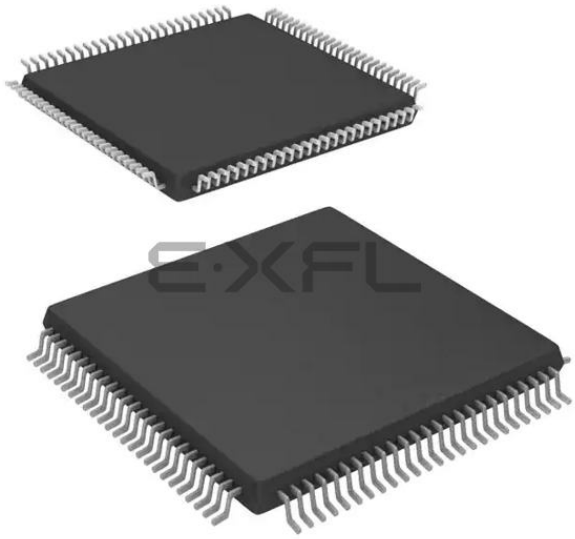

Welcome to E-XFL.COM

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-TQFP
Supplier Device Package	100-TQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/analog-devices/ia188emptq100ir03

ACRONYMS AND ABBREVIATIONS

AMD	Advanced Micro Devices
BIC	Bus Interface and Control
CDRAM	Count for Dynamic RAM
CSC	Chip Selects and Control
DA	Disable Address
DMA	Direct Memory Access
EOI	End of Interrupt
ISR	Interrupt Service Routine
LMCS	Low-Memory Chip Select
MC	Maximum Count
MDRAM	Memory Partition for Dynamic RAM
MILES™	Managed IC Lifetime Extension System
MMCS	Midrange Memory Chip Select
NMI	nonmaskable interrupt
PCB	peripheral control block
PIO	programmable I/O
PLL	phase-lock-loop
POR	power-on reset
PQFP	Plastic Quad Flat Package
PSRAM	Pseudo-Static RAM
RCU	Refresh Control Unit
RoHS	Restriction of Hazardous Substances
SFNM	Special Fully Nested mode
TQFP	Thin Quad Flat Package
UART	Universal Asynchronous Receiver-Transmitter
UMCS	Upper Memory Chip Select

Table 2. IA186EM TQFP Alphabetic Pin Listing

Name	Pin
a0	63
a1	62
a2	60
a3	59
a4	58
a5	57
a6	56
a7	55
a8	54
a9	53
a10	52
a11	51
a12	50
a13	49
a14	48
a15	47
a16	46
a17/pio7	45
a18/pio8	43
a19/pio9	42
ad0	1
ad1	3
ad2	5
ad3	7
ad4	9
ad5	11
ad6	14
ad7	17
ad8	2
ad9	4
ad10	6
ad11	8
ad12	10
ad13	13

Name	Pin
ad14	16
ad15	18
ale	30
ardy	30
bhe_n/aden_n	27
clkouta	39
clkoutb	40
den_n/pio5	72
drq0/pio12	100
drq1/pio13	99
dt/r_n/pio4	71
gnd	12
gnd	36
gnd	41
gnd	64
gnd	87
gnd	93
hlda	67
hold	68
int0	79
int1/select_n	78
int2/inta0_n/pio31	77
int3/inta1_n/irq	76
int4/pio30	75
lcs_n/once0_n	81
mcs0_n/pio14	73
mcs1_n/pio15	74
mcs2_n/pio24	91
mcs3_n/rfsh_n/pio25	92
nmi	70
pcs0_n/pio16	89
pcs1_npio	88
pcs2_n/pio18	86

Name	Pin
pcs3_n/pio19	85
pcs5_n/a1/pio3	83
pcs6_n/a2/pio2	82
rd_n	29
res_n	94
rxd/pio23	24
s0_n	34
s1_n	33
s2_n	32
s6/clkdiv2/pio29	19
sclk/pio20	26
sdata/pio21	23
sden0/pio22	25
sden1/pio23	24
srdy/pio6	69
tmrin0/pio11	98
tmrin1/pio0	95
tmrout0/pio10	97
tmrout1/pio1	96
txd/pio27	21
ucs_n/once1_n	80
uzi_n/pio26	20
V_{CC}	15
V_{CC}	38
V_{CC}	44
V_{CC}	61
V_{CC}	84
V_{CC}	90
whb_n	65
wlb_n	66
wr_n	28
x1	36
x2	37

Table 5. IA186EM PQFP Numeric Pin Listing

Pin	Name
1	sden1/pio23
2	sden0/pio22
3	sclk/pio20
4	bhe_n/aden_n
5	wr_n
6	rd_n
7	ale
8	ardy
9	s2_n
10	s1_n
11	s0_n
12	gnd
13	x1
14	x2
15	v_{cc}
16	clkouta
17	clkoutb
18	gnd
19	a19/pio29
20	a18/pio8
21	v_{cc}
22	a17/pio7
23	a16
24	a15
25	a14
26	a13
27	a12
28	a11
29	a10
30	a9
31	a8
32	a7
33	a6
34	a5

Pin	Name
35	a4
36	a3
37	a2
38	v_{cc}
39	a1
40	a0
41	gnd
42	whb_n
43	wlb_n
44	hlda
45	hold
46	srdy/pio6
47	nmi
48	dt/r_n/pio4
49	den_n/pio5
50	mcs0_n/pio14
51	mcs1_n/pio15
52	int4/pio30
53	int3/inta1_n/irq
54	int2/inta0_n/pio31
55	int1/select_n
56	int0
57	ucs_n/once1_n
58	lcs_n/once0_n
59	pcs6_n/a2/pio2
60	pcs5_n/a1/pio3
61	v_{cc}
62	pcs3_n/pio19
63	pcs2_n/pio18
64	gnd
65	pcs1_n/pio17
66	pcs0_n/pio16
67	v_{cc}

Pin	Name
68	mcs2_n/pio24
69	mcs3_n/rfsh_n/pio25
70	gnd
71	res_n
72	tmrin1/pio25
73	tmrout1/pio1
74	tmrout0/pio10
75	tmrin0/pio11
76	drq1/pio13
77	drq0/pio12
78	ad0
79	ad8
80	ad1
81	ad9
82	ad2
83	ad10
84	ad3
85	ad11
86	ad4
87	ad12
88	ad5
89	gnd
90	ad13
91	ad6
92	v_{cc}
93	ad14
94	ad7
95	ad15
96	s6/clkdiv2_n/pio29
97	uzi_n/pio26
98	txd/pio27
99	rxd/pio28
100	sdata/pio21

Table 6. IA186EM 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
ad8	79
ad9	81
ad10	83
ad11	85
ad12	87
ad13	90

Name	Pin
ad14	93
ad15	95
ale	7
ardy	8
bhe_n/aden_n	4
clkouta	16
clkoutb	17
den_n/pio5	49
drq0/pio12	77
drq1/pio13	76
dt/r_n/pio4	48
gnd	12
gnd	18
gnd	41
gnd	64
gnd	70
gnd	89
hlda	44
hold	45
int0	56
int1/select_n	55
int2/inta0_n/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/pio18	63

Name	Pin
pcs3_n/pio19	62
pcs5_n/a1/pio3	60
pcs6_n/a2/pio2	59
rd_n	6
res_n	71
rxd/pio28	99
s0_n	11
s1_n	10
s2_n	9
s6/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
whb_n	42
wlb_n	43
wr_n	5
x1	13
x2	14

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

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

Holding aden_n low during POR, 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.

2.2.8 clkouta—Clock Output A (synchronous output)

This pin is the internal clock output to the system. Bits [9–8] and Bits [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). The clkouta can be used as a full-speed clock source in power-save mode. The AC timing specifications that are clock-related refer to clkouta, which remains active during reset and hold conditions.

2.2.9 clkoutb—Clock Output B (synchronous output)

This pin is an additional clock output 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). The clkoutb remains active during reset and hold conditions.

2.2.10 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.

2.2.11 drq1/pio12–drq0/pio13—DMA Requests (synchronous level-sensitive inputs)

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.

2.2.12 dt/r_n/pio4—Data Transmit or Receive (synchronous output with tristate)

The microcontroller 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.

2.2.13 gnd—Ground

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

Note: 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.

Note: Caution is advised in the use of the DA bit in the LMCS or UMCS registers when overlapping an additional chip select with either the lcs_n or ucs_n. 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 for which 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, the respective ready- and wait-state configurations for their chip-select function will be in effect regardless of the function for which these two pins are 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 them must agree with those for any overlapping chip selects as though they were configured as chip selects.

Although pcs4_n is not available as an external pin, it has ready- and wait-state logic and must follow the rules for overlapping chip-selects. Conversely, pins pcs6_n and pcs5_n have ready- and wait-state logic that is disabled when configured as address bits a2 and a1, respectively.

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

4.11 Upper Memory Chip Select

The ucs_n chip select is for the top of memory. On reset, the microcontroller begins fetching and executing instructions at memory location FFFF0h. As a result, upper memory is usually used for instruction memory. To this end, ucs_n is active on reset and has a memory range of 64 Kbytes (F0000h to FFFFFh) by default, along with external ready required and 3 wait states automatically inserted. The lower boundary of ucs_n is programmable to provide ranges of 64 to 512 Kbytes.

- 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.

4.23 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 IA186EM/ IA188EM 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.

4.24 Programmable I/O (PIO)

Thirty-two pins are programmable as I/O signals (PIO). Table 15 presents them in both numeric and alphabetic order. Because programming a pin as a PIO disables its normal function, it should be done only if the normal function is not required. A PIO pin can be programmed as an input or output with or without a weak pull-up or pull-down. A PIO pin can be also programmed as an open-drain output. Each PIO pin regains default status after a POR.

These default status settings may be changed as desired.

After POR, a19–a17, the three most significant bits of the address bus, start with their normal function, allowing the processor to begin fetching instructions from the boot address FFFF0h. Normal function is also the default setting for dt/r_n, den_n, and srdy after POR.

If the ad15–ad0 bus override is enabled, s6/clkdiv2_n and uzi_n automatically return to normal operation. The ad15–ad0 bus override is enabled if either the bhe_n/aden_n for the IA186EM or the rfsh2_n/aden_n for the IA188EM is held low during POR.

5. Peripheral Architecture

5.1 Control and Registers

The on-chip peripherals in the IA186EM/IA188EM 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 space and are treated as a functional unit. A list of these registers is presented in Table 16.

Although a named register may be 8 bits, write operations performed on the IA188EM should be 8-bit writes, resulting in 16-bit data transfers to the Peripheral Control Block (PCB) register. Only word reads should be performed to the PCB registers. If unaligned read and write accesses are performed on either the IA186EM or IA188EM, indeterminate behavior may result.

Note: Adhere to these directions while writing code to avoid errors.

5.1.1 RELREG (0feh)

The Peripheral Control Block RELocation REGister maps the entire Peripheral Control Block Register Bank to either I/O or memory space. In addition, RELREG contains a bit that places the interrupt controller in either master or slave mode. The RELREG contains 20ffh at reset (see Table 17).

Table 17. Peripheral Control Block Relocation Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	S/Mn	Reserved	IO/Mn	RA19– RA8											

- Bit [15]—Reserved.
- Bit [14]—S/Mn → When set to 1, this bit places the interrupt controller into slave mode. When 0, it is in master mode.
- Bit [13]—Reserved.
- Bit [12]—IO/Mn → When set to 1, the Peripheral Control Block is mapped into memory space. When 0, this bit maps the Peripheral Control Block Register Bank into IO space.
- Bits [11–0]—RA19–RA8 → Sets the base address (upper 12 bits) of the Peripheral Control Block Register Bank. RA7–RA0 default to 0. When Bit [12] (IO/Mn) is set to 1, RA19–RA16 are ignored.

5.1.2 RESCON (0f6h)

The RESet CONfiguration Register latches user-defined information present at specified pins at the rising edge of reset. The contents of this register are read-only and remain valid until the next reset. The RESCON contains user-defined information at reset (see Table 18).

Table 18. Reset Configuration Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RC15–RC0															

- Bits [15–0]—RC15–RC0 → At the rising edge of reset, the values of specified pins (ad15–ad0 for the IA186EM and ao15–ao8 and ad7–ad0 for the IA188EM) are latched into this register.

5.1.3 PRL (0f4h)

The Processor Release Level Register contains a code corresponding to the latest processor production release. The PRL is a Read-Only Register. The PRL contains 0400h (see Table 19).

- 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

pcs3_n–pcs0_n 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

- Bit [2]—R2 → Ready Mode. When set to 1, external ready is ignored. When 0, it is required. In each case the number of wait states is determined according to the pcs3_n–pcs0_n Wait-State Encoding shown above.

- Bits [1–0]—R1–R0 → Wait-State Value (see pcs3_n–pcs0_n Wait-State Encoding above). The pcs6_n–pcs5_n and pcs3_n–pcs0_n pins are multiplexed with the PIO pins. For these to function as chip selects, the PIO mode and direction settings for these pins must be set to 0 for normal operation.

5.1.17 LMCS (0a2h)

The Low-Memory Chip Select (LMCS) Register configures the LMCS 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 (see Table 33).

Table 33. Low-Memory Chip Select Register

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

- Bit [15]—Reserved → Set to 0.

- Bits [14–12]—UB2–UB0 → Upper Boundary. These bits define the upper boundary of memory accessed by the lcs_n chip select. The list below presents the possible block-size configurations (a 512-Kbyte maximum).

LMCS Block-Size Programming Values

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

- Bits [11–8]—Reserved → Set to 1.
- Bit [7]—DA → Disable Address. When set to 1, the address bus is disabled, providing some measure of power saving. When 0, the address is driven onto the address bus ad15–ad0 during the address phase of a bus cycle. This bit is set to 0 at reset.
  - If bhe_n/aden_n (IA186EM) is held at 0 during the rising edge of res_n, the address bus is always driven, regardless of the setting of DA.
- Bit [6]—PSE → PSRAM Mode Enable. When set to 1, PSRAM support for the lcs_n chip select memory space is enabled. The EDRAM, MDRAM, and CDRAM RCU 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.
- Bits [5–3]—Reserved → Set to 1.
- Bit [2]—R2 → Ready Mode. When set to 1, the external ready is ignored. When 0, it is required. The value of R1–R0 bits determines the number of wait states inserted.
- Bits [1–0]—R1–R0 → Wait-State Value. The value of these bits determines the number of wait states inserted into an access to the lcs_n memory area. This number ranges from 0 to 3 (R1–R0 = 00b to 11b).

5.1.18 UMCS (0a0h)

The Upper Memory Chip Select Register configures the UMCS pin, 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. The 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. The lower end of this upper memory range is programmable. The value of the UMCS register is F03Bh at reset (see Table 34).

- Bits [11–6]—Reserved → Set to 0.

- Bit [5]—MC 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. If preferred, this bit may be used by software polling rather than by interrupts to monitor timer status.

- Bit [4]—RTG 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 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.

- Bit [3]—P 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.

- Bit [2]—EXT External Clock Bit → This bit determines whether an external or internal clock is used. If EXT is 1, an external clock is used. If 0, an internal is used.

- Bit [1]—ALT 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 0, it 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.

- Bit [0]—CONT Continuous Mode Bit → When set to 1, the timer runs continuously. When 0, the timer stops after each count run and EN will be cleared. 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 stops counting.

5.1.28 T2CON (066h)

Timer 2 Mode and CONtrol Register. This register controls the operation of Timer 2. The value of the T2CON register is 0000h at reset (see Table 50).

Table 50. Timer 2 Mode and Control Registers

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN	INHn	INT	Reserved							MC	Reserved				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. 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 and can only be written if the INHn bit (Bit [14]) is set to 1 in the same operation.

Table 83. Read Cycle Timing

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

[a]In nanoseconds.

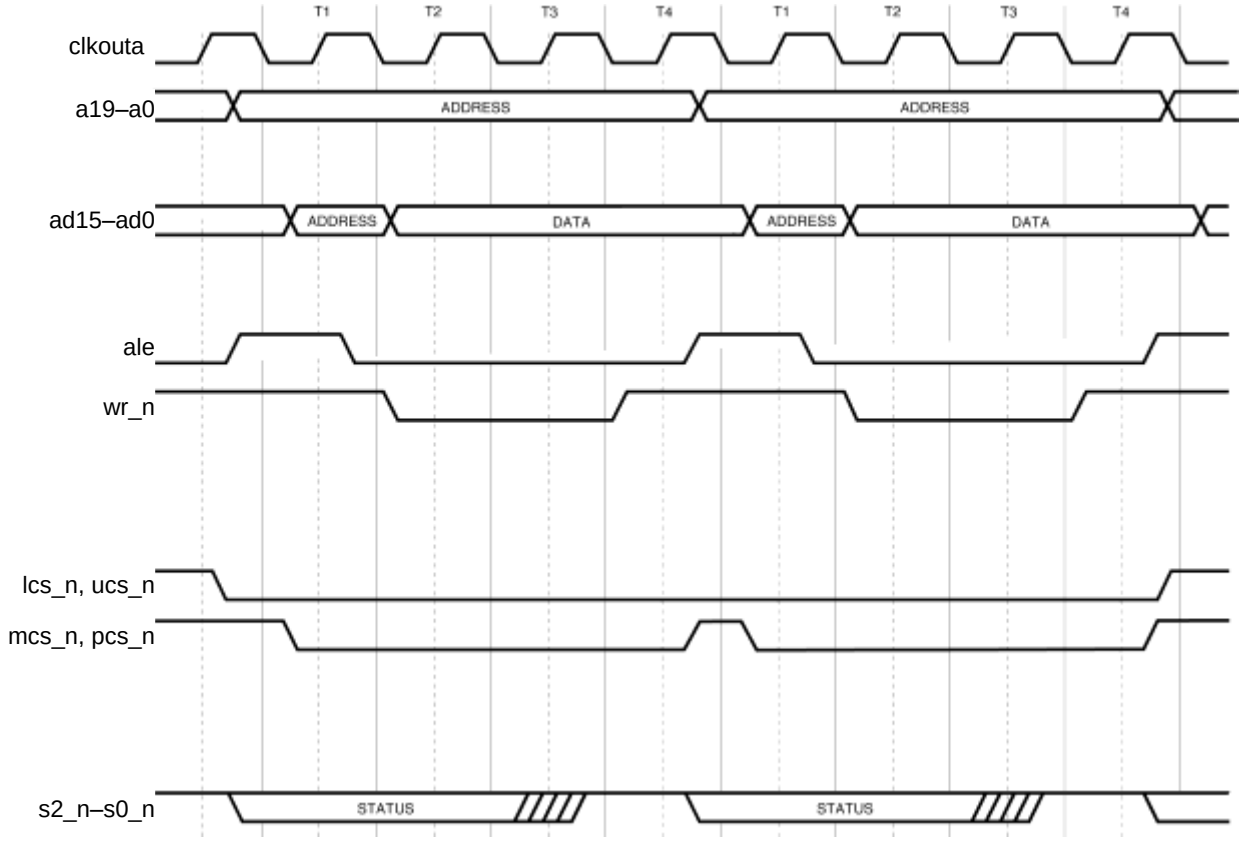

Figure 14. Multiple Write Cycles

Table 84. Write Cycle Timing

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

[a]In nanoseconds.

Table 86. PSRAM Write Cycle Timing

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

[a]In nanoseconds.

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
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									
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									

end of the timer interrupt routine being serviced, 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.

Errata No. 2

Problem: Lock up just after reset is released.

Description: 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 device instruction pointer was corrupted, causing the lockup. In summary, a short jump with zero displacement is a corner case that does not work in the device.

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

Errata No. 3

Problem: Intermittent startup.

Description: 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 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.

Errata No. 4

Problem: Timer operation in continuous mode.

Description: The timers (Timer 0 and Timer 1) do not function per the specification when set in continuous mode with no external timer input stimulus to initiate/continue count.

Workaround: None.

Errata No. 5

Problem: DMA interrupt will not bring device out of halt state.