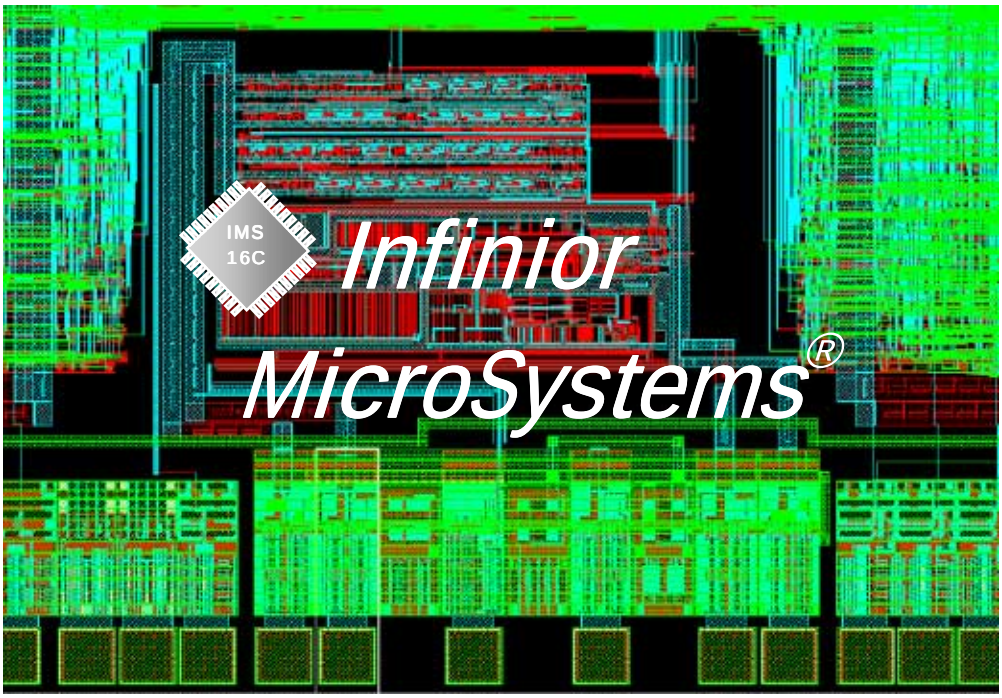
Infinior MicroSystems' IMS16C Tornado®
16bit Embedded processor

User's Manual



The IMS16C is available from the following manufacturer:

Infinior MicroSystems
5F Jeongam Bldg, 769-12, Yeoksam2-Dong, Gangnam-Gu,
Seoul, Korea 135-928
Tel: +82-2-6202-5500
FAX: +82-2-6202-5555
www.infinior.com
sales@infinior.com

2003-02-10 *applied_ ERRTA1_2*

Contents

1. Introduction

1.1. Overview

Infinior Microsystems is the leading provider of highly integrated IP cores, silicon and system solutions that enable Internet multimedia appliances. Using advanced design methodologies and proprietary technologies, We designs, develops and supplies SOC(System On a Chip) products and leading edge systems for communication markets, mobile computing, digital set-top box, VoIP/VoDSL and embedded networking devices.

Infinior Microsystems believes that one of its key competitive advantages is its high performance, highly integrated IP cores based on SOC design technologies encompassing the complete design space from system to silicon.
High performance 16bit microcontroller with variety of peripherals like SDRAM controller, MAC controller and HDLC controller still remains instruction level compatible with industry standard x86 microprocessors.

The IMS16B microcontroller is performance-enhanced implementations of the industry standard microprocessors with powerful up-to date peripherals such as SDRAM controller.

The IMS16C microcontroller is an enhanced version of IMS16B. In addition to IMS16B peripherals, IMS16C microcontroller also has 2 HDLC Controller and Internal 16-kbyte Sync RAM and 16 General Purpose Registers.

The IMS16N microcontroller is an enhanced version of IMS16B with 16-Mbyte memory address range. In addition to IMS16B peripherals, IMS16N microcontroller also has MAC Control Unit, HDLC controller and 8-kbyte instruction cache.

1.2. IMS16 Families and Road Map

	IMS16B	IMS16C	IMS16N
Speed	40/50/60MHz @3.3V	80/90MHz @3.3V	90/100MHz @3.3V
Address Bus	20bit	20bit	24bit
Data Bus	16bit	16bit	16bit
186/188 mode	186 mode only	186/188 mode	186/188 mode
Bus Timing	T1, T2, T3, T4	T1, T2, T3	T1, T2, T3
Wait State (UCS, LCS, MCS, PCS)	3, 3, 3, 7	15, 15, 15, 31	15, 15, 15, 31
5V Tolerant I/O	No	Yes	Yes
Package	128pin QFP 1420	128pin QFP 1420	144pin QFP
Power Consumption	1.65mA/MHz at 3.3V	1.3mA/MHz at 3.3V	
Deliverable	Silicon/IP/Verilog	Silicon/IP/Verilog	Silicon/IP/Verilog
Process	0.5um	0.35um	0.35um
Available	1Q 2001	2Q 2002	4Q 2002
Additional Peripherals			
UART	1	3	3
DMA Controller	2 External DMA	2 External DMA (2 UART DMA)	2 External DMA (2 UART DMA)
SDRAM Controller	Yes	Yes	Yes
GPIO	32 pin	44 pin	38 pin
Internal GPR	No	16bit x 16 General Purpose Register	16bit x 16 General Purpose Register
Internal Memory	No	16K bytes Sync RAM	No
Instruction Cache	No	No	16K bytes
WDT	S/W	H/W	H/W
Interrupt	6 Ext. /1 NMI	6 Ext. /1 NMI	6 Ext. /1 NMI
HDLC Controller	No	Yes (2 ch) (with 4 x 32bytes FIFO)	Yes (1 ch) (with 2 x 32bytes FIFO)
MAC Controller	No	No	Yes (10/100 Base T), MII I/F
Address Range	1MB	1MB	16MB

IMS16C

2. Key Features and Applications

The IMS16C is a performance-enhanced implementations of the industry standard AMD Am186EM® microprocessors with powerful up-to date peripherals such as SDRAM controller and two HDLC controllers. This microcontroller integrates the functions of the CPU, non-multiplexed address bus, timers, chip selects, interrupt controller, DMA controller, 3 asynchronous serial ports, 2 HDLC controllers, 16 general purpose registers, 16kbyte internal memory and 44 programmable I/O pins on one chip.

Compared to the existing 16bit microcontrollers, the IMS16C enables system designers to reduce the size, power consumption, and cost of embedded systems, while increasing functionality and performance. Development environments are also widely available.

The IMS16C has been totally re-architected from the ground up using the latest design techniques to produce an efficient, high clock-rate CPU core. It provides higher performance than Am186EM and the Intel 80C186 at the same clock frequency. Low power applications not needing this increased performance can run at 1/2 the clock frequency and still obtain more throughputs.

The IMS16C has been designed to meet the most common requirements of embedded products developed for the office automation, mass storage, communications and general embedded markets.
This device is ideal for use in a broad range of communications applications, including ISDN terminal adapters, low-end routers, digital subscriber line (xDSL) modems, PBX applications, digital phones and key telephone systems. This makes the IMS16C ideal devices for designs requiring high performance, serial communications, and a glueless bus interface to external memory systems.

2.1. Key Features

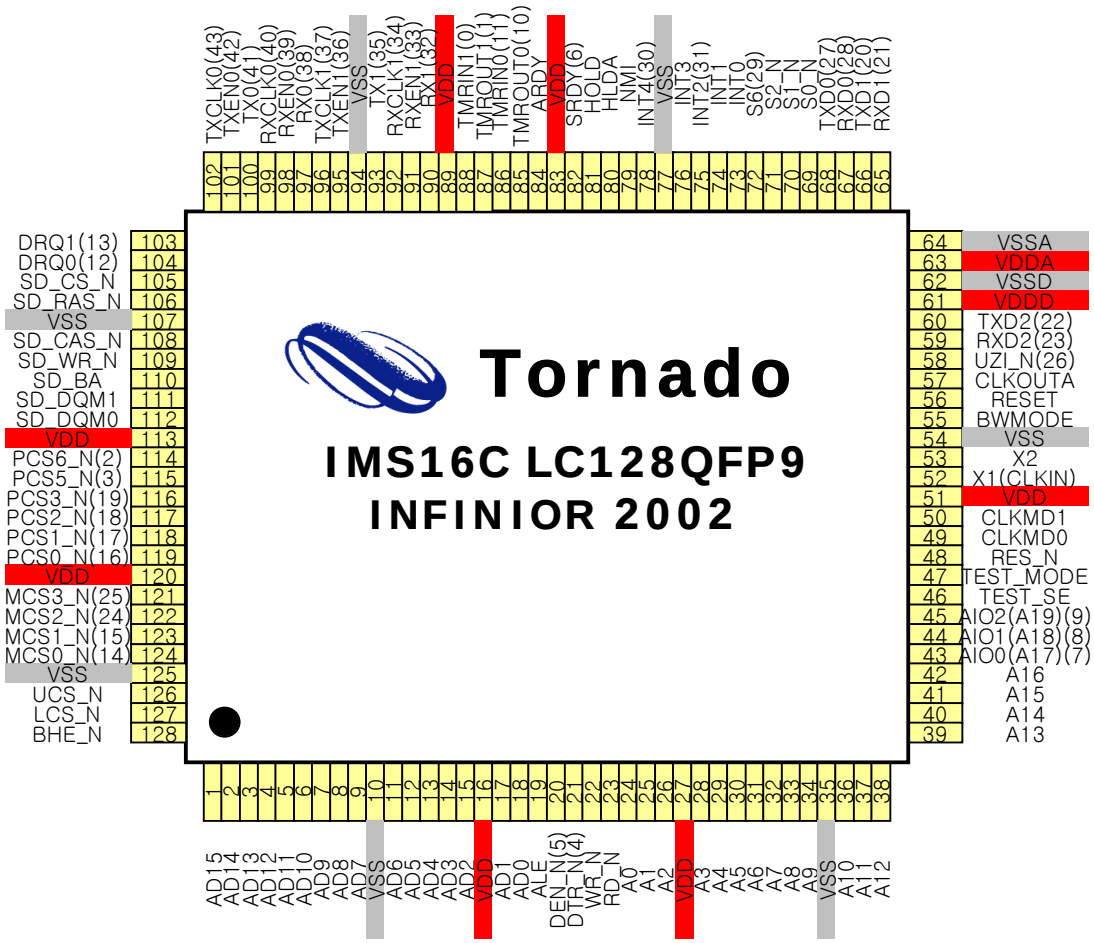
- 100% software compatible with the Intel 8086/80186®
- Supported by widely available native x86 development tools
- 1Mbyte Memory address space, 64Kbyte I/O space
- 16-bit ALU
- IMS16C includes all AMD Am186EM peripherals
  - Peripheral Interface Logic
  - Chip-Select and Ready Control Logic
  - 2 Channel DMA Controllers
  - 3 Programmable 16-bit Timers
  - Interrupt Controller
  - Power Save Logic
  - Chip Select Unit
  - Asynchronous Serial Port
  - Programmable I/O (PIO)
- 2 HDLC Controllers
- SDRAM Controller
- 2 UART Serial Ports with DMA Operation
- 16k bytes Internal Sync RAM
- 16 General Purpose Registers
- 44 General Purpose I/O Pins
- PLL Management Unit
- 81.1Mhz, 90.3Mhz, at 3.6864Mhz input clock and x6 Mode
- Fully synchronous and static design
- 80/90Mhz operation in 3.3V
- 0.35um 3.3V CMOS Process (commercial)
- 3.3V I/O pad (5V tolerant)
- 128-Pin Quad Flat Pack (QFP 1420)

2.2. Applications

- General Purpose Applications
  Toys, Automotive, Instrumentation, Medical, Process and Industrial control
- PC Peripheral Applications
  Disk and optical drive storage, Retail and consumer products, Various Peripheral Systems and PCI Add-on cards
- Telecommunication Applications
  Fax Machine, ADSL Modem cards, Key phone System
- Networking Applications
  VoIP Client, Intelligent Ethernet Hub, SOHO Router

3. System Overview

3.1. Pin Outs



3.2. Pin List

Pin No.	Pin Name	Pad Name	Pad Type	Pull-up/ Pull-down	Driving Current	Comments
1	**AD15**	PC3B02	IN/OUT		4mA	
2	**AD14**	PC3B02	IN/OUT		4mA	
3	**AD13**	PC3B02	IN/OUT		4mA	
4	**AD12**	PC3B02	IN/OUT		4mA	
5	**AD11**	PC3B02	IN/OUT		4mA	
6	**AD10**	PC3B02	IN/OUT		4mA	
7	**AD9**	PC3B02	IN/OUT		4mA	
8	**AD8**	PC3B02	IN/OUT		4mA	
9	**AD7**	PC3B02	IN/OUT		4mA	
10	**VSS**	VSS	GROUND			
11	**AD6**	PC3B02	IN/OUT		4mA	
12	**AD5**	PC3B02	IN/OUT		4mA	
13	**AD4**	PC3B02	IN/OUT		4mA	
14	**AD3**	PC3B02	IN/OUT		4mA	
15	**AD2**	PC3B02	IN/OUT		4mA	
16	**VDD**	POWER	POWER			
17	**AD1**	PC3B02	IN/OUT		4mA	
18	**AD0**	PC3B02	IN/OUT		4mA	
19	**ALE**	PC3O02	OUT		4mA	
20	**DEN_N**	PC3B02	IN/OUT		4mA	
21	**DTR_N**	PC3B02	IN/OUT		4mA	
22	**WR_N**	PC3T02	Tri-OUT		4mA	
23	**RD_N**	PC3T02	Tri-OUT		4mA	
24	**A0**	PC3T02	Tri-OUT		4mA	
25	**A1**	PC3T02	Tri-OUT		4mA	
26	**A2**	PC3T02	Tri-OUT		4mA	
27	**VDD**	VDD	POWER			
28	**A3**	PC3T02	Tri-OUT		4mA	
29	**A4**	PC3T02	Tri-OUT		4mA	
30	**A5**	PC3T02	Tri-OUT		4mA	
31	**A6**	PC3T02	Tri-OUT		4mA	
32	**A7**	PC3T02	Tri-OUT		4mA	

Pin No.	Pin Name	Pad Name	Pad Type	Pull-up/ Pull-down	Driving Current	Comments
33	**A8**	PC3T02	Tri-OUT		4mA	
34	**A9**	PC3T02	Tri-OUT		4mA	
35	**VSS**	VSS	GROUND			
36	**A10**	PC3T02	Tri-OUT		4mA	
37	**A11**	PC3T02	Tri-OUT		4mA	
38	**A12**	PC3T02	Tri-OUT		4mA	
39	**A13**	PC3T02	Tri-OUT		4mA	
40	**A14**	PC3T02	Tri-OUT		4mA	
41	**A15**	PC3T02	Tri-OUT		4mA	
42	**A16**	PC3T02	Tri-OUT		4mA	
43	**AIO0(A17)**	PC3B02	IN/OUT		4mA	5V Tolerant
44	**AIO1(A18)**	PC3B02	IN/OUT		4mA	5V Tolerant
45	**AIO2(A19)**	PC3B02	IN/OUT		4mA	5V Tolerant
46	**TEST_SE**	PC3D01	INPUT			5V Tolerant
47	**TEST_MODE**	PC3D01	INPUT			5V Tolerant
48	**RES_N**	PC3D21U	INPUT	Pull-up		5V Tolerant
49	**CLKMD0**	PC3D21	INPUT			5V Tolerant
50	**CLKMD1**	PC3D21	INPUT			5V Tolerant
51	**VDD**	VDD	POWER			
52	**X1(CLKIN)**	PC3X13	INPUT			Oscillator
53	**X2**	PC3X13	IN/OUT			Oscillator
54	**VSS**	VSS	GROUND			
55	**BWMODE**	PC3D21U	INPUT	Pull-up		5V Tolerant
56	**RESET**	PC3O02	OUT		4mA	
57	**CLKOUT**	PC3T02	Tri-OUT		4mA	
58	**UZI_N**	PC3B02U	IN/OUT	Pull-up	4mA	5V Tolerant
59	**RXD2**	PC3B02U	IN/OUT	Pull-up	4mA	5V Tolerant
60	**TXD2**	PC3B02U	IN/OUT	Pull-up	4mA	5V Tolerant
61	**VDDD**	VDDD	POWER			DIGITAL POWER
62	**VSSD**	VDDS	GROUND			DIGITAL GROUND
63	**VDDA**	VDDA	POWER			ANALOG POWER
64	**VSSA**	VSSA	GROUND			ANALOG GROUND

Pin No.	Pin Name	Pad Name	Pad Type	Pull-up/ Pull-down	Driving Current	Comments
65	**RXD1**	PC3B02U	IN/OUT	Pull-up	4mA	5V Tolerant
66	**TXD1**	PC3B02U	IN/OUT	Pull-up	4mA	5V Tolerant
67	**RXD0**	PC3B02U	IN/OUT	Pull-up	4mA	5V Tolerant
68	**TXD0**	PC3B02U	IN/OUT	Pull-up	4mA	5V Tolerant
69	**S0_N**	PC3T02	Tri-OUT		4mA	
70	**S1_N**	PC3T02	Tri-OUT		4mA	
71	**S2_N**	PC3T02	Tri-OUT		4mA	
72	**S6**	PC3B02U	IN/OUT	Pull-up	4mA	5V Tolerant
73	**INT0**	PC3D21D	INPUT	Pull-down		5V Tolerant
74	**INT1**	PC3D21D	INPUT	Pull-down		5V Tolerant
75	**INT2**	PC3B02U	IN/OUT	Pull-up	4mA	5V Tolerant
76	**INT3**	PC3B02U	IN/OUT	Pull-up	4mA	5V Tolerant
77	**VSS**	VSS	GROUND			
78	**INT4**	PC3B02U	IN/OUT	Pull-up	4mA	5V Tolerant
79	**NMI**	PC3D21D	INPUT	Pull-down		5V Tolerant
80	**HLDA**	PC3O02	OUT		4mA	
81	**HOLD**	PC3D21D	INPUT	Pull-down		5V Tolerant
82	**SRDY**	PC3B02	IN/OUT		4mA	5V Tolerant
83	**VDD**	VDD	POWER			
84	**ARDY**	PC3D21U	INPUT	Pull-up		5V Tolerant
85	**TMROUT0**	PC3B02U	IN/OUT	Pull-up		5V Tolerant
86	**TMRIN0**	PC3B02U	IN/OUT	Pull-up		5V Tolerant
87	**TMROUT1**	PC3B02U	IN/OUT	Pull-up		5V Tolerant
88	**TMRIN1**	PC3B02U	IN/OUT	Pull-up		5V Tolerant
89	**VDD**	VDD	POWER			
90	**RX1**	PC3B02U	IN/OUT	Pull-up	4mA	5V Tolerant
91	**RXEN1**	PC3B02U	IN/OUT	Pull-up	4mA	5V Tolerant
92	**RXCLK1**	PC3B02U	IN/OUT	Pull-up	4mA	5V Tolerant
93	**TX1**	PC3B02U	IN/OUT	Pull-up	4mA	5V Tolerant
94	**VSS**	VSS	GROUND			
95	**TXEN1**	PC3B02U	IN/OUT	Pull-up	4mA	5V Tolerant
96	**TXCLK1**	PC3B02U	IN/OUT	Pull-up	4mA	5V Tolerant

Pin No.	Pin Name	Pad Name	Pad Type	Pull-up/ Pull-down	Driving Current	Comments
97	**RX0**	PC3B02U	IN/OUT	Pull-up	4mA	5V Tolerant
98	**RXEN0**	PC3B02U	IN/OUT	Pull-up	4mA	5V Tolerant
99	**RXCLK0**	PC3B02U	IN/OUT	Pull-up	4mA	5V Tolerant
100	**TX0**	PC3B02U	IN/OUT	Pull-up	4mA	5V Tolerant
101	**TXEN0**	PC3B02U	IN/OUT	Pull-up	4mA	5V Tolerant
102	**TXCLK0**	PC3B02U	IN/OUT	Pull-up	4mA	5V Tolerant
103	**DRQ1**	PC3B02U	IN/OUT	Pull-up	4mA	5V Tolerant
104	**DRQ0**	PC3B02U	IN/OUT	Pull-up	4mA	5V Tolerant
105	**SD_CS_N**	PC3O02	OUT		4mA	
106	**SD_RAS_N**	PC3O02	OUT		4mA	
107	**VSS**	VSS	GROUND			
108	**SD_CAS_N**	PC3O02	OUT		4mA	
109	**SD_WR_N**	PC3O02	OUT		4mA	
110	**SD_BA**	PC3O02	OUT		4mA	
111	**SD_DQM1**	PC3O02	OUT		4mA	
112	**SD_DQM0**	PC3O02	OUT		4mA	
113	**VDD**	VDD	POWER			
114	**PCS6_N**	PC3B02U	IN/OUT	Pull-up	4mA	5V Tolerant
115	**PCS5_N**	PC3B02U	IN/OUT	Pull-up	4mA	5V Tolerant
116	**PCS3_N**	PC3B02U	IN/OUT	Pull-up	4mA	5V Tolerant
117	**PCS2_N**	PC3B02U	IN/OUT	Pull-up	4mA	5V Tolerant
118	**PCS1_N**	PC3B02U	IN/OUT	Pull-up	4mA	5V Tolerant
119	**PCS0_N**	PC3B02U	IN/OUT	Pull-up	4mA	5V Tolerant
120	**VDD**	VDD	POWER			
121	**MCS3_N**	PC3B02U	IN/OUT	Pull-up	4mA	5V Tolerant
122	**MCS2_N**	PC3B02U	IN/OUT	Pull-up	4mA	5V Tolerant
123	**MCS1_N**	PC3B02U	IN/OUT	Pull-up	4mA	5V Tolerant
124	**MCS0_N**	PC3B02U	IN/OUT	Pull-up	4mA	5V Tolerant
125	**VSS**	VSS	GROUND			
126	**UCS_N**	PC3T02	Tri-OUT		4mA	
127	**LCS_N**	PC3T02	Tri-OUT		4mA	
128	**BHE_N**	PC3T02	Tri-OUT		4mA	

3.3. Pad Description

Pad name	Comments
PC3X13	3.3V Crystal Oscillator Pads. 40Mhz ~ 100Mhz CLOCK INPUT
PC3D01	3.3V CMOS Input Only Pads
PC3D31U	3.3V CMOS Input Only Pads With Pull-up Register and Schmitt Trigger non-inverting
PC3D21	3.3V CMOS Input Only Pads With Schmitt Trigger non-inverting.
PC3D21U	3.3V CMOS Input Only Pads With Pull-up Register and Schmitt Trigger non-inverting.
PC3D21D	3.3V CMOS Input Only Pads With Pull-down Register and Schmitt Trigger non-inverting.
PC3O02	3.3V CMOS Output Pads with 4mA drive
PC3T02	3.3V CMOS 3-State Output Pads with 4mA drive
PC3B02	3.3V CMOS 3-State I/O Pads with 4mA drive
PC3B02D	3.3V CMOS 3-State I/O Pads With Pull-down Register with 4mA drive
PC3B02U	3.3V CMOS 3-State I/O Pads With Pull-up Register with 4mA drive

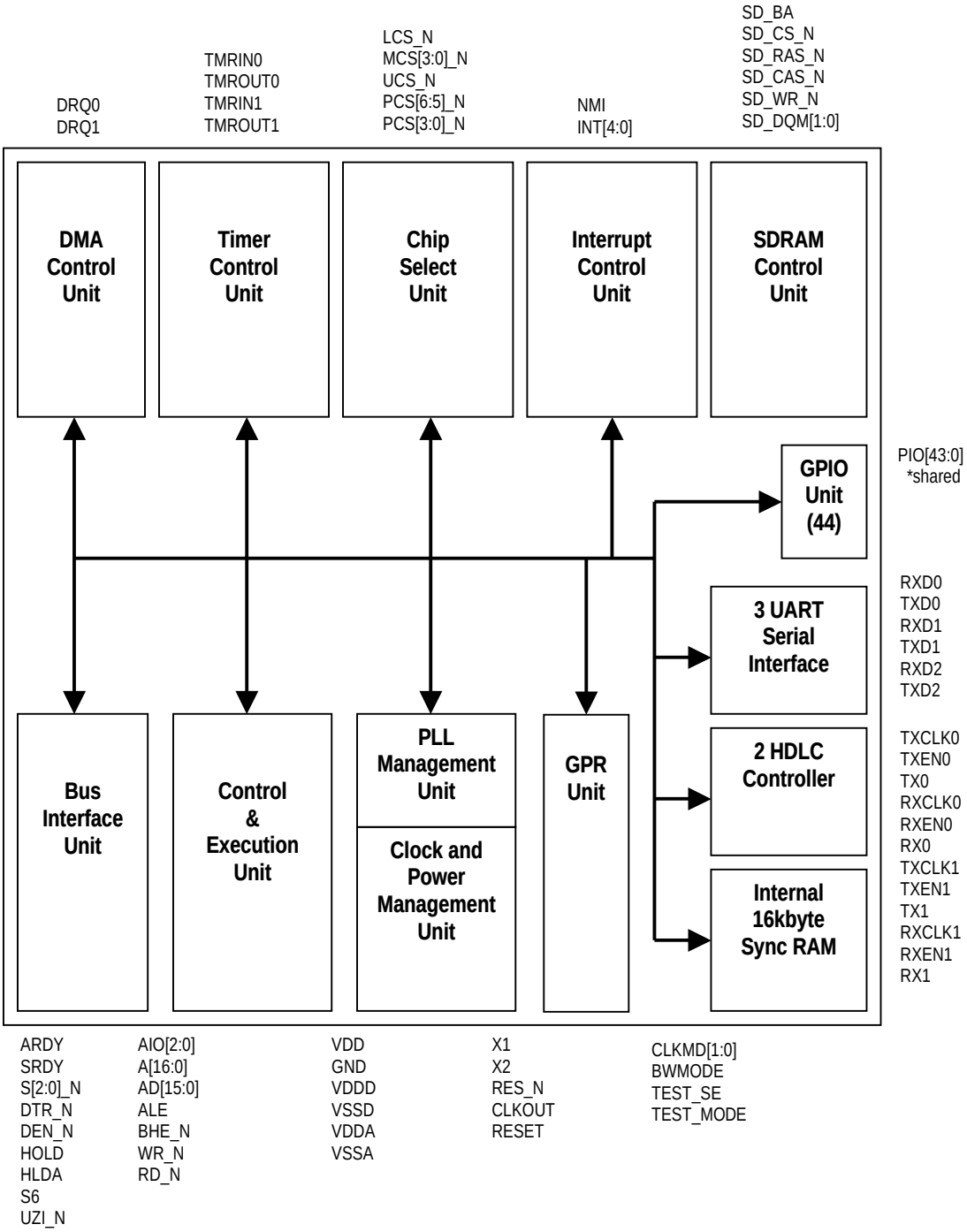
I/O Circuit Pullups
Unless otherwise specified, the following current values are used for I/Os with Internal pullup devices.

	Min Current(at Pad = 0 V)	Max Current (at Pad = 0V)
3.3V Pullup	-30 uA	-146 uA
Equivalent resistance	88.3 Kohms	24.7 Kohms

I/O Circuit Pulldowns
Unless otherwise specified, the following current values are used for I/Os with Internal pulldown devices.

	Min Current (at Pad = 2.65 V)	Max Current (at Pad = 3.6V)
Pulldown	31 uA	159 uA
Equivalent resistance	85.5 Kohms	22.6 Kohms

3.4. Block Diagram



3.5. Pin Description

<table>
<tr><th>Pin Name</th><th>Type</th><th>Description</th></tr>
<tr><td>A16 - A0</td><td>Address Bus (3-state output)</td><td>These pins supply non-multiplexed memory or I/O addresses to the system one-half of a CLKOUT period earlier than the multiplexed address and data bus. During a bus hold or reset condition, the address bus is in a high-impedance state.</td></tr>
<tr><td>AIO2 - AIO0 (A19/PIO9, A18/PIO8, A17/PIO7)</td><td>Address, PIO Bus (I/O)</td><td>These pins are used address A19-A17 and programmable Input/output pin PIO9-PIO7.</td></tr>
<tr><td>AD15-AD0</td><td>Address and Data Bus (I/O)</td><td>These time-multiplexed pins supply partial memory or I/O addresses, as well as data, to the system.
This bus supplies the 16bit address to the system during the first period of a bus cycle (t1), and it supplies data to the system during the remaining periods of that cycle (t2 and t3). During a bus hold or reset condition, the address and data bus is in a high-impedance state.</td></tr>
<tr><td>ALE</td><td>Address Latch Enable (output)</td><td>This pin indicates to the system that an address appears on the address and data bus. The address is guaranteed valid on the trailing edge of ALE. This pin is not three-stated during a bus hold or reset.</td></tr>
<tr><td>ARDY</td><td>Asynchronous Ready (input)</td><td>This pin indicates to the microcontroller that the addressed memory space or I/O device will complete a data transfer.
The ARDY pin accepts a rising edge that is asynchronous to CLKOUT and is active High. The falling edge of ARDY must be synchronized to CLKOUT. To always assert the ready condition to the microcontroller, tie ARDY High. If the system does not use ARDY, tie the pin Low to yield control to SRDY.</td></tr>
<tr><td>BHE_N</td><td>Bus High Enable (3-state output)</td><td>During a memory access, this pin and the least-significant address bit (AD0 or A0) indicate to the system, which bytes of the data bus (upper, lower, or both) participate in a bus cycle. The BHE_N and AD0 pins are encoded as shown in Table 1.
BHE_N is asserted during t1 and remains asserted through t2 and tw . BHE_N does not need to be latched. BHE_N floats during bus hold and reset.
<table>
<tr><th>BHE_N</th><th>AD0</th><th>Type of Bus Cycle</th></tr>
<tr><td>0</td><td>0</td><td>Word Transfer</td></tr>
<tr><td>0</td><td>1</td><td>High Byte Transfer with valid data on Dout(15:8)</td></tr>
<tr><td>1</td><td>0</td><td>Low Byte Transfer with valid data on Dout(7:0)</td></tr>
<tr><td>1</td><td>1</td><td>Reserved</td></tr>
</table>
Table 1. Data Byte Encoding</td></tr>
<tr><td>BWMODE</td><td>Byte/Word Mode (input)</td><td>IMS16C can be configured either 188 mode(byte mode, bwmode = 0) or 186(word mode, bwmode = 1). When high, ad[15:0] is fully used for IMS16C. When low, ad[7:0] is only used for external data access.</td></tr>
</table>

<table>
<tr><td>CLKMD0, CLKMD1</td><td>Core Clock Mode (input)</td><td>CLKMD0 and CLKMD1 pins determine internal core clock speed. These pins are used for setting PLL frequency coefficients.
<table><tr><th>CLKMD[1:0]</th><th>Internal Core Clock</th></tr><tr><td>00</td><td>External clock source bypass mode (1/2)</td></tr><tr><td>01</td><td>81.1Mhz core clock with 3.6864Mhz input.
Input clock range: 1.9MHz < X1 < 5.5MHz
It denotes x22 of Input.</td></tr><tr><td>10</td><td>90.3Mhz core clock with 3.6864Mhz input.
Input clock range: 1.7 < X1 < 5.1 MHz
It denotes x24.5 of Input.</td></tr><tr><td>11</td><td>External clock x6 Mode.
Input clock range: 6.7MHz < X1 < 15MHz
Ex) x1 = 11.05Mhz -> core clock = 66.3Mhz</td></tr></table>Table 2. core clock mode</td></tr>
<tr><td>CLKOUT</td><td>Clock Output (3-state output)</td><td>This pin is directly connected to inverted internal core clock. It supplies the internal core clock to the system. CLKOUT remains active during reset and bus hold conditions.</td></tr>
<tr><td>DEN_N/PIO5</td><td>Data Enable (I/O)</td><td>This pin supplies an output enable to an external data bus transceiver. DEN_N is asserted during memory, I/O, and interrupt acknowledge cycles. DEN_N is deasserted when DTR_N changes state. DEN_N floats during a bus hold or reset condition.</td></tr>
<tr><td>DRQ1 - DRQ0 (DRQ1/PIO13, DRQ0/PIO12)</td><td>DMA Requests (I/O)</td><td>These pins indicate to the microcontroller that an external device is ready for DMA channel 1 or channel 0 to perform a transfer. DRQ1 - DRQ0 are level-triggered and internally synchronized. The DRQ signals are not latched and must remain active until each serviced(source sync or destination sync).</td></tr>
<tr><td>DTR_N/PIO4</td><td>Data Transmit or Receive (I/O)</td><td>This pin indicates which direction data should flow through an external data bus transceiver. When DTR_N is asserted High, the microcontroller transmits data. When this pin is deasserted Low, the microcontroller receives data. DTR_N floats during a bus hold or reset condition.</td></tr>
<tr><td>GND</td><td>Core ground (input)</td><td>These pins connect the system ground to the IMS16C.</td></tr>
<tr><td>HLDA</td><td>Bus Hold Acknowledge (output)</td><td>This pin is asserted High to indicate to an external bus master that the microcontroller has released control of the local bus. When an external bus master requests control of the local bus (by asserting HOLD), the microcontroller completes the bus cycle in progress and then relinquishes control of the bus to the external bus master by asserting HLDA and floating DEN_N, RD_N, WR_N, S2_N - S0_N, AD15 - AD0, S6, A19 - A0, BHE_N and DTR_N, and then driving the chip selects UCS_N, LCS_N, MCS3_N - MCS0_N, PCS6_N - PCS5_N, and PCS3_N - PCS0_N High.
When the external bus master has finished using the local bus, it indicates this to the microcontroller by deasserting HOLD. The microcontroller responds by deasserting HLDA.</td></tr>
<tr><td>HOLD</td><td>Bus Hold Request (input)</td><td>This pin indicates to the microcontroller that an external bus master needs control of the local bus.
The IMS16C microcontroller' HOLD latency time is a function of the activity occurring in the processor when the HOLD request is received.</td></tr>
</table>

INT0	**Maskable Interrupt Request 0 (input)**	This pin indicates to the microcontroller that an interrupt request has occurred. If the INT0 pin is not masked, the microcontroller transfers program execution to the location specified by the INT0 vector in the microcontroller interrupt vector table. Interrupt requests are synchronized internally and can be edge-triggered or level-triggered. To guarantee interrupt recognition, the requesting device must continue asserting INT0 until the request is acknowledged.
INT1	**Maskable Interrupt Request 1 (input)**	This pin indicates to the microcontroller that an interrupt request has occurred. If INT1 is not masked, the microcontroller transfers program execution to the location specified by the INT1 vector in the microcontroller interrupt vector table. Interrupt requests are synchronized internally and can be edge-triggered or level-triggered. To guarantee interrupt recognition, the requesting device must continue asserting INT1 until the request is acknowledged.
INT2/PIO31	**Maskable Interrupt Request 2 (I/O)**	This pin indicates to the microcontroller that an interrupt request has occurred. If the INT2 pin is not masked, the microcontroller transfers program execution to the location specified by the INT2 vector in the microcontroller interrupt vector table. Interrupt requests are synchronized internally and can be edge-triggered or level-triggered. To guarantee interrupt recognition, the requesting device must continue asserting INT2 until the request is acknowledged.
INT3	**Maskable Interrupt Request 3 (input)**	This pin indicates to the microcontroller that an interrupt request has occurred. If the INT3 pin is not masked, the microcontroller then transfers program execution to the location specified by the INT3 vector in the microcontroller interrupt vector table. Interrupt requests are synchronized internally, and can be edge-triggered or level-triggered. To guarantee interrupt recognition, the requesting device must continue asserting INT3 until the request is acknowledged.
INT4/PIO30	**Maskable Interrupt Request 4 (I/O)**	This pin indicates to the microcontroller that an interrupt request has occurred. If the INT4 pin is not masked, the microcontroller then transfers program execution to the location specified by the INT4 vector in the microcontroller interrupt vector table. Interrupt requests are synchronized internally, and can be edge-triggered or level-triggered. To guarantee interrupt recognition, the requesting device must continue asserting INT4 until the request is acknowledged.
LCS_N	**Lower Memory Chip Select (output)**	This pin indicates to the system that a memory access is in progress to the lower memory block. The base address and size of the lower memory block are programmable up to 512 Kbytes. LCS is held High during a bus hold condition.
MCS3_N/PIO25	**Midrange Memory Chip Select 3 (I/O)**	This pin indicates to the system that a memory access is in progress to the fourth region of the midrange memory block. The base address and size of the midrange memory block are programmable. MCS3_N is held High during a bus hold condition.
MCS2_N - MCS0_N (MCS2/PIO24, MCS1/PIO15, MCS0/PIO14)	**Midrange Memory Chip Selects (I/O)**	These pins indicate to the system that a memory access is in progress to the corresponding region of the midrange memory block. The base address and size of the midrange memory block are programmable.

		MCS2_N - MCS0_N are held High during a bus hold condition.
NMI	**Nonmaskable Interrupt (input)**	This pin indicates to the microcontroller that an Not maskable interrupt request has occurred. The NMI is the highest priority hardware interrupt pin and, unlike the INT4 - INT0 pins, cannot be masked. The microcontroller always transfers program execution to the location specified by the nonmaskable interrupt vector in the microcontroller interrupt vector table when NMI is asserted. Although NMI is the highest priority interrupt source, it does not participate in the priority resolution process of the maskable interrupts. There is no bit associated with NMI in the interrupt in-service or interrupt request registers. This means that a new NMI request can interrupt an executing NMI interrupt service routine. As with all hardware interrupts, the IF(interrupt enable flag) is cleared when the processor takes the interrupt, disabling the maskable interrupt sources. However, if maskable interrupts are re-enabled by software in the NMI interrupt service routine, via the STI instruction for example, the fact that an NMI is currently in service will not have any effect on the priority resolution of maskable interrupt requests. For this reason, it is strongly advised that the interrupt service routine for NMI should not enable the maskable interrupts. An NMI transition from Low to High is latched and synchronized internally, and it initiates the interrupt at the next instruction boundary. To guarantee that the interrupt is recognized, the NMI pin must be asserted for at least one CLKOUT period.
PCS3_N - PCS0_N (PCS3_N/PIO19, PCS2_N/PIO18, PCS1_N/PIO17, PCS0_N/PIO16)	**Peripheral Chip Selects (I/O)**	These pins indicate to the system that a peripheral access is in progress to the corresponding region(either I/O or memory address space). The base address of the peripheral memory block is programmable. PCS3_N - PCS0_N are held High during a bus hold condition. They are also held High during reset. Unlike the UCS_N and LCS_N chip selects, the PCS_N outputs assert with the multiplexed AD address bus. Note also that each peripheral chip select asserts over a 256byte address range, which is twice the address range covered by peripheral chip selects in the 80C186 and 80C188 microcontrollers.
PCS5_N/A1/PIO3	**Peripheral Chip Select 5, Latched Address Bit 1 (I/O)**	PCS5_N: These pins indicate to the system that a peripheral access is in progress to the corresponding region(either I/O or memory address space). The base address of the peripheral memory block is programmable. PCS5_N is held High during a bus hold condition. It is also held High during reset. Unlike the UCS_N and LCS_N chip selects, the PCS_N outputs assert with the multiplexed AD address bus. Note also that each peripheral chip select asserts over a 256 byte address range, which is twice the address range covered by peripheral chip selects in the 80C186. A1: When the EX bit in the MCS_N and PCS_N auxiliary register is 0, this pin supplies an internally latched address bit 1 to the system. During a bus hold condition, A1 retains its previously latched value.
PCS6_N/A2/PIO2	**Peripheral Chip Select 6, Latched Address Bit 2 (I/O)**	PCS6_N: These pins indicate to the system that a peripheral access is in progress to the corresponding region(either I/O or memory address space). The base address of the peripheral memory

		block is programmable. PCS6_N is held High during a bus hold condition or reset. Unlike the UCS_N and LCS_N chip selects, the PCS outputs assert with the multiplexed AD address bus. Note also that each peripheral chip select asserts over a 256 byte address range, which is twice the address range covered by peripheral chip selects in the 80C186. A2: When the EX bit in the MCS and PCS Auxiliary Register is 0, this pin supplies an internally latched address bit 2 to the system. During a bus hold condition, A2 retains its previously latched value.
PIO43 - PIO0 (Shared)	**Programmable I/O Pins**	The IMS16C microcontroller provides 44 individually programmable I/O pins. Each PIO can be programmed with the following attributes: PIO function (enabled/disabled), direction (input/output). The pins that are multiplexed with PIO43 - PIO0 are listed in Table 3. After power-on reset, the PIO pins default to various configurations. The column titled Power-On Reset Status in Table 3 lists the defaults for the PIOs. The system initialization code must reconfigure any PIOs as required. The A19 - A17(AIO2 –AIO0) address pins default to normal operation on power-on reset, allowing the processor to correctly begin fetching instructions at the boot address FFFF0h. The DTR_N, DEN_N, and SRDY pins also default to normal operation on power-on reset.

PIO	Associated	Power-On Reset Status
0	TMRIN1	Input with pullup
1	TMROUT1	Input with pullup
2	PCS6_N	Input with pullup
3	PCS5_N	Input with pullup
4	DTR_N	Normal operation
5	DEN_N	Normal operation
6	SRDY	Normal operation
7	AIO0(A17)	Normal operation
8	AIO1(A18)	Normal operation
9	AIO2(A19)	Normal operation
10	TMROUT0	Input with pullup
11	TMRIN0	Input with pullup
12	DRQ0	Input with pullup
13	DRQ1	Input with pullup
14	MCS0_N	Input with pullup
15	MCS1_N	Input with pullup
16	PCS0_N	Input with pullup
17	PCS1_N	Input with pullup
18	PCS2_N	Input with pullup
19	PCS3_N	Input with pullup
20	TXD1	Input with pullup
21	RXD1	Input with pullup
22	TXD2	Input with pullup
23	RXD2	Input with pullup
24	MCS2_N	Input with pullup
25	MCS3_N	Input with pullup
26	UZI_N	Input with pullup
27	TXD0	Input with pullup
28	RXD0	Input with pullup

		28 RXD0 Input with pullup 29 S6 Input with pullup 30 INT4 Input with pullup 31 INT2 Input with pullup 32 RX1 Input with pullup 33 RXEN1 Input with pullup 34 RXCLK1 Input with pullup 35 TX1 Input with pullup 36 TXEN1 Input with pullup 37 TXCLK1 Input with pullup 38 RX0 Input with pullup 39 RXEN0 Input with pullup 40 RXCLK0 Input with pullup 41 TX0 Input with pullup 42 TXEN0 Input with pullup 43 TXCLK0 Input with pullup Table 3. Numeric PIO Pin Assignments
VDDA, VSSA	**Analog vdd, vss (input)**	PLL Core Analog power supply and ground
VDDD, VSSD	**Digital vdd, vss (input)**	PLL Core Digital power supply and ground
RESET	**Reset out (output)**	Reset Output indicates that the CPU is being reset, and can be used as a system reset, It is active high, synchronized with the processor clock, and lasts an integer number of clock periods corresponding to the length of the RES_N signal.
RD_N	**Read Strobe (3-state output)**	This pin indicates to the system that the microcontroller is performing a memory or I/O read cycle. RD_N pin is guaranteed not to be asserted before the address and data bus is floated during the address-to-data transition. RD_N floats during a bus hold condition
RES_N	**Reset (input)**	This pin requires the microcontroller to perform a reset. When RES_N is asserted, the microcontroller immediately terminates its present activity, clears its internal logic, and CPU control is transferred to the reset address FFFF0h. RES_N must be held Low for at least 1 ms. RES_N can be asserted asynchronously to CLKOUT because RES_N is synchronized internally. For proper initialization, VCC must be within specifications, and CLKOUT must be stable for more than ten CLKOUT periods during which RES_N is asserted. This input is provided with Schmitt trigger to facilitate power-on reset generation via an RC network.
RX0/PIO38 **RX1/PIO32**	**HDLC Receive Data (I/O)**	This pin supplies HDLC serial receive data from the system to the internal HDLC controller.
RXEN0/PIO39 **RXEN1/PIO33**	**HDLC Receive Enable (I/O)**	This pin controls HDLC serial receive data from the system to the internal HDLC controller.
RXCLK0/PIO40 **RXCLK1/PIO34**	**HDLC Receive Clock (I/O)**	This pin supplies HDLC serial receive data sync clock to the system or receives serial data sync clock from the system.
RXD0/PIO28 **RXD1/PIO21** **RXD2/PIO23**	**UART Receive Data (I/O)**	This pin supplies UART serial receive data from the system to the internal UART controller. Especially, RXD0, RXD1 can be configured to DMA operation. Otherwise RXD2 is restricted to normal UART operation only.

<table>
<tr><td>S2_N - S0_N</td><td>Bus Cycle Status (3-state output)</td><td>These pins indicate to the system the type of bus cycle in progress. S2_N can be used as a logical memory or I/O indicator, and S1_N can be used as a data transmit or receive indicator. S2_N - S0_N float during bus hold and hold acknowledges conditions.
The S2_N - S0_N pins are encoded as shown in Table 4.
<table><tr><th>S2_N</th><th>S1_N</th><th>S0_N</th><th>Bus Cycle</th></tr><tr><td>0</td><td>0</td><td>0</td><td>Interrupt acknowledge</td></tr><tr><td>0</td><td>0</td><td>1</td><td>Read data from I/O</td></tr><tr><td>0</td><td>1</td><td>0</td><td>Write data to I/O</td></tr><tr><td>0</td><td>1</td><td>1</td><td>Halt</td></tr><tr><td>1</td><td>0</td><td>0</td><td>Instruction fetch</td></tr><tr><td>1</td><td>0</td><td>1</td><td>Read data from memory</td></tr><tr><td>1</td><td>1</td><td>0</td><td>Write data to memory</td></tr><tr><td>1</td><td>1</td><td>1</td><td>No Bus Activity</td></tr></table>Table 4. Bus Cycle Encoding</td></tr>
<tr><td>S6/PIO29</td><td>Bus Cycle Status Bit 6 (I/O)</td><td>During the second and remaining periods of a cycle (t2 and t3), this pin is asserted High to indicate a DMA-initiated bus cycle. During a bus hold or reset condition, S6 floats.
If S6 is to be used as PIO29 in input mode, the device driving PIO29 must not drive the pin Low during power-on reset. S6/PIO29 defaults to a PIO input with pullup, so the pin does not need to be driven High externally.</td></tr>
<tr><td>TEST_SE</td><td>Scan enable (input)</td><td>TEST_SE pin is used for scan test. Active high.</td></tr>
<tr><td>TEST_MODE</td><td>Scan test (input)</td><td>TEST_MODE pin is used for scan test. Active high.</td></tr>
<tr><td>SD_BA</td><td>SDRAM control (output)</td><td>This is the bank selection pin connected to the SDRAM</td></tr>
<tr><td>SD_CS_N</td><td>SDRAM control (output)</td><td>This is the active low SDRAM chip select pin.</td></tr>
<tr><td>SD_CAS_N</td><td>SDRAM control (output)</td><td>This is the active low SDRAM column address select pin.</td></tr>
<tr><td>SD_LAS_N</td><td>SDRAM control (output)</td><td>This is the active low SDRAM low address select pin.</td></tr>
<tr><td>SD_WR_N</td><td>SDRAM control (output)</td><td>This is the active low SDRAM write pin.</td></tr>
<tr><td>SD_DQM1 - SD_DQM0</td><td>SDRAM control (output)</td><td>This is the active high byte mask signal. When high, no data can be read to written to the SDRAM.</td></tr>
<tr><td>SRDY/PIO6</td><td>Synchronous Ready (I/O)</td><td>This pin indicates to the microcontroller that the addressed memory space or I/O device will complete a data transfer. The SRDY pin accepts an active High input synchronized to CLKOUT. Using SRDY instead of ARDY allows relaxed system timing because of the elimination of the one-half clock period required to internally synchronize ARDY. To always assert the ready condition to the microcontroller, tie SRDY High. If the system does not use SRDY, tie the pin Low to yield control to ARDY.</td></tr>
<tr><td>TMRIN0/PIO11</td><td>Timer Input 0 (I/O)</td><td>This pin supplies a clock or control signal to the internal microcontroller timer 0. After internally synchronizing a Low-to-High transition on TMRIN0, the microcontroller increments the timer. TMRIN0 must be tied High if not being used.</td></tr>
<tr><td>TMRIN1/PIO0</td><td>Timer Input 1 (I/O)</td><td>This pin supplies a clock or control signal to the internal microcontroller timer 1. After internally synchronizing a Low-to-High transition on TMRIN1, the microcontroller increments the timer. TMRIN1 must be tied High if not being used.</td></tr>
</table>

TMROUT0/PIO10	**Timer Output 0 (I/O)**	This pin supplies the system with either a single pulse or a continuous waveform with a programmable duty cycle. TMROUT0 is floated during a bus hold or reset.
TMROUT1/PIO1	**Timer Output 1 (I/O)**	This pin supplies the system with either a single pulse or a continuous waveform with a programmable duty cycle. TMROUT1 can also be programmed as a watchdog timer. TMROUT1 is floated during a bus hold or reset.
TXD0/PIO27 **TXD1/PIO20** **TXD2/PIO22**	**Transmit Data (I/O)**	This pin supplies UART serial transmit data to the system from the internal UART controller. Especially, TXD0, TXD1 can be configured to DMA operation. Otherwise TXD2 is restricted to normal UART operation only.
TX0/PIO41 **TX1/PIO35**	**Transmit Data (I/O)**	This pin supplies HDLC serial transmit data from the system to the internal HDLC controller.
TXEN0/PIO42 **TXEN1/PIO36**	**Transmit Enable (I/O)**	This pin controls HDLC serial transmit data from the system to the internal HDLC controller.
TXCLK0/PIO43 **TXCLK1/PIO37**	**Transmit Clock (I/O)**	This pin supplies HDLC serial transmit data sync clock to the system or receives serial data sync clock from the system.
UCS_N	**Upper Memory Chip Select (output)**	This pin indicates to the system that a memory access is in progress to the upper memory block. The base address and size of the upper memory block are programmable up to 512 Kbytes. UCS is held High during a bus hold condition. After power-on reset, UCS is asserted because the processor begins executing at FFFF0h and the default configuration for the UCS chip select is 64 Kbytes from F0000h to FFFFFh.
UZI_N/PIO26	**Upper Zero Indicate (I/O)**	This pin lets the designer determine if an access to the interrupt vector table is in progress by ORing it with bits 15-10 of the address and data bus. UZI is the logical OR of the inverted A19 - A16 bits, and it asserts in the first period of a bus cycle and is held throughout the cycle.
VDD	**Digital Power supply (input)**	Digital Power Supply
WR_N	**Write Strobe (3-state output)**	This pin indicates to the system that the data on the bus is to be written to a memory or I/O device. WR_N floats during a bus hold or reset condition
X1(CLKIN)	**Crystal Input (input)**	This pin and the X2 pin provide connections for a fundamental mode or third-overtone parallel-resonant crystal used by the internal oscillator circuit. To provide the microcontroller with an external clock source, connect the source to the X1 pin and leave the X2 pin unconnected.
X2	**Crystal Output (output)**	This pin and the X1 pin provide connections for a fundamental mode or third-overtone parallel-resonant crystal used by the internal oscillator circuit. To provide the microcontroller with an external clock source, leave the X2 pin unconnected and connect the source to the X1 pin.

3.6. Architecture Overview

The IMS86 core in IMS16C is comprised of two major functions the Execution Unit (EU), and the Bus Interface Unit (BIU). Both units operate independent and asynchronous to the other. The EU is responsible for instruction execution and effective address calculations. The BIU performs all bus cycles and maintains the instruction queue.

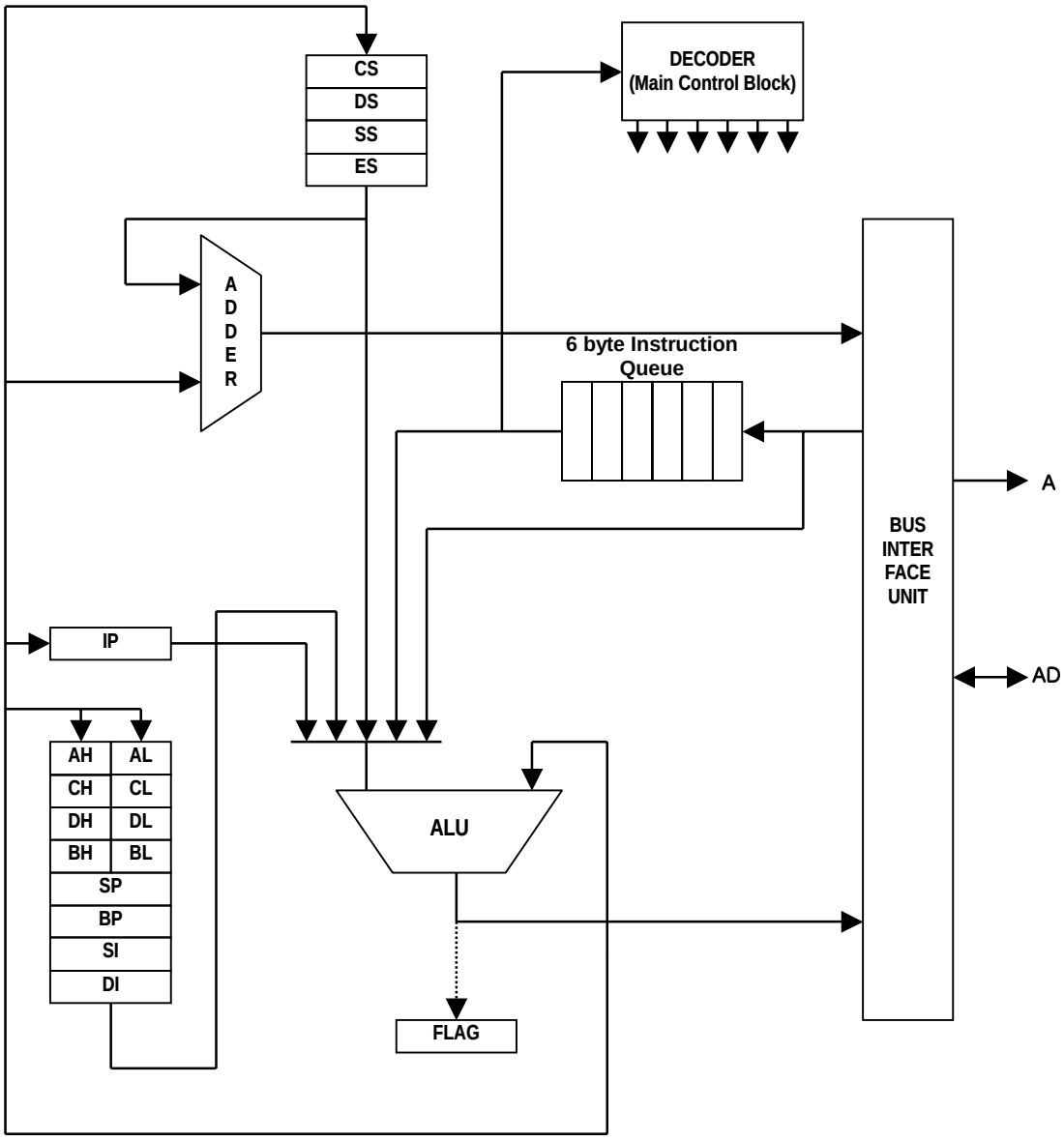


Figure 3.5 IMS86 Core Architecture

3.7. Bus Operation

The industry-standard 80C186 and 80C188 microcontrollers finishes 1 bus cycle with 4 clock cycles(t1, t2, t3, t4). The IMS16C reduces 4 clock cycles to 3 clock cycles(t1, t2, t3) for 1 bus cycle.
The IMS16C provides the multiplexed AD bus and a non-multiplexed address (A) bus. The A bus provides an address to the system for the complete bus cycle (t1 – t3).

Read and write bus cycles can be performed to memory or I/O space. Figure 3.6.1 and Figure 3.6.2 shows the affected signals during a normal read or write operation for an IMS16C microcontroller. The address and data will be multiplexed onto the AD bus.

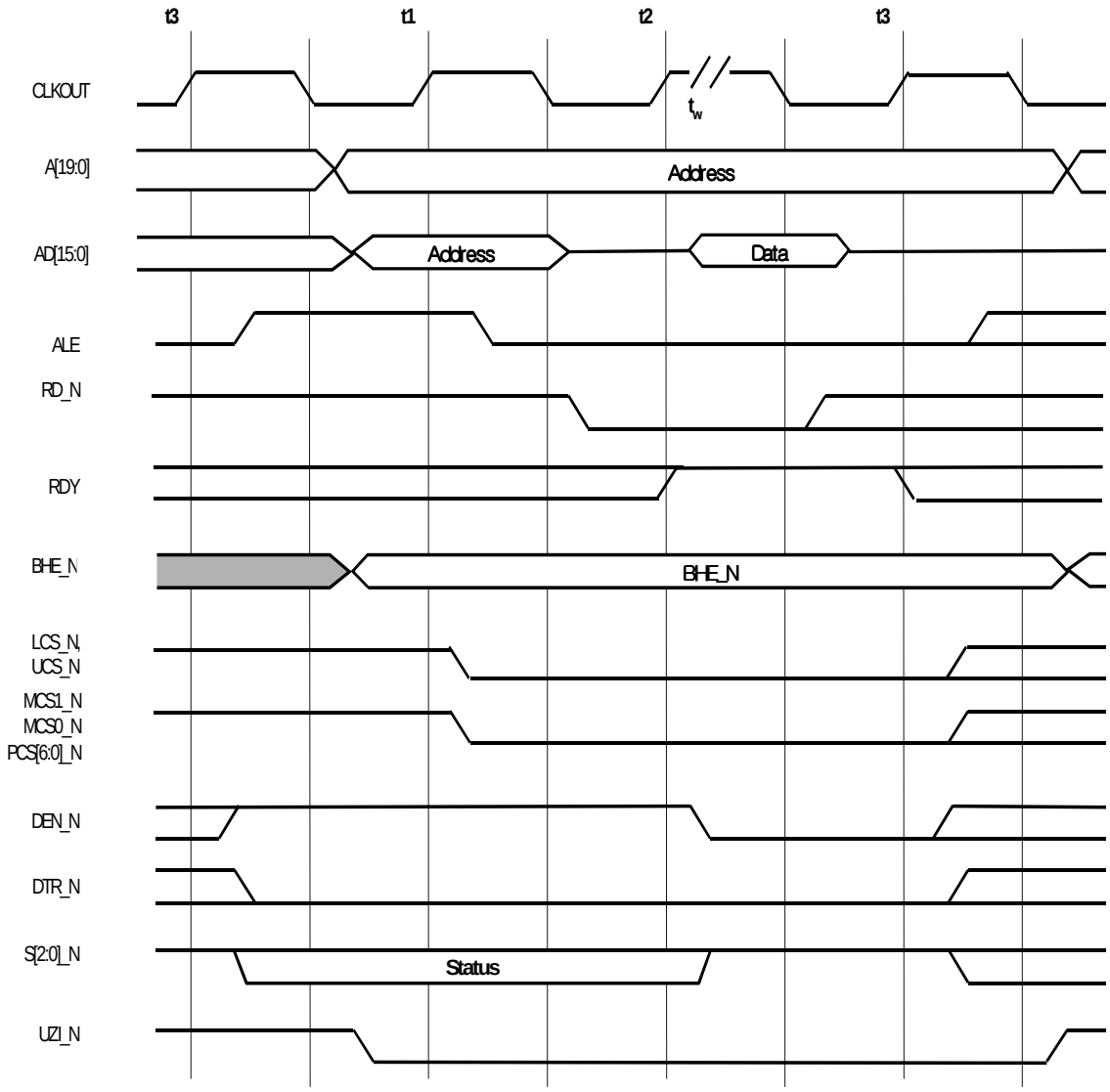


Figure 3.6.1 IMS16C Read Cycle Waveforms

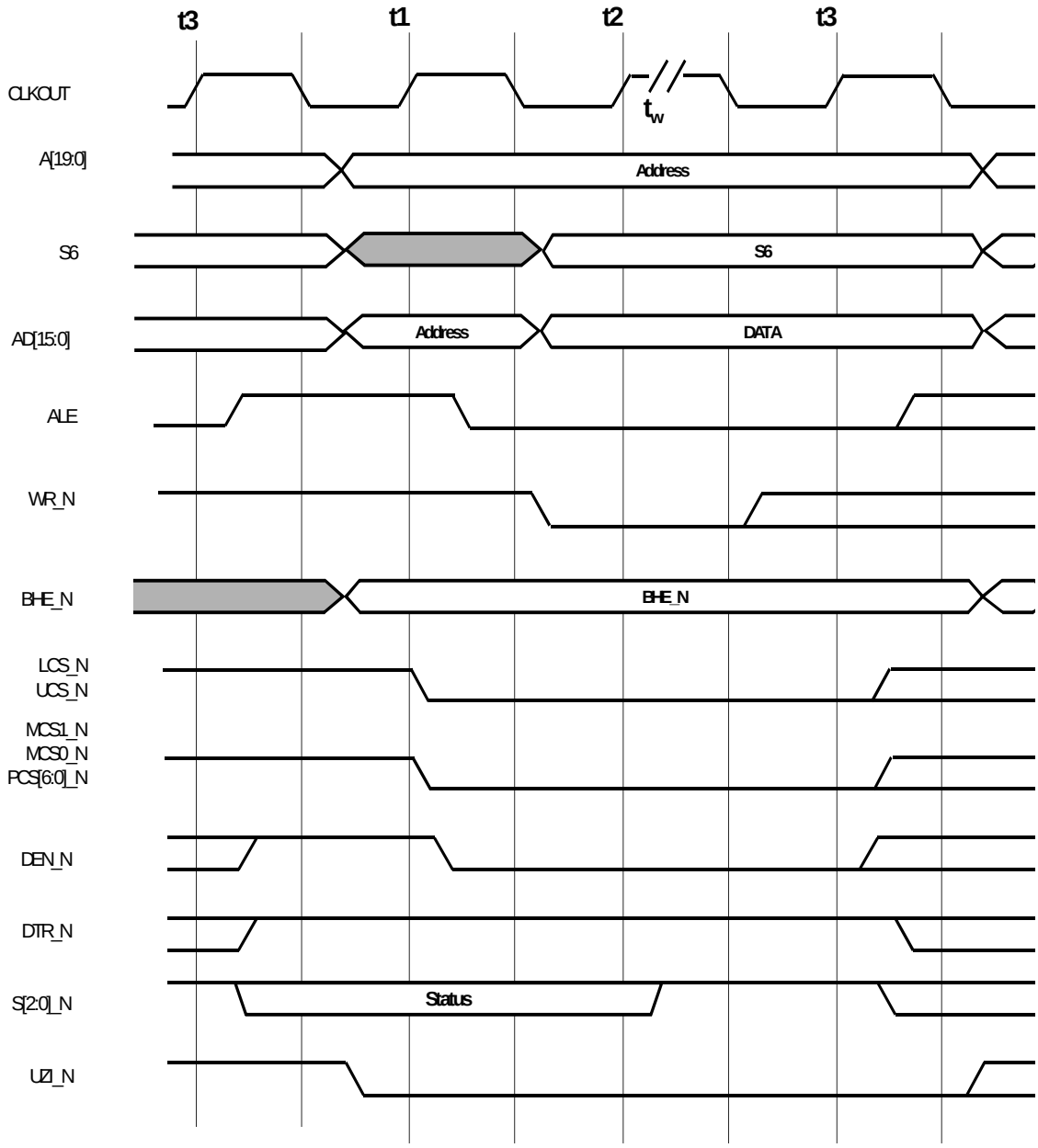


Figure 3.6.2 IMS16C Write Cycle Waveforms

4. Programming

All members of the IMS16 series microprocessors, including the IMS16C, contain the same basic set of registers, instructions, and addressing modes, and are compatible with the original industry-standard 186/188 parts.

4.1. Register Set

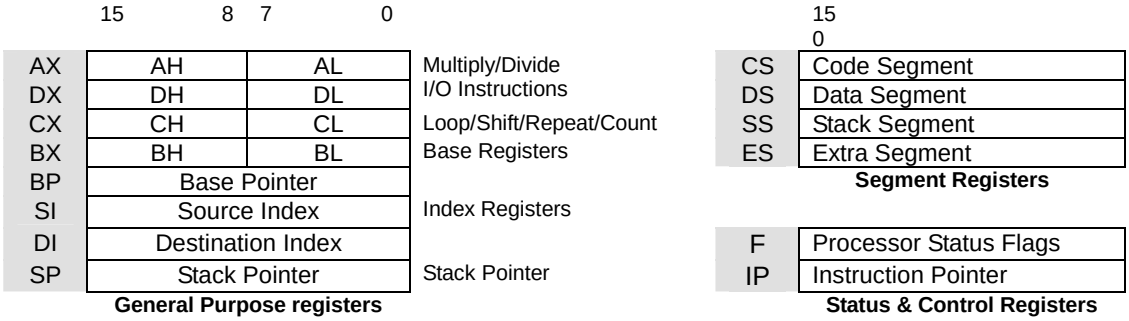


Figure 4.1 Programming Model

The base architecture of the IMS16C has 14 registers, as shown in Figure 4.1. These registers are grouped into the following categories:

General Purpose Registers

Eight 16-bit general purpose registers can be used for arithmetic and logical operands. Four of these (AX, BX, CX, and DX) can be used as 16-bit registers or split into pairs of separate 8-bit registers (AH, AL, BH, BL, CH, CL, DH, and DL). The Destination Index (DI) and Source Index (SI) registers are used for data movement and string instructions. The Base Pointer (BP) and Stack Pointer (SP) registers are used for the stack segment and point to the bottom and top of the stack, respectively.

Some of the general purpose registers also have dedicated functions with certain instructions as noted in Figure 4.1. The AX and DX registers are used during multiply, divide and I/O instructions. The CX register is used during instructions that have a repeat count for loop, shift and repeat. BX and BP can be used as base pointers used during effective address calculations. SI and DI can be used as index registers during effective address calculations.

Four of the general-purpose registers (BP, BX, DI, and SI) can also be used to determine offset addresses of operands in memory. These registers can contain base addresses or indexes to particular locations within a segment. The addressing mode selects the specific registers for operand and address calculations.

All stack operations (POP, POPA, POPF, PUSH, PUSHA, PUSHF) utilize the stack pointer. The Stack Pointer register is always offset from the Stack Segment (SS) register, and no segment override is allowed.

Besides, The IMS16C has 16 word Memory Mapped General Purpose Registers. It locates

I/O memory space ranges from FF00h to FF1Eh. Users can access these registers using IN, OUT instructions.

- **Segment Register**s

Four 16-bit special-purpose registers (CS, DS, ES, and SS) select, at any given time, the segments of memory that are immediately addressable for code (CS), data (DS and ES), and stack (SS) memory.
CS is used for instruction prefetch and immediate data. DS is used for all other data references. SS is used for all stack pushes and pops; any memory reference that uses BP as a base register. ES is used for all string instruction references that use the DI register as an index.

- **Status and Control Register**s

Two 16bit special purpose registers record or alter certain aspects of the processor state. The Instruction Pointer (IP) register contains the offset address of the next sequential instruction to be executed and the Processor Status Flags (FLAGS) register contains status and control flag bits.

4.2. Processor Status Flags Register

The 16-bit processor Status Flags register records specific characteristics of the result of logical and arithmetic instructions and controls the operation of the microcontroller within a given operating mode.
After an instruction is executed, the value of the flags may be set, cleared/reset, unchanged, or undefined. The term undefined means that the flag value prior to the execution of the instruction is not preserved, and the value of the flag after the instruction is executed cannot be predicted.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				OF	DF	IF	TF	SF	ZF		AF		PF		CF

Bit	Name	Function
11	Overflow Flag	Set if the signed result cannot be expressed within the number of bits in the destination operand, cleared otherwise
10	Direction Flag	Used during string instructions to determine auto decrement(DF=1), or auto increment (DF=0) of index value.
9	Interrupt-Enable Flag	When set, enables maskable interrupts to cause the CPU to transfer control to a location specified by an interrupt vector.
8	Trace Flag	When set, a trace interrupt occurs after instructions execute. TF is cleared by the trace interrupt after the processor status flags are pushed onto the stack. The trace service routine can continue tracing by popping the flags back with an interrupt return (IRET) instruction.
7	Sign Flag	Set equal to high-order bit of result (0 if 0 or positive, 1 if negative).
6	Zero Flag	Set if result is 0; cleared otherwise.
4	Auxiliary Carry	Set on carry from or borrow to the low-order 4 bits of the AL general-purpose register; cleared otherwise.
2	Parity Flag	Set if low-order 8 bits of result contain an even number of 1's; cleared otherwise.
0	Carry Flag	Set on high-order bit carry or borrow; cleared otherwise.

Figure 4.2 Status Flags Register

4.3. Memory Organizations and Address Generation

Memory is organized in sets of segments. Each segment is a linear contiguous sequence of 64K (2^{16}) 8-bit bytes. Memory is addressed using a two-component address that consists of a 16-bit segment value and a 16-bit offset. The offset is the number of bytes from the beginning of the segment (the segment address), to the data or instruction that is being accessed.

The processor forms the physical address of the target location by taking the segment address, shifting it to the left 4 bits (multiplying by 16), and adding this to the 16-bit offset. The result is the 20-bit address of the target data or instruction. This allows for a 1-Mbyte physical address size.
Figure 4.3 shows physical address generation example.

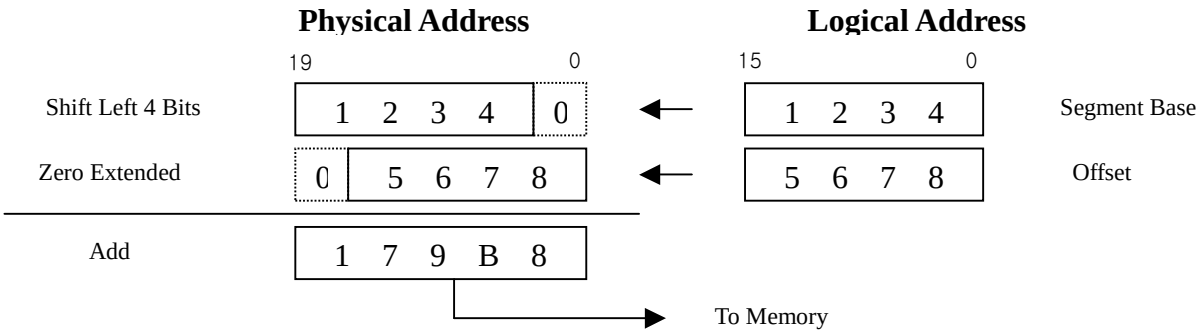


Figure 4.3 Physical Address Generation

4.4. Segments

The IMS16C uses four segment registers:

1. **Data Segment (DS):** The processor assumes that all accesses to the program's variables are from the 64K space pointed to by the DS register. The data segment holds data, operands, etc.
2. **Code Segment (CS):** This 64K space is the default location for all instructions. All code must be executed from the code segment.
3. **Stack Segment (SS):** The processor uses the SS register to perform operations that involve the stack, such as pushes and pops. The stack segment is used for temporary space.
4. **Extra Segment (ES):** Usually this segment is used for large string operations and for large data structures. Certain string instructions assume the extra segment as the segment portion of the address. The extra segment is also used (by using segment override) as a spare data segment.

When a segment is not defined for a data movement instruction, it's assumed to be a data segment. An instruction prefix can be used to override the segment register. For speed and compact instruction encoding, the segment register used for physical address generation is implied by the addressing mode used (see Figure 4.4).

Memory Reference Needed	Segment Register Used	Implicit Segment Selection Rule
Local Data	Data(DS)	All data references
Instructions	Code(CS)	Instructions, Immediate data
Stack	Stack(SS)	All stack pushes and pops Any memory references that use the BP register
External Data(Global)	Extra(ES)	All string instruction references that use the DI register as an index

Figure 4.4 Segment Register Selection Rules

4.5. Addressing Modes

The IMS16C microcontroller uses eight categories of addressing modes to specify operands.
- Two addressing modes are provided for instructions that operate on register or immediate operands.
- Six addressing modes are provided to specify the location of an operand in a memory segment.

Register and Immediate Operands

Register Operand Mode : The operand is located in one of the 8- or 16-bit registers.
Immediate Operand Mode :The operand is included in the instruction.

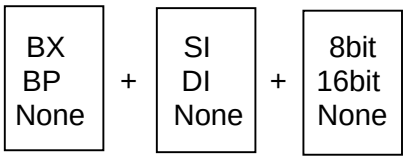
Memory Operands

A memory-operand address consists of two 16-bit components: a segment value and an offset.
The segment value is supplied by a 16 bit segment register either implicitly chosen by the addressing mode or explicitly chosen by a segment override prefix.
The offset, also called the effective address, is calculated by summing any combination of the following three address elements:

1. Displacement : an 8-bit or 16-bit immediate value contained in the instruction
2. Base : contents of either the BX or BP base registers
3. Index : contents of either the SI or DI index registers

Any carry from the 16-bit addition is ignored. Eight-bit displacements are sign-extended to 16-bit values.

Combinations of the above three address elements define the following six memory addressing modes:



1. Direct Mode : The operand offset is contained in the instruction as an 8- or 16-bit displacement element.
2. Register Indirect Mode : The operand offset is in one of the registers BP, BX, DI, or SI.
3. Based Mode : The operand offset is the sum of an 8- or 16-bit displacement and the contents of a base register (BX or BP).
4. Indexed Mode : The operand offset is the sum of an 8- or 16-bit displacement and the contents of an index register (DI or SI).

5. Based Indexed Mode : The operand offset is the sum of the contents of a base register (BP or BX) and an index register (DI or SI).
6. Based Indexed Mode with Displacement : The operand offset is the sum of a base register's contents, an index register's contents, and an 8-bit or 16-bit displacement.

Addressing Mode	Example
Direct	mov ax, ds:4
Register Indirect	mov ax, [si]
Based	mov ax, [bx]4
Indexed	mov ax, [si]4
Based Indexed	mov ax, [si][bx]
Based Indexed with displacement	mov ax, [si][bx]4

Figure 4.5 Memory Addressing Mode Examples

4.6. I/O Space

The I/O space consists of 64K 8-bit or 32K 16-bit ports. The IN and OUT instructions address the I/O space with either an 8-bit port address specified in the instruction, or a 16-bit port address in the DX register. Eight-bit port addresses are zero-extended so that A15–A8 are Low. I/O port addresses 00F8h through 00FFh are reserved.

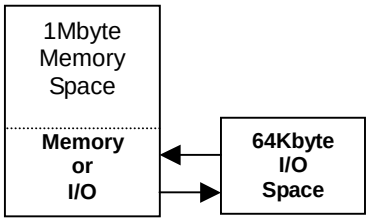


Figure 4.6 Memory and I/O Space

4.7. Data Types

The IMS16C supports the following data types:

Integer
A signed binary numeric value contained in an 8-bit byte or a 16-bit word. All operations assume a two's complement representation.

Ordinal
An unsigned binary numeric value contained in an 8-bit byte or a 16-bit word.

Double Word
A signed binary numeric value contained in two sequential 16-bit addresses, or in a DX::AX register pair.

Quad Word
A signed binary numeric value contained in four sequential 16-bit addresses.

BCD
An unpacked byte representation of the decimal digits 0-9.

ASCII
A byte representation of alphanumeric and control characters using the ASCII standard of character representation.

Packed BCD
A packed byte representation of two decimal digits (0-9). One digit is stored in each nibble (4 bits) of the byte.

String
A contiguous sequence of bytes or words. A string can contain from 1 byte up to 64 Kbytes.

Pointer
16-bit or 32-bit quantities composed of a 16-bit offset component or a 16-bit segment base component plus a 16-bit offset component.

In general, individual data elements must fit within defined segment limits.

4.8. Instruction Set

This chapter contains descriptions of the brief overview of IMS16C Instruction set constitution. The IMS16C shares the standard 186 instruction set. An instruction can reference from zero to several operands. An operand can reside in a register, in the instruction itself, or in memory.

OPCODE w	mod reg r/m	(disp-low)	(disp-high)	(data)	(data)
1 byte or 2bytes		8bit or 16bit displacement if memory operand needs.		8bit or 16bit immediate data	

r/m	mod				
	00	01	10	11	
				w=0	w=1
000	(BX) + (SI)	(BX) + (SI) + D8	(BX) + (SI) + D16	AL	AX
001	(BX) + (DI)	(BX) + (DI) + D8	(BX) + (DI) + D16	CL	CX
010	(BP) + (SI)	(BP) + (SI) + D8	(BP) + (SI) + D16	DL	DX
011	(BP) + (DI)	(BP) + (DI) + D8	(BP) + (DI) + D16	BL	BX
100	(SI)	(SI) + D8	(SI) + D16	AH	SP
101	(DI)	(DI) + D8	(DI) + D16	CH	BP
110	DIRECT	(BP) + D8	(BP) + D16	DH	SI
111	(BX) + [DI]	(BX) + D8	(BX) + D16	BH	DI

reg	w = 0	w = 1
000	AL	AX
001	CL	CX
010	DL	DX
011	BL	BX
100	AH	SP
101	CH	BP
110	DH	SI
111	BH	DI

Segment Register	Code
ES	00
CS	01
SS	10
DS	11

AAA	ASCII adjust for addition	AAD	ASCII adjust for division
AAM	ASCII adjust for multiplication	AAS	ASCII adjust for subtraction
ADC	Add byte or word with carry	ADD	Add byte or word
AND	Logical AND byte or word	BOUND	Detects values outside prescribed range

CALL	Call procedure	CBW	Convert byte to word
CLC	Clear carry flag	CLD	Clear direction flag
CLI	Clear interrupt-enable flag	CMC	Complement carry flag
CMP	Compare byte or word	CMPS	Compare byte or word string
CWD	Convert word to double word	DAA	Decimal adjust for addition
DAS	Decimal adjust for subtraction	DEC	Decrement byte or word by 1
DIV	Divide byte or word unsigned	ENTER	Format stack for procedure entry
ESC	Escape to extension processor	HLT	Halt until interrupt or reset
IDIV	Integer divide byte or word	IMUL	Integer multiply byte or word
IN	Input byte or word	INC	Increment byte or word by 1
INS	Input bytes or word string	INT	Interrupt
INTO	Interrupt if overflow	IRET	Interrupt return
JA/JNBE	Jump if above/not below or equal	JAE/JNB	Jump if above or equal/not below
JB/JNAE	Jump if below/not above or equal	JBE/JNA	Jump if below or equal/not above
JC	Jump if carry	JCXZ	Jump if register CX = 0
JE/JZ	Jump if equal/zero	JG/JNLE	Jump if greater/not less or equal
JGE/JNL	Jump if greater or equal/not less	JL/JNGE	Jump if less/not greater or equal
JLE/JNG	Jump if less or equal/not greater	JMP	Jump
JNC	Jump if not carry	JNE/JNZ	Jump if not equal/not zero
JNO	Jump if not overflow	JNP/JPO	Jump if not parity/parity odd
JNS	Jump if not sign	JO	Jump if overflow
JP/JPE	Jump if parity/parity even	JS	Jump if sign
LAHF	Load AH register from flags	LDS	Load pointer using DS
LEA	Load effective address	LEAVE	Restore stack for procedure exit
LES	Load pointer using ES	LOCK	Lock bus during next instruction
LODS	Load byte or word string	LOOP	Loop
LOOPE/ LOOPZ	Loop if equal/zero	LOOPNE/ LOOPNZ	Loop if not equal/not zero
MOV	Move byte or word	MOVS	Move byte or word string
MUL	Multiply byte or word unsigned	NEG	Negate byte or word
NOP	No operation	NOT	Logical NOT byte or word
OR	Logical Inclusive OR byte or word	OUT	Output byte or word
POP	Pop word off stack	POPA	Pop all general register off stack
POPF	Pop flags off stack	PUSH	Push word onto stack
PUSHA	Push all general registers onto stack	PUSHF	Push flags onto stack
RCL	Rotate left through carry byte or word	RCR	Rotate right through carry byte or word
REP	Repeat	REPE/REPZ	Repeat while equal/zero
REPNE/ REPNZ	Repeat while not equal/not zero	RET0	Return from procedure
ROL	Rotate left byte or word	ROR	Rotate right byte or word
SAHF	Store AH register in flags SF, ZF, AF, PF, and CF	SAL	Shift left arithmetic byte or word
SAR	Shift right arithmetic byte or word	SBB	Subtract byte or word with borrow
SCAS	Scan byte or word string	SHL	Shift left logical byte or word
SHR	Shift right logical byte or word	STC	Set carry flag
STD	Set direction flag	STI	Set interrupt-enable flag
STOS	Store byte or word string	SUB	Subtract byte or word
TEST	Test (Logical AND, flags only set) byte or word	XCHG	Exchange byte or word
XLAT	Translate byte	XOR	Logical exclusive OR byte or word

Figure 4.8 Major Instruction Set of IMS16C

5. Peripherals

This section documents the operation of the IMS16C integrated peripherals. The IMS16C includes the following peripherals:

- Peripheral Interface Logic
- Power-Save Logic
- Chip-Select and Ready Control Logic
- 2 Channel DMA controllers
- 3 Programmable 16-bit Timers
- Interrupt Controller
- 3 UART Controllers
- 2 HDLC Controllers
- SDRAM Controller
- Two 8kbytes Single port Sync RAM

The peripheral control block can be mapped into either memory or I/O space. The base address of the control block must be on an even 256-byte boundary (i.e., the lower eight bits of the base address are 00h). Internal logic recognizes control block addresses and responds to bus cycles. During bus cycles to internal registers, the bus controller signals the operation externally (i.e., the RD, WR, status, address, and data lines are driven as in a normal bus cycle), but the data bus, SRDY, and ARDY are ignored.
At reset, the Peripheral Control Block Relocation register is set to 00FFh, which maps the control block to start at FF00h in I/O space. An offset map of the 256-byte peripheral control register block is shown in Figure 5

Offset (hex)	Register Name
FE	Peripheral Control Block Relocation Register
FC	
FA	
F8	
F6	
F4	Processor Release Level Register
F2	
F0	Power-Save Control Register

EE	SDRAM Mode Set Register
EC	SDRAM Auto Refresh Duty Cycle Register
EA	SDRAM Enable Register
E8	Internal Sync RAM 8kw Enable Register
E6	
E4	
E2	
E0	

DE	
DC	

DA	DMA 1 Control Register
D8	DMA 1 Transfer Count Register
D6	DMA1 Destination Address High Register
D4	DMA 1 Destination Address Low Register
D2	DMA 1 Source Address High Register
D0	DMA 1 Source Address Low Register

CE	
CC	
CA	DMA 0 Control Register
C8	DMA 0 Transfer Count Register
C6	DMA0 Destination Address High Register
C4	DMA 0 Destination Address Low Register
C2	DMA 0 Source Address High Register
C0	DMA 0 Source Address Low Register

BE	HDLC 0 Control Register
BC	HDLC 0 Status Register
BA	HDLC 0 Tx Buffer Register
B8	HDLC 0 Rx Buffer Register
B6	HDLC 0 Buad Register
B4	HDLC 1 Control Register
B2	HDLC 1 Status Register
B0	HDLC 1 Tx Buffer Register

AE	HDLC 1 Rx Buffer Register
AC	HDLC 1 Buad Register
AA	
A8	PCS_N and MCS_N Auxiliary Register
A6	Midrange Memory Chip Select Register
A4	Peripheral Chip Select Register
A2	Lower Memory Chip Select Register
A0	Upper Memory Chip Select Register

9E	UART 2 Baud Rate Divisor Register
9C	UART 2 Receive Register
9A	UART 2 Transmit Register
98	UART 2 Status Register
96	UART 2 Control Register
94	UART 1 Baud Rate Divisor Register
92	UART 1 Receive Register
90	UART 1 Transmit Register

8E	UART 1 Status Register
8C	UART 1 Control Register
8A	
88	UART 0 Baud Rate Divisor Register
86	UART 0 Receive Register
84	UART 0 Transmit Register
82	UART 0 Status Register
80	UART 0 Control Register

7E	
7C	
7A	PIO Data 1 Register
78	PIO Direction 1 Register
76	PIO Mode 1 Register
74	PIO Data 0 Register

72	PIO Direction 0 Register
70	PIO Mode 0 Register

6E	PIO Data 2 Register
6C	PIO Direction 2 Register
6A	PIO Mode 2 Register
68	
66	Timer 2 Mode/Control Register
64	
62	Timer 2 Max count Compare A Register
60	Timer 2 Count Register

5E	Timer 1 Mode/Control Register
5C	Timer 1 Max count Compare B Register
5A	Timer 1 Max count Compare A Register
58	Timer 1 Count Register
56	Timer 0 Mode/Count Register
54	Timer 0 Max count Compare B Register
52	Timer 0 Max count Compare A Register
50	Timer 0 Count Register

4E	
4C	HDLC 1 Interrupt Control Register
4A	HDLC 0 Interrupt Control Register
48	UART 2 Serial Interrupt Control Register
46	UART 1 Serial Interrupt Control Register
44	UART 0 Serial Interrupt Control Register
42	Watchdog Timer Control Register
40	INT4 Control Register

3E	INT3 Control Register
3C	INT2 Control Register
3A	INT1 Control Register
38	INT0 Control Register
36	DMA 1 Interrupt Control Register
34	DMA 0 Interrupt Control Register
32	Timer Interrupt Control Register
30	Interrupt Status Register

2E	Interrupt Request Register
2C	In-service Register
2A	Priority Mask Register
28	Interrupt Mask Register
26	Poll Status Register
24	Poll Register
22	End-of Interrupt Register
20	

1E	General Purpose Register 15
1C	General Purpose Register 14
1A	General Purpose Register 13
18	General Purpose Register 12
16	General Purpose Register 11
14	General Purpose Register 10
12	General Purpose Register 9
10	General Purpose Register 8

0E	General Purpose Register 7

0C	General Purpose Register 6
0A	General Purpose Register 5
08	General Purpose Register 4
06	General Purpose Register 3
04	General Purpose Register 2
02	General Purpose Register 1
00	General Purpose Register 0

Figure 5 IMS16C Peripheral Control Block Register Map

6. Peripheral Control Register

6.1. Peripheral Control Block Relocation Register (RELREG, Offset FEh)

The peripheral control block is mapped into either memory or I/O space by programming the Peripheral Control Block Relocation (RELREG) register. This register is a 16bit register at offset FEh from the control block base address. The RELREG register provides the upper 12 bits of the base address of the control block. The control block is effectively an internal chip select range. Other chip selects can overlap the control block only if they are programmed to zero wait states and ignore external ready.
If the control register block is mapped into I/O space, the upper four bits of the base address must be programmed as 0000b (since I/O addresses are only 16 bits wide). At reset, the RELREG register is set to 00FFh, which maps the control block to start at FF00h in I/O space.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	M/IO	Relocation Address RA(19:8)											

Reset value : 0000_0000_1111_1111b

Bit	Name	Function
12	Memory/IO Space	When set to 1, the peripheral control block(PCB) is located in memory space. When set to 0, the PCB is located in I/O space.
11-0	Relocation Address Bits	RA19–RA8 define the upper address bits of the PCB base address. The lower eight bits (RA7–RA0) default to 00h. RA19–RA16 are ignored when the PCB is mapped to I/O space.

Figure 6.1 Peripheral Control Block Relocation Register

6.2. Power-Save Control Register (PDCON, Offset F0h)

The value of the PDCON register is 0000h at reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSEN	0	0	0	0	0	0	CDD	0	0	0	0	0	F2	F1	F0

Reset value : 0000_0000_0000_0000b

Bit	Name	Function
15	Enable Power-Save mode	When set to 1, enables Power-Save mode and divides the internal operating clock by the value in F2–F0. PSEN is automatically cleared when an external interrupt, including those generated by on-chip peripheral devices, occurs. The value of the PSEN bit is not restored by the execution of an IRET instruction. Software interrupts (INT instruction) and exceptions do not clear the PSEN bit, and interrupt service routines for these conditions should do so if desired. This bit is 0 after processor reset.
8	Clockout Drive Disable	When set to 1, CDD tri-states the clockout. When set to 0, clockout is normal function. This bit is 0 after processor reset.
2-0	Clock Divisor Select	Controls the division factor when Power-Save mode is enabled. Allowable values are as follows:

F2	F1	F0	Divider Factor
0	0	0	Divide by 1
0	0	1	Divide by 2
0	1	0	Divide by 4
0	1	1	Divide by 8
1	0	0	Divide by 16
1	0	1	Divide by 32
1	1	0	Divide by 64
1	1	1	Divide by 128

Figure 6.2 Power-Save Control Register

6.3. Initializations and Processor Reset

Processor initialization or startup is accomplished by driving the RES_N input pin Low. RES_N must be Low during power-up to ensure proper device initialization. RES_N forces the IMS16C microcontroller to terminate all execution and local bus activity. No instruction or bus activity occurs as long as RES_N is active. After RES_N is deasserted and an internal processing interval elapses, the microcontroller begins execution with the instruction at physical location FFFF0h.

7. Chip Select Unit

The IMS16C microcontroller contains logic that provides programmable chip select generation for both memories and peripherals. In addition, the logic can be programmed to provide ready or wait-state generation and latched address bits A1 and A2. The chip select lines are active for all memory and I/O cycles in their programmed area, whether they are generated by the CPU or by the integrated DMA unit. The IMS16C microcontroller provides six chip select outputs for use with memory devices and six more for use with peripherals in either memory space or I/O space. The six memory chip selects can be used to address three memory ranges. ***Note that only one chip select may be programmed to be active for any memory location at a time. (Chip selects cannot be overlapped.)**

Offset	Register Mnemonic	Register Name	Affected Pins	Comments
A0h	UMCS	Upper Memory Chip Select	UCS_N	Ending address is fixed at FFFFFh
A2h	LMCS	Lower Memory Chip Select	LCS_N	Starting address is fixed at 00000h
A4h	PACS	Peripheral Chip Select	PCS6_N - PCS5_N PCS3_N - PCS0_N	Block size is fixed at 256 bytes
A6h	MMCS	Midrange Chip Select	MCS3_N - MCS0_N	Starting address and block size are programmable
A8h	MPCS	PCS and MCS Auxiliary	PCS6_N - PCS5_N PCS3_N - PCS0_N MCS3_N - MCS0_N	Affects both PCS and MCS chip selects

Figure 7 Chip Select Register Summaries

Except for the UCS chip select, chip selects are not activated until the associated registers have been accessed. (An access is any read or write operation.) For this reason, the chip select registers should not be read by the processor initialization code until after they have been written with valid data. The LCS chip select is activated when the LMCS register is accessed, the MCS chip selects are activated after both the MMCS and MPCS registers have been accessed, and the PCS chip selects are activated after both the PACS and MPCS registers have been accessed.

7.1. Upper Memory Chip Select Register (UMCS, Offset A0h)

The IMS16C provides the UCS_N chip select pin for the top of memory. On reset, the microcontroller begins fetching and executing instructions starting at memory location FFFF0h, so upper memory is usually used as instruction memory. To facilitate this usage, UCS_N defaults to active on reset with a default memory range of 64Kbytes from F0000h to FFFFFh, external ready required, and three wait states automatically inserted.
The UCS_N memory range always ends at FFFFFh. The lower boundary is programmable. The Upper Memory Chip Select is configured through the UMCS register.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	LB2	LB1	LB0	0	0	0	0	0	0	R5	0	R3	R2	R1	R0

Reset value : 1111_0000_0010_1111b

Bit	Name	Function
14 -12	Lower Boundary	The LB2-LB0 bits define the lower bound of the memory accessed through the UCS_N chip selects.

Memory Block Size	Starting Address	LB[2:0]	Comments
64K	F0000h	111b	Default
128K	E0000h	110b	
256K	C0000h	100b	
512K	80000h	000b	

5	Ready Mode	The R5 bit is used to configure the ready mode for the UCS_N chip select. If R5 is set to 0, external ready is required. If R5 is set to 1, external ready is ignored. In each case, the processor also uses the value of the R3–R0 bits to determine the number of wait states to insert. R5 defaults to 1 at reset.
3-0	Wait-State Value	The value of R3-R0 determines the number of wait states inserted into an access to the UCS_N memory area. From 0 to 15 wait states can be inserted (R3–R0 = 0000b to 1111b). R3-R0 defaults to 1111b at reset.

Figure 7.1 Upper Memory Chip Select Register

7.2. Low Memory Chip Select Register (LMCS, Offset A2h)

The IMS16C microcontroller provides the LCS_N chip select pin for the bottom of memory. Since the interrupt vector table is located at 00000h at the bottom of memory, the LCS_N pin has been provided to facilitate this usage. The LCS_N pin is not active on reset, but any read or write access to the LMCS register activates this pin.
The Low Memory Chip Select is configured through the LMCS register.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	UB2	UB1	UB0	1	1	1	1	1	1	R5	0	R3	R2	R1	R0

Reset value : 0000_1111_1110_1111b

Bit	Name	Function
14 -12	Upper Boundary	The UB2-UB0 bits define the upper bound of the memory accessed through the LCS_N chip selects.
5	Ready Mode	The R5 bit is used to configure the ready mode for the LCS_N chip select. If R5 is set to 0, external ready is required. If R5 is set to 1, external ready is ignored. In each case, the processor also uses the value of the R3–R0 bits to determine the number of wait states to insert. R5 defaults to 1 at reset.
3-0	Wait-State Value	The value of R3-R0 determines the number of wait states inserted into an access to the LCS_N memory area. From 0 to 15 wait states can be inserted (R3–R0 = 0000b to 1111b). R3-R0 defaults to 1111b at reset.

Memory Block Size	Ending Address	UB[2:0]	Comments
64K	0FFFFh	000b	Default
128K	1FFFFh	001b	
256K	3FFFFh	011b	
512K	7FFFFh	111b	

Figure 7.2 Lower Memory Chip Select Register

7.3. Midrange Memory Chip Select Register (MMCS, Offset A6h)

The IMS16C provides four chip select pins, MCS3_N - MCS0_N, for use within a user-locatable memory block. The base address of the memory block can be located anywhere within the 1Mbyte memory address space, exclusive of the area associated with the UCS_N and LCS_N chip selects (and, if they are mapped to memory, the address range of the Peripheral Chip Selects, PCS6_N - PCS5_N and PCS3_N - PCS0_N). The MCS address range can overlap the PCS address range if the PCS chip selects are mapped to I/O space.

The Midrange Memory Chip Selects are programmed through two registers. The Midrange Memory Chip Select (MMCS) register determines the base address and the ready condition and wait states of the memory block accessed through the MCS pins. The PCS and MCS Auxiliary (MPCS) register is used to configure the block size. The MCS3_N - MCS0_N pins are not active on reset. Both the MMCS and MPCS registers must be accessed with a read or write to activate these chip selects.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA19	BA18	BA17	BA16	BA15	BA14	BA13	1	1	1	R5	0	R3	R2	R1	R0

Reset value : 0000_0001_1110_1111b

Bit	Name	Function
15 -9	Base Address	The base address of the memory block that is addressed by the MCS chip select pins is determined by the value of BA19 - BA13. These bits correspond to bits A19 - A13 of the 20-bit memory address. Bits A12 - A0 of the base address are always 0.
5	Ready Mode	The R5 bit is used to configure the ready mode for the MCS_N chip select. If R5 is set to 0, external ready is required. If R5 is set to 1, external ready is ignored. In each case, the processor also uses the value of the R3 - R0 bits to determine the number of wait states to insert. R5 defaults to 1 at reset.
3-0	Wait-State Value	The value of R3 - R0 determines the number of wait states inserted into an access to the MCS_N memory area. From 0 to 15 wait states can be inserted (R3 - R0 = 0000b to 1111b). R3- R0 defaults to 1111b at reset.

Figure 7.3 Midrange Memory Chip Select Register

The base address can be set to any integer multiple of the size of the memory block size selected in the MPCS register. For example, if the midrange block is 32 Kbytes, the block could be located at 10000h or 18000h but not at 14000h. The base address of the midrange chip selects can be set to 00000h only if the LCS chip select is not active. This is due to the fact that the LCS base address is defined to be address 00000h and chip select address ranges are not allowed to overlap. Because of the additional restriction that the base address must be a multiple of the block size, a 512K MMCS block size can only be used when located at address 00000h, and the LCS chip selects must not be active in this case. Use of the MCS chip selects to access low memory allows the timing of these accesses to follow the AD address bus rather than the A address bus. Locating a 512K MMCS block at 80000h always conflicts with the range of the UCS chip select and is not allowed.

7.4. PCS and MCS Auxiliary Register (MPCS, Offset A8h)

The PCS and MCS Auxiliary (MPCS) register differs from the other chip select control registers in that it contains fields that pertain to more than one type of chip select. The MPCS register fields provide program information for MCS3_N - MCS0_N as well as PCS6_N - PCS5_N and PCS3_N - PCS0_N. In addition to its function as a chip select control register, the MPCS register contains a field that configures the PCS6_N - PCS5_N pins as either chip selects or as alternate sources for the A2 and A1 address bits. When programmed to provide address bits A1 and A2, PCS6_N - PCS5_N cannot be used as peripheral chip selects. These outputs can be used to provide latched address bits for A2 and A1. On reset, PCS6_N - PCS5_N are not active. If PCS6_N - PCS5_N are configured as address pins, an access to the MPCS register causes the pins to activate. No corresponding access to the PACS register is required to activate the PCS6_N - PCS5_N pins as addresses.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	M6	M5	M4	M3	M2	M1	M0	EX	MS	R5	R4	R3	R2	R1	R0

Reset value : 1000_0001_0011_1111b

Bit	Name	Function
14 -8	MCS_N Block Size	This field determines the total block size for the MCS3 - MCS0 chip selects. Each individual chip select is active for one quarter of the total block size. Only one of the M6 - M0 bits can be set at any time. If more than one of the M6 - M0 bits is set, unpredictable operation of the MCS lines occurs.
7	Pin Selector	This bit determines whether the PCS6_N - PCS5_N pins are configured as chip selects or as alternate outputs for A2 - A1. When this bit is set to 1, PCS6_N - PCS5_N are configured as peripheral chip select pins. When EX is set to 0, PCS5_N becomes address bit A1 and PCS6_N becomes address bit A2.
6	Memory / IO Space Selector	This bit determines whether the PCS pins are active during memory bus cycles or I/O bus cycles. When MS is set to 1, the PCS outputs are active for memory bus cycles. When MS is set to 0, the PCS outputs are active for I/O bus cycles.
5	Ready Mode	This bit applies only to the PCS6–PCS5 chip selects. If R5 is set to 0, external ready is required. If R5 is set to 1, external ready is ignored. In each case, the processor also uses the value of the R4–R0 bits to determine the number of wait states to insert. R5 defaults to 1 at reset.
4-0	Wait-State Value	These bits apply only to the PCS6–PCS5 chip selects. The value of R4–R0 determines the number of wait states inserted into an access to the PCS memory or I/O area. From 0 to 31 wait states can be inserted (R4–R0 = 00000b to 11111b). R4-R0 defaults to 11111b at reset.

Total Block Size	Individual Select Size	M6 - M0	Comments
8K	2K	0000001b	Default
16K	4K	0000010b	
32K	8K	0000100b	
64K	16K	0001000b	
128K	32K	0010000b	
256K	64K	0100000b	
512K	128K	1000000b	

Figure 7.4 PCS and MCS Auxiliary Register

7.5. Peripheral Chip Select Register (PACS, Offset A4h)

Each peripheral chip select asserts over a 256-byte address range. The IMS16C provides six chip selects, PCS6_N - PCS5_N and PCS3_N - PCS0_N, for use within a user-locatable memory or I/O block. (PCS4_N is not implemented on the IMS16C.) The base address of the memory block can be located anywhere within the 1Mbyte memory address space, exclusive of the area associated with the UCS_N, LCS_N, and MCS chip selects, or they can be configured to access the 64Kbyte I/O space.

The Peripheral Chip Selects are programmed through two registers: the Peripheral Chip Select (PACS) register and the PCS and MCS Auxiliary (MPCS) register. The Peripheral Chip Select (PACS) register determines the base address, the ready condition, and the wait states for the PCS3_N - PCS0_N outputs. The PCS and MCS Auxiliary (MPCS) register contains bits that configure the PCS6_N -PCS5_N pins as either chip selects or address pins A1 and A2. When the PCS6_N - PCS5_N pins are chip selects, the MPCS register also determines whether PCS chip selects are active during memory or I/O bus cycles and specifies the ready and wait states for the PCS6_N - PCS5_N outputs.

The PCS pins are not active on reset. Both the PACS and MPCS registers must be accessed with a read or write to activate the PCS pins as chip selects. PCS6_N - PCS5_N can be configured and activated as address pins by writing only the MPCS register. No corresponding access to the PACS register is required in this case. PCS3_N - PCS0_N can be configured for 0 wait states to 31 wait states. PCS6_N - PCS5_N can be configured for 0 wait states to 31 wait states.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA19	BA18	BA17	BA16	BA15	BA14	BA13	BA12	BA11	1	R5	R4	R3	R2	R1	R0

Reset value : 0000_0000_0111_1111b

Bit	Name	Function
15 -7	Base Address	The base address of the peripheral chip select block is defined by BA19 - BA11 of the PACS register. BA19 - BA11 correspond to bits 19 - 11 of the 20-bit programmable base address of the peripheral chip select block. Bit 6 of the PACS register corresponds to BA10. When the PCS chip selects are mapped to I/O space, BA19 - 16 must be programmed to 0000b because the I/O address bus is only 16-bits wide.
5	Ready Mode	The R5 bit is used to configure the ready mode for the PCS3_N - PCS0_N chip selects. If R5 is set to 0, external ready is required. External ready is ignored when R5 is set to 1. In each case, the processor also uses the value of the R4 - R0 bits to determine the number of wait states to insert. The ready mode for PCS6_N - PCS5_N is configured through the MPCS register. R5 defaults to 1 at reset.

PCS_N Pins	Range	
	Low	High
PCS0_N	Base Address	Base Address + 255(FFh)
PCS1_N	Base Address + 256	Base Address + 511(1FFh)
PCS2_N	Base Address + 512	Base Address + 767(2FFh)
PCS3_N	Base Address + 768	Base Address + 1023(3FFh)
Reserved	N/A	N/A
PCS5_N	Base Address + 1280	Base Address + 1535(5FFh)
PCS6_N	Base Address + 1536	Base Address + 1791(6FFh)

4-0	Wait-State Value	The value of R4 - R0 determines the number of wait states inserted into a PCS3_N - PCS0_N access. From 0 to 31 wait states can be inserted (R4–R0 = 00000b to 11111b). R4-R0 defaults to 11111b at reset.

Figure 7.5 Peripheral Chip Select Register

8. Interrupt Control Unit

The IMS16C microcontroller can receive interrupt requests from a variety of sources, both internal and external. The internal interrupt controller arranges these requests by priority and presents them one at a time to the CPU. There are six external interrupt sources on the IMS16C microcontroller:
five maskable interrupt pins (INT4 - INT0) and the non-maskable interrupt (NMI) pin. There are ten internal interrupt sources that are not connected to external pins: three timers, two DMA channels, three UART serial ports and two HDLC channels. Interrupt Controller has the ability to resolve priority among simultaneous interrupt requests. **Note that IMS16C only supports Master Mode, and Fully Nested Mode.**

8.1. Definitions of Interrupt Terms

■ Interrupt Vector Table

The interrupt vector table is a memory area of 1Kbyte beginning at address 00000h that holds up to 256 four-byte address pointers containing the address for the interrupt service routine for each possible interrupt type. For each interrupt, an 8-bit interrupt type identifies the appropriate interrupt vector table entry. Interrupts 00h to 1Fh are reserved. The processor calculates the index to the interrupt vector table by shifting the interrupt type left 2 bits (multiplying by 4).

■ Maskable and Non-Maskable Interrupts

Interrupt types 08h through 1Fh are maskable. Of these, only 08h through 14h are actually in use.
The maskable interrupts are enabled and disabled by the interrupt enable flag (IF) in the processor status flags, but the INT command can execute any interrupt regardless of the setting of IF. Interrupt types 00h through 07h and all software interrupts (the INT instruction) are non-maskable. The non-maskable interrupts are not affected by the setting of the IF flag. The IMS16C provides two methods for masking and unmasking the maskable interrupt sources. Each interrupt source has an interrupt control register that contains a mask bit specific to that interrupt. In addition, the Interrupt Mask register is provided as a single source to access all of the mask bits. If the Interrupt Mask register is written while interrupts are enabled, it is possible that an interrupt could occur while the register is in an undefined state. This can cause interrupts to be accepted even though they were masked both before and after the write to the Interrupt Mask register. Therefore, the Interrupt Mask register should only be written when interrupts are disabled. Mask bits in the individual interrupt control registers can be written while interrupts are enabled, and there will be no erroneous interrupt operation.

■ Interrupt Enable Flag (IF)

The interrupt enable flag (IF) is part of the processor status flags. If Interrupt Enable Flag is set to 1, maskable interrupts are enabled and can cause processor interrupt. (Individual maskable interrupts can still be disabled by means of the mask bit in each control register.) If Interrupt Enable Flag is set to 0, all maskable interrupts are disabled. The IF flag does not affect the NMI or software exception interrupts (interrupt types 00h to 07h), and it does not

affect the execution of any interrupt through the INT instruction.

Interrupt Mask Bit

Each of the interrupt control registers for the maskable interrupts contains a mask bit (MSK). If MSK is set to 1 for a particular interrupt, that interrupt is disabled regardless of the IF setting.

Interrupt Priority

The column titled Overall Priority in Figure8.1 shows the fundamental priority breakdown for the interrupts at power-on reset. The non-maskable interrupts 00h through 07h are always prioritized ahead of the maskable interrupts. The maskable interrupts can be reprioritized by reconfiguring the PR2 - PR0 bits in the interrupt control registers. The PR2 - PR0 bits in all the maskable interrupts are set to priority level 7(lowest level) at power-on reset.

Software Interrupts

Software interrupts can be initiated by the INT instruction. Any of the 256 possible interrupts can be initiated by the INT instruction. INT 21h causes an interrupt to the vector located at 00084h in the interrupt vector table. INT FFh causes an interrupt to the vector located at 003FCh in the interrupt vector table. Software interrupts are not maskable and are not affected by the setting of the IF flag.

Software Exceptions

A software exception interrupt occurs when an instruction causes an interrupt due to some condition in the processor. Interrupt types 00h, 01h, 03h, 04h, 05h, 06h, and 07h are software exception interrupts. Software exceptions are not maskable and are not affected by the setting of the IF flag.

Interrupt Name	Interrupt Type	Vector Table Address	Default Priority	Related Instructions	Application Notes
Divide Error Exception	00h	00h	1	DIV, IDIV	1
Trace Interrupt	01h	04h	1A	All	2
Non-Maskable Interrupt(NMI)	02h	08h	1	All	
Breakpoint Interrupt	03h	0Ch	1	INT	1
INT0 Detected Overflow Exception	04h	10h	1	INT0	1
Array Bounds Exception	05h	14h	1	BOUND	1
Unused Opcode Exception	06h	18h	1	Undefined Opcodes	1
ESC Opcode Exception	07h	1Ch	1	ESC	1
Timer 0 Interrupt	08h	20h	2A		3
Timer 1 Interrupt	12h	48h	2B		3
Timer 2 Interrupt	13h	4Ch	2C		3
DMA 0 Interrupt	0Ah	28h	4		
DMA 1 Interrupt	0Bh	2Ch	5		
INT0 Interrupt	0Ch	30h	6		
INT1 Interrupt	0Dh	34h	7		
INT2 Interrupt	0Eh	38h	8		
INT3 Interrupt	0Fh	3Ch	9		
INT4 Interrupt	10h	40h	10		
Watchdog Timer Interrupt	11h	44h	11		
UART0 Serial Port Interrupt	14h	50h	12		
UART1 Serial Port Interrupt	15h	54h	13		
UART2 Serial Port Interrupt	16h	58h	14		
HDLC0 Interrupt	17h	5Ch	15		

HDLC1 Interrupt	18h	60h	16		

Figure 8.1 IMS16C Interrupt Types

NOTES:
Default priorities for interrupt sources are used only if the user does not program each source to a unique priority level.
1. Generated as a result of an instruction execution.
2. Performed in the same manner as the 80C186.
3. All three timers make up a single interrupt request from the interrupt controller. Because of this they share the same priority level. However, each timers has a defined priority with respect to each other. Priority level 2A is the highest, then 2B, followed by 2C.

8.2. Interrupt Conditions and Sequence

■ Non-Maskable Interrupts

Non-maskable interrupts—the trace interrupt, the NMI interrupt, and software interrupts [both user-defined (INT) and software exceptions]—are serviced regardless of the setting of the interrupt enable flag (IF) in the processor status flags.

■ Maskable Hardware Interrupts

In order for maskable hardware interrupt requests to be serviced, the IF flag must be set by the STI instruction, and the mask bit associated with each interrupt must be reset.

■ The Interrupt Request

When an interrupt is requested, the internal interrupt controller verifies that the interrupt is enabled and that there are no higher priority interrupt requests being serviced or pending. If the interrupt request is granted, the interrupt controller uses the interrupt type to access a vector from the interrupt vector table. Each interrupt type has a four-byte vector available in the interrupt vector table. The interrupt vector table is located in the 1024 bytes from 00000h to 003FFh. Each four-byte vector consists of a 16-bit offset (IP) value and a 16-bit segment (CS) value. The 8-bit interrupt type is shifted left 2 bit positions (multiplied by 4) to generate the index into the interrupt vector table.

■ Interrupt Servicing

A valid interrupt transfers execution to a new program location based on the vector in the interrupt vector table. The next instruction address (CS:IP) and the processor status flags are pushed onto the stack. The interrupt enable flag (IF) is cleared after the processor status flags are pushed on the stack, disabling maskable interrupts during the interrupt service routine (ISR). The segment:offset values from the interrupt vector table are loaded into the code segment(CS) and the instruction pointer (IP), and execution of the ISR begins.

■ Returning from the Interrupt

The interrupt return (IRET) instruction pops the processor status flags and the return address off the stack. Program execution resumes at the point where the interrupt occurred. The interrupt enable flag (IF) is restored by the IRET instruction along with the rest of the processor status flags. If the IF flag was set before the interrupt was serviced, interrupts are re-enabled when the IRET is executed. If there are valid interrupts pending when the IRET is executed, the instruction at the return address is not executed. Instead, the new interrupt is serviced immediately. If an ISR intends to permanently modify the value of any of the saved flags, it must modify the copy of the Processor Status Flags register that was pushed onto the stack.

8.3. Interrupt Priority

Figure 8.1 shows the predefined types and overall priority structure for the IMS16C. Non-maskable interrupts (interrupt types 0-7) are always higher priority than maskable interrupts. Maskable interrupts have a programmable priority that can override the default priorities relative to one another.

Non-Maskable Interrupts and Software Interrupt Priority

The non-maskable interrupts from 00h to 07h and software interrupts (INT instruction) always take priority over the maskable hardware interrupts. Within the non-maskable and software interrupts, the trace interrupt has the highest priority, followed by the NMI interrupt, followed by the remaining non-maskable and software interrupts. After the trace interrupt and the NMI interrupt, the remaining software exceptions are mutually exclusive and can only occur one at a time, so there is no further priority breakdown.

Maskable Hardware Interrupt Priority

Beginning with interrupt type 8 (the Timer 0 interrupt), the maskable hardware interrupts have both an overall priority (see Figure 8.1) and a programmable priority. The programmable priority is the primary priority for maskable hardware interrupts. The overall priority is the secondary priority for maskable hardware interrupts. Since all maskable interrupts are set to a programmable priority of seven on reset, the overall priority of the interrupts determines the priority in which each interrupt is granted by the interrupt controller until programmable priorities are changed by reconfiguring the control registers. The overall priority levels shown in Figure 8.1 are not the same as the programmable priority level that is associated with each maskable hardware interrupt. Each of the maskable hardware interrupts has a programmable priority from zero to seven, with zero being the highest priority.

For example, if the INT4 - INT0 interrupts are all changed to programmable priority six and no other programmable priorities are changed from the reset value of seven, then the INT4 - INT0 interrupts take precedence over all other maskable interrupts. (Within INT4 - INT0, INT0 takes precedence over INT1, and INT1 takes precedence over INT2, etc., because of the underlying hierarchy of the overall priority.).

8.4. Software Exceptions, Traps and NMI

User programming cannot mask the following predefined interrupts.

Divide Error Exception (Interrupt Type 00h)

Generated when a DIV or IDIV instruction quotient cannot be expressed in the number of destination bits.

Trace Interrupt (Interrupt Type 01h)

If the trace flag (TF) in the Processor Status flags register is set, the trace interrupt is generated after most instructions. This interrupt allows programs to execute in single-step mode. The interrupt is not generated after prefix instructions like REP, instructions that

modify segment registers like POP DS, or the WAIT instruction. Taking the trace interrupt clears the TF bit after the processor status flags are pushed onto the stack. The IRET instruction at the end of the single step interrupts service routine restores the processor status flags (and the TF bit) and transfers control to the next instruction to be traced. Pushing the processor status flags onto the stack, setting the TF flag on the stack, and then popping the flags initiate trace mode.

■ Non-Maskable Interrupt—NMI (Interrupt Type 02h)

The NMI pin provides an external interrupt source that is serviced regardless of the state of the IF (interrupt enable flag) bit. No external interrupt acknowledge sequence is performed for an NMI interrupt. A typical use of NMI is to activate a power failure routine.

■ Breakpoint Interrupt (Interrupt Type 03h)

An interrupt caused by the 1-byte version of the INT instruction (INT3).

■ INTO Detected Overflow Exception (Interrupt Type 04h)

Generated by an INTO instruction if the OF bit is set in the Processor Status Flags (FLAGS) register.

■ Array BOUNDS Exception (Interrupt Type 05h)

Generated by a BOUND instruction if the array index is outside the array bounds. The array bounds are located in memory at a location indicated by one of the instruction operands. The other operand indicates the value of the index to be checked.

■ Unused Opcode Exception (Interrupt Type 06h)

Generated if execution is attempted on undefined opcodes.

■ ESC Opcode Exception (Interrupt Type 07h)

Generated if execution of ESC opcodes (D8h-DFh) is attempted. The microcontrollers do not check the escape opcode trap bit. The return address of this exception points to the ESC instruction that caused the exception. If a segment override prefix preceded the ESC instruction, the return address points to the segment override prefix. All numeric coprocessor opcodes cause a trap. The IMS16C does not support the numeric coprocessor interface.

■ Interrupt Acknowledge

Interrupts can be acknowledged in two different ways : the internal interrupt controller can provide the interrupt type or an external interrupt controller can provide the interrupt type. The processor requires the interrupt type as an index into the interrupt vector table. When the internal interrupt controller is supplying the interrupt type, no bus cycles are generated. The only external indication that an interrupt is being serviced is the processor reading the interrupt vector table. When an external interrupt
controller is supplying the interrupt type, the processor generates two interrupt acknowledge bus cycles. The interrupt type is written to the AD7-AD0 lines by the external interrupt controller during the second bus cycle.

■ Interrupt Controller Reset Conditions

On reset, the interrupt controller performs the following nine actions:

1. All special fully nested mode (SFNM) bits are reset, implying fully nested mode.
2. All priority (PR) bits in the various control registers are set to 1. This places all sources

at the lowest priority (level 7).
3. All level-triggered mode (LTM) bits are reset to 0, resulting in edge-triggered mode.
4. All interrupt in-service bits are reset to 0.
5. All interrupt request bits are reset to 0.
6. All mask (MSK) bits are set to 1. All interrupts are masked.
7. All cascade (C) bits are reset to 0 (non-cascade).
8. The interrupt priority mask is set to 7, allowing interrupts of all priorities.
9. The interrupt controller is initialized to master mode.

8.5. Fully Nested Mode

In fully nested mode, five pins are used as direct interrupt requests as in Figure 8.5. The interrupt types for these five inputs are generated internally. An in-service bit is provided for every interrupt source. If a lower-priority device requests an interrupt while the in-service bit (IS) is set for a higher priority interrupt, the interrupt controller does not generate any interrupt. In addition, if another interrupts request occurs from the same interrupt source while the in-service bit is set; the interrupt controller generates no interrupt. This allows interrupt service routines operating with interrupts enabled to be suspended only by interrupts of equal or higher priority than the in-service interrupt. When an interrupt service routine is completed, the proper IS bit must be reset by writing the interrupt type to the EOI register. This is required to allow subsequent interrupts from this interrupt source and to allow servicing of lower-priority interrupts. A write to the EOI register should be executed at the end of the interrupt service routine just before the return from interrupt instruction.

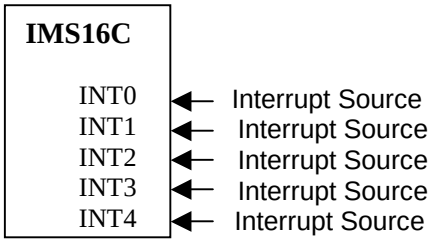


Figure 8.5 Fully Nested Mode Interrupt Controller Connections

8.6. Operation in a Polled Environment

To allow reading of the Poll register information without setting the indicated in-service bit, the IMS16C microcontroller provides a Poll Status register in addition to the Poll register. Poll register information is duplicated in the Poll Status register, but the Poll Status register can be read without setting the associated in-service bit. These registers are located in two adjacent memory locations in the peripheral control block. The interrupt controller can be used in polled mode if interrupts are not desired. When polling, interrupts are disabled and

software polls the interrupt controller as required. The interrupt controller is polled by reading the Poll Status register. Bit 15 in the Poll Status register indicates to the processor that an interrupt of high enough priority is requesting service. Bits 4 - 0 indicate to the processor the interrupt type of the highest priority source requesting service. After determining that an interrupt is pending, software reads the Poll register (rather than the Poll Status register), which causes the in-service bit of the highest priority source to be set.

8.7. End-of-Interrupt Write to the EOI Register

A program must write to the EOI register to reset the in-service (IS) bit when an interrupt service routine is completed. There are two types of writes to the EOI register: specific EOI and non-specific EOI. Non-specific EOI does not specify which IS bit is to be reset. Instead, the interrupt controller automatically resets the IS bit of the highest priority source with an active service routine. Specific EOI requires the program to send the interrupt type to the interrupt controller to indicate the source IS bit that is to be reset. Specific reset is applicable when interrupt nesting is possible or when the highest priority IS bit that was set does not belong to the service routine in progress.

8.8. Programmable Priority

Each interrupt source is equipped with a programmable priority level within its control register. These bits can be programmed to one of eight different priorities (000b - 111b). Priority level 000b has the highest priority and 111b has the lowest priority. These are the bits that determine which interrupt source will have priority over another.

8.9. Trigger Mode

The five external interrupts (INT4 - INT0) can have their inputs configured for either edge-triggered or level-triggered mode. In either mode all interrupt inputs are active high. When the inputs are configured for level-triggered mode (LTM=1) an interrupt request will occur when ever the input is pulled high. The input must remain high long enough for the processor to recognize it. If the input continues to stay high it will cause another interrupt only after the first interrupt has been processed.
To avoid this, users must mask current interrupt source. When the inputs are configured for edge-triggered mode (LTM=0) an interrupt request will occur when the input makes a low to high transition.

8.10. Interrupt Control Registers

The interrupt control registers for master mode are shown in Figure 8.10

Offset	Register Mnemonic	Register Name	Associated Pins	Comments
22h	EOI	EOI Register		Write-only Register
24h	POLL	Poll Register		Read-only Register
26h	POLLST	Poll Status Register		Read-only Register
28h	IMASK	Interrupt Mask Register	INT4 - INT0 DRQ1 - DRQ0	
2Ah	PRIMSK	Priority Mask Register		
2Ch	INSERV	In-Service Register	INT4 – INT0 DRQ1 – DRQ0	
2Eh	REQST	Interrupt Request Register	INT4 – INT0 DRQ1 – DRQ0	Read-only Register
30h	INTSTS	Interrupt Status Register		
32h	TCUCON	Timer Interrupt Control Register	TMRIN1 TMRIN0 TMROUT1 TMROUT0	
34h	DMA0CON	DMA0 Interrupt Control Register	DRQ0	
36h	DMA1CON	DMA1 Interrupt Control Register	DRQ1	
38h	I0CON	INT0 Control Register	INT0	
3Ah	I1CON	INT1 Control Register	INT1	
3Ch	I2CON	INT2 Control Register	INT2	
3Eh	I3CON	INT3 Control Register	INT3	
40h	I4CON	INT4 Control Register	INT4	
42h	WDCON	Watchdog Timer Interrupt Control Register		
44h	U0CON	UART0 Interrupt Control Register	TXD0, RXD0	
46h	U1CON	UART1 Interrupt Control Register	TXD1, RXD1	
48h	U2CON	UART2 Interrupt Control Register	TXD2, RXD2	
4Ah	H0CON	HDLC0 Interrupt Control Register	TX0, RX0	
4Ch	H1CON	HDLC1 Interrupt Control Register	TX1, RX1	

Figure 8.10 Interrupt Control Registers

8.11. INT0 and INT1 Control Registers (I0CON, Offset 38h, I1CON, Offset 3Ah) (Master Mode)

The INT0 interrupt is assigned to interrupt type 0Ch. The INT1 interrupt is assigned to interrupt type 0Dh. When cascade mode is enabled for INT0 by setting the C bit of I0CON to 1, the INT2 pin becomes INTA0, the interrupt acknowledge for INT0. When cascade mode is enabled for INT1 by setting the C bit of I1CON to 1, the INT3 pin becomes INTA1, the interrupt acknowledge for INT1.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	SFNM	C	LTM	MSK	PR2	PR1	PR0

Reset value : 0000_0000_0000_1111b

Bit	Name	Function
6	Special Fully Nested Mode	When set to 1, enables special fully nested mode.
5	Cascade Mode	When set to 1, this bit enables cascade mode.
4	Level Triggered Mode	This bit determines whether the microcontroller interprets an INT0 or INT1 interrupt request as edge- or level-sensitive. A 1 in this bit configures INT0 or INT1 as an active High, level-sensitive interrupt. A 0 in this bit configures INT0 or INT1 as a Low-to-High, edge-triggered interrupt. In either case, INT0 or INT1 must remain High until they are acknowledged.
3	Mask	This bit determines whether the INT0 or INT1 signal can cause an interrupt. A 1 in this bit masks this interrupt source, preventing INT0 or INT1 from causing an interrupt. A 0 in this bit enables INT0 or INT1 interrupts.
2-0	Priority Level	This field determines the priority of INT0 or INT1 relative to the other interrupt signals.

Priority	PR2 – PR0
(HIGH) 0	000b
1	001b
2	010b
3	011b
4	100b
5	101b
6	110b
(LOW)7	111b

Figure 8.11 INT0 and INT1 Control Registers

8.12. INT2 and INT3 Control Registers (I2CON, Offset 3Ch, I3CON, Offset 3Eh) (Master Mode)

The INT2 interrupt is assigned to interrupt type 0Eh. The INT3 interrupt is assigned to interrupt type 0Fh. The INT2 and INT3 pins can be configured as interrupt acknowledge pins INTA0 and INTA1 when cascade mode is implemented.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	LTM	MSK	PR2	PR1	PR0

Reset value : 0000_0000_0000_1111b

Bit	Name	Function
4	Level Triggered Mode	This bit determines whether the microcontroller interprets an INT2 or INT3 interrupt request as edge- or level-sensitive. A 1 in this bit configures INT2 or INT3 as an active High, level-sensitive interrupt. A 0 in this bit configures INT2 or INT3 as a Low-to-High, edge-triggered interrupt. In either case, INT2 or INT3 must remain High until they are acknowledged.
3	Mask	This bit determines whether the INT2 or INT3 signal can cause an interrupt. A 1 in this bit masks this interrupt source, preventing INT2 or INT3 from causing an interrupt. A 0 in this bit enables INT2 or INT3 interrupts.
2-0	Priority Level	This field determines the priority of INT2 or INT3 relative to the other interrupt signals.

Priority	PR2 – PR0
(HIGH) 0	000b
1	001b
2	010b
3	011b
4	100b
5	101b
6	110b
(LOW) 7	111b

Figure 8.12 INT2 and INT3 Control Registers

8.13. INT4 Control Register (I4CON, Offset 40h) (Master Mode)

The IMS16C microcontroller provides INT4, an additional external interrupt pin. This input behaves like INT3 - INT0 on the 80C186/188 microcontroller with the exception that INT4 is only intended for use as a nested-mode interrupt source.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	LTM	MSK	PR2	PR1	PR0

Reset value : 0000_0000_0000_1111b

Bit	Name	Function
4	Level Triggered Mode	This bit determines whether the microcontroller interprets an INT4 interrupt request as edge- or level-sensitive. A 1 in this bit configures INT4 as an active High, level-sensitive interrupt. A 0 in this bit configures INT4 as a Low-to-High, edge-triggered interrupt. In either case, INT4 must remain High until they are acknowledged.
3	Mask	This bit determines whether the INT4 signal can cause an interrupt. A 1 in this bit masks this interrupt source, preventing INT4 from causing an interrupt. A 0 in this bit enables INT4 interrupts.
2-0	Priority Level	This field determines the priority of INT4 relative to the other interrupt signals.

Priority	PR2 – PR0
(HIGH) 0	000b
1	001b
2	010b
3	011b
4	100b
5	101b
6	110b
(LOW) 7	111b

Figure 8.13 INT4 Control Registers

8.14. Timer and DMA Interrupt Control Registers (TCUCON, Offset 32h, DMA0CON, Offset 34h, DMA1CON, Offset 36h) (Master Mode)

The three timer interrupts are assigned to interrupt type 08h, 12h, and 13h. All three timer interrupts are configured through TCUCON, offset 32h. The DMA0 interrupt is assigned to interrupt type 0Ah. The DMA1 interrupt is assigned to interrupt type 0Bh.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	MSK	PR2	PR1	PR0

Reset value : 0000_0000_0000_1111b

Bit	Name	Function
3	Mask	This bit determines whether the corresponding signal can generate an interrupt. A 1 masks this interrupt source. A 0 enables the corresponding interrupt.
2-0	Priority Level	Sets the priority level for its corresponding source.

Priority	PR2 – PR0
(HIGH) 0	000b
1	001b
2	010b
3	011b
4	100b
5	101b
6	110b
(LOW) 7	111b

Figure 8.14 Timer and DMA Interrupt Control Registers

8.15. Watchdog Timer Interrupt Control Register (WDCON, Offset 42h) (Master Mode)

The IMS16C microcontroller provides an additional on-chip interrupt source, the watchdog timer.
This timer is constructed from existing 80C186 microcontroller pins. It is implemented by connecting the TMROUT1 output to an additional internal interrupt to create the watchdog timer interrupt. This interrupt is assigned to interrupt type 11h.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	MSK	PR2	PR1	PR0

Reset value : 0000_0000_0000_1111b

Bit	Name	Function
3	Mask	This bit determines whether the watchdog timer can cause an interrupt. A 1 in this bit masks this interrupt source, preventing the watchdog timer from causing an interrupt. A 0 in this bit enables watchdog timer interrupts.
2-0	Priority Level	This field determines the priority of the watchdog timer relative to the other interrupt signals.

Priority	PR2 – PR0
(HIGH) 0	000b
1	001b
2	010b
3	011b
4	100b
5	101b
6	110b
(LOW) 7	111b

Figure 8.15 Watchdog Timer Interrupt Control Register

8.16. UART Serial Interrupt Control Register (U0CON, Offset 44h, U1CON, Offset 46h, U2CON, Offset 48h) (Master Mode)

The Serial Port Interrupt Control register controls the operation of the asynchronous serial port interrupt source (SPI, bit 10 in the Interrupt Request register). This interrupt is assigned to interrupt type 14h.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	MSK	PR2	PR1	PR0

Reset value : 0000_0000_0000_1111b

Bit	Name	Function
3	Mask	This bit determines whether the serial port can cause an interrupt. A 1 in this bit masks this interrupt source, preventing the serial port from causing an interrupt. A 0 in this bit enables serial port interrupts.
2-0	Priority Level	This field determines the priority of the serial port relative to the other interrupt signals.

Priority	PR2 – PR0
(HIGH) 0	000b
1	001b
2	010b
3	011b
4	100b
5	101b
6	110b
(LOW) 7	111b

Figure 8.16 UART Serial Port Interrupt Control Register

8.17. Interrupt Status Register (INTSTS, Offset 30h) (Master Mode)

The Interrupt Status (INTSTS) register indicates the interrupt request status of the three Timers.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DHLT	0	0	0	0	0	0	0	0	0	0	0	0	TMR2	TMR1	TMR0

Reset value : 0000_0000_0000_0000b

Bit	Name	Function
15	DMA Halt	When set to 1, halts any DMA activity. This pin is automatically set to 1 when non-maskable interrupts occur and is reset when an IRET instruction is executed. Time-critical software, such as interrupt handlers, can modify this bit directly to inhibit DMA transfers. Because of the function of this register as an interrupt request register for the timers, the DHLT bit should not be modified by software when timer interrupts are enabled.
2-0	Timer Interrupt Request	When set to 1, these bits indicate that the corresponding timer has an interrupt request pending.

Figure 8.17 Interrupt Status Register

8.18. Interrupt Request Register (REQST, Offset 2Eh) (Master Mode)

The hardware interrupt sources have interrupt request bits inside the interrupt controller. A read from this register yields the status of these bits. The Interrupt Request register is a read-only register.

For internal interrupts, the corresponding bit is set to 1 when the device requests an interrupt. The bit is reset during the internally generated interrupt acknowledge.

For INT4 - INT0 external interrupts; the corresponding bit (I4 - I0) reflects the current value of the external signal. The device must hold this signal High until the interrupt is serviced. Generally the interrupt service routine signals the external device to remove the interrupt request.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	HDLC1	HDLC0	UART2	UART1	UART0	WDT	I4	I3	I2	I1	I0	D1	D0	0	TMR

Reset value : 0000_0000_0000_0000b

Bit	Name	Function
14	HDLC1 Interrupt Request	This bit indicates the interrupt state of the HDLC1. It is set when the HDLC channel 1 generates an interrupt request and cleared when the interrupt acknowledge cycle occurs
13	HDLC0 Interrupt Request	This bit indicates the interrupt state of the HDLC0. It is set when the HDLC channel 0 generates an interrupt request and cleared when the interrupt acknowledge cycle occurs
12	UART2 Serial Port Interrupt Request	This bit indicates the interrupt state of the UART2 serial port. It is set when the serial port generates an interrupt request and cleared when the interrupt acknowledge cycle occurs
11	UART1 Serial Port Interrupt Request	This bit indicates the interrupt state of the UART1 serial port. It is set when the serial port generates an interrupt request and cleared when the interrupt acknowledge cycle occurs
10	UART0 Serial Port Interrupt Request	This bit indicates the interrupt state of the UART0 serial port. It is set when the serial port generates an interrupt request and cleared when the interrupt acknowledge cycle occurs
9	Watchdog Timer Interrupt Request	When this bit is set to 1, the Watchdog Timer has an interrupt pending.
8-4	Interrupt Request	When set to 1, the corresponding INT pin has an interrupt pending (i.e., when INT0 is pending, I0 is set). These bits reflect the status of the external pin.
3-2	DMA Channel Interrupt Request	When set to 1, the corresponding DMA channel has an interrupt pending.
0	Timer Interrupt Request	This bit indicates the state of the timer interrupts. This bit is the logical OR of the timer interrupts requests. When set to a 1, this bit indicates that the timer control unit has an interrupt pending.

Figure 8.18 Interrupt Request Register

8.19. In-Service Register (INSERV, Offset 2Ch) (Master Mode)

The interrupt controller sets the bits in the INSERV register when the interrupt is taken. Writing the corresponding interrupt type to the End-of-Interrupt (EOI) register clears each bit in the register.

When an in-service bit is set, the microcontroller will not generate an interrupt request for the associated source, preventing an interrupt from interrupting itself if interrupts are enabled in the ISR.
In-Service Register is set to 0000h on reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	HDLC1	HDLC0	UART2	UART1	UART0	WDT	I4	I3	I2	I1	I0	D1	D0	0	TMR

Reset value : 0000_0000_0000_0000b

Bit	Name	Function
14	HDLC1 Interrupt In-Service	This bit indicates the in-service state of the HDLC Channel 1.
13	HDLC0 Interrupt In-Service	This bit indicates the in-service state of the HDLC Channel 0.
12	UART2 Serial Port Interrupt In-Service	This bit indicates the in-service state of the UART2 asynchronous serial port.
11	UART1 Serial Port Interrupt In-Service	This bit indicates the in-service state of the UART1 asynchronous serial port.
10	UART0 Serial Port Interrupt In-Service	This bit indicates the in-service state of the UART0 asynchronous serial port.
9	Watchdog Timer Interrupt In-Service	This bit indicates the in-service state of the Watchdog Timer.
8-4	Interrupt In-Service	These bits indicate the in-service state of the corresponding INT pin.
3-2	DMA Channel Interrupt In-Service	These bits indicate the in-service state of the corresponding DMA channel.
0	Timer Interrupt In-Service	This bit indicates the state of the in-service timer interrupts. This bit is the logical OR of all the timer interrupt status bits. When set to a 1, this bit indicates that the corresponding timer interrupt status bit is in-service.

Figure 8.19 In-Service Register

8.20. Priority Mask Register (PRIMSK, Offset 2Ah) (Master Mode)

The Priority Mask (PRIMSK) register provides the value that determines the minimum priority level at which maskable interrupts can generate an interrupt. The value of PRIMSK at reset is 0007h.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	PRM2	PRM1	PRM0

Reset value : 0000_0000_0000_0111b

Bit	Name	Function
2-0	Priority Field Mask	This field determines the minimum priority that is required in order for a maskable interrupt source to generate an interrupt. Maskable interrupts with programmable priority values that are numerically higher than this field are masked. The possible values are zero (000b) to seven (111b).

Priority	PR2 – PR0
(HIGH) 0	000b
1	001b
2	010b
3	011b
4	100b
5	101b
6	110b
(LOW) 7	111b

Figure 8.20 Priority Mask Register

8.21. Interrupt Mask Register (IMASK, Offset 28h) (Master Mode)

The Interrupt Mask (IMASK) register is a read/write register. Programming a bit in the IMASK register has the effect of programming the MSK bit in the associated control register.

Do not write to the interrupt mask register while interrupts are enabled. To modify mask bits while interrupts are enabled, use the individual interrupt control registers. The IMASK register is set to 7FFDh on reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	HDLC1	HDLC0	UART2	UART1	UART0	WDT	I4	I3	I2	I1	I0	D1	D0	0	TMR

Reset value : 0111_1111_1111_1101b

Bit	Name	Function
14	HDLC1 Interrupt Mask	When set to 1, this bit indicates that the HDLC1 interrupt is masked.
13	HDLC0 Interrupt Mask	When set to 1, this bit indicates that the HDLC0 interrupt is masked.
12	UART2 Serial Port Interrupt Mask	When set to 1, this bit indicates that the UART2 serial port interrupt is masked.
11	UART1 Serial Port Interrupt Mask	When set to 1, this bit indicates that the UART1 serial port interrupt is masked.
10	UART0 Serial Port Interrupt Mask	When set to 1, this bit indicates that the UART0 serial port interrupt is masked.
9	Watchdog Timer Interrupt Mask	When set to 1, this bit indicates that the Watchdog Timer interrupt is masked.
8-4	Interrupt Mask	When set to 1, an I4 - I0 bit indicates that the corresponding interrupt is masked.
3-2	DMA Channel Interrupt Mask	When set to 1, a D1 - D0 bit indicates that the corresponding DMA channel interrupt is masked.
0	Timer Interrupt Mask	When set to 1, this bit indicates that interrupt requests from the timer control unit are masked.

Figure 8.21 Interrupt Mask Register

8.22. Poll Status Register (POLLST, Offset 26h) (Master Mode)

The Poll Status (POLLST) register mirrors the current state of the Poll register. The POLLST register can be read without affecting the current interrupt request. But when the Poll register is read, the current interrupt is acknowledged and the next interrupt takes its place in the Poll register.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IREQ	0	0	0	0	0	0	0	0	0	0	S4	S3	S2	S1	S0

Reset value : 0000_0000_0000_0000b

Bit	Name	Function
15	Interrupt Request	Set to 1 if an interrupt is pending. When this bit is set to 1, the S4 - S0 field contains valid data.
4-0	Poll Status	Indicates the interrupt type of the highest priority pending interrupt. These bits are only valid if IREQ = 1.

Interrupt Type S[4:0]	Interrupt Name
08h	Timer 0 Interrupt
12h	Timer 1 Interrupt
13h	Timer 2 Interrupt
0Ah	DMA 0 Interrupt
0Bh	DMA 1 Interrupt
0Ch	INT0 Interrupt
0Dh	INT1 Interrupt
0Eh	INT2 Interrupt
0Fh	INT3 Interrupt
10h	INT4 Interrupt
11h	Watchdog Timer Interrupt
14h	UART0 Serial Port Interrupt
15h	UART1 Serial Port Interrupt
16h	UART2 Serial Port Interrupt
17h	HDLC0 Interrupt
18h	HDLC1 Interrupt

Figure 8.22 Poll Status Register

8.23. Poll Register (POLL, Offset 24h) (Master Mode)

When the Poll register is read, the current interrupt is acknowledged and the next interrupt takes its place in the Poll register. The Poll Status register mirrors the current state of the Poll register, but the Poll Status register can be read without affecting the current interrupt request.

Although the IS bit is set, the interrupt service routine does not begin execution automatically. The application software must execute the appropriate ISR.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IREQ	0	0	0	0	0	0	0	0	0	0	S4	S3	S2	S1	S0

Reset value : 0000_0000_0000_0000b

Bit	Name	Function
15	Interrupt Request	Set to 1 if an interrupt is pending. When this bit is set to 1, the S4 - S0 field contains valid data.
4-0	Poll Status	Indicates the interrupt type of the highest priority pending interrupt. Reading the Poll register acknowledges the highest priority pending interrupts and enables the next interrupt to advance into the register.

Interrupt Type S[4:0]	Interrupt Name
08h	Timer 0 Interrupt
12h	Timer 1 Interrupt
13h	Timer 2 Interrupt
0Ah	DMA 0 Interrupt
0Bh	DMA 1 Interrupt
0Ch	INT0 Interrupt
0Dh	INT1 Interrupt
0Eh	INT2 Interrupt
0Fh	INT3 Interrupt
10h	INT4 Interrupt
11h	Watchdog Timer Interrupt
14h	UART0 Serial Port Interrupt
15h	UART1 Serial Port Interrupt
16h	UART2 Serial Port Interrupt
17h	HDLC0 Interrupt
18h	HDLC1 Interrupt

Figure 8.23 Poll Register

8.24. End-of-Interrupt Register (EOI, Offset 22h) (Master Mode)

The End-of-Interrupt (EOI) register is a write-only register. Writing to the EOI register resets the in-service flags in the In-Service register. Before executing the IRET instruction that ends an interrupt service routine (ISR), the ISR should write to the EOI register to reset the IS bit for the interrupt. The specific EOI reset is the most secure method to use for resetting IS bits.

Example code for a specific EOI reset:

```
            ...
            ...                     ;ISR code
            …
            mov dx, EOI_ADDR
exit:        mov ax, int_type        ;load the interrupt type in ax
            out dx, ax              ;write the interrupt type to EOI
            popa
            iret                    ;return from interrupt
```

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NSPEC	0	0	0	0	0	0	0	0	0	0	S4	S3	S2	S1	S0

Reset value : 0000_0000_0000_0000b

Bit	Name	Function
15	Non-Specific EOI	The NSPEC bit determines the type of EOI command. When written as a 1, NSPEC indicates non-specific EOI. When written as a 0, NSPEC indicates the specific EOI interrupt type in S4–S0.
4-0	Source EOI Type	Specifies the EOI type of the interrupt that is currently being processed.

Interrupt Type S[4:0]	Interrupt Name
08h	Timer 0 Interrupt
12h	Timer 1 Interrupt
13h	Timer 2 Interrupt
0Ah	DMA 0 Interrupt
0Bh	DMA 1 Interrupt
0Ch	INT0 Interrupt
0Dh	INT1 Interrupt
0Eh	INT2 Interrupt
0Fh	INT3 Interrupt
10h	INT4 Interrupt
11h	Watchdog Timer Interrupt
14h	UART0 Serial Port Interrupt
15h	UART1 Serial Port Interrupt
16h	UART2 Serial Port Interrupt
17h	HDLC0 Interrupt
18h	HDLC1 Interrupt

Figure 8.24 End-of-Interrupt Register

9. Timer Control Unit

There are three 16-bit programmable timers in the IMS16C. Timers 0 and 1 are highly versatile and are each connected to two external pins (TMRIN0, TMROUT0, TMRIN1, TMROUT1). These two timers can be used to count or time external events, or they can be used to generate non-repetitive or variable-duty-cycle waveforms.

Timer 1 can also be configured as a watchdog timer. The watchdog timer provides a mechanism for detecting software crashes or hangs. The timer 1 interrupt request is directly connected to the watchdog timer interrupt request. The watchdog timer interrupts active makes IMS16C hardware reset. Software developers must first program the TIMER1 Mode/Control, Count, and Max Count registers, and then program the Watchdog Timer Interrupt Control register. The TIMER1 Count register must be reloaded at intervals less than the TIMER1 max count to assure the watchdog interrupt is not taken. If the code crashes or hangs, the TIMER1 countdown can cause a watchdog interrupt.
Timer 2 is not connected to any external pins. It can be used for real-time coding and time-delay applications. It can also be used as a presale to timer 0 and timer 1 or as a DMA request source.

Offset	Register Mnemonic	Register Name
50h	T0CNT	Timer 0 Count
52h	T0CMPA	Timer 0 Maxcount Compare A
54h	T0CMPB	Timer 0 Maxcount Compare B
56h	T0CON	Timer 0 Mode/Control
58h	T1CNT	Timer 1 Count
5Ah	T1CMPA	Timer 1 Maxcount Compare A
5Ch	T1CMPB	Timer 1 Maxcount Compare B
5Eh	T1CON	Timer 1 Mode/Control
60h	T2CNT	Timer 2 Count
62h	T2CMPA	Timer 2 Maxcount Compare A
66h	T2CON	Timer 2 Mode/Control

Figure 9 Timer Control Unit Register Summaries

The timer-count registers contain the current value of a timer. The timer-count registers can be read or written at any time, regardless of whether the corresponding timer is running. The microcontroller increments the value of a timer-count register each time a timer event occurs. When the timer reaches the maximum value, it resets to 0 during the same clock cycle. In addition, timers 0 and 1 have a secondary maximum-count register.

Using both the primary and secondary maximum-count registers lets the timer alternate between two maximum values. If the timer is programmed to use only the primary maximum-count register, the timer output pin switches Low for one clock cycle, the clock cycle after the maximum value is reached. If the timer is programmed to use both of its maximum-count registers, the output pin creates a waveform by indicating which maximum-count register is currently in control. The duty cycle and frequency of the waveform depend on the values in the alternating maximum-count registers.

Each timer is serviced on every fourth clock cycle. Therefore, a timer can operate at a

maximum speed of one-quarter of the internal clock frequency. A timer can be clocked externally at the same maximum frequency of one-fourth of the internal clock frequency. However, because of internal synchronization and pipelining of the timer circuitry, the timer output takes up to six clock cycles to respond to the clock or gate input. The timers are run by the processor's internal clock. If power-save mode is in effect, the timers operate at the reduced power-save clock rate.

9.1. Timer 0 and Timer 1 Mode and Control Registers (T0CON, Offset 56h, T1CON, Offset 5Eh)

These registers control the functionality of timer 0 and timer 1. The value of T0CON and T1CON at reset is 0002h.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN	INH	INT	RIU	0	0	0	0	0	0	MC	RTG	P	EXT	ALT	CONT

Reset value : 0000_0000_0000_0010b

Bit	Name	Function
15	Enable Bit	This is the count enable bit for the timer. When set, the timer is enabled to increment. When cleared, the timer is disabled from counting. If CONT=0, the EN bit is automatically cleared upon a maximum count.
14	Inhibit Bit	INH is a write protect for the EN bit. If 1 during a write operation to the control registers, then EN is modified by the write. If 0 during a write, then EN is not modified by the write. This bit is not stored in the control register, and is always 0 when read.
13	Interrupt Bit	When set to 1, an interrupt request is generated when the count register equals a maximum count. If the timer is configured in dual maxcount Mode, an interrupt is generated each time the count reaches maxcount A or maxcount B. When INT is set to 0, the timer will not issue interrupt requests. If the enable bit is cleared after an interrupt request has been generated but before the pending interrupt is serviced, the interrupt request will still be present.
12	Register in Use Bit	When the Maxcount Compare A register is being used for comparison to the timer count value, this bit is set to 0. When the Maxcount Compare B register is being used, this bit is set to 1.
5	Maximum Count Bit	The MC bit is set to 1 when the timer reaches a maximum count. In dual maxcount mode, the bit is set each time either Maxcount Compare A or B register is reached. This bit is set regardless of the timer interrupt-enable bit. The MC bit can be used to monitor timer status through software polling instead of through interrupts.
4	Retrigger Bit	Determines the control function provided by the timer input pin. When set to 1, a 0 to 1 edge transition on TMRIN0 or TMRIN1 resets the count. When set to 0, a High input enables counting and a Low input holds the timer value. This bit is ignored when external clocking (EXT=1) is selected.
3	Prescaler Bit	When set to 1, the timer is prescaled by timer 2. When set to 0, the timer counts up every fourth CLKOUT period. This bit is ignored when external clocking is enabled (EXT=1).
2	External Clock Bit	When set to 1, an external clock is used. When set to 0, the internal clock is used.
1	Alternate Compare Bit	When set to 1, the timer counts to maxcount compare A, then resets the count register to 0. Then the timer counts to maxcount compare B, resets the count register to zero, and starts over with maxcount compare A. If ALT is clear, the timer counts to maxcount compare A and then resets the count register to zero and starts counting again against maxcount compare A. In this case, maxcount compare B is not used.
0	Continuous Mode Bit	When set to 1, CONT causes the associated timer to run in the normal continuous mode.

Figure 9.1 Timer 0 and Timer 1 Mode and Control Registers

9.2. Timer 2 Mode and Control Register (T2CON, Offset 66h)

This register controls the functionality of timer 2. The value of T2CON at reset is 0000h.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN	INH	INT	0	0	0	0	0	0	0	MC	0	0	0	0	CONT

Reset value : 0000_0000_0000_0000b

Bit	Name	Function
15	Enable Bit	When EN is set to 1, the timer is enabled. When set to 0, the timer is inhibited from counting. Do not write to this bit unless the INH bit is set to 1 during the same write.
14	Inhibit Bit	Allows selective updating of enable (EN) bit. When INH is set to 1 during a write, EN can be modified on the same write. When INH is set to 0 during a write, writes to EN are ignored. This bit is not stored and is always read as 0.
13	Interrupt Bit	When INT is set to 1, an interrupt request is generated when the count register equals a maximum count. When INT is set to 0, the timer will not issue interrupt requests. If the EN enable bit is cleared after an interrupt request has been generated, but before the pending interrupt is serviced, the interrupt request remains active.
5	Maximum Count Bit	The MC bit is set to 1 when the timer reaches its maximum count. This bit is set regardless of the timer interrupt-enable bit. The MC bit can be used to monitor timer status through software polling instead of through interrupts.
0	Continuous Mode Bit	When CONT is set to 1, it causes the associated timer to run continuously. When set to 0, EN is cleared after each timer counts sequence and the timer halts on reaching the maximum count.

Figure 9.2 Timer 2 Mode and Control Register

9.3. Timer Count Registers (T0CNT, Offset 50h, T1CNT, Offset 58h, T2CNT, Offset 60h)

These registers can be incremented by one every four internal processor clocks. Timer 0 and timer 1 can also be configured to increment based on the TMRIN0 and TMRIN1 external signals, or timer 2 can rescale them.

The count registers are compared to maximum count registers and various actions are triggered based on reaching a maximum count.

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

Reset value : 0000_0000_0000_0000b

Bit	Name	Function
15 - 0	Timer Count Value	This register contains the current count of the associated timer. The count is incremented every fourth processor clock in internal clocked mode, or each time the timer 2 maxcount is reached if prescaled by timer 2. Timer 0 and timer 1 can be configured for external clocking based on the TMRIN0 and TMRIN1 signals.

Figure 9.3 Timer Count Registers

9.4. Timer Maxcount Compare Registers (T0CMPA, Offset 52h, T0CMPB, Offset 54h, T1CMPA, Offset 5Ah, T1CMPB, Offset 5Ch, T2CMPA, Offset 62h)

These registers serve as comparators for their associated count registers. Timer 0 and timer 1 each have two maximum count compare registers. Timer 0 and timer 1 can be configured to count and compare to register A and then count and compare to register B. Using this method, the TMROUT0 or TMROUT1 signals can be used to generate waveforms of various duty cycles.

Timer 2 has one compare register, T2CMPA. If a maximum count compare register is set to 0000h, the timer associated with that compare register will count from 0000h to FFFFh before requesting an interrupt.

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

Reset value : 0000_0000_0000_0000b

Bit	Name	Function
15 - 0	Timer Count Value	This register contains the maximum value a timer will count to before resetting its count register to 0.

Figure 9.4 Timer Maxcount Compare Registers

10. DMA Control Unit

Direct memory access (DMA) permits transfer of data between memory and peripherals without CPU involvement. The DMA unit in the IMS16C microcontroller provides two high-speed DMA channels. Data transfers can occur between memory and I/O spaces (e.g., memory to I/O) or within the same space (e.g., memory-to-memory or I/O-to- I/O).

Either bytes or words can be transferred to or from even or odd addresses on the IMS16C. Two bus cycles (a minimum of eight clocks) are necessary for each data transfer. Each channel accepts a DMA request from one of three sources: the channel request pin (DRQ1, DRQ0) , (UART0, UART1) and Timer 2. The two DMA channels can be programmed with different priorities to resolve simultaneous DMA requests, and transfers on one channel can interrupt the other channel. Figure 10 Shows DMA Controller Block Diagram in IMS16C.

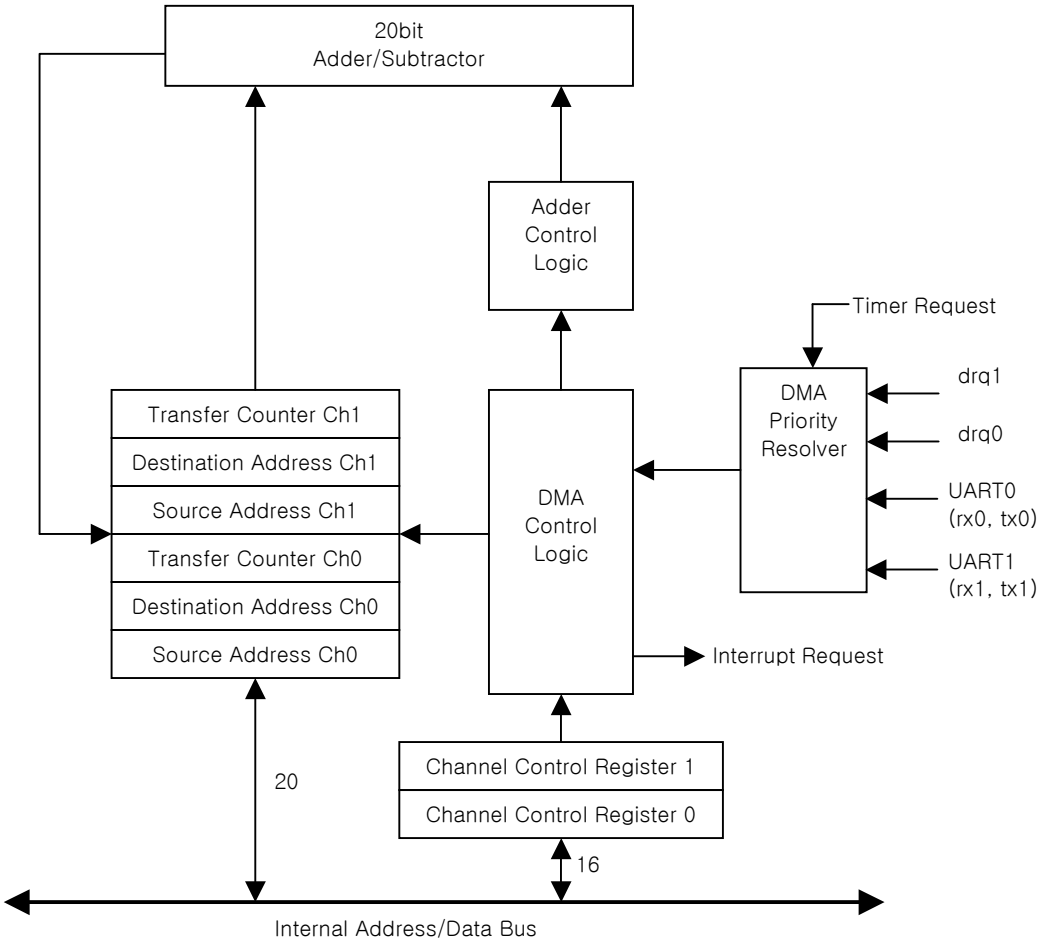


Figure 10 DMA Controller Block Diagram

10.1. DMA Operation

The format of the DMA control block is shown in Figure10.1. Six registers in the peripheral control block define the operation of each channel. The DMA registers consist of a 20-bit source address (2 registers), a 20-bit destination address (2 registers), a 16-bit transfer count register, and a 16-bit control register.

Offset	Register Mnemonic	Register Name
C0h	D0SRCL	DMA 0 Source Address Low
C2h	D0SRCH	DMA 0 Source Address high
C4h	D0DSTL	DMA 0 Destination Address Low
C6h	D0DSTH	DMA 0 Destination Address High
C8h	D0TC	DMA 0 Transfer Count
CAh	D0CON	DMA 0 Control
D0h	D1SRCL	DMA 1 Source Address Low
D2h	D1SRCH	DMA 1 Source Address high
D4h	D1DSTL	DMA 1 Destination Address Low
D6h	D1DSTH	DMA 1 Destination Address High
D8h	D1TC	DMA 1 Transfer Count
DAh	D1CON	DMA 1 Control

Figure 10.1 DMA Controller Register Summary

The DMA transfer count register (DTC) specifies the number of DMA transfers to be performed. Up to 64 Kbytes or 64K words can be transferred with automatic termination. The DMA control registers define the channel operations (see Figure 10.1). All registers can be modified or altered during any DMA activity. Any changes made to these registers are reflected immediately in DMA operation.
The sections on the following pages describe the control registers that are used to configure and operate the two DMA channels.

The IMS16C has the added feature of being able to DMA to and from the serial ports(UART0, UART1). This is accomplished by programming the DMA controller to perform transfers between a data buffer and a serial port peripheral control register. ***It is important to note that when a DMA channel is in use by a serial port, the corresponding external DMA request signal is deactivated.**

For DMA to the serial port, the transmit data register address, either I/O mapped or memory mapped, should be specified as a byte destination for the DMA by writing the address of the register into the DMA destination low and DMA destination high registers. The destination address (the address of the transmit data register) should be configured as a constant throughout the DMA operation. The serial port transmitter acts as the synchronizing device so the DMA channel should be configured as destination synchronized.

For DMA from the serial port, the receive data register address, either I/O mapped or memory mapped, should be specified as a byte source for the DMA by writing the address of the register into the DMA source and DMA source high registers. The source address (the address of the receive data register) should be configured as a constant throughout the DMA. The serial port receiver acts as the synchronizing device so the DMA channel should be configured as source synchronized.

10.2. DMA Control Registers (D0CON, Offset CAh, D1CON, Offset DAh)

The DMA control registers determine the mode of operation for the DMA channels. These registers specify the following options:

- Whether the destination address is memory or I/O space
- Whether the destination address is incremented, decremented, or maintained constant after each transfer
- Whether the source address is memory or I/O space
- Whether the source address is incremented, decremented, or maintained constant after each transfer
- If DMA activity ceases after a programmed number of DMA cycles
- If an interrupt is generated after the last transfer
- The mode of synchronization
- The relative priority of one DMA channel with respect to the other DMA channel
- Whether timer 2 DMA requests are enabled or disabled
- Whether bytes or words are transferred

The DMA channel control registers can be changed while the channel is operating. Any changes made during DMA operations affect the current DMA transfer. The value of D0CON and D1CON at reset is 0000h.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMIO	DDEC	DINC	SMIO	SDEC	SINC	TC	INT	SYN1	SYN0	P	TDRQ	0	CHG	ST	BW

Reset value : 0000_0000_0000_0000b

Bit	Name	Function
15	Destination Address Space Select	Selects memory or I/O space for the destination address. When DMIO is set to 1, the destination address is in memory space. When set to 0, the destination address is in I/O space.
14	Destination Decrement	When DDEC is set to 1, the destination address is automatically decremented after each transfer. The address decrements by 1 or 2, depending on the byte/word bit (BW, bit 0). The address remains constant if the increment and decrement bits are set to the same value (00b or 11b).
13	Destination Increment	When DINC is set to 1, the destination address is automatically incremented after each transfer. The address increments by 1 or 2, depending on the byte/word bit (BW, bit 0). The address remains constant if the increment and decrement bits are set to the same value (00b or 11b).
12	Source Address Space Select	When SMIO is set to 1, the source address is in memory space. When set to 0, the source address is in I/O space.
11	Source Decrement	When SDEC is set to 1, the source address is automatically decremented after each transfer. The address decrements by 1 or 2 depending on the byte/word bit (BW, bit 0). The address remains constant if the increment and decrement bits are set to the same value (00b or 11b).
10	Source Increment	When SINC is set to 1, the source address is automatically incremented after each transfer. The address increments by 1 or 2 depending on the byte/word bit (B/W, bit 0). The address remains constant if the increment and decrement bits are set to the same value (00b or 11b).
9	Terminal Count	The DMA decrements the transfer count for each DMA transfer. When TC is set to 1, source or destination synchronized DMA transfers terminate when the count reaches 0. When TC is set to 0, source or destination synchronized DMA transfers do not terminate when the count reaches 0.

<table>
<tr><td></td><td></td><td>Unsynchronized DMA transfers always terminate when the count reaches 0, regardless of the setting of this bit.</td></tr>
<tr><td>8</td><td>Interrupt</td><td>When INT is set to 1, the DMA channel generates an interrupt request on completion of the transfer count. The TC bit must also be set to generate an interrupt.</td></tr>
<tr><td>7-6</td><td>Synchronization Type</td><td>The SYN1, SYN0 bits select channel synchronization as shown below.

<table><tr><td>SYN1</td><td>SYN0</td><td>Sync Type</td></tr><tr><td>0</td><td>0</td><td>Unsynchronized</td></tr><tr><td>0</td><td>1</td><td>Source Synch</td></tr><tr><td>1</td><td>0</td><td>Destination Synch</td></tr><tr><td>1</td><td>1</td><td>Reserved</td></tr></table></td></tr>
<tr><td>5</td><td>Relative Priority</td><td>When P is set to 1, it selects high priority for this channel relative to the other channel during simultaneous transfers.</td></tr>
<tr><td>4</td><td>Timer Enable/Disable Request</td><td>When TDRQ is set to 1, it enables DMA requests from timer 2. When set to 0, TDRQ disables DMA requests from timer 2.</td></tr>
<tr><td>2</td><td>Change Start Bit</td><td>This bit must be set to 1 during a write to allow modification of the ST bit. When change start bit is set to 0 during a write, ST is not altered when writing the control word.</td></tr>
<tr><td>1</td><td>Start/Stop DMA Channel</td><td>The DMA channel is started when the start bit is set to 1. This bit can be modified only when the change start bit is set to a 1 during the same register writes.</td></tr>
<tr><td>0</td><td>Byte/Word Select</td><td>On the IMS16C microcontroller, when BW is set to 1, word transfers are selected. When BW is set to 0, byte transfers are selected.</td></tr>
</table>

Figure 10.2 DMA Control Registers

10.3. DMA Transfer Count Registers (D0TC, Offset C8h, D1TC, Offset D8h)

Each DMA channel maintains a 16-bit DMA Transfer Count register (DTC). This register is decremented after every DMA cycle, regardless of the state of the TC bit in the DMA Control register. However, if the TC bit in the DMA control word is set or if unsynchronized transfers are programmed, DMA activity terminates when the Transfer Count register reaches 0.

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

Reset value : 0000_0000_0000_0000b

Bit	Name	Function
15-0	DMA Transfer Count	Contains the transfer count for a DMA channel. Value is decremented by 1 after each transfer.

Figure 10.3 DMA Transfer Count Registers

10.4. DMA Destination Address High Register (High Order Bits) (D0DSTH, Offset C6h, D1DSTH, Offset D6h)

Each DMA channel maintains a 20bit destination and a 20bit source register. Each register takes up two full 16-bit registers (the high register and the low register) in the peripheral control block. For each DMA channel to be used, all four registers must be initialized. These registers can be individually incremented or decremented after each transfer. If word transfers are performed, the address is incremented or decremented by 2 after each transfer. If byte transfers are performed, the address is incremented or decremented by 1.
Each register can point into either memory or I/O space. The user must program the upper four bits to 0000b in order to address the normal 64K I/O space. Since the DMA channels can perform transfers to or from odd addresses, there is no restriction on values for the destination and source address registers. Higher transfer rates can be achieved on the IMS16C microcontroller if all word transfers are performed to or from even addresses so that accesses occur in single, 16-bit bus cycles.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	DDA[19:16]			

Reset value : 0000_0000_0000_0000b

Bit	Name	Function
3-0	DMA Destination Address High	These bits are driven onto A19-A16 during the write phase of a DMA transfer.

Figure 10.4 DMA Destination Address High Register

10.5. DMA Destination Address Low Register (Low Order Bits) (D0DSTL, Offset C4h, D1DSTL, Offset D4h)

The Figure 10.5 shows the DMA Destination Address Low Register.

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

Reset value : 0000_0000_0000_0000b

Bit	Name	Function
15-0	DMA Destination Address Low	These bits are driven onto A15-A0 during the write phase of a DMA transfer.

Figure 10.5 DMA Destination Address Low Register

10.6. DMA Source Address High Register (High Order Bits) (D0SRCH, Offset C2h, D1SRCH, Offset D2h)

Each DMA channel maintains a 20bit destination and a 20bit source register. Each register takes up two full 16bit registers (the high register and the low register) in the peripheral control block. For each DMA channel to be used, all four registers must be initialized. These registers can be individually incremented or decremented after each transfer. If word transfers are performed, the address is incremented or decremented by 2 after each transfer. If byte transfers are performed, the address is incremented or decremented by 1.
Each register can point into either memory or I/O space. The user must program the upper four bits to 0000b in order to address the normal 64K I/O space. Since the DMA channels can perform transfers to or from odd addresses, there is no restriction on values for the destination and source address registers. Higher transfer rates can be achieved on the IMS16C microcontroller if all word transfers are performed to or from even addresses so that accesses occur in single, 16-bit bus cycles.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	DSA[19:16]			

Reset value : 0000_0000_0000_0000b

Bit	Name	Function
3-0	DMA Source Address High	These bits are driven onto A19-A16 during the read phase of a DMA transfer.

Figure 10.6 DMA Source Address High Register

10.7. DMA Source Address Low Register (Low Order Bits) (D0SRCL, Offset C0h, D1SRCL, Offset D0h)

The Figure 10.7 shows the DMA Destination Address Low Register.

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

Reset value : 0000_0000_0000_0000b

Bit	Name	Function
15-0	DMA Source Address Low	These bits are driven onto A15-A0 during the read phase of a DMA transfer.

Figure 10.7 DMA Source Address Low Register

10.8. DMA Requests

Data transfers can be either source or destination synchronized - either the source of the data or the destination of the data can request the data transfer. DMA transfers can also be unsynchronized (i.e., the transfer takes place continually until the correct number of transfers has occurred).

During source synchronized or unsynchronized transfers, the DMA channel can begin a transfer immediately after the end of the previous DMA transfer, and a complete transfer can occur every two bus cycles or six clock cycles (assuming no wait states). When destination synchronization is performed, data is not fetched from the source address until the destination device signals that it is ready to receive it.

When destination synchronized transfers are requested, the DMA controller relinquishes control of the bus after every transfer. If no other bus activity is initiated, another DMA cycle begins after two processor clocks. This allows the destination device time to remove its request if another transfer is not desired.

When the DMA controller relinquishes the bus during destination-synchronized transfers, the CPU can initiate a bus cycle. As a result, a complete bus cycle is often inserted between destination-synchronized transfers.

Synchronization Type	**Maximum DMA Transfer Rate(Mbytes/sec)**
Unsynchronized	30 Mbytes/s
Source Synchronization	30 Mbytes/s
Destination Synchronization	18 Mbytes/s

Figure 10.8 Maximum DMA Transfer Rates at 90Mhz for IMS16C

10.9. Synchronization Timing

DRQ1 or DRQ0 must be de-asserted before the end of the DMA transfer to prevent another DMA cycle from occurring. The timing for the required de-assertion depends on whether the transfer is source-synchronized or destination-synchronized.

Source Synchronization Timing

Figure 10.9.1 shows a typical source-synchronized DMA transfer. Next DMA transition is sampled at every T3 cycle falling edge.

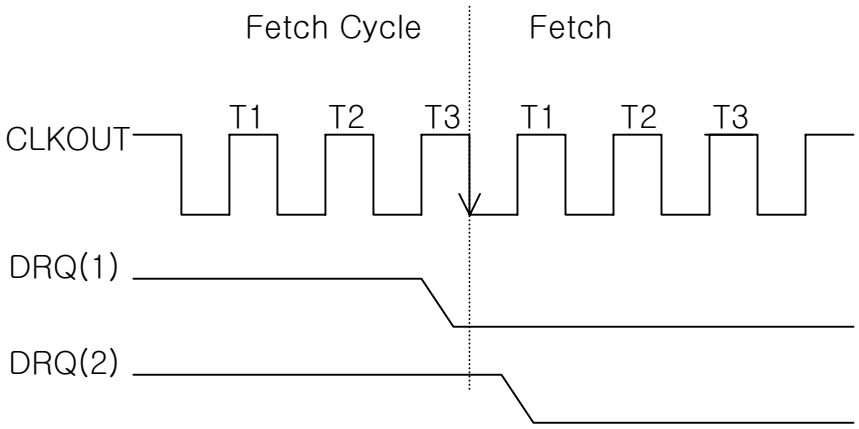


Figure 10.9.1 Source-Synchronized DMA Transfer

Notes:
(1). This source-synchronized transfer is not followed immediately by another DMA transfer.
(2). This source-synchronized transfer is immediately followed by another DMA transfer because DRQ is not de-asserted soon enough.

Destination Synchronization Timing

Figure 10.9.2 shows a typical destination-synchronized DMA transfer. A destination-synchronized transfer differs from a source-synchronized transfer in that two idle states are added to the end of the deposit cycle. Without the two idle states, the destination device would not have time to de-assert its DRQ signal. DMA transition is sampled at every T2 Deposit cycles.

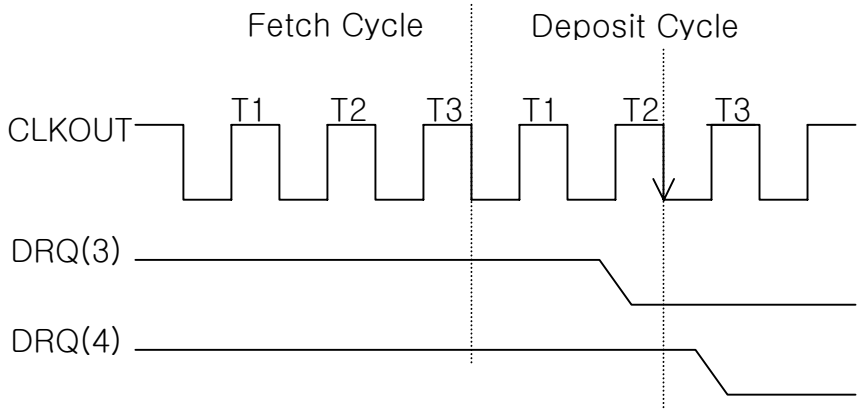


Figure 10.9.2 Destination-Synchronized DMA Transfer

Notes:
(3). This destination-synchronized transfer is not followed immediately by another DMA transfer.
(4). This destination-synchronized transfer is immediately followed by another DMA transfer because DRQ is not de-asserted soon enough.

10.10. DMA Acknowledge

No explicit DMA acknowledge signal is provided. But, status bit S6 pin of IMS16C will be set to 1 during DMA operations, and reset to 0 during CPU bus cycles.

10.11. DMA Priority

The DMA channels can be programmed so that one channel is always given priority over the other, or they can be programmed to alternate cycles when both have DMA requests. The default priority for alternating cycles is DMA0(highest), followed by DMA1(lowest).

DMA cycles always have priority over internal CPU cycles except between internally locked memory accesses or word accesses to odd memory locations. However, an external bus hold takes priority over an internal DMA cycle.

Because an interrupt request cannot suspend a DMA operation and the CPU cannot access memory during a DMA cycle, interrupt latency time suffers during sequences of continuous DMA cycles. An NMI request, however, causes all internal DMA activity to halt. This allows the CPU to respond quickly to the NMI request.

10.12. DMA Programming

DMA cycles occur whenever the ST bit of the control register is set. If synchronized transfers are programmed, a DRQ must also be generated. Therefore, the source and destination transfer address registers and the transfer count register (if used) must be programmed before the ST bit is set.
Each DMA register can be modified while the channel is operating. If the CHG bit is set to 0 when the control register is written, the ST bit of the control register will not be modified by the write. If multiple channel registers are modified, an internally Locked string transfer should be used to prevent a DMA transfer from occurring between updates to the channel registers.

10.13. DMA Channels on Reset

On reset, the state of the DMA channels is as follows:

- The ST bit for each channel is reset.
- Any transfer in progress is aborted.
- The values of the transfer count registers, source address registers, and destination address registers are undefined.

11. UART(Asynchronous) Serial Interface

The IMS16C microcontroller provides three asynchronous serial ports. The asynchronous serial port is a two-pin interface that permits full-duplex bi-directional data transfer. The asynchronous serial port supports the following features:

- Full-duplex operation
- 7-bit or 8-bit data transfers
- Odd parity, even parity, or no parity
- 1 or 2 stop bits
- Loop back mode
- Ability to break character transmit
- 2 UART Interfaces (UART0, UART1) support DMA Operation and 1 normal UART(UART2) operation

The asynchronous serial port transmit and receive sections are double-buffered. Break character recognition, framing, parity, and overrun error detection are provided. The user programs exception interrupt generation. The transmit/receive clock is based on the internal processor clock internally divided down to the serial port operating frequency. If power-save mode is in effect, the divide factor must be reprogrammed. The serial port permits 7-bit and 8-bit data transfers. DMA transfers through the serial port are not supported. The serial port generates one interrupt for all serial port events (transmit complete, data received, or error). The Serial Port Status register contains the reason for the serial port interrupt. The interrupt type assigned to the serial port is 14h. The serial port can be used in power-save mode, but the transfer rate must be adjusted to correctly reflect the new internal operating frequency and the serial port must not receive any information until the frequency is changed.

To transmit a word of data, the word must be written to the transmit data register while the register is empty. When the transmit shift register is empty the word will be transferred freeing the transmit data register for the next word of data. Therefore the next word of data may be written while the previous word is still being transmitted. The software may poll the transfer transmit data register empty bit in the status register, or the asynchronous serial port may be configured to generate an interrupt whenever the transmit data register is empty. When the receive portion of the serial port receives a valid word of data the data ready bit in the status register will be asserted. The software then has until the next word of data is received to read the previous word. If the word is not read on time it will be overwritten by the next word of data and the overrun error bit in the status register will be set.

11.1. PROGRAMMABLE REGISTERS

The asynchronous serial port is programmed through the use of five, 16-bit peripheral registers.

Offset	Register Mnemonic	Register Name
80h	U0CT	UART0 Control
82h	U0STS	UART0 Status
84h	U0TD	UART0 Transmit Data
86h	U0RD	UART0 Receive Data
88h	U0BAUD	UART0 Baud Rate Divisor

Offset	Register Mnemonic	Register Name
8Ch	U1CT	UART1 Control
8Eh	U1STS	UART1 Status
90h	U1TD	UART1 Transmit Data
92h	U1RD	UART1 Receive Data
94h	U1BAUD	UART1 Baud Rate Divisor

Offset	Register Mnemonic	Register Name
96h	U2CT	UART2 Control
98h	U2STS	UART2 Status
9Ah	U2TD	UART2 Transmit Data
9Ch	U2RD	UART2 Receive Data
9Eh	U2BAUD	UART2 Baud Rate Divisor

Figure 11.1 Asynchronous Serial Port Register Summary

11.2. Serial Port Control Register (U0CT, Offset 80h, U1CT, Offset 8Ch, U2CT, Offset 96h)

The Serial Port Control register controls both transmit and receive sections of the serial port.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMA			0	TXIE	RXIE	LOOP	BRK	BRKVAL	PMODE1	PMODE0	WLGN	STP	TMODE	RSIE	RMODE

Reset value : 0000_0000_0000_0000b

Bit	Name	Function
15-13	DMA Control Field (U0CT, U1CT)	This Field configures the serial port for use with DMA transfers according to the following table. **DMA Bits / Receive / Transmit** 000b / No DMA / No DMA 001b / DMA0 / DMA1 010b / DMA1 / DMA0 011b / Reserved / Reserved 100b / DMA0 / No DMA 101b / DMA1 / No DMA 110b / No DMA / DMA0 111b / No DMA / DMA1 DMA transfers to a serial port function as destination-synchronized DMA transfers. A new transfer is requested when the transmit data register is empty. This corresponds with the assertion of the THRE bit in the serial port status register in non-DMA mode. When the port is configured for DMA transmits, the corresponding transmit interrupt is disabled regardless of the setting of the TXIE bit. DMA transfers from the serial port function as source-synchronized DMA transfers. A new transfer is requested when the serial port receive register contains valid data. This corresponds with the assertion of the RDR bit in the serial port status register in non-DMA mode. When the port is configured for DMA receives, the corresponding receive interrupt is disabled regardless of the setting of the RSIE bit. Receive status interrupts may still be taken, as configured by the RSIE bit. Hardware handshaking may be used in conjunction with serial port DMA transfers. When a DMA channel is being used for serial port transmits or receives, the DMA request is generated internally. The corresponding external DMA request signals, DRQ0 or DRQ1, are not active for serial port DMA transfers.
11	Transmit Data Register Empty Interrupt Enable	This bit enables the serial port to generate an interrupt for transmit data register empty condition, indicating that the serial port is ready to accept a new character for transmission. If this bit is 1 and the Serial Port Transmit Data register does not contain valid data, the serial port generates an interrupt request. This bit is reset to 0.
10	Receive Data Ready Interrupt Enable	This bit enables the serial port to generate an interrupt for the receive data ready condition. If this bit is 1 and the Serial Port Receive Buffer register contains data that has been received on the serial port, the serial port generates an interrupt request. This bit is reset to 0.
9	Loop Back Test Enable	Setting this bit to 1 places the serial port in the loopback mode. In this mode, the TXD output is set High and the transmit shift register is connected to the receive shift register. Data transmitted by the transmit section is immediately received by the receive section. The loopback mode

<table>
<tr><td></td><td></td><td>is provided for testing the serial port. This bit is reset to 0.</td></tr>
<tr><td>8</td><td>Send Break</td><td>Setting this bit to 1 causes the serial port to send a continuous level on the TXD output. A break is a continuous Low on the TXD output for duration of more than one frame transmission time. The BRKVAL bit determines the level driven on the TXD output. To use the transmitter to time the frame, set the BRK bit when the transmitter is empty (indicated by the TEMT bit of the Serial Port Status register), write the serial port transmit data register, then wait until the TEMT bit is again set before resetting the BRK bit. Since the TXD output is held constant while BRK is set, the data written to transmit data register will not appear on the pin. This bit is reset to 0.</td></tr>
<tr><td>7</td><td>Break Value</td><td>This bit determines the output value transmitted on the TXD pin during a send break operation. If BRKVAL is 1, a continuous High level is driven on the TXD output. If BRKVAL is 0, a continuous Low level is driven on the TXD output. Only a continuous Low value (BRKVAL=0) will result in a break being detected by the receiver. This bit is reset to 0.</td></tr>
<tr><td>6-5</td><td>Parity Mode</td><td>This field specifies how parity generation and checking are performed during transmission and reception. If parity checking and generation is selected, a parity bit is received or sent in addition to the specified number of data bits. The value of PMODE after power-on reset is 00b.

<table><tr><th>PMODE</th><th>Parity</th></tr><tr><td>0 X</td><td>None(No parity bit in frame)</td></tr><tr><td>1 0</td><td>Odd(Odd number of 1s in frame)</td></tr><tr><td>1 1</td><td>Even(Even number of 1s in frame)</td></tr></table></td></tr>
<tr><td>4</td><td>Word Length</td><td>This bit determines the number of bits transmitted or received in a frame. If WLGN is 0, the serial port sends and receives 7 bits of data per frame. If WLGN is 1, the serial port sends and receives 8 bits of data per frame. The value of WLGN after power-on reset is 0.</td></tr>
<tr><td>3</td><td>Stop Bits</td><td>A 0 in the STP bit specifies that one stop bit is used to signify the end of a frame. A 1 in this bit specifies that two stop bits be used to signify the end of a frame. The value of STP after power-on reset is 0.</td></tr>
<tr><td>2</td><td>Transmit Mode</td><td>The TMODE bit enables data transmission and controls the operational mode of the serial port for the transmission of data. If TMODE is 0, the transmit section and transmit interrupts of the serial port are disabled. If TMODE is 1, the transmit section of the serial port is enabled. The value of TMODE after power-on reset is 0.</td></tr>
<tr><td>1</td><td>Receive Status Interrupt Enable</td><td>This bit enables the serial port to generate an interrupt because of an exception during reception. If this bit is 1 and the serial port receives a break, or experiences a framing error, parity error, or overrun error, the serial port generates a serial port interrupt. The value of RSIE after power-on reset is 0.</td></tr>
<tr><td>0</td><td>Receive Mode</td><td>This field enables data reception and controls the operational mode of the serial port for the reception of data. If RMODE is 0, the receive section and receive interrupts of the serial port are disabled. If RMODE is 1, receive section of the serial port is enabled. The value of RMODE after power-on reset is 0.</td></tr>
</table>

Figure 11.2 Serial Port Control Register

11.3. Serial Port Status Register (U0STS, Offset 82h, U1STS, Offset 8Eh, U2STS, Offset 98h)

The Serial Port Status register indicates the status of transmits and receives sections of the serial port.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	TEMT	THRE	RDR	BRKI	FER	PER	OER

Reset value : 0000_0000_0110_0000b

Bit	Name	Function
6	Transmit Empty	The TEMT bit is 1 when the transmitter has no data to transmit and the transmit shift register is empty. This indicates to software that it is safe to disable the transmit section. This bit is read-only.
5	Transmit Data Register Empty	When the THRE bit is 1, transmit data register contains invalid data and can be written with data to be transmitted. When the THRE bit is 0, transmit data register cannot be written because it contains valid data that has not yet been copied to the transmit shift register for transmission. If transmit interrupts are enabled by the TMODE and TXIE fields, a serial port interrupt request is generated when the THRE bit is 1. The THRE bit is reset automatically by writing transmit data register. This bit is read-only, allowing other bits of the Serial Port Status register to be written (i.e., resetting the BRKI bit) without interfering with the current data request.
4	Receive Data Ready	When the RDR bit is 1, receive buffer register contains data that can be read. When the RDR bit is 0, receive buffer register does not contain valid data. This bit is read-only. If receive interrupts are enabled by the RMODE and RXIE fields, a serial port interrupt request is generated when the RDR bit is 1. Reading receive buffer register resets the RDR bit.
3	Break Interrupt	The BRKI bit is set to indicate that a break has been received. If the RSIE bit is 1, the BRKI bit being set causes a serial port interrupt request. The BRKI bit should be reset by software.
2	Framing Error	The FER bit is set to indicate that a framing error occurred during reception of data. If the RSIE bit is 1, the FER bit being set causes a serial port interrupt request. The FER bit should be reset by software.
1	Parity Error	The PER bit is set to indicate that a parity error occurred during reception of data. If the RSIE bit is 1, the PER bit being set causes a serial port interrupt request. The PER bit should be reset by software.
0	Overrun Error	The OER bit is set when an overrun error occurs during reception of data. If the RSIE bit is 1, the OER bit being set causes a serial port interrupt request. The OER bit should be reset by software.

Figure 11.3 Serial Port Status Register

11.4. Serial Port Transmit Data Register (U0TD, Offset 84h, U1TD, Offset 90h, U2TD, Offset 9Ah)

Software writes this register with data to be transmitted on the serial port. The transmitter is double-buffered, and the transmit section copies data from the transmit data register to the transmit shift register (which is not accessible to software) before transmitting the data.

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

Reset value : 0000_0000_0000_0000b

Bit	Name	Function
7-0	Transmit Data	This field is written with data to be transmitted on the serial port. The THRE bit in the Serial Port Status register indicates whether there is valid data in the SPTD register. To avoid overwriting data in the SPTD register, the THRE bit should be read as a 1 before writing this register. Writing this register causes the THRE bit to be reset.

Figure 11.4 Serial Port Transmit Data Register

11.5. Serial Port Receive Data Register (U0RD, Offset 86h, U1RD, Offset 92h, U2RD, Offset 9Ch)

This register contains data received over the serial port. The receiver is double-buffered, and the receive section can be receiving a subsequent frame of data in the receive shift register (which is not accessible to software) while the receive data register is being read by software.

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

Reset value : Not Defined

Bit	Name	Function
7-0	Receive Data	This field contains data received on the serial port. The RDR bit of the Serial Port Status register indicates valid data in the SPRD register. To avoid reading invalid data, the RDR bit should be read as a 1 before the SPRD register is read. Reading this register causes the RDR bit to be reset.

Figure 11.5 Serial Port Receive Data Register

11.6. Serial Port Baud Rate Divisor Register (U0BAUD, Offset 88h, U1BAUD, Offset 94h, U2BAUD, Offset 96h)

This register specifies a clock divisor for the generation of the serial clock that controls the serial port. The serial clock rate is 16 times the baud rate of transmission or reception of data. The UART baud register specifies the number of internal processor cycles in one phase (half period) of the 16x serial clock. If power-save mode is in effect, the baud rate divisor must be reprogrammed to reflect the new processor clock frequency.

A general formula for the baud rate divisor is:

$BAUDDIV_{(decimal)}$ = (Internal Core Clock Frequency / (32 *Baud Rate)) – 1

The maximum baud rate is 1/32 of the internal Core clock and is achieved by setting BAUDDIV = 0000h.

For a 40MHz internal core clock frequency, a baud rate of 9600 bps can be achieved by

40000000/(32*9600) = 40000000/307200 = 130.208
130 – 1 = 129

BAUDDIV=129 (81h). A 1% error applies.

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

Reset value : 0000_0000_0010_0110b

Bit	Name	Function
15-0	Baud Rate Divisor	This field specifies the divisor for the internal processor clock that generates one phase (half period) of the serial clock. The serial clock operates at 16 times the data transmission or reception baud rate.

Figure 11.6 Serial Port Baud Rate Divisor Register

12. Programmable I/O Unit

The 44 pins on the IMS16C microcontroller are available as user-programmable I/O signals (PIOs). Each of these pins can be used as a PIO if the normal function of the pin is not needed. If a pin is enabled to function as a PIO signal, the normal function is disabled and does not affect the pin. A PIO signal can be configured to operate as an input or output.

After power-on reset, the PIO pins default to various configurations. The column titled Power-On Reset State in Figure12 lists the defaults for the PIOs. The system initialization code must reconfigure PIOs as required.

The AIO[2:0](A[19:17]) address pins default to normal operation on power-on reset, allowing the processor to correctly begin fetching instructions at the boot address FFFF0h. The DTR_N, DEN_N, and SRDY pins also default to normal operation on power-on reset.

PIO No.	Associated Pin	Power-On Reset Status
0	TMRIN1	Input with pullup
1	TMROUT1	Input with pullup
2	PCS6_N	Input with pullup
3	PCS5_N	Input with pullup
4	DTR_N	Normal operation
5	DEN_N	Normal operation
6	SRDY	Normal operation
7	AIO0(A17)	Normal operation
8	AIO1(A18)	Normal operation
9	AIO2(A19)	Normal operation
10	TMROUT0	Input with pullup
11	TMRIN0	Input with pullup
12	DRQ0	Input with pullup
13	DRQ1	Input with pullup
14	MCS0_N	Input with pullup
15	MCS1_N	Input with pullup
16	PCS0_N	Input with pullup
17	PCS1_N	Input with pullup
18	PCS2_N	Input with pullup
19	PCS3_N	Input with pullup
20	TXD1	Input with pullup
21	RXD1	Input with pullup
22	TXD2	Input with pullup
23	RXD2	Input with pullup
24	MCS2_N	Input with pullup
25	MCS3_N	Input with pullup
26	UZI_N	Input with pullup
27	TXD0	Input with pullup
28	RXD0	Input with pullup
29	S6	Input with pullup
30	INT4	Input with pullup
31	INT2	Input with pullup
32	RX1	Input with pullup
33	RXEN1	Input with pullup
34	RXCLK1	Input with pullup

35	TX1	Input with pullup
36	TXEN1	Input with pullup
37	TXCLK1	Input with pullup
38	RX0	Input with pullup
39	RXEN0	Input with pullup
40	RXCLK0	Input with pullup
41	TX0	Input with pullup
42	TXEN0	Input with pullup
43	TXCLK0	Input with pullup

Figure 12 PIO Pin Assignments

PIO Mode	PIO Direction	Pin Function
0	0	Normal operation
0	1	PIO input
1	0	PIO output
1	1	Reserved

Figure 12-1 PIO Mode and PIO Direction Settings

Offset	Register Mnemonic	Register Name
70h	PIOMODE0	PIO Mode 0 Register
72h	PDIR0	PIO Direction 0 Register
74h	PDATA0	PIO Data Register 0
76h	PIOMODE1	PIO Mode 1 Register
78h	PDIR1	PIO Direction 1 Register
7Ah	PDATA1	PIO Data Register 1
6Ah	PIOMODE2	PIO Mode 2 Register
6Ch	PDIR2	PIO Direction 2 Register
6Eh	PDATA2	PIO Data Register 2

Figure 12-2 Programmable I/O's Register Summary

12.1. PIO Mode 2 Register (PIOMODE2, Offset 6Ah)

The value of PIOMODE2 at reset is 0000h.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
"0000" & PIOMODE2[43:32]															

Bit	Name	Function
11-0	PIO Mode Bits[43:32]	This field with the PIO direction registers determines whether each PIO pin performs its pre-assigned function or is enabled as a custom PIO signal. The most significant bit of the PMODE field determines whether PIO43 is enabled, the next bit determines whether PIO42 is enabled, and so on.

Figure 12.1 PIO Mode 2 Register

12.2. PIO Mode 1 Register (PIOMODE1, Offset 76h)

The value of PIOMODE1 at reset is 0000h.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIOMODE1[31:16]															

Bit	Name	Function
15-0	PIO Mode Bits[31:16]	This field with the PIO direction registers determines whether each PIO pin performs its pre-assigned function or is enabled as a custom PIO signal. The most significant bit of the PMODE field determines whether PIO31 is enabled, the next bit determines whether PIO30 is enabled, and so on.

Figure 12.2 PIO Mode 1 Register

12.3. PIO Mode 0 Register (PIOMODE0, Offset 70h)

The value of PIOMODE0 at reset is 0000h.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIOMODE0[15:0]															

Bit	Name	Function
15-0	PIO Mode Bits[15:0]	This field is a continuation of the PMODE field in the PIO Mode 1 register.

Figure 12.3 PIO Mode 0 Register

12.4. PIO Direction 2 Register (PDIR2, Offset 6Ch)

The value of PDIR2 at reset is 0FFFh.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
"0000" & PDIR2[43:32]															

Bit	Name	Function
11-0	PIO Direction Bits[47:32]	This field determines whether each PIO pin acts as an input or an output. The most significant bit of the PDIR field determines the direction of PIO43, the next bit determines the direction of PIO42, and so on. A 1 in the bit configures the PIO signal as an input, and a 0 in the bit configures it as an output or as normal pin function.

Figure 12.4 PIO Direction 1 Register

12.5. PIO Direction 1 Register (PDIR1, Offset 78h)

The value of PDIR1 at reset is FFFFh.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PDIR1[31:16]															

Bit	Name	Function
15-0	PIO Direction Bits[31:16]	This field determines whether each PIO pin acts as an input or an output. The most significant bit of the PDIR field determines the direction of PIO31, the next bit determines the direction of PIO30, and so on. A 1 in the bit configures the PIO signal as an input, and a 0 in the bit configures it as an output or as normal pin function.

Figure 12.5 PIO Direction 1 Register

12.6. PIO Direction 0 Register (PDIR0, Offset 72h)

The value of PDIR0 at reset is FC0Fh.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PDIR0[15:0]															

Bit	Name	Function
15-0	PIO Direction Bits[15:0]	This field is a continuation of the PDIR field in the PIO Direction 1 register.

Figure 12.6 PIO Direction 0 Register

12.7. PIO Data Register 2 (PDATA2, Offset 6Eh)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
"0000" & PDATA2[43:32]															

Bit	Name	Function
11-0	PIO Data Bits[43:32]	This field determines the level driven on each PIO pin or reflects the external level of the pin, depending upon whether the pin is configured as an output or an input in the PIO Direction registers. The most significant bit of the PDATA field indicates the level of PIO43, the next bit indicates the level of PIO42, and so on.

Figure 12.7 PIO Data 1 Register

12.8. PIO Data Register 1 (PDATA1, Offset 7Ah)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PDATA1[31:16]															

Bit	Name	Function
15-0	PIO Data Bits[31:16]	This field determines the level driven on each PIO pin or reflects the external level of the pin, depending upon whether the pin is configured as an output or an input in the PIO Direction registers. The most significant bit of the PDATA field indicates the level of PIO31, the next bit indicates the level of PIO30, and so on.

Figure 12.8 PIO Data 1 Register

12.9. PIO Data Register 0 (PDATA0, Offset 74h)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PDATA0[15:0]															

Bit	Name	Function
15-0	PIO Data Bits[15:0]	This field is a continuation of the PDATA field in the PIO Data 1 register.

Figure 12.9 PIO Data 0 Register

13. General Purpose Register Unit

In addition to the programmable I/Os, IMS16C provides 16 words general-purpose registers. Each Register's size is 16bit wide, so these registers are restricted to word operation only. Users can access these 16 general-purpose registers through the IN or OUT instructions.

Offset	Register Mnemonic	Register Name
1Eh	GPR15	General Purpose Register 15
1Ch	GPR14	General Purpose Register 14
1Ah	GPR13	General Purpose Register 13
18h	GPR12	General Purpose Register 12
16h	GPR11	General Purpose Register 11
14h	GPR10	General Purpose Register 10
12h	GPR9	General Purpose Register 9
10h	GPR8	General Purpose Register 8
0Eh	GPR7	General Purpose Register 7
0Ch	GPR6	General Purpose Register 6
0Ah	GPR5	General Purpose Register 5
08h	GPR4	General Purpose Register 4
06h	GPR3	General Purpose Register 3
04h	GPR2	General Purpose Register 2
02h	GPR1	General Purpose Register 1
00h	GPR0	General Purpose Register 0

Figure 13 General Purpose Register Summaries

13.1. General Purpose Register (GPR0 ~ GPR15, Offset 00h ~ 1Eh)

This General Purpose Register is not initialized at power on reset.

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

Reset value : Not defined

Bit	Name	Function
15 -0	General Purpose Register	This Register is for general purpose. So, users can freely read or write word data to this register through the IN or OUT instructions.

Figure 13.1 General Purpose Register

14. SDRAM Control Unit

Synchronous DRAMs (SDRAMs) provide a significant improvement in bandwidth performance over traditional asynchronous DRAMs such as "FPM" (Fast Page Mode) and "EDO" (Extended Data Out). Synchronous DRAMs latch input address, data, and control signals on the clock rising edge, freeing the controller from having to drive address and control for the entire read or write transaction. After a preset number of clock cycles, the data is available on the output latches of the SDRAM for a READ, or can be written into its memory for a WRITE. Figure14 shows MICRON's 16Mbit SDRAM functional block diagram.

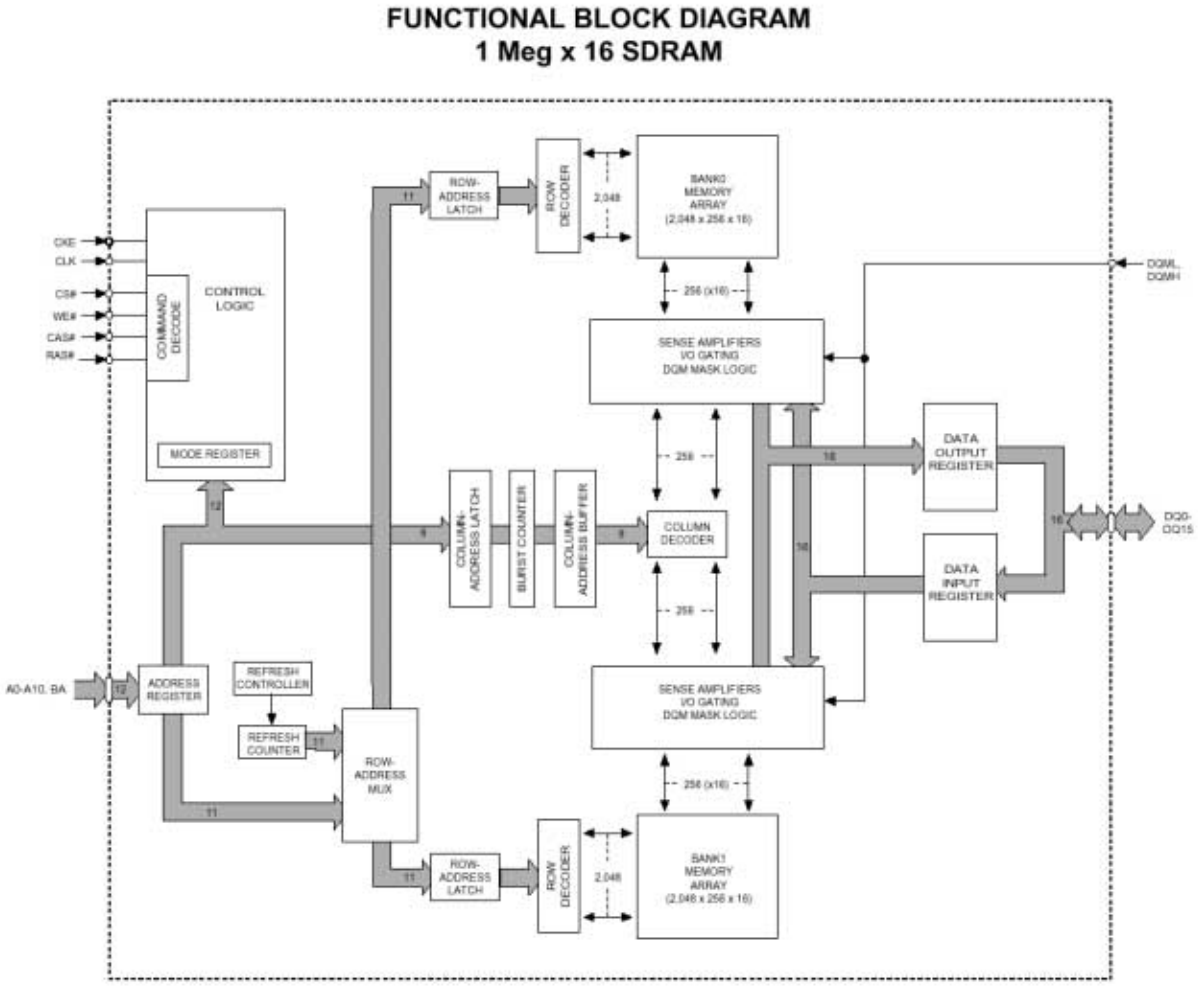


Figure 14 MICRON's 16Mbit SDRAM Block Diagram

In general, SDRAMs are multi-bank DRAM arrays that operate at 3.3V and include a command-driven synchronous interface (all signals are registered on the positive edge of the clock signal). For example, Figure 14 shows a dual 512K x 16 array architecture. Each of the two 512K x 16bit banks is organized as 2,048 rows by 256 columns by 16 bits. Read and write accesses to the SDRAM are burst oriented; accesses start at a selected location and

continue for a programmed number of locations in a programmed sequence.

Accesses begin with the registration of the active low command, followed by a read or write command. The address bits registered coincident with the active low commands are used to select the intended bank and row. The SDRAM must be initialized according to manufacturer specifications. Initialization usually consists of a sequence of precharge-all-banks, auto-refresh, and mode-register-set commands.

The majority of SDRAM employs a simple set of signal combinations called commands to carry out the basic IO. Most of these commands are one clock cycle in duration, and are clocked by the SDRAM on the rising edge. A proper setup and hold times must be observed. A sequence of these commands comprises the primitive operation of read, write and refresh.

The 1M x 16 SDRAM is organized as 512K x 16 x 2 banks. The bank selection is done by a pin called BA (bank address). The address organization of these SDRAMs are 11 rows by 8 columns. During the row address strobing, A[10:0] provides the row to select and during the column strobing, A[7:0] provides the column to select (A[10:8] should be set to logic low).

14.1. OVERVIEW OF SDRAM COMMANDS

■ Mode register set command

This command is used to program the SDRAM's mode register. The mode register controls the operation of the SDRAM, including the CAS latency, burst type, burst length, test mode and other vendor specific options. Most SDRAMs do not initialize the mode register upon power up, thus it is critical this register be initialized prior to the normal use. The SDRAM controller initializes the mode register following every system reset with a default value. During the mode register programming, the SDRAM receives the data from A[10:0] and BA, rather than from the data bus.

Most SDRAMs follows the bit field definition illustrated below, but make sure to consult the specific SDRAM vendor specs as they might contain subtle differences.

Figure 14.1 shows mode register definition for SDRAMs.

11 10	9	8 7	6 5 4	3	2 1 0
res	WBL	TM	CAS Latency	BT	Burst Length

Bit	Name	Function
9	Write Burst Length	**A9** / **Length**: 0 = Burst; 1 = Single bit
8-7	Test Mode	**A[8:7]** / **Type**: 0 = Mode register set; 1 = res; 2 = res; 3 = res
6-4	CAS Latency	**A[6:4]** / **Type**: 0 = res; 1 = res; 2 = 2; 3 = 3; 4 = res; 5 = res; 6 = res; 7 = res
3	Burst Type	**A3** / **Length**: 0 = sequential; 1 = interleave
2-0	Burst Length	**A[2:0]** / **BT=0** / **BT=1**: 0 = 1, 1; 1 = 2, 2; 2 = 4, 4; 3 = 8, 8; 4 = res, res; 5 = res, res; 6 = res, res; 7 = page, res

Figure 14.1 Mode register definition for general SDRAMs

■ Row address strobe and bank active command
This command is used to select the bank and the row where the data access is to take place. The BA input selects the bank, while A[10:0] provides the row address.

■ Precharge
The precharge command is used to begin the precharge operation to the selected bank(s). When A[10] is high, both banks are activated, while when A[10] is low, the bank selected by BA is activated.

■ Column address and write command
This command is used to select the column of the selected bank where the write is to take place. It is important that the bank selected at the row address strobe be selected again. The address bus A[7:0]
selects the column. A[10:8] should be logic low.

■ Column address and read command
This command is used to select the column of the selected bank where the read is to take place. The address bus A[7:0] selects the column. A[10:8] should be logic low.

■ Auto refresh command
This command is used to start the internal refresh cycle. An internal row counter is adjusted to point to the next row.

Figure 14.1-1 shows timing diagrams of basic commands for SDRAM control.

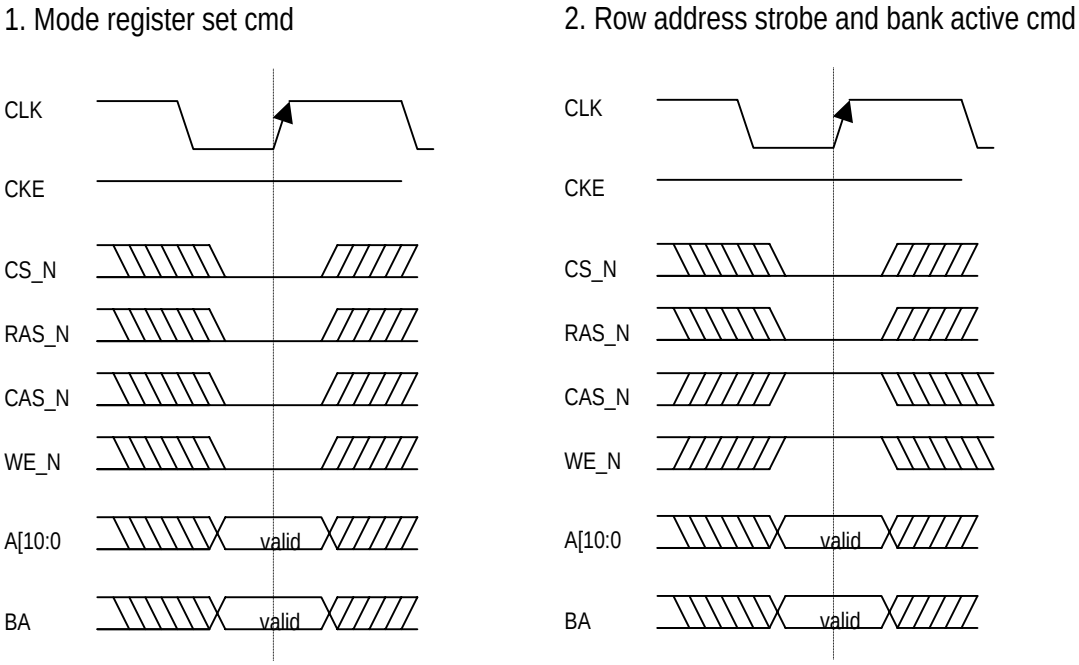
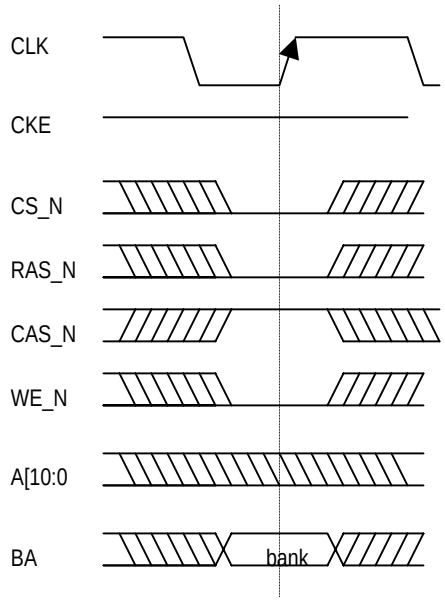
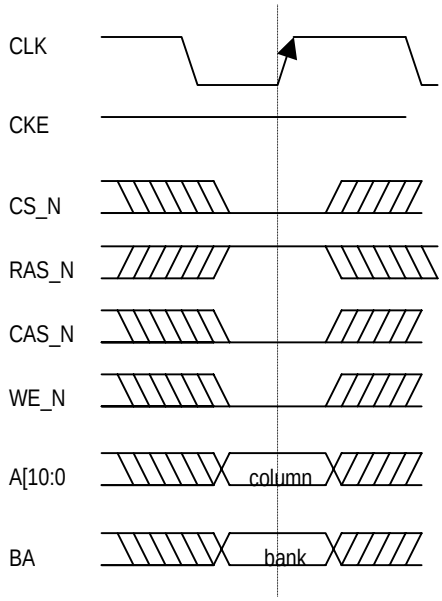


Figure 14.1-1 Timing Diagrams of basic commands for SDRAM control

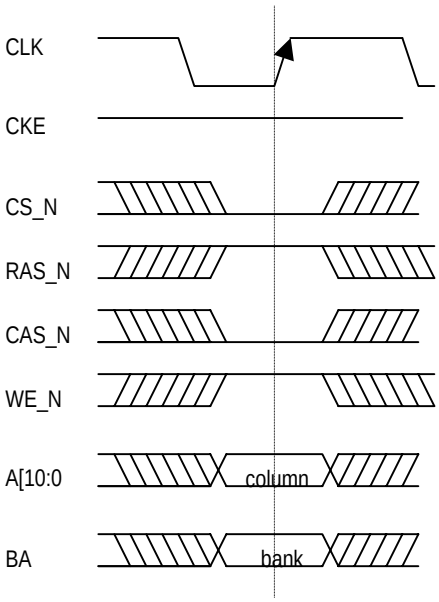
3. Precharge cmd



4. Column address and write cmd



5. Column address and read cmd



6. Autorefresh cmd

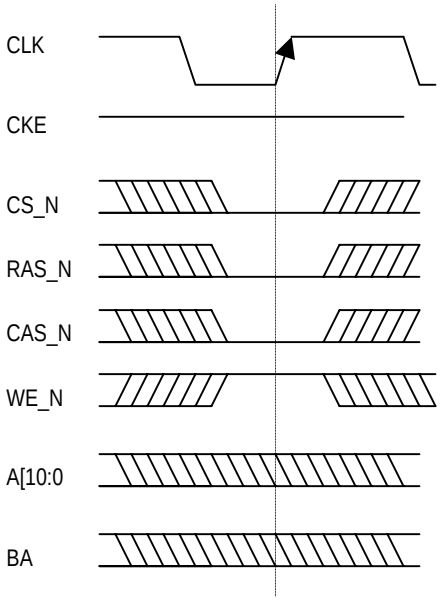


Figure 14.1-1 Timing Diagrams of basic commands for SDRAM control (Contd.)

14.2. Basic Operation

The following section describes the basic operations of single word write, single word read and refresh. Burst write and burst read are not presently supported. Upon power-up, the SDRAM's mode register must be initialized for proper operation. This is due to the fact that most SDRAMs do not initialize the mode register following a reset. However, before the register can be initialized, the SDRAM must be subjected to an initialization sequence. Notice that following power up (or reset), a repeated number of auto refresh command is necessary. The power-up sequence is also highly dependent on the SDRAM maker, so consult the specification.
In general, the sequence consists of the following:

1. After a stable power (Vcc) has been reached, the SDRAM should see a stable clock for about 200uS. During this time, no valid command should be issued.
2. Both banks must be pre-charged (precharge command)
3. One or more auto-refresh commands must be issued.
4. The mode register can now be initialized.

IMS16C SDRAM controller generates control signals for above power-up sequences for SDRAM automatically at power-on reset.

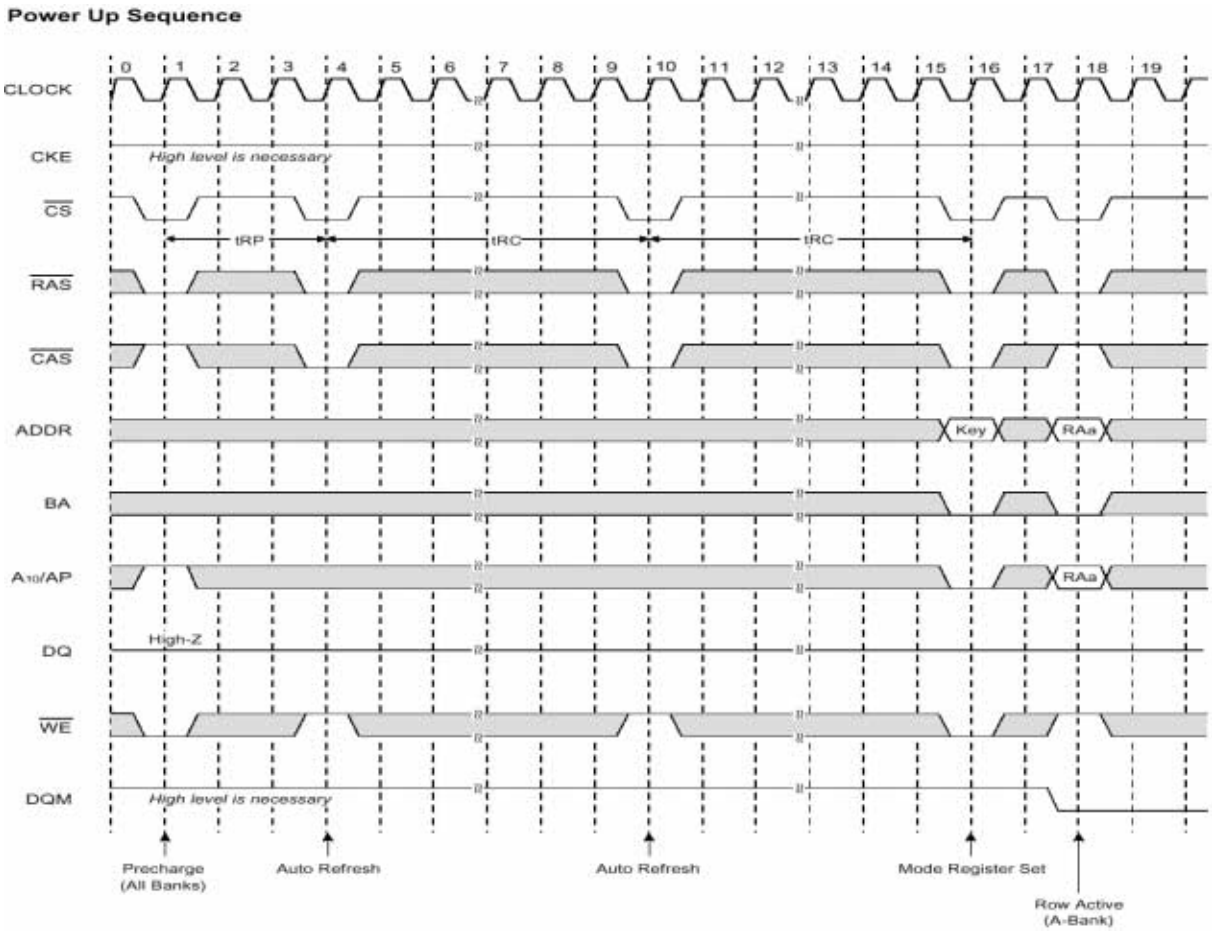


Figure 14.2 Timing Diagrams of Power-Up Sequence (Samsung SDRAM)

The basic read and write operation in Figure 14.2-1 consist of the following command sequences:

1. Row address strobe and bank active command,
2. Wait to meet tRCD
3. Column address and read command
4. Wait to meet CAS latency
5. Column address and write command
6. Column address and read command
7. Precharge all bank command
8. Wait to meet tRP
9. Wait to meet tRC before a new row activate command

Single Bit Read-Write-Read Cycle(Same Page) @CAS Latency=3, Burst Length=1

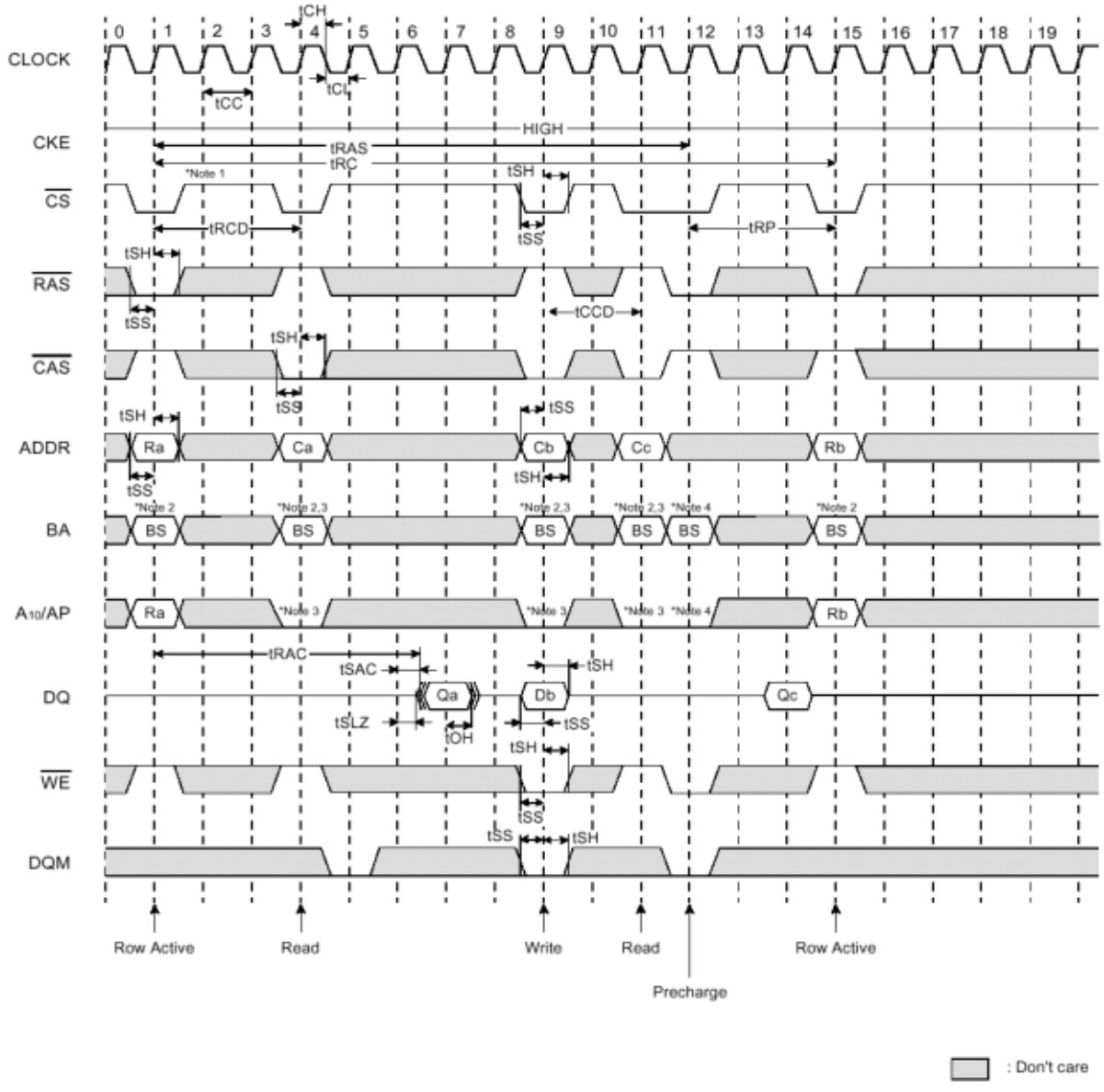


Figure 14.2-1 Timing Diagrams of read and write Sequence (Samsung SDRAM)

Once the SDRAM receives the column address strobe command, it makes the data available on its bus CAS latency cycles later. The CAS latency is programmed through the mode register. In theory this value can be programmed from 2 to 7 cycles, but most SDRAM vendors limit the choices to 2 or 3 cycles. After the data has been read, the SDRAM controller issues a precharge-all command to ready for the next data access.
The write operation inherently takes fewer cycles than the write because there are not CAS latencies involved. The SDRAM latched in the data on the rising edge of column strobe command.

The refresh operation consists of initiating the internal auto-refresh cycle of the SDRAM, and consists of the following command sequence:

1. N number of autorefresh commands
2. Delay to meet tRC before issuing another command

N represents the number of rows to be refreshed. Following the autorefresh command, the SDRAM cannot accept any new commands until tRC later. This parameter varies by SDRAM vendor.

The number of autorefresh cycles, N, to be issued depends on the type of refresh desired. As previously mentioned, in a typical 16Mbit SDRAM, there are 2048 rows per bank. And the typical refresh interval is 32mS. For an evenly distributed refresh type, this represents 1 refresh operation every 15.6uS(32ms/2048). Alternatively, a "burst" type of refresh can perform all 2048 refresh operations every 32mS. The latter approach is highly advantageous for very slow hosts. Thus N can take a value from 1 to 2048.
Note that the IMS16C SDRAM controller does not support self-refresh mode.

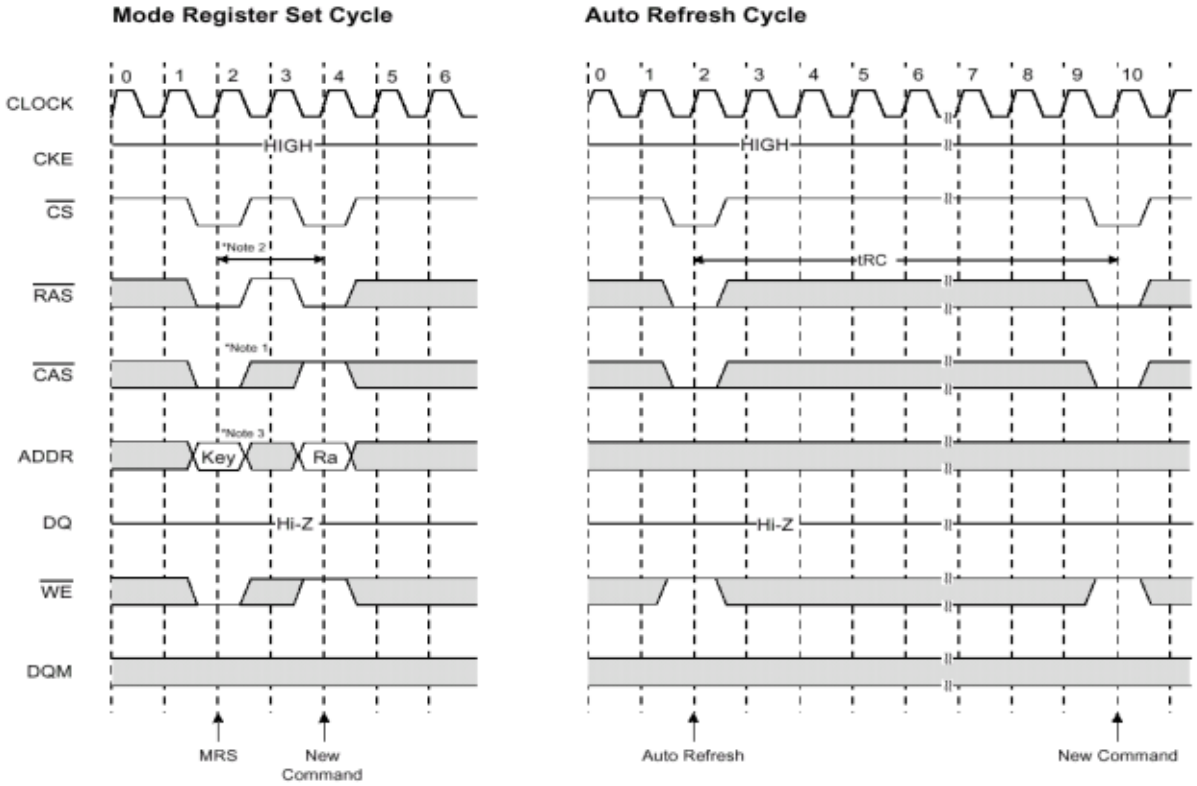


Figure 14.2-2 Timing Diagrams of mode register set and auto refresh cycle (Samsung SDRAM)

14.3. Key Features in IMS16C SDRAM Controller

The SDRAM controller in IMS16C can fully use 16Mbit SDRAM area. There are several SDRAM access modes in IMS16C SDRAM controller.

- 512k bytes mode with 4 banks (bank0, bank1, bank2, bank3)
- 1M bytes mode with 2 banks (bank01, bank23)

In 512k bytes mode, users can access whole 512k bytes in LCS area with 4 bank selection. In 1M bytes mode, users can access IMS16C memory area from 512k bytes to 960k bytes. There are 8 memory selection mode which closely related to UCS and MCS area. Figure 14.3 shows 1M bytes range selection in SDRAM control register.

SDCR[6:4]	1Mbytes Range Selection
000	0x00000 ~ 0x7FFFF (512k bytes)
001	0x00000 ~ 0x8FFFF (576k bytes)
010	0x00000 ~ 0x9FFFF (640k bytes)
011	0x00000 ~ 0xAFFFF (704k bytes)
100	0x00000 ~ 0xBFFFF (768k bytes)
101	0x00000 ~ 0xCFFFF (832k bytes)
110	0x00000 ~ 0xDFFFF (896k bytes)
111	0x00000 ~ 0xEFFFF (960k bytes)

Figure 14.3 1Mbytes range selection

If users want to access over 512k bytes memory area, careful concerns must be needed in selecting corresponding upper and middle memory area. That is, if users want to access over 512k bytes SDRAM area, UCS_N and MCS_N are not activated in SDRAM access range.

The IMS16C's AD bus shares SDRAM data bus, and IMS16C A[11:0] bus shares SDRAM address bus. IMS16C has 7 pins, which related to SDRAM interfaces (sd_cs_n, sd_ras_n, sd_cas_n, sd_wr_n, sd_ba, sd_dqm1, sd_dqm0). IMS16C's clockout pin is also used for SDRAM clock source.

SDRAM shows best performances when multi-bit burst mode is selected. This feature is suitable for cache operation. IMS16C microcontroller does not have cache feature, it can only access SDRAM in single bit. But, to the low speed and high memory capacity systems, IMS16C SDRAM controller is one of the best choices.

The IMS16C microcontroller has 3 peripheral registers related to SDRAM controller. Figure 14.3.1 shows SDRAM Control Registers.

Offset	Register Mnemonic	Register Name
EAh	SDCR	SDRAM Control Register
ECh	SDDUTY	SDRAM Autorefresh Duty Cycle Register
EEh	SDMODE	SDRAM Mode Set Register

Figure 14.3.1 SDRAM Control Registers Summary

14.4. SDRAM Control Register (SDCR, Offset EAh)

Figure 14.4 shows SDRAM control register. SDRAM controller is disabled on power on reset. If internal sync ram is enabled, SDRAM controller does not operate regardless of SDEN bit value.

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

Reset value : 0000_0000_0000_0000b

Bit	Name	Function
12	MRS	This bit selects Memory Range. When set to '0', SDRAM controller operates 512k bytes memory access. When set to '1', SDRAM controller operates 1M bytes memory access.
9-8	BANK	This bit is used for bank selection. Note that if users select 1Mbytes area, then sdcr[9:8] is fixed to "0-" or "1-". That is, sdcr[8] is don't care. If MRS is set to '1' and users want to use bank23, SDCR[9:8] must be set to either "10" or "11".
6-4	1MRS	This bit is used for 1Mbytes range selection.
0	SDEN	This bit enables SDRAM Controller. When set to '0', SDRAM controller is disabled. When set to '1', SDRAM controller is enabled.

SDCR[9:8]	512 bytes Bank	1Mbytes Bank
00	Bank0 (512k bytes)	Bank01 (1M bytes)
01	Bank1 (512k bytes)	
10	Bank2 (512k bytes)	Bank23 (1M bytes)
11	Bank3 (512k bytes)	

SDCR[6:4]	1Mbytes Range Selection
000	0x00000 ~ 0x7FFFF (512k bytes)
001	0x00000 ~ 0x8FFFF (576k bytes)
010	0x00000 ~ 0x9FFFF (640k bytes)
011	0x00000 ~ 0xAFFFF (704k bytes)
100	0x00000 ~ 0xBFFFF (768k bytes)
101	0x00000 ~ 0xCFFFF (832k bytes)
110	0x00000 ~ 0xDFFFF (896k bytes)
111	0x00000 ~ 0xEFFFF (960k bytes)

Figure 14.4 SDRAM Control Register

14.5. SDRAM Auto-refresh Duty Cycle Register (SDDUTY, Offset ECh)

Most of the storage cells of SDRAM need to be refreshed every 32ms to maintain data. An auto refresh cycle accomplishes refresh of a single row of storage cells. The internal counter increments automatically on every auto refresh cycle to refresh all the rows. An auto refresh command is issued by asserting low on CS, RAS and CAS with high on CKE and WE.

This register contains dividing value for generating autorefresh command. Internal clock of IMS16C is used for generating autorefresh command period. Core clock period is the same as CLKOUT.
The equation is as follows:

Autorefresh Duty Cycle = SDRAM Refresh Time / (SDRAM Row Num * core clock period)

For example, If IMS16C core clock speed is 50Mhz(20ns), and if SDRAM refresh time is 32ms, and SDRAM Row number is 2048 then

Autorefresh Duty Cycle = $32*10^{-3}$ / (2048 * $20*10^{-9}$) = 781.25

That is, If auto-refresh commend is generated once every 781 core clock cycles, it refreshes 2048 rows in 32ms. In this case, programmer must write 1100001101b(781_{dec}) to SDRAM Auto-refresh Duty Cycle Register.

Default value at power-on is 1200_{dec}(4B0h). It assumed that core clock speed is 80Mhz and refresh cycle is once every 15us for 2048 rows(2k/32ms).

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

Reset value : 0000_0100_1011_0000b

Bit	Name	Function
10-0	SDDUTY	SDRAM Auto-refresh Duty Cycle Value.

Figure 14.5 SDRAM Auto-refresh Duty Cycle Register

14.6. SDRAM Mode Set Register (SDMODE, Offset EEh)

Figure 14.6 shows SDRAM mode set register. Power-on reset value is 0020h. Only CAS Latency can be configured.

10	9	8 7	6 5 4	3	2 1 0
0	0	00	CAS Latency	0	000

Bit	Name	Function
6-4	CAS Latency	A[6:4] = Type: 0 = Res; 1 = Res; 2 = 2; 3 = 3; 4 = Res; 5 = Res; 6 = Res; 7 = Res

Figure 14.6 SDRAM Mode Set Register

15. On-Chip RAM Control Unit

The IMS16C microcontroller has two 8kbytes on-chip synchronous single ports RAM. It can locate in internal address space from 0x00000 to 0x03FFF. The integration of memory reduces a system design's overall cost, size and power consumption.

Users can read or write 16k bytes on-chip RAM either 8bit byte mode or 16bit word mode. Internal RAM provides the same performance as 8/16bit external zero-wait-state RAM.

There are six memory configuration methods in IMS16C microcontroller. Figure 15 shows IMS16C memory map.

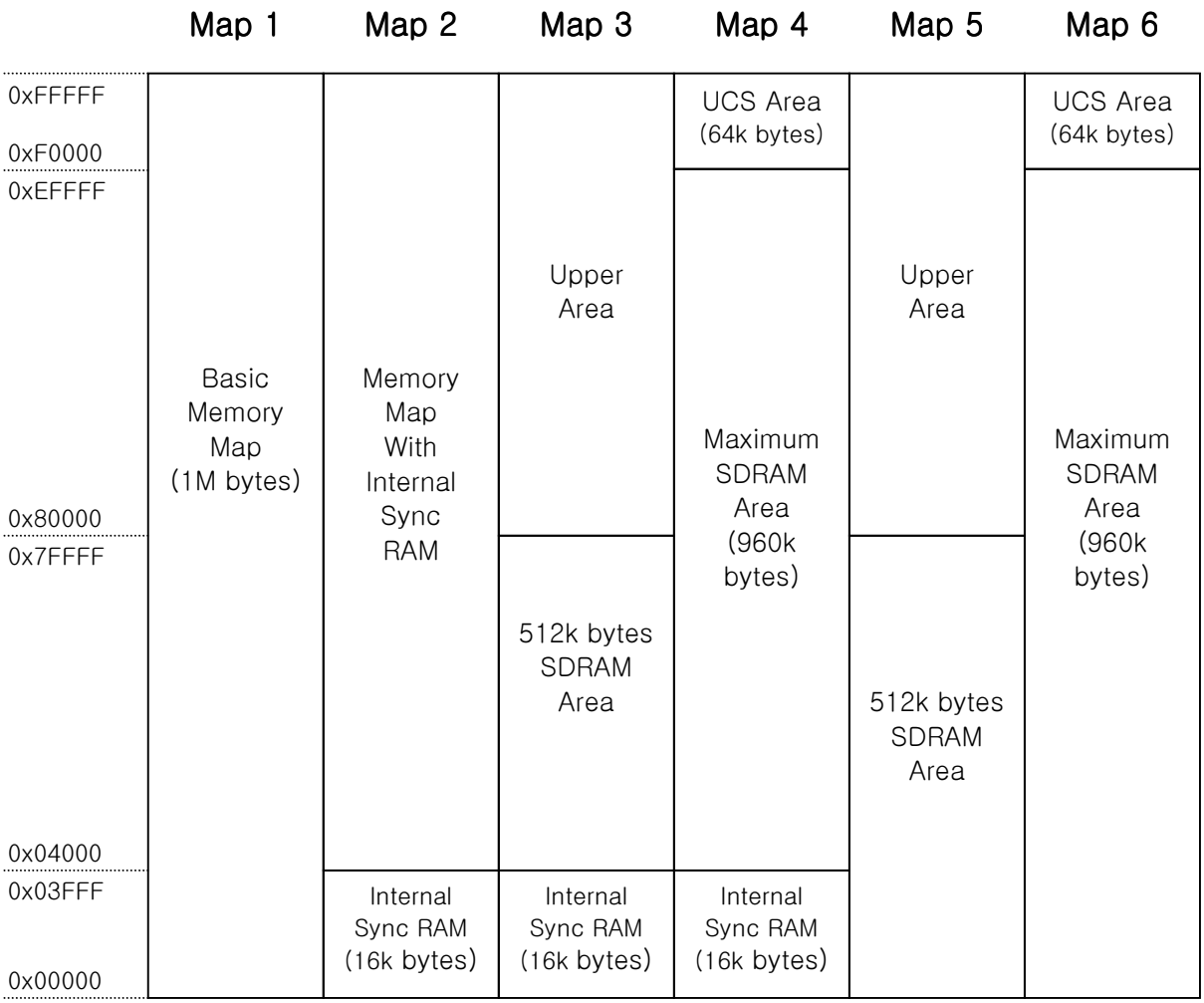


Figure 15 IMS16C Memory Configuration Map

Map 1 in Figure 15 shows IMS16C standard memory map. Internal RAM and SDRAM controller are disabled.

Map 2 in Figure 15 shows IMS16C memory map with 16k bytes on-chip RAM. On-chip RAM is located between 0x0000 and 0x3FFFF and the other address space is configured to UCS, LCS, MCS and PCS area. When internal RAM area and LCS area are overlapped, the priority goes to internal RAM, and LCS_N pin goes high in internal memory area.

Map 3 in Figure 15 shows IMS16C memory map with 16k bytes on-chip RAM and external 512k bytes SDRAM. On-chip RAM is located between 0x0000 and 0x3FFFF and SDRAM address space occupies the range from 0x04000 to 0x7FFFF. Upper 512k bytes share UCS, MCS and PCS spaces.

Map 4 in Figure 15 shows IMS16C memory map with 16k bytes on-chip RAM and external 960k bytes SDRAM. On-chip RAM is located between 0x0000 and 0x3FFFF and SDRAM address space occupies the range from 0x04000 to 0xEFFFF. Upper 64k bytes are used for UCS area. MCS and PCS (memory space shared) are prohibited.

Map 5 in Figure 15 shows IMS16C memory map with external 512k bytes SDRAM. Internal RAM is disabled. Upper 512k bytes share UCS, MCS and PCS spaces.

Map 6 in Figure 15 shows IMS16C memory map with external 960k bytes SDRAM. Internal RAM is disabled. Upper 64k bytes are used for UCS area. MCS and PCS (memory space shared) are prohibited.

15.1. On-chip RAM Control Register (SRCR, Offset E8h)

The sync ram control register has a control bit to enable the internal memory. Power-on reset value is fixed to zero.

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

Reset value : 0000_0000_0000_0000b

Bit	Name	Function
0	SYEN	This bit enables on-chip sync RAM. When set to '0', internal ram is disabled. When set to '1', internal ram is enabled.

Figure 15.1 Sync RAM Control Register

16. HDLC Controller

This chapter describes the HDLC controller on the IMS16C. High-Level Data Link Control, also known as HDLC, is a bit oriented, switched and non-switched protocol. It is a data link control protocol, and falls within layer 2, the Data Link Layer, of the Open Systems Interface (OSI) model. HDLC protocol is used as a data link of most of the current communication systems like ISDN, Frame Relay etc.

The IMS16C provides two HDLC channels. The HDLCs are typically used to transmit and receive frames based on the HDLC format. This format uses flags to determine the start and stop of a frame, and it uses "bit stuffing" to maintain data transparency.

In the transmit direction, the HDLCs add the required error detection bytes (cycle redundancy check, or CRC) at the end of the frame, bit stuff the data as needed, and surround it with start and stop flags.

In the receive direction, the HDLCs search for flags to determine the start and stop for the frame, remove any bit stuffing, and check the CRC bytes.

First-in-first-out buffers (FIFOs) are used in both directions to isolate data requests from the execution unit. Each channel has 64bytes FIFO Memory, which shares 32bytes receive FIFOs and 32bytes transmit FIFOs.

Below list shows some features in IMS16C HDLC controller.

- Synchronous operation
- Start and End of frame pattern generation
- Start and End of frame pattern checking
- Idle pattern generation and detection (all ones)
- Zero insertion and removal for transparent transmission
- Abort pattern generation and checking (seven ones)
- Address insertion and detection by software
- CRC generation and checking (CRC16, CRC-CCITT)
- Byte aligned data
- Q.921, LAPB and LAPD compliant
- Non-Return-To-Zero (NRZ) Data Encoding
- 32 bytes Receive FIFO and 32 bytes Transmit FIFO
- Retransmission is not supported when there is collision in the bus connection mode
- Use both internal and external RX, TX clocks.
- TX and RX interrupt source selectable

Transmit clock speed can be configured using internal tx baud register. It also can be synchronized through the external tx clock source. Receive clock has the same feature of tx clock.
Since the HDLC controller can operate in different clock domains, all control signals pass through two flip-flops to reduce the meta-stability. These flip-flops are clocked with the same clock of the interface that read these signals.

Figure 16 shows HDLC block diagram of IMS16C.

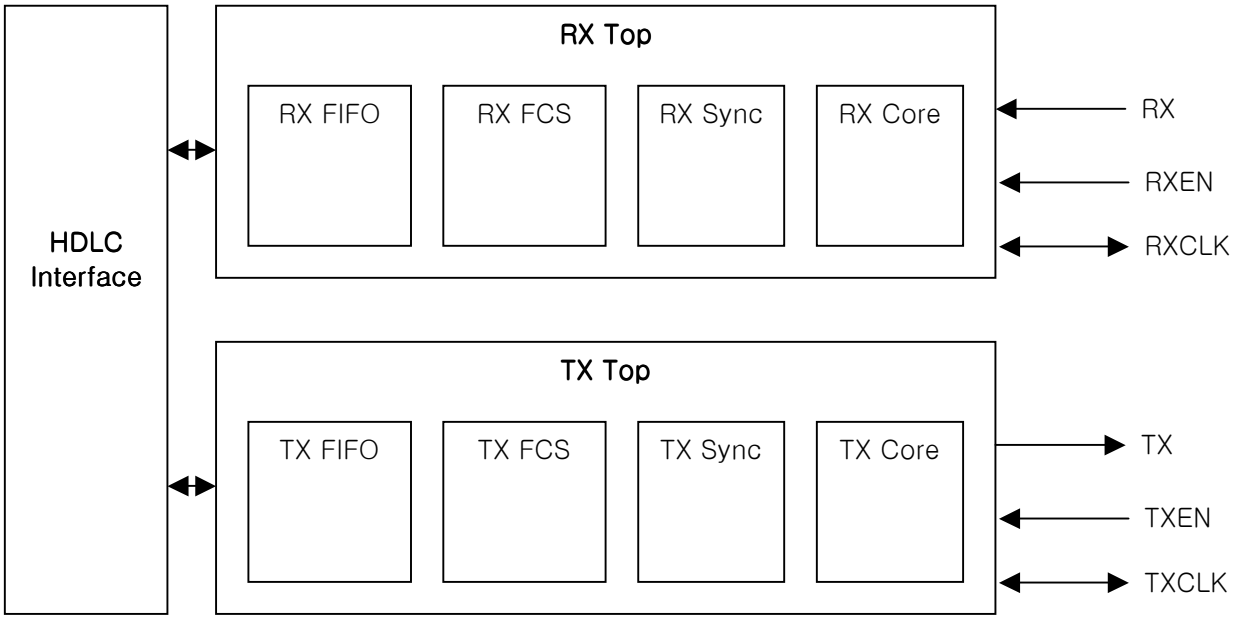


Figure 16 HDLC Block diagram

The IMS16C microcontroller has 10 peripheral registers related to HDLC controller. Figure 16-1 shows HDLC Control Registers.

Offset	Register Mnemonic	Register Name
BEh	HDCTR0	HDLC 0 Control Register
BCh	HDSTR0	HDLC 0 Status Register
BAh	TXBFR0	HDLC 0 Transmit Buffer Register
B8h	RXBFR0	HDLC 0 Receive Buffer Register
B6h	BDREG0	HDLC 0 Baud Clock Setting Register
B4h	HDCTR1	HDLC 1 Control Register
B2h	HDSTR1	HDLC 1 Status Register
B0h	TXBFR1	HDLC 1 Transmit Buffer Register
AEh	RXBFR1	HDLC 1 Receive Buffer Register
ACh	BDREG1	HDLC 1 Buad Clock Setting Register

Figure 16-1 HDLC Control Registers Summary

16.1. HDLC Control Register (HDCTR0, Offset BEh, HDCTR1, Offset B4h)

Figure 16.1 shows HDLC control register.

15	14 13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX INT EN	TX INT SEL	0	RX INT EN	RX INT SEL	0	RX ERR INT EN	TX CLK EN	RX CLK EN	0	0	0	CRC SEL	TX ABORT EN	TX START EN

Reset value : 0000_0000_0000_0000b

Bit	Name	Function
15	TX INT EN	-Transmit Interrupt Enable. When set to '0', Transmit Interrupt is disabled. When set to '1', Transmit Interrupt is enabled.
14-13	TX INT SEL	-Transmit Interrupt Source Select HDCTR[14:13] / Interrupt Source 00 – Tx_half_full (HDSTR[13]) 01 – Tx_almost_full (HDSTR[14]) 10 – Tx_full (HDSTR[12])
11	RX INT EN	-Receive Interrupt Enable When set to '0', Receive Interrupt is disabled. When set to '1', Receive Interrupt is enabled.
10	RX INT SEL	-Receive Interrupt Source Select HDCTR[10] / Interrupt Source 0 – Rx_half_full (HDSTR[5]) 1 – Rx_almost_empty (HDSTR[6])
8	RX ERR INT EN	-Receive Error Interrupt Enable : Rx Frame Error, Rx FCS Error When set to '0', Receive Error Interrupt is disabled. When set to '1', Receive Error Interrupt is enabled.
7	TX CLK EN	-Transmit Internal Clock Source Enable When set to '0', Transmit Internal Clock Source is disabled. External TXCLK is used. (TXCLK Direction: IN) When set to '1', Transmit Internal Clock Source is enabled. Internal TXCLK is used. (TXCLK Direction: OUT)
6	RX CLK EN	-Receive Internal Clock Source Enable When set to '0', Receive Internal Clock Source is disabled. External RXCLK is used. (RXCLK Direction: IN) When set to '1', Receive Internal Clock Source is enabled. Internal RXCLK is used. (RXCLK Direction: OUT)
2	CRC SEL	-CRC Selection When set to '0', CRC-16 is selected. When set to '1', CRC-CCITT is selected.
1	TX ABORT EN	-Transmit Data Abort Enable When set to '0', Transmit data is not aborted. When set to '1', Transmit data is aborted.
0	TX START EN	-Transmit Start Enable When set to '0', HDLC data transmission is not started. When set to '1', HDLC data transmission is started.

Figure 16.1 HDLC Control Register

16.2. HDLC Status Register (HDSTR0, Offset BCh, HDSTR1, Offset B2h)

Figure 16.2 shows HDLC status register. Transmit FIFO depth is 32. Receive FIFO depth is 32.

15	14	13	12	11	10	9	8
TX EMPTY	TX ALMOST FULL	TX HALF FULL	TX FULL	TX ABORT	TX TRANSMITTING	0	RX OVERRUN

7	6	5	4	3	2	1	0
RX EMPTY	RX ALMOST EMPTY	RX HALF FULL	RX FULL	RX ABORT	RX FRAME ERROR	RX FCS ERROR	RX EOF

Reset value : 0000_0000_0000_0000b

Bit	Name	Function
15	TX EMPTY	-Transmit FIFO Empty. When set to '0', Transmit FIFO is not empty. When set to '1', Transmit FIFO is empty.
14	TX ALMOST FULL	-Transmit FIFO Almost Full When set to '0', Transmit FIFO is not almost full. When set to '1', Transmit FIFO is almost full. (FIFO with valid data is 27)
13	TX HALF FULL	-Transmit FIFO Half Full When set to '0', Transmit FIFO is not half full. When set to '1', Transmit FIFO is half full. (FIFO with valid data is 16)
12	TX FULL	-Transmit FIFO Full. When set to '0', Transmit FIFO is not full. When set to '1', Transmit FIFO is full.
11	TX ABORT	-Transmit Data Abort When set to '0', Transmit data is not aborted. When set to '1', Transmit data is aborted.
10	TX TRANSMITTING	-Transmit Data is being Transmitting When set to '0', Data Transmission is not active. When set to '1', Data Transmission is active.
8	RX OVERRUN	-Receive Data FIFO Push Error When set to '0', Receive FIFO Data Push is not overrun. When set to '1', Receive FIFO Data Push is overrun.
7	RX EMPTY	-Receive FIFO Empty When set to '0', Receive FIFO is not empty. When set to '1', Receive FIFO is empty.
6	RX ALMOST EMPTY	-Receive FIFO Almost Empty When set to '0', Receive FIFO is not almost empty. When set to '1', Receive FIFO is almost empty. (FIFO with valid data is 5)
5	RX HALF FULL	-Receive FIFO Half Full When set to '0', Receive FIFO is not half full. When set to '1', Receive FIFO is half full. (FIFO with valid data is 16)
4	RX FULL	-Receive FIFO Full When set to '0', Receive FIFO is not full. When set to '1', Receive FIFO is full.
3	RX ABORT	-Receive Data Abort When set to '0', Receive data is not aborted. When set to '1', Receive data is aborted.
2	RX FRAME ERROR	-Receive Frame Error When set to '0', Receive Frame Error is not occurred. When set to '1', Receive Frame Error is occurred.
1	RX FCS ERROR	-Receive Frame Check Sequence Error When set to '0', Receive FCS Error is not occurred.

		When set to '1', Receive FCS Error is occurred.
0	RX EOF	-Receive End of Frame When set to '0', Receiving Current Frame is not finished. When set to '1', Receiving Current Frame is finished.

Figure 16.2 HDLC Status Register

16.3. HDLC Transmit Data Buffer Register (TXBFR0, Offset BAh, TXBFR1, Offset B0h)

Figure 16.3 shows HDLC Transmit Data Buffer Register.

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

Reset value : Not Defined

Bit	Name	Function
7 - 0	TX Data Buffer Register	This Register is used for Transmit Data Buffer. Low 8 bits are used for data transmission. High 8 bits are not used.

Figure 16.3 HDLC Transmit Data Buffer Register

16.4. HDLC Receive Data Buffer Register (RXBFR0, Offset B8h, RXBFR1, Offset AEh)

Figure 16.4 shows HDLC Receive Data Buffer Register.

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

Reset value : Not Defined

Bit	Name	Function
7 - 0	RX Data Buffer Register	This Register is used for Receive Data Buffer. Low 8 bits are used for data receiving. High 8 bits are not used.

Figure 16.4 HDLC Receive Data Buffer Register

16.5. HDLC Baud Rate Divisor Register (BDREG0, Offset B6h, BDREG1, Offset ACh)

This register specifies a clock divisor for the generations of the internal HDLC receive and transmit clock.

A general formula for the baud rate divisor is:

$BAUDDIV_{(decimal)}$ = (Internal Core Clock Frequency / (32 *Baud Rate)) – 1

The maximum baud rate is 1/32 of the internal Core clock and is achieved by setting BAUDDIV = 0000h.

For a 40MHz internal core clock frequency, a baud rate of 9600 bps can be achieved by

40000000/(32*9600) = 40000000/307200 = 130.208
130 – 1 = 129

BAUDDIV=129 (81h). A 1% error applies.

Figure 16.5 shows HDLC baud rate divisor register.

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

Reset value : 0000_0000_0000_0100b

Bit	Name	Function
15-0	Baud Divisor Register	This register generates internal RXCLK and TXCLK speed. Note that Internal RXCLK and TXCLK have the same speed, which decided by setting Baud Divisor Register.

Figure 16.5 HDLC Baud Rate Divisor Register

17. Byte/Word Mode Control Unit

IMS16C has an input pin, "BWMODE", which converts IMS16C external AD bus configuration either 80c186 word mode or 80c188 byte mode.

If bwmode pin is set to low, IMS16C operates 80c188 mode. External 16bit AD bus is divided to two buses. High 8bit of AD bus is configured to the same as A[15:8]. Low 8bit of AD bus is configured to 8bit data bus. When external access is occurred, external A bus is incremented or decremented by one in normal read or write operations. But, internal IMS16C core operation is 16bit word operation. Only external data access is fixed to 8bit byte operation.

If bwmode pin is set to high, IMS16C operates 80c186 mode. External 16bit AD bus is used for address or data bus. IMS16C accesses external data in 16bit word mode. Internal core operates 16bit word operation.

Figure 17 shows external AD bus timing diagram when bwmode pin is low(80c188 byte mode).

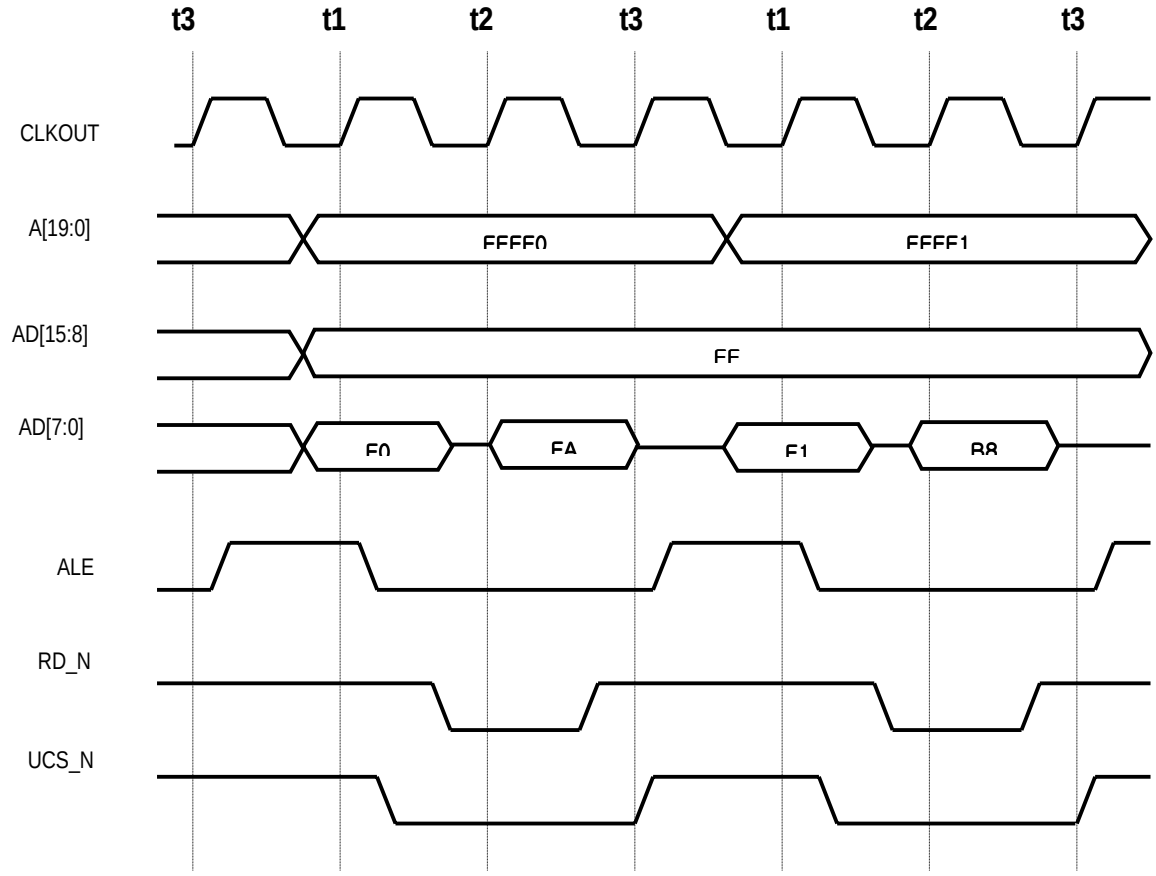


Figure 17 IMS16C 80c188 Byte Mode AD Bus Timing

APPENDIX

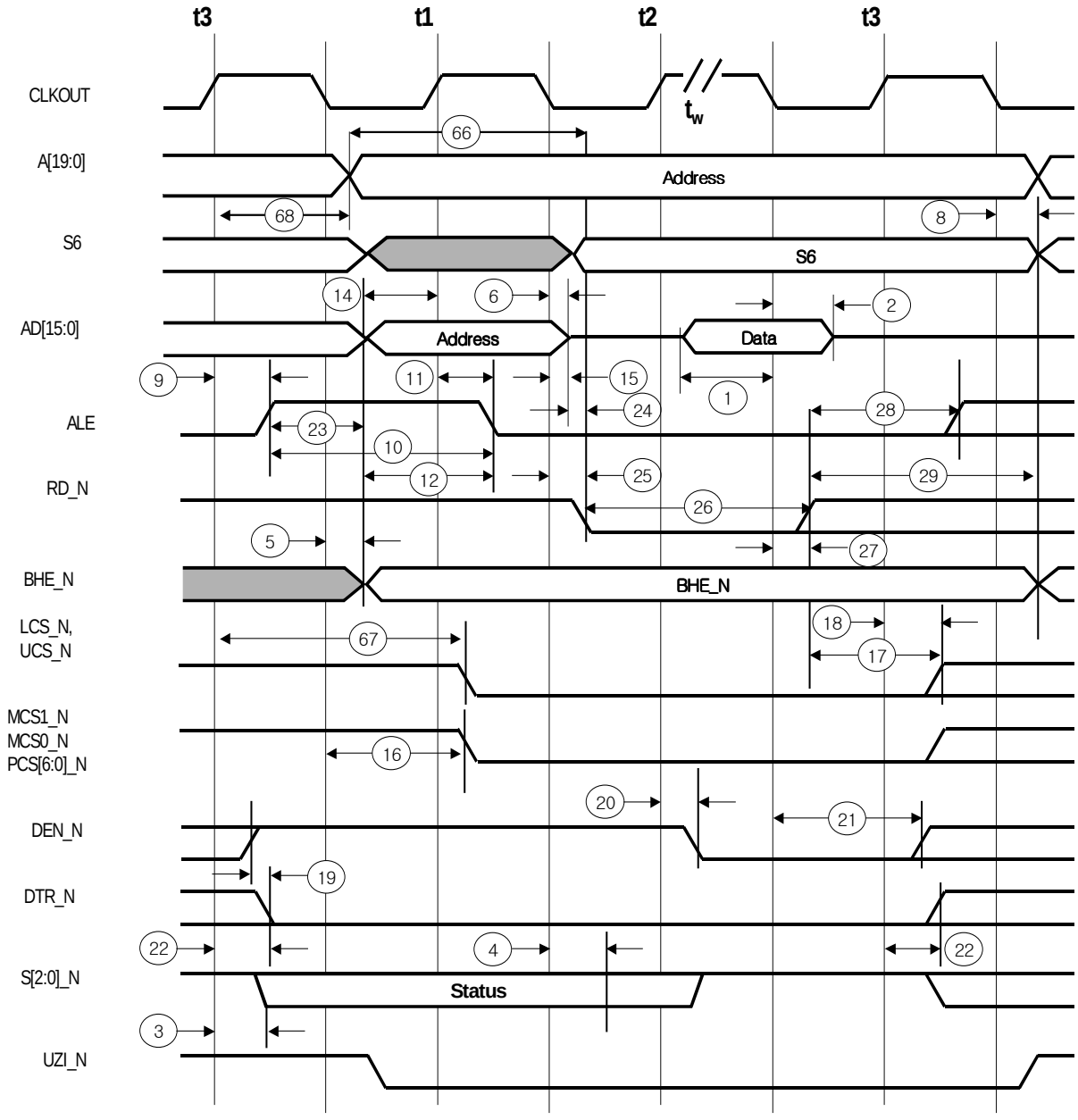
A. Electrical Characteristics

A.1 DC Electrical Characteristics (TA=0~70°C, Vcc=3.3V±5%, Vss=0V)

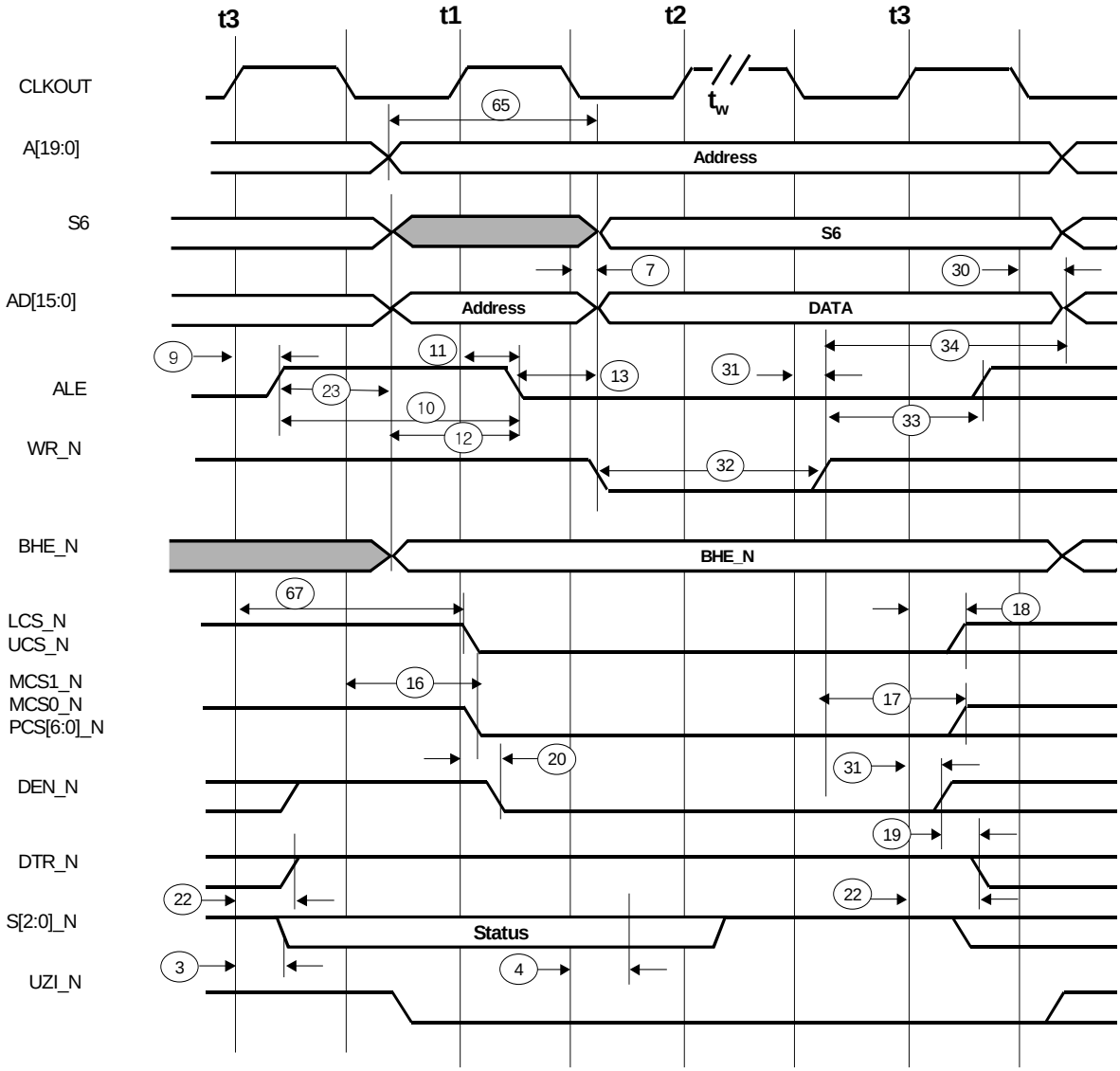
Parameters	Minimum	Maximum	Unit	NOTE
V_{IL}, (Input Low voltage CMOS input)	-0.5	0.3 x Vcc	V	Guaranteed Input Low Voltage
V_{IH}, (Input High voltage CMOS Input)	0.7 x Vcc	Vcc + 0.3	V	Guaranteed Input High Voltage
V_{OL} (Output Low voltage)		0.8	V	For I_{OL}=2mA/4mA/ 8mA/12mA
V_{OH} (Output High voltage)	2.4		V	For I_{OH}=-2mA/-4mA/ -8mA/-12mA
I_{In} (Input Leakage current)	-1	1	μ A	V_{IN}=Vcc or Vss
I_{OZ} (3-state Output Leakage current)	-1	1	μ A	V_{OUT}=Vcc or Vss
Junction temperature	0	100	^{o}C	-
Main Clock Frequency	-	90	Mhz	-

A.2 AC Electrical Characteristics

A.2.1 Read Cycle Waveforms



A.2.2 Write Cycle Waveforms



A.2.3 Read/Write Cycle Timing Table (60Mhz)

General Timing Response

No	Symbol	Description	Min Time(ns)	Max Time(ns)
1	Tdvcl	Data in Setup	3	
2	Tcldx	Data in Hold	2	
3	Tchsv	Status Active Delay	0	3
4	Tclsh	Status Inactive Delay	0	11
5	Tclav	AD Address Valid Delay and BHE	0	8
6	Tclax	Address Hold	0	7
7	Tcldv	Data Valid Delay	0	7
8	Tchdx	Status Hold Time	0	7
9	Tchlh	ALE Active Delay	0	3
10	Tlhll	ALE Width	17	
11	Tchll	ALE Inactive Delay		3
12	Tavll	AD Address Valid to ALE Low	4	
13	Tllax	AD Address Hold from ALE inactive	13	
14	Tavch	AD Address Valid to Clock High	2	
15	Tclaz	AD Address Float Delay	1	
16	Tclcsv	MCS_n/PCS_n Active Delay	18	
17	Tcxcsx	MCS_n/PCS_n Hold from Command Inactive	4	
18	Tchcsx	MCS_n/PCS_n Inactive Delay	0	3
19	Tdxdl	DEN_N Inactive to DT/R_n Low	0	
20	Tcvctv	Contol Active Delay	0	3
21	Tcvdex	DEN_N Active Delay	0	11
22	Tchctv	Control Active Delay	0	3
23	Tlhav	ALE High to Address Valid	14	

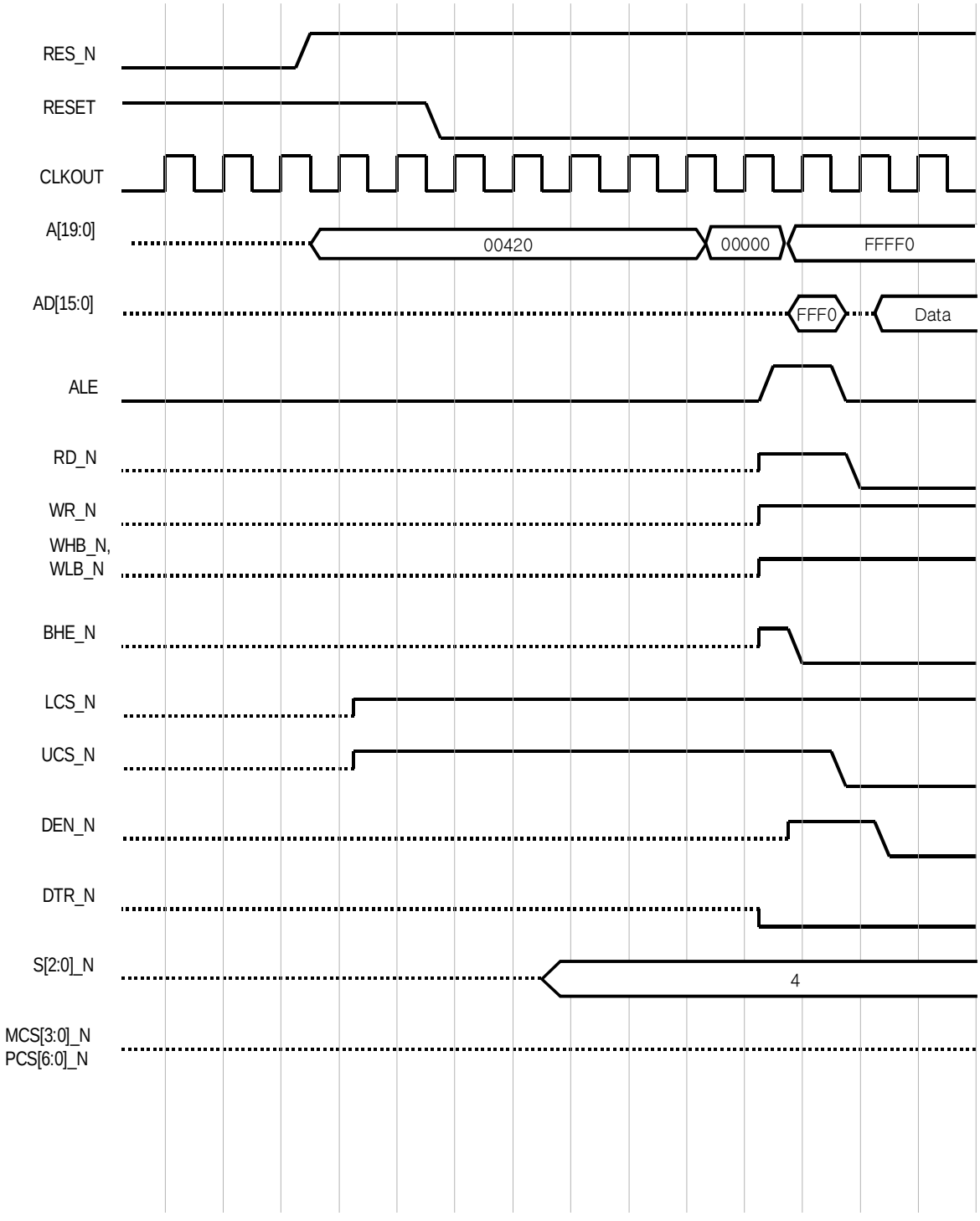
Read Cycle Timing Response

No	Symbol	Description	Min Time(ns)	Max Time (ns)
24	Tazrl	AD Address Float to RD_n Active	0	
25	Tclrl	RD_n Active Delay	0	7
26	Trlrh	RD_n Pulse Width	16	
27	Tclrh	RD_n Inactive Delay	0	7
28	Trhlh	RD_n Inactive to ALE High	4	
29	Trhav	RD_n Inactive to AD Address Active	16	
59	Trhdx	RD_n High to Data Hold on AD Bus	0	
66	Tavrl	A Address Valid to RD_n Low	17	
67	Tchcsv	CLKOUTA High to LCS_n/UCS_n Valid	0	19
68	Tchav	CLKOUTA High to A Address Valid	0	16

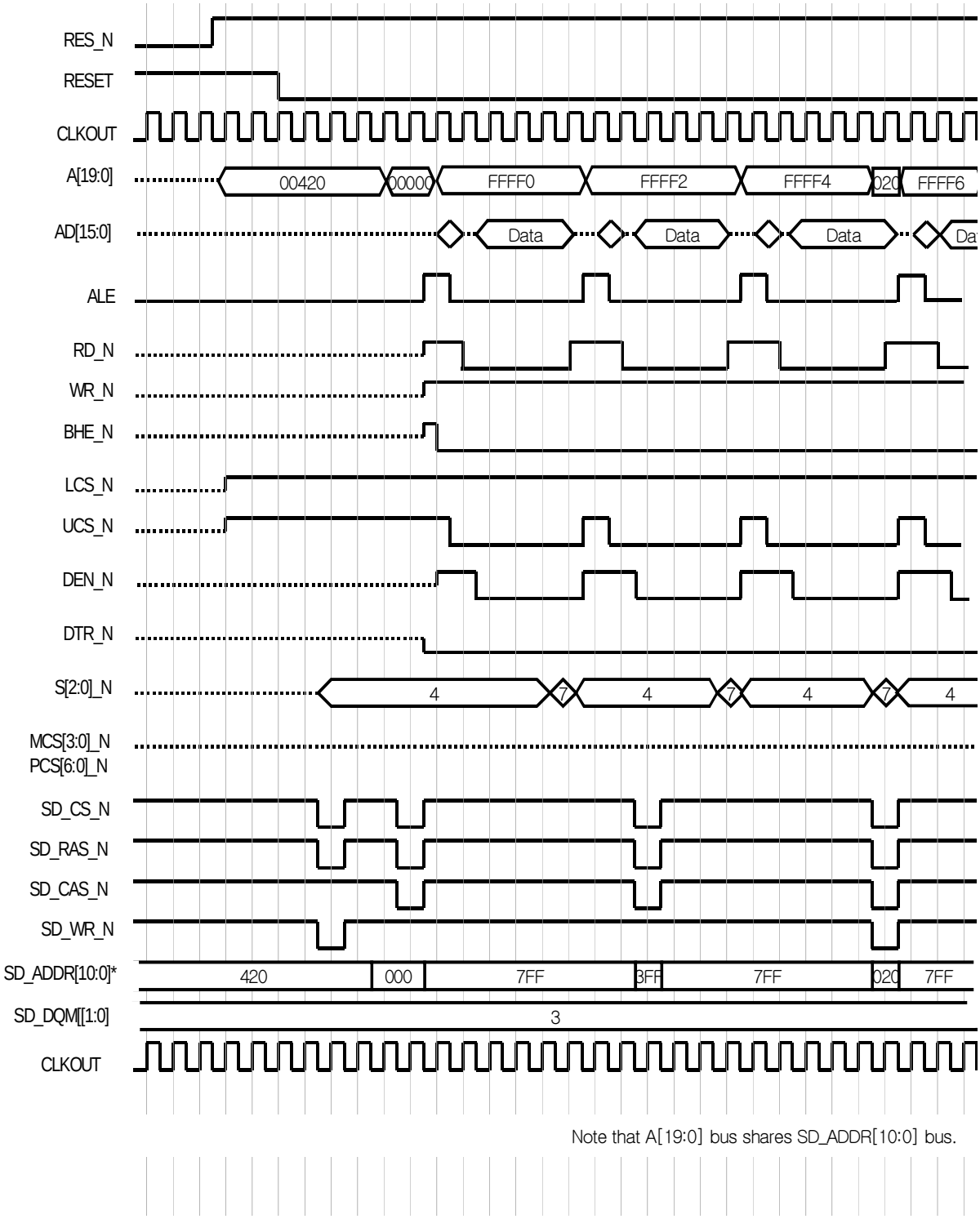
Write Cycle Timing Response

No	Symbol	Description	Min Time(ns)	Max Time(ns)
30	Tcldox	Data Hold Time	7	
31	Tcvctx	Control Inactive Delay	0	7
32	Twlwh	WR_n Pulse Width	16	
33	Twhlh	WR_n Inactive to ALE High	4	
34	Twhdx	Data Hold after WR_n	16	
65	Tavwl	A Address Valid to WR_n Low	17	
68	Tchav	CLOCKA High to A Address Valid	0	16

A.2.4 Reset Waveforms

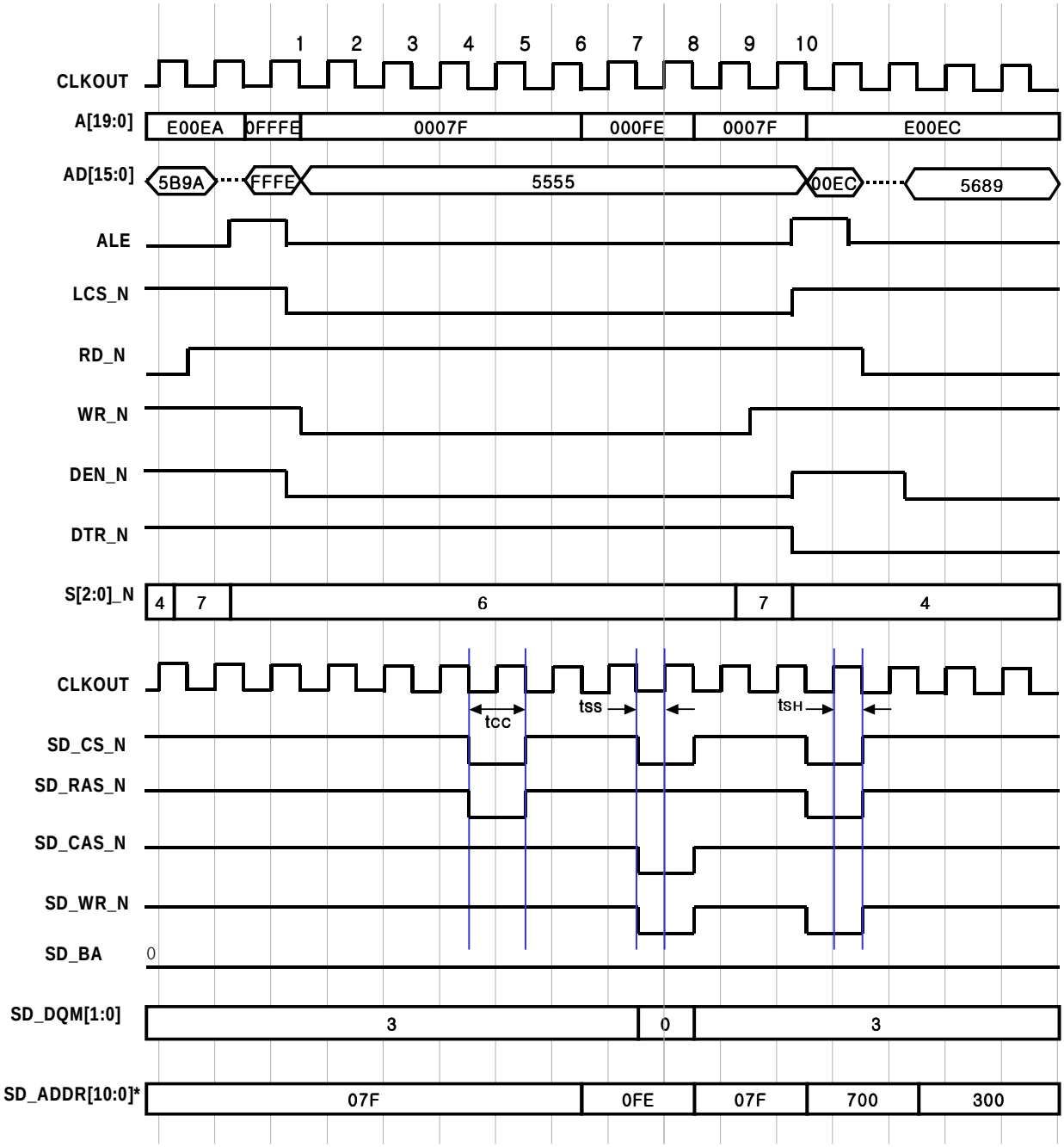


A.2.5 SDRAM Initial Waveforms



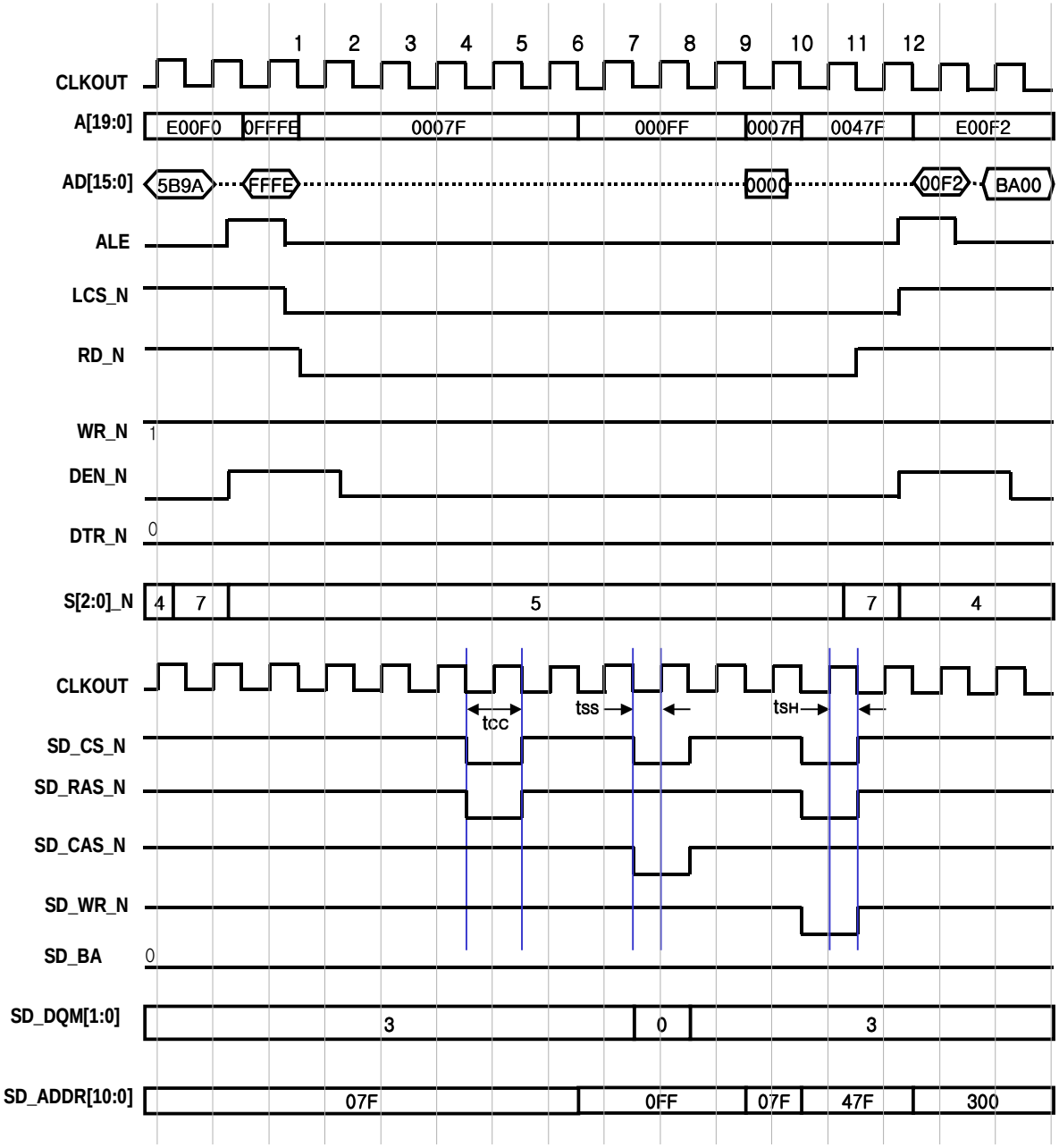
Note that A[19:0] bus shares SD_ADDR[10:0] bus.

A.2.6 SDRAM Write Waveforms



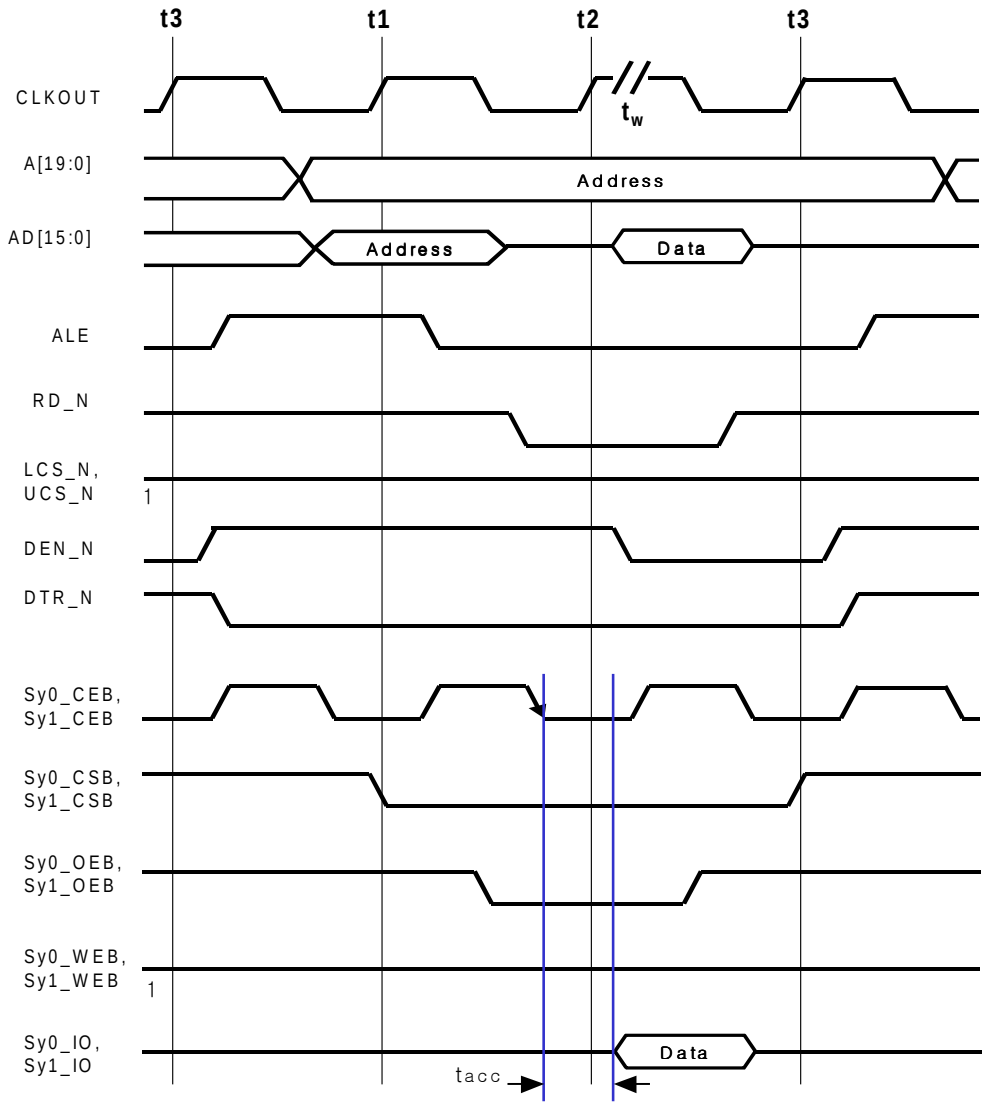
Symbol	Description	Min Time (ns)	Max Time (ns)
tCC	SDRAM Clock Cycle Time	16.4	
tSS	SDRAM Control Setup Time	2	
tSH	SDRAM Control Hold Time	1	

A.2.7 SDRAM Read Waveforms



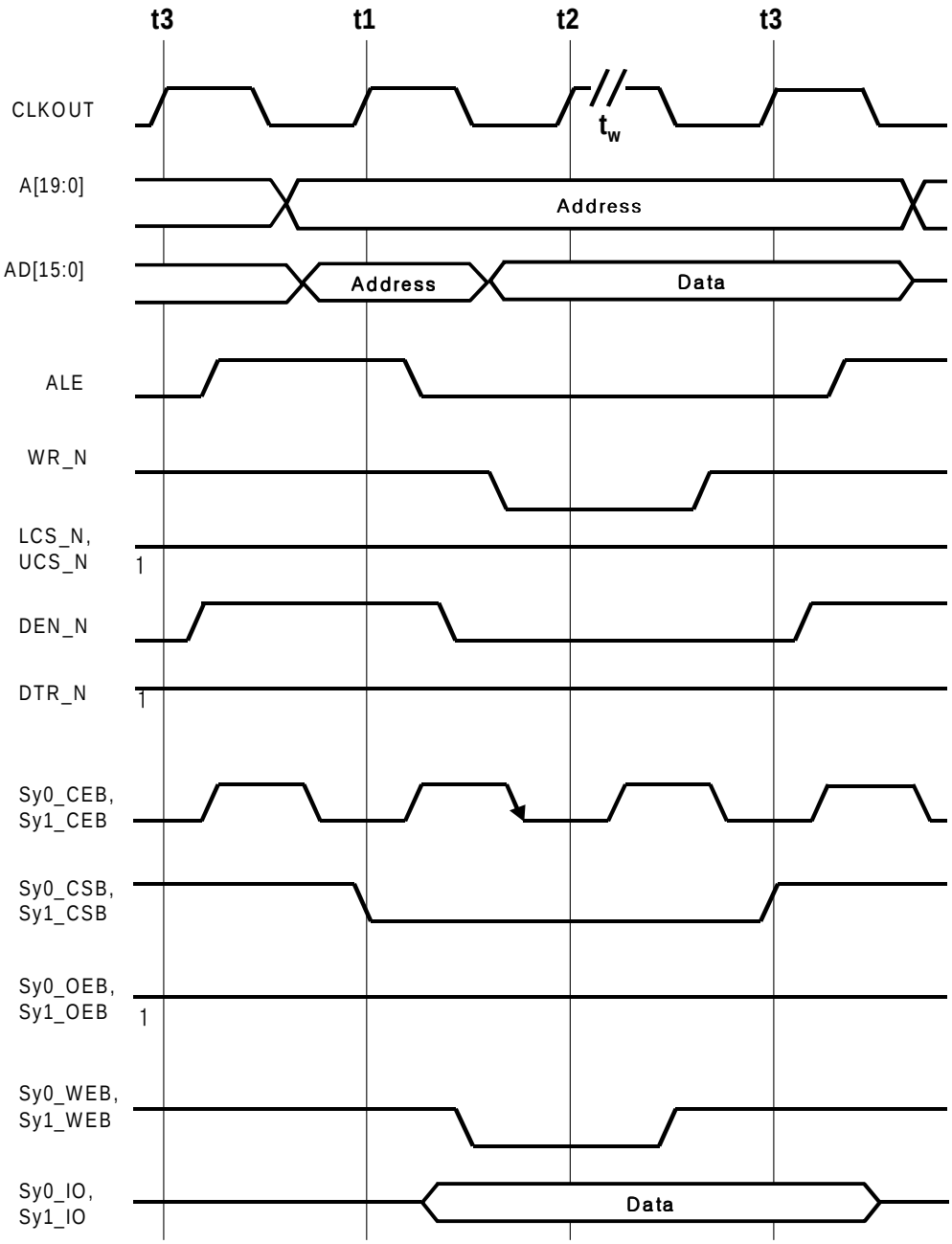
Symbol	Description	Min Time (ns)	Max Time (ns)
tCC	SDRAM Clock Cycle Time	11	
tSS	SDRAM Control Setup Time	2	
tSH	SDRAM Control Hold Time	1	

A.2.8 On-Chip Single Port Sync RAM Read Cycle Waveforms

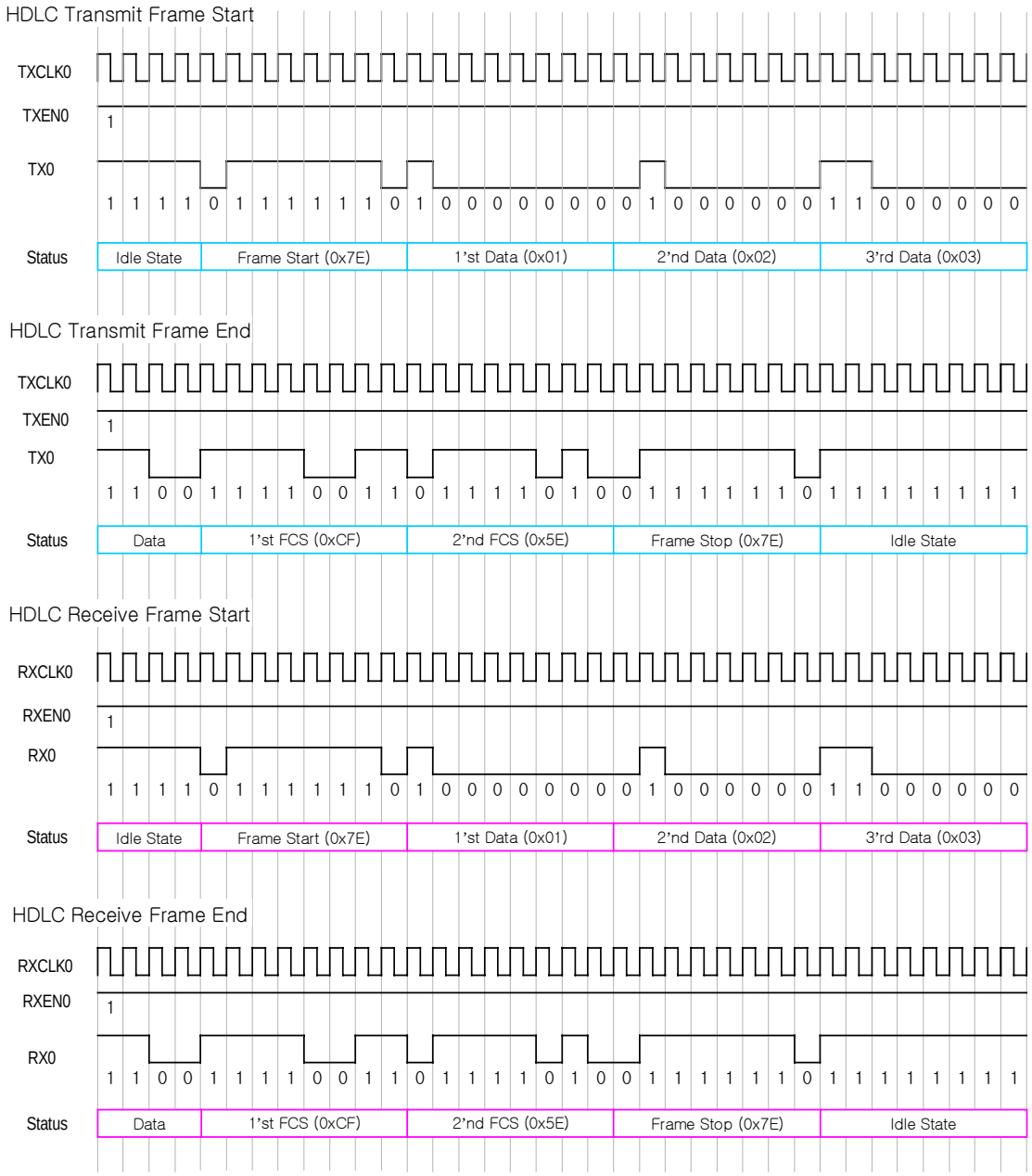


Symbol	Description	Min Time (ns)	Max Time (ns)
tacc	Internal Sync RAM Access Time	5.8	

A.2.9 On-Chip Single Port Sync RAM Write Cycle Waveforms



A.2.10 HDLC Start and End Frame Timing Diagram



B. Package

The Plastic Quad Flatpack (PQFP) is a high-density, low-cost package for high leadcount applications. It uses a smaller lead-to-lead spacing than the PLCC and has gull-wing leads which permit better inspection of leads and solder joints. PQFPs are assembled with the latest technology of low stress die-attach material and molding compound.

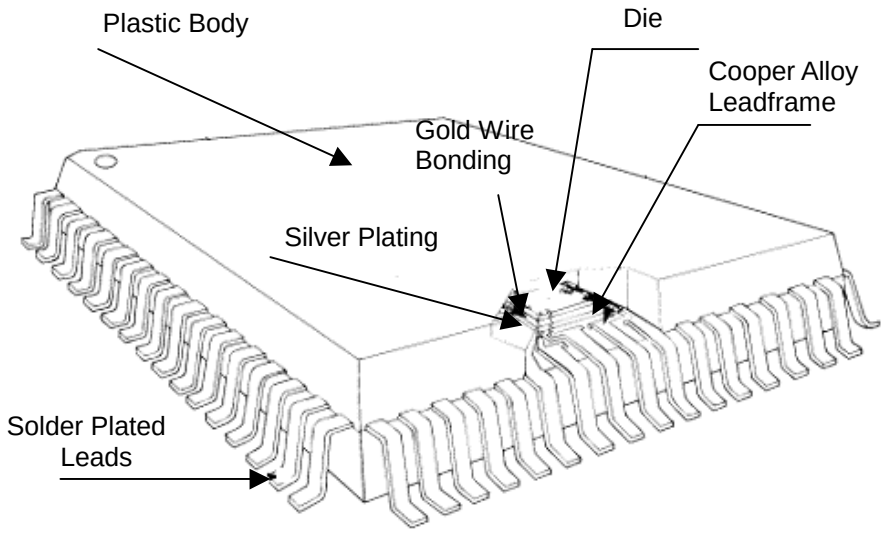


Figure B.1 Packaging

128-QFP-1420 (Dimensions in Millimeters)

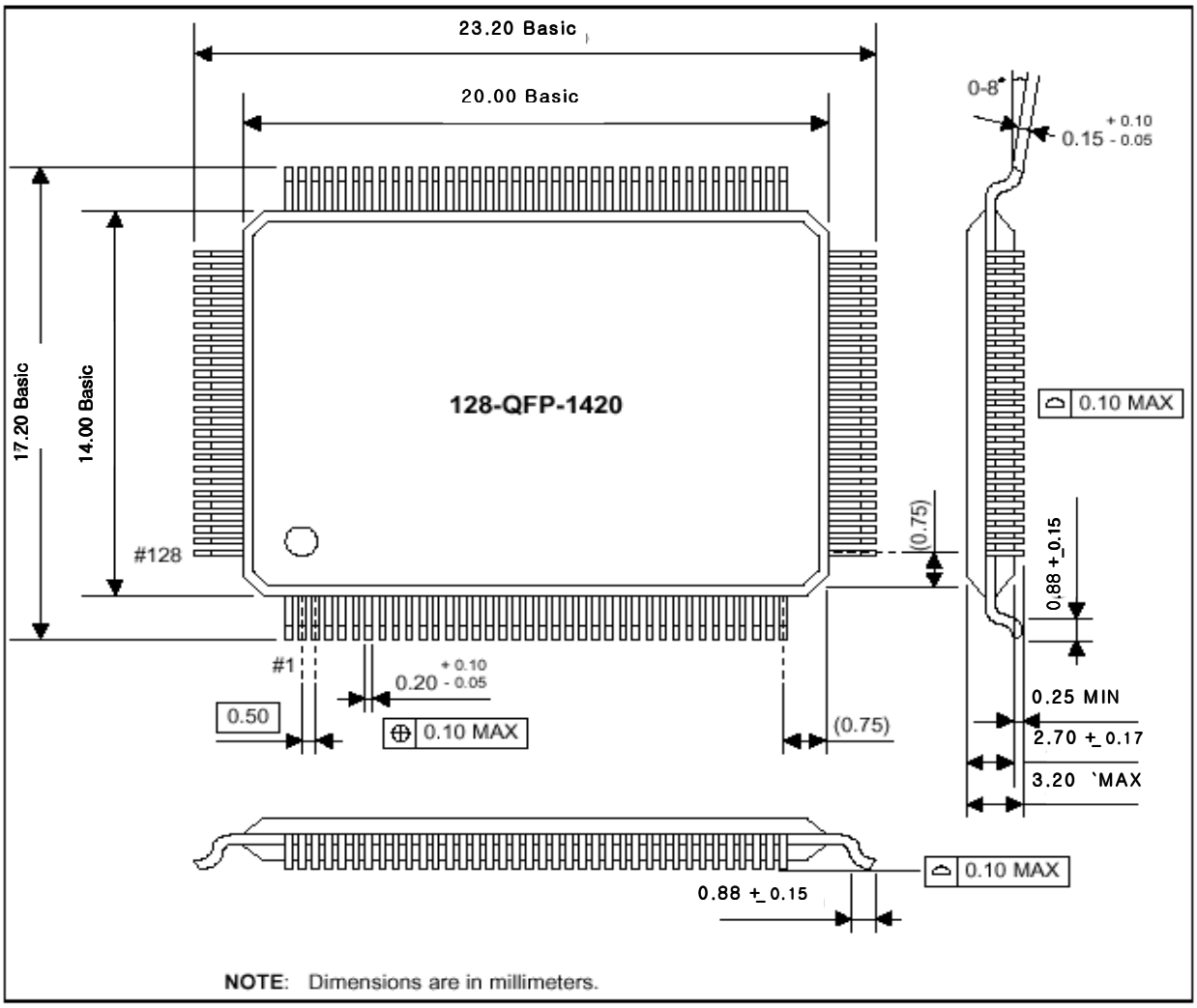
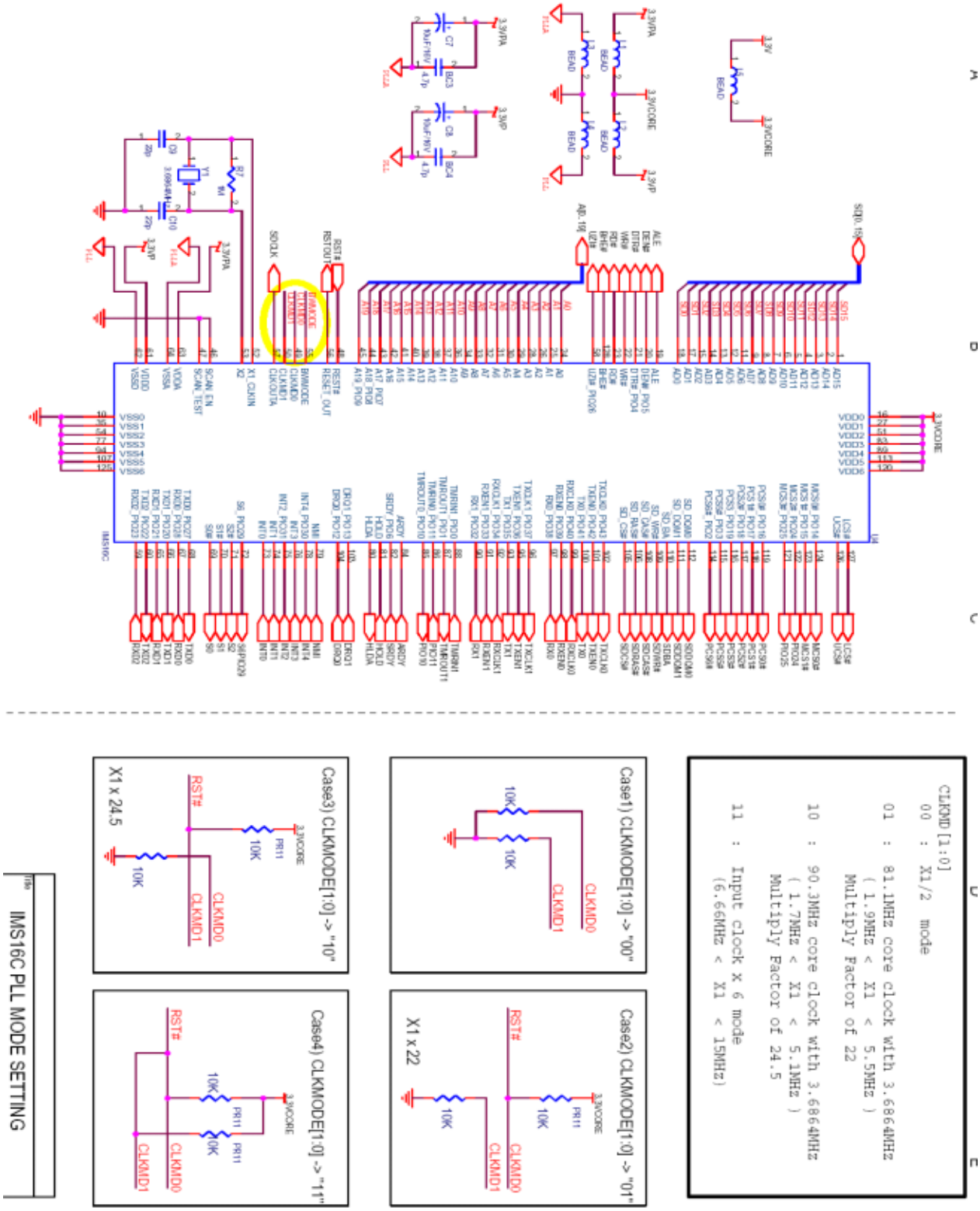


Figure B.2 Physical Dimension

NOTE
1. DIMENSIONS DO NOT INCLUDE MOLD PROTRUSION AND
   DAMBAR PROTRUSION.
   ALLOWABLE MOLD PROTRUSION IS 0.254mm.
   ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.08mm
   TOTAL AT MAXIMUM MATERIAL CONDITION.
2. FORMED LEAD SHALL BE PLANAR WITH RESPECT ANOTHER
   WITHIN 0.10mm
3. CONTROLLING DIMENSION : MILLIMETER.
   THIS OUTLINE CONFIRMS TO JEDEC PUBLICATION 95
   RESISTRATION MO-112.

C. Typical Applications (IMS16C evaluation B'd schematic)



Note: CLKMD1 has multiply factor of **22**, CLKMD2 has multiply factor of **24.5**.

C.1 PLL Frequency Synthesizer Circuit

The IMS16C has a PLL frequency synthesizer on chip to provide frequency multiplication capabilities. The PLL circuit consists of a pre-divider, a PFD(Phase/Frequency Detector), a charge pump, a VCO (Voltage Controlled Oscillator), a post-scalar and a external loop filter.

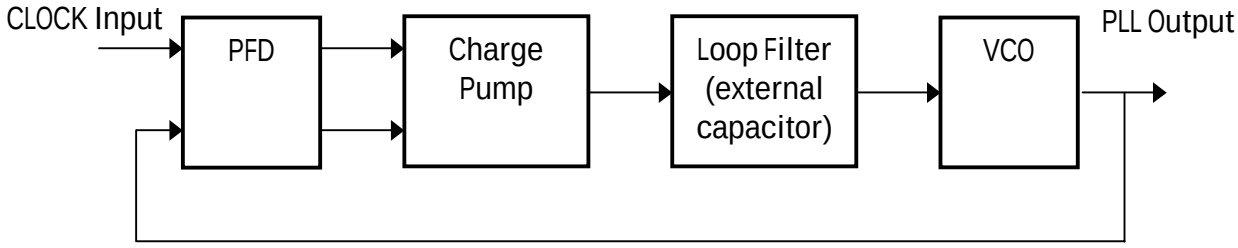


Figure C.2 Main Component of PLL Frequency Synthesizer

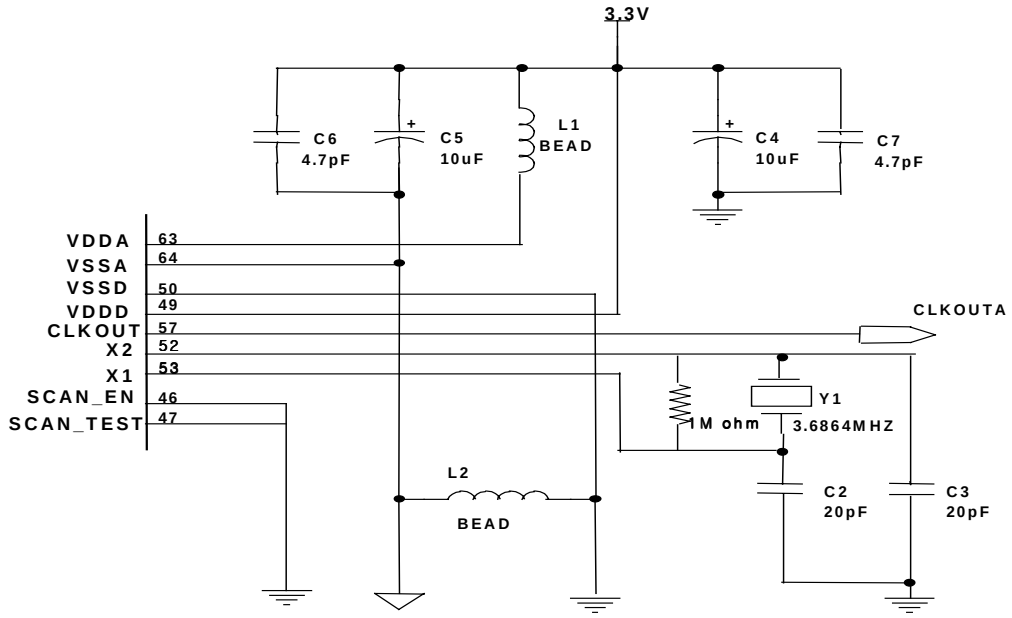


Figure C.3 Example circuit for PLL Frequency Synthesizer

Phase tolerance and jitter of the PLL circuit are independent of the input and PLL frequencies, those are affected by the noise frequency in the power line. Jitter increases when the noise level increases.
For signal compatibility with the PLL circuit, a CMOS-level crystal or oscillator is recommended.
The capacitor C2 and C3 are varied with the characteristics of crystal Y1. C4 and C5 de-coupling capacitors are 20uF electrolytic capacitors. 10uF capacitors would work well also. C6 and C7 de-coupling capacitors are 4.7pF ceramic capacitors.

D. Instruction Clock Cycles

Note : EA : Effective Address (for calculation method, refer to D.11)

D.1 Data Transfer Instructions

INSTRUCTIONS	Function	Clocks	
MOV	Register to register/memory	2	5 + EA
	Register/memory to register	2	7 + EA
	Immediate to register/memory	5 + EA	6 + EA
	Immediate to register	2	
	Memory to accumulator	8	
	Accumulator to memory	8	
	Register/memory to segment register	2	7 + EA
	Segment register to register/memory	2	5 + EA
PUSH	Memory	8 + EA	
	Register	6	
	Segment register	6	
	Immediate	7	
POP	Memory	12 + EA	
	Register	8	
	Segment register	8	
PUSHA	Push all	27	
POPA	Pop all	30	
XCHG	Register/memory with register	3	10 + EA
XLAT	Translate byte to AL	9	
	Register with accumulator	3	
IN	Fixed port	8	
	Variable port	8	
OUT	Fixed port	5	
	Variable port	5	
LEA	Load EA to register	2 + EA	
LDS	Load pointer to DS	10	
LES	Load pointer to ES	10 + EA	
ENTER	L=0	12	
	L=1	17	
	L>1	17 + 9(L-1)	
LEAVE	Leave procedure	8	
LAHF	Load AH with flags	2	
SAHF	Store AH into flags	3	
PUSHF	Push flags	8	
POPF	Pop flags	8	

D.2 Arithmetic Instructions

INSTRUCTIONS	Function	Clocks	
ADD	Register/memory with register to either	3	8 + EA
	Immediate byte to register/memory	3	8 + EA
	Immediate word to register/memory	4	9 + EA
	Immediate to accumulator	2	

ADC	Register/memory with register to either	3	8 + EA
	Immediate byte to register/memory	3	8 + EA
	Immediate word to register/memory	4	9 + EA
	Immediate to accumulator	2	3
INC	Register/memory	2	8 + EA
	Register	2	
SUB	Register/memory and register to either	3	8 + EA
	Immediate byte to register/memory	3	8 + EA
	Immediate word to register/memory	4	9 + EA
	Immediate from accumulator	2	
SBB	Register/memory and register to either	3	8 + EA
	Immediate byte to register/memory	3	8 + EA
	Immediate word to register/memory	4	9 + EA
	Immediate from accumulator	2	
DEC	Register/memory	2	8 + EA
	Register	2	
NEG	Register/memory	3	10 + EA
CMP	Register/memory with register to either	3	8 + EA
	Immediate byte to register/memory	3	8 + EA
	Immediate word to register/memory	4	9 + EA
	Immediate with accumulator	2	
MUL	Register-byte	19	
	Register-word	19	
	Memory-byte	21 + EA	
	Memory-word	21 + EA	
IMUL	Register-byte	24	
	Register-word	24	
	Memory-byte	26 + EA	
	Memory-word	26 + EA	
	Register-immediate8/16	20	
	Memory-immediate8/16	22 + EA	
DIV	Register-byte	29	
	Register-word	38	
	Memory-byte	35	
	Memory-word	44	
IDIV	Register-byte	52	
	Register-word	61	
	Memory-byte	58	
	Memory-word	67	
AAS	ASCII adjust for subtraction	5	
DAS	Decimal adjust for subtraction	3	
AAA	ASCII adjust for addition	5	
DAA	Decimal adjust for addition	3	
AAD	ASCII adjust for division	15	
AAM	ASCII adjust for multiplication	19	
CBW	Convert byte to word	2	
CWD	Convert word to double word	2	

D.3 Logical Instructions

INSTRUCTIONS	Function	Clocks	
NOT	Register/memory	3	8 + EA
AND	Register/memory and register to either	3	8 + EA
	Immediate byte to register/memory	3	8 + EA
	Immediate word to register/memory	4	9 + EA
	Immediate to accumulator	2	
OR	Register/memory and register to either	3	8 + EA
	Immediate byte to register/memory	3	8 + EA
	Immediate word to register/memory	4	9 + EA
	Immediate to accumulator	2	
XOR	Register/memory and register to either	3	8 + EA
	Immediate byte to register/memory	3	8 + EA
	Immediate word to register/memory	4	9 + EA
	Immediate to accumulator	2	
TEST	Register/memory and register	3	8 + EA
	Immediate byte to register/memory	3	8 + EA
	Immediate word to register/memory	4	9 + EA
	Immediate to accumulator	2	
Shifts/Rotates	Register/memory by 1	2	8 + EA
	Register/memory by CL	5 + n	11+n + EA
	Register/memory by Count	5 + n	11+n + EA

D.4 String Manipulation Instructions

INSTRUCTIONS	Function	Clocks	
MOVS	Move byte/word	11	
INS	IN byte/word from DX port	11	
OUTS	OUT byte/word to DX port	11	
CMPS	Compare byte/word	12	
SCAS	Scan byte/word	9	
LODS	Load byte/word to AL/AX	8	
STOS	Store byte/word from AL/AX	6	
Repeated by count in CX			
MOVS	Move string	7 + 6n	
INS	IN string	7 + 6n	
OUTS	OUT string	7 + 6n	
CMPS	Compare string	8 + 12n	
SCAS	Scan string	8 + 9n	
LODS	Load string	4 + 6n	
STOS	Store string	9 + 3n	

D.5 Conditional Transfer Instructions

INSTRUCTIONS	Function	Clocks	
JE/JZ	Equal/zero	2	8
JL/JNGE	Less/not greater or equal	2	8
JLE/JNG	Less or equal/not greater	2	8
JC/JB/JNAE	Carry/below/not above or equal	2	8

JBE/JNA	Below or equal/not above	2	8
JP/JPE	Parity/ parity even	2	8
JO	Overflow	2	8
JS	Sign	2	8
JNE/JNZ	Not equal/not zero	2	8
JNL/JGE	Not less/greater or equal	2	8
JNLE/JG	Not less or equal/greater	2	8
JNC/JNB/JAE	Not carry/not below/above or equal	2	8
JNBE/JA	Not below or equal/above	2	8
JNP/JPO	Not parity/parity odd	2	8
JNO	Not overflow	2	8
JNS	Not sign	2	8

D.6 Unconditional Transfer Instructions

INSTRUCTIONS	Function	Clocks	
CALL	Direct within segment	12	
	Register/memory indirect within segment	15	18 + EA
	Direct inter-segment	17	
	Indirect inter-segment	24 + EA	
RET	Within segment	16	
	Within segment adding immediate to SP	16	
	Inter-segment	18	
	Inter-segment adding immediate to SP	18	
JMP	Short/long	8	
	Direct within segment	9	
	Register/memory indirect within segment	8	13 + EA
	Indirect inter-segment	16 + EA	
	Direct inter-segment	10	

D.7 Iteration Control Instructions

INSTRUCTIONS	Function	Clocks	
LOOP	Loop CX times	4	10
LOOPZ/LOOPE	Loop while zero/equal	4	10
LOOPNZ/LOOPNE	Loop while not zero/equal	4	10
JCXZ	Jump if CX is zero	4	10

D.8 Interrupt Instructions

INSTRUCTIONS	Function	Clocks	
INT	Type specified	30	
	Type 3	30	
INTO	Interrupt on overflow	30	
BOUND	Detect value out of range	13 + EA	
IRET	Interrupt return	19	

D.9 Processor Control Instructions

INSTRUCTIONS	Function	Clocks	
CLC	Clear carry	2	
CMC	Complement carry	2	
STC	Set carry	2	
CLD	Clear direction	2	
STD	Set direction	2	
CLI	Clear interrupt	2	
STI	Set interrupt	2	
HLT	Halt	3	
LOCK	Bus lock prefix	2	
ESC	Math coprocessor escape	6	
NOP	No operation	3	

D.10 Segment Override Prefix

Segment	Clocks
CS	2
SS	2
DS	2
ES	2

D.11 Effective address calculation

(R/m : Register Memory)

MOD = 00	R/m = 0--	EA = 2
	R/m = 100	EA = 1
	R/m = 101	EA = 1
	R/m = 110	EA = 2
	R/m = 111	EA = 1
MOD = 01	R/m = 0--	EA = 3
	R/m = 1--	EA = 2
MOD = 10	R/m = 0--	EA = 4
	R/m = 1--	EA = 3

OPCODE w	**mod reg R/m**	**(disp-low)**	**(disp-high)**	**(data)**	**(data)**
1 byte or 2bytes		8bit or 16bit displacement if memory operand needs.		8bit or 16bit immediate data	

R/m	**MOD**		
	00	01	10
000	(BX) + (SI)	(BX) + (SI) + D8	(BX) + (SI) + D16
001	(BX) + (DI)	(BX) + (DI) + D8	(BX) + (DI) + D16
010	(BP) + (SI)	(BP) + (SI) + D8	(BP) + (SI) + D16
011	(BP) + (DI)	(BP) + (DI) + D8	(BP) + (DI) + D16
100	(SI)	(SI) + D8	(SI) + D16
101	(DI)	(DI) + D8	(DI) + D16
110	DIRECT	(BP) + D8	(BP) + D16
111	(BX) + [DI]	(BX) + D8	(BX) + D16

E. Operating Characteristics

E.1 Absolute Maximum conditions (referenced to Vss)

PARAMETER	SYMBOL	LIMITS	UNIT
DC Supply Voltage	V_{DD}	2.7 ~ 3.6	V
Input Voltage	V_I	-0.5 ~ V_{DD}+0.3	V
DC Input Current	I_I	0.8	mA
Storage Temp. Range	T_{STG}	-55 ~ +125	℃

Notes : Stresses greater than those shown above may cause permanent damage to the device. Exposure of the device to these conditions for extended periods may also affect device reliability

E.2 Recommended Operating Conditions (referenced to Vss)

PARAMETER	SYMBOL	LIMITS	UNIT
DC Supply Voltage	V_{DD}	2.7 ~ 3.6	V
Operating Ambient Temp. Range	T_A	0 ~ 70	℃

E.3 Capacitance (specified at Vcc and ambient temperature over the designated range, Note1 and 2)

PARAMETER	SYMBOL	LIMITS	UNIT
Input Capacitance	C_I	3.0 ~ 5.2	pF
Output Capacitance	C_O	3.7 ~ 6.1	pF

Notes : 1. Commercial temperature range is 0 to 70, ±5% power supply:
2. For cell pad only.