Innovasic Semiconductor
Intelligent Embedded Control Solutions

IA186EM/IA188EM
8/16-Bit Microcontrollers

Data Sheet

Copyright © 2004

Contents

Features....4
Description....5
Bus Interface and Control....7
Peripheral Control and Registers....7
Clock and Power Management....49
- System Clocks....49
- Power-Save Mode....50
- Initialization and Reset....50
- Reset Configuration Register....50
- Chip-Selects....50
- Chip-Select Timing....50
- Ready and Wait-State Programming....50
- Chip-Select Overlap....51
- Upper Memory Chip Select....52
- Low Memory Chip Select....52
- Midrange Memory Chip Selects....52
- Peripheral Chip Selects....52
- Refresh Control....53
- Interrupt Control....53
  - Interrupt Types....54
  - Interrupt Table Notes....55
- Timer Control....55
- Direct Memory Access (DMA)....56
- DMA Operation....56
  - DMA Channel Control Registers....56
  - DMA Priority....57

Asynchronous Serial Port....58
Synchronous Serial Port....58
Programmable I/O (PIO)....59
Pin Descriptions....60
Instruction Set Summary....71
- Key to Abbreviations Used Instruction Summary Table....85

Absolute Maximum Ratings....90
DC Characteristics Over Commercial Operating Ranges....90
AC Characteristics Over Commercial Operating Ranges (40 MHz)....91
Waveforms....94
- Alphabetic Key to Waveform Parameters....94
- Numeric Key to Waveform Parameters....95
- Read Cycle Timing....97

Write Cycle 98
Write Cycle Timing 99
PSRAM Read Cycle....100
PSRAM Read Cycle Timing....101
PSRAM Write Cycle....102
PSRAM Write Cycle Timing....103
PSRAM Refresh Cycle 104
PSRAM Refresh Cycle 104
Interrupt Acknowledge Cycle 105
Interrupt Acknowledge Cycle Timing 106
Software Halt Cycle 107
Software Halt Cycle Timing 107
Clock – Active Mode 108
Clock – Power-Save Mode 108
Clock Timing 108
srdy – Synchronous Ready....109
ardy - Asynchronous Ready....109
Peripherals....109
Ready and Peripheral Timing....109
Reset 1....110
Reset 2....110
Bus Hold Entering....111
Bus Hold Leaving 111
Reset and Bus Hold Timing....111
Synchronous Serial Interface 112
Synchronous Serial Interface Timing 112
IA186EM 100-Pin PQFP 113
IA186EM TQFP 100-Pin....116
IA188EM 100-Pin PQFP 119
IA188EM 100-Pin TQFP....122
Physical Dimensions....125
PQFP 100 125
TQFP 100 127
Ordering Information....128
Errata....129

Please Note

Included in the Ordering Information section on page 128 of this manual are enhanced RoHS-compliant versions of the IA186 and IA188 family of microcontrollers. However, standard packaged or non RoHS-compliant versions of the IA186 and IA188 microcontrollers are still available.

Features

- Pin-for-pin compatible with AMD® Am186EM/188EM devices
- All features are retained, including:
  - PLL allowing same crystal/system clock frequency
  - 8086/8088 instruction set with additional 186 instruction set extensions
  - Programmable interrupt controller
  - Two DMA channels
  - Three 16-bit timers
  - Programmable chip select logic and wait-state generator
  - Dedicated watch dog timer
  - Two independent asynchronous serial ports (UARTs)
    - DMA capability
    - Hardware flow control
    - 7-, 8-, or 9-bit data capability
  - Pulse Width Demodulator feature
  - Up to 32 programmable I/O pins (PIO)
- Pseudo-static/dynamic RAM controller
- Fully static CMOS design
- 40 MHz operation at industrial operating conditions
- +5 VDC power supply
- Available packages:
    - 100-pin Thin Quad Flat Pack (TQFP)
    - 100-pin Plastic Quad Flat Pack (PQFP)

The IA186EM/188EM is a form, fit and function replacement for the original Advanced Micro Devices® Am186EM/188EM family of microcontrollers. Innovasic produces replacement ICs using its MILES™, or Managed IC Lifetime Extension System cloning technology. This technology produces replacement ICs far more complex than "emulation" while ensuring they are compatible with the original IC. MILES™ captures the design of a clone so it can be produced even as silicon technology advances. MILES™ also verifies the clone against the original IC so that even the "undocumented features" are duplicated.

This data sheet contains preliminary information for the IA186EM/188EM. The complete data sheet which documents all necessary engineering information about the IA186EM/188EM including functional and I/O descriptions, electrical characteristics, and applicable timing will be available when the device nears completion.

Description

The IA186EM/188EM family of microcontrollers replaces obsolete AMD® Am186EM/188EM devices, allowing customers to retain existing board designs, software compilers/assemblers and emulation tools, thereby avoiding expensive redesign efforts.

The IA186EM/188EM microcontrollers are an upgrade for the 80C186/188 microcontroller designs, with integrated peripherals to provide increased functionality and reduce system costs. The Innovasic devices are created to satisfy requirements of embedded products designed for telecommunications, office automation and storage and industrial controls.

A block diagram of the IA186EM/188EM microcontroller is depicted in Figure 1. The IA186EM/188EM microcontroller consists of the following functional blocks, with brief discussions of each afterwards.

- Bus Interface and Control
- Peripheral Control and Registers
- Chip Selects and Control
- Programmable I/O
- Clock and Power Management
- Direct Memory Access (DMA)
- Interrupt Controller
- Timers
- Asynchronous Serial Ports
- Synchronous Serial Interface.

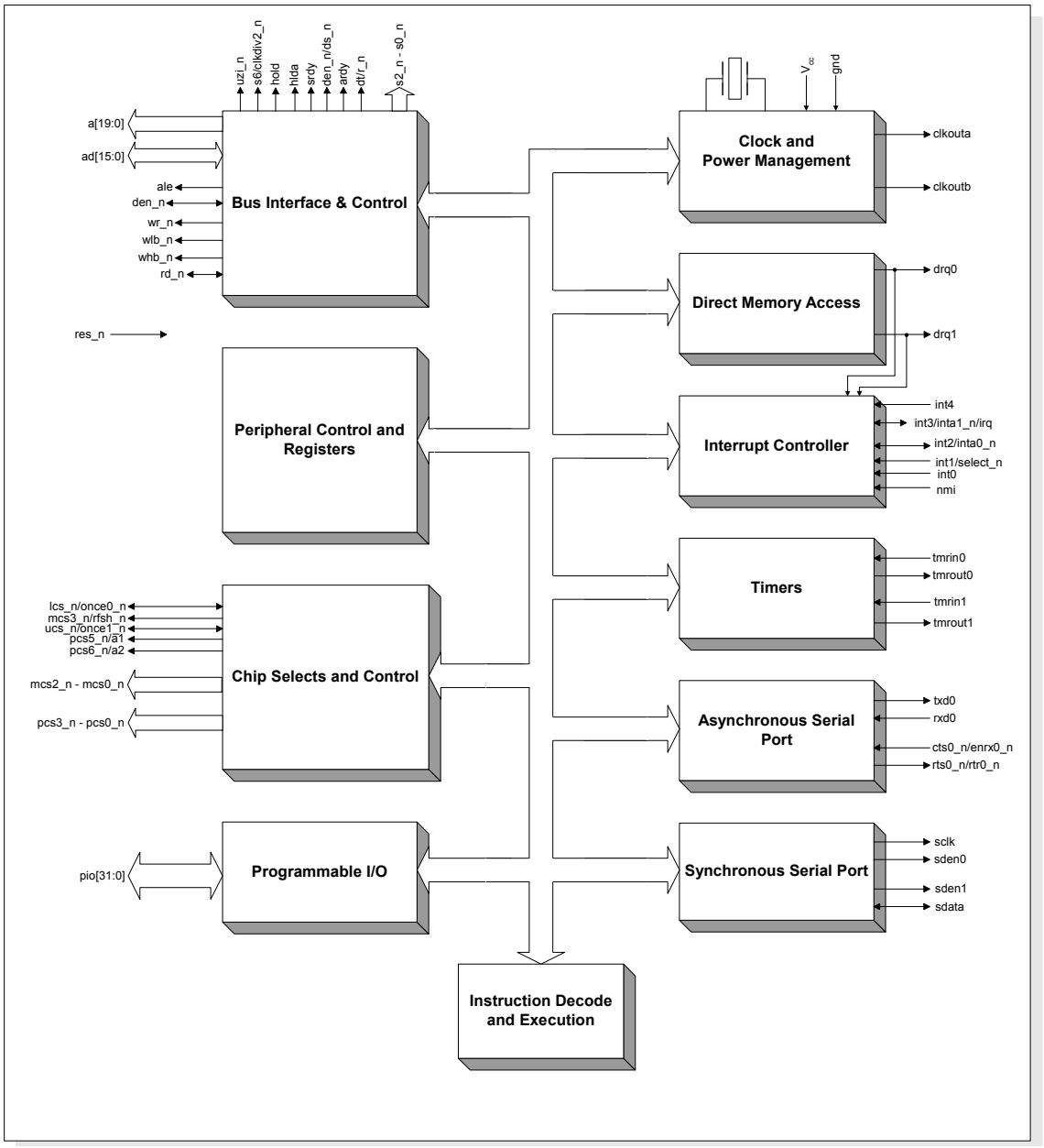


Figure 1. IA186/88EM Block Diagram

NOTE

See pin descriptions for pins that share other functions with PIO pins.
pwd, int5, int6, rts1_n/rtr1_n, **and** cts1_n/enrx1_n **are multiplexed** with int2_n/inta0_n, drq0_n, drq0_n, pcs3_n, **and** pcs2_n **respectively**.

Bus Interface and Control

Bus Interface and Control (BIC) manages all accesses to external memory and external peripherals. These peripherals may be mapped either in memory space or I/O space. The BIC supports both multiplexed and non-multiplexed bus operations. Multiplexed address and data are provided on the **ad [15:0]** bus, while a non-multiplexed address is provided on the **a [19:0]** bus. The A bus provides address information for the entire bus cycle (t1-t4), while the **ad** bus provides address information only during the first (t1) phase of the bus cycle. For more details regarding bus cycles, see the AC waveforms at the end of this datasheet.

The IA186EM microcontroller provides two signals that serve as byte write enables: write high byte (**whb_n**) and write low byte (**wlb_n**). Obviously, the IA188EM microcontroller requires only a single write byte (**wb_n**) signal to support its 8-bit data bus. **whb_n** is the logical OR of the **bhe_n** and **wr_n**. **wlb_n** is the logical OR of **ad0** and **wr_n**. **wlb_n** is the logical OR of **ad0** and **wr_n**. **wb_n** is low whenever a byte is written to the IA188EM data bus **ad[7:0].**

The byte write enables are driven in conjunction with the non-multiplexed address bus **a[19:0]** to facilitate meeting the timing requirements of common SRAMs.

The BIC also provides support for Pseudo-Static RAM (PSRAM) devices. PSRAM is supported in the lower chip select (**lcs_n**) area only. In order to support PSRAM, the Chip Selects and Control (CSC) must be appropriately programmed. For details regarding this operation, see Chip Selects.

Peripheral Control and Registers

The on-chip peripherals in the IA186EM/188EM microcontroller are controlled from a 256-byte block of internal registers. Although these registers are actually located in the peripherals they control, they are addressed within a single 256-byte block of I/O spaced and are therefore treated as a functional unit for the purposes of this document.

A map of these registers is depicted in Table 1.

All write operations performed on the IA188EM should be 8-bit writes, which will still result in 16-bit data transfers to the Peripheral Control Block (PCB) register even if the named register is an 8-bit register. Any read performed to the PCB registers should be word reads. Code written with these points in mind will run correctly on both the IA186EM and IA188EM.

However, unpredictable behavior will result in both the IA186EM and IA188EM processors if unaligned read and write accesses are performed.

Register Name	Offset
Peripheral Control Block Registers	
PCB Relocation Register	FEh
Reset Configuration Register	F6h
Processor Release Level Register	F4h
Power-Save Control Register	F0h
Enable RCU Register	E4h
Clock Prescaler Register	E2h
Memory Partition Register	E0h
DMA Registers	
DMA1 Control Register	DAh
DMA1 Transfer Count Register	D8h
DMA1 Destination Address High Register	D6h
DMA1 Destination Address Low Register	D4h
DMA1 Source Address High Register	D2h
DMA1 Source Address Low Register	D0h
DMA0 Control Register	CAh
DMA0 Transfer Count Register	C8h
DMA0 Destination Address High Register	C6h
DMA0 Destination Address Low Register	C4h
DMA0 Source Address High Register	C2h
DMA0 Source Address Low Register	C0h
Chip-Select Registers	
pcs_n and mcs_n Auxiliary Register	A8h
Mid-Range Memory Chip-Select Register	A6h
Peripheral Chip-Select Register	A4h
Low-Memory Chip-Select Register	A2h
Upper-Memory Chip-Select Register	A0h
Asynchronous Serial Port Register	
Serial Port Baud Rate Divisor Register	88h
Serial Port Receive Register	86h
Serial Port Transmit Register	84h
Serial Port Status Register	82h
Serial Port Control Register	80h
PIO Registers	
PIO Data 1 Register	7Ah
PIO Direction 1 Register	78h
PIO Mode 1 Register	76h
PIO Data 0 Register	74h
PIO Direction 0 Register	72h
PIO Mode 0 Register	70h

Register Name	Offset
Timer Registers	
Timer 2 Mode & Control Register	66h
Timer 2 Max Count Compare A Register	62h
Timer 2 Count Register	60h
Timer 1 Mode & Control Register	5Eh
Timer 1 Max Count Compare B Register	5Ch
Timer 1 Max Count Compare A Register	5Ah
Timer 1 Count Register	58h
Timer 0 Mode & Control Register	56h
Timer 0 Max Count Compare B Register	54h
Timer 0 Max Count Compare A Register	52h
Timer 0 Count Register	50h
Interrupt Registers	
Serial Port 0 Interrupt Control Register	44h
Watchdog Timer Control Register	42h
INT4 Interrupt Control Register	40h
INT3 Interrupt Control Register	3Eh
INT2 Interrupt Control Register	3Ch
INT1 Interrupt Control Register	3Ah
INT0 Interrupt Control Register	38h
DMA1 Interrupt Control Register	36h
DMA0 Interrupt Control Register	34h
Timer Interrupt Control Register	32h
Interrupt Status Register	30h
Interrupt Request Register	2Eh
In-Service Register	2Ch
Priority Mask Register	2Ah
Interrupt Mask Register	28h
Poll Status Register	26h
Poll Register	24h
End-of-Interrupt (EOI) Register	22h
Interrupt Vector Register	20h
Synchronous Serial Port Register	
Synchronous Serial Receive Register	18h
Synchronous Serial Transmit 0 Register	16h
Synchronous Serial Transmit 1 Register	14h
Synchronous Serial Enable Register	12h
Synchronous Serial Status Register	10h

Table 1. Map of Peripheral Control Registers

RELREG (0feh) - The Peripheral Control Block *REL*ocation *REG*ister maps the entire Peripheral Control Block Register Bank to either I/O or memory space. In addition, RELREG contains a bit which places the Interrupt Controller in either Master or Slave mode.

The RELREG contains 20ffh at reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES	*S/Mn*	*RES*	*IO/Mn*	*RA [19:8]*											

RES (bit 15) - Reserved.

S/Mn (bit 14) – A 1 in this bit places the Interrupt Controller into slave mode. When set to zero, the Interrupt Controller is in master mode.

RES (bit 13) - Reserved.

IO/Mn (bit 12)- A 1 in this bit maps the Peripheral Control Block Register Bank into IO space. When set to zero, the Peripheral Control Block is mapped into memory space.

RA [19:8] (bits 11-0) – Sets the base address (upper 12 bits) of the Peripheral Control Block Register Bank. RA [7:0} default to zero. Note that when bit 12 (*IO/M_n*) is a 1, *RA [19:16]* are ignored.

RESCON (0f6h) - The *RES*et *CON*figuration Register latches user-defined information present at specified pins at the rising edge of reset. This contents of this register are read-only and remain valid until the next reset.

The RESCON contains user-defined information at reset.

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

RC [15:0] (bits 15-0) – At the rising edge of reset, the values of specified pins (**ad [15:0]** for the IA186Es and {**ao [15:8], ad [7:0]**} for the IA188EM) are latched into this register.

PRL (0f4h) - The *P*rocessor *R*elease *L*evel Register contains a code corresponding to the latest processor production release. The PRL is a Read-Only Register

The PRL contains 0400h.

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

PRL [7:0] (bits 15-8) – The latest Processor Release Level.

PRL Value	Processor Release Level
01h	C
02h	D
03h	E
04h	F

RES (bits 7-0) – Reserved.

PDCON (0f0h) - The *Power-save CON*trol Register controls several miscellaneous system I/O and timing functions.

The SYSCON contains 0000h at reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSEN	*RES*			*CBF*	*CBD*	*CAF*	*CAD*	*RES*					*F2*	*F1*	*F0*

PSEN (bit 15) – When set to 1, enables the power-save mode causing the internal operating clock to be divided by the value in *F2-F0*. External interrupts or interrupts from internal interrupts automatically clear *PSEN*. Software interrupts and exception do not clear *PSEN*. Note that the value of *PSEN* is not restored upon execution of an IRET instruction.

RES (bit 14-12) – Reserved. These bits read back as zeros.

CBF (bit 11) – When set to 1, the **clkoutb** output follows the input crystal (PLL) frequency. When this bit is 0, the **clkoutb** follows the internal clock frequency after the clock divider.

CBD (bit 10) – When set to 1, the **clkoutb** output is pulled low. When this bit is 0, the **clkoutb** is driven as an output per the *CBF* bit.

CAF (bit 9) – When set to 1, the **clkouta** output follows the input crystal (PLL) frequency. When this bit is 0, the **clkouta** follows the internal clock frequency after the clock divider.
CAD (bit 8) – When set to 1, the **clkouta** output is pulled low. When this bit is 0, the **clkouta** is driven as an output per the *CBF* bit.

RES (bits 7-3) – Reserved. These bits read back as zeros.

F2-F0 (bits 2-0) – These bits control the clock divider as shown below. Note that *PSEN* must be 1 for the clock divider to function.

F2	F1	F0	Divider Factor
0	0	0	Divide by 1 (2^0)
0	0	1	Divide by 2 (2^1)
0	1	0	Divide by 4 (2^2)
0	1	1	Divide by 8 (2^3)
1	0	0	Divide by 16 (2^4)
1	0	1	Divide by 32 (2^5)
1	1	0	Divide by 64 (2^6)
1	1	1	Divide by 128 (2^7)

EDRAM (0e4h) - The *E*nable RCU Register provides control and status for the refresh counter.

The EDRAM register contains 0000h at reset.

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

E (bit 15) – When set to 1, the refresh counter is enabled and **msc3_n** is configured to act as **rfsh_n**. Clearing *E* clears the refresh counter and disables refresh requests. The refresh address is unaffected by clearing *E*.

RES (bits 14-9) – Reserved. These bits read back as 0.

T [8:0] (bits 8-0) – These bits hold the current value of the refresh counter. These bits are read-only.

CDRAM (0e2h) - The *C*lock Prescaler Register determines the period between refresh cycles.

The CDRAM register is undefined at reset.

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

RES (bits 15-9) – Reserved. These bits read back as 0.
RC [8:0] (bits 8-0) – These bits hold the clock count interval between refresh cycles. This value should not be set to less than 18 (12h), else there would never be sufficient bus cycles available for the processor to execute code.

In power-save mode, the refresh counter value should be adjusted to account for the clock divider value in SYSCON.

MDRAM (0e0h) - The *M*emory Partition Register holds the A19-A13 address bits of the 20-bit base refresh address.

The MDRAM register contains 0000h at reset.

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

M [6:0] (bits 15-9) – Upper bits corresponding to address bits **a19-a13** of the 20-Bit memory refresh address. These bits are not available on the **a19-a0** bus. When using PSRAM mode, *M6-M0* must be programmed to 0000000b.

Reserved [8:0] (bits 8-0) – Reserved. These bits read back as 0.

D1CON (0dah) - *D*MA *CON*trol Registers.

D0CON (0cah)

DMA Control Registers control operation of the two DMA channels. The D0CON and D1CON registers are undefined at reset, except *ST* that is set to 0.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DM/IOn	*DDEC*	*DINC*	*SM/IOn*	*SDEC*	*SINC*	*TC*	*INT*	*SYN1-SYN0*		*P*	*TDRQ*	*Res*	*CHG*	*ST*	*Bn/W*

DM/IOn (bit 15) – Destination Address Space Select selects memory or I/O space for the destination address. When *DM/IO* is set to 1, the destination address is in memory space. When set to 0, the destination address is in I/O space.

DDEC (bit 14) – Destination Decrement automatically decrements the destination address after each transfer when set to 1. The address is decremented by 1 or 2, depending on the byte/word bit (*Bn/W*, bit 0). The address does not change if the increment and decrement bits are set to the same value (00b or 11b).

DINC (bit 13) – Destination Increment, when set to 1, automatically increments the destination address after each transfer. The address is incremented by 1 or 2, depending on the byte/word bit (*Bn/W*, bit 0). The address does not change if the increment and decrement bits are set to the same value (00b or 11b).

SM/IOn (bit 12) – Source Address Space Select selects memory or I/O space for the source address. When *SM/IOn* is set to 1, the source address is in memory space, while when 0, the source address is in I/O space.

SDEC (bit 11) – Source Decrement, when set to 1, automatically decrements the destination address after each transfer. The address is decremented by 1 or 2, depending on the byte/word bit (*Bn/W*, bit 0). The address does not change if the increment and decrement bits are set to the same value (00b or 11b).

SINC (bit 10) – Source Increment, when set to 1, automatically increments the destination address after each transfer. The address is incremented by 1 or 2, depending on the byte/word bit (*Bn/W*, bit 0). The address does not change if the increment and decrement bits are set to the same value (00b or 11b).

TC (bit 9) – Terminal Count. *The DMA* decrements the transfer count for each DMA transfer. When *TC* is set to 1, the source or destination synchronized DMA transfers terminate when the count reaches 0, but when *TC* is set to 0, source or destination synchronized DMA transfers do not terminate when the count reaches 0. Unsynchronized DMA transfers always end when the count reaches 0, irrespective of the setting of this bit.

INT (bit 8) – Interrupt. The DMA channel generates an interrupt request on completion of the transfer count when this bit is set to 1. However, for an interrupt to be generated, the *TC* bit must also be set to 1.

SYN1-SYN0 (bits 7-6) – Synchronization Type bits select channel synchronization as shown in the following table. The value of these bits is ignored if *TDRQ* (bit 4) is set to 1. A processor reset causes these bits to be set to 11b.

SYN1	***SYN0***	**Sync Type**
0	0	Unsynchronized
0	1	Source Synchronized
1	0	Destination Synchronized
1	1	Reserved

P (bit 5) – Relative Priority. Selects high priority for this channel relative to the other channel during simultaneous transfers when set to 1.

TDRQ (bit 4) - Timer 2 Synchronization. Enables DMA requests from timer 2, when set to 1, but disables DMA requests from timer 2 when set to 0.

EXT (bit 3) – Reserved.

CHG (bit 2) – Change Start Bit. This bit must be set to 1, to allow modification of the *ST* bit during a write. During a write, when *CHG* is set to 0, *ST* is not changed when writing the control word. The result of reading this bit is always 0.

ST (bit 1) – Start/Stop DMA Channel. When the start bit is set to 1, the DMA channel is started. The *CHG* bit must be set to 1 for this bit to be modified and only during the same register write. A processor reset causes this bit to be set to 0.

Bn/W (bit 0) – Byte/Word Select. When set to 1, word transfers are selected. When set to 0, byte transfers are selected. (The IA188EM does not support word transfers and furthermore they are not supported if the chip selects are programmed for 8-bit transfers.)

D1TC (0d8h) - *D*MA *T*ransfer *C*oun*t* Registers.

D0TC (0c8h)

The DMA Transfer Count registers are maintained by each DMA channel. They are decremented after each DMA cycle. The state of the *TC* bit in the DMA control register has no influence on this activity. But, if unsynchronized transfers are programmed or if the *TC* bit in the DMA control word is set, DMA activity ceases when the transfer count register reaches 0.

The D0TC and D1TC registers are undefined at reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TC15 – TC0															

TC [15:0] (bits 15-0) – DMA Transfer Count contains the transfer count for the respective DMA channel. Its value is decremented after each transfer.

D1DSTH (0d6h) - The *D*MA *De**ST**i*nation Address *H*igh Register.
D0DSTH (0c6h)

The 20-bit destination address consists of these four bits combined with the 16-bits of the respective Destination Address Low Register. A DMA transfer requires that two complete 16-bit registers (high and low registers) be used for both the source and destination addresses of each DMA channel involved. These four registers must be initialized. Each address may be incremented or decremented independently of the other after each transfer. The addresses are incremented or decremented by two for word transfers and incremented or decremented by 1 for byte transfers.

The D0DSTH and D1DSTH registers are undefined at reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												*DDA19-DDA16*			

Reserved [15:4] (bits 15-4) – Reserved.

DDA [19:16] (bits 3-0) – DMA Destination Address High bits are driven onto A19-A16 during the write phase of a DMA transfer.

DIDSTL (0d4h) - *D*MA *De**ST**i*nation Address *L*ow Register.

D0DSTL (0c4h)

The sixteen bits of these registers are combined with the four bits of the respective DMA Destination Address High Register to produce a 20-bit destination address.

The D0DSTL and D1DSTL registers are undefined at reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DDA15 – DDA0															

DDA [15:0] (bits 15-0) – DMA Destination Address Low bits are driven onto A15-A0 during the write phase of a DMA transfer.

D1SRCH (0d2h) - *D*MA *S*ou*RC*e Address *H*igh Register.
D0SRCH (0c2h)

The 20-bit source address consists of these four bits combined with the 16-bits of the respective Source Address Low Register. A DMA transfer requires that two complete 16-bit registers in the peripheral control block (high and low registers) be used for both the source and destination addresses of each DMA channel involved. Each DMA channel requires that all four address registers be initialized. Each address may be incremented or decremented independently of the other after each transfer. The addresses are incremented or decremented by 2 for word transfers and incremented or decremented by 1 for byte transfers.

The D0SRCH and D1SRCHL registers are undefined at reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												*DSA19 –DSA16*			

Reserved [15:4] (bits 15-4) – Reserved

DSA [19:16] (bits 3-0) – DMA Source Address High bits are driven onto A19-A16 during the read phase of a DMA transfer.

D1SRCL (0d0h) - *D*MA *S*ou*RC*e Address *L*ow Register.
D0SRCL (0c0h)

The sixteen bits of these registers are combined with the four bits of the respective DMA Source Address High register to produce a 20-bit source address.

The D0SRCL and D1SRCL registers are undefined at reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSA15-DSA0															

DSA [15:0] (bits 15-0) – DMA Source Address Low bits are placed onto **a15-a0** during the read phase of a DMA transfer.

MPCS (0a8h) - *MCS* and *PCS* Auxiliary Register.

This register controls more than one type of chip select, making it different from the other chip select control registers. The MPCS register contains information for the following, **mcs3_n - mcs0_n** as well as **pcs6_n - pcs5_n** and **pcs3_n - pcs0_n**.

The MPCS register also contains a bit that configures the **pcs6_n - pcs5_n** pins as either chip selects or as alternate sources for the A2 and A1 address bits. Either address bits **a1** & **a2** or **pcs6_n - pcs5_n** are selected to the exclusion of the other. When programmed for address bits, these outputs can be used to provide latched address bits for **a2** & **a1**

pcs6_n - pcs5_n are high and not active on processor reset. An access to the MPCS register causes the pins to activate, when the **pcs6_n - pcs5_n** are configured as address pins. The **pcs6_n - pcs5_n** pins do not require corresponding access to the PACS register to be activated.

The value of the MPCS register is undefined at reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	*M6-M0*							*EX*	*MS*	*1*	*1*	*1*	*R2*	*R1-R0*	

Reserved (bit 15) – Set to 1.

M [6:0] (bits14-8) MCS_n Block Size – These seven bits determine the total block size for the MCS3_n - MCS0_n chip selects. The total block size is divided equally among the four chip selects. The following table shows the relationship between *M [6:0]* and the size of the memory block.

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

EX (bit7) 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, whereas when set to 0, **pcs6_n - pcs5_n** become address bit **a1** and address bit **a2** respectively.

MS (bit 6) Memory/ I/O Space Selector determines whether the **pcs_n** pins are active either during memory or I/O bus cycles. When *MS* is set to 1, the **pcs_n** outputs are active for memory bus cycles, and active for I/O bus cycles when set to 0.

Reserved (bits 5:3) – Set to 1.

R2 (bit 2) Ready Mode – This bit influences only the **pcs6_n - pcs5_n** chip selects. If *R2* is set to 0, external ready is required. If *R2* is set to 1, external ready is ignored. In each case, the values of the *R1-R0* bits determine the number of wait states to be inserted.

R [1:0] (bits 1-0) Wait-State Value – These bits influence only the **pcs6_n - pcs5_n** chip selects. The value of *R1-R0* determines the number of wait states inserted into an access depending on whether its to the PCS_n memory or I/O area. Up to three wait states can be inserted (*R1 - R0* = 00b to 11b).

MMCS (0a6h) - *M*idrange *M*emory *C*hip *S*elect Register.

Four chip-select pins, **mcs3_n - mcs0_n**, are provided for use within a user-locatable memory block. The memory block base address can be located anywhere within the 1-Mbyte memory address space, excluding the areas associated with the **ucs_n** and **lcs_n** chip selects (and, if mapped to memory, the address range of the Peripheral Chip Selects, **pcs6_n - pcs5_n** and **pcs3_n to pcs0_n**). If the **pcs_n** chip selects are mapped to I/O space, the **mcs_n** address range can overlap the **pcs_n** address range

Two registers program the Midrange Chip Selects. The Midrange Memory Chip Select (MMCS) register determines the base address, the ready condition and wait states of the memory block that are accessed through the **mcs_n** pins. The **pcs_n** and **mcs_n** Auxiliary (MPCS) register configures the block size. On reset the **mcs3_n - mcs0_n** pins are not active. Accessing with a write both the MMCS and MPCS registers activates these chip selects.

The **mcs3_n - mcs0_n** outputs assert with the multiplexed AD address bus (**ad15 – ad0** or **ao15 – ao8** and **ad7 – ad0**) rather than the earlier timing of the **a19 – a0** bus unlike the **ucs_n** and **lcs_n** chip selects. The timing is delayed for a half cycle later than that for **ucs_n** and **lcs_n** if the **a19 – a0** bus is used for address selection.

The value of the MMCS register is undefined at reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA19 – BA13							*1*	*1*	*1*	*1*	*1*	*1*	*R2*	*R1 - R0*	

BA [15:9] (bits 15-9) – Base Address. The value of the BA19 – BA13 determines the Base Address of the memory block that is addressed by the **mcs_n** chip select pins. These bits correspond to bits **a19 – a13** of the 20-bit memory address. The remaining bits **a12 – a0** of the base address are always 0.

The base address may be any integer multiple of the size of the memory clock selected in the MPCS register. For example, if the midrange block is 32 Kbytes, the block could be located at 20000h or 28000h but not at 24000h.

If the **lcs_n** chip select is inactive, the base address of the midrange chip selects can be set to 00000h, because the **lcs_n** chip select is defined to be 00000h but is unused. The further limitation that the base address must be an integer multiple of the block size means that a 512K MMCS block size can only be used with the **lcs_n** chip select inactive and the base address of the midrange chip selects set to 00000h.

Reserved [8:3] (bits 8-3) - Set to 1.

R2 (bit 2) – Ready mode. This bit determines the **mcs_n** chip selects ready mode. When *R2* is 0, an external ready is necessary. If *R2* is 1, an external ready is ignored. In each case, the number of wait states inserted in an access is determined by the value of the *R1 & R0* bits.

R [1:0] (bits 1-0) – Wait-State Value. The number of wait states inserted in an access is determined by the value of the *R1 & R0* bits. Up to three wait states can be inserted (*R1 - R0* = 00b to 11b).

PACS (0a4h) - *P*eripher*A*l *C*hip *S*elec*t* Register.

The Peripheral Chip Selects are asserted over 256-byte range with the same timing as the AD address bus. There are six chip selects, **pcs6_n - pcs5_n** and **pcs3_n - pcs0_n**, that are utilized in either the user-locatable memory or I/O blocks. The **pcs4_n** chip select is not implemented in the ia18xEM family of Micro controllers. Excluding the areas utilized by the **ucs_n, lcs_n,** and **mcs_n** chip selects, the memory block can be located anywhere within the 1-Mbyte address space. These chip selects may also be configured to access the 64-Kbyte I/O space.

Programming the Peripheral Chip Selects uses two registers, The Peripheral Chip Select (PACS) register and the **pcs_n** and **mcs_n** Auxiliary (MPCS) register. The PACS register establishes the base address, configures the ready mode, and determines the number of wait states for the **pcs3_n - pcs0_n** outputs.

The MPCS register configures the **pcs6_n – pcs5_n** pins to be either chip selects or address pins **a1** and **a2**. When these pins are configured as chip selects, the MPCS register determines whether they are active during memory or I/O bus cycles and determines the ready state and wait states for these output pins. These pins are not active on reset but are activated as chip selects by writing to the two registers (PACS and MPCS). To configure and activate them as address pins it is necessary to write to both the PACS and MPCS registers. **pcs6_n – pcs5_n** can be configured for 0 to 3 wait states while **pcs3_n - pcs0_n** can be programmed for 0 to 15 wait states.

The value of the PACS register is undefined at reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA19 – BA11									*1*	*1*	*1*	*R3*	*R2*	*R1 –R0*	

BA [19:11] (bits 15-7) - Base Address bits determine the base address and correspond to bits 19 - 11 of the 20-bit programmable base address of the peripheral chip select block. However, if the PCS_n chip selects are mapped to I/O space, these bits must be set to 0000b, as I/O addresses are only 16 bits wide.

PCS Address Ranges

PCSn Line	Range	
	Low	High
PCS0n	Base Address	Base Address + 255
PCS1n	Base Address + 256	Base Address + 511
PCS2n	Base Address + 512	Base Address + 767
PCS3n	Base Address + 768	Base Address + 1023
Reserved	N/A	N/A
PCS5n	Base Address + 1280	Base Address
PCS6n	Base Address + 1536	Base Address

Reserved [6:4] (bits 6-4) – Set to 1.

R [3] (bit 3) – Wait State Value. See the following table.

R [2] (bit 2) – Ready Mode. When 0, external ready is required. When 1, external ready is ignored. But in each case the number of wait states is determined as in the following table.

R [1:0] (bits 1 – 0) – Wait-State Value. See following table. It should be noted that **pcs6_n – pcs5_n** and **pcs3_n – pcs0_n** pins are multiplexed with the programmable I/O pins. And for them to function as chip selects, the PIO mode and direction settings for these pins must be set to 0 for normal operation.

PCS3n – PCS0n Wait–State Encoding

R3	*R1*	*R0*	Wait States
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	5
1	0	1	7
1	1	0	9
1	1	1	15

LMCS (0a2h) - *Low Memory Chip Select* Register configures the Low Memory Chip Select that has been provided to facilitate access to the interrupt vector table located at 00000h or the bottom of memory. The **lcs_n** pin is not active at reset.

The width of the data bus for the **lcs_n** space should be configured in the AUXCON register before activating the **lcs_n** chip select pin, by any write access to the LMCS register.

The value of the LMCS register is undefined at reset except DA, which is set to 0.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	*UB2 – UB0*			*1*	*1*	*1*	*1*	*DA*	*PSE*	*1*	*1*	*1*	*R2*	*R1-R0*	

Reserved [15] (bit 15) – Set to 0

UB [2:0] (bits 14 – 12) - Upper Boundary. These bits define the upper boundary of memory accessed by the **lcs_n** chip select. The following table gives the possible configurations of block size (max 512Kbytes).

LMCS Block Size Programming Values

Memory Block Size	**Ending Address**	***UB2 – UB0***
64K	0FFFFh	000b
128K	1FFFFh	001b
256K	3FFFFh	011b
512K	7FFFFh	111b

Reserved [11:8] (bits 11-8) - Set to 1.

DA (bit 7) Disable Address - When set to 0, the address is driven onto the address bus (**ad15 – ad0**) during the address phase of a bus cycle. If *DA* is set to 1, the address bus is disabled, providing some measure of power saving. This bit is set to 0 at reset.

If BHE_n/ADEN_n is held at 0 during the rising edge of **res_n**, then the address bus is always driven, independent of the setting of *DA*.

PSE (bit 6) PSRAM Mode Enable – PSRAM support for the **lcs_n** chip select memory space is enabled when the PSE is set to 1. The EDRAM, MDRAM, and CDRAM refresh control unit registers must be configured for auto refresh before PSRAM support is enabled. Setting the enable bit (*EN*) in the enable RCU register (EDRAM, offset e4h) configures the **mcs3_n/rfsh_n** as **rfsh_n**.

Reserved (bits 5-3) – Set to 1.

R2 (bit 2) - Ready Mode. When this bit is set to 0, an external ready is required. When set to 1, the external ready is ignored. In either case, however, the value of the *R1 - R0* bits determine the number of wait states inserted.

R [1:0] (bits R1-R0) - Wait-State Value. The number of wait states inserted into an access to the LCS_n memory area is determined by the value of these bits. This number ranges from 0 to 3 (*R1 – R0* = 00b to 11b)

UMCS (0a0h) - *U*pper *M*emor*y C*hip *S*elect Register configures the Upper Memory Chip Select pin, which is used for the top of memory. On reset, the first fetch takes place at memory location FFFF0h and thus this area of memory is usually used for instruction memory. With this in mind, UCS_n defaults to an

active state at reset with a memory range of 64 Kbytes (F0000h to FFFFFh), external ready required, and three wait states automatically inserted. The upper end of the memory range always ends at FFFFFh, whereas the lower end of this upper memory range is programmable.

The value of the UMCS register is F03Bh at reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	*LB2 – LB0*			*0*	*0*	*0*	*0*	*DA*	*0*	*1*	*1*	*1*	*R2*	*R1-R0*	

Reserved [15] (bit 15) – Set to 1.

LB [2:0] (bits 14–12) – Lower Boundary. These bits determine the bottom of the memory accessed by the **ucs_n** chip selects.

UMCS Block Size Programming Values

Memory Block Size	Starting Address	*LB2 – LB0*	Comments
64K	F0000h	111b	Default
128K	E0000h	110b	
256K	C0000h	100b	
512K	80000h	000b	

Reserved (bits 11 – 8)

DA (bit 7) – Disable Address. When set to 0, the address is driven onto the address bus (**ad15 – ad0**) during the address phase of a bus cycle when **ucs_n** is asserted. If *DA* is set to 1, the address bus is disabled, and the address is not driven on the address bus when **ucs_n** is asserted, providing some measure of power saving. This bit is set to 0 at reset.

If **bhe_n/aden_n** is held at 0 during the rising edge of **res_n**, then the address bus is always driven independent of the setting of *DA*.

Reserved (bit 6) – Set to 0.

Reserved (bit 5 – 3) – Set to 1.

R2 (bit 2) Ready Mode – When this bit is set to 0, an external ready is required. But when set to 1, the external ready is ignored. In either case, however, the value of the *R1 - R0* bits determine the number of wait states inserted.

R [1:0] (bits 1-0) - Wait-State Value. The number of wait states inserted into an access to the **lcs_n** memory area is determined by the value of these bits. This number ranges from 0 to 3 (*R1 – R0* = 00b to 11b).

SPBAUD (088h) - *S*erial *P*ort *BAUD* Rate Divisor Register.

The value in this register determines the number of internal processor cycles in one phase (half-period) of the 32 x serial clock.

The contents of these registers must be adjusted to reflect the new processor clock frequency if power-save mode is in effect.

The baud rate divisor may be calculated from:

BAUDDIV = (Processor Frequency / (32 x baud rate)) -1

By setting the BAUDDIV to 0000h, the maximum baud rate of 1/32 of the internal processor frequency clock is set. Setting BAUDDIV to 129 (81h) provides a baud rate of 9600 at 40MHz. The baud rate tolerance is +4.6% and –1.9% with respect to the actual serial port baud rate, not the target baud rate.

Baud Rates

Baud Rate	Divisor Based on CPU Clock Rate			
	20 MHz	**25 MHz**	**33 MHz**	**40 MHz**
300	2082	2603	3471	4165
600	1040	1301	1735	2082
1200	519	650	867	1040
2400	259	324	433	519
4800	129	161	216	259
9600	64	80	107	129
14400	42	53	71	85
19200	31	39	53	64
625 Kbaud	0	n/a	n/a	1
781.25 Kbaud	n/a	0	n/a	n/a
1.041 Mbaud	n/a	n/a	0	n/a
1.25 Mbaud	n/a	n/a	n/a	0

The value of the SPBAUD register at reset is undefined.

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

BAUDDIV [15:0] (bits 15-0) – Baud Rate Divisor. Defines the divisor for the internal processor clock.

SPRD (086h) - *S*erial *P*ort *R*eceive *D*ata Register.

Data received over the serial port are stored in this register until read. The data are received initially by the receive shift register (no software access) permitting data to be received while the previous data are being read.

The *RDR* bit (Receive Data Ready) in the serial port status register indicates the status of the SPRD register. Setting the *RDR* bit 1 indicates that there is valid data in the receive register.

The value of the SPRD register is undefined at reset.

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

Reserved (bits 15-8) – Reserved.

RDATA [7:0] (bits 7-0) – Holds valid data while the *RDR* bit of the status register is set.

SPTD (084h) - *S*erial *P*ort *T*ransmit *D*ata Register.

Data is written to this register by software, with the values to be transmitted by the serial port. Double buffering of the transmitter allows for the transmission of data from the transmit shift register (no software access), while the next data are written into the transmit register.

The *THRE* bit in the Serial Port Status register indicates whether there is valid data in the SPDT register. The *THRE* bit must be a 1 before writing data to this register to prevent overwriting valid data that is already in the SPDT register.

The value of the SPTD register is undefined at reset.

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

Reserved (bits 15-8) – Reserved.

TDATA [7:0] (bits 7-0) – Holds the data to be transmitted.

SPSTS (082h) – *S*erial *P*ort *ST*atu*S* Register.

This register stores information concerning the current status of the port. The status bits are described below.

The value of the SPSTS register is undefined at reset.

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

Reserved (bits 15-7) – Reserved – Set to 0.

TEMT (bit 6) – Transmitter Empty. When both the transmit shift register and the transmit register are empty, this bit is set indicating to software that it is safe to disable the transmitter.
This bit is read-only.

THRE (bit 5) – Transmit Holding Register Empty. When this bit is 1, the corresponding transmit holding register is ready to accept data. This is a read-only bit.

RDR (bit 4) – Receive Data Ready. When this bit is 1, the respective SPRD register contains valid data. This is a read/write bit and can be reset only by reading the corresponding Receive register.

BRKI (bit 3) –Break Interrupt. This bit indicates that a break has been received when this bit is set to 1 and causes a serial port interrupt request.
NOTE: This bit should be reset by software.

FER (bit 2) – Framing Error Detected. When the receiver samples the **rxd** line as low when a stop bit is expected (line high) a framing error is generated setting this bit.
NOTE: This bit should be reset by software.

PER (bit 1) - Parity Error Detected. When a parity error is detected in either mode 1 or 3, this bit is set.
NOTE: This bit should be reset by software.

OER (bit 0) – Overrun Error Detected. When new data overwrites valid data in the receive register (because it hasn't been read) an overrun error is detected setting this bit.
NOTE: This bit should be reset by software.

SPCT (080h) - *S*erial *P*ort *ConTrol* Register.

This register controls both transmit and receive parts of the serial port.

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				*TX IE*	*RX IE*	*LOOP*	*BRK*	*BRK VAL*	*PMODE*		*WLGN*	*STP*	*TMODE*	*RSIE*	*RMODE*

Reserved (bits 15-12) – Reserved. Set to 0.

TXIE (bit 11) – Transmitter Ready Interrupt Enable. This bit enables the generation of an interrupt requests whenever the transmit holding register is empty (*THRE* bit 1). The respective port does not generate interrupts when this bit is 0. Interrupts continue to be generated as long as *THRE* and the *TXIE* are 1.

RXIE (bit 10) – Receive Data Ready Interrupt Enable. This bit enables the generation of an interrupt requests whenever the receive register contains valid data (*RDR* bit 1). The respective port does not generate interrupts when this bit is 0. Interrupts continue to be generated as long as *RDR* and the *RXIE* are 1.

LOOP (bit 9) – Loop-back. The serial port is placed into the loop-back mode when this bit is set.

BRK (bit 8) – Send Break. When this bit is set to 1, the **txd** pin is driven low, overriding any data that may be in the course of being shifted out of the transmit shift register.

See the definitions of long and short break in the Serial Port Status register definition.

BRKVAL (bit 7) – Break Value. This is the ninth data bit transmitted when in modes 2 and 3. This bit is cleared at each transmitted word and is not buffered. To transmit data with this bit set high, the following procedure is recommended.

1. The TEMT bit in the serial port status register must go high.
2. Set the TB8 bit by writing it to the serial port control register.
3. Finally write the transmit character to the serial port transmit register.

Serial port 0 is a special case in that if this bit is 1, the associated pins are used for flow control overriding the Peripheral Chip Select signals. This bit is 0 at reset.

PMODE (bits6:5) – Parity Mode. When this bit is set to 1, the **txd** pin is driven low overriding any data that may be in the course of being shifted out of the transmit shift register.

See the definitions of long and short break in the Serial Port Status register definition.

WLGN (bit 4) – Word Length. The number of bits transmitted or received in a frame is determined by the value of this bit. When this bit is 0, the number of data bits in a frame is 7 whereas when this bit is 1 the number of data bits in a frame is 8. This bit is 0 at reset.

STP (bit 3) – Stop Bits. This bit specifies the number of stop bits used to indicate the end of a frame. When this bit is 0, the number of stop bits is 1. When it is 1, the number of stop bits is 2. This bit is 0 at reset.

TMODE (bit 2) – Transmit Mode. The transmit section of the serial port is enabled when this bit is 1 and disabled when this bit is 0.

RSIE (bit 1) – Receive Status Interrupt Enable. When an exception occurs during data reception an interrupt request is generated if enabled by this bit (*RSIE* = 1). Interrupt requests are made for the error conditions listed (*BRK, OER, PER,* and *FER*) in the serial port status register. This bit is 0 at reset.

RMODE (bit 0) – Receive Mode. The receive section of the serial port is enabled when this bit is 1 and disabled when this bit is 0. This bit is 0 at reset.

PDATA1 (07ah) - *P*IO *DATA* Registers.

PDATA0 (074h),

When a PIO pin is configured as an output the value in the corresponding PIO data register bit is driven onto the pin. On the other hand, if the PIO pin is configured as an input, the value on the pin is input into the corresponding bit of the PIO data register.

The following table lists the default states for the PIO pins.

PIO Pin Assignments

PIO Number	Associated Pin Name	Power-On Reset Status
0	tmrin1	Input with pull-up
1	tmrout1	Input with pull-down
2	pcs6/A2	Input with pull-up
3	pcs5/A1	Input with pull-up
4	dt/r_n	Normal operation(c)
5	den_n/ds_n	Normal operation(c)
6	srdy	Normal operation(d)

PIO Number	Associated Pin Name	Power-On Reset Status
7[a]	a17	Normal operation[c]
8[a]	a18	Normal operation[c]
9[a]	a19	Normal operation[c]
10	tmrout0	Input with pull-down
11	tmrin0	Input with pull-up
12	drq0	Input with pull-up
13	drq1	Input with pull-up
14	mcs0_n	Input with pull-up
15	mcs1_n	Input with pull-up
16	pcs0_n	Input with pull-up
17	pcs1_n	Input with pull-up
18	pcs2_n	Input with pull-up
19	pcs3_n	Input with pull-up
20	sclk	Input with pull-up
21	sdata	Input with pull-up
22	sden0	Input with pull- down
23	sden1	Input with pull- down
24	mcs2_n	Input with pull-up
25	mcs3_n/rfsh_n	Input with pull-up
26[a, b]	uzi	Input with pull-up
27	txd	Input with pull-up
28	rxd	Input with pull-up
29[a, b]	s6/clkdiv2_n	Input with pull-up
30	int4	Input with pull-up
31	int2	Input with pull-up

NOTES

1. Emulators use these pins. (**s2_n-s0_n, res_n, nmi, clkouta, bhe_n, ale, ad15 – ad0**, and **a16 – a0** are used by emulators also.)
2. If **bhe_n/aden_n** (ia186EM) or rfsh2_n/aden (ia188EM) is held low during power-on reset, these pins revert to normal operation.

3. When used as a PIO pin, it is an input with an available pull-up option.
4. When used as a PIO pin, it is an input with an available pull-down option.

The value of the PDATA registers is undefined at reset.

PDATA 0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PDATA (15 – 0)															

PDATA 1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PDATA (31 – 16)															

PDATA [15:0] (bits 15-0) – PIO Data 0 Bits. This register contains the values of the bits that are either driven on or received from the corresponding PIO pins, depending on its configuration each pin as either an output or an input. The values of these bits correspond to those in the PIO direction registers and PIO Mode registers.

PDATA [31:16] (bits 15-0) – PIO Data 1 Bits. This register contains the values of the bits that are either driven on, or received from, the corresponding PIO pins depending on its configuration each pin as either an output or an input. The values of these bits correspond to those in the PIO direction registers and PIO Mode registers

The PIO pins can be operated as open-drain outputs by:
1. Maintaining the data constant in the appropriate bit of the PIO data register.
2. Writing the value of the data bit into the respective bit position of the PIO Direction register, so that the output is either 0 or disabled depending on the value of the data bit.

PDIR1 (078h) - *P*IO *DIR*ection Registers.

PDIR0 (072h)

Each PIO pin is configured as an input or an output by the corresponding bit in the PIO direction register.

PIO Mode and PIO Direction Settings

PIO Mode	PIO Direction	Pin function
0	0	Normal operation
0	1	PIO input with pullup/pulldown
1	0	PIO output
1	1	PIO input without pullup/pulldown

PDIR0

The value of the PDIR0 register is FC0Fh at reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PDIR (15 – 0)															

PDIR1

The value of the PDIR1 register is FFFFh at reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PDIR (31 – 16)															

PDIR [15:0] (bits 15-0) – PIO Direction 0 Bits. For each bit, if the value is 1, the pin is configured as an input and as an output if the value is 0. The values of these bits correspond to those in the PIO data registers and PIO Mode registers.

PDIR [31:16] (bits 15-0) – PIO Direction 1 Bits. For each bit, if the value is 1, the pin is configured as an input and as an output if the value is 0. The values of these bits correspond to those in the PIO data registers and PIO Mode registers.

PIOMODE1 (076h) - *P*IO *MODE* Registers.

PIOMODE0 (070h)

Each PIO pin is configured as an input or an output by the corresponding bit in the PIO direction register.

The bit number of PMODE corresponds to the PIO number.

See the table *PIO Mode and PIO Direction Settings* in PDIR description above**.**

PIOMODE0

The value of the PIOMODE0 register is 0000h at reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMODE (15 – 0)															

PIOMODE1

The value of the PIOMODE1 register is 0000h at reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMODE (31 – 16)															

PMODE [15:0] (bits 15-0) – PIO Mode 0 Bits. For each bit, if the value is 1 then the pin is configured as an input and as an output if the value is 0. The values of these bits correspond to those in the PIO data registers and PIO Mode registers.

PMODE [31:16] (bits 15-0) – PIO Mode 1 Bits. For each bit, if the value is 1 then the pin is configured as an input and as an output if the value is 0. The values of these bits correspond to those in the PIO data registers and PIO Mode registers.

T1CON (05eh) - *T*ime*r 0 and T*ime*r 1* Mode and *CON*trol Registers.

T0CON (056h)

This registers controls the operation of the Timer 1 and Timer 0 respectively.

The value of both the T0CON and T1CON registers is 0000h at reset.

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

EN (bit 15) – Enable Bit. The timer is enabled when the *EN* bit is 1. The timer count is inhibited when the EN bit is 0. This bit is write-only, but with the *INHn* bit set to 1 in the same write operation.

INHn (bit 14) – Inhibit Bit. Gates the setting of the enable (*EN*) bit. This bit must be set to 1 in the same write operation that sets the enable (*EN*) bit. Otherwise, the EN bit will not be changed. This bit always reads as 0.

INT (bit 13) – Interrupt Bit. An interrupt request is generated when the Count register reaches its maximum, *MC* = 1, by setting the *INT* bit to 1. In dual maxcount mode, an interrupt request is generated when the count register reaches the value in maxcount A or maxcount B. No interrupt requests are generated if this bit is set to 0. If an interrupt request is generated and then the enable bit is cleared before said interrupt is serviced, the interrupt request will remain.

RIU (bit 12) – Register in Use Bit. This bit is set to 1 when the maxcount register B is used to compare to the timer count value. It is set to 0 when the maxcount compare A register is used.

Reserved (bits 11-6) – Set to 0.

MC (bit 5) – Maximum Count. When the timer reaches its maximum count this bit is set to 1 regardless of the interrupt enable bit. This bit is also set every time Maxcount Compare register A or B is reached, when in dual maxcount mode. This bit may be used by software polling to monitor timer status rather than through interrupts if desired.

RTG (bit 4) – Retrigger Bit. This pin controls the timer function of the timer input pin. When set to 1, the count is reset by a 0 to 1 transition on **timrin0** or **tmrin1.** When set to 0, a high input on **tmrin0** or **tmrin1** enables the count and a 1 holds the timer value. This bit is ignored if the external clocking (EXT=1) bit is set.

P (bit 3) – Prescaler Bit. P is ignored if external clocking is enabled (*EXT* = 1). Timer 2 prescales the timer when P is set to 1. Otherwise, the timer is incremented on every fourth CLKOUT cycle.

EXT (bit 2) – External Clock Bit. This bit determines whether an external or internal clock is used. *EXT* = 1, an external clock is used and *EXT* = 0, an internal is used.

ALT (bit 1) – Alternate Compare Bit. If set to 1, the timer will count to Maxcount Compare A, reset the count register to 0, count to maxcount compare B, reset the count register to 0 and begin again at maxcount compare A.

If set to 0, the timer will count to maxcount compare A, reset the count register to 0, and begin again at maxcount compare A. Maxcount compare B is not used in this case.

CONT (bit 0) – Continuous Mode Bit. The timer will run continuously when this bit is set to 1. The timer will stop after each count run and *EN* will be cleared if the *CONT* bit is set to 0. If *CONT* = 1 and *ALT* = 1, the respective timer counts to the maxcount compare A value and resets, then commences counting to maxcount compare B value, resets and ceases counting.

T2CON (066h) - *T*im*er 2 Mode* and *CON*trol Register.

This register controls the operation of the Timer 2.

The value of the T2CON register is 0000h at reset.

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

EN (bit 15) – Enable Bit. The timer is enabled when the *EN* bit is 1. The timer count is inhibited when the *EN* bit is 0. Setting this bit to 1 by writing to the T2CON register requires that the *INH* bit be set to 1 during the same write. This bit is write-only, but with the *INHn* bit set to 1 in the same write operation.

INHn (bit 14) – Inhibit Bit. Gates the setting of the enable (*EN*) bit. This bit must be set to 1 in the same write operation that sets the enable (*EN*) bit. This bit always reads as 0.

INT (bit 13) – Interrupt Bit. An interrupt request is generated, by setting the *INT* bit to 1, when the Count register reaches its maximum, *MC* = 1.

Reserved (bits 12-6) – Set to 0.
MC (bit 5) – Maximum Count. When the timer reaches its maximum count this bit is set to 1, regardless of the interrupt enable bit. This bit may be used by software polling to monitor timer status rather than through interrupts if desired.

Reserved (bits 4-1) – Set to 0.

CONT (bit 0) – Continuous Mode Bit. The timer will run continuously when this bit is set to 1. The timer will stop after each count run and *EN* will be cleared if this bit is set to 0.

T2COMPA (062h), - *T*im*er* Maxcount *COM*pare Registers.

T1COMPB (05ch)

T1COMPA (05ah)

T0COMPB (054h)

T0COMPA (052h)

These registers contain the maximum count value that is compared to the respective count register. Timer 0 and Timer 1 have two of these compare registers each.

If Timer 0 or Timer 1 or both are configured to count and compare firstly to register A and then register B, the **tmrout0** or **tmrout1** signals may be used to generate various duty-cycle wave forms.

Timer 2 has only one compare register, T2COMPA.

If one of these timer maxcount compare registers is set to 0000h, the respective timer will count from 0000h to FFFFh before generating an interrupt request. For example, a timer configured in this manner with a 40MHz clock will interrupt every 6.5536 mS.

The value of these registers is undefined at reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TC15 – TC0															

TC [15:0] (bits 15-0) – Timer Compare Value. The timer will count to the value in the respective register before resetting the count value to 0.

T2CNT (060h) - *T*imer *C*ou*NT* Registers.

T1CNT (058h)

T0CNT (050h),

These registers are incremented by one every four internal clock cycles if the relevant timer is enabled.

The Increment of Timer 0 and Timer 1 may also be controlled by external signals **tmrin0** and **tmrin1** respectively, or prescaled by Timer 2.

Comparisons are made between the count registers and maxcount registers and action taken dependent on achieving the maximum count.

The value of these registers is undefined at reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TC15 – TC0															

TC [15:0] (bits 15-0) – Timer Count Value. This register has the value of the current count of the related timer that is incremented every fourth processor clock in internal clocked mode. Alternatively, the register is incremented each time the Timer 2 maxcount is reached if using Timer 2 as a prescaler. Timer 0 and Timer 1 may be externally clocked by tmrin0 and tmrin1 signals.

SPICON (044h) - *S*erial *P*ort *I*nterrupt *CON*trol Register.

Master Mode

This register controls the operation of the asynchronous serial port interrupt source (SPI, bit 10 in of the Interrupt Request register)

The value of this register is 001Fh at reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											*Res*	*MSK*	*PR2-PR0*		

Reserved (bits 15-5) – Set to 0.

Reserved (bit 4) – Set to1.

MSK (bit 3) – Mask. This bit, when 0, enables the serial port to cause an interrupt. When this bit is 1, the serial port is prevented from generating an interrupt.

PR2-PR0 (bits 2-0) – Priority. These bits define the priority of the serial port interrupt in relation to other interrupt signals. The interrupt priority is the lowest at 7 at reset. The values of *PR2 – PR0* are shown in the following table.

Priority Level

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

WDCON (044h) – *WatchDog timer interrupt CON*trol Register.

Master Mode

These registers control the operation of the Watchdog Timer interrupt source. The value of this register is 000Fh at reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											*Res*	*MSK*	*PR2-PR0*		

Reserved (bits 15-5) – Set to 0.

Reserved (bit 4) – Set to 0.

MSK (bit 3) – Mask. This bit, when 0, enables the Watchdog Timer to cause an interrupt. When this bit is 1 prevents the Watchdog Timer from generating an interrupt.

PR2-PR0 (bits 2-0) – Priority. These bits define the priority of the Watchdog Timer interrupt in relation to other interrupt signals. The interrupt priority is the lowest at 7 at reset. The values of *PR2 – PR0* are shown in the above table (Priority Level).

I4CON (040h) – *I*NT*4 CON*trol Register.

Master Mode

This register controls the operation of the **int4** signal, which is only intended for use in fully nested mode. The interrupt is assigned to type 10h.

The value of the I4CON register is 000Fh at reset.

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

Reserved (bits 15-5) – Set to 0.

LTM (bit 4) – Level-Triggered Mode. The int4 interrupt may be edge or level triggered depending on the value of the bit. If *LTM* is 1, int4 is active high-level sensitive interrupt l. If *LTM* is 0, int4 is a rising edge triggered interrupt. The interrupt int4 must remain active (high) until serviced.

MSK (bit 3) – Mask. The int4 signal can cause an interrupt if the *MSK* bit is 0. The int4 signal cannot cause an interrupt if the *MSK* bit is 1.

PR2-PR0 (bit 2-0) – Priority. These bits define the priority of the serial port interrupt in relation to other interrupt signals. The interrupt priority is the lowest at 7 at reset. The values of *PR2 – PR0* are shown in the above table (Priority Level).

I3CON (03eh) – *I*NT*2*/*I*NT*3 CON*trol Register.

I2CON (03ch),

Master Mode

INT2 and INT3 are designated as interrupt type 0eh and 0fh respectively.

The **int2** and **int3** pins may be configured as the interrupt acknowledge pins **inta0_n** and **inta1_n** respectively in cascade mode.

The value of these registers is 000Fh at reset.

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

Reserved (bits 15-5) – Set to 0.

LTM (bit 4) – Level-Triggered Mode The **int2** or **int3** interrupt may be edge or level triggered depending on the value of this bit. If *LTM* is 1, **int2** or **int3** is an active high level-sensitive interrupt. If *LTM* is 0, **int2** or **int3** is a rising edge triggered interrupt. The interrupt **int2** or **int3** must remain active (high) until acknowledged.

MSK (bit 3) – Mask. The int2 or int3 signal can cause an interrupt if the *MSK* bit is 0. The **int2** or **int3** signal cannot cause an interrupt if the *MSK* bit is 1. The Interrupt Mask Register has a duplicate of this bit.

PR2-PR0 (bit 2-0) – Priority. These bits define the priority of the serial port interrupt **int2** or **int3** in relation to other interrupt signals. The interrupt priority is the lowest at 7 at reset. The values of *PR2 – PR0* are shown in the above table (Priority Level).

I1CON (03ah) – *I*NT*0*/*I*NT*1 CON*trol Register.

I0CON (038h),

(Master Mode)

IINT0 and INT1 are designated as interrupt type 0ch and 0dh respectively.

The **int2** and **int3** pins may be configured as the interrupt acknowledge pins **inta0** and **inta1** respectively, the interrupt acknowledge signals for **int0** and **int1** in cascade mode.

The value of these registers is 000Fh at reset.

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

Reserved (bits 15-7) – Set to 0.

SPNM (bit 6) – Special Fully Nested Mode. This bit enables fully nested mode for int0 or int1 when set to 1.

C (bit 5) – Cascade Mode. This bit enables cascade mode for int0 or int1 when set to 1.

LTM (bit 4) – Level-Triggered Mode. The **int0** or **int1** interrupt may be edge or level triggered depending on the value of the bit. If *LTM* is 1, **int0** or **int1** is an active high level-sensitive interrupt. If *LTM* is 0, **int0** or **int1** is a rising edge triggered interrupt. The interrupt **int0** or **int1** must remain active (high) until acknowledged.

MSK (bit 3) – Mask. The **int0** or **int1** signal can cause an interrupt if the *MSK* bit is 0. The **int0** or **int1** signal cannot cause an interrupt if the *MSK* bit is 1. The Interrupt Mask Register has a duplicate of this bit.

PR2-PR0 (bit 2-0) – Priority. These bits define the priority of the serial port interrupt **int0** or **int1** in relation to other interrupt signals. The interrupt priority is the lowest at 7 at reset. The values of *PR2 – PR0* are shown in the above table (Priority Level).

TCUCON (032h) - *T*imer *C*ontrol *U*nit Interrupt *CON*trol Register.

Master Mode

The three timers have their interrupts assigned to types 08h, 12h, and 13h and are configured by this register.

The value of this register is 000Fh at reset.

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

Reserved (bits 15-4) – Set to 0.

MSK (bit 3) – Interrupt Mask. An interrupt source may cause an interrupt if the *MSK* bit is 0. The interrupt source cannot cause an interrupt if the *MSK* bit is 1. The Interrupt Mask Register has a duplicate of this bit.

PR2-PR0 (bit 2-0) – Priority. These bits define the priority of the serial port interrupt in relation to other interrupt signals. The interrupt priority is the lowest at 7 at reset. The values of *PR2 – PR0* are shown in the above table (Priority Level).

T2INTCON (03ah) - *T*imer *INT*errupt *CON*trol Register.

T1INTCON (038h)

T0INTCON (032h)

Slave Mode

The three timers, Timer2, Timer1, and Timer0, each have an interrupt control register, whereas in master mode all three are masked and prioritized in one register (TCUCON).

The value of these registers is 000Fh at reset.

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

Reserved (bits 15-4) – Set to 0.

MSK (bit 3) – Mask. Any of the interrupt sources may cause an interrupt if the *MSK* bit is 0. The interrupt sources cannot cause an interrupt if the *MSK* bit is 1. The Interrupt Mask Register has a duplicate of this bit.

PR2-PR0 (bit 2-0) – Priority. These bits define the priority of the serial port interrupts in relation to other interrupt signals. The interrupt priority is the lowest at 7 at reset. The values of *PR2 – PR0* are shown in the above table (Priority Level).

DMA1CON/INT6CON (036h) – *DMA* and *INT*errupt *CON*trol Register. **DMA0CON/INT5CON (034h)**

Master Mode

The DMA0 and DMA1 interrupts have interrupt type 0ah and 0bh respectively. These pins are configured as external interrupts or DMA requests in the respective DMA Control register.

The value of these registers is 000Fh at reset.

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

Reserved (bits 15-4) – Set to 0.

MSK (bit 3) – Mask. Any of the interrupt sources may cause an interrupt if the *MSK* bit is 0. The interrupt sources cannot cause an interrupt if the *MSK* bit is 1. The Interrupt Mask Register has a duplicate of this bit.

PR2-PR0 (bits 2-0) – Priority. These bits define the priority of the serial port interrupts in relation to other interrupt signals. The interrupt priority is the lowest at 7 at reset. The values of *PR2 – PR0* are shown in the above table (Priority Level).

DMA1CON/INT6 (036h) – *DMA* and *INT*errupt *CON*trol Register.

DMA0CON/INT5 (034h)

Slave Mode

The two DMA control registers maintain their original functions and addressing that they possessed in Master Mode. These pins are configured as external interrupts or DMA requests in the respective DMA Control register.

The value of these registers is 000Fh at reset.

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

Reserved (bits 15-4) – Set to 0.

MSK (bit 3) – Mask. Any of the interrupt sources may cause an interrupt if the *MSK* bit is 0. The interrupt sources cannot cause an interrupt if the *MSK* bit is 1. The Interrupt Mask Register has a duplicate of this bit.

PR2-PR0 (bits 2-0) – Priority. These bits define the priority of the serial port interrupts in relation to other interrupt signals. The interrupt priority is the lowest at 7 at reset. The values of *PR2 – PR0* are shown in the above table (Priority Level).

INTSTS (030h) – *INT*errupt *ST*atu*S* Register.

Master Mode

The Interrupt status register contains the interrupt request status of each of the three timers, Timer2, Timer1, and Timer0.

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

DHLT (bit 15) – DMA Halt. DMA activity is halted when this bit is 1. It is set to 1 automatically when any non-maskable interrupt occurs and is cleared to 0 when an IRET instruction is executed. Interrupt handlers and other time critical software may modify this bit directly to disable DMA transfers. However, the *DHLT* bit should not be modified by software if the timer interrupts are enabled as the function of this register as an interrupt request register for the timers would be compromised.

Reserved (bits 14-3)

TMR [2:0] (bit 2-0) – Timer Interrupt Request. A pending interrupt request is indicated by the respective timer, when any of these bits is 1. (N.B. the *TMR* bit in the *REQST* register is a logical OR of these timer interrupt requests)

Slave Mode

When nonmaskable interrupts occur the interrupt status register controls DMA operation and the interrupt request status of each of the three timers, Timer2, Timer1, and Timer0.

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

DHLT (bit 15) – DMA Halt. DMA activity is halted when this bit is 1. It is set to 1 automatically when any non-maskable interrupt occurs and is cleared to 0 when an IRET instruction is executed.

Reserved (bits 14-3)

TMR [2:0] (bit 2-0) – Timer Interrupt Request. A pending interrupt request is indicated by the respective timer, when any of these bits is 1. (N.B. the *TMR* bit in the *REQST* register is a logical OR of these timer interrupt requests.)

REQST (02eh) – Interrupt *REQ*ue*ST* Register.

Master Mode

This is a read-only register and such a read results in the status of the interrupt request bits presented to the interrupt controller.

The REQST register is undefined on reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					*SPI*	*WD*	*I4*	*I3*	*I2*	*I1*	*IO*	*D1- D0*		*Res*	*TMR*

Reserved (bits 15 – 11)

SPI (bit 10) – Serial Port Interrupt Request. This is the serial port interrupt state and when enabled is the logical OR of all the serial port 0 interrupt sources: - *THRE, RDR, BRKI, FER, PER,* and *OER*.

WD (bit 9) – Watchdog Timer Interrupt Request. This is the watchdog interrupt state and indicates that an interrupt is pending when it is a 1.

I [4:0] (bits 8 - 4) Interrupt Requests. Setting any of these bits to 1 indicates that the relevant interrupt has a pending interrupt.

D1-D0 (bit 3:2) DMA Channel Interrupt 6 Request. Setting either bit to 1 indicates that either the respective DMA channel has a pending interrupt.

Reserved (bit 1)

TMR (bit 0) – Timer Interrupt Request. This is the timer interrupt state and is the logical OR of the timer interrupt requests. Setting this bit to 1 indicates that the timer control unit has a pending interrupt.

Slave Mode

This is a read-only register and such a read results in the status of the interrupt request bits presented to the interrupt controller. The status of these bits is available when this register is read.

When an internal interrupt request (*D1, D0, TMR2, TMR1*, or *TMR0*) occurs, the respective bit is set to 1. The internally generated interrupt acknowledge resets these bits.

The REQST register contains 0000h on reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										*TMR2*	*TMR1*	*D1*	*D0*	*Res*	*TMR0*

Reserved (bits 15 – 6)

TMR2 (bit 5) Interrupt Requests. Setting this bit to 1 indicates that timer 2 has a pending interrupt.

TMR1 (bit 4) Interrupt Requests. Setting this bit to 1 indicates that timer 1 has a pending interrupt.

D1:D0 (bits 3:2) DMA Channel Interrupt Request. Setting either bit to 1 indicates that the respective DMA channel has a pending interrupt.

Reserved (bit 1)

TMR0 (bit 0) – Timer0 Interrupt Request. Setting this bit to 1 indicates that timer 0 has a pending interrupt.

INSERV (02ch) – *IN-SERV*ice Register.

Master Mode

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

When one of these bits is set, an interrupt request will not be generated by the microcontroller for the respective source. This prevents an interrupt from interrupting itself if interrupts are enabled in the ISR. This restriction is bypassed in fully Special Fully nested mode for the INT0 and INT1 sources.

The INSERV register contains 0000h on reset

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					*SPI*	*WD*	*I4*	*I3*	*I2*	*I1*	*IO*	*D1*	*D0*	*Res*	*TMR*

Reserved (bits 15 – 11)

SPI (bit 10) – Serial Port Interrupt Request. This is the serial port 0 interrupt state.

WD (bit 9) – Watchdog Timer Interrupt In-Service Request. This bit is the In-Service state of the Watchdog Timer.

I [4:0] (bits 8 - 4) Interrupt Requests. Setting any of these bits to 1 indicates that the relevant interrupt has a pending interrupt.

D1-D0 (bit 3:2) DMA Channel Interrupt In-Service. This bit is the In-Service state of the respective DMA channel.

Reserved (bit 1)

TMR (bit 0) – Timer Interrupt Request. This is the timer interrupt state and is the logical OR of the timer interrupt requests. Setting this bit to 1 indicates that the timer control unit has a pending interrupt.

Slave Mode

This is a read-only register and such a read supplies the status of the interrupt request bits presented to the interrupt controller.

When an internal interrupt request (*D1, D0, TMR2, TMR1*, and *TMR0*) occurs, the respective bit is set to 1. The internally generated interrupt acknowledge resets these bits.

The *REQST* register contains 0000h on reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										*TMR2*	*TMR1*	*D1*	*D0*	*Res*	*TMR0*

Reserved (bits 15 – 6)

TMR2 (bit 5) Timer2 Interrupt In Service. Timer 2 is being serviced when this bit is set to 1.
TMR1 (bit 4) Timer1 Interrupt IN Service. Timer 1 is being serviced when this bit is set to 1.

D1-D0 (bit 3:2) DMA Channel Interrupt In Service. The respective DMA channel is being serviced when this bit is set to 1.

Reserved (bit 1)

TMR0 (bit 0) – Timer Interrupt In Service. Timer 0 is being serviced when this bit is set to 1.

PRIMSK (02ah) – *PRI*ority *MaSK* Register.
Master and Slave Mode

This register contains a value that sets the minimum priority level at which an interrupt can be generated by a maskable interrupt.

The PRIMSK register contains 0007h on reset

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 – PRM0*		

Reserved (bits 15 – 3) – Set to 0.

PRM 2-PRM0 (bits 2 - 0) Priority Field Mask. This three-bit field sets the minimum priority necessary for a maskable interrupt to generate an interrupt. Any maskable interrupt with a numerically higher value than that contained by these three bits is masked.

Priority Level

Priority	***PR2 – PR0***
(High) 0	0 0 0b
1	0 0 1b
2	0 1 0b
3	0 1 1b
4	1 0 0b
5	1 0 1b
6	1 1 0b
(Low) 7	1 1 1b

Any unmasked interrupt can generate an interrupt if the priority level is set to 7. On the other hand, if the priority level is set to say 4, only unmasked interrupts with a priority of 0 to 5 are permitted to generate interrupts.

IMASK (028h) – *I*nterrupt *MASK* Register.

Master Mode

The interrupt mask register is read/write. Setting a bit in this register is effectively the same as setting the *MSK* bit in the corresponding interrupt control register. Setting a bit to 1 masks the interrupt. The interrupt request is enabled when the corresponding bit is set to 0.

The IMASK register contains 07fdh on reset

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					*SPI*	*WD*	*I4*	*I3*	*I2*	*I1*	*I0*	*D1-D0*		*Res*	*TMR*

Reserved (bits 15 – 11)

SPI (bit 10) – Serial Port Interrupt Mask. Setting this bit to 1 is an indication that the asynchronous serial port interrupt is masked.

WD (bit 9) – Watchdog Timer Interrupt In-Service Request. Setting this bit to 1 is an indication that the Watchdog Timer interrupt is masked.

I [4:0] (bits 8 - 4) Interrupt Mask. Setting any of these bits to 1 is an indication that the relevant interrupt is masked.

D1-D0 (bit 3:2) DMA Channel Interrupt Mask. Setting this bit to 1 is an indication that the respective DMA channel interrupt is masked.

Reserved (bit 1)

TMR (bit 0) – Timer Interrupt Mask. When set to 1, it indicates that the timer control unit interrupt is masked.

Slave Mode

The interrupt mask register is read/write. Setting a bit in this register is effectively the same as setting the *MSK* bit in the corresponding interrupt control register. Setting a bit to 1 masks the interrupt request. The interrupt request is enabled when the corresponding bit is set to 0.

The IMASK register contains 003dh on reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										*TMR2*	*TMR1*	*D1*	*D0*	*Res*	*TMR0*

Reserved (bits 15 – 6)

TMR2 (bit 5) Timer2 Interrupt Mask. This bit provides an indication of the state of the mask bit in the Timer Interrupt Control register. When it is set to 1, it indicates that the interrupt request is masked.

TMR1 (bit 4) Timer1 Interrupt Mask. This bit provides an indication of the state of the mask bit in the Timer Interrupt Control register. When it is set to 1, it indicates that the interrupt request is masked.

D1-D0 (bit 3:2) DMA Channel Interrupt Mask. This bit provides an indication of the state of the mask bit in the respective DMA channel Interrupt Control register. When it is set to 1, it indicates that the interrupt request is masked.

Reserved (bit 1)

TMR0 (bit 0) – Timer Interrupt Mask. This bit provides an indication of the state of the mask bit in the Timer Interrupt Control register. When it is set to 1, it indicates that the interrupt request is masked.

POLLST (026h) – *POLL ST*atus Register.

Master Mode

This register reflects the current state of the Poll register and can be read without affecting its contents. However, when the Poll Register is read, it causes the current interrupt to be acknowledged and be replaced by the next interrupt.

The poll status register is read-only.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IREQ	*Reserved*										*S4 – S0*				

IREQ (bit 15) – Interrupt Request. This bit is set to 1 when an interrupt is pending. And during this state, the *S4 - S0* bits contain valid data.

Reserved (bits 14-6) Set to 0

S [4:0] (bit 4-0) – Poll Status. These bits show the interrupt type of the highest priority pending interrupt.

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

POLL (024h) – *POLL* Register.

Master Mode

When the Poll Register is read, it causes the current interrupt to be acknowledged and be replaced by the next interrupt. The poll status register reflects the current state of the Poll register and can be read without affecting its contents.

The POLL register is read-only.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IREQ	*Reserved*										*S4 – S0*				

IREQ (bit 15) – Interrupt Request. This bit is set to 1 when an interrupt is pending. And during this state, the *S4 - S0* bits contain valid data.

Reserved (bits 14-6)

S [4:0] (bit 4-0) – Poll Status. These bits show the interrupt type of the highest priority pending interrupt.

EOI (022h) – *E*nd-*O*f-*I*nterrupt Register.

Master Mode

The In Service flags of the In-Service register are reset when a write is made to the EOI register.

The interrupt service routine (ISR) should write to the EOI to reset the IS bit, in the In-Service register, for the interrupt before executing an IRET instruction that ends an interrupt service routine.

The specific EOI reset is the preferred method for resetting the IS bits as it is most secure.

The EOI register is write-only.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NSPEC	*Reserved*										*S4 – S0*				

NSPEQ (bit 15) – Non-Specific EOI. This bit is set to 1 a non-specific EOI and when set to 0 it indicates the specific EOI.

Reserved (bits 14-5)

S [4:0] (bit 4-0) – Source Interrupt Type. These bits show the interrupt type of the highest priority pending interrupt.

EOI (022h) – Specific *E*nd-*O*f-*I*nterrupt Register.

Slave Mode

An In Service flag of a specific priority, in the In-Service register, is reset when a write is made to the EOI register.

A three-bit user supplied priority-level value that points to the in-service bit that is to be reset. Writing this value to this register resets the specific bit.

The EOI register is write only and undefined at reset.

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*	*L2 – L0*		

Reserved (bits 15-3) – Write as 0.

L[2:0] (bit 2-0) – Interrupt Type. The priority or the IS (interrupt service) bit to be reset is encoded in these three bits. Writing to these bits caused the issuance of an EOI for the interrupt type. See Table 3 - Interrupt Types.

INTVEC (020h) –*INT*errupt *VEC*tor Register.

Slave Mode

The CPU shifts left 2 bits (multiplies by 4) an 8-bit interrupt type, generated by the interrupt controller, to produce an offset into the interrupt vector table.

The INTVEC register is undefined at reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	*0*	*0*	*0*	*0*	*0*	*0*	*0*	*T4 – T0*					*0*	*0*	*0*

Reserved (bits 15-8) – Read as 0.
T [4:0] (bits 7-3) – Interrupt Type. These five bits contain the five most significant bits of the interrupt types used for the internal interrupt type. The least significant three bits of the interrupt type are supplied by the interrupt controller, as set by the priority level of the interrupt request.

Reserved (bits 2-0) – Read as 0.

SSR (018h) – *S*ynchronous *S*erial *R*eceive Register.

This register holds the serial data received on the SSI port.

The value of the SSR register is undefined at reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								*SR7-SR0*							

Reserved (bits 15-8) – Reserved Bits.

SR[7:0] (bits 7-0) – Data received over the SDATA pin.

SSD0 (016h) – *S*ynchronous *S*erial *Transmit* Registers.

SSD0 (014h)

These registers hold the data to be transmitted by the SSI ports.

The value of these registers is undefined at reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								*SD7-SD0*							

Reserved (bits 15-8) – Reserved Bits.

SD[7:0] (bits 7-0) – Data to be transmitted over the SDATA pin.

SSC (012h) – *S*ynchronous *S*erial *C*ontrol Register.

This register controls the operation of the **sden1** and **sden0** outputs and the baud rate of the SSI port. The **sden1** and **sden0** outputs are held high when the respective bit is set to 1, but in the event that both *DE1* and *DE0* are set to 1 then only **sden0** will be held high.

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										*SCLKDIV*		*RES*		*DE1-DE0*	

Reserved (bits 15-6) – Reserved Bits.

SCLKDIV (bits 5-4) – SCLK Divide. These bits set the SCLK frequency. SCLK is the result of dividing the internal processor clock by 2, 4, 8, or 16 as in the following table.

SCLKDIV	SCLK Frequency Divider
00b	Processor Clock /2
01b	Processor Clock /4
10b	Processor Clock /8
11b	Processor Clock /16

RES (bits 3-2) – Reserved Bits.

DE1 (bit1) - SDEN1. The SDEN1 bit is held high when this bit is set to 1 and SDEN1 is held low when this bit is set to 0.

DE0 (bit0) – SDEN0. The SDEN0 bit is held high when this bit is set to 1 and SDEN0 is held low when this bit is set to 0.

SSS (010h) – *S*ynchronous *S*erial *S*tatus Register.

This is a read only register that indicates the state of the SSI port.

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													*RE/TE*	*DR/DT*	*PB*

Reserved (bits 15-3) – Reserved Bits.

RE/TE (bit 2) – Receive/Transmit Error Detect. This bit is set to 1 when a read of the Synchronous Serial Received register or a write to one of the transmit register is detected while the interface is busy

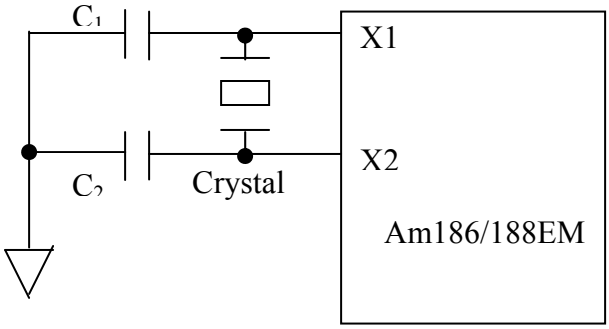
(*PB* = 1). This bit is reset to 0 when the *SDEN* output is not active (DE1-DE0 in the SSC register are 00h).

DR/DT (bit 1) – Data Receive/Transmit Complete. This bit is set to a 1 when the transmission of data bit 7 is completed (SCLK rising edge) during a transmit or receive operation. This bit is reset by a read of the SSR register, when either the SSD0 or SSD1 register is written, when the SSS register is read (unless the SSI completes an operation and sets the bit in the same cycle), or when both SDEN0 and SDEN1 become inactive.

PB (bit 0) SSI Port busy. This bit indicates that a data transmit or receive is occurring when it is set to 1. When set to 0 it indicates that the port is ready to transmit or receive data.

Clock and Power Management

A phase-lock-loop (PLL) and a second programmable system clock output (CLKOUTB) are included in the clock and power management unit. The internal clock is the same frequency as the crystal but with a duty cycle of 45% - 55 %, as a worse case, generated by the PLL obviating the need for an x2 external clock. A power-on reset (POR) resets the PLL.



Recommended range of values for C_1 and C_2 are:

C_1 = 15pF +/- 20%

C_2 = 22pF +/- 20%

Figure 2. Crystal Configuration

System Clocks

If required, the internal oscillator can be driven by an external clock source that should be connected to X1, leaving X2 unconnected.

The clock outputs **clkouta** and **clkoutb** may be enabled or disabled individually (Power-Save Control register (PDCON) bits (11 – 8)). These clock control bits allow one clock output to run at PLL frequency and the other to run at the power-save frequency.

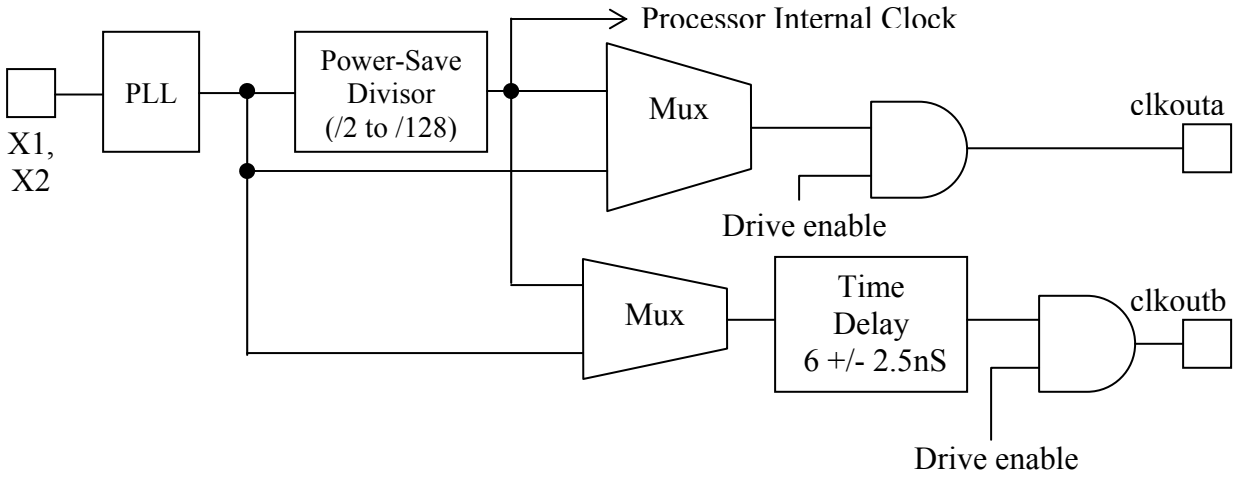


Figure 3. Organization of Clock

Power-Save Mode

The operation of the CPU and peripherals operate at a slower clock frequency when in power save mode reducing power consumption and thermal dissipation. Should an interrupt occur, the microcontroller returns to its normal operating frequency automatically on the internal clock's next rising edge in t_3. Any clock dependent devices should be reprogrammed for the changed in frequency during the power-save mode period.

Initialization and Reset

res_n (Reset), the highest priority interrupt, must be held low for 1mS during power-up to initialize the microcontroller correctly. This operation makes the device cease all instruction execution and local bus activity. The microcontoller begins instruction execution at physical address FFFF0h when **res_n** becomes inactive and after an internal processing interval with **ucs_n** asserted and three wait states. Reset also sets up certain registers to predetermined values and resets the Watchdog timer.

Reset Configuration Register

The data on the address/data bus (**ad15 – ad0** for the Am186EM and **ao15 – ao8** and **ad7 – ad0** for the Am188EM) are written into the Reset Configuration register when reset is low. This data is system dependent and is held in the Reset Configuration register after Reset is de-asserted. This configuration data may be placed on the address/data bus by using weak external pull-up and pull-down resistors or applied to the bus by an external driver, as the processor does not drive the bus during reset. It is a method of supplying the software with some initial data after a reset; for example, option jumper positions.

Chip Selects

Chip select generation is programmable for memories and peripherals. Programming is also available to produce ready and wait-state generation plus latched address bits **a1** and **a2**. For all memory and I/O cycles, the chip-select lines are active within their programmed areas, regardless of whether they are generated by the internal DMA unit or the CPU.

There are six chip selects outputs for memories and a further six for peripherals whether in memory or I/O space. The memory chip-selects are able to address three memory ranges, whereas the peripheral chip-selects are used to address 256-byte blocks that are offset from a programmable base address. Writing to a chip-select register enables the related logic even in the event that the pin in question has another function, as for example in the case that the pin is programmed to be a PIO.

Chip Select Timing

For normal timing, the **ucs_n** and **lcs_n** outputs are asserted with the non-multiplexed address bus.

Ready and Wait-State Programming

Each of the memory or peripheral chip-select lines can have a ready signal programmed that can be the **ardy** or **srdy** signal. The chip-select control registers (UMCS, LMCS, MMCS, PACS, and MPCS) have a single bit that selects if the external ready signal is to be used or not (*R2*, bit 2). *R1 & R0* (bits 1-0) in these registers control the number of wait-states that are inserted during each access to a memory or

peripheral location (from 0 to 3). The control registers for **pcs3_n – pcs0_n** utilize three bits, *R3, R1 – R0* (bits 3, 1 – 0) to provide 5, 7, 9, and 15 wait-states in addition to the original values of 0 – 3 wait states.

In the case where an external ready has been selected as required, internally programmed wait-states will always be completed before the external ready can finish or extend a bus cycle. As an example, consider a system in which the number of wait-states to be inserted has been set to three. The external ready pin is sampled by the processor during the first wait cycle. The access is completed after seven cycles (4 cycles plus 3 wait-cycles) if the ready is asserted. Alternatively, if the ready is not asserted during the first wait cycle the access is prolonged until ready is asserted and two more wait-states are inserted followed by t_4.

Chip Select Overlap

Overlapping chip selects are those configurations in which more than one chip-select is asserted for the same physical address. For example, if PCS is configured in I/O space with LCS or any other chip select configured for memory, address 00000h is not overlapping the chip selects. It is not recommended that multiple chip-select signals be asserted for the same physical address, although it may be inescapable in certain systems. If this is the case, then all overlapping chip-selects must have the same external ready configuration and the same number of wait-states to be inserted into access cycles.

Internal signals are employed to access the peripheral control block (PCB) and these signals serve as chip selects that are configured with no wait-states and no external ready. Therefore, the PCB can be programmed with addresses that overlap external chip-selects only if these chip selects are configured in the same manner.

Care should be exercised in the use of the Disable Address (*DA*) bit in the LMCS or UMCS registers when overlapping an additional chip-select with either the **lcs_n** or **ucs_n** chip-selects. Setting the DA bit to 1 prevents the address from being driven onto the AD bus for all accesses for which the respective chip-select is active, including those accesses for which the multiple selects are active.

The **mcs_n** and **pcs_n** pins are dual-purpose pins, either as chip-selects or PIO inputs or outputs. However, their respective ready and wait-state configurations for their chip-select function will be in effect no matter for which function these two pins are actually programmed. This requires that even if these pins are configured as PIO and enabled (by writing to the MMCS and MPCS registers for the **mcs_n** chip-selects and to the PACS and MPCS registers for the **pcs_n** chip-selects), the ready and wait-state settings for these signals must agree with the settings for any over-lapping chip-selects as if they had been configured as chip-selects.

Even though **pcs4_n** is not available as an external pin it has ready and wait-state logic and must therefore follow the rules for overlapping chip-selects. **pcs6_n** and **pcs5_n** on the other hand have ready and wait-state logic that is disabled when these pins are configured as address bits **a2** and **a1** respectively.

If the chip-select configuration rules are not followed, the processor may hang with the appearance of waiting for a ready signal even in a system in which ready (**ardy** or **srdy**) is always set to 1.

Upper Memory Chip Select

The **ucs_n** chip-select is for the top of memory. On reset, the micro controller begins fetching and executing instructions at memory location FFFF0h. As a result, upper memory is usually utilized for instruction memory. To this end, **ucs_n** is active on reset and has a memory range of 64Kbytes (F0000h to FFFFFh) as default along with external ready required and three wait-states automatically inserted. The lower boundary of **ucs_n** is programmable to provide ranges of 64Kbytes to 512Kbytes.

Low Memory Chip Select

The **lcs_n** chip-select is for lower memory. As the interrupt vector table is at the bottom of memory beginning at 00000h, this pin us usually utilized for control data memory. Unlike **ucs_n**, this pin is inactive on reset, but it can be activated by any read or write to the LMCS register.

Midrange Memory Chip Selects

There are four midrange chip-selects, **mcs3_n-mcs0_n**, which may be used in a user-located memory block. The base address of the memory block may be located anywhere in the 1-Mbyte memory address space with some exceptions. The memory spaces used by the **ucs_n** and **lcs_n** chip-selects are excluded, as are the **pcs6_n, pcs5_n,** and **pcs3_n – pcs0_n**. If the **pcs_n** chip-selects are mapped to I/O space then the MCS address range can overlap the PCS address range.

Both the Midrange Memory Chip Select (MMCS) register and the MCS and PCS Auxiliary register (MPCS) registers are used to program the four midrange chip-selects. The MPCS register is used to configure the block size, whereas the MMCS register configures the base address, the ready condition, and the wait states of the memory block accessed by the mcs_n pin. The chip selects (**mcs3_n-mcs0_n)** are activated by performing a read or write operation of the MMCS and MPCS registers. The assertion of the MCS outputs occurs with the same timing as the multiplexed AD address bus (**ad15-ad0** or **ao15-ao8** and **ad7-ad0**). The **a19-a0** may be used for address selection, but the timing will be delayed by a half clock cycle over the timing used for the **ucs_n** and **lcs_n.**

Peripheral Chip Selects

There are six peripheral chip-selects, **pcs6_n, pcs5_n,** and **pcs3_n – pcs0_n**, that may be used within a user-defined memory or I/O block. The base address of this user-defined memory block can be located anywhere within the 1-Mbyte memory address space except for the spaces associated with the **ucs_n**, **lcs_n**, and **mcs_n** chip selects. Or it may be programmed to the 64Kbyte I/O space. **pcs4_n** is not available.

Both the Peripheral Chip Select (PACS) register and the MCS and PCS Auxiliary register (MPCS) registers are used to program the six peripheral chip-selects, **pcs6_n, pcs5_n,** and **pcs3_n – pcs0_n.** The PACS register sets the base address, the ready condition, and the wait states for the **pcs3_n – pcs0_n** outputs.

The MPCS register configures **pcs6_n** and **pcs5_n** pins as either chip selects or address pins **a1** and **a2** respectively. When these pins are chip selects the MPCS register also configures them as being active during memory or I/O bus cycles, and their ready and wait states.

None of the **pcs_n** pins are active at reset. Both the Peripheral Chip Select (PACS) register and the MCS and PCS Auxiliary register (MPCS) registers must be read or written to activate the **pcs_n** pins aws chip selects.

pcs6_n and **pcs5_n** may be programmed to have 0 to 3 wait-states, whereas **pcs3_n – pcs0_n** may be programmed to have these and 5, 7, 9, and 15 wait-states.

Refresh Control

The Refresh Control Unit (RCU) generates refresh bus cycles. The RCU generates a memory read request after a programmable period of time to the bus interface unit.

The *ENA* bit in the Enable RCU register (EDRAM) enables refresh cycles, operating off the processor internal clock. If the processor is in power-save mode, the RCU must be reconfigured for the new clock rate.

If the **hlda** pin is asserted when a refresh request is initiated (indicating a bus hold condition), the processor disables the **hlda** pin to allow a refresh cycle to be performed. The external circuit bus master must deassert the **hold** signal for at least one clock period to permit the execution of the refresh cycle.

Interrupt Control

Interrupt requests originate from a variety of internal and external sources that are arranged by the internal interrupt controller in priority order and presented one by one to the processor.

Six external interrupt sources, five maskable (**int4-int0**) and one nonmaskable (NMI), are connected to the processor and six internal interrupt sources (three timers, two DMA channels, and the asynchronous serial port that are not brought out to external pins).

The five external maskable interrupt request pins can be used as direct interrupt requests. However, should more interrupts be needed, **int3-int0** may be with an external interrupt controller of the 82C59A type. By programming the internal interrupt controller to slave mode, an external 82C59A compatible interrupt controller can be used as the system master. Interrupt nesting can be used in all cases that permit interrupts of a higher priority to interrupt those of a lower priority.

When an interrupt is accepted, other interrupts are disabled, but may be re-enabled by setting the Interrupt Enable Flag (*IF*), in the Processor Status Flags register, during the Interrupt Service Routine (ISR). Setting *IF* permits interrupts of equal or greater priority to interrupt the currently running ISR.

Further interrupts from the same source will be blocked until the corresponding bit in the In-Service register (INSERV) is cleared. Special Fully Nested mode is invoked for **int0** and **int1** by the *SFNM* bit in the INT0 and INT1 control registerm, respectively, when this bit is set to 1. In this mode a new interrupt may be generated by these sources regardless of the in-service bit. The following table shows the priorities of the interrupts at power-on reset.

Interrupt Types

Interrupt Name	Interrupt Type	Vector Table Address	EOI Type	Overall Priority	Related Instructions
Divide Error Exception (1)	00h	00h	N/A	1	DIV, IDIV
Trace Interrupt (2)	01h	04h	N/A	1A	All
Non-maskable Interrupt (NMI)	02h	08h	N/A	1B	
Breakpoint Interrupt (1)	03h	0ch	N/A	1	INT3
INT0 Detected Overflow Exception (1)	04h	10h	N/A	1	INT0
Array Bounds Exception (1)	05h	14h	N/A	1	BOUND
Unused Opcode Exception (1)	06h	18h	N/A	1	Undefined Opcodes
ESC Opcode Exception (1, 3)	07h	1ch	N/A	1	ESC Opcodes
Timer 0 Interrupt (4, 5)	08h	20h	08h	2A	
Timer 1 Interrupt (4, 5)	12h	48h	08h	2B	
Timer 2 Interrupt (4, 5)	13h	4ch	08h	2C	
Reserved	09h	24h			
DMA 0 Interrupt (5)	0ah	28h	0ah	3	
DMA 1 Interrupt (5)	0bh	2ch	0bh	4	
INT0 Interrupt	0ch	30h	0ch	5	
INT1 Interrupt	0dh	34h	0dh	6	
INT2 Interrupt	0eh	38h	0eh	7	
INT3 Interrupt	0fh	3ch	0fh	8	
INT4 Interrupt (6)	10h	40h	10h	9	
Watchdog Timer Interrupt (6)	11h	44h	11h	9	
Asynchronous Serial Port Interrupt (6)	14h	50h	14h	9	
Reserved	15h – 1fh	54h – 7ch			

Interrupt Table Notes

If the user does not change priority levels, the default priority level will be used for the interrupt sources.

1. Instruction execution generates interrupts.
2. Performed in the same manner as for the 8086 & 8088.
3. An ESC opcode causes a trap.
4. Only one IRQ is generated for the three timers so they share priority level with regard to other sources. The timers themselves have an interrupt priority order among themselves (2A > 2B > 2C).
5. These interrupt types are programmable in Slave mode.
6. Not available in slave mode.

Timer Control

The IA186EM and IA188EM each have three 16-bit programmable timers.

Timer0 and timer1 each have an input and an output connected to external pins that permit them to count or time events, produce variable duty-cycle waveforms or non-repetitive waveforms. Timer1 can also be configured as a Watchdog timer.

Timer2 does not have any external connections. Therefore, it is confined to internal functions such as real-time coding, time-delay applications, a prescaler for timer0 and timer1, or to synchronize DMA transfers.

The Peripheral Control Block contains eleven 16-bit registers to control the programmable timers. The present value of the timer is located in the associated timer-count register, which may be read from or written to at any time regardless of whether the timer is in operation or not. The value of the timer-count register is incremented by the microcontroller every time a timer event takes place.

The maximum value that each timer can reach is determined by the value stored in the associated maximum count register. Upon reaching this maximum count value, the timer count register is reset to 0 in the same clock cycle that this count was attained, so that the timer count register does not store this maximum value. Both timer0 and timer1 have two maximum count registers, a primary and a secondary register, permitting each timer to alternate between two discrete maximum values.

Timer0 and timer1 can have the maximum count registers configured in one of two ways, primary only or both primary and secondary. If only the primary is configured to operate, on reaching the maximum count the output pin will go low for one clock period. If both the primary and secondary registers are enabled, the output pin reflects the state of whichever of the two registers is in control at the time, generating the required waveform that is dependent on the two values in the maximum count registers.

The timers can operate at a quarter of the internal clock frequency, as they are polled every fourth clock period. Alternatively, an external clock can be used. However, in this case the timer output can take six clock cycles to respond to the input.

Direct Memory Access (DMA)

Direct memory access (DMA) relieves the CPU of involvement in the transfer of data between memory and peripherals over either one or both high-speed DMA channels. Data can be transferred from memory to I/O, I/O to memory, memory-to-memory, or I/O-to-I/O. Furthermore, the DMA channels can be connected to the asynchronous serial port.

The IA186EM microcontroller supports the transfer of both bytes and words, to and from, even or odd addresses, but it does not support word transfers to memory that is configured for byte accesses. The IA188EM does not support word transfers at all. Each data transfer will take two bus cycles (a minimum of 8 clock cycles).

There are three sources of DMA requests for each DMA channel: the channel request pin (**drq1 – drq0**), Timer2, or the system software. The two channels can be programmed to have different priorities to facilitate the resolution of simultaneous DMA requests or to interrupt a transfer on the other channel.

DMA Operation

The Peripheral Control Block contains six registers for each DMA channel to control and specify the operation of the channels. The six registers consist of a pair of registers to store a 20-bit source address, a pair of registers to store a 20-bit destination address, a 16-bit transfer count register, and a 16-bit control register.

The number of DMA transfers required is designated in the DMA Transfer Count register and can be up to 64K bytes or words and, furthermore, will end automatically. DMA channel function is defined by the Control registers, which along with the other 5 registers can be changed at any time, including during a DMA transfer and are implemented immediately.

DMA Channel Control Registers

See **D1CON (0dah)** & **D0CON (0cah)** - *D*MA *CON*trol Registers above. Briefly, these registers specify the following:

- Is the data destination in memory or I/O space? (Bit 15).
- Is the destination address incremented, decremented, or unchanged after each transfer? (Bit 14 & 13).
- Is the data source in memory or I/O space? (Bit 12).
- Is the source address incremented, decremented, or unchanged after each transfer? (Bit 11 & 10).
- Do DMA transfers cease upon reaching a designated count? (Bit 9).
- Does the last transfer generate an interrupt? (Bit 8).
- Synchronization mode. (Bits 7 & 6).
- The relative priority of one DMA channel with respect to the other. (Bit 5).
- Acceptance of DMA requests from Timer2. (Bit 4).
- Byte or word transfers. (Bit 0).

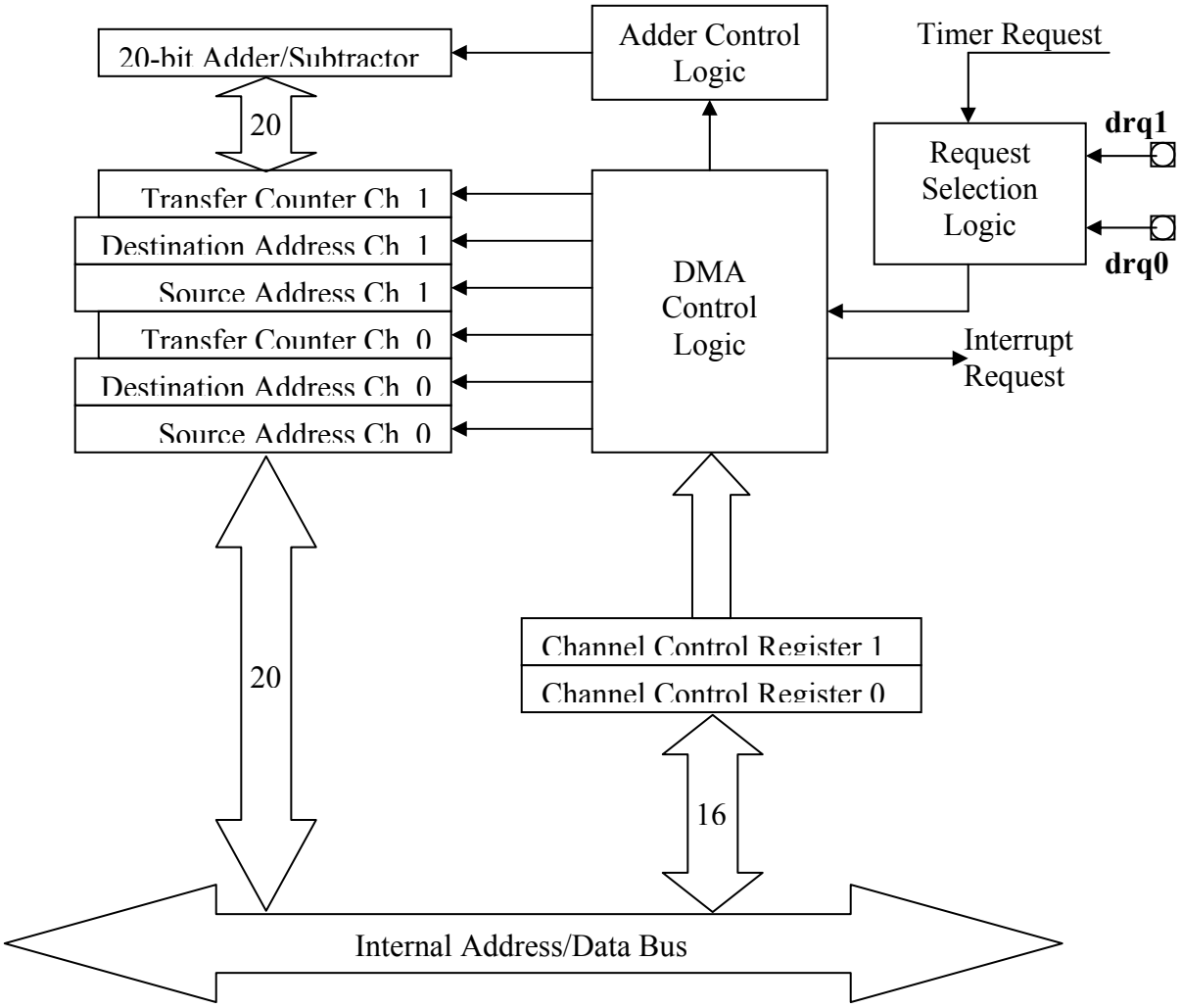


Figure 4. DMA Unit

DMA Priority

DMA transfers have a higher priority than CPU transfers, with the exception of word accesses to odd memory locations or between locked memory addresses. The CPU cannot access memory during a DMA transfer and a DMA transfer cannot be suspended by an interrupt request. Continuous DMA activity will thus cause interrupt latency to suffer. However, an NMI request halts any DMA activity, enabling the CPU to respond promptly to the request.

Asynchronous Serial Port

The asynchronous serial port employs standard industry communication protocols in its implementation of full duplex, bi-directional data transfers. The port can be the source or destination of DMA transfers.

The following features are supported:

- Full-duplex data transfers
- 7-, 8-, or 9-bit data transfers
- Odd, even, or no parity
- One or two stop bits
- Error detection provided by Parity, Framing, or Overrun errors
- Hardware handshaking is achieved with the following selectable control signals: Clear-to-send (**cts_n**)
    - Enable receiver request (**enrx_n**)
    - Ready to send (**rts_n**)
    - Ready to receive (**rtr_n**)
- DMA to and from the port
- The port has its own maskable interrupt
- The port has an independent baud rate generator
- Maximum baud rate is 1/32 of the processor clock
- Transmit and Receive lines are double buffered

In power-save mode the baud rate generator divide factor must be re-programmed to compensate for the change in clock rate.

Synchronous Serial Port

The synchronous serial port allows the microcontrollers to communicate with ASICs that are required to be programmed but have a pin shortage. The four-pin interface allows half-duplex, bi-directional data transfer at a maximum of 20 Mbits/sec with a 40 MHz CPU clock.

The synchronous serial interface of the AI186EM/AI188EM operates as the master port in a master/slave arrangement.

There are four pins in the synchronous serial interface for communication with the system elements. These pins are two enables (SDEN0 and SDEN1), a clock (SCLK) and a data pin (SDATA).

In power-save mode, the baud rate generator divide factor must be re-programmed to compensate for the change in clock rate.

Programmable I/O (PIO)

32 pins are programmable as I/O signals. The following tables list these pins with their pin name and PIO number, first in PIO numerical order, then in pin name alphabetical order. Programming a pin as a PIO should only be performed if the normal pin function is not required as the normal function is disabled and no longer has any affect on the pin. A PIO pin can be programmed as an input or output with or without either a weak pull-up or pull-down, or as an open-drain output. Following a power-on reset, the PIO pins have default status as shown in the following tables.

PIO No.	Associated Pin	Power-On Reset Status
0	**tmrin1**	Input with pull-up
1	**tmrout1**	Input with pull-down
2	**pcs6_n/a2**	Input with pull-up
3	**pcs5_n/a1**	Normal operation(3)
4	**dt/r_n**	Normal operation(3)
5	**den_n**	Normal operation(3)
6	**srdy**	Normal operation(3)
7(1)	**a17**	Normal operation(3)
8(1)	**a18**	Normal operation(3)
9(1)	**a19**	Normal operation(3)
10	**tmrout0**	Input with pull-down
11	**tmrin0**	Input with pull-up
12	**drq0**	Input with pull-up
13	**drq1**	Input with pull-up
14	**mcs0_n**	Input with pull-up
15	**mcs1_n**	Input with pull-up
16	**pcs0_n**	Input with pull-up
17	**pcs1_n**	Input with pull-up
18	**pcs2_n**	Input with pull-up
19	**pcs3_n**	Input with pull-up
20	**sclk**	Input with pull-up
21	**sdata**	Input with pull-up
22	**sden0**	Input with pull-up
23	**sden1**	Input with pull-up
24	**mcs2_n**	Input with pull-up
25	**mcs3_n/rfsh_n**	Input with pull-up
26(1,2)	**uzi_n**	Input with pull-up
27	**txd**	Input with pull-up
28	**rxd**	Input with pull-up
29(1,2)	**s6/clkdiv2**	Input with pull-up
30	**int4**	Input with pull-up
31	**int2**	Input with pull-up

Associated Pin	PIO No.	Power-On Reset Status
a17	7	Normal operation(3)
a18	8	Normal operation(3)
a19	9	Normal operation(3)
den_n/ds_n	5	Normal operation(3)
drq0	12	Input with pull-up
drq1	13	Input with pull-up
dt/r_n	4	Normal operation(3)
int2/	31	Input with pull-up
int4	30	Input with pull-up
mcs0_n	14	Input with pull-up
mcs1_n	15	Input with pull-up
mcs2_n	24	Input with pull-up
mcs3_n/rfsh_n	25	Input with pull-up
pcs0_n	16	Input with pull-up
pcs1_n	17	Input with pull-up
pcs2_n	18	Input with pull-up
pcs3_n	19	Input with pull-up
pcs5_n/a1	3	Input with pull-up
pcs6_n/a2	2	Input with pull-up
rxd	28	Input with pull-up
s6/clkdiv2	29	Input with pull-up(1,2)
sclk	20	Input with pull-up
sdata	21	Input with pull-up
sden0	22	Input with pull-up
sden1	23	Input with pull-up
srdy	6	Normal operation(4)
tmrin0	11	Input with pull-up
tmrin1	0	Input with pull-up
tmrout0	10	Input with pull-down
tmrout1	1	Input with pull-down
txd	27	Input with pull-up
uzi_n	26	Input with pull-up

NOTES

These notes apply to both tables:

1. Emulators use these pins and also **s2_n-s0_n, res_n, nmi, clkouta, bhe_n ale, ad15-ad0,** and **a15-a0**.
2. If **bhe_n/aden_n** (IA186EM) or **rfsh_n/aden_n** (IA188M) is held low during power-on reset, these pins will revert to normal operation.
3. Input with pull-up option available when used as PIO.
4. Input with pull-down option available when used as PIO.

These default status setting may be changed as desired.

The three most significant bits of the address bus (**a19 – a17**) start with their normal function on power-on reset, permitting the processor to begin fetching instructions from the boot address FFFF0h. Furthermore, normal function is the default setting for **dt/r_n**, **den_n**, and **srdy** on power-on reset.

s6/clkdiv2_n and **uzi_n** automatically return to normal operation in the event that the **ad15-ad0** bus override is enabled. The **ad15-ad0** bus override is enabled if the **bhe_n/aden_n** for the IA186EM, or the **rfsh2_n/aden_n** for the IA188EM, is held low during power-on reset.

Pin Descriptions

a19 (pio9), a18 (pio8), a17 (pio7), a16 – a0 Address Bus (synchronous outputs with tristate)
These pins are the system's source of non-multiplexed I/O or memory addresses and occur a half CLKOUTA cycle before the multiplexed address/data bus (**ad15-ad0** for the IA186EM or **ao15_ao8** and **ad7-ad0** for the AI188EM). The address bus is tristated during a bus hold or reset.

ad15 – ad8 ***IA186EM*** **Address/Data bus (level-sensitive synchronous inouts with tristate)**
These pins are the system's source of time-multiplexed I/O or memory addresses and data. The address function of these pins can be disabled. (See **bhe_n/aden_n** pin description.) If the address function of these pins is enabled, the address will be present on this bus during t_1 of the bus cycle and data will be present during t_2, t_3, and t_4 of the same bus cycle.

If **whb_n** is not active, these pins are tristated during t_2, t_3, and t_4 of the bus cycle.

The address/data bus is tristated during a bus hold or reset.

These pins can be used to load the internal Reset Configuration register (RESCON, offset 0F6h) with configuration data during a power-on reset.

ad7 – ad0 Address/Data bus (level-sensitive synchronous inouts with tristate)
These pins are the system's source of time-multiplexed low-order byte of the addresses for I/O or memory and 8-bit data. The low-order address byte will be present on this bus during t_1 of the bus cycle and the 8-bit data will be present during t_2, t_3, and t_4 of the same bus cycle.

The address function of these pins can be disabled. (See **bhe_n/aden_n** pin description.)

If **wlb_n** is not active, these pins are tristated during t_2, t_3, and t_4 of the bus cycle. The address/data bus is tristated during a bus hold or reset.

ao15 – ao8 ***IA188EM*** **Address-only bus (level-sensitive synchronous outputs with tristate)**
The address-only bus will contain valid high-order address bits during the bus cycle (t_1, t_2, t_3, and t_4) if the bus is enabled.

These pins are combined with **ad7-ad0** to complete the multiplexed address bus and are tristated during a bus hold or reset condition.

ale Address Latch Enable (synchronous output)
This signal indicates the presence of an address on the address bus (**ad15-ad0** for the IA186EM or **ao15-ao8** and **ad7-ad0** for the AI188EM), which is guaranteed to be valid on the falling edge of **ale**.

ardy Asynchronous Ready (level-sensitive asynchronous input)
This asynchronous signal provides an indication to the microcontroller that the addressed I/O device or memory space will complete a data transfer. This active high signal is asynchronous with respect to **clkouta** and if the falling edge of **ardy** is not synchronized to **clkouta** and additional clock cycle may be added

ardy should be tied high to maintain a permanent assertion of the ready condition. On the other hand, if the **ardy** signal is not used by the system it should be tied low, which passes control to the **srdy** signal.

bhe_n/aden_n *IA186EM only* Bus High Enable (synchronous output with tristate) /Address Enable (input with internal pull-up)
bhe_n - bhe_n and address **bit ad0** or **a0** inform the system which bytes of the data bus (upper, lower, or both) are involved in the current memory access bus cycle as shown in the following table.

bhe_n	ad0	Type of Bus Cycle
0	0	Word Transfer
0	1	High-Byte Transfer (Bits 15-8)
1	0	Low-Byte Transfer (Bits 7-0)
1	1	Refresh

bhe_n does not require latching and during bus hold and reset is tristated. It is asserted during t_1 and remains so through t_3 and t_w.

The high and low byte write enable functions of **bhe_n** and **ad0** are performed by **whb_n** and **wlb_n** respectively.

When using the **ad** bus, DRAM refresh cycles are indicated by **bhe_n/aden_n** and **ad0** both being high. During refresh cycles the **a** and **ad** busses may not have the same address during the address phase of the **ad** bus cycle necessitating the use of **ad0** as a determinant for the refresh cycle rather than A0.

An additional signal is utilized for PSRAM refreshes (see **mcs3_n/rfsh_n** pin description).

aden_n
There is a weak internal pull-up on **bhe_n/aden_n** obviating the need for an external pull-up and reducing power consumption.

Holding **aden_n** high or letting it float during power-on reset passes control of the address function of the **ad** bus (**ad15-ad0**) during LCS and UCS bus cycles from **aden_n** to the DA bit in LMCS and UMCS registers. When the address function is selected, the memory address is placed on the **a19-a0** pins.

Holding **aden_n** low during power-on reset, both the address and data are driven onto the **ad** bus independently of the DA bit setting. This pin is normally sampled one clock cycle after the rising edge of **res_n**.

clkouta – Clock Output A (synchronous output)
This pin is the internal clock output to the system. Bits 9, 8, and 2-0 of the Power-Save Control register (PDCON) control the output of this pin, which may be tristated, output the crystal input frequency (X1), or output the power save frequency (internal processor frequency after divisor). **clkouta** can be used as a full speed clock source in power-save mode. The A.C. timing specifications that are clock-related refer to **clkouta**, which remains active during reset and hold conditions.

clkoutb – Clock Output B (synchronous output)
This pin is an additional clock out put to the system**.** Bits 11, 10, and 2-0 of the Power-Save Control register (PDCON) control the output of this pin, which may be tristated, output the PLL frequency, or may output the power save frequency (internal processor frequency after divisor). **clkoutb** remains active during reset and hold conditions.

den_n (pio5) – Data Enable Strobe (synchronous output with tristate)
This pin provides an output enable to an external bus data bus transmitter or receiver. This signal is asserted during I/O, memory, and interrupt acknowledge processes and is deasserted when **dt/r_n** undergoes a change of state. It is tristated for a bus hold or reset.

drq1-drq (pio12-pio13) – DMA Requests (synchronous level-sensitive inputs)
drq0 – An external device that is ready for DMA channel 1 or 0 to carry out a transfer indicates to the microcontroller this readiness on these pins. They are level triggered, internally synchronized, not latched, and must remain asserted until dealt with.

dt/r_n (pio4) – Data Transmit or Receive (synchronous output with tristate)
The microntroller transmits data when **dt/r_n** is pulled high and receives data when this pin is pulled low. It floats during a reset or bus hold condition.

gnd – Ground
Six or seven pins, depending on package, connect the microcontroller to the system ground.

hlda – Bus Hold Acknowledge (synchronous output)
This pin is pulled high to signal the system that the microntroller has ceded control of the local bus, in response to a high on the **hold** signal by an external bus master, after the microcontroller has completed the current bus cycle. The assertion of **hlda** is accompanied by the tristating of **den_n, rd_n, wr_n, s2_n-s0_n, ad15-ad0, s6, a19-a0, bhe_n, whb_n, wlb_n,** and **dt/r_n**, followed by the driving high of the chip selects **ucs_n, lcs_n, mcs3_n - mcs0_n, pcs6_n – pcs5_n,** and **pcs3_n – pcs0_n**. The external bus master releases control of the local bus by the deassertion of **hold** that in turn induces the microcontroller to deassert the **hlda**. The microcontroller can take control of the bus if necessary (to execute a refresh for example), by deasserting **hlda** without the bus master first deasserting **hold**. This requires that the external bus master be able to deassert **hold** to permit the microcontroller to access the bus.

hold – Bus Hold Request (synchronous level-sensitive input)
This pin is pulled high to signal the microcontroller that the system requires control of the local bus.

hold latency time (time between the **hold** and **hlda**) depends on the current processor activity when the **hold** is received. A hold request is second only do a DMA refresh request in priority of processor activity requests. If a **hold** request is received at the moment a DMA transfer starts, the **hold** latency can be up to 4 bus cycles. (On the IA186EM only, this happens when a word transfer is taking place from an odd to an odd address). This means that the latency may be 16 clock cycles without wait states. Furthermore, if lock transfers are being performed, then the latency time is increased by the during of the locked transfer.

int0 – Maskable Interrupt Request 0 (asynchronous input)
The **int0** pin provides an indication that an interrupt request has occurred, and provided that **int0** is not masked, program execution will continue at the location specified by the INT0 vector in the interrupt vector table. Although interrupt requests are asynchronous, they are synchronized internally and may be edge- or level-triggered. The assertion of the interrupt request must be maintained until it is handled, to ensure that it is recognized.

int1/select_n – Maskable Interrupt Request 1/Slave Select (both are asynchronous inputs)
int1 - The **int1** pin provides an indication that an interrupt request has occurred, and provided that **int1** is not masked, program execution will continue at the location specified by the INT1 vector in the interrupt vector table. Although interrupt requests are asynchronous, they are synchronized internally and may be edge- or level-triggered. The assertion of the interrupt request must be maintained until it is handled, to ensure that it is recognized.

select_n – This pin provides an indication to the microcontroller that an interrupt type has been placed on the address/data bus when the internal Interrupt Control Unit is slaved to an external interrupt controller. Before this occurs, however, the **int0** pin must have indicated an interrupt request has occurred.

int2/inta0_n (pio31) – Maskable Interrupt Request 2 (asynchronous input) / Interrupt Acknowledge 0 (synchronous output)
int2 - The **int2** pin provides an indication that an interrupt request has occurred, and provided that **int2** is not masked, program execution will continue at the location specified by the int2 vector in the interrupt vector table. Although interrupt requests are asynchronous, they are synchronized internally and may be edge- or level-triggered. The assertion of the interrupt request must be maintained until it is handled, to ensure that it is recognized. When **int0** is configured to be in cascade mode, **int2** changes its function to **inta0_n**.

inta0_n – this function indicates to the system that the microcontroller requires an interrupt type in response to the interrupt request **int0** when the microcontroller's Interrupt Control Unit is in cascade mode. The peripheral device that issued the interrupt must provide the interrupt type.

int3/inta1_n/irq – Maskable Interrupt Request 3 (asynchronous input) / Interrupt Acknowledge 1 (synchronous output) / Interrupt Acknowledge (synchronous output)
int3 - The **int3** pin provides an indication that an interrupt request has occurred, and provided that **int3** is not masked, program execution will continue at the location specified by the **int3** vector in the interrupt vector table. Although interrupt requests are asynchronous, they are synchronized internally and may be edge- or level-triggered. The assertion of the interrupt request must be maintained until it is handled, to ensure that it is recognized. When **int1** is configured to be in cascade mode, **int3** changes its function to **inta1_n**.

inta1_n – this function indicates to the system that the microcontroller requires an interrupt type in response to the interrupt request **int1** when the microcontroller's Interrupt Control Unit is in cascade mode. The peripheral device that issued the interrupt must provide the interrupt type.

irq – With the Interrupt Control Unit of the microcontroller in slave mode, the signal on this pin allows the microcontroller to output an interrupt request to the external master interrupt controller.

int4/pio30 – Maskable Interrupt Request 4 (asynchronous input)
int4 - The **int4** pin provides an indication that an interrupt request has occurred, and provided that **int4** is not masked, program execution will continue at the location specified by the **int4** vector in the interrupt vector table. Although interrupt requests are asynchronous, they are synchronized internally and may be edge- or level-triggered. The assertion of the interrupt request must be maintained until it is handled, to ensure that it is recognized.

lcs_n/once0_n – Lower Memory Chip Select (synchronous output with internal pull-up) / ONCE Mode Request (input)
lcs_n - The **lcs_n** pin provides an indication that a memory access is occurring to the lower memory block. The size of the Lower Memory Block and its base address are programmable, with the size adjustable up to 512 Kbytes. **lcs_n** is held high during bus hold.

once0_n – (ONCE – *ON C*ircuit *E*mulation). This pin and its companion pin once1_n define the microcontroller mode during reset. These two pins are sampled on the rising edge of **res_n** and if both are asserted low the microcontroller starts in ONCE mode, else it starts normally. In ONCE mode, all pins are tristated and remain so until a subsequent reset. To prevent the microcontroller from entering ONCE mode inadvertently, this pin has a weak pull-up that is only present during reset. Finally, this pin is not tristated during bus hold.

mcs2_n – mcs0_n (no pio - pio15 – pio 14) – Midrange Memory Chip Selects (synchronous outputs with internal pull-up)
mcs0_n - The **mcs2_n** and **mcs0_n** pins provide an indication that a memory access is in train to either the second or third midrange memory block. The size of the Midrange Memory Block and its base address are programmable. **mcs2_n – mcs0_n** are held high during bus hold and have weak pull-ups that are only present during reset.

mcs3_n/rfsh_n (pio25) – Midrange Memory Chip Select (synchronous output with internal pull-up) / Automatic Refresh (synchronous output)
mcs3_n - The **mcs3_n** pin provides an indication that a memory access is in train to the fourth region of the midrange memory block. The size of the Midrange Memory Block and its base address are programmable. **mcs3_n** is held high during bus hold and has a weak pull-up that is present only during reset.

rfsh_n – This signal is timed for auto refresh to PSRAM or DRAM devices. The refresh pulse is output only when the PSRAM or DRAM mode bit is set (EDRAM register bit 15). This pulse is of 1.5 clock pulse duration with the rest of the refresh cycle made up of a deassertion period such that the overall refresh time is met. Finally, this pin is not tristated during a bus hold.

nmi – Nonmaskable Interrupt (synchronous edge-sensitive input)
This is the highest priority interrupt signal and cannot be masked, unlike **int4 – int0**.
Program execution is transferred to the nonmaskable interrupt vector in the interrupt vector table, upon the assertion of this interrupt (transition from Low to High), and this interrupt is initiated at the next instruction boundary. For recognition to be assured the **nmi** pin must be held high for at least a **clkouta** period so that the transition from low to high is latched and synchronized internally. The interrupt will begin at the next instruction boundary.

The NMI is not involved in the priority resolution process that deals with the maskable interrupts, and does not have an associated interrupt flag. This allows for a new NMI request to interrupt an NMI service routine that is already underway. The interrupt flag IF is cleared, disabling the maskable interrupts, when an interrupt is taken by the processor. If, during the NMI service routine, the maskable interrupts are re-enabled, by use of STI instruction for example, the priority resolution of maskable interrupts will be unaffected by the servicing of the NMI. For this reason, it is strongly recommended that the NMI interrupt service routine does not enable the maskable interrupts.

pcs3_n - pcs0_n (pio19 – pio16) – Peripheral Chip Selects 3-0 (synchronous outputs)
These pins provide an indication that a memory access is under way for the corresponding region of the peripheral memory block (I/O or memory address space). The base address of the Peripheral memory block is programmable. **pcs3_n – pcs0_n** are held high during both bus hold and reset. These outputs are asserted with the **ad** address bus over a 256-byte range each.

pcs5_n/A1– Peripheral Chip Select 5 (synchronous output) / latched Address Bit 1 (synchronous output)
pcs5_n – This signal provides an indication that a memory access is under way for the sixth region of the peripheral memory block (I/O or memory address space). The base address of the Peripheral memory block is programmable. **pcs5_n** is held high during both bus hold and reset. This output is asserted with the **ad** address bus over a 256-byte range.

A1 – This pin provides and internally latched address bit 1 to the system when the EX bit (bit 7) in the MCS_n and PCS_n auxiliary (MPCS) register is 0. It retains its previously latched value during a bus hold.

pcs6_n/A2/ – Peripheral Chip Select 6 (synchronous output) / latched Address Bit 2 (synchronous output)
pcs6_n – This signal provides an indication that a memory access is under way for the seventh region of the peripheral memory block (I/O or memory address space). The base address of the Peripheral memory block is programmable. **pcs6_n** is held high during both bus hold and reset. This output is asserted with the **ad** address bus over a 256-byte range.

A2 – This pin provides and internally latched address bit 2 to the system when the EX bit (bit 7) in the MCS_n and PCS_n auxiliary (MPCS) register is 0. It retains its previously latched value during a bus hold.

pio31 – pio0
Programmable I/O Pins (asynchronous input/output open –drain)
32 individually programmable I/O pins are provided. See page 62.

rd_n - Read strobe (synchronous output with tristate)
This pin provides an indication to the system that a memory or I/O read cycle is under way. It will not to be asserted before the **ad** bus is floated during the address to data transition. **rd_n** is tristated during bus hold.

res_n - Reset (asynchronous level-sensitive input)
This pin forces a reset on the microcontroller. It has a Schmitt trigger to allow power-on reset generation via an RC network. When this signal is asserted, the microcontroller immediately terminates its present activity, clears its internal logic, and transfers CPU control to the reset address, FFFF0h.

res_n must be asserted for at least 1ms and may be asserted asynchronously to **clkouta** as it is synchronized internally. Furthermore, V_{cc} must be within specification and clkouta must be stable for more than four of its clock periods for the period that **res_n** is asserted.

The microcontroller starts to fetch instructions 6.5 **clkouta** clock periods after the deassertion of **res_n**.

rfsh2_n/aden_n - *IA188EM only* - Refresh 2 (synchronous output with tristate) / Address Enable (input with internal pull-up)
rfsh2_n – Indicates that a DRAM refresh cycle is being performed when it is asserted low. However, this is not valid in PSRAM mode where **mcs3_n/rfsh_n** is used instead.

aden_n – If this pin is held high during power-on reset, the **ad** bus (**ao15-ao8 & ad7-ad0**) is controlled during the address portion of the LCS and UCS bus cycles by the DA bit (bit 7) in the LCS and UCS registers. If the DA bit is 1, the address is accessed on the **a19-a0** pins reducing power consumption. The weak pull-up on this pin obviates the necessity of an external pull-up.

If this pin is held low during power-on reset, the ad bus is used for both addresses and data without regard for the setting of the DA bits. **rfsh2_n/aden_n** is sampled one crystal clock cycle after the rising edge of **res_n** and is tristated during bus holds and ONCE mode.

rxd (pio28) - Receive Data (asynchronous input)
This signal connects asynchronous serial receive data from the system to the asynchronous serial port.

s2_n-s0_n - Bus Cycle Status (synchronous outputs with tristate)
These three signals inform the system of the type of bus cycle is in progress. **s2_n** may be used to indicate whether the current access is to memory or I/O, and **s1_n** may be used to indicate whether data is being transmitted or received. These signals are tristated during bus hold and hold acknowledge.

The coding for these pins is shown in the following table.

s2_n	s1_n	s0_n	Bus Cycle
0	0	0	Interrupt acknowledge
0	0	1	Read data from I/O
0	1	0	Write data to I/O
0	1	1	Halt
1	0	0	Instruction fetch
1	0	1	Read data from memory
1	1	0	Write data to memory
1	1	1	None (passive)

s6/clkdiv2_n (pio29) - Bus Cycle Status Bit 6 (synchronous output) /Clock Divide by 2 (input with internal pull-up)
s6 - This signal is high during the second and remaining cycle periods, i.e. $t_2 - t_4$, indicating that a DMA–initiated bus cycle is under way. **s6** is tristated during bus hold or reset.

clkdiv2_n – The microcontroller enters clock divide-by-2 mode, if this signal is held low during power-on-reset. In this mode, the PLL is disabled and the processor receives the external clock divided by 2. Sampling of this pin occurs on the rising edge of **res_n**.
Should this pin be used as **pio29** configured as an input, care should be taken that it is not driven low during power-on-reset. This pin has an internal pull-up so it is not necessary to drive the pin high even though it defaults to an input PIO.

sclk – Serial Clock (synchronous outputs with tristate)
This pin provides a slave device with a synchronous serial clock permitting synchronization of the transmit and receive data exchanges between the slave and the microcontroller. **sclk** is the result of dividing the internal clock by 2, 4, 8, or 16 dependent on the contents of the Synchronous Serial Control (SSC) register bits 5-4. Accessing either the SSR of SSD registers activates the **sclk** for eight cycles. When **sclk** is not active the microcontroller hold is high.

sdata – Serial Data (synchronous inout)
This pin connects a slave device to synchronous serial transmit and receive data. The last value is maintained on this pin when it is inactive.

sden1 - sden0 – Serial Data Enables (synchronous outputs with tristate)
These pins facilitate the transfer of data on ports 1 and 0 of the Synchronous Serial Interface (SSI). Either **sden1** or **sden0** is asserted by the microcontroller at the start of the data transfer and is de-asserted it when the transfer is completed. These pins are held low by the microcontroller when they are inactive.

srdy/pio6 - Synchronous Ready (synchronous level-sensitive input)
This signal is an active high input synchronized to **clkouta** and indicates to the microcontroller that a data transfer will be completed by the addressed memory space or I/O device.

In contrast to the Asynchronous Ready (**ardy**), which requires internal synchronization, srdy permits easier system timing as it already synchronized. Tying **srdy** high will always assert this ready condition, whereas tying it low will give control to **ardy**.

tmrin0/pio11 - Timer Input 0 (synchronous edge-sensitive input)
This signal may be either a clock or control signal for the internal timer 0. The timer is incremented by the microcontroller after it synchronizes a rising edge of **tmrin0**. When not used, **tmrin0** must be tied high, or when used as **pio11** it is pulled up internally.

tmrin1/pio0 - Timer Input 1 (synchronous edge-sensitive input)
This signal may be either a clock or control signal for the internal timer 1. The timer is incremented by the microcontroller after it synchronizes a rising edge of **tmrin1**. When not used, **tmrin1** must be tied high, or when used as **pio0** it is pulled up internally.

tmrout0/pio10 - Timer Output 0 (synchronous output)
This signal provides the system with a single pulse or a continuous waveform with a programmable duty cycle. It is tristated during a bus hold or reset.

tmrout1/pio1 - Timer Output 1 (synchronous output)
This signal provides the system with a single pulse or a continuous waveform with a programmable duty cycle. It is tristated during a bus hold or reset.

txd/pio22 - Transmit Data (asynchronous output)
This pin provides the system with asynchronous serial transmit data from the serial port.

ucs_n/once1_n - Upper Memory Chip Select (synchronous output) / ONCE Mode Request 1 (input with internal pull-up)

ucs_n - This pin provides an indication that a memory access is in train to the upper memory block. The size of the Upper Memory Block and its base address are programmable, with the size adjustable up to 512 Kbytes. **ucs_n** is held high during bus hold.

After power-on-reset**, ucs_n** is active low and program execution begins at FFFF0h with the default configuration of the **ucs_n** chip select is for 64 Kbytes memory range from F0000h to FFFFFh.

once1_n – (ONCE – *ON C*ircuit *E*mulation). This pin and its companion pin once0_n define the microcontroller mode during reset. These two pins are sampled on the rising edge of **res_n** and if both are asserted low the microcontroller starts in ONCE mode, else it starts normally. In ONCE mode all pins are tristated and remain so until a subsequent reset. To prevent the microcontroller from entering ONCE mode inadvertently, this pin has a weak pull-up that is only present during reset. Finally, this pin is not tristated during bus hold.

uzi_n/pio26 – Upper Zero Indicate (synchronous output)
This pin allows the designer to 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 (**ad15-ad10** on the AI186EM and **ao15-ao10** on the AI188EM). **uzi_n** is the logical OR of the inverted **a19-a16** bits. It asserts in the first period of a bus cycle and is held throughout the cycle.

At reset **uzi_n** should be pulled high or should be allowed to float. If this pin is pulled low at reset, the microcontroller enters a reserved clock test mode.

v_{cc} – Power Supply (input)
These pins supply power (+5V) to the microcontroller.

whb_n – Write High Byte - *IA186EM only* - (synchronous output with tristate)
This pin and **wlb_n** provide an indication to the system of which bytes of the data bus (upper, lower or both) are taking part in a write cycle. **whb_n** is asserted with **ad15_ad8** and is the logical OR of **bhe_n** and **wr_n**. It is tristated during reset.

wlb_n/wb_n – Write Low Byte - *IA186EM only* - (synchronous output with tristate) / Write Byte – *IA188EM only* - (synchronous output with tristate)
wlb_n - wlb_n and **whb_n** provide an indication to the system of which bytes of the data bus (upper, lower, or both) are taking part in a write cycle. **wlb_n** is asserted with **ad7_ad0** and is the logical OR of **ad0** and **wr_n**. It is tristated during reset.

wb_n – On the IA188EM microcontroller, **wb_n** provides an indication that a write to the bus is occurring. It shares the same early timing as that of the non-multiplexed address bus, and is associated with **ad7-ad0**. It is tristated during reset.

wr_n – Write Strobe (synchronous output)
This pin provides an indication to the system that the data currently on the bus is to be written to a memory or I/O device. It is tristated during a bus hold or reset.

x1 – Crystal Input (input)
This pin and **x2** are the connections for a fundamental mode or third-overtone, parallel-resonant crystal used by the internal oscillator circuit. An external clock source for the microcontroller is connected to **x1** while the **x2** pin is left unconnected.

x2 – Crystal Input (input)
This pin and **x1** are the connections for a fundamental mode or third-overtone, parallel-resonant crystal used by the internal oscillator circuit. An external clock source for the microcontroller is connected to **x1** while the **x2** pin is left unconnected.

Pins Used by Emulators

The following pins are used by emulators:
a19-a0
ao15-ao8
ad7-ad0
ale
bhe_n/aden_n (on the AI186EM)
clkouta
rfsh2_n/aden_n (on the AI188EM)
rd_n
s2_n-s0_n
s6/lock_n/clkdiv2_n
uzi_n

Emulators require that **s6/lock_n/clkdiv2_n** and **uzi_n** be configured as their normal functions, i.e. as **s6** and **uzi_n** respectively. Holding **bhe_n/aden_n** (AI186EM) or **rfsh_n/aden_n** (AI188EM) low during the rising edge of **res_n, s6** and **uzi_n** will be configured in their normal functions instead of as PIOs, at reset.

Instruction Set Summary

NOTE
Key to abbreviations appears at the end of the table.

Instruction		Opcode - Hex			Clock Cycles		Flags Affected								
Mnemonic	Description	byte 1	byte 2	byte 3-6	IA186	IA188	O	D	I	T	S	Z	A	P	C
AAA	ASCII adjust AL after add	37	-	-	8	8	U	-	-	-	U	U	R	U	R
AAD	ASCII adjust AX before divide.	D5	0A	-	15	15	U	-	-	-	R	R	U	R	U
AAM	ASCII adjust AL after multiply	D4	0A	-	19	19	U	-	-	-	R	R	U	R	U
AAS	ASCII adjust AL after subtract	3F	-	-	7	7	U	-	-	-	U	U	R	U	R
ADC	Add imm8 to AL with carry	14	ib	-	3	3	R	-	-	-	R	R	R	R	R
	Add imm16 to AX with carry	15	iw	-	4	4									
	Add imm8 to r/m8 with carry	80	/2 ib	-	4/16	4/16									
	Add imm16 to r/m16 with carry	81	/2 iw	-	4/16	4/20									
	Add sign extended imm8 to r/m16 with carry	83	/2 ib	-	4/16	4/20									
	Add byte reg to r/m8 with carry	10	/r	-	3/10	3/10									
	Add word reg to r/m16 with carry	11	/r	-	3/10	3/14									
	Add r/m8 to byte reg with carry	12	/r	-	3/10	3/10									
	Add r/m16 to word reg with carry	13	/r	-	3/10	3/14									
ADD	Add imm8 to AL	04	ib	-	3	3	R	-	-	-	R	R	R	R	R
	Add imm16 to AX	05	iw	-	4	4									
	Add imm8 to r/m8	80	/0 ib	-	4/16	4/16									
	Add imm16 to r/m16	81	/0 iw	-	4/16	4/20									
	Add sign extended imm8 to r/m16	83	/0 ib	-	4/16	4/20									
	Add byte reg. to r/m8	00	/r	-	3/10	3/10									
	Add word reg. to r/m16	01	/r	-	3/10	3/14									
	Add r/m8 to byte reg	02	/r	-	3/10	3/10									
	Add r/m16 to word reg	03	/r	-	3/10	3/14									
AND	And imm8 with AL	24	ib		3	3	0	-	-	-	R	R	U	R	0
	And imm16 with AX	25	iw		4	4									
	And imm8 with r/m8	80	/4 ib		4/16	4/16									
	And imm16 with r/m16	81	/4 iw		4/16	4/20									
	And sign-extended imm8 with r/m16	83	/4 ib		4/16	4/20									
	And byte reg. with r/m8	20	/r		3/10	3/10									

	And word reg. with r/m16	21	/r		3/10	3/14									
	And r/m8 with byte reg	22	/r		3/10	3/10									
	And r/m16 with word reg	23	/r		3/10	3/14									

Instruction		Opcode - Hex			Clock Cycles		Flags Affected								
Mnemonic	Description	byte 1	byte 2	byte 3-6	IA186	IA188	O	D	I	T	S	Z	A	P	C
BOUND	Check array index against bounds	62	/r	-	33-35	33-35	-	-	-	-	-	-	-	-	-
CALL	Call near, disp relative to next instruction	E8	cw	-	15	19	-	-	-	-	-	-	-	-	-
	Call near, reg indirect mem	FF	/2	-	13/19	17/27									
	Call far to full address given	9A	cd	-	23	31									
	Call far to address at m16:16 word	FF	/3	-	38	54									
CBW	Convert byte integer to word	98	-	-	2	2	-	-	-	-	-	-	-	-	-
CLC	Clear carry flag	F8	-	-	2	2	-	-	-	-	-	-	-	-	-
CLD	Clear direction flag	FC	-	-	2	2	-	0	-	-	-	-	-	-	-
CLI	Clear interrupt-enable flag	FA	-	-	2	2	-	-	0	-	-	-	-	-	-
CMC	Complement carry flag	F5	-	-	2	2	-	-	-	-	-	-	-	-	R
CMP	Compare imm8 to AL	3C	ib		3	3	R	-	-	-	R	R	R	R	R
	Compare imm16 to AX	3D	iw	-	4	4									
	Compare imm8 to r/m8	80	/7	ib	3/10	3/10									
	Compare imm16 to r/m16	81	/7	iw	3/10	3/14									
	Compare sign-extended imm8 to r/m16	83	/7	ib	3/10	3/14									
	Compare byte reg to r/m8	38	/r		3/10	3/10									
	Compare word reg to r/m16	39	/r	-	3/10	3/14									
	Compare r/m8 to byte reg	3A	/r	-	3/10	3/10									
	Compare r/m16 to word reg	3B	/r	-	3/10	3/14									
CMPS	Compare byte ES: [DI] to byte segment: [SI]	A6	-	-	22	22	R	-	-	-	R	R	R	R	R
	Compare word ES: [DI] to word segment: [SI]	A7	-	-	22	26									
CMPSB	Compare byte ES: [DI] to byte DS: [SI]	A6	-	-	22	22	R	-	-	-	R	R	R	R	R
CMPSW	Compare word ES: [DI] to word DS: [SI]	A7	-	-	22	26	R	-	-	-	R	R	R	R	R
CS	CS segment reg override prefix	2E	-	-	-	-	-	-	-	-	-	-	-	-	-
CWD	Convert word integer to double word	99	-	-	4	4	-	-	-	-	-	-	-	-	-
DAA	Decimal adjust AL after addition	27	-	-	4	4	U	-	-	-	R	R	R	R	R
DAS	Decimal adjust AL after subtraction	2F	-	-	4	4	U	-	-	-	R	R	R	R	R
DEC	Subtract 1 from r/m8	FE	/1	-	3/15	3/15	R	-	-	-	R	R	R	R	R
	Subtract 1 from r/m16	FF	/1	-	3/15	3/19									
	Subtract 1 from word reg	48+ rw			3	3									
DIV	Divide unsigned numbers	F6	mod 110 r/m	-	29/35	29/35	U	-	-	-	U	U	U	U	U
DS	DS segment override prefix	3E	-	-	-	-	-	-	-	-	-	-	-	-	-

Instruction		Opcode - Hex			Clock Cycles		Flags Affected								
Mnemonic	Description	byte 1	byte 2	byte 3-6	IA186	IA188	O	D	I	T	S	Z	A	P	C
ENTER	Create stack frame for nested procedure	C8	iw ib	-	22+16 (n-1)	26+20 (n-1)	-	-	-	-	-	-	-	-	-
	Create stack frame for non-nested procedure	C8	iw 00	-	15	19									
	Create stack frame for nested procedure	C8	iw 01	-	25	29									
ES	ES segment reg override prefix	26	-	-	-	-	-	-	-	-	-	-	-	-	-
ESC	Escape - takes a Trap 7	D8	/0	-	-	-	-	-	0	0	-	-	-	-	-
	Escape - takes a Trap 7	D9	/1	-	-	-									
	Escape - takes a Trap 7	DA	/2	-	-	-									
	Escape - takes a Trap 7	DB	/3	-	-	-									
	Escape - takes a Trap 7	DC	/4	-	-	-									
	Escape - takes a Trap 7	DD	/5	-	-	-									
	Escape - takes a Trap 7	DE	/6	-	-	-									
	Escape - takes a Trap 7	DF	/7	-	-	-									
HLT	Suspend instruction execution	F4	-	-	2	2	-	-	-	-	-	-	-	-	-
IDIV	Divide Integers AL = AX/(r/m8); AH = remainder	F6	/7	-	44-52 / 50-58	44-52 / 50-58	U	-	-	-	U	U	U	U	U
	Divide Integers AX = DX : AX/(r/m16); DX = remainder	F7	/7	-	53-61 / 59-67	53-61 / 63-71									
IMUL	Multiply Integers AX=(r/m8)*Al	F6	/5	-	25-28 / 31-34	25-28 / 31-34	R	-	-	-	U	U	U	U	R
	Multiply Integers DX=(r/m16)*AX	F7	/5	-	34-37 / 40-43	34-37 / 44-47									
	Multiply Integers (word reg) = (r/m16)*(sign-ext. byte integer)	6B	/r ib	-	22-25	22-25									
	Multiply Integers (word reg) = (word reg)*(sign-ext. byte integer)	6B	/r ib	-	22-25	22-25									
	Multiply Integers (word reg) = (r/m16)*(sign-ext. byte integer)	69	/r iw	-	29-32	29-32									
	Multiply Integers (word reg) = (word reg)*(sign-ext. byte integer)	69	/r iw	-	29-32	29-32									

Instruction		Opcode - Hex			Clock Cycles		Flags Affected								
Mnemonic	Description	byte 1	byte 2	byte 3-6	IA186	IA188	O	D	I	T	S	Z	A	P	C
IN	Input byte from imm port to AL	E4	ib	-	10	10	-	-	-	-	-	-	-	-	-
	Input word from imm port to AX	E5	ib	-	10	14									
	Input byte from port in DX to AL	EC		-	8	8									
	Input word from port in DX to AX	ED		-	8	12									
INC	Increment r/m8 by 1	FE	/0	-	3/15	3/15	R	-	-	-	R	R	R	R	R
	Increment r/m16 by 1	FF	/0	-	3/15	3/19									
	Increment word reg by 1	40+ rw	-	-	3	3									
INS	Input byte from port in DX to ES : [DI]	6C	-	-	14	14	-	-	-	-	-	-	-	-	-
	Input word from port in DX to ES : [DI]	6D													
INSB	Input byte from port in DX to ES : [DI]	6C													
INSW	Input word from port in DX to ES : [DI]	6D													
INT 3	Generate interrupt 3 (trap to debug)	CC	-	-	45	45	-	-	0	0	-	-	-	-	-
INT	Generate type of interrupt specified by imm8	CD	ib	-	47	47									
INTO	Generate interrupt 4 if Overflow Flag (O) is 1	CE	-	-	48, 4	48, 4									
IRET	Interrupt return	CF	-	-	28	28	Restores value of flags reg that was stored on the stack when the interrupt was taken								
JA	Jump short if above (C & Z = 0)	77	cb	-	13, 4	13, 4	-	-	-	-	-	-	-	-	-
JNBE	Jump short if not below or equal														
JAE	Jump short if above or equal(C=0)	73	cb	-	13, 4	13, 4	-	-	-	-	-	-	-	-	-
JNB	Jump short if not below (C=0)														
JNC	Jump short if not carry (C=0)														
JB	Jump short if below (C=1)	72	cb	-	13, 4	13, 4	-	-	-	-	-	-	-	-	-
JC	Jump short if carry (C=1)														
JNAE	Jump short if not above or equal (C=1)														
JBE	Jump short if below or equal (C & Z = 0)	76	cb	-	13, 4	13, 4	-	-	-	-	-	-	-	-	-
JNA	Jump short if not above (C & Z = 0)														
JCXZ	Jump short if CX reg is 0	E3	cb	-	15,5	15,5	-	-	-	-	-	-	-	-	-
JE	Jump short if equal (Z=1)	74	cb	-	13, 4	13, 4	-	-	-	-	-	-	-	-	-
JZ	Jump short if 0 (Z=1)														

Instruction		Opcode - Hex			Clock Cycles		Flags Affected								
Mnemonic	Description	byte 1	byte 2	byte 3-6	IA186	IA188	O	D	I	T	S	Z	A	P	C
JG	Jump short if greater (Z & S = O)	7F	cb	-	13, 4	13, 4	-	-	-	-	-	-	-	-	-
JNLE	Jump short if not less or equal (Z & S = O)														
JGE	Jump short if greater or equal (S=O)	7D	cb	-	13, 4	13, 4	-	-	-	-	-	-	-	-	-
JNL	Jump short if not less (S = O)														
JLE	Jump short if less or equal (Z & S = O)	7E	cb	-	13, 4	13, 4	-	-	-	-	-	-	-	-	-
JNG	Jump short if not greater (Z & S = O)														
JMP	Jump short direct, disp relative to next instruction	EB	cb	-	14	14	-	-	-	-	-	-	-	-	-
	Jump near direct, disp relative to next instruction	E9	cw	-	14	14									
	Jump near indirect	FF	/4	-	11/17	11/21									
	Jump far direct to doubleword imm address	EA	cd	-	14	14									
	Jump m16: 16 indirect and far	FF	/5	-	26	34									
JNE	Jump short if not equal (Z=0)	75	cb	-	13, 4	13, 4	-	-	-	-	-	-	-	-	-
JNZ	Jump short if not zero (Z=0)														
JNO	Jump short if not overflow (O=1)	71	cb	-	13, 4	13, 4	-	-	-	-	-	-	-	-	-
JNP	Jump short if not parity (P=0)	7B	cb	-	13, 4	13, 4	-	-	-	-	-	-	-	-	-
JPO	Jump short if parity odd (P=0)														
JNS	Jump short if not sign (S=0)	79	cb	-	13, 4	13, 4	-	-	-	-	-	-	-	-	-
JO	Jump short if overflow (O=1)	70	cb	-	13, 4	13, 4	-	-	-	-	-	-	-	-	-
JP	Jump short if parity (P=1)	7A	cb	-	13, 4	13, 4	-	-	-	-	-	-	-	-	-
JPE	Jump short if parity (P=1)														
JS	Jump short if sign (S=1)	78	cb	-	13, 4	13, 4	-	-	-	-	-	-	-	-	-
LAHF	Load AH with low byte of flags reg	9F	-	-	2	2	-	-	-	-	-	-	-	-	-
LDS	Load DS:r16 with segment :offset from memory	C5	/r	-	18	26	-	-	-	-	-	-	-	-	-
LEA	Load offset for m16 word in 16-bit reg	8D	/r	-	6	6	-	-	-	-	-	-	-	-	-
LEAVE	Destroy procedure stack frame	C9	-	-	8	8	-	-	-	-	-	-	-	-	-
LES	Load ES:r16 with segment offset from memory	C4	/r	-	18	26	-	-	-	-	-	-	-	-	-
LOCK	Asserts lock_n during an instruction execution	F0	-	-	1	1	-	-	-	-	-	-	-	-	-
LODS	Load byte segment :[SI] in AL	AC	-	-	12	12	-	-	-	-	-	-	-	-	-
	Load word segment :[SI] in AX	AD			12	16									

Instruction		Opcode - Hex			Clock Cycles		Flags Affected								
Mnemonic	Description	byte 1	byte 2	byte 3-6	IA186	IA188	O	D	I	T	S	Z	A	P	C
LODSB	Load byte DS: [SI] in AL	AC			12	12									
LODSW	Load word DS: [SI] in AX	AD			12	16									
LOOP	Decrement count ; jump short if CX ≠ 0	E2	-	-	16, 6	16, 6	-	-	-	-	-	-	-	-	-
LOOPE	Decrement count ; jump short if CX ≠ 0 and Z = 1	E1	cb	-											
LOOPZ	Decrement count ; jump short if CX ≠ 0 and Z = 1														
LOOPNE	Decrement count ; jump short if CX ≠ 0 and Z = 0	E0	cb	-	16, 6	16, 6	-	-	-	-	-	-	-	-	-
LOOPNZ	Decrement count ; jump short if CX ≠ 0 and Z = 0														
MOV	Copy reg to r/m8	88	/r	-	2/12	2/12	-	-	-	-	-	-	-	-	-
	Copy reg to r/m16	89	/r	-	2/12	2/16									
	Copy r/m8 to reg	8A	/r	-	2/9	2/9									
	Copy r/m16 to reg	8B	/r	-	2/9	2/13									
	Copy segment reg to r/m16	8C	/sr	-	2/11	2/15									
	Copy r/m16 to segment reg	8E	/sr	-	2/9	2/13									
	Copy byte at segment offset to AL	A0	-	-	8	8									
	Copy word at segment offset to AX	A1	-	-	8	12									
	Copy AL to byte at segment offset	A2	-	-	9	9									
	Copy AX to word at segment offset	A3	-	-	9	13									
	Copy imm8 to reg	B0 +rb	-	-	3	3									
	Copy imm16 to reg	B8 +rw			3	4									
	Copy imm8 to r/m8	C6	/0	-	12	12									
	Copy imm16 to r/m16	C7	/0	-	12	13									
MOVS	Copy byte segment [SI] to ES:[DI]	A4	-	-	14	14	-	-	-	-	-	-	-	-	-
	Copy word segment [SI] to ES:[DI]	A5	-	-	14	18									
MOVSB	Copy byte DS:[SI] to ES:[DI]	A4	-	-	14	14									
MOVSW	Copy word DS:[SI] to ES:[DI]	A5	-	-	14	18									
MUL	AX = (r/m8) * AL	F6	/4	-	26-28 / 32-34	26-28 / 32-34	R	-	-	-	-	-	-	-	R
	DX :: AX = (r/m16) * AX	F7	/4		35-37 / 41-43	35-37 / 45-47									
NEG	Perform 2's complement negation of r/m8	F6	/3	-	3/10	3/10	R	-	-	-	R	R	R	R	R
	Perform 2's complement negation of r/m16	F7	/3	-	3/10	3/14									

Instruction	Opcode - Hex	Clock Cycles	Flags Affected

Mnemonic	Description	byte 1	byte 2	byte 3-6	IA186	IA188	O	D	I	T	S	Z	A	P	C
NOP	Perform no operation	90	-	-	3	3	-	-	-	-	-	-	-	-	-
NOT	Complement each bit in r/m8	F6	/2	-	3/10	3/10	-	-	-	-	-	-	-	-	-
	Complement each bit in r/m16	F7	/2		3/10	3/14									
OR	OR imm8 with AL	0C	ib	-	3	3	0	-	-	-	R	R	U	R	0
	OR imm16 with AX	0D	iw		4	4									
	OR imm8 with r/m8	80	/1 ib	-	4/16	4/16									
	OR imm16 with r/m16	81	/1 iw	-	4/16	4/20									
	OR imm8 with r/m16	83	/1 ib	-	4/16	4/20									
	OR byte reg with r/m8	08	/r	-	3/10	3/10									
	OR word reg with r/m16	09	/r	-	3/10	3/14									
	OR r/m8 with byte reg	0A	/r	-	3/10	3/10									
	OR r/m16 with word reg	0B	/r	-	3/10	3/14									
OUT	Output AL to imm port	E6	ib	-	9	9	-	-	-	-	-	-	-	-	-
	Output AX to imm port	E7	ib	-	9	13									
	Output AL to port in DX	EE	-	-	7	7									
	Output AX to port in DX	EF	-	-	7	11									
OUTS	Output byte DS:[SI] to port in DX	6E	-	-	14	14	-	-	-	-	-	-	-	-	-
	Output word DS:[SI] to port in DX	6F	-	-											
OUTSB	Output byte DS:[SI] to port in DX	6E	-	-											
OUTSW	Output word DS:[SI] to port in DX	6F	-	-											
POP	Pop top word of stack into memory word	8F	/0	-	20	24	-	-	-	-	-	-	-	-	-
	Pop top word of stack into word reg	58+ rw	-	-	10	14									
	Pop top word of stack into DS	1F	-	-	8	12									
	Pop top word of stack into ES	07	-	-											
	Pop top word of stack into SS	17	-	-											
POPA	Pop DI, SI, BP, BX, DX, CX, & AX	61	-	-	51	83	Values in word at top of stack are copied into FLAGS reg bits								
POPF	Pop top word of stack into Processor Status Flags reg	9D	-	-	8	12									
PUSH	Push memory word onto stack	FF	/6	-	16	20	-	-	-	-	-	-	-	-	-
	Push reg word onto stack	50+ rw	-	-	10	14									
	Push sign-extended imm8 onto stack	6A	-	-	10	14									
	Push imm16 onto stack	68	-	-	10	14									
	Push CS onto stack	0E	-	-	9	13									
	Push SS onto stack	16	-	-	9	13									
	Push DS onto stack	1E	-	-	9	13									
	Push ES onto stack	06	-	-	9	13									

Instruction		Opcode - Hex			Clock Cycles		Flags Affected								
Mnemonic	Description	byte 1	byte 2	byte 3-6	IA186	IA188	O	D	I	T	S	Z	A	P	C
PUSHA	Push AX, CX, DX, BX, original SP, BP, SI, and DI	60	-	-	36	68	-	-	-	-	-	-	-	-	-
PUSHF	Push Processor Status Flags reg	9C	-	-	9	13	-	-	-	-	-	-	-	-	-
RCL	Rotate 9 bits of C and r/m8 left once	D0	/2	-	2/15	2/15	-	-	-	-	-	-	-	-	-
	Rotate 9 bits of C and r/m8 left CL times	D2	/2	-	5+n/ 17+n	5+n/ 17+n									
	Rotate 9 bits of C and r/m8 left imm8 times	C0	/2 ib	-	5+n/ 17+n	5+n/ 17+n									
	Rotate 17 bits of C and r/m16 left once	D1	/2	-	2/15	2/15									
	Rotate 17 bits of C and r/m16 left CL times	D3	/2	-	5+n/ 17+n	5+n/ 17+n									
	Rotate 17 bits of C and r/m16 left imm8 times	C1	/2 ib	-	5+n/ 17+n	5+n/ 17+n									
RCR	Rotate 9 bits of C and r/m8 right once	D0	/3	-	2/15	2/15	-	-	-	-	-	-	-	-	-
	Rotate 9 bits of C and r/m8 right CL times	D2	/3	-	5+n/ 17+n	5+n/ 17+n									
	Rotate 9 bits of C and r/m8 right imm8 times	C0	/3 ib	-	5+n/ 17+n	5+n/ 17+n									
	Rotate 17 bits of C and r/m16 right once	D1	/3	-	2/15	2/15									
	Rotate 17 bits of C and r/m16 right CL times	D3	/3	-	5+n/ 17+n	5+n/ 17+n									
	Rotate 17 bits of C and r/m16 right imm8 times	75	/3 ib	-	5+n/ 17+n	5+n/ 17+n									
REP INS	Input CX bytes from port in DX to ES : [DI]	F3	6C	-	8+8n	8+8n	-	-	-	-	-	-	-	-	-
	Input CX bytes from port in DX to ES : [DI]	F3	6D	-	8+8n	12+8n									
REP LODS	Load CX bytes from segment :[SI] in AL	F3	AC	-	6+11n	6+11n	-	-	-	-	-	-	-	-	-
	Load CX words from segment :[SI] in AX	F3	AD	-	6+11n	10+ 11n									
REP MOVS	Copy CX bytes from segments : [SI] to ES:[DI]	F3	A4	-	8+8n	8+8n	-	-	-	-	-	-	-	-	-
	Copy CX words from segments : [SI] to ES:[DI]	F3	A5	-	8+8n	12+8n									
REP OUTS	Output CX bytes from DS:[SI] to port in DX	F3	6E	-	8+8n	8+8n	-	-	-	-	-	-	-	-	-
	Output CX bytes from DS:[SI] to port in DX	F3	6F	-	8+8n	12+8n									

Instruction		Opcode - Hex			Clock Cycles		Flags Affected								
Mnemonic	Description	byte 1	byte 2	byte 3-6	IA186	IA188	O	D	I	T	S	Z	A	P	C
REP STOS	Fill CX bytes at ES:[DI] with AL	F3	AA	-	8+8n	8+8n	-	-	-	-	-	-	-	-	-
	Fill CX words at ES:[DI] with AL	F3	AB	-	8+8n	12+8n									
REPE CMPS	Find non-matching bytes in ES:[DI] and segment :[SI]	F3	A6	-	5+22n	5+22n									
	Find non-matching words in ES:[DI] and segment :[SI]	F3	A7	-	5+22n	9+22n									
REPE SCAS	Find non-AL byte starting at ES:[DI]	F3	AE	-	5+15n	5+15n									
	Find non-AX word starting at ES:[DI]	F3	AF	-	5+15n	9+15n	-	-	-	-	-	-	-	-	-
REPZ CMPS	Find non-matching bytes in ES:DI and segment :[SI]	F3	A6	-	5+22n	5+22n									
	Find non-matching words in ES:DI and segment :[SI]	F3	A7	-	5+22n	9+22n									
REPZ SCAS	Find non-AL byte starting at ES:DI	F3	AE	-	5+15n	5+15n									
	Find non-AX word starting at ES:DI	F3	AF	-	5+15n	9+15n									
REPNE CMPS	Find matching bytes in ES:[DI] and segment :[SI]	F2	A6	-	5+22n	5+22n									
	Find matching words in ES:[DI] and segment :[SI]	F2	A7	-	5+22n	9+22n									
REPNZ CMPS	Find AL byte starting at ES:[DI]	F2	A6	-	5+22n	5+22n									
	Find AX word starting at ES:[DI]	F2	A7	-	5+22n	9+22n	-	-	-	-	-	-	-	-	-
REPNE SCAS	Find matching bytes in ES:DI and segment :[SI]	F2	AE	-	5+15n	5+15n									
	Find matching words in ES:DI and segment :[SI]	F2	AF	-	5+15n	9+15n									
REPNZ SCAS	Find AL byte starting at ES:DI	F2	AE	-	5+15n	5+15n									
	Find AX word starting at ES:DI	F2	AF	-	5+15n	9+15n									
RET	Return near to calling procedure	C3			16	20									
	Return far to calling procedure	CB	data low	data high	22	30									
	Return near; pop imm16 parameters	C2			18	22	-	-	-	-	-	-	-	-	-
	Return far; pop imm16 parameters	CA	data low	data high	25	33									
ROL	Rotate 8 bits of r/m8 left once	D0	/0	-	2/15	2/15									
	Rotate 8 bits or r/m8 left CL times	D2	/0	-	5+n/ 17+n	5+n/ 17+n	U	-	-	-	-	-	-	-	R
	Rotate 8 bits or r/m8 left imm8 times	C0	/0 ib	data 8	5+n/ 17+n	5+n/ 17+n									
	Rotate 16 bits of r/m8 left once	D1	/0	-	2/15	2/15									

Instruction		Opcode - Hex			Clock Cycles		Flags Affected								
Mnemonic	Description	byte 1	byte 2	byte 3-6	IA186	IA188	O	D	I	T	S	Z	A	P	C
ROL	Rotate 16 bits or r/m8 left CL times	D3	/0	-	5+n/ 17+n	5+n/ 17+n	U	-	-	-	-	-	-	-	R
	Rotate 16 bits or r/m8 left imm8 times	C1	/0 ib	data 8	5+n/ 17+n	5+n/ 17+n									
ROR	Rotate 8 bits of r/m8 right once	D0	/1	-	2/15	2/15	U	-	-	-	-	-	-	-	R
	Rotate 8 bits or r/m8 right CL times	D2	/1	-	5+n/ 17+n	5+n/ 17+n									
	Rotate 8 bits or r/m8 right imm8 times	C0	/1 ib	data 8	5+n/ 17+n	5+n/ 17+n									
	Rotate 16 bits of r/m8 right once	D1	/1	-	2/15	2/15									
	Rotate 16 bits or r/m8 right CL times	D3	/1	-	5+n/ 17+n	5+n/ 17+n									
	Rotate 16 bits or r/m8 right imm8 times	C1	/1 ib	data 8	5+n/ 17+n	5+n/ 17+n									
SAHF	Show AH in low byte of the Status Flags reg	9E	-	-	3	3	-	-	-	-	R	R	R	R	R
SAL/SHL	Multiply r/m8 by 2, once	D0	/4	-	2/15	2/15	U	-	-	-	-	R	R	R	R
	Multiply r/m8 by 2, CL times	D2	/4	-	5+n/ 17+n	5+n/ 17+n									
	Multiply r/m8 by 2, imm8 times	C0	/4 ib	data 8	5+n/ 17+n	5+n/ 17+n									
	Multiply r/m16 by 2, once	D1	/4	-	2/15	2/15									
	Multiply r/m16 by 2, CL times	D3	/4	-	5+n/ 17+n	5+n/ 17+n									
	Multiply r/m16 by 2, imm8 times	C1	/4 ib	data 8	5+n/ 17+n	5+n/ 17+n									
	Multiply r/m8 by 2, once	D0	/4	-	2/15	2/15									
	Multiply r/m8 by 2, CL times	D2	/4	-	5+n/ 17+n	5+n/ 17+n									
	Multiply r/m8 by 2, imm8 times	C0	/4 ib	data 8	5+n/ 17+n	5+n/ 17+n									
	Multiply r/m16 by 2, once	D1	/4	-	2/15	2/15									
	Multiply r/m16 by 2, CL times	D3	/4	-	5+n/ 17+n	5+n/ 17+n									
	Multiply r/m16 by 2, imm8 times	C1	/4 ib	data 8	5+n/ 17+n	5+n/ 17+n									

Instruction		Opcode - Hex			Clock Cycles		Flags Affected								
Mnemonic	Description	byte 1	byte 2	byte 3-6	IA186	IA188	O	D	I	T	S	Z	A	P	C
SAR	Perform a signed division of r/m8 by 2, once	D0	/7	-	2/15	2/15	U	-	-	-	R	R	U	R	R
	Perform a signed division of r/m8 by 2, CL times	D2	/7	-	5+n/ 17+n	5+n/ 17+n									
	Perform a signed division of r/m8 by 2, imm8 times	C0	/7 ib	data 8	5+n/ 17+n	5+n/ 17+n									
	Perform a signed division of r/m16 by 2, once	D1	/7	-	2/15	2/15									
	Perform a signed division of r/m16 by 2, CI times	D3	/7	-	5+n/ 17+n	5+n/ 17+n									
	Perform a signed division of r/m16 by 2, imm8 times	C1	/7 ib	data 8	5+n/ 17+n	5+n/ 17+n									
SBB	Subtract imm8 from AI with borrow	1C	ib	-	3	3	R	-	-	-	R	R	R	R	R
	Subtract imm16 from AX with borrow	1D	iw	data 8	4	4									
	Subtract imm8 from r/m8 with borrow	80	/3 ib	-	4/16	4/16									
	Subtract imm16 from r/m16 with borrow	81	/3 iw	-	4/16	4/20									
	Subtract sign-extended imm8 from r/m16 with borrow	83	/3 ib	-	4/16	4/20									
	Subtract byte reg from r/m8 with borrow	18	/r	data 8	3/10	3/10									
	Subtract word reg from r/m16 with borrow	19	/r	-	3/10	3/14									
	Subtract r/m8 from r/m8 with borrow	1A	/r	-	3/10	3/10									
	Subtract r/m8 reg from byte with borrow	1B	/r	data 8	3/10	3/14									
SCAS	Compare byte AL to ES:[DI]; update DI	AE	-	-	15	19	R	-	-	-	R	R	R	R	R
	Compare word AL to ES:[DI]; update DI	AF	-	-	15	19									
SCASB	Compare byte AL to ES:[DI]; update DI	AE	-	-	15	19									
SCASW	Compare word AL to ES:[DI]; update DI	AF	-	-	15	19									

Instruction		Opcode - Hex			Clock Cycles		Flags Affected								
Mnemonic	Description	byte 1	byte 2	byte 3-6	IA186	IA188	O	D	I	T	S	Z	A	P	C
SHR	Divide unsigned of r/m8 by 2, once	D0	/7	-	2/15	2/15	U	-	-	-	R	R	U	R	0
	Divide unsigned of r/m8 by 2, CL times	D2	/7	-	5+n/ 17+n	5+n/ 17+n									
	Divide unsigned of r/m8 by 2, imm8 times	C0	/7 ib	data 8	5+n/ 17+n	5+n/ 17+n									
	Divide unsigned of r/m16 by 2, once	D1	/7	-	2/15	2/15									
	Divide unsigned of r/m16 by 2, CL times	D3	/7	-	5+n/ 17+n	5+n/ 17+n									
	Divide unsigned of r/m16 by 2, imm8 times	C1	/7 ib	data 8	5+n/ 17+n	5+n/ 17+n									
SS	SS segment reg override prefix	36	-	-	-	-	-	-	-	-	-	-	-	-	-
STC	Set the Carry Flag to 1	F9	-		2	2	-	-	-	-	-	-	-	-	1
STD	Set the Direction Flag so the source Index (SI) and/or the Destination Index (DI) regs will decrement during string instructions	FD	-	-	2	2	-	1	-	-	-	-	-	-	-
STI	Enable maskable interrupts after the next instruction	FB	-	-	2	2	-	-	1	-	-	-	-	-	-
STOS	Store AL in byte ES:[DI]; update DI	AA	-	-	10	10	-	-	-	-	-	-	-	-	-
	Store AX in word ES:[DI]; update DI	AB	-	-	10	14									
STOSB	Store AL in byte ES:[DI]; update DI	AA	-	-	10	10									
STOSW	Store AX in word ES:[DI]; update DI	AB	-	-	10	14									
SUB	Subtract imm8 from AL	2C	ib	-	3	3	R	-	-	-	R	R	R	R	R
	Subtract imm16 from AX	2D	iw	-	4	4									
	Subtract imm8 from r/m8	80	/5 ib	-	4/16	4/16									
	Subtract imm16 from r/m16	81	/5 iw	-	4/16	4/20									
	Subtract sign-extended imm8 from r/m16	83	/5 ib	-	4/16	4/20									
	Subtract byte reg from r/m8	28	/r	-	3/10	3/10									
	Subtract word reg from r/m16	29	/r	-	3/10	3/14									
	Subtract r/m8 from byte reg	2A	/r	-	3/10	3/10									
	Subtract r/m16 from word reg	2B	/r	-	3/10	3/14									

Instruction		Opcode - Hex			Clock Cycles		Flags Affected								
Mnemonic	Description	byte 1	byte 2	byte 3-6	IA186	IA188	O	D	I	T	S	Z	A	P	C
TEST	AND imm8 with AL	A8	ib	-	3	3	0	-	-	-	R	R	U	R	0
	AND imm16 with AX	A9	iw	-	4	4									
	AND imm8 with r/m8	F6	/0 ib	data 8	4/10	4/10									
	AND imm16 with r/m16	F7	/0 iw	-	4/10	4/14									
	AND byte reg with r/m8	84	/r	-	3/10	3/10									
	AND word reg with r/m16	85	/r	data 8	3/10	3/14									
WAIT	Performs a NOP	9B	-	-	-	-	-	-	-	-	-	-	-	-	-
XCHG	Exchange word reg with AX	90 +rw	-	-	3	3	-	-	-	-	-	-	-	-	-
	Exchange AX with word reg		-	-	3	3									
	Exchange byte reg with r/byte	86 /r	-	-	4/17	4/17									
	Exchange r/m8 with byte reg		-	-	4/17	4/17									
	Exchange word reg with r/m16	87 /r	-	-	4/17	4/21									
	Exchange r/m16 with word reg		-	-	4/17	4/21									
XLAT	Set AL to memory byte segment :[BX+unsigned AL]	D7	-	-	11	15	-	-	-	-	-	-	-	-	-
XLATB	Set AL to memory byte DS :[BX+unsigned AL]	D7	-	-	11	15									
XOR	XOR imm8 with AL	34	ib	-	3	3	0	-	-	-	R	R	U	R	0
	XOR imm16 with AX	35	iw	-	4	4									
	XOR imm8 with r/m8	80	/6 ib	-	4/16	4/16									
	XOR imm16 with r/m16	81	/6 iw	-	4/16	4/20									
	XOR sign-extended imm8 with r/m16	83	/6 ib	-	4/16	4/20									
	XOR byte reg with r/m8	30	/r	-	3/10	3/10									
	XOR word reg with r/m16	31	/r	-	3/10	3/14									
	XOR r/m8 with byte reg	32	/r	-	3/10	3/10									
	XOR r/m16 with word reg	33	/r	-	3/10	3/14									

Key to Abbreviations Used Instruction Summary Table

The Operand Address byte is configured as follows.

7	6	5	4	3	2	1	0
mod field		aux field			r/m field		

mod field (Modifier Field)

mod	Description
11	r/m is treated as a register field
00	DISP = 0, disp-low and disp-high are absent - address displacement is 0.
01	DISP = disp-low sign-extended to 16-bits, disp-high is absent.
10	DISP = disp-high: disp-low.

aux field (Auxiliary Field)

aux	If mod = 11 and word = 0	If mod = 11 and word = 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

When mod ≠ 11, depends on instruction

r/m field

r/m	Description
000	EA = (BX) + (SI) + DISP [where EA is the Effective Address]
001	EA = (BX) + (DI) + DISP
010	EA = (BP) + (SI) + DISP
011	EA = (BX) + (DI) + DISP
100	EA = (SI) + DISP
101	EA = (DI) + DISP
110	EA = (BP) + DISP [except if mod = 00, then EA = disp-high: disp-low]
111	EA = (BX) + DISP

Displacement

The displacement is an 8 or 16 bit value added to the offset portion of the address.

Immediate

The immediate bytes consist of up to 16 bits of immediate data.

Segment Override Prefix

The Operand Address byte is configured as follows.

7	6	5	4	3	2	1	0
0	0	1	**SR**	**SR**	1	1	0

SR	Segment Register
00	ES
01	CS
10	SS
11	DS

Notation

Parameter	Indication
:	The component of the left is the segment for a component located in memory. The component on the right is the offset.
::	The component of the left is concatenated with the component on the right.

Operand	Translation
imm8	Immediate byte: signed number between –128 and 127
imm16	Immediate word: signed number between –32768 and 32767
m	Operand in memory
m8	Byte string in memory pointed to by DS:SI or ES:DI
m16	Word string in memory pointed to by DS:SI or ES:DI
r/m8	General byte register or a byte in memory
r/m16	General word register or a word in memory

Opcode

Parameter	
/0 - /7	The Auxiliary Field in the Operand Address byte specifies an extension (from 000 to 111, i.e. 0 to 7) to the opcode instead of a register. Thus the opcode for adding (AND) an immediate byte to a general byte register or a byte in memory is '*80 /4 ib'*. This indicates that the second byte of the opcode is '*mod 100 r/m'*.
/r	The Auxiliary Field in the Operand Address byte specifies a register rather that an opcode extension. The opcode byte specifies which register, either byte size or word size, is assigned as in the aux code above.

/sr	This byte is placed before the instruction as shown above under Segment Override Prefix.
cb	The byte following the Opcode byte specifies the offset.
cd	The double-word following the Opcode byte specifies the offset and is some cases a segment.
ib	Immediate byte – signed or unsigned determined by the Opcode byte.
iw	Immediate word – signed or unsigned determined by the Opcode byte.
rw	Word register operand as determined by the Opcode byte, aux field.

Flags Affected After Instruction

U	Undefined
-	Unchanged
R	Result dependent

Absolute Maximum Ratings

Storage Temperature	-65°C to +125°C
Voltage on any pin with respect to ground	-0.5 V to V_{CC} +0.5 V
Operating Range	
Industrial (T_A)	-40°C to +85°C
T_A = ambient temperature	

DC Characteristics Over Commercial Operating Ranges

Symbol	Parameter Description	Test Conditions	Preliminary		Unit
			Min	Max	
V_{IL}	Input Low Voltage (Except X1)		-0.5	0.8	V
V_{IL1}	Clock Input Low Voltage (X1)		-0.5	0.8	V
V_{IH}	Input High Voltage (Except res_n and X1)		2.0	V_{CC} + 0.5	V
V_{IH1}	Input High Voltage (res_n)		2.4	V_{CC} + 0.5	V
V_{IH1}	Clock Input High Voltage (X1)		V_{CC} - 0.8	V_{CC} + 0.5	V
V_{OL}	**Output Low Voltages**[1]	I_{OL} = 2.5 mA (s2_n -s0_n)		0.45	V
		I_{OL} = 2.0 mA (other)		0.45	V
V_{OH}	**Output High Voltages**	I_{OH} = -2.4 mA @ 2.4 V	2.4	V_{CC} + 0.5	V
		I_{OH} = -200 μA @ V_{CC} □ 0.5	V_{CC} - 0.5	V_{CC}	V
I_{CC}	**Power Supply Current @ 0°C**	V_{CC} = 5.5 V[2]		5.9	mA/MHz
I_{LI}	Input Leakage Current @ 0.5 MHz	0.45 V ≤ V_{IN} ≤ V_{CC}		±10	μA
I_{LO}	Output Leakage Current @ 0.5 MHz	0.45 V ≤ V_{OUT} ≤ V_{CC}[3]		±10	μA
V_{CLO}	Clock Output Low	I_{CLO} = 4.0 mA		0.45	V
V_{CHO}	Clock Output High	I_{CHO} = -500 μA	V_{CC} - 0.5		V

NOTES

1. The **lcs_n/once0_n**, **mcs3_n – mcs0_n**, **ucs_n/once1_n**, and **rd_n** pins have weak internal pull-up resistors. Loading the **lcs_n/once0_n** and **ucs_n/once1_n** pins in excess of I_{OH} = -200 μA during reset can cause the device to go into ONCE mode.

2. Current is measured with the device in reset with the **x1** and **x2** driven and all other non-power pins open but held High or Low.

3. Testing is performed with the pins floating, either during **hold** or by invoking the ONCE mode.

AC Characteristics Over Commercial Operating Ranges (40 MHz)

No.	Name	Description	MIN	MAX	Units
General Timing Requirements					
1	tDVCL	Data in Setup	10		ns
2	tCLDX	Data in Hold	0		ns
General Timing Responses					
3	tCHSV	Status Active Delay	0	6	ns
4	tCLSH	Status Inactive Delay	0	6	ns
5	tCLAV	**ad** Address Valid Delay	0	12	ns
6	tCLAX	Address Hold	0	12	ns
8	tCHDX	Status Hold Time	0		ns
9	tCHLH	**ale** Active Delay	0	8	ns
10	tLHLL	**ale** Width	tCLCH-5		ns
11	tCHLL	**ale** Inactive Delay	0	8	ns
12	tAVLL	**ad** Address Valid to **ale** Low	tCLCH		ns
13	tLLAX	**ad** Address Hold from **ale** Inactive	tCHCL		ns
14	tAVCH	**ad** Address Valid to Clock High	0		ns
15	tCLAZ	**ad** Address Float Delay	0	12	ns
16	tCLCSV	**mcs_n/pcs_n** Inactive Delay	0	12	ns
17	tCXCSX	**mcs_n/pcs_n** Hold from Command Inactive	tCLCH		ns
18	tCHCSX	**mcs_n/pcs_n** Inactive Delay	0	12	ns
19	tDXDL	**den_n** Inactive to **dt_r_n** Low	0		ns
20	tCVCTV	Control Active Delay 1	0	10	ns
21	tCVDEX	**den_n** Inactive Delay	0	0	ns
22	tCHCTV	Control Active Delay 2	0	10	ns
23	tLHAV	**ale** High to Address Valid	7.5		ns
80	tCLCLX	**lcs_n** Inactive Delay	0	9	ns
81	tCLCSL	**lcs_n** Active Delay	0	9	ns
82	tCLRF	**clkoutA** High to **rfsh_n** Invalid	0	12	ns
84	tLRLL	**lcs_n** Precharge Pulse Width	tCLCL + tCLCH		ns
Read Cycle Timing Responses					
24	tAZRL	**ad** Address Float to **rd_n** Active	0		ns
25	tCLRL	**rd_n** Active Delay	0	10	ns
26	tRLRH	**rd_n** Pulse Width	tCLCL		ns
27	tCLRH	**rd_n** Inactive Delay	0	10	ns
28	tRHLH	**rd_n** Inactive to ale High	tCLCH		ns
29	tRHAV	**rd_n** Inactive to **ad** Address Active	tCLCL		ns
30	tCLDOX	Data Hold Time	0		ns
Write Cycle Timing Responses					
31	tCVCTX	Control Inactive Delay	0	10	ns
32	tWLWH	**wr_n** Pulse Width	2tCLCL		ns
33	tWHLH	**wr_n** Inactive to **ale** High	tCLCH		ns

No.	Name	Description	MIN	MAX	Units
34	tWHDX	Data Hold after **wr_n**	tCLCL		ns
35	tWHDEX	**wr_n** Inactive to **den_n** Inactive	tCLCH		ns
41	tDSHLH	**ds_n** Inactive to **ale** Inactive	tCLCH		ns
59	tRHDX	**rd_n** High to Data Hold on **ad** Bus	0		ns
65	tAVWL	a Address Valid to wr_n Low	tCLCL + tCHCL		ns
66	tAVRL	**a** Address Valid to **rd_n** Low	tCLCL + tCHCL		ns
67	tCHCSV	**clkoutA** High to **lcs_n/usc_n** Valid	0	9	ns
68	tCHAV	**clkoutA** High to **a** Address Valid	0	8	ns
87	tAVBL	**a** Address Valid to **whb_n/wlb_n** Low	tCHCL - 1.5	tCHCL	ns
Refresh Timing Cycle Parameters					
79	tCHRFD	**clkoutA** High to **rfsh_n** Valid	0	12	ns
82	tCLRF	**clkoutA** High to **rfsh_n** Invalid	0	12	ns
85	tRFCY	**rfsh_n** Cycle Time	6tCLCL		ns
86	tLCRF	**lcs_n** Inactive to **rfsh_n** Active Delay	2tCLCL		ns
clkin Timing					
36	tCKIN	**X1** Period	25	66	ns
37	tCLCK	**X1** Low Time	7.5		ns
38	tCHCK	**X1** High Time	7.5		ns
39	tCKHL	**X1** Fall Time		5	ns
40	tCKLH	**X1** Rise time		5	ns
clkout Timing					
42	tCLCL	**clkoutA** Period	25		ns
43	tCLCH	**clkoutA** Low Time	TCLCL/2		ns
44	tCHCL	**clkoutA** High Time	TCLCL/2		ns
45	tCH1CH2	**clkoutA** Rise Time		3	ns
46	tCL2CL1	**clkoutA** Fall Time		3	ns
61	tLOCK	Maximum PLL Lock Time		0.5	ms
69	tCICOA	**X1** to **clkoutA** Skew		25	ns
70	tCICOB	**X1** to **clkoutB** Skew		35	ns
Ready & Peripheral Timing Requirements					
47	tSRYCL	**srdy** Transition Setup Time	10		ns
48	tCLSRY	**srdy** Transistion Hold Time	3		ns
49	tARYCH	**ardy** Resolution Transition Setup Time	9		ns
50	tCLARX	**ardy** Active Hold Time	4		ns
51	tARYCHL	**ardy** Inactive Holding Time	6		ns
52	tARYLCL	**ardy** Setup Time	9		ns
53	tINVCH	Peripheral Setup Time	10		ns
54	tINVCL	**drq** Setup Time	10		ns

No.	Name	Description	MIN	MAX	Units
Peripheral Timing Responses					
55	tCLTMV	Timer Output Delay	0	12	ns
Reset & Hold Timing Requirements					
57	tRESIN	**res_n** Setup Time	10		ns
58	tHVCL	**hld** Setup Time	10		ns
Reset & Hold Timing Responses					
62	tCLHAV	**hlda** Valid Delay	0	7	ns
63	tCHCZ	Command Lines Float Delay	0	12	ns
64	tCHCV	Command Lines Valid Delay (after Float)	0	12	ns
Synchronous Serial Port Timing Requirements					
75	tDVSH	Data Valid to **sclk** High	10		ns
77	tSHDX	**sclk** High to SPI Data Hold	3		ns
Synchronous Serial Port Timing Responses					
71	tCLEV	clkouta Low to sden Valid	0	12	ns
72	tCLSL	clkouta Low to sclk High	0	12	ns
78	tSLDV	sclk Low to Data Valid	0	12	ns

Waveforms

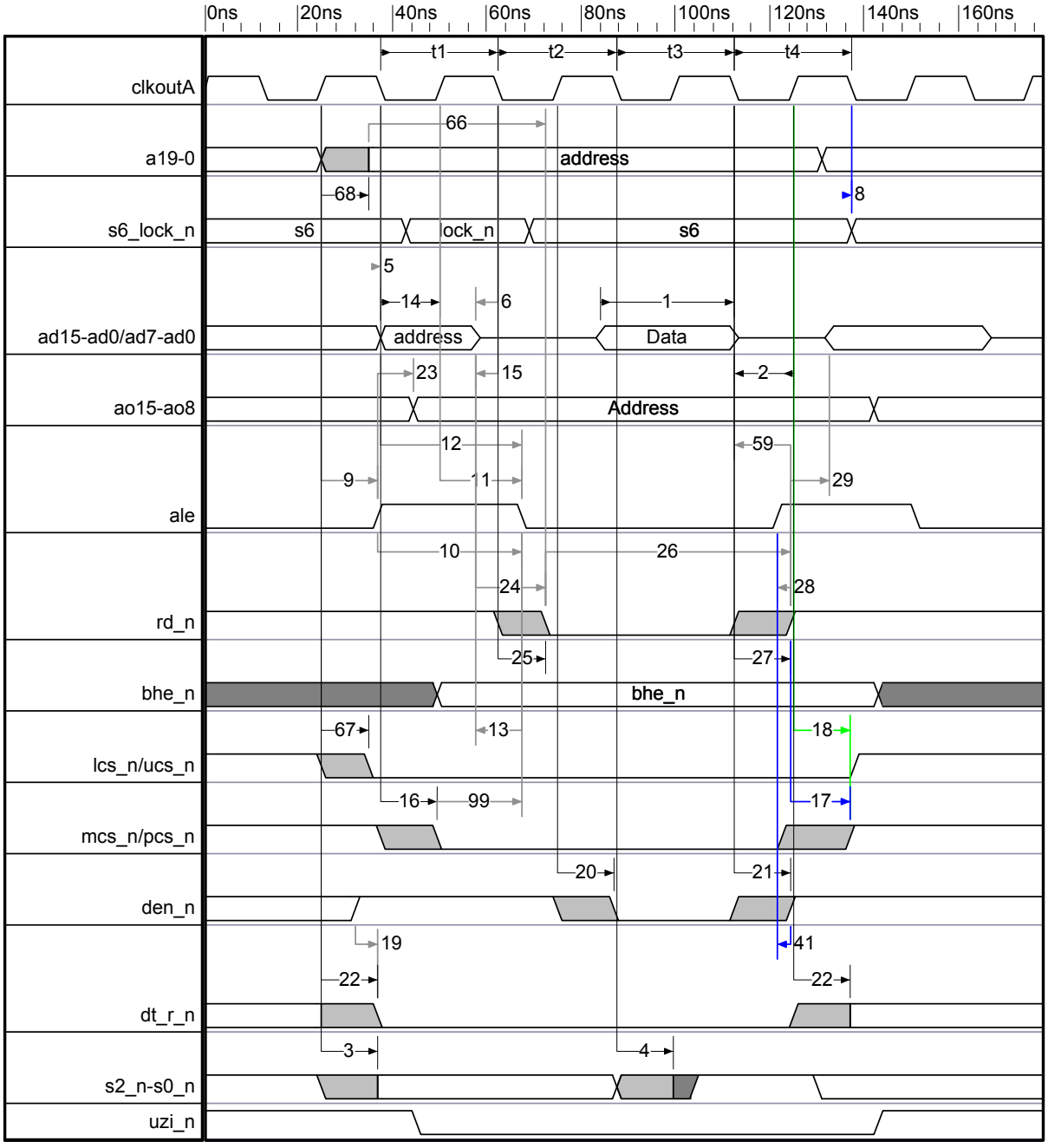
Alphabetic Key to Waveform Parameters

No.	Name	Description	No.	Name	Description
49	tARYCH	**ardy** Resolution Transition Setup Time	2	tCLDX	Data in Hold
51	tARYCHL	**ardy** Inactive Holding Time	71	tCLEV	**clkoutA** Low to **sden** Valid
52	tARYLCL	**ardy** Setup Time	62	tCLHAV	**hlda** Valid Delay
87	tAVBL	**a** Address Valid to **whb_n/wlb_n** Low	82	tCLRF	**clkoutA** High to **rfsh_n** Invalid
14	tAVCH	**ad** Address Valid to Clock High	27	tCLRH	**rd_n** Inactive Delay
12	tAVLL	**ad** Address Valid to **ale** Low	25	tCLRL	**rd_n** Active Delay
66	tAVRL	**a** Address Valid to **rd_n** Low	4	tCLSH	Status Inactive Delay
65	tAVWL	**a** Address Valid to **wr_n** Low	72	tCLSL	**clkoutA** Low to **sclk** Low
24	tAZRL	**ad** Address Float to **rd_n** Active	48	tCLSRY	**srdy** Transistion Hold Time
45	tCH1CH2	**clkoutA** Rise Time	55	tCLTMV	Timer Output Delay
68	tCHAV	**clkoutA** High to A Address Valid	83	tCOAOB	**clkoutA** to **clkoutB** Skew
38	tCHCK	**X1** High Time	20	tCVCTV	Control Active Delay 1
44	tCHCL	**clkoutA** High Time	31	tCVCTX	Control Inactive Delay
67	tCHCSV	**clkoutA** High to **lcs_n/usc_n** Valid	21	tCVDEX	**den_n** Inactive Delay
18	tCHCSX	**mcs_n/pcs_n** Inactive Delay	17	tCXCSX	**mcs_n/pcs_n** Hold from Command Inactive
22	tCHCTV	Control Active Delay 2	1	tDVCL	Data in Setup
64	tCHCV	Command Lines Valid Delay (after Float)	75	tDVSH	Data Valid to SCLK High
63	tCHCZ	Command Lines Float Delay	19	tDXDL	**den_n** Inactive to **dt_r_n** Low
8	tCHDX	Status Hold Time	58	tHVCL	**hld** Setup Time
9	tCHLH	**ale** Active Delay	53	tINVCH	Peripheral Setup Time
11	tCHLL	**ale** Inactive Delay	54	tINVCL	**drq** Setup Time
79	tCHRFD	**clkoutA** High to **rfsh_n** Valid	86	tLCRF	**lcs_n** Inactive to **rfsh_n** Active Delay
3	tCHSV	Status Active Delay	23	tLHAV	**ale** High to Address Valid
69	tCICOA	**X1** to **clkoutA** Skew	10	tLHLL	**ale** Width
70	tCICOB	**X1** to **clkoutB** Skew	13	tLLAX	**ad** Address Hold from ALE Inactive
39	tCKHL	**X1** Fall Time	61	tLOCK	Maximum PLL Lock Time
36	tCKIN	**X1** Period	84	tLRLL	**lcs_n** Precharge Pulse Width
40	tCKLH	**X1** Rise time	57	tRESIN	**res_n** Setup Time
46	tCL2CL1	**clkoutA** Fall Time	85	tRFCY	**rfsh_n** Cycle Time
50	tCLARX	**ardy** Active Hold Time	29	tRHAV	**rd_n** Inactive to **ad** Address Active
5	tCLAV	**ad** Address Valid Delay	59	tRHDX	**rd_n** High to Data Hold on **ad** Bus
6	tCLAX	Address Hold	28	tRHLH	**rd_n** Inactive to **ale** High
15	tCLAZ	**ad** Address Float Delay	26	tRLRH	**rd_n** Pulse Width
43	tCLCH	**clkoutA** Low Time	77	tSHDX	**sclk** High to SPI Data Hold
37	tCLCK	**X1** Low Time	78	tSLDV	**sclk** Low SPI Data Hold
42	tCLCL	**clkoutA** Period	47	tSRYCL	**srdy** Transition Setup Time
80	tCLCLX	**lcs_n** Inactive Delay	35	tWHDEX	**wr_n** Inactive to **den_n** Inactive
81	tCLCSL	**lcs_n** Active Delay	34	tWHDX	Data Hold after **wr_n**
16	tCLCSV	**mcs_n/pcs_n** Inactive Delay	33	tWHLH	**wr_n** Inactive to **ale** High
30	tCLDOX	Data Hold Time	32	tWLWH	**wr_n** Pulse Width
7	tCLDV	Data Valid Delay			

Numeric Key to Waveform Parameters

No.	Name	Description	No.	Name	Description
1	tDVCL	Data in Setup	43	tCLCH	**clkoutA** Low Time
2	tCLDX	Data in Hold	44	tCHCL	**clkoutA** High Time
3	tCHSV	Status Active Delay	45	tCH1CH2	**clkoutA** Rise Time
4	tCLSH	Status Inactive Delay	46	tCL2CL1	**clkoutA** Fall Time
5	tCLAV	**ad** Address Valid Delay	47	tSRYCL	**srdy** Transition Setup Time
6	tCLAX	Address Hold	48	tCLSRY	**srdy** Transistion Hold Time
7	tCLDV	Data Valid Delay	49	tARYCH	**ardy** Resolution Transition Setup Time
8	tCHDX	Status Hold Time	50	tCLARX	**ardy** Active Hold Time
9	tCHLH	**ale** Active Delay	51	tARYCHL	**ardy** Inactive Holding Time
10	tLHLL	**ale** Width	52	tARYLCL	**ardy** Setup Time
11	tCHLL	**ale** Inactive Delay	53	tINVCH	Peripheral Setup Time
12	tAVLL	**ad** Address Valid to ALE Low	54	tINVCL	**drq** Setup Time
13	tLLAX	**ad** Address Hold from ALE Inactive	55	tCLTMV	Timer Output Delay
14	tAVCH	**ad** Address Valid to Clock High	57	tRESIN	**res_n** Setup Time
15	tCLAZ	**ad** Address Float Delay	58	tHVCL	**hld** Setup Time
16	tCLCSV	**mcs_n/pcs_n** Inactive Delay	59	tRHDX	**rd_n** High to Data Hold on **ad** Bus
17	tCXCSX	**mcs_n/pcs_n** Hold from Command Inactive	61	tLOCK	Maximum PLL Lock Time
18	tCHCSX	**mcs_n/pcs_n** Inactive Delay	62	tCLHAV	**hlda** Valid Delay
19	tDXDL	**den_n** Inactive to **dt_r_n** Low	63	tCHCZ	Command Lines Float Delay
20	tCVCTV	Control Active Delay 1	64	tCHCV	Command Lines Valid Delay (after Float)
21	tCVDEX	**den_n** Inactive Delay	65	tAVWL	**a** Address Valid to **wr_n** Low
22	tCHCTV	Control Active Delay 2	66	tAVRL	**a** Address Valid to **rd_n** Low
23	tLHAV	**ale** High to Address Valid	67	tCHCSV	**clkoutA** High to **lcs_n/usc_n** Valid
24	tAZRL	**ad** Address Float to **rd_n** Active	68	tCHAV	**clkoutA** High to A Address Valid
25	tCLRL	**rd_n** Active Delay	69	tCICOA	**X1** to **clkoutA** Skew
26	tRLRH	**rd_n** Pulse Width	70	tCICOB	**X1** to **clkoutB** Skew
27	tCLRH	**rd_n** Inactive Delay	71	tCLEV	**clkouta** Low to **sden** Valid
28	tRHLH	**rd_n** Inactive to **ale** High	72	tCLSL	**clkouta** Low to **sclk** High
29	tRHAV	**rd_n** Inactive to **ad** Address Active	75	tDVSH	Data Valid to **sclk** High
30	tCLDOX	Data Hold Time	77	tSHDX	**sclk** High to SPI Data Hold
31	tCVCTX	Control Inactive Delay	78	tSLDV	**sclk** Low to Data Valid
32	tWLWH	**wr_n** Pulse Width	79	tCHRFD	**clkoutA** High to **rfsh_n** Valid
33	tWHLH	**wr_n** Inactive to **ale** High	80	tCLCLX	**lcs_n** Inactive Delay
34	tWHDX	Data Hold after **wr_n**	81	tCLCSL	**lcs_n** Active Delay
35	tWHDEX	**wr_n** Inactive to **den_n** Inactive	82	tCLRF	**clkoutA** High to **rfsh_n** Invalid
36	tCKIN	**X1** Period	83	tCOAOB	**clkoutA** to **clkoutB** Skew
37	tCLCK	**X1** Low Time	84	tLRLL	**lcs_n** Precharge Pulse Width
38	tCHCK	**X1** High Time	85	tRFCY	**rfsh_n** Cycle Time
39	tCKHL	**X1** Fall Time	86	tLCRF	**lcs_n** Inactive to **rfsh_n** Active Delay
40	tCKLH	**X1** Rise time	87	tAVBL	**a** Address Valid to **whb_n/wlb_n** Low
42	tCLCL	**clkoutA** Period			

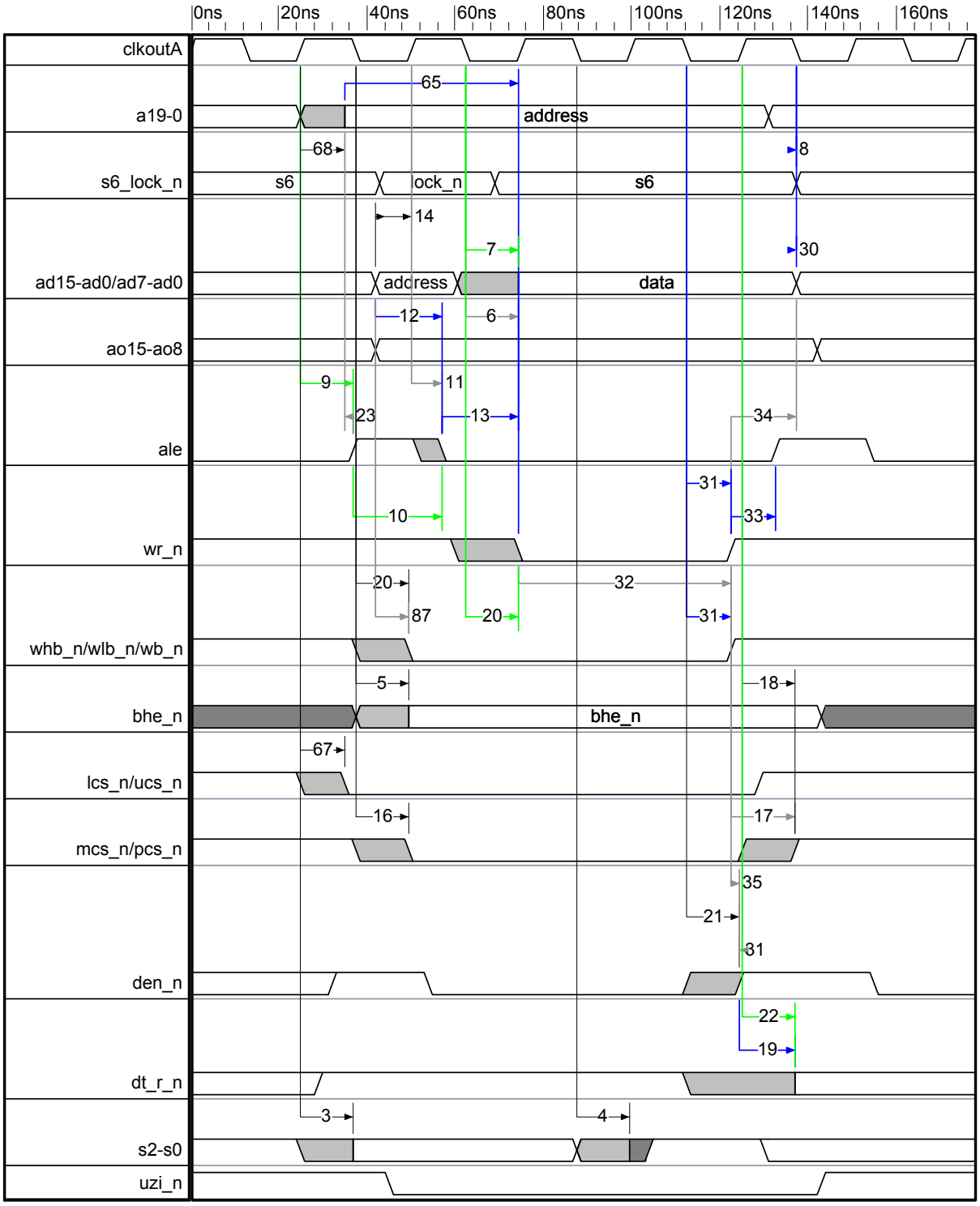
Read Cycle


0ns
20ns
40ns
60ns
80ns
100ns
120ns
140ns
160ns
t1
t2
t3
t4
clkoutA
a19-0
address
s6_lock_n
s6
ock_n
s6
ad15-ad0/ad7-ad0
address
Data
ao15-ao8
Address
ale
rd_n
bhe_n
bhe_n
lcs_n/ucs_n
mcs_n/pcs_n
den_n
dt_r_n
s2_n-s0_n
uzi_n
66
68
8
5
14
6
1
23
15
2
12
59
9
11
29
10
26
24
28
25
27
67
13
18
16
99
17
20
21
19
41
22
22
3
4

Read Cycle Timing

No.	Name	Description	MIN	MAX	Units
General Timing Requirements					
1	tDVCL	Data in Setup	10		ns
2	tCLDX	Data in Hold	0		ns
General Timing Responses					
3	tCHSV	Status Active Delay	0	6	ns
4	tCLSH	Status Inactive Delay	0	6	ns
5	tCLAV	**ad** Address Valid Delay	0	12	ns
6	tCLAX	Address Hold	0	12	ns
8	tCHDX	Status Hold Time	0		ns
9	tCHLH	**ale** Active Delay	0	8	ns
10	tLHLL	**ale** Width	tCLCH-5		ns
11	tCHLL	**ale** Inactive Delay	0	8	ns
12	tAVLL	**ad** Address Valid to **ale** Low	tCLCH		ns
13	tLLAX	**ad** Address Hold from **ale** Inactive	tCHCL		ns
14	tAVCH	**ad** Address Valid to Clock High	0		ns
15	tCLAZ	**ad** Address Float Delay	0	12	ns
16	tCLCSV	**mcs_n/pcs_n** Inactive Delay	0	12	ns
17	tCXCSX	**mcs_n/pcs_n** Hold from Command Inactive	tCLCH		ns
18	tCHCSX	**mcs_n/pcs_n** Inactive Delay	0	12	ns
19	tDXDL	**den_n** Inactive to **dt_r_n** Low	0		ns
20	tCVCTV	Control Active Delay 1	0	10	ns
21	tCVDEX	**den_n** Inactive Delay	0	9	ns
22	tCHCTV	Control Active Delay 2	0	10	ns
23	tLHAV	**ale** High to Address Valid	7.5		ns
Read Cycle Timing Responses					
24	tAZRL	**ad** Address Float to **rd_n** Active	0		ns
25	tCLRL	**rd_n** Active Delay	0	10	ns
26	tRLRH	**rd_n** Pulse Width	tCLCL		ns
27	tCLRH	**rd_n** Inactive Delay	0	10	ns
28	tRHLH	**rd_n** Inactive to ale High	tCLCH		ns
29	tRHAV	**rd_n** Inactive to **ad** Address Active	tCLCL		ns
66	tAVRL	**a** Address Valid to **rd_n** Low	tCLCL + tCHCL		ns
67	tCHCSV	**clkoutA** High to **lcs_n/usc_n** Valid	0	9	ns
68	tCHAV	**clkoutA** High to **a** Address Valid	0	8	ns

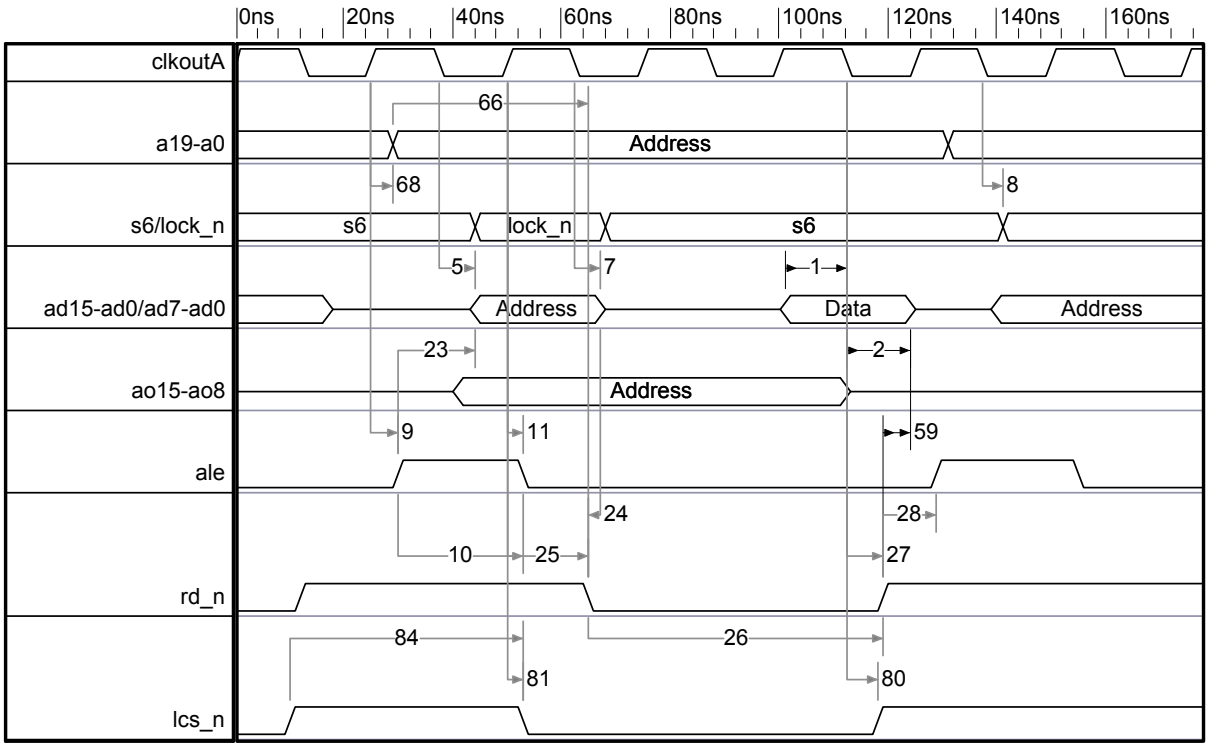
Write Cycle


0ns
20ns
40ns
60ns
80ns
100ns
120ns
140ns
160ns
clkoutA
a19-0
s6_lock_n
ad15-ad0/ad7-ad0
ao15-ao8
ale
wr_n
whb_n/wlb_n/wb_n
bhe_n
lcs_n/ucs_n
mcs_n/pcs_n
den_n
dt_r_n
s2-s0
uzi_n
address
s6
ock_n
s6
address
data
bhe_n
65
68
8
14
7
30
12
6
9
11
23
13
34
31
10
33
20
32
87
20
31
5
18
67
16
17
35
21
31
22
19
3
4

Write Cycle Timing

No.	Name	Description	MIN	MAX	Units
General Timing Requirements					
1	tDVCL	Data in Setup	10		ns
2	tCLDX	Data in Hold	0		ns
General Timing Responses					
3	tCHSV	Status Active Delay	0	6	ns
4	tCLSH	Status Inactive Delay	0	6	ns
5	tCLAV	**ad** Address Valid Delay	0	12	ns
6	tCLAX	Address Hold	0	12	ns
7	tCLDV	Data Valid Delay	0	12	ns
8	tCHDX	Status Hold Time	0		ns
9	tCHLH	**ale** Active Delay	0	8	ns
10	tLHLL	**ale** Width	tCLCH-5		ns
11	tCHLL	**ale** Inactive Delay	0	8	ns
12	tAVLL	**ad** Address Valid to **ale** Low	tCLCH		ns
13	tLLAX	**ad** Address Hold from **ale** Inactive	tCHCL		ns
14	tAVCH	**ad** Address Valid to Clock High	0		ns
16	tCLCSV	**mcs_n/pcs_n** Inactive Delay	0	12	ns
17	tCXCSX	**mcs_n/pcs_n** Hold from Command Inactive	tCLCH		ns
18	tCHCSX	**mcs_n/pcs_n** Inactive Delay	0	12	ns
19	tDXDL	**den_n** Inactive to **dt_r_n** Low	0		ns
20	tCVCTV	Control Active Delay 1	0	10	ns
22	tCHCTV	Control Active Delay 2	0	9	ns
23	tLHAV	**ale** High to Address Valid	7.5		ns
Write Cycle Timing Responses					
30	tCLDOX	Data Hold Time	0		ns
31	tCVCTX	Control Inactive Delay	0	10	ns
32	tWLWH	**wr_n** Pulse Width	2tCLCL		ns
33	tWHLH	**wr_n** Inactive to **ale** High	tCLCH		ns
34	tWHDX	Data Hold after **wr_n**	tCLCL		ns
35	tWHDEX	**wr_n** Inactive to **den_n** Inactive	tCLCH		ns
65	tAVWL	**a** Address Valid to **wr_n** Low	tCLCL + tCHCL		ns
67	tCHCSV	**clkoutA** High to **lcs_n/usc_n** Valid	0	9	ns
68	tCHAV	**clkoutA** High to **a** Address Valid	0	8	ns
87	tAVBL	**a** Address Valid to **whb_n/wlb_n** Low	tCHCL -1.5		ns

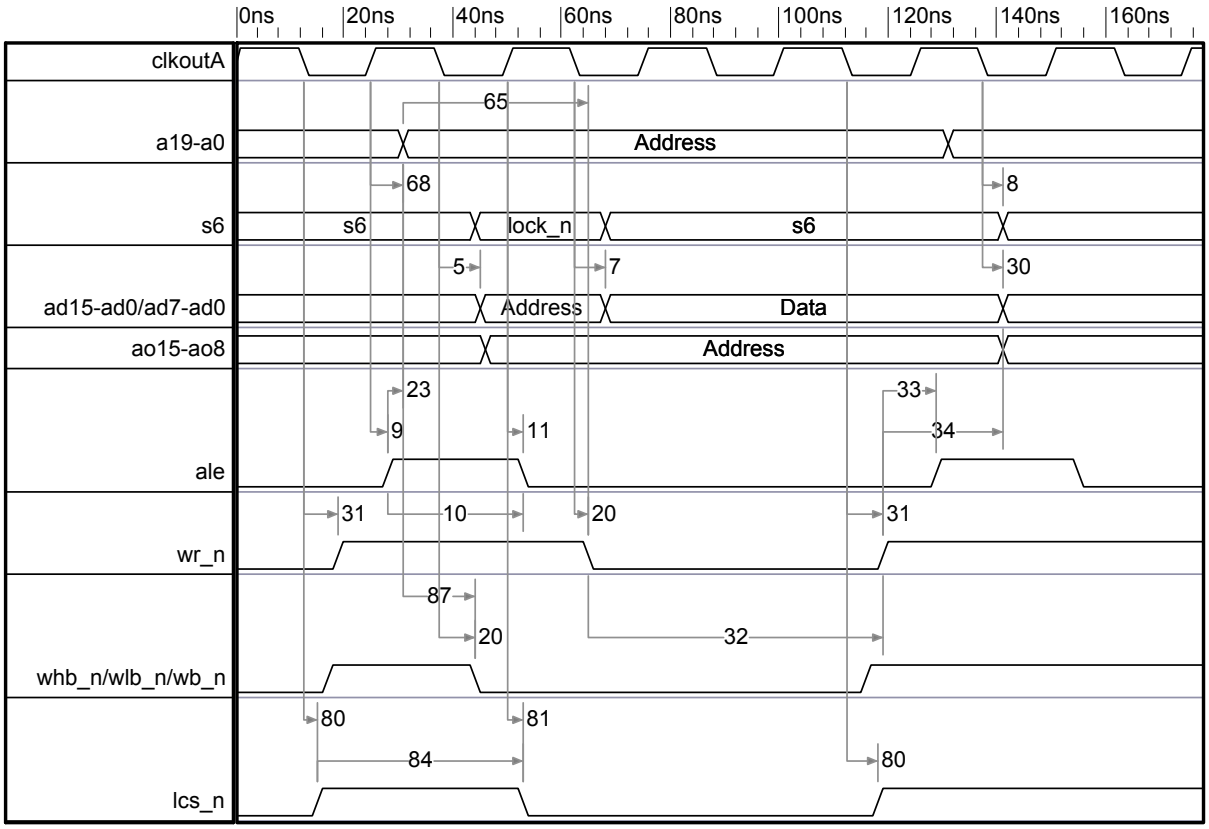
PSRAM Read Cycle



PSRAM Read Cycle Timing

No.	Name	Comment	MIN	MAX	Units
General Timing Requirements					
1	tDVCL	Data in Setup	10	NLL	ns
2	tCLDX	Data in Hold	0		ns
General Timing Responses					
5	tCLAV	**ad** Address Valid Delay	0	12	ns
7	tCLDV	Data Valid Delay	0	12	ns
8	tCHDX	Status Hold Time	0		ns
9	tCHLH	**ale** Active Delay	0	8	ns
10	tLHLL	**ale** Width	tCHCL-5		ns
11	tCHLL	**ale** Inactive Delay	0	8	ns
23	tLHAV	**ale** High to Address Valid	7.5		ns
80	tCLCLX	**lcs_n** Inactive Delay	0	9	ns
81	tCLCSL	**lcs_n** Active Delay	0	9	ns
84	tLRLL	**lcs_n** Precharge Pulse Width	tCLCL + tCLCH		ns
Read Cycle Timing Responses					
24	tAZRL	**ad** Address Float to **rd_n** Active	0		ns
25	tCLRL	**rd_n** Active Delay	0	10	ns
26	tRLRH	**rd_n** Pulse Width	tCLCL		ns
27	tCLRH	**rd_n** Inactive Delay	0	10	ns
28	tRHLH	**rd_n** Inactive to **ale** High	tCLCH		ns
59	tRHDX	**rd_n** High to Data Hold on **ad** Bus	0		ns
66	tAVRL	**a** Address Valid to **rd_n** Low	tCLCL + tCHCL		
68	tCHAV	**clkoutA** High to **a** Address Valid	0	8	ns

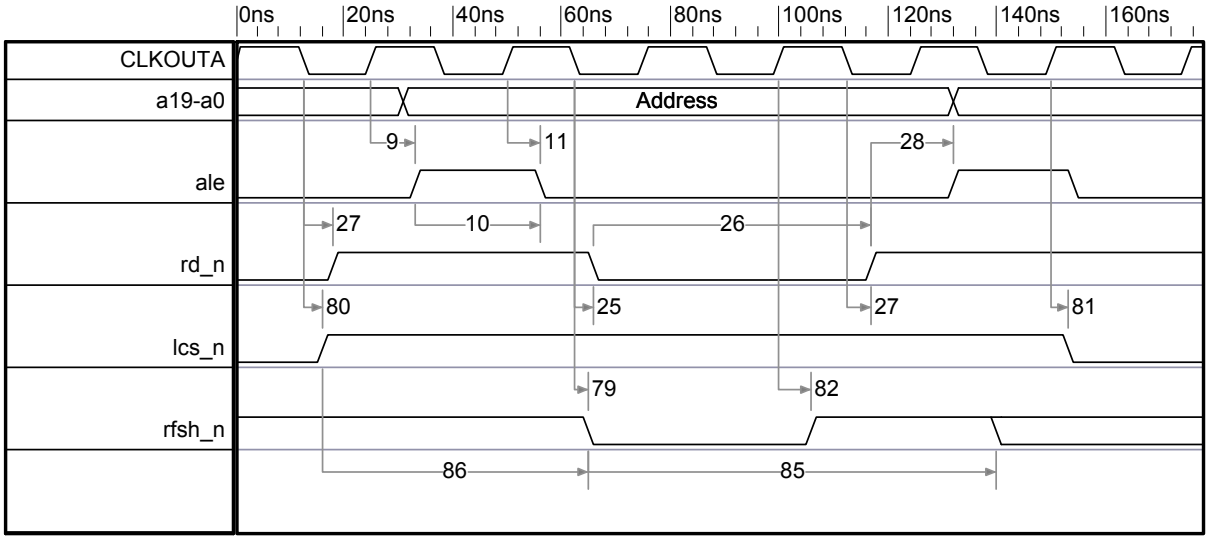
PSRAM Write Cycle



PSRAM Write Cycle Timing

No.	Name	Comment	MIN	MAX	Units
General Timing Requirements					
1	tDVCL	Data in Setup	10		ns
2	tCLDX	Data in Hold	0		ns
General Timing Responses					
5	tCLAV	**ad** Address Valid Delay	0	12	ns
7	tCLDV	Data Valid Delay	0	12	ns
8	tCHDX	Status Hold Time	0		ns
9	tCHLH	**ale** Active Delay	0	8	ns
10	tLHLL	**ale** Width	tCLCH-5		ns
11	tCHLL	**ale** Inactive Delay	NULL	NULL	ns
20	tCVCTV	Control Active Delay 1	0	10	ns
23	tLHAV	**ale** High to Address Valid	7.5		ns
80	tCLCLX	**lcs_n** Inactive Delay	0	9	ns
81	tCLCSL	**lcs_n** Active Delay	0	9	ns
84	tLRLL	**lcs_n** Precharge Pulse Width	tCLCL+ tCLCH		ns
Write Cycle Timing Responses					
30	tCLDOX	Data Hold Time	0		ns
31	tCVCTX	Control Inactive Delay	0	10	ns
32	tWLWH	**wr_n** Pulse Width	2tCLCL		ns
33	tWHLH	**wr_n** Inactive to **ale** High	tCLCH		ns
34	tWHDX	Data Hold after **wr_n**	tCLCL		ns
65	tAVWL	**a** Address Valid to **wr_n** Low	tCLCL+ tCHCL		ns
68	tCHAV	**clkoutA** High to **a** Address Valid	0	8	ns
87	tAVBL	**a** Address Valid to **whb_n/wlb_n** Low	tCHCL - 1.5		ns

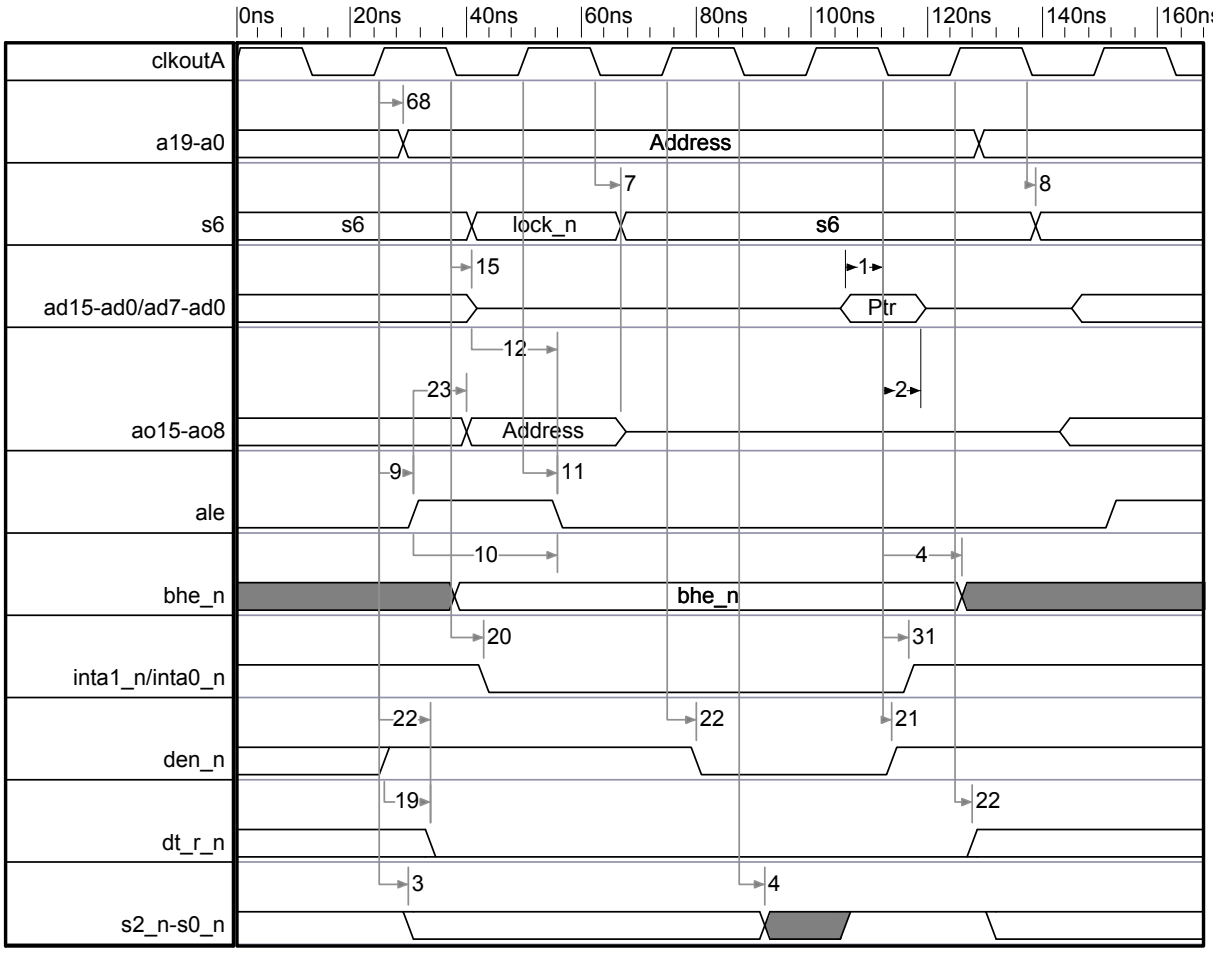
PSRAM Refresh Cycle



PSRAM Refresh Cycle

No.	Name	Comment	MIN	MAX	Units
General Timing Requirements					
1	tDVCL	Data in Setup	10		ns
2	tCLDX	Data in Hold	0		ns
General Timing Responses					
9	tCHLH	**ale** Active Delay	0	8	ns
10	tLHLL	**ale** Width	tCLCH-5		ns
11	tCHLL	**ale** Inactive Delay	0	8	ns
Read/Write Cycle Timing Responses					
25	tCLRL	**rd_n** Active Delay	0	10	ns
26	tRLRH	**rd_n** Pulse Width	tCLCL		ns
27	tCLRH	**rd_n** Inactive Delay	0	10	ns
28	tRHLH	**rd_n** Inactive to ale High	tCLCH		ns
80	tCLCLX	**lcs_n** Inactive Delay	0	9	ns
81	tCLCSL	**lcs_n** Active Delay	0	9	ns
Refresh Cycle Timing Responses					
79	tCHRFD	**clkoutA** High to **rfsh_n** Valid	0	12	ns
82	tCLRF	**clkoutA** High to **rfsh_n** Invalid	0	12	ns
85	tRFCY	**rfsh_n** Cycle Time	6tCLCL		ns
86	tLCRF	**lcs_n** Inactive to **rfsh_n** Active Delay	2tCLCL	NULL	ns

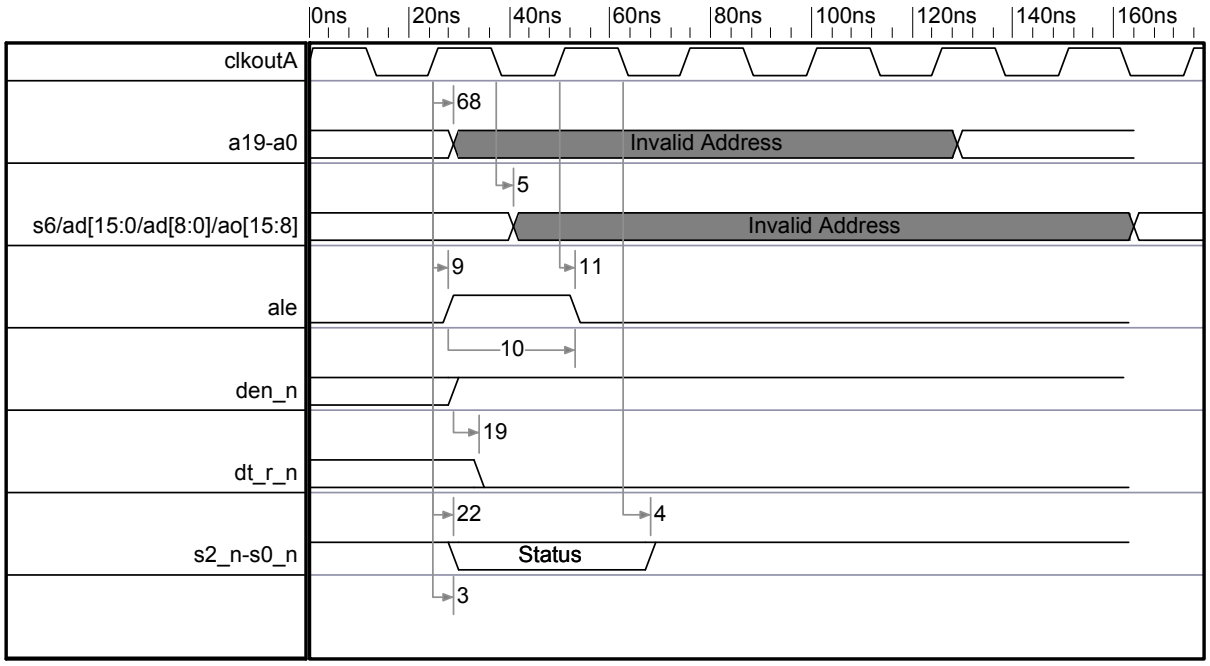
Interrupt Acknowledge Cycle



Interrupt Acknowledge Cycle Timing

No.	Name	Description	MIN	MAX	Units
General Timing Requirements					
1	tDVCL	Data in Setup	10		ns
2	tCLDX	Data in Hold	0		ns
General Timing Responses					
3	tCHSV	Status Active Delay	0	6	ns
4	tCLSH	Status Inactive Delay	0	6	ns
7	tCLDV	Data Valid Delay	0	12	ns
8	tCHDX	Status Hold Time	0		ns
9	tCHLH	**ale** Active Delay	0	8	ns
10	tLHLL	**ale** Width	tCLCH-5		ns
11	tCHLL	**ale** Inactive Delay	0	8	ns
12	tAVLL	**ad** Address Valid to **ale** Low	tCLCH		ns
15	tCLAZ	**ad** Address Float Delay	0	12	ns
19	tDXDL	**den_n** Inactive to **dt_r_n** Low	0		ns
20	tCVCTV	Control Active Delay 1	0	10	ns
21	tCVDEX	**den_n** Inactive Delay	0	9	ns
22	tCHCTV	Control Active Delay 2	0	10	ns
23	tLHAV	**ale** High to Address Valid	7.5		ns
31	tCVCTX	Control Inactive Delay	0	10	ns
68	tCHAV	**clkoutA** High to **a** Address Valid	0	8	ns

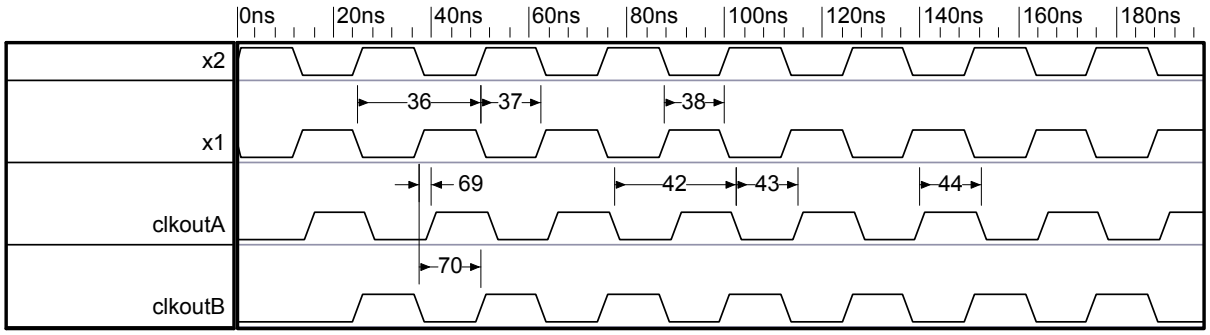
Software Halt Cycle



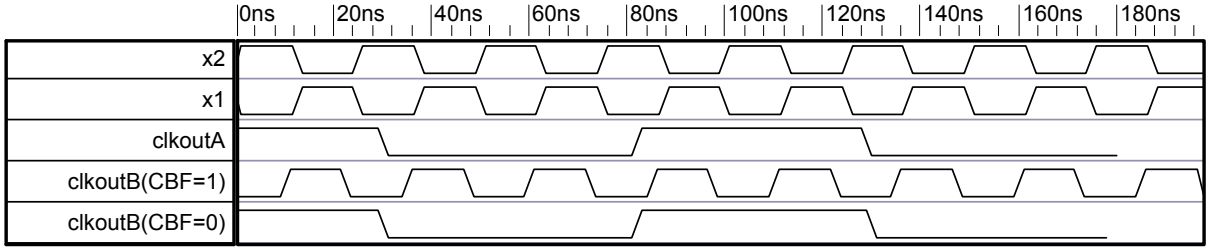
Software Halt Cycle Timing

No.	Name	Description	MIN	MAX	Units
General Timing Responses					
3	tCHSV	Status Active Delay	0	6	ns
4	tCLSH	Status Inactive Delay	0	6	ns
5	tCLAV	**ad** Address Valid Delay	0	12	ns
9	tCHLH	**ale** Active Delay	0	8	ns
10	tLHLL	**ale** Width	tCLCH-5		ns
11	tCHLL	**ale** Inactive Delay	0	8	ns
19	tDXDL	**den_n** Inactive to **dt_r_n** Low	0		ns
22	tCHCTV	Control Active Delay 2	0	10	ns
68	tCHAV	**clkoutA** High to **a** Address Valid	0	8	ns

Clock – Active Mode



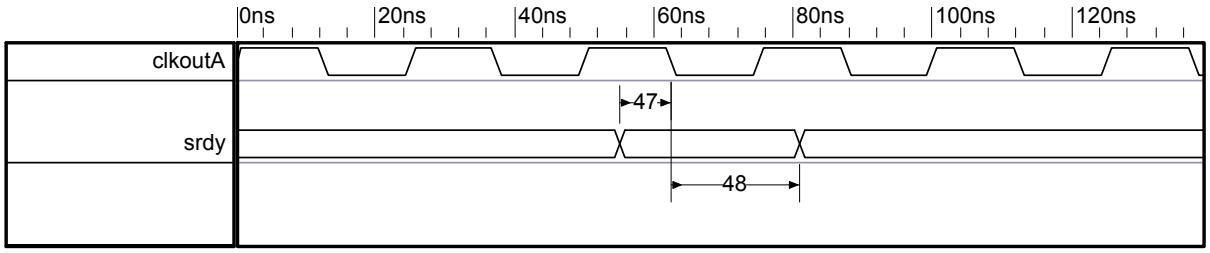
Clock – Power-Save Mode



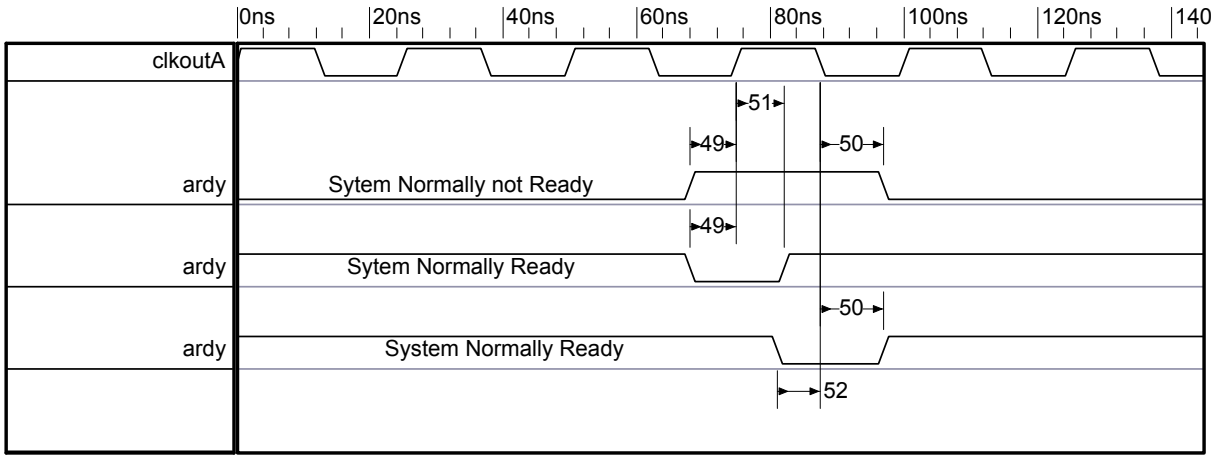
Clock Timing

No.	Name	Description	MIN	MAX	Units
CLKIN Requirements					
36	tCKIN	**X1** Period	25	66	ns
37	tCLCK	**X1** Low Time	7.5		ns
38	tCHCK	**X1** High Time	7.5		ns
39	tCKHL	**X1** Fall Time		5	ns
40	tCKLH	**X1** Rise time		5	ns
CLKOUT Timing					
42	tCLCL	**clkoutA** Period	25		ns
43	tCLCH	**clkoutA** Low Time	tCLCL/2		ns
44	tCHCL	**clkoutA** High Time	tCLCL/2		ns
45	tCH1CH2	**clkoutA** Rise Time		3	ns
46	tCL2CL1	**clkoutA** Fall Time		3	ns
61	tLOCK	Maximum PLL Lock Time		0.5	ms
69	tCICOA	**X1** to **clkoutA** Skew		25	ns
70	tCICOB	**X1** to **clkoutB** Skew		35	ns

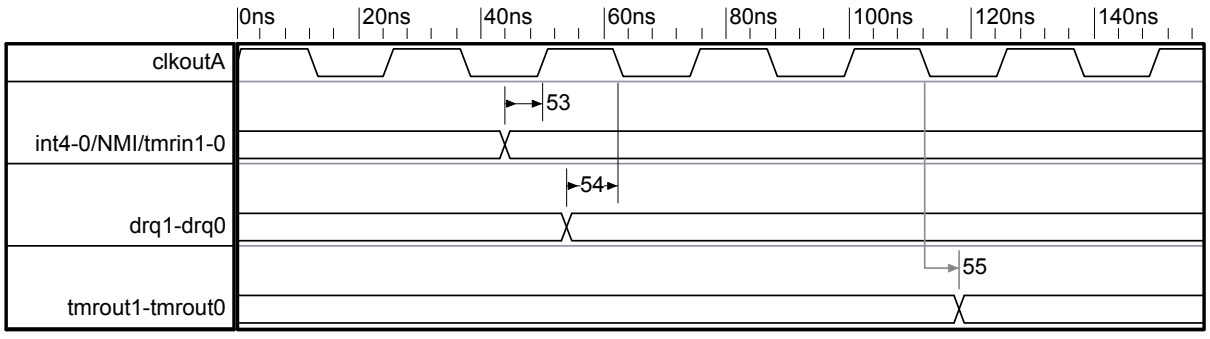
srdy – Synchronous Ready



ardy - Asynchronous Ready



Peripherals

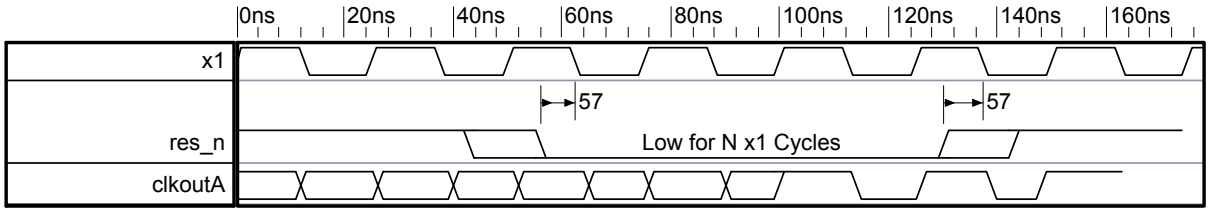


Ready and Peripheral Timing

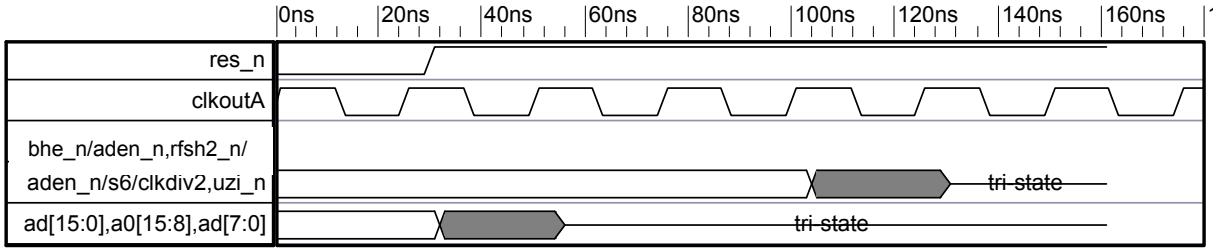
No.	Name	Description	MIN	MAX	Units

Ready and Peripheral Timing Requirements					
47	tSRYCL	**srdy** Transition Setup Time	10		ns
48	tCLSRY	**srdy** Transistion Hold Time	3		ns
49	tARYCH	**ardy** Resolution Transition Setup Time	9		ns
50	tCLARX	**ardy** Active Hold Time	4		ns
51	tARYCHL	**ardy** Inactive Holding Time	6		ns
52	tARYLCL	**ardy** Setup Time	9		ns
53	tINVCH	Peripheral Setup Time	10		ns
54	tINVCL	**drq** Setup Time	10		ns
Peripheral Timing Responses					
55	tCLTMV	Timer Output Delay	0	12	ns

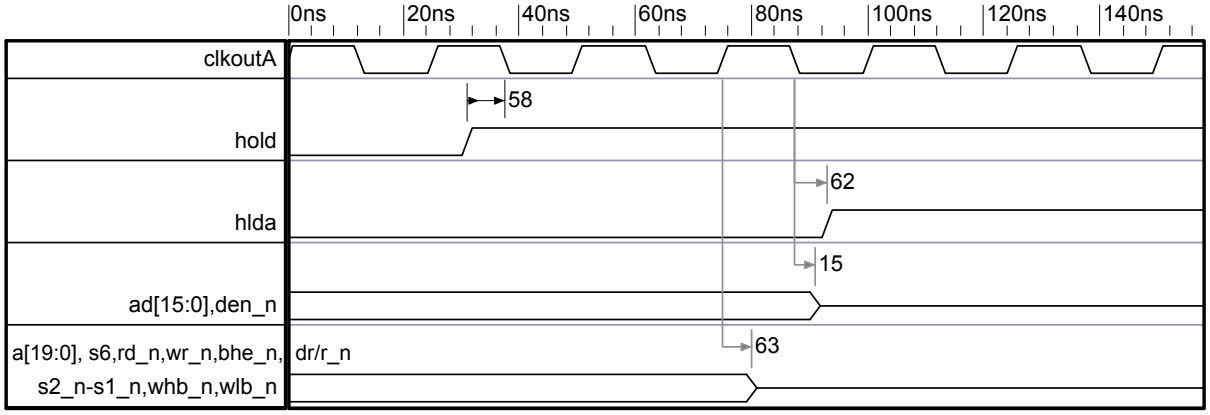
Reset 1



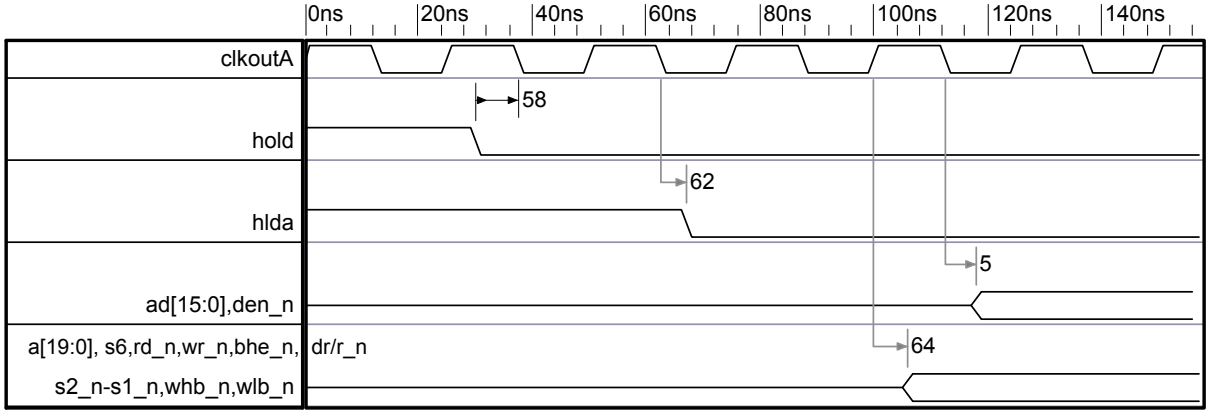
Reset 2



Bus Hold Entering



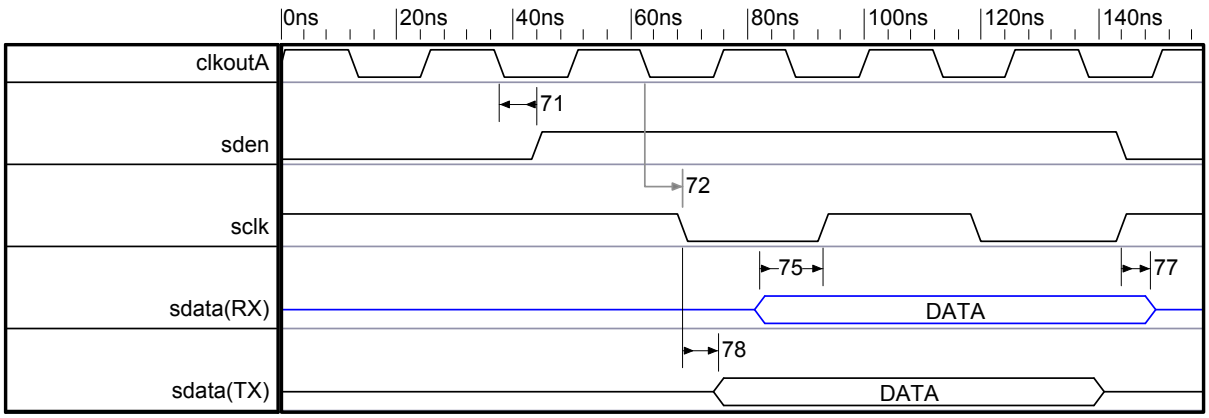
Bus Hold Leaving



Reset and Bus Hold Timing

No.	Name	Description	MIN	MAX	Units
Reset and Bus Hold Timing Requirements					
5	tCLAV	**ad** Address Valid Delay	0	12	ns
15	tCLAZ	**ad** Address Float Delay	tCLCH		ns
57	tRESIN	**res_n** Setup Time	10		ns
58	tHVCL	**hld** Setup Time	10		ns
Reset and Bus Hold Timing Responses					
62	tCLHAV	**hlda** Valid Delay	0	7	ns
63	tCHCZ	Command Lines Float Delay	0	12	ns
64	tCHCV	Command Lines Valid Delay (after Float)	0	12	ns

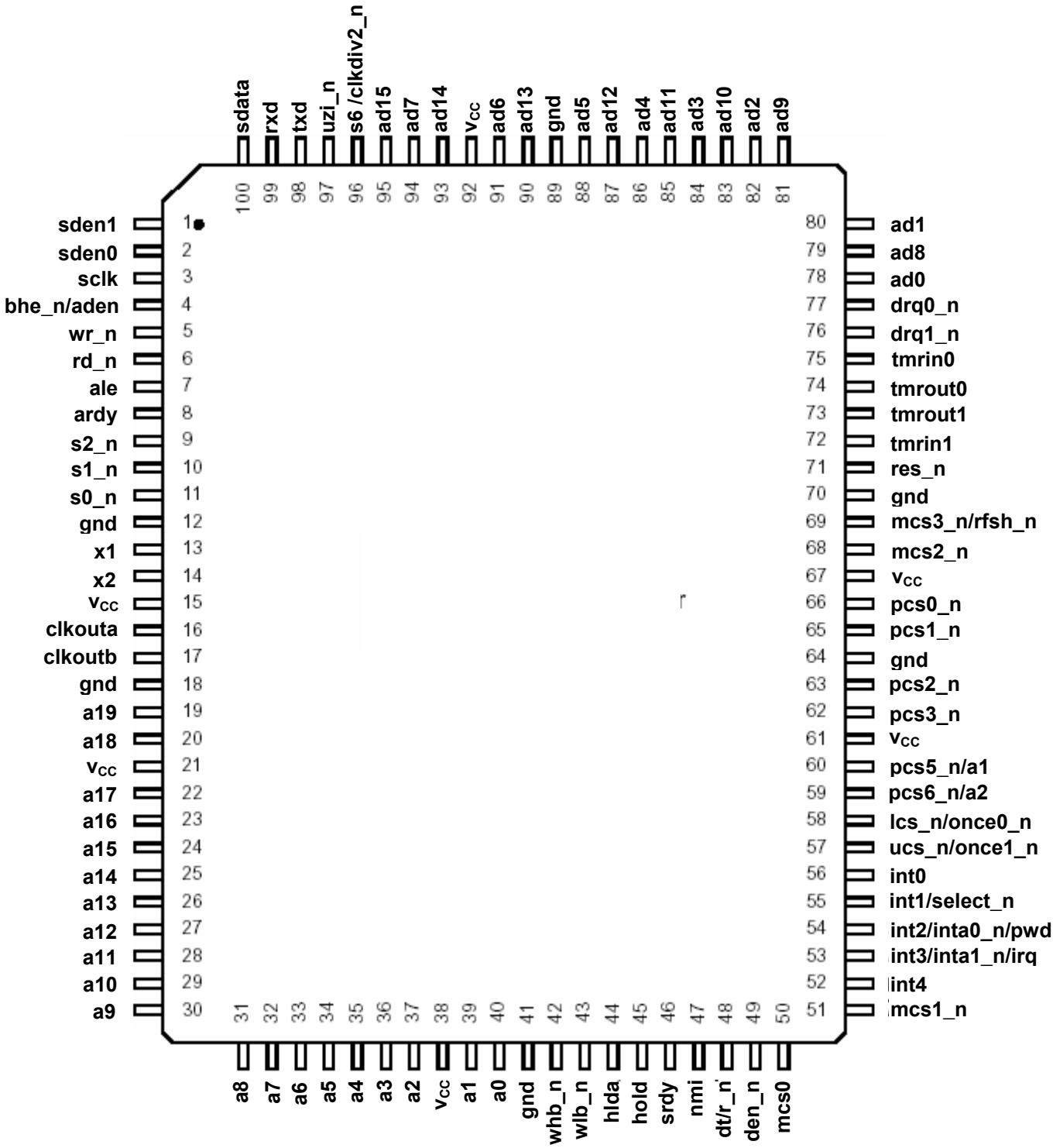
Synchronous Serial Interface



Synchronous Serial Interface Timing

No.	Name	Description	MIN	MAX	Units
Synchronous Serial Port Timing Requirements					
75	tDVSH	Data Valid to **sclk** high	10		ns
77	tSHDX	**sclk** High to SPI Data Hold	3		ns
Synchronous Serial Port Timing Responses					
71	tCLEV	**clkoutA** Low to **sden** Valid	0	12	ns
72	tCLSL	**clkoutA** Low to **sclk** Low	0	12	ns
78	tSLDV	**sclk** Low to Data Valid	0	12	ns

IA186EM 100-Pin PQFP

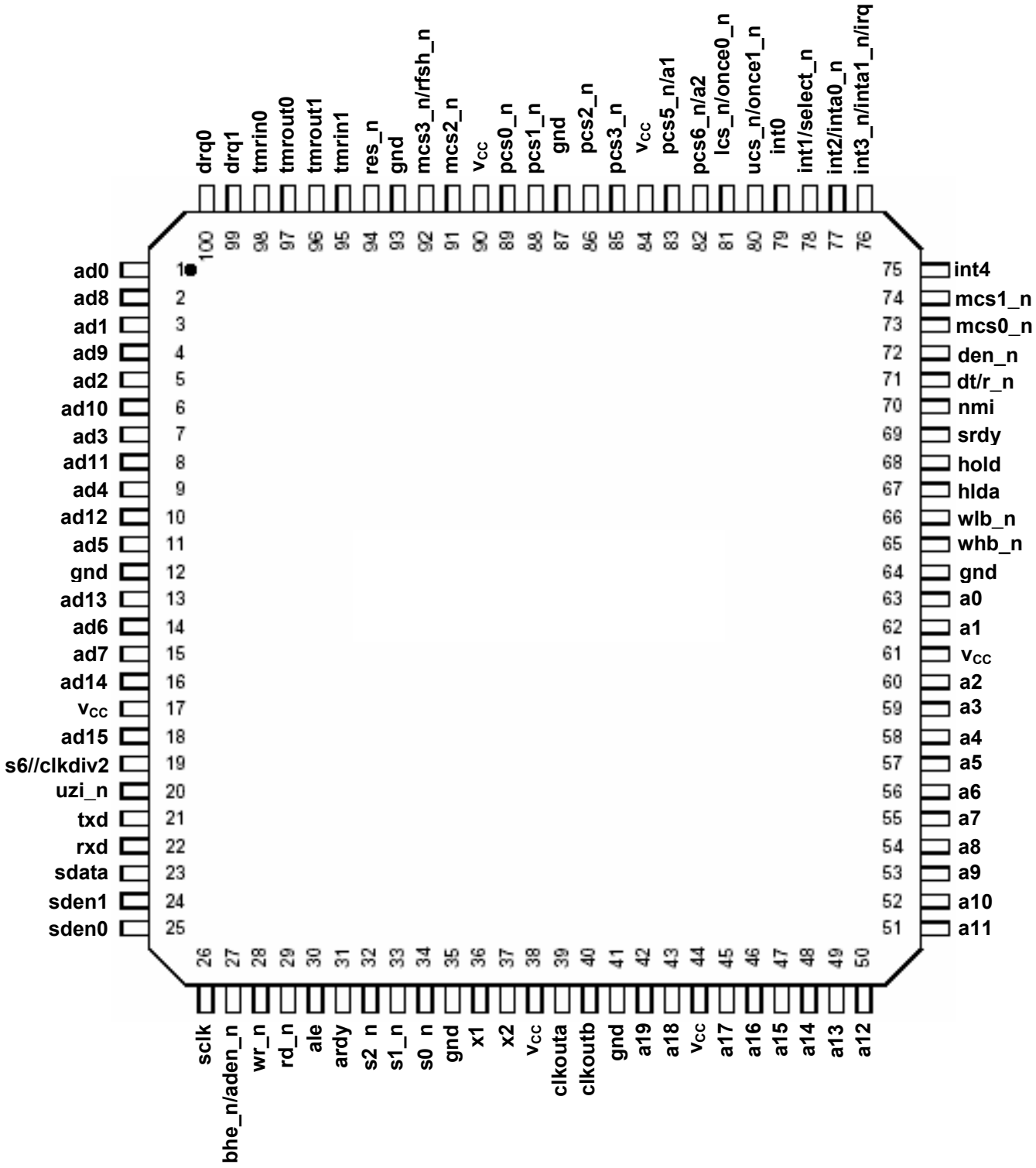


A186EM 100-Pin PQFP Assignments (Sorted by Pin Number)

Pin #	Name	Pin #	Name
1	sden1/pio23	51	mcs1_n/pio15
2	sden0/pio22	52	int4/pio30
3	sclk/pio20	53	int3/inta1_n/irq
4	bhe_n/aden_n	54	int2/inta0_n/pio31
5	wr_n	55	int1/select_n
6	rd_n	56	int0
7	ale	57	ucs_n/once1_n
8	ardy	58	lcs_n/once0_n
9	s2_n	59	pcs6_n/a2/pio2
10	s1_n	60	pcs5_n/a1/pio3
11	s0_n	61	v_{CC}
12	gnd	62	pcs3_n/pio19
13	x1	63	pcs2_n/pio18
14	x2	64	gnd
15	v_{CC}	65	pcs1_n/pio17
16	clkouta	66	pcs0_n/pio16
17	clkoutb	67	v_{CC}
18	gnd	68	mcs2_n/pio24
19	a19/pio29	69	mcs3_n/rfsh_n/pio25
20	a18/pio8	70	gnd
21	v_{CC}	71	res_n
22	a17/pio7	72	tmrin1/pio25
23	a16	73	tmrout1/pio1
24	a15	74	tmrout0/pio10
25	a14	75	tmrin0/pio11
26	a13	76	drq1/pio13
27	a12	77	drq0/pio12
28	a11	78	ad0
29	a10	79	ad8
30	a9	80	ad1
31	a8	81	ad9
32	a7	82	ad2
33	a6	83	ad10
34	a5	84	ad3
35	a4	85	ad11
36	a3	86	ad4
37	a2	87	ad12
38	v_{CC}	88	ad5
39	a1	89	gnd
40	a0	90	ad13
41	gnd	91	ad6
42	whb_n	92	v_{CC}
43	wlb_n	93	ad14
44	hlda	94	ad7
45	hold	95	ad15
46	srdy/pio6	96	s6/clkdiv2_n/pio29
47	nmi	97	uzi_n/pio26
48	dt/r_n/pio4	98	txd/pio27
49	den_n/pio5	99	rxd/pio28
50	mcs0_n/pio14	100	sdata/pio21

Pin Name	Number	Pin Name	Number
a0	40	gnd	89
a1	39	hlda	44
a2	37	hold	45
a3	36	int0	56
a4	35	int1/select_n	55
a5	34	int2/inta0_n/pio31	54
a6	33	int3/inta1_n/irq	53
a7	32	int4/pio30	52
a8	31	lcs_n/once0_n	58
a9	30	mcs0_n/pio14	50
a10	29	mcs1_n/pio15	51
a11	28	mcs2_n/pio24	68
a12	27	mcs3_n/rfsh_n/pio25	69
a13	26	nmi	47
a14	25	pcs0_n/pio16	66
a15	24	pcs1_n/pio17	65
a16	23	pcs2_n/pio18	63
a17/pio7	22	pcs3_n/pio19	62
a18/pio8	20	pcs5_n/a1/pio3	60
a19/pio9	19	pcs6_n/a2/pio2	59
ad0	78	rd_n	6
ad1	80	res_n	71
ad2	82	rxd/pio28	99
ad3	84	s0_n	11
ad4	86	s1_n	10
ad5	88	s2_n	9
ad6	91	s6/clkdiv2/pio29	96
ad7	94	sclk/pio20	3
ad8	79	sdata/pio21	100
ad9	81	sden0/pio22	2
ad10	83	sden1/pio23	1
ad11	85	srdy/pio6	46
ad12	87	tmrin0/pio11	75
ad13	90	tmrin1/pio0	72
ad14	93	tmrout0/pio10	74
ad15	95	tmrout1/pio1	73
ale	7	txd/pio27	98
ardy	8	ucs_n/once1_n	57
bhe_n/aden_n	4	uzi_n/pio26	97
clkouta	16	v_{CC}	15
clkoutb	17	v_{CC}	21
den_n/pio5	49	v_{CC}	38
drq0/pio12	77	v_{CC}	61
drq1/pio13	76	v_{CC}	67
dt/r_n/pio4	48	v_{CC}	92
gnd	12	whb_n	42
gnd	18	wlb_n	43
gnd	41	wr_n	5
gnd	64	x1	13
gnd	70	x2	14

IA186EM TQFP 100-Pin

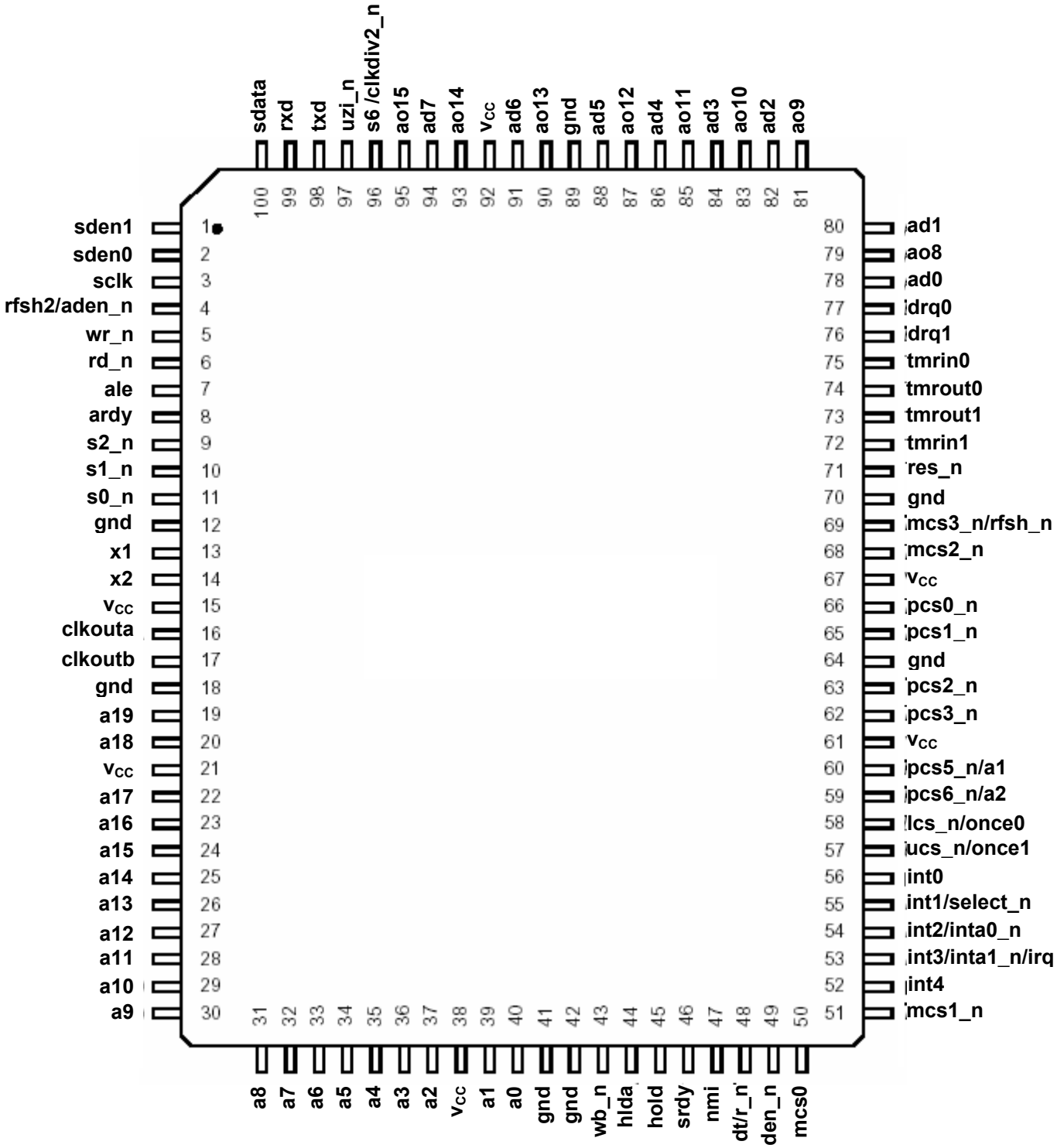


IA186EM 100-Pin TQFP Pin Assignments (sorted by Pin number)

Pin #	Name	Pin #	Name
1	ad0	51	a11
2	ad8	52	a10
3	ad1	53	a9
4	ad9	54	a8
5	ad2	55	a7
6	ad10	56	a6
7	ad3	57	a5
8	ad11	58	a4
9	ad4	59	a3
10	ad12	60	a2
11	ad5	61	v_{CC}
12	gnd	62	a1
13	ad13	63	a0
14	ad6	64	gnd
15	v_{CC}	65	whb_n
16	ad14	66	wlb_n
17	ad7	67	hlda
18	ad15	68	hold
19	s6/clkdiv2/pio29	69	srdy/pio6
20	uzi_n/pio26	70	nmi
21	txd	71	dt/r_n/pio4
22	rxd	72	den_n/pio5
23	sdata/pio21	73	mcs0_n/pio14
24	sden1/pio23	74	mcs1_n/pio15
25	sden0/pio22	75	int4/pio30
26	sclk/pio20	76	int3/inta1_n/irq
27	bhe_n/aden_n	77	int2/inta0_n/pio31
28	wr_n	78	int1/select_n
29	rd_n	79	int0
30	ale	80	ucs_n/once1_n
31	ardy	81	lcs_n/once0_n
32	s2_n	82	pcs6_n/a2/pio2
33	s1_n	83	pcs5_n/a1/pio3
34	s0_n	84	v_{CC}
35	gnd	85	pcs3_n/pio19
36	x1	86	pcs2_n/pio18
37	x2	87	gnd
38	v_{CC}	88	pcs1_n/pio17
39	clkouta	89	pcs0_n/pio16
40	clkoutb	90	v_{CC}
41	gnd	91	mcs2_n/pio24
42	a19/pio9	92	mcs3_n/rfsh_n/pio25
43	a18/pio8	93	gnd
44	v_{CC}	94	res_n
45	a17/pio7	95	tmrin1/pio0
46	a16	96	tmrout1/pio1
47	a15	97	tmrout0/pio10
48	a14	98	tmrin0/pio11
49	a13	99	drq1/pio13
50	a12	100	drq0/pio12

Pin Name	Number	Pin Name	Number
a0	63	gnd	93
a1	62	hlda	67
a2	60	hold	68
a3	59	int0	79
a4	58	int1/select_n	78
a5	57	int2/inta0_n/pio31	77
a6	56	int3/inta1_n/irq	76
a7	55	int4/pio30	75
a8	54	lcs_n/once0_n	81
a9	53	mcs0_n/pio14	73
a10	52	mcs1_n/pio15	74
a11	51	mcs2_n/pio24	91
a12	50	mcs3_n/rfsh_n/pio25	92
a13	49	nmi	70
a14	48	pcs0_n/pio16	89
a15	47	pcs1_npio	88
a16	46	pcs2_n/pio18	86
a17/pio7	45	pcs3_n/pio19	85
a18/pio8	43	pcs5_n/a1/pio3	83
a19/pio9	42	pcs6_n/a2/pio2	82
ad0	1	rd_n	29
ad1	3	res_n	94
ad2	5	rxd/pio23	24
ad3	7	s0_n	34
ad4	9	s1_n	33
ad5	11	s2_n	32
ad6	14	s6/clkdiv2/pio29	19
ad7	17	sclk/pio20	26
ad8	2	sdata/pio21	23
ad9	4	sden0/pio22	25
ad10	6	sden1/pio23	24
ad11	8	srdy/pio6	69
ad12	10	tmrin0/pio11	98
ad13	13	tmrin1/pio0	95
ad14	16	tmrout0/pio10	97
ad15	18	tmrout1/pio1	96
ale	30	txd/pio27	21
ardy	30	ucs_n/once1_n	80
bhe_n/aden_n	27	uzi_n/pio26	20
clkouta	39	v_{CC}	15
clkoutb	40	v_{CC}	38
den_n/pio5	72	v_{CC}	44
drq0/pio12	100	v_{CC}	61
drq1/pio13	99	v_{CC}	84
dt/r_n/pio4	71	v_{CC}	90
gnd	12	whb_n	65
gnd	36	wlb_n	66
gnd	41	wr_n	28
gnd	64	x1	36
gnd	87	x2	37

IA188EM 100-Pin PQFP

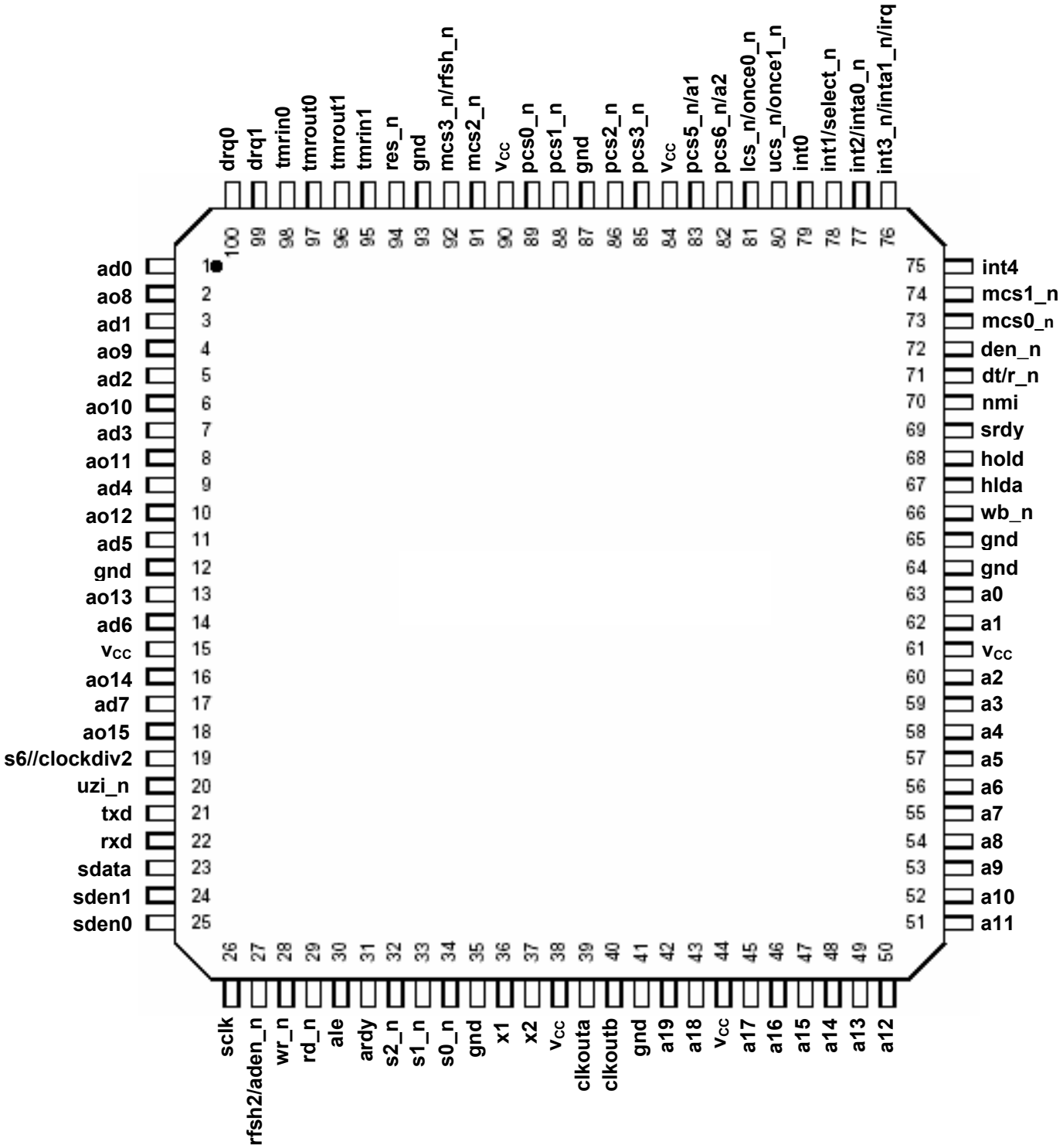


IA188EM 100 Pin PQFP Assignments (sorted by Pin number)

Pin #	Name	Pin #	Name
1	sden1/pio23	51	mcs1_n/pio15
2	sden0/pio22	52	int4/pio30
3	sclk/pio20	53	int3/inta1_n/irq
4	rfsh2_n/aden_n	54	int2/inta0_n/pwd/pio31
5	wr_n	55	int1/select_n
6	rd_n	56	int0
7	ale	57	ucs_n/once1_n
8	ardy	58	lcs_n/once0_n
9	s2_n	59	pcs6_n/a2/pio2
10	s1_n	60	pcs5_n/a1/pio3
11	s0_n	61	v_{CC}
12	gnd	62	pcs3_n/pio19
13	x1	63	pcs2_n/pio18
14	x2	64	gnd
15	v_{CC}	65	pcs1_n/pio17
16	clkouta	66	pcs0_n/pio16
17	clkoutb	67	v_{CC}
18	gnd	68	mcs2_n/pio24
19	a19/pio29	69	mcs3_n/rfsh_n/pio25
20	a18/pio8	70	gnd
21	v_{CC}	71	res_n
22	a17/pio7	72	tmrin1/pio25
23	a16	73	tmrout1/pio1
24	a15	74	tmrout0/pio10
25	a14	75	tmrin0/pio11
26	a13	76	drq1/pio13
27	a12	77	drq0/pio12
28	a11	78	ad0
29	a10	79	ao8
30	a9	80	ad1
31	a8	81	ao9
32	a7	82	ad2
33	a6	83	ao10
34	a5	84	ad3
35	a4	85	ao11
36	a3	86	ad4
37	a2	87	ao12
38	v_{CC}	88	ad5
39	a1	89	gnd
40	a0	90	ao13
41	gnd	91	ad6
42	gnd	92	V_{CC}
43	wb_n	93	ao14
44	hlda	94	ad7
45	hold	95	ao15
46	srdy/pio6	96	s6/clkdiv2_n/pio29
47	nmi	97	uzi_n/pio26
48	dt/r_n/pio4	98	txd/pio27
49	den_n/pio5	99	rxd/pio28
50	mcs0_n/pio14	100	sdata/pio21

Pin Name	Number	Pin Name	Number
a0	40	gnd	89
a1	39	hlda	44
a2	37	hold	45
a3	36	int0	56
a4	35	int1/select_n	55
a5	34	int2/inta0_n/pwd/pio31	54
a6	33	int3/inta1_n/irq	53
a7	32	int4/pio30	52
a8	31	lcs_n/once0_n	58
a9	30	mcs0_n/pio14	50
a10	29	mcs1_n/pio15	51
a11	28	mcs2_n/pio24	68
a12	27	mcs3_n/rfsh_n/pio25	69
a13	26	nmi	47
a14	25	pcs0_n/pio16	66
a15	24	pcs1_n/pio17	65
a16	23	pcs2_n/cts1_n/enrx1_n/pio18	63
a17/pio7	22	pcs3_n/rts1_n/rtr1_n/pio19	62
a18/pio8	20	pcs5_n/a1/pio3	60
a19/pio9	19	pcs6_n/a2/pio2	59
ad0	78	rd_n	6
ad1	80	res_n	71
ad2	82	rfsh2_n/aden_n	4
ad3	84	rxd/pio28	99
ad4	86	s0_n	11
ad5	88	s1_n	10
ad6	91	s2_n	9
ad7	94	s6/lock_n/clkdiv2/pio29	96
ale	7	sclk/pio20	3
ao8	79	sdata/pio21	100
ao9	81	sden0/pio22	2
ao10	83	sden1/pio23	1
ao11	85	srdy/pio6	46
ao12	87	tmrin0/pio11	75
ao13	90	tmrin1/pio0	72
ao14	93	tmrout0/pio10	74
ao15	95	tmrout1/pio1	73
ardy	8	txd/pio27	98
clkouta	16	ucs_n/once1_n	57
clkoutb	17	uzi_n/pio26	97
den_n/ds_n/pio5	49	v_{CC}	15
drq0/pio12	77	v_{CC}	21
drq1/pio13	76	v_{CC}	38
dt/r_n/pio4	48	v_{CC}	61
gnd	12	v_{CC}	67
gnd	18	v_{CC}	92
gnd	41	wb_n	42
gnd	42	wr_n	5
gnd	64	x1	13
gnd	70	x2	14

IA188EM 100-Pin TQFP



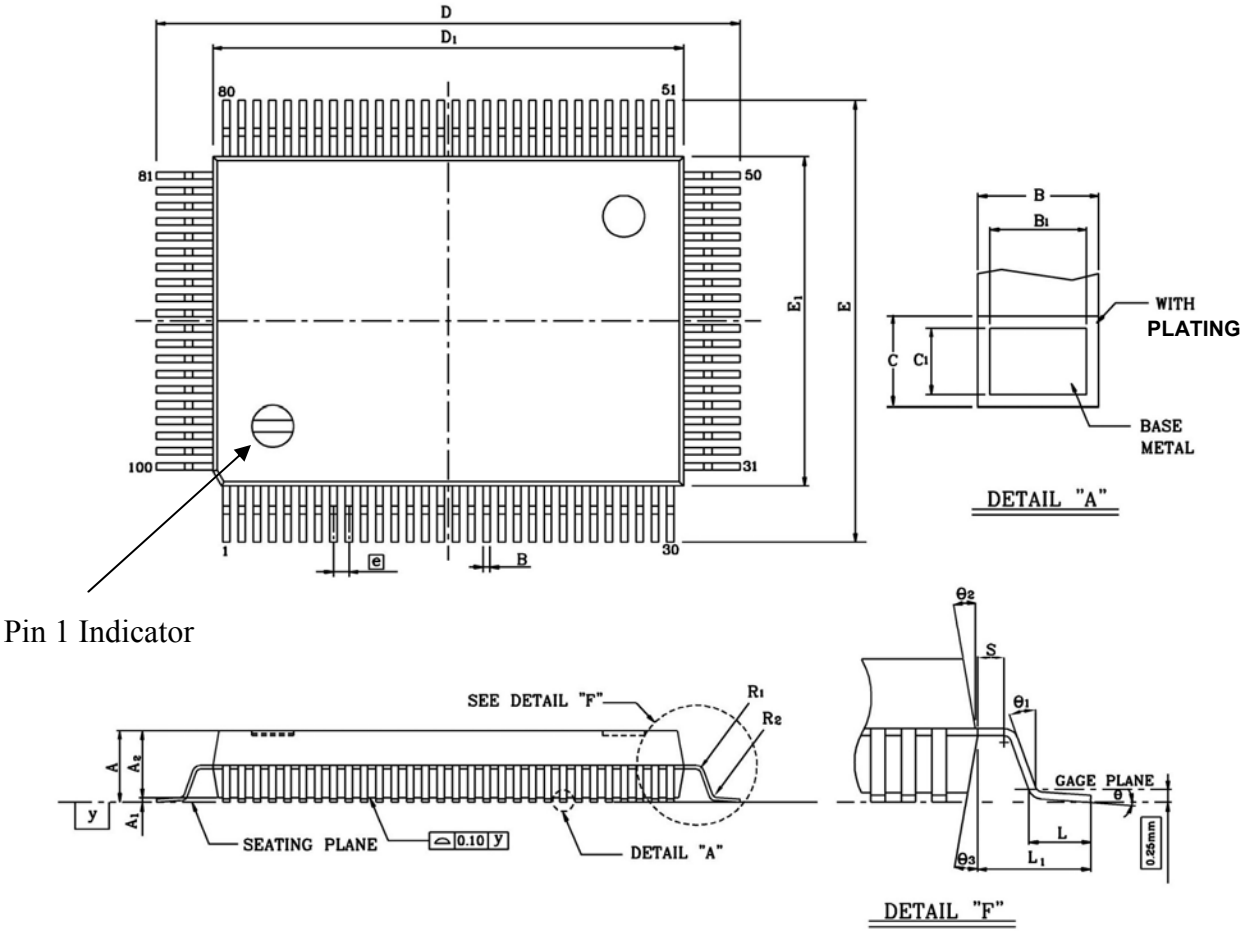
IA188EM 100-Pin TQFP Pin Assignments (sorted by Pin number)

Pin #	Name	Pin #	Name
1	ad0	51	a11
2	ao8	52	a10
3	ad1	53	a9
4	ao9	54	a8
5	ad2	55	a7
6	ao10	56	a6
7	ad3	57	a5
8	ao11	58	a4
9	ad4	59	a3
10	ao12	60	a2
11	ad5	61	v_{CC}
12	gnd	62	a1
13	ao13	63	a0
14	ad6	64	gnd
15	v_{CC}	65	gnd
16	ao14	66	wb_n
17	ad7	67	hlda
18	ao15	68	hold
19	s6/clkdiv2/pio29	69	srdy/pio6
20	uzi_n/pio26	70	nmi
21	txd/pio27	71	dt/r_n/pio4
22	rxd/pio28	72	den_n/pio5
23	sdata/pio21	73	mcs0_n/pio14
24	sden1/pio23	74	mcs1_n/pio15
25	sden0/pio22	75	int4/pio30
26	sclk/pio20	76	int3/inta1_n/irq
27	rfsh2_n/aden_n	77	int2/inta0_n/pio31
28	wr_n	78	int1/select_n
29	rd_n	79	int0
30	ale	80	ucs_n/once1_n
31	ardy	81	lcs_n/once0_n
32	s2_n	82	pcs6_n/a2/pio2
33	s1_n	83	pcs5_n/a1/pio3
34	s0_n	84	v_{CC}
35	gnd	85	pcs3_n/pio19
36	x1	86	pcs2_n/pio18
37	x2	87	gnd
38	v_{CC}	88	pcs1_n/pio17
39	clkouta	89	pcs0_n/pio16
40	clkoutb	90	v_{CC}
41	gnd	91	mcs2_n/pio24
42	a19/pio9	92	mcs3_n/rfsh_n/pio25
43	a18/pio8	93	gnd
44	v_{CC}	94	res_n
45	a17/pio7	95	tmrin1/pio0
46	a16	96	tmrout1/pio1
47	a15	97	tmrout0/pio10
48	a14	98	tmrin0/pio11
49	a13	99	drq1/pio13
50	a12	100	drq0/pio12

Pin Name	Number	Pin Name	Number
a0	63	gnd	93
a1	62	hlda	67
a2	60	hold	68
a3	59	int0	79
a4	58	int1/select_n	78
a5	57	int2/inta0_n/pio31	77
a6	56	int3/inta1_n/irq	76
a7	55	int4/pio30	75
a8	54	lcs_n/once0_n	81
a9	53	mcs0_n/pio14	73
a10	52	mcs1_n/pio15	74
a11	51	mcs2_n/pio24	91
a12	50	mcs3_n/rfsh_n/pio25	92
a13	49	nmi	70
a14	48	pcs0_n/pio16	89
a15	47	pcs1_n/pio17	88
a16	46	pcs2_n/pio18	86
a17/pio7	45	pcs3_n/pio19	85
a18/pio8	43	pcs5_n/a1/pio3	83
a19/pio9	42	pcs6_n/a2/pio2	82
ale	30	rd_n	29
ad0	1	res_n	94
ad1	3	rfsh2_n/aden_n	27
ad2	5	rxd/pio28	22
ad3	7	s0_n	34
ad4	9	s1_n	33
ad5	11	s2_n	32
ad6	14	s6/lock_n/clkdiv2/pio29	19
ad7	17	sclk/pio20	26
ao8	2	sdata/pio21	23
ao9	4	sden0/pio22	25
ao10	6	sden1/pio23	24
ao11	8	srdy/pio6	69
ao12	10	tmrin0/pio11	98
ao13	13	tmrin1/pio0	95
ao14	16	tmrout0/pio10	97
ao15	18	tmrout1/pio1	96
ardy	30	txd/pio27	21
clkouta	39	ucs_n/once1_n	80
clkoutb	40	uzi_n/pio26	20
den_/pio5	72	v_{CC}	15
drq0/pio12	100	v_{CC}	38
drq1/pio13	99	v_{CC}	44
dt/r_n/pio4	71	v_{CC}	61
gnd	12	v_{CC}	84
gnd	35	v_{CC}	90
gnd	41	wb_n	66
gnd	64	wr_n	28
gnd	65	x1	36
gnd	87	x2	37

Physical Dimensions

PQFP 100

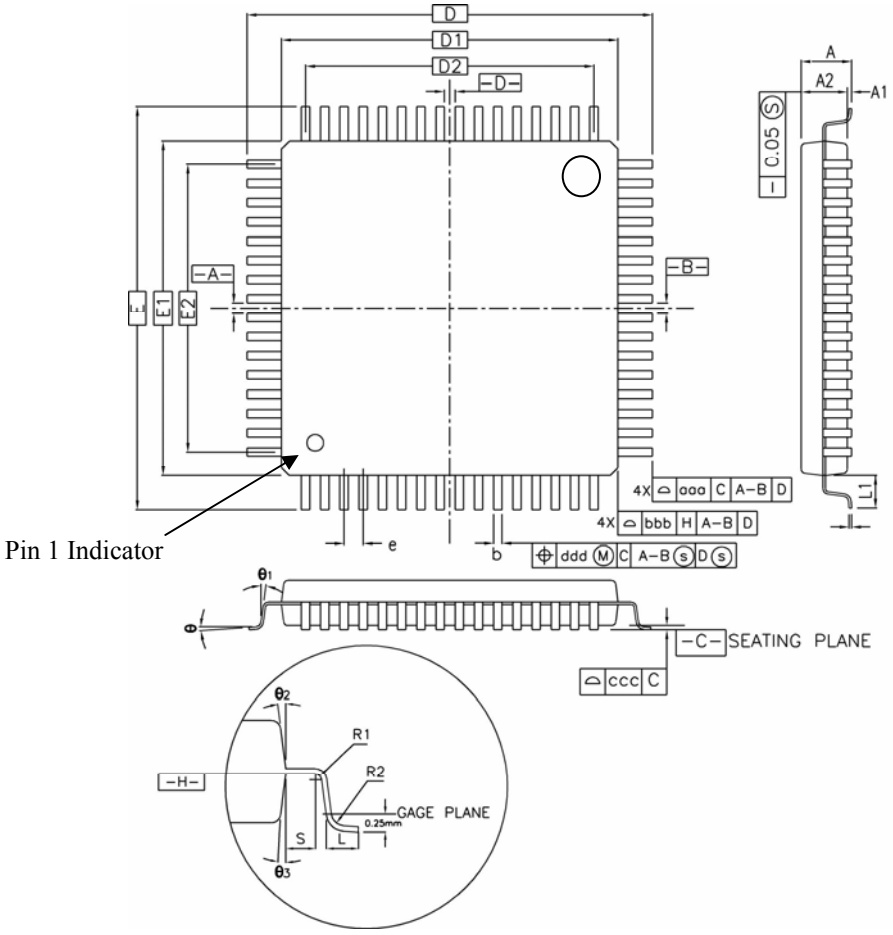


Symbol	Dimensions in Inches			Dimensions in mm		
	Minimum	Nominal	Maximum	Minimum	Nominal	Maximum
A	----	----	0.134	----	----	3.40
A_1	0.010	----	----	0.25	----	----
A_2	0.107	0.112	0.117	2.73	2.85	2.97
B	0.010	0.012	0.015	0.25	0.30	0.38
B_1	0.009	0.012	0.013	0.22	0.30	0.33
C	0.005	0.006	0.009	0.13	0.15	0.23
C_1	0.004	0.006	0.007	0.11	0.15	0.17
D	0.906	0.913	0.921	23.00	23.20	23.40
D_1	0.783	0.787	0.791	19.90	20.00	20.10
E	0.669	0.677	0.685	17.00	17.20	17.40
E_1	0.547	0.551	0.555	13.90	14.00	14.10
e		0.026 BSC			0.65 BSC	
L	0.029	0.035	0.041	0.73	0.88	1.03
L_1		0.063 BSC			1.60 BSC	
R_1	0.005	----	----	0.13	----	----
R_2	0.005	----	0.012	0.13	----	0.30
S	0.008	----	----	0.20	----	----
Y	----	----	0.004	----	----	0.10
θ	0°	----	7°	0°	----	7°
θ_1	0°	----	----	0°	----	----
θ_2	9°	10°	11°	9°	10°	11°
θ_3	9°	10°	11°	9°	10°	11°

NOTES

1. Dimensions D_1 and E_1 do not include mold protrusion. But mold mismatch is included. Allowable Protrusion is 0.25mm/0.010″ per side.
2. Dimension B does not include Dambar protrusion. Allowable protrusion is 0.08mm/0.003″ total in excess of the B dimension at maximum material condition. Dambar cannot be located on the lower radius or the foot.
3. Controlling dimension: millimeter.

TQFP 100



COTROL DIMENSIONS ARE IN MILLIMETERS.

SYMBOL	MILLIMETER			INCH		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	—	—	1.20	—	—	0.047
A1	0.05	—	0.15	0.002	—	0.006
A2	0.95	1.00	1.05	0.037	0.039	0.041
D	16.00 BSC.			0.630 BSC.		
D1	14.00 BSC.			0.551 BSC.		
E	16.00 BSC.			0.630 BSC.		
E1	14.00 BSC.			0.551 BSC.		
R2	0.08	—	0.20	0.003	—	0.008
R1	0.08	—	—	0.003	—	—
θ	0°	3.5°	7°	0°	3.5°	7°
θ1	0°	—	—	0°	—	—
θ2	11°	12°	13°	11°	12°	13°
θ3	11°	12°	13°	11°	12°	13°
c	0.09	—	0.20	0.004	—	0.008
L	0.45	0.60	0.75	0.018	0.024	0.030
L1	1.00 REF			0.039 REF		
S	0.20	—	—	0.008	—	—

Symbol	Dimensions in mm			Dimensions in Inches		
	Minimum	Nominal	Maximum	Minimum	Nominal	Maximum
b	0.17	0.20	0.27	0.007	0.008	0.011
e	0.50 BSC			0.02 BSC		
D2	12.00			0.472		
E2	12.00			0.472		
aaa	0.20			0.008		
bbb	0.20			0.008		
ccc	0.08			0.003		
ddd	0.08			0.003		

Ordering Information

Innovasic® Part Number	AMD® Part Number	Package Type	Temperature Grades
IA186EM-PQF100I (standard packaging) **IA186EM-PQF100I-R** (RoHS packaging)	**AM186EM-20KC\W** **AM186EM-20KI\W** **AM186EM-25KC\W** **AM186EM-25KI\W** **AM186EM-33KC\W** **AM186EM-40KC\W** **AM186EM-20KC\W** **AM186EM-20KI\W** **AM186EM-25KC\W** **AM186EM-25KI\W** **AM186EM-33KC\W** **AM186EM-40KC\W**	100-Pin Plastic Quad Flat Package (PQFP)	Commercial and Industrial
IA186EM-PTQ100I (standard packaging) **IA186EM-PTQ100I-R** (RoHS packaging)	**AM186EM-20VC\W** **AM186EM-25VC\W** **AM186EM-33VC\W** **AM186EM-40VC\W** **AM186EM-20VC\W** **AM186EM-25VC\W** **AM186EM-33VC\W** **AM186EM-40VC\W**	100-Pin Thin Quad Flat Package (TQFP)	Commercial and Industrial
IA188EM-PQF100I (standard packaging)	**AM188EM-20KC\W** **AM188EM-20KI\W** **AM188EM-25KC\W** **AM188EM-25KI\W** **AM188EM-33KC\W** **AM188EM-40KC\W**	100-Pin Plastic Quad Flat Package (PQFP)	Commercial and Industrial
IA188EM-PQF100I-R (RoHS packaging)	**AM188EM-20KC\W** **AM188EM-20KI\W** **AM188EM-25KC\W** **AM188EM-25KI\W** **AM188EM-33KC\W** **AM188EM-40KC\W**		
IA188EM-PTQ100I (standard packaging)	**AM188EM-20VC\W** **AM188EM-25VC\W** **AM188EM-33VC\W** **AM188EM-40VC\W**	100-Pin Thin Quad Flat Package (TQFP)	Commercial and Industrial
IA188EM-PTQ100I-R (RoHSpackaging)	**AM188EM-20VC\W** **AM188EM-25VC\W** **AM188EM-33VC\W** **AM188EM-40VC\W**		
Other packages and temperature grades may be available for an additional cost and lead time.			

Errata

Version -01

1) Problem: MCS chip select signals (MCS0-3) are intermittently suppressed. All other signals in bus cycle appear correct (i.e. address, data, write/read strobes).

   Analysis: Anomaly occurs when an access to I/O space is immediately followed by an MCS access. Given the way instruction prefetches naturally separate such accesses, one known scenario for this anomaly is via a DMA sequence. Possible combinations are: (1) DMA write to destination is followed by previously scheduled MCS read or write, (2) DMA from I/O space to MCS space.

   Customers using the UART DMA feature of the ES products may be particularly sensitive to this, because when the TX data register of the PCB is in I/O space, eventually an MCS access will be corrupted.

   Another known scenario occurs when auxiliary flash (containing executable code) is selected by an MCS signal and the PCB or PCS selects are in I/O space.

   The PCB register block and the PCS address spaces are the only areas that can be assigned to I/O space. The PCB register block is configured by bit 12 of the RELREG, and defaults to I/O space. PCS space is configured by bit 6 of the MPCS, and must be initialized by the user.

   Workaround: If possible, assign PCB and PCS address locations to memory space instead of I/O space.

2) Problem: IA186ES devices do not work in a 188ES socket.

   Analysis: The WHB pin should be sampled at reset to configure the bus width. This pin is always grounded in 188 applications, and floats high during reset in 186 applications. The bus width of the Innovasic devices are configured via in-package bonding.

   Workaround: Use IA188ES devices for 188 sockets.

3) Problem: Noise on TMROUT0 (PIO10) and TMROUT1 (PIO1) when in PIO output mode.

   Analysis: Only occurs when application is using HOLD/HLDA function, and either TMROUT pin is in PIO output mode. Improper logic allows the TMROUT pin to tri-state when HLDA is asserted. Analysis shows that UZI (PIO26), S6CLK2 (PIO29), DEN (PIO5), and DT_R (PIO4) may also be affected. PIO input modes and normal operation modes are not affected.

   Workaround: If possible, use a PIO pin other than those listed above when utilizing HOLD/HLDA feature. An external pullup/pulldown may also help.

4) Problem: An extra DMA cycle occurs after ending DMA transfers via a DMA control register write. In certain applications, this extra DMA cycle occurrence will hang the device because of DREQ/SRDY dependency.

Analysis: The string of DMA transfers is ended by writing x0004 to PCB address xCA or xDA while DMA request line is asserted. By this time, the sequence to initiate the DMA transfer cannot be suppressed or recalled, so the IA device executes the spurious transfer.

Workaround: None.

5) Problem: In the 186/188ES devices, the TB8 bit of the UART control register (offset x10 or x80) does not automatically reset after transmitting the initial word when using 9-bit formats (modes 2 or 3).

   Analysis: This feature is used to designate the "address" byte when using the UART in a psuedo-LAN configuration. The automatic reset of TB8 allows a convenient means to send a block of words with little software interaction.

   Workaround: Manually reset TB8 after detecting the end of the first transmitted word.

6) Problem: In the 186/188ES devices, the Power Save clock speed is not working correctly.

   Analysis: A logic error causes the device to incorrectly clear bits [2:0] of the PDCON when the device leaves power save mode by clearing bit 15 of the PDCON.

   Workaround: Every time the programmer desires to go into power save mode by setting bit 15 of the PDCON register, then bits [2:0] should also be set according to the desired clock divide factor. It should not be assumed that once written to, bits [2:0] will retain their values when entering and exiting the power save mode.

7) Problem: The device responds incorrectly to false start bits.

   Analysis: If a start bit is less than half width, the device should ignore this start bit completely (no data byte, no errors). Instead the device treats the data that follows as a valid byte, and generates a framing error.

   Workaround: Eliminate false start bits, or revise how the resulting framing error and extra byte are handled.

8) Problem: The UART is disabled when an external system generates a break condition.

   Analysis: The device should not be disabled when an external system generates a break condition, instead once the break condition is deasserted the UART should start receiving data. However, the UART in the Innovasic device is disabled by the externally generated break condition and can only receive data once the break flags (Bit 9: BRK0 of registers SP0ST and/or SP1ST) are cleared.

   Workaround: Ensure that every time an external break condition is acknowledged, that the break flag bits are cleared.

9) Problem: The MOV instruction does work when an attempt is made to load the CS register.

   Analysis: On the OEM AMD part a MOV CS, AX command loads CS with the contents of AX. The Innovasic part never loads CS with AX by use of a MOV instruction.

Workaround: To load the CS register use a far JMP command.

10) Problem: There is a difference in how priority of timer interrupts are asserted between the original AMD part and the Innovasic part.

Analysis: In the original AMD part, timer interrupts cannot be interrupted by another timer interrupt, even if the new timer interrupt is of a higher priority. The Innovasic part will interrupt a timer interrupt with a higher priority timer interrupt. Additionally, if a lower priority timer interrupt is interrupted with a higher priority timer interrupt and another occurrence of the lower priority interrupt occurs during the processing of the higher priority interrupt, upon execution of the EOI a new lower priority interrupt will be initiated, possibly orphaning the original lower priority timer interrupt.

Workaround: When using nested interrupts, at the beginning of the interrupt routine before the global interrupts are enabled with a CLI, timer interrupts must be specifically masked. At the end of the timer interrupt routine being serviced, you need to set the Interrupt Enable Bit in the Process Status Word to globally disable interrupts prior to clearing the timer interrupt being serviced.

11) Problem: UART will not respond to break condition if RXD is low when receiver is enabled.

Analysis: Detection of a break only occurs with a falling edge of RXD while receiver is enabled.

Workaround: None.

12) Problem: UART transmitter will not start if TX interrupt conditions exist prior to enabling transmitter.

Analysis: Priority of logic design inadvertently causes this lock up condition .

Workaround: Need to have transmitter enabled prior to any expected data transfers, or clear any spurious interrupts before enabling .

13) Problem: Lock up just after reset is released.

Analysis: Usually, the first instruction is a long jump to the start of the user's code. In this case, the compiler apparently inserted a short jump instruction with zero displacement before the expected long jump instruction. The OEM device stuttered, but recovered to execute the long jump, while the IA device instruction pointer was corrupted, causing the lock up. In summary, a short jump with zero displacement is a corner case that does not work in the IA device.

Workaround: Do not use a short jump instruction with zero displacement.

14) Problem: Intermittent startup.

Analysis: Processor either came out of reset normally, or would go into a series of watchdog timeouts. The addition of 10K ohm pullups to the WR_n and RD_n outputs seemed to solve the issue. Further analysis of the OEM device shows the presence of undocumented pullups on these pins,

which will pull them high when the reset condition tristates these pins. The Innovasic device does not include internal pullups on these pins allowing these outputs to float during reset.

Workaround: Add 10K ohm pullups to WR_n and RD_n pins to guarantee proper logic levels at the end of reset.

Version -03

1) Problem: There is a difference in how priority of timer interrupts are asserted between the original AMD part and the Innovasic part.

   Analysis: In the original AMD part, timer interrupts cannot be interrupted by another timer interrupt, even if the new timer interrupt is of a higher priority. The Innovasic part will interrupt a timer interrupt with a higher priority timer interrupt. Additionally, if a lower priority timer interrupt is interrupted with a higher priority timer interrupt and another occurrence of the lower priority interrupt occurs during the processing of the higher priority interrupt, upon execution of the EOI a new lower priority interrupt will be initiated, possibly orphaning the original lower priority timer interrupt.

   Workaround: When using nested interrupts, at the beginning of the interrupt routine before the global interrupts are enabled with a CLI, timer interrupts must be specifically masked. At the end of the timer interrupt routine being serviced, you need to set the Interrupt Enable Bit in the Process Status Word to globally disable interrupts prior to clearing the timer interrupt being serviced and unmask the appropriate timer interrupts.

2) Problem: Lock up just after reset is released.

   Analysis: Usually, the first instruction is a long jump to the start of the user's code. In this case, the compiler apparently inserted a short jump instruction with zero displacement before the expected long jump instruction. The OEM device stuttered, but recovered to execute the long jump, while the IA device instruction pointer was corrupted, causing the lock up. In summary, a short jump with zero displacement is a corner case that does not work in the IA device.

   Workaround: Do not use a short jump instruction with zero displacement.

3) Problem: Intermittent startup.

   Analysis: Processor either came out of reset normally, or would go into a series of watchdog timeouts. The addition of 10K ohm pullups to the WR_n and RD_n outputs seemed to solve the issue. Further analysis of the OEM device shows the presence of undocumented pullups on these pins, which will pull them high when the reset condition tristates these pins. The Innovasic device does not include internal pullups on these pins allowing these outputs to float during reset.

   Workaround: Add 10K ohm pullups to WR_n and RD_n pins to guarantee proper logic levels at the end of reset.

4) Problem: Timer Operation in continuous mode.

Analysis The timers (Timer0 and Timer1) do not function per the specification when set in continuous mode with no external timer input stimulus to initiate/continue count.

Workaround: None.