Innovasic®
Semiconductor
Extended Life Semiconductor Solutions

IA186ES/IA188ES
8-Bit/16-Bit Microcontrollers
Data Sheet

Published by Innovasic Semiconductor, Inc.
3737 Princeton Drive NE, Suite 130, Albuquerque, NM 87107

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

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.").

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
INSERV	In-Service
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
SYSCON	System Configuration Register
TQFP	Thin Quad Flat Package
UART	Universal Asynchronous Receiver-Transmitter
UMCS	Upper Memory Chip Select
WDT	Watchdog Timer

1. Introduction

The IA186ES/IA188ES is a form, fit, and function replacement for the original Advanced Micro Devices (AMD®) Am186ES/188ES family of microcontrollers. Innovasic produces replacement ICs using its MILES™, or Managed IC Lifetime Extension System, cloning technology that produces replacement ICs far more complex than "emulation" while ensuring they are compatible with the original IC. MILES captures the design of a clone so it can be produced even as silicon technology advances. MILES also verifies the clone against the original IC so that even the "undocumented features" are duplicated.

1.1 General Description

The IA186ES/IA188ES family of microcontrollers replaces obsolete AMD Am186ES/188ES devices, allowing customers to retain existing board designs, software compilers/assemblers, and emulation tools and thus avoid expensive redesign efforts.

The IA186ES/IA188ES microcontrollers are an upgrade for the 80C186/188 microcontroller designs, with integrated peripherals to provide increased functionality and reduce system costs. The Innovasic devices are designed to satisfy requirements of embedded products designed for telecommunications, office automation/storage, and industrial controls.

1.2 Features

- Pin-for-pin compatible with AMD Am186ES/188ES devices
- All features are retained, including:
  - A phase-lock loop (PLL) allowing same crystal/system clock frequency
  - An 8086/8088 instruction set with additional 186 instruction set extensions
  - A programmable interrupt controller
  - Two Direct Memory Access (DMA) channels
  - Three 16-bit timers
  - A wait-state generator and programmable chip select logic
  - A dedicated watchdog timer (WDT)
  - Two independent asynchronous serial ports (UARTs)
    - DMA capability
    - Hardware flow control
    - 7-, 8-, or 9-bit data capability
  - Pulse width demodulator feature
  - Up to 32 programmable I/O (PIO) pins
- A pseudo-static/dynamic RAM controller
- A fully static CMOS design
- 40-MHz operation at industrial operating conditions
- +5-VDC power supply

2. Packaging, Pin Descriptions, and Physical Dimensions

Information on the packages and pin descriptions for the IA186ES and the IA188ES is provided separately. Refer to sections, figures, and tables for information on the device of interest.

2.1 Packages and Pinouts

The Innovasic Semiconductor IA186ES and IA188ES microcontroller is available in the following packages:

- 100-Pin Thin Quad Flat Package (TQFP), equivalent to original SQFP package
- 100-Plastic Quad Flat Package (PQFP), equivalent to original PQFP package

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.

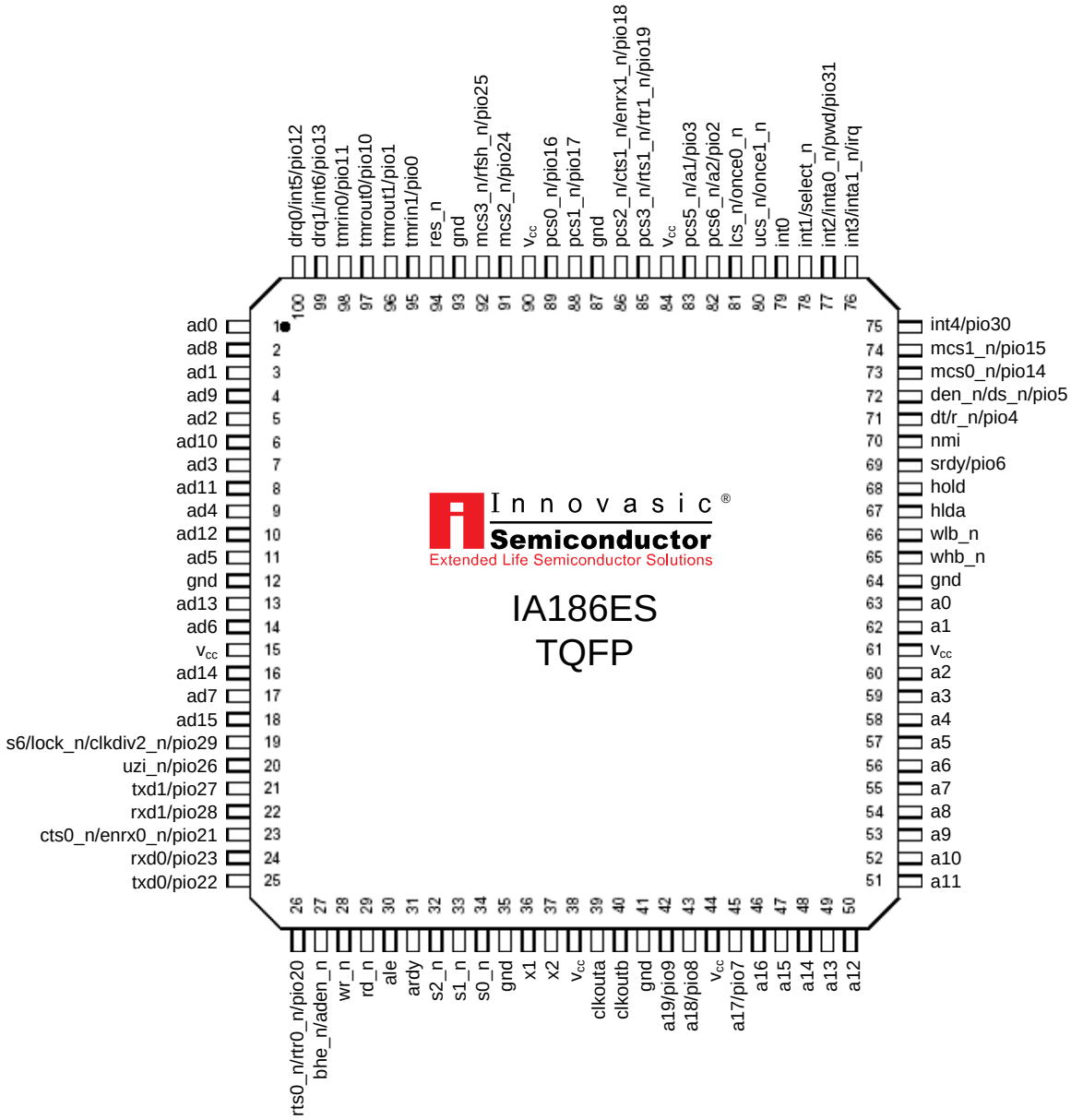


Figure 1. IA186ES TQFP Package Diagram

Table 1. IA186ES 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/lock_n/clkdiv2_n/pio29
20	uzi_n/pio26
21	txd1/pio27
22	rxd1/pio28
23	cts0_n/enrx0_n/pio21
24	rxd0/pio23
25	txd0/pio22
26	rts0_n/rtr0_n/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
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
68	hold
69	srdy/pio6
70	nmi
71	dt/r_n/pio4
72	den_n/ds_n/pio5
73	mcs0_n/pio14
74	mcs1_n/pio15
75	int4/pio30
76	int3/inta1_n/irq
77	int2/inta0_n/pwd/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/rts1_n/rtr1_n/pio19
86	pcs2_n/cts1_n/enrx1_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/int6/pio13
100	drq0/int5/pio12

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

Name	Pin
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

Name	Pin
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

2.1.2 IA188ES TQFP Package

The pinout for the IA186ES TQFP package is as shown in Figure 2. The corresponding pinout is provided in Tables 3 and 4.

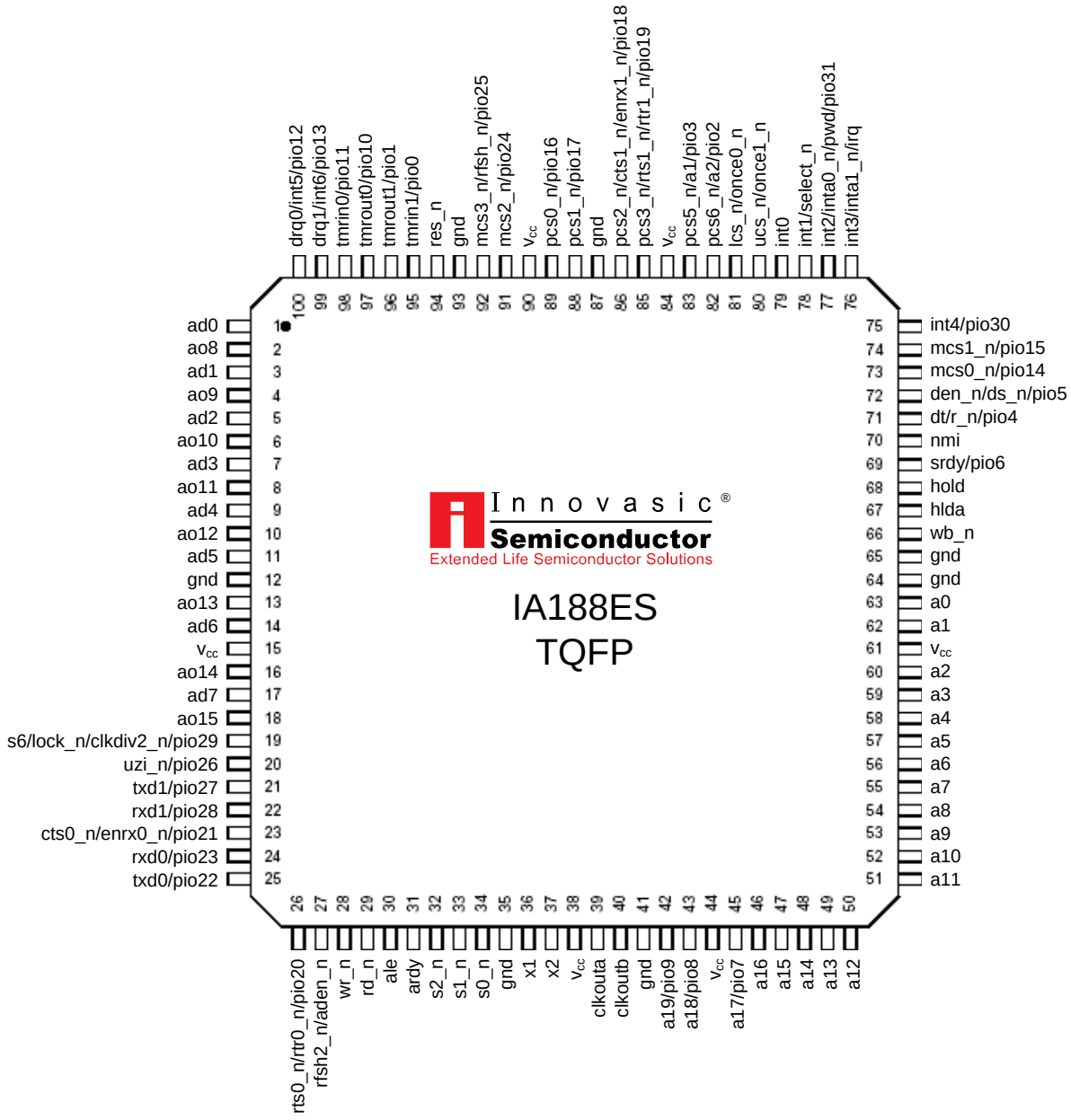


Figure 2. IA188ES TQFP Package Diagram

Table 3. IA188ES TQFP Numeric Pin Listing

Pin	Name
1	ad0
2	ao8
3	ad1
4	ao9
5	ad2
6	ao10
7	ad3
8	ao11
9	ad4
10	ao12
11	ad5
12	gnd
13	ao13
14	ad6
15	v_{CC}
16	ao14
17	ad7
18	ao15
19	s6/lock_n/clkdiv2_n/pio29
20	uzi_n/pio26
21	txd1/pio27
22	rxd1/pio28
23	cts0_n/enrx0_n/pio21
24	rxd0/pio23
25	txd0/pio22
26	rts0_n/rtr0_n/pio20
27	rfsh2_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	gnd
66	wb_n
67	hlda

Pin	Name
68	hold
69	srdy/pio6
70	nmi
71	dt/r_n/pio4
72	den_n/ds_n/pio5
73	mcs0_n/pio14
74	mcs1_n/pio15
75	int4/pio30
76	int3/inta1_n/irq
77	int2/inta0_n/pwd/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/rts1_n/rtr1_n/pio19
86	pcs2_n/cts1_n/enrx1_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/int6/pio13
100	drq0/int5/pio12

Table 4. IA188ES TQFP Alphabetic Pin Listing

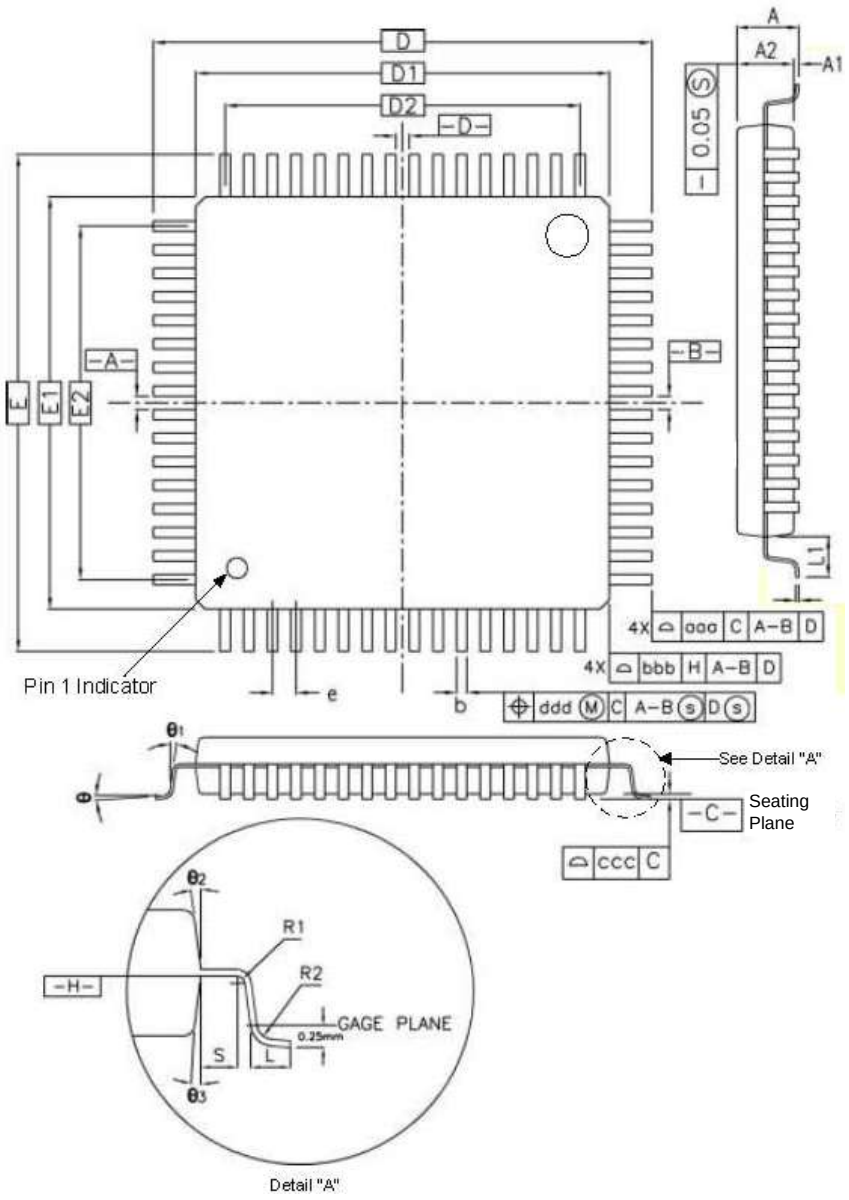
Name	Pin
a0	63
a1	62
a2	60
a3	59
a4	58
a5	57
a6	56
a7	55
a8	54
a9	53
a10	52
a11	51
a12	50
a13	49
a14	48
a15	47
a16	46
a17/pio7	45
a18/pio8	43
a19/pio9	42
ale	30
ad0	1
ad1	3
ad2	5
ad3	7
ad4	9
ad5	11
ad6	14
ad7	17
ao8	2
ao9	4
ao10	6
ao11	8
ao12	10

Name	Pin
ao13	13
ao14	16
ao15	18
ardy	30
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	35
gnd	41
gnd	64
gnd	65
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_n/pio17	88

Name	Pin
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
rfsh_n/aden_n	27
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
wb_n	66
wr_n	28
x1	36
x2	37

2.1.3 TQFP Physical Dimensions

The physical dimensions for the TQFP are as shown in Figure 3.



Legend:

Symbol	Millimeter Min	Nom	Max	Inch Min	Nom	Max
A	–	–	1.20	–	–	0.047
A1	0.05	–	0.15	0.002	–	0.006
A2	0.95	1.00	1.05	0.037	0.039	0.041
b	0.17	0.20	0.27	0.007	0.008	0.011
c	0.09	–	0.20	0.004	–	0.008
D	16.00 BSC.			0.630 BSC.		
D1	14.00 BSC.			0.551 BSC.		
D2	12.00			0.472		
e	0.50 BSC.			0.02 BSC.		
E	16.00 BSC.			0.630 BSC.		
E1	14.00 BSC.			0.551 BSC.		
E2	12.00			0.472		
L	0.45	0.60	0.75	0.018	0.024	0.030
L1	1.00 REF			0.039 REF		
R1	0.08	–	–	0.003	–	–
R2	0.08	–	0.20	0.003	–	0.008
S	0.20	–	–	0.008	–	–
θ	0°	3.5°	7°	0°	3.5°	7°
θ_1	0°	–	–	0°	–	–
θ_2	11°	12°	13°	11°	12°	13°
θ_3	11°	12°	13°	11°	12°	13°
Tolerances of Form and Position						
aaa	0.20			0.008		
bbb	0.20			0.008		
ccc	0.08			0.003		
ddd	0.08			0.003		

Note:
Control dimensions are in millimeters.

Figure 3. TQFP Package Dimensions

2.1.4 IA186ES PQFP Package

The pinout for the IA186ES PQFP package is as shown in Figure 4. The corresponding pinout is provided in Tables 5 and 6.

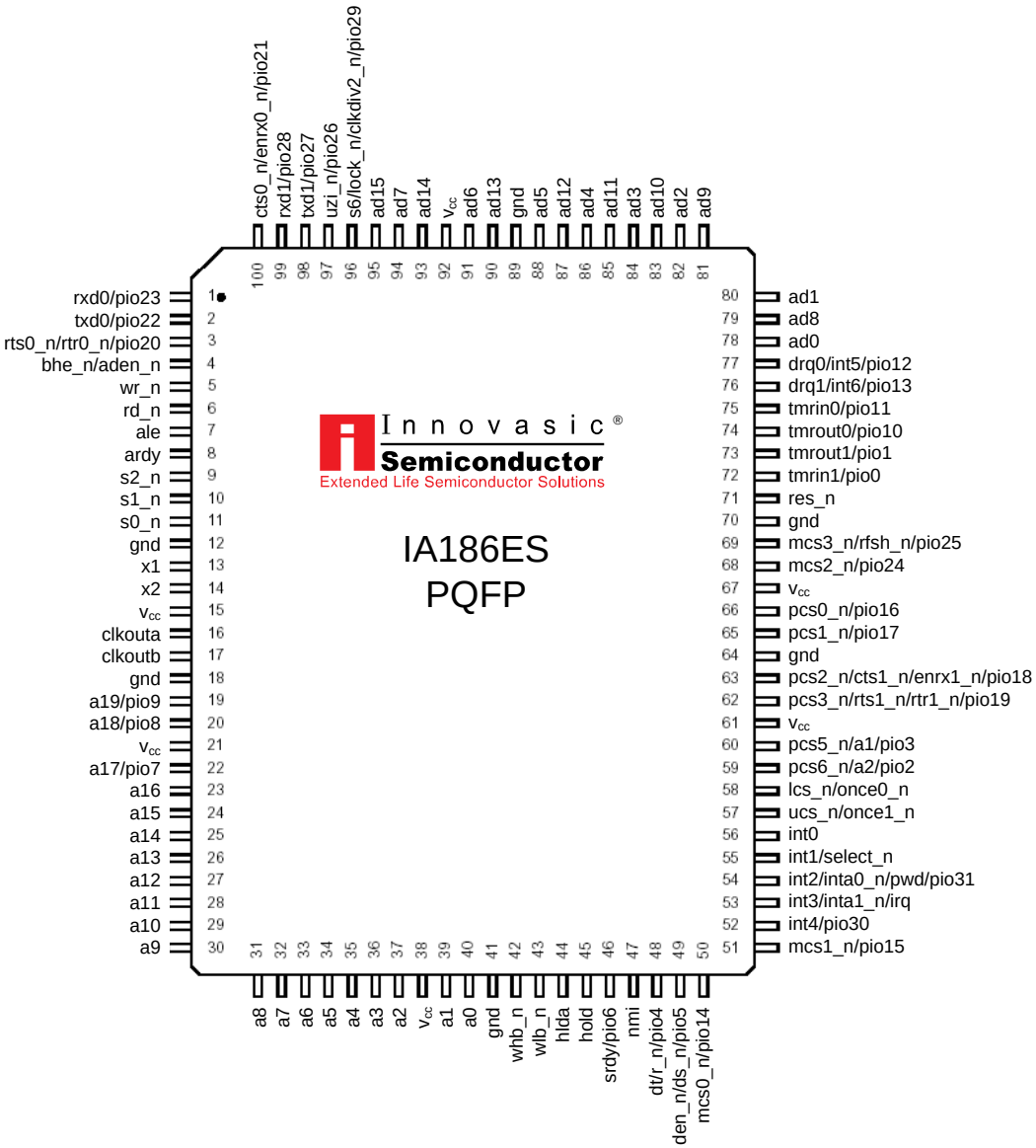


Figure 4. IA186ES PQFP Package Diagram

Table 5. IA186ES PQFP Numeric Pin Listing

Pin	Name
1	rxd0/pio23
2	txd0/pio22
3	rts0_n/rtr0_n/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/pio9
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/ds_n/pio5
50	mcs0_n/pio14
51	mcs1_n/pio15
52	int4/pio30
53	int3/inta1_n/irq
54	int2/inta0_n/pwd/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/rts1_n/rtr1_n/pio19
63	pcs2_n/cts1_n/enrx1_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/pio0
73	tmrout1/pio1
74	tmrout0/pio10
75	tmrin0/pio11
76	drq1/int6/pio13
77	drq0/int5/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/lock_n/clkdiv2_n/pio29
97	uzi_n/pio26
98	txd1/pio27
99	rxd1/pio28
100	cts0_n/enrx0_n/pio21

Table 6. IA186ES 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
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	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

Name	Pin
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
whb_n	42
wlb_n	43
wr_n	5
x1	13
x2	14

2.1.5 IA188ES PQFP Package

The pinout for the IA188ES PQFP package is as shown in Figure 5. The corresponding pinout is provided in Tables 7 and 8.

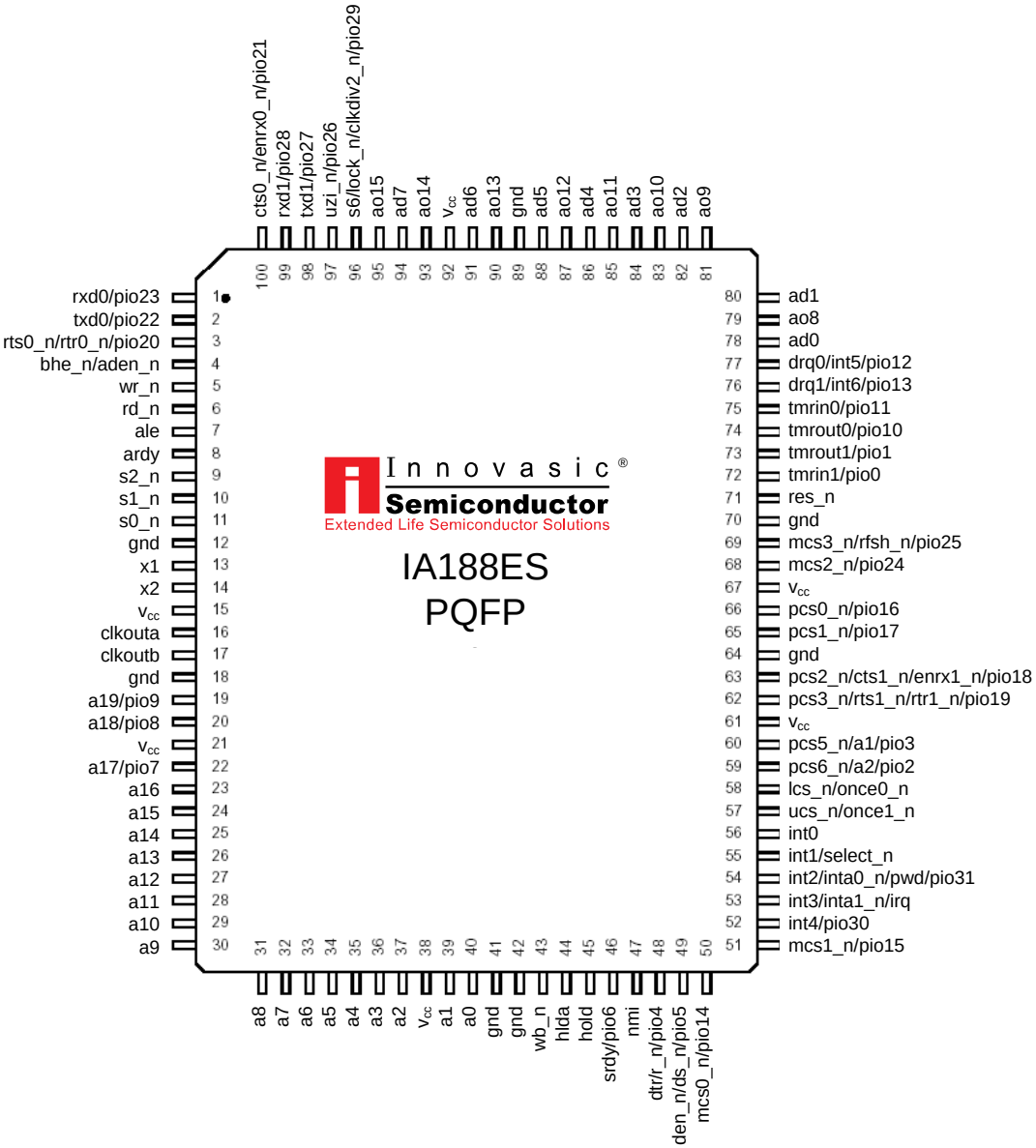


Figure 5. IA188ES PQFP Package Diagram

Table 7. IA188ES PQFP Numeric Pin Listing

Pin	Name
1	rxd0/pio23
2	txd0/pio22
3	rts0_n/rtr0_n/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/pio9
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	gnd
43	wb_n
44	hlda
45	hold
46	srdy/pio6
47	nmi
48	dt/r_n/pio4
49	den_n/ds_n/pio5
50	mcs0_n/pio14
51	mcs1_n/pio15
52	int4/pio30
53	int3/inta1_n/irq
54	int2/inta0_n/pwd/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/rts1_n/rtr1_n/pio19
63	pcs2_n/cts1_n/enrx1_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/pio0
73	tmrout1/pio1
74	tmrout0/pio10
75	tmrin0/pio11
76	drq1/int6/pio13
77	drq0/int5/pio12
78	ad0
79	ao8
80	ad1
81	ao9
82	ad2
83	ao10
84	ad3
85	ao11
86	ad4
87	ao12
88	ad5
89	gnd
90	ao13
91	ad6
92	v_{CC}
93	ao14
94	ad7
95	ao15
96	s6/lock_n/clkdiv2_n/pio29
97	uzi_n/pio26
98	txd1/pio27
99	rxd1/pio28
100	cts0_n/enrx0_n/pio21

Table 8. IA188ES PQFP Alphabetic Pin Listing

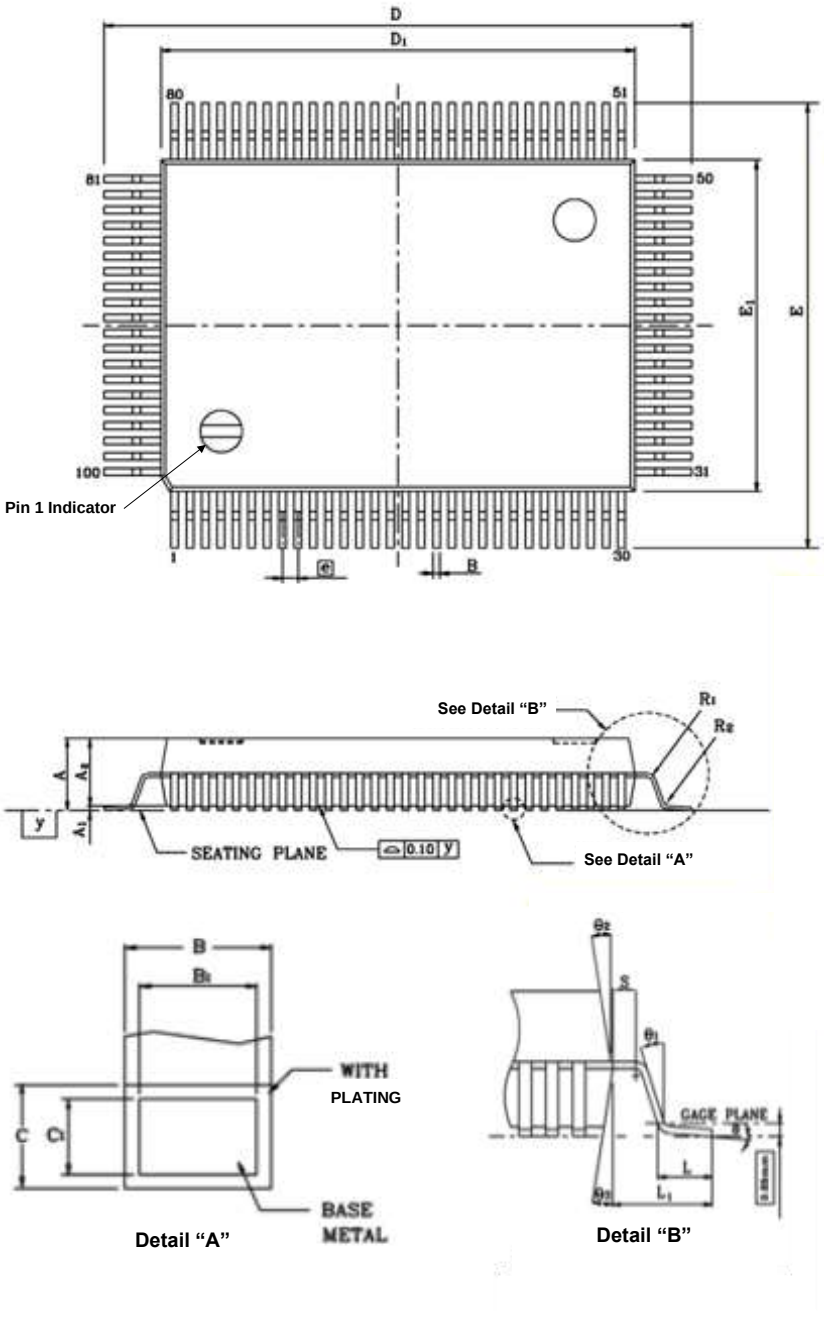
Name	Pin
a0	40
a1	39
a2	37
a3	36
a4	35
a5	34
a6	33
a7	32
a8	31
a9	30
a10	29
a11	28
a12	27
a13	26
a14	25
a15	24
a16	23
a17/pio7	22
a18/pio8	20
a19/pio9	19
ad0	78
ad1	80
ad2	82
ad3	84
ad4	86
ad5	88
ad6	91
ad7	94
ale	7
ao8	79
ao9	81
ao10	83
ao11	85
ao12	87

Name	Pin
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
Iint4/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

Name	Pin
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

2.1.6 PQFP Physical Dimensions

The physical dimensions for the PQFP are as shown in Figure 6.



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

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

Figure 6. PQFP Package Dimensions

2.2 Pin Descriptions

2.2.1 a19/pio9, a18/pio8, a17/pio7, a16–a0—Address Bus (synchronous outputs with tristate)

These pins are the system's source of non-multiplexed I/O or memory addresses and occur a half clkouta cycle before the multiplexed address/data bus (ad15–ad0 for the IA186ES or ao15–ao8 and ad7–ad0 for the IA188ES).

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

2.2.2 ad15–ad8 (IA186ES)—Address/Data Bus (level-sensitive synchronous inouts with tristate)

These pins are the system's source of time-multiplexed I/O or memory addresses and data. The address function of these pins may be disabled (see bhe_n/aden_n pin description). If the address function of these pins is enabled, the address will be present on this bus during t_1 of the bus cycle and data will be present during t_2, t_3, and t_4 of the same bus cycle.

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

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

These pins may be used to load the internal Reset Configuration register (RESCON, offset 0F6h) with configuration data during a POR.

2.2.3 ao15–ao8 (IA188ES)—Address Bus (level-sensitive synchronous outputs with tristate)

The address bus will contain valid high order address bits during the bus cycle (t_1, t_2, t_3, and t_4) if the bus is enabled by the AD bit in the Upper and Lower Memory Chip Select registers (UMCS, offset 0a0h, and LMCS, offset 0a2h).

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

2.2.4 ad7–ad0—Address/Data Bus (level-sensitive synchronous inouts with tristate)

These pins are the system's source of time-multiplexed low order byte of the addresses for I/O or memory and 8-bit data. The low order address byte will be present on this bus during t_1 of the bus cycle and the 8-bit data will be present during t_2, t_3, and t_4 of the same bus cycle.

The address function of these pins may be disabled (see bhe_n/aden_n pin description).

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

2.2.5 ale—Address Latch Enable (synchronous output)

This signal indicates the presence of an address on the address bus (ad15–ad0 for the IA186ES or ao15–ao8 and ad7–ad0 for the IA188ES), which is guaranteed to be valid on the falling edge of ale.

In ONCE mode, this pin is tristated but not during bus hold or reset.

2.2.6 ardy—Asynchronous Ready (level-sensitive asynchronous input)

This asynchronous signal provides an indication to the microcontroller that the addressed I/O device or memory space will complete a data transfer. This active high signal is asynchronous with respect to clkouta. If the falling edge of ardy is not synchronized to clkouta, an additional clock cycle may be added. The ardy or srdy must be synchronized to clkouta to guarantee the number of inserted wait states.

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

2.2.7 bhe_n/aden_n (IA186ES only)—Bus High Enable (synchronous output with tristate)/Address Enable (input with internal pullup)

The bhe_n–bhe_n and address bit ad0 or a0 inform the system which bytes of the data bus (upper, lower, or both) are involved in the current memory access bus cycle, as shown in Table 9.

Table 9. Bus Cycle Types for bhe_n and ad0

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

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

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

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

The enrx0_n is the Enable-Receiver-Request for asynchronous serial port 0 when Bit [4] (ENRX0) in the AUXCON register is 1, and Bit [9] (FC) in the SP0CT register is 1, and it enables the asynchronous serial port receiver.

2.2.11 den_n/ds_n/pio5—Data Enable /Data Strobe (both are synchronous outputs with tristate)

den_n 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. After reset, this pin defaults to den_n.

The data strobe ds_n is used under conditions in which a write cycle has the same timing as a read cycle. It is used with other control signals to interface with 68-Kbyte-type peripherals without further system interface logic. When it is asserted, addresses are valid. During a write, the data is valid, while during a read, data may be applied to the *ad* bus.

2.2.12 drq0/int5/pio12—DMA Request 0 (synchronous level-sensitive input)/Maskable Interrupt Request 5 (asynchronous edge-triggered input)

The drq0 is an external device that is ready for DMA channel 0 to carry out a transfer. It indicates to the microcontroller this readiness on this pin. It is not latched and must remain asserted until it is dealt with.

If DMA channel 0 is not required, int5 may be used as an extra interrupt request sharing the DMA0 interrupt type (0ah) and control bits. It is not latched and must remain asserted until it is dealt with.

2.2.13 drq1/int6/pio13—DMA Request 1 (synchronous level-sensitive input)/Maskable Interrupt Request 6 (asynchronous edge-triggered input)

The drq1 is an external device that is ready for DMA channel 1 to carry out a transfer. It indicates to the microcontroller this readiness on this pin. It is not latched and must remain asserted until it is dealt with.

If DMA channel 1 is not required, int6 may be used as an extra interrupt request sharing the DMA1 interrupt type (0bh) and control bits. It is not latched and must remain asserted until it is dealt with.

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

The microntroller transmits data when dt/r_n is pulled high and receives data when this pin is pulled low. It floats during a reset or bus hold condition.

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.

2.2.19 int2/inta0_n/pwd/pio31—Maskable Interrupt Request 2 (asynchronous input)/Interrupt Acknowledge 0 (synchronous output)/Pulse Width Demodulator (Schmitt trigger input)

The int2 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. 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 pwd processes a signal via the Schmitt trigger when pulse width demodulation is enabled. It drives timrin0 and int2 and its inverse signal drives timrin1 and int4. Provided that int2 and int4 are enabled and timer0 and timer1 are configured correctly, the pulse width of the alternating signal on pwd may be calculated from the values in timer0 and timer1.

While in pwd mode, tmrin0/pio11, tmrin1/pio0, and int4/pio31 signals are free for use as PIOs or may be ignored. The level on this pin is held in the PIO data register in the pio31 position, just as if it were a PIO.

2.2.20 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. Provided that int3 is not masked, program execution will continue at the location specified by the int3 vector in the interrupt vector table. Although interrupt requests are asynchronous, they are synchronized internally and may be edge- or level-triggered. 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.

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

2.2.21 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. In the case where PWD mode is selected, int4 indicates a High-to-Low transition of the PWD signal. Conversely, in the event that PWD mode is not selected, int4 may be used as a PIO.

2.2.22 lcs_n/once0_n—Lower Memory Chip Select (synchronous output with internal pullup)/ONCE Mode Request (input)

The lcs_n pin provides an indication that a memory access is in progress to the lower memory block. The size of the Lower Memory Block and its base address are programmable, with the size adjustable up to 512 Kbytes. The lcs_n may be configured for either an 8- or 16-bit bus width for the IA186ES microcontroller by the Auxiliary Configuration Register (AUXCON Bit [2]) and is held high during bus hold.

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

2.2.23 mcs0_n/pio14—Midrange Memory Chip Select (synchronous output with internal pullup)

The mcs0_n pin provides an indication that a memory access is in progress to the midrange memory block. The size of the Midrange Memory Block and its base address are programmable. The mcs0_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. The mcs0_n may be programmed as the chip select for the whole middle chip select address range. Furthermore, this pin has a weak pullup that is only present during reset.

2.2.24 mcs2_n–mcs1_n (pio24–pio 15)—Midrange Memory Chip Selects (synchronous outputs with internal pullup)

The mcs2_n and mcs1_n pins provide an indication that a memory access is in progress to the second or third midrange memory block. The size of the Midrange Memory Block and its base address are programmable. The mcs2_n and mcs1_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 are held high during bus hold. Furthermore, these pins have weak pullups that are

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

during both bus hold and reset. These outputs are asserted with the *ad* address bus over a 256 byte range each.

2.2.28 pcs2_n/cts1_n/enrx1_n/pio18—Peripheral Chip Select 2 (synchronous output)/Clear-to-Send 1 (asynchronous input)/Enable-Receiver-Request 1 (asynchronous input)

The pcs2_n signal provides an indication that a memory access is under way for the third region of the peripheral memory block (I/O or memory address space). The base address of the peripheral memory block is programmable. The pcs2_n is held high during both bus hold and reset. This output is asserted with the *ad* address bus over a 256-byte range.

The cts1_n is the Clear-to-Send signal for asynchronous serial port 1 when the ENRX1 bit (Bit [6]) in the auxiliary control register (AUXCON) is 0 and hardware flow control is enabled for this port (FC bit [Bit (9)] in the serial port 1 control register [SP1CT]). This signal controls the transmission of data from the serial port transmit register 1. When this signal is asserted, the transmitter begins sending out a frame of data if any is in the transmit register, whereas if the signal is deasserted, the data will be held in the transmit register. The signal is checked at the beginning of each frame of transmit data.

The enrx1_n is the Enable-Receiver-Request for asynchronous serial port 1 when the enrx1 bit (Bit [6]) in the auxiliary control register (AUXCON) is 1 and hardware flow control is enabled for this port (FC bit [Bit (9)] in the serial port 1 control register [SP1CT]). This signal enables the receiver of asynchronous serial port 1.

2.2.29 pcs3_n/rts1_n/rtr1_n/pio18—Peripheral Chip Select 3 (synchronous output)/Ready-to-Send 1 (asynchronous output)/Ready-to-Receive 1 (asynchronous input)

The pcs3_n signal provides an indication that a memory access is under way for the fourth region of the peripheral memory block (I/O or memory address space). The base address of the Peripheral memory block is programmable. The pcs3_n is held high during both bus hold and reset. This output is asserted with the *ad* address bus over a 256-byte range.

The rts1-n is the Ready-to-Send signal for asynchronous serial port 1 when the RTS1 bit (Bit [5]) in the auxiliary control register (AUXCON) is 1 and hardware flow control is enabled for this port (FC bit [Bit (9)] in the serial port 1 control register [SP1CT]). This signal is asserted when the serial port transmit register contains untransmitted data.

The rtr1-n is the Ready-to-Receive signal for asynchronous serial port 1 when the rts1 bit (Bit [5]) in the auxiliary control register (AUXCON) is 0 and hardware flow control is enabled for this port (FC bit [Bit (9)] in the serial port 1 control register [SP1CT]). This signal is asserted when the serial port receive register does not contain valid, unread data.

2.2.30 pcs5_n/A1/pio3—Peripheral Chip Select 5 (synchronous output)/Latched Address Bit [1] (synchronous output)

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

The A1 pin provides and internally latched address Bit [1] to the system when the EX bit (Bit [7]) in the mcs_n and pcs_n auxiliary (MPCS) register is 0. It retains its previously latched value during a bus hold.

2.2.31 pcs6_n/A2/pio2—Peripheral Chip Select 6 (synchronous output)/Latched Address Bit [2] (synchronous output)

The pcs6_n signal provides an indication that a memory access is underway for the seventh region of the peripheral memory block (I/O or memory address space). The base address of the peripheral memory block is programmable. The pcs6_n is held high during both bus hold and reset. This output is asserted with the *ad* address bus over a 256-byte range.

The A2 pin provides an internally latched address Bit [2] to the system when the EX bit (Bit [7]) in the MPCS register is 0. It retains its previously latched value during a bus hold.

2.2.32 pio31–pio0—Programmable I/O Pins (asynchronous input/output open-drain)

There are 32 individually PIO pins provided.

2.2.33 rd_n—Read strobe (synchronous output with tristate)

This pin provides an indication to the system that a memory or I/O read cycle is underway. It will not to be asserted before the *ad* bus is floated during the address to data transition. The rd_n is tristated during bus hold.

2.2.34 res_n—Reset (asynchronous level-sensitive input)

This pin forces a reset on the microcontroller. It has a Schmitt trigger to allow POR generation via an RC network. When this signal is asserted, the microcontroller immediately terminates its present activity, clears its internal logic, and transfers CPU control to the reset address, FFFF0h.

The res_n must be asserted for at least 1 ms and may be asserted asynchronously to clkouta as it is synchronized internally. Furthermore, v_{cc} must be within specification and clkouta must be stable for more than four of its clock periods for the period that res_n is asserted. The microcontroller starts to fetch instructions 6.5 clkouta clock periods after the deassertion of res_n.

2.2.35 rfsh2_n/aden_n (IA188ES only)—Refresh 2 (synchronous output with tristate)/Address Enable (input with internal pullup)

The rfsh2_n indicates that a DRAM refresh cycle is being performed when it is asserted low. However, this is not valid in PSRAM mode where mcs3_n/rfsh_n is used instead.

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

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

2.2.36 rts0_n/rtr0_n/pio20—Ready-to-Send 0 (asynchronous output)/Ready-to-Receive 0 (asynchronous input)

The rts0-n is the Ready-to-Send signal for asynchronous serial port 0 when the RTS0 bit (Bit [3]) in the auxiliary control register (AUXCON) is 1 and hardware flow control is enabled for this port (FC bit [Bit (9)] in the serial port 1 control register [SP1CT]). This signal is asserted when the serial port transmit register contains untransmitted data.

The rtr0-n is the Ready-to-Receive signal for asynchronous serial port 0 when the rts0 bit (Bit [3]) in the auxiliary control register (AUXCON) is 0 and hardware flow control is enabled for this port (FC bit [Bit (9)] in the serial port 1 control register [SP1CT]). This signal is asserted when the serial port receive register does not contain valid unread data.

2.2.37 rxd0_n/pio23—Receive Data 0 (asynchronous input)

This signal connects asynchronous serial receive data from the system to the asynchronous Serial Port 0.

2.2.38 rxd1_n/pio28—Receive Data 1 (asynchronous input)

This signal connects asynchronous serial receive data from the system to the asynchronous Serial Port 1.

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

These three signals inform the system of the type of bus cycle is in progress. 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)

2.2.40 s6/lock_n/clkdiv2_n/pio29—Bus Cycle Status Bit [6] (synchronous output)/Bus Lock (synchronous output)/Clock Divide by 2 (input with internal pullup)

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 under way. The s6 is tristated during bus hold or reset.

The lock_n signal is held low to indicate to other system bus masters that the system bus is being used and that no attempt should be made to try to gain control of it. This signal is only available during t_1 and is intended for emulator use.

The microcontroller enters clock divide-by-2 mode, if clkdiv2_n is held low during power-on-reset. In this mode, the PLL is disabled and the processor receives the external clock divided by 2. Sampling of this pin occurs on the rising edge of res_n.

Should this pin be used as pio29 configured as an input, care should be taken that it is not driven low during power-on-reset. This pin has an internal pullup so it is not necessary to drive the pin high even though it defaults to an input PIO.

2.2.41 srdy/pio6—Synchronous Ready (synchronous level-sensitive input)

This signal is an active high input synchronized to clkouta and indicates to the microcontroller that a data transfer will be completed by the addressed memory space or I/O device.

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

2.2.42 tmrin0/pio11—Timer Input 0 (synchronous edge-sensitive input)

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

When Pulse Width Demodulation mode is enabled, tmrin0 is driven internally by int2/inta0_n/pwd allowing for the pin to be configured as pio11.

2.2.43 tmrin1/pio0—Timer Input 1 (synchronous edge-sensitive input)

This signal may be either a clock or control signal for the internal timer 1. The timer is incremented by the microcontroller after it synchronizes a rising edge of tmrin1. When not used, tmrin1 must be tied high. When used as pio0, it is pulled up internally. When pulse width demodulation mode is enabled, tmrin1 is driven internally by int2/inta0_n/pwd, allowing for the pin to be configured as pio0.

2.2.44 tmrout0/pio10—Timer Output 0 (synchronous output)

This signal provides the system with a single pulse or a continuous waveform with a programmable duty cycle. It is tristated during a bus hold or reset.

2.2.45 tmrout1/pio1—Timer Output 1 (synchronous output)

This signal provides the system with a single pulse or a continuous waveform with a programmable duty cycle. It is tristated during a bus hold or reset.

2.2.46 txd0/pio22—Transmit Data 0 (asynchronous output)

This pin provides the system with asynchronous serial transmit data from serial port 0.

2.2.47 txd1/pio27—Transmit Data 1 (asynchronous output)

This pin provides the system with asynchronous serial transmit data from serial port 1.

2.2.48 ucs_n/once1_n—Upper Memory Chip Select (synchronous output)/ONCE Mode Request 1 (input with internal pullup)

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

After reset, ucs_n is active for the 64-Kbyte memory range from F0000h to FFFFFh, which includes the reset address at FFFF0h.

The once1_n pin (ON Circuit Emulation) and its companion pin, once0_n, define the microcontroller mode during reset. These two pins are sampled on the rising edge of res_n and if both are asserted low, the microcontroller starts in ONCE mode, otherwise it starts normally. In ONCE mode, all pins are tristated and remain so until a subsequent reset. To prevent the microcontroller from entering ONCE mode inadvertently, this pin has a weak pullup that is only present during reset. This pin is not tristated during bus hold.

2.2.49 uzi_n/pio26—Upper Zero Indicate (synchronous output)

This pin allows the designer to determine if an access to the interrupt vector table is in progress by ORing it with Bits [15–10] of the address and data bus (ad15–ad10 on the AI186 and ao15–ao10 on the IA188ES). The uzi_n is the logical AND of the inverted a19–a16 bits. It asserts in the first period of a bus cycle and is held throughout the cycle.

2.2.50 v_{cc}—Power Supply (input)

These pins supply power (+5V ±10%) to the microcontroller.

2.2.51 whb_n (IA186ES only)—Write High Byte (synchronous output with tristate)

This pin and wlb_n provide an indication to the system of which bytes of the data bus (upper, lower, or both) are taking part in a write cycle. The whb_n is asserted with ad15–ad8 and is the logical OR of bhe_n and wr_n. It is tristated during reset.

2.2.52 wlb_n/wb_n—Write Low Byte (IA186ES only) (synchronous output with tristate)/Write Byte (IA188ES only) (synchronous output with tristate)

The wlb_n and whb_n provide an indication to the system of which bytes of the data bus (upper, lower, or both) are taking part in a write cycle. The wlb_n is asserted with ad7–ad0 and is the logical OR of ad0 and wr_n. It is tristated during reset.

On the IA188ES microcontroller, wb_n provides an indication that a write to the bus is occurring. It shares the same early timing as that of the non-multiplexed address bus, and is associated with ad7–ad0. It is tristated during reset.

2.2.53 wr_n—Write Strobe (synchronous output)

This pin provides an indication to the system that the data currently on the bus is to be written to a memory or I/O device. It is tristated during a bus hold or reset.

2.2.54 x1—Crystal Input

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

2.2.55 x2—Crystal Input

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

2.3 Pins Used by Emulators

The following pins are used by emulators:

- a19–a0
- ao15–ao8
- ad7–ad0
- ale
- bhe_n
- aden_n (on the IA186ES)
- clkouta
- rfsh2_n/aden_n (on the IA188ES)
- rd_n
- s2_n–s0_n
- s6/lock_n/clkdiv2_n
- uzi_n

Emulators require that s6/lock_n/clkdiv2_n and uzi_n be configured as their normal functions (i.e., as s6 and uzi_n, respectively). Holding bhe_n/aden_n (IA186ES) or rfsh_n/aden_n (IA188ES) low during the rising edge of res_n, s6, and uzi_n will be configured at reset in their normal functions instead of as PIOs.

3. Maximum Ratings, Thermal Characteristics, and DC Parameters

For the Innovasic Semiconductor IA186ES and IA188ES microcontrollers, the absolute maximum ratings, thermal characteristics, and DC parameters are provided in Tables 11 through 13, respectively.

Table 11. IA186ES and IA188ES Absolute Maximum Ratings

Parameter	Rating
Storage Temperature	−65°C to +125°C
Voltage on any Pin with Respect to v_{ss}	−0.5V to +(v_{cc} + 0.5)V

Table 12. IA186ES and IA188ES Thermal Characteristics

Symbol	Characteristic	Value
T_A	Ambient Temperature	-40°C to 85°C

Table 13. DC Characteristics Over Commercial Operating Ranges

Symbol	Parameter Description	Test Conditions	Preliminary		Unit
			Min	Max	
V_{IL}	Input Low Voltage (Except x1)	–	−0.5	0.8	V
V_{IL1}	Clock Input Low Voltage (x1)	–	−0.5	0.8	V
V_{IH}	Input High Voltage (Except res_n and x1)	–	2.0	v_{cc} +0.5	V
V_{IH1}	Input High Voltage (res_n)	–	2.4	v_{cc} +0.5	V
V_{IH2}	Clock Input High Voltage (x1)	–	v_{cc}–0.8	v_{cc} +0.5	V
V_{OL}	Output Low Voltages	I_{OL} = 2.5 mA (s2_n–s0_n)	–	0.45	V
		I_{OL} = 2.0 mA (other)	–	0.45	V
V_{OH}	Output High Voltages[a]	I_{OH} = −2.4 mA @ 2.4 V	2.4	v_{cc} +0.5	V
		I_{OH} = −200 µA @ v_{cc} −0.5	v_{cc} −0.5	v_{cc}	V
I_{CC}	Power Supply Current @ 0°C	v_{cc} = 5.5 V[b]	–	5.9	mA/ MHz
I_{LI}	Input Leakage Current @ 0.5 MHz	0.45 V ≤ V_{IN} ≤ v_{cc}	–	±10	µA
I_{LO}	Output Leakage Current @ 0.5 MHz	0.45 V ≤ V_{OUT} ≤ v_{cc} [c]	–	±10	µA
V_{CLO}	Clock Output Low	I_{CLO} = 4.0 mA	–	0.45	V
V_{CHO}	Clock Output High	I_{CHO} = −500 µA	v_{cc} −0.5	–	V

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

[b]Current is measured with the device in reset with the x1 and x2 driven and all other non-power pins open but held high or low.

[c]Testing is performed with the pins floating, either during hold or by invoking the ONCE mode.

4. Device Architecture

A functional block diagram of the IA186ES/IA188ES is shown in Figure 7. This microcontroller consists of the following functional blocks.

- Bus Interface and Control (BIC)
- Chip Selects and Control (CSC)
- Programmable I/O
- Clock and Power Management
- DMA
- Interrupt Controller
- Timers
- Asynchronous Serial Ports (2).

4.1 Bus Interface and Control

The BIC manages all accesses to external memory and external peripherals. These peripherals may be mapped either in memory space or I/O space. The BIC supports both multiplexed and non-multiplexed bus operations. Multiplexed address and data are provided on the AD [15–0] bus, while a non-multiplexed address is provided on the A [19–0] bus. The A bus provides address information for the entire bus cycle (t_1–t_4), while the AD bus provides address information only during the first (t_1) phase of the bus cycle. For details regarding bus cycles, see Chapter 6, AC Specifications.

The BIC provide the capability to dynamically alter the size of the data bus. By programming the auxiliary control register (AUXCON), a user may easily support external peripherals and memory devices of both 8- and 16-bit widths without specialized micro-code managing the data accesses. The AUXCON register contains 3 programmable bits for this purpose: LSIZ, MSIZ, and IOSIZ. For details regarding the operation of these bits, see Section 5.1, Control and Registers. The IA186ES microcontroller provides two signals to support this functionality, write high byte (whb_n) and write low byte (wlb_n). The IA188ES microcontroller requires only a single write byte (wb_n) signal to support its 8-bit data bus.

The BIC also provides support for PSRAM devices. PSRAM is supported in the lower chip select (lcs_n) area only. In order to support PSRAM, the CSC must be appropriately programmed. For details regarding this operation, see Section 4.7, Chips Selects.

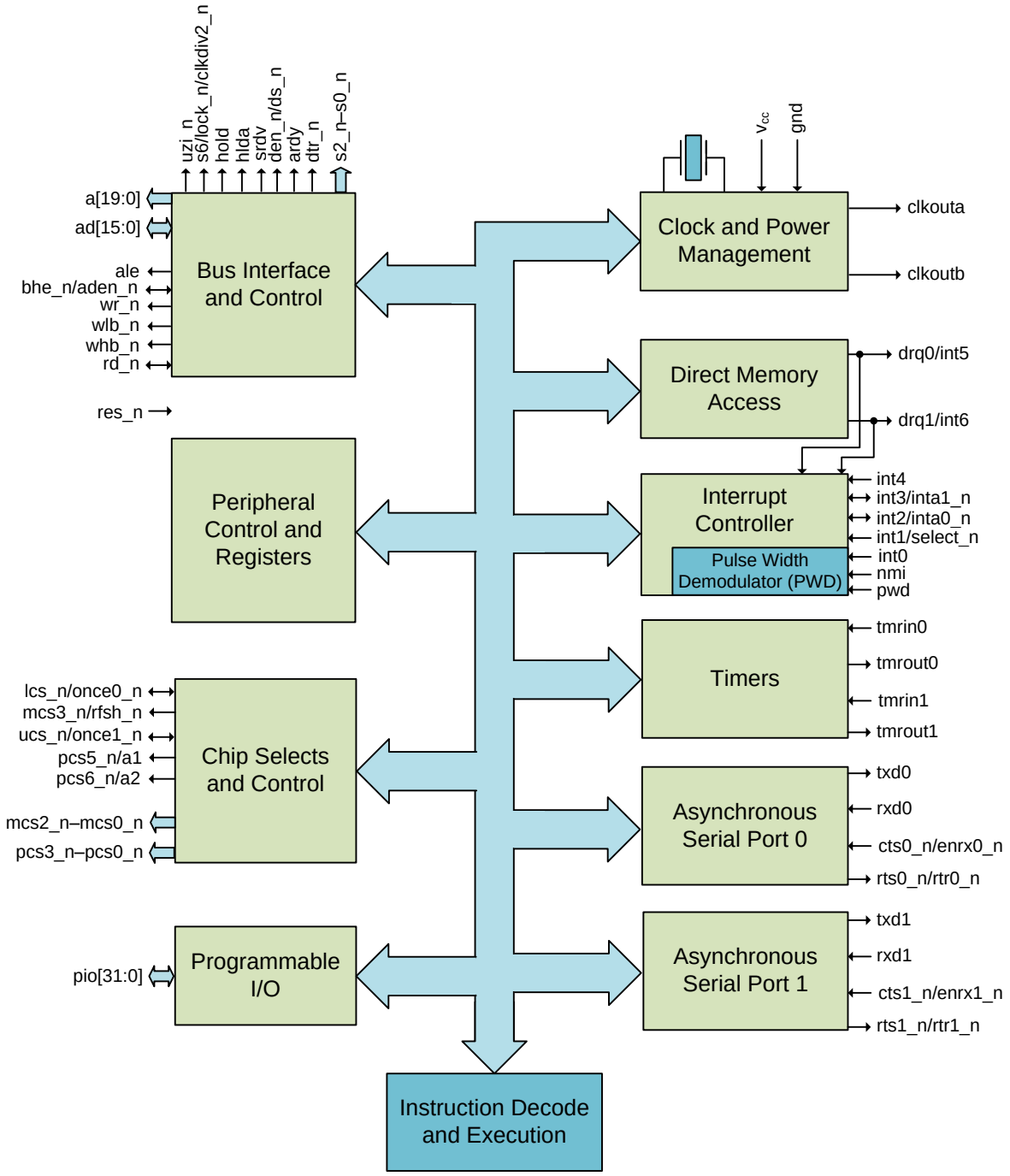


Figure 7. Functional Block Diagram

Note: See pin descriptions for pins that share other functions with PIO pins. Pins pwd, int5, int6, rts1_n/rtr1_n, and cts1_n/enrx1_n are multiplexed with int2_n/inta0_n, drq0, drq0, pcs3_n, and pcs2_n, respectively.

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).

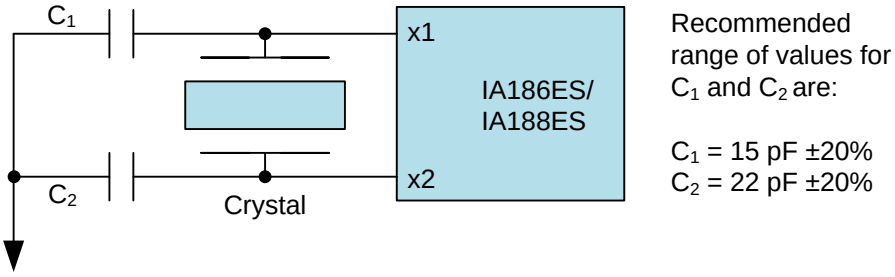


Figure 8. Crystal Configuration

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).

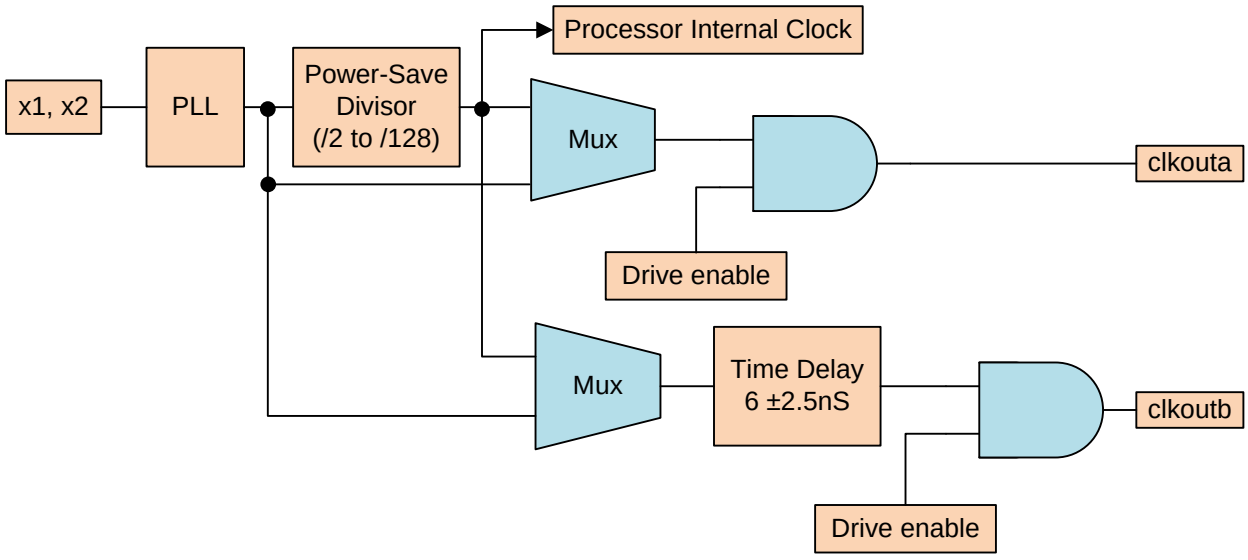


Figure 9. Organization of Clock

4.4 Power-Save Mode

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

4.5 Initialization and Reset

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

4.6 Reset Configuration Register

The data on the address/data bus (ad15–ad0 for the IA186ES and ao15–ao8 and ad7–ad0 for the IA188ES) are written into the Reset Configuration Register when reset is low. This data is system-dependent and is held in the Reset Configuration Register after reset is de-asserted. This configuration data may be placed on the address/data bus by using weak external pull-up and pull-down resistors or applied to the bus by an external driver, as the processor does not drive the bus during reset. It is a method of supplying the software with some initial data after a reset (e.g., option jumper positions).

4.7 Chip Selects

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

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

4.8 Chip-Select Timing

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

4.9 Ready- and Wait-State Programming

Each of the memory or peripheral chip-select lines can have a ready signal programmed that can be the ardy or srdy signal. The chip-select control registers (UMCS, LMCS, MMCS, PACS, and MPCS) have a single bit that selects if the external ready signal is to be used or not (R2, Bit [2]). R1 and R0 (Bits [1–0]) in these registers control the number of wait states that are inserted during each access to a memory or peripheral location (from 0 to 3). The control registers for pcs3_n–pcs0_n use three bits, R3, R1–R0 (Bit [3] and Bits [1–0]) to provide 5, 7, 9, and 15 wait states in addition to the original values of 0–3 wait states.

When an external ready has been selected as required, internally programmed wait states will always be completed before the external ready can finish or extend a bus cycle. Consider a system in which the number of wait states to be inserted has been set to three. The external ready pin is sampled by the processor during the first wait cycle. If the ready is asserted, the access is completed after seven cycles (4 cycles plus 3 wait cycles). If the ready is not asserted, during the first wait cycle the access is prolonged until ready is asserted and two more wait states are inserted followed by t_4. The ardy signal is an asynchronous ready with a pin that is active high and accepts a rising edge asynchronous to clkouta. However, an additional clock period may be necessary if the falling edge of ardy is not synchronized to clkouta.

4.10 Chip-Select Overlap

Overlapping chip selects are those configurations in which more than one chip select is asserted for the same physical address. If PCS is configured in I/O space with LCS or any other chip select configured for memory, address 00000h is not overlapping the chip selects.

It is not recommended that multiple chip-select signals be asserted for the same physical address, although it may be unavoidable in certain systems. If this is the case, 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). These signals serve as chip selects that are configured with no wait states and no external ready. Only when these chip selects are configured in the same manner can the PCB be programmed with addresses that overlap external chip selects.

Care should be exercised 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 chip selects. Setting the DA bit to “1” prevents the address from being driven onto the AD bus for all accesses for which the respective chip select is active, including those for which the multiple selects are active.

The mcs_n and pcs_n pins are dual-purpose pins, either as chip selects or PIO inputs or outputs. However, their respective ready- and wait-state configurations for their chip-select function will be in effect regardless of the function for which they are actually programmed. Their ready- and

wait-state settings must agree with those for any overlapping chip selects as though they had been configured as chip selects. This is true regardless of whether 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).

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

If the chip-select-configuration rules are not followed, the processor may hang with the appearance of waiting for a ready signal—even in a system 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, so upper memory is usually used for instruction. To this end, ucs_n is active on reset and has a memory range of 64 Kbytes (F0000h to FFFFFh) as default, along with external ready required and three wait states automatically inserted. The lower boundary of ucs_n is programmable to provide ranges of 64 to 512 Kbytes.

4.12 Low-Memory Chip Select

The lcs_n chip select is for lower memory and may be configured for 8- or 16-bit accesses by the AUXCON register. Because the interrupt vector table is at the bottom of memory beginning at 00000h, this pin is usually used for control data memory. Unlike ucs_n, this pin is inactive on reset.

4.13 Midrange-Memory Chip Selects

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

The mcs0_n chip select may be programmed to be active over the entire MCS range, leaving the mcs3_n–mcs1_n free for use as PIO pins.

The MCS may be configured for 8- or 16-bit accesses by the AUXCON register. The width of the non-UCS/non-LCS memory ranges determines the MCS range bus width. The assertion of the MCS outputs occurs with the same timing as the multiplexed AD address bus.

4.14 Peripheral Chip Selects

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

None of the pcs_n pins are active at reset. The pcs6_n–pcs5_n and pcs3_n–pcs0_n may be programmed to have 0 to 3 wait states. The pcs3_n–pcs0_n may be also programmed to have 5, 7, 9, and 15 wait states.

The PCS may be configured for 8- or 16-bit accesses by the AUXCON register. The PCS range bus width is determined either by that of the non-UCS/non-LCS memory range or by the width of the I/O space. The assertion of the PCS outputs occurs with the same timing as the multiplexed AD address bus. Each of the PCS operates over a 256-byte address range.

4.15 Refresh Control

The Refresh Control Unit (RCU) automatically generates refresh bus cycles with a fixed wait-state value of three for the PSRAM automatic refresh mode. The RCU generates a memory-read request after a programmable period of time to the bus interface unit.

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

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

4.16 Interrupt Control

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

Eight external-interrupt sources are connected to the processor. These include seven maskable and one nonmaskable interrupt (NMI). Eight internal-interrupt sources are also connected to the processor. These include those not brought out to external pins—three timers, two DMA channels, two asynchronous serial ports, and the WDT NMI. Interrupts int6 and int5 (multiplexed with drq1 and drq0) are available if the respective DMA is not enabled or is internally synchronized.

With the exception of int0, the seven external maskable interrupt request pins are multifunctional. One function is for direct-interrupt requests. The int6 and int5 are edge-triggered. The int4–int0 may be either level- or edge-triggered.

When configured in cascade mode, int1 and int0 interface with an external 82C59A-type interrupt controller. When int0 is configured for cascade mode, the function of int2 is automatically switched to its inta0_n role. Similarly, when int1 is configured for cascade mode, int3 is switched to its inta1_n role. An external 82C59A-compatible interrupt controller may be used as the system master by programming the internal interrupt controller to slave mode, but int6–int4 cannot be used.

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

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

4.17 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[b]	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,b]	07h	1ch	NA	1	ESC Opcodes
Timer0 Interrupt[d,e]	08h	20h	08h	2A	–
Timer1 Interrupt[d,e]	12h	48h	08h	2B	–

Table 14. Interrupt Types *(Continued)*

Interrupt Name	Interrupt Type	Vector Table Address	EOI Type	Overall Priority	Related Instructions
Timer2 Interrupt[d,e]	13h	4ch	08h	2C	–
Reserved	09h	24h	–	–	–
DMA0 Interrupt/INT5[e]	0ah	28h	0ah	3	–
DMA1 Interrupt/INT6[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	–
Asynchronous Serial Port 1 Interface[f]	11h	44h	11h	9	–
Asynchronous Serial Port 0 Interface[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.

4.18 Timer Control

The IA186ES and IA188ES have a WDT and three 16-bit programmable timers. Timer0 and timer1 each has an input and output connected to external pins that permits it to count or time events as well as produce variable duty-cycle waveforms or non-repetitive waveforms. These same timers are used to measure the high- and low-pulse widths of the Pulse Width Demodulator on the pwd pin.

Because timer2 does not have external connections, it is confined to internal functions such as real-time coding, time-delay applications, a prescaler for timer0 and timer1, or to synchronize DMA transfers.

The Peripheral Control Block contains eleven 16-bit registers to control the programmable timers. Each timer-count register holds the present value of its associated timer and may be read from or written to whether the timer is in operation or not. The microcontroller increments the value of the timer-count register when a timer event takes place.

The value stored in a timer's associated maximum count register determines its maximum count value. Upon reaching it, the timer count register is reset to 0 in the same clock cycle that this count was attained. The timer count register does not store this maximum value. Both timer0 and timer1 have a primary and a secondary maximum count register that permits each to alternate between two discrete maximum values.

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

Because they are polled every fourth clock period, the timers can operate at a quarter of the internal clock frequency. Although an external clock may be used, the timer output may take six clock cycles to respond to the input.

4.19 Watchdog Timer

The WDT operates in real WDT fashion and may be used to prevent loss of control in the event that software does not respond in an expected manner. The WDT is active after reset, has a maximum timeout count, and is programmed for system reset mode. The WDT control register (WDTCON) may be written to only once after reset. This is accomplished by writing 3333h, then CCCCh followed by the new configuration data to the WDTCON register. Provided they do not include access to the WDTCON register, any number of operations may be performed between these two words, including memory and I/O reads and writes.

Writing AAAAh then 5555h to the WDTCON register resets the current count. This count cannot be read. Provided they do not include access to the WDTCON register, any number of operations may be performed between these two words, including memory and I/O reads and writes. Use of these sequences is intended to prevent executing code from blocking a WDT event. With the WDT, a maximum 1.67-second timeout period is possible in a 40-MHz system.

The WDT can be programmed to generate either an NMI or a system reset when it times out. If programmed to generate an NMI, the NMIFLAG (Bit [12]) in the WDTCON register will be set when it occurs. This flag should be tested by the NMI interrupt service routine (ISR) to establish whether the WDT or an external source generated the interrupt. If set by writing the 3333h and CCCCh sequence followed by the configuration data that includes clearing NMIFLAG, the ISR should clear this flag. If the NMIFLAG is set while a second WDT timeout occurs, a WDT system reset is generated in place of a second NMI interrupt.

The RSTFLAG (Bit [13]) in the WDTCON register is set if a WDT reset is generated, due to one WDT occurrence while the WDT is programmed to generate resets, or because a WDT event

occurred with the NMIFLAG set. This permits system initialization code to distinguish between a WDT reset and hardware reset and take appropriate action. The RSTFLAG is cleared by a read or write to the WDTCON register. During a WDT reset, the external pins are not re-sampled, ensuring that clocking, reset configuration register, and any other features that are user programmable during reset do not change when a WDT system reset occurs. All other activities are the same as those of a normal system reset.

4.20 Direct Memory Access

DMA frees the CPU from involvement in transferring data between memory and peripherals over either one or both high-speed DMA channels. Data may be transferred from memory to I/O, I/O to memory, memory to memory, or I/O to I/O. DMA channels may be connected to asynchronous serial ports.

The IA186ES microcontroller supports the transfer of both bytes and words to and from even or odd addresses. It does not support word transfers to memory that is configured for byte accesses. The IA188ES does not support word transfers at all. Each data transfer will take two bus cycles (a minimum of 8 clock cycles).

There are four sources of DMA requests for both DMA channels:

- The channel request pin (drq1–drq0)
- Timer2
- A serial port
- The system software.

Each channel may be programmed to have a different priority either to resolve a simultaneous DMA request or to interrupt a transfer on the other channel.

4.21 DMA Operation

The PCB contains six registers for each DMA channel to control and specify the operation of the channel (see Figure 10):

- Two registers to store a 20-bit source address
- Two registers to store a 20-bit destination address
- One 16-bit transfer-count register
- One 16-bit control register

The number of DMA transfers required is designated in the DMA Transfer Count register and may contain up to 64 Kbytes or words. It will end automatically. DMA channel function is defined by the Control registers. Like the other five registers, these may be changed at any time (including during a DMA transfer) and are implemented immediately.

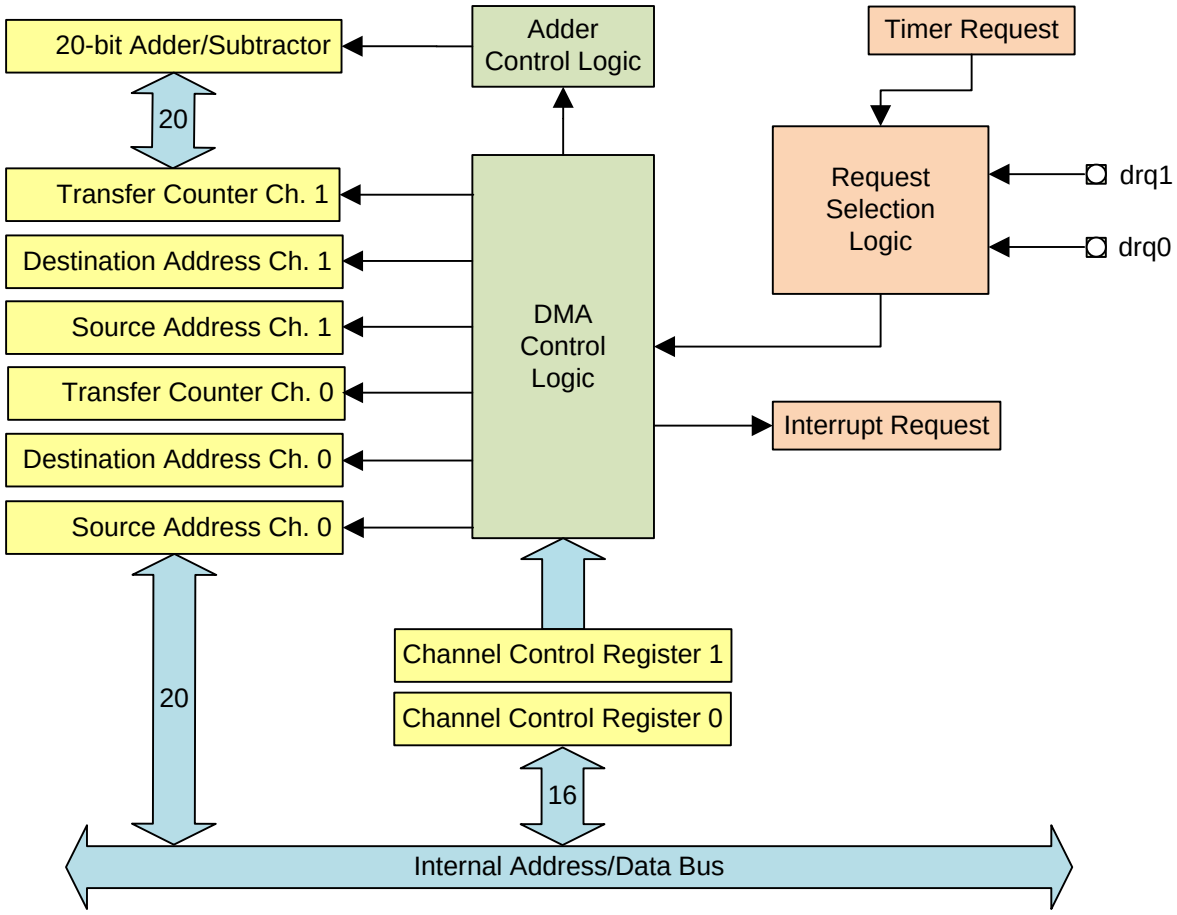


Figure 10. DMA Unit

4.22 DMA Channel Control Registers

See Section 5.1.10, D1CON (0dah) and D0CON (0cah). The DMA channel control registers specify the following:

- Whether the data destination is in memory or I/O space (Bit [15])
- Whether the destination address is incremented, decremented, or unchanged after each transfer (Bits [14–13])
- Whether the data source is in memory or I/O space (Bit [12])
- Whether the source address is incremented, decremented, or unchanged after each transfer (Bits [11–10])

- Whether DMA transfers cease upon reaching a designated count (Bit [9])
- Whether the last transfer generates an interrupt (Bit [8])
- Synchronization mode (Bits [7–6])
- The relative priority of one DMA channel with respect to the other (Bit [5])
- Acceptance of DMA requests from Timer2 (Bit [4])
- Configuration of DRQ pins as INT (Bit [3])
- Byte or Word transfers (Bit [0])

4.23 DMA Priority

With the exception of word accesses to odd memory locations or between locked memory addresses, DMA transfers have a higher priority than CPU transfers. Because the CPU cannot access memory during a DMA transfer and a DMA transfer cannot be suspended by an interrupt request, continuous DMA activity will increase interrupt delay. An NMI request halts any DMA activity, however, enabling the CPU to respond promptly to the request.

4.24 Pulse Width Demodulation

Note: There is no support for analog-to-digital conversion. This feature provides a means of measuring the width of a pulse in both its high and low phases. Its enabled by the PWD bit (Bit [6]) in the SYSCON.

TMRIN0, TMRIN1, INT2, and INT4 are internally configured to support the detection of rising and falling edges on the PWD pin (int2/int0_n/pwd) and to enable either timer0 or timer1, depending on whether the signal is high or low (see Figure 11). Because they are not used in this mode, the tmrin0, tmrin1, and int4 pins are available as PIO pins.

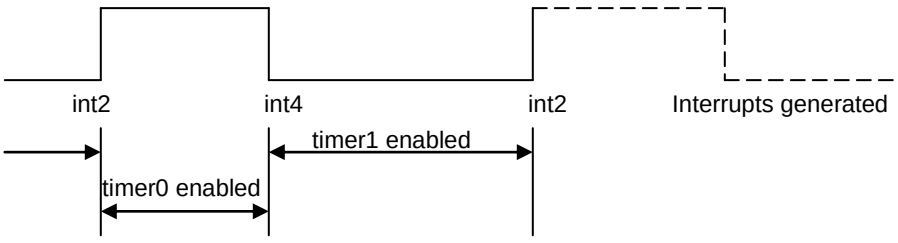


Figure 11. Typical Waveform at the int2/int0_n/pwd pin

The current count of timer1 for INT2 and timer0 for INT4 should be inspected by the ISR to determine the pulse width. The timer count register should then be reset by the ISR in readiness for the next pulse.

The timer count rate (one-fourth of the processor clock rate) determines the maximum resolution of the timers. To avoid the delay in servicing a timer interrupt in cases where the pulse width is short, the INT2 and INT4 request bits in the Interrupt Request register may be polled. In cases where the pulse width is greater than the maximum count of the timer, detection is achieved either by monitoring the Maximum Count (MC) bit of the respective timer or by enabling the timer interrupt requests by setting the INT bit in the respective Timer Mode and Control Register.

4.25 Asynchronous Serial Ports

There are two independent asynchronous serial ports that employ standard industry communication protocols in their implementation of full duplex, bi-directional data transfers. Functioning independently, either port may be the source or destination of DMA transfers.

The following features are supported:

- Full-duplex data transfers
- 7-, 8-, or 9-bit data transfers
- Odd, even, or no parity
- One or two stop bits
- Break characters of two lengths
- Error detection provided by parity, framing, or overrun errors
- Hardware handshaking 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 ports
- Each port has its own maskable interrupts
- 9-bit multidrop protocol
- Each port has an independent baud-rate generator
- Maximum baud rate is 1/16 of the processor clock
- Transmit and receive lines are double buffered

4.26 Programmable I/O

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.

Table 15. Default Status of PIO Pins at Reset

PIO No.	Associated Pin	Power-On Reset Status
0	tmrin1	Input with pullup
1	tmrout1	Input with pulldown
2	pcs6_n/a2	Input with pullup
3	pcs5_n/a1	Normal operation[a]
4	dt/r_n	Normal operation[a]
5	den_n/ds_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 pulldown
11	tmrin0	Input with pullup
12	drq0/int5	Input with pullup
13	drq1/int6	Input with pullup
14	mcs0_n	Input with pullup
15	mcs1_n	Input with pullup
16	pcs0_n	Input with pullup
17	pcs1_n	Input with pullup
18	pcs2_n/cts1_n/enrx1_n	Input with pullup
19	pcs3_n/rts1_n/rtr1_n	Input with pullup
20	rts0_n/rtr0_n	Input with pullup
21	cts0_n/enrx0_n	Input with pullup
22	txd0	Input with pullup
23	rxd0	Input with pullup
24	mcs2_n	Input with pullup
25	mcs3_n/rfsh_n	Input with pullup
26[b,c]	uzi_n	Input with pullup
27	txd1	Input with pullup
28	rxd1	Input with pullup
29[b,c]	s6/lock_n/clkdiv2	Input with pullup
30	int4	Input with pullup
31	int2/inta0_n/pwd	Input with pullup

Associated Pin	A	Power-On Reset Status
a17	7	Normal operation[a]
a18	8	Normal operation[a]
a19	9	Normal operation[a]
cts0_n/enrx0_n	21	Input with pullup
den_n/ds_n	5	Normal operation[a]
drq0/int5	12	Input with pullup
drq1/int6	13	Input with pullup
dt/r_n	4	Normal operation[a]
int2/inta0_n/pwd	31	Input with pullup
int4	30	Input with pullup
mcs0_n	14	Input with pullup
mcs1_n	16	Input with pullup
mcs2_n	24	Input with pullup
mcs3_n/rfsh_n	25	Input with pullup
pcs0_n	16	Input with pullup
pcs1_n	17	Input with pullup
pcs2_n/cts1_n/enrx1_n	18	Input with pullup
pcs3_n/rts1_n/rtr1_n	19	Input with pullup
pcs5_n/a1	3	Input with pullup
pcs6_n/a2	2	Input with pullup
rts0_n/rtr0_n	20	Input with pullup
rxd0	23	Input with pullup
rxd1	28	Input with pullup
s6/lock_n/clkdiv2	29	Input with pullup
srdy	6	Normal operation[d]
timrin0	11	Input with pullup
tmrin1	0	Input with pullup
tmrout0	10	Input with pulldown
tmrout1	1	Input with pulldown
txd0	22	Input with pullup
txd1	27	Input with pullup
uzi_n	26	Input with pullup

[a]Input with pullup when used as PIO.
[b]Emulators use these pins and also s2_n–s0_n, res_n, nmi, clkouta, bhe_n, ale, ad15–ad0, and a16–a0.
[c]If bhe_n/aden_n (IA186ES) or rfsh_n/aden_n (IA188ES) is held low during POR, these pins will revert to normal operation.
[d]Input with pulldown option available when used as PIO.

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 IA186ES or the rfsh2_n/aden_n for the IA188ES is held low during POR.

5. Peripheral Architecture

5.1 Control and Registers

The on-chip peripherals in the IA186ES/IA188ES 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 IA188ES 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 IA186ES or IA188ES, indeterminate behavior may result.

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

Table 16. Peripheral Control Registers

Register Name	Offset
Peripheral Control Block Registers	
PCB Relocation Register	FEh
Reset Configuration Register	F6h
Processor Release Level Register	F4h
Auxiliary Configuration Register	F2h
System Configuration Register	F0h
Watchdog Timer Control Register	E6h
Enable RCU Register	E4h
Clock Prescaler Register	E2h
Memory Partition Register	E0h
DMA Registers	
DMA1 Control Register	DAh
DMA1 Transfer Count Register	D8h
DMA1 Destination Address High Register	D6h
DMA1 Destination Address Low Register	D4h
DMA1 Source Address High Register	D2h
DMA1 Source Address Low Register	D1h
DMA0 Control Register	CAh
DMA0 Transfer Count Register	C8h
DMA0 Destination Address High Register	C6h
DMA0 Destination Address Low Register	C4h
DMA0 Source Address High Register	C2h
DMA0 Source Address Low Register	C0h
Chip-Select Registers	
pcs_n and mcs_n Auxiliary Register	A8h
Mid-Range Memory Chip-Select Register	A6h
Peripheral Chip-Select Register	A4h
Low-Memory Chip-Select Register	A2h
Upper-Memory Chip-Select Register	A0h
Serial Port 0 Registers	
Serial Port 0 Baud Rate Divisor Register	88h
Serial Port 0 Receive Register	86h
Serial Port 0 Transmit Register	84h
Serial Port 0 Status Register	82h
Serial Port 0 Control Register	80h
PIO Registers	
PIO Data 1 Register	7Ah
PIO Direction Register	78h
PIO Mode 1 Register	76h
PIO Data 0 Register	74h
PIO Direction 0 Register	72h
PIO Mode 0 Register	70h

Register Name	Offset
Timer Registers	
Timer2 Mode and Control Register	66h
Timer2 Max Count Compare A Register	62h
Timer2 Count Register	60h
Timer1 Mode and Control Register	5Eh
Timer1 Max Count Compare B Register	5Ch
Timer1 Max Count Compare A Register	5Ah
Timer1 Count Register	58h
Timer0 Mode and Control Register	56h
Timer0 Max Count Compare B Register	54h
Timer0 Max Count Compare A Register	52h
Timer0 Count Register	50h
Interrupt Registers	
Serial Port 0 Interrupt Control Register	44h
Serial Port 1 Interrupt Control Register	42h
INT4 Interrupt Control Register	40h
INT3 Interrupt Control Register	3Eh
INT2 Interrupt Control Register	3Ch
INT1 Interrupt Control Register	3Ah
INT0 Interrupt Control Register	38h
DMA1/INT6 Interrupt Control Register	36h
DMA0/INT5 Interrupt Control Register	34h
Timer Interrupt Control Register	32h
Interrupt Status Register	30h
Interrupt Request Register	2Eh
Interrupt In-Service Register	2Ch
Interrupt Priority Mask Register	2Ah
Interrupt Mask Register	28h
Interrupt Poll Status Register	26h
Interrupt Poll Register	24h
End-of-Interrupt (EOI) Register	22h
Interrupt Vector Register	20h
Serial Port 1 Registers	
Serial Port 1 Baud Rate Divisor Register	18h
Serial Port 1 Receive Register	16h
Serial Port 1 Transmit Register	14h
Serial Port 1 Status Register	12h
Serial Port 1 Control Register	10h

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	RA [19–8]											

- 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]—RA [19–8] → Sets the base address (upper 12 bits) of the Peripheral Control Block Register Bank. RA [7–0] default to zero. When Bit [12] (IO/Mn) is set to 1, RA [19–16] 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
RC [15–0]															

- Bits [15–0]—RC [15–0] → At the rising edge of reset, the values of specified pins (ad15–ad0 for the IA186ES and ao15–ao8 and ad7–ad0 for the IA188ES) 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 1100h (see Table 19).

Table 19. Processor Release Level Register

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

- Bits [15–8]—PRL [7–0] → The latest Processor Release Level.

PRL Value	Processor Release Level
10h	A
11h	B
12h	C
13h	D
14h	E

- Bits [7–0]—Reserved.

5.1.4 AUXCON (0f2h)

The AUXiliary CONfiguration Register configures the flow control signals for the asynchronous serial ports. AUXCON controls data bus width (8- or 16-bit) for lower memory, middle memory, and IO accesses and contains 0000h at reset (see Table 20).

Table 20. Auxiliary Configuration Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES									ENRX1	RTS1	ENRX0	RTS0	LSIZ	MSIZ	IOSIZ

- Bit [15–7]—Reserved.
- Bit [6]—ENRX1 → When set to 1, the cts1_n/enrx1_n pin functions as cts1_n. When 0, it functions as enrx1_n.
- Bit [5]—RTS1 → When set to 1, the rtr1_n/rts1_n pin functions as rts1_n. When 0, it functions as rtr1_n.
- Bit [4]—ENRX0 → When set to 1, the cts0_n/enrx0_n pin functions as cts0_n. When 0, it functions as enrx0_n.
- Bit [3]—RTS0 → When set to 1, the rtr0_n/rts0_n pin functions as rts0_n. When 0, it functions as rtr0_n.
- Bit [2]—LSIZ (IA186ES only) → When set to 1, 8-bit data accesses are performed in lower chip-select (lcs_n) space. When 0, 16-bit data accesses are performed.

- 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.

- Bit [8]—CAD → When set to 1, the clkouta output is driven low. When 0, it is driven as an output per the CBF bit.
- Bits [7–3]—Reserved → The bits read back as zeros.
- Bits [2–0]—F2–F0 → These bits control the clock divider as shown below.

Note: PSEN must be 1 for the clock divider to function.

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

5.1.6 WDTCON (0e6h)

The WatchDog Timer CONtrol Register provides control and status for the WDT. The WDTCON contains c080h at reset (see Table 22).

Table 22. Watchdog Timer Control Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENA	WRST	RSTFLAG	NMIFLAG	TEST	RES			COUNT							

- Bit [15]—ENA → When set to 1, the WDT is enabled. When 0, it is disabled.
- Bit [14]—WRST → When set to 1, an internal WDT reset is generated when the WDT timeout count (COUNT) is reached. When 0, an NMI will be generated once WDT timeout count is reached and the NMIFLAG bit is 0. If the NMIFLAG bit is 1, an internal WDT reset is generated when the WDT timeout count is reached.
- Bit [13]—RSTFLAG → When set to 1, a WDT timeout event has occurred. This bit may be cleared by software or by an external reset.
- Bit [12]—NMIFLAG → When set to 1, a WDT NMI event has occurred. This bit may be cleared by software or by an external reset. If this bit is 1 when WDT timeout occurs, an internal WDT reset is generated regardless of the state of WRST.
- Bit [11]—TEST → This bit is reserved for chip test and should be always set to 0.

- Bits [10–8]—Reserved.
- Bits [7–0]—COUNT → Control the timeout period for the WDT as follows:

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

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.

5.1.8 CDRAM (0e2h)

The Count for Dynamic RAM (CDRAM) Refresh Control Register determines the period between refresh cycles. The CDRAM register is undefined at reset (see Table 24).

Table 24. Count for Dynamic RAM Refresh Control Register

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

- Bits [15–9]—Reserved → These bits read back as 0.
- Bits [8–0]—RC [8–0] → These bits hold the clock count interval between refresh cycles. In power-save mode, the refresh counter value should be adjusted to account for the clock divider value in SYSCON.

5.1.9 MDRAM (0e0h)

The Memory Partition for Dynamic RAM (MDRAM) Refresh Control Register holds the a19–a13 address bits of the 20-bit base refresh address. The MDRAM register contains 0000h at reset (see Table 25).

Table 25. Memory Partition for Dynamic RAM Refresh Control Register

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

- Bits [15–9]—M [6–0] → Upper bits corresponding to address bits a19–a13 of the 20-bit memory refresh address. These bits are not available on the a19–a0 bus. When using PSRAM mode, M6–M0 must be programmed to 0000000b.
- Bits [8–0]—Reserved → These bits read back as 0.

5.1.10 D1CON (0dah) and D0CON (0cah)

DMA CONtrol Registers. DMA Control Registers control operation of the two DMA channels. The D0CON and D1CON registers are undefined at reset, except ST which is set to 0 (see Table 26).

Table 26. DMA Control Registers

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

- 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.

Synchronization Bit Channel Selection

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

- 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]—EXT → External Interrupt Enable Bit. When set to 1, if the respective DMA channel does not respond to changes on the drq pin, this pin functions as an int pin and the interrupt controller processes requests on the pin. When 0, it functions as a drq pin.
- 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 IA188ES does not support word transfers

5.1.11 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 27).

Table 27. 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]—TC [15–0] → DMA Transfer Count contains the transfer count for the respective DMA channel. Its value is decremented after each transfer.

5.1.12 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 initialized. Each address may be incremented or decremented independently of each other after each transfer. The addresses are incremented or decremented by two for word transfers and incremented or decremented by one for byte transfers. They are undefined at reset (see Table 28).

Table 28. DMA Destination Address High Register

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

- Bits [15–4]—Reserved.
- Bits [3–0]—DDA [19–16] → DMA Destination Address High bits are driven onto a19–a16 during the write phase of a DMA transfer.

5.1.13 DIDSTL (0d4h) and D0DSTL (0c4h)

DMA DeSTination Address Low Register. The 16 bits of these registers are combined with the 4 bits of the respective DMA Destination Address High Register to produce a 20-bit destination address. They are undefined at reset (see Table 29).

Table 29. DMA Destination Address Low Register

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

- Bits [15–0]—DDA [15–0] → DMA Destination Address Low bits are driven onto a19–a16 during the write phase of a DMA transfer.

5.1.14 D1SRCH (0d2h) and D0SRCH (0c2h)

DMA SouRCe Address High Register. The 20-bit source address consists of these 4 bits combined with the 16 bits of the respective Source Address Low Register. A DMA transfer requires that two complete 16-bit registers in the PCB (high and low registers) be used for both the source and destination addresses of each DMA channel involved. Each channel requires that

all four address registers be initialized. Each address may be independently incremented or decremented after each word transfer by 2 or by 1 for byte transfers. They are undefined at reset (see Table 30).

Table 30. DMA Source Address High Register

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

- Bits [15–4]—Reserved.
- Bits [3–0]—DSA [19–16] → DMA Source Address High bits are driven onto a19–a16 during the read phase of a DMA transfer.

5.1.15 D1SRCL (0d0h) and D0SRCL (0c0h)

DMA SouRCe Address Low Register. The 16 bits of these registers are combined with the 4 bits of the respective DMA Source Address High register to produce a 20-bit source address. They are undefined at reset (see Table 31).

Table 31. DMA Source Address Low Register

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

- Bits [15–0]—DSA [15–0] → DMA Source Address Low bits are placed onto a15–a0 during the read phase of a DMA transfer.

5.1.16 MPCS (0a8h)

MCS and PCS (MPCS) Auxiliary Register. Because this register controls more than one type of chip select, it is unlike other chip select control registers. The MPCS register contains information for mcs3_n–mcs0_n, pcs6_n–pcs5_n, and pcs3_n–pcs0_n.

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

The pcs6_n–pcs5_n pins are high and not active on processor reset. When the pcs6_n–pcs5_n are configured as address pins, an access to the MPCS register causes them to activate. They do not require corresponding access to the PACS register to be activated. The value of the MPCS register is undefined at reset (see Table 32).

Table 32. MCS and PCS Auxiliary Register

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

- Bit [15]—Reserved → Set to 1.
- Bits [14–8]—M [6–0] mcs_n Block Size → These seven bits determine the total memory block size for the mcs3_n–mcs0_n chip selects. The size is divided equally among them. The relationship between M [6–0] and the size is shown below.

Select Sizes of M6–M0 by Total Block Size

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

- Bit [7]—EX Pin Selector → This bit determines whether the pcs6_n–pcs5_n pins are configured as chip selects or as alternate outputs for a2 and a1. When set to 1, they are configured as peripheral chip select pins. When 0, they become address bits a1 and a2, respectively.
- Bit [6]—MS → Memory/ I/O Space Selector determines whether the pcs_n pins are active during either memory or I/O bus cycles. When set to 1, the outputs are active for memory bus cycles. When 0, they are active for I/O bus cycles.
- Bits [5–3]—Reserved → Set to 1.
- Bit [2]—R2 Ready Mode → This bit influences only the pcs6_n–pcs5_n chip selects. When set to 1, external ready is ignored. When 0, it is required. Values determine the number of wait states to be inserted.
- Bits [1–0]—R [1–0] Wait-State Value → These bits influence only the pcs6_n–pcs5_n chip selects. Their value determines the number of wait states inserted into an access, depending on whether it is to the pcs_n memory or I/O area. Up to three wait states can be inserted (R1–R0 = 00b to 11b).

5.1.17 MMCS (0a6h)

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

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

Unlike the ucs_n and lcs_n chip selects, the mcs3_n–mcs0_n outputs assert with the multiplexed *ad* address bus (ad15–ad0 or ao15–ao8 and ad7–ad0), rather than the earlier timing of the a19–a0 bus. If the a19–a0 bus is used for address selection, the timing is delayed for a half cycle later than that for ucs_n and lcs_n. The value is undefined at reset (see Table 33).

Table 33. Midrange Memory Chip Select Register

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

- Bits [15–9]—BA [15–9] Base Address → The value of the BA19–BA13 determines the base address of the memory block that is addressed by the mcs_n chip select pins. These bits correspond to a19–a13 of the 20-bit memory address. The remaining bits a12–a0 of the base address are always 0.
  - The base address may be any integer multiple of the size of the memory clock selected in the MPCS register. For example, if the midrange block is 32 Kbytes, the block could be located at 20000h or 28000h but not at 24000h.
  - If the lcs_n chip select is inactive, the base address of the midrange chip selects can be set to 00000h, because the lcs_n chip select is defined to be 00000h but is unused. Because the base address must be an integer multiple of the block size, a 512K MMCS block size can only be used with the lcs_n chip select inactive and the base address of the midrange chip selects set to 00000h.
- Bits [8–3]—Reserved. Set to 1.
- Bit [2]—R2 Ready mode → This bit determines the mcs_n chip selects ready mode. When set to 1, an external ready is ignored. When 0, an external ready is necessary. Its value determines the number of wait states inserted into an access.

- Bits [1–0]—R [1–0] → Wait-State Value. The value of these bits determines the number of wait states inserted in an access. Up to three wait states can be inserted (R1–R0 = 00b to 11b).

5.1.18 PACS (0a4h)

PeripherAl Chip Select Register. These Peripheral Chip Selects are asserted over 256-byte range with the same timing as the *ad* address bus. There are six chip selects, pcs6_n–pcs5_n and pcs3_n–pcs0_n that are used in either the user-locatable memory or I/O blocks. Excluding the areas used by the ucs_n, lcs_n, and mcs_n chip selects, the memory block can be located anywhere within the 1-Mbyte address space. These chip selects may also be configured to access the 64-Kbyte I/O space.

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

The MPCS register configures the pcs6_n–pcs5_n pins to be either chip selects or address pins a1 and a2. When these pins are configured as chip selects, the MPCS register determines the ready state and wait states for these output pins and whether they are active during memory or I/O bus cycles. These pins are activated as chip selects by writing to the two registers (PACS and MPCS). They are not active on reset. To configure and activate them as address pins, it is necessary to write to both the PACS and MPCS registers. Pins pcs6_n–pcs5_n can be configured for 0 to 3 wait states and pcs3_n–pcs0_n can be programmed for 0 to 15 wait states. The value of the PACS register is undefined at reset (see Table 34).

Table 34. Peripheral Chip Select Register

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

- Bits [15–7]—BA [19–11] → Base Address bits correspond to Bits [19–11] of the 20-bit programmable base address of the peripheral chip select block and determine the base address. Because I/O addresses are only 16 bits wide, if the pcs_n chip selects are mapped to I/O space, these bits must be set to 0000b. The pcs address ranges are shown below.

Address Ranges of pcs Chip Selects

pcs_n Line	Range	
	Low	High
pcs0_n	Base Address	Base Address + 255
pcs1_n	Base Address + 256	Base Address + 511
pcs2_n	Base Address + 512	Base Address + 767
pcs3_n	Base Address + 768	Base Address + 1023
Reserved	NA	NA
pcs5_n	Base Address + 1280	Base Address
pcs6_n	Base Address + 1536	Base Address

- Bits [6–4]—Reserved. Set to 1.
- Bit [3]—R [3] → Wait State Value. See pcs3_n–pcs0_n Wait-State Encoding shown below.

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]—R [2] → 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]—R [1–0] → Wait-State Value (see pcs3_n–pcs0_n Wait-State Encoding shown above). The pcs6_n–pcs5_n and pcs3_n–pcs0_n pins are multiplexed with the PIO pins. For them to function as chip selects, the PIO mode and direction settings for these pins must be set to 0 for normal operation.

5.1.19 LMCS (0a2h)

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

- Bits [R1–R0]—R [1–0] → 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).

5.1.20 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 36).

Table 36. Upper-Memory Chip Select Register

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

- Bit [15]—Reserved. Set to 1.

- Bits [14–12]—LB [2–0] → Lower Boundary. These bits determine the bottom of the memory accessed by the ucs_n chip selects. The UMCS Block-Size Programming Values shown below list the possible block-size configurations (a 512-Kbyte maximum).

  UMCS Block-Size Programming Values

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

- Bits [11–8]—Reserved. Set to 0.

- Bit [7]—DA Disable Address → When set to 1, the address bus is disabled and the address is not driven on the address bus when ucs_n is asserted, providing some measure of power saving. When 0, the address is driven onto the address bus (ad15–ad0) during the address phase of a bus cycle when ucs_n is asserted. 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]—Reserved. Set to 0.

- Bit [5–3]—Reserved. Set to 1.
- Bit [2]—R2 Ready Mode → When set to 1, the external ready is ignored. When 0, an external ready is required. The value of these bits determines the number of wait states inserted.
- Bits [1–0]—R [1–0] 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).

5.1.21 SP0BAUD (088h)

Serial Port BAUD Rate Divisor Registers.

5.1.22 SP1BAUD (018h)

Two baud-rate divisor registers, one for each port, allow the two ports to operate at different baud rates. The value in these registers determines the number of internal processor cycles in one phase (one half period) of the 16x serial clock.

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

The baud rate divisor may be calculated from:

$$\text{BAUDDIV} = (\text{Processor Frequency}/(16 \times \text{baud rate})) \qquad \text{(Equation 2)}$$

By setting the BAUDDIV to 0001h, the maximum baud rate of 1/16 of the internal processor frequency clock is set. This provides a baud rate of 2500 Kbytes at 40 MHz. If the BAUDDIV is set to zero, transmission or reception of data does not occur. The baud rate tolerance is +3.0% and −2.5% with respect to the actual serial port baud rate, not the target baud rate (see Table 37).

Table 37. Baud Rates

Baud Rate	Divisor Based on CPU Clock Rate			
	20 MHz	25 MHz	33 MHz	40 MHz
300	4166	5208	6875	8333
600	2083	2604	3437	4166
1050	1190	1488	1964	2380
1200	1041	1302	1718	2083
1800	694	868	1145	1388
2400	520	651	859	1041
4800	260	325	429	520
7200	173	217	286	347
9600	130	162	214	260
19200	65	81	107	130
28800	43	54	71	86
38400	33	40	53	65
56000	22	28	36	45
57600	22	27	35	43
76800	16	20	26	32
115200	10	13	18	22
128000	9	12	16	19
153600	8	10	13	16
Special	15 MHz	21 MHz	24 MHz	30 MHz
187500	5	7	8	10

The value of the SP0BAUD and SP1BAUD registers at reset is 0000h (see Table 38).

Table 38. Serial Port Baud Rate Divisor Registers

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

- Bits [15–0]—BAUDDIV Baud Rate Divisor → Defines the divisor for the internal processor clock.

5.1.23 SP0RD (086h) and SP1RD (016h)

Serial Port Receive Registers. Data received over the serial ports are stored in these registers until read. The data are received initially by the receive shift register (no software access) permitting data to be received while the previous data are being read.

The status of these registers is indicated by the RDR bit (Receive Data Ready) in the serial port status registers. Setting the RDR bit to 1 indicates that there is valid data in the receive register. The RDR bit is cleared automatically when the receive register is read.

If handshaking is employed, the control signals cts_n/enrx_n are deasserted while the receive register has valid unread data. The cts_n/enrx_n signal is reasserted after the data in the receive register is read. The value of the SP0RD and SP1RD registers is undefined at reset (see Table 39).

Table 39. Serial Port Receive Registers

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

- Bits [15–8]—Reserved.
- Bits [7–0]—RDATA → Holds valid data while the RDR bit of the respective status register is set.

5.1.24 SP0TD (084h) and SP1TD (014h)

Serial Port Transmit Registers. Data is written to these registers by software with the values to be transmitted by the serial port. Double buffering of these transmitters allows for the transmission of data from the transmit shift registers (no software access), while the next data are written into the transmit registers.

The TEMT and THRE bits in the respective Serial Port Status registers indicate the status of these two pairs of registers.

Invoking handshaking requires that rts_n/rtr_n inputs be asserted before the transmitters can send any data which remain held in the transmit and shift registers without affecting the transmit pin. The value of the SPTD registers is undefined at reset (see Table 40).

Table 40. Serial Port Transmit Registers

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

- Bits [15–8]—Reserved.
- Bits [7–0]—TDATA → Holds the data to be transmitted.

5.1.25 SP0STS (082h) and SP1STS (012h)

Serial Port Status Register. These registers store information concerning the current status of the respective ports. The value of the SP0STS and SP1STS registers is undefined at reset (see Table 41).

Table 41. Serial Port Status Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					BRK1	BRK0	RB8	RDR	THRE	FER	OER	PER	TEMT	HS0	Res

- Bits [15–11]—Reserved.
- Bit [10]—BRK1 Long Break Detected → A long break is a low signal level on the rxd pin for a period greater that 2*M* + 3 bit times, where:

  M = start bit + number of data bits + parity bits + stop bit (Equation 3)

  – Should data reception be in progress when the break starts, the reception of the current word will be completed and the timing for the break will begin. Because the stop bit will not be detected due to the break, this will generate a framing error.

  – Detection of the break with the 2M + 3 bit time period can only be guaranteed if the break commences outside of a frame.

  Note: This bit should be reset by software.

- Bit [9]—BRK0 Short Break Detected → A short break is a low on the rxd pin for a period greater than M bit times (see Equation 3 above).

  – Should data reception be in progress when the break starts, the reception of the current word will be completed and the timing for the break will begin. Because the stop bit will not be detected due to the break, this will generate a framing error.

  – Detection of the break with the M bit time period can only be guaranteed if the break commences outside of a frame.

  Note: This bit should be reset by software.

- Bit [8]—RB8 Received Bit [8] → This is the ninth data bit received in modes 2 and 3 (see Section 5.1.26, SP0CT (080h) and SP1CT (010h)).

  Note: This bit should be reset by software.

- Bit [7]—RDR Receive Data Ready → When this bit is 1, it indicates that the respective SPRD register contains valid data. This is a read-only bit and can be reset only by reading the corresponding receive register.

- Bit [6]—THRE Transmit Holding Register Empty → When this bit is 1, it indicates that the corresponding transmit holding register is ready to accept data. This is a read-only bit.

- Bit [5]—FER Framing Error Detected → When the receiver samples the rxd line as low when a stop bit is expected (line high), a framing error is generated setting this bit.

Note: This bit should be reset by software.

- Bit [4]—OER Overrun Error Detected → When new data overwrites valid data in the receive register (because it has not been read), an overrun error is detected setting this bit.

Note: This bit should be reset by software.

- Bit [3]—PER Parity Error Detected → When a parity error is detected in either mode 1 or 3, this bit is set.

Note: This bit should be reset by software.

- Bit [2]—TEMT Transmitter Empty → When both the transmit shift register and the transmit register are empty, this bit is set indicating to software that it is safe to disable the transmitter. This bit is read-only.

- Bit [1]—HS0 Handshake Signal 0 → This bit is the inverted value of cts_n and is read only.

- Bit [0]—RES Reserved.

5.1.26 SP0CT (080h) and SP1CT (010h)

Serial Port ConTrol Registers. These registers control both transmit and receive parts of the respective serial ports. The value of the SP0CT and SP1CT registers is 0000h at reset (see Table 42).

Table 42. Serial Port Control Registers

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMA			RSIE	BRK	TB8	FC	TXIE	RXIE	TMODE	RMODE	EVN	PE	MODE		

- Bits [15–13]—DMA DMA Control Field → These bits set up the respective ports for use with DMA transfers as shown below.

DMA Control Bits

DMA Bits	Receive	Transmit
000b	No DMA	No DMA
001b	DMA0	DMA1
010b	DMA1	DMA0
011b	Reserved	Reserved
100b	DMA0	No DMA
101b	DMA1	No DMA
110b	No DMA	DMA0
111b	No DMA	DMA1

– DMA transfers to both serial ports are destination-synchronized operations. When the transmit holding register is empty, a new transfer is requested, corresponding with the assertion of the THRE bit in the status register in non-DMA mode. However, when configured for DMA transfers, the respective transmit interrupt is disabled without regard for the TXIE bit.

– DMA transfers from both serial ports are source-synchronized operations. When the receive holding register contains valid data, a new transfer is requested, corresponding with the assertion of the RDR bit in the status register in non-DMA mode. However, when configured for DMA receives, the respective receive interrupt is disabled without regard for the RXIE bit. This is despite the fact that the RSIE bit may still permit receive status interrupts, depending on its setting.

– DMA transfers do not preclude the use of hardware handshaking.

– If either or both serial ports are configured for DMA transfers, the DMA request is internally generated and the corresponding external DMA signals, drq0 and/or drq1 do not play a role.

- Bit [12]—RSIE Receive Status Interrupt Enable → When an exception occurs during data reception, an interrupt request is generated if enabled by this bit (RSIE = 1). Interrupt requests are made for the error conditions listed (BRK0, BRK1, OER, PER, and FER) in the serial port status register.

- Bit [11]—BRK Send Break → When this bit is set to 1, the txd pin is driven low overriding any data that may be in the course of being shifted out of the transmit shift register.

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

- Bit [10]—TB8 Transmit Bit 8 → This is the ninth data bit transmitted when in modes 2 and 3. This bit is cleared at each transmitted word and is not buffered. To transmit data with this bit set high, the following procedure is recommended.

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

- Bit [9]—FC Flow Control Enable → This bit controls the hardware handshake (flow control) by enabling it when set to 1, and vice versa. The type of flow control depends on the value of the ENRX0/ENRX1 and RTS0/RTS1 bits in the AUXCON register.

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

  – This bit is 0 at reset.

- Bit [8]—TXIE Transmitter Ready Interrupt Enable → This bit enables the generation of an interrupt request whenever the transmit holding register is empty (THRE Bit [1]). The respective port does not generate interrupts when this bit is 0. Interrupts continue to be generated as long as THRE and the TXIE are 1.

- Bit [7]—RXIE Receive Data Ready Interrupt Enable → This bit enables the generation of an interrupt request whenever the receive register contains valid data (RDR Bit [1]). The respective port does not generate interrupts when this bit is 0. Interrupts continue to be generated as long as RDR and the RXIE are 1.

- Bit [6]—TMODE Transmit Mode → The transmit section of the serial port is enabled when this bit is 1. Conversely, the transmit section of the serial port is disabled when this bit is 0.

- Bit [5]—RMODE Receive Mode → The receive section of the serial port is enabled when this bit is 1. Conversely, the receive section of the serial port is disabled when this bit is 0.

- Bit [4]—EVN Even Parity → When this bit is 1, even parity protocol is established. Conversely, odd parity is established when this bit is 0. This bit is valid only when parity is enabled (PE).

- Bit [3]—PE Parity Enable → Parity is enabled when this bit is 1 and disabled when this bit is 0.

- Bit [2–0]—MODE Mode of Operation → These three bits establish the mode of operation of the respective serial port. The valid modes and their functions are shown below.

Serial Port MODE Settings

MODE	Description	Data Bits	Parity Bits	Stop Bits
0[a]	Data Mode 0	7	–	2
1	Data Mode 1	7 or 8	1 or 0	1
2	Data Mode 2	9	–	1
3	Data Mode 3	8 or 9	1 or 0	1
4	Data Mode 4	7	–	1
5[a]	Data Mode 5	7 or 8	1 or 0	2
6[a]	Data Mode 6	9	0	2
7[a]	Data Mode 7	8 or 9	1 or 0	2

[a]These were originally reserved modes that have been implemented to provide 2 stop bits.

– Mode 2 requires that the ninth data is set to a 1 state. Otherwise, the character will be ignored by the receiver. The transmit section, however, operates as if it were in Mode 3.

– This is designed to facilitate multidrop communication over a common serial data link. For this purpose, the port in question is initially programmed to mode 2 and for each data received with the ninth bit (Bit [8]) set as 1. It is compared by software with a unique identifier for this port. If the identifier comparison does not find a match, the port is left in mode 2. If the comparison finds an identifier match, the port should be reprogrammed to mode 3 so that the ninth bit is allowed to be 0.

– Handshaking should only be employed in such a multidrop system by ports that are exchanging data (mode 3) to prevent multiple ports from attempting to drive the handshake signals. Mode 2 does not support handshaking for this reason and should not be enabled. If it is possible that more than 2 ports be configured in mode 3 at the same time, handshaking should not be implemented.

– Mode 3 allows for 8 data bits if parity is enabled or 9 data bits if parity is not enabled. If parity is not used, the ninth data bit for the transmit section is set by writing a 1 to the TB8 bit in the serial port control register. The ninth bit is read at the receive port from the RB8 bit in the serial port status register.

– Mode 4 allows for a start bit, 7 data bits, and a stop bit without parity, which is not available.

5.1.27 PDATA1 (07ah) and PDATA0 (074h)

PIO DATA Registers. When a PIO pin is configured as an output, the value in the corresponding PIO data register bit is driven onto the pin. However, if the PIO pin is configured

as an input, the value on the pin is put into the corresponding bit of the PIO data register. Table 43 lists the default states for the PIO pins.

Table 43. PIO Pin Assignments

PIO Number	Associated Pin Name	Power-On Reset Status
0	tmrin1	Input with pull-up
1	tmrout1	Input with pull-down
2	pcs6/A2	Input with pull-up
3	pcs5/A1	Input with pull-up
4	dt/r_n	Normal operation[a]
5	den_n/ds_n	Normal operation[a]
6	srdy	Normal operation[b]
7[c]	a17	Normal operation[a]
8[c]	a18	Normal operation[a]
9[c]	a19	Normal operation[a]
10	tmrout0	Normal operation[a]
11	tmrin0	Input with pull-up
12	drq0/int5	Input with pull-up
13	drq1/int6	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/cts1_n/enrx1_n	Input with pull-up
19	pcs3_n/rts1_n/rtr1_n	Input with pull-up
20	rts0_n/rtr0_n	Input with pull-up
21	cts0_n/enrx0_n	Input with pull-up
22	txd0	Input with pull-up
23	rxd0	Input with pull-up
24	mcs2_n	Input with pull-up
25	mcs3_n/rfsh_n	Input with pull-up
26[c,d]	uzi	Input with pull-up
27	txd1	Input with pull-up
28	rxd1	Input with pull-up
29[c,d]	s6/lock_n/clkdiv2_n	Input with pull-up
30	int4	Input with pull-up
31	int2/inta0_n/pwd	Input with pull-up

[a]When used as a PIO pin, it is an input with an available pull-up option.
[b]When used as a PIO pin, it is an input with an available pull-down option.
[c]Emulators use these pins. (The s2_n–s0_n, res_n, nmi, clkouta, bhe_n, ale, ad15–ad0, and a16–a0 pins are used by emulators also.)
[d]If bhe_n/aden_n is held low during POR, these pins revert to normal operation.

The 32 PIO pins initialize to either 00b or 01b as shown in Tables 44 and 45. The value of the PDATA registers is undefined at reset.

Table 44. PDATA 0

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

Table 45. PDATA 1

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

- Bits [15–0]—PDATA [15–0] PIO Data 0 Bits → This register contains the values of the bits that are either driven on, or received from, the corresponding PIO pins. Depending on its configuration each pin is either an output or an input. The values of these bits correspond to those in the PIO direction registers and PIO Mode registers.
- Bits [15–0]—PDATA [31–16] PIO Data 1 Bits → This register contains the values of the bits that are either driven on, or received from, the corresponding PIO pins. Depending on its configuration each pin is either an output or an input. The values of these bits correspond to those in the PIO direction registers and PIO Mode registers
- The PIO pins may be operated as open-drain outputs by:
  - Maintaining the data constant in the appropriate bit of the PIO data register.
  - Writing the value of the data bit into the respective bit position of the PIO Direction register, so that the output is either 0 or disabled depending on the value of the data bit.

5.1.28 PDIR1 (078h) and PDIR0 (072h)

PIO DIRection Registers. Each PIO pin is configured as an input or an output by the corresponding bit in the PIO direction register (see Table 46).

Table 46. PIO Mode and PIO Direction Settings

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

The value of the PDIR0 register is FC0Fh at reset (see Table 48).

Table 47. PDIR0

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

The value of the PDIR1 register is FFFFh at reset (see Table 48).

Table 48. PDIR1

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

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

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

5.1.29 PMODE1 (076h) and PMODE0 (070h)

PIO MODE Registers. Each PIO pin is configured as an input or an output by the corresponding bit in the PIO direction register. The bit number of PMODE corresponds to the PIO number (see Table 46, PIO Mode and PIO Direction Settings). The value of the PMODE0 register is 0000h at reset (see Table 49).

Table 49. PMODE0

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

The value of the PMODE1 register is 0000h at reset (see Table 50).

Table 50. PMODE1

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

- Bits [15–0]—PMODE [15–0] 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]—PMODE [31–16] 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.

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

Timer0 and Timer1 Mode and CONtrol Registers. These registers control the operation of Timer0 and Timer1, respectively. The value of the T0CON and T1CON registers is 0000h at reset (see Table 51).

Table 51. Timer0 and Timer1 Mode and Control Registers

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

- 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 INHn 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.

- 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. This bit always reads as 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 then the enable bit is cleared before said 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 set to 0 when the maxcount compare A register is used.

- 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. This bit may be used by software polling to monitor timer status rather than through interrupts, if desired.

- Bit [4]—RTG Retrigger Bit.

- 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 = 1, an external clock is used. If EXT = 0, an internal is used.

- Bit [1]—ALT Alternate Compare Bit → If set to 1, the timer will count to maxcount compare register A, reset the count register to 0, and then count to maxcount compare register B, reset the count register to 0, and begin again at maxcount compare register A. If set to 0, the timer will count to maxcount compare register A, reset the count register to 0, and begin again at maxcount compare register A. Maxcount compare register B is not used in this case.

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

5.1.31 T2CON (066h)

Timer2 Mode and CONtrol Registers. This register controls the operation of the Timer2. The value of the T2CON register is 0000h at reset (see Table 52).

Table 52. Timer2 Mode and Control Registers

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

- 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, but with the INHn bit set to 1 in the same write operation.

- Bit [14]—INH Inhibit Bit → Gates the setting of the enable (EN) bit. This bit must be set to 1 in the same write operation that sets the enable (EN) bit. This bit always reads as 0.

- 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.

- Bits [12–6]—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 may be used by software polling to monitor timer status rather than through interrupts if desired.

- Bits [4–1]—Reserved. Set to 0.

- Bit [0]—CONT Continuous Mode Bit → The timer will run continuously when this bit is set to 1. The timer will stop after each count run and EN will be cleared if this bit is set to 0.

5.1.32 T2COMPA (062h), T1COMPB (05ch), T1COMPA (05ah), T0COMPB (054h), and T0COMPA (052h)

Timer Maxcount COMpare Registers. These registers contain the maximum count value that is compared to the respective count register. Timer0 and Timer1 have two of these compare registers each.

If Timer0 and/or Timer1 is/are configured to count and compare firstly to register A and then register B, the TMROUT0 or TMROUT1 signals may be used to generate various duty-cycle wave forms.

Timer2 has only one compare register, T2COMPA.

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

The value of these registers is 0000h at reset (see Table 53).

Table 53. Timer Maxcount Compare Registers

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

- Bits [15–0]—TC [15–0] Timer Compare Value → The timer will count to the value in the respective register before resetting the count value to 0.

5.1.33 T2CNT (060h), T1CNT (058h), and T0CNT (050h)

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

The Increment of Timer0 and Timer1 may also be controlled by external signals tmrin0 and tmrin1 respectively, or prescaled by Timer2.

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 0000h at reset (see Table 54).

Table 54. Timer Count Registers

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

- Bits [15–0]—TC [15–0] Timer Count Value → This register has the value of the current count of the related timer that is incremented every fourth processor clock in internal clocked mode. Alternatively, the register is incremented each time the Timer2 maxcount is reached if using Timer2 as a prescaler. Timer0 and Timer1 may be externally clocked by tmrin0 and tmrin1 signals.

5.1.34 SP0CON (044h) and SP1CON (042h) (Master Mode)

Serial Port Interrupt CONtrol Registers. These registers control the operation of the serial ports' interrupt source. The value of these registers is 001Fh at reset (see Table 55).

Table 55. Serial Port Interrupt Control Registers

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

- Bits [15–5]—Reserved. Set to 0.
- Bit [4]—Reserved. Set to 1.
- Bit [3]—MSK Mask → When 0, this bit enables the serial port to cause an interrupt. When 1, it prevents the serial port from generating an interrupt.
- Bits [2–0]—PR [2–0] Priority. These bits define the priority of the serial port interrupt in relation to other interrupt signals. The interrupt priority is the lowest at 7 upon reset. The values of PR2–PR0 are shown below.

Values of PR2–PR0 by Priority

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

5.1.35 I4CON (040h) (Master Mode)

INT4 CONtrol Register. The int4 signal is intended only for use in fully nested mode and is not available in cascade mode. The value of the I4CON register is 000Fh at reset (see Table 56).

Table 56. INT4 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 int4 interrupt may be edge- or level-triggered, depending on the value of the bit. If LTM is 1, int4 is active high level-sensitive interrupt. If 0, it is a rising-edge triggered interrupt. The interrupt int4 must remain active (high) until serviced.
- Bit [3]—MSK Mask → The int4 signal can cause an interrupt if the MSK bit is 0. The int4 signal cannot cause an interrupt if the MSK bit is 1.
- Bit [2–0]—PR [2–0] Priority → These bits define the priority of the serial port interrupt in relation to other interrupt signals. The interrupt priority is the lowest at 7 upon reset. The values of PR2–PR0 are shown above.

5.1.36 I3CON (03eh) and I2CON (03ch) (Master Mode)

INT2/INT3 CONtrol Register. INT2 and INT3 are designated as interrupt type 0eh and 0fh, respectively.

The int2 and int3 pins may be configured as the interrupt acknowledge pins inta0 and inta1, respectively, the signals in cascade mode. The value of these registers is 000Fh at reset (see Table 57).

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

level-sensitive interrupt. If 0, it is a rising-edge triggered interrupt. The interrupt int0 or int1 must remain active (high) until acknowledged.

- Bit [3]—MSK Mask → The int0 or int1 signal can cause an interrupt if the MSK bit is 0. The int0 or int1 signal cannot cause an interrupt if the MSK bit is 1. The Interrupt Mask Register has a duplicate of this bit.

- Bit [2–0]—PR [2–0] Priority → These bits define the priority of the serial port interrupt int0 or int1 in relation to other interrupt signals. The interrupt priority is the lowest at 7 at reset. The values of PR2–PR0 are shown above.

5.1.38 TCUCON (032h) (Master Mode)

Timer Control Unit Interrupt CONtrol Register. The three timers, Timer2, Timer1, and Timer0, have their interrupts assigned to types 08h, 12h, and 13h and are configured by this register. The value of these registers is 000Fh at reset (see Table 59).

Table 59. Timer Control Unit Interrupt Control Register

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

- Bits [15–4]—Reserved. Set to 0.

- Bit [3]—MSK Mask → Any of the interrupt sources may cause an interrupt if the MSK bit is 0. The interrupt sources cannot cause an interrupt if the MSK bit is 1. The Interrupt Mask Register has a duplicate of this bit.

- Bit [2–0]—PR [2–0] Priority → These bits define the priority of the serial port interrupts in relation to other interrupt signals. The interrupt priority is the lowest at 7 upon reset. The values of PR2–PR0 are shown above.

5.1.39 T2INTCON (03ah), T1INTCON (038h), and T0INTCON (032h) (Slave Mode)

Timer INTerrupt CONtrol Register. The three timers, Timer2, Timer1, and Timer0, each have an interrupt control register, whereas in master mode all three are masked and prioritized in one register (TCUCON). The value of these registers is 000Fh at reset (see Table 60).

Table 60. Timer Interrupt Control Register

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

- Bits [15–4]—Reserved. Set to 0.
- Bit [3]—MSK Mask → Any of the interrupt sources may cause an interrupt if the MSK bit is 0. The interrupt sources cannot cause an interrupt if the MSK bit is 1. The Interrupt Mask Register has a duplicate of this bit.
- Bit [2–0]—PR [2–0] Priority → These bits define the priority of the serial port interrupts in relation to other interrupt signals. The interrupt priority is the lowest at 7 upon reset. The values of PR2–PR0 are shown above.

5.1.40 DMA1CON/INT6CON (036h) and DMA0CON/INT5CON (034h) (Master Mode)

DMA and INTerrupt CONtrol Register. The DMA0 and DMA1 interrupts have interrupt type 0ah and 0bh, respectively. These pins are configured as external interrupts or DMA requests in the respective DMA Control register. The value of these registers is 000Fh at reset (see Table 61).

Table 61. DMA and Interrupt Control Register (Master Mode)

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

- Bits [15–4]—Reserved. Set to 0.
- Bit [3]—MSK Mask → Any of the interrupt sources may cause an interrupt if the MSK bit is 0. The interrupt sources cannot cause an interrupt if the MSK bit is 1. The Interrupt Mask Register has a duplicate of this bit.
- Bits [2–0]—PR [2–0] Priority → These bits define the priority of the serial port interrupts in relation to other interrupt signals. The interrupt priority is the lowest at 7 upon reset. The values of PR2–PR0 are shown above.

5.1.41 DMA1CON/INT6 (036h) and DMA0CON/INT5 (034h) (Slave Mode)

DMA and INTerrupt CONtrol Register. The two DMA control registers maintain their original functions and addressing that they possessed in Master Mode. These pins are configured as external interrupts or DMA requests in the respective DMA Control register. The value of these registers is 000Fh at reset (see Table 62).

Table 62. DMA and Interrupt Control Register (Slave Mode)

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

- Bits [15–4]—Reserved. Set to 0.
- Bit [3]—MSK Mask → Any of the interrupt sources may cause an interrupt if the MSK bit is 0. The interrupt sources cannot cause an interrupt if the MSK bit is 1. The Interrupt Mask Register has a duplicate of this bit.
- Bits [2–0]—PR [2–0] Priority → These bits define the priority of the serial port interrupts in relation to other interrupt signals. The interrupt priority is the lowest at 7 upon reset. The values of PR2–PR0 are shown above.

5.1.42 INTSTS (030h) (Master Mode)

INTerrupt STatuS Register. The Interrupt status register contains the interrupt request status of each of the three timers, Timer2, Timer1, and Timer0 (see Table 63).

Table 63. Interrupt Status Register (Master Mode)

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

- Bit [15]—DHLT DMA Halt → DMA activity is halted when this bit is 1. It is set to 1 automatically when any non-maskable interrupt occurs and is cleared to 0 when an IRET instruction is executed. Interrupt handlers and other time-critical software may modify this bit directly to disable DMA transfers. However, the DHLT bit should not be modified by software if the timer interrupts are enabled as the function of this register as an interrupt request register for the timers would be compromised.
- Bits [14–3]—Reserved.
- Bits [2–0]—TMR [2–0] Timer Interrupt Request → When any of these bits is 1, a pending interrupt request is indicated by the respective timer.

Note: The TMR bit in the REQST register is a logical OR of these timer interrupt requests.

5.1.43 INTSTS (030h) (Slave Mode)

When NMIs occur, the interrupt status register controls DMA operation and the interrupt request status of each of the three timers, Timer2, Timer1, and Timer0 (see Table 64).

Table 64. Interrupt Status Register (Slave Mode)

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

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

- Bits [14–3]—Reserved.

- Bit [2–0]—TMR [2–0] Timer Interrupt Request → A pending interrupt request is indicated by the respective timer, when any of these bits is 1.

Note: The TMR bit in the REQST register is a logical OR of these timer interrupt requests.

5.1.44 REQST (02eh) (Master Mode)

Interrupt REQueST Register. This is a read-only register and such a read results in the status of the interrupt request bits presented to the interrupt controller. The REQST register is undefined on reset (see Table 65).

Table 65. Interrupt Request Register (Master Mode)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					SP0	SP1	I4	I3	I2	I1	IO	D1/I6	D0/I5	Res	TMR

- Bits [15–11]—Reserved.

- Bit [10]—SP0 Serial Port 0 Interrupt Request → This is the serial port 0 interrupt state and when enabled is the logical OR of all the serial port 0 interrupt sources, THRE, RDR, BRK1, BRK0, FER, PER, and OER.

- Bit [9]—SP1 Serial Port 1 Interrupt Request → This is the serial port 1 interrupt state and when enabled is the logical OR of all the serial port 1 interrupt sources, THRE, RDR, BRK1, BRK0, FER, PER, and OER.

- Bits [8–4]—I [4–0] Interrupt Requests → When any of these bits is set to 1, it indicates that the relevant interrupt has a pending interrupt.

- Bit [3]—D1/I6 DMA Channel 1/Interrupt 6 Request → When set to 1, it indicates that either the DMA channel 1 or int6 has a pending interrupt.

- Bit [2]—D0/I5 DMA Channel 0/Interrupt 5 Request → When set to 1, it indicates that either the DMA channel 0 or int5 has a pending interrupt.

- Bit [1]—Reserved.
- Bit [0]—TMR Timer Interrupt Request → This is the timer interrupt state and is the logical OR of the timer interrupt requests. When set to 1, it indicates that the timer control unit has a pending interrupt.

5.1.45 REQST (02eh) (Slave Mode)

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

When an internal interrupt request (D1/I6, D0/I5, 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 66).

Table 66. 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/I6	D0/I5	Res	TMR0

- Bits [15–6]—Reserved.
- Bit [5]—TMR2 Interrupt Requests → When set to 1, it indicates that Timer2 has a pending interrupt.
- Bit [4]—TMR1 Interrupt Requests → When set to 1, it indicates that Timer1 has a pending interrupt.
- Bit [3]—D1/I6 DMA Channel 1/Interrupt 6 Request → When set to 1, it indicates that either the DMA channel 1 or int6 has a pending interrupt.
- Bit [2]—D0/I5 DMA Channel 0/Interrupt 5 Request → When set to 1, it indicates that either the DMA channel 0 or int5 has a pending interrupt.
- Bit [1]—Reserved.
- Bit [0]—TMR0 Timer Interrupt Request → When set to 1, it indicates that Timer0 has a pending interrupt.

5.1.46 INSERV (02ch) (Master Mode)

IN-SERVice Register. The interrupt controller sets the bits in this register when the interrupt is taken. The INSERV register contains 0000h on reset (see Table 67).

Table 67. In-Service Register (Master Mode)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					SP0	SP1	I4	I3	I2	I1	IO	D1/I6	D0/I5	Res	TMR

- Bits [15–11]—Reserved.
- Bit [10]—SP0 Serial Port 0 Interrupt Request → This is the Serial Port 0 interrupt state.
- Bit [9]—SP1 Serial Port 1 Interrupt Request → This is the Serial Port 1 interrupt state.
- Bits [8–4]—I [4–0] Interrupt Requests → When any of these bits is set to 1, it indicates that the relevant interrupt has a pending interrupt.
- Bit [3]—D1/I6 DMA Channel 1/Interrupt 6 Request → When set to 1, it indicates that either the DMA channel 1 or int6 has a pending interrupt.
- Bit [2]—D0/I5 DMA Channel 0/Interrupt 5 Request → When set to 1 it indicates that either the DMA channel 0 or int5 has a pending interrupt.
- Bit [1]—Reserved.
- Bit [0]—TMR Timer Interrupt Request → This is the timer interrupt state and is the logical OR of the timer interrupt requests. When set to 1, it indicates that the timer control unit has a pending interrupt.

5.1.47 INSERV (02ch) (Slave Mode)

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

When an internal interrupt request (D1/I6, D0/I5, TMR2, TMR1, and TMR0) occurs, the respective bit is set to 1. The internally generated interrupt acknowledge resets these bits. The REQST register contains 0000h on reset (see Table 68).

Table 68. In-Service Register (Slave Mode)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										TMR2	TMR1	D1/I6	D0/I5	Res	TMR0

- Bits [15–6]—Reserved.
- Bit [5]—TMR2 Timer2 Interrupt In Service → Timer2 is being serviced when this bit is set to 1.

- Bit [4]—TMR1 Timer1 Interrupt IN Service → Timer1 is being serviced when this bit is set to 1.
- Bit [3]—D1/I6 DMA Channel Interrupt 6 In Service → DMA channel 1 or int6 is being serviced when this bit is set to 1.
- Bit [2]—D0/I5 DMA Channel Interrupt 5 IN Service → DMA channel 0 or int5 is being serviced when this bit is set to 1.
- Bit [1]—Reserved.
- Bit [0]—TMR0 Timer Interrupt In Service → Timer0 is being serviced when this bit is set to 1.

5.1.48 PRIMSK (02ah) (Master and Slave Mode)

PRIority MaSK Register. This register contains a value that sets the minimum priority level that a maskable interrupt must have to generate an interrupt. The PRIMSK register contains 0007h on reset (see Table 69).

Table 69. Priority Mask Register

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

- Bits [15–3]—Reserved. Set to 0.
- Bits [2–0]—PRM [2–0] Priority Field Mask → This three-bit field sets the minimum priority necessary for a maskable interrupt to generate an interrupt. Any maskable interrupt with a numerically higher value than that contained by these three bits, are masked. The values of PR2–PR0 are shown above.
  - Any unmasked interrupt may generate an interrupt if the priority level is set to 7. However, by way of example, if the priority level is set to 4, only unmasked interrupts with a priority of 0 to 5 are permitted to generate interrupts.

5.1.49 IMASK (028h) (Master Mode)

Interrupt MASK Register. The interrupt mask register is read/write. Setting a bit in this register is effectively the same as setting the MSK bit in the corresponding interrupt control register. Setting a bit to 1 masks the interrupt. The interrupt request is enabled when the corresponding bit is set to 0. The IMASK register contains 07fdh on reset (see Table 70).

Table 70. Interrupt MASK Register (Master Mode)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					SP0	SP1	I4	I3	I2	I1	IO	D1/I6	D0/I5	Res	TMR

- Bits [15–11]—Reserved.
- Bit [10]—SP0 Serial Port 0 Interrupt Mask → Setting this bit to 1 is an indication that the serial port 0 interrupt is masked.
- Bit [9]—SP1 Serial Port 1 Interrupt Mask → Setting this bit to 1 is an indication that the serial port 0 interrupt is masked.
- Bits [8–4]—I [4–0] Interrupt Mask → When any of these bits is set to 1, it is an indication that the relevant interrupt is masked.
- Bit [3]—D1/I6 DMA Channel 1/Interrupt 6 Mask → Setting this bit to 1, is an indication that either the DMA channel 1 or int6 interrupt is masked.
- Bit [2]—D0/I5 DMA Channel 0/Interrupt 5 Mask → When set to 1, it indicates that either the DMA channel 0 or int5 interrupt is masked.
- Bit [1]—Reserved.
- Bit [0]—TMR Timer Interrupt Mask → When set to 1, it indicates that the timer control unit interrupt is masked.

5.1.50 IMASK (028h) (Slave Mode)

The interrupt mask register is read/write. Setting a bit in this register has the effect of setting the MSK bit in the corresponding interrupt control register. Setting a bit to 1, masks the interrupt request. The interrupt request is enabled when the corresponding bit is set to 0. The IMASK register contains 003dh on reset (see Table 71).

Table 71. Interrupt MASK Register (Slave Mode)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										TMR2	TMR1	D1/I6	D0/I5	Res	TMR0

- Bits [15–6]—Reserved.
- Bit [5]—TMR2 Timer2 Interrupt Mask → This bit provides the state of the mask bit in the Timer Interrupt Control register. When set to 1, it indicates that the interrupt request is masked.

- Bit [4]—TMR1 Timer1 Interrupt Mask → This bit provides the state of the mask bit in the Timer Interrupt Control register. When set to 1, it indicates that the interrupt request is masked.

- Bit [3]—D1/I6 DMA Channel Interrupt 6 Mask → This bit provides the state of the mask bit in the DMA channel 0 or int5 Interrupt Control register. When set to 1, it indicates that the interrupt request is masked.

- Bit [2]—D0/I5 DMA Channel Interrupt 5 Mask → This bit provides the state of the mask bit in the DMA channel 1 or int6 Interrupt Control register. When set to 1, it indicates that the interrupt request is masked.

- Bit [0]—TMR0 Timer Interrupt Mask → This bit provides the state of the mask bit in the Timer Interrupt Control register. When set to 1, it indicates that the interrupt request is masked.

5.1.51 POLLST (026h) (Master Mode)

POLL STatus Register. This register reflects the current state of the poll register and can be read without affecting its contents. However, the current interrupt is acknowledged and replaced by the next interrupt when the poll register is read. The poll status register is read-only (see Table 72).

Table 72. POLL Status Register

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

- Bit [15]—IREQ Interrupt Request → This bit is set to 1 when an interrupt is pending and during this state, the S4–S0 bits contain valid data.

- Bits [14–6]—Reserved.

- Bit [5–0]—S [5–0] Poll Status → These bits show the interrupt type of the highest priority pending interrupt.

5.1.52 POLL (024h) (Master Mode)

POLL Register. The current interrupt is acknowledged and replaced by the next interrupt when the poll register is read. The poll status register reflects the current state of the poll register and can be read without affecting its contents. The poll register is read-only (see Table 73).

Table 73. Poll Register

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

- Bit [15]—IREQ Interrupt Request → This bit is set to 1 when an interrupt is pending and during this state, the S4–S0 bits contain valid data.
- Bits [14–6]—Reserved.
- Bit [4–0]—S [4–0] Poll Status → These bits show the interrupt type of the highest priority pending interrupt.

5.1.53 EOI (022h) End-Of-Interrupt Register (Master Mode)

The In Service flags of the INSERV register are reset when a write is made to the EOI register. The interrupt service routine (ISR) should write to the EOI to reset the IS bit, in the INSERV register, for the interrupt before executing an IRET instruction than ends an interrupt service routine. Because it is the most secure, the specific EOI reset is the preferred method for resetting the IS bits. The EOI register is write-only (see Table 74).

Table 74. End-Of-Interrupt Register

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

- Bit [15]—NSPEQ Non-Specific EOI → When set to 1, this bit is a non-specific EOI. When 0, it indicates the specific EOI.
- Bits [14–5]—Reserved.
- Bit [4–0]—S [4–0] Source Interrupt Type → These bits show the interrupt type of the highest priority pending interrupt.

5.1.54 EOI (022h) Specific End-Of-Interrupt Register (Slave Mode)

Specific End-Of-Interrupt Register. An In Service flag of a specific priority in the INSERV register is reset when a write is made to the EOI register. A three-bit, user-supplied priority-level value points to the in-service bit that is to be reset. Writing this value to this register resets the specific bit. Because it is the most secure, the specific EOI reset is the preferred method for resetting the IS bits. The EOI register is write-only and undefined at reset (see Table 75).

Table 75. Specific End-Of-Interrupt Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	L2–L0		

- Bits [15–3]—Reserved. Write as 0.
- Bit [2–0]—L [2–0] Interrupt Type → The priority or the IS (interrupt service) bit to be reset is encoded in these three bits. Writing to these bits causes the issuance of an EOI for the interrupt type (see Table 14, Interrupt Types).

5.1.55 INTVEC (020h) Interrupt Vector Register (Slave Mode)

The CPU shifts left 2 bits (i.e., it multiplies by 4) an 8-bit interrupt type, generated by the interrupt controller, to produce an offset into the interrupt vector table. The INTVEC register is undefined at reset (see Table 76).

Table 76. Interrupt Vector Register

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

- Bits [15–8]—Reserved. Read as 0.
- Bits [7–3]—T [4–0] Interrupt Type → These five bits contain the five most significant bits of the types used for the internal interrupt. The least significant three bits of the interrupt type are supplied by the interrupt controller, as set by the priority level of the interrupt request.
- Bits [2–0]—Reserved. Read as 0.

5.2 Reference Documents

Additional information on the operation and programming of the IA186ES/ IA188ES can be found in the following AMD publications:

- *Am186™ES and Am188™ES User's Manual* (Publication 21096).
- *Am186™ES/ESLV and Am188™ES/ESLV Preliminary Data Sheet* (Publication 20002).

6. AC Specifications

Table 77 and Table 78 present the alphabetic and numeric keys to waveform parameters, respectively. Figure 12 presents the read cycle. Figure 13 presents the multiple read cycles. Table 79 presents the read cycle timing. Figure 14 presents the write cycle. Table 80 presents the write cycle timing. Figure 15 presents the multiple write cycles.

Figure 16 presents the PSRAM read cycle. Table 81 presents the PSRAM read cycle timing. Figure 17 presents the PSRAM write cycle. Table 82 presents the PSRAM write cycle timing. Figure 18 presents the PSRAM refresh cycle. Table 83 presents the PSRAM refresh cycle timing. Figure 19 presents the interrupt acknowledge cycle. Table 84 presents the interrupt acknowledge cycle timing. Figure 20 presents the software halt cycle. Table 85 presents the software halt cycle timing. Figure 21 presents the active mode. Figure 22 presents the power-save mode. Table 86 presents the clock timing.

Figure 23 presents the srdy—synchronous ready. Figure 24 presents the ardy—asynchronous ready. Figure 25 presents the peripherals. Table 87 presents the ready and peripheral timing. Figure 26 and Figure 27 present Reset 1 and Reset 2, respectively. Figure 28 and Figure 29 present the bus hold entering and bus hold leaving, respectively. Table 88 presents the reset and bus hold timing.

Table 77. Alphabetic Key to Waveform Parameters

No.	Name	Description	Min[a]	Max[a]
49	tARYCH	ardy Resolution Transition Setup Time	9	–
51	tARYCHL	ardy Inactive Holding Time	6	–
52	tARYLCL	ardy Setup Time	9	–
87	tAVBL	*a* Address Valid to whb_n/wlb_n Low	tCHCL-1.5	tCHCL
14	tAVCH	*ad* Address Valid to Clock High	0	–
12	tAVLL	*ad* Address Valid to ale Low	tCHCL	–
66	tAVRL	*a* Address Valid to rd_n Low	tCLCL+tCHCL	–
65	tAVWL	*a* Address Valid to wr_n Low	tCLCL+tCHCL	–
24	tAZRL	*ad* Address Float to rd_n Active	0	–
45	tCH1CH2	clkouta Rise Time	0	3
68	tCHAV	clkouta High to a Address Valid	0	8
38	tCHCK	x1 High Time	7.5	–
44	tCHCL	clkouta High Time	tCLCL/2	–
67	tCHCSV	clkouta High to lcs_n/usc_n Valid	0	9
18	tCHCSX	mcs_n/pcs_n Inactive Delay	0	12
22	tCHCTV	Control Active Delay 2	0	10
64	tCHCV	Command Lines Valid Delay (after Float)	0	12
63	tCHCZ	Command Lines Float Delay	0	12

[a]In nanoseconds.

Table 77. Alphabetic Key to Waveform Parameters *(Continued)*

No.	Name	Description	Min[a]	Max[a]
8	tCHDX	Status Hold Time	0	–
9	tCHLH	ale Active Delay	0	8
11	tCHLL	ale Inactive Delay	0	8
79	tCHRFD	clkouta High to rfsh_n Valid	0	12
3	tCHSV	Status Active Delay	0	6
69	tCICOA	x1 to clkouta Skew	–	25
70	tCICOB	x1 to clkoutb Skew	–	35
39	tCKHL	x1 Fall Time	–	5
36	tCKIN	x1 Period	25	66
40	tCKLH	x1 Rise time	–	5
46	tCL2CL1	clkouta Fall Time	–	3
50	tCLARX	ardy Active Hold Time	4	–
5	tCLAV	*ad* Address Valid Delay	0	12
6	tCLAX	Address Hold	0	12
15	tCLAZ	*ad* Address Float Delay	0	12
43	tCLCH	clkouta Low Time	tCLCL/2	–
37	tCLCK	x1 Low Time	7.5	–
42	tCLCL	clkouta Period	25	–
80	tCLCLX	lcs_n Inactive Delay	0	9
81	tCLCSL	lcs_n Active Delay	0	9
16	tCLCSV	mcs_n/pcs_n Inactive Delay	0	12
30	tCLDOX	Data Hold Time	0	–
7	tCLDV	Data Valid Delay	0	12
2	tCLDX	Data in Hold	0	–
62	tCLHAV	hlda Valid Delay	0	7
82	tCLRF	clkouta High to rfsh_n Invalid	0	12
27	tCLRH	rd_n Inactive Delay	0	10
25	tCLRL	rd_n Active Delay	0	10
4	tCLSH	Status Inactive Delay	0	6
48	tCLSRY	srdy Transition Hold Time	3	–
55	tCLTMV	Timer Output Delay	0	12
83	tCOAOB	clkouta to clkoutb Skew	3	1
20	tCVCTV	Control Active Delay 1	–	10
31	tCVCTX	Control Inactive Delay	0	10
21	tCVDEX	den_n Inactive Delay	0	9
17	tCXCSX	mcs_n/pcs_n Hold from Command Inactive	tCLCH	–
98	tDSHDIW	ds_n High to Data Invalid (Write)	0	tCHCL
41	tDSHLH	ds_n Inactive to ale Inactive	tCLCH	–

[a]In nanoseconds.

Table 77. Alphabetic Key to Waveform Parameters *(Continued)*

No.	Name	Description	Min[a]	Max[a]
1	tDVCL	Data in Setup	10	–
19	tDXDL	den_n Inactive to dt/r_n Low	0	–
58	tHVCL	hld Setup Time	10	–
53	tINVCH	Peripheral Setup Time	10	–
54	tINVCL	drq Setup Time	10	–
86	tLCRF	lcs_n Inactive to rfsh_n Active Delay	2tCLCL	–
23	tLHAV	ale High to Address Valid	7.5	–
10	tLHLL	ale Width	tCLCH-5	–
13	tLLAX	*ad* Address Hold from ale Inactive	tCHCL	–
61	tLOCK	Maximum PLL Lock Time	–	0.5
84	tLRLL	lcs_n Precharge Pulse Width	tCLCL+tCLCHH	–
57	tRESIN	res_n Setup Time	10	–
85	tRFCY	rfsh_n Cycle Time	6tCLCL	–
29	tRHAV	rd_n Inactive to *ad* Address Active	tCLCL	–
59	tRHDX	rd_n High to Data Hold on *ad* Bus	0	–
28	tRHLH	rd_n Inactive to ale High	tCLCH	–
26	tRLRH	rd_n Pulse Width	tCLCL	–
47	tSRYCL	srdy Transition Setup Time	10	–
35	tWHDEX	wr_n Inactive to den_n Inactive	tCLCH	–
34	tWHDX	Data Hold after wr_n	tCLCL	–
33	tWHLH	wr_n Inactive to ale High	tCLCH	–
32	tWLWH	wr_n Pulse Width	2tCLCL	–

[a]In nanoseconds.

Table 78. Numeric Key to Waveform Parameters

No.	Name	Description	Min[a]	Max[a]
1	tDVCL	Data in Setup	10	–
2	tCLDX	Data in Hold	0	–
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	–

[a]In nanoseconds.

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

No.	Name	Description	Min[a]	Max[a]
52	tARYLCL	ardy Setup Time	9	–
53	tINVCH	Peripheral Setup Time	10	–
54	tINVCL	drq Setup Time	10	–
55	tCLTMV	Timer Output Delay	0	12
57	tRESIN	res_n Setup Time	10	–
58	tHVCL	hld Setup Time	10	–
59	tRHDX	rd_n High to Data Hold on *ad* Bus	0	–
61	tLOCK	Maximum PLL Lock Time	–	0.5
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
65	tAVWL	a Address Valid to wr_n Low	tCLCL+tCHCL	–
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
69	tCICOA	x1 to clkouta Skew	–	25
70	tCICOB	x1 to clkoutb Skew	–	35
79	tCHRFD	clkouta High to rfsh_n Valid	0	12
80	tCLCLX	lcs_n Inactive Delay	0	9
81	tCLCSL	lcs_n Active Delay	0	9
82	tCLRF	clkouta High to rfsh_n Invalid	0	12
83	tCOAOB	clkouta to clkoutb Skew	3	11
84	tLRLL	lcs_n Precharge Pulse Width	tCLCL+tCLCH	–
85	tRFCY	rfsh_n Cycle Time	6tCLCL	–
86	tLCRF	lcs_n Inactive to rfsh_n Active Delay	2tCLCL	–
87	tAVBL	a Address Valid to whb_n/wlb_n Low	tCHCL-1.5	tCHCL
98	tDSHDIW	ds_n High to Data Invalid (Write)	0	tCLCL
99	tPLAL	pcs Low to ale Low	–	tCLCH

[a]In nanoseconds.

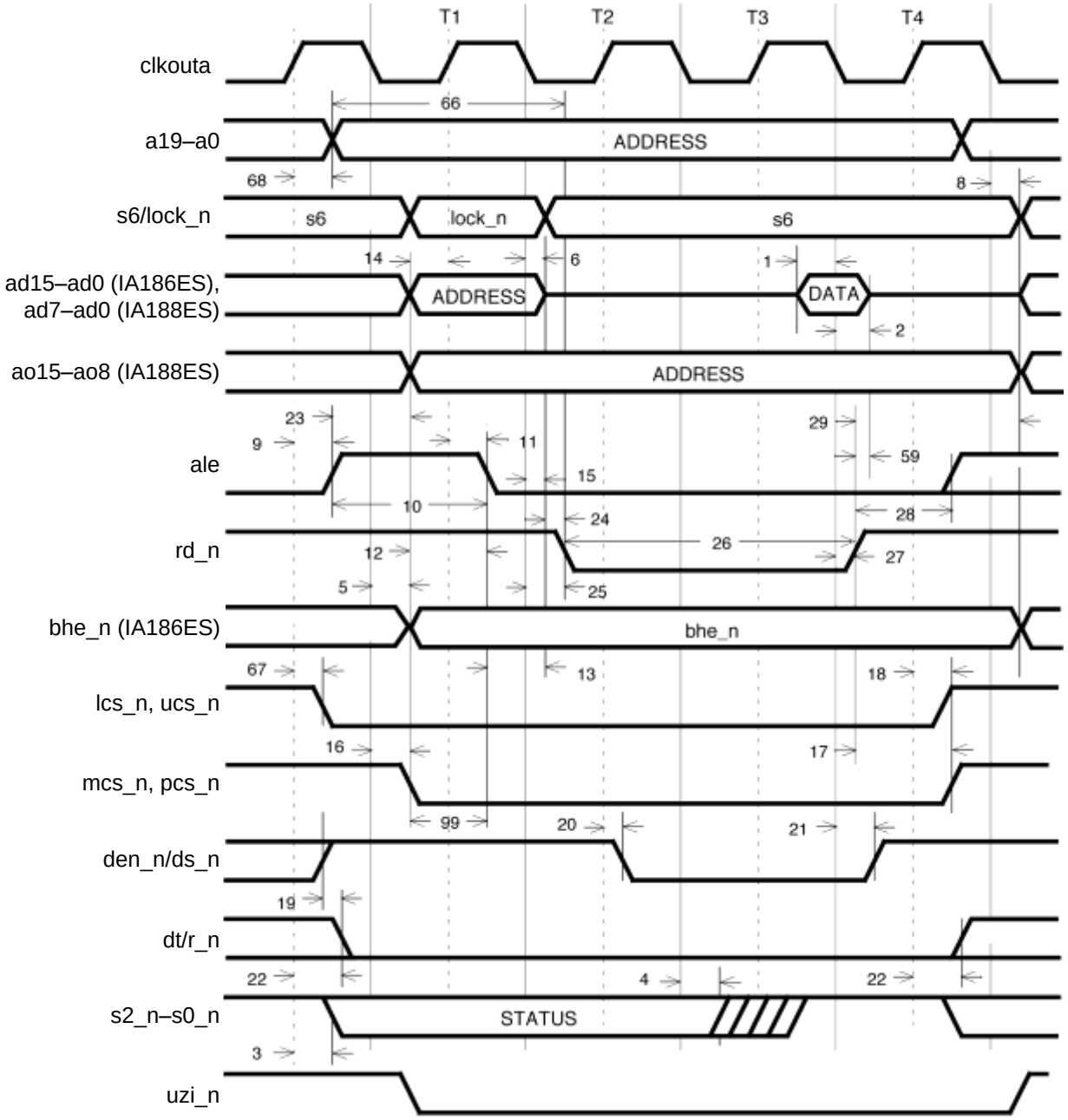


Figure 12. Read Cycle

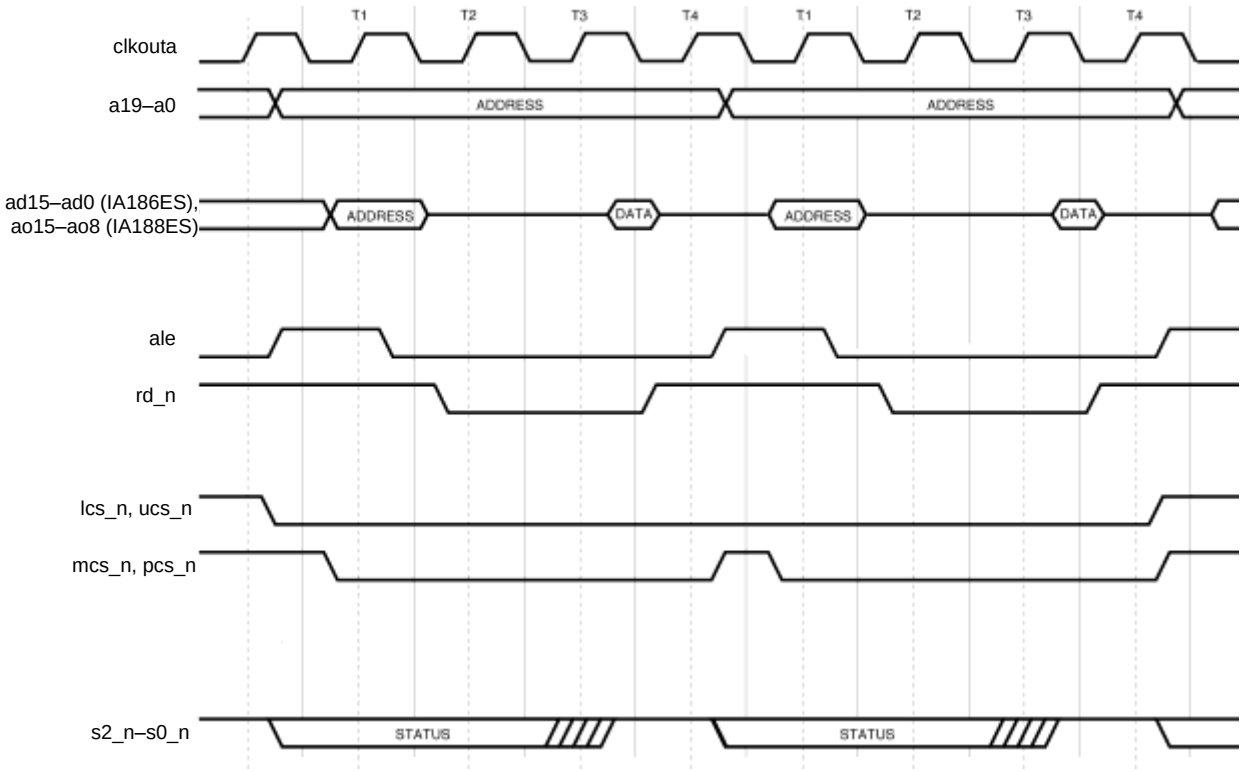


Figure 13. Multiple Read Cycles

Table 79. 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	–
99	tPLAL	pcs Low to ale Low	–	tCLCH
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	–
41	tDSHLH	ds_n Inactive to ale Inactive	tCLCH	–
59	tRHDX	rd_n High to Data Hold on *ad* Bus	0	–
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.

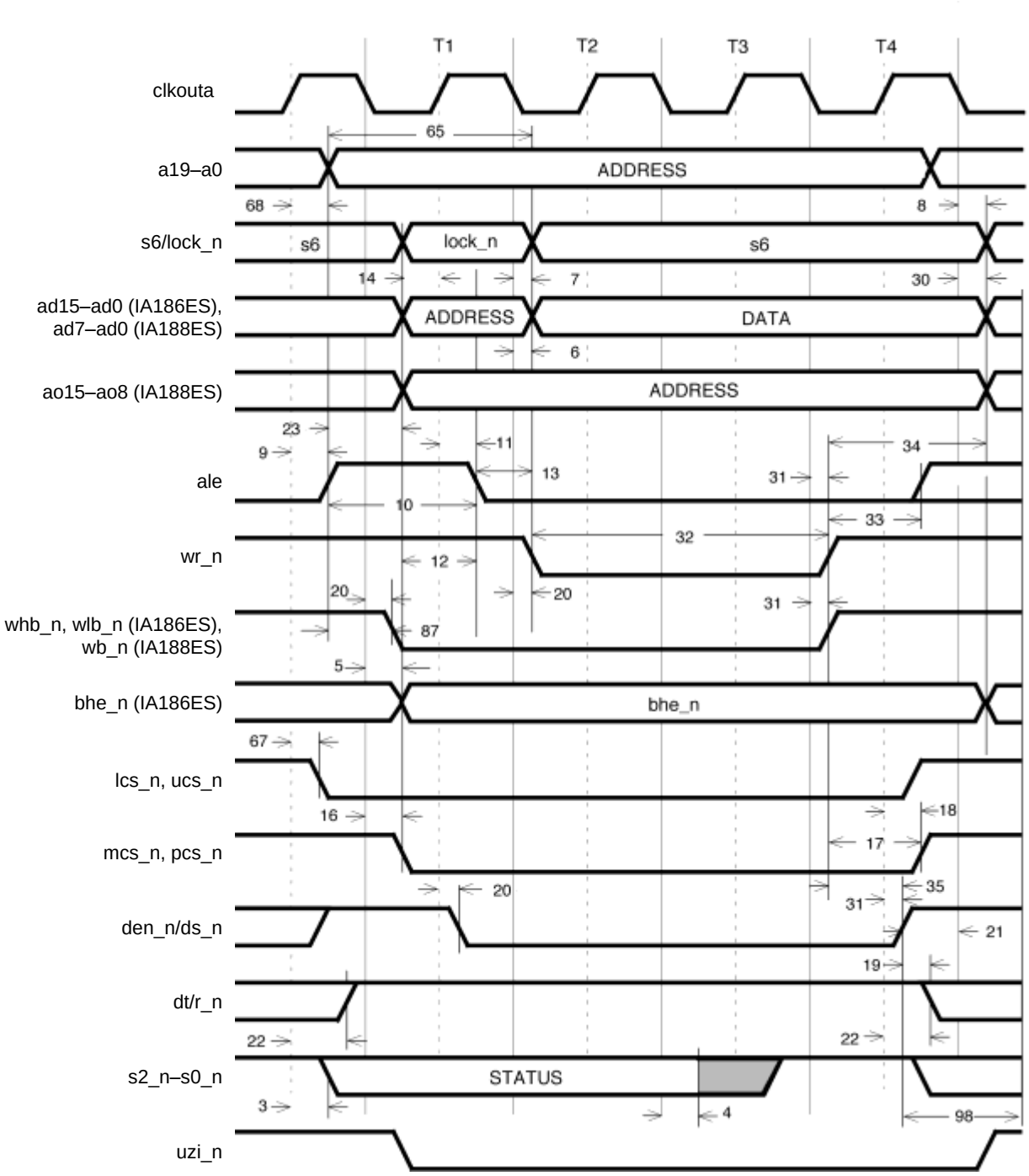


Figure 14. Write Cycle

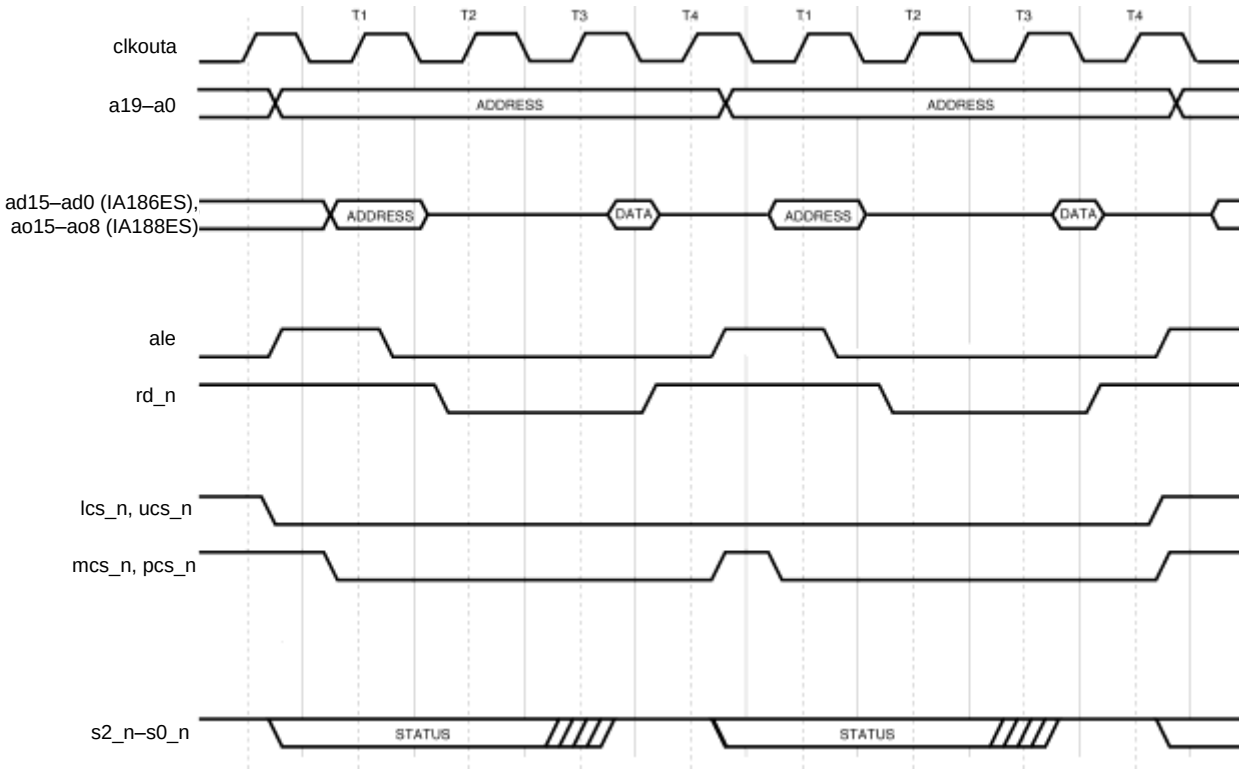


Figure 15. Multiple Write Cycles

Table 80. 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
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	–
99	tPLAL	pcs Low to ale Low	–	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	–
35	tWHDEX	wr_n Inactive to den_n Inactive	tCLCH	–
41	tDSHLH	ds_n Inactive to ale 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	tCHCL
98	tDSHDIW	ds_n High to Data Invalid (Write)	0	tCLCL

[a]In nanoseconds.

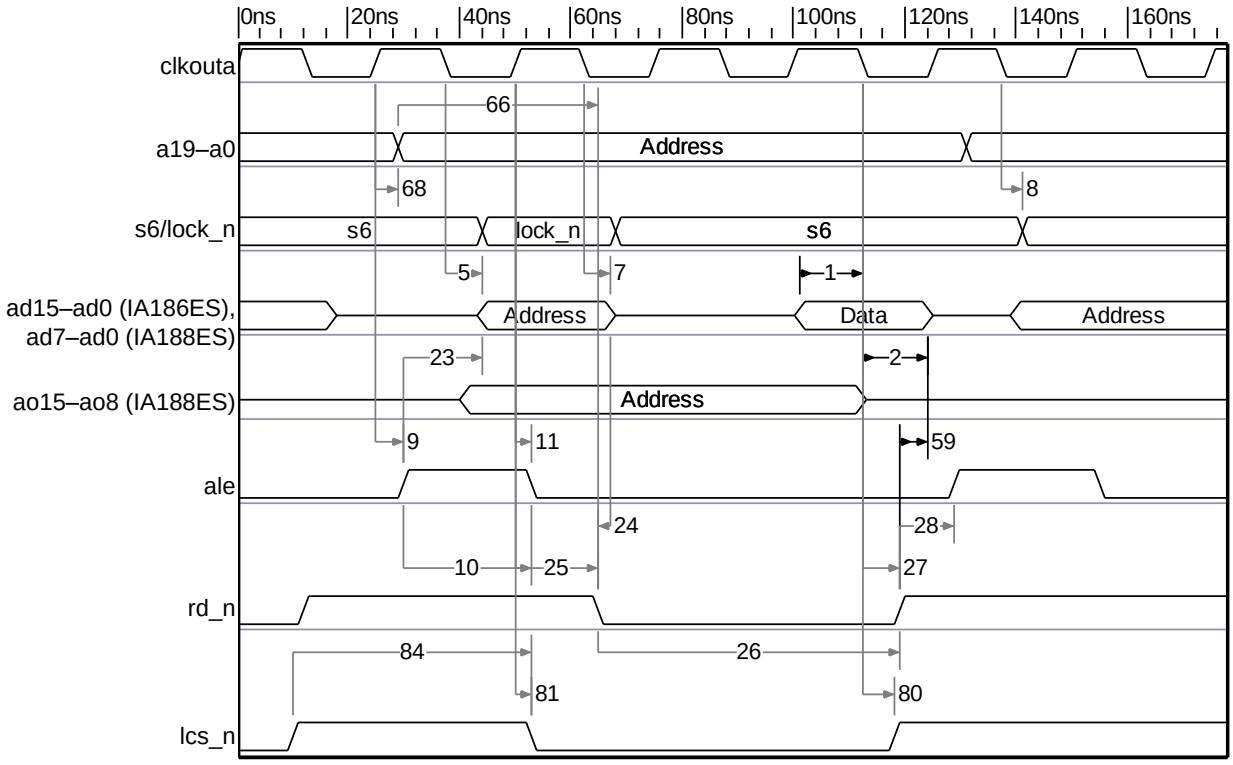


Figure 16. PSRAM Read Cycle

Table 81. PSRAM 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				
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	–
99	tPLAL	pcs Low to ale Low	–	tCLCH
Read 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	–
41	tDSHLH	ds_n Inactive to ale 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	tCHCL
98	tDSHDIW	ds_n High to Data Invalid (Write)	0	tCLCL

[a]In nanoseconds.

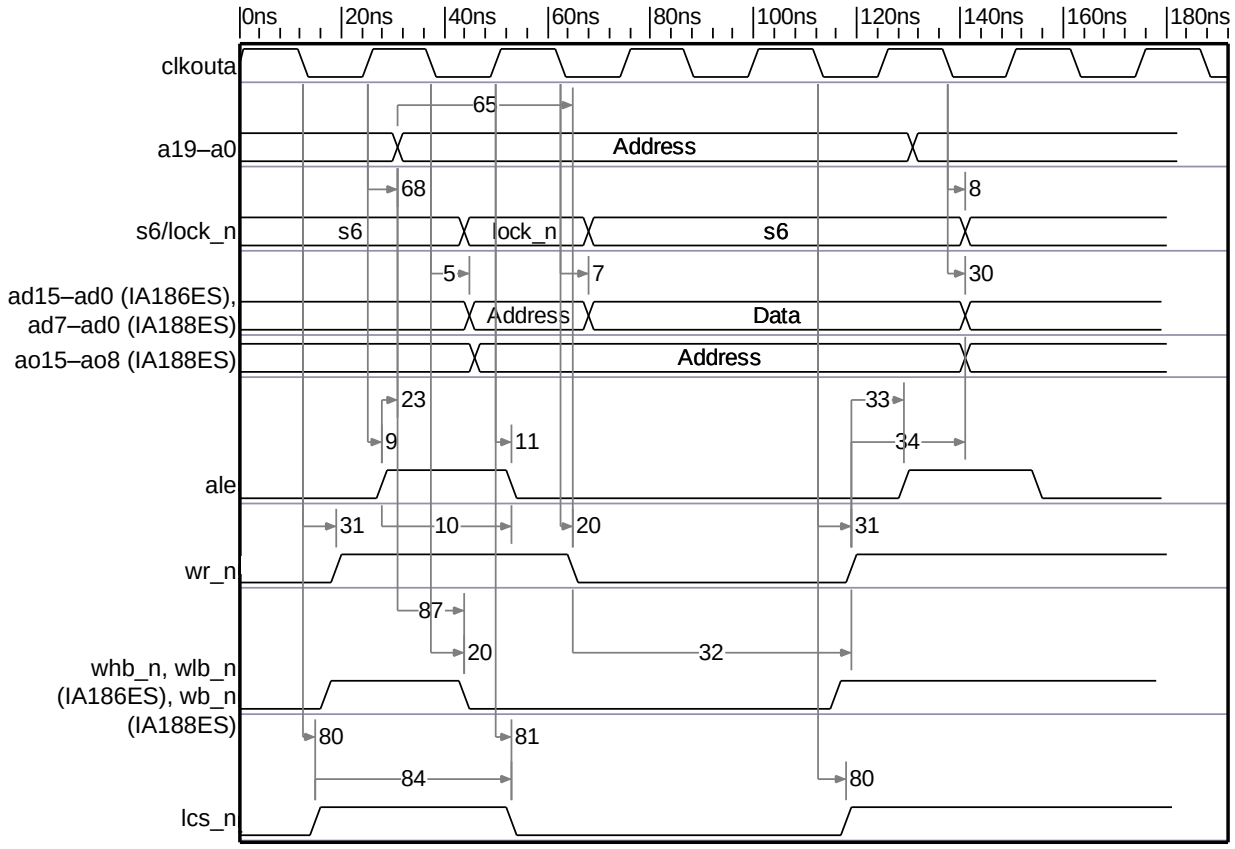


Figure 17. PSRAM Write Cycle

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.

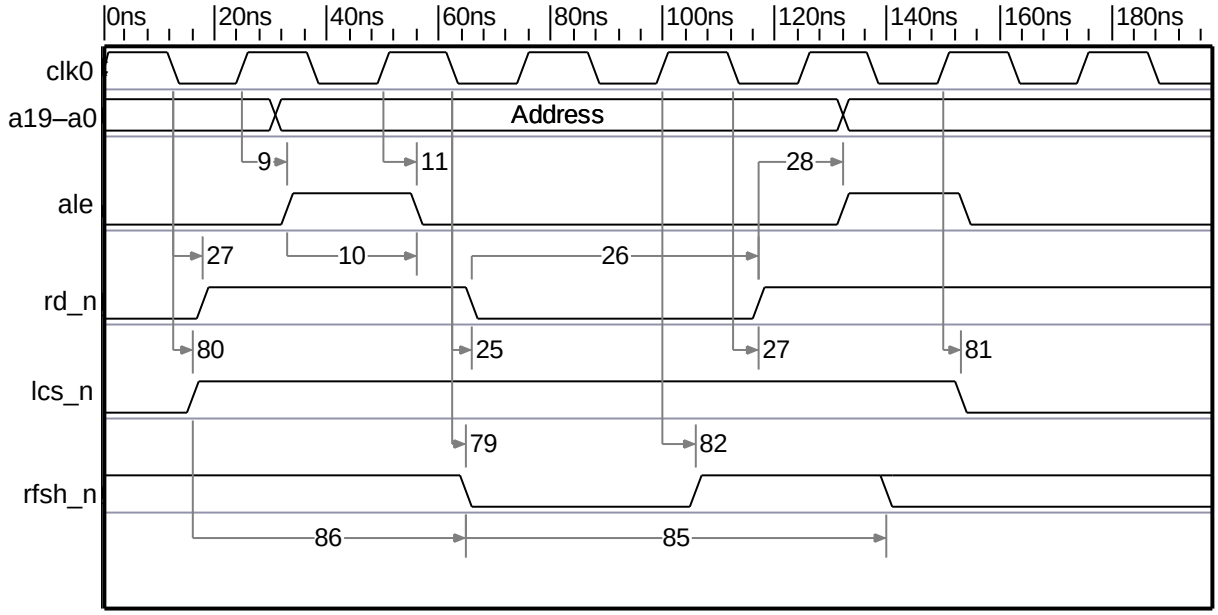


Figure 18. PSRAM Refresh Cycle

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.

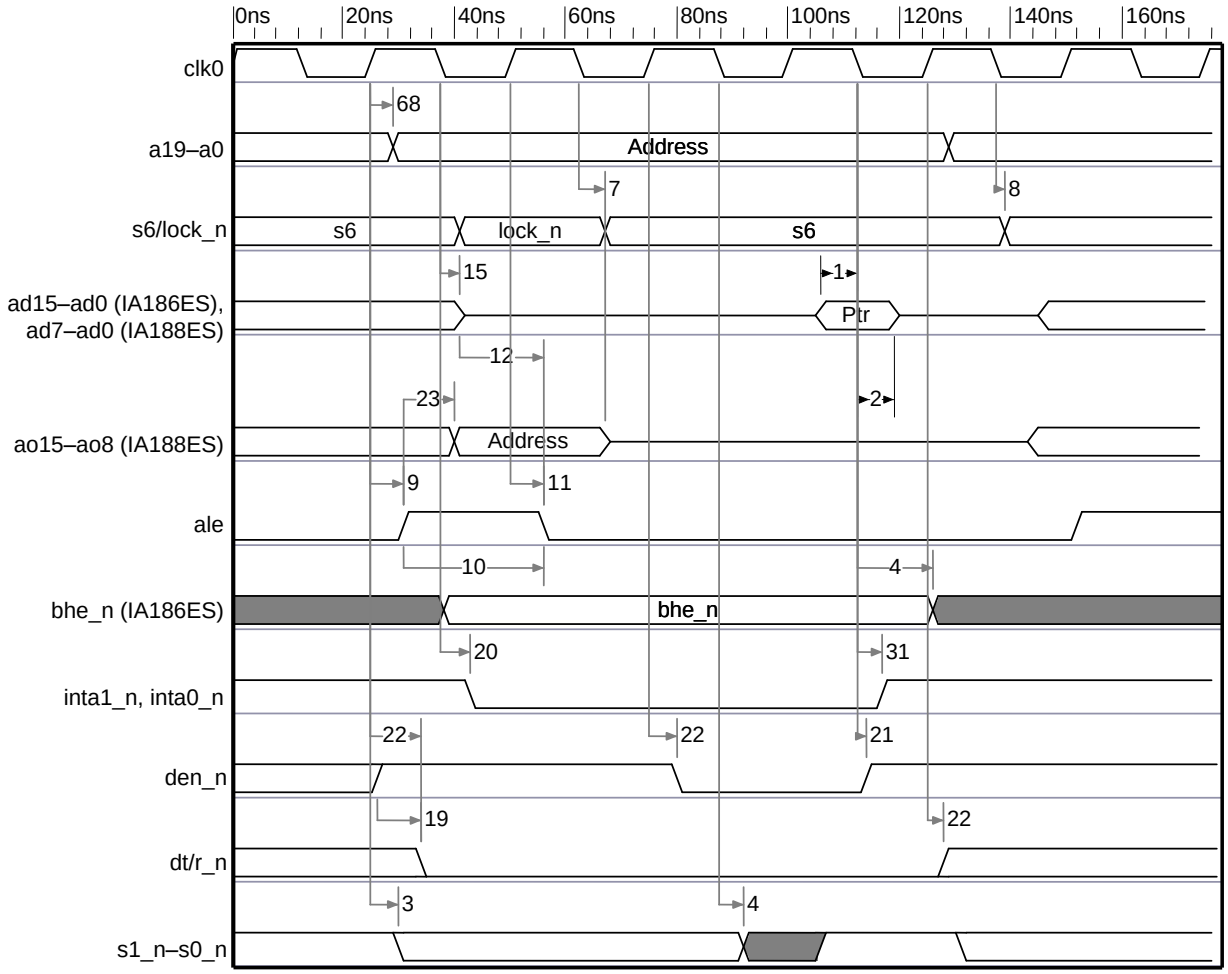


Figure 19. Interrupt Acknowledge Cycle

Table 84. Interrupt Acknowledge 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
9	tCHLH	ale Active Delay	0	8
10	tLHLL	ale Width	tCLCH-5	–
11	tCHLL	ale Inactive Delay	0	8
19	tDXDL	den_n Inactive to dt/r_n Low	0	–
22	tCHCTV	Control Active Delay 2	0	10
68	tCHAV	clkouta High to a Address Valid	0	8

[a]In nanoseconds.

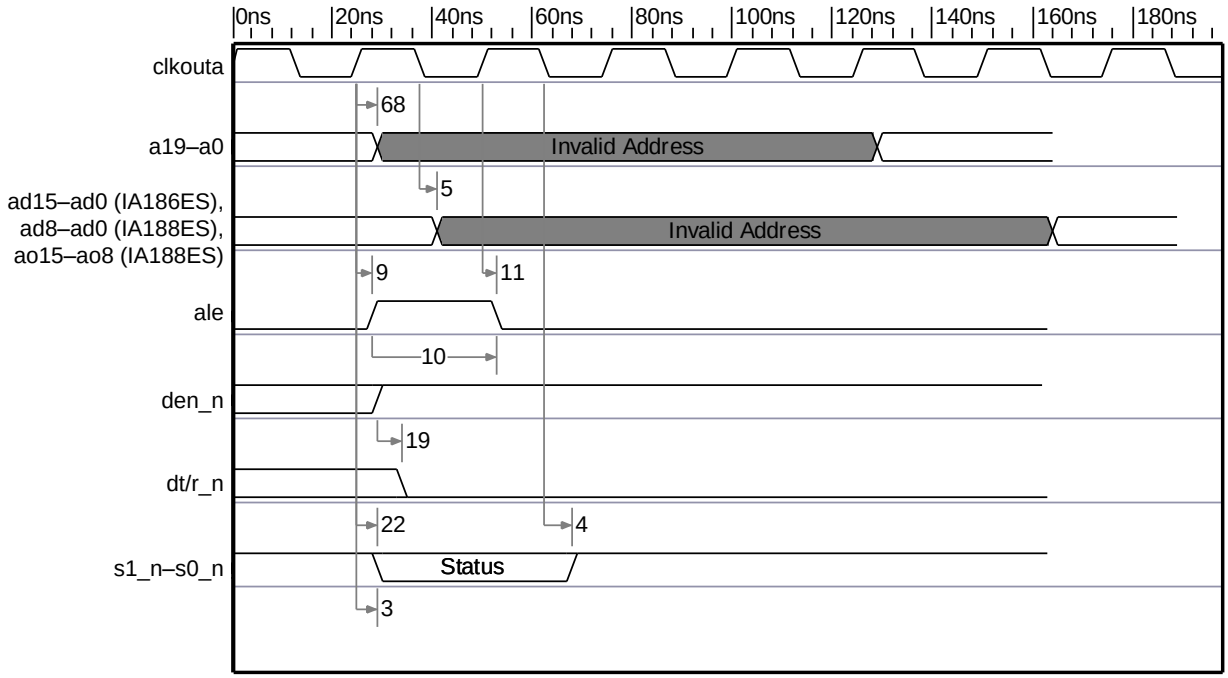


Figure 20. Software Halt Cycle

Table 85. Software Halt Cycle Timing

No.	Name	Description	Min[a]	Max[a]
General Timing Responses				
3	tCHSV	Status Active Delay	0	6
4	tCLSH	Status Inactive Delay	0	6
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	–
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	–
31	tCVCTX	Control Inactive Delay	0	10
68	tCHAV	clkouta High to a Address Valid	0	8

[a]In nanoseconds.

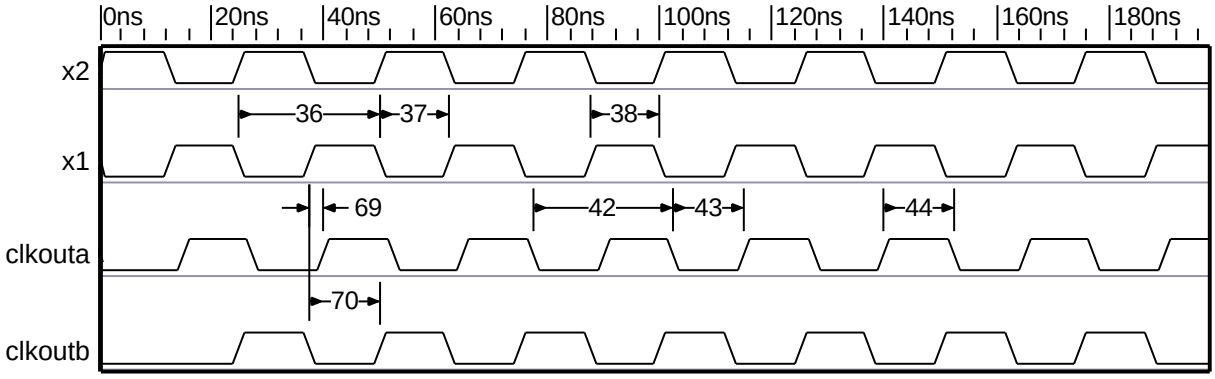


Figure 21. Clock—Active Mode

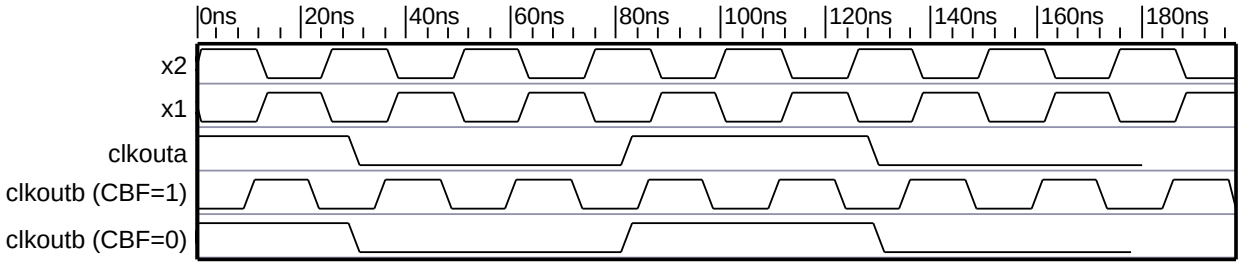


Figure 22. Clock—Power-Save Mode

Table 86. Clock Timing

No.	Name	Description	Min[a]	Max[a]
clkin Requirements				
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
clkout Requirements				
42	tCLCL	clkouta Period	25	–
43	tCLCH	clkouta Low Time	tCLCL/2	–
44	tCHCL	clkouta High Time	tCLCL/2	–
45	tCH1CH2	clkouta Rise Time	0	3
46	tCL2CL1	clkouta Fall Time	0	3
61	tLOCK	Maximum PLL Lock Time	–	0.5
69	tCICOA	x1 to clkouta Skew	–	25
70	tCICOB	x1 to clkoutb Skew	–	35

[a]In nanoseconds.

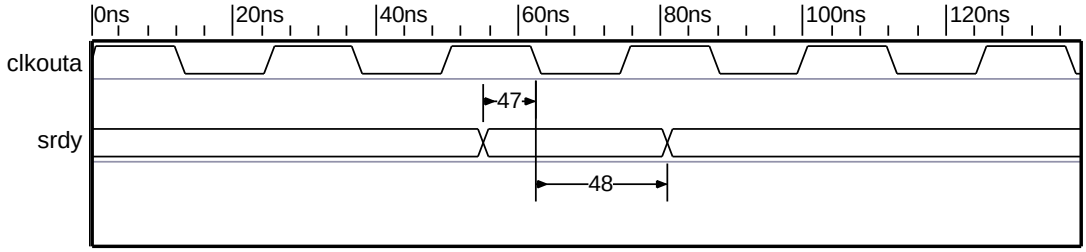


Figure 23. srdy—Synchronous Ready

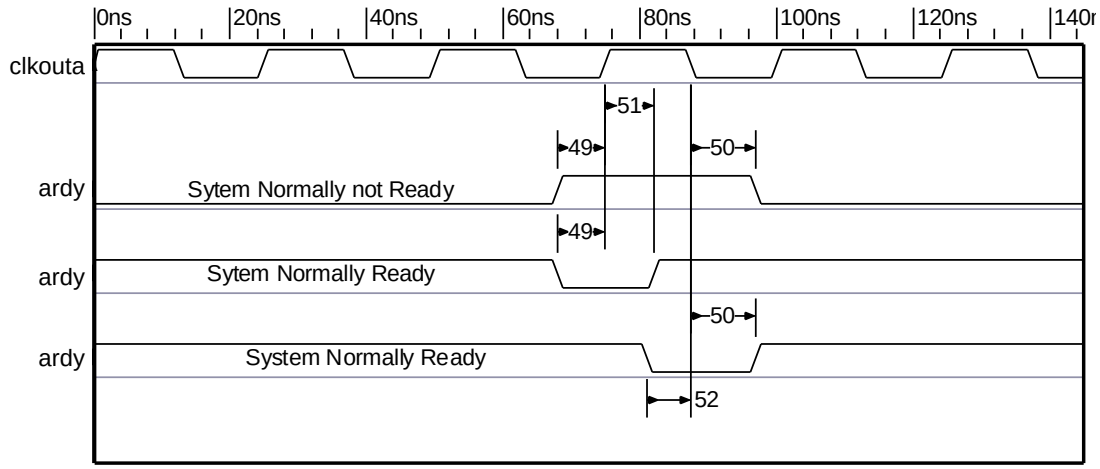


Figure 24. ardy—Asynchronous Ready

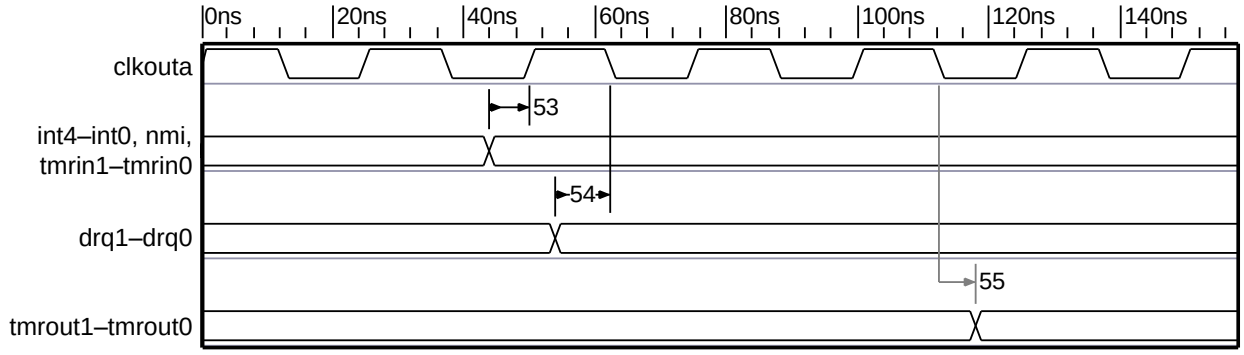


Figure 25. Peripherals

Table 87. Ready and Peripheral Timing

No.	Name	Description	Min[a]	Max[a]
Ready and Peripheral Timing Requirements				
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	–
52	tARYLCL	ardy Setup Time	9	–
53	tINVCH	Peripheral Setup Time	10	–
54	tINVCL	drq Setup Time	10	–
Peripheral Timing Responses				
55	tCLTMV	Timer Output Delay	0	12

[a]In nanoseconds.

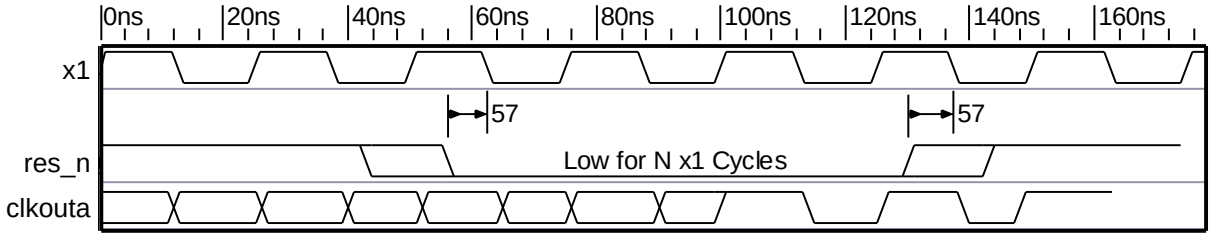


Figure 26. Reset 1

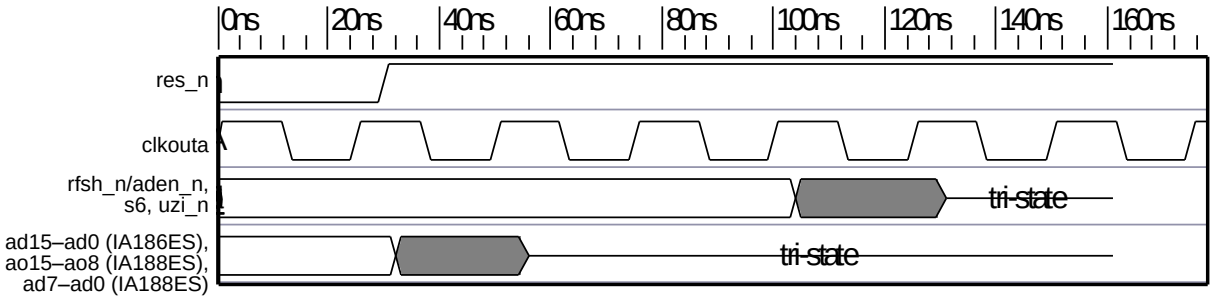


Figure 27. Reset 2

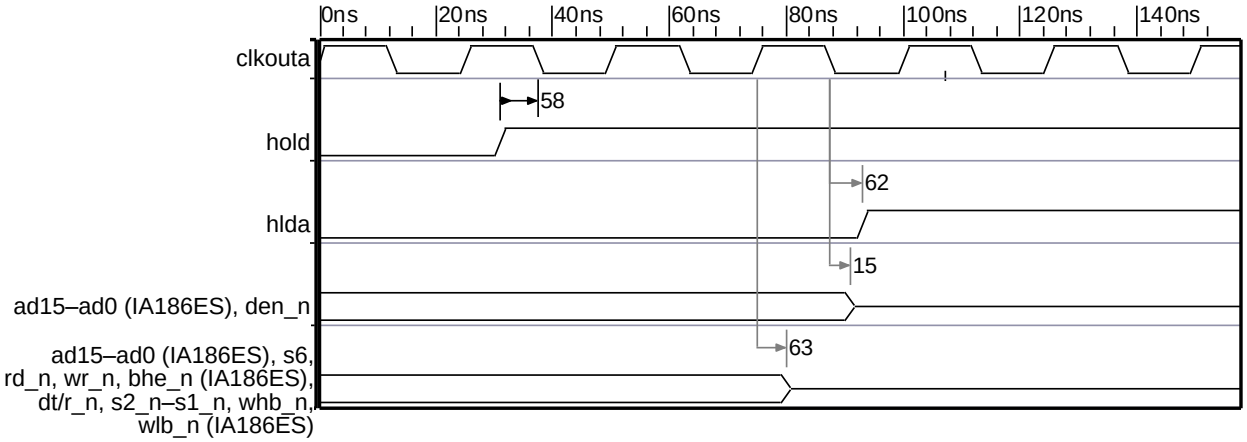


Figure 28. Bus Hold Entering

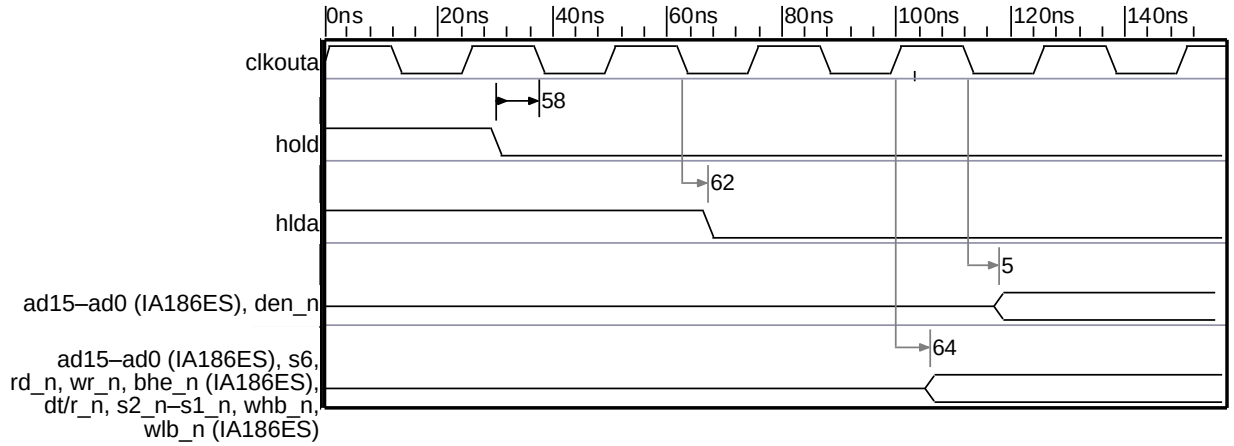


Figure 29. Bus Hold Leaving

Table 88. 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	0	12
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.

7. Instruction Set Summary Table

Table 89 summarizes each instruction. A key to abbreviations is presented at the end of the table.

Table 89. Instruction Set Summary

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
AAA	ASCII adjust AL after add	37	–	–	8	8	U	–	–	–	U	U	R	U	R
AAD	ASCII adjust AX before divide	D5	0A	–	15	15	U	–	–	–	R	R	U	R	U
AAM	ASCII adjust AL after multiply	D4	0A	–	19	19	U	–	–	–	R	R	U	R	U
AAS	ASCII adjust AL after subtract	3F	–	–	7	7	U	–	–	–	U	U	R	U	R
ADC	Add imm8 to AL with carry	14	ib	–	3	3	R	–	–	–	R	R	R	R	R
	Add imm16 to AX with carry	15	iw	–	4	4									
	Add imm8 to r/m8 with carry	80	/2 ib	–	4/16	4/16									
	Add imm16 to r/m16 with carry	81	/2 iw	–	4/16	4/20									
	Add sign extended imm8 to r/m16 with carry	83	/2 ib	–	4/16	4/20									
	Add byte reg to r/m8 with carry	10	/r	–	3/10	3/10									
	Add word reg to r/m16 with carry	11	/r	–	3/10	3/14									
	Add r/m8 to byte reg with carry	12	/r	–	3/10	3/10									
	Add r/m16 to word reg with carry	13	/r	–	3/10	3/14									
ADD	Add imm8 to AL	04	ib	–	3	3	R	–	–	–	R	R	R	R	R
	Add imm16 to AX	05	iw	–	4	4									
	Add imm8 to r/m8	80	/0 ib	–	4/16	4/16									
	Add imm16 to r/m16	81	/0 iw	–	4/16	4/20									
	Add sign extended imm8 to r/m16	83	/0 ib	–	4/16	4/20									
	Add byte reg to r/m8	00	/r	–	3/10	3/10									
	Add word reg to r/m16	01	/r	–	3/10	3/14									
	Add r/m8 to byte reg	02	/r	–	3/10	3/10									
	Add r/m16 to word reg	03	/r	–	3/10	3/14									
AND	And imm8 with AL	24	ib	–	3	3	0	–	–	–	R	R	U	R	0
	And imm16 with AX	25	iw	–	4	4									
	And imm8 with r/m8	80	/4 ib	–	4/16	4/16									
	And imm16 with r/m16	81	/4 iw	–	4/16	4/20									
	And sign-extended imm8 with r/m16	83	/4 ib	–	4/16	4/20									
	And byte reg with r/m8	20	/r	–	3/10	3/10									
	And word reg with r/m16	21	/r	–	3/10	3/14									
	And r/m8 with byte reg	22	/r	–	3/10	3/10									
	And r/m16 with word reg	23	/r	–	3/10	3/14									
BOUND	Check array index against bounds	62	/r	–	33–35	33–35	–	–	–	–	–	–	–	–	–
CALL	Call near, disp relative to next instruction	E8	cw	–	15	19	–	–	–	–	–	–	–	–	–
	Call near, reg indirect mem	FF	/2	–	13/19	17/27									
	Call far to full address given	9A	cd	–	23	31									
	Call far to address at m16:16 word	FF	/3	–	38	54									
CBW	Convert byte integer to word	98	–	–	2	2	–	–	–	–	–	–	–	–	–
CLC	Clear carry flag	F8	–	–	2	2	–	–	–	–	–	–	–	–	–
CLD	Clear direction flag	FC	–	–	2	2	–	0	–	–	–	–	–	–	–
CLI	Clear interrupt-enable flag	FA	–	–	2	2	–	–	0	–	–	–	–	–	–
CMC	Complement carry flag	F5	–	–	2	2	–	–	–	–	–	–	–	–	R

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

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
CMP	Compare imm8 to AL	3C	ib	–	3	3	R	–	–	–	R	R	R	R	R
	Compare imm16 to AX	3D	iw	–	4	4									
	Compare imm8 to r/m8	80	/7	ib	3/10	3/10									
	Compare imm16 to r/m16	81	/7	iw	3/10	3/14									
	Compare sign-extended imm8 to r/m16	83	/7	ib	3/10	3/14									
	Compare byte reg to r/m8	38	/r	–	3/10	3/10									
	Compare word reg to r/m16	39	/r	–	3/10	3/14									
	Compare r/m8 to byte reg	3A	/r	–	3/10	3/10									
	Compare r/m16 to word reg	3B	/r	–	3/10	3/14									
CMPS	Compare byte ES:[DI] to byte segment:[SI]	A6	–	–	22	22	R	–	–	–	R	R	R	R	R
	Compare word ES:[DI] to word segment:[SI]	A7	–	–	22	26									
CMPSB	Compare byte ES:[DI] to byte DS:[SI]	A6	–	–	22	22	R	–	–	–	R	R	R	R	R
CMPSW	Compare word ES:[DI] to word DS:[SI]	A7	–	–	22	26	R	–	–	–	R	R	R	R	R
CS	CS segment reg override prefix	2E	–	–	–	–	–	–	–	–	–	–	–	–	–
CWD	Convert word integer to double word	99	–	–	4	4	–	–	–	–	–	–	–	–	–
DAA	Decimal adjust AL after addition	27	–	–	4	4	?	–	–	–	R	R	R	R	R
DAS	Decimal adjust AL after subtraction	2F	–	–	4	4	?	–	–	–	R	R	R	R	R
DEC	Subtract 1 from r/m8	FE	/1	–	3/15	3/15	R	–	–	–	R	R	R	R	R
	Subtract 1 from r/m16	FF	/1	–	3/15	3/19									
	Subtract 1 from word reg	48+ rw			3	3									
DIV	Divide unsigned numbers	F6	mod 110 r/m	–	29/35	29/35	U	–	–	–	U	U	U	U	U
DS	DS segment override prefix	3E	–	–	–	–	–	–	–	–	–	–	–	–	–
ENTER	Create stack frame for nested procedure	C8	iw ib	–	22+16 (n–1)	26+20 (n–1)	–	–	–	–	–	–	–	–	–
	Create stack frame for non-nested procedure	C8	iw 00	–	15	19									
	Create stack frame for nested procedure	C8	iw 01	–	25	29									
ES	ES segment reg override prefix	26	–	–	–	–	–	–	–	–	–	–	–	–	–
ESC	Escape - takes a Trap 7	D8	/0	–	–	–	–	–	0	0	–	–	–	–	–
	Escape - takes a Trap 7	D9	/1	–	–	–									
	Escape - takes a Trap 7	DA	/2	–	–	–									
	Escape - takes a Trap 7	DB	/3	–	–	–									
	Escape - takes a Trap 7	DC	/4	–	–	–									
	Escape - takes a Trap 7	DD	/5	–	–	–									
	Escape - takes a Trap 7	DE	/6	–	–	–									
	Escape - takes a Trap 7	DF	/7	–	–	–									
HLT	Suspend instruction execution	F4	–	–	2	2	–	–	–	–	–	–	–	–	–
IDIV	Divide Integers AL = AX/(r/m8); AH = remainder	F6	/7	–	44–52 / 50–58	44–52 / 50–58	U	–	–	–	U	U	U	U	U
	Divide Integers AX = DX:AX/(r/m16); DX = remainder	F7	/7	–	53–61 / 59–67	53–61 / 63–71									

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

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
IMUL	Multiply Integers AX=(r/m8)*Al	F6	/5	–	25–28 / 31–34	25–28 / 31–34	R	–	–	–	U	U	U	U	R
	Multiply Integers DX=(r/m16)*AX	F7	/5	–	34–37 / 40–43	34–37 / 44–47									
	Multiply Integers (word reg) = (r/m16)*(sign-ext. byte integer)	6B	/r ib	–	22–25	22–25									
	Multiply Integers (word reg) = (word reg)*(sign-ext. byte integer)	6B	/r ib	–	22–25	22–25									
	Multiply Integers (word reg) = (r/m16)*(sign-ext. byte integer)	69	/r iw	–	29–32	29–32									
	Multiply Integers (word reg) = (word reg)*(sign-ext. byte integer)	69	/r iw	–	29–32	29–32									
IN	Input byte from imm port to AL	E4	ib	–	10	10	–	–	–	–	–	–	–	–	–
	Input word from imm port to AX	E5	ib	–	10	14									
	Input byte from port in DX to AL	EC	–	–	8	8									
	Input word from port in DX to AX	ED	–	–	8	12									
INC	Increment r/m8 by 1	FE	/0	–	3/15	3/15	R	–	–	–	R	R	R	R	R
	Increment r/m16 by 1	FF	/0	–	3/15	3/19									
	Increment word reg by 1	40+rw	–	–	3	3									
INS	Input byte from port in DX to ES:[DI]	6C	–	–	14	14	–	–	–	–	–	–	–	–	–
	Input word from port in DX to ES:[DI]	6D													
INSB	Input byte from port in DX to ES:[DI]	6C													
INSW	Input word from port in DX to ES:[DI]	6D													
INT 3	Generate interrupt 3 (trap to debug)	CC	–	–	45	45	–	–	0	0	–	–	–	–	–
INT	Generate type of interrupt specified by imm8	CD	ib	–	47	47									
INTO	Generate interrupt 4 if Overflow Flag (O) is 1	CE	–	–	48, 4	48, 4									
IRET	Interrupt return	CF	–	–	28	28	Restores value of flags reg that was stored on the stack when the interrupt was taken								
JA	Jump short if above (C & Z = 0)	77	cb	–	13, 4	13, 4	–	–	–	–	–	–	–	–	–
JNBE	Jump short if not below or equal														
JAE	Jump short if above or equal (C=0)	73	cb	–	13, 4	13, 4	–	–	–	–	–	–	–	–	–
JNB	Jump short if not below (C=0)														
JNC	Jump short if not carry (C=0)														
JB	Jump short if below (C=1)	72	cb	–	13, 4	13, 4	–	–	–	–	–	–	–	–	–
JC	Jump short if carry (C=1)														
JNAE	Jump short if not above or equal (C=1)														
JBE	Jump short if below or equal (C & Z = 0)	76	cb	–	13, 4	13, 4	–	–	–	–	–	–	–	–	–
JNA	Jump short if not above (C & Z = 0)														

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

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
JCXZ	Jump short if CX reg is 0	E3	cb	–	15,5	15,5	–	–	–	–	–	–	–	–	–
JE	Jump short if equal (Z=1)	74	cb	–	13, 4	13, 4	–	–	–	–	–	–	–	–	–
JZ	Jump short if 0 (Z=1)														
JG	Jump short if greater (Z & S = O)	7F	cb	–	13, 4	13, 4	–	–	–	–	–	–	–	–	–
JNLE	Jump short if not less or equal (Z & S = O)														
JGE	Jump short if greater or equal (S=O)	7D	cb	–	13, 4	13, 4	–	–	–	–	–	–	–	–	–
JNL	Jump short if not less (S = O)														
JLE	Jump short if less or equal (Z & S = O)	7E	cb	–	13, 4	13, 4	–	–	–	–	–	–	–	–	–
JNG	Jump short if not greater (Z & S = O)														
JMP	Jump short direct, disp relative to next instruction	EB	cb	–	14	14	–	–	–	–	–	–	–	–	–
	Jump near direct, disp relative to next instruction	E9	cw	–	14	14									
	Jump near indirect	FF	/4	–	11/17	11/21									
	Jump far direct to doubleword imm address	EA	cd	–	14	14									
	Jump m16:16 indirect and far	FF	/5	–	26	34									
JNE	Jump short if not equal (Z=0)	75	cb	–	13, 4	13, 4	–	–	–	–	–	–	–	–	–
JNZ	Jump short if not zero (Z=0)														
JNO	Jump short if not overflow (O=1)	71	cb	–	13, 4	13, 4	–	–	–	–	–	–	–	–	–
JNP	Jump short if not parity (P=0)	7B	cb	–	13, 4	13, 4	–	–	–	–	–	–	–	–	–
JPO	Jump short if parity odd (P=0)														
JNS	Jump short if not sign (S=0)	79	cb	–	13, 4	13, 4	–	–	–	–	–	–	–	–	–
JO	Jump short if overflow (O=1)	70	cb	–	13, 4	13, 4	–	–	–	–	–	–	–	–	–
JP	Jump short if parity (P=1)	7A	cb	–	13, 4	13, 4	–	–	–	–	–	–	–	–	–
JPE	Jump short if parity (P=1)														
JS	Jump short if sign (S=1)	78	cb	–	13, 4	13, 4	–	–	–	–	–	–	–	–	–
LAHF	Load AH with low byte of flags reg	9F	–	–	2	2	–	–	–	–	–	–	–	–	–
LDS	Load DS:r16 with segment:offset from memory	C5	/r	–	18	26	–	–	–	–	–	–	–	–	–
LEA	Load offset for m16 word in 16-bit reg	8D	/r	–	6	6	–	–	–	–	–	–	–	–	–
LEAVE	Destroy procedure stack frame	C9	–	–	8	8	–	–	–	–	–	–	–	–	–
LES	Load ES:r16 with segment offset from memory	C4	/r	–	18	26	–	–	–	–	–	–	–	–	–
LOCK	Asserts lock_n during an instruction execution	F0	–	–	1	1	–	–	–	–	–	–	–	–	–
LODS	Load byte segment:[SI] in AL	AC	–	–	12	12	–	–	–	–	–	–	–	–	–
	Load word segment:[SI] in AX	AD			12	16									
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														

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

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
MOV	Copy reg to r/m8	88	/r	–	2/12	2/12	–	–	–	–	–	–	–	–	–
	Copy reg to r/m16	89	/r	–	2/12	2/16									
	Copy r/m8 to reg	8A	/r	–	2/9	2/9									
	Copy r/m16 to reg	8B	/r	–	2/9	2/13									
	Copy segment reg to r/m16	8C	/sr	–	2/11	2/15									
	Copy r/m16 to segment reg	8E	/sr	–	2/9	2/13									
	Copy byte at segment offset to AL	A0	–	–	8	8									
	Copy word at segment offset to AX	A1	–	–	8	12									
	Copy AL to byte at segment offset	A2	–	–	9	9									
	Copy AX to word at segment offset	A3	–	–	9	13									
	Copy imm8 to reg	B0+rb	–	–	3	3									
	Copy imm16 to reg	B8+rw	–	–	3	4									
	Copy imm8 to r/m8	C6	/0	–	12	12									
	Copy imm16 to r/m16	C7	/0	–	12	13									
MOVS	Copy byte segment [SI] to ES:[DI]	A4	–	–	14	14	–	–	–	–	–	–	–	–	–
	Copy word segment [SI] to ES:[DI]	A5	–	–	14	18									
MOVSB	Copy byte DS:[SI] to ES:[DI]	A4	–	–	14	14									
MOVSW	Copy word DS:[SI] to ES:[DI]	A5	–	–	14	18									
MUL	AX = (r/m8) * AL	F6	/4	–	26–28 / 32–34	26–28 / 32–34	R	–	–	–	–	–	–	–	R
	DX::AX = (r/m16) * AX	F7	/4		35–37 / 41–43	35–37 / 45–47									
NEG	Perform 2's-complement negation of r/m8	F6	/3	–	3/10	3/10	R	–	–	–	R	R	R	R	R
	Perform 2's-complement negation of r/m16	F7	/3	–	3/10	3/14									
NOP	Perform no operation	90	–	–	3	3	–	–	–	–	–	–	–	–	–
NOT	Complement each bit in r/m8	F6	/2	–	3/10	3/10	–	–	–	–	–	–	–	–	–
	Complement each bit in r/m16	F7	/2		3/10	3/14									
OR	OR imm8 with AL	0C	ib	–	3	3	0	–	–	–	R	R	U	R	0
	OR imm16 with AX	0D	iw	–	4	4									
	OR imm8 with r/m8	80	/1 ib	–	4/16	4/16									
	OR imm16 with r/m16	81	/1 iw	–	4/16	4/20									
	OR imm8 with r/m16	83	/1 ib	–	4/16	4/20									
	OR byte reg with r/m8	08	/r	–	3/10	3/10									
	OR word reg with r/m16	09	/r	–	3/10	3/14									
	OR r/m8 with byte reg	0A	/r	–	3/10	3/10									
	OR r/m16 with word reg	0B	/r	–	3/10	3/14									
OUT	Output AL to imm port	E6	ib	–	9	9	–	–	–	–	–	–	–	–	–
	Output AX to imm port	E7	ib	–	9	13									
	Output AL to port in DX	EE	–	–	7	7									
	Output AX to port in DX	EF	–	–	7	11									
OUTS	Output byte DS:[SI] to port in DX	6E	–	–	14	14	–	–	–	–	–	–	–	–	–
	Output word DS:[SI] to port in DX	6F	–	–											
OUTSB	Output byte DS:[SI] to port in DX	6E	–	–											
OUTSW	Output word DS:[SI] to port in DX	6F	–	–											

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

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.

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
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									
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									

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

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
ROR	Rotate 8 bits of r/m8 right once	D0	/1	–	2/15	2/15	U	–	–	–	–	–	–	–	R
	Rotate 8 bits or r/m8 right CL times	D2	/1	–	5+n/ 17+n	5+n/ 17+n									
	Rotate 8 bits or r/m8 right imm8 times	C0	/1 ib	data 8	5+n/ 17+n	5+n/ 17+n									
	Rotate 16 bits of r/m8 right once	D1	/1	–	2/15	2/15									
	Rotate 16 bits or r/m8 right CL times	D3	/1	–	5+n/ 17+n	5+n/ 17+n									
	Rotate 16 bits or r/m8 right imm8 times	C1	/1 ib	data 8	5+n/ 17+n	5+n/ 17+n									
SAHF	Show AH in low byte of the Status Flags reg	9E	–	–	3	3	–	–	–	–	R	R	R	R	R
SAL/SHL	Multiply r/m8 by 2, once	D0	/4	–	2/15	2/15	U	–	–	–	–	R	R	R	R
	Multiply r/m8 by 2, CL times	D2	/4	–	5+n/ 17+n	5+n/ 17+n									
	Multiply r/m8 by 2, imm8 times	C0	/4 ib	data 8	5+n/ 17+n	5+n/ 17+n									
	Multiply r/m16 by 2, once	D1	/4	–	2/15	2/15									
	Multiply r/m16 by 2, CL times	D3	/4	–	5+n/ 17+n	5+n/ 17+n									
	Multiply r/m16 by 2, imm8 times	C1	/4 ib	data 8	5+n/ 17+n	5+n/ 17+n									
	Multiply r/m8 by 2, once	D0	/4	–	2/15	2/15									
	Multiply r/m8 by 2, CL times	D2	/4	–	5+n/ 17+n	5+n/ 17+n									
	Multiply r/m8 by 2, imm8 times	C0	/4 ib	data 8	5+n/ 17+n	5+n/ 17+n									
	Multiply r/m16 by 2, once	D1	/4	–	2/15	2/15									
	Multiply r/m16 by 2, CL times	D3	/4	–	5+n/ 17+n	5+n/ 17+n									
	Multiply r/m16 by 2, imm8 times	C1	/4 ib	data 8	5+n/ 17+n	5+n/ 17+n									
SAR	Perform a signed division of r/m8 by 2, once	D0	/7	–	2/15	2/15	U	–	–	–	R	R	U	R	R
	Perform a signed division of r/m8 by 2, CL times	D2	/7	–	5+n/ 17+n	5+n/ 17+n									
	Perform a signed division of r/m8 by 2, imm8 times	C0	/7 ib	data 8	5+n/ 17+n	5+n/ 17+n									
	Perform a signed division of r/m16 by 2, once	D1	/7	–	2/15	2/15									
	Perform a signed division of r/m16 by 2, CI times	D3	/7	–	5+n/ 17+n	5+n/ 17+n									
	Perform a signed division of r/m16 by 2, imm8 times	C1	/7 ib	data 8	5+n/ 17+n	5+n/ 17+n									

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

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
SBB	Subtract imm8 from Al with borrow	1C	ib	–	3	3	R	–	–	–	R	R	R	R	R
	Subtract imm16 from AX with borrow	1D	iw	data 8	4	4									
	Subtract imm8 from r/m8 with borrow	80	/3 ib	–	4/16	4/16									
	Subtract imm16 from r/m16 with borrow	81	/3 iw	–	4/16	4/20									
	Subtract sign-extended imm8 from r/m16 with borrow	83	/3 ib	–	4/16	4/20									
	Subtract byte reg from r/m8 with borrow	18	/r	data 8	3/10	3/10									
	Subtract word reg from r/m16 with borrow	19	/r	–	3/10	3/14									
	Subtract r/m8 from r/m8 with borrow	1A	/r	–	3/10	3/10									
	Subtract r/m8 reg from byte with borrow	1B	/r	data 8	3/10	3/14									
SCAS	Compare byte AL to ES:[DI]; update DI	AE	–	–	15	19	R	–	–	–	R	R	R	R	R
	Compare word AL to ES:[DI]; update DI	AF	–	–	15	19									
SCASB	Compare byte AL to ES:[DI]; update DI	AE	–	–	15	19									
SCASW	Compare word AL to ES:[DI]; update DI	AF	–	–	15	19									
SHR	Divide unsigned of r/m8 by 2, once	D0	/7	–	2/15	2/15	U	–	–	–	R	R	U	R	0
	Divide unsigned of r/m8 by 2, CL times	D2	/7	–	5+n/ 17+n	5+n/ 17+n									
	Divide unsigned of r/m8 by 2, imm8 times	C0	/7 ib	data 8	5+n/ 17+n	5+n/ 17+n									
	Divide unsigned of r/m16 by 2, once	D1	/7	–	2/15	2/15									
	Divide unsigned of r/m16 by 2, CL times	D3	/7	–	5+n/ 17+n	5+n/ 17+n									
	Divide unsigned of r/m16 by 2, imm8 times	C1	/7 ib	data 8	5+n/ 17+n	5+n/ 17+n									
SS	SS segment reg override prefix	36	–	–	–	–	–	–	–	–	–	–	–	–	–
STC	Set the Carry Flag to 1	F9	–	–	2	2	–	–	–	–	–	–	–	–	1
STD	Set the Direction Flag so the source Index (SI) and/or the Destination Index (DI) regs will decrement during string instructions	FD	–	–	2	2	–	1	–	–	–	–	–	–	–
STI	Enable maskable interrupts after the next instruction	FB	–	–	2	2	–	–	1	–	–	–	–	–	–
STOS	Store AL in byte ES:[DI]; update DI	AA	–	–	10	10	–	–	–	–	–	–	–	–	–
	Store AX in word ES:[DI]; update DI	AB	–	–	10	14									
STOSB	Store AL in byte ES:[DI]; update DI	AA	–	–	10	10									
STOSW	Store AX in word ES:[DI]; update DI	AB	–	–	10	14									

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

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
SUB	Subtract imm8 from AL	2C	ib	–	3	3	R	–	–	–	R	R	R	R	R
	Subtract imm16 from AX	2D	iw	–	4	4									
	Subtract imm8 from r/m8	80	/5 ib	–	4/16	4/16									
	Subtract imm16 from r/m16	81	/5 iw	–	4/16	4/20									
	Subtract sign-extended imm8 from r/m16	83	/5 ib	–	4/16	4/20									
	Subtract byte reg from r/m8	28	/r	–	3/10	3/10									
	Subtract word reg from r/m16	29	/r	–	3/10	3/14									
	Subtract r/m8 from byte reg	2A	/r	–	3/10	3/10									
	Subtract r/m16 from word reg	2B	/r	–	3/10	3/14									
TEST	AND imm8 with AL	A8	ib	–	3	3	0	–	–	–	R	R	U	R	0
	AND imm16 with AX	A9	iw	–	4	4									
	AND imm8 with r/m8	F6	/0 ib	data 8	4/10	4/10									
	AND imm16 with r/m16	F7	/0 iw	–	4/10	4/14									
	AND byte reg with r/m8	84	/r	–	3/10	3/10									
	AND word reg with r/m16	85	/r	data 8	3/10	3/14									
WAIT	Performs a NOP	9B	–	–	–	–	–	–	–	–	–	–	–	–	–
XCHG	Exchange word reg with AX	90 +rw	–	–	3	3	–	–	–	–	–	–	–	–	–
	Exchange AX with word reg		–	–	3	3									
	Exchange byte reg with r/byte	86 /r	–	–	4/17	4/17									
	Exchange r/m8 with byte reg		–	–	4/17	4/17									
	Exchange word reg with r/m16	87 /r	–	–	4/17	4/21									
	Exchange r/m16 with word reg		–	–	4/17	4/21									
XLAT	Set AL to memory byte segment:[BX+unsigned AL]	D7	–	–	11	15	–	–	–	–	–	–	–	–	–
XLATB	Set AL to memory byte DS:[BX+unsigned AL]	D7	–	–	11	15									
XOR	XOR imm8 with AL	34	ib	–	3	3	0	–	–	–	R	R	U	R	0
	XOR imm16 with AX	35	iw	–	4	4									
	XOR imm8 with r/m8	80	/6 ib	–	4/16	4/16									
	XOR imm16 with r/m16	81	/6 iw	–	4/16	4/20									
	XOR sign-extended imm8 with r/m16	83	/6 ib	–	4/16	4/20									
	XOR byte reg with r/m8	30	/r	–	3/10	3/10									
	XOR word reg with r/m16	31	/r	–	3/10	3/14									
	XOR r/m8 with byte reg	32	/r	–	3/10	3/10									
	XOR r/m16 with word reg	33	/r	–	3/10	3/14									

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.

7.1.4 r/m Field

The r/m field is defined below.

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

7.1.5 Displacement

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

7.1.6 Immediate Bytes

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

7.1.7 Segment Override Prefix

The segment override prefix is configured as shown below.

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

7.1.8 Segment Register

The segment register is shown below.

SR	Segment Register
00	ES
01	CS
10	SS
11	DS

7.2 Explanation of Notation Used in Instruction Set Summary Table

Notation used in the Instruction Set Summary Table is explained below.

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

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

7.2.1 Opcode

Opcode parameters and definitions are provided below.

Parameter	Definition
/0 - /7	The Auxiliary Field in the Operand Address byte specifies an extension (from 000 to 111, i.e., 0 to 7) to the opcode instead of a register. Thus, the opcode for adding (AND) an immediate byte to a general byte register or a byte in memory is “80 /4 ib.” This indicates that the second byte of the opcode is “mod 100 r/m.”
/r	The Auxiliary Field in the Operand Address byte specifies a register rather that an opcode extension. The opcode byte specifies which register, either byte size or word size, is assigned as in the aux code above.
/sr	This byte is placed before the instruction as shown in Section 7.1.7, Segment Override Prefix.
cb	The byte following the Opcode byte specifies the offset.
cd	The double word following the Opcode byte specifies the offset and is some cases a segment.
ib	Immediate byte—signed or unsigned determined by the Opcode byte.
iw	Immediate word—signed or unsigned determined by the Opcode byte.
rw	Word register operand as determined by the Opcode byte, aux field.

7.2.2 Flags Affected After Instruction

Flags affected after instruction are shown below.

U	Undefined
-	Unchanged
R	Result dependent

8. Innovasic/AMD Part Number Cross-Reference Tables

Tables 90 and 91 show Innovasic part numbers cross-referenced with the corresponding AMD part number.

Table 90. Innovasic/AMD Part Number Cross-Reference for the TQFP

Innovasic Part Number	AMD Part Number	Package Type	Temperature Grades
IA186ES-PTQ100I-R-03 lead free (RoHS-compliant)	AM186ES-20VC\W AM186ES-25VC\W AM186ES-33VC\W AM186ES-40VC\W AM186ES-20VI\W AM186ES-25VI\W AM186ES-33VI\W AM186ES-40VI\W	100-Pin Thin Quad Flat Package (TQFP)	Industrial
IA188ES-PTQ100I-R-03 lead free (RoHS-compliant)	AM188ES-20VC\W AM188ES-25VC\W AM188ES-33VC\W AM188ES-40VC\W AM188ES-20VI\W AM188ES-25VI\W AM188ES-33VI\W AM188ES-40VI\W		

Table 91. Innovasic/AMD Part Number Cross-Reference for the PQFP

Innovasic Part Number	AMD Part Number	Package Type	Temperature Grade
IA186ES-PQF100I-R-03 lead free (RoHS-compliant)	AM186ES-20KC\W AM186ES-25KC\W AM186ES-33KC\W AM186ES-40KC\W AM186ES-20KI\W AM186ES-25KI\W AM186ES-33KI\W AM186ES-40KI\W	100-Pin Plastic Quad Flat Package (PQFP)	Industrial
IA188ES-PQF100I-R-03 lead free (RoHS-compliant)	AM188ES-20KC\W AM188ES-25KC\W AM188ES-33KC\W AM188ES-40KC\W AM188ES-20KI\W AM188ES-25KI\W AM188ES-33KI\W AM188ES-40KI\W		

9. Errata

The following errata are associated with Version 03 of the IA186ES/IA188ES. A workaround to the identified problem has been provided where possible.

9.1 Errata Summary

Table 92 presents a summary of errata.

Table 92. 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 req 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 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.

Errata No. 5

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

Description: When device is in halt state, the interrupt caused by a DMA completion will not bring the CPU out of the halt state.

Workaround: Use idle mode instead of halt.

Errata No. 6

Problem: Does not clear the interrupt req bit for INT0 upon entering the ISR.

Description: The interrupt bit for INT0 is not cleared until the interrupt routine is complete.

Workaround: Do not rely on the bit to be cleared when nesting interrupts.

Errata No. 7

Problem: How hardware handshaking for UARTs during a bus hold cycle is handled differently between the AMD part and the Innovasic part.

Description: In the AMD part, hardware handshaking works per the data sheet. The Innovasic part will occasionally drive a handshake signal to the incorrect state during bus hold instead of tristating the pin.

Workaround: None. Avoid using hardware handshaking in conjunction with the bus hold operation with the Innovasic part UARTs.

10. Revision History

Table 93 presents the sequence of revisions to document IA211050902.

Table 93. Revision History

Date	Revision	Description	Page(s)
August 17, 2007	11	Edition released.	NA
January 31, 2008	12	Errata 6 and 7 added.	136
February 19, 2008	13	Errata 7 clarified.	136
August 7, 2008	14	Column 2 of Peripheral Control Registers table changed from “Serial Port 0” to “Serial Port 1” in 6 bottom rows.	9
December 24, 2008	15	Document reformatted and elements added to meet publication standards. Improved figures and tables. Added Conventions, Acronyms and Abbreviations, and Summary of Errata table.	All
January 25, 2010	16	Corrected PQFP Package Dimensions table.	30
February 25, 2011	17	Updated section 2.2.50 to clarify that power rating is ±10%; Removed packaging options to support the elimination of SnPb lead plating options.	44, 148, 149
July 22, 2011	18	Corrected pin names for various pins.	17-29
November 15, 2011	19	Corrected pin names for pins 42, 43	29

11. For Additional Information

The IA186ES/IA188ES is a form, fit, and function replacement for the original AMD Am186ES/188ES family of microcontrollers. Innovasic produces replacement ICs using its MILES system cloning technology that produces replacement ICs far more complex than "emulation" while ensuring they are compatible with the original IC. MILES captures the design of a clone so it can be produced even as silicon technology advances. MILES also verifies the clone against the original IC so that even the "undocumented features" are duplicated.

The IA186ES/IA188ES family of microcontrollers replaces obsolete AMD Am186ES/188ES devices, allowing customers to retain existing board designs, software compilers/assemblers, and emulation tools and thus avoid expensive redesign efforts.

The Innovasic Support Team wants its information to be complete, accurate, useful, and easy to understand. Please feel free to contact experts at Innovasic with suggestions, comments, or questions at any time.

Innovasic Support Team
3737 Princeton NE
Suite 130
Albuquerque, NM 87107

Phone: +1 505-883-5263
Fax: +1 505-883-5477
Toll Free: (888) 824-4184
Website: http://www.innovasic.com