Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	-
Core Size	8/16-Bit
Speed	40MHz
Connectivity	UART/USART
Peripherals	DMA, PWM, WDT
Number of I/O	32
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	-
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-BQFP
Supplier Device Package	100-PQFP (20x14)
Purchase URL	https://www.e-xfl.com/product-detail/analog-devices/ia186em-pqf100i-r-03

CONVENTIONS

Arial Bold	Designates headings, figure captions, and table captions.
Blue	Designates hyperlinks (PDF copy only).
Italics	Designates emphasis or caution related to nearby information. Italics is also used to designate variables, refer to related documents, and to differentiate terms from other common words (e.g., "During refresh cycles, the *a* and *ad* busses may not have the same address during the address phase of the *ad* bus cycle." "The *hold* latency time [time between the *hold* and hlda] depends on the current processor activity when the hold is received.").

Table 1. IA186EM TQFP Numeric Pin Listing

Pin	Name
1	ad0
2	ad8
3	ad1
4	ad9
5	ad2
6	ad10
7	ad3
8	ad11
9	ad4
10	ad12
11	ad5
12	gnd
13	ad13
14	ad6
15	v_{cc}
16	ad14
17	ad7
18	ad15
19	s6/clkdiv2/pio29
20	uzi_n/pio26
21	txd
22	rxd
23	sdata/pio21
24	sden1/pio23
25	sden0/pio22
26	sclk/pio20
27	bhe_n/aden_n
28	wr_n
29	rd_n
30	ale
31	ardy
32	s2_n
33	s1_n
34	s0_n

Pin	Name
35	gnd
36	x1
37	x2
38	v_{cc}
39	clkouta
40	clkoutb
41	gnd
42	a19/pio9
43	a18/pio8
44	v_{cc}
45	a17/pio7
46	a16
47	a15
48	a14
49	a13
50	a12
51	a11
52	a10
53	a9
54	a8
55	a7
56	a6
57	a5
58	a4
59	a3
60	a2
61	v_{cc}
62	a1
63	a0
64	gnd
65	whb_n
66	wlb_n
67	hlda

Pin	Name
68	hold
69	srdy/pio6
70	nmi
71	dt/r_n/pio4
72	den_n/pio5
73	mcs0_n/pio14
74	mcs1_n/pio15
75	int4/pio30
76	int3/inta1_n/irq
77	int2/inta0_n/pio31
78	int1/select_n
79	int0
80	ucs_n/once1_n
81	lcs_n/once0_n
82	pcs6_n/a2/pio2
83	pcs5_n/a1/pio3
84	v_{cc}
85	pcs3_n/pio19
86	pcs2_n/pio18
87	gnd
88	pcs1_n/pio17
89	pcs0_n/pio16
90	v_{cc}
91	mcs2_n/pio24
92	mcs3_n/rfsh_n/pio25
93	gnd
94	res_n
95	tmrin1/pio0
96	tmrout1/pio1
97	tmrout0/pio10
98	tmrin0/pio11
99	drq1/pio13
100	drq0/pio12

Table 8. IA188EM PQFP Alphabetic Pin Listing

Name	Pin
a0	40
a1	39
a2	37
a3	36
a4	35
a5	34
a6	33
a7	32
a8	31
a9	30
a10	29
a11	28
a12	27
a13	26
a14	25
a15	24
a16	23
a17/pio7	22
a18/pio8	20
a19/pio9	19
ad0	78
ad1	80
ad2	82
ad3	84
ad4	86
ad5	88
ad6	91
ad7	94
ale	7
ao8	79
ao9	81
ao10	83
ao11	85
ao12	87

Name	Pin
ao13	90
ao14	93
ao15	95
ardy	8
clkouta	16
clkoutb	17
den_n/ds_n/pio5	49
drq0/pio12	77
drq1/pio13	76
dt/r_n/pio4	48
gnd	12
gnd	18
gnd	41
gnd	42
gnd	64
gnd	70
gnd	89
hlda	44
hold	45
int0	56
int1/select_n	55
int2/inta0_n/pwd/pio31	54
int3/inta1_n/irq	53
int4/pio30	52
lcs_n/once0_n	58
mcs0_n/pio14	50
mcs1_n/pio15	51
mcs2_n/pio24	68
mcs3_n/rfsh_n/pio25	69
nmi	47
pcs0_n/pio16	66
pcs1_n/pio17	65
pcs2_n/cts1_n/enrx1_n/pio18	63

Name	Pin
pcs3_n/rts1_n/rtr1_n/pio19	62
pcs5_n/a1/pio3	60
pcs6_n/a2/pio2	59
rd_n	6
res_n	71
rfsh2_n/aden_n	4
rxd/pio28	99
s0_n	11
s1_n	10
s2_n	9
s6/lock_n/clkdiv2/pio29	96
sclk/pio20	3
sdata/pio21	100
sden0/pio22	2
sden1/pio23	1
srdy/pio6	46
tmrin0/pio11	75
tmrin1/pio0	72
tmrout0/pio10	74
tmrout1/pio1	73
txd/pio27	98
ucs_n/once1_n	57
uzi_n/pio26	97
V_{CC}	15
V_{CC}	21
V_{CC}	38
V_{CC}	61
V_{CC}	67
V_{CC}	92
wb_n	42
wr_n	5
x1	13
x2	14

The select_n 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 can occur, however, the int0 pin must have already indicated an interrupt request has occurred.

2.2.18 int2/inta0_n/pio31—Maskable Interrupt Request 2 (asynchronous input)/Interrupt Acknowledge 0 (synchronous output)

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. To ensure that it is recognized, the assertion of the interrupt request must be maintained until it is handled. When int0 is configured to be in cascade mode, int2 changes its function to inta0_n.

The inta0_n 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.

2.2.19 int3/inta1_n/irq—Maskable Interrupt Request 3 (asynchronous input)/Interrupt Acknowledge 1 (synchronous output)/Interrupt Acknowledge (synchronous output)

The int3 pin provides an indication that an interrupt request has occurred. If 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. To ensure that it is recognized, the assertion of the interrupt request must be maintained until it is handled. When int1 is configured to be in cascade mode, int3 changes its function to inta1_n.

The inta1_n 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.

The signal on irq allows the microcontroller to output an interrupt request to the external master interrupt controller when the Interrupt Control Unit of the microcontroller is in slave mode.

2.2.20 int4/pio30—Maskable Interrupt Request 4 (asynchronous input)

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. To ensure that it is recognized, the assertion of the interrupt request must be maintained until it is handled.

2.2.33 s2_n–s0_n—Bus Cycle Status (synchronous outputs with tristate)

These three signals inform the system of the type of bus cycle in progress. The 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 presented in Table 10.

Table 10. Bus Cycle Types for s2_n, s1_n, and s0_n

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)

2.2.34 s6/clkdiv2_n/pio29—Bus Cycle Status Bit 6 (synchronous output)/Clock Divide by 2 (input with internal pull-up)

The s6 signal is high during the second and remaining cycle periods (i.e., $t_2 - t_4$), indicating that a DMA-initiated bus cycle is underway. The s6 is tristated during bus hold or reset.

If the clkdiv2_n signal is held low during power-on-reset, the microcontroller enters clock divide-by-2 mode. 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.

> *Note: If this pin is used as pio29 and configured as an input, care should be taken that it is not driven low during POR.*

Because this pin has an internal pull-up, it is not necessary to drive the pin high even though it defaults to an input PIO.

2.2.35 sclk—Serial Clock (synchronous outputs with tristate)

Because this pin provides a slave device with a synchronous serial clock it permits synchronization of the transmit and receive data exchanges between the slave and the microcontroller. The sclk is the result of dividing the internal clock by 2, 4, 8, or 16, depending on the contents of the Synchronous Serial Control (SSC) register Bits [5–4]. Accessing either the SSR or SSD registers activates the sclk for eight cycles. When sclk is not active, the microcontroller hold is high.

4.2 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% to 55 %, as a worst case, generated by the PLL obviating the need for an x2 external clock. A POR resets the PLL (see Figure 8).

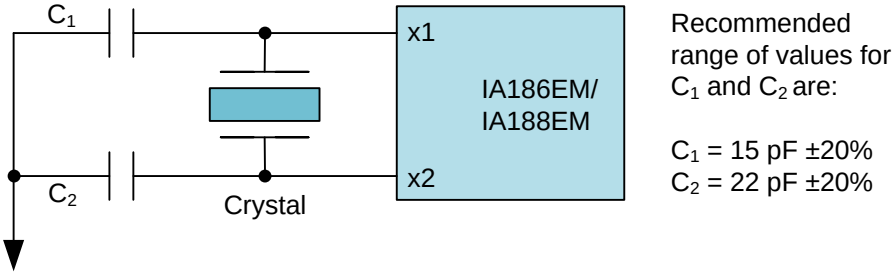

Figure 8. Crystal Configuration

4.3 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 (see Figure 9).

4.16.1 Interrupt Types

Table 14 presents interrupt names, types, vector table address, End-of-Interrupt (EOI) type, overall priority, and related instructions.

Table 14. Interrupt Types

Interrupt Name	Interrupt Type	Vector Table Address	EOI Type	Overall Priority	Related Instructions
Divide Error Exception[a]	00h	00h	NA	1	DIV, IDIV
Trace Interrupt[b]	01h	04h	NA	1A	All
Non-maskable Interrupt (NMI)	02h	08h	NA	1B	–
Breakpoint Interrupt[a]	03h	0ch	NA	1	INT3
INT0 Detected Overflow Exception[a]	04h	10h	NA	1	INT0
Array Bounds Exception[a]	05h	14h	NA	1	BOUND
Unused Opcode Exception[a]	06h	18h	NA	1	Undefined Opcodes
ESC Opcode Exception[a,c]	07h	1ch	NA	1	ESC Opcodes
Timer 0 Interrupt[d,e]	08h	20h	08h	2A	–
Timer 1 Interrupt[d,e]	12h	48h	08h	2B	–
Timer 2 Interrupt[d,e]	13h	4ch	08h	2C	–
Reserved	09h	24h	–	–	–
DMA 0 Interrupt[e]	0ah	28h	0ah	3	–
DMA 1 Interrupt[e]	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[f]	10h	40h	10h	9	–
Watchdog Timer Interrupt[f]	11h	44h	11h	9	–
Asynchronous Serial Port Interrupt[f]	14h	50h	14h	9	–
Reserved	15h–1fh	54h–7ch	–	–	–

Note: If the priority levels are not changed, the default priority level will be used for the interrupt sources.
[a]Instruction execution generates interrupts.
[b]Performed in the same manner as for the 8086 and 8088.
[c]An ESC opcode causes a trap.
[d]Because only one IRQ is generated for the three timers, they share priority level with other sources. The timers have an interrupt priority order among themselves (2A > 2B > 2C).
[e]These interrupt types are programmable in slave mode.
[f]Not available in slave mode.

Table 15. Default Status of PIO Pins at Reset

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[a]
4	dt/r_n	Normal operation[a]
5	den_n	Normal operation[a]
6	srdy	Normal operation[a]
7[b]	a17	Normal operation[a]
8[b]	a18	Normal operation[a]
9[b]	a19	Normal operation[a]
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[b,c]	uzi_n	Input with pull-up
27	txd	Input with pull-up
28	rxd	Input with pull-up
29[b,c]	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[a]
a18	8	Normal operation[a]
a19	9	Normal operation[a]
den_n/ds_n	5	Normal operation[a]
drq0	12	Input with pull-up
drq1	13	Input with pull-up
dt/r_n	4	Normal operation[a]
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[b,c]
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[d]
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

[a]Input with pullup option available when used as PIO.
[b]Emulators use these pins and also a15–a0, ad15–ad0 (IA186EM), ale, bhe_n (IA186EM), clkouta, nmi, res_n, and s2_n–s0_n.
[c]If bhe_n/aden_n (IA186EM) or rfsh_n/aden_n (IA188EM) is held low during POR, these pins will revert to normal operation.
[d]Input with pulldown option available when used as PIO.

- Bit [5]—P → Relative Priority. When set to 1, selects high priority for this channel relative to the other channel during simultaneous transfers.
- Bit [4]—TDRQ → Timer 2 Synchronization. When set to 1, enables DMA requests from Timer 2. When 0, disables them.
- Bit [3]—Reserved.
- Bit [2]—CHG → 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.
- Bit [1]—ST → Start/Stop DMA Channel. When 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.
- Bit [0]—Bn/W → Byte/Word Select. When set to 1, word transfers are selected. When 0, byte transfers are selected.

Note: Word transfers are not supported if the chip selects are programmed for 8-bit transfers. The IA188EM does not support word transfers

5.1.9 D1TC (0d8h) and D0TC (0c8h)

DMA Transfer Count Registers. 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 (see Table 25).

Table 25. DMA Transfer Count Registers

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

- Bits [15–0]—TC15–TC0 → DMA Transfer Count contains the transfer count for the respective DMA channel. Its value is decremented after each transfer.

5.1.10 D1DSTH (0d6h) and D0DSTH (0c6h)

The DMA DeSTination Address High Register. The 20-bit destination address consists of these 4 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

The value of the PIOMODE1 register is 0000h at reset (see Table 48).

Table 48. PMODE1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMODE31–PMODE16															

- Bits [15–0]—PMODE15–PMODE0 PIO Mode 0 Bits → For each bit, if the value is 1, the pin is configured as an input. If 0, an output. The values of these bits correspond to those in the PIO data registers and PIO Mode registers.

- Bits [15–0]—PMODE31–PMODE16 PIO Mode 1 Bits → For each bit, if the value is 1, the pin is configured as an input. If 0, an output. The values of these bits correspond to those in the PIO data registers and PIO Mode registers.

5.1.27 T1CON (05eh) and T0CON (056h)

Timer 0 and Timer 1 Mode and CONtrol Registers. These registers control the operation of Timer 0 and Timer 1, respectively. The value of the T0CON and T1CON registers is 0000h at reset (see Table 49).

Table 49. Timer 0 and Timer 1 Mode and Control Registers

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

- Bit [15]—EN 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 and can only be written if the INHn bit (Bit [14]) is set to 1 in the same operation.

- Bit [14]—INHn 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 0.

- Bit [13]—INT 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 the enable bit is then cleared before the interrupt is serviced, the interrupt request will remain.

- Bit [12]—RIU 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 0 when the Maxcount Compare A register is used.

5.1.30 T2CNT (060h), T1CNT (058h), and 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 (see Table 52).

Table 52. Timer Count Registers

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

- Bits [15–0]—TC15–TC0 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.

5.1.31 SPICON (044h) (Master Mode)

Serial Port Interrupt CONtrol Register. This register controls the operation of the asynchronous serial port interrupt source (SPI, Bit [10] in the Interrupt Request register). The value of this register is 001Fh at reset (see Table 53).

Table 53. Serial Port Interrupt Control Registers

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

- Bits [15–5]—Reserved → Set to 0.
- Bit [4]—Reserved → Set to 1.
- Bit [3]—MSK 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.
- Bits [2–0]—PR2–PR0 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 below.

- Bit [0]—TMR 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.

5.1.43 REQST (02eh) (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 (see Table 65).

Table 65. Interrupt Request Register (Slave Mode)

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

- Bits [15–6]—Reserved.
- Bit [5]—TMR2 Interrupt Requests → Setting this bit to 1 indicates that Timer 2 has a pending interrupt.
- Bit [4]—TMR1 Interrupt Requests → Setting this bit to 1 indicates that Timer 1 has a pending interrupt.
- Bits [3–2]—D1–D0 DMA Channel Interrupt Request → Setting either bit to 1 indicates that the respective DMA channel has a pending interrupt.
- Bit [1]—Reserved.
- Bit [0]—TMR0 Timer Interrupt Request → Setting this bit to 1 indicates that Timer 0 has a pending interrupt.

5.1.44 INSERV (02ch) (Master Mode)

IN-SERVice Register. 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 Special Fully nested mode for the int0 and int1 sources. The INSERV register contains 0000h on reset (see Table 66).

Table 77. Synchronous Serial Transmit Registers

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

- Bits [15–8]—Reserved.
- Bits [7–0]—SD7–SD0 → Data to be transmitted over the SDATA pin.

5.1.56 SSC (012h)

Synchronous Serial Control 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 (see Table 78).

Table 78. Synchronous Serial Control Registers

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

- Bits [15–6]—Reserved.
- Bits [5–4]—SCLKDIV 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 shown below.

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

- Bits [3–2]—Reserved.
- Bit [1]—DE1 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.
- Bit [0]—DE0 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.

5.1.57 SSS (010h)

Synchronous Serial Status 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 (see Table 79).

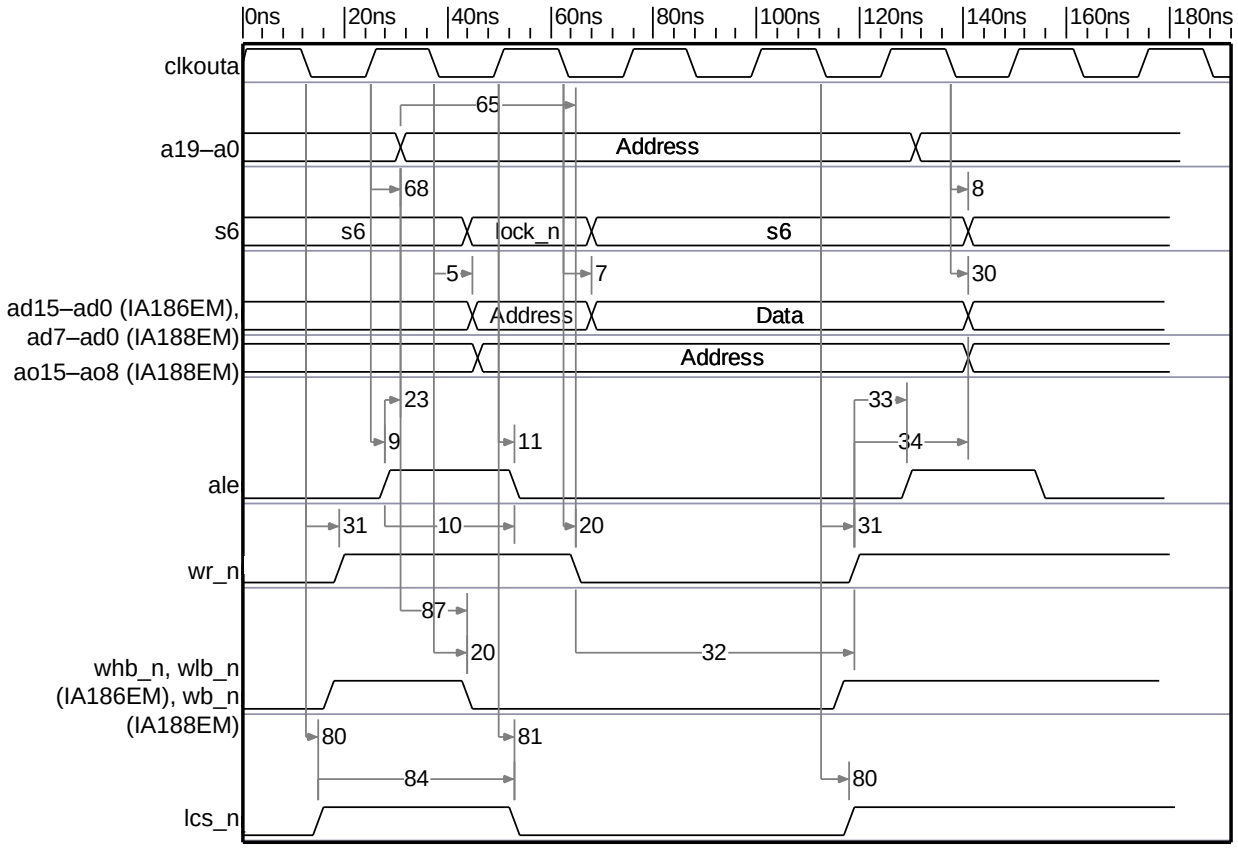

Figure 16. PSRAM Write Cycle

Table 90. 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

[a]In nanoseconds.

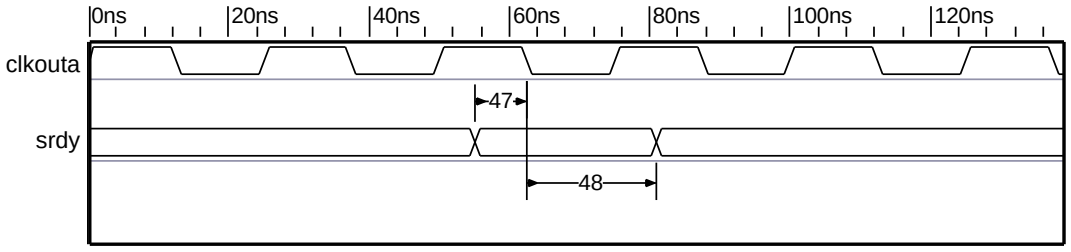

Figure 22. srdy—Synchronous Ready

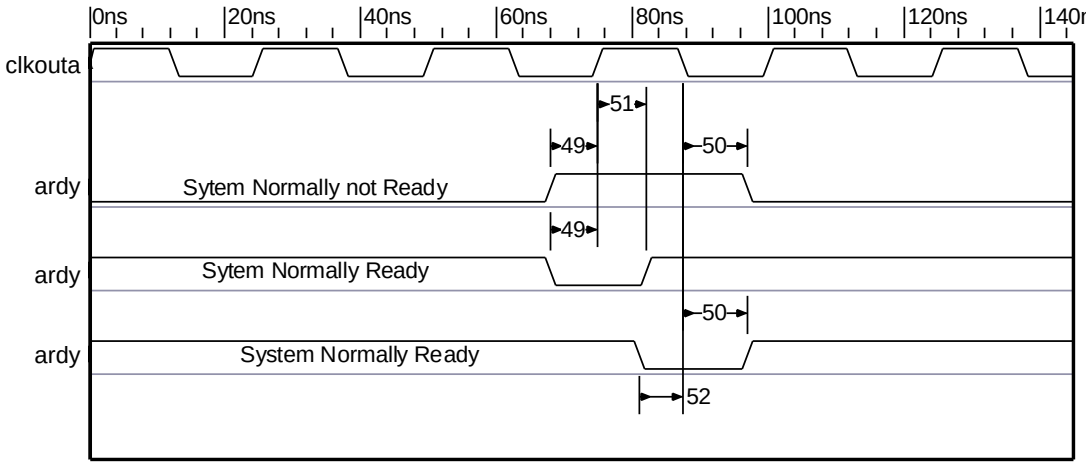

Figure 23. ardy—Asynchronous Ready

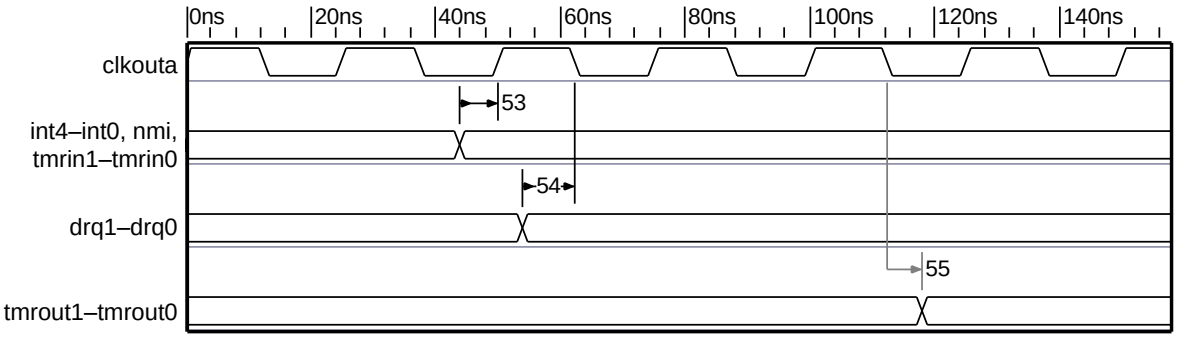

Figure 24. Peripherals

Table 92. Reset and Bus Hold Timing

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

[a]In nanoseconds.

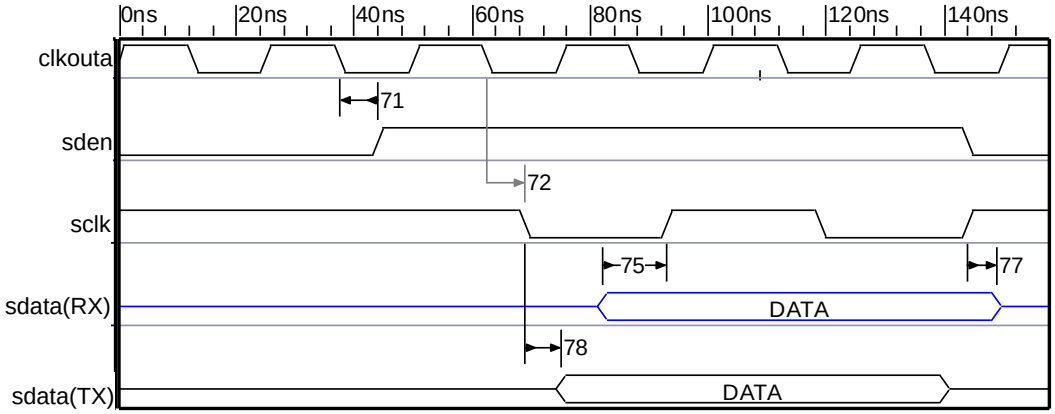

Figure 29. Synchronous Serial Interface

Table 94. Instruction Set Summary *(Continued)*

Instruction		Opcode - Hex			Clock Cycles		Flags Affected								
Mnemonic	Description	Byte 1	Byte 2	Bytes 3–6	IA186EM	IA188EM	O	D	I	T	S	Z	A	P	C
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									
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									

Table 94. Instruction Set Summary *(Continued)*

Instruction		Opcode - Hex			Clock Cycles		Flags Affected								
Mnemonic	Description	Byte 1	Byte 2	Bytes 3–6	IA186EM	IA188EM	O	D	I	T	S	Z	A	P	C
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	U	–	–	–	–	–	–	–	R
	Rotate 8 bits or r/m8 left CL times	D2	/0	–	5+n/ 17+n	5+n/ 17+n									
	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									
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

9. Errata

The following errata are associated with Version 03 of the IA186EM/IA188EM. A workaround to the identified problem has been provided where possible.

9.1 Errata Summary

Table 97 presents a summary of errata.

Table 97. Summary of Errata

Errata No.	Problem	Ver. 03
1	There is a difference in how priority of timer interrupts are asserted between the original AMD part and the Innovasic part.	Exists
2	Lock up just after reset is released.	Exists
3	Intermittent startup.	Exists
4	Timer Operation in continuous mode.	Exists
5	DMA interrupt will not bring device out of halt state.	Exists
6	Does not clear the interrupt request bit for INT0 upon entering the ISR.	Exists
7	There is a difference in how hardware handshaking for UARTs during a bus hold cycle is handled between the original AMD part and the Innovasic part.	Exists

9.2 Errata Detail

Errata No. 1

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

Description: 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 incident 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