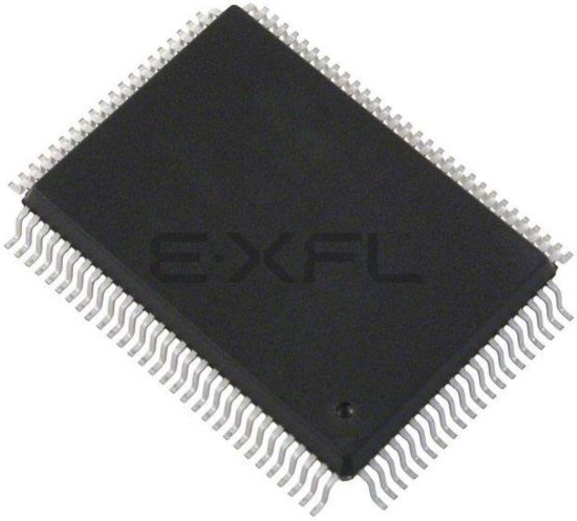
Analog Devices Inc. - IA188ESPQF100IR03 Datasheet



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	Active
Core Processor	-
Core Size	8/16-Bit
Speed	40MHz
Connectivity	EBI/EMI, 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/ia188espqf100ir03

TABLE OF CONTENTS

2.1.1 IA186ES TQFP Package

The pinout for the IA186ES TQFP package is as shown in Figure 1. The corresponding pinout is provided in Tables 1 and 2.

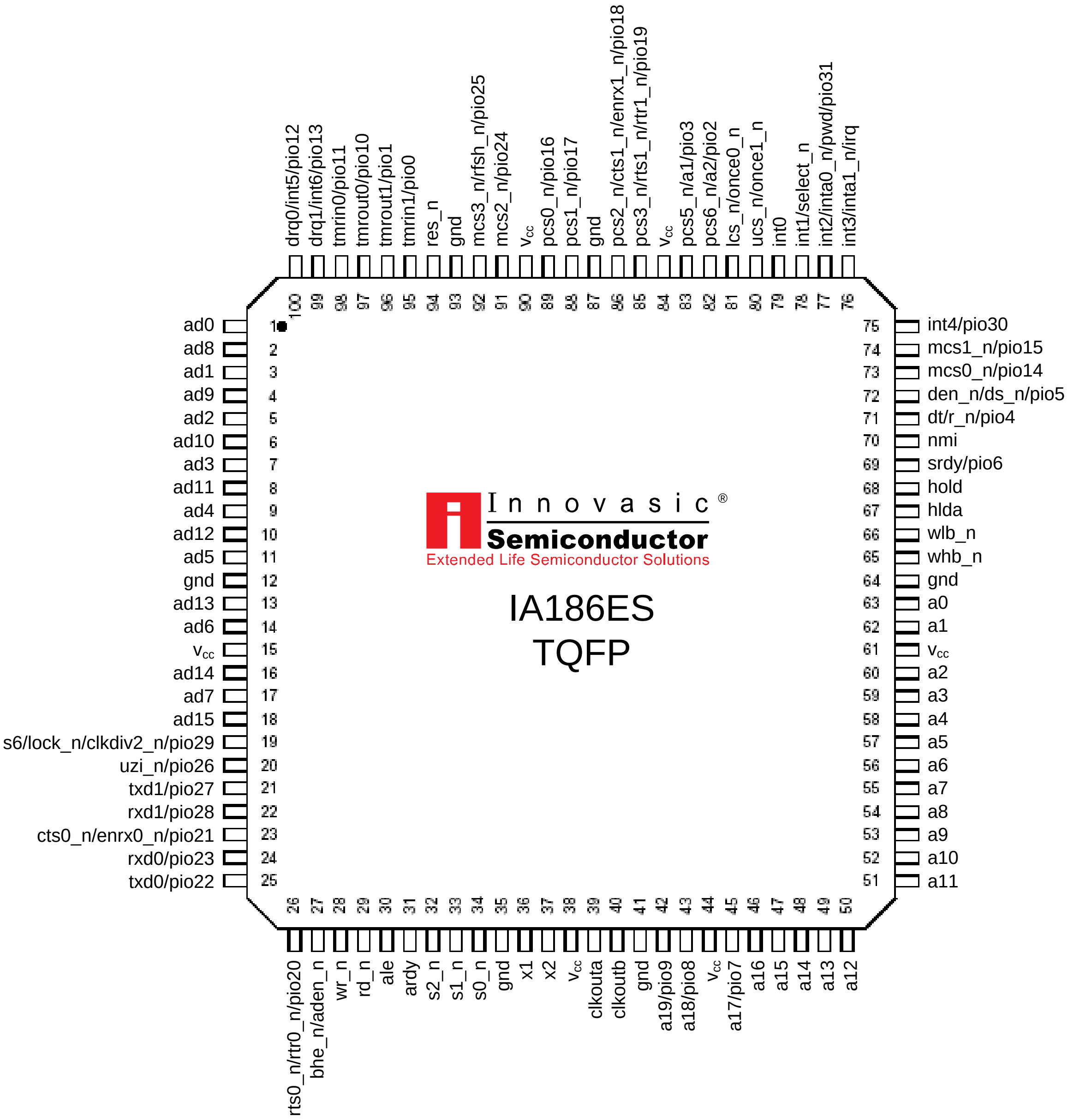


Figure 1. IA186ES TQFP Package Diagram

Table 2. IA186ES 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
ad14	16
ad15	18
ale	30
ardy	30
bhe_n/aden_n	27
clkouta	39
clkoutb	40
cts0_n/enrx0_n/pio21	23
den_n/ds_n/pio5	72
drq0/int5/pio12	100
drq1/int6/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/pwd/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/cts1_n/enrx1_n/pio18	86
pcs3_n/rts1_n/rtr1_n/pio19	85
pcs5_n/a1/pio3	83
pcs6_n/a2/pio2	82
rd_n	29
res_n	94
rts0_n/rtr0_n/pio20	26
rxd0/pio23	24
rxd1/pio28	22
s0_n	34
s1_n	33
s2_n	32
s6/lock_n/clkdiv2_n/pio29	19
srdy/pio6	69
tmrin0/pio11	98
tmrin1/pio0	95
tmrout0/pio10	97
tmrout1/pio1	96
txd0/pio22	25
txd1/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 8. IA188ES 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
ao13	90
ao14	93
ao15	95
ardy	8
bhe_n/aden_n	4
clkouta	16
clkoutb	17
cts0_n/enrx0_n/pio21	100
den_n/ds_n/pio5	49
drq0/int5/pio12	77
drq1/int6/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
lint4/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
pcs3_n/rts1_n/rtr1_n/pio19	62
pcs5_n/a1/pio3	60
pcs6_n/a2/pio2	59
rd_n	6
res_n	71
rts0_n/rtr0_n/pio20	3
rxd0/pio23	1
rxd1/pio28	99
s0_n	11
s1_n	10
s2_n	9
s6/lock_n/clkdiv2/pio29	96
srdy/pio6	46
tmrin0/pio11	75
tmrin1/pio0	72
tmrout0/pio10	74
tmrout1/pio1	73
txd0/pio22	2
txd1/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	43
wr_n	5
x1	13
x2	14

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. This would necessitate the use of ad0 as a determinant for the refresh cycle, rather than A0.

An additional signal is used for Pseudo-Static RAM (PSRAM) refreshes (see mcs3_n/rfsh_n pin description), aden_n. There is a weak internal pullup on bhe_n/aden_n , eliminating the need for an external pullup and reducing power consumption.

Holding aden_n high or letting it float during power-on reset (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 LMCS and UMCS registers. When the address function is selected, the memory address is placed on the a19–a0 pins.

When 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 on the rising edge of res_n and the condition of s6 and uzi_n default to their normal functions.

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 System Configuration (SYSCON) register control the output of this pin, which may be disabled, output the PLL frequency, or output the power save frequency (internal processor frequency after divisor). The clkouta may 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 with and output delayed with respect to clkouta. Bits [11–10] and Bits [2–0] of the SYSCON register control the output of this pin, which may be disabled, output the PLL frequency, or may output the power save frequency (internal processor frequency after divisor). The clkoutb may be used as a full-speed clock source in power-save mode and remains active during reset and hold conditions.

2.2.10 cts0_n/enrx0_n/pio21—Clear-to-Send 0/Enable-Receive-Request 0 (both are asynchronous inputs)

The cts0_n is the Clear-to-Send signal for asynchronous serial port 0, provided that Bit [4] (ENRX0) in the AUXCON register is 0, and Bit [9] (FC) in the SP0CT register is 1. The cts0_n controls the transmission of data from asynchronous serial port 0. When it is asserted, the transmitter begins transmitting the next frame of data. When it is not asserted, the data to be transmitted is held in the transmit register. This signal is checked only at the start of data frame transmission.

2.2.15 gnd—Ground

Depending on the package, six or seven pins connect the microcontroller to the system ground.

2.2.16 hlda—Bus Hold Acknowledge (synchronous output)

This pin is pulled high to signal the system that the microntroller has relinquished 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–s0, ad15–ad0, s6, a19–a0, bhe_n, whb_n, wlb_n, and dr/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 may 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 must be able to deassert hold to permit the microcontroller to access the bus.

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

2.2.18 int1/select_n—Maskable Interrupt Request 1/Slave Select (both are asynchronous inputs)

The int1 pin provides an indication that an interrupt request has occurred. 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. To ensure that it is recognized, the assertion of the interrupt request must be maintained until it is handled.

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. However, before this occurs, the int0 pin must have indicated an interrupt request has occurred.

present only during reset. If mcs0_n has been programmed as the chip select for the whole middle chip select address range, these pins may be used as PIOs.

2.2.25 mcs3_n/rfsh_n/pio25—Midrange Memory Chip Select (synchronous outputs with internal pullup)/Automatic Refresh (synchronous output)

The mcs3_n pin provides an indication that a memory access is in progress to the fourth region of the midrange memory block. The size of the Midrange Memory Block and its base address are programmable. The mcs3_n may be configured for either an 8- or 16-bit bus width for the IA186ES microcontroller by the Auxiliary Configuration Register (AUXCON Bit [1]) and is held high during bus hold. If mcs0_n has been programmed as the chip select for the whole middle chip select address range, this pin may be used as PIO. Furthermore, this pin has a weak pullup that is only present during reset.

The rfsh_n signal is timed for auto refresh to PSRAM or DRAM devices. The refresh pulse is only output 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.

2.2.26 nmi—Nonmaskable Interrupt (synchronous edge-sensitive input)

This is the highest priority interrupt signal and cannot be masked, unlike int6–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.

The nmi is not involved in the priority resolution process, which 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.

2.2.27 pcs1_n–pcs0_n (pio17–pio16)—Peripheral Chip Selects 1–0 (synchronous outputs)

These pins provide an indication that a memory access is under way for the second and first regions, respectively, of the peripheral memory block (I/O or memory address space). The base address of the peripheral memory block is programmable. The pcs3_n–pcs0_n are held 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 a 2X external clock. A POR resets the PLL (see Figure 8).

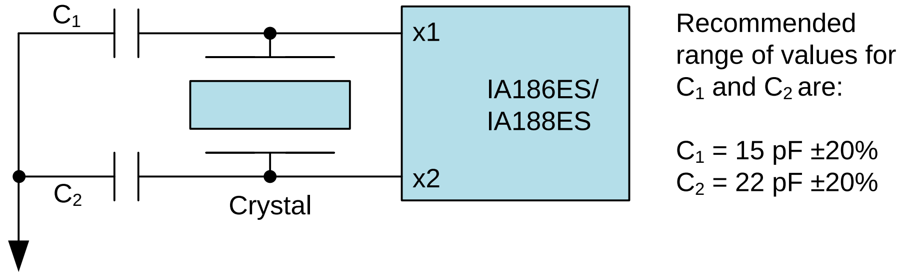


Figure 8. Crystal Configuration

4.3 System Clocks

If required, the internal oscillator may 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 (SYSCON register 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).

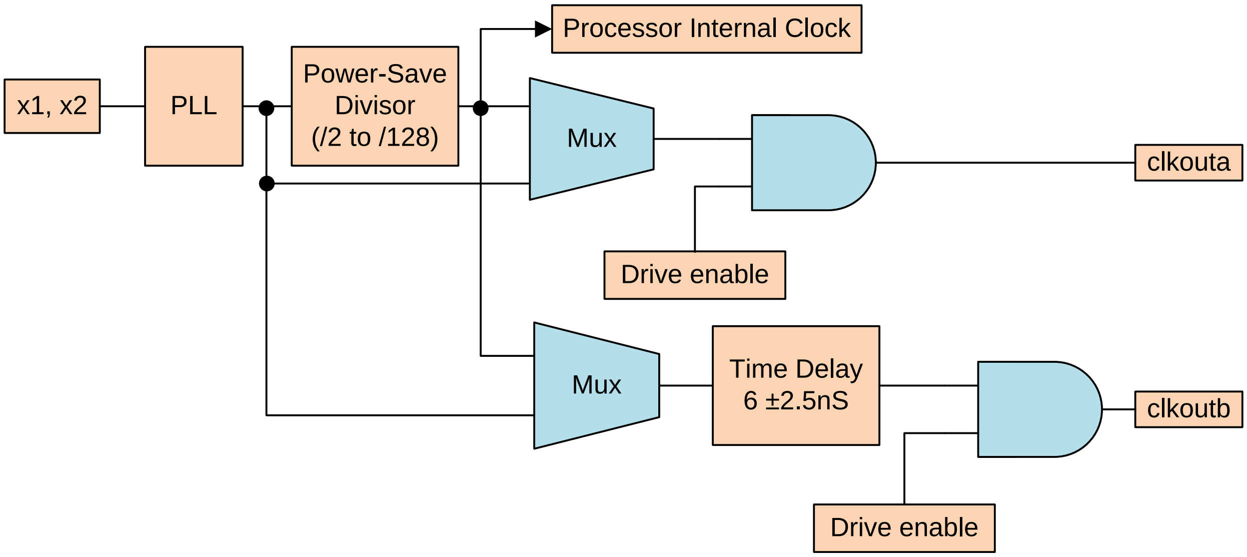


Figure 9. Organization of Clock

- Bit [1]—MSIZ (IA186ES only) → When set to 1, 8-bit data accesses are performed in middle chip-select (mcs_n) space and peripheral chip-select space (psc_n—but only if pcs_n is mapped to memory). When 0, 16-bit data accesses are performed.

- Bit [0]—IOSIZ (IA186ES only) → When set to 1, 8-bit data accesses are performed in all I/O space. When 0, 16-bit data accesses are performed.

5.1.5 SYSCON (0f0h)

The SYStem CONfiguration Register controls several miscellaneous system I/O and timing functions. The SYSCON contains 0000h at reset (see Table 21).

Table 21. System Configuration Register

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

- Bit [15]—PSEN → When set to 1, enables the power-save mode causing the internal operating clock to be divided by the value in F2–F0. External or internal interrupts clear PSEN automatically. Software interrupts and exceptions do not.

> *Note: The value of PSEN is not restored upon execution of an IRET instruction.*

- Bit [14]—MCSBIT → When set to 1, mcs0_n is active over the entire MCS range, thus freeing msc2 _n and mcs1_n to be used as PIO. When 0, it behaves normally.

- Bit [13]—DSDEN → When set to 1, the ds_n/den_n pin functions as ds_n. When 0, it functions as den_n. See the individual pin descriptions for details of data strobe (ds_n) mode versus data enable (den_n) mode.

- Bit [12]—PWD → When set to 1, the pulse width demodulator is enabled. When 0, it is disabled.

- Bit [11]—CBF → When set to 1, the clkoutb output follows the input crystal (PLL) frequency. When 0, it follows the internal clock frequency after the clock divider.

- Bit [10]—CBD → When set to 1, the clkoutb output is driven low. When 0, it is driven as an output per the CBF bit.

- Bit [9]—CAF → When set to 1, the clkouta output follows the input crystal (PLL) frequency. When 0, it follows the internal clock frequency after the clock divider.

- Bits [10–8]—Reserved.

- Bits [7–0]—COUNT → Control the timeout period for the WDT as follows:

$$T_{timeout} = 2^{exponent}/frequency \qquad \text{(Equation 1)}$$

Where:

$T_{timeout}$ = The WDT timeout period in seconds.
frequency = The processor frequency in hertz.
exponent = Is based upon count as shown below:

Bit [7]	Bit [6]	Bit [5]	Bit [4]	Bit [3]	Bit [2]	Bit [1]	Bit [0]	Exponent
0	0	0	0	0	0	0	0	NA
X	X	X	X	X	X	X	1	10
X	X	X	X	X	X	1	0	20
X	X	X	X	X	1	0	0	21
X	X	X	X	1	0	0	0	22
X	X	X	1	0	0	0	0	23
X	X	1	0	0	0	0	0	24
X	1	0	0	0	0	0	0	25
1	0	0	0	0	0	0	0	26

5.1.7 EDRAM (0e4h)

The Enable Dynamic RAM Refresh Control Register provides control and status for the refresh counter. The EDRAM register contains 0000h at reset (see Table 23).

Table 23. Enable Dynamic RAM Refresh Control Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN	0	0	0	0	0	0	T [8–0]								

- Bit [15]—EN → When set to 1, the refresh counter is enabled and msc3_n is configured to act as rfsh_n. Clearing EN clears the refresh counter and disables refresh requests. The refresh address is unaffected by clearing EN.

- Bits [14–9]—Reserved → These bits read back as 0.

- Bits [8–0]—T [8–0] → These bits hold the current value of the refresh counter. These bits are read-only.

- Bit [15]—DM/IOn → 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 0, it is in I/O space.

- Bit [14]—DDEC → Destination 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).

- Bit [13]—DINC → 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).

- Bit [12]—SM/IOn → Source Address Space Select selects memory or I/O space for the source address. When set to 1, the source address is in memory space. When 0, it is in I/O space.

- Bit [11]—SDEC → 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).

- Bit [10]—SINC → 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).

- Bit [9]—TC → Terminal Count. The DMA decrements the transfer count for each DMA transfer. When set to 1, the source or destination synchronized DMA transfers terminate when the count reaches 0. When 0, they do not terminate when the count reaches 0. Unsynchronized DMA transfers always end when the count reaches 0, regardless of the setting of this bit.

- Bit [8]—INT → Interrupt. When this bit is set to 1, the DMA channel generates an interrupt request on completion of the transfer count. However, for an interrupt to be generated, the TC bit must also be set to 1.

- Bits [7–6]—SYN1–SYN0 → Synchronization Type bits each select channel synchronization types as shown below. 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.

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 35).

Table 35. Low-Memory Chip Select Register

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	

- Bit [15]—Reserved. Set to 0.

- Bits [14–12]—UB [2–0] → Upper Boundary. These bits define the upper boundary of memory accessed by the lcs_n chip select. The LMCS Block-Size Programming Values shown below list 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 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 these bits determines the number of wait states inserted.

Table 57. INT2/INT3 Control Register

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

- Bits [15–5]—Reserved. Set to 0.
- Bit [4]—LTM 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 0, it is a rising-edge triggered interrupt. The interrupt int2 or int3 must remain active (high) until acknowledged.
- Bit [3]—MSK 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.
- Bit [2–0]—PR [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 upon reset. The values of PR2–PR0 are shown above.

5.1.37 I1CON (03ah) and I0CON (038h) (Master Mode)

INT0/INT1 CONtrol Register. 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 signals in cascade mode. The value of these registers is 000Fh at reset (see Table 58).

Table 58. INT0/INT1 Control Register

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

- Bits [15–7]—Reserved. Set to 0.
- Bit [6]—SPNM Special Fully Nested Mode → This bit enables fully-nested mode for int0 or int1 when set to 1.
- Bit [5]—C Cascade Mode → This bit enables cascade mode for int0 or int1 when set to 1.
- Bit [4]—LTM 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

Table 78. Numeric Key to Waveform Parameters *(Continued)*

No.	Name	Description	Min[a]	Max[a]
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	–
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	–
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	–
36	tCKIN	x1 Period	25	66
37	tCLCK	x1 Low Time	7.5	–
38	tCHCK	x1 High Time	7.5	–
39	tCKHL	x1 Fall Time	–	5
40	tCKLH	x1 Rise time	–	5
41	tDSHLH	ds_n Inactive to ale Inactive	tCLCH	–
42	tCLCL	clkouta Period	25	–
43	tCLCH	clkouta Low Time	tCLCL/2	–
44	tCHCL	clkouta High Time	tCLCL/2	–
45	tCH1CH2	clkouta Rise Time	–	3
46	tCL2CL1	clkouta Fall Time	–	3
47	tSRYCL	srdy Transition Setup Time	10	–
48	tCLSRY	srdy Transition Hold Time	3	–
49	tARYCH	ardy Resolution Transition Setup Time	9	–
50	tCLARX	ardy Active Hold Time	4	–
51	tARYCHL	ardy Inactive Holding Time	6	–

[a] In nanoseconds.

Table 82. PSRAM 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				
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	0	8
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	tCHCL

[a]In nanoseconds.

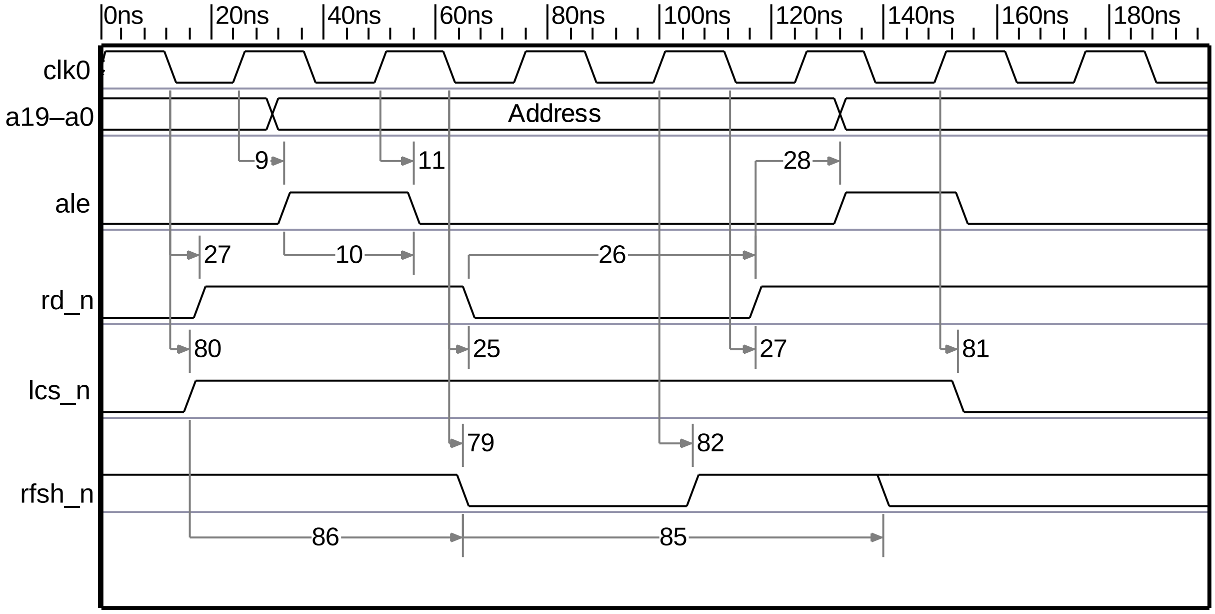


Figure 18. PSRAM Refresh Cycle

Table 83. PSRAM Refresh 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				
9	tCHLH	ale Active Delay	0	8
10	tLHLL	ale Width	tCLCH-5	–
11	tCHLL	ale Inactive Delay	0	8
Read/Write Cycle Timing Responses				
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	–
80	tCLCLX	lcs_n Inactive Delay	0	9
81	tCLCSL	lcs_n Active Delay	0	9
Refresh Cycle Timing Responses				
79	tCHRFD	clkouta High to rfsh_n Valid	0	12
82	tCLRF	clkouta High to rfsh_n Invalid	0	12
85	tRFCY	rfsh_n Cycle Time	6tCLCL	–
86	tLCRF	lcs_n Inactive to rfsh_n Active Delay	2tCLCL	–

[a] In nanoseconds.

Table 89. Instruction Set Summary *(Continued)*

Instruction		Opcode – Hex			Clock Cycles		Flags Affected								
Mnemonic	Description	Byte 1	Byte 2	Byte 3–6	IA186ES	IA188ES	O	D	I	T	S	Z	A	P	C
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									
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									

Refer to the key for abbreviations and an explanation of notation at the end of this table.

7.1 Key to Abbreviations Used in Instruction Set Summary Table

Abbreviations used in the Instruction Set Summary Table are explained below.

7.1.1 Operand Address Byte

The operand address byte is configured as shown below.

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

7.1.2 Modifier Field

The modifier field is defined below.

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

7.1.3 Auxiliary Field

The Auxiliary Field is defined below.

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

Note: When mod ≠ 11, depends on instruction.

7.2.2 Flags Affected After Instruction

Flags affected after instruction are shown below.

U	Undefined
-	Unchanged
R	Result dependent

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, 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 (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.