONTREG through FD[7:0]. In other mode except mode1, CONTREG register write and read are both possible.

**EBR Erase Block Select Register**

Bit	7	6	5	4	3	2	1	0
	SEC7	SEC6	SEC5	SEC4	SEC3	SEC2	SEC1	SEC0
Initial Value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The Bits of this register is used as selector for 'Erase block(sector)' and each bit of this register is matched to each erase block(sector). After setting CONTREG for erase, block erase is performed by setting each corresponding block number bit in EBR to 1. Table 12.5 depicts bit-corresponding sector number and the size of sector and their address of lower 18bits. Capable of multiple block erase by setting multiple bits of this register. (Maximum 4 blocks at a time)

In Mode1, if FR\_SEL[2:0] is '100' & FWEB is rising\_edge, FD[7:0] is passed into EBR. If FR\_SEL[2:0] is '100', FWEB is '1' & FOEB is '0', users are able to read 8-bit value of EBR[16:1] through FD[7:0].

In other mode except mode1, EBR register write and read are both possible.

**Table 12.5 Erase Block Register**

Sector #	Sector Size	Address Range
Sector 0	8KB (4K-word)	0x00000 ~ 0x01FFF
Sector 1	8KB	0x02000 ~ 0x03FFF
Sector 2	24KB	0x04000 ~ 0x0BFFF
Sector 3	24KB	0x0B000 ~ 0x0FFFF
Sector 4	32KB	0x10000 ~ 0x17FFF
Sector 5	32KB	0x18000 ~ 0x1FFFF
Sector 6	32KB	0x20000 ~ 0x27FFF
Sector 7	32KB	0x28000 ~ 0x2FFFF

**STATPWR Status & Power Register**

Bit	8	7	6	5	4	3	2	1	0
	HVEEI	LVEEI	LVCC	VEEI[1:0]		reserved		VPEI[1:0]	
Initial Value	0	0	0	00		00		00	
Read/Write	R/W	R/W	R/W	R/W		-		R/W	

This is register regulates high voltage pump output voltage and indicates status of pump in Program mode and Erase mode. In the following table, bit[8:6] are status bits to indicate status of pump and bit[5:0] are power control bits to regulate high voltage pump output voltage. Bit[5:0] are read/write register that controls voltage of wordline/ bulk needed in Program, erase and verify mode.

In Mode1, if FR\_SEL[2:0] is '101' & FWEB is rising\_edge, FD[5:0] is passed into STATPWR[5:0] (STATPWR[8:6] are read only). If FR\_SEL[2:0] is '101' and FWEB is '1', users are able to read 9-bit of STATPWR through FD[8:0].

In other mode except mode1, STATPWR write and read are both possible.

Table 12.6 Status &amp; Power Register

Bit	Name	Function
8	HVEEI	It's 1, when the 'ER_PWR' in <b>CONTREG</b> is 1 and VEEI(Negative Gate pump output voltage) is below -7V(i.e. -7.1V)
7	LVEEI	It's 1 when VEEI voltage is risen over -1V to discharge.
6	LVCC	It's 1 when Pump is running (PGM_PWR=1 or ER_PWR=1) and VDD becomes below 2.9V.
5,4	VEEI[1:0]	These bits define VEEI (Negative Gate Pump output voltage) when the 'ER_PWR' of <b>CONTREG</b> is 1. (“00” : -9V, “01” : -10V, “10” : -8V, “11” : -10V)
3,2	reserved	These bits are reserved by future use
1,0	VPPI[1:0]	These bits define VPPI(Positive Gate Pump output voltage) when either PGM_PWR or ER_PWR is 1. These bits also define VPPI value differently as program or erase mode, when one of verify mode enable bits in <b>CONTREG</b> is 1. Program/Erase Mode (“00” : 9V, “01” : 8V, “10” : 10V, “11” : 7V) Verify_Mode (“00” : 4V, “01” : 5V, “10” : 6V, “11” : 7V)



## 12.5 On-Board Programming Mode

When pins are set to on-board programming mode and a reset-start is executed, the chip enters the on-board programming state in which on-chip flash memory programming, erasing, and verifying can be carried out. There are two operating modes in this – boot mode and user program mode – set by the mode pins.

Boot mode is for use when user program mode is not available, such as the first time on-board programming is performed, or if the program activated in user program mode is accidentally erased.

### 12.5.1 Boot Mode

When mode pins are set to 6 or 7 and reset-start is executed, the HMS39C7092 enters the Boot Mode programming state in which on-chip flash memory programming, erasing, verifying can be carried out. There are two operating modes in this mode – mode6 is extended mode, mode7 is one-chip micro-controller mode. This device has Internal ROM area for booting. This ROM area locates in 0x00000000, when SBM(Serial Boot Mode) = '1' (i.e. boot mode (Mode6 or 7)), and used for Serial Boot when device is reset.

If boot mode is used, a flash memory programming control program must be prepared beforehand in the host, and UART channel 0 is to be used.

When a reset-start is executed after setting the HMS39C7092 to mode6 or mode7, the boot program in is activated – the bit rate register value determined to 38400 bps(in 33.86MHz), then on-chip UART receives a user program (flash memory programming control program) from off-chip. The received user program is written into RAM (0x08030000 ~ 0x0803FFFF).

Figure 12.2 shows a system configuration diagram when using mode6 or mode7, and figure 12.3 show the boot program mode execution procedure.

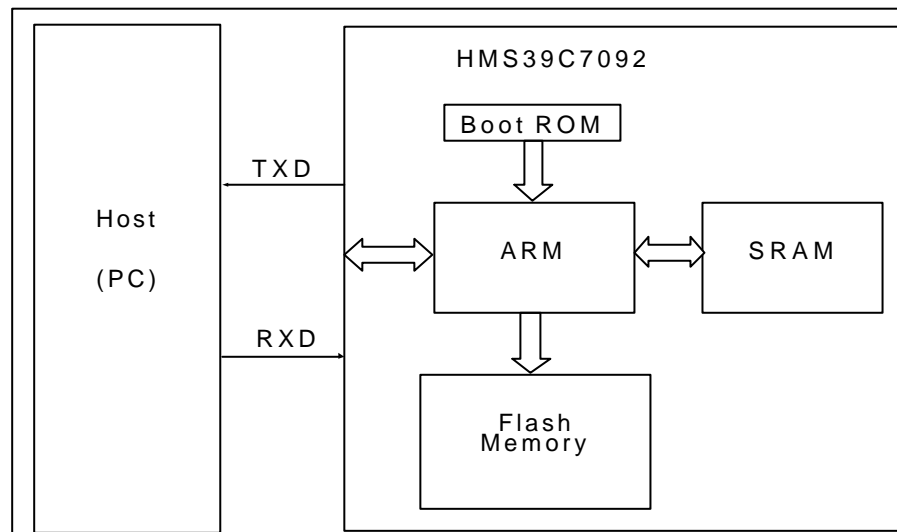


Figure 12.2 System Configuration When Using On-Board Boot Mode

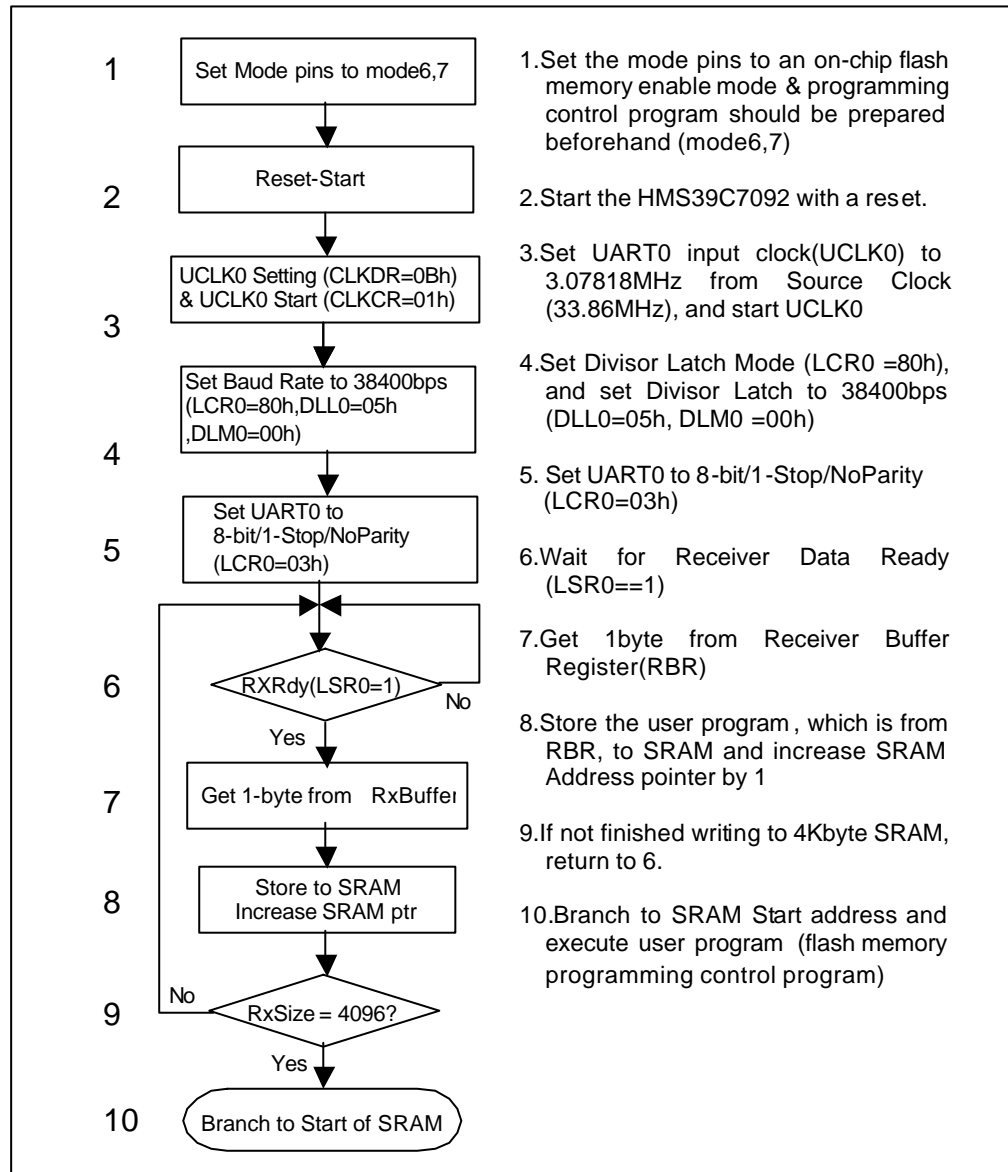
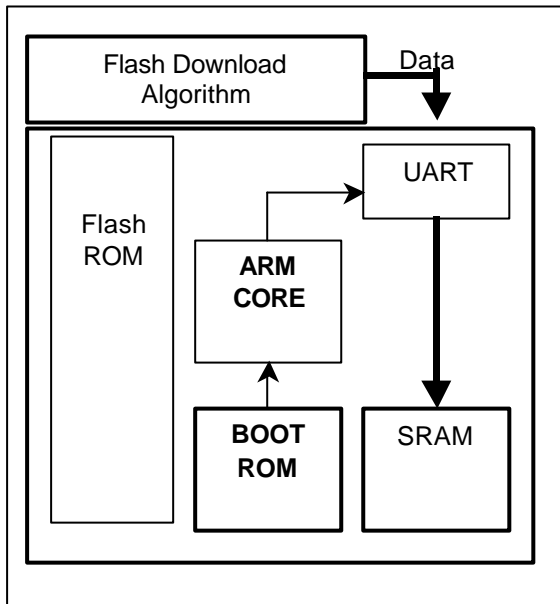


Figure 12.3 Boot Mode Execution Procedure

When boot mode is initiated, the HMS39C7092 measures the low period of the asynchronous communication data transmitted continuously from the host. The UART transmit/receive format should be set as 8-bit data, 1 stop bit, no parity. To ensure correct UART operation, the host's transfer bit rate should be set to 38400 bps, and the operating frequency for this process should be 33.86MHz.

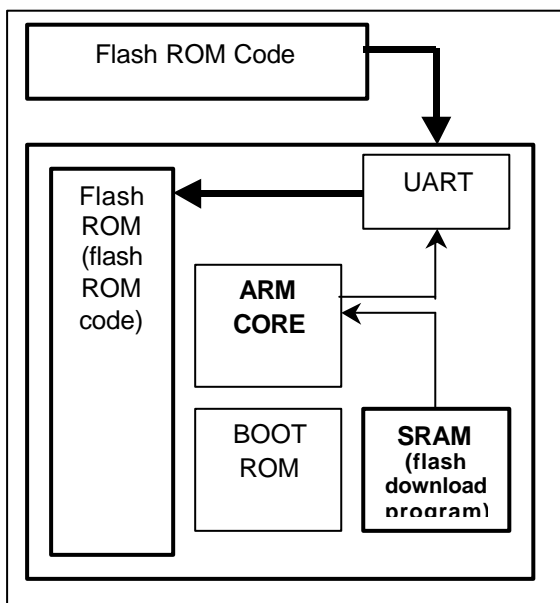
## Application example (Boot Mode)

### 1. Download Application Program



- Step 1. Set Serial Boot Mode to ' 1' (flash download algorithm program should be prepared in the host beforehand)
- Step 2. Reset system
- Step 3. ARM runs Boot Program in internal ROM.
- Step 4. Boot program receives flash download algorithm program through UART from Host.
- Step 5. Store the flash download algorithm program into the internal SRAM.

### 2. Run Downloaded Application Program



- Step 6. ARM branches to the start address of the flash download algorithm program.
- Step 7. The algorithm program gets flash ROM code from host through UART and executes flash ROM Write operation with the code.
- Step 8. End the flash ROM Write operation and Host changes system mode to Normal.
- Step 9. Reset
- Step 10. ARM Execute New Program in the flash ROM.

12.5.2 User Program Mode

When set to user program mode, the HMS39C7092 can program and erase its flash memory by executing a user program/erase control program. Therefore, on-board reprogramming of the on-chip flash memory can be carried out by providing programming data and program/erase control program at the host, and storing transfer program of a program/erase control program in flash memory area beforehand.

To select user program mode, select a mode that enable the on-chip flash memory (mode 4 or 5). Mode4 is extended mode, and mode5 is one-chip micro-controller mode. The flash memory itself cannot be read while being programmed or erased, so the control program that performs program/erase should transferred from external memory to RAM and executed in RAM. Figure 12.4 shows the execution procedure when user program mode is entered during program execution in RAM.

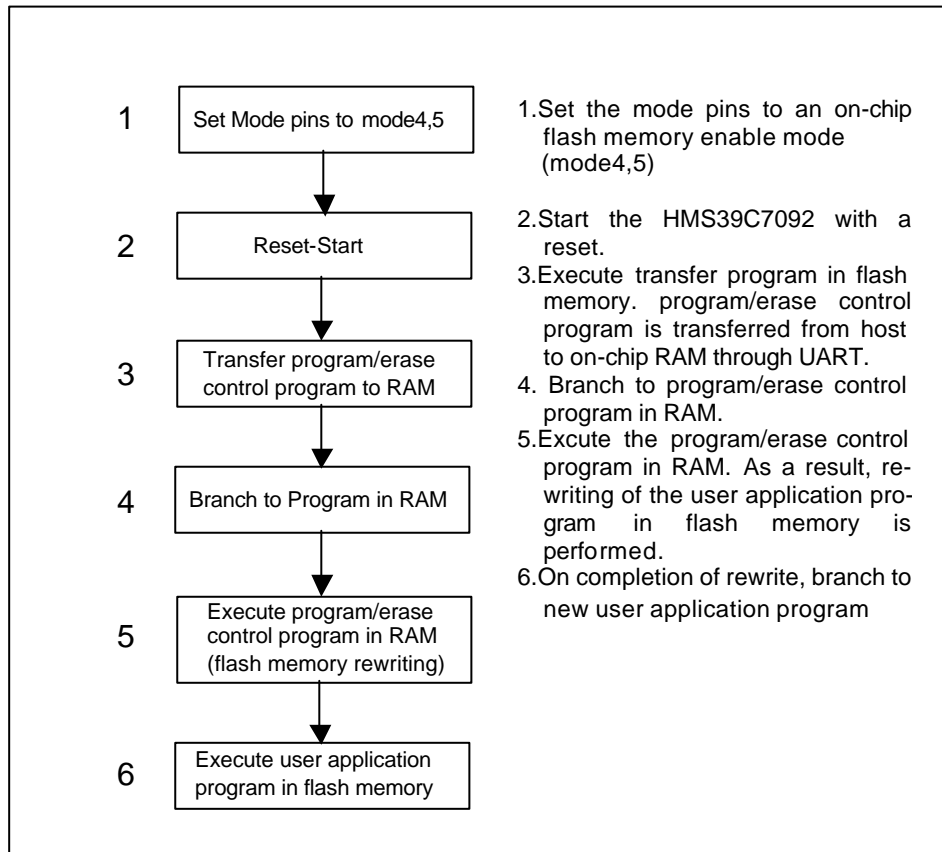
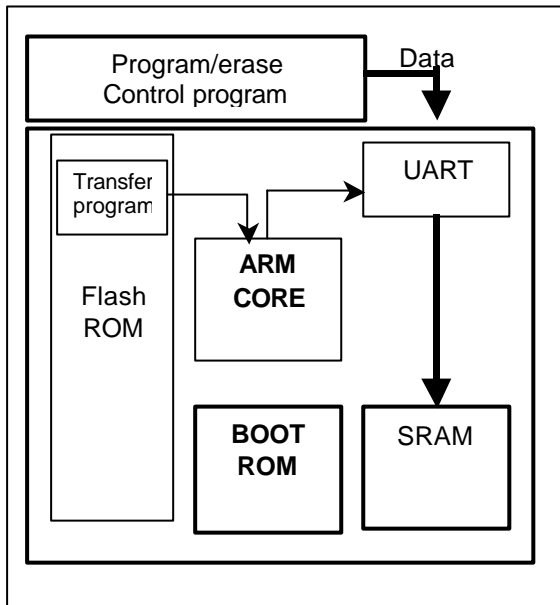


Figure 12.4 User Mode Execution Procedure

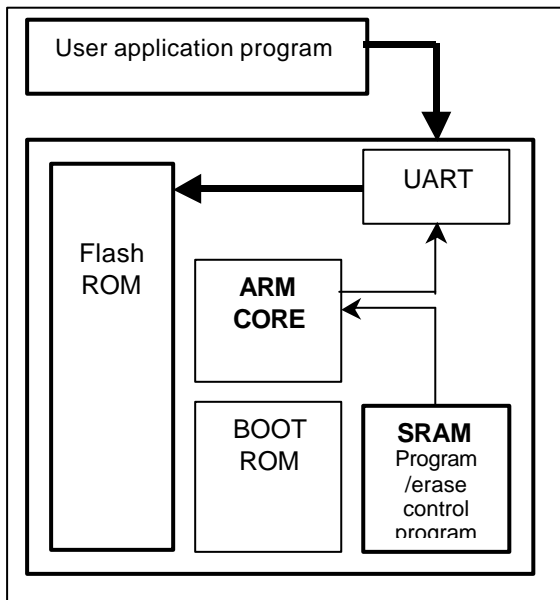
## Application example (User Program Mode)

### 1. Download Application Program



- Step 1. Set Serial Boot Mode to ' 1' Boot mode (mode 4 or 5)
- Step 2. Reset system
- Step 3. ARM runs Transfer Program in the flash memory, and transfer the program/erase control program to internal SRAM.

### 2. Run Downloaded Application Program



- Step 4. ARM branches to the start address of the Program/erase control program in SRAM
- Step 5. If flash memory has been programmed, Program/erase control program erase flash memory in block Units before program
- Step 6. Program/erase control program gets user application program from host through UART and executes flash memory program operation with the code in the erased flash memory block.
- Step 7. End the flash memory program operation and Host changes system mode to Normal.
- Step 8. Reset
- Step 9. ARM Execute New Program in the flash ROM.

**12.6 Flash Memory Programming/Erasing**

A software method, using the CPU, is employed to program and erase flash memory in the on-board programming modes. There are five flash memory operation modes: pre-program/program mode, post-program mode, erase mode, pre-program/program-verify mode, and erase verify mode. The transitions to these modes are made by setting CONTREG register.

The flash memory cannot be read while being programmed or erased. Therefore, the program (user program) that controls flash memory program/erase should be located and executed in on-chip RAM or external memory.

**12.6.1 Program & Program-Verify Mode**

When writing data or programs to flash memory, the program flowchart shown in figure 12.5 should be followed. Flash Memory of HMS39C7092 can be programmed 16bits at one time. In Program Verify, the data written in program mode is read to check whether it has been correctly written in the flash memory. If result of verify read at a certain address is not same as the programmed data of this address, program must be retried to the time when Verify read result and the programmed data are matched.

However, if the program/program verify sequence is repeated N times and verify read result is not same as programmed data, it is program fail.

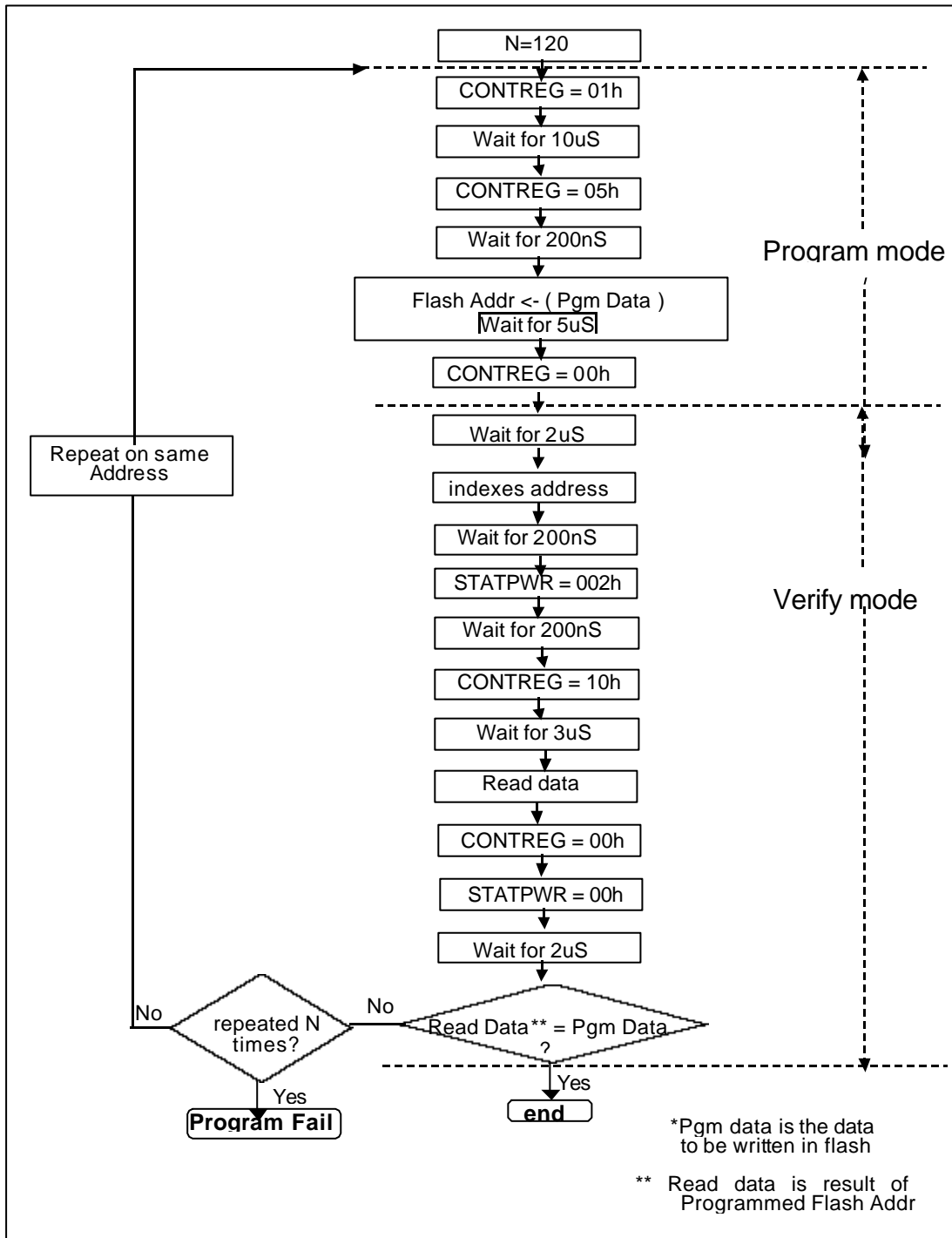


Figure 12.5 Flash Program & Program Verify Sequence

### 12.6.2 Pre-program & Pre-program Verify Mode

This is the first step of flash memory erase algorithm. Pre-program & Pre-program Verify must be done before block erase.

The difference between Program and Pre-program is that the purpose of Pre-program is programming not-programmed cell in a certain block that will be erased. Due to Pre-programming before block erase, every cell in the block that will be erased goes to program state, so it is possible to prevent cell from being over-erased after block erase.

When Pre-program mode, program address must start at first address of block to be erased, and increase by 2 to the last address of that block.

The relation between each erase block and corresponding flash memory address is shown at the Table 12.5 of chapter 12.3.

Pre-program needs to do pre-program verify read to ensure that every cell in the block are programmed successfully.

The CONTREG setting are the same as program & program verify mode.

The Flow of pre-program and pre-program verify is shown at Figure 12.6



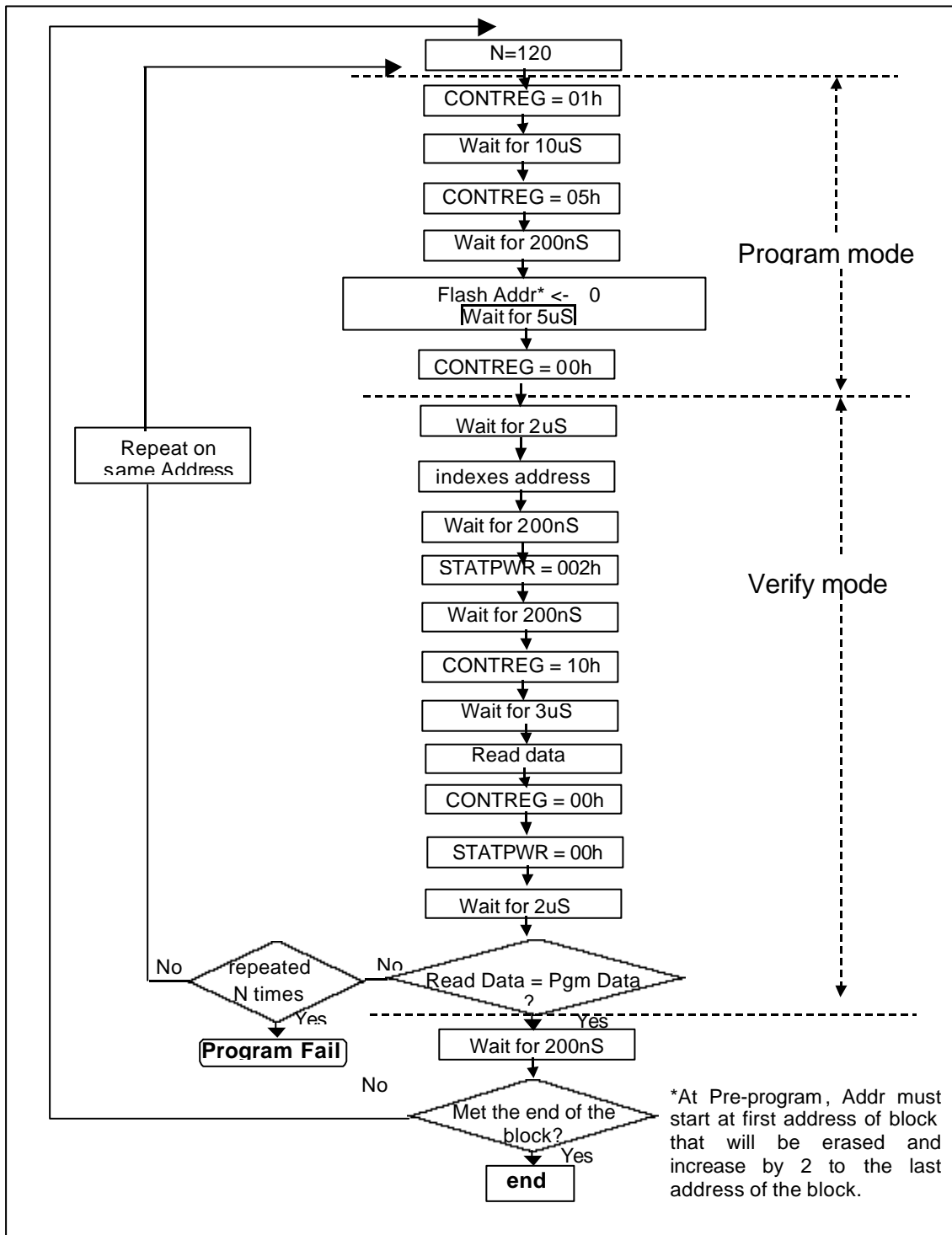


Figure 12.6 Flash Pre-program &amp; Pre-program Verify Sequence

### 12.6.3 Erase & Erase Verify Mode

Flash memory erase operation are performed block by block. To erase flash memory, make a setting for the flash memory area to be erased in erase block register(EBR). If multiple bits of EBR register are set, multiple block are erased at one time. The Maximum number of blocks that can be erased at one time is four. After Erase, it is necessary to do Erase verify read to ensure that every cell in the block are erased successively. When Erase verify read mode, verify address must start at first address of block to be erased, and increase by 2 to the last address of that block. The relation between each erase block and corresponding flash memory address is shown at the table 12.5 of chapter 12.3.

If the result of verify read at a certain address is not 0xFFFF, erase must be retried until the result of Verify read is 0xFFFF. However, if the erase/erase verify sequence is repeated N times and the result of verify read is not 0xFFFF, device is erase fail. The Flow of erase & erase verify is shown at Figure 12.7.

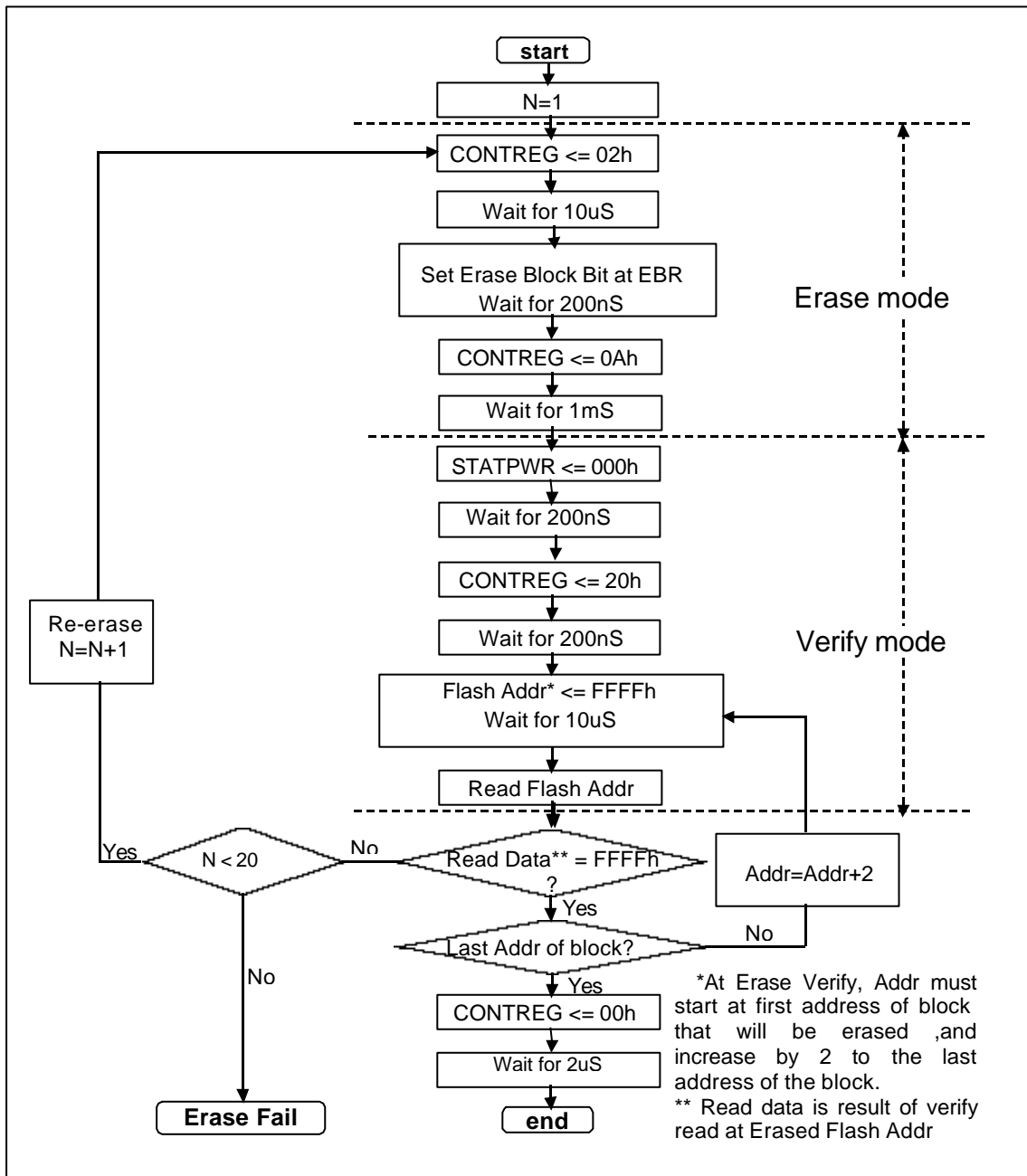


Figure 12.7 Flash Erase &amp; Erase Verify Sequence

12.6.4 Erase Algorithm

When erasing flash memory, the sequence of Figure 12.8 should be followed. It is composed of pre-program & pre-program verify, erase & erase-verify and post-program to prevent all cells in the erased block from being over-erased.

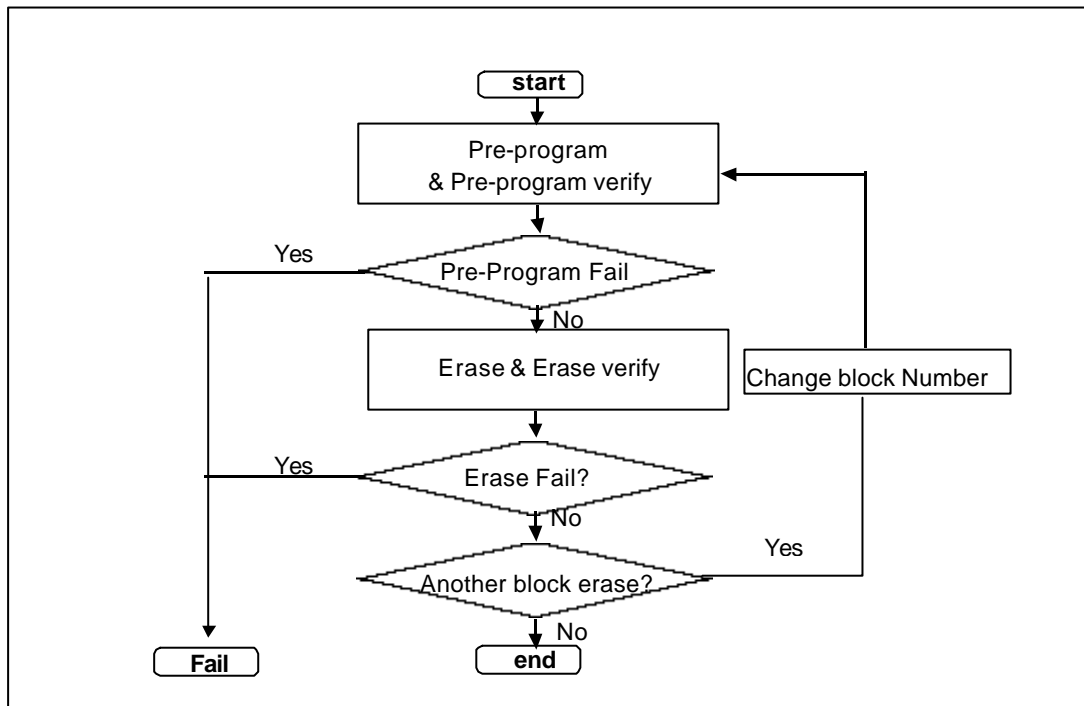


Figure 12.8 Flash Erase Algorithm

## 12.7 Flash Memory PROM Mode

The HMS39C7092 has a PROM mode as well as the on-board programming modes for programming and erase flash memory. In PROM mode, the on-chip flash memory can be freely programmed using a PROM writer.

### 12.7.1 PROM Mode Setting

By setting FR\_SEL signal, internal register of flash memory are directly write or read through FD[15:0] as table 12.7. When value of FR\_SEL[2:0] is set and FWEB = rising\_edge, FD[15:0] signals are passed into the register that FR\_SEL select. When value of FR\_SEL[2:0] is set and FOEB is low, the register's value is read through the FD[15:0].

Table 12.8 shows how the different external pins are set to write and read internal register.

**Table 12.7 FR\_SEL Value for access to internal Register**

FR_SEL[2:0]	Register Read	Register Write
000	Read Data	Reserved
001	ADDREG	ADDREG & DATAREG
010	DATAREG	Reserved
011	CONTREG	CONTREG
100	EBR	EBR
101	STATPWR[8:0]	STATPWR[5:0]

**Table 12.8 Setting for Register read/write**

Register Mode	Pin Name					
	FRSTB	FCEB	FWEB	FOEB	FD[15:0]	FA[16:0]
Read	High	Low	High	Low	Read value output	Address input
Write	High	Low	Rising edge	High	Write Value input	Address input

### 12.7.2 Memory Map

The memory map of PROM mode are shown at Table 12.9

At PROM mode, on-chip flash is 96K x 16 memory. Therefore, In order to access very next 16bit data to the currently accessed address, address should be changed by '1'(not by '2'), Erase operation is performed by sector, and corresponding address of each sector are shown.

**Table 12.9 Erase Block Register**

Sector #	Sector Size	FA [16:0]
Sector 0	8KB (4K-word)	0x00000 ~ 0x00FFF
Sector 1	8KB	0x01000 ~ 0x01FFF
Sector 2	24KB	0x02000 ~ 0x04FFF
Sector 3	24KB	0x05000 ~ 0x07FFF
Sector 4	32KB	0x08000 ~ 0x0BFFF
Sector 5	32KB	0x0C000 ~ 0x0FFFF
Sector 6	32KB	0x10000 ~ 0x13FFF
Sector 7	32KB	0x14000 ~ 0x17FFF

### 12.7.3 PROM Mode Operation

Each flash memory operation, such as program, erase, read are made by writing and reading the flash memory internal register. Table 12.10 shows different flash memory operation and register read/write sequence of each operation. Every operation except for memory normal read and erase, the 1st and 2nd cycles are deciding which operation will be performed, and 3rd cycle is setting flash memory address to be programmed or verified. Therefore, only 3rd cycle need to be repeated if another flash memory address is programmed or verified repeatedly after first address. At Erase operation, 2'nd and 3'rd cycle have to be repeated.

At Verify read operation (Pre-Program/program Verify and erase verify), In order to get the result of verify read, it is necessary to execute memory normal read operation after 3'rd cycle.

Table 12.10 Setting for Flash PROM read/write

Operation	1 <sup>st</sup> Cycle			2 <sup>nd</sup> Cycle			3 <sup>rd</sup> Cycle		
	FR_SEL	Mode	Address	FR_SEL	Mode	Address	FR_SEL	Mode	Address
			Data			Data			Data
Memory Normal Read	000	R	RA						
			Dout						
Memory Program /Pre-program	011	W	X	011	W	X	001	W	WA
			0X0001			0X0005			Din
Memory Post-program	011	W	X	011	W	X	001	W	WA
			0X0001			0X0041			Din
Memory Erase	011	W	X	100	W	X	011	W	X
			0X0002			BN			0X000A
Program/Pre-program verify	101	W	X	011	W	X	001	W	RA
			0X0001			0X0010			X
Erase Verify read	101	W	X	011	W	X	001	W	RA
			0X0000			0X0020			X

&lt; Legend &gt;

RA: Read Address WA: Write address Dout: Read data Din: Program data  
 X: don't care R: Read W: Write BN: Erase Block Number(see Table 12.6)

#### 12.7.4 Timing Diagram and AC/DC Characteristics

This timing diagram follows the sequence that is shown on Table 12.11.

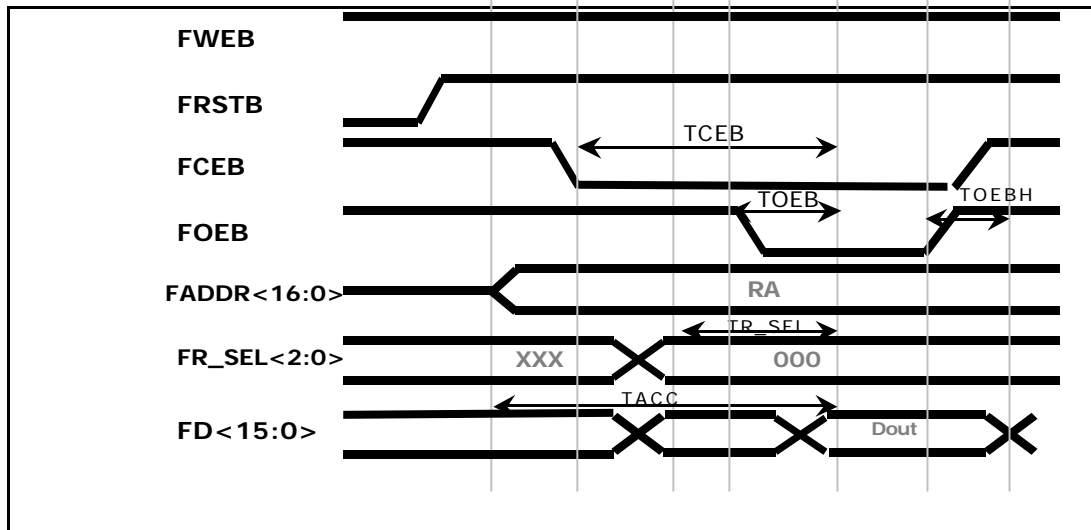


Figure 12.9 Timing Diagram of Read

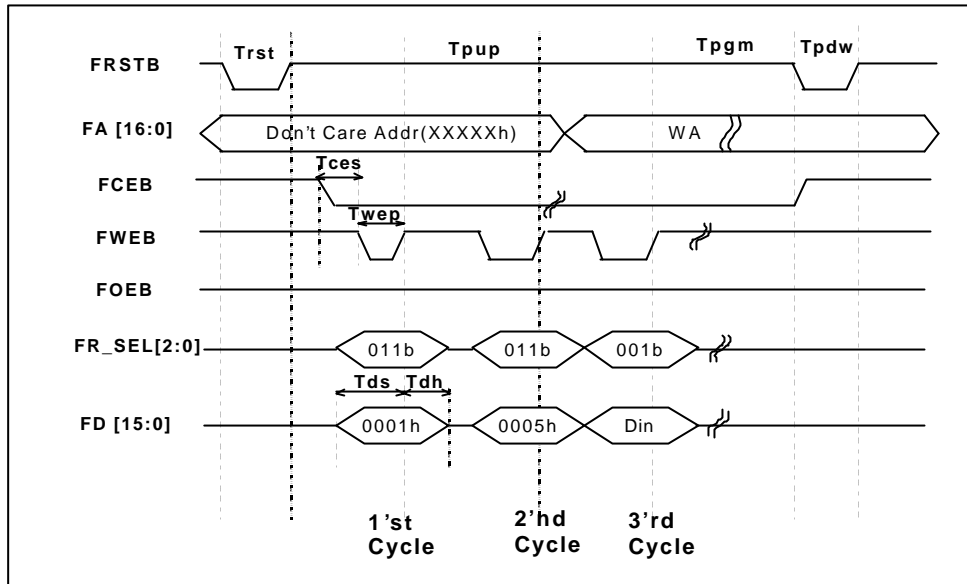


Figure 12.10 Timing Diagram of Pre-Program/Program

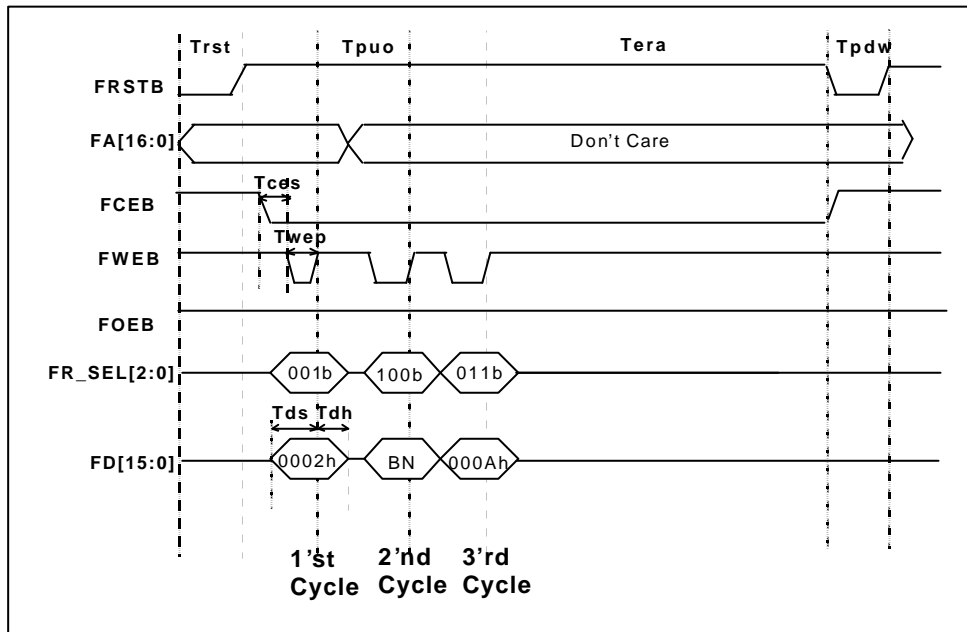


Figure 12.11 Timing Diagram of Erase

< Legend >  
 RA: Read Address WA: Write address Dout: Read data Din: Program data  
 X: don't care R: Read W: Write BN: Erase Block Number(see Table 12.6)



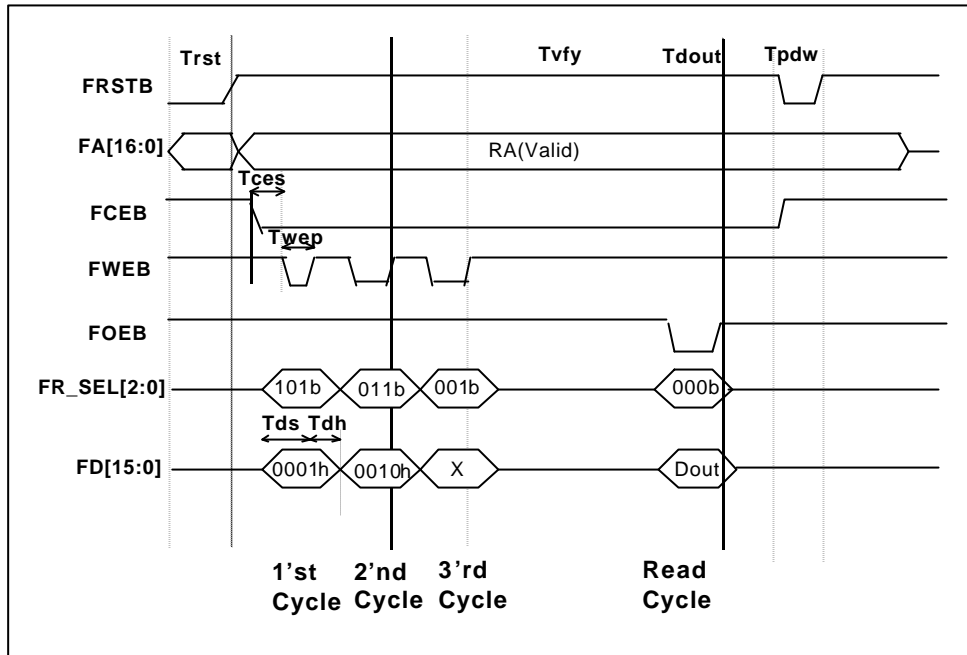


Figure 12.12 Timing Diagram of Pre-Program/Program Verify

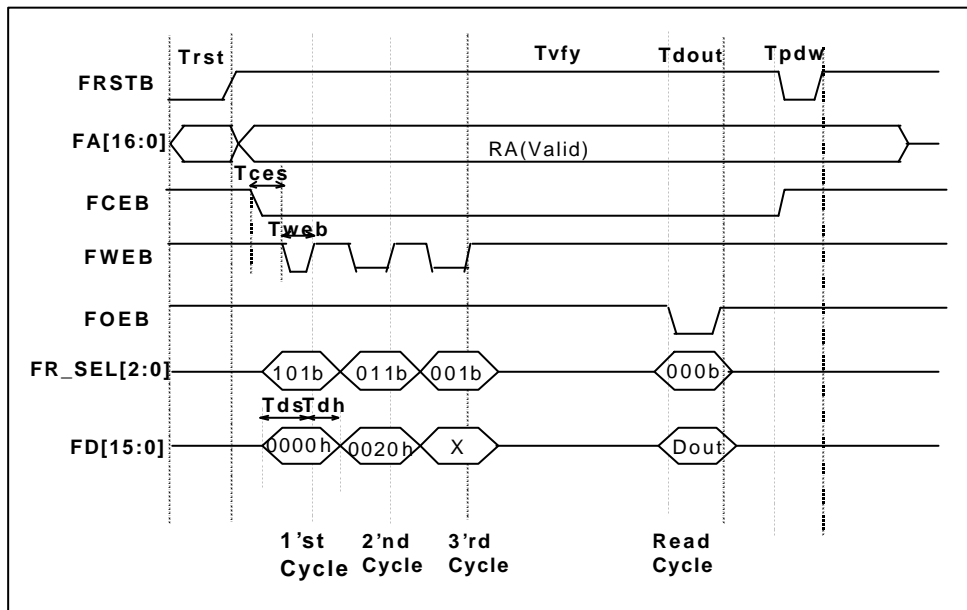


Figure 12.13 Timing Diagram of Erase Verify

< Legend >  
 RA: Read Address WA: Write address Dout: Read data Din: Program data  
 X: don't care R: Read W: Write BN: Erase Block Number(see Table 12.6)

**Table 12.11 DC Characteristics** - Preliminary  
 (  $V_{DD} = 3.3V \pm 10\%$ ,  $V_{SS} = 0V$ ,  $V_{SS} = 0V$ ,  $FTVPPD = 5V \pm 10\%$ ,  $T_a = 25^\circ C \pm 10\%$  )

Item	Symbol	Min	Typ	Max	Unit	Test Condition
Input high voltage	Vih	$0.7 \times V_{DD}$	---	$V_{DD} + 0.5$	V	
Input low voltage	Vil	-0.5	---	$0.3 \times V_{DD}$	V	
Output high voltage	Voh	2.4	---	---	V	Ioh=0.8mA
Output low voltage	Vol	---	---	0.4	V	Iol=0.8mA
Vcc current	Reading	Idd	---	20	40	mA
	Programming	Idd	---	40	80	mA
	Erasing	Idd	---	25	50	mA
FXTVPPD Current	Programming	Ippd	---	10	20	mA

**Table 12.12 AC Characteristics** - Preliminary  
 (  $V_{DD} = 3.3V \pm 10\%$ ,  $V_{SS} = 0V$ ,  $V_{SS} = 0V$ ,  $FXTVPPD = 5V \pm 10\%$ ,  $T_a = 25^\circ C \pm 10\%$  )

Item	Symbol	Min	Typ	Max	Unit
CEB output delay time	TCEB	---	70	90	ns
OEB output delay time	TOEB	---	5	10	ns
Output disable delay time	TOEBH	1	2	---	ns
R_SEL output delay time	TR_SEL	---	1	2	ns
Access time	TACC	---	70	90	ns
Reset Pulse Width	Trst	300	500	---	us
Power up time	Tpup	8	10	---	us
Discharge time(Program,Verify)	Tpdw	1	10	---	us
		10	20	---	us
Discharge time(Erase)					
Program time	Tpgm	30	30	---	us
CEB Setup time	Tces	100	200	---	us
WEB Pulse Width	Twep	100	200	---	ns
WEB rise time	Tr	---	20	30	ns
WEB fall time	Tf	---	20	30	ns
Data Setup time	Tds	50	150	---	ns
Data Hold time	Tdh	50	80	---	ns
Erase time	Tera	1	2	2	ms
Verify Setup time	Tvfy	8	10	---	us
Verify data out time	Tdout	2	2	---	us

Chapter 13  
**A/D Converter**

13.1 Overview

The HMS39C7092 has a 10-bit successive-approximations A/D converter with a selection of up to five analog input channels. The A/D converter has multiplexed five input channels. The serial output is configured to interface with standard shift registers. The differential analog voltage input allows for common-mode rejection or offset of the analog zero input voltage value. The voltage reference input can be adjusted to allow encoding any smaller analog voltage span to the full 10bits of resolution.

13.1.1 Features

A/D converter features are listed below.

- 10-bit resolution
- 5 input channels
- Selectable analog conversion voltage range:  
The analog voltage conversion range can be programmed by input of an analog reference voltage at the V<sub>REF</sub> pin.
- High-speed conversion:  
Conversion time: minimum 2us per channel (with 8Mhz ADC clock)
- Analog input range: GND ~ AVREF
- Five 10-bit data registers
- A/D conversion results are transferred for storage into data registers corresponding to the channels.
- Sample-and-hold function
- A/D interrupt requested at end of conversion:  
At the end of A/D conversions, an A/D End Interrupt (ADI) can be requested.

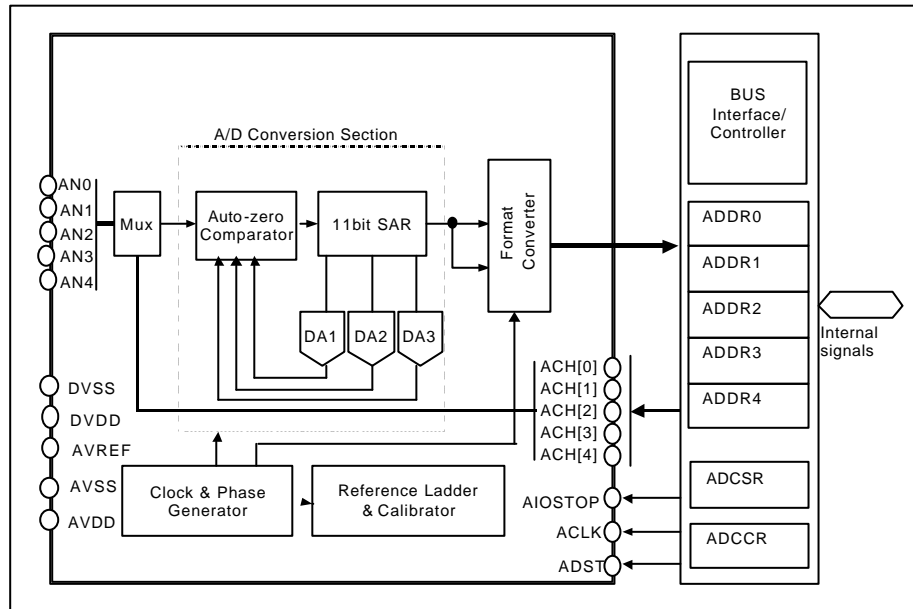


Figure 13.1 Block Diagram of A/D Converter

## 13.1.2 Pin Configuration

**Table 13.1** summarizes the A/D converter's input pins.  $AV_{DD}$  and  $AV_{SS}$  are the power supply for the analog circuits in the A/D converter.  $V_{REF}$  is the A/D conversion reference voltage.

**Table 13.1 A/D Converter Pins**

Pin Name	I/O	Function
$AV_{DD}$	Input	Analog power supply
$AV_{SS}$	Input	Analog ground
$AV_{REF}$	Input	Analog reference voltage
$AN_0$	Input	Analog input channel 0
$AN_1$	Input	Analog input channel 1
$AN_2$	Input	Analog input channel 2
$AN_3$	Input	Analog input channel 3
$AN_4$	Input	Analog input channel 4

13.2 A/D Converter Registers

The registers used to control the A/D converter when enabled are shown in **Table 13.2**. The base address of the A/D converter is **0x0900\_1700**.

**Table 13.2 Summarizes the A/D converter's registers.**

Reg.Name	I/O Offset	R/W	Name	Initial Value
<b>ADCSR</b>	0x1700	R/W	Control & Status Register	0x00
<b>ADCCR</b>	0x1704	R/W	Control Register	0x1
<b>ADDR0</b>	0x1708	R	Data Register 0	0x0000
<b>ADDR1</b>	0x170C	R	Data Register 1	0x0000
<b>ADDR2</b>	0x1710	R	Data Register 2	0x0000
<b>ADDR3</b>	0x1714	R	Data Register 3	0x0000
<b>ADDR4</b>	0x1718	R	Data Register 4	0x0000

13.2.1 Register Descriptions

**ADCSR AD Control & Status Register (0x0900\_1700 R/W)**

Bit	b7	b6	b5	b4	b3	b2	b1	b0
<b>ADCSR</b>	ADF	ADST	ADIE	ACKS		ACHS		
Init. Val.	0	0	0	0	0	0	0	0
RD/WR	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial Value: 0x00

- ACHS** Channel select (Select the analog input channel)  
 000 : Analog input channel 0  
 001 : Analog input channel 1  
 010 : Analog input channel 2  
 011 : Analog input channel 3  
 100 : Analog input channel 4
- ACKS** Clock select (Select the A/D conversion time)  
 00 : 1/8 times of the ADC input clock (ADCLK)  
 01 : 1/4 times of the ADC input clock (ADCLK)  
 10 : 1/2 times of the ADC input clock (ADCLK)  
 11 : ADC input clock is from the timer block
- ADIE** A/D interrupt enable (Enables and disables A/D conversion)  
 0 : A/D end interrupt request (INT\_ADC) is disabled.  
 1 : A/D end interrupt request (INT\_ADC) is enabled.
- ADST** A/D start (Starts or stops A/D conversion)  
 0 : A/D conversion is stopped  
 1 : A/D conversion start; ADST is automatically cleared to 0 when conversion end.

- ADF** A/D end flag (Indicates end of A/D conversion)  
 0 : [Clearing condition] Read when ADF=1, then write 0 in ADF.  
 1 : [Setting condition] Automatically set when conversion end

ADCSR is the control and status register for AD converter. ACH[2:0] is used for selection of the analog input channel. CKS[1:0] is used for selection of the AD converter input clock. When these signals are '00', the main clock of the A/D converter is 1/8 times of input clock (ADCLK), which is the same cycle with the system operation clock.. When these signals are '01', then the main clock of the A/D converter is 1/4 times of ADCLK. When these signals are '10', then the main clock of the A/D converter is 1/2 times of ADCLK. When these signals are '11', then the main clock of the A/D converter is external clock from Timer block. ADIE bit is interrupt enable signal. When this signal is '0', then A/D converter does not generate the interrupt of the end of A/D conversion. When this signal is '1', then A/D converter generates the end of conversion interrupt. ADST bit indicate the start of A/D conversion. When this signal is '1', the A/D converter start the A/D conversion and remain high during A/D conversion. ADF indicate the end of A/D conversion. When this bit is '1', then A/D converter indicates the end of A/D conversion. This signal is auto-cleared by reading this bit.

**ADCCR****AD Control Register (0X0900\_1704 R/W)**

Bit	b15-b2	b1	b0
<b>ADCCR</b>	Reserved	CALEND	AIOSTOP
Init. Val.		0	1
RD/WR		R	R/W

Initial value: 0x01

bit field: I0-I1

- CALEND** Calibration end (Indicate the calibration end time)  
 0 : Indicate not end of calibration time.  
 1 : Indicate the end of calibration time.

- AIOSTOP** Power save mode  
 0 : A/D converter is normal operation mode  
 1 : A/D converter is power save mode, so not operate

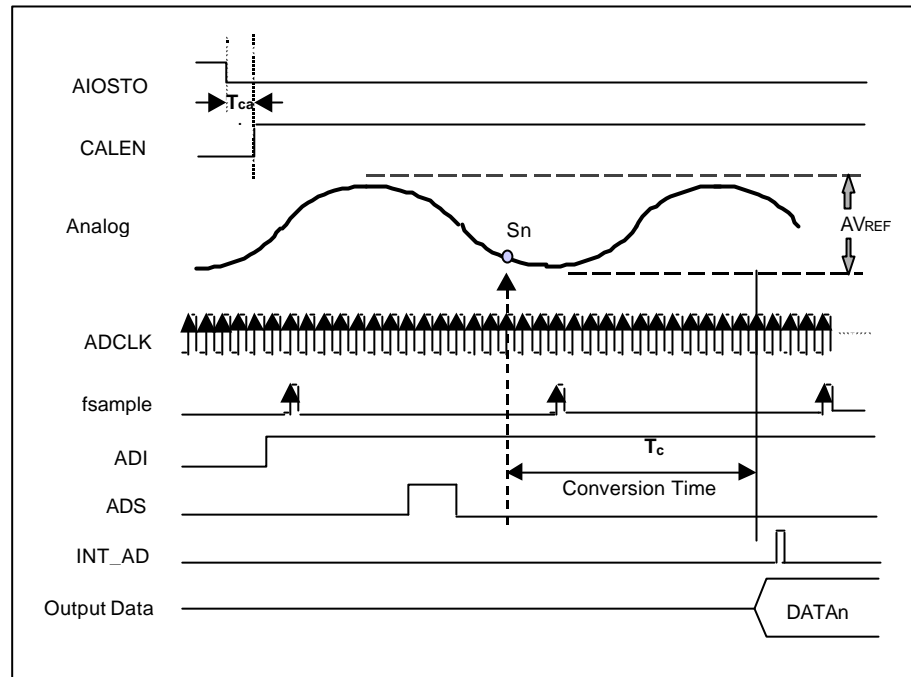
CALEND indicate the end of calibration time ( $T_{cal}$ ). This signal is read only. AIOSTOP is used to set the power save mode of A/D converter. When this signal is '0', then A/D converter is entering normal operation mode after calibration time, or power up time. See **Figure 13.2 A/D converter operation** for detailed timing diagram.





### 13.3 Operation

The A/D converter operates by successive approximations with 10-bit resolution. **Figure 13.2** show the operation of A/D converter.



**Figure 13.2** A/D converter Operation

**13.4 Interrupts**

The A/D converter generates an interrupt (INT\_ADC) at the end of A/D conversion. The INT\_ADC interrupt request can be enabled or disabled by the ADIE bit in ADCSR.

### 13.5 Usage Notes

When using the A/D converter, note the following points:

1. Analog Input Voltage Range: During A/D conversion, the voltages input to the analog input pins  $AN_n$  should be in the range  $AV_{SS} \leq AN_n \leq V_{REF}$ .
2. Relationships of  $AV_{CC}$  and  $AV_{SS}$  to  $V_{CC}$  and  $V_{SS}$ :  $AV_{CC}$ ,  $AV_{SS}$ ,  $V_{CC}$ , and  $V_{SS}$  should be related as follows:  $AV_{SS} = V_{SS}$ .  $AV_{CC}$  and  $AV_{SS}$  must not be left open, even if the A/D converter is not used.
3.  $V_{REF}$  Programming Range: The reference voltage input at the  $V_{REF}$  pin should be in the range  $V_{REF} \leq AV_{CC}$ .
4. Note on Board Design: In board layout, separate the digital circuits from the analog circuits as much as possible. Particularly avoid layouts in which the signal lines of digital circuits cross or closely approach the signal lines of analog circuits. Induction and other effects may cause the analog circuits to operate incorrectly, or may adversely affect the accuracy of A/D conversion. The analog input signals ( $AN_0$  to  $AN_4$ ), analog reference voltage ( $V_{REF}$ ), and analog supply voltage ( $AV_{CC}$ ) must be separated from digital circuit by the analog ground ( $AV_{SS}$ ). The analog ground ( $AV_{SS}$ ) should be connected to a stable digital ground ( $V_{SS}$ ) at one point on the board.
5. Note on Noise: To prevent damage from surges and other abnormal voltages at the analog input pins ( $AN_0$  to  $AN_4$ ) and analog reference voltage pin ( $V_{REF}$ ), connect a protection circuit like the one in figure 13.3 between  $AV_{CC}$  and  $AV_{SS}$ . The bypass capacitors connected to  $AV_{CC}$  and  $V_{REF}$  and the filter capacitors connected to  $AN_0$  to  $AN_4$  must be connected to  $AV_{SS}$ .
6. A/D Conversion Accuracy Definitions: A/D conversion accuracy in the HMS39C7092 is defined as follows:
  - Resolution: Digital output code length of A/D converter
  - Offset error: Deviation from ideal A/D conversion characteristic of analog input voltage required to raise digital output from minimum voltage value 0000000000 to 0000000001 (figure 13.4)
  - Full-scale error: Deviation from ideal A.D conversion characteristic of analog input voltage required to raise digital output from 1111111110 to 1111111111 (figure 13.4)
  - Quantization error: Intrinsic error of the A/D converter; 1/2 LSB (figure 13.5)
  - Nonlinearity error: Deviation from ideal A/D conversion characteristic in range from zero volts to full scale, exclusive of offset error, full-scale error, and quantization error.
  - Absolute accuracy: Deviation of digital value from analog input value, including offset error, full-scale error, quantization error, and nonlinearity error.

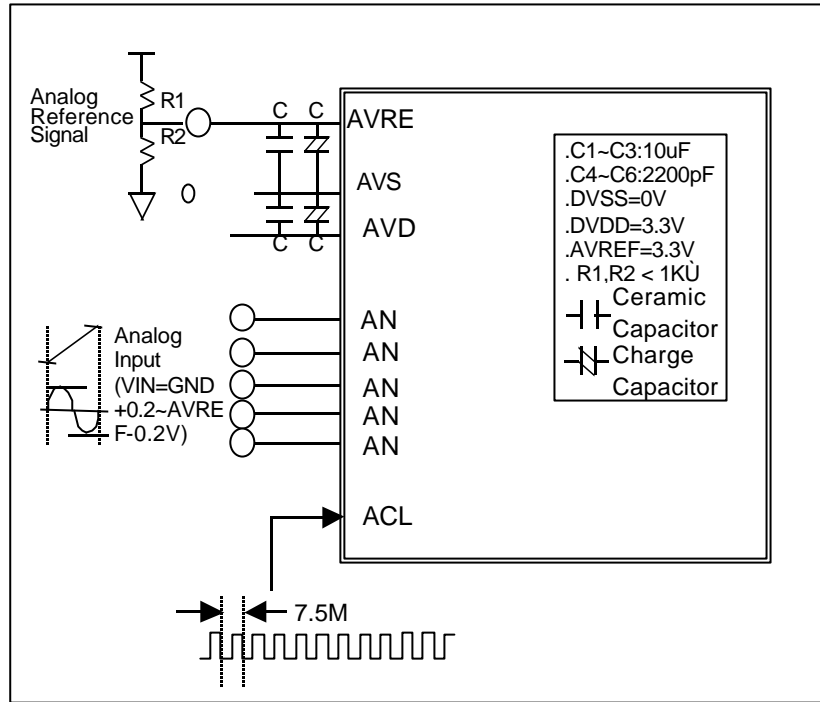


Figure 13.3 Example of Analog Input Circuit

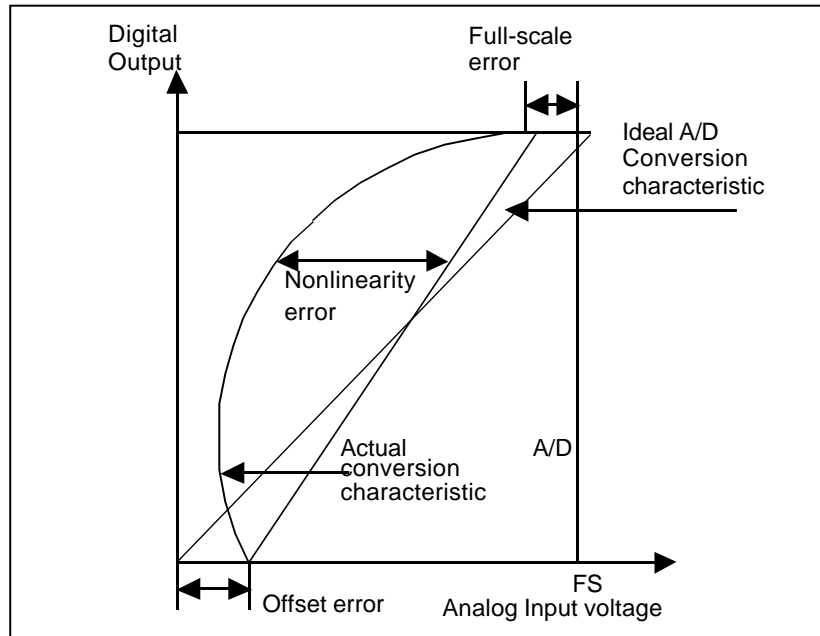
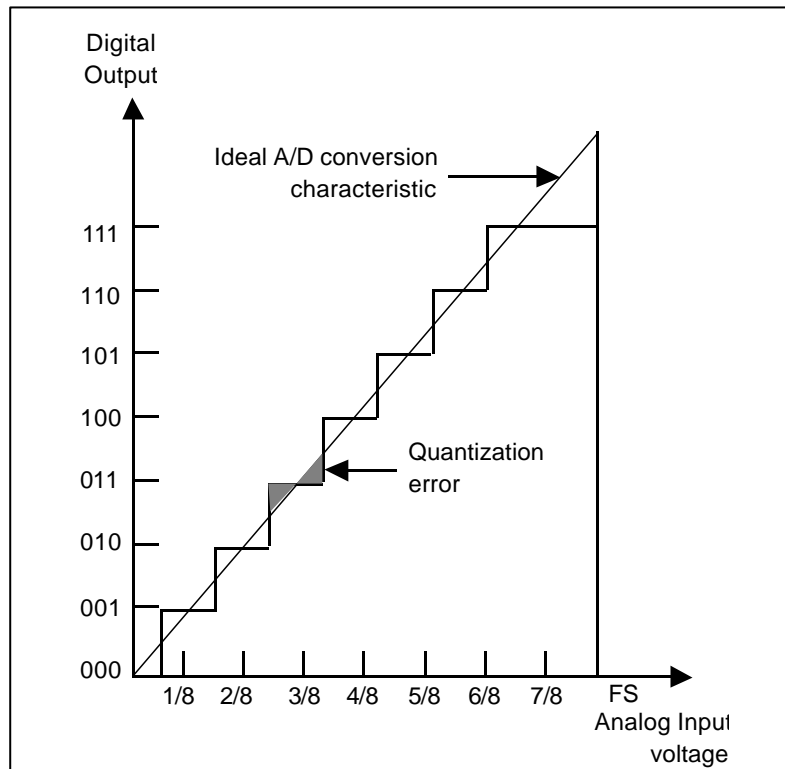


Figure 13.4 A/D Converter Accuracy Definitions (1)



**Figure 13.5 A/D Converter Accuracy Definitions (2)**

7. Effect on Absolute Accuracy: Attaching an external capacitor creates a coupling with ground, so if there is noise on the ground line, it may degrade absolute accuracy. The capacitor must be connected to an electrically stable ground, such as  $AV_{SS}$ . If a filter circuit is used, be careful of interference with digital signals on the same board, and make sure the circuit does not act as an antenna.

## 13.6 Example

```

AREA          ADDONE, CODE, READONLY
ENTRY

ldr          r0, =ADC_base          ; Make AOPSTOP to LOW to release power down mode,
add         r0, r0, #ADCCR          ; then set normal operation mode.
mov         r1, #0
str         r1, [r0]

loop
ldr         r2, [r0]                ; Check whether CALEND is set to 1 or not.
cmp         r2, #2                  ; (Check it' s in the range of calibration time)
bne        loop

ldr         r0, =ADC_base          ; Set the control bit in ADCSR register
add         r0, r0, #ADCSR          ; AD conversion start, CKS=1/8 ADCLK, ACH=0ch
mov         r1, #0x40              ;(set ADST, CKS=0, ACH=0)
str         r1, [r0]

loop_adf
;check ADF (AD conversion END)
adr         r2, [r0]
and         r2, r2, #0x80
cmp         r2, #0x80
bne        loop_adf
ldr         r1, [r0]
and         r1, r1, #0x7f ; Clear ADF to 0
str         r1, [r0]

ldr         r0, =ADC_base          ;read ADDR0 register into R1 register
add         r0, r0, #ADDR0
ldr         r1, [r0]

END

```

Chapter 14

**Electrical Characteristics**

## Electrical Characteristics

## Flash MCU(HMS39C7092)

### 14.1 Absolute Maximum Ratings

Table 14.1 lists the absolute maximum ratings(Note1 and 2).

Item	Symbol	Value
Power supply voltage	$V_{DD}$	-0.5V to 4.6V
DC Input Voltage (except I/O pins)	$V_{IN}$	-0.5V to 6.0V
DC Output Voltage (Output in high or low state)	$V_{OUT}$	-0.5V to $V_{DD}+0.5V$
DC Output Voltage (output in 3-state)	$V_{OUT}$	-0.5V to +6.0V
Reference Voltage	$V_{REF}$	-0.3V to $AV_{DD}+0.3$
Analog Power supply voltage	$AV_{CC}$	-0.3V to 3.6V
Analog Input Voltage	$V_{AN}$	-0.3V to $AV_{DD}+0.3$
Storage Temperature range	$T_S$	-65 to +150°C

Note1: Absolute maximum continuous ratings are those values which damage to the device may occur. Exposure to these conditions or conditions beyond those indicated may adversely affect device reliability. Functional operation under absolute-maximum-rated conditions is not implied.

Note2: Under transient conditions these ratings may be exceeded as elsewhere in this specification.

### 14.2 Recommended Operating Conditions:

Table 14.2 lists the recommended operating conditions.

Symbol	Parameter	MIN	MAX	UNIT
$V_{DD}$	Supply voltage	3.0	3.6	V
$V_{IN}$	Input voltage	0	5.5	V
$V_{OUT}$	Output voltage outputs active	0	$V_{DD}$	V
$V_{OUT}$	Output voltage outputs disabled	0	5.5	V
$V_{PPD}$	Flash program/erase voltage	4.5	5.5	V
$T_A$	Operating free-air temperature	0	70	°C



## 14.3 DC Characteristics

Table 14.3 lists the DC characteristics.

**Table 14.3 DC Characteristics**

**- Preliminary-**

ITEM	SYM BOL	MIN	MAX	UNIT	TEST Conditions
Input Low Voltage	$V_{IL}$	-0.5	$0.3XV_{DD}$	V	$V_{DD}=3.0V$ to $3.6V$
Input High Voltage	$V_{IH}$	$0.7XV_{DD}$	$V_{DD}+0.5$	V	$V_{DD}=3.0V$ to $3.6V$
Output Low Voltage	$V_{OL}$	-	0.4	V	$V_{DD}=3.0V$ $I_{OL}=0.8mA$
Output High Voltage	$V_{OH}$	2.4	-	V	$V_{DD}=3.0V$ $I_{OH}=-0.8mA$
Input current at maximum voltage	$I_I$	-	1	mA	$V_{DD}=3.0V$ to $3.6V$ Input= $5.5V$

Table 14.4 lists the IO circuit with pull-ups

**Table 14.4 IO Circuits with pull-ups**

**- Preliminary-**

	Min Current(at PAD = 0V)	Max Current (at PAD = 0V)
3.3V Pull-up	30uA	-146uA
Equivalent resistance	88.3kOhms	24.7kOhms

Table 14.5 lists the IO circuit with pull-downs

**Table 14.5 IO Circuits with pull-downs**

**- Preliminary-**

	Min Current(at PAD = 2.65V)	Max Current (at PAD = 3.6V)
Pull-down	31uA	159uA
Equivalent resistance	85.5kOhms	22.6kOhms

#### 14.4 AC Characteristics

Timing measurement conditions is following that

Unless otherwise specified:

VDD = 3.3V

Junction Temperature = 25 degrees C

Process = Typical

Low voltage input signal input signal rising and falling edges switching time = 0.3ns

Clock timing parameters are listed in table 14.6, control signal timing parameters in table 14.7, and bus timing parameters in table 14.8.

**Table 14.6 Clock Timing**

**- Preliminary-**

Item	Symbol	Min.	Max.	Units	Test Conditions
Clock cycle time	$t_{CYC}$	20	1000	ns	Figure 14.3
Clock pulse low width	$t_{CL}$	10	-	ns	
Clock pulse high width	$t_{CH}$	10	-	ns	
Clock rise time	$t_{Cr}$	-	10	ns	
Clock fall time	$t_{Cf}$	-	10	ns	
Clock oscillator Settling time at reset	$t_{OSC1}$	20	-	ms	Figure 14.1

**Table 14.7 Control Signal Timing**

**- Preliminary-**

Item	Symbol	Min.	Max.	Units	Test Conditions
XnRES setup time	$t_{RESS}$	200	-	ns	Figure 14.2
XnRES pulse width	$t_{RESW}$	5	-	$t_{CYC}$	
Mode programming setup time	$t_{MDS}$	200	-	ns	

Table 14.8 Bus Timing

- Preliminary-

Item	Symbol	Min.	Max.	Units	Test Conditions
Address delay time	$t_{AD}$	-	20	ns	Figure 14.3
Address hold time	$t_{AH}$	0	-	ns	Figure 14.4
Read strobe delay time	$t_{RSD}$	-	20	ns	
Address strobe delay time	$t_{ASD}$	-	20	ns	
Write strobe delay time	$t_{WSD}$	-	20	ns	
Strobe delay time	$t_{SD}$	-	20	ns	
Write strobe pulse width 1	$t_{WSW1}$	20	-	ns	
Address setup time 1	$t_{AS1}$	10	-	ns	
Read data setup time	$t_{RDS}$	20	-	ns	
Read data hold time	$t_{RDH}$	0	-	ns	
Write data delay time	$t_{WDD}$	-	20	ns	
Write data setup time 1	$t_{WDS1}$	10	-	ns	
Write data hold time	$t_{WDH}$	0	-	ns	
Read data access time 1	$t_{ACC1}$	-	40	ns	
Read data access time 3	$t_{ACC3}$	-	40	ns	
Precharge time 1	$t_{PCH1}$	20	-	ns	Figure 14.5
Precharge time 2	$t_{PCH2}$	0	-	ns	
Wait setup time	$t_{WTS}$	20	-	ns	
Wait hold time	$t_{WTH}$	5	-	ns	
Bus request setup time	$t_{BRQS}$	20	-	ns	Figure 14.6
Bus acknowledge time 1	$t_{BACD1}$	-	30	ns	
Bus acknowledge time 2	$t_{BACD2}$	-	30	ns	
Bus-floating time	$t_{BZD}$	-	30	ns	

14.4 AD Conversion characteristics (Preliminary)

Table 14.9 lists the operation conditions of the AD Conversion

**Table 14.9 Operating Conditions of the AD Conversion - Preliminary-**

Parameter	Symbol	Min.	Max.	Units
Power Supply	AVDD	3.0	3.6	V
Analog Input	AN	GND+0.2	AVREF-0.2	V
Clock Pulse Width	T <sub>PWL</sub>	62.5		ns
Operating Temperature	T <sub>OP</sub>	0	100	°C

Table 14.10 lists the electrical characteristics of the AD converter

**Table 14.10 Electrical characteristics of the AD converter - Preliminary-**

Conditions : Analog input frequency F<sub>IN</sub>=1.26KHz, ADCLK=7.5MHz,  
AVDD=DVDD=AVREF=3.3V T=25 °C

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Units
I <sub>DD</sub>	Normal	ADCLK=7.5MHz Input=AV <sub>REF</sub> F <sub>IN</sub> =1.26KHz ramp			2.0	mA
	Power Down	ADCLK=7.5MHz			50	uA
AN	Analog input voltage		GND+0.2		AV <sub>REF</sub> -0.2	V
Accuracy	Resolution				10	bits
INL	Integral Non-linearity	ADCLK=7.5MHz Input=0-AV <sub>REF</sub> (V) (F <sub>IN</sub> =1.26KHz)			±2.0	LSB
DNL	Differential Non-linearity	ADCLK=7.5MHz Input=0-AV <sub>REF</sub> (F <sub>IN</sub> =1.26KHz ramp)			±1.0	LSB
SNR	Signal-to-Noise Ratio	F <sub>sample</sub> =500Ksps, F <sub>IN</sub> =1.26KHz	48	54		dB
SNDR	Signal-to-Noise Distortion Ratio		45	54		dB
ADCLK			2	4	8	MHz
t <sub>c</sub>	Conversion time		2	4	8	us
C <sub>o</sub>	Output capacitance			20		pF
R <sub>ref</sub>	Reference resistance			10K		U
AV <sub>REF</sub>	Analog Reference Voltage				AV <sub>DD</sub>	V
T <sub>CAL</sub>	Power up time	Calibration time		22		ms
THD	Total harmonic distortion		55	60		dB
AVDD	Analog power		3.0	3.3	3.6	V
DVDD	Digital power		3.0	3.3	3.6	V
F <sub>IN</sub>	Analog input frequency				5	KHz

## 14.5 Operational Timing

### 14.5.1 Clock Timing

Figure. 14.1 shows the settling time of the crystal oscillator.

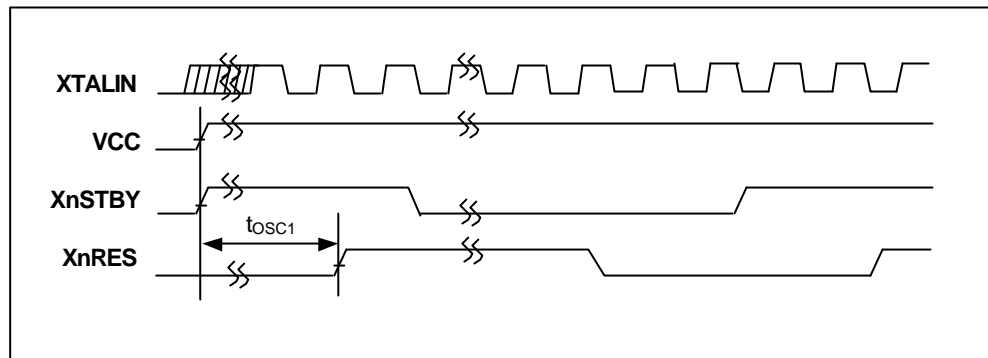


Figure 14.1 The settling time of the crystal oscillator

### 14.5.2 Reset Timing

Figure 14.2 show the reset input timing and reset output timing.

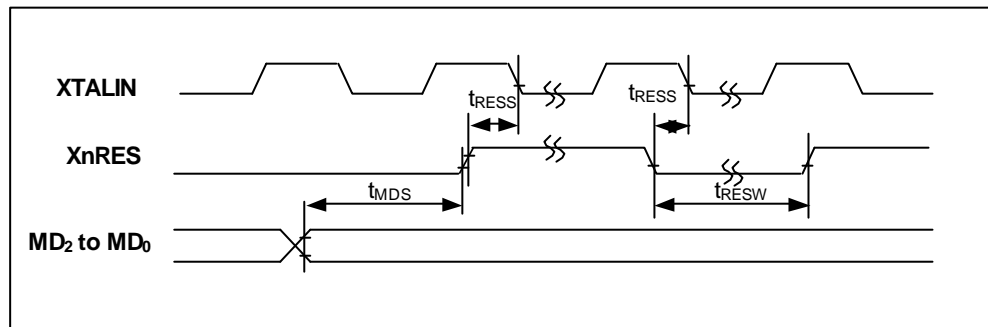


Figure 14.2 Reset Input Timing

14.5.3 Bus Timing

Figure 14.3 and Figure 14.6 show the timing diagram of the bus controller.

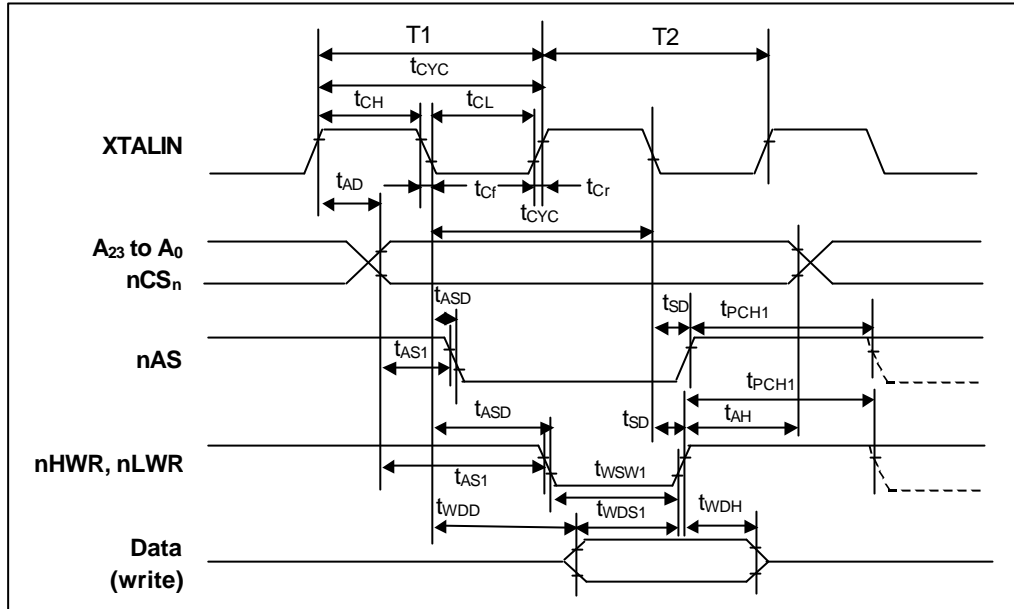


Figure 14.3 The Write Timing Diagram of the Bus Controller

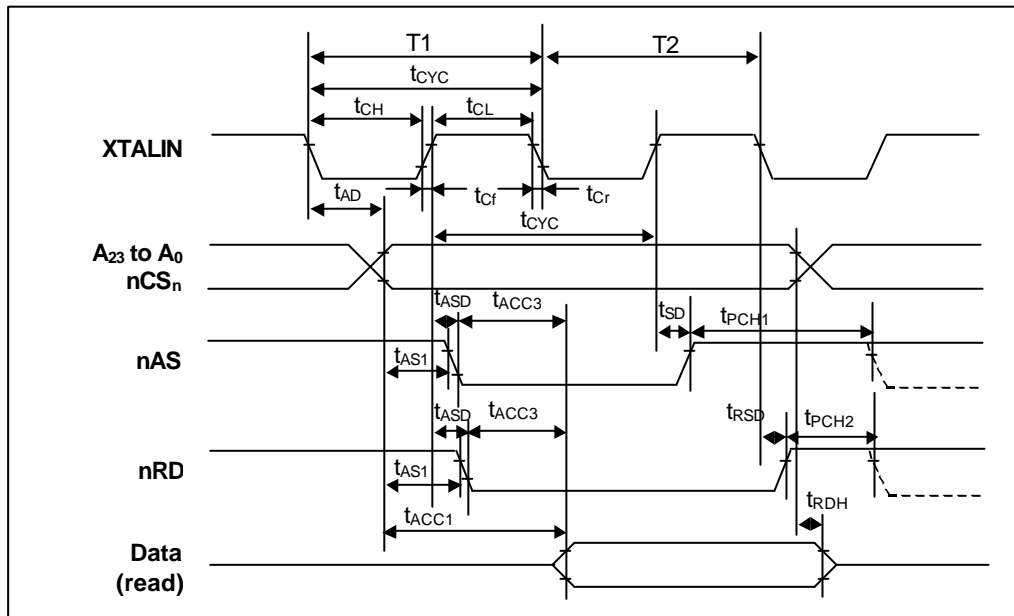


Figure 14.4 The Read Timing Diagram of the Bus Controller

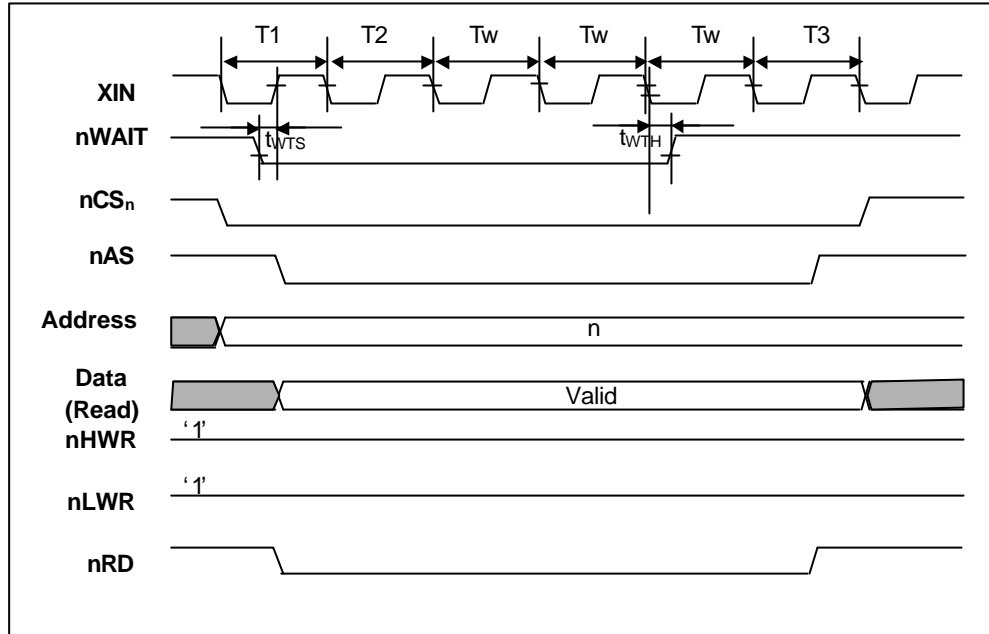


Figure 14.5 Basic Bus Cycle with External Wait State

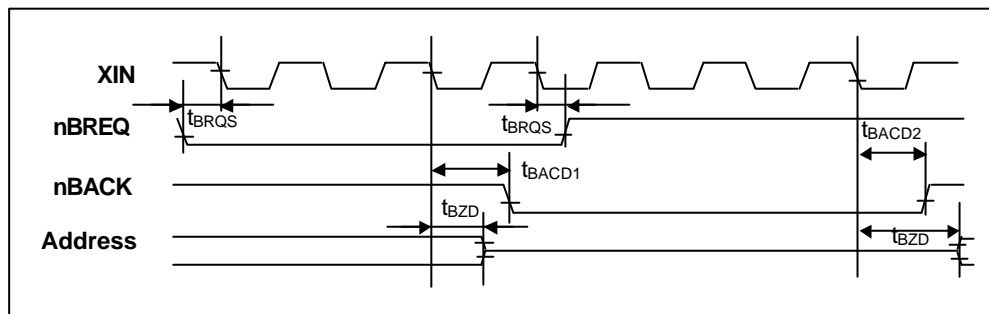


Figure 14.6 Bus Release Mode Timing