PIC24FJ256DA210 Family Data Sheet

64/100-Pin, 16-Bit Flash Microcontrollers with Graphics Controller and USB On-The-Go (OTG)

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Trademarks

ISBN: 978-1-60932-235-9

Microchip received ISO/TS-16949:2002 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2002

64/100-Pin, 16-Bit Flash Microcontrollers with Graphics Controller and USB On-The-Go (OTG)

Graphics Controller Features:

- Three Graphics Hardware Accelerators to Facilitate Rendering of Block Copying, Text and Unpacking of Compressed Data
- Color Look-up Table (CLUT) with Maximum of 256 Entries
- 1/2/4/8/16 bits-per-pixel (bpp) Color Depth Set at Run Time
- Display Resolution Programmable According to Frame Buffer:
  - Supports direct access to external memory on devices with EPMP
  - Resolution supported is up to 480x272 @ 60 Hz, 16 bpp; 640x480 @ 30 Hz, 16 bpp or 640x480 @ 60 Hz, 8 bpp
- Supports Various Display Interfaces:
  - 4/8/16-bit Monochrome STN
  - 4/8/16-bit Color STN
  - 9/12/18/24-bit Color TFT (18 and 24-bit displays are connected as 16-bit, 5-6-5 RGB color format)

Universal Serial Bus Features:

- USB v2.0 On-The-Go (OTG) Compliant
- Dual Role Capable – Can act as either Host or Peripheral
- Low-Speed (1.5 Mbps) and Full-Speed (12 Mbps) USB Operation in Host mode
- Full-Speed USB Operation in Device mode
- High-Precision PLL for USB
- Supports up to 32 Endpoints (16 bidirectional):
  - USB module can use the internal RAM location from 0x800 to 0xFFFF as USB endpoint buffers
- On-Chip USB Transceiver with Interface for Off-Chip Transceiver
- Supports Control, Interrupt, Isochronous and Bulk Transfers
- On-Chip Pull-up and Pull-Down Resistors

Peripheral Features:

- Enhanced Parallel Master Port/Parallel Slave Port (EPMP/PSP), 100-pin devices only:
  - Direct access from CPU with an Extended Data Space (EDS) interface
  - 4, 8 and 16-bit wide data bus
  - Up to 23 programmable address lines
  - Up to 2 chip select lines
  - Up to 2 Acknowledgement lines (one per chip select)
  - Programmable address/data multiplexing
  - Programmable address and data Wait states
  - Programmable polarity on control signals
- Peripheral Pin Select:
  - Up to 44 available pins (100-pin devices)
- Three 3-Wire/4-Wire SPI modules (supports 4 Frame modes)
- Three I^2C™ modules Supporting Multi-Master/Slave modes and 7-Bit/10-Bit Addressing
- Four UART modules:
  - Supports RS-485, RS-232, LIN/J2602 protocols and IrDA®
- Five 16-Bit Timers/Counters with Programmable Prescaler
- Nine 16-Bit Capture Inputs, each with a Dedicated Time Base
- Nine 16-Bit Compare/PWM Outputs, each with a Dedicated Time Base
- Hardware Real-Time Clock and Calendar (RTCC)
- Enhanced Programmable Cyclic Redundancy Check (CRC) Generator
- Up to 5 External Interrupt Sources

PIC24FJ Device	Pins	Program Memory (bytes)	SRAM (bytes)	Remappable Peripherals					I^2C™	10-Bit A/D (ch)	Comparators	CTMU	EPMP/PSP	RTCC	Graphics Controller	USB OTG
				Remappable Pins	16-Bit Timers	IC/OC PWM	UART w/IrDA®	SPI								
PIC24FJ128DA106	64	128K	24K	29	5	9/9	4	3	3	16	3	Y	N	Y	Y	Y
PIC24FJ256DA106	64	256K	24K	29	5	9/9	4	3	3	16	3	Y	N	Y	Y	Y
PIC24FJ128DA110	100/121	128K	24K	44	5	9/9	4	3	3	24	3	Y	Y	Y	Y	Y
PIC24FJ256DA110	100/121	256K	24K	44	5	9/9	4	3	3	24	3	Y	Y	Y	Y	Y
PIC24FJ128DA206	64	128K	96K	29	5	9/9	4	3	3	16	3	Y	N	Y	Y	Y
PIC24FJ256DA206	64	256K	96K	29	5	9/9	4	3	3	16	3	Y	N	Y	Y	Y
PIC24FJ128DA210	100/121	128K	96K	44	5	9/9	4	3	3	24	3	Y	Y	Y	Y	Y
PIC24FJ256DA210	100/121	256K	96K	44	5	9/9	4	3	3	24	3	Y	Y	Y	Y	Y

High-Performance CPU

- Modified Harvard Architecture
- Up to 16 MIPS Operation at 32 MHz
- 8 MHz Internal Oscillator
- 17-Bit x 17-Bit Single-Cycle Hardware Multiplier
- 32-Bit by 16-Bit Hardware Divider
- 16 x 16-Bit Working Register Array
- C Compiler Optimized Instruction Set Architecture with Flexible Addressing modes
- Linear Program Memory Addressing, up to 12 Mbytes
- Data Memory Addressing, up to 16 Mbytes:
  - 2K SFR space
  - 30K linear data memory
  - 66K extended data memory
  - Remaining (from 16 Mbytes) memory (external) can be accessed using extended data Memory (EDS) and EPMP (EDS is divided into 32-Kbyte pages)
- Two Address Generation Units for Separate Read and Write Addressing of Data Memory

Power Management:

- On-Chip Voltage Regulator of 1.8V
- Switch between Clock Sources in Real Time
- Idle, Sleep and Doze modes with Fast Wake-up and Two-Speed Start-up
- Run Mode: 800 μA/MIPS, 3.3V Typical
- Sleep mode Current Down to 20 μA, 3.3V Typical
- Standby Current with 32 kHz Oscillator: 22 μA, 3.3V Typical

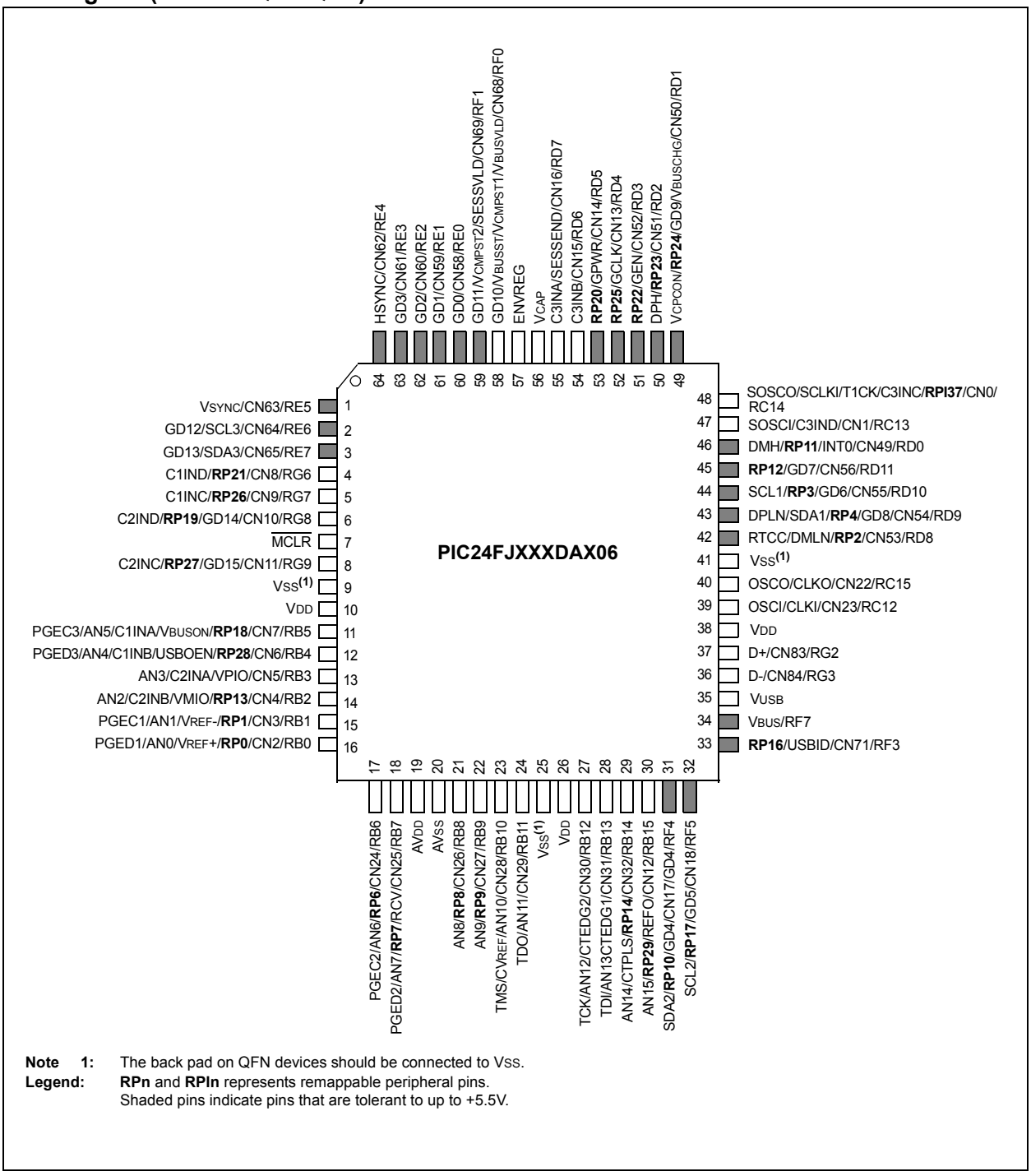
Analog Features:

- 10-Bit, up to 24-Channel Analog-to-Digital (A/D) Converter at 500 ksps:
  - Operation is possible in Sleep mode
  - Band gap reference input feature
- Three Analog Comparators with Programmable Input/Output Configuration
- Charge Time Measurement Unit (CTMU):
  - Supports capacitive touch sensing for touch screens and capacitive switches
  - Minimum time measurement setting at 100 ps
- Available LVD Interrupt V_{LVD} Level

Special Microcontroller Features:

- Operating Voltage Range of 2.2V to 3.6V
- 5.5V Tolerant Input (digital pins only)
- Configurable Open-Drain Outputs on Digital I/O Ports
- High-Current Sink/Source (18 mA/18 mA) on all I/O Ports
- Selectable Power Management modes:
  - Sleep, Idle and Doze modes with fast wake-up
- Fail-Safe Clock Monitor (FSCM) Operation:
  - Detects clock failure and switches to on-chip, FRC oscillator
- On-Chip LDO Regulator
- Power-on Reset (POR) and Oscillator Start-up Timer (OST)
- Brown-out Reset (BOR)
- Flexible Watchdog Timer (WDT) with On-Chip Low-Power RC Oscillator for Reliable Operation
- In-Circuit Serial Programming™ (ICSP™) and In-Circuit Debug (ICD) via 2 Pins
- JTAG Boundary Scan Support
- Flash Program Memory:
  - 10,000 erase/write cycle endurance (minimum)
  - 20-year data retention minimum
  - Selectable write protection boundary
  - Self-reprogrammable under software control
  - Write protection option for Configuration Words

Pin Diagram (64-Pin TQFP/QFN)



Note 1: The back pad on QFN devices should be connected to VSS.

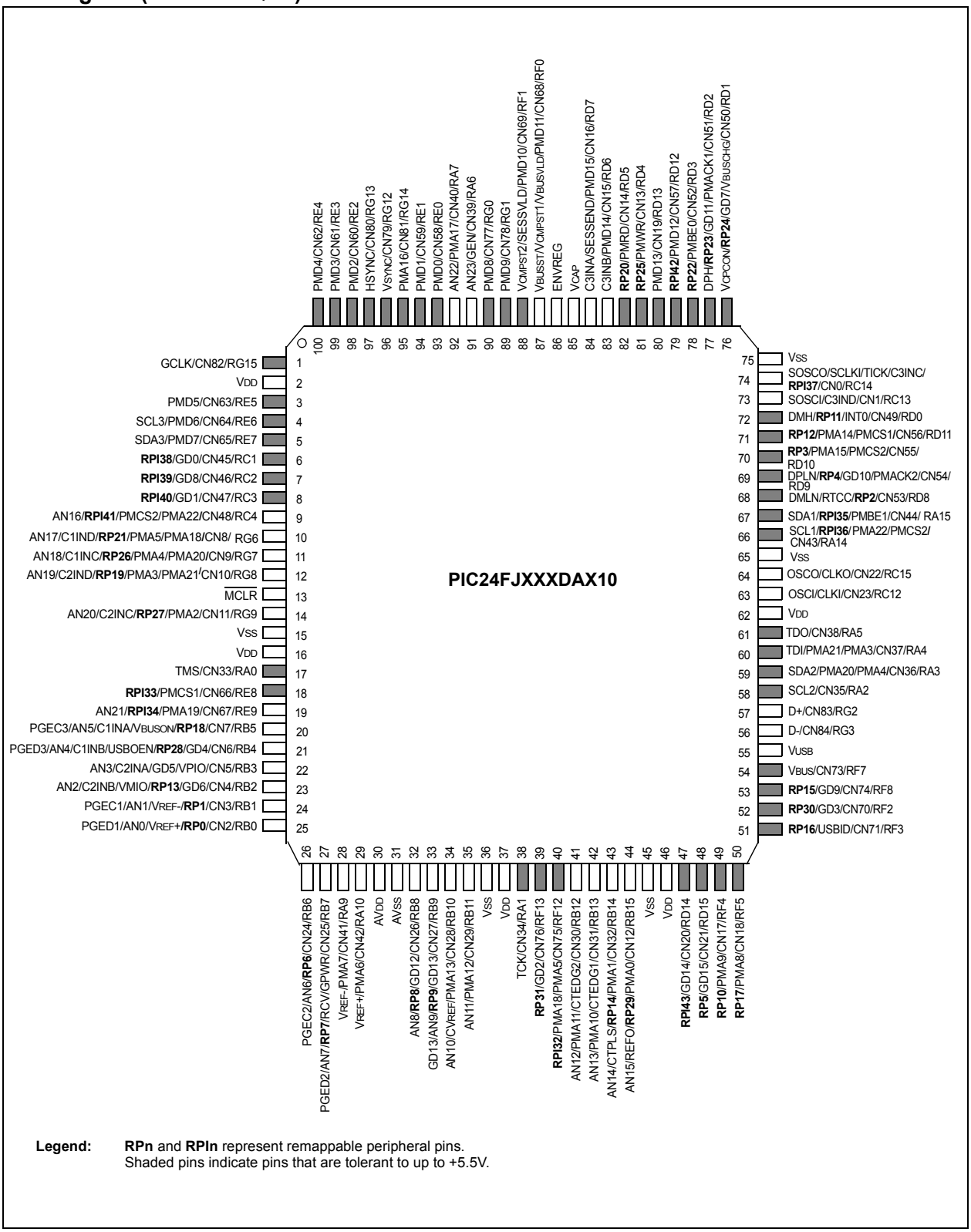
Legend: **RPn** and **RPIn** represents remappable peripheral pins.
Shaded pins indicate pins that are tolerant to up to +5.5V.

TABLE 1: COMPLETE PIN FUNCTION DESCRIPTIONS FOR 64-PIN DEVICES

Pin	Function	Pin	Function
1	VSYNC/CN63/RE5	33	**RP16**/USBID/CN71/RF3
2	GD12/SCL3/CN64/RE6	34	VBUS/RF7
3	GD13/SDA3/CN65/RE7	35	VUSB
4	C1IND/**RP21**/CN8/RG6	36	D-/CN84/RG3
5	C1INC/**RP26**/CN9/RG7	37	D+/CN83/RG2
6	C2IND/**RP19**/GD14/CN10/RG8	38	VDD
7	$\overline{\text{MCLR}}$	39	OSCI/CLKI/CN23/RC12
8	C2INC/**RP27**/GD15/CN11/RG9	40	OSCO/CLKO/CN22/RC15
9	VSS	41	VSS
10	VDD	42	RTCC/DMLN/**RP2**/CN53/RD8
11	PGEC3/AN5/C1INA/VBUSON/**RP18**/CN7/RB5	43	DPLN/SDA1/**RP4**/GD8/CN54/RD9
12	PGED3/AN4/C1INB/USBOEN/**RP28**/CN6/RB4	44	SCL1/**RP3**/GD6/CN55/RD10
13	AN3/C2INA/VPIO/CN5/RB3	45	**RP12**/GD7/CN56/RD11
14	AN2/C2INB/VMIO/**RP13**/CN4/RB2	46	DMH/**RP11**/INT0/CN49/RD0
15	PGEC1/AN1/VREF-/**RP1**/CN3/RB1	47	SOSCI/C3IND/CN1/RC13
16	PGED1/AN0/VREF+/**RP0**/CN2/RB0	48	SOSCO/SCLKI/T1CK/C3INC/**RPI37**/CN0/RC14
17	PGEC2/AN6/**RP6**/CN24/RB6	49	VCPCON/**RP24**/GD9/VBUSCHG/CN50/RD1
18	PGED2/AN7/**RP7**/RCV/CN25/RB7	50	DPH/**RP23**/CN51/RD2
19	AVDD	51	**RP22**/GEN/CN52/RD3
20	AVSS	52	**RP25**/GCLK/CN13/RD4
21	AN8/**RP8**/CN26/RB8	53	**RP20**/GPWR/CN14/RD5
22	AN9/**RP9**/CN27/RB9	54	C3INB/CN15/RD6
23	TMS/CVREF/AN10/CN28/RB10	55	C3INA/SESSEND/CN16/RD7
24	TDO/AN11/CN29/RB11	56	VCAP
25	VSS	57	ENVREG
26	VDD	58	GD10/VBUSST/VCMPST1/VBUSVLD/CN68/RF0
27	TCK/AN12/CTEDG2/CN30/RB12	59	GD11/VCMPST2/SESSVLD/CN69/RF1
28	TDI/AN13/CTEDG1/CN31/RB13	60	GD0/CN58/RE0
29	AN14/CTPLS/**RP14**/CN32/RB14	61	GD1/CN59/RE1
30	AN15/**RP29**/REFO/CN12/RB15	62	GD2/CN60/RE2
31	SDA2/**RP10**/GD4/CN17/RF4	63	GD3/CN61/RE3
32	SCL2/**RP17**/GD5/CN18/RF5	64	HSYNC/CN62/RE4

Legend: **RPn** and **RPIn** represent remappable pins for Peripheral Pin Select functions.

Pin Diagram (100-Pin TQFP)



Legend: **RPn** and **RPIn** represent remappable peripheral pins.
Shaded pins indicate pins that are tolerant to up to +5.5V.

TABLE 2: COMPLETE PIN FUNCTION DESCRIPTIONS FOR 100-PIN DEVICES

Pin	Function	Pin	Function
1	GCLK/CN82/RG15	41	AN12/PMA11/CTEDG2/CN30/RB12
2	VDD	42	AN13/PMA10/CTEDG1/CN31/RB13
3	PMD5/CN63/RE5	43	AN14/CTPLS/**RP14**/PMA1/CN32/RB14
4	SCL3/PMD6/CN64/RE6	44	AN15/REFO/**RP29**/PMA0/CN12/RB15
5	SDA3/PMD7/CN65/RE7	45	VSS
6	**RPI38**/GD0/CN45/RC1	46	VDD
7	**RPI39**/GD8/CN46/RC2	47	**RPI43**/GD14/CN20/RD14
8	**RPI40**/GD1/CN47/RC3	48	**RP5**/GD15/CN21/RD15
9	AN16/**RPI41**/PMCS2/PMA22(2)/CN48/RC4	49	**RP10**/PMA9/CN17/RF4
10	AN17/C1IND/**RP21**/PMA5/PMA18(2)/CN8/RG6	50	**RP17**/PMA8/CN18/RF5
11	AN18/C1INC/**RP26**/PMA4/PMA20(2)/CN9/RG7	51	**RP16**/USBID/CN71/RF3
12	AN19/C2IND/**RP19**/PMA3/PMA21(2)/CN10/RG8	52	**RP30**/GD3/CN70/RF2
13	$\overline{\text{MCLR}}$	53	**RP15**/GD9/CN74/RF8
14	AN20/C2INC/**RP27**/PMA2/CN11/RG9	54	VBUS/CN73/RF7
15	VSS	55	VUSB
16	VDD	56	D-/CN84/RG3
17	TMS/CN33/RA0	57	D+/CN83/RG2
18	**RPI33**/PMCS1/CN66/RE8	58	SCL2/CN35/RA2
19	AN21/**RPI34**/PMA19/CN67/RE9	59	SDA2/PMA20/PMA4(2)/CN36/RA3
20	PGEC3/AN5/C1INA/VBUSON/**RP18**/CN7/RB5	60	TDI/PMA21/PMA3(2)/CN37/RA4
21	PGED3/AN4/C1INB/USBOEN/**RP28**/GD4/CN6/RB4	61	TDO/CN38/RA5
22	AN3/C2INA/GD5/VPIO/CN5/RB3	62	VDD
23	AN2/C2INB/VMIO/**RP13**/GD6/CN4/RB2	63	OSCI/CLKI/CN23/RC12
24	PGEC1/AN1/VREF-(1)/**RP1**/CN3/RB1	64	OSCO/CLKO/CN22/RC15
25	PGED1/AN0/VREF+(1)/**RP0**/CN2/RB0	65	VSS
26	PGEC2/AN6/**RP6**/CN24/RB6	66	SCL1/**RPI36**/PMA22/PMCS2(2)/CN43/RA14
27	PGED2/AN7/**RP7**/RCV/GPWR/CN25/RB7	67	SDA1/**RPI35**/PMBE1/CN44/RA15
28	VREF-/PMA7/CN41/RA9	68	DMLN/RTCC/**RP2**/CN53/RD8
29	VREF+/PMA6/CN42/RA10	69	DPLN/**RP4**/GD10/PMACK2/CN54/RD9
30	AVDD	70	**RP3**/PMA15/PMCS2(3)/CN55/RD10
31	AVSS	71	**RP12**/PMA14/PMCS1(3)/CN56/RD11
32	AN8/**RP8**/GD12/CN26/RB8	72	DMH/**RP11**/INT0/CN49/RD0
33	AN9/**RP9**/GD13/CN27/RB9	73	SOSCI/C3IND/CN1/RC13
34	AN10/CVREF/PMA13/CN28/RB10	74	SOSCO/SCLKI/T1CK/C3INC/**RPI37**/CN0/RC14
35	AN11/PMA12/CN29/RB11	75	VSS
36	VSS	76	VCPCON/**RP24**/GD7/VBUSCHG/CN50/RD1
37	VDD	77	DPH/**RP23**/GD11/PMACK1/CN51/RD2
38	TCK/CN34/RA1	78	**RP22**/PMBE0/CN52/RD3
39	**RP31**/GD2/CN76/RF13	79	**RPI42**/PMD12/CN57/RD12
40	**RPI32**/PMA18/PMA5(2)/CN75/RF12	80	PMD13/CN19/RD13

Legend: **RPn** and **RPIn** represent remappable pins for Peripheral Pin Select (PPS) functions.

Note 1: Alternate pin assignments for VREF+ and VREF- when the $\overline{\text{ALTVREF}}$ Configuration bit is programmed.

2: Alternate pin assignments for EPMP when the $\overline{\text{ALTPMP}}$ Configuration bit is programmed.

3: Pin assignment for PMCSx when CSF<1:0> is not equal to '00'.

TABLE 2: COMPLETE PIN FUNCTION DESCRIPTIONS FOR 100-PIN DEVICES

Pin	Function	Pin	Function
81	**RP25**/PMWR/CN13/RD4	91	AN23/GEN/CN39/RA6
82	**RP20**/PMRD/CN14/RD5	92	AN22/PMA17/CN40/RA7
83	C3INB/PMD14/CN15/RD6	93	PMD0/CN58/RE0
84	C3INA/SESSEND/PMD15/CN16/RD7	94	PMD1/CN59/RE1
85	VCAP	95	PMA16/CN81/RG14
86	ENVREG	96	VSYNC/CN79/RG12
87	VBUSST/VCMPST1/VBUSVLD/PMD11/CN68/RF0	97	HSYNC/CN80/RG13
88	VCMPST2/SESSVLD/PMD10/CN69/RF1	98	PMD2/CN60/RE2
89	PMD9/CN78/RG1	99	PMD3/CN61/RE3
90	PMD8/CN77/RG0	100	PMD4/CN62/RE4

Legend: **RPn** and **RPIn** represent remappable pins for Peripheral Pin Select (PPS) functions.

Note 1: Alternate pin assignments for VREF+ and VREF- when the ALTVREF Configuration bit is programmed.

2: Alternate pin assignments for EPMP when the ALTPMP Configuration bit is programmed.

3: Pin assignment for PMCSx when CSF<1:0> is not equal to '00'.

Pin Diagram – Top View (121-Pin BGA)(1)

	1	2	3	4	5	6	7	8	9	10	11
A	RE4	RE3	HSYNC/ RG13	RE0	RG0	RF1	ENVREG	N/C	RD12	GD11/ RD2	GD7/ RD1
B	N/C	GCLK/ RG15	RE2	RE1	RA7	RF0	VCAP	RD5	RD3	VSS	RC14
C	RE6	VDD	VSYNC/ RG12	RG14	GEN/ RA6	N/C	RD7	RD4	VDD	RC13	RD11
D	GD0/ RC1	RE7	RE5	VSS	VSS	N/C	RD6	RD13	RD0	n/c	RD10
E	RC4	GD1/ RC3	RG6	GD8/ RC2	VDD	RG1	N/C	RA15	RD8	GD10/ RD9	RA14
F	$\overline{\text{MCLR}}$	RG8	RG9	RG7	VSS	n/c	N/C	VDD	OSCI/ RC12	VSS	OSCO/ RC15
G	RE8	RE9	RA0	N/C	VDD	VSS	VSS	N/C	RA5	RA3	RA4
H	PGEC3/ RB5	PGED3/ GD4/RB4	VSS	VDD	N/C	VDD	n/c	VBUS/ RF7	VUSB	D+/RG2	RA2
J	GD5/ RB3	GD6/ RB2	PGED2/RB7 GPWR	AVDD	RB11	RA1	RB12	N/C	N/C	GD9/RF8	D-/RG3
K	PGEC1/ RB1	PGED1/ RB0	RA10	GD12/ RB8	N/C	RF12	RB14	VDD	GD15/ RD15	USBID/ RF3	GD3/ RF2
L	PGEC2/ RB6	RA9	AVSS	GD13/ RB9	RB10	GD2/ RF13	RB13	RB15	GD14/ RD14	RF4	RF5

Note 1: See Table 3 for complete functional pinout descriptions.

Legend: **RPn** and **RPIn** represent remappable pins for Peripheral Pin Select functions.
Shaded pins indicate pins tolerant to up to +5.5V.

TABLE 3: COMPLETE PIN FUNCTION DESCRIPTIONS FOR 121-PIN (BGA) DEVICES

Pin	Function	Pin	Function
A1	PMD4/CN62/RE4	E5	VDD
A2	PMD3/CN61/RE3	E6	PMD9/CN78/RG1
A3	HSYNC/CN80/RG13	E7	N/C
A4	PMD0/CN58/RE0	E8	SDA1/**RPI35**/PMBE1/CN44/RA15
A5	PMD8/CN77/RG0	E9	DMLN/RTCC/**RP2**/CN53/RD8
A6	VCMPST2/SESSVLD/PMD10/CN69/RF1	E10	DPLN/**RP4**/GD10/PMACK2/CN54/RD9
A7	ENVREG	E11	SCL1/**RPI36**/PMA22/PMCS2(2)/CN43/RA14
A8	N/C	F1	$\overline{\text{MCLR}}$
A9	**RPI42**/PMD12/CN57/RD12	F2	AN19/C2IND/**RP19**/PMA3/PMA21(2)/CN10/RG8
A10	DPH/**RP23**/GD11/PMACK1/CN51/RD2	F3	AN20/C2INC/**RP27**/PMA2/CN11/RG9
A11	VCPCON/**RP24**/GD7/VBUSCHG/CN50/RD1	F4	AN18/C1INC/**RP26**/PMA4/PMA20(2)/CN9/RG7
B1	N/C	F5	VSS
B2	GCLK/CN82/RG15	F6	N/C
B3	PMD2/CN60/RE2	F7	N/C
B4	PMD1/CN59/RE1	F8	VDD
B5	AN22/PMA17/CN40/RA7	F9	OSCI/CLKI/CN23/RC12
B6	VBUSST/VCMPST1/VBUSVLD/PMD11/CN68/RF0	F10	VSS
B7	VCAP	F11	OSCO/CLKO/CN22/RC15
B8	**RP20**/PMRD/CN14/RD5	G1	**RPI33**/PMCS1/CN66/RE8
B9	**RP22**/PMBE0/CN52/RD3	G2	AN21/**RPI34**/PMA19/CN67/RE9
B10	VSS	G3	TMS/CN33/RA0
B11	SOSCO/SCLKI/T1CK/C3INC/**RPI37**/CN0/RC14	G4	N/C
C1	SCL3/PMD6/CN64/RE6	G5	VDD
C2	VDD	G6	VSS
C3	VSYNC/CN79/RG12	G7	VSS
C4	PMA16/CN81/RG14	G8	N/C
C5	AN23/GEN/CN39/RA6	G9	TDO/CN38/RA5
C6	N/C	G10	SDA2/PMA20/PMA4(2)/CN36/RA3
C7	C3INA/SESSEND/PMD15/CN16/RD7	G11	TDI/PMA21/PMA3(2)/CN37/RA4
C8	**RP25**/PMWR/CN13/RD4	H1	PGEC3/AN5/C1INA/VBUSON/**RP18**/CN7/RB5
C9	VDD	H2	PGED3/AN4/C1INB/USBOEN/**RP28**/GD4/CN6/RB4
C10	SOSCI/C3IND/CN1/RC13	H3	VSS
C11	**RP12**/PMA14/PMCS1(3)/CN56/RD11	H4	VDD
D1	**RPI38**/GD0/CN45/RC1	H5	N/C
D2	SDA3/PMD7/CN65/RE7	H6	VDD
D3	PMD5/CN63/RE5	H7	N/C
D4	VSS	H8	VBUS/CN73/RF7
D5	VSS	H9	VUSB
D6	N/C	H10	D+/CN83/RG2
D7	C3INB/PMD14/CN15/RD6	H11	SCL2/CN35/RA2
D8	PMD13/CN19/RD13	J1	AN3/C2INA/GD5/VPIO/CN5/RB3
D9	DMH/**RP11**/INT0/CN49/RD0	J2	AN2/C2INB/VMIO/**RP13**/GD6/CN4/RB2
D10	N/C	J3	PGED2/AN7/**RP7**/RCV/GPWR/CN25/RB7
D11	**RP3**/PMA15/PMCS2(3)/CN55/RD10	J4	AVDD
E1	AN16/**RPI41**/PMCS2/PMA22(2)/CN48/RC4	J5	AN11/PMA12/CN29/RB11
E2	**RPI40**/GD1/CN47/RC3	J6	TCK/CN34/RA1
E3	AN17/C1IND/**RP21**/PMA5/PMA18(2)/CN8/RG6	J7	AN12/PMA11/CTEDG2/CN30/RB12
E4	**RPI39**/GD8/CN46/RC2	J8	N/C

Legend: **RPn** and **RPIn** represent remappable pins for Peripheral Pin Select functions.

Note 1: Alternate pin assignments for VREF+ and VREF- when the ALTVREF Configuration bit is programmed.

2: Alternate pin assignments for EPMP when the ALTPMP Configuration bit is programmed.

3: Pin assignment for PMCSx when CSF<1:0> is not equal to '00'.

TABLE 3: COMPLETE PIN FUNCTION DESCRIPTIONS FOR 121-PIN (BGA) DEVICES

Pin	Function	Pin	Function
J9	N/C	L1	PGEC2/AN6/**RP6**/CN24/RB6
J10	**RP15**/GD9/CN74/RF8	L2	VREF-(1)/PMA7/CN41/RA9
J11	D-/CN84/RG3	L3	AVSS
K1	PGEC1/AN1/VREF-(1)/**RP1**/CN3/RB1	L4	AN9/**RP9**/GD13/CN27/RB9
K2	PGED1/AN0/VREF+(1)/**RP0**/CN2/RB0	L5	AN10/CVREF/PMA13/CN28/RB10
K3	VREF+(1)/PMA6/CN42/RA10	L6	**RP31**/GD2/CN76/RF13
K4	AN8/**RP8**/GD12/CN26/RB8	L7	AN13/PMA10/CTEDG1/CN31/RB13
K5	N/C	L8	AN15/REFO/**RP29**/PMA0/CN12/RB15
K6	**RPI32**/PMA18/PMA5(2)/CN75/RF12	L9	**RPI43**/GD14/CN20/RD14
K7	AN14/CTPLS/**RP14**/PMA1/CN32/RB14	L10	**RP10**/PMA9/CN17/RF4
K8	VDD	L11	**RP17**/GD5/PMA8/SCL2/CN18/RF5
K9	**RP5**/GD15/CN21/RD15	—	—
K10	**RP16**/USBID/CN71/RF3	—	—
K11	**RP30**/GD3/CN70/RF2	—	—

Legend: **RPn** and **RPIn** represent remappable pins for Peripheral Pin Select functions.

Note 1: Alternate pin assignments for VREF+ and VREF- when the ALTVREF Configuration bit is programmed.

2: Alternate pin assignments for EPMP when the ALTPMP Configuration bit is programmed.

3: Pin assignment for PMCSx when CSF<1:0> is not equal to '00'.

Table of Contents

1.0 Device Overview 15
2.0 Guidelines for Getting Started with 16-bit Microcontrollers 33
3.0 CPU 39
4.0 Memory Organization 45
5.0 Flash Program Memory 81
6.0 Resets 87
7.0 Interrupt Controller 93
8.0 Oscillator Configuration 141
9.0 Power-Saving Features 155
10.0 I/O Ports 157
11.0 Timer1 189
12.0 Timer2/3 and Timer4/5 191
13.0 Input Capture with Dedicated Timers 197
14.0 Output Compare with Dedicated Timers 201
15.0 Serial Peripheral Interface (SPI) 211
16.0 Inter-Integrated Circuit™ (I^2C™) 223
17.0 Universal Asynchronous Receiver Transmitter (UART) 231
18.0 Universal Serial Bus with On-The-Go Support (USB OTG) 239
19.0 Enhanced Parallel Master Port (EPMP) 273
20.0 Real-Time Clock and Calendar (RTCC) 285
21.0 32-Bit Programmable Cyclic Redundancy Check (CRC) Generator 297
22.0 Graphics Controller Module (GFX) 305
23.0 10-Bit High-Speed A/D Converter 325
24.0 Triple Comparator Module 335
25.0 Comparator Voltage Reference 341
26.0 Charge Time Measurement Unit (CTMU) 343
27.0 Special Features 347
28.0 Development Support 359
29.0 Instruction Set Summary 363
30.0 Electrical Characteristics 371
31.0 Packaging Information 387
Appendix A: Revision History 397
Index 399
The Microchip Web Site 405
Customer Change Notification Service 405
Customer Support 405
Reader Response 406
Product Identification System 407

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@microchip.com** or fax the **Reader Response Form** in the back of this data sheet to (480) 792-4150. We welcome your feedback.

Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000A is version A of document DS30000).

Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; http://www.microchip.com
- Your local Microchip sales office (see last page)

When contacting a sales office, please specify which device, revision of silicon and data sheet (include literature number) you are using.

Customer Notification System

Register on our web site at **www.microchip.com** to receive the most current information on all of our products.

1.0 DEVICE OVERVIEW

This document contains device-specific information for the following devices:

- PIC24FJ128DA106
- PIC24FJ256DA106
- PIC24FJ128DA110
- PIC24FJ256DA110
- PIC24FJ128DA206
- PIC24FJ256DA206
- PIC24FJ128DA210
- PIC24FJ256DA210

The PIC24FJ256DA210 family enhances on the existing line of Microchip's 16-bit microcontrollers, adding a new Graphics Controller (GFX) module to interface with a graphical LCD display and also adds large data RAM, up to 96 Kbytes. The PIC24FJ256DA210 family allows the CPU to fetch data directly from an external memory device using the EPMP module.

1.1 Core Features

1.1.1 16-BIT ARCHITECTURE

Central to all PIC24F devices is the 16-bit modified Harvard architecture, first introduced with Microchip's dsPIC® Digital Signal Controllers (DSCs). The PIC24F CPU core offers a wide range of enhancements, such as:

- 16-bit data and 24-bit address paths with the ability to move information between data and memory spaces
- Linear addressing of up to 12 Mbytes (program space) and 32 Kbytes (data)
- A 16-element working register array with built-in software stack support
- A 17 x 17 hardware multiplier with support for integer math
- Hardware support for 32 by 16-bit division
- An instruction set that supports multiple addressing modes and is optimized for high-level languages, such as 'C'
- Operational performance up to 16 MIPS

1.1.2 POWER-SAVING TECHNOLOGY

All of the devices in the PIC24FJ256DA210 family incorporate a range of features that can significantly reduce power consumption during operation. Key items include:

- **On-the-Fly Clock Switching:** The device clock can be changed under software control to the Timer1 source or the internal, low-power RC oscillator during operation, allowing the user to incorporate power-saving ideas into their software designs.
- **Doze Mode Operation:** When timing-sensitive applications, such as serial communications, require the uninterrupted operation of peripherals, the CPU clock speed can be selectively reduced, allowing incremental power savings without missing a beat.
- **Instruction-Based Power-Saving Modes:** The microcontroller can suspend all operations, or selectively shut down its core while leaving its peripherals active with a single instruction in software.

1.1.3 OSCILLATOR OPTIONS AND FEATURES

All of the devices in the PIC24FJ256DA210 family offer five different oscillator options, allowing users a range of choices in developing application hardware. These include:

- Two Crystal modes using crystals or ceramic resonators.
- Two External Clock modes offering the option of a divide-by-2 clock output.
- A Fast Internal Oscillator (FRC) with a nominal 8 MHz output, which can also be divided under software control to provide clock speeds as low as 31 kHz.
- A Phase Lock Loop (PLL) frequency multiplier, available to the external oscillator modes and the FRC oscillator, which allows clock speeds of up to 32 MHz.
- A separate Low-Power Internal RC Oscillator (LPRC) with a fixed 31 kHz output, which provides a low-power option for timing-insensitive applications.

The internal oscillator block also provides a stable reference source for the Fail-Safe Clock Monitor (FSCM). This option constantly monitors the main clock source against a reference signal provided by the internal oscillator and enables the controller to switch to the internal oscillator, allowing for continued low-speed operation or a safe application shutdown.

1.1.4 EASY MIGRATION

Regardless of the memory size, all devices share the same rich set of peripherals, allowing for a smooth migration path as applications grow and evolve. The consistent pinout scheme used throughout the entire family also aids in migrating from one device to the next larger, or even in jumping from 64-pin to 100-pin devices.

The PIC24F family is pin compatible with devices in the dsPIC33 family, and shares some compatibility with the pinout schema for PIC18 and dsPIC30. This extends the ability of applications to grow from the relatively simple, to the powerful and complex, yet still selecting a Microchip device.

1.2 Graphics Controller

With the PIC24FJ256DA210 family of devices, Microchip introduces the Graphics Controller module, which acts as an interface between the CPU (mainly through SFRs) and a display. On-board RAM is provided for display buffer, scratch areas, images and fonts. In some cases, the RAM requirements for the display used exceeds the on-board RAM; external memory connected through EPMP can be used.

This module provides acceleration for drawing points, vertical and horizontal lines, rectangles, copying rectangles between different locations on screen, drawing text and decompressing compressed data.

1.3 USB On-The-Go

The USB On-The-Go (USB OTG) module provides on-chip functionality as a target device compatible with the USB 2.0 standard, as well as limited stand-alone functionality as a USB embedded host. By implementing USB Host Negotiation Protocol (HNP), the module can also dynamically switch between device and host operation, allowing for a much wider range of versatile USB enabled applications on a microcontroller platform.

In addition to USB host functionality, PIC24FJ256DA210 family devices provide a true single chip USB solution, including an on-chip transceiver and voltage regulator, and a voltage boost generator for sourcing bus power during host operations.

1.4 Other Special Features

- **Peripheral Pin Select:** The Peripheral Pin Select (PPS) feature allows most digital peripherals to be mapped over a fixed set of digital I/O pins. Users may independently map the input and/or output of any one of the many digital peripherals to any one of the I/O pins.
- **Communications:** The PIC24FJ256DA210 family incorporates a range of serial communication peripherals to handle a range of application requirements. There are three independent I^2C™ modules that support both Master and Slave modes of operation. Devices also have, through the PPS feature, four independent UARTs with built-in IrDA® encoders/decoders and three SPI modules.
- **Analog Features:** All members of the PIC24FJ256DA210 family include a 10-bit A/D Converter (ADC) module and a triple comparator module. The ADC module incorporates programmable acquisition time, allowing for a channel to be selected and a conversion to be initiated without waiting for a sampling period, and faster sampling speeds. The comparator module includes three analog comparators that are configurable for a wide range of operations.
- **CTMU Interface:** In addition to their other analog features, members of the PIC24FJ256DA210 family include the CTMU interface module. This provides a convenient method for precision time measurement and pulse generation, and can serve as an interface for capacitive sensors.
- **Enhanced Parallel Master/Parallel Slave Port:** There are general purpose I/O ports, which can be configured for parallel data communications. In this mode, the device can be master or slave on the communication bus. 4-bit, 8-bit and 16-bit data transfers, with up to 23 external address lines are supported in Master modes.
- **Real-Time Clock and Calendar: (RTCC)** This module implements a full-featured clock and calendar with alarm functions in hardware, freeing up timer resources and program memory space for use of the core application.

1.5 Details on Individual Family Members

Devices in the PIC24FJ256DA210 family are available in 64-pin and 100-pin packages. The general block diagram for all devices is shown in Figure 1-1.

The devices are differentiated from each other in seven ways:

1. Flash program memory (128 Kbytes for PIC24FJ128DAXXX devices and 256 Kbytes for PIC24FJ256DAXXX devices).
2. Data memory (24 Kbytes for PIC24FJXXXDA1XX devices, and 96 Kbytes for PIC24FJXXXDA2XX devices).
3. Available I/O pins and ports (52 pins on 6 ports for PIC24FJXXXDAX06 devices and 84 pins on 7 ports for PIC24FJXXXDAX10 devices).
4. Available Interrupt-on-Change Notification (ICN) inputs (52 on PIC24FJXXXDAx06 devices and 84 on PIC24FJXXXDAX10 devices).
5. Available remappable pins (29 pins on PIC24FJXXXDAX06 devices and 44 pins on PIC24FJXXXDAX10 devices).
6. Analog channels for ADC (16 channels for PIC24FJXXXDAX06 devices and 24 channels for PIC24FJxxxDAx10 devices).
7. EPMP module (available in PIC24FJXXXDAX10 devices and not in PIC24FJXXXDAX06 devices).

All other features for devices in this family are identical. These are summarized in Table 1-1 and Table 1-2.

A list of the pin features available on the PIC24FJ256DA210 family devices, sorted by function, is shown in Table 1-1. Note that this table shows the pin location of individual peripheral features and not how they are multiplexed on the same pin. This information is provided in the pinout diagrams in the beginning of the data sheet. Multiplexed features are sorted by the priority given to a feature, with the highest priority peripheral being listed first.

TABLE 1-1: DEVICE FEATURES FOR THE PIC24FJ256DA210 FAMILY: 64-PIN

Features	PIC24FJ128DA106	PIC24FJ256DA106	PIC24FJ128DA206	PIC24FJ256DA206
Operating Frequency	DC – 32 MHz			
Program Memory (bytes)	128K	256K	128K	256K
Program Memory (instructions)	44,032	87,552	44,032	87,552
Data Memory (bytes)	24K		96K	
Interrupt Sources (soft vectors/ NMI traps)	65 (61/4)			
I/O Ports	Ports B, C, D, E, F, G			
Total I/O Pins	52			
Remappable Pins	29 (28 I/O, 1 Input only)			
Timers: Total Number (16-bit)	5(1)			
32-Bit (from paired 16-bit timers)	2			
Input Capture Channels	9(1)			
Output Compare/PWM Channels	9(1)			
Input Change Notification Interrupt	52			
Serial Communications: UART	4(1)			
SPI (3-wire/4-wire)	3(1)			
I^2C™	3			
Parallel Communications (EPMP/PSP)	No			
JTAG Boundary Scan	Yes			
10-Bit Analog-to-Digital Converter (ADC) Module (input channels)	16			
Analog Comparators	3			
CTMU Interface	Yes			
USB OTG	Yes			
Graphics Controller	Yes			
Resets (and Delays)	POR, BOR, RESET Instruction, $\overline{MCLR}$, WDT; Illegal Opcode, REPEAT Instruction, Hardware Traps, Configuration Word Mismatch (OST, PLL Lock)			
Instruction Set	76 Base Instructions, Multiple Addressing Mode Variations			
Packages	64-Pin TQFP and QFN			

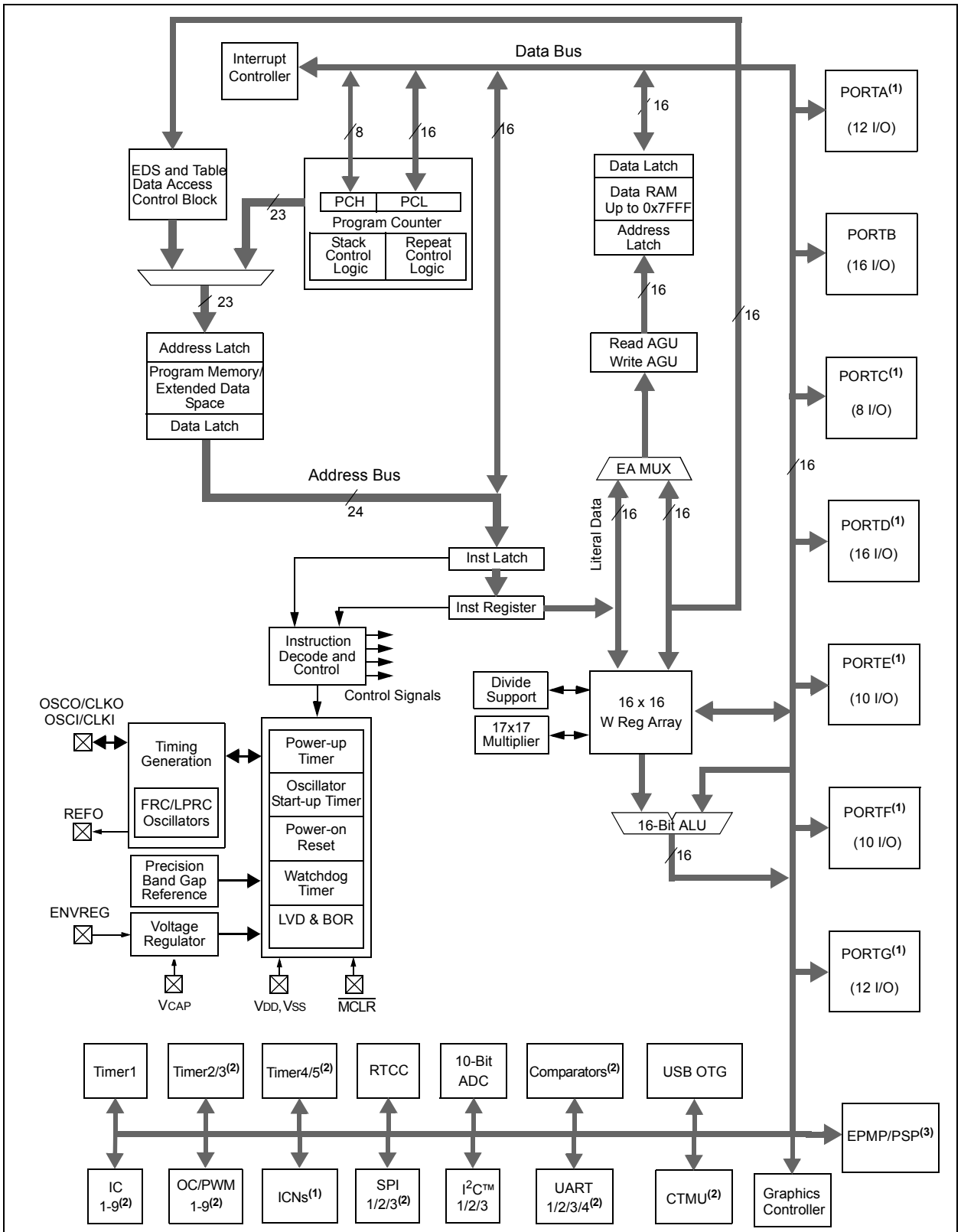
Note 1: Peripherals are accessible through remappable pins.

TABLE 1-2: DEVICE FEATURES FOR THE PIC24FJ256DA210 FAMILY: 100-PIN DEVICES

Features	PIC24FJ128DA110	PIC24FJ256DA110	PIC24FJ128DA210	PIC24FJ256DA210
Operating Frequency	DC – 32 MHz			
Program Memory (bytes)	128K	256K	128K	256K
Program Memory (instructions)	44,032	87,552	44,032	87,552
Data Memory (bytes)	24K		96K	
Interrupt Sources (soft vectors/NMI traps)	66 (62/4)			
I/O Ports	Ports A, B, C, D, E, F, G			
Total I/O Pins	84			
Remappable Pins	44 (32 I/O, 12 input only)			
Timers:				
Total Number (16-bit)	5(1)			
32-Bit (from paired 16-bit timers)	2			
Input Capture Channels	9(1)			
Output Compare/PWM Channels	9(1)			
Input Change Notification Interrupt	84			
Serial Communications:				
UART	4(1)			
SPI (3-wire/4-wire)	3(1)			
I^2C™	3			
Parallel Communications (EPMP/PSP)	Yes			
JTAG Boundary Scan	Yes			
10-Bit Analog-to-Digital Converter (ADC) Module (input channels)	24			
Analog Comparators	3			
CTMU Interface	Yes			
USB OTG	Yes			
Graphics Controller	Yes			
Resets (and delays)	POR, BOR, RESET Instruction, $\overline{MCLR}$, WDT; Illegal Opcode, REPEAT Instruction, Hardware Traps, Configuration Word Mismatch (OST, PLL Lock)			
Instruction Set	76 Base Instructions, Multiple Addressing Mode Variations			
Packages	100-Pin TQFP and 121-Pin BGA			

Note 1: Peripherals are accessible through remappable pins.

FIGURE 1-1: PIC24FJ256DA210 FAMILY GENERAL BLOCK DIAGRAM



Note 1: Not all I/O pins or features are implemented on all device pinout configurations. See Table 1-1 for specific implementations by pin count.
2: These peripheral I/Os are only accessible through remappable pins.
3: Not available on 64-pin devices (PIC24FJxxxDAx06).

TABLE 1-3: PIC24FJ256DA210 FAMILY PINOUT DESCRIPTIONS

Function	Pin Number			I/O	Input Buffer	Description
	64-Pin TQFP/QFN	100-Pin TQFP	121-Pin BGA			
AN0	16	25	K2	I	ANA	A/D Analog Inputs.
AN1	15	24	K1	I	ANA	
AN2	14	23	J2	I	ANA	
AN3	13	22	J1	I	ANA	
AN4	12	21	H2	I	ANA	
AN5	11	20	H1	I	ANA	
AN6	17	26	L1	I	ANA	
AN7	18	27	J3	I	ANA	
AN8	21	32	K4	I	ANA	
AN9	22	33	L4	I	ANA	
AN10	23	34	L5	I	ANA	
AN11	24	35	J5	I	ANA	
AN12	27	41	J7	I	ANA	
AN13	28	42	L7	I	ANA	
AN14	29	43	K7	I	ANA	
AN15	30	44	L8	I	ANA	
AN16	—	9	E1	I	ANA	
AN17	—	10	E3	I	ANA	
AN18	—	11	F4	I	ANA	
AN19	—	12	F2	I	ANA	
AN20	—	14	F3	I	ANA	
AN21	—	19	G2	I	ANA	
AN22	—	92	B5	I	ANA	
AN23	—	91	C5	I	ANA	
AVDD	19	30	J4	P	—	Positive Supply for Analog modules.
AVSS	20	31	L3	P	—	Ground Reference for Analog modules.
C1INA	11	20	H1	I	ANA	Comparator 1 Input A.
C1INB	12	21	H2	I	ANA	Comparator 1 Input B.
C1INC	5	11	F4	I	ANA	Comparator 1 Input C.
C1IND	4	10	E3	I	ANA	Comparator 1 Input D.
C2INA	13	22	J1	I	ANA	Comparator 2 Input A.
C2INB	14	23	J2	I	ANA	Comparator 2 Input B.
C2INC	8	14	F3	I	ANA	Comparator 2 Input C.
C2IND	6	12	F2	I	ANA	Comparator 2 Input D.
C3INA	55	84	C7	I	ANA	Comparator 3 Input A.
C3INB	54	83	D7	I	ANA	Comparator 3 Input B.
C3INC	48	74	B11	I	ANA	Comparator 3 Input C.
C3IND	47	73	C10	I	ANA	Comparator 3 Input D.
CLKI	39	63	F9	I	ST	Main Clock Input Connection.
CLKO	40	64	F11	O	—	System Clock Output.

Legend: TTL = TTL input buffer; ANA = Analog level input/output; ST = Schmitt Trigger input buffer; I^2C™ = I^2C/SMBus input buffer

Note 1: The alternate EPMP pins are selected when the $\overline{\text{ALTPMP}}$ (CW3<12>) bit is programmed to '0'.

2: The PMSC2 signal will replace the PMA15 signal on the 15-pin PMA when CSF<1:0> = 01 or 10.

3: The PMCS1 signal will replace the PMA14 signal on the 14-pin PMA when CSF<1:0> = 10.

4: The alternate VREF pins selected when the $\overline{\text{ALTVREF}}$ (CW1<5>) bit is programmed to '0'.

TABLE 1-3: PIC24FJ256DA210 FAMILY PINOUT DESCRIPTIONS (CONTINUED)

Function	Pin Number			I/O	Input Buffer	Description
	64-Pin TQFP/QFN	100-Pin TQFP	121-Pin BGA			
CN0	48	74	B11	I	ST	Interrupt-on-Change Inputs.
CN1	47	73	C10	I	ST	
CN2	16	25	K2	I	ST	
CN3	15	24	K1	I	ST	
CN4	14	23	J2	I	ST	
CN5	13	22	J1	I	ST	
CN6	12	21	H2	I	ST	
CN7	11	20	H1	I	ST	
CN8	4	10	E3	I	ST	
CN9	5	11	F4	I	ST	
CN10	6	12	F2	I	ST	
CN11	8	14	F3	I	ST	
CN12	30	44	L8	I	ST	
CN13	52	81	C8	I	ST	
CN14	53	82	B8	I	ST	
CN15	54	83	D7	I	ST	
CN16	55	84	C7	I	ST	
CN17	31	49	L10	I	ST	
CN18	32	50	L11	I	ST	
CN19	—	80	D8	I	ST	
CN20	—	47	L9	I	ST	
CN21	—	48	K9	I	ST	
CN22	40	64	F11	I	ST	
CN23	39	63	F9	I	ST	
CN24	17	26	L1	I	ST	
CN25	18	27	J3	I	ST	
CN26	21	32	K4	I	ST	
CN27	22	33	L4	I	ST	
CN28	23	34	L5	I	ST	
CN29	24	35	J5	I	ST	
CN30	27	41	J7	I	ST	
CN31	28	42	L7	I	ST	
CN32	29	43	K7	I	ST	
CN33	—	17	G3	I	ST	
CN34	—	38	J6	I	ST	
CN35	—	58	H11	I	ST	
CN36	—	59	G10	I	ST	
CN37	—	60	G11	I	ST	
CN38	—	61	G9	I	ST	
CN39	—	91	C5	I	ST	

Legend: TTL = TTL input buffer; ANA = Analog level input/output; ST = Schmitt Trigger input buffer; I^2C™ = I^2C/SMBus input buffer

Note 1: The alternate EPMP pins are selected when the $\overline{\text{ALTPMP}}$ (CW3<12>) bit is programmed to '0'.

2: The PMSC2 signal will replace the PMA15 signal on the 15-pin PMA when CSF<1:0> = 01 or 10.

3: The PMCS1 signal will replace the PMA14 signal on the 14-pin PMA when CSF<1:0> = 10.

4: The alternate V_{REF} pins selected when the $\overline{\text{ALTVREF}}$ (CW1<5>) bit is programmed to '0'.

TABLE 1-3: PIC24FJ256DA210 FAMILY PINOUT DESCRIPTIONS (CONTINUED)

Function	Pin Number			I/O	Input Buffer	Description
	64-Pin TQFP/QFN	100-Pin TQFP	121-Pin BGA			
CN40	—	92	B5	I	ST	Interrupt-on-Change Inputs.
CN41	—	28	L2	I	ST	
CN42	—	29	K3	I	ST	
CN43	—	66	E11	I	ST	
CN44	—	67	E8	I	ST	
CN45	—	6	D1	I	ST	
CN46	—	7	E4	I	ST	
CN47	—	8	E2	I	ST	
CN48	—	9	E1	I	ST	
CN49	46	72	D9	I	ST	
CN50	49	76	A11	I	ST	
CN51	50	77	A10	I	ST	
CN52	51	78	B9	I	ST	
CN53	42	68	E9	I	ST	
CN54	43	69	E10	I	ST	
CN55	44	70	D11	I	ST	
CN56	45	71	C11	I	ST	
CN57	—	79	A9	I	ST	
CN58	60	93	A4	I	ST	
CN59	61	94	B4	I	ST	
CN60	62	98	B3	I	ST	
CN61	63	99	A2	I	ST	
CN62	64	100	A1	I	ST	
CN63	1	3	D3	I	ST	
CN64	2	4	C1	I	ST	
CN65	3	5	D2	I	ST	
CN66	—	18	G1	I	ST	
CN67	—	19	G2	I	ST	
CN68	58	87	B6	I	ST	
CN69	59	88	A6	I	ST	
CN70	—	52	K11	I	ST	
CN71	33	51	K10	I	ST	
CN73	—	54	H8	I	ST	
CN74	—	53	J10	I	ST	
CN75	—	40	K6	I	ST	
CN76	—	39	L6	I	ST	
CN77	—	90	A5	I	ST	
CN78	—	89	E6	I	ST	
CN79	—	96	C3	I	ST	
CN80	—	97	A3	I	ST	

Legend: TTL = TTL input buffer; ST = Schmitt Trigger input buffer; ANA = Analog level input/output; I^2C™ = I^2C/SMBus input buffer

Note 1: The alternate EPMP pins are selected when the $\overline{\text{ALTPMP}}$ (CW3<12>) bit is programmed to '0'.

2: The PMSC2 signal will replace the PMA15 signal on the 15-pin PMA when CSF<1:0> = 01 or 10.

3: The PMCS1 signal will replace the PMA14 signal on the 14-pin PMA when CSF<1:0> = 10.

4: The alternate V_{REF} pins selected when the $\overline{\text{ALTVREF}}$ (CW1<5>) bit is programmed to '0'.

TABLE 1-3: PIC24FJ256DA210 FAMILY PINOUT DESCRIPTIONS (CONTINUED)

Function	Pin Number			I/O	Input Buffer	Description
	64-Pin TQFP/QFN	100-Pin TQFP	121-Pin BGA			
CN81	—	95	C4	I	ST	Interrupt-on-Change Inputs.
CN82	—	1	B2	I	ST	
CN83	37	57	H10	I	ST	
CN84	36	56	J11	I	ST	
CTEDG1	28	42	L7	I	ANA	CTMU External Edge Input 1.
CTEDG2	27	41	J7	I	ANA	CTMU External Edge Input 2.
CTPLS	29	43	K7	O	—	CTMU Pulse Output.
CVREF	23	34	L5	O	—	Comparator Voltage Reference Output.
D+	37	57	H10	I/O	—	USB Differential Plus Line (internal transceiver).
D-	36	56	J11	I/O	—	USB Differential Minus Line (internal transceiver).
DMH	46	72	D9	O	—	D- External Pull-up Control Output.
DMLN	42	68	E9	O	—	D- External Pull-down Control Output.
DPH	50	77	A10	O	—	D+ External Pull-up Control Output.
DPLN	43	69	E10	O	—	D+ External Pull-down Control Output.
ENVREG	57	86	J7	I	ST	Voltage Regulator Enable.
GCLK	52	1	B2	O	—	Graphics Display Pixel Clock.
GD0	60	6	D1	O	—	Graphics Controller Data Output.
GD1	61	8	E2	O	—	
GD2	62	39	L6	O	—	
GD3	63	52	K11	O	—	
GD4	31	21	H2	O	—	
GD5	32	22	J1	O	—	
GD6	44	23	J2	O	—	
GD7	45	76	A11	O	—	
GD8	43	7	E4	O	—	
GD9	49	53	J10	O	—	
GD10	58	69	E10	O	—	
GD11	59	77	A10	O	—	
GD12	2	32	K4	O	—	
GD13	3	33	L4	O	—	
GD14	6	47	L9	O	—	
GD15	8	48	K9	O	—	
GEN	51	91	C5	O	—	Graphics Display Enable Output.
GPWR	53	27	J3	O	—	Graphics Display Power System Enable.
HSYNC	64	97	A3	O	—	Graphics Display Horizontal Sync Pulse.
INT0	46	72	D9	I	ST	External Interrupt Input.
$\overline{\text{MCLR}}$	7	13	F1	I	ST	Master Clear (device Reset) Input. This line is brought low to cause a Reset.
OSCI	39	63	F9	I	ANA	Main Oscillator Input Connection.
OSCO	40	64	F11	O	ANA	Main Oscillator Output Connection.

Legend: TTL = TTL input buffer; ST = Schmitt Trigger input buffer; ANA = Analog level input/output; I^2C™ = I^2C/SMBus input buffer

Note 1: The alternate EPMP pins are selected when the $\overline{\text{ALTPMP}}$ (CW3<12>) bit is programmed to '0'.

2: The PMSC2 signal will replace the PMA15 signal on the 15-pin PMA when CSF<1:0> = 01 or 10.

3: The PMCS1 signal will replace the PMA14 signal on the 14-pin PMA when CSF<1:0> = 10.

4: The alternate VREF pins selected when the $\overline{\text{ALTVREF}}$ (CW1<5>) bit is programmed to '0'.

TABLE 1-3: PIC24FJ256DA210 FAMILY PINOUT DESCRIPTIONS (CONTINUED)

Function	Pin Number			I/O	Input Buffer	Description
	64-Pin TQFP/QFN	100-Pin TQFP	121-Pin BGA			
PGEC1	15	24	K1	I/O	ST	In-Circuit Debugger/Emulator/ICSP™ Programming Clock 1.
PGED1	16	25	K2	I/O	ST	In-Circuit Debugger/Emulator/ICSP Programming Data 1.
PGEC2	17	26	L1	I/O	ST	In-Circuit Debugger/Emulator/ICSP Programming Clock 2.
PGED2	18	27	J3	I/O	ST	In-Circuit Debugger/Emulator/ICSP Programming Data 2.
PGEC3	11	20	H1	I/O	ST	In-Circuit Debugger/Emulator/ICSP Programming Clock 3.
PGED3	12	21	H2	I/O	ST	In-Circuit Debugger/Emulator/ICSP Programming Data 3.
PMA0	—	44	L8	I/O	ST	Parallel Master Port Address bit 0 Input (Buffered Slave modes) and Output (Master modes).
PMA1	—	43	K7	I/O	ST	Parallel Master Port Address Bit 1 Input (Buffered Slave modes) and Output (Master modes).
PMA2	—	14	F3	O	—	Parallel Master Port Address bits<22:2>.
PMA3	—	12, 60(1)	F2, G11(1)	O	—	
PMA4	—	11,59(1)	F4,G10(1)	O	—	
PMA5	—	10,40(1)	E3,K6(1)	O	—	
PMA6	—	29	K3	O	—	
PMA7	—	28	L2	O	—	
PMA8	—	50	L11	O	—	
PMA9	—	49	L10	O	—	
PMA10	—	42	L7	O	—	
PMA11	—	41	J7	O	—	
PMA12	—	35	J5	O	—	
PMA13	—	34	L5	O	—	
PMA14	—	71	C11	O	—	
PMA15	—	70	D11	O	—	
PMA16	—	95	C4	O	—	
PMA17	—	92	B5	O	—	
PMA18	—	40,10(1)	K6,E3(1)	O	—	
PMA19	—	19	G2	O	—	
PMA20	—	59, 11(1)	G10, F4(1)	O	—	
PMA21	—	60,12(1)	G11,F2(1)	O	—	
PMA22	—	66,9(1)	E11,E1(1)	O	—	
PMACK1	—	77	A10	I	ST/TTL	Parallel Master Port Acknowledge Input 1.
PMACK2	—	69	E10	I	ST/TTL	Parallel Master Port Acknowledge Input 2.
PMALL	—	44	L8	O	—	Parallel Master Port Lower Address Latch Strobe.
PMALH	—	43	K7	O	—	Parallel Master Port Higher Address Latch Strobe.
PMALU	—	14	F3	O	—	Parallel Master Port Upper Address Latch Strobe.
PMBE0	—	78	B9	O	—	Parallel Master Port Byte Enable Strobe 0.
PMBE1	—	67	E8	O	—	Parallel Master Port Byte Enable Strobe 1.
PMCS1	—	71(3),18	C11(3),G1	I/O	ST/TTL	Parallel Master Port Chip Select Strobe 1.
PMCS2	—	70(2),9, 66(1)	D11(2),E1, E11(1)	O	—	Parallel Master Port Chip Select Strobe 2.

Legend: TTL = TTL input buffer; ST = Schmitt Trigger input buffer; ANA = Analog level input/output; I^2C™ = I^2C/SMBus input buffer

Note 1: The alternate EPMP pins are selected when the $\overline{\text{ALTPMP}}$ (CW3<12>) bit is programmed to '0'.

2: The PMSC2 signal will replace the PMA15 signal on the 15-pin PMA when CSF<1:0> = 01 or 10.

3: The PMCS1 signal will replace the PMA14 signal on the 14-pin PMA when CSF<1:0> = 10.

4: The alternate VREF pins selected when the $\overline{\text{ALTVREF}}$ (CW1<5>) bit is programmed to '0'.

TABLE 1-3: PIC24FJ256DA210 FAMILY PINOUT DESCRIPTIONS (CONTINUED)

Function	Pin Number			I/O	Input Buffer	Description
	64-Pin TQFP/QFN	100-Pin TQFP	121-Pin BGA			
PMD0	—	93	A4	I/O	ST/TTL	Parallel Master Port Data bits<15:0>.
PMD1	—	94	B4	I/O	ST/TTL	
PMD2	—	98	B3	I/O	ST/TTL	
PMD3	—	99	A2	I/O	ST/TTL	
PMD4	—	100	A1	I/O	ST/TTL	
PMD5	—	3	D3	I/O	ST/TTL	
PMD6	—	4	C1	I/O	ST/TTL	
PMD7	—	5	D2	I/O	ST/TTL	
PMD8	—	90	A5	I/O	ST/TTL	
PMD9	—	89	E6	I/O	ST/TTL	
PMD10	—	88	A6	I/O	ST/TTL	
PMD11	—	87	B6	I/O	ST/TTL	
PMD12	—	79	A9	I/O	ST/TTL	
PMD13	—	80	D8	I/O	ST/TTL	
PMD14	—	83	D7	I/O	ST/TTL	
PMD15	—	84	C7	I/O	ST/TTL	
PMRD	—	82	B8	I/O	ST/TTL	Parallel Master Port Read Strobe.
PMWR	—	81	C8	I/O	ST/TTL	Parallel Master Port Write Strobe.
RA0	—	17	G3	I/O	ST	PORTA Digital I/O.
RA1	—	38	J6	I/O	ST	
RA2	—	58	H11	I/O	ST	
RA3	—	59	G10	I/O	ST	
RA4	—	60	G11	I/O	ST	
RA5	—	61	G9	I/O	ST	
RA6	—	91	C5	I/O	ST	
RA7	—	92	B5	I/O	ST	
RA9	—	28	L2	I/O	ST	
RA10	—	29	K3	I/O	ST	
RA14	—	66	E11	I/O	ST	
RA15	—	67	E8	I/O	ST	

Legend: TTL = TTL input buffer; ST = Schmitt Trigger input buffer; ANA = Analog level input/output; I^2C™ = I^2C/SMBus input buffer

Note 1: The alternate EPMP pins are selected when the $\overline{\text{ALTPMP}}$ (CW3<12>) bit is programmed to '0'.

2: The PMSC2 signal will replace the PMA15 signal on the 15-pin PMA when CSF<1:0> = 01 or 10.

3: The PMCS1 signal will replace the PMA14 signal on the 14-pin PMA when CSF<1:0> = 10.

4: The alternate V_{REF} pins selected when the $\overline{\text{ALTVREF}}$ (CW1<5>) bit is programmed to '0'.

TABLE 1-3: PIC24FJ256DA210 FAMILY PINOUT DESCRIPTIONS (CONTINUED)

Function	Pin Number			I/O	Input Buffer	Description
	64-Pin TQFP/QFN	100-Pin TQFP	121-Pin BGA			
RB0	16	25	K2	I/O	ST	PORTB Digital I/O.
RB1	15	24	K1	I/O	ST	
RB2	14	23	J2	I/O	ST	
RB3	13	22	J1	I/O	ST	
RB4	12	21	H2	I/O	ST	
RB5	11	20	H1	I/O	ST	
RB6	17	26	L1	I/O	ST	
RB7	18	27	J3	I/O	ST	
RB8	21	32	K4	I/O	ST	
RB9	22	33	L4	I/O	ST	
RB10	23	34	L5	I/O	ST	
RB11	24	35	J5	I/O	ST	
RB12	27	41	J7	I/O	ST	
RB13	28	42	L7	I/O	ST	
RB14	29	43	K7	I/O	ST	
RB15	30	44	L8	I/O	ST	
RC1	—	6	D1	I/O	ST	PORTC Digital I/O.
RC2	—	7	E4	I/O	ST	
RC3	—	8	E2	I/O	ST	
RC4	—	9	E1	I/O	ST	
RC12	39	63	F9	I/O	ST	
RC13	47	73	C10	I/O	ST	
RC14	48	74	B11	I/O	ST	
RC15	40	64	F11	I/O	ST	
RCV	18	27	J3	I	ST	USB Receive Input (from external transceiver).

Legend: TTL = TTL input buffer; ST = Schmitt Trigger input buffer; ANA = Analog level input/output; I^2C™ = I^2C/SMBus input buffer

Note 1: The alternate EPMP pins are selected when the $\overline{\text{ALTPMP}}$ (CW3<12>) bit is programmed to '0'.

2: The PMSC2 signal will replace the PMA15 signal on the 15-pin PMA when CSF<1:0> = 01 or 10.

3: The PMCS1 signal will replace the PMA14 signal on the 14-pin PMA when CSF<1:0> = 10.

4: The alternate VREF pins selected when the $\overline{\text{ALTVREF}}$ (CW1<5>) bit is programmed to '0'.

TABLE 1-3: PIC24FJ256DA210 FAMILY PINOUT DESCRIPTIONS (CONTINUED)

Function	Pin Number			I/O	Input Buffer	Description
	64-Pin TQFP/QFN	100-Pin TQFP	121-Pin BGA			
RD0	46	72	D9	I/O	ST	PORTD Digital I/O.
RD1	49	76	A11	I/O	ST	
RD2	50	77	A10	I/O	ST	
RD3	51	78	B9	I/O	ST	
RD4	52	81	C8	I/O	ST	
RD5	53	82	B8	I/O	ST	
RD6	54	83	D7	I/O	ST	
RD7	55	84	C7	I/O	ST	
RD8	42	68	E9	I/O	ST	
RD9	43	69	E10	I/O	ST	
RD10	44	70	D11	I/O	ST	
RD11	45	71	C11	I/O	ST	
RD12	—	79	A9	I/O	ST	
RD13	—	80	D8	I/O	ST	
RD14	—	47	L9	I/O	ST	
RD15	—	48	K9	I/O	ST	
RE0	60	93	A4	I/O	ST	PORTE Digital I/O.
RE1	61	94	B4	I/O	ST	
RE2	62	98	B3	I/O	ST	
RE3	63	99	A2	I/O	ST	
RE4	64	100	A1	I/O	ST	
RE5	1	3	D3	I/O	ST	
RE6	2	4	C1	I/O	ST	
RE7	3	5	D2	I/O	ST	
RE8	—	18	G1	I/O	ST	
RE9	—	19	G2	I/O	ST	
REFO	30	44	L8	O	—	Reference Clock Output.
RF0	58	87	B6	I/O	ST	PORTF Digital I/O.
RF1	59	88	A6	I/O	ST	
RF2	—	52	K11	I/O	ST	
RF3	33	51	K10	I/O	ST	
RF4	31	49	L10	I/O	ST	
RF5	32	50	L11	I/O	ST	
RF7	34	54	H8	I/O	ST	
RF8	—	53	J10	I/O	ST	
RF12	—	40	K6	I/O	ST	
RF13	—	39	L6	I/O	ST	

Legend: TTL = TTL input buffer; ST = Schmitt Trigger input buffer; ANA = Analog level input/output; I^2C™ = I^2C/SMBus input buffer

Note 1: The alternate EPMP pins are selected when the $\overline{\text{ALTPMP}}$ (CW3<12>) bit is programmed to '0'.

2: The PMSC2 signal will replace the PMA15 signal on the 15-pin PMA when CSF<1:0> = 01 or 10.

3: The PMCS1 signal will replace the PMA14 signal on the 14-pin PMA when CSF<1:0> = 10.

4: The alternate VREF pins selected when the $\overline{\text{ALTVREF}}$ (CW1<5>) bit is programmed to '0'.

TABLE 1-3: PIC24FJ256DA210 FAMILY PINOUT DESCRIPTIONS (CONTINUED)

Function	Pin Number			I/O	Input Buffer	Description
	64-Pin TQFP/QFN	100-Pin TQFP	121-Pin BGA			
RG0	—	90	A5	I/O	ST	PORTG Digital I/O.
RG1	—	89	E6	I/O	ST	
RG2	37	57	H10	I/O	ST	
RG3	36	56	J11	I/O	ST	
RG6	4	10	E3	I/O	ST	
RG7	5	11	F4	I/O	ST	
RG8	6	12	F2	I/O	ST	
RG9	8	14	F3	I/O	ST	
RG12	—	96	C3	I/O	ST	
RG13	—	97	A3	I/O	ST	
RG14	—	95	C4	I/O	ST	
RG15	—	1	B2	I/O	ST	
RP0	16	25	K2	I/O	ST	Remappable Peripheral (input or output).
RP1	15	24	K1	I/O	ST	
RP2	42	68	E9	I/O	ST	
RP3	44	70	D11	I/O	ST	
RP4	43	69	E10	I/O	ST	
RP5	—	48	K9	I/O	ST	
RP6	17	26	L1	I/O	ST	
RP7	18	27	J3	I/O	ST	
RP8	21	32	K4	I/O	ST	
RP9	22	33	L4	I/O	ST	
RP10	31	49	L10	I/O	ST	
RP11	46	72	D9	I/O	ST	
RP12	45	71	C11	I/O	ST	
RP13	14	23	J2	I/O	ST	
RP14	29	43	K7	I/O	ST	
RP15	—	53	J10	I/O	ST	
RP16	33	51	K10	I/O	ST	
RP17	32	50	L11	I/O	ST	
RP18	11	20	H1	I/O	ST	
RP19	6	12	F2	I/O	ST	

Legend: TTL = TTL input buffer; ST = Schmitt Trigger input buffer; ANA = Analog level input/output; I^2C™ = I^2C/SMBus input buffer

Note 1: The alternate EPMP pins are selected when the $\overline{\text{ALTPMP}}$ (CW3<12>) bit is programmed to '0'.

2: The PMSC2 signal will replace the PMA15 signal on the 15-pin PMA when CSF<1:0> = 01 or 10.

3: The PMCS1 signal will replace the PMA14 signal on the 14-pin PMA when CSF<1:0> = 10.

4: The alternate V_{REF} pins selected when the $\overline{\text{ALTVREF}}$ (CW1<5>) bit is programmed to '0'.

TABLE 1-3: PIC24FJ256DA210 FAMILY PINOUT DESCRIPTIONS (CONTINUED)

Function	Pin Number			I/O	Input Buffer	Description
	64-Pin TQFP/QFN	100-Pin TQFP	121-Pin BGA			
RP20	53	82	B8	I/O	ST	Remappable Peripheral (input or output).
RP21	4	10	E3	I/O	ST	
RP22	51	78	B9	I/O	ST	
RP23	50	77	A10	I/O	ST	
RP24	49	76	A11	I/O	ST	
RP25	52	81	C8	I/O	ST	
RP26	5	11	F4	I/O	ST	
RP27	8	14	F3	I/O	ST	
RP28	12	21	H2	I/O	ST	
RP29	30	44	L8	I/O	ST	
RP30	—	52	K11	I/O	ST	
RP31	—	39	L6	I/O	ST	
RPI32	—	40	K6	I	ST	Remappable Peripheral (input only).
RPI33	—	18	G1	I	ST	
RPI34	—	19	G2	I	ST	
RPI35	—	67	E8	I	ST	
RPI36	—	66	E11	I	ST	
RPI37	48	74	B11	I	ST	
RPI38	—	6	D1	I	ST	
RPI39	—	7	E4	I	ST	
RPI40	—	8	E2	I	ST	
RPI41	—	9	E1	I	ST	
RPI42	—	79	A9	I	ST	
RPI43	—	47	L9	I	ST	
RTCC	42	68	E9	O	—	Real-Time Clock Alarm/Seconds Pulse Output.
SCL1	44	66	E11	I/O	I^2C™	I2C1 Synchronous Serial Clock Input/Output.
SCL2	32	58	H11	I/O	I^2C	I2C2 Synchronous Serial Clock Input/Output.
SCL3	2	4	C1	I/O	I^2C	I2C3 Synchronous Serial Clock Input/Output.
SCLKI	48	74	B11	O	ANA	Secondary Clock Input.
SDA1	43	67	E8	I/O	I^2C	I2C1 Data Input/Output.
SDA2	31	59	G10	I/O	I^2C	I2C2 Data Input/Output.
SDA3	3	5	D2	I/O	I^2C	I2C3 Data Input/Output.
SESSEND	55	84	C7	I	ST	USB VBUS Boost Generator, Comparator Input 3.
SESSVLD	59	88	A6	I	ST	USB VBUS Boost Generator, Comparator Input 2.
SOSCI	47	73	C10	I	ANA	Secondary Oscillator/Timer1 Clock Input.
SOSCO	48	74	B11	O	ANA	Secondary Oscillator/Timer1 Clock Output.
T1CK	48	74	B11	I	ST	Timer1 Clock.

Legend: TTL = TTL input buffer; ST = Schmitt Trigger input buffer; ANA = Analog level input/output; I^2C™ = I^2C/SMBus input buffer

Note 1: The alternate EPMP pins are selected when the $\overline{\text{ALTPMP}}$ (CW3<12>) bit is programmed to '0'.

2: The PMSC2 signal will replace the PMA15 signal on the 15-pin PMA when CSF<1:0> = 01 or 10.

3: The PMCS1 signal will replace the PMA14 signal on the 14-pin PMA when CSF<1:0> = 10.

4: The alternate VREF pins selected when the $\overline{\text{ALTVREF}}$ (CW1<5>) bit is programmed to '0'.

TABLE 1-3: PIC24FJ256DA210 FAMILY PINOUT DESCRIPTIONS (CONTINUED)

Function	Pin Number			I/O	Input Buffer	Description
	64-Pin TQFP/QFN	100-Pin TQFP	121-Pin BGA			
TCK	27	38	J6	I	ST	JTAG Test Clock Input.
TDI	28	60	G11	I	ST	JTAG Test Data Input.
TDO	24	61	G9	O	—	JTAG Test Data Output.
TMS	23	17	G3	I	ST	JTAG Test Mode Select Input.
USBID	33	51	K10	I	ST	USB OTG ID (OTG mode only).
USBOEN	12	21	H2	O	—	USB Output Enable Control (for external transceiver).
VBUS	34	54	H8	I	ANA	USB Voltage, Host mode (5V).
VBUSCHG	49	76	A11	O	—	External USB VBUS Charge Output.
VBUSON	11	20	H1	O	—	USB OTG External Charge Pump Control.
VBUSST	58	87	B6	I	ANA	USB OTG Internal Charge Pump Feedback Control.
VBUSVLD	58	87	B6	I	ST	USB VBUS Boost Generator, Comparator Input 1.
VCAP	56	85	B7	P	—	External Filter Capacitor Connection (regulator enabled).
VCMPST1	58	87	B6	I	ST	USB VBUS Boost Generator, Comparator Input 1.
VCMPST2	59	88	A6	I	ST	USB VBUS Boost Generator, Comparator Input 2.
VCPCON	49	76	A11	O	—	USB OTG VBUS PWM/Charge Output.
VDD	10, 26, 38	2, 16, 37, 46, 62	C2, C9, F8, G5, H6, K8, H4, E5	P	—	Positive Supply for Peripheral Digital Logic and I/O Pins.
VMIO	14	23	J2	I	ST	USB Differential Minus Input/Output (external transceiver).
VPIO	13	22	J1	I	ST	USB Differential Plus Input/Output (external transceiver).
VREF-	15	28, 24(4)	L2, K1(4)	I	ANA	A/D and Comparator Reference Voltage (low) Input.
VREF+	16	29, 25(4)	K3, K2(4)	I	ANA	A/D and Comparator Reference Voltage (high) Input.
VSS	9, 25, 41	15, 36, 45, 65, 75	B10, F5, F10, G6, G7, H3, D4, D5	P	—	Ground Reference for Logic and I/O Pins.
VSYNC	1	96	C3	O	—	Graphics Display Vertical Sync Pulse.
VUSB	35	55	H9	P	—	USB Voltage (3.3V).

Legend: TTL = TTL input buffer; ST = Schmitt Trigger input buffer; ANA = Analog level input/output; I^2C™ = I^2C/SMBus input buffer

Note 1: The alternate EPMP pins are selected when the $\overline{\text{ALTPMP}}$ (CW3<12>) bit is programmed to '0'.

2: The PMSC2 signal will replace the PMA15 signal on the 15-pin PMA when CSF<1:0> = 01 or 10.

3: The PMCS1 signal will replace the PMA14 signal on the 14-pin PMA when CSF<1:0> = 10.

4: The alternate VREF pins selected when the $\overline{\text{ALTVREF}}$ (CW1<5>) bit is programmed to '0'.

NOTES:

2.0 GUIDELINES FOR GETTING STARTED WITH 16-BIT MICROCONTROLLERS

2.1 Basic Connection Requirements

Getting started with the PIC24FJ256DA210 family of 16-bit microcontrollers requires attention to a minimal set of device pin connections before proceeding with development.

The following pins must always be connected:

- All VDD and VSS pins
  (see **Section 2.2 "Power Supply Pins"**)
- All AVDD and AVSS pins, regardless of whether or not the analog device features are used
  (see **Section 2.2 "Power Supply Pins"**)
- MCLR pin
  (see **Section 2.3 "Master Clear (MCLR) Pin"**)
- ENVREG/DISVREG and VCAP/VDDCORE pins (PIC24FJ devices only)
  (see **Section 2.4 "Voltage Regulator Pins (ENVREG/DISVREG and VCAP/VDDCORE)"**)

These pins must also be connected if they are being used in the end application:

- PGECx/PGEDx pins used for In-Circuit Serial Programming™ (ICSP™) and debugging purposes
  (see **Section 2.5 "ICSP Pins"**)
- OSCI and OSCO pins when an external oscillator source is used
  (see **Section 2.6 "External Oscillator Pins"**)

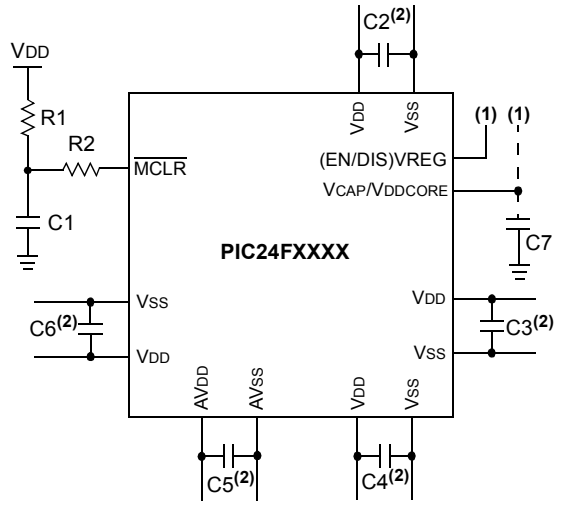
Additionally, the following pins may be required:

- VREF+/VREF- pins used when external voltage reference for analog modules is implemented

Note:	The AVDD and AVSS pins must always be connected, regardless of whether any of the analog modules are being used.

The minimum mandatory connections are shown in Figure 2-1.

FIGURE 2-1: RECOMMENDED MINIMUM CONNECTIONS



Key (all values are recommendations):

C1 through C6: 0.1 µF, 20V ceramic

C7: 10 µF, 6.3V or greater, tantalum or ceramic

R1: 10 kΩ

R2: 100Ω to 470Ω

Note 1: See **Section 2.4 "Voltage Regulator Pins (ENVREG/DISVREG and VCAP/VDDCORE)"** for explanation of ENVREG/DISVREG pin connections.

2: The example shown is for a PIC24F device with five VDD/VSS and AVDD/AVSS pairs. Other devices may have more or less pairs; adjust the number of decoupling capacitors appropriately.

2.2 Power Supply Pins

2.2.1 DECOUPLING CAPACITORS

The use of decoupling capacitors on every pair of power supply pins, such as VDD, VSS, AVDD and AVSS is required.

Consider the following criteria when using decoupling capacitors:

- **Value and type of capacitor:** A 0.1 μF (100 nF), 10-20V capacitor is recommended. The capacitor should be a low-ESR device with a resonance frequency in the range of 200 MHz and higher. Ceramic capacitors are recommended.
- **Placement on the printed circuit board:** The decoupling capacitors should be placed as close to the pins as possible. It is recommended to place the capacitors on the same side of the board as the device. If space is constricted, the capacitor can be placed on another layer on the PCB using a via; however, ensure that the trace length from the pin to the capacitor is no greater than 0.25 inch (6 mm).
- **Handling high-frequency noise:** If the board is experiencing high-frequency noise (upward of tens of MHz), add a second ceramic type capacitor in parallel to the above described decoupling capacitor. The value of the second capacitor can be in the range of 0.01 μF to 0.001 μF. Place this second capacitor next to each primary decoupling capacitor. In high-speed circuit designs, consider implementing a decade pair of capacitances as close to the power and ground pins as possible (e.g., 0.1 μF in parallel with 0.001 μF).
- **Maximizing performance:** On the board layout from the power supply circuit, run the power and return traces to the decoupling capacitors first, and then to the device pins. This ensures that the decoupling capacitors are first in the power chain. Equally important is to keep the trace length between the capacitor and the power pins to a minimum, thereby reducing PCB trace inductance.

2.2.2 TANK CAPACITORS

On boards with power traces running longer than six inches in length, it is suggested to use a tank capacitor for integrated circuits including microcontrollers to supply a local power source. The value of the tank capacitor should be determined based on the trace resistance that connects the power supply source to the device, and the maximum current drawn by the device in the application. In other words, select the tank capacitor so that it meets the acceptable voltage sag at the device. Typical values range from 4.7 μF to 47 μF.

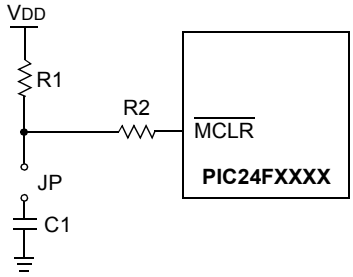
2.3 Master Clear ($\overline{\text{MCLR}}$) Pin

The $\overline{\text{MCLR}}$ pin provides two specific device functions: device Reset, and device programming and debugging. If programming and debugging are not required in the end application, a direct connection to VDD may be all that is required. The addition of other components, to help increase the application's resistance to spurious Resets from voltage sags, may be beneficial. A typical configuration is shown in Figure 2-1. Other circuit designs may be implemented, depending on the application's requirements.

During programming and debugging, the resistance and capacitance that can be added to the pin must be considered. Device programmers and debuggers drive the $\overline{\text{MCLR}}$ pin. Consequently, specific voltage levels (VIH and VIL) and fast signal transitions must not be adversely affected. Therefore, specific values of R1 and C1 will need to be adjusted based on the application and PCB requirements. For example, it is recommended that the capacitor, C1, be isolated from the $\overline{\text{MCLR}}$ pin during programming and debugging operations by using a jumper (Figure 2-2). The jumper is replaced for normal run-time operations.

Any components associated with the $\overline{\text{MCLR}}$ pin should be placed within 0.25 inch (6 mm) of the pin.

FIGURE 2-2: EXAMPLE OF $\overline{\text{MCLR}}$ PIN CONNECTIONS



Note 1: R1 ≤ 10 kΩ is recommended. A suggested starting value is 10 kΩ. Ensure that the $\overline{\text{MCLR}}$ pin VIH and VIL specifications are met.

2: R2 ≤ 470Ω will limit any current flowing into $\overline{\text{MCLR}}$ from the external capacitor, C, in the event of $\overline{\text{MCLR}}$ pin breakdown, due to Electrostatic Discharge (ESD) or Electrical Overstress (EOS). Ensure that the $\overline{\text{MCLR}}$ pin VIH and VIL specifications are met.

2.4 Voltage Regulator Pins (ENVREG/DISVREG and VCAP/VDDCORE)

Note: This section applies only to PIC24FJ devices with an on-chip voltage regulator.

The on-chip voltage regulator enable/disable pin (ENVREG or DISVREG, depending on the device family) must always be connected directly to either a supply voltage or to ground. The particular connection is determined by whether or not the regulator is to be used:

- For ENVREG, tie to VDD to enable the regulator, or to ground to disable the regulator
- For DISVREG, tie to ground to enable the regulator or to VDD to disable the regulator

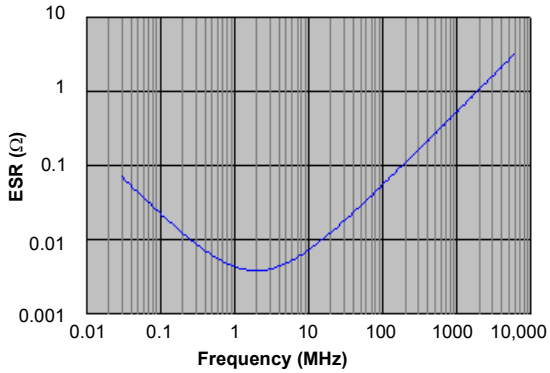
Refer to **Section 27.2 "On-Chip Voltage Regulator"** for details on connecting and using the on-chip regulator.

When the regulator is enabled, a low-ESR (<5Ω) capacitor is required on the VCAP/VDDCORE pin to stabilize the voltage regulator output voltage. The VCAP/VDDCORE pin must not be connected to VDD, and must use a capacitor of 10 μF connected to ground. The type can be ceramic or tantalum. A suitable example is the Murata GRM21BF50J106ZE01 (10 μF, 6.3V) or equivalent. Designers may use Figure 2-3 to evaluate ESR equivalence of candidate devices.

The placement of this capacitor should be close to VCAP/VDDCORE. It is recommended that the trace length not exceed 0.25 inch (6 mm). Refer to **Section 30.0 "Electrical Characteristics"** for additional information.

When the regulator is disabled, the VCAP/VDDCORE pin must be tied to a voltage supply at the VDDCORE level. Refer to **Section 30.0 "Electrical Characteristics"** for information on VDD and VDDCORE.

FIGURE 2-3: FREQUENCY vs. ESR PERFORMANCE FOR SUGGESTED VCAP



Note: Data for Murata GRM21BF50J106ZE01 shown. Measurements at 25°C, 0V DC bias.

2.5 ICSP Pins

The PGECx and PGEDx pins are used for In-Circuit Serial Programming (ICSP) and debugging purposes. It is recommended to keep the trace length between the ICSP connector and the ICSP pins on the device as short as possible. If the ICSP connector is expected to experience an ESD event, a series resistor is recommended, with the value in the range of a few tens of ohms, not to exceed 100Ω.

Pull-up resistors, series diodes and capacitors on the PGECx and PGEDx pins are not recommended as they will interfere with the programmer/debugger communications to the device. If such discrete components are an application requirement, they should be removed from the circuit during programming and debugging. Alternatively, refer to the AC/DC characteristics and timing requirements information in the respective device Flash programming specification for information on capacitive loading limits and pin input voltage high (VIH) and input low (VIL) requirements.

For device emulation, ensure that the "Communication Channel Select" (i.e., PGECx/PGEDx pins) programmed into the device matches the physical connections for the ICSP to the Microchip debugger/emulator tool.

For more information on available Microchip development tools connection requirements, refer to **Section 28.0 "Development Support"**.

2.6 External Oscillator Pins

Many microcontrollers have options for at least two oscillators: a high-frequency primary oscillator and a low-frequency secondary oscillator (refer to **Section 8.0 "Oscillator Configuration"** for details).

The oscillator circuit should be placed on the same side of the board as the device. Place the oscillator circuit close to the respective oscillator pins with no more than 0.5 inch (12 mm) between the circuit components and the pins. The load capacitors should be placed next to the oscillator itself, on the same side of the board.

Use a grounded copper pour around the oscillator circuit to isolate it from surrounding circuits. The grounded copper pour should be routed directly to the MCU ground. Do not run any signal traces or power traces inside the ground pour. Also, if using a two-sided board, avoid any traces on the other side of the board where the crystal is placed.

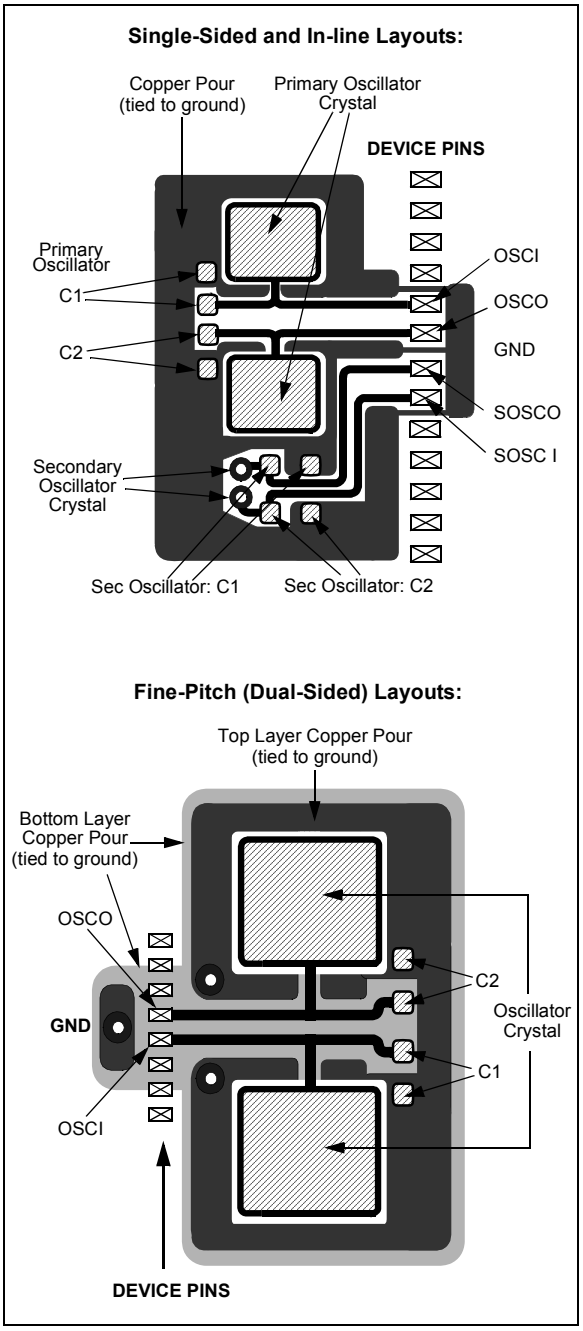
Layout suggestions are shown in Figure 2-4. In-line packages may be handled with a single-sided layout that completely encompasses the oscillator pins. With fine-pitch packages, it is not always possible to completely surround the pins and components. A suitable solution is to tie the broken guard sections to a mirrored ground layer. In all cases, the guard trace(s) must be returned to ground.

In planning the application's routing and I/O assignments, ensure that adjacent port pins and other signals in close proximity to the oscillator are benign (i.e., free of high frequencies, short rise and fall times and other similar noise).

For additional information and design guidance on oscillator circuits, please refer to these Microchip Application Notes, available at the corporate web site (www.microchip.com):

- *AN826, "Crystal Oscillator Basics and Crystal Selection for rfPIC™ and PICmicro® Devices"*
- *AN849, "Basic PICmicro® Oscillator Design"*
- *AN943, "Practical PICmicro® Oscillator Analysis and Design"*
- *AN949, "Making Your Oscillator Work"*

FIGURE 2-4: SUGGESTED PLACEMENT OF THE OSCILLATOR CIRCUIT



2.7 Configuration of Analog and Digital Pins During ICSP Operations

If an ICSP compliant emulator is selected as a debugger, it automatically initializes all of the A/D input pins (ANx) as “digital” pins. Depending on the particular device, this is done by setting all bits in the ADnPCFG register(s), or clearing all bit in the ANSx registers.

All PIC24F devices will have either one or more ADnPCFG registers or several ANSx registers (one for each port); no device will have both. Refer to (**Section 23.0 “10-Bit High-Speed A/D Converter”**) for more specific information.

The bits in these registers that correspond to the A/D pins that initialized the emulator must not be changed by the user application firmware; otherwise, communication errors will result between the debugger and the device.

If your application needs to use certain A/D pins as analog input pins during the debug session, the user application must modify the appropriate bits during initialization of the ADC module, as follows:

- For devices with an ADnPCFG register, clear the bits corresponding to the pin(s) to be configured as analog. Do not change any other bits, particularly those corresponding to the PGECx/PGEDx pair, at any time.
- For devices with ANSx registers, set the bits corresponding to the pin(s) to be configured as analog. Do not change any other bits, particularly those corresponding to the PGECx/PGEDx pair, at any time.

When a Microchip debugger/emulator is used as a programmer, the user application firmware must correctly configure the ADnPCFG or ANSx registers. Automatic initialization of this register is only done during debugger operation. Failure to correctly configure the register(s) will result in all A/D pins being recognized as analog input pins, resulting in the port value being read as a logic '0', which may affect user application functionality.

2.8 Unused I/Os

Unused I/O pins should be configured as outputs and driven to a logic low state. Alternatively, connect a 1 kΩ to 10 kΩ resistor to Vss on unused pins and drive the output to logic low.

NOTES:

3.0 CPU

Note: This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the *"PIC24F Family Reference Manual"*, **Section 44. "CPU with Extended Data Space (EDS)"** (DS39732). The information in this data sheet supersedes the information in the FRM.

The PIC24F CPU has a 16-bit (data) modified Harvard architecture with an enhanced instruction set and a 24-bit instruction word with a variable length opcode field. The Program Counter (PC) is 23 bits wide and addresses up to 4M instructions of user program memory space. A single-cycle instruction prefetch mechanism is used to help maintain throughput and provides predictable execution. All instructions execute in a single cycle, with the exception of instructions that change the program flow, the double-word move (MOV.D) instruction and the table instructions. Overhead-free program loop constructs are supported using the REPEAT instructions, which are interruptible at any point.

PIC24F devices have sixteen, 16-bit working registers in the programmer's model. Each of the working registers can act as a data, address or address offset register. The 16th working register (W15) operates as a Software Stack Pointer for interrupts and calls.

The lower 32 Kbytes of the data space can be accessed linearly. The upper 32 Kbytes of the data space are referred to as extended data space to which the extended data RAM, EPMP memory space or program memory can be mapped.

The Instruction Set Architecture (ISA) has been significantly enhanced beyond that of the PIC18, but maintains an acceptable level of backward compatibility. All PIC18 instructions and addressing modes are supported, either directly, or through simple macros. Many of the ISA enhancements have been driven by compiler efficiency needs.

The core supports Inherent (no operand), Relative, Literal, Memory Direct Addressing modes along with three groups of addressing modes. All modes support Register Direct and various Register Indirect modes. Each group offers up to seven addressing modes. Instructions are associated with predefined addressing modes depending upon their functional requirements.

For most instructions, the core is capable of executing a data (or program data) memory read, a working register (data) read, a data memory write and a program (instruction) memory read per instruction cycle. As a result, three parameter instructions can be supported, allowing trinary operations (that is, A + B = C) to be executed in a single cycle.

A high-speed, 17-bit x 17-bit multiplier has been included to significantly enhance the core arithmetic capability and throughput. The multiplier supports Signed, Unsigned and Mixed mode, 16-bit x 16-bit or 8-bit x 8-bit, integer multiplication. All multiply instructions execute in a single cycle.

The 16-bit ALU has been enhanced with integer divide assist hardware that supports an iterative non-restoring divide algorithm. It operates in conjunction with the REPEAT instruction looping mechanism and a selection of iterative divide instructions to support 32-bit (or 16-bit), divided by 16-bit, integer signed and unsigned division. All divide operations require 19 cycles to complete but are interruptible at any cycle boundary.

The PIC24F has a vectored exception scheme with up to 8 sources of non-maskable traps and up to 118 interrupt sources. Each interrupt source can be assigned to one of seven priority levels.

A block diagram of the CPU is shown in Figure 3-1.

3.1 Programmer's Model

The programmer's model for the PIC24F is shown in Figure 3-2. All registers in the programmer's model are memory mapped and can be manipulated directly by instructions. A description of each register is provided in Table 3-1. All registers associated with the programmer's model are memory mapped.

FIGURE 3-1: PIC24F CPU CORE BLOCK DIAGRAM

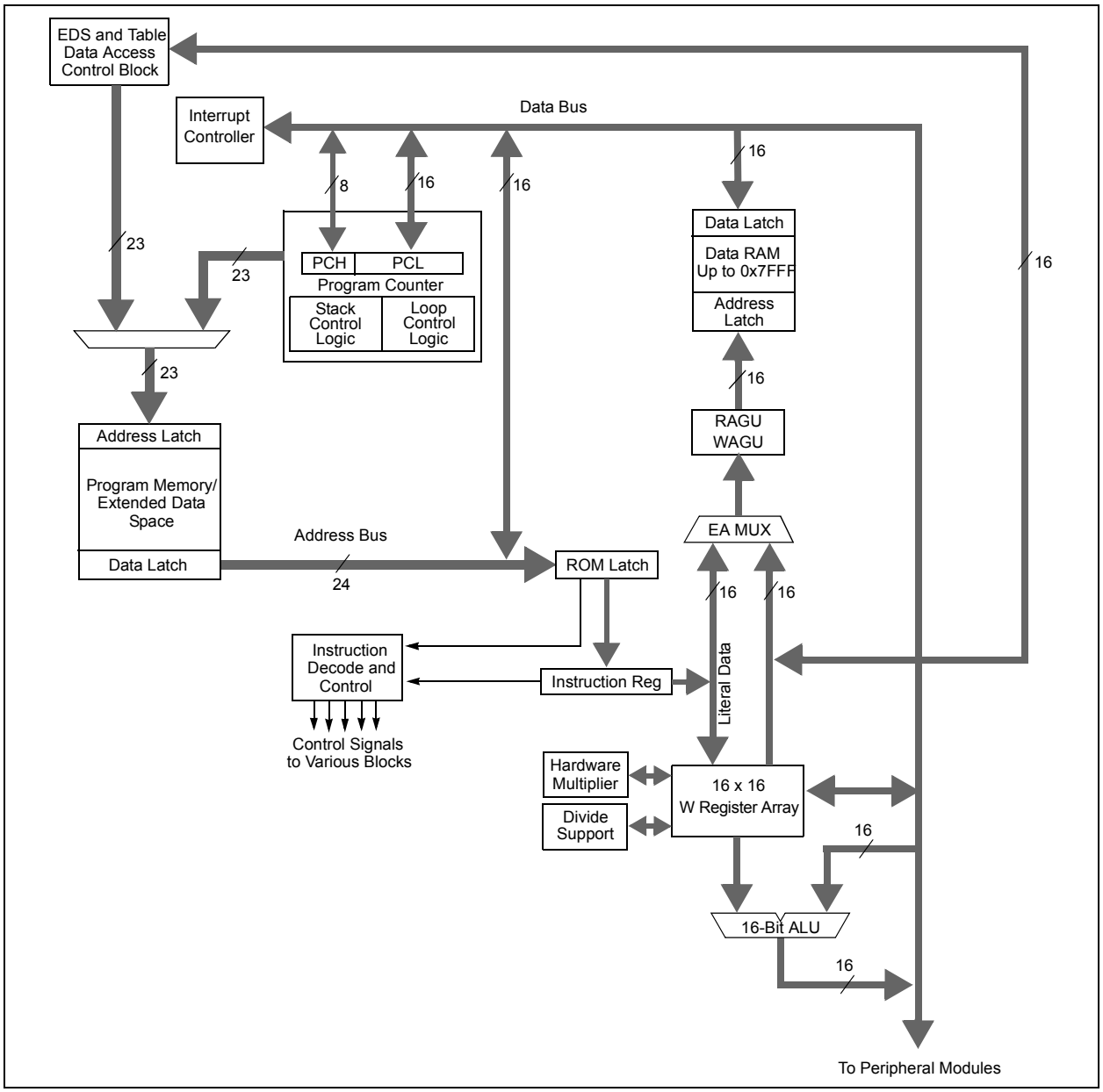
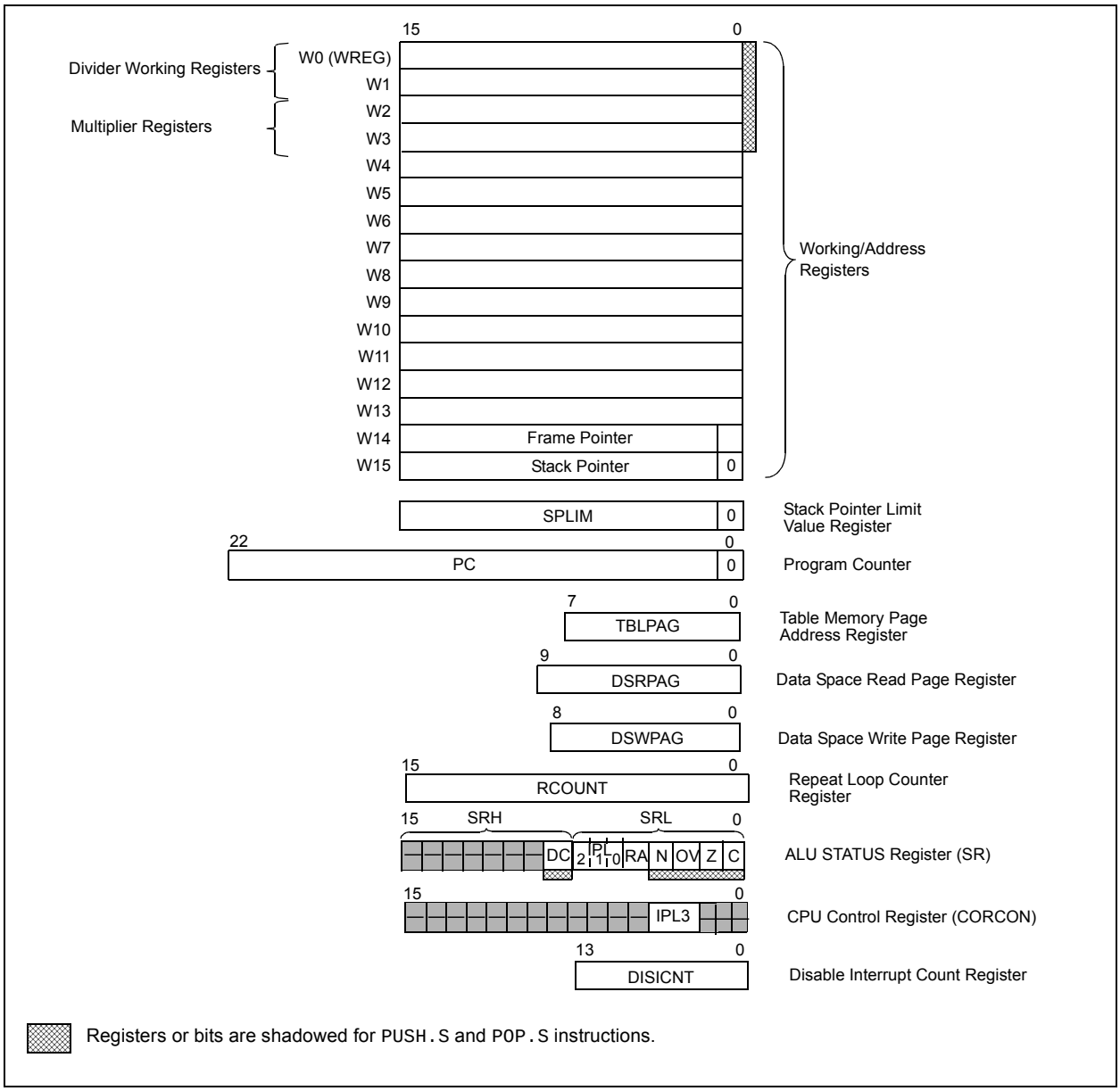


TABLE 3-1: CPU CORE REGISTERS

Register(s) Name	Description
W0 through W15	Working Register Array
PC	23-Bit Program Counter
SR	ALU STATUS Register
SPLIM	Stack Pointer Limit Value Register
TBLPAG	Table Memory Page Address Register
RCOUNT	Repeat Loop Counter Register
CORCON	CPU Control Register
DISICNT	Disable Interrupt Count Register
DSRPAG	Data Space Read Page Register
DSWPAG	Data Space Write Page Register

FIGURE 3-2: PROGRAMMER'S MODEL



3.2 CPU Control Registers

REGISTER 3-1: SR: ALU STATUS REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0, HSC
—	—	—	—	—	—	—	DC
bit 15							bit 8

R/W-0, HSC(1)	R/W-0, HSC(1)	R/W-0, HSC(1)	R-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC
IPL2(2)	IPL1(2)	IPL0(2)	RA	N	OV	Z	C
bit 7							bit 0

Legend:	HSC = Hardware Settable/Clearable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-9 **Unimplemented:** Read as '0'

bit 8 **DC:** ALU Half Carry/Borrow bit
1 = A carry out from the 4th low-order bit (for byte-sized data) or 8th low-order bit (for word-sized data) of the result occurred
0 = No carry out from the 4th or 8th low-order bit of the result has occurred

bit 7-5 **IPL<2:0>:** CPU Interrupt Priority Level Status bits(1,2)
111 = CPU interrupt priority level is 7 (15); user interrupts are disabled
110 = CPU interrupt priority level is 6 (14)
101 = CPU interrupt priority level is 5 (13)
100 = CPU interrupt priority level is 4 (12)
011 = CPU interrupt priority level is 3 (11)
010 = CPU interrupt priority level is 2 (10)
001 = CPU interrupt priority level is 1 (9)
000 = CPU interrupt priority level is 0 (8)

bit 4 **RA:** REPEAT Loop Active bit
1 = REPEAT loop in progress
0 = REPEAT loop not in progress

bit 3 **N:** ALU Negative bit
1 = Result was negative
0 = Result was not negative (zero or positive)

bit 2 **OV:** ALU Overflow bit
1 = Overflow occurred for signed (2's complement) arithmetic in this arithmetic operation
0 = No overflow has occurred

bit 1 **Z:** ALU Zero bit
1 = An operation, which affects the Z bit, has set it at some time in the past
0 = The most recent operation, which affects the Z bit, has cleared it (i.e., a non-zero result)

bit 0 **C:** ALU Carry/$\overline{\text{Borrow}}$ bit
1 = A carry out from the Most Significant bit of the result occurred
0 = No carry out from the Most Significant bit of the result occurred

Note 1: The IPL Status bits are read-only when NSTDIS (INTCON1<15>) = 1.

2: The IPL Status bits are concatenated with the IPL3 (CORCON<3>) bit to form the CPU Interrupt Priority Level (IPL). The value in parentheses indicates the IPL when IPL3 = 1.

REGISTER 3-2: CORCON: CPU CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	R/C-0, HSC	R-1	U-0	U-0
—	—	—	—	IPL3(1)	r	—	—
bit 7							bit 0

Legend:	C = Clearable bit	r = Reserved bit	HSC = Hardware Settable/Clearable bit
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-4 **Unimplemented:** Read as '0'

bit 3 **IPL3:** CPU Interrupt Priority Level Status bit(1)
1 = CPU interrupt priority level is greater than 7
0 = CPU interrupt priority level is 7 or less

bit 2 **Reserved:** Read as '1'

bit 1-0 **Unimplemented:** Read as '0'

Note 1: The IPL3 bit is concatenated with the IPL<2:0> bits (SR<7:5>) to form the CPU interrupt priority level; see Register 3-1 for bit description.

3.3 Arithmetic Logic Unit (ALU)

The PIC24F ALU is 16 bits wide and is capable of addition, subtraction, bit shifts and logic operations. Unless otherwise mentioned, arithmetic operations are 2's complement in nature. Depending on the operation, the ALU may affect the values of the Carry (C), Zero (Z), Negative (N), Overflow (OV) and Digit Carry (DC) Status bits in the SR register. The C and DC Status bits operate as Borrow and Digit Borrow bits, respectively, for subtraction operations.

The ALU can perform 8-bit or 16-bit operations, depending on the mode of the instruction that is used. Data for the ALU operation can come from the W register array, or data memory, depending on the addressing mode of the instruction. Likewise, output data from the ALU can be written to the W register array or a data memory location.

The PIC24F CPU incorporates hardware support for both multiplication and division. This includes a dedicated hardware multiplier and support hardware for 16-bit divisor division.

3.3.1 MULTIPLIER

The ALU contains a high-speed, 17-bit x 17-bit multiplier. It supports unsigned, signed or mixed sign operation in several multiplication modes:

1. 16-bit x 16-bit signed
2. 16-bit x 16-bit unsigned
3. 16-bit signed x 5-bit (literal) unsigned
4. 16-bit unsigned x 16-bit unsigned
5. 16-bit unsigned x 5-bit (literal) unsigned
6. 16-bit unsigned x 16-bit signed
7. 8-bit unsigned x 8-bit unsigned

3.3.2 DIVIDER

The divide block supports signed and unsigned integer divide operations with the following data sizes:

1. 32-bit signed/16-bit signed divide
2. 32-bit unsigned/16-bit unsigned divide
3. 16-bit signed/16-bit signed divide
4. 16-bit unsigned/16-bit unsigned divide

The quotient for all divide instructions ends up in W0 and the remainder in W1. 16-bit signed and unsigned DIV instructions can specify any W register for both the 16-bit divisor (Wn), and any W register (aligned) pair (W(m + 1):Wm) for the 32-bit dividend. The divide algorithm takes one cycle per bit of divisor, so both 32-bit/16-bit and 16-bit/16-bit instructions take the same number of cycles to execute.

3.3.3 MULTI-BIT SHIFT SUPPORT

The PIC24F ALU supports both single bit and single-cycle, multi-bit arithmetic and logic shifts. Multi-bit shifts are implemented using a shifter block, capable of performing up to a 15-bit arithmetic right shift, or up to a 15-bit left shift, in a single cycle. All multi-bit shift instructions only support Register Direct Addressing for both the operand source and result destination.

A full summary of instructions that use the shift operation is provided in Table 3-2.

TABLE 3-2: INSTRUCTIONS THAT USE THE SINGLE BIT AND MULTI-BIT SHIFT OPERATION

Instruction	Description
ASR	Arithmetic shift right source register by one or more bits.
SL	Shift left source register by one or more bits.
LSR	Logical shift right source register by one or more bits.

4.0 MEMORY ORGANIZATION

As Harvard architecture devices, PIC24F microcontrollers feature separate program and data memory spaces and busses. This architecture also allows direct access of program memory from the data space during code execution.

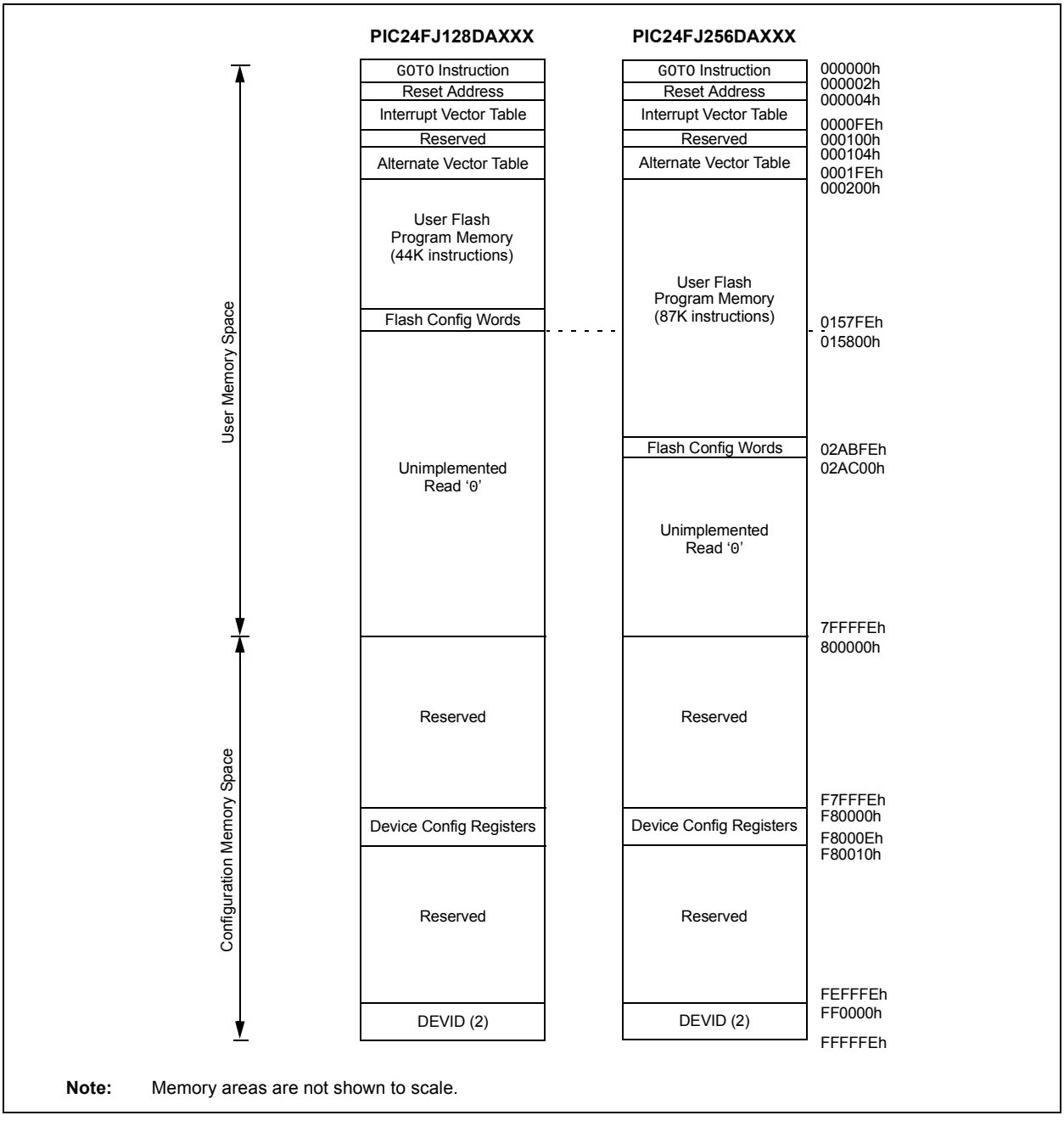
4.1 Program Memory Space

The program address memory space of the PIC24FJ256DA210 family devices is 4M instructions. The space is addressable by a 24-bit value derived from either the 23-bit Program Counter (PC) during program execution, or from table operation or data space remapping, as described in **Section 4.3 "Interfacing Program and Data Memory Spaces"**.

User access to the program memory space is restricted to the lower half of the address range (000000h to 7FFFFFh). The exception is the use of TBLRD/TBLWT operations, which use TBLPAG<7> to permit access to the Configuration bits and Device ID sections of the configuration memory space.

Memory maps for the PIC24FJ256DA210 family of devices are shown in Figure 4-1.

FIGURE 4-1: PROGRAM SPACE MEMORY MAP FOR PIC24FJ256DA210 FAMILY DEVICES



Note: Memory areas are not shown to scale.

4.1.1 PROGRAM MEMORY ORGANIZATION

The program memory space is organized in word-addressable blocks. Although it is treated as 24 bits wide, it is more appropriate to think of each address of the program memory as a lower and upper word, with the upper byte of the upper word being unimplemented. The lower word always has an even address, while the upper word has an odd address (Figure 4-2).

Program memory addresses are always word-aligned on the lower word and addresses are incremented or decremented by two during code execution. This arrangement also provides compatibility with data memory space addressing and makes it possible to access data in the program memory space.

4.1.2 HARD MEMORY VECTORS

All PIC24F devices reserve the addresses between 0x00000 and 0x000200 for hard coded program execution vectors. A hardware Reset vector is provided to redirect code execution from the default value of the PC on device Reset to the actual start of code. A GOTO instruction is programmed by the user at 0x000000 with the actual address for the start of code at 0x000002.

PIC24F devices also have two interrupt vector tables, located from 0x000004 to 0x0000FF and 0x000100 to 0x0001FF. These vector tables allow each of the many device interrupt sources to be handled by separate ISRs. A more detailed discussion of the interrupt vector tables is provided in **Section 7.1 "Interrupt Vector Table"**.

4.1.3 FLASH CONFIGURATION WORDS

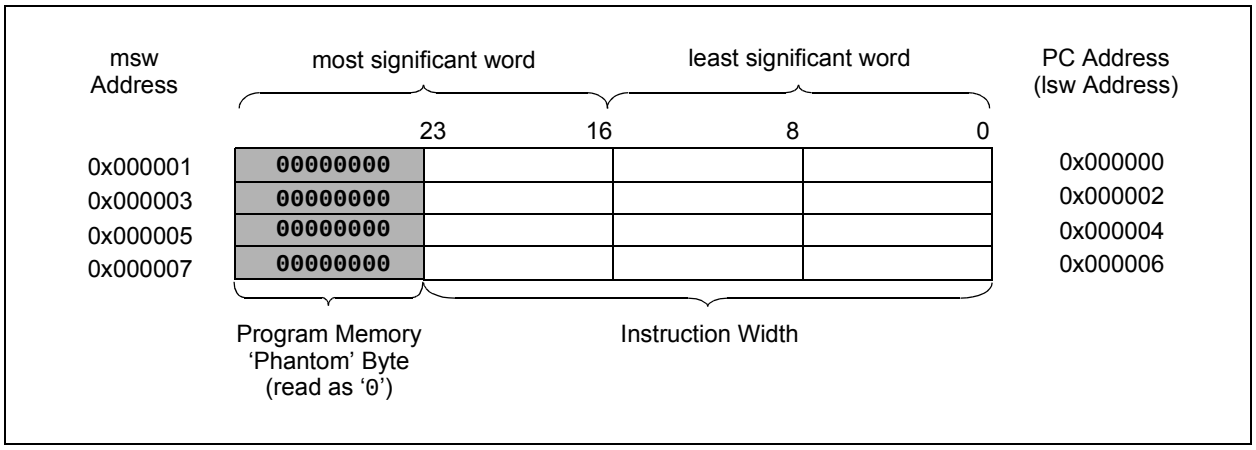
In PIC24FJ256DA210 family devices, the top four words of on-chip program memory are reserved for configuration information. On device Reset, the configuration information is copied into the appropriate Configuration register. The addresses of the Flash Configuration Word for devices in the PIC24FJ256DA210 family are shown in Table 4-1. Their location in the memory map is shown with the other memory vectors in Figure 4-1.

The Configuration Words in program memory are a compact format. The actual Configuration bits are mapped in several different registers in the configuration memory space. Their order in the Flash Configuration Words does not reflect a corresponding arrangement in the configuration space. Additional details on the device Configuration Words are provided in **Section 27.1 "Configuration Bits"**.

TABLE 4-1: FLASH CONFIGURATION WORDS FOR PIC24FJ256DA210 FAMILY DEVICES

Device	Program Memory (Words)	Configuration Word Addresses
PIC24FJ128DAXXX	44,032	0x0157F8:0x0157FE
PIC24FJ256DAXXX	87,552	0x02ABF8:0x02ABFE

FIGURE 4-2: PROGRAM MEMORY ORGANIZATION



4.2 Data Memory Space

Note: This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the "*PIC24F Family Reference Manual*", **Section 45. "Data Memory with Extended Data Space**" (DS39733). The information in this data sheet supersedes the information in the FRM.

The PIC24F core has a 16-bit wide data memory space, addressable as a single linear range.

The data space is accessed using two Address Generation Units (AGUs), one each for read and write operations. The data space memory map is shown in Figure 4-3.

The 16-bit wide data addresses in the data memory space point to bytes within the Data Space (DS). This gives a DS address range of 64 Kbytes or 32K words. The lower 32 Kbytes (0x0000 to 0x7FFF) of DS is compatible with the PIC24F microcontrollers without EDS.

The upper 32 Kbytes of data memory address space (0x8000 - 0xFFFF) are used as an EDS window.

The EDS window is used to access all memory region implemented in EDS, as shown in Figure 4-4.

The EDS includes any additional internal data memory not accessible by the lower 32-Kbyte data address space and any external memory through EPMP. For more details on accessing internal extended data memory, refer to the "*PIC24F Family Reference Manual*", **Section 45. "Data Memory with Extended Data Space (EDS)"** (DS39733). For more details on accessing external memory using EPMP, refer to the "*PIC24F Family Reference Manual*", **Section 42. "Enhanced Parallel Master Port (EPMP)"** (DS39730). In PIC24F microcontrollers with EDS, the program memory can also be read from EDS. This is called Program Space Visibility (PSV). Table 4-2 lists the total memory accessible by each of the devices in this family.

The EDS is organized as pages, with a single page called an EDS page that equals the EDS window (32 Kbytes). A particular EDS page is selected through the Data Space Read register (DSRPAG) or Data Space Write register (DSWPAG). For PSV, only the DSRPAG register is used. The combination of the DSRPAG register value and the 16-bit wide data address forms a 24-bit Effective Address (EA). For more information on EDS, refer to **Section 4.3.3 "Reading Data from Program Memory Using EDS"**.

TABLE 4-2: TOTAL MEMORY ACCESSIBLE BY THE DEVICE

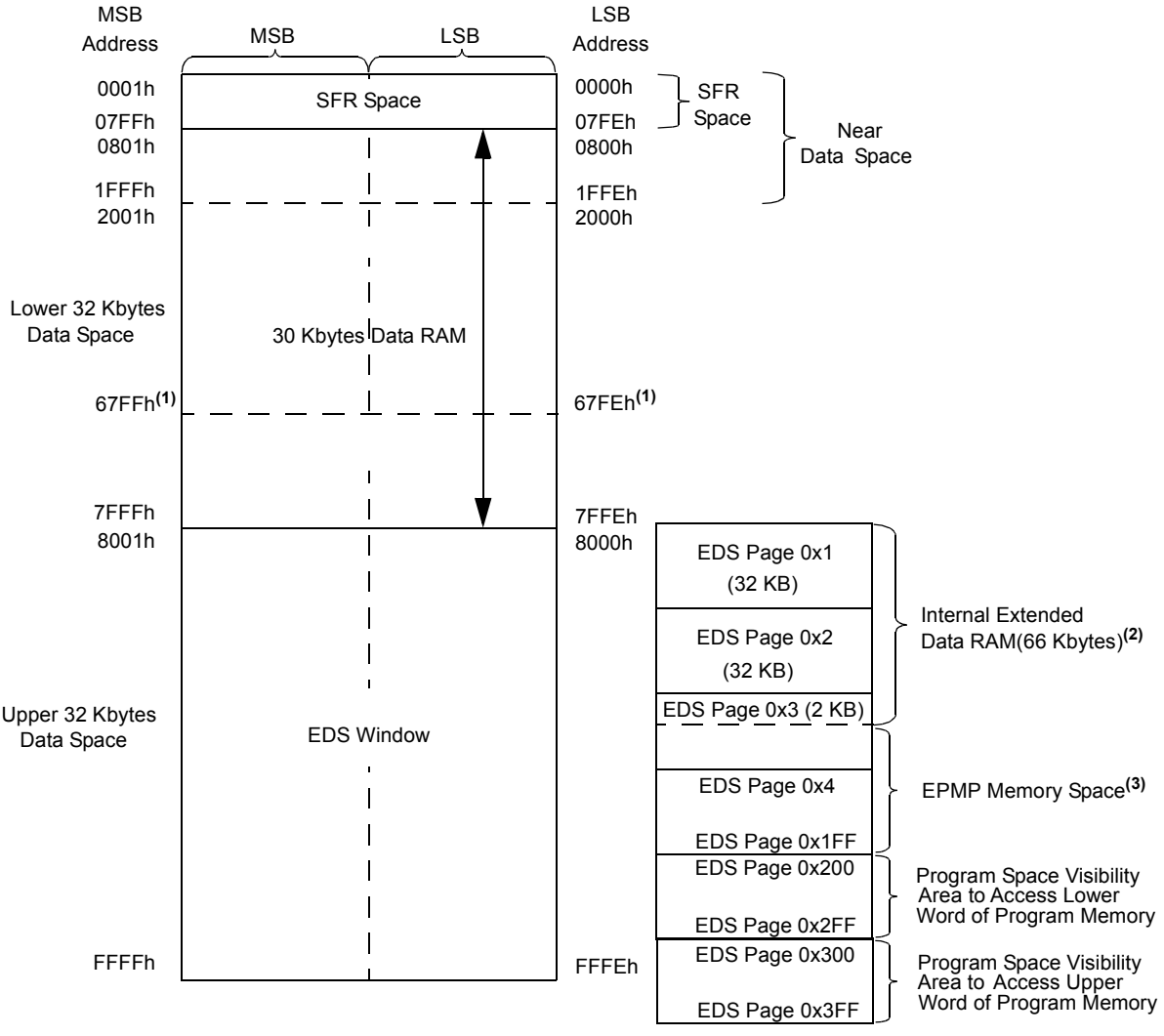
Devices	Internal RAM	External RAM Access Using EPMP	Program Memory Access Using EDS
PIC24FJXXXDA210	96 Kbytes (30K + 66K(1))	Yes (up to 16 MB)	Yes
PIC24FJXXXDA206	96 Kbytes (30K + 66K(1))	No	Yes
PIC24FJXXXDA110	24 Kbytes	Yes (up to 16 MB)	Yes
PIC24FJXXXDA106	24 Kbytes	No	Yes

Note 1: The internal RAM above 30 Kbytes can be accessed through EDS window.

4.2.1 DATA SPACE WIDTH

The data memory space is organized in byte-addressable, 16-bit wide blocks. Data is aligned in data memory and registers as 16-bit words, but all data space EAs resolve to bytes. The Least Significant Bytes (LSBs) of each word have even addresses, while the Most Significant Bytes (MSBs) have odd addresses.

FIGURE 4-3: DATA SPACE MEMORY MAP FOR PIC24FJ256DA210 FAMILY DEVICES(4)



Note 1: 24-Kbyte RAM variants (PIC24FJXXXDA1XX devices have implemented RAM only up to 0x67FF).
2: Valid only for 96-Kbyte RAM variants (PIC24FJXXXDA2XX).
3: Valid only for variants with the EPMP module (PIC24FJXXXDAX10).
4: Data memory areas are not shown to scale.

4.2.2 DATA MEMORY ORGANIZATION AND ALIGNMENT

To maintain backward compatibility with PIC® MCUs and improve data space memory usage efficiency, the PIC24F instruction set supports both word and byte operations. As a consequence of byte accessibility, all EA calculations are internally scaled to step through word-aligned memory. For example, the core recognizes that Post-Modified Register Indirect Addressing mode [Ws++] will result in a value of Ws + 1 for byte operations and Ws + 2 for word operations.

Data byte reads will read the complete word, which contains the byte, using the LSB of any EA to determine which byte to select. The selected byte is placed onto the LSB of the data path. That is, data memory and registers are organized as two parallel, byte-wide entities with shared (word) address decode but separate write lines. Data byte writes only write to the corresponding side of the array or register which matches the byte address.

All word accesses must be aligned to an even address. Misaligned word data fetches are not supported, so care must be taken when mixing byte and word operations or translating from 8-bit MCU code. If a misaligned read or write is attempted, an address error trap will be generated. If the error occurred on a read, the instruction underway is completed; if it occurred on a write, the instruction will be executed but the write will not occur. In either case, a trap is then executed, allowing the system and/or user to examine the machine state prior to execution of the address Fault.

All byte loads into any W register are loaded into the LSB. The Most Significant Byte (MSB) is not modified.

A sign-extend instruction (SE) is provided to allow users to translate 8-bit signed data to 16-bit signed values. Alternatively, for 16-bit unsigned data, users can clear the MSB of any W register by executing a zero-extend (ZE) instruction on the appropriate address.

Although most instructions are capable of operating on word or byte data sizes, it should be noted that some instructions operate only on words.

4.2.3 NEAR DATA SPACE

The 8-Kbyte area between 0000h and 1FFFh is referred to as the near data space. Locations in this space are directly addressable via a 13-bit absolute address field within all memory direct instructions. The remainder of the data space is addressable indirectly. Additionally, the whole data space is addressable using MOV instructions, which support Memory Direct Addressing with a 16-bit address field.

4.2.4 SPECIAL FUNCTION REGISTER (SFR) SPACE

The first 2 Kbytes of the near data space, from 0000h to 07FFh, are primarily occupied with Special Function Registers (SFRs). These are used by the PIC24F core and peripheral modules for controlling the operation of the device.

SFRs are distributed among the modules that they control and are generally grouped together by module. Much of the SFR space contains unused addresses; these are read as '0'. A diagram of the SFR space, showing where the SFRs are actually implemented, is shown in Table 4-3. Each implemented area indicates a 32-byte region where at least one address is implemented as an SFR. A complete list of implemented SFRs, including their addresses, is shown in Tables 4-4 throughTable 4-34.

TABLE 4-3: IMPLEMENTED REGIONS OF SFR DATA SPACE

	SFR Space Address							
	xx00	**xx20**	**xx40**	**xx60**	**xx80**	**xxA0**	**xxC0**	**xxE0**
000h	Core			ICN	Interrupts			
100h	Timers		Capture		Compare			
200h	I²C™	UART	SPI/UART	SPI/I²C	SPI	UART	I/O	
300h	ADC/CTMU			—	—	—	—	—
400h	—	—	—	—	USB			ANSEL
500h	—	—	—	—	—	—	—	—
600h	EPMP	RTC/Comp	CRC	—	PPS			—
700h	GFX Controller		System	NVM/PMD	—	—	—	—

Legend: — = No implemented SFRs in this block

TABLE 4-4: CPU CORE REGISTERS MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
WREG0	0000	Working Register 0																0000
WREG1	0002	Working Register 1																0000
WREG2	0004	Working Register 2																0000
WREG3	0006	Working Register 3																0000
WREG4	0008	Working Register 4																0000
WREG5	000A	Working Register 5																0000
WREG6	000C	Working Register 6																0000
WREG7	000E	Working Register 7																0000
WREG8	0010	Working Register 8																0000
WREG9	0012	Working Register 9																0000
WREG10	0014	Working Register 10																0000
WREG11	0016	Working Register 11																0000
WREG12	0018	Working Register 12																0000
WREG13	001A	Working Register 13																0000
WREG14	001C	Working Register 14																0000
WREG15	001E	Working Register 15																0800
SPLIM	0020	Stack Pointer Limit Value Register																xxxx
PCL	002E	Program Counter Low Word Register																0000
PCH	0030	—	—	—	—	—	—	—	—	Program Counter Register High Byte								0000
DSRPAG	0032	—	—	—	—	—	—	Extended Data Space Read Page Address Register										0001
DSWPAG(1)	0034	—	—	—	—	—	—	—	Extended Data Space Write Page Address Register									0001
RCOUNT	0036	Repeat Loop Counter Register																xxxx
SR	0042	—	—	—	—	—	—	—	DC	IPL2	IPL1	IPL0	RA	N	OV	Z	C	0000
CORCON	0044	—	—	—	—	—	—	—	—	—	—	—	—	IPL3	r	—	—	0004
DISICNT	0052	—	—	Disable Interrupts Counter Register														xxxx
TBLPAG	0054	—	—	—	—	—	—	—	—	Table Memory Page Address Register								0000

Legend: — = unimplemented, read as '0'; r = Reserved bit. Reset values are shown in hexadecimal.
Note 1: Reserved in PIC24FJXXXDA106 devices; do not use.

TABLE 4-5: ICN REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
CNPD1	0056	CN15PDE	CN14PDE	CN13PDE	CN12PDE	CN11PDE	CN10PDE	CN9PDE	CN8PDE	CN7PDE	CN6PDE	CN5PDE	CN4PDE	CN3PDE	CN2PDE	CN1PDE	CN0PDE	0000
CNPD2	0058	CN31PDE	CN30PDE	CN29PDE	CN28PDE	CN27PDE	CN26PDE	CN25PDE	CN24PDE	CN23PDE	CN22PDE	CN21PDE(1)	CN20PDE(1)	CN19PDE(1)	CN18PDE	CN17PDE	CN16PDE	0000
CNPD3	005A	CN47PDE(1)	CN46PDE(1)	CN45PDE(1)	CN44PDE(1)	CN43PDE(1)	CN42PDE(1)	CN41PDE(1)	CN40PDE(1)	CN39PDE(1)	CN38PDE(1)	CN37PDE(1)	CN36PDE(1)	CN35PDE(1)	CN34PDE(1)	CN33PDE(1)	CN32PDE	0000
CNPD4	005C	CN63PDE	CN62PDE	CN61PDE	CN60PDE	CN59PDE	CN58PDE	CN57PDE(1)	CN56PDE	CN55PDE	CN54PDE	CN53PDE	CN52PDE	CN51PDE	CN50PDE	CN49PDE	CN48PDE(1)	0000
CNPD5	005E	CN79PDE(1)	CN78PDE(1)	CN77PDE(1)	CN76PDE(1)	CN75PDE(1)	CN74PDE(1)	CN73PDE(1)	—	CN71PDE	CN70PDE(1)	CN69PDE	CN68PDE	CN67PDE(1)	CN66PDE(1)	CN65PDE	CN64PDE	0000
CNPD6	0060	—	—	—	—	—	—	—	—	—	—	—	CN84PDE	CN83PDE	CN82PDE(1)	CN81PDE(1)	CN80PDE(1)	0000
CNEN1	0062	CN15IE	CN14IE	CN13IE	CN12IE	CN11IE	CN10IE	CN9IE	CN8IE	CN7IE	CN6IE	CN5IE	CN4IE	CN3IE	CN2IE	CN1IE	CN0IE	0000
CNEN2	0064	CN31IE	CN30IE	CN29IE	CN28IE	CN27IE	CN26IE	CN25IE	CN24IE	CN23IE	CN22IE	CN21IE(1)	CN20IE(1)	CN19IE(1)	CN18IE	CN17IE	CN16IE	0000
CNEN3	0066	CN47IE(1)	CN46IE(1)	CN45IE(1)	CN44IE(1)	CN43IE(1)	CN42IE(1)	CN41IE(1)	CN40IE(1)	CN39IE(1)	CN38IE(1)	CN37IE(1)	CN36IE(1)	CN35IE(1)	CN34IE(1)	CN33IE(1)	CN32IE	0000
CNEN4	0068	CN63IE	CN62IE	CN61IE	CN60IE	CN59IE	CN58IE	CN57IE(1)	CN56IE	CN55IE	CN54IE	CN53IE	CN52IE	CN51IE	CN50IE	CN49IE	CN48IE(1)	0000
CNEN5	006A	CN79IE(1)	CN78IE(1)	CN77IE(1)	CN76IE(1)	CN75IE(1)	CN74IE(1)	CN73IE(1)	—	CN71IE	CN70IE(1)	CN69IE	CN68IE	CN67IE(1)	CN66IE(1)	CN65IE	CN64IE	0000
CNEN6	006C	—	—	—	—	—	—	—	—	—	—	—	CN84IE	CN83IE	CN82IE(1)	CN81IE(1)	CN80IE(1)	0000
CNPU1	006E	CN15PUE	CN14PUE	CN13PUE	CN12PUE	CN11PUE	CN10PUE	CN9PUE	CN8PUE	CN7PUE	CN6PUE	CN5PUE	CN4PUE	CN3PUE	CN2PUE	CN1PUE	CN0PUE	0000
CNPU2	0070	CN31PUE	CN30PUE	CN29PUE	CN28PUE	CN27PUE	CN26PUE	CN25PUE	CN24PUE	CN23PUE	CN22PUE	CN21PUE(1)	CN20PUE(1)	CN19PUE(1)	CN18PUE	CN17PUE	CN16PUE	0000
CNPU3	0072	CN47PUE(1)	CN46PUE(1)	CN45PUE(1)	CN44PUE(1)	CN43PUE(1)	CN42PUE(1)	CN41PUE(1)	CN40PUE(1)	CN39PUE(1)	CN38PUE(1)	CN37PUE(1)	CN36PUE(1)	CN35PUE(1)	CN34PUE(1)	CN33PUE(1)	CN32PUE	0000
CNPU4	0074	CN63PUE	CN62PUE	CN61PUE	CN60PUE	CN59PUE	CN58PUE	CN57PUE(1)	CN56PUE	CN55PUE	CN54PUE	CN53PUE	CN52PUE	CN51PUE	CN50PUE	CN49PUE	CN48PUE(1)	0000
CNPU5	0076	CN79PUE(1)	CN78PUE(1)	CN77PUE(1)	CN76PUE(1)	CN75PUE(1)	CN74PUE(1)	CN73PUE(1)	—	CN71PUE	CN70PUE(1)	CN69PUE	CN68PUE	CN67PUE(1)	CN66PUE(1)	CN65PUE	CN64PUE	0000
CNPU6	0078	—	—	—	—	—	—	—	—	—	—	—	CN84PUE	CN83PUE	CN82PUE(1)	CN81PUE(1)	CN80PUE(1)	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

Note 1: Unimplemented in 64-pin devices; read as '0'.

TABLE 4-6: INTERRUPT CONTROLLER REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
INTCON1	0080	NSTDIS	—	—	—	—	—	—	—	—	—	—	MATHERR	ADDRERR	STKERR	OSCFAIL	—	0000
INTCON2	0082	ALTIVT	DISI	—	—	—	—	—	—	—	—	—	INT4EP	INT3EP	INT2EP	INT1EP	INT0EP	0000
IFS0	0084	—	—	AD1IF	U1TXIF	U1RXIF	SPI1IF	SPF1IF	T3IF	T2IF	OC2IF	IC2IF	—	T1IF	OC1IF	IC1IF	INT0IF	0000
IFS1	0086	U2TXIF	U2RXIF	INT2IF	T5IF	T4IF	OC4IF	OC3IF	—	IC8IF	IC7IF	—	INT1IF	CNIF	CMIF	MI2C1IF	SI2C1IF	0000
IFS2	0088	—	—	PMPIF(1)	OC8IF	OC7IF	OC6IF	OC5IF	IC6IF	IC5IF	IC4IF	IC3IF	—	—	—	SPI2IF	SPF2IF	0000
IFS3	008A	—	RTCIF	—	—	—	—	—	—	—	INT4IF	INT3IF	—	—	MI2C2IF	SI2C2IF	—	0000
IFS4	008C	—	—	CTMUIF	—	—	—	—	LVDIF	—	—	—	—	CRCIF	U2ERIF	U1ERIF	—	0000
IFS5	008E	—	—	IC9IF	OC9IF	SPI3IF	SPF3IF	U4TXIF	U4RXIF	U4ERIF	USB1IF	MI2C3IF	SI2C3IF	U3TXIF	U3RXIF	U3ERIF	—	0000
IFS6	0090	—	—	—	—	—	—	—	—	—	—	—	GFX1IF	—	—	—	—	0000
IEC0	0094	—	—	AD1IE	U1TXIE	U1RXIE	SPI1IE	SPF1IE	T3IE	T2IE	OC2IE	IC2IE	—	T1IE	OC1IE	IC1IE	INT0IE	0000
IEC1	0096	U2TXIE	U2RXIE	INT2IE	T5IE	T4IE	OC4IE	OC3IE	—	IC8IE	IC7IE	—	INT1IE	CNIE	CMIE	MI2C1IE	SI2C1IE	0000
IEC2	0098	—	—	PMPIE(1)	OC8IE	OC7IE	OC6IE	OC5IE	IC6IE	IC5IE	IC4IE	IC3IE	—	—	—	SPI2IE	SPF2IE	0000
IEC3	009A	—	RTCIE	—	—	—	—	—	—	—	INT4IE	INT3IE	—	—	MI2C2IE	SI2C2IE	—	0000
IEC4	009C	—	—	CTMUIE	—	—	—	—	LVDIE	—	—	—	—	CRCIE	U2ERIE	U1ERIE	—	0000
IEC5	009E	—	—	IC9IE	OC9IE	SPI3IE	SPF3IE	U4TXIE	U4RXIE	U4ERIE	USB1IE	MI2C3IE	SI2C3IE	U3TXIE	U3RXIE	U3ERIE	—	0000
IEC6	00A0	—	—	—	—	—	—	—	—	—	—	—	GFX1IE	—	—	—	—	0000
IPC0	00A4	—	T1IP2	T1IP1	T1IP0	—	OC1IP2	OC1IP1	OC1IP0	—	IC1IP2	IC1IP1	IC1IP0	—	INT0IP2	INT0IP1	INT0IP0	4444
IPC1	00A6	—	T2IP2	T2IP1	T2IP0	—	OC2IP2	OC2IP1	OC2IP0	—	IC2IP2	IC2IP1	IC2IP0	—	—	—	—	4440
IPC2	00A8	—	U1RXIP2	U1RXIP1	U1RXIP0	—	SPI1IP2	SPI1IP1	SPI1IP0	—	SPF1IP2	SPF1IP1	SPF1IP0	—	T3IP2	T3IP1	T3IP0	4444
IPC3	00AA	—	—	—	—	—	—	—	—	—	AD1IP2	AD1IP1	AD1IP0	—	U1TXIP2	U1TXIP1	U1TXIP0	0044
IPC4	00AC	—	CNIP2	CNIP1	CNIP0	—	CMIP2	CMIP1	CMIP0	—	MI2C1IP2	MI2C1IP1	MI2C1IP0	—	SI2C1IP2	SI2C1IP1	SI2C1IP0	4444
IPC5	00AE	—	IC8IP2	IC8IP1	IC8IP0	—	IC7IP2	IC7IP1	IC7IP0	—	—	—	—	—	INT1IP2	INT1IP1	INT1IP0	4404
IPC6	00B0	—	T4IP2	T4IP1	T4IP0	—	OC4IP2	OC4IP1	OC4IP0	—	OC3IP2	OC3IP1	OC3IP0	—	—	—	—	4440
IPC7	00B2	—	U2TXIP2	U2TXIP1	U2TXIP0	—	U2RXIP2	U2RXIP1	U2RXIP0	—	INT2IP2	INT2IP1	INT2IP0	—	T5IP2	T5IP1	T5IP0	4444
IPC8	00B4	—	—	—	—	—	—	—	—	—	SPI2IP2	SPI2IP1	SPI2IP0	—	SPF2IP2	SPF2IP1	SPF2IP0	0044
IPC9	00B6	—	IC5IP2	IC5IP1	IC5IP0	—	IC4IP2	IC4IP1	IC4IP0	—	IC3IP2	IC3IP1	IC3IP0	—	—	—	—	4440
IPC10	00B8	—	OC7IP2	OC7IP1	OC7IP0	—	OC6IP2	OC6IP1	OC6IP0	—	OC5IP2	OC5IP1	OC5IP0	—	IC6IP2	IC6IP1	IC6IP0	4444
IPC11	00BA	—	—	—	—	—	—	—	—	—	PMPIP2(1)	PMPIP1(1)	PMPIP0(1)	—	OC8IP2	OC8IP1	OC8IP0	0044(2)
IPC12	00BC	—	—	—	—	—	MI2C2IP2	MI2C2IP1	MI2C2IP0	—	SI2C2IP2	SI2C2IP1	SI2C2IP0	—	—	—	—	0440
IPC13	00BE	—	—	—	—	—	INT4IP2	INT4IP1	INT4IP0	—	INT3IP2	INT3IP1	INT3IP0	—	—	—	—	0440
IPC15	00C2	—	—	—	—	—	RTCIP2	RTCIP1	RTCIP0	—	—	—	—	—	—	—	—	0400

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

Note 1: Unimplemented in 64-pin devices, read as '0'.

2: The Reset value in 64-pin devices are '0004'.

TABLE 4-6: INTERRUPT CONTROLLER REGISTER MAP (CONTINUED)

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
IPC16	00C4	—	CRCIP2	CRCIP1	CRCIP0	—	U2ERIP2	U2ERIP1	U2ERIP0	—	U1ERIP2	U1ERIP1	U1ERIP0	—	—	—	—	4440
IPC18	00C8	—	—	—	—	—	—	—	—	—	—	—	—	—	LVDIP2	LVDIP1	LVDIP0	0004
IPC19	00CA	—	—	—	—	—	—	—	—	—	CTMUIP2	CTMUIP1	CTMUIP0	—	—	—	—	0040
IPC20	00CC	—	U3TXIP2	U3TXIP1	U3TXIP0	—	U3RXIP2	U3RXIP1	U3RXIP0	—	U3ERIP2	U3ERIP1	U3ERIP0	—	—	—	—	4440
IPC21	00CE	—	U4ERIP2	U4ERIP1	U4ERIP0	—	USB1IP2	USB1IP1	USB1IP0	—	MI2C3IP2	MI2C3IP1	MI2C3IP0	—	SI2C3IP2	SI2C3IP1	SI2C3IP0	4444
IPC22	00D0	—	SPI3IP2	SPI3IP1	SPI3IP0	—	SPF3IP2	SPF3IP1	SPF3IP0	—	U4TXIP2	U4TXIP1	U4TXIP0	—	U4RXIP2	U4RXIP1	U4RXIP0	4444
IPC23	00D2	—	—	—	—	—	—	—	—	—	IC9IP2	IC9IP1	IC9IP0	—	OC9IP2	OC9IP1	OC9IP0	0044
IPC25	00D6	—	—	—	—	—	—	—	—	—	—	—	—	—	GFX1IP2	GFX1IP1	GFX1IP0	0004
INTTREG	00E0	CPUIRQ	—	VHOLD	—	ILR3	ILR2	ILR1	ILR0	—	VECNUM6	VECNUM5	VECNUM4	VECNUM3	VECNUM2	VECNUM1	VECNUM0	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.
Note 1: Unimplemented in 64-pin devices, read as '0'.
2: The Reset value in 64-pin devices are '0004'.

TABLE 4-7: TIMER REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
TMR1	0100	Timer1 Register																0000
PR1	0102	Timer1 Period Register																FFFF
T1CON	0104	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	TSYNC	TCS	—	0000
TMR2	0106	Timer2 Register																0000
TMR3HLD	0108	Timer3 Holding Register (for 32-bit timer operations only)																0000
TMR3	010A	Timer3 Register																0000
PR2	010C	Timer2 Period Register																FFFF
PR3	010E	Timer3 Period Register																FFFF
T2CON	0110	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	T32	—	TCS	—	0000
T3CON	0112	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	—	TCS	—	0000
TMR4	0114	Timer4 Register																0000
TMR5HLD	0116	Timer5 Holding Register (for 32-bit operations only)																0000
TMR5	0118	Timer5 Register																0000
PR4	011A	Timer4 Period Register																FFFF
PR5	011C	Timer5 Period Register																FFFF
T4CON	011E	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	T45	—	TCS	—	0000
T5CON	0120	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	—	TCS	—	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-8: INPUT CAPTURE REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
IC1CON1	0140	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC1CON2	0142	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D
IC1BUF	0144	Input Capture 1 Buffer Register																0000
IC1TMR	0146	Input Capture 1 Timer Value Register																xxxx
IC2CON1	0148	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC2CON2	014A	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D
IC2BUF	014C	Input Capture 2 Buffer Register																0000
IC2TMR	014E	Input Capture 2 Timer Value Register																xxxx
IC3CON1	0150	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC3CON2	0152	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D
IC3BUF	0154	Input Capture 3 Buffer Register																0000
IC3TMR	0156	Input Capture 3 Timer Value Register																xxxx
IC4CON1	0158	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC4CON2	015A	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D
IC4BUF	015C	Input Capture 4 Buffer Register																0000
IC4TMR	015E	Input Capture 4 Timer Value Register																xxxx
IC5CON1	0160	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC5CON2	0162	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D
IC5BUF	0164	Input Capture 5 Buffer Register																0000
IC5TMR	0166	Input Capture 5 Timer Value Register																xxxx
IC6CON1	0168	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC6CON2	016A	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D
IC6BUF	016C	Input Capture 6 Buffer Register																0000
IC6TMR	016E	Input Capture 6 Timer Value Register																xxxx
IC7CON1	0170	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC7CON2	0172	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D
IC7BUF	0174	Input Capture 7 Buffer Register																0000
IC7TMR	0176	Input Capture 7 Timer Value Register																xxxx
IC8CON1	0178	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC8CON2	017A	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D
IC8BUF	017C	Input Capture 8 Buffer Register																0000
IC8TMR	017E	Input Capture 8 Timer Value Register																xxxx
IC9CON1	0180	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC9CON2	0182	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D
IC9BUF	0184	Input Capture 9 Buffer Register																0000
IC9TMR	0186	Input Capture 9 Timer Value Register																xxxx

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-9: OUTPUT COMPARE REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
OC1CON1	0190	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	ENFLT2	ENFLT1	ENFLT0	OCFLT2	OCFLT1	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000
OC1CON2	0192	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	DCB1	DCB0	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000C
OC1RS	0194	Output Compare 1 Secondary Register																0000
OC1R	0196	Output Compare 1 Register																0000
OC1TMR	0198	Output Compare 1 Timer Value Register																xxxx
OC2CON1	019A	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	ENFLT2	ENFLT1	ENFLT0	OCFLT2	OCFLT1	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000
OC2CON2	019C	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	DCB1	DCB0	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000C
OC2RS	019E	Output Compare 2 Secondary Register																0000
OC2R	01A0	Output Compare 2 Register																0000
OC2TMR	01A2	Output Compare 2 Timer Value Register																xxxx
OC3CON1	01A4	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	ENFLT2	ENFLT1	ENFLT0	OCFLT2	OCFLT1	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000
OC3CON2	01A6	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	DCB1	DCB0	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000C
OC3RS	01A8	Output Compare 3 Secondary Register																0000
OC3R	01AA	Output Compare 3 Register																0000
OC3TMR	01AC	Output Compare 3 Timer Value Register																xxxx
OC4CON1	01AE	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	ENFLT2	ENFLT1	ENFLT0	OCFLT2	OCFLT1	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000
OC4CON2	01B0	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	DCB1	DCB0	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000C
OC4RS	01B2	Output Compare 4 Secondary Register																0000
OC4R	01B4	Output Compare 4 Register																0000
OC4TMR	01B6	Output Compare 4 Timer Value Register																xxxx
OC5CON1	01B8	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	ENFLT2	ENFLT1	ENFLT0	OCFLT1	OCFLT1	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000
OC5CON2	01BA	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	DCB1	DCB0	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000C
OC5RS	01BC	Output Compare 5 Secondary Register																0000
OC5R	01BE	Output Compare 5 Register																0000
OC5TMR	01C0	Output Compare 5 Timer Value Register																xxxx
OC6CON1	01C2	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	ENFLT2	ENFLT1	ENFLT0	OCFLT2	OCFLT1	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000
OC6CON2	01C4	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	DCB1	DCB0	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000C
OC6RS	01C6	Output Compare 6 Secondary Register																0000
OC6R	01C8	Output Compare 6 Register																0000
OC6TMR	01CA	Output Compare 6 Timer Value Register																xxxx
OC7CON1	01CC	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	ENFLT2	ENFLT1	ENFLT0	OCFLT2	OCFLT1	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000
OC7CON2	01CE	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	DCB1	DCB0	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000C
OC7RS	01D0	Output Compare 7 Secondary Register																0000
OC7R	01D2	Output Compare 7 Register																0000
OC7TMR	01D4	Output Compare 7 Timer Value Register																xxxx

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-9: OUTPUT COMPARE REGISTER MAP (CONTINUED)

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
OC8CON1	01D6	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	ENFLT2	ENFLT1	ENFLT0	OCFLT2	OCFLT1	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000
OC8CON2	01D8	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	DCB1	DCB0	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000C
OC8RS	01DA	Output Compare 8 Secondary Register																0000
OC8R	01DC	Output Compare 8 Register																0000
OC8TMR	01DE	Output Compare 8 Timer Value Register																xxxx
OC9CON1	01E0	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	ENFLT2	ENFLT1	ENFLT0	OCFLT2	OCFLT1	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000
OC9CON2	01E2	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	DCB1	DCB0	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000C
OC9RS	01E4	Output Compare 9 Secondary Register																0000
OC9R	01E6	Output Compare 9 Register																0000
OC9TMR	01E8	Output Compare 9 Timer Value Register																xxxx

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-10: I²C™ REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
I2C1RCV	0200	—	—	—	—	—	—	—	—	I2C1 Receive Register								0000
I2C1TRN	0202	—	—	—	—	—	—	—	—	I2C1 Transmit Register								00FF
I2C1BRG	0204	—	—	—	—	—	—	—	I2C1 Baud Rate Generator Register									0000
I2C1CON	0206	I2CEN	—	I2CSIDL	SCLREL	IPMIEN	A10M	DISSLW	SMEN	GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	1000
I2C1STAT	0208	ACKSTAT	TRSTAT	—	—	—	BCL	GCSTAT	ADD10	IWCOL	I2COV	D/$\overline{A}$	P	S	R/$\overline{W}$	RBF	TBF	0000
I2C1ADD	020A	—	—	—	—	—	—	I2C1 Address Register										0000
I2C1MSK	020C	—	—	—	—	—	—	I2C1 Address Mask Register										0000
I2C2RCV	0210	—	—	—	—	—	—	—	—	I2C2 Receive Register								0000
I2C2TRN	0212	—	—	—	—	—	—	—	—	I2C2 Transmit Register								00FF
I2C2BRG	0214	—	—	—	—	—	—	—	I2C2 Baud Rate Generator Register									0000
I2C2CON	0216	I2CEN	—	I2CSIDL	SCLREL	IPMIEN	A10M	DISSLW	SMEN	GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	1000
I2C2STAT	0218	ACKSTAT	TRSTAT	—	—	—	BCL	GCSTAT	ADD10	IWCOL	I2COV	D/$\overline{A}$	P	S	R/$\overline{W}$	RBF	TBF	0000
I2C2ADD	021A	—	—	—	—	—	—	I2C2 Address Register										0000
I2C2MSK	021C	—	—	—	—	—	—	I2C2 Address Mask Register										0000
I2C3RCV	0270	—	—	—	—	—	—	—	—	I2C3 Receive Register								0000
I2C3TRN	0272	—	—	—	—	—	—	—	—	I2C3 Transmit Register								00FF
I2C3BRG	0274	—	—	—	—	—	—	—	I2C3 Baud Rate Generator Register									0000
I2C3CON	0276	I2CEN	—	I2CSIDL	SCLREL	IPMIEN	A10M	DISSLW	SMEN	GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	1000
I2C3STAT	0278	ACKSTAT	TRSTAT	—	—	—	BCL	GCSTAT	ADD10	IWCOL	I2COV	D/$\overline{A}$	P	S	R/$\overline{W}$	RBF	TBF	0000
I2C3ADD	027A	—	—	—	—	—	—	I2C3 Address Register										0000
I2C3MSK	027C	—	—	—	—	—	—	I2C3 Address Mask Register										0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-11: UART REGISTER MAPS

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
U1MODE	0220	UARTEN	—	USIDL	IREN	RTSMD	—	UEN1	UEN0	WAKE	LPBACK	ABAUD	RXINV	BRGH	PDSEL1	PDSEL0	STSEL	0000
U1STA	0222	UTXISEL1	UTXINV	UTXISEL0	—	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA	0110
U1TXREG	0224	—	—	—	—	—	—	—	UART1 Transmit Register									xxxx
U1RXREG	0226	—	—	—	—	—	—	—	UART1 Receive Register									0000
U1BRG	0228	UART1 Baud Rate Generator Prescaler Register																0000
U2MODE	0230	UARTEN	—	USIDL	IREN	RTSMD	—	UEN1	UEN0	WAKE	LPBACK	ABAUD	RXINV	BRGH	PDSEL1	PDSEL0	STSEL	0000
U2STA	0232	UTXISEL1	UTXINV	UTXISEL0	—	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA	0110
U2TXREG	0234	—	—	—	—	—	—	—	UART2 Transmit Register									xxxx
U2RXREG	0236	—	—	—	—	—	—	—	UART2 Receive Register									0000
U2BRG	0238	UART2 Baud Rate Generator Prescaler Register																0000
U3MODE	0250	UARTEN	—	USIDL	IREN	RTSMD	—	UEN1	UEN0	WAKE	LPBACK	ABAUD	RXINV	BRGH	PDSEL1	PDSEL0	STSEL	0000
U3STA	0252	UTXISEL1	UTXINV	UTXISEL0	—	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA	0110
U3TXREG	0254	—	—	—	—	—	—	—	UART3 Transmit Register									xxxx
U3RXREG	0256	—	—	—	—	—	—	—	UART3 Receive Register									0000
U3BRG	0258	UART3 Baud Rate Generator Prescaler Register																0000
U4MODE	02B0	UARTEN	—	USIDL	IREN	RTSMD	—	UEN1	UEN0	WAKE	LPBACK	ABAUD	RXINV	BRGH	PDSEL1	PDSEL0	STSEL	0000
U4STA	02B2	UTXISEL1	UTXINV	UTXISEL0	—	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA	0110
U4TXREG	02B4	—	—	—	—	—	—	—	UART4 Transmit Register									xxxx
U4RXREG	02B6	—	—	—	—	—	—	—	UART4 Receive Register									0000
U4BRG	02B8	UART4 Baud Rate Generator Prescaler Register																0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-12: SPI REGISTER MAPS

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
SPI1STAT	0240	SPIEN	—	SPISIDL	—	—	SPIBEC2	SPIBEC1	SPIBEC0	SRMPT	SPIROV	SRXMPT	SISEL2	SISEL1	SISEL0	SPITBF	SPIRBF	0000
SPI1CON1	0242	—	—	—	DISSCK	DISSDO	MODE16	SMP	CKE	SSEN	CKP	MSTEN	SPRE2	SPRE1	SPRE0	PPRE1	PPRE0	0000
SPI1CON2	0244	FRMEN	SPIFSD	SPIFPOL	—	—	—	—	—	—	—	—	—	—	—	SPIFE	SPIBEN	0000
SPI1BUF	0248	SPI1 Transmit and Receive Buffer																0000
SPI2STAT	0260	SPIEN	—	SPISIDL	—	—	SPIBEC2	SPIBEC1	SPIBEC0	SRMPT	SPIROV	SRXMPT	SISEL2	SISEL1	SISEL0	SPITBF	SPIRBF	0000
SPI2CON1	0262	—	—	—	DISSCK	DISSDO	MODE16	SMP	CKE	SSEN	CKP	MSTEN	SPRE2	SPRE1	SPRE0	PPRE1	PPRE0	0000
SPI2CON2	0264	FRMEN	SPIFSD	SPIFPOL	—	—	—	—	—	—	—	—	—	—	—	SPIFE	SPIBEN	0000
SPI2BUF	0268	SPI2 Transmit and Receive Buffer																0000
SPI3STAT	0280	SPIEN	—	SPISIDL	—	—	SPIBEC2	SPIBEC1	SPIBEC0	SRMPT	SPIROV	SRXMPT	SISEL2	SISEL1	SISEL0	SPITBF	SPIRBF	0000
SPI3CON1	0282	—	—	—	DISSCK	DISSDO	MODE16	SMP	CKE	SSEN	CKP	MSTEN	SPRE2	SPRE1	SPRE0	PPRE1	PPRE0	0000
SPI3CON2	0284	FRMEN	SPIFSD	SPIFPOL	—	—	—	—	—	—	—	—	—	—	—	SPIFE	SPIBEN	0000
SPI3BUF	0288	SPI3 Transmit and Receive Buffer																0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-13: PORTA REGISTER MAP(1)

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit2	Bit 1	Bit 0	All Resets
TRISA	02C0	TRISA15	TRISA14	—	—	—	TRISA10	TRISA9	—	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	C6FF
PORTA	02C2	RA15	RA14	—	—	—	RA10	RA9	—	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	xxxx
LATA	02C4	LATA15	LATA14	—	—	—	LATA10	LATA9	—	LATA7	LATA6	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0	xxxx
ODCA	02C6	ODA15	ODA14	—	—	—	ODA10	ODA9	—	ODA7	ODA6	ODA5	ODA4	ODA3	ODA2	ODA1	ODA0	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal. Reset values shown are for 100-pin devices.
Note 1: PORTA and all associated bits are unimplemented on 64-pin devices and read as '0'. Bits are available on 100-pin devices only, unless otherwise noted.

TABLE 4-14: PORTB REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
TRISB	02C8	TRISB15	TRISB14	TRISB13	TRISB12	TRISB11	TRISB10	TRISB9	TRISB8	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	FFFF
PORTB	02CA	RB15	RB14	RB13	RB12	RB11	RB10	RB9	RB8	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx
LATB	02CC	LATB15	LATB14	LATB13	LATB12	LATB11	LATB10	LATB9	LATB8	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	xxxx
ODCB	02CE	ODB15	ODB14	ODB13	ODB12	ODB11	ODB10	ODB9	ODB8	ODB7	ODB6	ODB5	ODB4	ODB3	ODB2	ODB1	ODB0	0000

Legend: Reset values are shown in hexadecimal.

TABLE 4-15: PORTC REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4(1)	Bit 3(1)	Bit 2(1)	Bit 1(1)	Bit 0	All Resets
TRISC	02D0	TRISC15	TRISC14	TRISC13	TRISC12	—	—	—	—	—	—	—	TRISC4	TRISC3	TRISC2	TRISC1	—	F01E
PORTC	02D2	RC15(2,3)	RC14	RC13	RC12(2)	—	—	—	—	—	—	—	RC4	RC3	RC2	RC1	—	xxxx
LATC	02D4	LATC15	LATC14	LATC13	LATC12	—	—	—	—	—	—	—	LATC4	LATC3	LATC2	LATC1	—	xxxx
ODCC	02D6	ODC15	ODC14	ODC13	ODC12	—	—	—	—	—	—	—	ODC4	ODC3	ODC2	ODC1	—	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal. Reset values shown are for 100-pin devices.
Note 1: Bits are unimplemented in 64-pin devices; read as '0'.
2: RC12 and RC15 are only available when the primary oscillator is disabled or when EC mode is selected (POSCMD<1:0> Configuration bits = 11 or 00); otherwise read as '0'.
3: RC15 is only available when the POSCMD<1:0> Configuration bits = 11 or 00 and the OSCIOFN Configuration bit = 1.

TABLE 4-16: PORTD REGISTER MAP

File Name	Addr	Bit 15(1)	Bit 14(1)	Bit 13(1)	Bit 12(1)	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
TRISD	02D8	TRISD15	TRISD14	TRISD13	TRISD12	TRISD11	TRISD10	TRISD9	TRISD8	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	FFFF
PORTD	02DA	RD15	RD14	RD13	RD12	RD11	RD10	RD9	RD8	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	xxxx
LATD	02DC	LATD15	LATD14	LATD13	LATD12	LATD11	LATD10	LATD9	LATD8	LATD7	LATD6	LATD5	LATD4	LATD3	LATD2	LATD1	LATD0	xxxx
ODCD	02DE	ODD15	ODD14	ODD13	ODD12	ODD11	ODD10	ODD9	ODD8	ODD7	ODD6	ODD5	ODD4	ODD3	ODD2	ODD1	ODD0	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal. Reset values shown are for 100-pin devices.
Note 1: Bits are unimplemented in 64-pin devices; read as '0'.

TABLE 4-17: PORTE REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9(1)	Bit 8(1)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
TRISE	02E0	—	—	—	—	—	—	TRISE9	TRISE8	TRISE7	TRISE6	TRISE5	TRISE4	TRISE3	TRISE2	TRISE1	TRISE0	03FF
PORTE	02E2	—	—	—	—	—	—	RE9	RE8	RE7	RE6	RE5	RE4	RE3	RE2	RE1	RE0	xxxx
LATE	02E4	—	—	—	—	—	—	LATE9	LATE8	LATE7	LATE6	LATE5	LATE4	LATE3	LATE2	LATE1	LATE0	xxxx
ODCE	02E6	—	—	—	—	—	—	ODE9	ODE8	ODE7	ODE6	ODE5	ODE4	ODE3	ODE2	ODE1	ODE0	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal. Reset values shown are for 100-pin devices.
Note 1: Bits are unimplemented in 64-pin devices; read as '0'.

TABLE 4-18: PORTF REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13(1)	Bit 12(1)	Bit 11	Bit 10	Bit 9	Bit 8(1)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2(1)	Bit 1	Bit 0	All Resets
TRISF	02E8	—	—	TRISF13	TRISF12	—	—	—	TRISF8	TRISF7	—	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	TRISF0	31BF
PORTF	02EA	—	—	RF13	RF12	—	—	—	RF8	RF7	—	RF5	RF4	RF3	RF2	RF1	RF0	xxxx
LATF	02EC	—	—	LATF13	LATF12	—	—	—	LATF8	LATF7	—	LATF5	LATF4	LATF3	LATF2	LATF1	LATF0	xxxx
ODCF	02EE	—	—	ODF13	ODF12	—	—	—	ODF8	ODF7	—	ODF5	ODF4	ODF3	ODF2	ODF1	ODF0	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal. Reset values shown are for 100-pin devices.
Note 1: Bits are unimplemented in 64-pin devices; read as '0'.

TABLE 4-19: PORTG REGISTER MAP

File Name	Addr	Bit 15(1)	Bit 14(1)	Bit 13(1)	Bit 12(1)	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1(1)	Bit 0(1)	All Resets
TRISG	02F0	TRISG15	TRISG14	TRISG13	TRISG12	—	—	TRISG9	TRISG8	TRISG7	TRISG6	—	—	TRISG3	TRISG2	TRISG1	TRISG0	F3CF
PORTG	02F2	RG15	RG14	RG13	RG12	—	—	RG9	RG8	RG7	RG6	—	—	RG3	RG2	RG1	RG0	xxxx
LATG	02F4	LATG15	LATG14	LATG13	LATG12	—	—	LATG9	LATG8	LATG7	LATG6	—	—	LATG3	LATG2	LATG1	LATG0	xxxx
ODCG	02F6	ODG15	ODG14	ODG13	ODG12	—	—	ODG9	ODG8	ODG7	ODG6	—	—	ODG3	ODG2	ODG1	ODG0	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal. Reset values shown are for 100-pin devices.
Note 1: Bits are unimplemented in 64-pin devices; read as '0'.

TABLE 4-20: PAD CONFIGURATION REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
PADCFG1	02FC	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RTSECSEL	PMPTTL(1)	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.
Note 1: Bits are unimplemented in 64-pin devices; read as '0'.

TABLE 4-21: ADC REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
ADC1BUF0	0300	ADC Data Buffer 0																xxxx
ADC1BUF1	0302	ADC Data Buffer 1																xxxx
ADC1BUF2	0304	ADC Data Buffer 2																xxxx
ADC1BUF3	0306	ADC Data Buffer 3																xxxx
ADC1BUF4	0308	ADC Data Buffer 4																xxxx
ADC1BUF5	030A	ADC Data Buffer 5																xxxx
ADC1BUF6	030C	ADC Data Buffer 6																xxxx
ADC1BUF7	030E	ADC Data Buffer 7																xxxx
ADC1BUF8	0310	ADC Data Buffer 8																xxxx
ADC1BUF9	0312	ADC Data Buffer 9																xxxx
ADC1BUFA	0314	ADC Data Buffer 10																xxxx
ADC1BUFB	0316	ADC Data Buffer 11																xxxx
ADC1BUFC	0318	ADC Data Buffer 12																xxxx
ADC1BUFD	031A	ADC Data Buffer 13																xxxx
ADC1BUFE	031C	ADC Data Buffer 14																xxxx
ADC1BUFF	031E	ADC Data Buffer 15																xxxx
ADC1BUF10	0340	ADC Data Buffer 16																xxxx
ADC1BUF11	0342	ADC Data Buffer 17																xxxx
ADC1BUF12	0344	ADC Data Buffer 18																xxxx
ADC1BUF13	0346	ADC Data Buffer 19																xxxx
ADC1BUF14	0348	ADC Data Buffer 20																xxxx
ADC1BUF15	034A	ADC Data Buffer21																xxxx
ADC1BUF16	034C	ADC Data Buffer 22																xxxx
ADC1BUF17	034E	ADC Data Buffer 23																xxxx
ADC1BUF18	0350	ADC Data Buffer 24																xxxx
ADC1BUF19	0352	ADC Data Buffer 25																xxxx
ADC1BUF1A	0354	ADC Data Buffer 26																xxxx
ADC1BUF1B	0356	ADC Data Buffer 27																xxxx
ADC1BUF1C	0358	ADC Data Buffer 28																xxxx
ADC1BUF1D	035A	ADC Data Buffer 29																xxxx
ADC1BUF1E	035C	ADC Data Buffer 30																xxxx
ADC1BUF1F	035E	ADC Data Buffer 31																xxxx

Legend: — = unimplemented, read as '0', r = reserved, maintain as '0'. Reset values are shown in hexadecimal.
Note 1: Unimplemented in 64-pin devices, read as '0'

TABLE 4-21: ADC REGISTER MAP (CONTINUED)

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
AD1CON1	0320	ADON	—	ADSIDL	—	—	—	FORM1	FORM0	SSRC2	SSRC1	SSRC0	—	—	ASAM	SAMP	DONE	0000
AD1CON2	0322	VCFG2	VCFG1	VCFG0	r	—	CSCNA	—	—	BUFS	SMPI4	SMPI3	SMPI2	SMPI1	SMPI0	BUFM	ALTS	0000
AD1CON3	0324	ADRC	r	r	SAMC4	SAMC3	SAMC2	SAMC1	SAMC0	ADCS7	ADCS6	ADCS5	ADCS4	ADCS3	ADCS2	ADCS1	ADCS0	0000
AD1CHS	0328	CH0NB	—	—	CH0SB4	CH0SB3	CH0SB2	CH0SB1	CH0SB0	CH0NA	—	—	CH0SA4	CH0SA3	CH0SA2	CH0SA1	CH0SA0	0000
AD1CSSH	032E	—	—	—	—	CSSL27	CSSL26	CSSL25	CSSL24	CSSL23(1)	CSSL22(1)	CSSL21(1)	CSSL20(1)	CSSL19(1)	CSSL18(1)	CSSL17(1)	CSSL16(1)	0000
AD1CSSL	0330	CSSL15	CSSL14	CSSL13	CSSL12	CSSL11	CSSL10	CSSL9	CSSL8	CSSL7	CSSL6	CSSL5	CSSL4	CSSL3	CSSL2	CSSL1	CSSL0	0000

Legend: — = unimplemented, read as '0', r = reserved, maintain as '0'. Reset values are shown in hexadecimal.
Note 1: Unimplemented in 64-pin devices, read as '0'

TABLE 4-22: CTMU REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
CTMUCON	033C	CTMUEN	—	CTMUSIDL	TGEN	EDGEN	EDGSEQEN	IDISSEN	CTTRIG	EDG2POL	EDG2SEL1	EDG2SEL0	EDG1POL	EDG1SEL1	EDG1SEL0	EDG2STAT	EDG1STAT	0000
CTMUICON	033E	ITRIM5	ITRIM4	ITRIM3	ITRIM2	ITRIM1	ITRIM0	IRNG1	IRNG0	—	—	—	—	—	—	—	—	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-23: USB OTG REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
U1OTGIR(2)	0480	—	—	—	—	—	—	—	—	IDIF	T1MSECIF	LSTATEIF	ACTVIF	SESVDIF	SESENDIF	—	VBUSVDIF	0000
U1OTGIE(2)	0482	—	—	—	—	—	—	—	—	IDIE	T1MSECIE	LSTATEIE	ACTVIE	SESVDIE	SESENDIE	—	VBUSVDIE	0000
U1OTGSTAT2)	0484	—	—	—	—	—	—	—	—	ID	—	LSTATE	—	SESVD	SESEND	—	VBUSVD	0000
U1OTGCON(2)	0486	—	—	—	—	—	—	—	—	DPPULUP	DMPULUP	DPPULDWN	DMPULDWN	VBUSON	OTGEN	VBUSCHG	VBUSDIS	0000
U1PWRC	0488	—	—	—	—	—	—	—	—	UACTPND	—	—	USLPGRD	—	—	USUSPND	USBPWR	0000
U1IR	048A(1)	—	—	—	—	—	—	—	—	STALLIF	—	RESUMEIF	IDLEIF	TRNIF	SOFIF	UERRIF	URSTIF	0000
		—	—	—	—	—	—	—	—	STALLIF	ATTACHIF(1)	RESUMEIF	IDLEIF	TRNIF	SOFIF	UERRIF	DETACHIF(1)	0000
U1IE	048C(1)	—	—	—	—	—	—	—	—	STALLIE	—	RESUMEIE	IDLEIE	TRNIE	SOFIE	UERRIE	URSTIE	0000
		—	—	—	—	—	—	—	—	STALLIE	ATTACHIE(1)	RESUMEIE	IDLEIE	TRNIE	SOFIE	UERRIE	DETACHIE(1)	0000
U1EIR	048E(1)	—	—	—	—	—	—	—	—	BTSEF	—	DMAEF	BTOEF	DFN8EF	CRC16EF	CRC5EF	PIDEF	0000
		—	—	—	—	—	—	—	—	BTSEF	—	DMAEF	BTOEF	DFN8EF	CRC16EF	EOFEF(1)	PIDEF	0000
U1EIE	0490(1)	—	—	—	—	—	—	—	—	BTSEE	—	DMAEE	BTOEE	DFN8EE	CRC16EE	CRC5EE	PIDEE	0000
		—	—	—	—	—	—	—	—	BTSEE	—	DMAEE	BTOEE	DFN8EE	CRC16EE	EOFEE(1)	PIDEE	0000
U1STAT	0492	—	—	—	—	—	—	—	—	ENDPT3	ENDPT2	ENDPT1	ENDPT0	DIR	PPBI	—	—	0000
U1CON	0494(1)	—	—	—	—	—	—	—	—	—	SE0	PKTDIS	—	HOSTEN	RESUME	PPBRST	USBEN	0000
		—	—	—	—	—	—	—	—	JSTATE(1)	SE0	TOKBUSY	USBRST	HOSTEN	RESUME	PPBRST	SOFEN(1)	0000
U1ADDR	0496	—	—	—	—	—	—	—	—	LSPDEN(1)	USB Device Address (DEVADDR) Register							0000
U1BDTP1	0498	—	—	—	—	—	—	—	—	Buffer Descriptor Table Base Address Register							—	0000
U1FRML	049A	—	—	—	—	—	—	—	—	Frame Count Register Low Byte								0000
U1FRMH	049C	—	—	—	—	—	—	—	—	—	—	—	—	—	Frame Count Register High Byte			0000
U1TOK(2)	049E	—	—	—	—	—	—	—	—	PID3	PID2	PID1	PID0	EP3	EP2	EP1	EP0	0000
U1SOF(2)	04A0	—	—	—	—	—	—	—	—	Start-Of-Frame Count Register								0000
U1CNFG1	04A6	—	—	—	—	—	—	—	—	UTEYE	UOEMON	—	USBSIDL	—	—	PPB1	PPB0	0000
U1CNFG2	04A8	—	—	—	—	—	—	—	—	—	—	UVCMPSEL	PUVBUS	EXTI2CEN	UVBUSDIS	UVCMPDIS	UTRDIS	0000
U1EP0	04AA	—	—	—	—	—	—	—	—	LSPD(1)	RETRYDIS(1)	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP1	04AC	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP2	04AE	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP3	04B0	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP4	04B2	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP5	04B4	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP6	04B6	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP7	04B8	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP8	04BA	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP9	04BC	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

Note 1: Alternate register or bit definitions when the module is operating in Host mode.

2: This register is available in Host mode only.

TABLE 4-23: USB OTG REGISTER MAP (CONTINUED)

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
U1EP10	04BE	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP11	04C0	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP12	04C2	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP13	04C4	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP14	04C6	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1EP15	04C8	—	—	—	—	—	—	—	—	—	—	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK	0000
U1PWMRRS	04CC	USB Power Supply PWM Duty Cycle Register								USB Power Supply PWM Period Register								0000
U1PWMCON	04CE	PWMEN	—	—	—	—	—	PWMPOL	CNTEN	—	—	—	—	—	—	—	—	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.
Note 1: Alternate register or bit definitions when the module is operating in Host mode.
2: This register is available in Host mode only.

TABLE 4-24: ANCFG REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
ANCFG	04DE	—	—	—	—	—	—	—	—	—	—	—	—	—	VBG6EN	VBG2EN	VBGEN	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-25: ANSEL REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets(2)
ANSA(1)	04E0	—	—	—	—	—	ANSA10(1)	ANSA9(1)	—	ANSA7(1)	ANSA6(1)	—	—	—	—	—	—	06C0
ANSB	04E2	ANSB15	ANSB14	ANSB13	ANSB12	ANSB11	ANSB10	ANSB9	ANSB8	ANSB7	ANSB6	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	FFFF
ANSC	04E4	—	ANSC14	ANSC13	—	—	—	—	—	—	—	—	ANSC4(1)	—	—	—	—	6010
ANSD	04E6	—	—	—	—	—	—	—	—	ANSD7	ANSD6	—	—	—	—	—	—	00C0
ANSE(1)	04E8	—	—	—	—	—	—	ANSE9(1)	—	—	—	—	—	—	—	—	—	0200
ANSF	04EA	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ANSF0	0001
ANSG	04EC	—	—	—	—	—	—	ANSG9	ANSG8	ANSG7	ANSG6	—	—	—	—	—	—	03C0

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.
Note 1: Unimplemented in 64-pin devices, read as '0'.
2: Reset values are valid for 100-pin devices only.

TABLE 4-26: ENHANCED PARALLEL MASTER/SLAVE PORT REGISTER MAP(1)

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
PMCON1	0600	PMPEN	—	PSIDL	ADRMUX1	ADRMUX0	—	MODE1	MODE0	CSF1	CSF0	ALP	ALMODE	—	BUSKEEP	IRQM1	IRQM0	0000
PMCON2	0602	BUSY	—	ERROR	TIMEOUT	AMREQ	CURMST	MSTSEL1	MSTSEL0	RADDR23	RADDR22	RADDR21	RADDR20	RADDR19	RADDR18	RADDR17	RADDR16	0000
PMCON3	0604	PTWREN	PTRDEN	PTBE1EN	PTBE0EN	—	AWAITM1	AWAITM0	AWAITE	—	PTEN22	PTEN21	PTEN20	PTEN19	PTEN18	PTEN17	PTEN16	0000
PMCON4	0606	PTEN15	PTEN14	PTEN13	PTEN12	PTEN11	PTEN10	PTEN9	PTEN8	PTEN7	PTEN6	PTEN5	PTEN4	PTEN3	PTEN2	PTEN1	PTEN0	0000
PMCS1CF	0608	CSDIS	CSP	CSPTEN	BEP	—	WRSP	RDSP	SM	ACKP	PTSZ1	PTSZ0	—	—	—	—	—	0000
PMCS1BS	060A	BASE23	BASE22	BASE21	BASE20	BASE19	BASE18	BASE17	BASE16	BASE15	—	—	—	BASE11	—	—	—	0200
PMCS1MD	060C	ACKM1	ACKM0	AMWAIT2	AMWAIT1	AMWAIT0	—	—	—	DWAITB1	DWAITB0	DWAITM3	DWAITM2	DWAITM1	DWAITM0	DWAITE1	DWAITE0	0000
PMCS2CF	060E	CSDIS	CSP	CSPTEN	BEP	—	WRSP	RDSP	SM	ACKP	PTSZ1	PTSZ0	—	—	—	—	—	0000
PMCS2BS	0610	BASE23	BASE22	BASE21	BASE20	BASE19	BASE18	BASE17	BASE16	BASE15	—	—	—	BASE11	—	—	—	0600
PMCS2MD	0612	ACKM1	ACKM0	AMWAIT2	AMWAIT1	AMWAIT0	—	—	—	DWAITB1	DWAITB0	DWAITM3	DWAITM2	DWAITM1	DWAITM0	DWAITE1	DWAITE0	0000
PMDOUT1	0614	EPMP Data Out Register 1<15:8>								EPMP Data Out Register 1<7:0>								xxxx
PMDOUT2	0616	EPMP Data Out Register 2<15:8>								EPMP Data Out Register 2<7:0>								xxxx
PMDIN1	0618	EPMP Data In Register 1<15:8>								EPMP Data In Register 1<7:0>								xxxx
PMDIN2	061A	EPMP Data In Register 2<15:8>								EPMP Data In Register 2<7:0>								xxxx
PMSTAT	061C	IBF	IBOV	—	—	IB3F	IB2F	IB1F	IB0F	OBE	OBUF	—	—	OB3E	OB2E	OB1E	OB0E	008F

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.
Note 1: Unimplemented in 64-pin devices, read as '0'.

TABLE 4-27: REAL-TIME CLOCK AND CALENDAR REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
ALRMVAL	0620	Alarm Value Register Window Based on ALRMPTR<1:0>																xxxx
ALCFGRPT	0622	ALRMEN	CHIME	AMASK3	AMASK2	AMASK1	AMASK0	ALRMPTR1	ALRMPTR0	ARPT7	ARPT6	ARPT5	ARPT4	ARPT3	ARPT2	ARPT1	ARPT0	0000
RTCVAL	0624	RTCC Value Register Window Based on RTCPTR<1:0>																xxxx
RCFGCAL	0626	RTCEN	—	RTCWREN	RTCSYNC	HALFSEC	RTCOE	RTCPTR1	RTCPTR0	CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0	(Note 1)

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.
Note 1: The status of the RCFGCAL register on POR is '0000' and on other Resets is unchanged.

TABLE 4-28: COMPARATORS REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
CMSTAT	0630	CMIDL	—	—	—		C3EVT	C2EVT	C1EVT	—	—	—	—	—	C3OUT	C2OUT	C1OUT	0000
CVRCON	0632	—	—	—	—	—	CVREFP	CVREFM1	CVREFM0	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0	0000
CM1CON	0634	CON	COE	CPOL	—	—	—	CEVT	COUT	EVPOL1	EVPOL0	—	CREF	—	—	CCH1	CCH0	0000
CM2CON	0636	CON	COE	CPOL	—	—	—	CEVT	COUT	EVPOL1	EVPOL0	—	CREF	—	—	CCH1	CCH0	0000
CM3CON	0638	CON	COE	CPOL	—	—	—	CEVT	COUT	EVPOL1	EVPOL0	—	CREF	—	—	CCH1	CCH0	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-29: CRC REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
CRCCON1	0640	CRCEN	—	CSIDL	VWORD4	VWORD3	VWORD2	VWORD1	VWORD0	CRCFUL	CRCMPT	CRCISEL	CRCGO	LENDIAN	—	—	—	0040
CRCCON2	0642	—	—	—	DWIDTH4	DWIDTH3	DWIDTH2	DWIDTH1	DWIDTH0	—	—	—	PLEN4	PLEN3	PLEN2	PLEN1	PLEN0	0000
CRCXORL	0644	X15	X14	X13	X12	X11	X10	X9	X8	X7	X6	X5	X4	X3	X2	X1	—	0000
CRCXORH	0646	X31	X30	X29	X28	X27	X26	X25	X24	X23	X22	X21	X20	X19	X18	X17	X16	0000
CRCDATL	0648	CRC Data Input Register Low																0000
CRCDATH	064A	CRC Data Input Register High																0000
CRCWDATL	064C	CRC Result Register Low																0000
CRCWDATH	064E	CRC Result Register High																0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-30: PERIPHERAL PIN SELECT REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
RPINR0	0680	—	—	INT1R5	INT1R4	INT1R3	INT1R2	INT1R1	INT1R0	—	—	—	—	—	—	—	—	3F00
RPINR1	0682	—	—	INT3R5	INT3R4	INT3R3	INT3R2	INT3R1	INT3R0	—	—	INT2R5	INT2R4	INT2R3	INT2R2	INT2R1	INT2R0	3F3F
RPINR2	0684	—	—	—	—	—	—	—	—	—	—	INT4R5	INT4R4	INT4R3	INT4R2	INT4R1	INT4R0	003F
RPINR3	0686	—	—	T3CKR5	T3CKR4	T3CKR3	T3CKR2	T3CKR1	T3CKR0	—	—	T2CKR5	T2CKR4	T2CKR3	T2CKR2	T2CKR1	T2CKR0	3F3F
RPINR4	0688	—	—	T5CKR5	T5CKR4	T5CKR3	T5CKR2	T5CKR1	T5CKR0	—	—	T4CKR5	T4CKR4	T4CKR3	T4CKR2	T4CKR1	T4CKR0	3F3F
RPINR7	068E	—	—	IC2R5	IC2R4	IC2R3	IC2R2	IC2R1	IC2R0	—	—	IC1R5	IC1R4	IC1R3	IC1R2	IC1R1	IC1R0	3F3F
RPINR8	0690	—	—	IC4R5	IC4R4	IC4R3	IC4R2	IC4R1	IC4R0	—	—	IC3R5	IC3R4	IC3R3	IC3R2	IC3R1	IC3R0	3F3F
RPINR9	0692	—	—	IC6R5	IC6R4	IC6R3	IC6R2	IC6R1	IC6R0	—	—	IC5R5	IC5R4	IC5R3	IC5R2	IC5R1	IC5R0	3F3F
RPINR10	0694	—	—	IC8R5	IC8R4	IC8R3	IC8R2	IC8R1	IC8R0	—	—	IC7R5	IC7R4	IC7R3	IC7R2	IC7R1	IC7R0	3F3F
RPINR11	0696	—	—	OCFBR5	OCFBR4	OCFBR3	OCFBR2	OCFBR1	OCFBR0	—	—	OCFAR5	OCFAR4	OCFAR3	OCFAR2	OCFAR1	OCFAR0	3F3F
RPINR15	069E	—	—	IC9R5	IC9R4	IC9R3	IC9R2	IC9R1	IC9R0	—	—	—	—	—	—	—	—	3F00
RPINR17	06A2	—	—	U3RXR5	U3RXR4	U3RXR3	U3RXR2	U3RXR1	U3RXR0	—	—	—	—	—	—	—	—	3F00
RPINR18	06A4	—	—	U1CTSR5	U1CTSR4	U1CTSR3	U1CTSR2	U1CTSR1	U1CTSR0	—	—	U1RXR5	U1RXR4	U1RXR3	U1RXR2	U1RXR1	U1RXR0	3F3F
RPINR19	06A6	—	—	U2CTSR5	U2CTSR4	U2CTSR3	U2CTSR2	U2CTSR1	U2CTSR0	—	—	U2RXR5	U2RXR4	U2RXR3	U2RXR2	U2RXR1	U2RXR0	3F3F
RPINR20	06A8	—	—	SCK1R5	SCK1R4	SCK1R3	SCK1R2	SCK1R1	SCK1R0	—	—	SDI1R5	SDI1R4	SDI1R3	SDI1R2	SDI1R1	SDI1R0	3F3F
RPINR21	06AA	—	—	U3CTSR5	U3CTSR4	U3CTSR3	U3CTSR2	U3CTSR1	U3CTSR0	—	—	SS1R5	SS1R4	SS1R3	SS1R2	SS1R1	SS1R0	3F3F
RPINR22	06AC	—	—	SCK2R5	SCK2R4	SCK2R3	SCK2R2	SCK2R1	SCK2R0	—	—	SDI2R5	SDI2R4	SDI2R3	SDI2R2	SDI2R1	SDI2R0	3F3F
RPINR23	06AE	—	—	—	—	—	—	—	—	—	—	SS2R5	SS2R4	SS2R3	SS2R2	SS2R1	SS2R0	003F
RPINR27	06B6	—	—	U4CTSR5	U4CTSR4	U4CTSR3	U4CTSR2	U4CTSR1	U4CTSR0	—	—	U4RXR5	U4RXR4	U4RXR3	U4RXR2	U4RXR1	U4RXR0	3F3F
RPINR28	06B8	—	—	SCK3R5	SCK3R4	SCK3R3	SCK3R2	SCK3R1	SCK3R0	—	—	SDI3R5	SDI3R4	SDI3R3	SDI3R2	SDI3R1	SDI3R0	3F3F
RPINR29	06BA	—	—	—	—	—	—	—	—	—	—	SS3R5	SS3R4	SS3R3	SS3R2	SS3R1	SS3R0	003F

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.
Note 1: Bits are unimplemented in 64-pin devices; read as '0'.

TABLE 4-30: PERIPHERAL PIN SELECT REGISTER MAP (CONTINUED)

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
RPOR0	06C0	—	—	RP1R5	RP1R4	RP1R3	RP1R2	RP1R1	RP1R0	—	—	RP0R5	RP0R4	RP0R3	RP0R2	RP0R1	RP0R0	0000
RPOR1	06C2	—	—	RP3R5	RP3R4	RP3R3	RP3R2	RP3R1	RP3R0	—	—	RP2R5	RP2R4	RP2R3	RP2R2	RP2R1	RP2R0	0000
RPOR2	06C4	—	—	RP5R5(1)	RP5R4(1)	RP5R3(1)	RP5R2(1)	RP5R1(1)	RP5R0(1)	—	—	RP4R5	RP4R4	RP4R3	RP4R2	RP4R1	RP4R0	0000
RPOR3	06C6	—	—	RP7R5	RP7R4	RP7R3	RP7R2	RP7R1	RP7R0	—	—	RP6R5	RP6R4	RP6R3	RP6R2	RP6R1	RP6R0	0000
RPOR4	06C8	—	—	RP9R5	RP9R4	RP9R3	RP9R2	RP9R1	RP9R0	—	—	RP8R5	RP8R4	RP8R3	RP8R2	RP8R1	RP8R0	0000
RPOR5	06CA	—	—	RP11R5	RP11R4	RP11R3	RP11R2	RP11R1	RP11R0	—	—	RP10R5	RP10R4	RP10R3	RP10R2	RP10R1	RP10R0	0000
RPOR6	06CC	—	—	RP13R5	RP13R4	RP13R3	RP13R2	RP13R1	RP13R0	—	—	RP12R5	RP12R4	RP12R3	RP12R2	RP12R1	RP12R0	0000
RPOR7	06CE	—	—	RP15R5(1)	RP15R4(1)	RP15R3(1)	RP15R2(1)	RP15R1(1)	RP15R0(1)	—	—	RP14R5	RP14R4	RP14R3	RP14R2	RP14R1	RP14R0	0000
RPOR8	06D0	—	—	RP17R5	RP17R4	RP17R3	RP17R2	RP17R1	RP17R0	—	—	RP16R5	RP16R4	RP16R3	RP16R2	RP16R1	RP16R0	0000
RPOR9	06D2	—	—	RP19R5	RP19R4	RP19R3	RP19R2	RP19R1	RP19R0	—	—	RP18R5	RP18R4	RP18R3	RP18R2	RP18R1	RP18R0	0000
RPOR10	06D4	—	—	RP21R5	RP21R4	RP21R3	RP21R2	RP21R1	RP21R0	—	—	RP20R5	RP20R4	RP20R3	RP20R2	RP20R1	RP20R0	0000
RPOR11	06D6	—	—	RP23R5	RP23R4	RP23R3	RP23R2	RP23R1	RP23R0	—	—	RP22R5	RP22R4	RP22R3	RP22R2	RP22R1	RP22R0	0000
RPOR12	06D8	—	—	RP25R5	RP25R4	RP25R3	RP25R2	RP25R1	RP25R0	—	—	RP24R5	RP24R4	RP24R3	RP24R2	RP24R1	RP24R0	0000
RPOR13	06DA	—	—	RP27R5	RP27R4	RP27R3	RP27R2	RP27R1	RP27R0	—	—	RP26R5	RP26R4	RP26R3	RP26R2	RP26R1	RP26R0	0000
RPOR14	06DC	—	—	RP29R5	RP29R4	RP29R3	RP29R2	RP29R1	RP29R0	—	—	RP28R5	RP28R4	RP28R3	RP28R2	RP28R1	RP28R0	0000
RPOR15(1)	06DE	—	—	RP31R5(1)	RP31R4(1)	RP31R3(1)	RP31R2(1)	RP31R1(1)	RP31R0(1)	—	—	RP30R5(1)	RP30R4(1)	RP30R3(1)	RP30R2(1)	RP30R1(1)	RP30R0(1)	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.
Note 1: Bits are unimplemented in 64-pin devices; read as '0'.

TABLE 4-31: GRAPHICS REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
G1CMDL	0700	Graphics Command Register<15:0>																0000
G1CMDH	0702	Graphics Command Register<31:16>																0000
G1CON1	0704	G1EN	—	G1SIDL	GCMDWMK4	GCMDWMK3	GCMDWMK2	GCMDWMK1	GCMDWMK0	PUBPP2	PUBPP1	PUBPP0	GCMDCNT4	GCMDCNT3	GCMDCNT2	GCMDCNT1	GCMDCNT0	0000
G1STAT	0706	PUBUSY	—	—	—	—	—	—	—	IPUBUSY	RCCBUSY	CHRBUSY	VMRGN	HMRGN	CMDLV	CMDFUL	CMDMPT	0000
G1IE	0708	PUIE	—	—	—	—	—	—	—	IPUIE	RCCIE	CHRIE	VMRGNIE	HMRGNIE	CMDLVIE	CMDFULIE	CMDMPTIE	0000
G1IR	070A	PUIF	—	—	—	—	—	—	—	IPUIF	RCCIF	CHRIF	VMRGNIF	HMRGNIF	CMDLVIF	CMDFULIF	CMDMPTIF	0000
G1W1ADRL	070C	GPU Work Area 1 Start Address Register<15:0>																0000
G1W1ADRH	070E	—	—	—	—	—	—	—	—	GPU Work Area 1 Start Address Register<23:16>								0000
G1W2ADRL	0710	GPU Work Area 2 Start Address Register<15:0>																0000
G1W2ADRH	0712	—	—	—	—	—	—	—	—	GPU Work Area 2 Start Address Register<23:16>								0000
G1PUW	0714	—	—	—	—	—	GPU Work Area Width Register											0000
G1PUH	0716	—	—	—	—	—	GPU Work Area Height Register											0000
G1DPADRL	0718	Display Buffer Start Address Register<15:0>																0000
G1DPADRH	071A	—	—	—	—	—	—	—	—	Display Buffer Start Address Register<23:16>								0000
G1DPW	071C	—	—	—	—	—	Display Frame Width Register											0000
G1DPH	071E	—	—	—	—	—	Display Frame Height Register											0000
G1DPWT	0720	—	—	—	—	—	Display Total Width Register											0000
G1DPHT	0722	—	—	—	—	—	Display Total Height Register											0000
G1CON2	0724	DPGWDTH1	DPGWDTH0	DPSTGER1	DPSTGER0	—	—	DPTEST1	DPTEST0	DPBPP2	DPBPP1	DPBPP0	—	—	DPMODE2	DPMODE1	DPMODE0	0000
G1CON3	0726	—	—	—	—	—	—	DPPINOE	DPPOWER	DPCLKPOL	DPENPOL	DPVSPOL	DPHSPOL	DPPWROE	DPENOE	DPVSOE	DPHSOE	0000
G1ACTDA	0728	Number of Lines Before the First Active Line Register								Number of Pixels Before the First Active PIxel Register								0000
G1HSYNC	072A	HSYNC Pulse-Width Configuration Register								HSYNC Start Delay Configuration Register								0000
G1VSYNC	072C	VSYNC Pulse-Width Configuration Register								VSYNC Start Delay Configuration Register								0000
G1DBLCON	072E	Vertical Blanking Start to First Displayed Line Configuration Regsiter								Horizontal Blanking Start to First Displayed Line Configuration Regsiter								0000
G1CLUT	0730	CLUTEN	CLUTBUSY	—	—	—	—	CLUTTRD	CLUTRWEN	Color Look-Up Table Memory Address Register								0000
G1CLUTWR	0732	Color Look-up Table Memory Write Data Register																0000
G1CLUTRD	0734	Color Look-up Table Memory Read Data Register																0000
G1MRGN	0736	Vertical Blanking Advance Register								Horizontal Blanking Advance Register								0000
G1CHRX	0738	—	—	—	—	—	Current Character X-Coordinate Position Register											0000
G1CHRY	073A	—	—	—	—	—	Current Character Y-Coordinate Position Register											0000
G1IPU	073C	—	—	—	—	—	—	—	—	—	—	HUFFERR	BLCKERR	LENERR	WRAPERR	IPUDONE	BFINAL	0000
G1DBEN	073E	GDBEN15	GDBEN14	GDBEN13	GDBEN12	GDBEN11	GDBEN10	GDBEN9	GDBEN8	GDBEN7	GDBEN6	GDBEN5	GDBEN4	GDBEN3	GDBEN2	GDBEN1	GDBEN0	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-32: SYSTEM REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
RCON	0740	TRAPR	IOPUWR	—	—	—	—	CM	VREGS	EXTR	SWR	SWDTEN	WDTO	SLEEP	IDLE	BOR	POR	Note 1
OSCCON	0742	—	COSC2	COSC1	COSC0	—	NOSC2	NOSC1	NOSC0	CLKLOCK	IOLOCK	LOCK	—	CF	POSCEN	SOSCEN	OSWEN	Note 2
CLKDIV	0744	ROI	DOZE2	DOZE1	DOZE0	DOZEN	RCDIV2	RCDIV1	RCDIV0	CPDIV1	CPDIV0	PLLEN	G1CLKSEL	—	—	—	—	0100
CLKDIV2	0746	GCLKDIV6	GCLKDIV5	GCLKDIV4	GCLKDIV3	GCLKDIV2	GCLKDIV1	GCLKDIV0	—	—	—	—	—	—	—	—	—	0000
OSCTUN	0748	—	—	—	—	—	—	—	—	—	—	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0	0000
REFOCON	074E	ROEN	—	ROSSLP	ROSEL	RODIV3	RODIV2	RODIV1	RODIV0	—	—	—	—	—	—	—	—	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.
Note 1: The Reset value of the RCON register is dependent on the type of Reset event. See **Section 6.0 "Resets"** for more information.
2: The Reset value of the OSCCON register is dependent on both the type of Reset event and the device configuration. See **Section 8.0 "Oscillator Configuration"** for more information.

TABLE 4-33: NVM REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
NVMCON	0760	WR	WREN	WRERR	—	—	—	—	—	—	ERASE	—	—	NVMOP3	NVMOP2	NVMOP1	NVMOP0	0000(1)
NVMKEY	0766	—	—	—	—	—	—	—	—	NVMKEY Register<7:0>								0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.
Note 1: Reset value shown is for POR only. Value on other Reset states is dependent on the state of memory write or erase operations at the time of Reset.

TABLE 4-34: PMD REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
PMD1	0770	T5MD	T4MD	T3MD	T2MD	T1MD	—	—	—	I2C1MD	U2MD	U1MD	SPI2MD	SPI1MD	—	—	ADC1MD	0000
PMD2	0772	IC8MD	IC7MD	IC6MD	IC5MD	IC4MD	IC3MD	IC2MD	IC1MD	OC8MD	OC7MD	OC6MD	OC5MD	OC4MD	OC3MD	OC2MD	OC1MD	0000
PMD3	0774	—	—	—	—	—	CMPMD	RTCCMD	PMPMD(1)	CRCMD	—	—	—	U3MD	I2C3MD	I2C2MD	—	0000
PMD4	0776	—	—	—	—	—	—	—	—	—	UPWMMD	U4MD	—	REFOMD	CTMUMD	LVDMD	USB1MD	0000
PMD5	0778	—	—	—	—	—	—	—	IC9MD	—	—	—	—	—	—	—	OC9MD	0000
PMD6	077A	—	—	—	—	—	—	—	—	—	GFX1MD	—	—	—	—	—	SPI3MD	0000

Legend: — = unimplemented, read as '0'. Reset values are shown in hexadecimal.
Note 1: Unimplemented in 64-pin devices, read as '0'.

4.2.5 EXTENDED DATA SPACE (EDS)

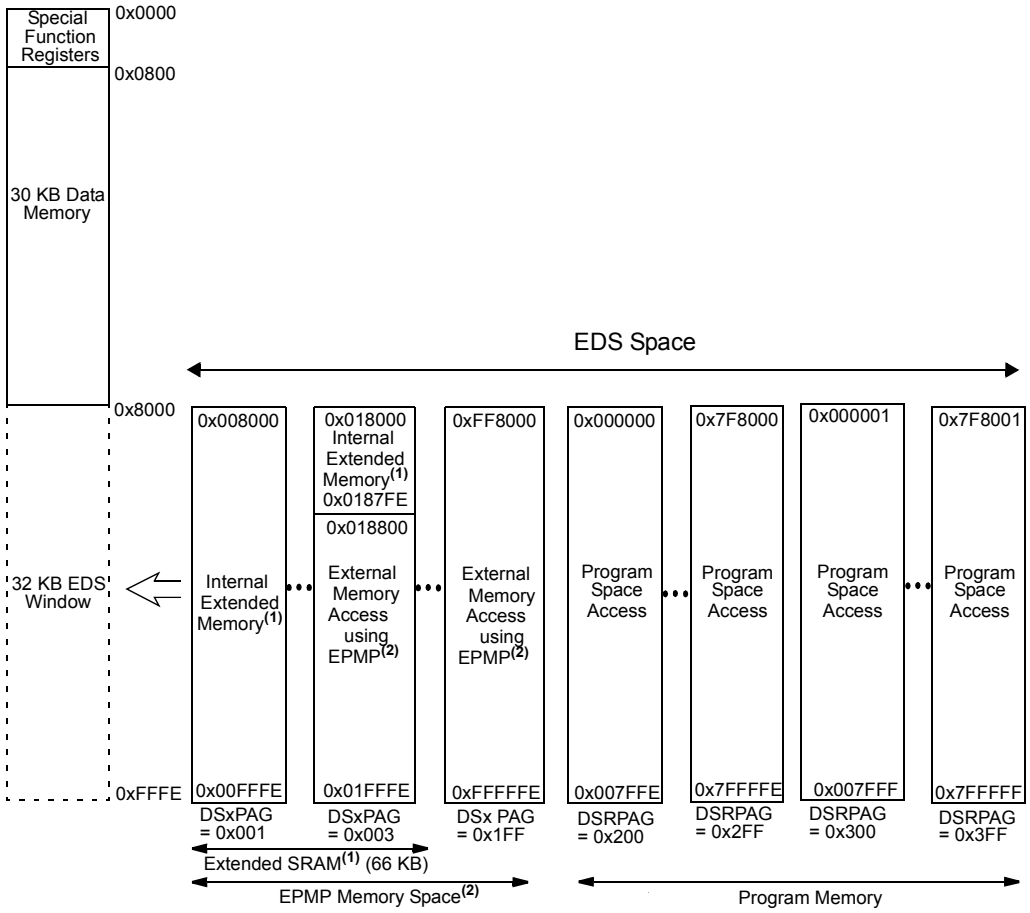
The enhancement of the data space in PIC24FJ256DA210 family devices has been accomplished by a new technique, called the Extended Data Space (EDS).

The EDS includes any additional internal extended data memory not accessible by the lower 32 Kbytes data address space, any external memory through EPMP and the Program Space Visibility (PSV).

The extended data space is always accessed through the EDS window, the upper half of data space. The entire extended data space is organized into EDS pages, each having 32 Kbytes of data. Mapping of the EDS page into the EDS window is done using the Data Space Read register (DSRPAG<9:0>) for read operations and Data Space Write register (DSWPAG<8:0>) for write operations. Figure 4-4 displays the entire EDS space.

Note:	Accessing Page 0 in the EDS window will generate an address error trap as Page 0 is the base data memory (data locations, 0x0800 to 0x7FFF, in the lower data space).

FIGURE 4-4: EXTENDED DATA SPACE



Note 1: Available only in PIC24FJXXXDA2XX devices. In the PIC24FJXXXDA110 devices, this space can be used to access external memory using EPMP.

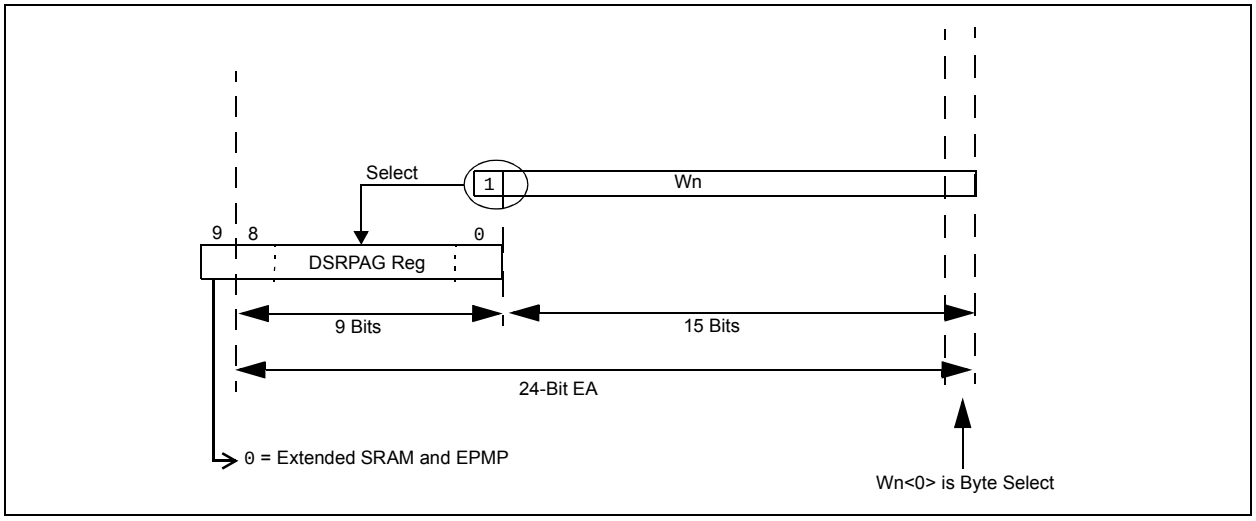
2: Available only in PIC24FJXXXDAX10 devices (100-pin).

4.2.5.1 Data Read from EDS Space

In order to read the data from the EDS space, first, an Address Pointer is set up by loading the required EDS page number into the DSRPAG register and assigning the offset address to one of the W registers. Once the above assignment is done, the EDS window is enabled by setting bit 15 of the working register, assigned with the offset address; then, the contents of the pointed EDS location can be read.

Figure 4-5 illustrates how the EDS space address is generated for read operations.

FIGURE 4-5: EDS ADDRESS GENERATION FOR READ OPERATIONS



When the Most Significant bit (MSBs) of EA is '1' and DSRPAG<9> = 0, the lower 9 bits of DSRPAG are concatenated to the lower 15 bits of EA to form a 24-bit EDS space address for read operations.

Example 4-1 shows how to read a byte, word and double-word from EDS.

Note: All read operations from EDS space have an overhead of one instruction cycle. Therefore, a minimum of two instruction cycles is required to complete an EDS read. EDS reads under the REPEAT instruction; the first two accesses take three cycles and the subsequent accesses take one cycle.

EXAMPLE 4-1: EDS READ CODE IN ASSEMBLY

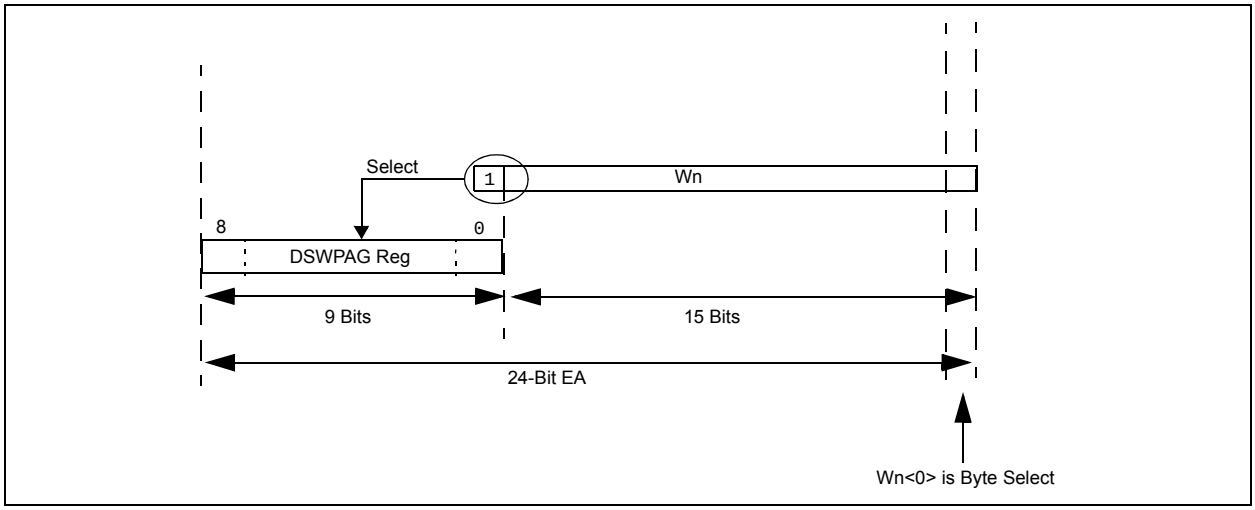
```
; Set the EDS page from where the data to be read
    mov        #0x0002 , w0
    mov        w0 , DSRPAG    ;page 2 is selected for read
    mov        #0x0800 ,  w1  ;select the location (0x800) to be read
bset       w1 , #15       ;set the MSB of the base address, enable EDS mode

;Read a byte from the selected location
    mov.b      [w1++] ,   w2  ;read Low byte
    mov.b      [w1++] , w3    ;read High byte

;Read a word from the selected location
    mov        [w1] , w2      ;

;Read Double - word from the selected location
    mov.d      [w1] , w2      ;two word read, stored in w2 and w3
```

4.2.5.2 Data Write into EDS Space

In order to write data to EDS space, such as in EDS reads, an Address Pointer is set up by loading the required EDS page number into the DSWPAG register, and assigning the offset address to one of the W registers. Once the above assignment is done, then the EDS window is enabled by setting bit 15 of the working register, assigned with the offset address, and the accessed location can be written.

Figure 4-2 illustrates how the EDS space address is generated for write operations.

FIGURE 4-6: EDS ADDRESS GENERATION FOR WRITE OPERATIONS



When the MSBs of EA is '1', the lower 9 bits of DSWPAG are concatenated to the lower 15 bits of EA to form a 24-bit EDS address for write operations. Example 4-2 shows how to write a byte, word and double-word to EDS.

EXAMPLE 4-2: EDS WRITE CODE IN ASSEMBLY

```
; Set the EDS page where the data to be written
    mov     #0x0002 , w0
    mov     w0 , DSWPAG     ;page 2 is selected for write
    mov     #0x0800 ,  w1   ;select the location (0x800) to be written
    bset    w1 , #15        ;set the MSB of the base address, enable EDS mode

;Write a byte to the selected location
    mov     #0x00A5 , w2
    mov     #0x003C , w3
    mov.b   w2 , [w1++]         ;write Low byte
    mov.b   w3 , [w1++]     ;write High byte

;Write a word to the selected location
    mov     #0x1234   , w2  ;
    mov     w2 , [w1]       ;

;Write a Double - word to the selected location
    mov     #0x1122 , w2
    mov     #0x4455 , w3
    mov.d   w2 , [w1]       ;2 EDS writes
```

The page registers (DSRPAG/DSWPAG) do not update automatically while crossing a page boundary, when the rollover happens from 0xFFFF to 0x8000. While developing code in assembly, care must be taken to update the page registers when an Address Pointer crosses the page boundary. The 'C' compiler keeps track of the addressing, and increments or decrements the page registers accordingly while accessing contiguous data memory locations.

Note 1: All write operations to EDS are executed in a single cycle.

2: Use of Read/Modify/Write operation on any EDS location under a `REPEAT` instruction is not supported. For example, `BCLR, BSW, BTG, RLC f, RLNC f, RRC f, RRNC f, ADD f, SUB f, SUBR f, AND f, IOR f, XOR f, ASR f, ASL f`.

3: Use the DSRPAG register while performing Read/Modify/Write operation.

TABLE 4-35: EDS MEMORY ADDRESS WITH DIFFERENT PAGES AND ADDRESSES

DSRPAG (Data Space Read Register)	DSWPAG (Data Space Write Register)	Source/Destination Address while Indirect Addressing	24-Bit EA Pointing to EDS	Comment
x(1)	x(1)	0x0000 to 0x1FFF	0x000000 to 0x001FFF	Near data space(2)
		0x2000 to 0x7FFF	0x002000 to 0x007FFF	
0x001	0x001	0x8000 to 0xFFFF	0x008000 to 0x00FFFE	32 Kbytes on each page
0x002	0x002		0x010000 to 0x017FFE	
0x003	0x003		0x018000 to 0x0187FE	Only 2 Kbytes of extended SRAM on this page
0x004	0x004		0x018800 to 0x027FFE	EPMP memory space(4)
• • •	• • •		• • •	
0x1FF	0x1FF		0xFF8000 to 0xFFFFFE	
0x000	0x000		Invalid Address	Address error trap(3)

Note 1: If the source/destination address is below 0x8000, the DSRPAG and DSWPAG registers are not considered.

2: This data space can also be accessed by Direct Addressing.

3: When the source/destination address is above 0x8000 and DSRPAG/DSWPAG is '0', an address error trap will occur.

4: EPMP memory space can start from location, 0x008000, in the parts with 24 Kbytes of data memory (PIC24FJXXXDA1XX)

4.2.6 SOFTWARE STACK

Apart from its use as a working register, the W15 register in PIC24F devices is also used as a Software Stack Pointer (SSP). The pointer always points to the first available free word and grows from lower to higher addresses. It pre-decrements for stack pops and post-increments for stack pushes, as shown in Figure 4-7. Note that for a PC push during any CALL instruction, the MSB of the PC is zero-extended before the push, ensuring that the MSB is always clear.

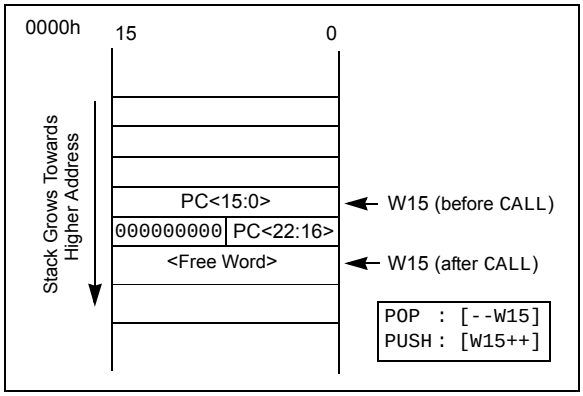
Note:	A PC push during exception processing will concatenate the SRL register to the MSB of the PC prior to the push.

The Stack Pointer Limit Value register (SPLIM), associated with the Stack Pointer, sets an upper address boundary for the stack. SPLIM is uninitialized at Reset. As is the case for the Stack Pointer, SPLIM<0> is forced to '0' as all stack operations must be word-aligned. Whenever an EA is generated using W15 as a source or destination pointer, the resulting address is compared with the value in SPLIM. If the contents of the Stack Pointer (W15) and the SPLIM register are equal, and a push operation is performed, a stack error trap will not occur. The stack error trap will occur on a subsequent push operation. Thus, for example, if it is desirable to cause a stack error trap when the stack grows beyond address 2000h in RAM, initialize the SPLIM with the value, 1FFEh.

Similarly, a Stack Pointer underflow (stack error) trap is generated when the Stack Pointer address is found to be less than 0800h. This prevents the stack from interfering with the SFR space.

A write to the SPLIM register should not be immediately followed by an indirect read operation using W15.

FIGURE 4-7: CALL STACK FRAME



4.3 Interfacing Program and Data Memory Spaces

The PIC24F architecture uses a 24-bit wide program space and 16-bit wide data space. The architecture is also a modified Harvard scheme, meaning that data can also be present in the program space. To use this data successfully, it must be accessed in a way that preserves the alignment of information in both spaces.

Aside from normal execution, the PIC24F architecture provides two methods by which program space can be accessed during operation:

- Using table instructions to access individual bytes or words anywhere in the program space
- Remapping a portion of the program space into the data space (program space visibility)

Table instructions allow an application to read or write to small areas of the program memory. This makes the method ideal for accessing data tables that need to be updated from time to time. It also allows access to all bytes of the program word. The remapping method allows an application to access a large block of data on a read-only basis, which is ideal for look ups from a large table of static data. It can only access the least significant word of the program word.

4.3.1 ADDRESSING PROGRAM SPACE

Since the address ranges for the data and program spaces are 16 and 24 bits, respectively, a method is needed to create a 23-bit or 24-bit program address from 16-bit data registers. The solution depends on the interface method to be used.

For table operations, the 8-bit Table Memory Page Address register (TBLPAG) is used to define a 32K word region within the program space. This is concatenated with a 16-bit EA to arrive at a full 24-bit program space address. In this format, the MSBs of TBLPAG is used to determine if the operation occurs in the user memory (TBLPAG<7> = 0) or the configuration memory (TBLPAG<7> = 1).

For remapping operations, the 10-bit Extended Data Space Read register (DSRPAG) is used to define a 16K word page in the program space. When the Most Significant bit (MSb) of the EA is '1', and the MSb (bit 9) of DSRPAG is '1', the lower 8 bits of DSRPAG are concatenated with the lower 15 bits of the EA to form a 23-bit program space address. The DSRPAG<8> bit decides whether the lower word (when bit is '0') or the higher word (when bit is '1') of program memory is mapped. Unlike table operations, this strictly limits remapping operations to the user memory area.

Table 4-36 and Figure 4-8 show how the program EA is created for table operations and remapping accesses from the data EA. Here, P<23:0> refers to a program space word, whereas D<15:0> refers to a data space word.

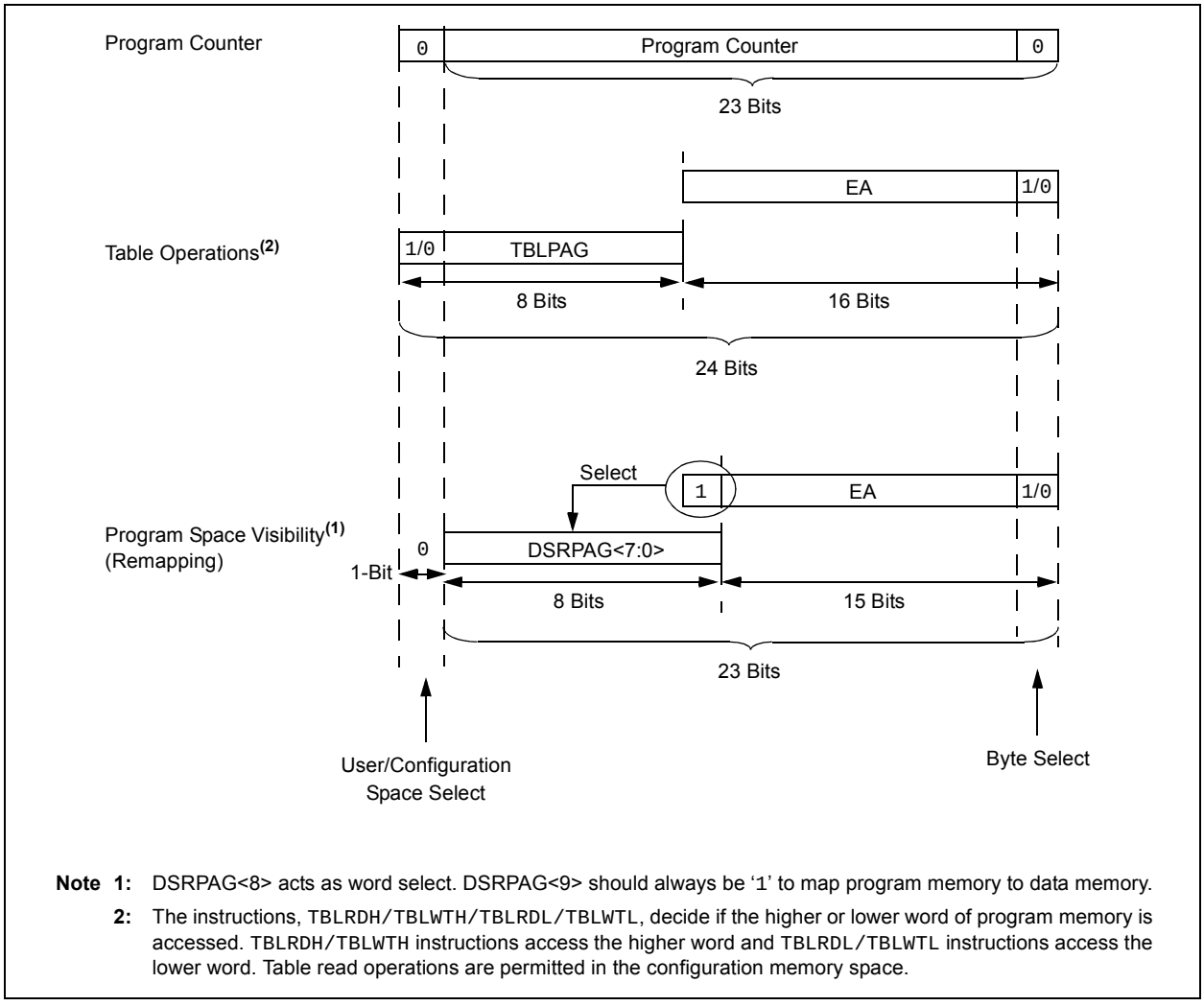
TABLE 4-36: PROGRAM SPACE ADDRESS CONSTRUCTION

Access Type	Access Space	Program Space Address				
		<23>	<22:16>	<15>	<14:1>	<0>
Instruction Access (Code Execution)	User	0	PC<22:1>			0
		0xx xxxx xxxx xxxx xxxx xxx0				
TBLRD/TBLWT (Byte/Word Read/Write)	User	TBLPAG<7:0>		Data EA<15:0>		
		0xxx xxxx		xxxx xxxx xxxx xxxx		
	Configuration	TBLPAG<7:0>		Data EA<15:0>		
		1xxx xxxx		xxxx xxxx xxxx xxxx		
Program Space Visibility (Block Remap/Read)	User	0	DSRPAG<7:0>(2)		Data EA<14:0>(1)	
		0	xxxx xxxx		xxx xxxx xxxx xxxx	

Note 1: Data EA<15> is always '1' in this case, but is not used in calculating the program space address. Bit 15 of the address is DSRPAG<0>.

2: DSRPAG<9> is always '1' in this case. DSRPAG<8> decides whether the lower word or higher word of program memory is read. When DSRPAG<8> is '0', the lower word is read and when it is '1', the higher word is read.

FIGURE 4-8: DATA ACCESS FROM PROGRAM SPACE ADDRESS GENERATION



Note 1: DSRPAG<8> acts as word select. DSRPAG<9> should always be '1' to map program memory to data memory.

2: The instructions, TBLRDH/TBLWTH/TBLRDL/TBLWTL, decide if the higher or lower word of program memory is accessed. TBLRDH/TBLWTH instructions access the higher word and TBLRDL/TBLWTL instructions access the lower word. Table read operations are permitted in the configuration memory space.

4.3.2 DATA ACCESS FROM PROGRAM MEMORY USING TABLE INSTRUCTIONS

The TBLRDL and TBLWTL instructions offer a direct method of reading or writing the lower word of any address within the program space without going through data space. The TBLRDH and TBLWTH instructions are the only method to read or write the upper 8 bits of a program space word as data.

The PC is incremented by two for each successive 24-bit program word. This allows program memory addresses to directly map to data space addresses. Program memory can thus be regarded as two, 16-bit word-wide address spaces, residing side by side, each with the same address range. TBLRDL and TBLWTL access the space which contains the least significant data word, and TBLRDH and TBLWTH access the space which contains the upper data byte.

Two table instructions are provided to move byte or word-sized (16-bit) data to and from program space. Both function as either byte or word operations.

1. TBLRDL (Table Read Low): In Word mode, it maps the lower word of the program space location (P<15:0>) to a data address (D<15:0>).

   In Byte mode, either the upper or lower byte of the lower program word is mapped to the lower byte of a data address. The upper byte is selected when byte select is '1'; the lower byte is selected when it is '0'.

2. TBLRDH (Table Read High): In Word mode, it maps the entire upper word of a program address (P<23:16>) to a data address. Note that D<15:8>, the 'phantom' byte, will always be '0'.

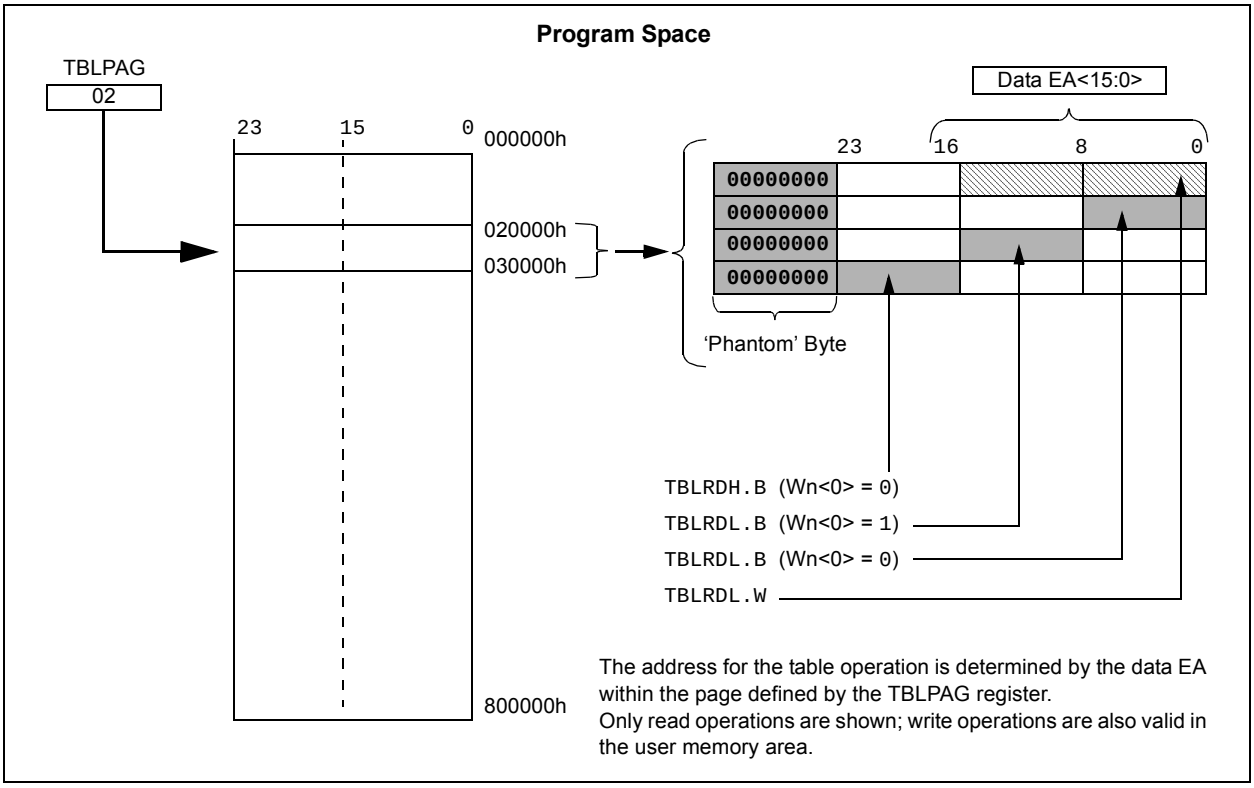
   In Byte mode, it maps the upper or lower byte of the program word to D<7:0> of the data address, as above. Note that the data will always be '0' when the upper 'phantom' byte is selected (byte select = 1).

In a similar fashion, two table instructions, TBLWTH and TBLWTL, are used to write individual bytes or words to a program space address. The details of their operation are described in **Section 5.0 "Flash Program Memory"**.

For all table operations, the area of program memory space to be accessed is determined by the Table Memory Page Address register (TBLPAG). TBLPAG covers the entire program memory space of the device, including user and configuration spaces. When TBLPAG<7> = 0, the table page is located in the user memory space. When TBLPAG<7> = 1, the page is located in configuration space.

Note:	Only table read operations will execute in the configuration memory space, where Device IDs are located. Table write operations are not allowed.

FIGURE 4-9: ACCESSING PROGRAM MEMORY WITH TABLE INSTRUCTIONS



The address for the table operation is determined by the data EA within the page defined by the TBLPAG register.
Only read operations are shown; write operations are also valid in the user memory area.

4.3.3 READING DATA FROM PROGRAM MEMORY USING EDS

The upper 32 Kbytes of data space may optionally be mapped into any 16K word page of the program space. This provides transparent access of stored constant data from the data space without the need to use special instructions (i.e., TBLRDL/H).

Program space access through the data space occurs when the MSb of EA is '1' and the DSRPAG<9> is also '1'. The lower 8 bits of DSRPAG are concatenated to the Wn<14:0> bits to form a 23-bit EA to access program memory. The DSRPAG<8> decides which word should be addressed; when the bit is '0', the lower word and when '1', the upper word of the program memory is accessed.

The entire program memory is divided into 512 EDS pages, from 0x200 to 0x3FF, each consisting of 16K words of data. Pages, 0x200 to 0x2FF, correspond to the lower words of the program memory, while 0x300 to 0x3FF correspond to the upper words of the program memory.

Using this EDS technique, the entire program memory can be accessed. Previously, the access to the upper word of the program memory was not supported.

Table 4-37 provides the corresponding 23-bit EDS address for program memory with EDS page and source addresses.

For operations that use PSV and are executed outside a REPEAT loop, the MOV and MOV.D instructions will require one instruction cycle in addition to the specified execution time. All other instructions will require two instruction cycles in addition to the specified execution time.

For operations that use PSV, which are executed inside a REPEAT loop, there will be some instances that require two instruction cycles in addition to the specified execution time of the instruction:

- Execution in the first iteration
- Execution in the last iteration
- Execution prior to exiting the loop due to an interrupt
- Execution upon re-entering the loop after an interrupt is serviced

Any other iteration of the REPEAT loop will allow the instruction accessing data, using PSV, to execute in a single cycle.

TABLE 4-37: EDS PROGRAM ADDRESS WITH DIFFERENT PAGES AND ADDRESSES

DSRPAG (Data Space Read Register)	Source Address while Indirect Addressing	23-Bit EA Pointing to EDS	Comment
0x200	0x8000 to 0xFFFF	0x000000 to 0x007FFE	Lower words of 4M program instructions; (8 Mbytes) for read operations only.
• • •		• • •	
0x2FF		0x7F8000 to 0x7FFFFE	
0x300		0x000001 to 0x007FFF	Upper words of 4M program instructions (4 Mbytes remaining, 4 Mbytes are phantom bytes) for read operations only.
• • •		• • •	
0x3FF		0x7F8001 to 0x7FFFFF	
0x000		Invalid Address	Address error trap(1)

Note 1: When the source/destination address is above 0x8000 and DSRPAG/DSWPAG is '0', an address error trap will occur.

FIGURE 4-10: PROGRAM SPACE VISIBILITY OPERATION TO ACCESS LOWER WORD

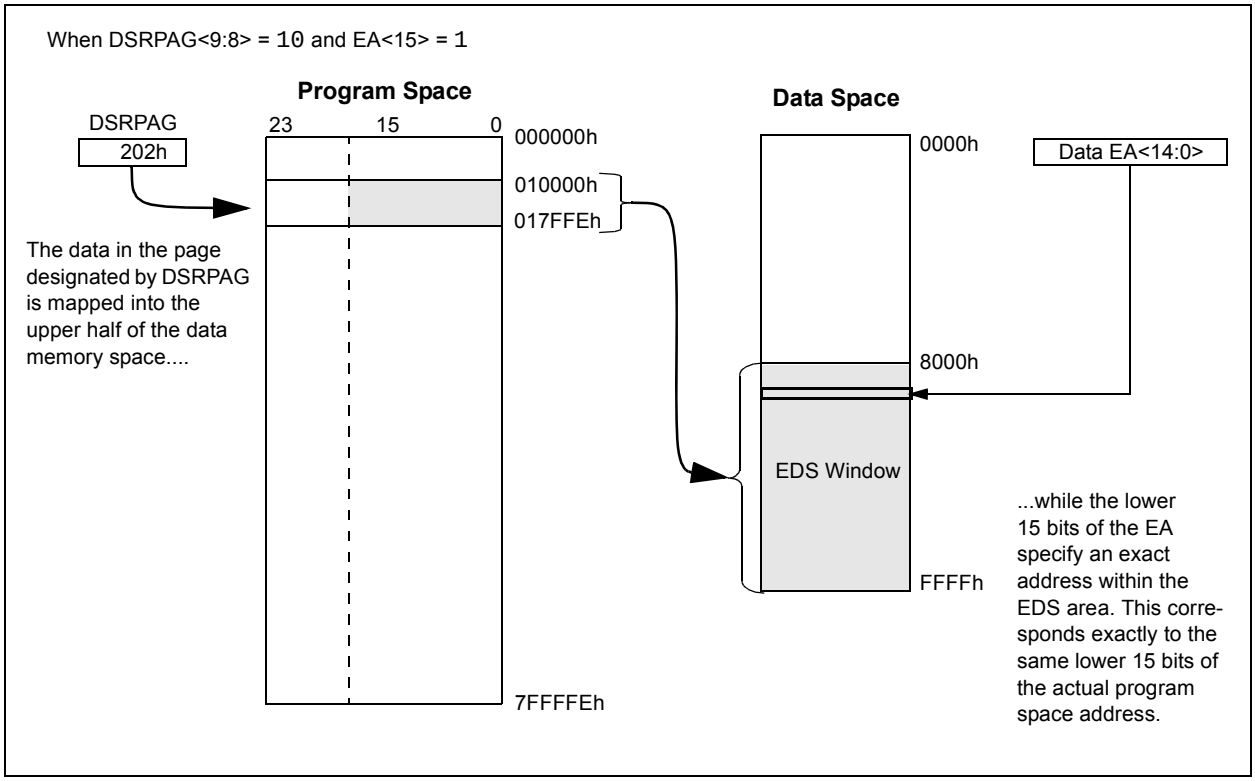
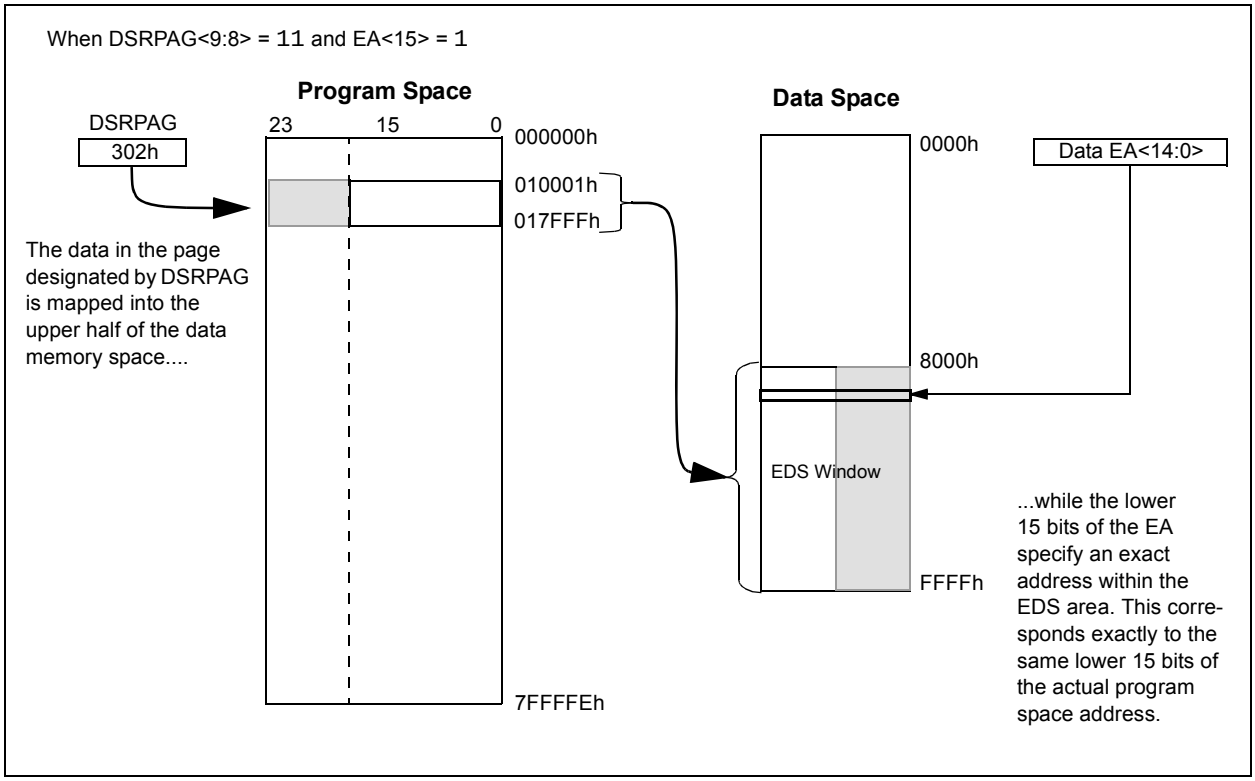


FIGURE 4-11: PROGRAM SPACE VISIBILITY OPERATION TO ACCESS HIGHER WORD



EXAMPLE 4-3: EDS READ CODE FROM PROGRAM MEMORY IN ASSEMBLY

```
; Set the EDS page from where the data to be read
    mov     #0x0202 , w0
    mov     w0 , DSRPAG               ;page 0x202, consisting lower words, is selected for read
    mov     #0x000A ,  w1             ;select the location (0x0A) to be read
    bset    w1 , #15                  ;set the MSB of the base address, enable EDS mode

;Read a byte from the selected location
    mov.b   [w1++] , w2               ;read Low byte
    mov.b   [w1++] , w3               ;read High byte

;Read a word from the selected location
    mov     [w1] , w2                 ;

;Read Double - word from the selected location
    mov.d   [w1] , w2                 ;two word read, stored in w2 and w3
```

5.0 FLASH PROGRAM MEMORY

Note: This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the *"PIC24F Family Reference Manual"*, **Section 4. "Program Memory"** (DS39715). The information in this data sheet supersedes the information in the FRM.

The PIC24FJ256DA210 family of devices contains internal Flash program memory for storing and executing application code. The program memory is readable, writable and erasable. The Flash can be programmed in four ways:

- In-Circuit Serial Programming™ (ICSP™)
- Run-Time Self-Programming (RTSP)
- JTAG
- Enhanced In-Circuit Serial Programming (Enhanced ICSP)

ICSP allows a PIC24FJ256DA210 family device to be serially programmed while in the end application circuit. This is simply done with two lines for the programming clock and programming data (named PGECx and PGEDx, respectively), and three other lines for power (V_{DD}), ground (V_{SS}) and Master Clear ($\overline{MCLR}$). This allows customers to manufacture boards with unprogrammed devices and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

RTSP is accomplished using TBLRD (table read) and TBLWT (table write) instructions. With RTSP, the user may write program memory data in blocks of 64 instructions (192 bytes) at a time and erase program memory in blocks of 512 instructions (1536 bytes) at a time.

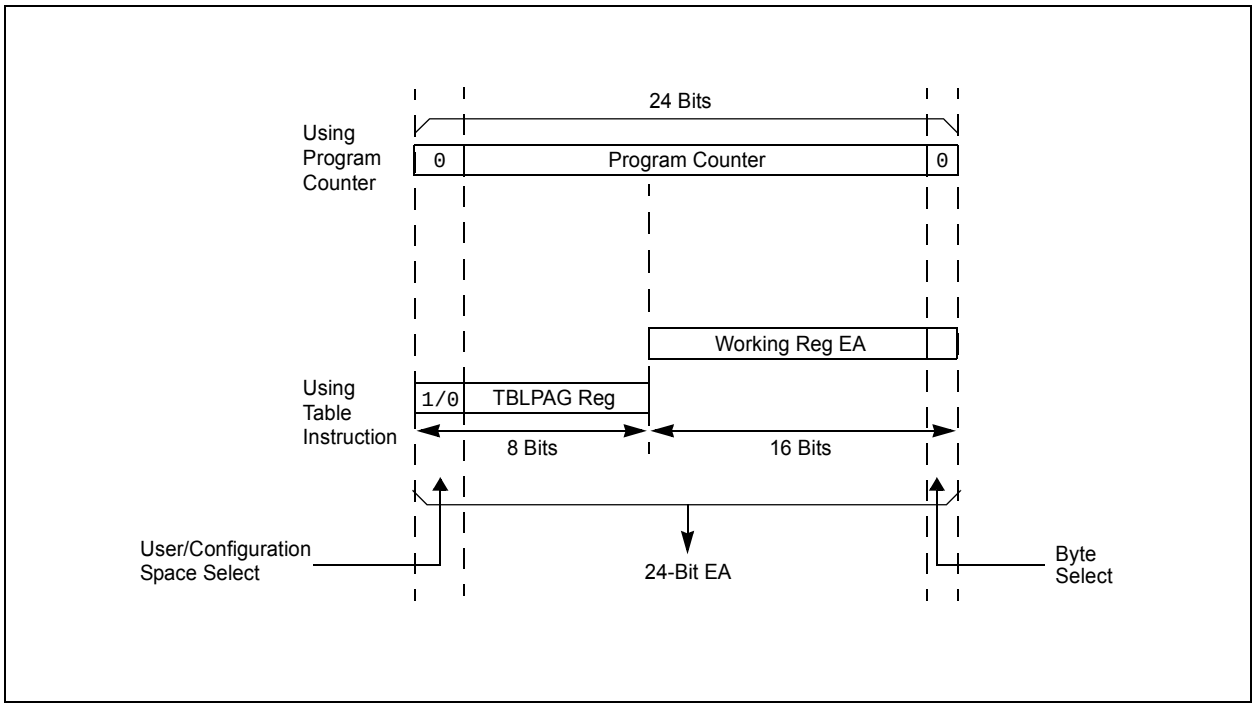
5.1 Table Instructions and Flash Programming

Regardless of the method used, all programming of Flash memory is done with the table read and write instructions. These allow direct read and write access to the program memory space from the data memory while the device is in normal operating mode. The 24-bit target address in the program memory is formed using the TBLPAG<7:0> bits and the Effective Address (EA) from a W register, specified in the table instruction, as shown in Figure 5-1.

The TBLRDL and the TBLWTL instructions are used to read or write to bits<15:0> of program memory. TBLRDL and TBLWTL can access program memory in both Word and Byte modes.

The TBLRDH and TBLWTH instructions are used to read or write to bits<23:16> of program memory. TBLRDH and TBLWTH can also access program memory in Word or Byte mode.

FIGURE 5-1: ADDRESSING FOR TABLE REGISTERS



5.2 RTSP Operation

The PIC24F Flash program memory array is organized into rows of 64 instructions or 192 bytes. RTSP allows the user to erase blocks of eight rows (512 instructions) at a time and to program one row at a time. It is also possible to program single words.

The 8-row erase blocks and single row write blocks are edge-aligned, from the beginning of program memory, on boundaries of 1536 bytes and 192 bytes, respectively.

When data is written to program memory using TBLWT instructions, the data is not written directly to memory. Instead, data written using table writes is stored in holding latches until the programming sequence is executed.

Any number of TBLWT instructions can be executed and a write will be successfully performed. However, 64 TBLWT instructions are required to write the full row of memory.

To ensure that no data is corrupted during a write, any unused address should be programmed with FFFFFFh. This is because the holding latches reset to an unknown state, so if the addresses are left in the Reset state, they may overwrite the locations on rows which were not rewritten.

The basic sequence for RTSP programming is to set up a Table Pointer, then do a series of TBLWT instructions to load the buffers. Programming is performed by setting the control bits in the NVMCON register.

Data can be loaded in any order and the holding registers can be written to multiple times before performing a write operation. Subsequent writes, however, will wipe out any previous writes.

Note:	Writing to a location multiple times without erasing is *not* recommended.

All of the table write operations are single-word writes (2 instruction cycles), because only the buffers are written. A programming cycle is required for programming each row.

5.3 JTAG Operation

The PIC24F family supports JTAG boundary scan. Boundary scan can improve the manufacturing process by verifying pin to PCB connectivity.

5.4 Enhanced In-Circuit Serial Programming

Enhanced In-Circuit Serial Programming uses an on-board bootloader, known as the program executive, to manage the programming process. Using an SPI data frame format, the program executive can erase, program and verify program memory. For more information on Enhanced ICSP, see the device programming specification.

5.5 Control Registers

There are two SFRs used to read and write the program Flash memory: NVMCON and NVMKEY.

The NVMCON register (Register 5-1) controls which blocks are to be erased, which memory type is to be programmed and when the programming cycle starts.

NVMKEY is a write-only register that is used for write protection. To start a programming or erase sequence, the user must consecutively write 55h and AAh to the NVMKEY register. Refer to **Section 5.6 "Programming Operations"** for further details.

5.6 Programming Operations

A complete programming sequence is necessary for programming or erasing the internal Flash in RTSP mode. During a programming or erase operation, the processor stalls (waits) until the operation is finished. Setting the WR bit (NVMCON<15>) starts the operation and the WR bit is automatically cleared when the operation is finished.

REGISTER 5-1: NVMCON: FLASH MEMORY CONTROL REGISTER

R/S-0, HC(1)	R/W-0(1)	R-0, HSC(1)	U-0	U-0	U-0	U-0	U-0
WR	WREN	WRERR	—	—	—	—	—
bit 15							bit 8

U-0	R/W-0(1)	U-0	U-0	R/W-0(1)	R/W-0(1)	R/W-0(1)	R/W-0(1)
—	ERASE	—	—	NVMOP3(2)	NVMOP2(2)	NVMOP1(2)	NVMOP0(2)
bit 7							bit 0

Legend:	S = Settable bit	HSC = Hardware Settable/Clearable bit	
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown
HC = Hardware Clearable bit			

bit 15 **WR:** Write Control bit(1)
- 1 = Initiates a Flash memory program or erase operation; the operation is self-timed and the bit is cleared by hardware once the operation is complete
- 0 = Program or erase operation is complete and inactive

bit 14 **WREN:** Write Enable bit(1)
- 1 = Enable Flash program/erase operations
- 0 = Inhibit Flash program/erase operations

bit 13 **WRERR:** Write Sequence Error Flag bit(1)
- 1 = An improper program or erase sequence attempt or termination has occurred (bit is set automatically on any set attempt of the WR bit)
- 0 = The program or erase operation completed normally

bit 12-7 **Unimplemented:** Read as '0'

bit 6 **ERASE:** Erase/Program Enable bit(1)
- 1 = Perform the erase operation specified by NVMOP<3:0> on the next WR command
- 0 = Perform the program operation specified by NVMOP<3:0> on the next WR command

bit 5-4 **Unimplemented:** Read as '0'

bit 3-0 **NVMOP<3:0>:** NVM Operation Select bits(1,2)
- 1111 = Memory bulk erase operation (ERASE = 1) or no operation (ERASE = 0)(3)
- 0011 = Memory word program operation (ERASE = 0) or no operation (ERASE = 1)
- 0010 = Memory page erase operation (ERASE = 1) or no operation (ERASE = 0)
- 0001 = Memory row program operation (ERASE = 0) or no operation (ERASE = 1)

Note 1: These bits can only be reset on POR.

2: All other combinations of NVMOP<3:0> are unimplemented.

3: Available in ICSP™ mode only; refer to the device programming specification.

5.6.1 PROGRAMMING ALGORITHM FOR FLASH PROGRAM MEMORY

The user can program one row of Flash program memory at a time. To do this, it is necessary to erase the 8-row erase block containing the desired row. The general process is:

1. Read eight rows of program memory (512 instructions) and store in data RAM.
2. Update the program data in RAM with the desired new data.
3. Erase the block (see Example 5-1):
   a) Set the NVMOP bits (NVMCON<3:0>) to '0010' to configure for block erase. Set the ERASE (NVMCON<6>) and WREN (NVMCON<14>) bits.
   b) Write the starting address of the block to be erased into the TBLPAG and W registers.
   c) Write 55h to NVMKEY.
   d) Write AAh to NVMKEY.
   e) Set the WR bit (NVMCON<15>). The erase cycle begins and the CPU stalls for the duration of the erase cycle. When the erase is done, the WR bit is cleared automatically.
4. Write the first 64 instructions from data RAM into the program memory buffers (see Example 5-3).
5. Write the program block to Flash memory:
   a) Set the NVMOP bits to '0001' to configure for row programming. Clear the ERASE bit and set the WREN bit.
   b) Write 55h to NVMKEY.
   c) Write AAh to NVMKEY.
   d) Set the WR bit. The programming cycle begins and the CPU stalls for the duration of the write cycle. When the write to Flash memory is done, the WR bit is cleared automatically.
6. Repeat steps 4 and 5, using the next available 64 instructions from the block in data RAM by incrementing the value in TBLPAG, until all 512 instructions are written back to Flash memory.

For protection against accidental operations, the write initiate sequence for NVMKEY must be used to allow any erase or program operation to proceed. After the programming command has been executed, the user must wait for the programming time until programming is complete. The two instructions following the start of the programming sequence should be NOPs, as shown in Example 5-4.

EXAMPLE 5-1: ERASING A PROGRAM MEMORY BLOCK (ASSEMBLY LANGUAGE CODE)

```
; Set up NVMCON for block erase operation
    MOV #0x4042, W0 ;
    MOV W0, NVMCON                    ; Initialize NVMCON
; Init pointer to row to be ERASED
    MOV #tblpage(PROG_ADDR), W0       ;
    MOV W0, TBLPAG                    ; Initialize Program Memory (PM) Page Boundary SFR
    MOV #tbloffset(PROG_ADDR), W0     ; Initialize in-page EA<15:0> pointer
    TBLWTL W0, [W0]                   ; Set base address of erase block
    DISI #5                           ; Block all interrupts with priority <7
                                      ; for next 5 instructions
    MOV.B #0x55, W0
    MOV W0, NVMKEY                    ; Write the 0x55 key
    MOV.B #0xAA, W1 ;
    MOV W1, NVMKEY                    ; Write the 0xAA key
    BSET NVMCON, #WR                  ; Start the erase sequence
    NOP                               ; Insert two NOPs after the erase
    NOP                               ; command is asserted
```

EXAMPLE 5-2: ERASING A PROGRAM MEMORY BLOCK ('C' LANGUAGE CODE)

```
// C example using MPLAB C30
    unsigned long progAddr = 0xXXXXXX;          // Address of row to write
    unsigned int offset;
//Set up pointer to the first memory location to be written
    TBLPAG = progAddr>>16;                      // Initialize PM Page Boundary SFR
    offset = progAddr & 0xFFFF;                 // Initialize lower word of address
    __builtin_tblwtl(offset, 0x0000);           // Set base address of erase block
                                                // with dummy latch write
    NVMCON = 0x4042;                            // Initialize NVMCON
    asm("DISI #5");                             // Block all interrupts with priority <7
                                                // for next 5 instructions
    __builtin_write_NVM();                      // check function to perform unlock
                                                // sequence and set WR
```

EXAMPLE 5-3: LOADING THE WRITE BUFFERS

```
; Set up NVMCON for row programming operations
       MOV     #0x4001, W0                      ;
       MOV     W0, NVMCON                       ; Initialize NVMCON
; Set up a pointer to the first program memory location to be written
; program memory selected, and writes enabled
       MOV     #0x0000, W0                      ;
       MOV     W0, TBLPAG                       ; Initialize PM Page Boundary SFR
       MOV     #0x6000, W0                      ; An example program memory address
; Perform the TBLWT instructions to write the latches
; 0th_program_word
       MOV     #LOW_WORD_0, W2                  ;
       MOV     #HIGH_BYTE_0, W3                 ;
       TBLWTL  W2, [W0]                         ; Write PM low word into program latch
       TBLWTH  W3, [W0++]                       ; Write PM high byte into program latch
; 1st_program_word
       MOV     #LOW_WORD_1, W2                  ;
       MOV     #HIGH_BYTE_1, W3                 ;
       TBLWTL  W2, [W0]                         ; Write PM low word into program latch
       TBLWTH  W3, [W0++]                       ; Write PM high byte into program latch
;  2nd_program_word
       MOV     #LOW_WORD_2, W2                  ;
       MOV     #HIGH_BYTE_2, W3                 ;
       TBLWTL  W2, [W0]                         ; Write PM low word into program latch
       TBLWTH  W3, [W0++]                       ; Write PM high byte into program latch
       •
       •
       •
; 63rd_program_word
       MOV     #LOW_WORD_63, W2                 ;
       MOV     #HIGH_BYTE_63, W3                ;
       TBLWTL  W2, [W0]                         ; Write PM low word into program latch
       TBLWTH  W3, [W0]                         ; Write PM high byte into program latch
```

EXAMPLE 5-4: INITIATING A PROGRAMMING SEQUENCE

```
       DISI    #5                          ; Block all interrupts with priority <7
                                           ; for next 5 instructions
       MOV.B   #0x55, W0
       MOV     W0, NVMKEY                  ; Write the 0x55 key
       MOV.B    #0xAA, W1                  ;
       MOV     W1, NVMKEY                  ; Write the 0xAA key
       BSET    NVMCON, #WR                 ; Start the programming sequence
       NOP                                 ; Required delays
       NOP
       BTSC    NVMCON, #15                 ; and wait for it to be
       BRA     $-2                         ; completed
```

5.6.2 PROGRAMMING A SINGLE WORD OF FLASH PROGRAM MEMORY

If a Flash location has been erased, it can be programmed using table write instructions to write an instruction word (24-bit) into the write latch. The TBLPAG register is loaded with the 8 Most Significant Bytes (MSB) of the Flash address. The TBLWTL and TBLWTH instructions write the desired data into the write latches and specify the lower 16 bits of the program memory address to write to. To configure the NVMCON register for a word write, set the NVMOP bits (NVMCON<3:0>) to '0011'. The write is performed by executing the unlock sequence and setting the WR bit (see Example 5-5). An equivalent procedure in 'C' compiler, using the MPLAB C30 compiler and built-in hardware functions is shown in Example 5-6.

EXAMPLE 5-5: PROGRAMMING A SINGLE WORD OF FLASH PROGRAM MEMORY

```
; Setup a pointer to data Program Memory
    MOV     #tblpage(PROG_ADDR), W0       ;
    MOV     W0, TBLPAG                    ;Initialize PM Page Boundary SFR
    MOV     #tbloffset(PROG_ADDR), W0     ;Initialize a register with program memory address

    MOV     #LOW_WORD_N, W2               ;
    MOV     #HIGH_BYTE_N, W3              ;
    TBLWTL  W2, [W0]                      ; Write PM low word into program latch
    TBLWTH  W3, [W0++]                    ; Write PM high byte into program latch

; Setup NVMCON for programming one word to data Program Memory
    MOV     #0x4003, W0                   ;
    MOV     W0, NVMCON                    ; Set NVMOP bits to 0011

    DISI    #5                            ; Disable interrupts while the KEY sequence is written
    MOV.B   #0x55, W0                     ; Write the key sequence
    MOV     W0, NVMKEY
    MOV.B   #0xAA, W0
    MOV     W0, NVMKEY
    BSET    NVMCON, #WR                   ; Start the write cycle
    NOP                                   ; Required delays
    NOP
```

EXAMPLE 5-6: PROGRAMMING A SINGLE WORD OF FLASH PROGRAM MEMORY ('C' LANGUAGE CODE)

```
// C example using MPLAB C30
unsigned int offset;
unsigned long progAddr = 0xXXXXXX;          // Address of word to program
unsigned int progDataL = 0xXXXX;            // Data to program lower word
unsigned char progDataH = 0xXX;             // Data to program upper byte

//Set up NVMCON for word programming
NVMCON = 0x4003;                            // Initialize NVMCON

//Set up pointer to the first memory location to be written
TBLPAG = progAddr>>16;                      // Initialize PM Page Boundary SFR
offset = progAddr & 0xFFFF;                 // Initialize lower word of address

//Perform TBLWT instructions to write latches
__builtin_tblwtl(offset, progDataL);        // Write to address low word
__builtin_tblwth(offset, progDataH);        // Write to upper byte
asm("DISI #5");                             // Block interrupts with priority <7
                                            // for next 5 instructions
__builtin_write_NVM();                      // C30 function to perform unlock
                                            // sequence and set WR
```

6.0 RESETS

Note: This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the "*PIC24F Family Reference Manual*", **Section 7. "Reset"** (DS39712). The information in this data sheet supersedes the information in the FRM.

The Reset module combines all Reset sources and controls the device Master Reset Signal, SYSRST. The following is a list of device Reset sources:

- POR: Power-on Reset
- MCLR: Pin Reset
- SWR: RESET Instruction
- WDT: Watchdog Timer Reset
- BOR: Brown-out Reset
- CM: Configuration Mismatch Reset
- TRAPR: Trap Conflict Reset
- IOPUWR: Illegal Opcode Reset
- UWR: Uninitialized W Register Reset

A simplified block diagram of the Reset module is shown in Figure 6-1.

Any active source of Reset will make the SYSRST signal active. Many registers associated with the CPU and peripherals are forced to a known Reset state. Most registers are unaffected by a Reset; their status is unknown on POR and unchanged by all other Resets.

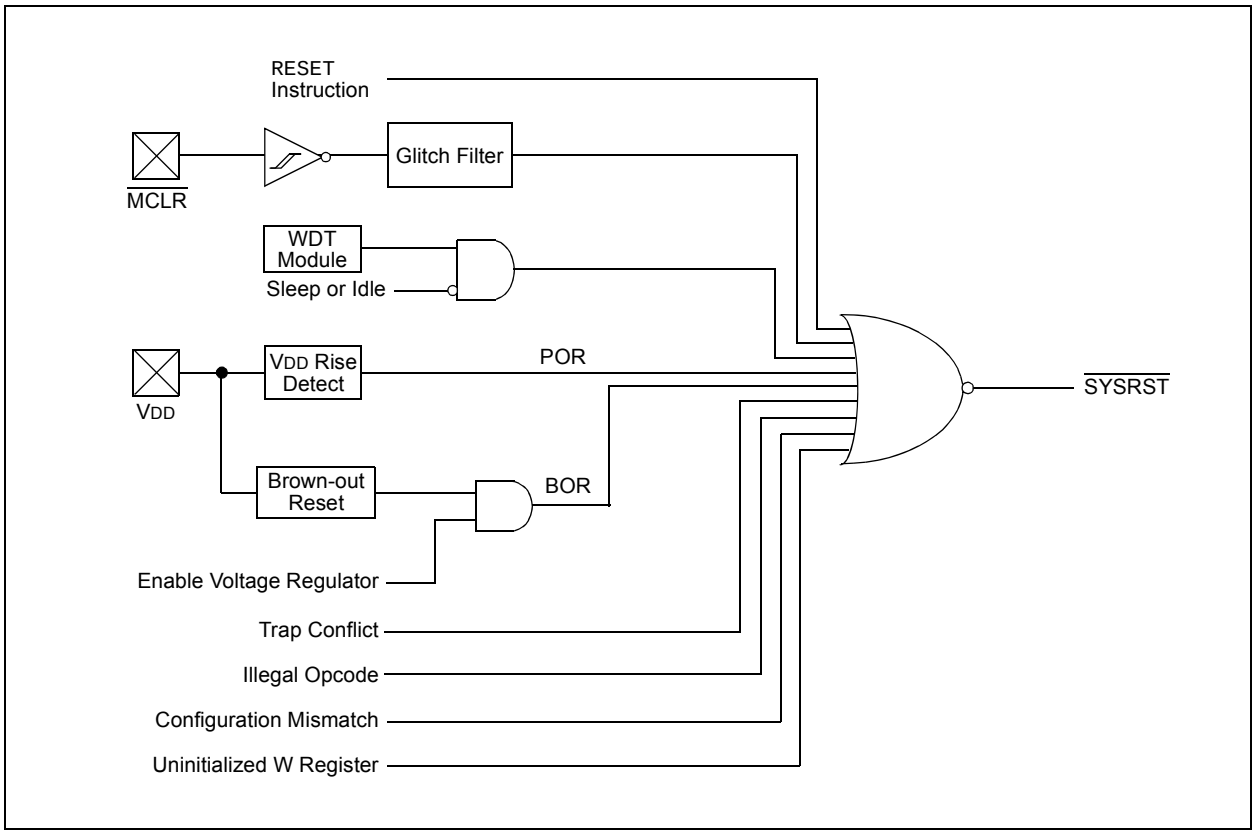
Note: Refer to the specific peripheral or CPU section of this manual for register Reset states.

All types of device Reset will set a corresponding status bit in the RCON register to indicate the type of Reset (see Register 6-1). A POR will clear all bits, except for the BOR and POR (RCON<1:0>) bits, which are set. The user may set or clear any bit at any time during code execution. The RCON bits only serve as status bits. Setting a particular Reset status bit in software will not cause a device Reset to occur.

The RCON register also has other bits associated with the Watchdog Timer and device power-saving states. The function of these bits is discussed in other sections of this data sheet.

Note: The status bits in the RCON register should be cleared after they are read so that the next RCON register value after a device Reset will be meaningful.

FIGURE 6-1: RESET SYSTEM BLOCK DIAGRAM



REGISTER 6-1: RCON: RESET CONTROL REGISTER(1)

R/W-0, HS	R/W-0, HS	U-0	U-0	U-0	U-0	R/W-0, HS	R/W-0
TRAPR	IOPUWR	—	—	—	—	CM	VREGS(3)
bit 15							bit 8

R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-1, HS	R/W-1, HS
EXTR	SWR	SWDTEN(2)	WDTO	SLEEP	IDLE	BOR	POR
bit 7							bit 0

Legend:	HS = Hardware Settable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **TRAPR:** Trap Reset Flag bit
1 = A Trap Conflict Reset has occurred
0 = A Trap Conflict Reset has not occurred

bit 14 **IOPUWR:** Illegal Opcode or Uninitialized W Access Reset Flag bit
1 = An illegal opcode detection, an illegal address mode or uninitialized W register is used as an Address Pointer and caused a Reset
0 = An illegal opcode or uninitialized W Reset has not occurred

bit 13-10 **Unimplemented:** Read as '0'

bit 9 **CM:** Configuration Word Mismatch Reset Flag bit
1 = A Configuration Word Mismatch Reset has occurred
0 = A Configuration Word Mismatch Reset has not occurred

bit 8 **VREGS:** Voltage Regulator Standby Enable bit(3)
1 = Program memory and regulator remain active during Sleep/Idle
0 = Program memory power is removed and regulator goes to standby during Seep/Idle

bit 7 **EXTR:** External Reset ($\overline{MCLR}$) Pin bit
1 = A Master Clear (pin) Reset has occurred
0 = A Master Clear (pin) Reset has not occurred

bit 6 **SWR:** Software Reset (Instruction) Flag bit
1 = A RESET instruction has been executed
0 = A RESET instruction has not been executed

bit 5 **SWDTEN:** Software Enable/Disable of WDT bit(2)
1 = WDT is enabled
0 = WDT is disabled

bit 4 **WDTO:** Watchdog Timer Time-out Flag bit
1 = WDT time-out has occurred
0 = WDT time-out has not occurred

bit 3 **SLEEP:** Wake From Sleep Flag bit
1 = Device has been in Sleep mode
0 = Device has not been in Sleep mode

Note 1: All of the Reset status bits may be set or cleared in software. Setting one of these bits in software does not cause a device Reset.

2: If the FWDTEN Configuration bit is '1' (unprogrammed), the WDT is always enabled, regardless of the SWDTEN bit setting.

3: Re-enabling the regulator after it enters Standby mode will add a delay, T_{VREG}, when waking up from Sleep. Applications that do not use the voltage regulator should set this bit to prevent this delay from occurring.

REGISTER 6-1: RCON: RESET CONTROL REGISTER(1) (CONTINUED)

bit 2 **IDLE:** Wake-up From Idle Flag bit
1 = Device has been in Idle mode
0 = Device has not been in Idle mode

bit 1 **BOR:** Brown-out Reset Flag bit
1 = A Brown-out Reset has occurred
Note that BOR is also set after a Power-on Reset.
0 = A Brown-out Reset has not occurred

bit 0 **POR:** Power-on Reset Flag bit
1 = A Power-on Reset has occurred
0 = A Power-on Reset has not occurred

Note 1: All of the Reset status bits may be set or cleared in software. Setting one of these bits in software does not cause a device Reset.

2: If the FWDTEN Configuration bit is '1' (unprogrammed), the WDT is always enabled, regardless of the SWDTEN bit setting.

3: Re-enabling the regulator after it enters Standby mode will add a delay, TVREG, when waking up from Sleep. Applications that do not use the voltage regulator should set this bit to prevent this delay from occurring.

TABLE 6-1: RESET FLAG BIT OPERATION

Flag Bit	Setting Event	Clearing Event
TRAPR (RCON<15>)	Trap Conflict Event	POR
IOPUWR (RCON<14>)	Illegal Opcode or Uninitialized W Register Access	POR
CM (RCON<9>)	Configuration Mismatch Reset	POR
EXTR (RCON<7>)	$\overline{\text{MCLR}}$ Reset	POR
SWR (RCON<6>)	RESET Instruction	POR
WDTO (RCON<4>)	WDT Time-out	CLRWDT, PWRSAV Instruction, POR
SLEEP (RCON<3>)	PWRSAV #0 Instruction	POR
IDLE (RCON<2>)	PWRSAV #1 Instruction	POR
BOR (RCON<1>)	POR, BOR	—
POR (RCON<0>)	POR	—

Note: All Reset flag bits may be set or cleared by the user software.

6.1 Special Function Register Reset States

Most of the Special Function Registers (SFRs) associated with the PIC24F CPU and peripherals are reset to a particular value at a device Reset. The SFRs are grouped by their peripheral or CPU function and their Reset values are specified in each section of this manual.

The Reset value for each SFR does not depend on the type of Reset, with the exception of four registers. The Reset value for the Reset Control register, RCON, will depend on the type of device Reset. The Reset value for the Oscillator Control register, OSCCON, will depend on the type of Reset and the programmed values of the FNOSC bits in Flash Configuration Word 2 (CW2) (see Table 6-2). The RCFGCAL and NVMCON registers are only affected by a POR.

6.2 Device Reset Times

The Reset times for various types of device Reset are summarized in Table 6-3. Note that the system Reset signal, $\overline{\text{SYSRST}}$, is released after the POR delay time expires.

The time at which the device actually begins to execute code will also depend on the system oscillator delays, which include the Oscillator Start-up Timer (OST) and the PLL lock time. The OST and PLL lock times occur in parallel with the applicable $\overline{\text{SYSRST}}$ delay times.

The Fail-Safe Clock Monitor (FSCM) delay determines the time at which the FSCM begins to monitor the system clock source after the $\overline{\text{SYSRST}}$ signal is released.

6.3 Clock Source Selection at Reset

If clock switching is enabled, the system clock source at device Reset is chosen, as shown in Table 6-2. If clock switching is disabled, the system clock source is always selected according to the oscillator Configuration bits. Refer to the "*PIC24F Family Reference Manual*", **Section 8.0 "Oscillator Configuration"** for further details.

TABLE 6-2: OSCILLATOR SELECTION vs. TYPE OF RESET (CLOCK SWITCHING ENABLED)

Reset Type	Clock Source Determinant
POR	FNOSC Configuration bits (CW2<10:8>)
BOR	
$\overline{\text{MCLR}}$	COSC Control bits (OSCCON<14:12>)
WDTO	
SWR	

TABLE 6-3: RESET DELAY TIMES FOR VARIOUS DEVICE RESETS

Reset Type	Clock Source	$\overline{\text{SYSRST}}$ Delay	System Clock Delay	Notes
POR(7)	EC	TPOR + TSTARTUP + TRST	—	1, 2, 3
	ECPLL	TPOR + TSTARTUP + TRST	TLOCK	1, 2, 3, 5
	XT, HS, SOSC	TPOR + TSTARTUP + TRST	TOST	1, 2, 3, 4
	XTPLL, HSPLL	TPOR + TSTARTUP + TRST	TOST + TLOCK	1, 2, 3, 4, 5
	FRC, FRCDIV	TPOR + TSTARTUP + TRST	TFRC	1, 2, 3, 6, 7
	FRCPLL	TPOR + TSTARTUP + TRST	TFRC + TLOCK	1, 2, 3, 5, 6
	LPRC	TPOR + TSTARTUP + TRST	TLPRC	1, 2, 3, 6
BOR	EC	TSTARTUP + TRST	—	2, 3
	ECPLL	TSTARTUP + TRST	TLOCK	2, 3, 5
	XT, HS, SOSC	TSTARTUP + TRST	TOST	2, 3, 4
	XTPLL, HSPLL	TSTARTUP + TRST	TOST + TLOCK	2, 3, 4, 5
	FRC, FRCDIV	TSTARTUP + TRST	TFRC	2, 3, 6, 7
	FRCPLL	TSTARTUP + TRST	TFRC + TLOCK	2, 3, 5, 6
	LPRC	TSTARTUP + TRST	TLPRC	2, 3, 6
$\overline{\text{MCLR}}$	Any Clock	TRST	—	3
WDT	Any Clock	TRST	—	3
Software	Any clock	TRST	—	3
Illegal Opcode	Any Clock	TRST	—	3
Uninitialized W	Any Clock	TRST	—	3
Trap Conflict	Any Clock	TRST	—	3

Note 1: TPOR = Power-on Reset delay (10 μs nominal).

2: TSTARTUP = TVREG (10 μs nominal when VREGS = 1 and when VREGS = 0; depends upon WUTSEL<1:0> bits setting).

3: TRST = Internal State Reset time (32 μs nominal).

4: TOST = Oscillator Start-up Timer. A 10-bit counter counts 1024 oscillator periods before releasing the oscillator clock to the system.

5: TLOCK = PLL lock time.

6: TFRC and TLPRC = RC Oscillator start-up times.

7: If Two-speed Start-up is enabled, regardless of the primary oscillator selected, the device starts with FRC so the system clock delay is just TFRC, and in such cases, FRC start-up time is valid. It switches to the primary oscillator after its respective clock delay.

6.3.1 POR AND LONG OSCILLATOR START-UP TIMES

The oscillator start-up circuitry and its associated delay timers are not linked to the device Reset delays that occur at power-up. Some crystal circuits (especially low-frequency crystals) will have a relatively long start-up time. Therefore, one or more of the following conditions is possible after $\overline{\text{SYSRST}}$ is released:

- The oscillator circuit has not begun to oscillate.
- The Oscillator Start-up Timer has not expired (if a crystal oscillator is used).
- The PLL has not achieved a lock (if PLL is used).

The device will not begin to execute code until a valid clock source has been released to the system. Therefore, the oscillator and PLL start-up delays must be considered when the Reset delay time must be known.

6.3.2 FAIL-SAFE CLOCK MONITOR (FSCM) AND DEVICE RESETS

If the FSCM is enabled, it will begin to monitor the system clock source when $\overline{\text{SYSRST}}$ is released. If a valid clock source is not available at this time, the device will automatically switch to the FRC oscillator and the user can switch to the desired crystal oscillator in the Trap Service Routine (TSR).

NOTES:

7.0 INTERRUPT CONTROLLER

Note: This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the "*PIC24F Family Reference Manual*", **Section 8. "Interrupts"** (DS39707). The information in this data sheet supersedes the information in the FRM.

The PIC24F interrupt controller reduces the numerous peripheral interrupt request signals to a single interrupt request signal to the PIC24F CPU. It has the following features:

- Up to 8 processor exceptions and software traps
- Seven user-selectable priority levels
- Interrupt Vector Table (IVT) with up to 118 vectors
- Unique vector for each interrupt or exception source
- Fixed priority within a specified user priority level
- Alternate Interrupt Vector Table (AIVT) for debug support
- Fixed interrupt entry and return latencies

7.1 Interrupt Vector Table

The Interrupt Vector Table (IVT) is shown in Figure 7-1. The IVT resides in program memory, starting at location 000004h. The IVT contains 126 vectors, consisting of 8 non-maskable trap vectors, plus up to 118 sources of interrupt. In general, each interrupt source has its own vector. Each interrupt vector contains a 24-bit wide address. The value programmed into each interrupt vector location is the starting address of the associated Interrupt Service Routine (ISR).

Interrupt vectors are prioritized in terms of their natural priority; this is linked to their position in the vector table. All other things being equal, lower addresses have a higher natural priority. For example, the interrupt associated with Vector 0 will take priority over interrupts at any other vector address.

PIC24FJ256DA210 family devices implement non-maskable traps and unique interrupts. These are summarized in Table 7-1 and Table 7-2.

7.1.1 ALTERNATE INTERRUPT VECTOR TABLE

The Alternate Interrupt Vector Table (AIVT) is located after the IVT, as shown in Figure 7-1. The ALTIVT (INTCON2<15>) control bit provides access to the AIVT. If the ALTIVT bit is set, all interrupt and exception processes will use the alternate vectors instead of the default vectors. The alternate vectors are organized in the same manner as the default vectors.

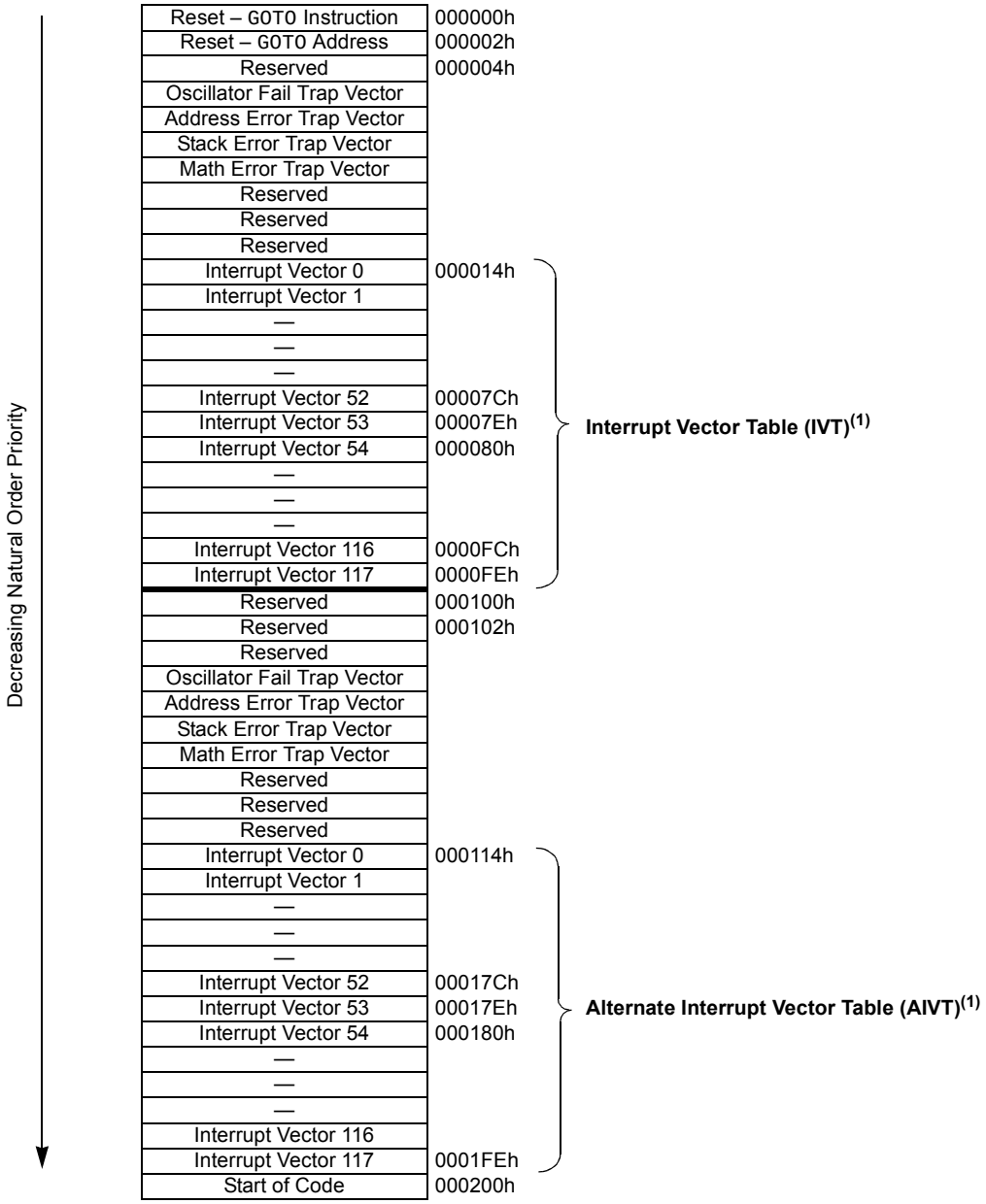
The AIVT supports emulation and debugging efforts by providing a means to switch between an application and a support environment without requiring the interrupt vectors to be reprogrammed. This feature also enables switching between applications for evaluation of different software algorithms at run time. If the AIVT is not needed, the AIVT should be programmed with the same addresses used in the IVT.

7.2 Reset Sequence

A device Reset is not a true exception because the interrupt controller is not involved in the Reset process. The PIC24F devices clear their registers in response to a Reset, which forces the PC to zero. The microcontroller then begins program execution at location, 000000h. The user programs a GOTO instruction at the Reset address, which redirects program execution to the appropriate start-up routine.

Note: Any unimplemented or unused vector locations in the IVT and AIVT should be programmed with the address of a default interrupt handler routine that contains a RESET instruction.

FIGURE 7-1: PIC24F INTERRUPT VECTOR TABLE



Note 1: See Table 7-2 for the interrupt vector list.

TABLE 7-1: TRAP VECTOR DETAILS

Vector Number	IVT Address	AIVT Address	Trap Source
0	000004h	000104h	Reserved
1	000006h	000106h	Oscillator Failure
2	000008h	000108h	Address Error
3	00000Ah	00010Ah	Stack Error
4	00000Ch	00010Ch	Math Error
5	00000Eh	00010Eh	Reserved
6	000010h	000110h	Reserved
7	000012h	000112h	Reserved

TABLE 7-2: IMPLEMENTED INTERRUPT VECTORS

Interrupt Source	Vector Number	IVT Address	AIVT Address	Interrupt Bit Locations		
				Flag	Enable	Priority
ADC1 Conversion Done	13	00002Eh	00012Eh	IFS0<13>	IEC0<13>	IPC3<6:4>
Comparator Event	18	000038h	000138h	IFS1<2>	IEC1<2>	IPC4<10:8>
CRC Generator	67	00009Ah	00019Ah	IFS4<3>	IEC4<3>	IPC16<14:12>
CTMU Event	77	0000AEh	0001AEh	IFS4<13>	IEC4<13>	IPC19<6:4>
External Interrupt 0	0	000014h	000114h	IFS0<0>	IEC0<0>	IPC0<2:0>
External Interrupt 1	20	00003Ch	00013Ch	IFS1<4>	IEC1<4>	IPC5<2:0>
External Interrupt 2	29	00004Eh	00014Eh	IFS1<13>	IEC1<13>	IPC7<6:4>
External Interrupt 3	53	00007Eh	00017Eh	IFS3<5>	IEC3<5>	IPC13<6:4>
External Interrupt 4	54	000080h	000180h	IFS3<6>	IEC3<6>	IPC13<10:8>
Graphics Controller	100	0000DCh	0001DCh	IFS6<4>	IEC6<4>	IPC25<2:0>
I2C1 Master Event	17	000036h	000136h	IFS1<1>	IEC1<1>	IPC4<6:4>
I2C1 Slave Event	16	000034h	000134h	IFS1<0>	IEC1<0>	IPC4<2:0>
I2C2 Master Event	50	000078h	000178h	IFS3<2>	IEC3<2>	IPC12<10:8>
I2C2 Slave Event	49	000076h	000176h	IFS3<1>	IEC3<1>	IPC12<6:4>
I2C3 Master Event	85	0000BEh	0001BEh	IFS5<5>	IEC5<5>	IPC21<6:4>
I2C3 Slave Event	84	0000BCh	0001BCh	IFS5<4>	IEC5<4>	IPC21<2:0>
Input Capture 1	1	000016h	000116h	IFS0<1>	IEC0<1>	IPC0<6:4>
Input Capture 2	5	00001Eh	00011Eh	IFS0<5>	IEC0<5>	IPC1<6:4>
Input Capture 3	37	00005Eh	00015Eh	IFS2<5>	IEC2<5>	IPC9<6:4>
Input Capture 4	38	000060h	000160h	IFS2<6>	IEC2<6>	IPC9<10:8>
Input Capture 5	39	000062h	000162h	IFS2<7>	IEC2<7>	IPC9<14:12>
Input Capture 6	40	000064h	000164h	IFS2<8>	IEC2<8>	IPC10<2:0>
Input Capture 7	22	000040h	000140h	IFS1<6>	IEC1<6>	IPC5<10:8>
Input Capture 8	23	000042h	000142h	IFS1<7>	IEC1<7>	IPC5<14:12>
Input Capture 9	93	0000CEh	0001CEh	IFS5<13>	IEC5<13>	IPC23<6:4>
Input Change Notification (ICN)	19	00003Ah	00013Ah	IFS1<3>	IEC1<3>	IPC4<14:12>
Low-Voltage Detect (LVD)	72	0000A4h	0001A4h	IFS4<8>	IEC4<8>	IPC18<2:0>
Output Compare 1	2	000018h	000118h	IFS0<2>	IEC0<2>	IPC0<10:8>
Output Compare 2	6	000020h	000120h	IFS0<6>	IEC0<6>	IPC1<10:8>
Output Compare 3	25	000046h	000146h	IFS1<9>	IEC1<9>	IPC6<6:4>
Output Compare 4	26	000048h	000148h	IFS1<10>	IEC1<10>	IPC6<10:8>
Output Compare 5	41	000066h	000166h	IFS2<9>	IEC2<9>	IPC10<6:4>
Output Compare 6	42	000068h	000168h	IFS2<10>	IEC2<10>	IPC10<10:8>
Output Compare 7	43	00006Ah	00016Ah	IFS2<11>	IEC2<11>	IPC10<14:12>
Output Compare 8	44	00006Ch	00016Ch	IFS2<12>	IEC2<12>	IPC11<2:0>
Output Compare 9	92	0000CCh	0001CCh	IFS5<12>	IEC5<12>	IPC23<2:0>
Enhanced Parallel Master Port (EPMP)(1)	45	00006Eh	00016Eh	IFS2<13>	IEC2<13>	IPC11<6:4>
Real-Time Clock and Calendar (RTCC)	62	000090h	000190h	IFS3<14>	IEC3<14>	IPC15<10:8>
SPI1 Error	9	000026h	000126h	IFS0<9>	IEC0<9>	IPC2<6:4>
SPI1 Event	10	000028h	000128h	IFS0<10>	IEC0<10>	IPC2<10:8>
SPI2 Error	32	000054h	000154h	IFS2<0>	IEC2<0>	IPC8<2:0>
SPI2 Event	33	000056h	000156h	IFS2<1>	IEC2<1>	IPC8<6:4>
SPI3 Error	90	0000C8h	0001C8h	IFS5<10>	IEC5<10>	IPC22<10:8>
SPI3 Event	91	0000CAh	0001CAh	IFS5<11>	IEC5<11>	IPC22<14:12>

Note 1: Not available in 64-pin devices (PIC24FJXXXDAX06).

TABLE 7-2: IMPLEMENTED INTERRUPT VECTORS (CONTINUED)

Interrupt Source	Vector Number	IVT Address	AIVT Address	Interrupt Bit Locations		
				Flag	Enable	Priority
Timer1	3	00001Ah	00011Ah	IFS0<3>	IEC0<3>	IPC0<14:12>
Timer2	7	000022h	000122h	IFS0<7>	IEC0<7>	IPC1<14:12>
Timer3	8	000024h	000124h	IFS0<8>	IEC0<8>	IPC2<2:0>
Timer4	27	00004Ah	00014Ah	IFS1<11>	IEC1<11>	IPC6<14:12>
Timer5	28	00004Ch	00014Ch	IFS1<12>	IEC1<12>	IPC7<2:0>
UART1 Error	65	000096h	000196h	IFS4<1>	IEC4<1>	IPC16<6:4>
UART1 Receiver	11	00002Ah	00012Ah	IFS0<11>	IEC0<11>	IPC2<14:12>
UART1 Transmitter	12	00002Ch	00012Ch	IFS0<12>	IEC0<12>	IPC3<2:0>
UART2 Error	66	000098h	000198h	IFS4<2>	IEC4<2>	IPC16<10:8>
UART2 Receiver	30	000050h	000150h	IFS1<14>	IEC1<14>	IPC7<10:8>
UART2 Transmitter	31	000052h	000152h	IFS1<15>	IEC1<15>	IPC7<14:12>
UART3 Error	81	0000B6h	0001B6h	IFS5<1>	IEC5<1>	IPC20<6:4>
UART3 Receiver	82	0000B8h	0001B8h	IFS5<2>	IEC5<2>	IPC20<10:8>
UART3 Transmitter	83	0000BAh	0001BAh	IFS5<3>	IEC5<3>	IPC20<14:12>
UART4 Error	87	0000C2h	0001C2h	IFS5<7>	IEC5<7>	IPC21<14:12>
UART4 Receiver	88	0000C4h	0001C4h	IFS5<8>	IEC5<8>	IPC22<2:0>
UART4 Transmitter	89	0000C6h	0001C6h	IFS5<9>	IEC5<9>	IPC22<6:4>
USB Interrupt	86	0000C0h	0001C0h	IFS5<6>	IEC5<6>	IPC21<10:8>

Note 1: Not available in 64-pin devices (PIC24FJXXXDAX06).

7.3 Interrupt Control and Status Registers

The PIC24FJ256DA210 family of devices implements a total of 40 registers for the interrupt controller:

- INTCON1
- INTCON2
- IFS0 through IFS6
- IEC0 through IEC6
- IPC0 through IPC25 (except IPC14, IPC17 and IPC24)
- INTTREG

Global interrupt control functions are controlled from INTCON1 and INTCON2. INTCON1 contains the Interrupt Nesting Disable (NSTDIS) bit, as well as the control and status flags for the processor trap sources. The INTCON2 register controls the external interrupt request signal behavior and the use of the Alternate Interrupt Vector Table (AIVT).

The IFSx registers maintain all of the interrupt request flags. Each source of interrupt has a status bit, which is set by the respective peripherals or an external signal and is cleared via software.

The IECx registers maintain all of the interrupt enable bits. These control bits are used to individually enable interrupts from the peripherals or external signals.

The IPCx registers are used to set the interrupt priority level for each source of interrupt. Each user interrupt source can be assigned to one of eight priority levels.

The INTTREG register contains the associated interrupt vector number and the new CPU interrupt priority level, which are latched into the Vector Number (VECNUM<6:0>) and the Interrupt Level (ILR<3:0>) bit fields in the INTTREG register. The new interrupt priority level is the priority of the pending interrupt.

The interrupt sources are assigned to the IFSx, IECx and IPCx registers in the order of their vector numbers, as shown in Table 7-2. For example, the INT0 (External Interrupt 0) is shown as having a vector number and a natural order priority of 0. Thus, the INT0IF status bit is found in IFS0<0>, the INT0IE enable bit in IEC0<0> and the INT0IP<2:0> priority bits in the first position of IPC0 (IPC0<2:0>).

Although they are not specifically part of the interrupt control hardware, two of the CPU Control registers contain bits that control interrupt functionality. The ALU STATUS register (SR) contains the IPL<2:0> bits (SR<7:5>). These indicate the current CPU interrupt priority level. The user can change the current CPU priority level by writing to the IPL bits.

The CORCON register contains the IPL3 bit, which, together with IPL<2:0>, indicates the current CPU priority level. IPL3 is a read-only bit so that trap events cannot be masked by the user software.

The interrupt controller has the Interrupt Controller Test register, INTTREG, which displays the status of the interrupt controller. When an interrupt request occurs, it's associated vector number and the new interrupt priority level are latched into INTTREG. This information can be used to determine a specific interrupt source if a generic ISR is used for multiple vectors (such as when ISR remapping is used in bootloader applications) or to check if another interrupt is pending while in an ISR.

All interrupt registers are described in Register 7-1 through Register 7-40 in the succeeding pages.

REGISTER 7-1: SR: ALU STATUS REGISTER (IN CPU)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R-0, HSC
—	—	—	—	—	—	—	DC(1)
bit 15							bit 8

R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC
IPL2(2,3)	IPL1(2,3)	IPL0(2,3)	RA(1)	N(1)	OV(1)	Z(1)	C(1)
bit 7							bit 0

Legend:	HSC = Hardware Settable/Clearable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-9 **Unimplemented:** Read as '0'

bit 7-5 **IPL<2:0>:** CPU Interrupt Priority Level Status bits(2,3)

111 = CPU interrupt priority level is 7 (15); user interrupts are disabled
110 = CPU interrupt priority level is 6 (14)
101 = CPU interrupt priority level is 5 (13)
100 = CPU interrupt priority level is 4 (12)
011 = CPU interrupt priority level is 3 (11)
010 = CPU interrupt priority level is 2 (10)
001 = CPU interrupt priority level is 1 (9)
000 = CPU interrupt priority level is 0 (8)

Note 1: See Register 3-1 for the description of the remaining bits (bit 8, 4, 3, 2, 1 and 0) that are not dedicated to interrupt control functions.

2: The IPL bits are concatenated with the IPL3 (CORCON<3>) bit to form the CPU interrupt priority level. The value in parentheses indicates the interrupt priority level if IPL3 = 1.

3: The IPL Status bits are read-only when NSTDIS (INTCON1<15>) = 1.

REGISTER 7-2: CORCON: CPU CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	R/C-0, HSC	r-1	U-0	U-0
—	—	—	—	IPL3(1)	r	—	—
bit 7							bit 0

Legend:	r = Reserved bit	C = Clearable bit	HSC = Hardware Settable/Clearable bit
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-4 **Unimplemented:** Read as '0'

bit 3 **IPL3:** CPU Interrupt Priority Level Status bit(1)
1 = CPU interrupt priority level is greater than 7
0 = CPU interrupt priority level is 7 or less

bit 2 **Reserved:** Read as '1'

bit 1-0 **Unimplemented:** Read as '0'

Note 1: The IPL3 bit is concatenated with the IPL<2:0> bits (SR<7:5>) to form the CPU interrupt priority level; see Register 3-2 for bit description.

REGISTER 7-3: INTCON1: INTERRUPT CONTROL REGISTER 1

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
NSTDIS	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	U-0
—	—	—	MATHERR	ADDRERR	STKERR	OSCFAIL	—
bit 7							bit 0

Legend:	HS = Hardware Settable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **NSTDIS:** Interrupt Nesting Disable bit
1 = Interrupt nesting is disabled
0 = Interrupt nesting is enabled

bit 14-5 **Unimplemented:** Read as '0'

bit 4 **MATHERR:** Arithmetic Error Trap Status bit
1 = Overflow trap has occurred
0 = Overflow trap has not occurred

bit 3 **ADDRERR:** Address Error Trap Status bit
1 = Address error trap has occurred
0 = Address error trap has not occurred

bit 2 **STKERR:** Stack Error Trap Status bit
1 = Stack error trap has occurred
0 = Stack error trap has not occurred

bit 1 **OSCFAIL:** Oscillator Failure Trap Status bit
1 = Oscillator failure trap has occurred
0 = Oscillator failure trap has not occurred

bit 0 **Unimplemented:** Read as '0'

REGISTER 7-4: INTCON2: INTERRUPT CONTROL REGISTER 2

R/W-0	R-0, HSC	U-0	U-0	U-0	U-0	U-0	U-0
ALTIVT	DISI	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	INT4EP	INT3EP	INT2EP	INT1EP	INT0EP
bit 7							bit 0

Legend:	HSC = Hardware Settable/Clearable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **ALTIVT:** Enable Alternate Interrupt Vector Table bit
1 = Use Alternate Interrupt Vector Table
0 = Use standard (default) vector table

bit 14 **DISI:** DISI Instruction Status bit
1 = DISI instruction is active
0 = DISI instruction is not active

bit 13-5 **Unimplemented:** Read as '0'

bit 4 **INT4EP:** External Interrupt 4 Edge Detect Polarity Select bit
1 = Interrupt on negative edge
0 = Interrupt on positive edge

bit 3 **INT3EP:** External Interrupt 3 Edge Detect Polarity Select bit
1 = Interrupt on negative edge
0 = Interrupt on positive edge

bit 2 **INT2EP:** External Interrupt 2 Edge Detect Polarity Select bit
1 = Interrupt on negative edge
0 = Interrupt on positive edge

bit 1 **INT1EP:** External Interrupt 1 Edge Detect Polarity Select bit
1 = Interrupt on negative edge
0 = Interrupt on positive edge

bit 0 **INT0EP:** External Interrupt 0 Edge Detect Polarity Select bit
1 = Interrupt on negative edge
0 = Interrupt on positive edge

REGISTER 7-5: IFS0: INTERRUPT FLAG STATUS REGISTER 0

U-0	U-0	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS
—	—	AD1IF	U1TXIF	U1RXIF	SPI1IF	SPF1IF	T3IF
bit 15							bit 8

R/W-0, HS	R/W-0, HS	R/W-0, HS	U-0	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS
T2IF	OC2IF	IC2IF	—	T1IF	OC1IF	IC1IF	INT0IF
bit 7							bit 0

Legend:	HS = Hardware Settable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13 **AD1IF:** A/D Conversion Complete Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 12 **U1TXIF:** UART1 Transmitter Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 11 **U1RXIF:** UART1 Receiver Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 10 **SPI1IF:** SPI1 Event Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 9 **SPF1IF:** SPI1 Fault Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 8 **T3IF:** Timer3 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 7 **T2IF:** Timer2 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 6 **OC2IF:** Output Compare Channel 2 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 5 **IC2IF:** Input Capture Channel 2 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 4 **Unimplemented:** Read as '0'

bit 3 **T1IF:** Timer1 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 2 **OC1IF:** Output Compare Channel 1 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

REGISTER 7-5: IFS0: INTERRUPT FLAG STATUS REGISTER 0 (CONTINUED)

bit 1 **IC1IF:** Input Capture Channel 1 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 0 **INT0IF:** External Interrupt 0 Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

REGISTER 7-6: IFS1: INTERRUPT FLAG STATUS REGISTER 1

R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	U-0
U2TXIF	U2RXIF	INT2IF	T5IF	T4IF	OC4IF	OC3IF	—
bit 15							bit 8

R/W-0, HS	R/W-0, HS	U-0	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS
IC8IF	IC7IF	—	INT1IF	CNIF	CMIF	MI2C1IF	SI2C1IF
bit 7							bit 0

Legend:	HS = Hardware Settable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **U2TXIF:** UART2 Transmitter Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 14 **U2RXIF:** UART2 Receiver Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 13 **INT2IF:** External Interrupt 2 Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 12 **T5IF:** Timer5 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 11 **T4IF:** Timer4 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 10 **OC4IF:** Output Compare Channel 4 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 9 **OC3IF:** Output Compare Channel 3 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 8 **Unimplemented:** Read as '0'

bit 7 **IC8IF:** Input Capture Channel 8 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 6 **IC7IF:** Input Capture Channel 7 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

REGISTER 7-6: IFS1: INTERRUPT FLAG STATUS REGISTER 1 (CONTINUED)

bit 5 **Unimplemented:** Read as '0'

bit 4 **INT1IF:** External Interrupt 1 Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 3 **CNIF:** Input Change Notification Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 2 **CMIF:** Comparator Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 1 **MI2C1IF:** Master I2C1 Event Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 0 **SI2C1IF:** Slave I2C1 Event Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

REGISTER 7-7: IFS2: INTERRUPT FLAG STATUS REGISTER 2

U-0	U-0	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS
—	—	PMPIF(1)	OC8IF	OC7IF	OC6IF	OC5IF	IC6IF
bit 15							bit 8

R/W-0, HS	R/W-0, HS	R/W-0, HS	U-0	U-0	U-0	R/W-0, HS	R/W-0, HS
IC5IF	IC4IF	IC3IF	—	—	—	SPI2IF	SPF2IF
bit 7							bit 0

Legend:	HS = Hardware Settable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13 **PMPIF:** Parallel Master Port Interrupt Flag Status bit(1)
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 12 **OC8IF:** Output Compare Channel 8 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 11 **OC7IF:** Output Compare Channel 7 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 10 **OC6IF:** Output Compare Channel 6 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 9 **OC5IF:** Output Compare Channel 5 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

Note 1: Not available in PIC24FJXXXDAX06 devices.

REGISTER 7-7: IFS2: INTERRUPT FLAG STATUS REGISTER 2 (CONTINUED)

bit 8 **IC6IF:** Input Capture Channel 6 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 7 **IC5IF:** Input Capture Channel 5 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 6 **IC4IF:** Input Capture Channel 4 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 5 **IC3IF:** Input Capture Channel 3 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 4-2 **Unimplemented:** Read as '0'

bit 1 **SPI2IF:** SPI2 Event Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 0 **SPF2IF:** SPI2 Fault Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

Note 1: Not available in PIC24FJXXXDAX06 devices.

REGISTER 7-8: IFS3: INTERRUPT FLAG STATUS REGISTER 3

U-0	R/W-0, HS	U-0	U-0	U-0	U-0	U-0	U-0
—	RTCIF	—	—	—	—	—	—
bit 15							bit 8

U-0	R/W-0, HS	R/W-0, HS	U-0	U-0	R/W-0, HS	R/W-0, HS	U-0
—	INT4IF	INT3IF	—	—	MI2C2IF	SI2C2IF	—
bit 7							bit 0

Legend:	HS = Hardware Settable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as '0'

bit 14 **RTCIF:** Real-Time Clock/Calendar Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 13-7 **Unimplemented:** Read as '0'

bit 6 **INT4IF:** External Interrupt 4 Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 5 **INT3IF:** External Interrupt 3 Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 4-3 **Unimplemented:** Read as '0'

bit 2 **MI2C2IF:** Master I2C2 Event Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 1 **SI2C2IF:** Slave I2C2 Event Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 0 **Unimplemented:** Read as '0'

REGISTER 7-9: IFS4: INTERRUPT FLAG STATUS REGISTER 4

U-0	U-0	R/W-0, HS	U-0	U-0	U-0	U-0	R/W-0, HS
—	—	CTMUIF	—	—	—	—	LVDIF
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0, HS	R/W-0, HS	R/W-0, HS	U-0
—	—	—	—	CRCIF	U2ERIF	U1ERIF	—
bit 7							bit 0

Legend:	HS = Hardware Settable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13 **CTMUIF:** CTMU Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 12-9 **Unimplemented:** Read as '0'

bit 8 **LVDIF:** Low-Voltage Detect Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 7-4 **Unimplemented:** Read as '0'

bit 3 **CRCIF:** CRC Generator Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 2 **U2ERIF:** UART2 Error Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 1 **U1ERIF:** UART1 Error Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 0 **Unimplemented:** Read as '0'

REGISTER 7-10: IFS5: INTERRUPT FLAG STATUS REGISTER 5

U-0	U-0	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS
—	—	IC9IF	OC9IF	SPI3IF	SPF3IF	U4TXIF	U4RXIF
bit 15							bit 8

R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	U-0
U4ERIF	USB1IF	MI2C3IF	SI2C3IF	U3TXIF	U3RXIF	U3ERIF	—
bit 7							bit 0

Legend:	HS = Hardware Settable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13 **IC9IF:** Input Capture Channel 9 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 12 **OC9IF:** Output Compare Channel 9 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 11 **SPI3IF:** SPI3 Event Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 10 **SPF3IF:** SPI3 Fault Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 9 **U4TXIF:** UART4 Transmitter Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 8 **U4RXIF:** UART4 Receiver Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 7 **U4ERIF:** UART4 Error Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 6 **USB1IF:** USB1 (USB OTG) Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 5 **MI2C3IF:** Master I2C3 Event Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 4 **SI2C3IF:** Slave I2C3 Event Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 3 **U3TXIF:** UART3 Transmitter Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 2 **U3RXIF:** UART3 Receiver Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

REGISTER 7-10: IFS5: INTERRUPT FLAG STATUS REGISTER 5 (CONTINUED)

bit 1 **U3ERIF:** UART3 Error Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 0 **Unimplemented:** Read as '0'

REGISTER 7-11: IFS6: INTERRUPT FLAG STATUS REGISTER 6

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	R/W-0, HS	U-0	U-0	U-0	U-0
—	—	—	GFX1IF	—	—	—	—
bit 7							bit 0

Legend:	HS = Hardware Settable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-5 **Unimplemented:** Read as '0'

bit 4 **GFX1IF:** Graphics 1 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 3-0 **Unimplemented:** Read as '0'

REGISTER 7-12: IEC0: INTERRUPT ENABLE CONTROL REGISTER 0

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	AD1IE	U1TXIE	U1RXIE	SPI1IE	SPF1IE	T3IE
bit 15							bit 8

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
T2IE	OC2IE	IC2IE	—	T1IE	OC1IE	IC1IE	INT0IE
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13 **AD1IE:** A/D Conversion Complete Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 12 **U1TXIE:** UART1 Transmitter Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 11 **U1RXIE:** UART1 Receiver Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 10 **SPI1IE:** SPI1 Transfer Complete Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 9 **SPF1IE:** SPI1 Fault Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 8 **T3IE:** Timer3 Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 7 **T2IE:** Timer2 Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 6 **OC2IE:** Output Compare Channel 2 Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 5 **IC2IE:** Input Capture Channel 2 Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 4 **Unimplemented:** Read as '0'

bit 3 **T1IE:** Timer1 Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 2 **OC1IE:** Output Compare Channel 1 Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

REGISTER 7-12: IEC0: INTERRUPT ENABLE CONTROL REGISTER 0 (CONTINUED)

bit 1 **IC1IE:** Input Capture Channel 1 Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 0 **INT0IE:** External Interrupt 0 Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

REGISTER 7-13: IEC1: INTERRUPT ENABLE CONTROL REGISTER 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
U2TXIE	U2RXIE	INT2IE(1)	T5IE	T4IE	OC4IE	OC3IE	—
bit 15							bit 8

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
IC8IE	IC7IE	—	INT1IE(1)	CNIE	CMIE	MI2C1IE	SI2C1IE
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **U2TXIE:** UART2 Transmitter Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 14 **U2RXIE:** UART2 Receiver Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 13 **INT2IE:** External Interrupt 2 Enable bit(1)
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 12 **T5IE:** Timer5 Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 11 **T4IE:** Timer4 Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 10 **OC4IE:** Output Compare Channel 4 Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 9 **OC3IE:** Output Compare Channel 3 Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 8 **Unimplemented:** Read as '0'

bit 7 **IC8IE:** Input Capture Channel 8 Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

Note 1: If an external interrupt is enabled, the interrupt input must also be configured to an available RPx or RPIx pin. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

REGISTER 7-13: IEC1: INTERRUPT ENABLE CONTROL REGISTER 1 (CONTINUED)

bit 6 **IC7IE:** Input Capture Channel 7 Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 5 **Unimplemented:** Read as '0'

bit 4 **INT1IE:** External Interrupt 1 Enable bit(1)
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 3 **CNIE:** Input Change Notification Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 2 **CMIE:** Comparator Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 1 **MI2C1IE:** Master I2C1 Event Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 0 **SI2C1IE:** Slave I2C1 Event Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

Note 1: If an external interrupt is enabled, the interrupt input must also be configured to an available RPx or RPIx pin. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

REGISTER 7-14: IEC2: INTERRUPT ENABLE CONTROL REGISTER 2

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	PMPIE(1)	OC8IE	OC7IE	OC6IE	OC5IE	IC6IE
bit 15							bit 8

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
IC5IE	IC4IE	IC3IE	—	—	—	SPI2IE	SPF2IE
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13 **PMPIE:** Parallel Master Port Interrupt Enable bit(1)
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 12 **OC8IE:** Output Compare Channel 8 Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 11 **OC7IE:** Output Compare Channel 7 Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 10 **OC6IE:** Output Compare Channel 6 Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 9 **OC5IE:** Output Compare Channel 5 Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 8 **IC6IE:** Input Capture Channel 6 Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 7 **IC5IE:** Input Capture Channel 5 Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 6 **IC4IE:** Input Capture Channel 4 Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 5 **IC3IE:** Input Capture Channel 3 Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 4-2 **Unimplemented:** Read as '0'

bit 1 **SPI2IE:** SPI2 Event Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 0 **SPF2IE:** SPI2 Fault Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

Note 1: Not available in 64-pin devices (PIC24FJXXXDAX06).

REGISTER 7-15: IEC3: INTERRUPT ENABLE CONTROL REGISTER 3

U-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
—	RTCIE	—	—	—	—	—	—
bit 15							bit 8

U-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	U-0
—	INT4IE(1)	INT3IE(1)	—	—	MI2C2IE	SI2C2IE	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as '0'

bit 14 **RTCIE:** Real-Time Clock/Calendar Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 13-7 **Unimplemented:** Read as '0'

bit 6 **INT4IE:** External Interrupt 4 Enable bit(1)
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 5 **INT3IE:** External Interrupt 3 Enable bit(1)
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 4-3 **Unimplemented:** Read as '0'

bit 2 **MI2C2IE:** Master I2C2 Event Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 1 **SI2C2IE:** Slave I2C2 Event Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 0 **Unimplemented:** Read as '0'

Note 1: If an external interrupt is enabled, the interrupt input must also be configured to an available RPx or RPIx pin. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

REGISTER 7-16: IEC4: INTERRUPT ENABLE CONTROL REGISTER 4

U-0	U-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0
—	—	CTMUIE	—	—	—	—	LVDIE
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	U-0
—	—	—	—	CRCIE	U2ERIE	U1ERIE	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13 **CTMUIE:** CTMU Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 12-9 **Unimplemented:** Read as '0'

bit 8 **LVDIE:** Low-Voltage Detect Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 7-4 **Unimplemented:** Read as '0'

bit 3 **CRCIE:** CRC Generator Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 2 **U2ERIE:** UART2 Error Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 1 **U1ERIE:** UART1 Error Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 0 **Unimplemented:** Read as '0'

REGISTER 7-17: IEC5: INTERRUPT ENABLE CONTROL REGISTER 5

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	IC9IE	OC9IE	SPI3IE	SPF3IE	U4TXIE	U4RXIE
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
U4ERIE	USB1IE	MI2C3IE	SI2C3IE	U3TXIE	U3RXIE	U3ERIE	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13 **IC9IE:** Input Capture Channel 9 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 12 **OC9IE:** Output Compare Channel 9 Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 11 **SPI3IE:** SPI3 Event Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 10 **SPF3IE:** SPI3 Fault Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 9 **U4TXIE:** UART4 Transmitter Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 8 **U4RXIE:** UART4 Receiver Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 7 **U4ERIE:** UART4 Error Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 6 **USB1IE:** USB1 (USB OTG) Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 5 **MI2C3IE:** Master I2C3 Event Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 4 **SI2C3IE:** Slave I2C3 Event Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 3 **U3TXIE:** UART3 Transmitter Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

bit 2 **U3RXIE:** UART3 Receiver Interrupt Enable bit
1 = Interrupt request enabled
0 = Interrupt request not enabled

REGISTER 7-17: IEC5: INTERRUPT ENABLE CONTROL REGISTER 5 (CONTINUED)

bit 1 **U3ERIE:** UART3 Error Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 0 **Unimplemented:** Read as '0'

REGISTER 7-18: IEC6: INTERRUPT ENABLE CONTROL REGISTER 6

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	R/W-0	U-0	U-0	U-0	U-0
—	—	—	GFX1IE	—	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-5 **Unimplemented:** Read as '0'

bit 4 **GFX1IE:** Graphics 1 Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 3-0 **Unimplemented:** Read as '0'

REGISTER 7-19: IPC0: INTERRUPT PRIORITY CONTROL REGISTER 0

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	T1IP2	T1IP1	T1IP0	—	OC1IP2	OC1IP1	OC1IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	IC1IP2	IC1IP1	IC1IP0	—	INT0IP2	INT0IP1	INT0IP0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as '0'

bit 14-12 **T1IP<2:0>:** Timer1 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as '0'

bit 10-8 **OC1IP<2:0>:** Output Compare Channel 1 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 7 **Unimplemented:** Read as '0'

bit 6-4 **IC1IP<2:0>:** Input Capture Channel 1 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3 **Unimplemented:** Read as '0'

bit 2-0 **INT0IP<2:0>:** External Interrupt 0 Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

REGISTER 7-20: IPC1: INTERRUPT PRIORITY CONTROL REGISTER 1

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	T2IP2	T2IP1	T2IP0	—	OC2IP2	OC2IP1	OC2IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	IC2IP2	IC2IP1	IC2IP0	—	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as '0'

bit 14-12 **T2IP<2:0>:** Timer2 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as '0'

bit 10-8 **OC2IP<2:0>:** Output Compare Channel 2 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 7 **Unimplemented:** Read as '0'

bit 6-4 **IC2IP<2:0>:** Input Capture Channel 2 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3-0 **Unimplemented:** Read as '0'

REGISTER 7-21: IPC2: INTERRUPT PRIORITY CONTROL REGISTER 2

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	U1RXIP2	U1RXIP1	U1RXIP0	—	SPI1IP2	SPI1IP1	SPI1IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	SPF1IP2	SPF1IP1	SPF1IP0	—	T3IP2	T3IP1	T3IP0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as ‘0’

bit 14-12 **U1RXIP<2:0>:** UART1 Receiver Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as ‘0’

bit 10-8 **SPI1IP<2:0>:** SPI1 Event Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 7 **Unimplemented:** Read as ‘0’

bit 6-4 **SPF1IP<2:0>:** SPI1 Fault Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3 **Unimplemented:** Read as ‘0’

bit 2-0 **T3IP<2:0>:** Timer3 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

REGISTER 7-22: IPC3: INTERRUPT PRIORITY CONTROL REGISTER 3

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	AD1IP2	AD1IP1	AD1IP0	—	U1TXIP2	U1TXIP1	U1TXIP0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-7 **Unimplemented:** Read as ‘0’

bit 6-4 **AD1IP<2:0>:** A/D Conversion Complete Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3 **Unimplemented:** Read as ‘0’

bit 2-0 **U1TXIP<2:0>:** UART1 Transmitter Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

REGISTER 7-23: IPC4: INTERRUPT PRIORITY CONTROL REGISTER 4

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	CNIP2	CNIP1	CNIP0	—	CMIP2	CMIP1	CMIP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	MI2C1IP2	MI2C1IP1	MI2C1IP0	—	SI2C1IP2	SI2C1IP1	SI2C1IP0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as ‘0’

bit 14-12 **CNIP<2:0>:** Input Change Notification Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as ‘0’

bit 10-8 **CMIP<2:0>:** Comparator Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 7 **Unimplemented:** Read as ‘0’

bit 6-4 **MI2C1IP<2:0>:** Master I2C1 Event Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3 **Unimplemented:** Read as ‘0’

bit 2-0 **SI2C1IP<2:0>:** Slave I2C1 Event Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

REGISTER 7-24: IPC5: INTERRUPT PRIORITY CONTROL REGISTER 5

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	IC8IP2	IC8IP1	IC8IP0	—	IC7IP2	IC7IP1	IC7IP0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	INT1IP2	INT1IP1	INT1IP0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as '0'

bit 14-12 **IC8IP<2:0>:** Input Capture Channel 8 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as '0'

bit 10-8 **IC7IP<2:0>:** Input Capture Channel 7 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 7-3 **Unimplemented:** Read as '0'

bit 2-0 **INT1IP<2:0>:** External Interrupt 1 Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

REGISTER 7-25: IPC6: INTERRUPT PRIORITY CONTROL REGISTER 6

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	T4IP2	T4IP1	T4IP0	—	OC4IP2	OC4IP1	OC4IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	OC3IP2	OC3IP1	OC3IP0	—	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as '0'

bit 14-12 **T4IP<2:0>:** Timer4 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as '0'

bit 10-8 **OC4IP<2:0>:** Output Compare Channel 4 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 7 **Unimplemented:** Read as '0'

bit 6-4 **OC3IP<2:0>:** Output Compare Channel 3 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3-0 **Unimplemented:** Read as '0'

REGISTER 7-26: IPC7: INTERRUPT PRIORITY CONTROL REGISTER 7

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	U2TXIP2	U2TXIP1	U2TXIP0	—	U2RXIP2	U2RXIP1	U2RXIP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	INT2IP2	INT2IP1	INT2IP0	—	T5IP2	T5IP1	T5IP0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as '0'

bit 14-12 **U2TXIP<2:0>:** UART2 Transmitter Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as '0'

bit 10-8 **U2RXIP<2:0>:** UART2 Receiver Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 7 **Unimplemented:** Read as '0'

bit 6-4 **INT2IP<2:0>:** External Interrupt 2 Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3 **Unimplemented:** Read as '0'

bit 2-0 **T5IP<2:0>:** Timer5 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

REGISTER 7-27: IPC8: INTERRUPT PRIORITY CONTROL REGISTER 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	SPI2IP2	SPI2IP1	SPI2IP0	—	SPF2IP2	SPF2IP1	SPF2IP0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-7 **Unimplemented:** Read as ‘0’

bit 6-4 **SPI2IP<2:0>:** SPI2 Event Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3 **Unimplemented:** Read as ‘0’

bit 2-0 **SPF2IP<2:0>:** SPI2 Fault Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

REGISTER 7-28: IPC9: INTERRUPT PRIORITY CONTROL REGISTER 9

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	IC5IP2	IC5IP1	IC5IP0	—	IC4IP2	IC4IP1	IC4IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	IC3IP2	IC3IP1	IC3IP0	—	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as '0'

bit 14-12 **IC5IP<2:0>:** Input Capture Channel 5 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as '0'

bit 10-8 **IC4IP<2:0>:** Input Capture Channel 4 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 7 **Unimplemented:** Read as '0'

bit 6-4 **IC3IP<2:0>:** Input Capture Channel 3 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3-0 **Unimplemented:** Read as '0'

REGISTER 7-29: IPC10: INTERRUPT PRIORITY CONTROL REGISTER 10

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	OC7IP2	OC7IP1	OC7IP0	—	OC6IP2	OC6IP1	OC6IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	OC5IP2	OC5IP1	OC5IP0	—	IC6IP2	IC6IP1	IC6IP0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as '0'

bit 14-12 **OC7IP<2:0>:** Output Compare Channel 7 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as '0'

bit 10-8 **OC6IP<2:0>:** Output Compare Channel 6 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 7 **Unimplemented:** Read as '0'

bit 6-4 **OC5IP<2:0>:** Output Compare Channel 5 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3 **Unimplemented:** Read as '0'

bit 2-0 **IC6IP<2:0>:** Input Capture Channel 6 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

REGISTER 7-30: IPC11: INTERRUPT PRIORITY CONTROL REGISTER 11

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	PMPIP2(1)	PMPIP1(1)	PMPIP0(1)	—	OC8IP2	OC8IP1	OC8IP0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-7 **Unimplemented:** Read as '0'

bit 6-4 **PMPIP<2:0>:** Parallel Master Port Interrupt Priority bits(1)

111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3 **Unimplemented:** Read as '0'

bit 2-0 **OC8IP<2:0>:** Output Compare Channel 8 Interrupt Priority bits

111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

Note 1: Not available in 64-pin devices (PIC24FJXXXDAX06).

REGISTER 7-31: IPC12: INTERRUPT PRIORITY CONTROL REGISTER 12

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	MI2C2IP2	MI2C2IP1	MI2C2IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	SI2C2IP2	SI2C2IP1	SI2C2IP0	—	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-11 **Unimplemented:** Read as '0'

bit 10-8 **MI2C2IP<2:0>:** Master I2C2 Event Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 7 **Unimplemented:** Read as '0'

bit 6-4 **SI2C2IP<2:0>:** Slave I2C2 Event Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3-0 **Unimplemented:** Read as '0'

REGISTER 7-32: IPC13: INTERRUPT PRIORITY CONTROL REGISTER 13

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	INT4IP2	INT4IP1	INT4IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	INT3IP2	INT3IP1	INT3IP0	—	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-11 **Unimplemented:** Read as '0'

bit 10-8 **INT4IP<2:0>:** External Interrupt 4 Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 7 **Unimplemented:** Read as '0'

bit 6-4 **INT3IP<2:0>:** External Interrupt 3 Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3-0 **Unimplemented:** Read as '0'

REGISTER 7-33: IPC15: INTERRUPT PRIORITY CONTROL REGISTER 15

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	RTCIP2	RTCIP1	RTCIP0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-11 **Unimplemented:** Read as ‘0’

bit 10-8 **RTCIP<2:0>:** Real-Time Clock and Calendar Interrupt Priority bits

111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 7-0 **Unimplemented:** Read as ‘0’

REGISTER 7-34: IPC16: INTERRUPT PRIORITY CONTROL REGISTER 16

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	CRCIP2	CRCIP1	CRCIP0	—	U2ERIP2	U2ERIP1	U2ERIP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	U1ERIP2	U1ERIP1	U1ERIP0	—	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as ‘0’

bit 14-12 **CRCIP<2:0>:** CRC Generator Error Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as ‘0’

bit 10-8 **U2ERIP<2:0>:** UART2 Error Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 7 **Unimplemented:** Read as ‘0’

bit 6-4 **U1ERIP<2:0>:** UART1 Error Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3-0 **Unimplemented:** Read as ‘0’

REGISTER 7-35: IPC18: INTERRUPT PRIORITY CONTROL REGISTER 18

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	LVDIP2	LVDIP1	LVDIP0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-3 **Unimplemented:** Read as ‘0’

bit 2-0 **LVDIP<2:0>:** Low-Voltage Detect Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

REGISTER 7-36: IPC19: INTERRUPT PRIORITY CONTROL REGISTER 19

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	CTMUIP2	CTMUIP1	CTMUIP0	—	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-7 **Unimplemented:** Read as ‘0’

bit 6-4 **CTMUIP<2:0>:** CTMU Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3-0 **Unimplemented:** Read as ‘0’

REGISTER 7-37: IPC20: INTERRUPT PRIORITY CONTROL REGISTER 20

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	U3TXIP2	U3TXIP1	U3TXIP0	—	U3RXIP2	U3RXIP1	U3RXIP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	U3ERIP2	U3ERIP1	U3ERIP0	—	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as ‘0’

bit 14-12 **U3TXIP<2:0>:** UART3 Transmitter Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as ‘0’

bit 10-8 **U3RXIP<2:0>:** UART3 Receiver Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 7 **Unimplemented:** Read as ‘0’

bit 6-4 **U3ERIP<2:0>:** UART3 Error Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3-0 **Unimplemented:** Read as ‘0’

REGISTER 7-38: IPC21: INTERRUPT PRIORITY CONTROL REGISTER 21

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	U4ERIP2	U4ERIP1	U4ERIP0	—	USB1IP2	USB1IP1	USB1IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	MI2C3IP2	MI2C3IP1	MI2C3IP0	—	SI2C3IP2	SI2C3IP1	SI2C3IP0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as ‘0’

bit 14-12 **U4ERIP<2:0>:** UART4 Error Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as ‘0’

bit 10-8 **USB1IP<2:0>:** USB1 (USB OTG) Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 7 **Unimplemented:** Read as ‘0’

bit 6-4 **MI2C3IP<2:0>:** Master I2C3 Event Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3 **Unimplemented:** Read as ‘0’

bit 2-0 **SI2C3IP<2:0>:** SIave I2C3 Event Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

REGISTER 7-39: IPC22: INTERRUPT PRIORITY CONTROL REGISTER 22

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	SPI3IP2	SPI3IP1	SPI3IP0	—	SPF3IP2	SPF3IP1	SPF3IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	U4TXIP2	U4TXIP1	U4TXIP0	—	U4RXIP2	U4RXIP1	U4RXIP0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as ‘0’

bit 14-12 **SPI3IP<2:0>:** SPI3 Event Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 11 **Unimplemented:** Read as ‘0’

bit 10-8 **SPF3IP<2:0>:** SPI3 Fault Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 7 **Unimplemented:** Read as ‘0’

bit 6-4 **U4TXIP<2:0>:** UART4 Transmitter Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3 **Unimplemented:** Read as ‘0’

bit 2-0 **U4RXIP<2:0>:** UART4 Receiver Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

REGISTER 7-40: IPC23: INTERRUPT PRIORITY CONTROL REGISTER 23

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	IC9IP2	IC9IP1	IC9IP0	—	OC9IP2	OC9IP1	OC9IP0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-7 **Unimplemented:** Read as ‘0’

bit 6-4 **IC9IP<2:0>:** Input Capture Channel 9 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

bit 3 **Unimplemented:** Read as ‘0’

bit 2-0 **OC9IP<2:0>:** Output Compare Channel 9 Interrupt Priority bits
111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

REGISTER 7-41: IPC25: INTERRUPT PRIORITY CONTROL REGISTER 25

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	GFX1IP2	GFX1IP1	GFX1IP0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-3 **Unimplemented:** Read as ‘0’

bit 2-0 **GFX1IP<2:0>:** Graphics 1 Interrupt Priority bits

111 = Interrupt is priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is priority 1
000 = Interrupt source is disabled

REGISTER 7-42: INTTREG: INTERRUPT CONTROLLER TEST REGISTER

R-0, HSC	U-0	R/W-0	U-0	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC
CPUIRQ	—	VHOLD	—	ILR3	ILR2	ILR1	ILR0
bit 15							bit 8

U-0	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC
—	VECNUM6	VECNUM5	VECNUM4	VECNUM3	VECNUM2	VECNUM1	VECNUM0
bit 7							bit 0

Legend:	HSC = Hardware Settable/Clearable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **CPUIRQ:** Interrupt Request from Interrupt Controller CPU bit
1 = An interrupt request has occurred but has not yet been Acknowledged by the CPU; this happens when the CPU priority is higher than the interrupt priority
0 = No interrupt request is unacknowledged

bit 14 **Unimplemented:** Read as '0'

bit 13 **VHOLD:** Vector Number Capture Configuration bit
1 = The VECNUM bits contain the value of the highest priority pending interrupt
0 = The VECNUM bits contain the value of the last Acknowledged interrupt (i.e., the last interrupt that has occurred with higher priority than the CPU, even if other interrupts are pending)

bit 12 **Unimplemented:** Read as '0'

bit 11-8 **ILR<3:0>:** New CPU Interrupt Priority Level bits
1111 = CPU Interrupt Priority Level is 15
•
•
•
0001 = CPU Interrupt Priority Level is 1
0000 = CPU Interrupt Priority Level is 0

bit 7 **Unimplemented:** Read as '0'

bit 6-0 **VECNUM<5:0>:** Vector Number of Pending Interrupt or Last Acknowledged Interrupt bits
VHOLD = 1: The VECNUM bits indicate the vector number (from 0 to 118) of the last interrupt to occur
VHOLD = 0: The VECNUM bits indicate the vector number (from 0 to 118) of the interrupt request currently being handled

7.4 Interrupt Setup Procedures

7.4.1 INITIALIZATION

To configure an interrupt source:

1. Set the NSTDIS (INTCON1<15>) control bit if nested interrupts are not desired.
2. Select the user-assigned priority level for the interrupt source by writing the control bits in the appropriate IPCx register. The priority level will depend on the specific application and type of interrupt source. If multiple priority levels are not desired, the IPCx register control bits for all enabled interrupt sources may be programmed to the same non-zero value.

Note:	At a device Reset, the IPCx registers are initialized, such that all user interrupt sources are assigned to priority level 4.

3. Clear the interrupt flag status bit associated with the peripheral in the associated IFSx register.
4. Enable the interrupt source by setting the interrupt enable control bit associated with the source in the appropriate IECx register.

7.4.2 INTERRUPT SERVICE ROUTINE (ISR)

The method that is used to declare an Interrupt Service Routine (ISR) and initialize the IVT with the correct vector address will depend on the programming language (i.e., 'C' or assembler) and the language development toolsuite that is used to develop the application. In general, the user must clear the interrupt flag in the appropriate IFSx register for the source of the interrupt that the ISR handles. Otherwise, the ISR will be re-entered immediately after exiting the routine. If the ISR is coded in assembly language, it must be terminated using a `RETFIE` instruction to unstack the saved PC value, SRL value and old CPU priority level.

7.4.3 TRAP SERVICE ROUTINE (TSR)

A Trap Service Routine (TSR) is coded like an ISR, except that the appropriate trap status flag in the INTCON1 register must be cleared to avoid re-entry into the TSR.

7.4.4 INTERRUPT DISABLE

All user interrupts can be disabled using the following procedure:

1. Push the current SR value onto the software stack using the `PUSH` instruction.
2. Force the CPU to priority level 7 by inclusive ORing the value 0Eh with SRL.

To enable user interrupts, the `POP` instruction may be used to restore the previous SR value.

Note that only user interrupts with a priority level of 7 or less can be disabled. Trap sources (Level 8-15) cannot be disabled.

The `DISI` instruction provides a convenient way to disable interrupts of priority levels, 1-6, for a fixed period of time. Level 7 interrupt sources are not disabled by the `DISI` instruction.

8.0 OSCILLATOR CONFIGURATION

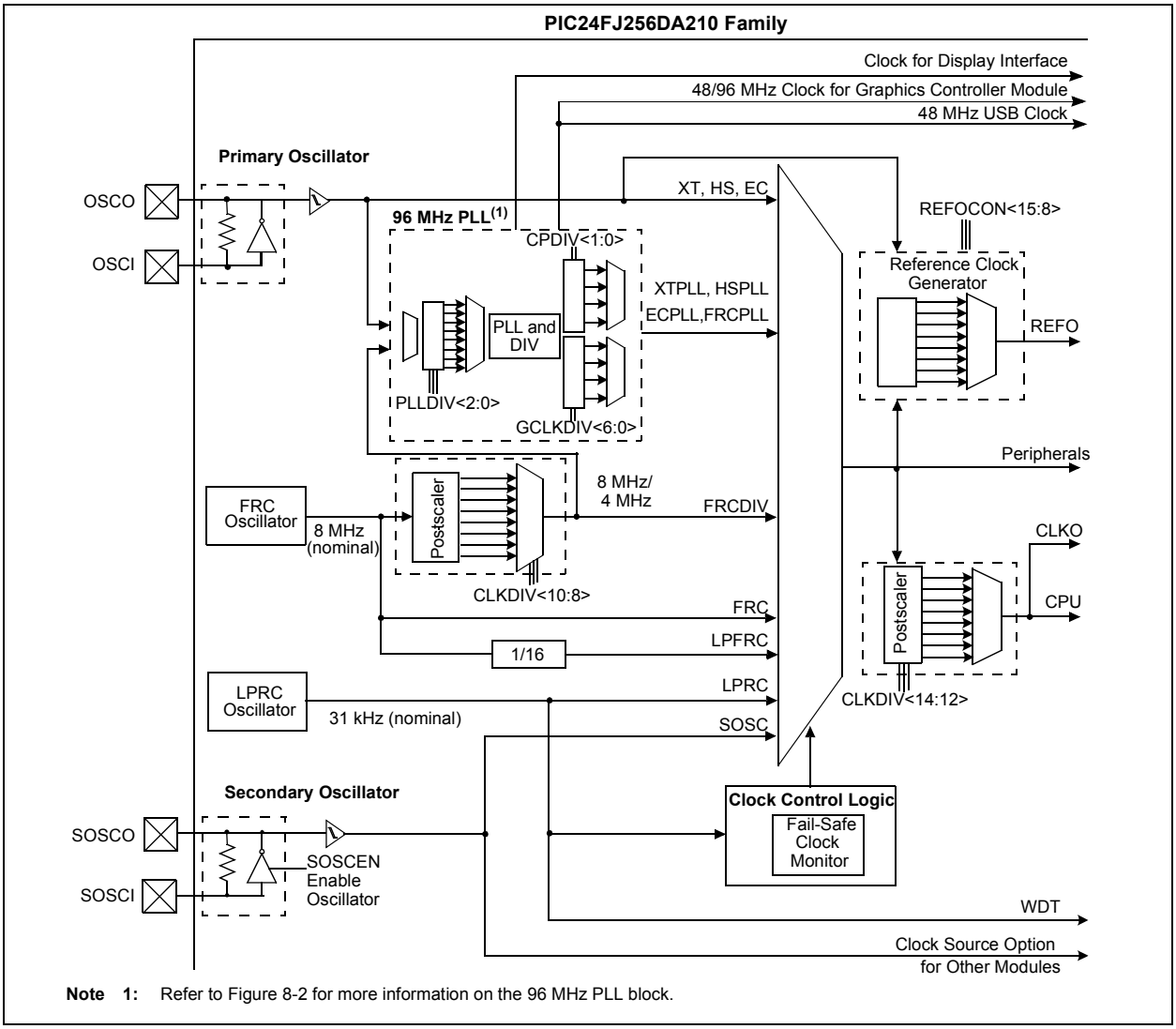
Note: This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the *"PIC24F Family Reference Manual"*, **Section 6. "Oscillator"** (DS39700). The information in this data sheet supersedes the information in the FRM.

The oscillator system for PIC24FJ256DA210 family devices has the following features:

- A total of four external and internal oscillator options as clock sources, providing 11 different clock modes
- An on-chip PLL block to boost internal operating frequency on select internal and external oscillator sources, and to provide a precise clock source for peripherals, such as USB and graphics
- Software controllable switching between various clock sources
- Software controllable postscaler for selective clocking of CPU for system power savings
- A Fail-Safe Clock Monitor (FSCM) that detects clock failure and permits safe application recovery or shutdown
- A separate and independently configurable system clock output for synchronizing external hardware

A simplified diagram of the oscillator system is shown in Figure 8-1.

FIGURE 8-1: PIC24FJ256DA210 FAMILY CLOCK DIAGRAM



Note 1: Refer to Figure 8-2 for more information on the 96 MHz PLL block.

8.1 CPU Clocking Scheme

The system clock source can be provided by one of four sources:

- Primary Oscillator (POSC) on the OSCI and OSCO pins
- Secondary Oscillator (SOSC) on the SOSCI and SOSCO pins
- Fast Internal RC (FRC) Oscillator
- Low-Power Internal RC (LPRC) Oscillator

The primary oscillator and FRC sources have the option of using the internal 24x PLL block, which generates the USB module clock, the Graphics module clock and a separate system clock through the 96 MHZ PLL. Refer to **Section 8.5 "96 MHz PLL Block"** for additional information.

The internal FRC provides an 8 MHz clock source. It can optionally be reduced by the programmable clock divider to provide a range of system clock frequencies.

The selected clock source generates the processor and peripheral clock sources. The processor clock source is divided by two to produce the internal instruction cycle clock, FCY. In this document, the instruction cycle clock is also denoted by FOSC/2. The internal instruction cycle clock, FOSC/2, can be provided on the OSCO I/O pin for some operating modes of the primary oscillator.

8.2 Initial Configuration on POR

The oscillator source (and operating mode) that is used at a device Power-on Reset (POR) event is selected using Configuration bit settings. The oscillator Configuration bit settings are located in the Configuration registers in the program memory (refer to **Section 27.1 "Configuration Bits"** for further details). The Primary Oscillator Configuration bits, POSCMD<1:0> (Configuration Word 2<1:0>) and the Initial Oscillator Select Configuration bits, FNOSC<2:0> (Configuration Word 2<10:8>), select the oscillator source that is used at a POR. The FRC primary oscillator with postscaler (FRCDIV) is the default (unprogrammed) selection. The secondary oscillator, or one of the internal oscillators, may be chosen by programming these bit locations.

The Configuration bits allow users to choose between the various clock modes, shown in Table 8-1.

8.2.1 CLOCK SWITCHING MODE CONFIGURATION BITS

The FCKSM Configuration bits (Configuration Word 2<7:6>) are used to jointly configure device clock switching and the Fail-Safe Clock Monitor (FSCM). Clock switching is enabled only when FCKSM1 is programmed ('0'). The FSCM is enabled only when FCKSM<1:0> are both programmed ('00').

TABLE 8-1: CONFIGURATION BIT VALUES FOR CLOCK SELECTION

Oscillator Mode	Oscillator Source	POSCMD<1:0>	FNOSC<2:0>	Notes
Fast RC Oscillator with Postscaler (FRCDIV)	Internal	11	111	**1, 2**
FRC Oscillator/16 (500 KHz)	Internal	11	110	**1**
Low-Power RC Oscillator (LPRC)	Internal	11	101	**1**
Secondary (Timer1) Oscillator (SOSC)	Secondary	11	100	**1**
Primary Oscillator (XT) with PLL Module (XTPLL)	Primary	01	011	**—**
Primary Oscillator (EC) with PLL Module (ECPLL)	Primary	00	011	**1**
Primary Oscillator (HS)	Primary	10	010	**—**
Primary Oscillator (XT)	Primary	01	010	**—**
Primary Oscillator (EC)	Primary	00	010	**1**
Fast RC Oscillator with PLL Module (FRCPLL)	Internal	11	001	**1**
Fast RC Oscillator (FRC)	Internal	11	000	**1**

Note 1: OSCO pin function is determined by the OSCIOFCN Configuration bit.

2: This is the default oscillator mode for an unprogrammed (erased) device.

8.3 Control Registers

The following five Special Function Registers control the operation of the oscillator:

- OSCCON
- CLKDIV
- OSCTUN
- CLKDIV2
- REFOCON

The OSCCON register (Register 8-1) is the main control register for the oscillator. It controls clock source switching and allows the monitoring of clock sources.

The CLKDIV register (Register 8-2) controls the features associated with Doze mode, as well as the postscaler for the FRC oscillator.

The OSCTUN register (Register 8-3) allows the user to fine tune the FRC oscillator over a range of approximately ±1.5%.

The CLKDIV2 register (Register 8-4) controls the clock to the display glass, with the frequency ranging from 750 kHz to 96 MHz.

The REFOCON register (Register 8-5) controls the frequency of the reference clock out.

REGISTER 8-1: OSCCON: OSCILLATOR CONTROL REGISTER

U-0	R-x, HSC(1)	R-x, HSC(1)	R-x, HSC(1)	U-0	R/W-x(1)	R/W-x(1)	R/W-x(1)
—	COSC2	COSC1	COSC0	—	NOSC2	NOSC1	NOSC0
bit 15							bit 8

R/S-0	R/W-0	R-0, HSC(3)	U-0	R/C-0, HS	R/W-0	R/W-0	R/W-0
CLKLOCK	IOLOCK(2)	LOCK	—	CF	POSCEN	SOSCEN	OSWEN
bit 7							bit 0

Legend:	C = Clearable bit	S = Settable bit	HSC = Hardware Settable/Clearable bit
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown
HS = Hardware Settable bit			

bit 15 **Unimplemented:** Read as '0'

bit 14-12 **COSC<2:0>:** Current Oscillator Selection bits(1)

- 111 = Fast RC Oscillator with Postscaler (FRCDIV)
- 110 = Fast RC/16 Oscillator
- 101 = Low-Power RC Oscillator (LPRC)
- 100 = Secondary Oscillator (SOSC)
- 011 = Primary Oscillator with PLL module (XTPLL, HSPLL, ECPLL)
- 010 = Primary Oscillator (XT, HS, EC)
- 001 = Fast RC Oscillator with Postscaler and PLL module (FRCPLL)
- 000 = Fast RC Oscillator (FRC)

bit 11 **Unimplemented:** Read as '0'

Note 1: Reset values for these bits are determined by the FNOSC Configuration bits.

2: The state of the IOLOCK bit can only be changed once an unlocking sequence has been executed. In addition, if the IOL1WAY Configuration bit is '1', once the IOLOCK bit is set, it cannot be cleared.

3: Also resets to '0' during any valid clock switch or whenever a non PLL Clock mode is selected.

REGISTER 8-1: OSCCON: OSCILLATOR CONTROL REGISTER (CONTINUED)

bit 10-8 **NOSC<2:0>:** New Oscillator Selection bits(1)
- 111 = Fast RC Oscillator with Postscaler (FRCDIV)
- 110 = Fast RC/16 Oscillator
- 101 = Low-Power RC Oscillator (LPRC)
- 100 = Secondary Oscillator (SOSC)
- 011 = Primary Oscillator with PLL module (XTPLL, HSPLL, ECPLL)
- 010 = Primary Oscillator (XT, HS, EC)
- 001 = Fast RC Oscillator with Postscaler and PLL module (FRCPLL)
- 000 = Fast RC Oscillator (FRC)

bit 7 **CLKLOCK:** Clock Selection Lock Enabled bit

If FSCM is enabled (FCKSM1 = 1):
- 1 = Clock and PLL selections are locked
- 0 = Clock and PLL selections are not locked and may be modified by setting the OSWEN bit

If FSCM is disabled (FCKSM1 = 0):
Clock and PLL selections are never locked and may be modified by setting the OSWEN bit.

bit 6 **IOLOCK:** I/O Lock Enable bit(2)
- 1 = I/O lock is active
- 0 = I/O lock is not active

bit 5 **LOCK:** PLL Lock Status bit(3)
- 1 = PLL module is in lock or PLL module start-up timer is satisfied
- 0 = PLL module is out of lock, PLL start-up timer is running or PLL is disabled

bit 4 **Unimplemented:** Read as '0'

bit 3 **CF:** Clock Fail Detect bit
- 1 = FSCM has detected a clock failure
- 0 = No clock failure has been detected

bit 2 **POSCEN:** Primary Oscillator Sleep Enable bit
- 1 = Primary Oscillator continues to operate during Sleep mode
- 0 = Primary Oscillator is disabled during Sleep mode

bit 1 **SOSCEN:** 32 kHz Secondary Oscillator (SOSC) Enable bit
- 1 = Enable the Secondary Oscillator
- 0 = Disable the Secondary Oscillator

bit 0 **OSWEN:** Oscillator Switch Enable bit
- 1 = Initiate an oscillator switch to the clock source specified by the NOSC<2:0> bits
- 0 = Oscillator switch is complete

Note 1: Reset values for these bits are determined by the FNOSC Configuration bits.

2: The state of the IOLOCK bit can only be changed once an unlocking sequence has been executed. In addition, if the IOL1WAY Configuration bit is '1', once the IOLOCK bit is set, it cannot be cleared.

3: Also resets to '0' during any valid clock switch or whenever a non PLL Clock mode is selected.

REGISTER 8-2: CLKDIV: CLOCK DIVIDER REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1
ROI	DOZE2	DOZE1	DOZE0	DOZEN(1)	RCDIV2	RCDIV1	RCDIV0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
CPDIV1	CPDIV0	PLLEN	G1CLKSEL	—	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **ROI:** Recover on Interrupt bit
1 = Interrupts clear the DOZEN bit and reset the CPU peripheral clock ratio to 1:1
0 = Interrupts have no effect on the DOZEN bit

bit 14-12 **DOZE<2:0>:** CPU Peripheral Clock Ratio Select bits
111 = 1:128
110 = 1:64
101 = 1:32
100 = 1:16
011 = 1:8
010 = 1:4
001 = 1:2
000 = 1:1

bit 11 **DOZEN:** DOZE Enable bit(1)
1 = DOZE<2:0> bits specify the CPU peripheral clock ratio
0 = CPU peripheral clock ratio is set to 1:1

bit 10-8 **RCDIV<2:0>:** FRC Postscaler Select bits
111 = 31.25 kHz (divide by 256)
110 = 125 kHz (divide by 64)
101 = 250 kHz (divide by 32)
100 = 500 kHz (divide by 16)
011 = 1 MHz (divide by 8)
010 = 2 MHz (divide by 4)
001 = 4 MHz (divide by 2)
000 = 8 MHz (divide by 1)

bit 7-6 **CPDIV<1:0>:** System Clock Select bits (postscaler select from 32 MHz clock branch)
11 = 4 MHz (divide by 8)(2)
10 = 8 MHz (divide by 4)(2)
01 = 16 MHz (divide by 2)
00 = 32 MHz (divide by 1)

Note 1: This bit is automatically cleared when the ROI bit is set and an interrupt occurs.
2: This setting is not allowed while the USB module is enabled.

REGISTER 8-2: CLKDIV: CLOCK DIVIDER REGISTER (CONTINUED)

bit 5 **PLLEN:** 96 MHz PLL Enable bit
The 96 MHz PLL must be enabled when the USB or graphics controller module is enabled. This control bit can be overridden by the PLL96MHZ (Configuration Word 2 <11>) Configuration bit.
1 = Enable the 96 MHz PLL for USB, graphics controller or HSPLL/ECPLL/FRCPLL operation
0 = Disable the 96 MHz PLL

bit 4 **G1CLKSEL:** Display Controller Module Clock Select bit
1 = Use the 96 MHz clock as a graphics controller module clock
0 = Use the 48 MHz clock as a graphics controller module clock

bit 3-0 **Unimplemented:** Read as '0'

Note 1: This bit is automatically cleared when the ROI bit is set and an interrupt occurs.
2: This setting is not allowed while the USB module is enabled.

REGISTER 8-3: OSCTUN: FRC OSCILLATOR TUNE REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	TUN5(1)	TUN4(1)	TUN3(1)	TUN2(1)	TUN1(1)	TUN0(1)
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-6 **Unimplemented:** Read as '0'

bit 5-0 **TUN<5:0>:** FRC Oscillator Tuning bits(1)
011111 = Maximum frequency deviation
011110 =
.
.
.
000001 =
000000 = Center frequency, oscillator is running at factory calibrated frequency
111111 =
.
.
.
100001 =
100000 = Minimum frequency deviation

Note 1: Increments or decrements of TUN<5:0> may not change the FRC frequency in equal steps over the FRC tuning range and may not be monotonic.

REGISTER 8-4: CLKDIV2: CLOCK DIVIDER REGISTER 2

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
GCLKDIV6(1)	GCLKDIV5(1)	GCLKDIV4(1)	GCLKDIV3(1)	GCLKDIV2(1)	GCLKDIV1(1)	GCLKDIV0(1)	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at all Resets	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-9 **GCLKDIV<6:0>:** Display Module Interface Clock Divider Selection bits(1)
(Values are based on a 96 MHz clock source set by G1CLKSEL (CLKDIV<4>) = 1. When the 48 MHz clock source is selected, G1CLKSEL (CLKDIV<4>) = 0; all values are divided by 2.)(1)
1111111 = (127) 1.50 MHz (divide by 64)
1111110 = (126) 1.52 MHz (divide by 63)
.
.
.
1100001 = (97) 2.82 MHz (divide by 34)
1100000 = (96) 2.91 MHz (divide by 33); from here, increment the divisor by 1.00
1011111 = (95) 2.95 MHz (divide by 32.50)
.
.
.
1000000 = (65) 5.49 MHz (divide by 17.50)
1000000 = (64) 5.65 MHz (divide by 17.00); from here, increment the divisor by 0.50
0111111 = (63) 5.73 MHz (divide by 16.75)
.
.
.
0000011 = (3) 54.86 MHz (divide by 1.75)
0000010 = (2) 64.00 MHz (divide by 1.5)
0000001 = (1) 76.80 MHz (divide by 1.25); from here, increment the divisor by 0.25
0000000 = (0) 96.00 MHz (divide by 1)

bit 8-0 **Unimplemented:** Read as '0'

Note 1: These bits take effect only when the 96 MHz PLL is enabled.

8.4 Clock Switching Operation

With few limitations, applications are free to switch between any of the four clock sources (POSC, SOSC, FRC and LPRC) under software control and at any time. To limit the possible side effects that could result from this flexibility, PIC24F devices have a safeguard lock built into the switching process.

Note:	The Primary Oscillator mode has three different submodes (XT, HS and EC) which are determined by the POSCMDx Configuration bits. While an application can switch to and from Primary Oscillator mode in software, it cannot switch between the different primary submodes without reprogramming the device.

8.4.1 ENABLING CLOCK SWITCHING

To enable clock switching, the FCKSM1 Configuration bit in CW2 must be programmed to '0'. (Refer to **Section 27.1 "Configuration Bits"** for further details.) If the FCKSM1 Configuration bit is unprogrammed ('1'), the clock switching function and Fail-Safe Clock Monitor function are disabled. This is the default setting.

The NOSCx (OSCCON<10:8>) control bits do not control the clock selection when clock switching is disabled. However, the COSCx (OSCCON<14:12>) control bits will reflect the clock source selected by the FNOSCx Configuration bits.

The OSWEN (OSCCON<0>) control bit has no effect when clock switching is disabled; It is held at '0' at all times.

8.4.2 OSCILLATOR SWITCHING SEQUENCE

At a minimum, performing a clock switch requires this basic sequence:

1. If desired, read the COSCx (OSCCON<14:12>) control bits to determine the current oscillator source.
2. Perform the unlock sequence to allow a write to the OSCCON register high byte.
3. Write the appropriate value to the NOSCx (OSCCON<10:8>) control bits for the new oscillator source.
4. Perform the unlock sequence to allow a write to the OSCCON register low byte.
5. Set the OSWEN bit to initiate the oscillator switch.

Once the basic sequence is completed, the system clock hardware responds automatically as follows:

1. The clock switching hardware compares the COSCx bits with the new value of the NOSCx bits. If they are the same, then the clock switch is a redundant operation. In this case, the OSWEN bit is cleared automatically and the clock switch is aborted.
2. If a valid clock switch has been initiated, the LOCK (OSCCON<5>) and CF (OSCCON<3>) bits are cleared.
3. The new oscillator is turned on by the hardware if it is not currently running. If a crystal oscillator must be turned on, the hardware will wait until the OST expires. If the new source is using the PLL, then the hardware waits until a PLL lock is detected (LOCK = 1).
4. The hardware waits for 10 clock cycles from the new clock source and then performs the clock switch.
5. The hardware clears the OSWEN bit to indicate a successful clock transition. In addition, the NOSCx bit values are transferred to the COSCx bits.
6. The old clock source is turned off at this time, with the exception of LPRC (if WDT or FSCM are enabled) or SOSC (if SOSCEN remains set).

Note 1:	The processor will continue to execute code throughout the clock switching sequence. Timing-sensitive code should not be executed during this time.
2:	Direct clock switches between any Primary Oscillator mode with PLL and FRCPLL modes are not permitted. This applies to clock switches in either direction. In these instances, the application must switch to FRC mode as a transition clock source between the two PLL modes.

A recommended code sequence for a clock switch includes the following:

1. Disable interrupts during the OSCCON register unlock and write sequence.
2. Execute the unlock sequence for the OSCCON high byte by writing 78h and 9Ah to OSCCON<15:8> in two back-to-back instructions.
3. Write new oscillator source to the NOSCx bits in the instruction immediately following the unlock sequence.
4. Execute the unlock sequence for the OSCCON low byte by writing 46h and 57h to OSCCON<7:0> in two back-to-back instructions.
5. Set the OSWEN bit in the instruction immediately following the unlock sequence.
6. Continue to execute code that is not clock-sensitive (optional).
7. Invoke an appropriate amount of software delay (cycle counting) to allow the selected oscillator and/or PLL to start and stabilize.
8. Check to see if OSWEN is '0'. If it is, the switch was successful. If OSWEN is still set, then check the LOCK bit to determine the cause of failure.

The core sequence for unlocking the OSCCON register and initiating a clock switch is shown in Example 8-1.

EXAMPLE 8-1: BASIC CODE SEQUENCE FOR CLOCK SWITCHING IN ASSEMBLY

```
;Place the new oscillator selection in W0
;OSCCONH (high byte) Unlock Sequence
MOV         #OSCCONH, w1
MOV         #0x78, w2
MOV         #0x9A, w3
MOV.b       w2, [w1]
MOV.b       w3, [w1]
;Set new oscillator selection
MOV.b       WREG, OSCCONH
;OSCCONL (low byte) unlock sequence
MOV         #OSCCONL, w1
MOV         #0x46, w2
MOV         #0x57, w3
MOV.b       w2, [w1]
MOV.b       w3, [w1]
;Start oscillator switch operation
BSET        OSCCON,#0
```

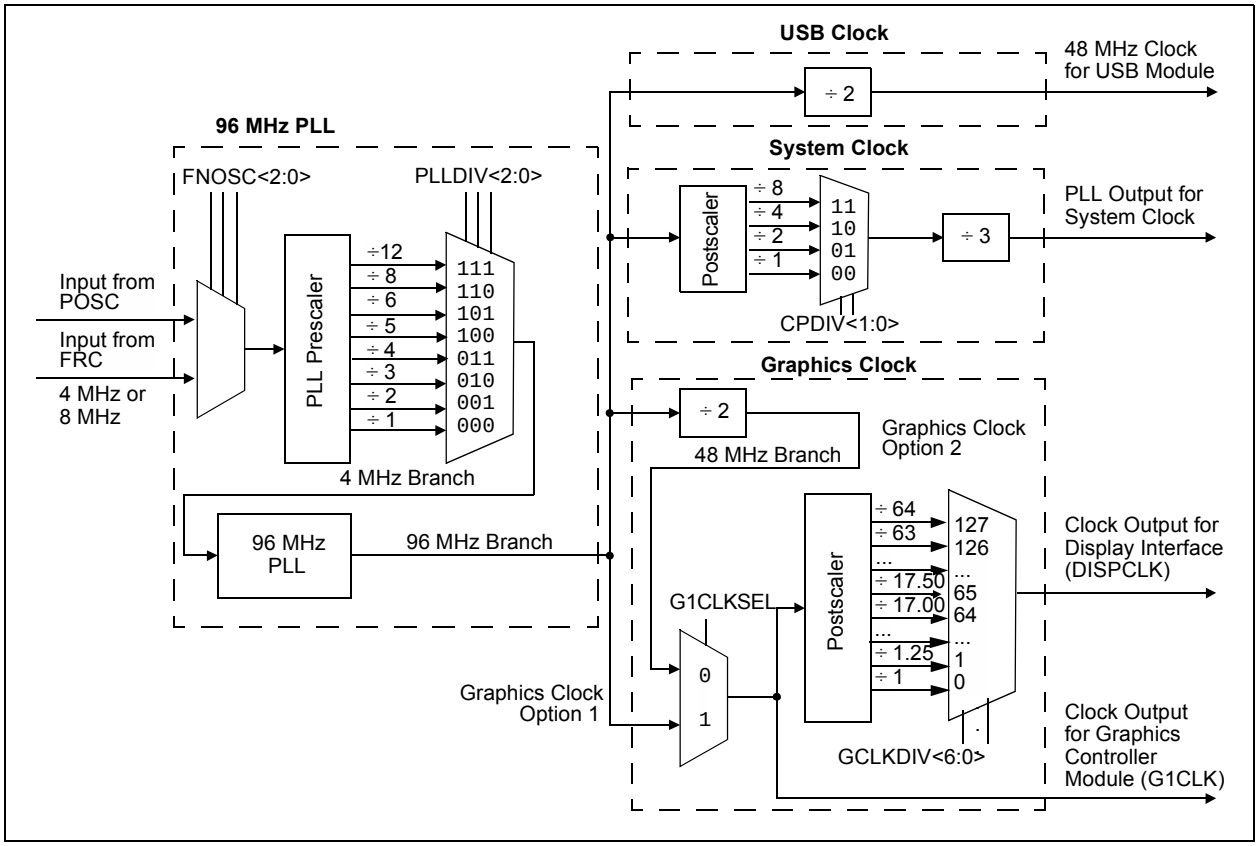
8.5 96 MHz PLL Block

The 96 MHz PLL block is implemented to generate the stable 48 MHz clock required for full-speed USB operation, a programmable clock output for the graphics controller module and the system clock from the same oscillator source. The 96 MHz PLL block is shown in Figure 8-2.

The 96 MHz PLL block requires a 4 MHz input signal; it uses this to generate a 96 MHz signal from a fixed, 24x PLL. This is, in turn, divided into three branches. The first branch generates the USB clock, the second branch generates the system clock and the third branch generates the graphics clock. The 96 MHz PLL block can be enabled and disabled using the PLL96MHZ Configuration bit (Configuration Word<11>) or through the PLLEN (CLKDIV<5>) control bit when the PLL96MHZ Configuration bit is not set. Note that the PLL96MHZ Configuration bit and PLLEN register bit are available only for PIC24F devices with USB and graphics controller modules.

The 96 MHz PLL prescaler does not automatically sense the incoming oscillator frequency. The user must manually configure the PLL divider to generate the required 4 MHz output, using the PLLDIV<2:0> Configuration bits (Configuration Word 2<14:12> in most devices).

FIGURE 8-2: 96 MHz PLL BLOCK



8.5.1 SYSTEM CLOCK GENERATION

The system clock is generated from the 96 MHz branch using a configurable postscaler/divider to generate a range of frequencies for the system clock multiplexer. The output of the multiplexer is further passed through a fixed divide-by-3 divider and the final output is used as the system clock. Figure 8-2 shows this logic in the system clock sub-block. Since the source is a 96 MHz signal, the possible system clock frequencies are listed in Table 8-2. The available system clock options are always the same, regardless of the setting of the PLLDIV Configuration bits.

TABLE 8-2: SYSTEM CLOCK OPTIONS FOR 96 MHz PLL BLOCK

MCU Clock Division (CPDIV<1:0>)	System Clock Frequency (Instruction Rate in MIPS)
None (00)	32 MHz (16)
÷2 (01)	16 MHz (8)
÷4 (10)	8 MHz (4)(1)
÷8 (11)	4 MHz (2)(1)

Note 1: These options are not compatible with USB operation. They may be used whenever the PLL branch is selected and the USB module is disabled.

8.5.2 USB CLOCK GENERATION

In the USB-On-The-Go module in PIC24FJ256DA210 family of devices, the primary oscillator with the PLL block can be used as a valid clock source for USB operation. The FRC oscillator (implemented with ±0.25% accuracy) can be combined with a PLL block, providing another option for a valid USB clock source. There is no provision to provide a separate external 48 MHz clock to the USB module. The USB module sources its clock signal from 96 MHz PLL. Due to the requirement that a 4 MHz input must be provided to generate the 96 MHz signal, the oscillator operation is limited to a range of possible values. Table 8-3 shows the valid oscillator configurations (i.e., ECPLL, HSPLL, XTPLL and FRCPLL) for USB operation. This sets the correct PLLDIV configuration for the specified oscillator frequency and the output frequency of the USB clock branch is always 48 MHz.

TABLE 8-3: VALID OSCILLATOR CONFIGURATIONS FOR USB OPERATIONS

Input Oscillator Frequency	Clock Mode	PLL Division (PLLDIV<2:0>)
48 MHz	ECPLL	÷12 (111)
32 MHz	HSPLL, ECPLL	÷ 8 (110)
24 MHz	HSPLL, ECPLL	÷6 (101)
20 MHz	HSPLL, ECPLL	÷5 (100)
16 MHz	HSPLL, ECPLL	÷4 (011)
12 MHz	HSPLL, ECPLL	÷3 (010)
8 MHz	ECPLL, HSPLL, XTPLL, FRCPLL	÷2 (001)
4 MHz	ECPLL, HSPLL, XTPLL, FRCPLL	÷1 (000)

Note: For USB devices, the use of a primary oscillator or external clock source, with a frequency above 32 MHz, does not imply that the device's system clock can be run at the same speed when the USB module is not used. The maximum system clock for all PIC24F devices is 32 MHz.

8.5.3 CONSIDERATIONS FOR USB OPERATION

When using the USB On-The-Go module in PIC24FJ256DA210 family devices, users must always observe these rules in configuring the system clock:

- For USB operation, the selected clock source (EC, HS or XT) must meet the USB clock tolerance requirements.
- The Primary Oscillator/PLL modes are the only oscillator configurations that permit USB operation. There is no provision to provide a separate external clock source to the USB module.
- While the FRCPLL Oscillator mode is used for USB applications, users must always ensure that the FRC source is configured to provide a frequency of 4 MHz or 8 MHz (RCDIV<2:0> = 001 or 000) and that the USB PLL prescaler is configured appropriately.

All other oscillator modes are available; however, USB operation is not possible when these modes are selected. They may still be useful in cases where other power levels of operation are desirable and the USB module is not needed (e.g., the application is sleeping and waiting for a bus attachment).

8.5.4 GRAPHICS CLOCK GENERATION

Two stable clock signals are generated for the graphics controller in the PIC24FJ256DA210 family of devices. The first clock is for the graphics controller module logic and the second clock is for the display module interface logic that generates the signals for the display glass. Figure 8-2 shows this logic in the graphics clock sub-block. Both clock signals are generated either from the Graphics Clock Option 1 (96 MHz branch) or the Graphics Clock Option 2 (48 MHz branch). Selection is set in the multiplexer using the G1CLKSEL (CLKDIV<4>) control bit. Graphics controller module logic directly uses the output of that multiplexer while the display module interface clock is further conditioned through a postscaler to generate 128 possible frequencies. The final clock output signal is selected through a multiplexer using the GCLKDIV<6:0> (CLKDIV2<15:9>) control bits. The 128 selections vary in increments of 0.25, 0.5, and 1.0. Refer to Table 8-4 for details. Note that for applications that use the graphics controller (GFX) module, the 96 MHz PLL must be enabled.

TABLE 8-4: DISPLAY MODULE CLOCK FREQUENCY DIVISION

GCLKDIV<6:0>	Frequency Divisor	Display Module Clock Frequency 96 MHz Input (48 MHz Input)
0000000	1	96 MHz (48 MHz)
0000001	1.25 (start incrementing by 0.25)	76.80 MHz (38.4 MHz)
0000010	1.5	64 MHz (32 MHz)
…	…	…
0111111	16.75	5.73 MHz (2.86 MHz)
1000000	17	5.65 MHz (2.82 MHz)
1000001	17.5 (start incrementing by 0.5)	5.49 MHz (2.74 MHz)
1000010	18	5.33 MHz (2.66 MHz)
…	…	…
1011111	32.5	2.95 MHz (1.47 MHz)
1100000	33	2.91 MHz (1.45 MHz)
1100001	34 (start incrementing by 1)	2.82 MHz (1.41 MHz)
1100010	35	2.74 MHz (1.37 MHz)
…	…	…
1111110	63	1.52 MHz (762 kHz)
1111111	64	1.50 MHz (750 kHz)

8.6 Reference Clock Output

In addition to the CLKO output (FOSC/2) available in certain oscillator modes, the device clock in the PIC24FJ256DA210 family devices can also be configured to provide a reference clock output signal to a port pin. This feature is available in all oscillator configurations and allows the user to select a greater range of clock submultiples to drive external devices in the application.

This reference clock output is controlled by the REFOCON register (Register 8-5). Setting the ROEN bit (REFOCON<15>) makes the clock signal available on the REFO pin. The RODIV bits (REFOCON<11:8>) enable the selection of 16 different clock divider options.

The ROSSLP and ROSEL bits (REFOCON<13:12>) control the availability of the reference output during Sleep mode. The ROSEL bit determines if the oscillator on OSCI and OSCO, or the current system clock source, is used for the reference clock output. The ROSSLP bit determines if the reference source is available on REFO when the device is in Sleep mode.

To use the reference clock output in Sleep mode, both the ROSSLP and ROSEL bits must be set. The device clock must also be configured for one of the primary modes (EC, HS or XT); otherwise, if the POSCEN bit is not also set, the oscillator on OSCI and OSCO will be powered down when the device enters Sleep mode. Clearing the ROSEL bit allows the reference output frequency to change as the system clock changes during any clock switches.

REGISTER 8-5: REFOCON: REFERENCE OSCILLATOR CONTROL REGISTER

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ROEN	—	ROSSLP	ROSEL(1)	RODIV3	RODIV2	RODIV1	RODIV0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **ROEN:** Reference Oscillator Output Enable bit
1 = Reference oscillator enabled on REFO pin
0 = Reference oscillator disabled

bit 14 **Unimplemented:** Read as '0'

bit 13 **ROSSLP:** Reference Oscillator Output Stop in Sleep bit
1 = Reference oscillator continues to run in Sleep
0 = Reference oscillator is disabled in Sleep

bit 12 **ROSEL:** Reference Oscillator Source Select bit(1)
1 = Primary oscillator used as the base clock
0 = System clock used as the base clock; base clock reflects any clock switching of the device

bit 11-8 **RODIV<3:0>:** Reference Oscillator Divisor Select bits
1111 = Base clock value divided by 32,768
1110 = Base clock value divided by 16,384
1101 = Base clock value divided by 8,192
1100 = Base clock value divided by 4,096
1011 = Base clock value divided by 2,048
1010 = Base clock value divided by 1,024
1001 = Base clock value divided by 512
1000 = Base clock value divided by 256
0111 = Base clock value divided by 128
0110 = Base clock value divided by 64
0101 = Base clock value divided by 32
0100 = Base clock value divided by 16
0011 = Base clock value divided by 8
0010 = Base clock value divided by 4
0001 = Base clock value divided by 2
0000 = Base clock value

bit 7-0 **Unimplemented:** Read as '0'

Note 1: Note that the crystal oscillator must be enabled using the FOSC<2:0> bits; the crystal maintains the operation in Sleep mode.

NOTES:

9.0 POWER-SAVING FEATURES

Note: This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the *"PIC24F Family Reference Manual"*, **Section 10. "Power-Saving Features"** (DS39698). The information in this data sheet supersedes the information in the FRM.

The PIC24FJ256DA210 family of devices provides the ability to manage power consumption by selectively managing clocking to the CPU and the peripherals. In general, a lower clock frequency and a reduction in the number of circuits being clocked constitutes lower consumed power. All PIC24F devices manage power consumption in four different ways:

- Clock Frequency
- Instruction-Based Sleep and Idle modes
- Software Controlled Doze mode
- Selective Peripheral Control in Software

Combinations of these methods can be used to selectively tailor an application's power consumption, while still maintaining critical application features, such as timing-sensitive communications.

9.1 Clock Frequency and Clock Switching

PIC24F devices allow for a wide range of clock frequencies to be selected under application control. If the system clock configuration is not locked, users can choose low-power or high-precision oscillators by simply changing the NOSC bits. The process of changing a system clock during operation, as well as limitations to the process, are discussed in more detail in **Section 8.0 "Oscillator Configuration"**.

9.2 Instruction-Based Power-Saving Modes

PIC24F devices have two special power-saving modes that are entered through the execution of a special PWRSAV instruction. Sleep mode stops clock operation and halts all code execution; Idle mode halts the CPU and code execution, but allows peripheral modules to continue operation. The assembly syntax of the PWRSAV instruction is shown in Example 9-1.

Sleep and Idle modes can be exited as a result of an enabled interrupt, WDT time-out or a device Reset. When the device exits these modes, it is said to "wake-up".

9.2.1 SLEEP MODE

Sleep mode has these features:

- The system clock source is shut down. If an on-chip oscillator is used, it is turned off.
- The device current consumption will be reduced to a minimum, provided that no I/O pin is sourcing current.
- The Fail-Safe Clock Monitor (FSCM) does not operate during Sleep mode since the system clock source is disabled.
- The LPRC clock will continue to run in Sleep mode if the WDT is enabled.
- The WDT, if enabled, is automatically cleared prior to entering Sleep mode.
- Some device features or peripherals may continue to operate in Sleep mode. This includes items such as the input change notification on the I/O ports or peripherals that use an external clock input. Any peripheral that requires the system clock source for its operation will be disabled in Sleep mode. Users can opt to make the voltage regulator enter standby mode on entering Sleep mode by clearing the VREGS bit (RCON<8>). This will decrease current consumption but will add a delay, TVREG, to the wake-up time. For this reason, applications that do not use the voltage regulator should set this bit.

The device will wake-up from Sleep mode on any of the these events:

- On any interrupt source that is individually enabled
- On any form of device Reset
- On a WDT time-out

On wake-up from sleep, the processor will restart with the same clock source that was active when Sleep mode was entered.

EXAMPLE 9-1: PWRSAV INSTRUCTION SYNTAX

```
PWRSAV    #0    ; Put the device into SLEEP mode
PWRSAV    #1    ; Put the device into IDLE mode
```

9.2.2 IDLE MODE

Idle mode has these features:

- The CPU will stop executing instructions.
- The WDT is automatically cleared.
- The system clock source remains active. By default, all peripheral modules continue to operate normally from the system clock source, but can also be selectively disabled (see **Section 9.4 "Selective Peripheral Module Control"**).
- If the WDT or FSCM is enabled, the LPRC will also remain active.

The device will wake from Idle mode on any of these events:

- Any interrupt that is individually enabled.
- Any device Reset.
- A WDT time-out.

On wake-up from Idle, the clock is reapplied to the CPU and instruction execution begins immediately, starting with the instruction following the PWRSAV instruction or the first instruction in the ISR.

9.2.3 INTERRUPTS COINCIDENT WITH POWER SAVE INSTRUCTIONS

Any interrupt that coincides with the execution of a PWRSAV instruction will be held off until entry into Sleep or Idle mode has completed. The device will then wake-up from Sleep or Idle mode.

9.3 Doze Mode

Generally, changing clock speed and invoking one of the power-saving modes are the preferred strategies for reducing power consumption. There may be circumstances, however, where this is not practical. For example, it may be necessary for an application to maintain uninterrupted synchronous communication, even while it is doing nothing else. Reducing system clock speed may introduce communication errors, while using a power-saving mode may stop communications completely.

Doze mode is a simple and effective alternative method to reduce power consumption while the device is still executing code. In this mode, the system clock continues to operate from the same source and at the same speed. Peripheral modules continue to be clocked at the same speed while the CPU clock speed is reduced. Synchronization between the two clock domains is maintained, allowing the peripherals to access the SFRs while the CPU executes code at a slower rate.

Doze mode is enabled by setting the DOZEN bit (CLKDIV<11>). The ratio between peripheral and core clock speed is determined by the DOZE<2:0> bits (CLKDIV<14:12>). There are eight possible configurations, from 1:1 to 1:128, with 1:1 being the default.

It is also possible to use Doze mode to selectively reduce power consumption in event driven applications. This allows clock-sensitive functions, such as synchronous communications, to continue without interruption while the CPU idles, waiting for something to invoke an interrupt routine. Enabling the automatic return to full-speed CPU operation on interrupts is enabled by setting the ROI bit (CLKDIV<15>). By default, interrupt events have no effect on Doze mode operation.

9.4 Selective Peripheral Module Control

Idle and Doze modes allow users to substantially reduce power consumption by slowing or stopping the CPU clock. Even so, peripheral modules still remain clocked, and thus, consume power. There may be cases where the application needs what these modes do not provide: the allocation of power resources to CPU processing with minimal power consumption from the peripherals.

PIC24F devices address this requirement by allowing peripheral modules to be selectively disabled, reducing or eliminating their power consumption. This can be done with two control bits:

- The Peripheral Enable bit, generically named, "XXXEN", located in the module's main control SFR.
- The Peripheral Module Disable (PMD) bit, generically named, "XXXMD", located in one of the PMD control registers.

Both bits have similar functions in enabling or disabling its associated module. Setting the PMD bit for a module disables all clock sources to that module, reducing its power consumption to an absolute minimum. In this state, the control and status registers associated with the peripheral will also be disabled, so writes to those registers will have no effect and read values will be invalid. Many peripheral modules have a corresponding PMD bit.

In contrast, disabling a module by clearing its XXXEN bit disables its functionality, but leaves its registers available to be read and written to. This reduces power consumption, but not by as much as setting the PMD bit does. Most peripheral modules have an enable bit; exceptions include input capture, output compare and RTCC.

To achieve more selective power savings, peripheral modules can also be selectively disabled when the device enters Idle mode. This is done through the control bit of the generic name format, "XXXIDL". By default, all modules that can operate during Idle mode will do so. Using the disable on Idle feature allows further reduction of power consumption during Idle mode, enhancing power savings for extremely critical power applications.

10.0 I/O PORTS

Note: This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the *"PIC24F Family Reference Manual"*, **Section 12. "I/O Ports with Peripheral Pin Select (PPS)"** (DS39711). The information in this data sheet supersedes the information in the FRM.

All of the device pins (except VDD, VSS, MCLR and OSCI/CLKI) are shared between the peripherals and the parallel I/O ports. All I/O input ports feature Schmitt Trigger (ST) inputs for improved noise immunity.

10.1 Parallel I/O (PIO) Ports

A parallel I/O port that shares a pin with a peripheral is, in general, subservient to the peripheral. The peripheral's output buffer data and control signals are provided to a pair of multiplexers. The multiplexers select whether the peripheral or the associated port has ownership of the output data and control signals of the I/O pin. The logic also prevents "loop through", in which a port's digital output can drive the input of a peripheral that shares the same pin. Figure 10-1 shows how ports are shared with other peripherals and the associated I/O pin to which they are connected.

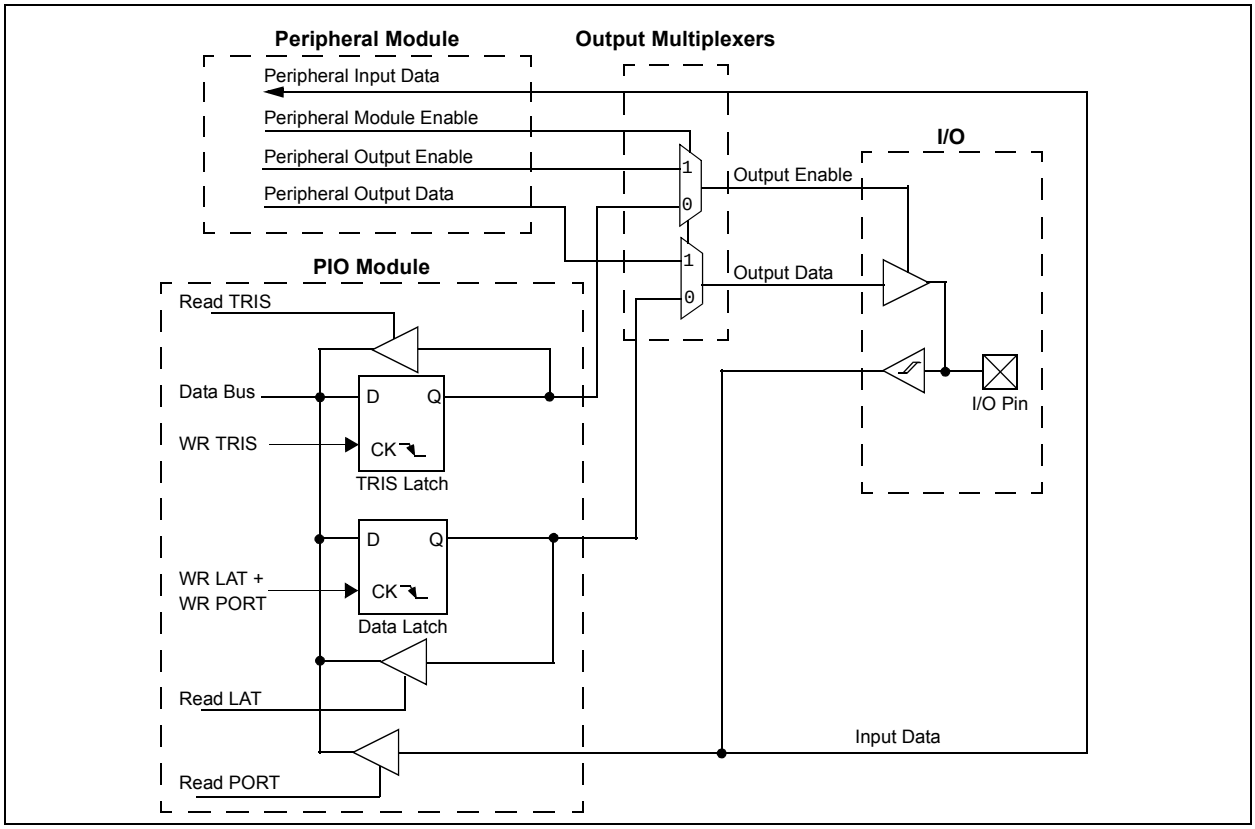
When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled. The I/O pin may be read, but the output driver for the parallel port bit will be disabled. If a peripheral is enabled, but the peripheral is not actively driving a pin, that pin may be driven by a port.

All port pins have three registers directly associated with their operation as digital I/O and one register associated with their operation as analog input. The Data Direction register (TRISx) determines whether the pin is an input or an output. If the data direction bit is a '1', then the pin is an input. All port pins are defined as inputs after a Reset. Reads from the Output Latch register (LATx), read the latch; writes to the latch, write the latch. Reads from the port (PORTx), read the port pins; writes to the port pins, write the latch.

Any bit and its associated data and control registers that are not valid for a particular device will be disabled. That means the corresponding LATx and TRISx registers, and the port pin will read as zeros.

When a pin is shared with another peripheral or function that is defined as an input only, it is regarded as a dedicated port because there is no other competing source of inputs.

FIGURE 10-1: BLOCK DIAGRAM OF A TYPICAL SHARED PORT STRUCTURE



10.1.1 I/O PORT WRITE/READ TIMING

One instruction cycle is required between a port direction change or port write operation and a read operation of the same port. Typically, this instruction would be a NOP.

10.1.2 OPEN-DRAIN CONFIGURATION

In addition to the PORT, LAT and TRIS registers for data control, each port pin can also be individually configured for either a digital or open-drain output. This is controlled by the Open-Drain Control register, ODCx, associated with each port. Setting any of the bits configures the corresponding pin to act as an open-drain output.

The open-drain feature allows the generation of outputs higher than VDD (e.g., 5V) on any desired digital only pins by using external pull-up resistors. The maximum open-drain voltage allowed is the same as the maximum VIH specification.

10.1.3 CONFIGURING D+ AND D- PINS (RG2 AND RG3)

The input buffers of the RG2 and RG3 pins are by default, tri-stated. To use these pins as input pins, the UTRDIS bit (U1CNFG2<0>) should be set which enables the input buffers on these pins.

10.2 Configuring Analog Port Pins (ANSEL)

The ANSx and TRISx registers control the operation of the pins with analog function. Each port pin with analog function is associated with one of the ANS bits (see Register 10-1 through Register 10-7), which decides if the pin function should be analog or digital. Refer to Table 10-1 for detailed behavior of the pin for different ANSx and TRISx bit settings.

When reading the PORT register, all pins configured as analog input channels will read as cleared (a low level).

10.2.1 ANALOG INPUT PINS AND VOLTAGE CONSIDERATIONS

The voltage tolerance of pins used as device inputs is dependent on the pin's input function. Pins that are used as digital only inputs are able to handle DC voltages of up to 5.5V, a level typical for digital logic circuits. In contrast, pins that also have analog input functions of any kind can only tolerate voltages up to VDD. Voltage excursions beyond VDD on these pins should always be avoided. Table 10-2 summarizes the input capabilities. Refer to **Section 30.1 "DC Characteristics"** for more details.

TABLE 10-1: CONFIGURING ANALOG/DIGITAL FUNCTION OF AN I/O PIN

Pin Function	ANSx Setting	TRISx Setting	Comments
Analog Input	1	1	It is recommended to keep ANSx = 1.
Analog Output	1	1	It is recommended to keep ANSx = 1.
Digital Input	0	1	Firmware must wait at least one instruction cycle after configuring a pin as a digital input before a valid input value can be read.
Digital Output	0	0	Make sure to disable the analog output function on the pin if any is present.

TABLE 10-2: INPUT VOLTAGE LEVELS FOR PORT OR PIN TOLERATED DESCRIPTION INPUT

Port or Pin	Tolerated Input	Description
PORTA(1)<10:9, 7:6>	VDD	Only VDD input levels are tolerated.
PORTB<15:0>		
PORTC(1)<15:12, 4>		
PORTD<7:6>		
PORTE(1)<9>		
PORTF<0>		
PORTG<9:6, 3:2>		
PORTA(1)<15:14, 5:0>	5.5V	Tolerates input levels above VDD, useful for most standard logic.
PORTC(1)<3:1>		
PORTD(1)<15:8, 5:0>		
PORTE(1)<8:0>		
PORTF(1)<13:12, 8:7, 5:1>		
PORTG(1)<15:12, 1:0>		

Note 1: Not all of the pins of these PORTS are implemented in 64-pin devices (PIC24FJXXXDAX06); refer to the device pinout diagrams for the details.

REGISTER 10-1: ANSA: PORTA ANALOG FUNCTION SELECTION REGISTER(1)

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-1	U-0
—	—	—	—	—	ANSA10	ANSA9	—
bit 15							bit 8

R/W-1	R/W-1	U-0	U-0	U-0	U-0	U-0	U-0
ANSA7	ANSA6	—	—	—	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-11 **Unimplemented:** Read as '0'

bit 10-9 **ANSA<10:9>:** Analog Function Selection bits
1 = Pin is configured in Analog mode; I/O port read is disabled
0 = Pin is configured in Digital mode; I/O port read is enabled

bit 8 **Unimplemented:** Read as '0'

bit 7-6 **ANSA<7:6>:** Analog Function Selection bits
1 = Pin is configured in Analog mode; I/O port read is disabled
0 = Pin is configured in Digital mode; I/O port read is enabled

bit 5-0 **Unimplemented:** Read as '0'

Note 1: This register is not available on 64-pin devices (PIC24FJXXXDAX06).

REGISTER 10-2: ANSB: PORTB ANALOG FUNCTION SELECTION REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
ANSB15	ANSB14	ANSB13	ANSB12	ANSB11	ANSB10	ANSB9	ANSB8
bit 15							bit 8

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
ANSB7	ANSB6	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-0 **ANSB<15:0>:** Analog Function Selection bits
1 = Pin is configured in Analog mode; I/O port read is disabled
0 = Pin is configured in Digital mode; I/O port read is enabled

REGISTER 10-3: ANSC: PORTC ANALOG FUNCTION SELECTION REGISTER

U-0	R/W-1	R/W-1	U-0	U-0	U-0	U-0	U-0
—	ANSC14	ANSC13	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	R/W-1	U-0	U-0	U-0	U-0
—	—	—	ANSC4(1)	—	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as '0'

bit 14-13 **ANSC<14:13>:** Analog Function Selection bits
1 = Pin is configured in Analog mode; I/O port read is disabled
0 = Pin is configured in Digital mode; I/O port read is enabled

bit 12-5 **Unimplemented:** Read as '0'

bit 4 **ANSC4:** Analog Function Selection bit(1)
1 = Pin is configured in Analog mode; I/O port read is disabled
0 = Pin is configured in Digital mode; I/O port read is enabled

bit 3-0 **Unimplemented:** Read as '0'

Note 1: This bit is not available on 64-pin devices (PIC24FJXXXDAX06).

REGISTER 10-4: ANSD: PORTD ANALOG FUNCTION SELECTION REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-1	R/W-1	U-0	U-0	U-0	U-0	U-0	U-0
ANSD7	ANSD6	—	—	—	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as ‘0’

bit 7-6 **ANSD<7:6>:** Analog Function Selection bits
1 = Pin is configured in Analog mode; I/O port read is disabled
0 = Pin is configured in Digital mode; I/O port read is enabled

bit 5-0 **Unimplemented:** Read as ‘0’

REGISTER 10-5: ANSE: PORTE ANALOG FUNCTION SELECTION REGISTER(1)

U-0	U-0	U-0	U-0	U-0	U-0	R/W-1	U-0
—	—	—	—	—	—	ANSE9	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-10 **Unimplemented:** Read as ‘0’

bit 9 **ANSE9:** Analog Function Selection bits
1 = Pin is configured in Analog mode; I/O port read is disabled
0 = Pin is configured in Digital mode; I/O port read is enabled

bit 8-0 **Unimplemented:** Read as ‘0’

Note 1: This register is not available in 64-pin devices (PIC24FJXXXDAX06).

REGISTER 10-6: ANSF: PORTF ANALOG FUNCTION SELECTION REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-1
—	—	—	—	—	—	—	ANSF0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-1 **Unimplemented:** Read as ‘0’

bit 0 **ANSF0:** Analog Function Selection bits
1 = Pin is configured in Analog mode; I/O port read is disabled
0 = Pin is configured in Digital mode; I/O port read is enabled

REGISTER 10-7: ANSG: PORTG ANALOG FUNCTION SELECTION REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-1
—	—	—	—	—	—	ANSG9	ANSG8
bit 15							bit 8

R/W-1	R/W-1	U-0	U-0	U-0	U-0	U-0	U-0
ANSG7	ANSG6	—	—	—	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-10 **Unimplemented:** Read as ‘0’

bit 9-6 **ANSG<9:6>:** Analog Function Selection bits
1 = Pin is configured in Analog mode; I/O port read is disabled
0 = Pin is configured in Digital mode; I/O port read is enabled

bit 5-0 **Unimplemented:** Read as ‘0’

10.3 Input Change Notification

The input change notification function of the I/O ports allows the PIC24FJ256DA210 family of devices to generate interrupt requests to the processor in response to a Change-Of-State (COS) on selected input pins. This feature is capable of detecting input Change-Of-States, even in Sleep mode, when the clocks are disabled. Depending on the device pin count, there are up to 84 external inputs that may be selected (enabled) for generating an interrupt request on a Change-Of-State.

Registers, CNEN1 through CNEN6, contain the interrupt enable control bits for each of the CN input pins. Setting any of these bits enables a CN interrupt for the corresponding pins.

Each CN pin has a both a weak pull-up and a weak pull-down connected to it. The pull-ups act as a current source that is connected to the pin, while the pull-downs act as a current sink that is connected to the pin. These eliminate the need for external resistors when push button or keypad devices are connected. The pull-ups and pull-downs are separately enabled using the CNPU1 through CNPU6 registers (for pull-ups), and the CNPD1 through CNPD6 registers (for pull-downs). Each CN pin has individual control bits for its pull-up and pull-down. Setting a control bit enables the weak pull-up or pull-down for the corresponding pin.

When the internal pull-up is selected, the pin pulls up to V_{DD} – 1.1V (typical). When the internal pull-down is selected, the pin pulls down to V_{SS}.

Note:	Pull-ups on change notification pins should always be disabled whenever the port pin is configured as a digital output.

Note:	To use CN83 and CN84, which are on the D+ and D- pins, the UTRDIS bit (U1CNFG2<0>) should be set.

EXAMPLE 10-1: PORT WRITE/READ IN ASSEMBLY

```
MOV   0xFF00, W0   ; Configure PORTB<15:8> as inputs
MOV   W0, TRISB    ; and PORTB<7:0> as outputs
NOP                ; Delay 1 cycle
BTSS  PORTB, #13   ; Next Instruction
```

EXAMPLE 10-2: PORT WRITE/READ IN 'C'

```
TRISB = 0xFF00;               // Configure PORTB<15:8> as inputs and PORTB<7:0> as outputs
Nop();                        // Delay 1 cycle
If (PORTBbits.RB13){ };       // Next Instruction
```

10.4 Peripheral Pin Select (PPS)

A major challenge in general purpose devices is providing the largest possible set of peripheral features while minimizing the conflict of features on I/O pins. In an application that needs to use more than one peripheral multiplexed on a single pin, inconvenient work arounds in application code or a complete redesign may be the only option.

The Peripheral Pin Select (PPS) feature provides an alternative to these choices by enabling the user's peripheral set selection and its placement on a wide range of I/O pins. By increasing the pinout options available on a particular device, users can better tailor the microcontroller to their entire application, rather than trimming the application to fit the device.

The Peripheral Pin Select feature operates over a fixed subset of digital I/O pins. Users may independently map the input and/or output of any one of many digital peripherals to any one of these I/O pins. PPS is performed in software and generally does not require the device to be reprogrammed. Hardware safeguards are included that prevent accidental or spurious changes to the peripheral mapping once it has been established.

10.4.1 AVAILABLE PINS

The PPS feature is used with a range of up to 44 pins, depending on the particular device and its pin count. Pins that support the Peripheral Pin Select feature include the designation, "RPn" or "RPIn", in their full pin designation, where "n" is the remappable pin number. "RP" is used to designate pins that support both remappable input and output functions, while "RPI" indicates pins that support remappable input functions only.

PIC24FJ256DA210 family devices support a larger number of remappable input only pins than remappable input/output pins. In this device family, there are up to 32 remappable input/output pins, depending on the pin count of the particular device selected; these are numbered, RP0 through RP31. Remappable input only pins are numbered above this range, from RPI32 to RPI43 (or the upper limit for that particular device).

See Table 1-1 for a summary of pinout options in each package offering.

10.4.2 AVAILABLE PERIPHERALS

The peripherals managed by the PPS are all digital only peripherals. These include general serial communications (UART and SPI), general purpose timer clock inputs, timer related peripherals (input capture and output compare) and external interrupt inputs. Also included are the outputs of the comparator module, since these are discrete digital signals.

PPS is not available for I^{2}C™, change notification inputs, RTCC alarm outputs, EPMP signals, graphics controller signals or peripherals with analog inputs.

A key difference between pin select and non-pin select peripherals is that pin select peripherals are not associated with a default I/O pin. The peripheral must always be assigned to a specific I/O pin before it can be used. In contrast, non pin select peripherals are always available on a default pin, assuming that the peripheral is active and not conflicting with another peripheral.

10.4.2.1 Peripheral Pin Select Function Priority

Pin-selectable peripheral outputs (e.g., OC, UART transmit) will take priority over general purpose digital functions on a pin, such as EPMP and port I/O. Specialized digital outputs, such as USB functionality, will take priority over PPS outputs on the same pin. The pin diagrams list peripheral outputs in the order of priority. Refer to them for priority concerns on a particular pin.

Unlike PIC24F devices with fixed peripherals, pin selectable peripheral inputs will never take ownership of a pin. The pin's output buffer will be controlled by the TRISx setting or by a fixed peripheral on the pin. If the pin is configured in digital mode then the PPS input will operate correctly. If an analog function is enabled on the pin the PPS input will be disabled.

10.4.3 CONTROLLING PERIPHERAL PIN SELECT

PPS features are controlled through two sets of Special Function Registers (SFRs): one to map peripheral inputs and one to map outputs. Because they are separately controlled, a particular peripheral's input and output (if the peripheral has both) can be placed on any selectable function pin without constraint.

The association of a peripheral to a peripheral-selectable pin is handled in two different ways, depending on if an input or an output is being mapped.

10.4.3.1 Input Mapping

The inputs of the Peripheral Pin Select options are mapped on the basis of the peripheral; that is, a control register associated with a peripheral dictates the pin it will be mapped to. The RPINRx registers are used to configure peripheral input mapping (see Register 10-8 through Register 10-28). Each register contains two sets of 6-bit fields, with each set associated with one of the pin-selectable peripherals. Programming a given peripheral's bit field with an appropriate 6-bit value maps the RPn/RPIn pin with that value to that peripheral. For any given device, the valid range of values for any of the bit fields corresponds to the maximum number of Peripheral Pin Selections supported by the device.

TABLE 10-3: SELECTABLE INPUT SOURCES (MAPS INPUT TO FUNCTION)(1)

Input Name	Function Name	Register	Function Mapping Bits
External Interrupt 1	INT1	RPINR0	INT1R<5:0>
External Interrupt 2	INT2	RPINR1	INT2R<5:0>
External Interrupt 3	INT3	RPINR1	INT3R<5:0>
External Interrupt 4	INT4	RPINR2	INT4R<5:0>
Input Capture 1	IC1	RPINR7	IC1R<5:0>
Input Capture 2	IC2	RPINR7	IC2R<5:0>
Input Capture 3	IC3	RPINR8	IC3R<5:0>
Input Capture 4	IC4	RPINR8	IC4R<5:0>
Input Capture 5	IC5	RPINR9	IC5R<5:0>
Input Capture 6	IC6	RPINR9	IC6R<5:0>
Input Capture 7	IC7	RPINR10	IC7R<5:0>
Input Capture 8	IC8	RPINR10	IC8R<5:0>
Input Capture 9	IC9	RPINR15	IC9R<5:0>
Output Compare Fault A	OCFA	RPINR11	OCFAR<5:0>
Output Compare Fault B	OCFB	RPINR11	OCFBR<5:0>
SPI1 Clock Input	SCK1IN	RPINR20	SCK1R<5:0>
SPI1 Data Input	SDI1	RPINR20	SDI1R<5:0>
SPI1 Slave Select Input	SS1IN	RPINR21	SS1R<5:0>
SPI2 Clock Input	SCK2IN	RPINR22	SCK2R<5:0>
SPI2 Data Input	SDI2	RPINR22	SDI2R<5:0>
SPI2 Slave Select Input	SS2IN	RPINR23	SS2R<5:0>
SPI3 Clock Input	SCK3IN	RPINR28	SCK3R<5:0>
SPI3 Data Input	SDI3	RPINR28	SDI3R<5:0>
SPI3 Slave Select Input	SS3IN	RPINR29	SS3R<5:0>
Timer2 External Clock	T2CK	RPINR3	T2CKR<5:0>
Timer3 External Clock	T3CK	RPINR3	T3CKR<5:0>
Timer4 External Clock	T4CK	RPINR4	T4CKR<5:0>
Timer5 External Clock	T5CK	RPINR4	T5CKR<5:0>
UART1 Clear To Send	$\overline{\text{U1CTS}}$	RPINR18	U1CTSR<5:0>
UART1 Receive	U1RX	RPINR18	U1RXR<5:0>
UART2 Clear To Send	$\overline{\text{U2CTS}}$	RPINR19	U2CTSR<5:0>
UART2 Receive	U2RX	RPINR19	U2RXR<5:0>
UART3 Clear To Send	$\overline{\text{U3CTS}}$	RPINR21	U3CTSR<5:0>
UART3 Receive	U3RX	RPINR17	U3RXR<5:0>
UART4 Clear To Send	$\overline{\text{U4CTS}}$	RPINR27	U4CTSR<5:0>
UART4 Receive	U4RX	RPINR27	U4RXR<5:0>

Note 1: Unless otherwise noted, all inputs use the Schmitt Trigger (ST) input buffers.

10.4.3.2 Output Mapping

In contrast to inputs, the outputs of the Peripheral Pin Select options are mapped on the basis of the pin. In this case, a control register associated with a particular pin dictates the peripheral output to be mapped. The RPORx registers are used to control output mapping. Each register contains two 6-bit fields, with each field being associated with one RPn pin (see Register 10-29 through Register 10-44). The value of the bit field corresponds to one of the peripherals and that peripheral's output is mapped to the pin (see Table 10-4).

Because of the mapping technique, the list of peripherals for output mapping also includes a null value of '000000'. This permits any given pin to remain disconnected from the output of any of the pin-selectable peripherals.

TABLE 10-4: SELECTABLE OUTPUT SOURCES (MAPS FUNCTION TO OUTPUT)

Output Function Number(1)	Function	Output Name
0	NULL(2)	Null
1	C1OUT	Comparator 1 Output
2	C2OUT	Comparator 2 Output
3	U1TX	UART1 Transmit
4	$\overline{\text{U1RTS}}$(3)	UART1 Request To Send
5	U2TX	UART2 Transmit
6	$\overline{\text{U2RTS}}$(3)	UART2 Request To Send
7	SDO1	SPI1 Data Output
8	SCK1OUT	SPI1 Clock Output
9	SS1OUT	SPI1 Slave Select Output
10	SDO2	SPI2 Data Output
11	SCK2OUT	SPI2 Clock Output
12	SS2OUT	SPI2 Slave Select Output
18	OC1	Output Compare 1
19	OC2	Output Compare 2
20	OC3	Output Compare 3
21	OC4	Output Compare 4
22	OC5	Output Compare 5
23	OC6	Output Compare 6
24	OC7	Output Compare 7
25	OC8	Output Compare 8
28	U3TX	UART3 Transmit
29	$\overline{\text{U3RTS}}$(3)	UART3 Request To Send
30	U4TX	UART4 Transmit
31	$\overline{\text{U4RTS}}$(3)	UART4 Request To Send
32	SDO3	SPI3 Data Output
33	SCK3OUT	SPI3 Clock Output
34	SS3OUT	SPI3 Slave Select Output
35	OC9	Output Compare 9
36	C3OUT	Comparator 3 Output
37-63	(unused)	NC

Note 1: Setting the RPORx register with the listed value assigns that output function to the associated RPn pin.

2: The NULL function is assigned to all RPn outputs at device Reset and disables the RPn output function.

3: IrDA® BCLK functionality uses this output.

10.4.3.3 Mapping Limitations

The control schema of the Peripheral Pin Select is extremely flexible. Other than systematic blocks that prevent signal contention, caused by two physical pins being configured as the same functional input or two functional outputs configured as the same pin, there are no hardware enforced lock outs. The flexibility extends to the point of allowing a single input to drive multiple peripherals or a single functional output to drive multiple output pins.

10.4.3.4 Mapping Exceptions for PIC24FJ256DA210 Devices

Although the PPS registers theoretically allow for up to 64 remappable I/O pins, not all of these are implemented in all devices. For PIC24FJ256DA210 family devices, the maximum number of remappable pins available are 44, which includes 12 input only pins. In addition, some pins in the RP and RPI sequences are unimplemented in lower pin count devices. The differences in available remappable pins are summarized in Table 10-5.

When developing applications that use remappable pins, users should also keep these things in mind:

- For the RPINRx registers, bit combinations corresponding to an unimplemented pin for a particular device are treated as invalid; the corresponding module will not have an input mapped to it. For all PIC24FJ256DA210 family devices, this includes all values greater than 43 (‘101011’).
- For RPORx registers, the bit fields corresponding to an unimplemented pin will also be unimplemented. Writing to these fields will have no effect.

10.4.4 CONTROLLING CONFIGURATION CHANGES

Because peripheral remapping can be changed during run time, some restrictions on peripheral remapping are needed to prevent accidental configuration changes. PIC24F devices include three features to prevent alterations to the peripheral map:

- Control register lock sequence
- Continuous state monitoring
- Configuration bit remapping lock

10.4.4.1 Control Register Lock

Under normal operation, writes to the RPINRx and RPORx registers are not allowed. Attempted writes will appear to execute normally, but the contents of the registers will remain unchanged. To change these registers, they must be unlocked in hardware. The register lock is controlled by the IOLOCK bit (OSCCON<6>). Setting IOLOCK prevents writes to the control registers; clearing IOLOCK allows writes.

To set or clear IOLOCK, a specific command sequence must be executed:

1. Write 46h to OSCCON<7:0>.
2. Write 57h to OSCCON<7:0>.
3. Clear (or set) IOLOCK as a single operation.

Unlike the similar sequence with the oscillator’s LOCK bit, IOLOCK remains in one state until changed. This allows all of the Peripheral Pin Selects to be configured with a single unlock sequence, followed by an update to all control registers, then locked with a second lock sequence.

10.4.4.2 Continuous State Monitoring

In addition to being protected from direct writes, the contents of the RPINRx and RPORx registers are constantly monitored in hardware by shadow registers. If an unexpected change in any of the registers occurs (such as cell disturbances caused by ESD or other external events), a Configuration Mismatch Reset will be triggered.

10.4.4.3 Configuration Bit Pin Select Lock

As an additional level of safety, the device can be configured to prevent more than one write session to the RPINRx and RPORx registers. The IOL1WAY (CW2<4>) Configuration bit blocks the IOLOCK bit from being cleared after it has been set once. If IOLOCK remains set, the register unlock procedure will not execute and the Peripheral Pin Select Control registers cannot be written to. The only way to clear the bit and re-enable peripheral remapping is to perform a device Reset.

In the default (unprogrammed) state, IOL1WAY is set, restricting users to one write session. Programming IOL1WAY allows users unlimited access (with the proper use of the unlock sequence) to the Peripheral Pin Select registers.

TABLE 10-5: REMAPPABLE PIN EXCEPTIONS FOR PIC24FJ256DA210 FAMILY DEVICES

Device Pin Count	RP Pins (I/O)		RPI Pins	
	Total	Unimplemented	Total	Unimplemented
64-Pin (PIC24FJXXXDAX06)	28	RP5, RP15, RP30, RP31	1	RPI32-36, RPI38-43
100/121-Pin (PIC24FJXXXDAX10)	32	—	12	—

10.4.5 CONSIDERATIONS FOR PERIPHERAL PIN SELECTION

The ability to control Peripheral Pin Selection introduces several considerations into application design that could be overlooked. This is particularly true for several common peripherals that are available only as remappable peripherals.

The main consideration is that the Peripheral Pin Selects are not available on default pins in the device's default (Reset) state. Since all RPINRx registers reset to '111111' and all RPORx registers reset to '000000', all Peripheral Pin Select inputs are tied to Vss and all Peripheral Pin Select outputs are disconnected.

Note:	In tying Peripheral Pin Select inputs to RP63, RP63 need not exist on a device for the registers to be reset to it.

This situation requires the user to initialize the device with the proper peripheral configuration before any other application code is executed. Since the IOLOCK bit resets in the unlocked state, it is not necessary to execute the unlock sequence after the device has come out of Reset. For application safety, however, it is best to set IOLOCK and lock the configuration after writing to the control registers.

Because the unlock sequence is timing-critical, it must be executed as an assembly language routine in the same manner as changes to the oscillator configuration. If the bulk of the application is written in 'C', or another high-level language, the unlock sequence should be performed by writing in-line assembly.

Choosing the configuration requires the review of all Peripheral Pin Selects and their pin assignments, especially those that will not be used in the application. In all cases, unused pin-selectable peripherals should be disabled completely. Unused peripherals should have their inputs assigned to an unused RPn/RPIn pin function. I/O pins with unused RPn functions should be configured with the null peripheral output.

The assignment of a peripheral to a particular pin does not automatically perform any other configuration of the pin's I/O circuitry. In theory, this means adding a pin-selectable output to a pin may mean inadvertently driving an existing peripheral input when the output is driven. Users must be familiar with the behavior of other fixed peripherals that share a remappable pin and know when to enable or disable them. To be safe, fixed digital peripherals that share the same pin should be disabled when not in use.

Along these lines, configuring a remappable pin for a specific peripheral does not automatically turn that feature on. The peripheral must be specifically configured for operation, and enabled as if it were tied to a fixed pin. Where this happens in the application code (immediately following device Reset and peripheral configuration or inside the main application routine) depends on the peripheral and its use in the application.

A final consideration is that Peripheral Pin Select functions neither override analog inputs nor reconfigure pins with analog functions for digital I/O. If a pin is configured as an analog input on device Reset, it must be explicitly reconfigured as digital I/O when used with a Peripheral Pin Select.

Example 10-3 shows a configuration for bidirectional communication with flow control using UART1. The following input and output functions are used:

- Input Functions: U1RX, $\overline{\text{U1CTS}}$
- Output Functions: U1TX, $\overline{\text{U1RTS}}$

EXAMPLE 10-3: CONFIGURING UART1 INPUT AND OUTPUT FUNCTIONS

```
// Unlock Registers
asm volatile(  "MOV    #OSCCON, w1    \n"
               "MOV    #0x46, w2      \n"
               "MOV    #0x57, w3      \n"
               "MOV.b  w2, [w1]       \n"
               "MOV.b  w3, [w1]       \n"
               "BCLR OSCCON,#6");

// or use C30 built-in macro:
// __builtin_write_OSCCONL(OSCCON & 0xbf);

// Configure Input Functions (Table 10-2))
    // Assign U1RX To Pin RP0
    RPINR18bits.U1RXR = 0;

    // Assign U1CTS To Pin RP1
    RPINR18bits.U1CTSR = 1;

// Configure Output Functions (Table 10-4)
    // Assign U1TX To Pin RP2
    RPOR1bits.RP2R = 3;

    // Assign U1RTS To Pin RP3
    RPOR1bits.RP3R = 4;

// Lock Registers
asm volatile   ("MOV   #OSCCON, w1    \n"
               "MOV    #0x46, w2      \n"
               "MOV    #0x57, w3      \n"
               "MOV.b  w2, [w1]\      n"
               "MOV.b  w3, [w1]       \n"
               "BSET   OSCCON, #6")   ;

// or use C30 built-in macro:
// __builtin_write_OSCCONL(OSCCON | 0x40);
```

10.4.6 PERIPHERAL PIN SELECT REGISTERS

The PIC24FJ256DA210 family of devices implements a total of 37 registers for remappable peripheral configuration:

- Input Remappable Peripheral Registers (21)
- Output Remappable Peripheral Registers (16)

Note: Input and output register values can only be changed if IOLOCK (OSCCON<6>) = 0. See **Section 10.4.4.1 "Control Register Lock"** for a specific command sequence.

REGISTER 10-8: RPINR0: PERIPHERAL PIN SELECT INPUT REGISTER 0

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	INT1R5	INT1R4	INT1R3	INT1R2	INT1R1	INT1R0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13-8 **INT1R<5:0>:** Assign External Interrupt 1 (INT1) to Corresponding RPn or RPIn Pin bits

bit 7-0 **Unimplemented:** Read as '0'

REGISTER 10-9: RPINR1: PERIPHERAL PIN SELECT INPUT REGISTER 1

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	INT3R5	INT3R4	INT3R3	INT3R2	INT3R1	INT3R0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	INT2R5	INT2R4	INT2R3	INT2R2	INT2R1	INT2R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13-8 **INT3R<5:0>:** Assign External Interrupt 3 (INT3) to Corresponding RPn or RPIn Pin bits

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **INT2R<5:0>:** Assign External Interrupt 2 (INT2) to Corresponding RPn or RPIn Pin bits

REGISTER 10-10: RPINR2: PERIPHERAL PIN SELECT INPUT REGISTER 2

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	INT4R5	INT4R4	INT4R3	INT4R2	INT4R1	INT4R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-6 **Unimplemented:** Read as '0'

bit 5-0 **INT4R<5:0>:** Assign External Interrupt 4 (INT4) to Corresponding RPn or RPIn Pin bits

REGISTER 10-11: RPINR3: PERIPHERAL PIN SELECT INPUT REGISTER 3

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	T3CKR5	T3CKR4	T3CKR3	T3CKR2	T3CKR1	T3CKR0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	T2CKR5	T2CKR4	T2CKR3	T2CKR2	T2CKR1	T2CKR0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13-8 **T3CKR<5:0>:** Assign Timer3 External Clock (T3CK) to Corresponding RPn or RPIn Pin bits

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **T2CKR<5:0>:** Assign Timer2 External Clock (T2CK) to Corresponding RPn or RPIn Pin bits

REGISTER 10-12: RPINR4: PERIPHERAL PIN SELECT INPUT REGISTER 4

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	T5CKR5	T5CKR4	T5CKR3	T5CKR2	T5CKR1	T5CKR0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	T4CKR5	T4CKR4	T4CKR3	T4CKR2	T4CKR1	T4CKR0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13-8 **T5CKR<5:0>:** Assign Timer5 External Clock (T5CK) to Corresponding RPn or RPIn Pin bits

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **T4CKR<5:0>:** Assign Timer4 External Clock (T4CK) to Corresponding RPn or RPIn Pin bits

REGISTER 10-13: RPINR7: PERIPHERAL PIN SELECT INPUT REGISTER 7

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC2R5	IC2R4	IC2R3	IC2R2	IC2R1	IC2R0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC1R5	IC1R4	IC1R3	IC1R2	IC1R1	IC1R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13-8 **IC2R<5:0>:** Assign Input Capture 2 (IC2) to Corresponding RPn or RPIn Pin bits

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **IC1R<5:0>:** Assign Input Capture 1 (IC1) to Corresponding RPn or RPIn Pin bits

REGISTER 10-14: RPINR8: PERIPHERAL PIN SELECT INPUT REGISTER 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC4R5	IC4R4	IC4R3	IC4R2	IC4R1	IC4R0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC3R5	IC3R4	IC3R3	IC3R2	IC3R1	IC3R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13-8 **IC4R<5:0>:** Assign Input Capture 4 (IC4) to Corresponding RPn or RPIn Pin bits

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **IC3R<5:0>:** Assign Input Capture 3 (IC3) to Corresponding RPn or RPIn Pin bits

REGISTER 10-15: RPINR9: PERIPHERAL PIN SELECT INPUT REGISTER 9

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC6R5	IC6R4	IC6R3	IC6R2	IC6R1	IC6R0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC5R5	IC5R4	IC5R3	IC5R2	IC5R1	IC5R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13-8 **IC6R<5:0>:** Assign Input Capture 6 (IC6) to Corresponding RPn or RPIn Pin bits

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **IC5R<5:0>:** Assign Input Capture 5 (IC5) to Corresponding RPn or RPIn Pin bits

REGISTER 10-16: RPINR10: PERIPHERAL PIN SELECT INPUT REGISTER 10

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC8R5	IC8R4	IC8R3	IC8R2	IC8R1	IC8R0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC7R5	IC7R4	IC7R3	IC7R2	IC7R1	IC7R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as ‘0’

bit 13-8 **IC8R<5:0>:** Assign Input Capture 8 (IC8) to Corresponding RPn or RPIn Pin bits

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **IC7R<5:0>:** Assign Input Capture 7 (IC7) to Corresponding RPn or RPIn Pin bits

REGISTER 10-17: RPINR11: PERIPHERAL PIN SELECT INPUT REGISTER 11

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	OCFBR5	OCFBR4	OCFBR3	OCFBR2	OCFBR1	OCFBR0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	OCFAR5	OCFAR4	OCFAR3	OCFAR2	OCFAR1	OCFAR0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as ‘0’

bit 13-8 **OCFBR<5:0>:** Assign Output Compare Fault B (OCFB) to Corresponding RPn or RPIn Pin bits

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **OCFAR<5:0>:** Assign Output Compare Fault A (OCFA) to Corresponding RPn or RPIn Pin bits

REGISTER 10-18: RPINR15: PERIPHERAL PIN SELECT INPUT REGISTER 15

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC9R5	IC9R4	IC9R3	IC9R2	IC9R1	IC9R0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13-8 **IC9R<5:0>:** Assign Input Capture 9 (IC9) to Corresponding RPn or RPIn Pin bits

bit 7-0 **Unimplemented:** Read as '0'

REGISTER 10-19: RPINR17: PERIPHERAL PIN SELECT INPUT REGISTER 17

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	U3RXR5	U3RXR4	U3RXR3	U3RXR2	U3RXR1	U3RXR0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13-8 **U3RXR<5:0>:** Assign UART3 Receive (U3RX) to Corresponding RPn or RPIn Pin bits

bit 7-0 **Unimplemented:** Read as '0'

REGISTER 10-20: RPINR18: PERIPHERAL PIN SELECT INPUT REGISTER 18

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	U1CTSR5	U1CTSR4	U1CTSR3	U1CTSR2	U1CTSR1	U1CTSR0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	U1RXR5	U1RXR4	U1RXR3	U1RXR2	U1RXR1	U1RXR0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13-8 **U1CTSR<5:0>:** Assign UART1 Clear to Send ($\overline{\text{U1CTS}}$) to Corresponding RPn or RPIn Pin bits

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **U1RXR<5:0>:** Assign UART1 Receive (U1RX) to Corresponding RPn or RPIn Pin bits

REGISTER 10-21: RPINR19: PERIPHERAL PIN SELECT INPUT REGISTER 19

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	U2CTSR5	U2CTSR4	U2CTSR3	U2CTSR2	U2CTSR1	U2CTSR0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	U2RXR5	U2RXR4	U2RXR3	U2RXR2	U2RXR1	U2RXR0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13-8 **U2CTSR<5:0>:** Assign UART2 Clear to Send ($\overline{\text{U2CTS}}$) to Corresponding RPn or RPIn Pin bits

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **U2RXR<5:0>:** Assign UART2 Receive (U2RX) to Corresponding RPn or RPIn Pin bits

REGISTER 10-22: RPINR20: PERIPHERAL PIN SELECT INPUT REGISTER 20

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SCK1R5	SCK1R4	SCK1R3	SCK1R2	SCK1R1	SCK1R0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SDI1R5	SDI1R4	SDI1R3	SDI1R2	SDI1R1	SDI1R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13-8 **SCK1R<5:0>:** Assign SPI1 Clock Input (SCK1IN) to Corresponding RPn or RPIn Pin bits

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **SDI1R<5:0>:** Assign SPI1 Data Input (SDI1) to Corresponding RPn or RPIn Pin bits

REGISTER 10-23: RPINR21: PERIPHERAL PIN SELECT INPUT REGISTER 21

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	U3CTSR5	U3CTSR4	U3CTSR3	U3CTSR2	U3CTSR1	U3CTSR0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SS1R5	SS1R4	SS1R3	SS1R2	SS1R1	SS1R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13-8 **U3CTSR<5:0>:** Assign UART3 Clear to Send ($\overline{\text{U3CTS}}$) to Corresponding RPn or RPIn Pin bits

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **SS1R<5:0>:** Assign SPI1 Slave Select Input (SS1IN) to Corresponding RPn or RPIn Pin bits

REGISTER 10-24: RPINR22: PERIPHERAL PIN SELECT INPUT REGISTER 22

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SCK2R5	SCK2R4	SCK2R3	SCK2R2	SCK2R1	SCK2R0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SDI2R5	SDI2R4	SDI2R3	SDI2R2	SDI2R1	SDI2R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as ‘0’

bit 13-8 **SCK2R<5:0>:** Assign SPI2 Clock Input (SCK2IN) to Corresponding RPn or RPIn Pin bits

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **SDI2R<5:0>:** Assign SPI2 Data Input (SDI2) to Corresponding RPn or RPIn Pin bits

REGISTER 10-25: RPINR23: PERIPHERAL PIN SELECT INPUT REGISTER 23

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SS2R5	SS2R4	SS2R3	SS2R2	SS2R1	SS2R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-6 **Unimplemented:** Read as ‘0’

bit 5-0 **SS2R<5:0>:** Assign SPI2 Slave Select Input (SS2IN) to Corresponding RPn or RPIn Pin bits

REGISTER 10-26: RPINR27: PERIPHERAL PIN SELECT INPUT REGISTER 27

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	U4CTSR5	U4CTSR4	U4CTSR3	U4CTSR2	U4CTSR1	U4CTSR0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	U4RXR5	U4RXR4	U4RXR3	U4RXR2	U4RXR1	U4RXR0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13-8 **U4CTSR<5:0>:** Assign UART4 Clear to Send (U4CTS) to Corresponding RPn or RPIn Pin bits

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **U4RXR<5:0>:** Assign UART4 Receive (U4RX) to Corresponding RPn or RPIn Pin bits

REGISTER 10-27: RPINR28: PERIPHERAL PIN SELECT INPUT REGISTER 28

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SCK3R5	SCK3R4	SCK3R3	SCK3R2	SCK3R1	SCK3R0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SDI3R5	SDI3R4	SDI3R3	SDI3R2	SDI3R1	SDI3R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13-8 **SCK3R<5:0>:** Assign SPI3 Clock Input (SCK3IN) to Corresponding RPn or RPIn Pin bits

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **SDI3R<5:0>:** Assign SPI3 Data Input (SDI3) to Corresponding RPn or RPIn Pin bits

REGISTER 10-28: RPINR29: PERIPHERAL PIN SELECT INPUT REGISTER 29

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SS3R5	SS3R4	SS3R3	SS3R2	SS3R1	SS3R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-6 **Unimplemented:** Read as ‘0’

bit 5-0 **SS3R<5:0>:** Assign SPI3 Slave Select Input (SS31IN) to Corresponding RPn or RPIn Pin bits

REGISTER 10-29: RPOR0: PERIPHERAL PIN SELECT OUTPUT REGISTER 0

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP1R5	RP1R4	RP1R3	RP1R2	RP1R1	RP1R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP0R5	RP0R4	RP0R3	RP0R2	RP0R1	RP0R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13-8 **RP1R<5:0>:** RP1 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP1 (see Table 10-4 for peripheral function numbers).

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **RP0R<5:0>:** RP0 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP0 (see Table 10-4 for peripheral function numbers).

REGISTER 10-30: RPOR1: PERIPHERAL PIN SELECT OUTPUT REGISTER 1

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP3R5	RP3R4	RP3R3	RP3R2	RP3R1	RP3R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP2R5	RP2R4	RP2R3	RP2R2	RP2R1	RP2R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13-8 **RP3R<5:0>:** RP3 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP3 (see Table 10-4 for peripheral function numbers).

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **RP2R<5:0>:** RP2 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP2 (see Table 10-4 for peripheral function numbers).

REGISTER 10-31: RPOR2: PERIPHERAL PIN SELECT OUTPUT REGISTER 2

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP5R5(1)	RP5R4(1)	RP5R3(1)	RP5R2(1)	RP5R1(1)	RP5R0(1)
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP4R5	RP4R4	RP4R3	RP4R2	RP4R1	RP4R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13-8 **RP5R<5:0>:** RP5 Output Pin Mapping bits(1)
Peripheral output number n is assigned to pin, RP5 (see Table 10-4 for peripheral function numbers).

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **RP4R<5:0>:** RP4 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP4 (see Table 10-4 for peripheral function numbers).

Note 1: Unimplemented in 64-pin devices; read as '0'.

REGISTER 10-32: RPOR3: PERIPHERAL PIN SELECT OUTPUT REGISTER 3

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP7R5	RP7R4	RP7R3	RP7R2	RP7R1	RP7R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP6R5	RP6R4	RP6R3	RP6R2	RP6R1	RP6R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13-8 **RP7R<5:0>:** RP7 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP7 (see Table 10-4 for peripheral function numbers).

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **RP6R<5:0>:** RP6 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP6 (see Table 10-4 for peripheral function numbers).

REGISTER 10-33: RPOR4: PERIPHERAL PIN SELECT OUTPUT REGISTER 4

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP9R5	RP9R4	RP9R3	RP9R2	RP9R1	RP9R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP8R5	RP8R4	RP8R3	RP8R2	RP8R1	RP8R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13-8 **RP9R<5:0>:** RP9 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP9 (see Table 10-4 for peripheral function numbers).

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **RP8R<5:0>:** RP8 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP8 (see Table 10-4 for peripheral function numbers).

REGISTER 10-34: RPOR5: PERIPHERAL PIN SELECT OUTPUT REGISTER 5

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP11R5	RP11R4	RP11R3	RP11R2	RP11R1	RP11R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP10R5	RP10R4	RP10R3	RP10R2	RP10R1	RP10R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13-8 **RP11R<5:0>:** RP11 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP11 (see Table 10-4 for peripheral function numbers).

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **RP10R<5:0>:** RP10 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP10 (see Table 10-4 for peripheral function numbers).

REGISTER 10-35: RPOR6: PERIPHERAL PIN SELECT OUTPUT REGISTER 6

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP13R5	RP13R4	RP13R3	RP13R2	RP13R1	RP13R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP12R5	RP12R4	RP12R3	RP12R2	RP12R1	RP12R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13-8 **RP13R<5:0>:** RP13 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP13 (see Table 10-4 for peripheral function numbers).

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **RP12R<5:0>:** RP12 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP12 (see Table 10-4 for peripheral function numbers).

REGISTER 10-36: RPOR7: PERIPHERAL PIN SELECT OUTPUT REGISTER 7

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP15R5(1)	RP15R4(1)	RP15R3(1)	RP15R2(1)	RP15R1(1)	RP15R0(1)
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP14R5	RP14R4	RP14R3	RP14R2	RP14R1	RP14R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13-8 **RP15R<5:0>:** RP15 Output Pin Mapping bits(1)
Peripheral output number n is assigned to pin, RP0 (see Table 10-4 for peripheral function numbers).

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **RP14R<5:0>:** RP14 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP14 (see Table 10-4 for peripheral function numbers).

Note 1: Unimplemented in 64-pin devices; read as '0'.

REGISTER 10-37: RPOR8: PERIPHERAL PIN SELECT OUTPUT REGISTER 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP17R5	RP17R4	RP17R3	RP17R2	RP17R1	RP17R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP16R5	RP16R4	RP16R3	RP16R2	RP16R1	RP16R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13-8 **RP17R<5:0>:** RP17 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP17 (see Table 10-4 for peripheral function numbers).

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **RP16R<5:0>:** RP16 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP16 (see Table 10-4 for peripheral function numbers).

REGISTER 10-38: RPOR9: PERIPHERAL PIN SELECT OUTPUT REGISTER 9

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP19R5	RP19R4	RP19R3	RP19R2	RP19R1	RP19R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP18R5	RP18R4	RP18R3	RP18R2	RP18R1	RP18R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13-8 **RP19R<5:0>:** RP19 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP19 (see Table 10-4 for peripheral function numbers).

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **RP18R<5:0>:** RP18 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP18 (see Table 10-4 for peripheral function numbers).

REGISTER 10-39: RPOR10: PERIPHERAL PIN SELECT OUTPUT REGISTER 10

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP21R5	RP21R4	RP21R3	RP21R2	RP21R1	RP21R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP20R5	RP20R4	RP20R3	RP20R2	RP20R1	RP20R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as ‘0’

bit 13-8 **RP21R<5:0>:** RP21 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP21 (see Table 10-4 for peripheral function numbers).

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **RP20R<5:0>:** RP20 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP20 (see Table 10-4 for peripheral function numbers).

REGISTER 10-40: RPOR11: PERIPHERAL PIN SELECT OUTPUT REGISTER 11

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP23R5	RP23R4	RP23R3	RP23R2	RP23R1	RP23R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP22R5	RP22R4	RP22R3	RP22R2	RP22R1	RP22R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as ‘0’

bit 13-8 **RP23R<5:0>:** RP23 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP23 (see Table 10-4 for peripheral function numbers).

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **RP22R<5:0>:** RP22 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP22 (see Table 10-4 for peripheral function numbers).

REGISTER 10-41: RPOR12: PERIPHERAL PIN SELECT OUTPUT REGISTER 12

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP25R5	RP25R4	RP25R3	RP25R2	RP25R1	RP25R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP24R5	RP24R4	RP24R3	RP24R2	RP24R1	RP24R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13-8 **RP25R<5:0>:** RP25 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP25 (see Table 10-4 for peripheral function numbers).

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **RP24R<5:0>:** RP24 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP24 (see Table 10-4 for peripheral function numbers).

REGISTER 10-42: RPOR13: PERIPHERAL PIN SELECT OUTPUT REGISTER 13

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP27R5	RP27R4	RP27R3	RP27R2	RP27R1	RP27R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP26R5	RP26R4	RP26R3	RP26R2	RP26R1	RP26R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13-8 **RP27R<5:0>:** RP27 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP27 (see Table 10-4 for peripheral function numbers).

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **RP26R<5:0>:** RP26 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP26 (see Table 10-4 for peripheral function numbers).

REGISTER 10-43: RPOR14: PERIPHERAL PIN SELECT OUTPUT REGISTER 14

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP29R5	RP29R4	RP29R3	RP29R2	RP29R1	RP29R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP28R5	RP28R4	RP28R3	RP28R2	RP28R1	RP28R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13-8 **RP29R<5:0>:** RP29 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP29 (see Table 10-4 for peripheral function numbers).

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **RP28R<5:0>:** RP28 Output Pin Mapping bits
Peripheral output number n is assigned to pin, RP28 (see Table 10-4 for peripheral function numbers).

REGISTER 10-44: RPOR15: PERIPHERAL PIN SELECT OUTPUT REGISTER 15(1)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP31R5	RP31R4	RP31R3	RP31R2	RP31R1	RP31R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP30R5	RP30R4	RP30R3	RP30R2	RP30R1	RP30R0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13-8 **RP31R<5:0>:** RP31 Output Pin Mapping bits(1)
Peripheral output number n is assigned to pin, RP31 (see Table 10-4 for peripheral function numbers).

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **RP30R<5:0>:** RP30 Output Pin Mapping bits(1)
Peripheral output number n is assigned to pin, RP30 (see Table 10-4 for peripheral function numbers).

Note 1: Unimplemented in 64-pin devices; read as '0'.

NOTES:

11.0 TIMER1

Note: This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the "*PIC24F Family Reference Manual*", **Section 14. "Timers" (DS39704)**. The information in this data sheet supersedes the information in the FRM.

The Timer1 module is a 16-bit timer, which can serve as the time counter for the Real-Time Clock (RTC) or operate as a free-running, interval timer/counter. Timer1 can operate in three modes:

- 16-Bit Timer
- 16-Bit Synchronous Counter
- 16-Bit Asynchronous Counter

Timer1 also supports these features:

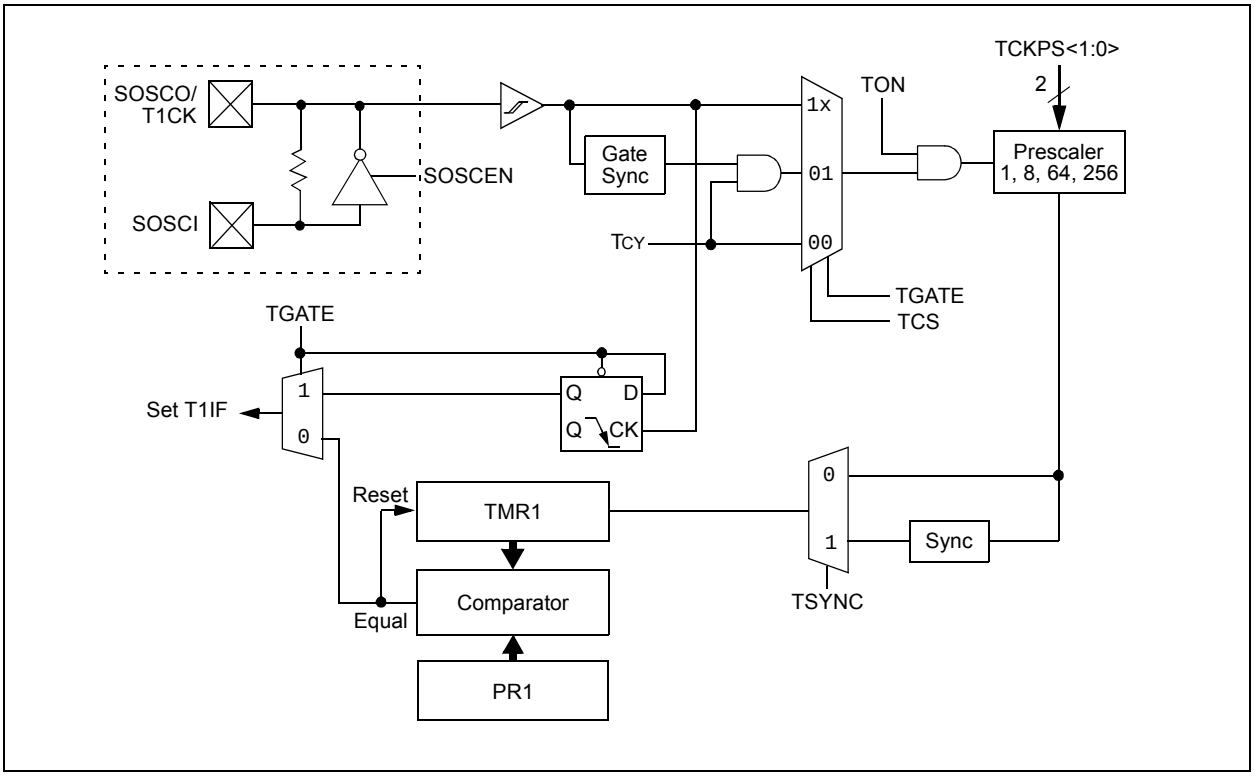
- Timer Gate Operation
- Selectable Prescaler Settings
- Timer Operation during CPU Idle and Sleep modes
- Interrupt on 16-Bit Period Register Match or Falling Edge of External Gate Signal

Figure 11-1 presents a block diagram of the 16-bit timer module.

To configure Timer1 for operation:

1. Set the TON bit (= 1).
2. Select the timer prescaler ratio using the TCKPS<1:0> bits.
3. Set the Clock and Gating modes using the TCS and TGATE bits.
4. Set or clear the TSYNC bit to configure synchronous or asynchronous operation.
5. Load the timer period value into the PR1 register.
6. If interrupts are required, set the interrupt enable bit, T1IE. Use the priority bits, T1IP<2:0>, to set the interrupt priority.

FIGURE 11-1: 16-BIT TIMER1 MODULE BLOCK DIAGRAM



REGISTER 11-1: T1CON: TIMER1 CONTROL REGISTER(1)

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON	—	TSIDL	—	—	—	—	—
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	U-0
—	TGATE	TCKPS1	TCKPS0	—	TSYNC	TCS	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **TON:** Timer1 On bit
1 = Starts 16-bit Timer1
0 = Stops 16-bit Timer1

bit 14 **Unimplemented:** Read as '0'

bit 13 **TSIDL:** Stop in Idle Mode bit
1 = Discontinue module operation when device enters Idle mode
0 = Continue module operation in Idle mode

bit 12-7 **Unimplemented:** Read as '0'

bit 6 **TGATE:** Timer1 Gated Time Accumulation Enable bit
When TCS = 1:
This bit is ignored.
When TCS = 0:
1 = Gated time accumulation enabled
0 = Gated time accumulation disabled

bit 5-4 **TCKPS<1:0>:** Timer1 Input Clock Prescale Select bits
11 = 1:256
10 = 1:64
01 = 1:8
00 = 1:1

bit 3 **Unimplemented:** Read as '0'

bit 2 **TSYNC:** Timer1 External Clock Input Synchronization Select bit
When TCS = 1:
1 = Synchronize external clock input
0 = Do not synchronize external clock input
When TCS = 0:
This bit is ignored.

bit 1 **TCS:** Timer1 Clock Source Select bit
1 = External clock from T1CK pin (on the rising edge)
0 = Internal clock (FOSC/2)

bit 0 **Unimplemented:** Read as '0'

Note 1: Changing the value of TxCON while the timer is running (TON = 1) causes the timer prescale counter to reset and is not recommended.

12.0 TIMER2/3 AND TIMER4/5

Note: This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the *"PIC24F Family Reference Manual"*, **Section 14. "Timers"** (DS39704). The information in this data sheet supersedes the information in the FRM.

The Timer2/3 and Timer4/5 modules are 32-bit timers, which can also be configured as four independent, 16-bit timers with selectable operating modes.

As 32-bit timers, Timer2/3 and Timer4/5 can each operate in three modes:

- Two independent 16-bit timers with all 16-bit operating modes (except Asynchronous Counter mode)
- Single 32-bit timer
- Single 32-bit synchronous counter

They also support these features:

- Timer Gate Operation
- Selectable Prescaler Settings
- Timer Operation during Idle and Sleep modes
- Interrupt on a 32-Bit Period Register Match
- ADC Event Trigger (only on Timer2/3 in 32-bit mode and Timer3 in 16-bit mode)

Individually, all four of the 16-bit timers can function as synchronous timers or counters. They also offer the features listed above, except for the ADC Event Trigger; this is implemented only on Timer2/3 in 32-bit mode and Timer3 in 16-bit mode. The operating modes and enabled features are determined by setting the appropriate bit(s) in the T2CON, T3CON, T4CON and T5CON registers. T2CON and T4CON are shown in generic form in Register 12-1; T3CON and T5CON are shown in Register 12-2.

For 32-bit timer/counter operation, Timer2 and Timer4 are the least significant word; Timer3 and Timer4 are the most significant word of the 32-bit timers.

Note: For 32-bit operation, T3CON and T5CON control bits are ignored. Only T2CON and T4CON control bits are used for setup and control. Timer2 and Timer4 clock and gate inputs are utilized for the 32-bit timer modules, but an interrupt is generated with the Timer3 or Timer5 interrupt flags.

To configure Timer2/3 or Timer4/5 for 32-bit operation:

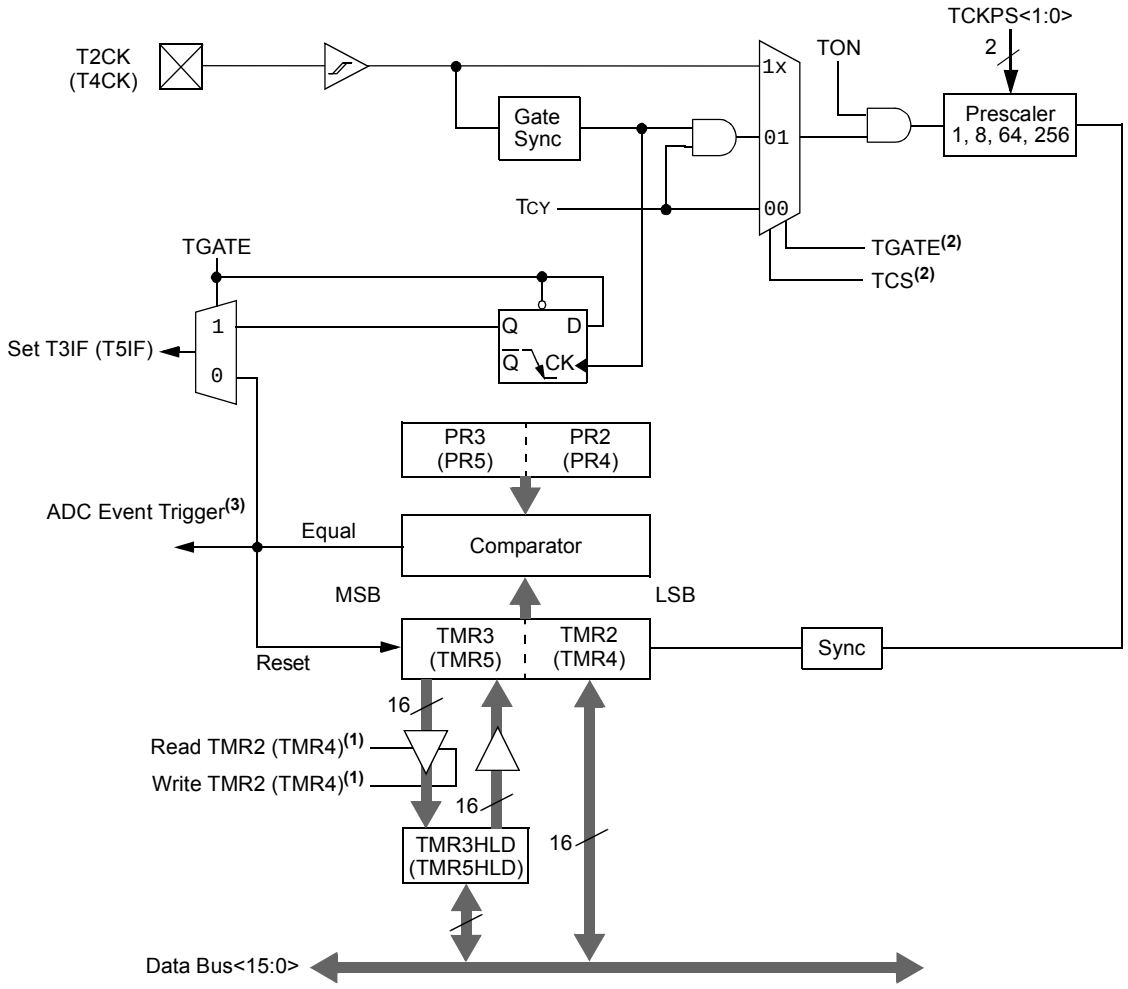
1. Set the T32 bit (T2CON<3> or T4CON<3> = 1).
2. Select the prescaler ratio for Timer2 or Timer4 using the TCKPS<1:0> bits.
3. Set the Clock and Gating modes using the TCS and TGATE bits. If TCS is set to an external clock, RPINRx (TxCK) must be configured to an available RPn/RPIn pin. For more information, see **Section 10.4 "Peripheral Pin Select (PPS)"**.
4. Load the timer period value. PR3 (or PR5) will contain the most significant word (msw) of the value while PR2 (or PR4) contains the least significant word (lsw).
5. If interrupts are required, set the interrupt enable bit, T3IE or T5IE; use the priority bits, T3IP<2:0> or T5IP<2:0>, to set the interrupt priority. Note that while Timer2 or Timer4 controls the timer, the interrupt appears as a Timer3 or Timer5 interrupt.
6. Set the TON bit (= 1).

The timer value, at any point, is stored in the register pair, TMR<3:2> (or TMR<5:4>). TMR3 (TMR5) always contains the most significant word of the count, while TMR2 (TMR4) contains the least significant word.

To configure any of the timers for individual 16-bit operation:

1. Clear the T32 bit corresponding to that timer (T2CON<3> for Timer2 and Timer3 or T4CON<3> for Timer4 and Timer5).
2. Select the timer prescaler ratio using the TCKPS<1:0> bits.
3. Set the Clock and Gating modes using the TCS and TGATE bits. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.
4. Load the timer period value into the PRx register.
5. If interrupts are required, set the interrupt enable bit, TxIE; use the priority bits, TxIP<2:0>, to set the interrupt priority.
6. Set the TON (TxCON<15> = 1) bit.

FIGURE 12-1: TIMER2/3 AND TIMER4/5 (32-BIT) BLOCK DIAGRAM

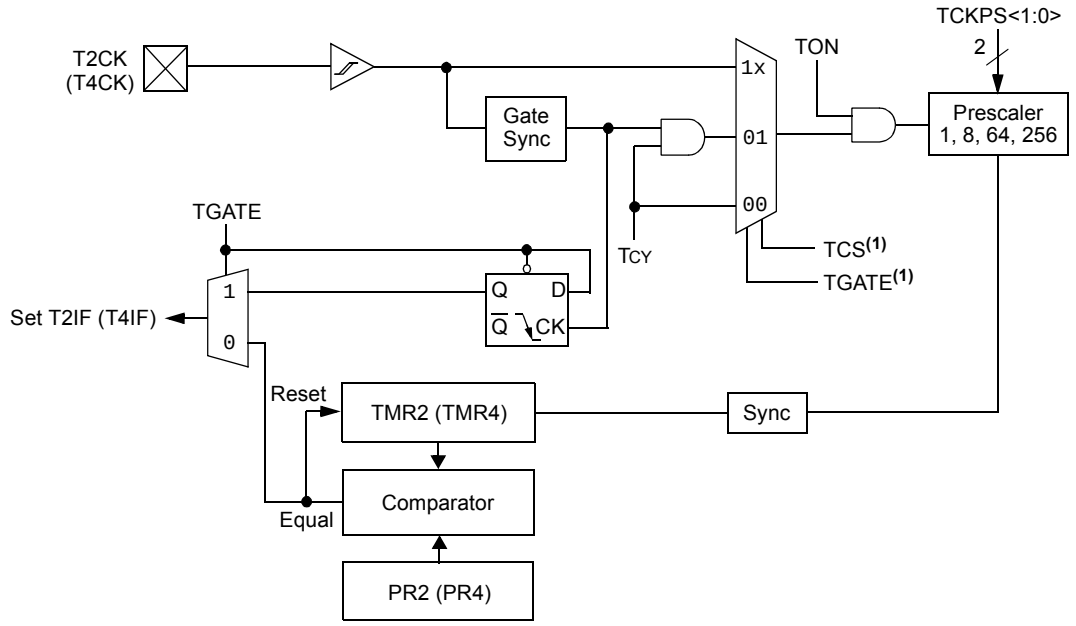


Note 1: The 32-Bit Timer Configuration bit, T32, must be set for 32-bit timer/counter operation. All control bits are respective to the T2CON and T4CON registers.

2: The timer clock input must be assigned to an available RPn/RPIn pin before use. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

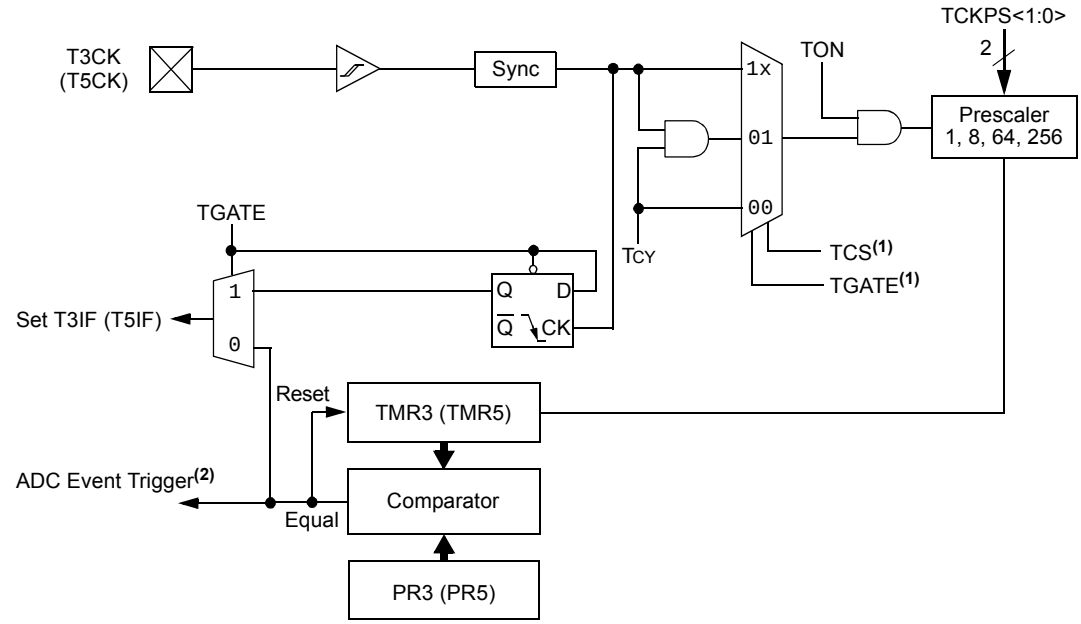
3: The ADC event trigger is available only on Timer 2/3 in 32-bit mode and Timer 3 in 16-bit mode.

FIGURE 12-2: TIMER2 AND TIMER4 (16-BIT SYNCHRONOUS) BLOCK DIAGRAM



Note 1: The timer clock input must be assigned to an available RPn/RPIn pin before use. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

FIGURE 12-3: TIMER3 AND TIMER5 (16-BIT ASYNCHRONOUS) BLOCK DIAGRAM



Note 1: The timer clock input must be assigned to an available RPn/RPIn pin before use. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

2: The ADC event trigger is available only on Timer3.

REGISTER 12-1: TxCON: TIMER2 AND TIMER4 CONTROL REGISTER(3)

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON	—	TSIDL	—	—	—	—	—
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	U-0
—	TGATE	TCKPS1	TCKPS0	T32(1)	—	TCS(2)	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **TON:** Timerx On bit
When TxCON<3> = 1:
1 = Starts 32-bit Timerx/y
0 = Stops 32-bit Timerx/y
When TxCON<3> = 0:
1 = Starts 16-bit Timerx
0 = Stops 16-bit Timerx

bit 14 **Unimplemented:** Read as '0'

bit 13 **TSIDL:** Stop in Idle Mode bit
1 = Discontinue module operation when device enters Idle mode
0 = Continue module operation in Idle mode

bit 12-7 **Unimplemented:** Read as '0'

bit 6 **TGATE:** Timerx Gated Time Accumulation Enable bit
When TCS = 1:
This bit is ignored.
When TCS = 0:
1 = Gated time accumulation enabled
0 = Gated time accumulation disabled

bit 5-4 **TCKPS<1:0>:** Timerx Input Clock Prescale Select bits
11 = 1:256
10 = 1:64
01 = 1:8
00 = 1:1

bit 3 **T32:** 32-Bit Timer Mode Select bit(1)
1 = Timerx and Timery form a single 32-bit timer
0 = Timerx and Timery act as two 16-bit timers
In 32-bit mode, T3CON control bits do not affect 32-bit timer operation.

bit 2 **Unimplemented:** Read as '0'

bit 1 **TCS:** Timerx Clock Source Select bit(2)
1 = External clock from pin, TxCK (on the rising edge)
0 = Internal clock (FOSC/2)

bit 0 **Unimplemented:** Read as '0'

Note 1: In T4CON, the T45 bit is implemented instead of T32 to select 32-bit mode. In 32-bit mode, the T3CON or T5CON control bits do not affect 32-bit timer operation.

2: If TCS = 1, RPINRx (TxCK) must be configured to an available RPn/RPIn pin. For more information, see **Section 10.4 "Peripheral Pin Select (PPS)"**.

3: Changing the value of TxCON while the timer is running (TON = 1) causes the timer prescale counter to reset and is not recommended.

REGISTER 12-2: TyCON: TIMER3 AND TIMER5 CONTROL REGISTER(3)

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON(1)	—	TSIDL(1)	—	—	—	—	—
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	U-0
—	TGATE(1)	TCKPS1(1)	TCKPS0(1)	—	—	TCS(1,2)	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **TON:** Timery On bit(1)
1 = Starts 16-bit Timery
0 = Stops 16-bit Timery

bit 14 **Unimplemented:** Read as '0'

bit 13 **TSIDL:** Stop in Idle Mode bit(1)
1 = Discontinue module operation when device enters Idle mode
0 = Continue module operation in Idle mode

bit 12-7 **Unimplemented:** Read as '0'

bit 6 **TGATE:** Timery Gated Time Accumulation Enable bit(1)
When TCS = 1:
This bit is ignored.
When TCS = 0:
1 = Gated time accumulation enabled
0 = Gated time accumulation disabled

bit 5-4 **TCKPS<1:0>:** Timery Input Clock Prescale Select bits(1)
11 = 1:256
10 = 1:64
01 = 1:8
00 = 1:1

bit 3-2 **Unimplemented:** Read as '0'

bit 1 **TCS:** Timery Clock Source Select bit(1,2)
1 = External clock from pin, TyCK (on the rising edge)
0 = Internal clock (FOSC/2)

bit 0 **Unimplemented:** Read as '0'

Note 1: When 32-bit operation is enabled (T2CON<3> or T4CON<3> = 1), these bits have no effect on Timery operation; all timer functions are set through T2CON and T4CON.

2: If TCS = 1, RPINRx (TxCK) must be configured to an available RPn/RPIn pin. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

3: Changing the value of TyCON while the timer is running (TON = 1) causes the timer prescale counter to reset and is not recommended.

NOTES:

13.0 INPUT CAPTURE WITH DEDICATED TIMERS

Note: This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the *"PIC24F Family Reference Manual"*, **Section 34. "Input Capture with Dedicated Timer"** (DS39722). The information in this data sheet supersedes the information in the FRM.

Devices in the PIC24FJ256DA210 family comprise nine independent input capture modules. Each of the modules offers a wide range of configuration and operating options for capturing external pulse events and generating interrupts.

Key features of the input capture module include:

- Hardware configurable for 32-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with up to 30 user-selectable sync/trigger sources available
- A 4-level FIFO buffer for capturing and holding timer values for several events
- Configurable interrupt generation
- Up to 6 clock sources available for each module, driving a separate internal 16-bit counter

The module is controlled through two registers: ICxCON1 (Register 13-1) and ICxCON2 (Register 13-2). A general block diagram of the module is shown in Figure 13-1.

13.1 General Operating Modes

13.1.1 SYNCHRONOUS AND TRIGGER MODES

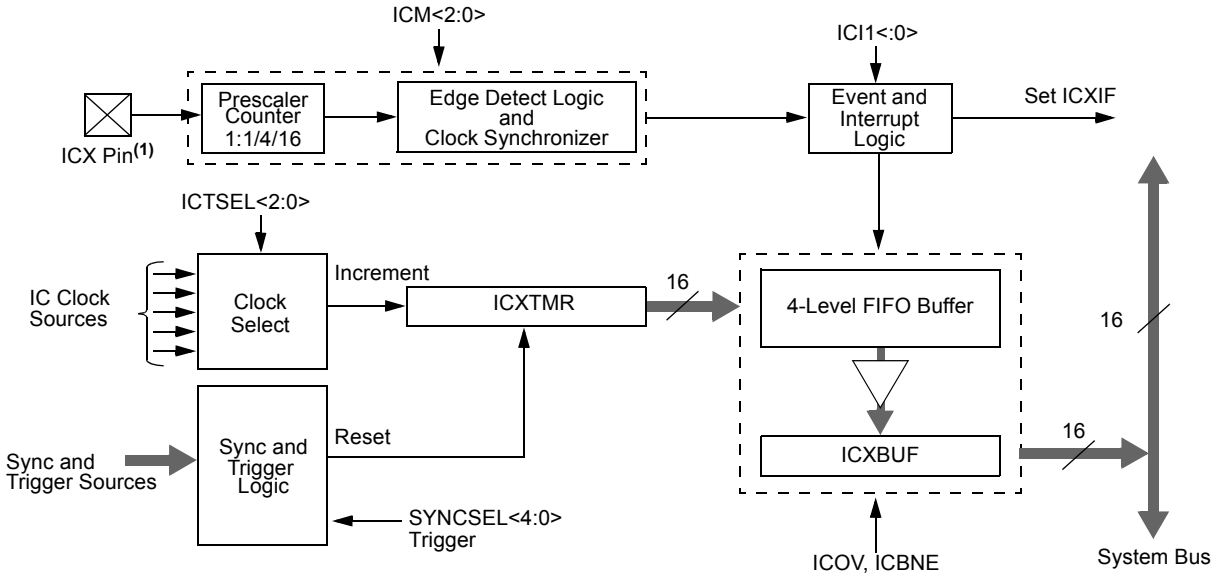
When the input capture module operates in a free-running mode, the internal 16-bit counter, ICxTMR, counts up continuously, wrapping around from FFFFh to 0000h on each overflow, with its period synchronized to the selected external clock source. When a capture event occurs, the current 16-bit value of the internal counter is written to the FIFO buffer.

In Synchronous mode, the module begins capturing events on the ICx pin as soon as its selected clock source is enabled. Whenever an event occurs on the selected sync source, the internal counter is reset. In Trigger mode, the module waits for a Sync event from another internal module to occur before allowing the internal counter to run.

Standard, free-running operation is selected by setting the SYNCSEL bits (ICxCON2<4:0>) to '00000' and clearing the ICTRIG bit (ICxCON2<7>). Synchronous and Trigger modes are selected any time the SYNCSEL bits are set to any value except '00000'. The ICTRIG bit selects either Synchronous or Trigger mode; setting the bit selects Trigger mode operation. In both modes, the SYNCSEL bits determine the sync/trigger source.

When the SYNCSEL bits are set to '00000' and ICTRIG is set, the module operates in Software Trigger mode. In this case, capture operations are started by manually setting the TRIGSTAT bit (ICxCON2<6>).

FIGURE 13-1: INPUT CAPTURE BLOCK DIAGRAM



Note 1: The ICx inputs must be assigned to an available RPn/RPIn pin before use. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

13.1.2 CASCADED (32-BIT) MODE

By default, each module operates independently with its own 16-bit timer. To increase resolution, adjacent even and odd modules can be configured to function as a single 32-bit module. (For example, Modules 1 and 2 are paired, as are Modules 3 and 4, and so on.) The odd numbered module (ICx) provides the Least Significant 16 bits of the 32-bit register pairs and the even module (ICy) provides the Most Significant 16 bits. Wrap-arounds of the ICx registers cause an increment of their corresponding ICy registers.

Cascaded operation is configured in hardware by setting the IC32 bits (ICxCON2<8>) for both modules.

13.2 Capture Operations

The input capture module can be configured to capture timer values and generate interrupts on rising edges on ICx or all transitions on ICx. Captures can be configured to occur on all rising edges or just some (every 4th or 16th). Interrupts can be independently configured to generate on each event or a subset of events.

To set up the module for capture operations:

1. Configure the ICx input for one of the available Peripheral Pin Select pins.
2. If Synchronous mode is to be used, disable the sync source before proceeding.
3. Make sure that any previous data has been removed from the FIFO by reading ICxBUF until the ICBNE bit (ICxCON1<3>) is cleared.
4. Set the SYNCSEL bits (ICxCON2<4:0>) to the desired sync/trigger source.
5. Set the ICTSEL bits (ICxCON1<12:10>) for the desired clock source.
6. Set the ICI bits (ICxCON1<6:5>) to the desired interrupt frequency
7. Select Synchronous or Trigger mode operation:
   a) Check that the SYNCSEL bits are not set to '00000'.
   b) For Synchronous mode, clear the ICTRIG bit (ICxCON2<7>).
   c) For Trigger mode, set ICTRIG, and clear the TRIGSTAT bit (ICxCON2<6>).
8. Set the ICM bits (ICxCON1<2:0>) to the desired operational mode.
9. Enable the selected sync/trigger source.

For 32-bit cascaded operations, the setup procedure is slightly different:

1. Set the IC32 bits for both modules (ICyCON2<8>) and (ICxCON2<8>), enabling the even numbered module first. This ensures the modules will start functioning in unison.
2. Set the ICTSEL and SYNCSEL bits for both modules to select the same sync/trigger and time base source. Set the even module first, then the odd module. Both modules must use the same ICTSEL and SYNCSEL settings.
3. Clear the ICTRIG bit of the even module (ICyCON2<7>). This forces the module to run in Synchronous mode with the odd module, regardless of its trigger setting.
4. Use the odd module's ICI bits (ICxCON1<6:5>) to set the desired interrupt frequency.
5. Use the ICTRIG bit of the odd module (ICxCON2<7>) to configure Trigger or Synchronous mode operation.

Note:	For Synchronous mode operation, enable the sync source as the last step. Both input capture modules are held in Reset until the sync source is enabled.

6. Use the ICM bits of the odd module (ICxCON1<2:0>) to set the desired capture mode.

The module is ready to capture events when the time base and the sync/trigger source are enabled. When the ICBNE bit (ICxCON1<3>) becomes set, at least one capture value is available in the FIFO. Read input capture values from the FIFO until the ICBNE clears to '0'.

For 32-bit operation, read both the ICxBUF and ICyBUF for the full 32-bit timer value (ICxBUF for the lsw, ICyBUF for the msw). At least one capture value is available in the FIFO buffer when the odd module's ICBNE bit (ICxCON1<3>) becomes set. Continue to read the buffer registers until ICBNE is cleared (performed automatically by hardware).

REGISTER 13-1: ICxCON1: INPUT CAPTURE x CONTROL REGISTER 1

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—
bit 15							bit 8

U-0	R/W-0	R/W-0	R-0, HSC	R-0, HSC	R/W-0	R/W-0	R/W-0
—	ICI1	ICI0	ICOV	ICBNE	ICM2(1)	ICM1(1)	ICM0(1)
bit 7							bit 0

Legend:	HSC = Hardware Settable/Clearable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13 **ICSIDL:** Input Capture x Module Stop in Idle Control bit
1 = Input capture module halts in CPU Idle mode
0 = Input capture module continues to operate in CPU Idle mode

bit 12-10 **ICTSEL<2:0>:** Input Capture Timer Select bits
111 = System clock (FOSC/2)
110 = Reserved
101 = Reserved
100 = Timer1
011 = Timer5
010 = Timer4
001 = Timer2
000 = Timer3

bit 9-7 **Unimplemented:** Read as '0'

bit 6-5 **ICI<1:0>:** Select Number of Captures Per Interrupt bits
11 = Interrupt on every fourth capture event
10 = Interrupt on every third capture event
01 = Interrupt on every second capture event
00 = Interrupt on every capture event

bit 4 **ICOV:** Input Capture x Overflow Status Flag bit (read-only)
1 = Input capture overflow occurred
0 = No input capture overflow occurred

bit 3 **ICBNE:** Input Capture x Buffer Empty Status bit (read-only)
1 = Input capture buffer is not empty, at least one more capture value can be read
0 = Input capture buffer is empty

bit 2-0 **ICM<2:0>:** Input Capture Mode Select bits(1)
111 = Interrupt mode: input capture functions as an interrupt pin only when the device is in Sleep or Idle mode (rising edge detect only, all other control bits are not applicable)
110 = Unused (module disabled)
101 = Prescaler Capture mode: capture on every 16th rising edge
100 = Prescaler Capture mode: capture on every 4th rising edge
011 = Simple Capture mode: capture on every rising edge
010 = Simple Capture mode: capture on every falling edge
001 = Edge Detect Capture mode: capture on every edge (rising and falling), ICI<1:0> bits do not control interrupt generation for this mode
000 = Input capture module turned off

Note 1: The ICx input must also be configured to an available RPn/RPIn pin. For more information, see **Section 10.4 "Peripheral Pin Select (PPS)"**.

REGISTER 13-2: ICxCON2: INPUT CAPTURE x CONTROL REGISTER 2

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	IC32
bit 15							bit 8

R/W-0	R/W-0 HS	U-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-1
ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0
bit 7							bit 0

Legend:	HS = Hardware Settable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-9 **Unimplemented:** Read as '0'

bit 8 **IC32:** Cascade Two IC Modules Enable bit (32-bit operation)
1 = ICx and ICy operate in cascade as a 32-bit module (this bit must be set in both modules)
0 = ICx functions independently as a 16-bit module

bit 7 **ICTRIG:** ICx Sync/Trigger Select bit
1 = Trigger ICx from the source designated by the SYNCSELx bits
0 = Synchronize ICx with the source designated by the SYNCSELx bits

bit 6 **TRIGSTAT:** Timer Trigger Status bit
1 = Timer source has been triggered and is running (set in hardware, can be set in software)
0 = Timer source has not been triggered and is being held clear

bit 5 **Unimplemented:** Read as '0'

bit 4-0 **SYNCSEL<4:0>:** Synchronization/Trigger Source Selection bits
11111 = Reserved
11110 = Input Capture 9(2)
11101 = Input Capture 6(2)
11100 = CTMU(1)
11011 = A/D(1)
11010 = Comparator 3(1)
11001 = Comparator 2(1)
11000 = Comparator 1(1)
10111 = Input Capture 4(2)
10110 = Input Capture 3(2)
10101 = Input Capture 2(2)
10100 = Input Capture 1(2)
10011 = Input Capture 8(2)
10010 = Input Capture 7(2)
1000x = Reserved
01111 = Timer5
01110 = Timer4
01101 = Timer3
01100 = Timer2
01011 = Timer1
01010 = Input Capture 5(2)
01001 = Output Compare 9
.
.
.
00010 = Output Compare 2
00001 = Output Compare 1
00000 = Not synchronized to any other module

Note 1: Use these inputs as trigger sources only and never as sync sources.
2: Never use an IC module as its own trigger source, by selecting this mode.

14.0 OUTPUT COMPARE WITH DEDICATED TIMERS

Note:	This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the *"PIC24F Family Reference Manual"*, **Section 35. "Output Compare with Dedicated Timer"** (DS39723). The information in this data sheet supersedes the information in the FRM.

Devices in the PIC24FJ256DA210 family feature all of the 9 independent output compare modules. Each of these modules offers a wide range of configuration and operating options for generating pulse trains on internal device events, and can produce pulse-width modulated waveforms for driving power applications.

Key features of the output compare module include:

- Hardware configurable for 32-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with up to 31 user-selectable trigger/sync sources available
- Two separate period registers (a main register, OCxR, and a secondary register, OCxRS) for greater flexibility in generating pulses of varying widths
- Configurable for single pulse or continuous pulse generation on an output event, or continuous PWM waveform generation
- Up to 6 clock sources available for each module, driving a separate internal 16-bit counter

14.1 General Operating Modes

14.1.1 SYNCHRONOUS AND TRIGGER MODES

When the output compare module operates in a free-running mode, the internal 16-bit counter, OCxTMR, runs counts up continuously, wrapping around from 0xFFFF to 0x0000 on each overflow, with its period synchronized to the selected external clock source. Compare or PWM events are generated each time a match between the internal counter and one of the period registers occurs.

In Synchronous mode, the module begins performing its compare or PWM operation as soon as its selected clock source is enabled. Whenever an event occurs on the selected sync source, the module's internal counter is reset. In Trigger mode, the module waits for a sync event from another internal module to occur before allowing the counter to run.

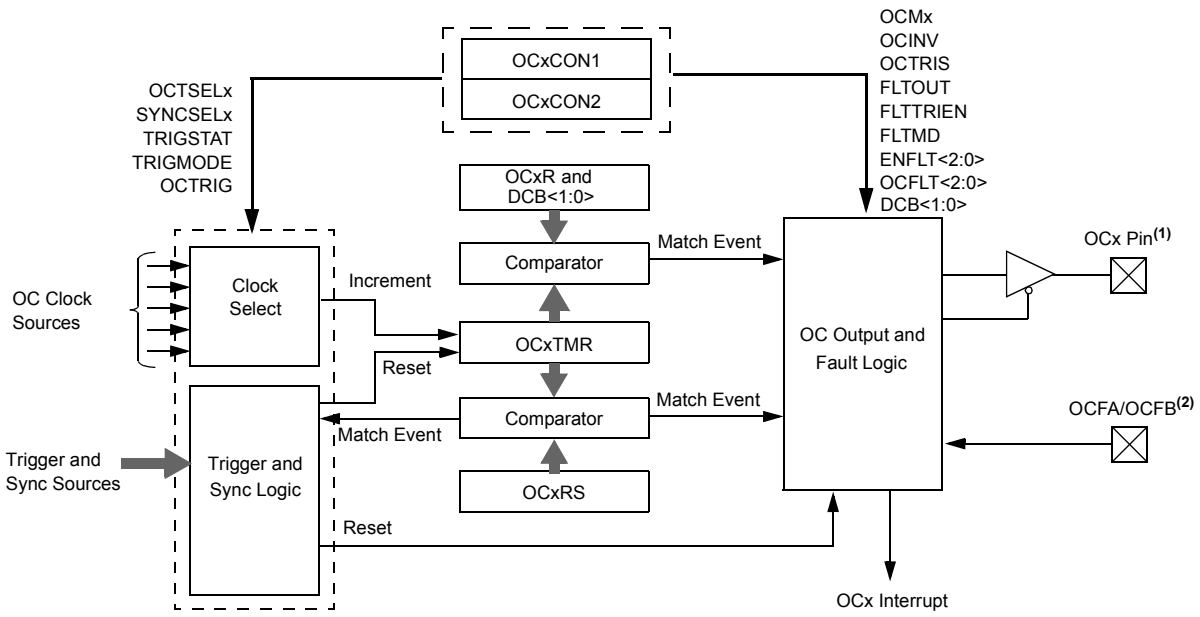
Free-running mode is selected by default or any time that the SYNCSEL bits (OCxCON2<4:0>) are set to '00000'. Synchronous or Trigger modes are selected any time the SYNCSEL bits are set to any value except '00000'. The OCTRIG bit (OCxCON2<7>) selects either Synchronous or Trigger mode; setting the bit selects Trigger mode operation. In both modes, the SYNCSEL bits determine the sync/trigger source.

14.1.2 CASCADED (32-BIT) MODE

By default, each module operates independently with its own set of 16-bit timer and duty cycle registers. To increase resolution, adjacent even and odd modules can be configured to function as a single 32-bit module. (For example, Modules 1 and 2 are paired, as are Modules 3 and 4, and so on.) The odd numbered module (OCx) provides the Least Significant 16 bits of the 32-bit register pairs and the even module (OCy) provides the Most Significant 16 bits. Wrap-arounds of the OCx registers cause an increment of their corresponding OCy registers.

Cascaded operation is configured in hardware by setting the OC32 bit (OCxCON2<8>) for both modules. For more details on cascading, refer to the "*PIC24F Family Reference Manual*", **Section 35. "Output Compare with Dedicated Timer"**.

FIGURE 14-1: OUTPUT COMPARE BLOCK DIAGRAM (16-BIT MODE)



Note 1: The OCx outputs must be assigned to an available RPn pin before use. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

2: The OCFA/OCFB fault inputs must be assigned to an available RPn/RPIn pin before use. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

14.2 Compare Operations

In Compare mode (Figure 14-1), the output compare module can be configured for single-shot or continuous pulse generation. It can also repeatedly toggle an output pin on each timer event.

To set up the module for compare operations:

1. Configure the OCx output for one of the available Peripheral Pin Select pins.
2. Calculate the required values for the OCxR and (for Double Compare modes) OCxRS Duty Cycle registers:
   a) Determine the instruction clock cycle time. Take into account the frequency of the external clock to the timer source (if one is used) and the timer prescaler settings.
   b) Calculate time to the rising edge of the output pulse relative to the timer start value (0000h).
   c) Calculate the time to the falling edge of the pulse based on the desired pulse width and the time to the rising edge of the pulse.
3. Write the rising edge value to OCxR and the falling edge value to OCxRS.
4. Set the Timer Period register, PRy, to a value equal to or greater than the value in OCxRS.
5. Set the OCM<2:0> bits for the appropriate compare operation (= 0xx).
6. For Trigger mode operations, set OCTRIG to enable Trigger mode. Set or clear TRIGMODE to configure trigger operation and TRIGSTAT to select a hardware or software trigger. For Synchronous mode, clear OCTRIG.
7. Set the SYNCSEL<4:0> bits to configure the trigger or synchronization source. If free-running timer operation is required, set the SYNCSEL bits to '00000' (no sync/trigger source).
8. Select the time base source with the OCTSEL<2:0> bits. If necessary, set the TON bits for the selected timer, which enables the compare time base to count. Synchronous mode operation starts as soon as the time base is enabled; Trigger mode operation starts after a trigger source event occurs.

For 32-bit cascaded operation, these steps are also necessary:

1. Set the OC32 bits for both registers (OCyCON2<8>) and (OCxCON2<8>). Enable the even numbered module first to ensure the modules will start functioning in unison.
2. Clear the OCTRIG bit of the even module (OCyCON2), so the module will run in Synchronous mode.
3. Configure the desired output and Fault settings for OCy.
4. Force the output pin for OCx to the output state by clearing the OCTRIS bit.
5. If Trigger mode operation is required, configure the trigger options in OCx by using the OCTRIG (OCxCON2<7>), TRIGMODE (OCxCON1<3>) and SYNCSEL (OCxCON2<4:0>) bits.
6. Configure the desired Compare or PWM mode of operation (OCM<2:0>) for OCy first, then for OCx.

Depending on the output mode selected, the module holds the OCx pin in its default state and forces a transition to the opposite state when OCxR matches the timer. In Double Compare modes, OCx is forced back to its default state when a match with OCxRS occurs. The OCxIF interrupt flag is set after an OCxR match in Single Compare modes and after each OCxRS match in Double Compare modes.

Single-shot pulse events only occur once, but may be repeated by simply rewriting the value of the OCxCON1 register. Continuous pulse events continue indefinitely until terminated.

14.3 Pulse-Width Modulation (PWM) Mode

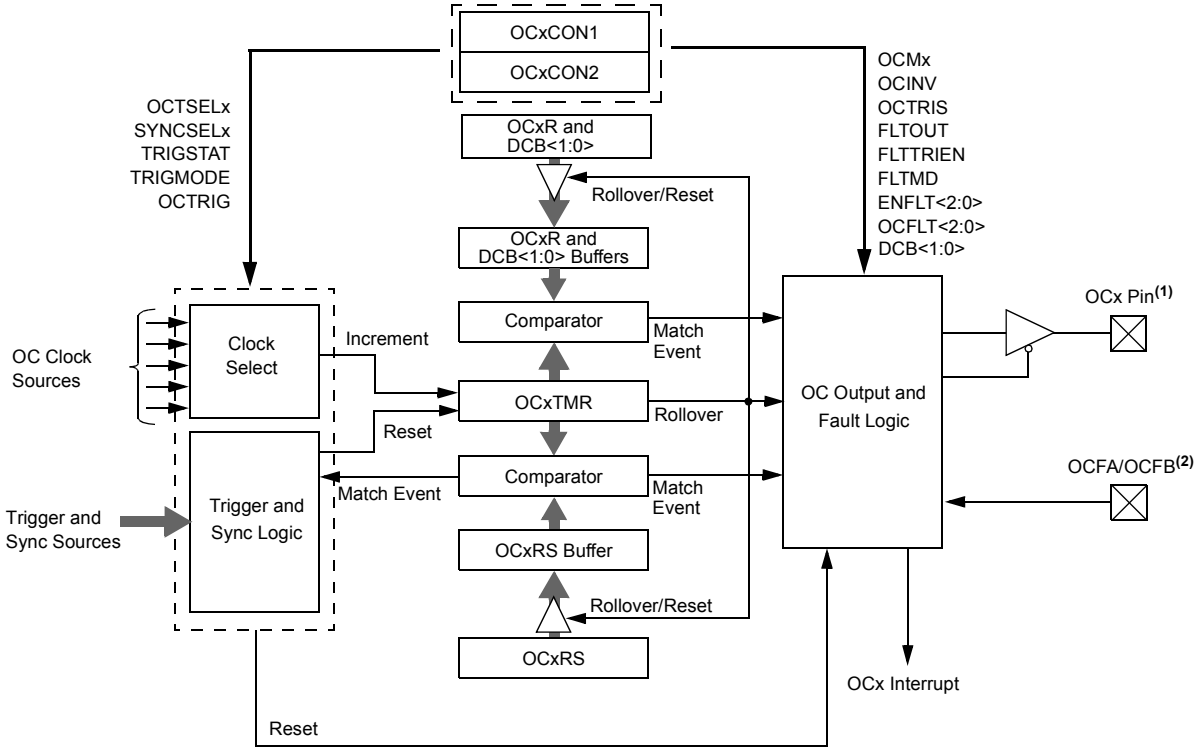
In PWM mode, the output compare module can be configured for edge-aligned or center-aligned pulse waveform generation. All PWM operations are double-buffered (buffer registers are internal to the module and are not mapped into SFR space).

To configure the output compare module for PWM operation:

1. Configure the OCx output for one of the available Peripheral Pin Select pins.
2. Calculate the desired duty cycles and load them into the OCxR register.
3. Calculate the desired period and load it into the OCxRS register.
4. Select the current OCx as the synchronization source by writing 0x1F to the SYNCSEL<4:0> bits (OCxCON2<4:0>) and '0' to the OCTRIG bit (OCxCON2<7>).
5. Select a clock source by writing to the OCTSEL<2:0> bits (OCxCON<12:10>).
6. Enable interrupts, if required, for the timer and output compare modules. The output compare interrupt is required for PWM Fault pin utilization.
7. Select the desired PWM mode in the OCM<2:0> bits (OCxCON1<2:0>).
8. Appropriate Fault inputs may be enabled by using the ENFLT<2:0> bits as described in Register 14-1.
9. If a timer is selected as a clock source, set the selected timer prescale value. The selected timer's prescaler output is used as the clock input for the OCx timer, and not the selected timer output.

Note:	This peripheral contains input and output functions that may need to be configured by the Peripheral Pin Select. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

FIGURE 14-2: OUTPUT COMPARE BLOCK DIAGRAM (DOUBLE-BUFFERED, 16-BIT PWM MODE)



Note 1: The OCx outputs must be assigned to an available RPn pin before use. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

2: The OCFA/OCFB fault inputs must be assigned to an available RPn/RPIn pin before use. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

14.3.1 PWM PERIOD

The PWM period is specified by writing to PRy, the Timer Period register. The PWM period can be calculated using Equation 14-1.

EQUATION 14-1: CALCULATING THE PWM PERIOD(1)

$$\text{PWM Period} = [(\text{PRy}) + 1 \cdot T_{CY} \cdot (\text{Timer Prescale Value})$$

where:

$$\text{PWM Frequency} = 1/[\text{PWM Period}]$$

Note 1: Based on $T_{CY} = T_{OSC} * 2$; Doze mode and PLL are disabled.

Note: A PRy value of N will produce a PWM period of N + 1 time base count cycles. For example, a value of 7 written into the PRy register will yield a period consisting of 8 time base cycles.

14.3.2 PWM DUTY CYCLE

The PWM duty cycle is specified by writing to the OCxRS and OCxR registers. The OCxRS and OCxR registers can be written to at any time, but the duty cycle value is not latched until a match between PRy and TMRy occurs (i.e., the period is complete). This provides a double buffer for the PWM duty cycle and is essential for glitchless PWM operation.

Some important boundary parameters of the PWM duty cycle include:

- If OCxR, OCxRS, and PRy are all loaded with 0000h, the OCx pin will remain low (0% duty cycle).
- If OCxRS is greater than PRy, the pin will remain high (100% duty cycle).

See Example 14-1 for PWM mode timing details. Table 14-1 and Table 14-2 show example PWM frequencies and resolutions for a device operating at 4 MIPS and 10 MIPS, respectively.

EQUATION 14-2: CALCULATION FOR MAXIMUM PWM RESOLUTION(1)

$$\text{Maximum PWM Resolution (bits)} = \frac{\log_{10}\left(\frac{FCY}{FPWM \bullet (\text{Timer Prescale Value})}\right)}{\log_{10}(2)} \text{ bits}$$

Note 1: Based on FCY = FOSC/2; Doze mode and PLL are disabled.

EXAMPLE 14-1: PWM PERIOD AND DUTY CYCLE CALCULATIONS(1)

1. Find the Timer Period register value for a desired PWM frequency of 52.08 kHz, where FOSC = 8 MHz with PLL (32 MHz device clock rate) and a Timer2 prescaler setting of 1:1.

   TCY = 2 * TOSC = 62.5 ns

   PWM Period = 1/PWM Frequency = 1/52.08 kHz = 19.2 ms

   PWM Period = (PR2 + 1) • TCY • (Timer2 Prescale Value)

   19.2 ms = PR2 + 1) • 62.5 ns • 1

   PR2 = 306

2. Find the maximum resolution of the duty cycle that can be used with a 52.08 kHz frequency and a 32 MHz device clock rate:

   PWM Resolution = $\log_{10}(FCY/FPWM)/\log_{10}2)$ bits

   = $(\log_{10}(16\text{ MHz}/52.08\text{ kHz})/\log_{10}2)$ bits

   = 8.3 bits

Note 1: Based on TCY = 2 * TOSC; Doze mode and PLL are disabled.

TABLE 14-1: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 4 MIPS (FCY = 4 MHz)(1)

PWM Frequency	7.6 Hz	61 Hz	122 Hz	977 Hz	3.9 kHz	31.3 kHz	125 kHz
Timer Prescaler Ratio	8	1	1	1	1	1	1
Period Register Value	FFFFh	FFFFh	7FFFh	0FFFh	03FFh	007Fh	001Fh
Resolution (bits)	16	16	15	12	10	7	5

Note 1: Based on FCY = FOSC/2; Doze mode and PLL are disabled.

TABLE 14-2: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 16 MIPS (FCY = 16 MHz)(1)

PWM Frequency	30.5 Hz	244 Hz	488 Hz	3.9 kHz	15.6 kHz	125 kHz	500 kHz
Timer Prescaler Ratio	8	1	1	1	1	1	1
Period Register Value	FFFFh	FFFFh	7FFFh	0FFFh	03FFh	007Fh	001Fh
Resolution (bits)	16	16	15	12	10	7	5

Note 1: Based on FCY = FOSC/2; Doze mode and PLL are disabled.

REGISTER 14-1: OCxCON1: OUTPUT COMPARE x CONTROL REGISTER 1

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	ENFLT2(2)	ENFLT1(2)
bit 15							bit 8

R/W-0	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0	R/W-0	R/W-0	R/W-0
ENFLT0(2)	OCFLT2(2)	OCFLT1(2)	OCFLT0(2)	TRIGMODE	OCM2(1)	OCM1(1)	OCM0(1)
bit 7							bit 0

Legend:	HSC = Hardware Settable/Clearable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13 **OCSIDL:** Stop Output Compare x in Idle Mode Control bit
1 = Output Compare x halts in CPU Idle mode
0 = Output Compare x continues to operate in CPU Idle mode

bit 12-10 **OCTSEL<2:0>:** Output Compare x Timer Select bits
111 = Peripheral clock (FCY)
110 = Reserved
101 = Reserved
100 = Timer1 clock (only synchronous clock is supported)
011 = Timer5 clock
010 = Timer4 clock
001 = Timer3 clock
000 = Timer2 clock

bit 9 **ENFLT2:** Fault Input 2 Enable bit(2)
1 = Fault 2 (Comparator 1/2/3 out) is enabled(3)
0 = Fault 2 is disabled

bit 8 **ENFLT1:** Fault Input 1 Enable bit(2)
1 = Fault 1 (OCFB pin) is enabled(4)
0 = Fault 1 is disabled

bit 7 **ENFLT0:** Fault Input 0 Enable bit(2)
1 = Fault 0 (OCFA pin) is enabled(4)
0 = Fault 0 is disabled

bit 6 **OCFLT2:** PWM Fault 2 (Comparator 1/2/3) Condition Status bit(2,3)
1 = PWM Fault 2 has occurred
0 = No PWM Fault 2 has occurred

bit 5 **OCFLT1:** PWM Fault 1 (OCFB pin) Condition Status bit(2,4)
1 = PWM Fault 1 has occurred
0 = No PWM Fault 1 has occurred

bit 4 **OCFLT0:** PWM Fault 0 (OCFA pin) Condition Status bit(2,4)
1 = PWM Fault 0 has occurred
0 = No PWM Fault 0 has occurred

Note 1: The OCx output must also be configured to an available RPn pin. For more information, see **Section 10.4 "Peripheral Pin Select (PPS)"**.

2: The Fault input enable and Fault status bits are valid when OCM<2:0> = 111 or 110.

3: The Comparator 1 output controls the OC1-OC3 channels; Comparator 2 output controls the OC4-OC6 channels; Comparator 3 output controls the OC7-OC9 channels.

4: The OCFA/OCFB Fault input must also be configured to an available RPn/RPIn pin. For more information, see **Section 10.4 "Peripheral Pin Select (PPS)"**.

REGISTER 14-1: OCxCON1: OUTPUT COMPARE x CONTROL REGISTER 1 (CONTINUED)

bit 3 **TRIGMODE:** Trigger Status Mode Select bit
1 = TRIGSTAT (OCxCON2<6>) is cleared when OCxRS = OCxTMR or in software
0 = TRIGSTAT is only cleared by software

bit 2-0 **OCM<2:0>:** Output Compare x Mode Select bits(1)
111 = Center-Aligned PWM mode on OCx(2)
110 = Edge-Aligned PWM Mode on OCx(2)
101 = Double Compare Continuous Pulse mode: Initialize the OCx pin low, the toggle OCx state continuously on alternate matches of OCxR and OCxRS
100 = Double Compare Single-Shot mode: Initialize the OCx pin low, toggle the OCx state on matches of OCxR and OCxRS for one cycle
011 = Single Compare Continuous Pulse mode: Compare events continuously toggle the OCx pin
010 = Single Compare Single-Shot mode: Initialize OCx pin high, compare event forces the OCx pin low
001 = Single Compare Single-Shot mode: Initialize OCx pin low, compare event forces the OCx pin high
000 = Output compare channel is disabled

Note 1: The OCx output must also be configured to an available RPn pin. For more information, see **Section 10.4 "Peripheral Pin Select (PPS)"**.

2: The Fault input enable and Fault status bits are valid when OCM<2:0> = 111 or 110.

3: The Comparator 1 output controls the OC1-OC3 channels; Comparator 2 output controls the OC4-OC6 channels; Comparator 3 output controls the OC7-OC9 channels.

4: The OCFA/OCFB Fault input must also be configured to an available RPn/RPIn pin. For more information, see **Section 10.4 "Peripheral Pin Select (PPS)"**.

REGISTER 14-2: OCxCON2: OUTPUT COMPARE x CONTROL REGISTER 2

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
FLTMD	FLTOUT	FLTTRIEN	OCINV	—	DCB1(3)	DCB0(3)	OC32
bit 15							bit 8

R/W-0	R/W-0 HS	R/W-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0
OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0
bit 7							bit 0

Legend:	HS = Hardware Settable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **FLTMD:** Fault Mode Select bit
- 1 = Fault mode is maintained until the Fault source is removed and the corresponding OCFLT0 bit is cleared in software
- 0 = Fault mode is maintained until the Fault source is removed and a new PWM period starts

bit 14 **FLTOUT:** Fault Out bit
- 1 = PWM output is driven high on a Fault
- 0 = PWM output is driven low on a Fault

bit 13 **FLTTRIEN:** Fault Output State Select bit
- 1 = Pin is forced to an output on a Fault condition
- 0 = Pin I/O condition is unaffected by a Fault

bit 12 **OCINV:** OCMP Invert bit
- 1 = OCx output is inverted
- 0 = OCx output is not inverted

bit 11 **Unimplemented:** Read as '0'

bit 10-9 **DCB<11:0>:** PWM Duty Cycle Least Significant bits(3)
- 11 = Delay OCx falling edge by ¾ of the instruction cycle
- 10 = Delay OCx falling edge by ½ of the instruction cycle
- 01 = Delay OCx falling edge by ¼ of the instruction cycle
- 00 = OCx falling edge occurs at the start of the instruction cycle

bit 8 **OC32:** Cascade Two OC Modules Enable bit (32-bit operation)
- 1 = Cascade module operation enabled
- 0 = Cascade module operation disabled

bit 7 **OCTRIG:** OCx Trigger/Sync Select bit
- 1 = Trigger OCx from the source designated by the SYNCSELx bits
- 0 = Synchronize OCx with the source designated by the SYNCSELx bits

bit 6 **TRIGSTAT:** Timer Trigger Status bit
- 1 = Timer source has been triggered and is running
- 0 = Timer source has not been triggered and is being held clear

bit 5 **OCTRIS:** OCx Output Pin Direction Select bit
- 1 = OCx pin is tri-stated
- 0 = Output compare peripheral x is connected to an OCx pin

Note 1: Never use an OC module as its own trigger source, either by selecting this mode or another equivalent SYNCSEL setting.

2: Use these inputs as trigger sources only and never as sync sources.

3: The DCB<1:0> bits are double-buffered in the PWM modes only (OCM<2:0> (OCxCON1<2:0>) = 111, 110).

REGISTER 14-2: OCxCON2: OUTPUT COMPARE x CONTROL REGISTER 2 (CONTINUED)

bit 4-0 **SYNCSEL<4:0>:** Trigger/Synchronization Source Selection bits

11111 = This OC module(1)
11110 = Input Capture 9(2)
11101 = Input Capture 6(2)
11100 = CTMU(2)
11011 = A/D(2)
11010 = Comparator 3(2)
11001 = Comparator 2(2)
11000 = Comparator 1(2)
10111 = Input Capture 4(2)
10110 = Input Capture 3(2)
10101 = Input Capture 2(2)
10100 = Input Capture 1(2)
10011 = Input Capture 8(2)
10010 = Input Capture 7(2)
1000x = Reserved
01111 = Timer5
01110 = Timer4
01101 = Timer3
01100 = Timer2
01011 = Timer1
01010 = Input Capture 5(2)
01001 = Output Compare 9(1)
01000 = Output Compare 8(1)
00111 = Output Compare 7(1)
00110 = Output Compare 6(1)
00101 = Output Compare 5(1)
00100 = Output Compare 4(1)
00011 = Output Compare 3(1)
00010 = Output Compare 2(1)
00001 = Output Compare 1(1)
00000 = Not synchronized to any other module

Note 1: Never use an OC module as its own trigger source, either by selecting this mode or another equivalent SYNCSEL setting.

2: Use these inputs as trigger sources only and never as sync sources.

3: The DCB<1:0> bits are double-buffered in the PWM modes only (OCM<2:0> (OCxCON1<2:0>) = 111, 110).

NOTES:

15.0 SERIAL PERIPHERAL INTERFACE (SPI)

Note:	This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the *"PIC24F Family Reference Manual"*, **Section 23. "Serial Peripheral Interface (SPI)"** (DS39699). The information in this data sheet supersedes the information in the FRM.

The Serial Peripheral Interface (SPI) module is a synchronous serial interface useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be serial EEPROMs, shift registers, display drivers, A/D Converters, etc. The SPI module is compatible with the SPI and SIOP Motorola® interfaces. All devices of the PIC24FJ256DA210 family include three SPI modules.

The module supports operation in two buffer modes. In Standard mode, data is shifted through a single serial buffer. In Enhanced Buffer mode, data is shifted through an 8-level FIFO buffer.

Note:	Do not perform read-modify-write operations (such as bit-oriented instructions) on the SPIxBUF register in either Standard or Enhanced Buffer mode.

The module also supports a basic framed SPI protocol while operating in either Master or Slave mode. A total of four framed SPI configurations are supported.

The SPI serial interface consists of four pins:

- SDIx: Serial Data Input
- SDOx: Serial Data Output
- SCKx: Shift Clock Input or Output
- $\overline{SS}$x: Active-Low Slave Select or Frame Synchronization I/O Pulse

The SPI module can be configured to operate using 2, 3 or 4 pins. In the 3-pin mode, $\overline{SS}$x is not used. In the 2-pin mode, both SDOx and $\overline{SS}$x are not used.

Block diagrams of the module in Standard and Enhanced modes are shown in Figure 15-1 and Figure 15-2.

Note:	In this section, the SPI modules are referred to together as SPIx or separately as SPI1, SPI2 or SPI3. Special Function Registers will follow a similar notation. For example, SPIxCON1 and SPIxCON2 refer to the control registers for any of the 3 SPI modules.

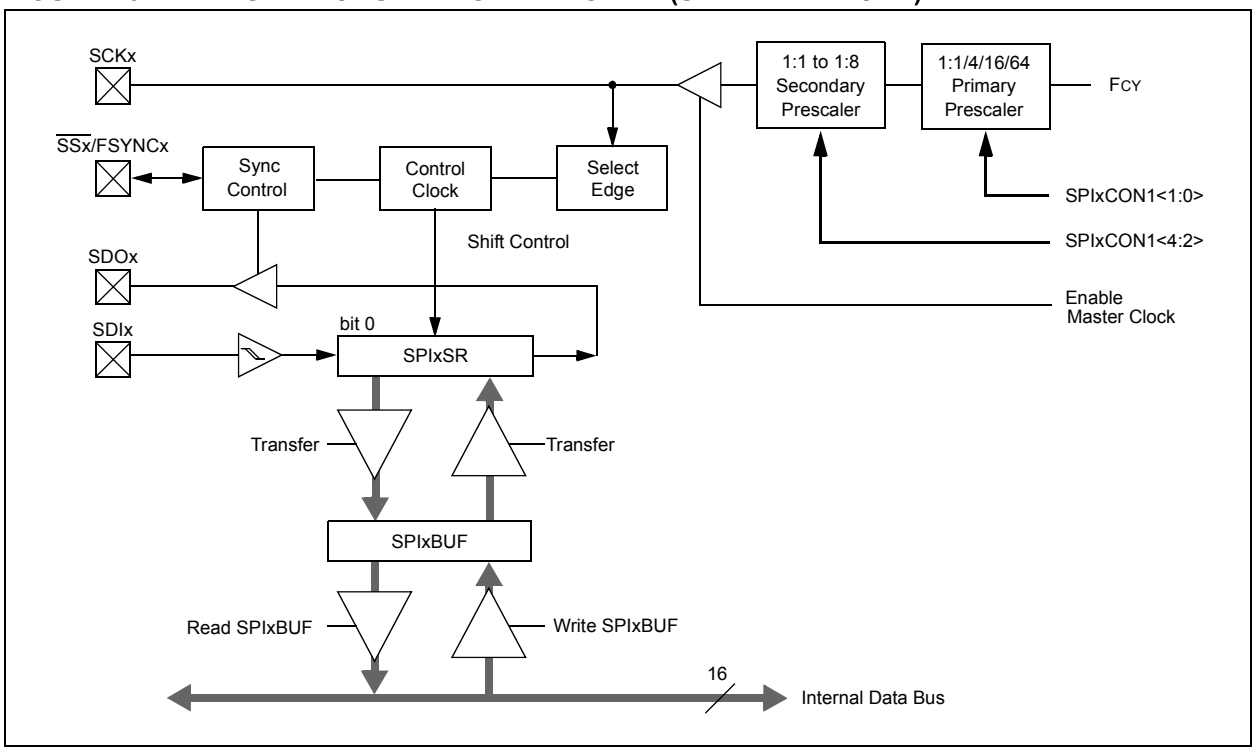
To set up the SPI module for the Standard Master mode of operation:

1. If using interrupts:
   a) Clear the SPIxIF bit in the respective IFS register.
   b) Set the SPIxIE bit in the respective IEC register.
   c) Write the SPIxIP bits in the respective IPC register to set the interrupt priority.
2. Write the desired settings to the SPIxCON1 and SPIxCON2 registers with MSTEN (SPIxCON1<5>) = 1.
3. Clear the SPIROV bit (SPIxSTAT<6>).
4. Enable SPI operation by setting the SPIEN bit (SPIxSTAT<15>).
5. Write the data to be transmitted to the SPIxBUF register. Transmission (and reception) will start as soon as data is written to the SPIxBUF register.

To set up the SPI module for the Standard Slave mode of operation:

1. Clear the SPIxBUF register.
2. If using interrupts:
   a) Clear the SPIxIF bit in the respective IFS register.
   b) Set the SPIxIE bit in the respective IEC register.
   c) Write the SPIxIP bits in the respective IPC register to set the interrupt priority.
3. Write the desired settings to the SPIxCON1 and SPIxCON2 registers with MSTEN (SPIxCON1<5>) = 0.
4. Clear the SMP bit.
5. If the CKE bit (SPIxCON1<8>) is set, then the SSEN bit (SPIxCON1<7>) must be set to enable the $\overline{SSx}$ pin.
6. Clear the SPIROV bit (SPIxSTAT<6>).
7. Enable SPI operation by setting the SPIEN bit (SPIxSTAT<15>).

FIGURE 15-1: SPIx MODULE BLOCK DIAGRAM (STANDARD MODE)



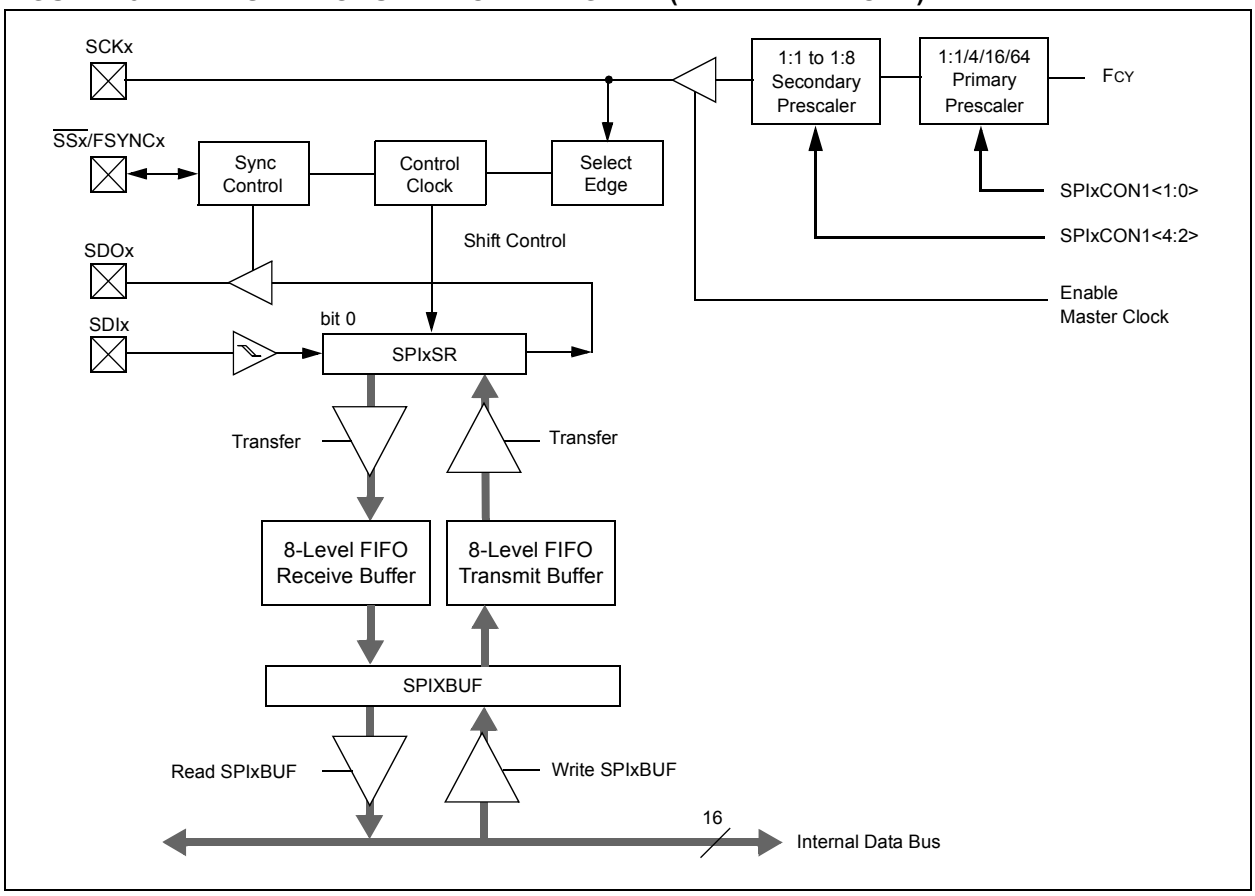
To set up the SPI module for the Enhanced Buffer Master mode of operation:

1. If using interrupts:
   a) Clear the SPIxIF bit in the respective IFS register.
   b) Set the SPIxIE bit in the respective IEC register.
   c) Write the SPIxIP bits in the respective IPC register.
2. Write the desired settings to the SPIxCON1 and SPIxCON2 registers with MSTEN (SPIxCON1<5>) = 1.
3. Clear the SPIROV bit (SPIxSTAT<6>).
4. Select Enhanced Buffer mode by setting the SPIBEN bit (SPIxCON2<0>).
5. Enable SPI operation by setting the SPIEN bit (SPIxSTAT<15>).
6. Write the data to be transmitted to the SPIxBUF register. Transmission (and reception) will start as soon as data is written to the SPIxBUF register.

To set up the SPI module for the Enhanced Buffer Slave mode of operation:

1. Clear the SPIxBUF register.
2. If using interrupts:
   a) Clear the SPIxIF bit in the respective IFS register.
   b) Set the SPIxIE bit in the respective IEC register.
   c) Write the SPIxIP bits in the respective IPC register to set the interrupt priority.
3. Write the desired settings to the SPIxCON1 and SPIxCON2 registers with MSTEN (SPIxCON1<5>) = 0.
4. Clear the SMP bit.
5. If the CKE bit is set, then the SSEN bit must be set, thus enabling the $\overline{SSx}$ pin.
6. Clear the SPIROV bit (SPIxSTAT<6>).
7. Select Enhanced Buffer mode by setting the SPIBEN bit (SPIxCON2<0>).
8. Enable SPI operation by setting the SPIEN bit (SPIxSTAT<15>).

FIGURE 15-2: SPIx MODULE BLOCK DIAGRAM (ENHANCED MODE)



REGISTER 15-1: SPIxSTAT: SPIx STATUS AND CONTROL REGISTER

R/W-0	U-0	R/W-0	U-0	U-0	R-0, HSC	R-0, HSC	R-0, HSC
SPIEN(1)	—	SPISIDL	—	—	SPIBEC2	SPIBEC1	SPIBEC0
bit 15							bit 8

R-0, HSC	R/C-0, HS	R-0, HSC	R/W-0	R/W-0	R/W-0	R-0, HSC	R-0, HSC
SRMPT	SPIROV	SRXMPT	SISEL2	SISEL1	SISEL0	SPITBF	SPIRBF
bit 7							bit 0

Legend:	C = Clearable bit	HS = Hardware Settable bit	
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown
HSC = Hardware Settable/Clearable bit			

bit 15 **SPIEN:** SPIx Enable bit(1)
1 = Enables module and configures SCKx, SDOx, SDIx and $\overline{SSx}$ as serial port pins
0 = Disables module

bit 14 **Unimplemented:** Read as '0'

bit 13 **SPISIDL:** Stop in Idle Mode bit
1 = Discontinue module operation when device enters Idle mode
0 = Continue module operation in Idle mode

bit 12-11 **Unimplemented:** Read as '0'

bit 10-8 **SPIBEC<2:0>:** SPIx Buffer Element Count bits (valid in Enhanced Buffer mode)
Master mode:
Number of SPI transfers pending.
Slave mode:
Number of SPI transfers unread.

bit 7 **SRMPT:** Shift Register (SPIxSR) Empty bit (valid in Enhanced Buffer mode)
1 = SPIx Shift register is empty and ready to send or receive
0 = SPIx Shift register is not empty

bit 6 **SPIROV:** Receive Overflow Flag bit
1 = A new byte/word is completely received and discarded
(The user software has not read the previous data in the SPIxBUF register.)
0 = No overflow has occurred

bit 5 **SRXMPT:** Receive FIFO Empty bit (valid in Enhanced Buffer mode)
1 = Receive FIFO is empty
0 = Receive FIFO is not empty

bit 4-2 **SISEL<2:0>:** SPIx Buffer Interrupt Mode bits (valid in Enhanced Buffer mode)
111 = Interrupt when the SPIx transmit buffer is full (SPITBF bit is set)
110 = Interrupt when the last bit is shifted into SPIxSR; as a result, the TX FIFO is empty
101 = Interrupt when the last bit is shifted out of SPIxSR; now the transmit is complete
100 = Interrupt when one data is shifted into the SPIxSR; as a result, the TX FIFO has one open spot
011 = Interrupt when the SPIx receive buffer is full (SPIRBF bit set)
010 = Interrupt when the SPIx receive buffer is 3/4 or more full
001 = Interrupt when data is available in the receive buffer (SRMPT bit is set)
000 = Interrupt when the last data in the receive buffer is read; as a result, the buffer is empty (SRXMPT bit set)

Note 1: If SPIEN = 1, these functions must be assigned to available RPn/RPIn pins before use. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

REGISTER 15-1: SPIxSTAT: SPIx STATUS AND CONTROL REGISTER (CONTINUED)

bit 1 **SPITBF:** SPIx Transmit Buffer Full Status bit
1 = Transmit not yet started, SPIxTXB is full
0 = Transmit started, SPIxTXB is empty
In Standard Buffer mode:
Automatically set in hardware when the CPU writes to the SPIxBUF location, loading SPIxTXB.
Automatically cleared in hardware when the SPIx module transfers data from SPIxTXB to SPIxSR.
In Enhanced Buffer mode:
Automatically set in hardware when the CPU writes to the SPIxBUF location, loading the last available buffer location.
Automatically cleared in hardware when a buffer location is available for a CPU write.

bit 0 **SPIRBF:** SPIx Receive Buffer Full Status bit
1 = Receive complete, SPIxRXB is full
0 = Receive is not complete, SPIxRXB is empty
In Standard Buffer mode:
Automatically set in hardware when SPIx transfers data from SPIxSR to SPIxRXB.
Automatically cleared in hardware when the core reads the SPIxBUF location, reading SPIxRXB.
In Enhanced Buffer mode:
Automatically set in hardware when SPIx transfers data from the SPIxSR to the buffer, filling the last unread buffer location.
Automatically cleared in hardware when a buffer location is available for a transfer from SPIxSR.

Note 1: If SPIEN = 1, these functions must be assigned to available RPn/RPIn pins before use. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

REGISTER 15-2: SPIxCON1: SPIx CONTROL REGISTER 1

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	DISSCK(1)	DISSDO(2)	MODE16	SMP	CKE(3)
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SSEN(4)	CKP	MSTEN	SPRE2	SPRE1	SPRE0	PPRE1	PPRE0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-13 **Unimplemented:** Read as '0'

bit 12 **DISSCK:** Disable SCKx Pin bit (SPI Master modes only)(1)
1 = Internal SPI clock is disabled; pin functions as I/O
0 = Internal SPI clock is enabled

bit 11 **DISSDO:** Disable SDOx Pin bit(2)
1 = SDOx pin is not used by the module; pin functions as I/O
0 = SDOx pin is controlled by the module

bit 10 **MODE16:** Word/Byte Communication Select bit
1 = Communication is word-wide (16 bits)
0 = Communication is byte-wide (8 bits)

bit 9 **SMP:** SPIx Data Input Sample Phase bit
Master mode:
1 = Input data sampled at the end of data output time
0 = Input data sampled at the middle of data output time
Slave mode:
SMP must be cleared when SPIx is used in Slave mode.

bit 8 **CKE:** SPIx Clock Edge Select bit(3)
1 = Serial output data changes on transition from active clock state to Idle clock state (see bit 6)
0 = Serial output data changes on transition from Idle clock state to active clock state (see bit 6)

bit 7 **SSEN:** Slave Select Enable (Slave mode) bit(4)
1 = $\overline{SSx}$ pin is used for Slave mode
0 = $\overline{SSx}$ pin is not used by the module; pin is controlled by the port function

bit 6 **CKP:** Clock Polarity Select bit
1 = Idle state for the clock is a high level; active state is a low level
0 = Idle state for the clock is a low level; active state is a high level

bit 5 **MSTEN:** Master Mode Enable bit
1 = Master mode
0 = Slave mode

Note 1: If DISSCK = 0, SCKx must be configured to an available RPn pin. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

2: If DISSDO = 0, SDOx must be configured to an available RPn pin. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

3: The CKE bit is not used in the Framed SPI modes. The user should program this bit to '0' for the Framed SPI modes (FRMEN = 1).

4: If SSEN = 1, $\overline{SSx}$ must be configured to an available RPn/PRIn pin. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

REGISTER 15-2: SPIxCON1: SPIx CONTROL REGISTER 1 (CONTINUED)

bit 4-2 **SPRE<2:0>:** Secondary Prescale bits (Master mode)
111 = Secondary prescale 1:1
110 = Secondary prescale 2:1
.
.
.
000 = Secondary prescale 8:1

bit 1-0 **PPRE<1:0>:** Primary Prescale bits (Master mode)
11 = Primary prescale 1:1
10 = Primary prescale 4:1
01 = Primary prescale 16:1
00 = Primary prescale 64:1

Note 1: If DISSCK = 0, SCKx must be configured to an available RPn pin. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

2: If DISSDO = 0, SDOx must be configured to an available RPn pin. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

3: The CKE bit is not used in the Framed SPI modes. The user should program this bit to '0' for the Framed SPI modes (FRMEN = 1).

4: If SSEN = 1, $\overline{SSx}$ must be configured to an available RPn/PRIn pin. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

REGISTER 15-3: SPIxCON2: SPIx CONTROL REGISTER 2

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0
FRMEN	SPIFSD	SPIFPOL	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	SPIFE	SPIBEN
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **FRMEN:** Framed SPIx Support bit
1 = Framed SPIx support is enabled
0 = Framed SPIx support is disabled

bit 14 **SPIFSD:** Frame Sync Pulse Direction Control on $\overline{SSx}$ Pin bit
1 = Frame sync pulse input (slave)
0 = Frame sync pulse output (master)

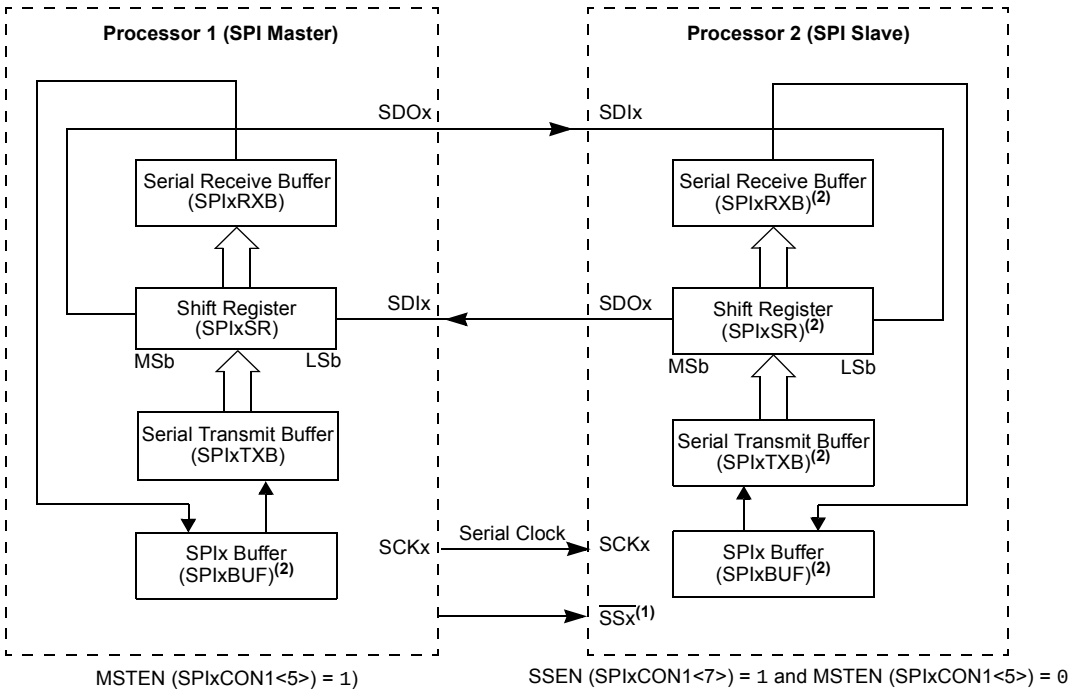
bit 13 **SPIFPOL:** Frame Sync Pulse Polarity bit (Frame mode only)
1 = Frame sync pulse is active-high
0 = Frame sync pulse is active-low

bit 12-2 **Unimplemented:** Read as '0'

bit 1 **SPIFE:** Frame Sync Pulse Edge Select bit
1 = Frame sync pulse coincides with the first bit clock
0 = Frame sync pulse precedes the first bit clock

bit 0 **SPIBEN:** Enhanced Buffer Enable bit
1 = Enhanced buffer is enabled
0 = Enhanced buffer is disabled (Legacy mode)

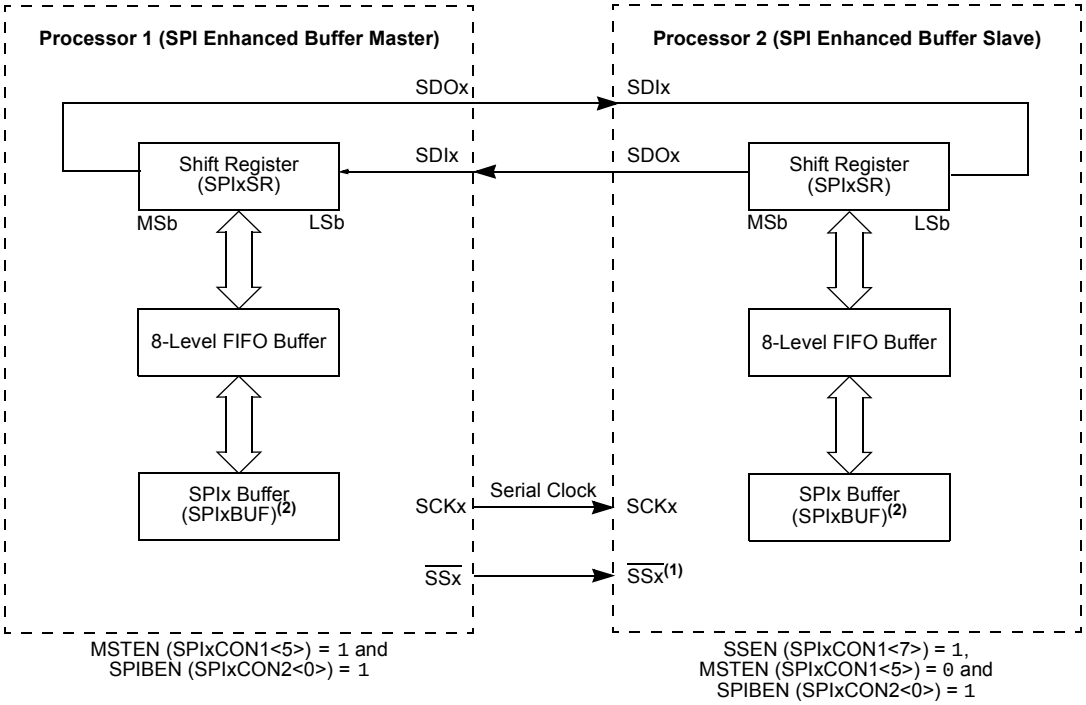
FIGURE 15-3: SPI MASTER/SLAVE CONNECTION (STANDARD MODE)



Note 1: Using the $\overline{SSx}$ pin in Slave mode of operation is optional.

2: User must write transmit data to read received data from SPIxBUF. The SPIxTXB and SPIxRXB registers are memory mapped to SPIxBUF.

FIGURE 15-4: SPI MASTER/SLAVE CONNECTION (ENHANCED BUFFER MODES)



Note 1: Using the $\overline{SSx}$ pin in Slave mode of operation is optional.

2: User must write transmit data to read received data from SPIxBUF. The SPIxTXB and SPIxRXB registers are memory mapped to SPIxBUF.

FIGURE 15-5: SPI MASTER, FRAME MASTER CONNECTION DIAGRAM

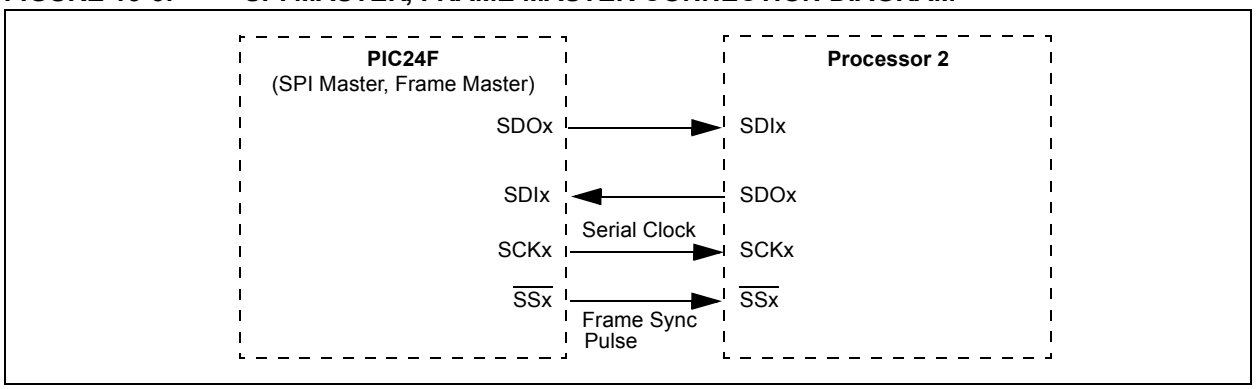


FIGURE 15-6: SPI MASTER, FRAME SLAVE CONNECTION DIAGRAM

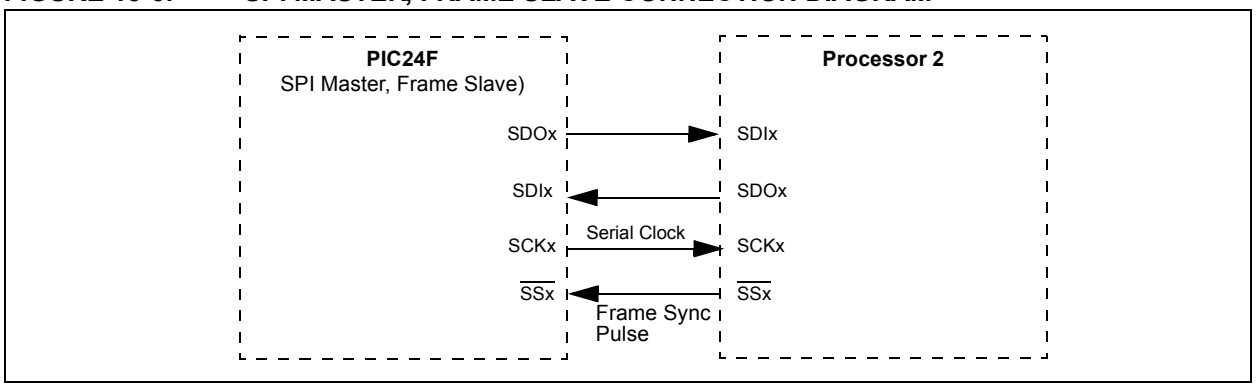


FIGURE 15-7: SPI SLAVE, FRAME MASTER CONNECTION DIAGRAM

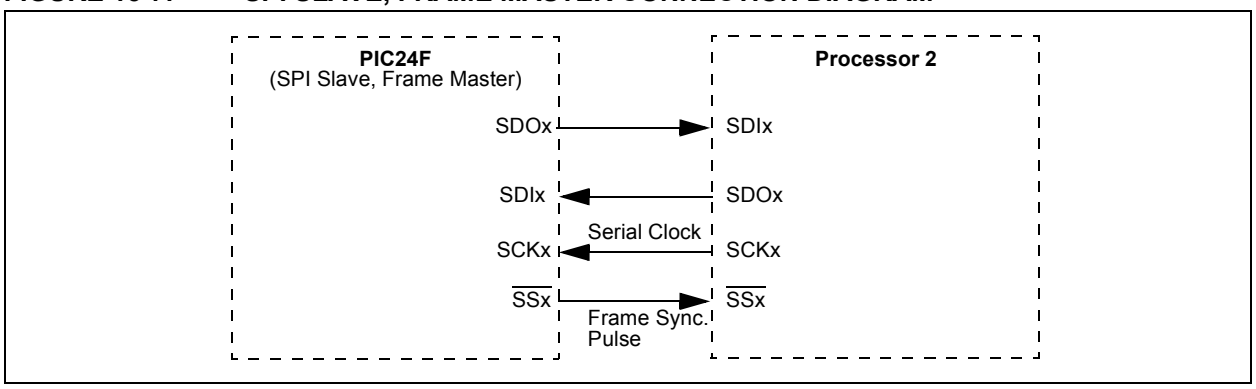
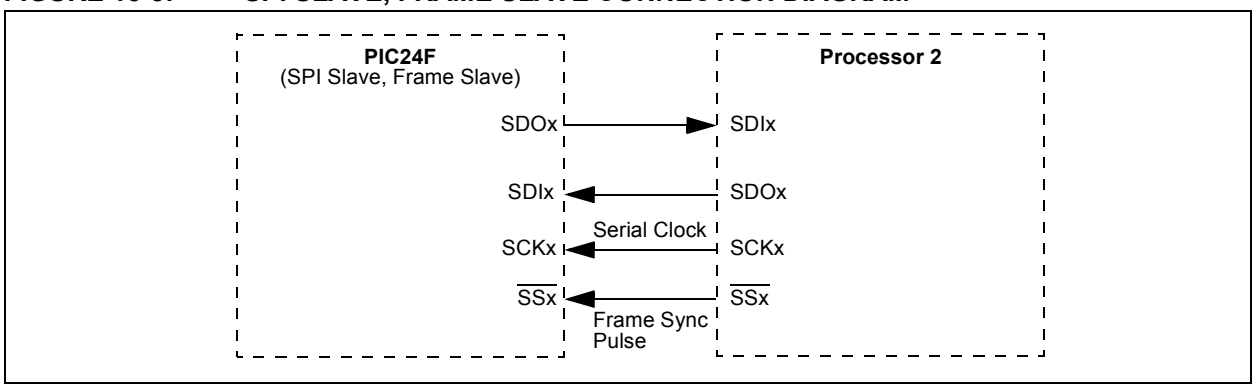


FIGURE 15-8: SPI SLAVE, FRAME SLAVE CONNECTION DIAGRAM



EQUATION 15-1: RELATIONSHIP BETWEEN DEVICE AND SPI CLOCK SPEED(1)

$$FSCK = \frac{FCY}{\text{Primary Prescaler x Secondary Prescaler}}$$

Note 1: Based on FCY = FOSC/2; Doze mode and PLL are disabled.

TABLE 15-1: SAMPLE SCKx FREQUENCIES(1,2)

FCY = 16 MHz		Secondary Prescaler Settings				
		1:1	2:1	4:1	6:1	8:1
Primary Prescaler Settings	1:1	Invalid	8000	4000	2667	2000
	4:1	4000	2000	1000	667	500
	16:1	1000	500	250	167	125
	64:1	250	125	63	42	31
FCY = 5 MHz						
Primary Prescaler Settings	1:1	5000	2500	1250	833	625
	4:1	1250	625	313	208	156
	16:1	313	156	78	52	39
	64:1	78	39	20	13	10

Note 1: Based on FCY = FOSC/2; Doze mode and PLL are disabled.

2: SCKx frequencies shown in kHz.

NOTES:

16.0 INTER-INTEGRATED CIRCUIT™ (I^2C™)

Note:	This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the *"PIC24F Family Reference Manual"*, **Section 24. "Inter-Integrated Circuit™ (I^2C™)"** (DS39702). The information in this data sheet supersedes the information in the FRM.

The Inter-Integrated Circuit™ (I^2C™) module is a serial interface useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be serial EEPROMs, display drivers, A/D Converters, etc.

The I^2C module supports these features:

- Independent master and slave logic
- 7-bit and 10-bit device addresses
- General call address, as defined in the I^2C protocol
- Clock stretching to provide delays for the processor to respond to a slave data request
- Both 100 kHz and 400 kHz bus specifications
- Configurable address masking
- Multi-Master modes to prevent loss of messages in arbitration
- Bus Repeater mode, allowing the acceptance of all messages as a slave regardless of the address
- Automatic SCL

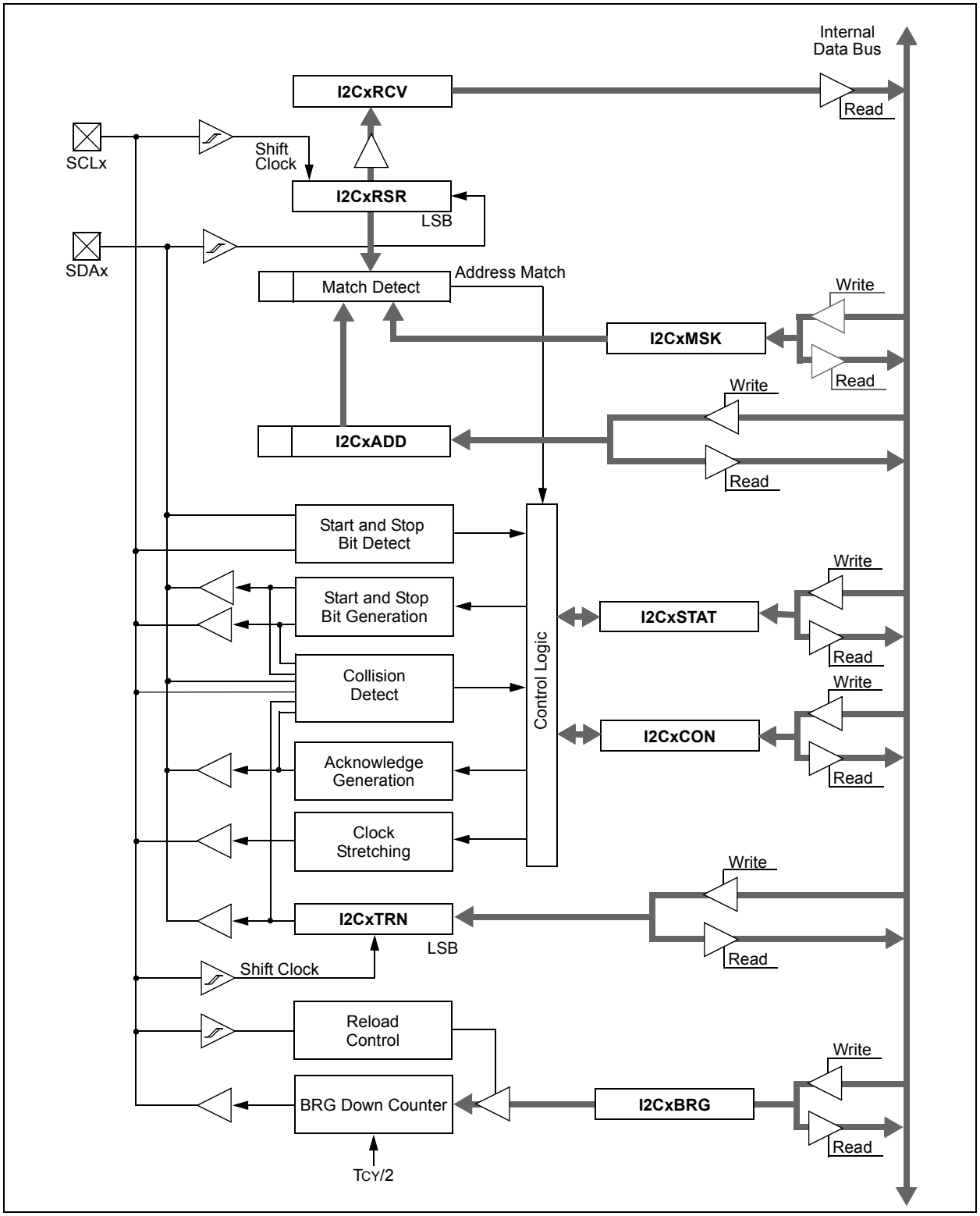
A block diagram of the module is shown in Figure 16-1.

16.1 Communicating as a Master in a Single Master Environment

The details of sending a message in Master mode depends on the communications protocol for the device being communicated with. Typically, the sequence of events is as follows:

1. Assert a Start condition on SDAx and SCLx.
2. Send the I^2C device address byte to the slave with a write indication.
3. Wait for and verify an Acknowledge from the slave.
4. Send the first data byte (sometimes known as the command) to the slave.
5. Wait for and verify an Acknowledge from the slave.
6. Send the serial memory address low byte to the slave.
7. Repeat steps 4 and 5 until all data bytes are sent.
8. Assert a Repeated Start condition on SDAx and SCLx.
9. Send the device address byte to the slave with a read indication.
10. Wait for and verify an Acknowledge from the slave.
11. Enable master reception to receive serial memory data.
12. Generate an ACK or NACK condition at the end of a received byte of data.
13. Generate a Stop condition on SDAx and SCLx.

FIGURE 16-1: I²C™ BLOCK DIAGRAM



16.2 Setting Baud Rate When Operating as a Bus Master

To compute the Baud Rate Generator reload value, use Equation 16-1.

EQUATION 16-1: COMPUTING BAUD RATE RELOAD VALUE(1,2)

$$FSCL = \frac{FCY}{I2CxBRG + 1 + \frac{FCY}{10{,}000{,}000}}$$

or:

$$I2CxBRG = \left(\frac{FCY}{FSCL} - \frac{FCY}{10{,}000{,}000} - 1\right)$$

Note 1: Based on FCY = FOSC/2; Doze mode and PLL are disabled.

2: These clock rate values are for guidance only. The actual clock rate can be affected by various system level parameters. The actual clock rate should be measured in its intended application.

16.3 Slave Address Masking

The I2CxMSK register (Register 16-3) designates address bit positions as “don’t care” for both 7-Bit and 10-Bit Addressing modes. Setting a particular bit location (= 1) in the I2CxMSK register causes the slave module to respond whether the corresponding address bit value is a ‘0’ or a ‘1’. For example, when I2CxMSK is set to ‘00100000’, the slave module will detect both addresses, ‘0000000’ and ‘0100000’.

To enable address masking, the Intelligent Peripheral Management Interface (IPMI) must be disabled by clearing the IPMIEN bit (I2CxCON<11>).

Note: As a result of changes in the I²C™ protocol, the addresses in Table 16-2 are reserved and will not be Acknowledged in Slave mode. This includes any address mask settings that include any of these addresses.

TABLE 16-1: I2C™ CLOCK RATES(1,2)

Required System FSCL	FCY	I2CxBRG Value		Actual FSCL
		(Decimal)	(Hexadecimal)	
100 kHz	16 MHz	157	9D	100 kHz
100 kHz	8 MHz	78	4E	100 kHz
100 kHz	4 MHz	39	27	99 kHz
400 kHz	16 MHz	37	25	404 kHz
400 kHz	8 MHz	18	12	404 kHz
400 kHz	4 MHz	9	9	385 kHz
400 kHz	2 MHz	4	4	385 kHz
1 MHz	16 MHz	13	D	1.026 MHz
1 MHz	8 MHz	6	6	1.026 MHz
1 MHz	4 MHz	3	3	0.909 MHz

Note 1: Based on FCY = FOSC/2; Doze mode and PLL are disabled.

2: These clock rate values are for guidance only. The actual clock rate can be affected by various system level parameters. The actual clock rate should be measured in its intended application.

TABLE 16-2: I²C™ RESERVED ADDRESSES(1)

Slave Address	R/$\overline{W}$ Bit	Description
0000 000	0	General Call Address(2)
0000 000	1	Start Byte
0000 001	x	CBus Address
0000 01x	x	Reserved
0000 1xx	x	HS Mode Master Code
1111 0xx	x	10-Bit Slave Upper Byte(3)
1111 1xx	x	Reserved

Note 1: The address bits listed here will never cause an address match, independent of address mask settings.

2: The address will be Acknowledged only if GCEN = 1.

3: A match on this address can only occur on the upper byte in 10-Bit Addressing mode.

REGISTER 16-1: I2CxCON: I2Cx CONTROL REGISTER

R/W-0	U-0	R/W-0	R/W-1, HC	R/W-0	R/W-0	R/W-0	R/W-0
I2CEN	—	I2CSIDL	SCLREL	IPMIEN	A10M	DISSLW	SMEN
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0, HC	R/W-0, HC	R/W-0, HC	R/W-0, HC	R/W-0, HC
GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
bit 7							bit 0

Legend:	HC = Hardware Clearable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **I2CEN:** I2Cx Enable bit
1 = Enables the I2Cx module and configures the SDAx and SCLx pins as serial port pins
0 = Disables the I2Cx module. All I^2C™ pins are controlled by port functions

bit 14 **Unimplemented:** Read as '0'

bit 13 **I2CSIDL:** Stop in Idle Mode bit
1 = Discontinues module operation when device enters an Idle mode
0 = Continues module operation in Idle mode

bit 12 **SCLREL:** SCLx Release Control bit (when operating as I^2C slave)
1 = Releases SCLx clock
0 = Holds SCLx clock low (clock stretch)
If STREN = 1:
Bit is R/W (i.e., software may write '0' to initiate stretch and write '1' to release clock). Hardware is clear at the beginning of slave transmission. Hardware is clear at the end of slave reception.
If STREN = 0:
Bit is R/S (i.e., software may only write '1' to release clock). Hardware clear at beginning of slave transmission.

bit 11 **IPMIEN:** Intelligent Platform Management Interface (IPMI) Enable bit
1 = IPMI Support mode is enabled; all addresses are Acknowledged
0 = IPMI mode is disabled

bit 10 **A10M:** 10-Bit Slave Addressing bit
1 = I2CxADD is a 10-bit slave address
0 = I2CxADD is a 7-bit slave address

bit 9 **DISSLW:** Disable Slew Rate Control bit
1 = Slew rate control disabled
0 = Slew rate control enabled

bit 8 **SMEN:** SMBus Input Levels bit
1 = Enables I/O pin thresholds compliant with SMBus specifications
0 = Disables the SMBus input thresholds

bit 7 **GCEN:** General Call Enable bit (when operating as I^2C slave)
1 = Enables interrupt when a general call address is received in the I2CxRSR (module is enabled for reception)
0 = General call address disabled

bit 6 **STREN:** SCLx Clock Stretch Enable bit (when operating as I^2C slave)
Used in conjunction with the SCLREL bit.
1 = Enables software or receive clock stretching
0 = Disables software or receive clock stretching

REGISTER 16-1: I2CxCON: I2Cx CONTROL REGISTER (CONTINUED)

bit 5 **ACKDT:** Acknowledge Data bit (when operating as I^2C master. Applicable during master receive.)
Value that will be transmitted when the software initiates an Acknowledge sequence.
1 = Sends NACK during Acknowledge
0 = Sends ACK during Acknowledge

bit 4 **ACKEN:** Acknowledge Sequence Enable bit (when operating as I^2C master. Applicable during master receive.)
1 = Initiates Acknowledge sequence on SDAx and SCLx pins and transmits the ACKDT data bit. Hardware is clear at the end of the master Acknowledge sequence.
0 = Acknowledge sequence is not in progress

bit 3 **RCEN:** Receive Enable bit (when operating as I^2C master)
1 = Enables Receive mode for I^2C. Hardware is clear at the end of the eighth bit of the master receive data byte.
0 = Receive sequence is not in progress

bit 2 **PEN:** Stop Condition Enable bit (when operating as I^2C master)
1 = Initiates Stop condition on the SDAx and SCLx pins. Hardware is clear at the end of the master Stop sequence.
0 = Stop condition is not in progress

bit 1 **RSEN:** Repeated Start Condition Enabled bit (when operating as I^2C master)
1 = Initiates Repeated Start condition on the SDAx and SCLx pins. Hardware is clear at the end of the master Repeated Start sequence
0 = Repeated Start condition is not in progress

bit 0 **SEN:** Start Condition Enabled bit (when operating as I^2C master)
1 = Initiates Start condition on SDAx and SCLx pins. Hardware is clear at end of the master Start sequence.
0 = Start condition is not in progress

REGISTER 16-2: I2CxSTAT: I2Cx STATUS REGISTER

R-0, HSC	R-0, HSC	U-0	U-0	U-0	R/C-0, HS	R-0, HSC	R-0, HSC
ACKSTAT	TRSTAT	—	—	—	BCL	GCSTAT	ADD10
bit 15							bit 8

R/C-0, HS	R/C-0, HS	R-0, HSC	R/C-0, HSC	R/C-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC
IWCOL	I2COV	D/$\overline{A}$	P	S	R/$\overline{W}$	RBF	TBF
bit 7							bit 0

Legend:	C = Clearable bit	HS = Hardware Settable bit	
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown
HSC = Hardware Settable/Clearable bit			

bit 15 **ACKSTAT:** Acknowledge Status bit
1 = NACK was detected last
0 = ACK was detected last
Hardware is set or clear at the end of Acknowledge.

bit 14 **TRSTAT:** Transmit Status bit
(When operating as I^2C™ master. Applicable to master transmit operation.)
1 = Master transmit is in progress (8 bits + ACK)
0 = Master transmit is not in progress
Hardware is set at the beginning of master transmission; hardware is clear at the end of slave Acknowledge.

bit 13-11 **Unimplemented:** Read as '0'

bit 10 **BCL:** Master Bus Collision Detect bit
1 = A bus collision has been detected during a master operation
0 = No collision
Hardware is set at the detection of a bus collision.

bit 9 **GCSTAT:** General Call Status bit
1 = General call address was received
0 = General call address was not received
Hardware is set when the address matches the general call address; hardware is clear at Stop detection.

bit 8 **ADD10:** 10-Bit Address Status bit
1 = 10-bit address was matched
0 = 10-bit address was not matched
Hardware is set at the match of the 2nd byte of the matched 10-bit address; hardware is clear at Stop detection.

bit 7 **IWCOL:** Write Collision Detect bit
1 = An attempt to write to the I2CxTRN register failed because the I^2C module is busy
0 = No collision
Hardware is set at an occurrence of write to I2CxTRN while busy (cleared by software).

bit 6 **I2COV:** Receive Overflow Flag bit
1 = A byte was received while the I2CxRCV register is still holding the previous byte
0 = No overflow
Hardware is set at an attempt to transfer I2CxRSR to I2CxRCV (cleared by software).

bit 5 **D/$\overline{A}$:** Data/$\overline{Address}$ bit (when operating as I^2C slave)
1 = Indicates that the last byte received was data
0 = Indicates that the last byte received was a device address
Hardware is clear at the device address match. Hardware is set after a transmission finishes or by reception of a slave byte.

REGISTER 16-2: I2CxSTAT: I2Cx STATUS REGISTER (CONTINUED)

bit 4 **P:** Stop bit
1 = Indicates that a Stop bit has been detected last
0 = Stop bit was not detected last
Hardware is set or clear when Start, Repeated Start or Stop is detected.

bit 3 **S:** Start bit
1 = Indicates that a Start (or Repeated Start) bit has been detected last
0 = Start bit was not detected last
Hardware is set or clear when Start, Repeated Start or Stop is detected.

bit 2 **R/$\overline{W}$:** Read/$\overline{\text{Write}}$ Information bit (when operating as I^2C slave)
1 = Read – indicates data transfer is output from slave
0 = Write – indicates data transfer is input to slave
Hardware is set or clear after the reception of an I^2C device address byte.

bit 1 **RBF:** Receive Buffer Full Status bit
1 = Receive is complete, I2CxRCV is full
0 = Receive not complete, I2CxRCV is empty
Hardware is set when I2CxRCV is written with the received byte; hardware is clear when the software reads I2CxRCV.

bit 0 **TBF:** Transmit Buffer Full Status bit
1 = Transmit is in progress, I2CxTRN is full
0 = Transmit is complete, I2CxTRN is empty
Hardware is set when software writes to I2CxTRN; hardware is clear at the completion of data transmission.

REGISTER 16-3: I2CxMSK: I2Cx SLAVE MODE ADDRESS MASK REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	AMSK9	AMSK8
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
AMSK7	AMSK6	AMSK5	AMSK4	AMSK3	AMSK2	AMSK1	AMSK0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-10 **Unimplemented:** Read as ‘0’

bit 9-0 **AMSK<9:0>:** Mask for Address Bit x Select bits

1 = Enable masking for bit x of the incoming message address; bit match is not required in this position

0 = Disable masking for bit x; bit match is required in this position

17.0 UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMITTER (UART)

Note: This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the *"PIC24F Family Reference Manual"*, **Section 21. "UART"** (DS39708). The information in this data sheet supersedes the information in the FRM.

The Universal Asynchronous Receiver Transmitter (UART) module is one of the serial I/O modules available in the PIC24F device family. The UART is a full-duplex, asynchronous system that can communicate with peripheral devices, such as personal computers, LIN/J2602, RS-232 and RS-485 interfaces. The module also supports a hardware flow control option with the UxCTS and UxRTS pins, and also includes an IrDA® encoder and decoder.

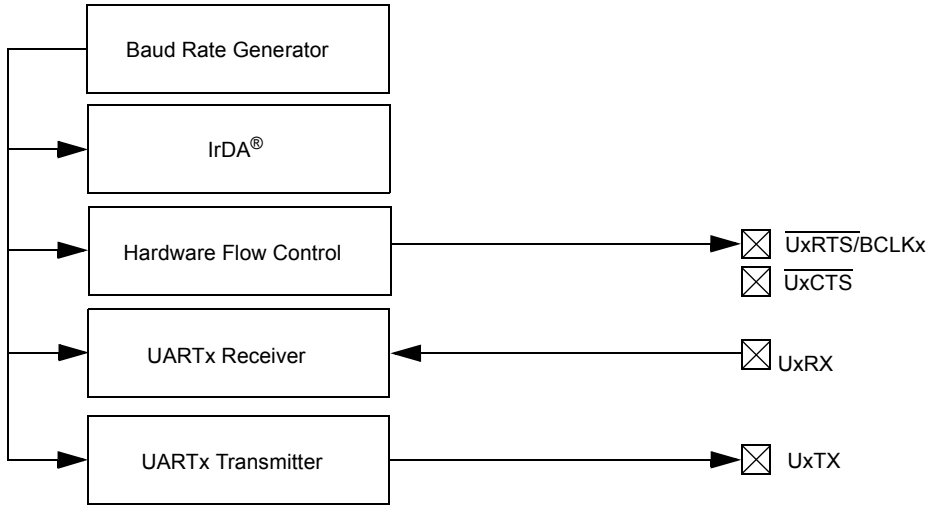
The primary features of the UART module are:

- Full-Duplex, 8 or 9-Bit data transmission through the UxTX and UxRX pins
- Even, Odd or No Parity options (for 8-bit data)
- One or two Stop bits
- Hardware Flow Control option with the UxCTS and UxRTS pins
- Fully Integrated Baud Rate Generator with 16-Bit Prescaler
- Baud Rates Ranging from 15 bps to 1 Mbps at 16 MIPS
- 4-Deep, First-In-First-Out (FIFO) Transmit Data Buffer
- 4-Deep FIFO Receive Data Buffer
- Parity, Framing and Buffer Overrun Error Detection
- Support for 9-bit mode with Address Detect (9^{th} bit = 1)
- Transmit and Receive Interrupts
- Loopback mode for Diagnostic Support
- Support for Sync and Break Characters
- Supports Automatic Baud Rate Detection
- IrDA® Encoder and Decoder Logic
- 16x Baud Clock Output for IrDA Support

A simplified block diagram of the UART is shown in Figure 17-1. The UART module consists of these key important hardware elements:

- Baud Rate Generator
- Asynchronous Transmitter
- Asynchronous Receiver

FIGURE 17-1: UART SIMPLIFIED BLOCK DIAGRAM



Note: The UART inputs and outputs must all be assigned to available RPn/RPIn pins before use. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

17.1 UART Baud Rate Generator (BRG)

The UART module includes a dedicated, 16-bit Baud Rate Generator. The UxBRG register controls the period of a free-running, 16-bit timer. Equation 17-1 shows the formula for computation of the baud rate with BRGH = 0.

EQUATION 17-1: UART BAUD RATE WITH BRGH = 0(1,2)

$$\text{Baud Rate} = \frac{F_{CY}}{16 \bullet (UxBRG + 1)}$$

$$UxBRG = \frac{F_{CY}}{16 \bullet \text{Baud Rate}} - 1$$

Note 1: FCY denotes the instruction cycle clock frequency (FOSC/2).

2: Based on FCY = FOSC/2; Doze mode and PLL are disabled.

Example 17-1 shows the calculation of the baud rate error for the following conditions:

- FCY = 4 MHz
- Desired Baud Rate = 9600

The maximum baud rate (BRGH = 0) possible is FCY/16 (for UxBRG = 0) and the minimum baud rate possible is FCY/(16 * 65536).

Equation 17-2 shows the formula for computation of the baud rate with BRGH = 1.

EQUATION 17-2: UART BAUD RATE WITH BRGH = 1(1,2)

$$\text{Baud Rate} = \frac{F_{CY}}{4 \bullet (UxBRG + 1)}$$

$$UxBRG = \frac{F_{CY}}{4 \bullet \text{Baud Rate}} - 1$$

Note 1: FCY denotes the instruction cycle clock frequency.

2: Based on FCY = FOSC/2; Doze mode and PLL are disabled.

The maximum baud rate (BRGH = 1) possible is FCY/4 (for UxBRG = 0) and the minimum baud rate possible is FCY/(4 * 65536).

Writing a new value to the UxBRG register causes the BRG timer to be reset (cleared). This ensures the BRG does not wait for a timer overflow before generating the new baud rate.

EXAMPLE 17-1: BAUD RATE ERROR CALCULATION (BRGH = 0)(1)

Desired Baud Rate = FCY/(16 (BRGx + 1))

Solving for BRGx Value:

BRGx = ((FCY/Desired Baud Rate)/16) – 1
BRGx = ((4000000/9600)/16) – 1
BRGx = 25

Calculated Baud Rate = 4000000/(16 (25 + 1))
= 9615

Error = (Calculated Baud Rate – Desired Baud Rate) / Desired Baud Rate
= (9615 – 9600)/9600

Note: Based on FCY = FOSC/2; Doze mode and PLL are disabled.

17.2 Transmitting in 8-Bit Data Mode

1. Set up the UART:
   a) Write appropriate values for data, parity and Stop bits.
   b) Write appropriate baud rate value to the UxBRG register.
   c) Set up transmit and receive interrupt enable and priority bits.
2. Enable the UART.
3. Set the UTXEN bit (causes a transmit interrupt two cycles after being set).
4. Write a data byte to the lower byte of UxTXREG word. The value will be immediately transferred to the Transmit Shift Register (TSR) and the serial bit stream will start shifting out with the next rising edge of the baud clock.
5. Alternately, the data byte may be transferred while UTXEN = 0 and then the user may set UTXEN. This will cause the serial bit stream to begin immediately because the baud clock will start from a cleared state.
6. A transmit interrupt will be generated as per interrupt control bit, UTXISELx.

17.3 Transmitting in 9-Bit Data Mode

1. Set up the UART (as described in **Section 17.2 "Transmitting in 8-Bit Data Mode"**).
2. Enable the UART.
3. Set the UTXEN bit (causes a transmit interrupt).
4. Write UxTXREG as a 16-bit value only.
5. A word write to UxTXREG triggers the transfer of the 9-bit data to the TSR. The serial bit stream will start shifting out with the first rising edge of the baud clock.
6. A transmit interrupt will be generated as per the setting of control bit, UTXISELx.

17.4 Break and Sync Transmit Sequence

The following sequence will send a message frame header, made up of a Break, followed by an auto-baud Sync byte.

1. Configure the UART for the desired mode.
2. Set UTXEN and UTXBRK to set up the Break character.
3. Load the UxTXREG with a dummy character to initiate transmission (value is ignored).
4. Write '55h' to UxTXREG; this loads the Sync character into the transmit FIFO.
5. After the Break has been sent, the UTXBRK bit is reset by hardware. The Sync character now transmits.

17.5 Receiving in 8-Bit or 9-Bit Data Mode

1. Set up the UART (as described in **Section 17.2 "Transmitting in 8-Bit Data Mode"**).
2. Enable the UART.
3. A receive interrupt will be generated when one or more data characters have been received as per interrupt control bit, URXISELx.
4. Read the OERR bit to determine if an overrun error has occurred. The OERR bit must be reset in software.
5. Read UxRXREG.

The act of reading the UxRXREG character will move the next character to the top of the receive FIFO, including a new set of PERR and FERR values.

17.6 Operation of $\overline{\text{UxCTS}}$ and $\overline{\text{UxRTS}}$ Control Pins

UARTx Clear to Send ($\overline{\text{UxCTS}}$) and Request to Send ($\overline{\text{UxRTS}}$) are the two hardware controlled pins that are associated with the UART module. These two pins allow the UART to operate in Simplex and Flow Control mode. They are implemented to control the transmission and reception between the Data Terminal Equipment (DTE). The UEN<1:0> bits in the UxMODE register configure these pins.

17.7 Infrared Support

The UART module provides two types of infrared UART support: one is the IrDA clock output to support an external IrDA encoder and decoder device (legacy module support), and the other is the full implementation of the IrDA encoder and decoder. Note that because the IrDA modes require a 16x baud clock, they will only work when the BRGH bit (UxMODE<3>) is '0'.

17.7.1 IrDA CLOCK OUTPUT FOR EXTERNAL IrDA SUPPORT

To support external IrDA encoder and decoder devices, the BCLKx pin (same as the $\overline{\text{UxRTS}}$ pin) can be configured to generate the 16x baud clock. With UEN<1:0> = 11, the BCLKx pin will output the 16x baud clock if the UART module is enabled. It can be used to support the IrDA codec chip.

17.7.2 BUILT-IN IrDA ENCODER AND DECODER

The UART has full implementation of the IrDA encoder and decoder as part of the UART module. The built-in IrDA encoder and decoder functionality is enabled using the IREN bit (UxMODE<12>). When enabled (IREN = 1), the receive pin (UxRX) acts as the input from the infrared receiver. The transmit pin (UxTX) acts as the output to the infrared transmitter.

REGISTER 17-1: UxMODE: UARTx MODE REGISTER

R/W-0	U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
UARTEN(1)	—	USIDL	IREN(2)	RTSMD	—	UEN1	UEN0
bit 15							bit 8

R/W-0, HC	R/W-0	R/W-0, HC	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WAKE	LPBACK	ABAUD	RXINV	BRGH	PDSEL1	PDSEL0	STSEL
bit 7							bit 0

Legend:	HC = Hardware Clearable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **UARTEN:** UARTx Enable bit(1)
- 1 = UARTx is enabled; all UARTx pins are controlled by UARTx as defined by UEN<1:0>
- 0 = UARTx is disabled; all UARTx pins are controlled by port latches; UARTx power consumption is minimal

bit 14 **Unimplemented:** Read as '0'

bit 13 **USIDL:** Stop in Idle Mode bit
- 1 = Discontinue module operation when device enters Idle mode
- 0 = Continue module operation in Idle mode

bit 12 **IREN:** IrDA® Encoder and Decoder Enable bit(2)
- 1 = IrDA encoder and decoder are enabled
- 0 = IrDA encoder and decoder are disabled

bit 11 **RTSMD:** Mode Selection for $\overline{\text{UxRTS}}$ Pin bit
- 1 = $\overline{\text{UxRTS}}$ pin is in Simplex mode
- 0 = $\overline{\text{UxRTS}}$ pin is in Flow Control mode

bit 10 **Unimplemented:** Read as '0'

bit 9-8 **UEN<1:0>:** UARTx Enable bits
- 11 = UxTX, UxRX and BCLKx pins are enabled and used; $\overline{\text{UxCTS}}$ pin is controlled by port latches
- 10 = UxTX, UxRX, $\overline{\text{UxCTS}}$ and $\overline{\text{UxRTS}}$ pins are enabled and used
- 01 = UxTX, UxRX and $\overline{\text{UxRTS}}$ pins are enabled and used; $\overline{\text{UxCTS}}$ pin is controlled by port latches
- 00 = UxTX and UxRX pins are enabled and used; $\overline{\text{UxCTS}}$ and $\overline{\text{UxRTS}}$/BCLKx pins are controlled by port latches

bit 7 **WAKE:** Wake-up on Start Bit Detect During Sleep Mode Enable bit
- 1 = UARTx will continue to sample the UxRX pin; interrupt is generated on the falling edge, bit is cleared in hardware on the following rising edge
- 0 = No wake-up is enabled

bit 6 **LPBACK:** UARTx Loopback Mode Select bit
- 1 = Enable Loopback mode
- 0 = Loopback mode is disabled

bit 5 **ABAUD:** Auto-Baud Enable bit
- 1 = Enable baud rate measurement on the next character – requires reception of a Sync field (55h); cleared in hardware upon completion
- 0 = Baud rate measurement is disabled or completed

Note 1: If UARTEN = 1, the peripheral inputs and outputs must be configured to an available RPn/RPIn pin. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

2: This feature is only available for the 16x BRG mode (BRGH = 0).

REGISTER 17-1: UxMODE: UARTx MODE REGISTER (CONTINUED)

bit 4 **RXINV:** Receive Polarity Inversion bit
1 = UxRX Idle state is '0'
0 = UxRX Idle state is '1'

bit 3 **BRGH:** High Baud Rate Enable bit
1 = High-Speed mode (4 BRG clock cycles per bit)
0 = Standard-Speed mode (16 BRG clock cycles per bit)

bit 2-1 **PDSEL<1:0>:** Parity and Data Selection bits
11 = 9-bit data, no parity
10 = 8-bit data, odd parity
01 = 8-bit data, even parity
00 = 8-bit data, no parity

bit 0 **STSEL:** Stop Bit Selection bit
1 = Two Stop bits
0 = One Stop bit

Note 1: If UARTEN = 1, the peripheral inputs and outputs must be configured to an available RPn/RPIn pin. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

2: This feature is only available for the 16x BRG mode (BRGH = 0).

REGISTER 17-2: UxSTA: UARTx STATUS AND CONTROL REGISTER

R/W-0	R/W-0	R/W-0	U-0	R/W-0 HC	R/W-0	R-0, HSC	R-1, HSC
UTXISEL1	UTXINV(1)	UTXISEL0	—	UTXBRK	UTXEN(2)	UTXBF	TRMT
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R-1, HSC	R-0, HSC	R-0, HSC	R/C-0, HS	R-0, HSC
URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA
bit 7							bit 0

Legend:	C = Clearable bit	HSC = Hardware Settable/Clearable bit	
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown
HS = Hardware Settable bit	HC = Hardware Clearable bit		

bit 15,13 **UTXISEL<1:0>:** Transmission Interrupt Mode Selection bits
- 11 = Reserved; do not use
- 10 = Interrupt when a character is transferred to the Transmit Shift Register (TSR) and as a result, the transmit buffer becomes empty
- 01 = Interrupt when the last character is shifted out of the Transmit Shift Register; all transmit operations are completed
- 00 = Interrupt when a character is transferred to the Transmit Shift Register (this implies there is at least one character open in the transmit buffer)

bit 14 **UTXINV:** IrDA® Encoder Transmit Polarity Inversion bit(1)

IREN = 0:
- 1 = UxTX is Idle '0'
- 0 = UxTX is Idle '1'

IREN = 1:
- 1 = UxTX is Idle '1'
- 0 = UxTX is Idle '0'

bit 12 **Unimplemented:** Read as '0'

bit 11 **UTXBRK:** Transmit Break bit
- 1 = Send Sync Break on next transmission – Start bit, followed by twelve '0' bits, followed by Stop bit; cleared by hardware upon completion
- 0 = Sync Break transmission is disabled or completed

bit 10 **UTXEN:** Transmit Enable bit(2)
- 1 = Transmit is enabled, UxTX pin controlled by UARTx
- 0 = Transmit is disabled, any pending transmission is aborted and the buffer is reset; UxTX pin is controlled by port.

bit 9 **UTXBF:** Transmit Buffer Full Status bit (read-only)
- 1 = Transmit buffer is full
- 0 = Transmit buffer is not full, at least one more character can be written

bit 8 **TRMT:** Transmit Shift Register Empty bit (read-only)
- 1 = Transmit Shift Register is empty and transmit buffer is empty (the last transmission has completed)
- 0 = Transmit Shift Register is not empty, a transmission is in progress or queued

Note 1: Value of bit only affects the transmit properties of the module when the IrDA® encoder is enabled (IREN = 1).

2: If UARTEN = 1, the peripheral inputs and outputs must be configured to an available RPn/RPIn pin. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

REGISTER 17-2: UxSTA: UARTx STATUS AND CONTROL REGISTER (CONTINUED)

bit 7-6 **URXISEL<1:0>:** Receive Interrupt Mode Selection bits
- 11 = Interrupt is set on an RSR transfer, making the receive buffer full (i.e., has 4 data characters)
- 10 = Interrupt is set on an RSR transfer, making the receive buffer 3/4 full (i.e., has 3 data characters)
- 0x = Interrupt is set when any character is received and transferred from the RSR to the receive buffer; receive buffer has one or more characters

bit 5 **ADDEN:** Address Character Detect bit (bit 8 of received data = 1)
- 1 = Address Detect mode is enabled

If 9-bit mode is not selected, this does not take effect.
- 0 = Address Detect mode is disabled

bit 4 **RIDLE:** Receiver Idle bit (read-only)
- 1 = Receiver is Idle
- 0 = Receiver is active

bit 3 **PERR:** Parity Error Status bit (read-only)
- 1 = Parity error has been detected for the current character (character at the top of the receive FIFO)
- 0 = Parity error has not been detected

bit 2 **FERR:** Framing Error Status bit (read-only)
- 1 = Framing error has been detected for the current character (character at the top of the receive FIFO)
- 0 = Framing error has not been detected

bit 1 **OERR:** Receive Buffer Overrun Error Status bit (clear/read-only)
- 1 = Receive buffer has overflowed
- 0 = Receive buffer has not overflowed (clearing a previously set OERR bit (1 → 0 transition); will reset the receiver buffer and the RSR to the empty state

bit 0 **URXDA:** Receive Buffer Data Available bit (read-only)
- 1 = Receive buffer has data, at least one more character can be read
- 0 = Receive buffer is empty

Note 1: Value of bit only affects the transmit properties of the module when the IrDA® encoder is enabled (IREN = 1).

2: If UARTEN = 1, the peripheral inputs and outputs must be configured to an available RPn/RPIn pin. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

NOTES:

18.0 UNIVERSAL SERIAL BUS WITH ON-THE-GO SUPPORT (USB OTG)

Note: This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the *"PIC24F Family Reference Manual"*, **Section 27. "USB On-The-Go (OTG)"** (DS39721). The information in this data sheet supersedes the information in the FRM.

PIC24FJ256DA210 family devices contain a full-speed and low-speed compatible, On-The-Go (OTG) USB Serial Interface Engine (SIE). The OTG capability allows the device to act either as a USB peripheral device or as a USB embedded host with limited host capabilities. The OTG capability allows the device to dynamically switch from device to host operation using OTG's Host Negotiation Protocol (HNP).

For more details on OTG operation, refer to the "*On-The-Go Supplement to the USB 2.0 Specification*", published by the USB-IF. For more details on USB operation, refer to the "*Universal Serial Bus Specification*", v2.0.

The USB OTG module offers these features:

- USB functionality in Device and Host modes, and OTG capabilities for application-controlled mode switching
- Software-selectable module speeds of full speed (12 Mbps) or low speed (1.5 Mbps, available in Host mode only)
- Support for all four USB transfer types: control, interrupt, bulk and isochronous
- 16 bidirectional endpoints for a total of 32 unique endpoints
- DMA interface for data RAM access
- Queues up to sixteen unique endpoint transfers without servicing
- Integrated, on-chip USB transceiver with support for off-chip transceivers via a digital interface
- Integrated VBUS generation with on-chip comparators and boost generation, and support of external VBUS comparators and regulators through a digital interface
- Configurations for on-chip bus pull-up and pull-down resistors

A simplified block diagram of the USB OTG module is shown in Figure 18-1.

The USB OTG module can function as a USB peripheral device or as a USB host, and may dynamically switch between Device and Host modes under software control. In either mode, the same data paths and Buffer Descriptors (BDs) are used for the transmission and reception of data.

In discussing USB operation, this section will use a controller-centric nomenclature for describing the direction of the data transfer between the microcontroller and the USB. RX (Receive) will be used to describe transfers that move data from the USB to the microcontroller and TX (Transmit) will be used to describe transfers that move data from the microcontroller to the USB. Table 18-1 shows the relationship between data direction in this nomenclature and the USB tokens exchanged.

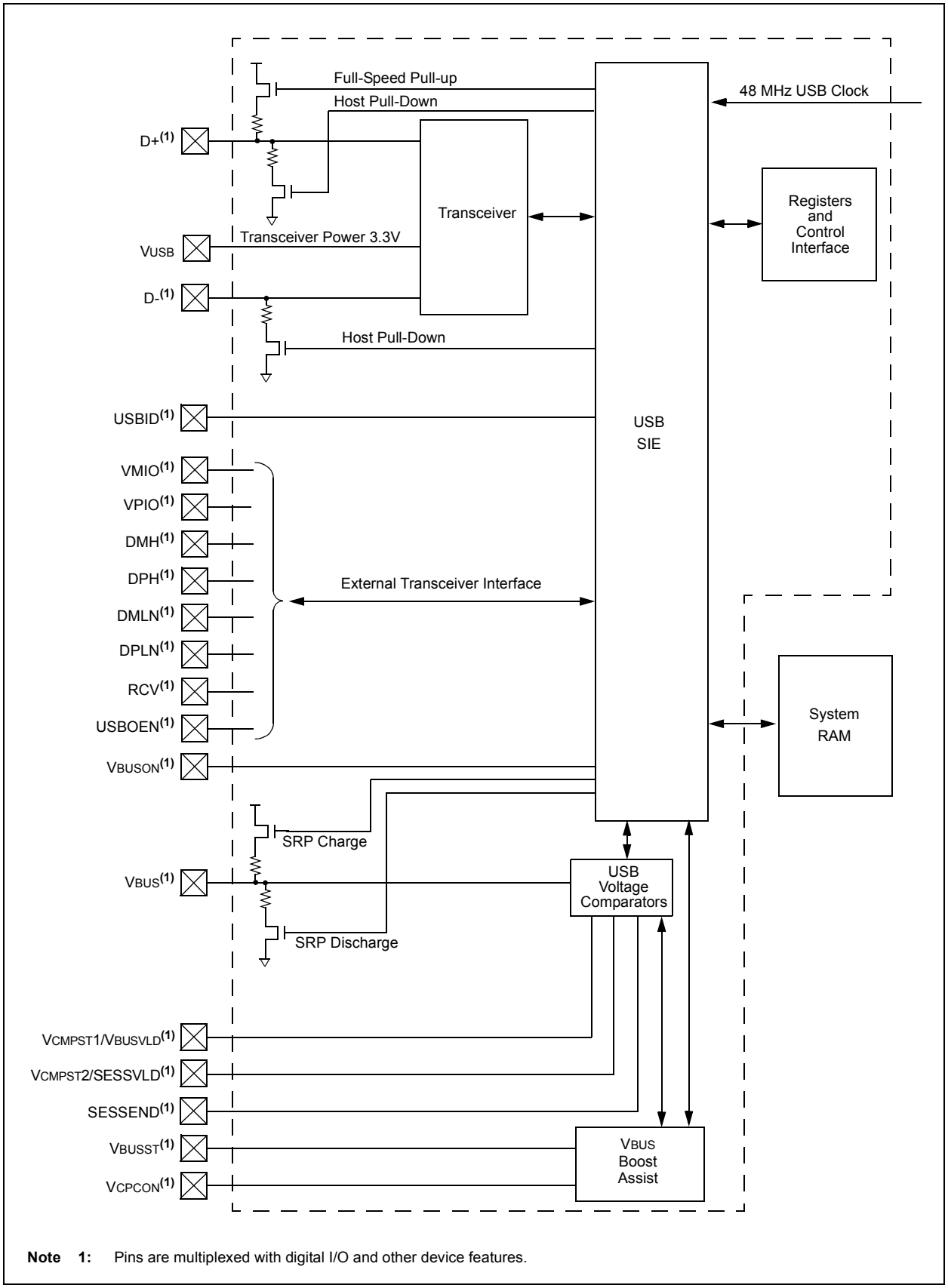
TABLE 18-1: CONTROLLER-CENTRIC DATA DIRECTION FOR USB HOST OR TARGET

USB Mode	Direction	
	RX	TX
Device	OUT or SETUP	IN
Host	IN	OUT or SETUP

This chapter presents the most basic operations needed to implement USB OTG functionality in an application. A complete and detailed discussion of the USB protocol and its OTG supplement are beyond the scope of this data sheet. It is assumed that the user already has a basic understanding of USB architecture and the latest version of the protocol.

Not all steps for proper USB operation (such as device enumeration) are presented here. It is recommended that application developers use an appropriate device driver to implement all of the necessary features. Microchip provides a number of application-specific resources, such as USB firmware and driver support. Refer to www.microchip.com/usb for the latest firmware and driver support.

FIGURE 18-1: USB OTG MODULE BLOCK DIAGRAM



Note 1: Pins are multiplexed with digital I/O and other device features.

18.1 Hardware Configuration

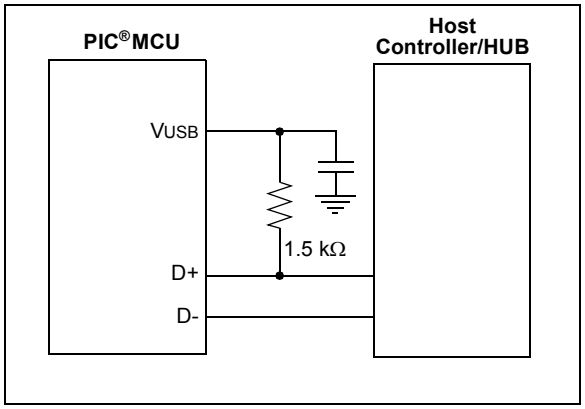
18.1.1 DEVICE MODE

18.1.1.1 D+ Pull-up Resistor

PIC24FJ256DA210 family devices have a built-in 1.5 kΩ resistor on the D+ line that is available when the microcontroller is operating in Device mode. This is used to signal an external Host that the device is operating in Full-Speed Device mode. It is engaged by setting the USBEN bit (U1CON<0>). If the OTGEN bit (U1OTGCON<2>) is set, then the D+ pull-up is enabled through the DPPULUP bit (U1OTGCON<7>).

Alternatively, an external resistor may be used on D+, as shown in Figure 18-2.

FIGURE 18-2: EXTERNAL PULL-UP FOR FULL-SPEED DEVICE MODE



18.1.1.2 Power Modes

Many USB applications will likely have several different sets of power requirements and configuration. The most common power modes encountered are:

- Bus Power Only mode
- Self-Power Only mode
- Dual Power with Self-Power Dominance

Bus Power Only mode (Figure 18-3) is effectively the simplest method. All power for the application is drawn from the USB.

To meet the inrush current requirements of the USB 2.0 Specification, the total effective capacitance appearing across VBUS and ground must be no more than 10 µF.

In the USB Suspend mode, devices must consume no more than 2.5 mA from the 5V VBUS line of the USB cable. During the USB Suspend mode, the D+ or D- pull-up resistor must remain active, which will consume some of the allowed suspend current.

In Self-Power Only mode (Figure 18-4), the USB application provides its own power, with very little power being pulled from the USB. Note that an attach indication is added to indicate when the USB has been connected and the host is actively powering VBUS.

To meet compliance specifications, the USB module (and the D+ or D- pull-up resistor) should not be enabled until the host actively drives VBUS high. One of the 5.5V tolerant I/O pins may be used for this purpose.

The application should never source any current onto the 5V VBUS pin of the USB cable.

The Dual Power mode with Self-Power Dominance (Figure 18-5) allows the application to use internal power primarily, but switch to power from the USB when no internal power is available. Dual power devices must also meet all of the special requirements for inrush current and Suspend mode current previously described, and must not enable the USB module until VBUS is driven high.

FIGURE 18-3: BUS POWER ONLY

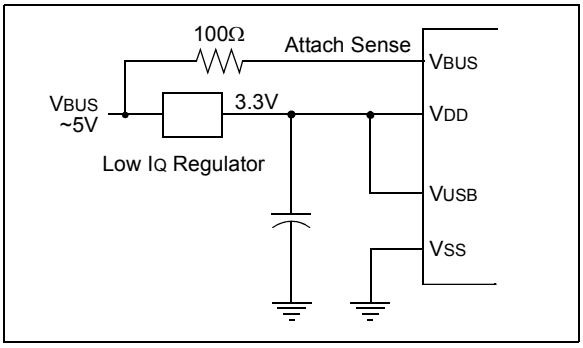


FIGURE 18-4: SELF-POWER ONLY

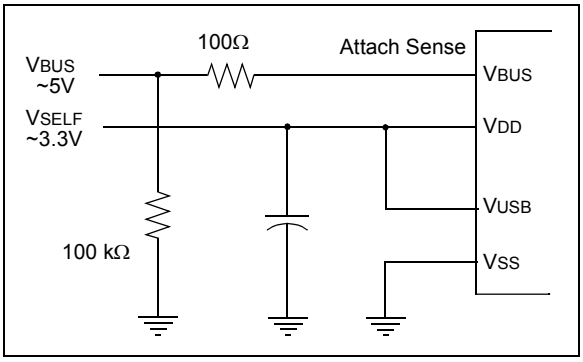
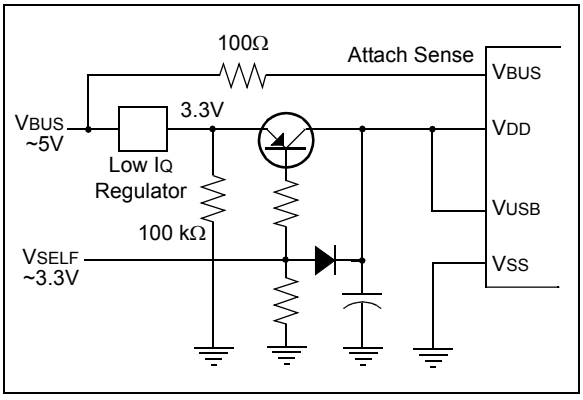


FIGURE 18-5: DUAL POWER EXAMPLE



18.1.2 HOST AND OTG MODES

18.1.2.1 D+ and D- Pull-Down Resistors

PIC24FJ256DA210 family devices have a built-in 15 kΩ pull-down resistor on the D+ and D- lines. These are used in tandem to signal to the bus that the microcontroller is operating in Host mode. They are engaged by setting the HOSTEN bit (U1CON<3>). If the OTGEN bit (U1OTGCON<2>) is set, then these pull-downs are enabled by setting the DPPULDWN and DMPULDWN bits (U1OTGCON<5:4>).

18.1.2.2 Power Configurations

In Host mode, as well as Host mode in On-The-Go operation, the USB 2.0 Specification requires that the host application should supply power on VBUS. Since the microcontroller is running below VBUS, and is not able to source sufficient current, a separate power supply must be provided.

When the application is always operating in Host mode, a simple circuit can be used to supply VBUS and regulate current on the bus (Figure 18-6). For OTG operation, it is necessary to be able to turn VBUS on or off as needed, as the microcontroller switches between Device and Host modes. A typical example using an external charge pump is shown in Figure 18-7.

FIGURE 18-6: HOST INTERFACE EXAMPLE

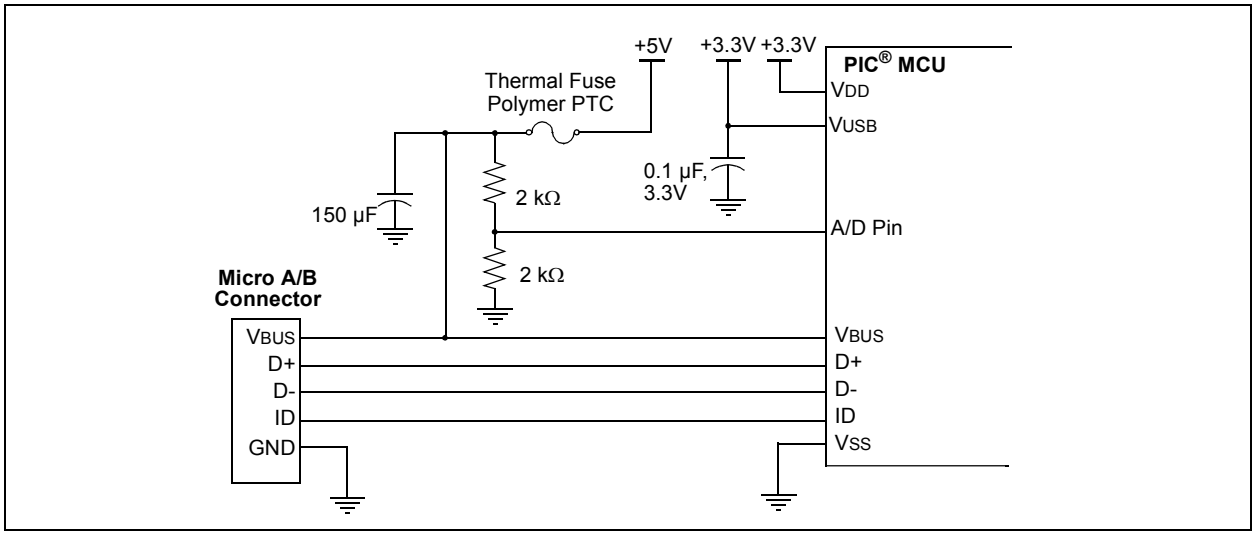
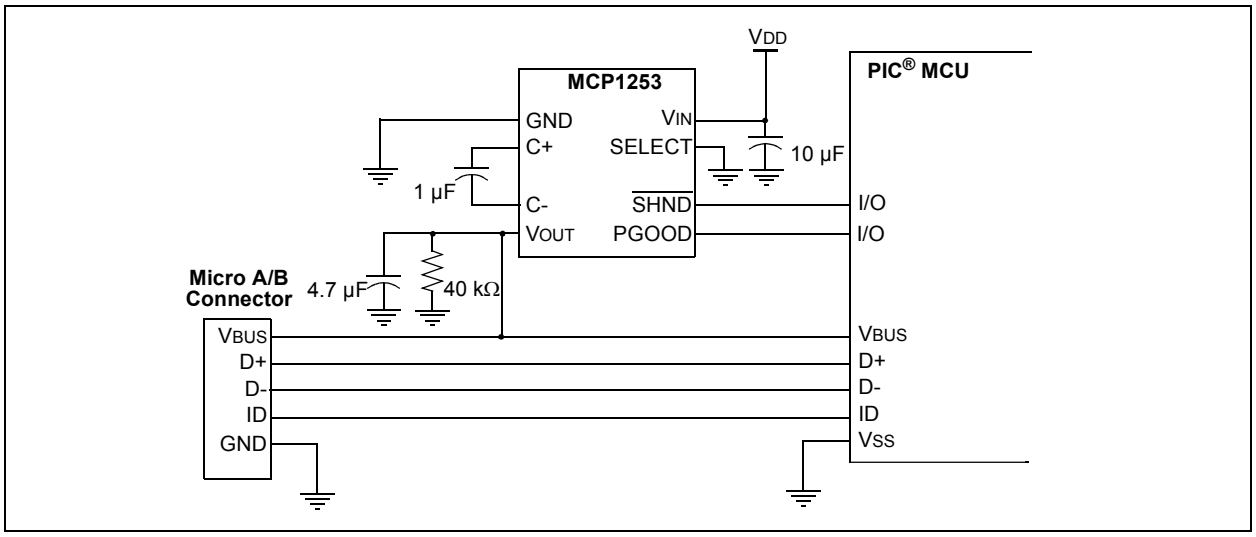


FIGURE 18-7: OTG INTERFACE EXAMPLE



18.1.2.3 VBUS Voltage Generation with External Devices

When operating as a USB host, either as an A-device in an OTG configuration or as an embedded host, VBUS must be supplied to the attached device. PIC24FJ256DA210 family devices have an internal VBUS boost assist to help generate the required 5V VBUS from the available voltages on the board. This is comprised of a simple PWM output to control a Switch mode power supply, and built-in comparators to monitor output voltage and limit current.

To enable voltage generation:

1. Verify that the USB module is powered (U1PWRC<0> = 1) and that the VBUS discharge is disabled (U1OTGCON<0> = 0).
2. Set the PWM period (U1PWMRRS<7:0>) and duty cycle (U1PWMRRS<15:8>) as required.
3. Select the required polarity of the output signal based on the configuration of the external circuit with the PWMPOL bit (U1PWMCON<9>).
4. Select the desired target voltage using the VBUSCHG bit (U1OTGCON<1>).
5. Enable the PWM counter by setting the CNTEN bit to '1' (U1PWMCON<8>).
6. Enable the PWM module by setting the PWMEN bit (U1PWMCON<15>) to '1'.
7. Enable the VBUS generation circuit (U1OTGCON<3> = 1).

Note:	This section describes the general process for VBUS voltage generation and control. Please refer to the "*PIC24F Family Reference Manual*" for additional examples.

18.1.3 USING AN EXTERNAL INTERFACE

Some applications may require the USB interface to be isolated from the rest of the system. PIC24FJ256DA210 family devices include a complete interface to communicate with and control an external USB transceiver, including the control of data line pull-ups and pull-downs. The VBUS voltage generation control circuit can also be configured for different VBUS generation topologies.

Refer to the "*PIC24F Family Reference Manual*", **Section 27. "USB On-The-Go (OTG)"** for information on using the external interface.

18.1.4 CALCULATING TRANSCEIVER POWER REQUIREMENTS

The USB transceiver consumes a variable amount of current depending on the characteristic impedance of the USB cable, the length of the cable, the VUSB supply voltage and the actual data patterns moving across the USB cable. Longer cables have larger capacitances and consume more total energy when switching output states. The total transceiver current consumption will be application-specific. Equation 18-1 can help estimate how much current actually may be required in full-speed applications.

Refer to the "*PIC24F Family Reference Manual*", **Section 27. "USB On-The-Go (OTG)"** for a complete discussion on transceiver power consumption.

EQUATION 18-1: ESTIMATING USB TRANSCEIVER CURRENT CONSUMPTION

$$IXCVR = \frac{40\ mA \bullet VUSB \bullet PZERO \bullet PIN \bullet LCABLE}{3.3V \bullet 5m} + IPULLUP$$

Legend: VUSB – Voltage applied to the VUSB pin in volts (3.0V to 3.6V).

PZERO – Percentage (in decimal) of the IN traffic bits sent by the PIC® microcontroller that are a value of '0'.

PIN – Percentage (in decimal) of total bus bandwidth that is used for IN traffic.

LCABLE – Length (in meters) of the USB cable. The USB 2.0 Specification requires that full-speed applications use cables no longer than 5m.

IPULLUP – Current which the nominal, 1.5 kΩ pull-up resistor (when enabled) must supply to the USB cable.

18.2 USB Buffer Descriptors and the BDT

Endpoint buffer control is handled through a structure called the Buffer Descriptor Table (BDT). This provides a flexible method for users to construct and control endpoint buffers of various lengths and configurations.

The BDT can be located in any available, 512-byte aligned block of data RAM. The BDT Pointer (U1BDTP1) contains the upper address byte of the BDT and sets the location of the BDT in RAM. The user must set this pointer to indicate the table's location.

The BDT is composed of Buffer Descriptors (BDs) which are used to define and control the actual buffers in the USB RAM space. Each BD consists of two, 16-bit "soft" (non-fixed-address) registers, BDnSTAT and BDnADR, where n represents one of the 64 possible BDs (range of 0 to 63). BDnSTAT is the status register for BDn, while BDnADR specifies the starting address for the buffer associated with BDn.

Note:	Since BDnADR is a 16-bit register, only the first 64 Kbytes of RAM can be accessed by the USB module.

Depending on the endpoint buffering configuration used, there are up to 64 sets of Buffer Descriptors, for a total of 256 bytes. At a minimum, the BDT must be at least 8 bytes long. This is because the USB Specification mandates that every device must have Endpoint 0 with both input and output for initial setup.

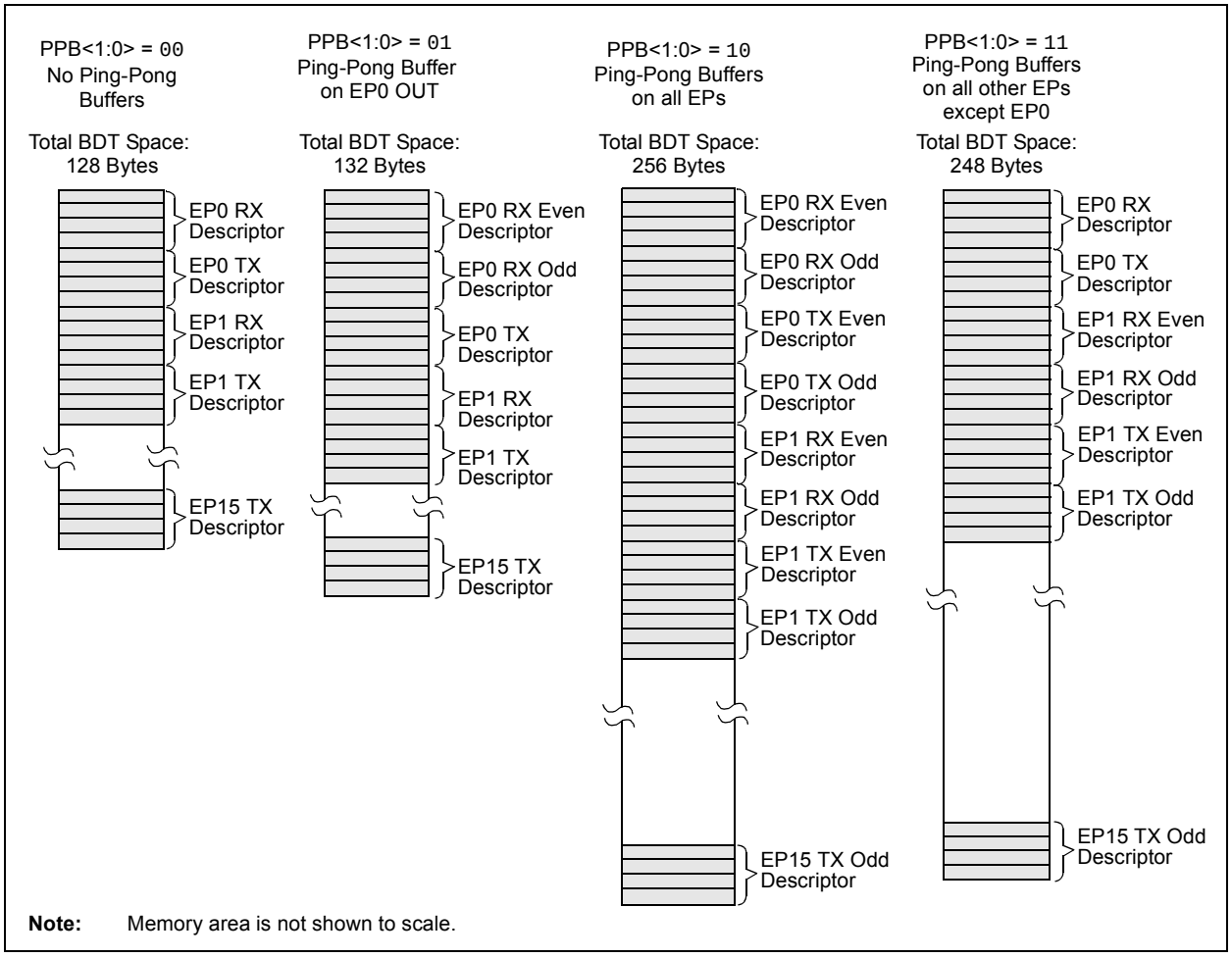
Endpoint mapping in the BDT is dependent on three variables:

- Endpoint number (0 to 15)
- Endpoint direction (RX or TX)
- Ping-pong settings (U1CNFG1<1:0>)

Figure 18-8 illustrates how these variables are used to map endpoints in the BDT.

In Host mode, only Endpoint 0 Buffer Descriptors are used. All transfers utilize the Endpoint 0 Buffer Descriptor and Endpoint Control register (U1EP0). For received packets, the attached device's source endpoint is indicated by the value of ENDPT<3:0> in the USB status register (U1STAT<7:4>). For transmitted packet, the attached device's destination endpoint is indicated by the value written to the Token register (U1TOK).

FIGURE 18-8: BDT MAPPING FOR ENDPOINT BUFFERING MODES



Note: Memory area is not shown to scale.

BDs have a fixed relationship to a particular endpoint, depending on the buffering configuration. Table 18-2 provides the mapping of BDs to endpoints. This relationship also means that gaps may occur in the BDT if endpoints are not enabled contiguously. This, theoretically, means that the BDs for disabled endpoints could be used as buffer space. In practice, users should avoid using such spaces in the BDT unless a method of validating BD addresses is implemented.

18.2.1 BUFFER OWNERSHIP

Because the buffers and their BDs are shared between the CPU and the USB module, a simple semaphore mechanism is used to distinguish which is allowed to update the BD and associated buffers in memory. This is done by using the UOWN bit as a semaphore to distinguish which is allowed to update the BD and associated buffers in memory. UOWN is the only bit that is shared between the two configurations of BDnSTAT.

When UOWN is clear, the BD entry is “owned” by the microcontroller core. When the UOWN bit is set, the BD entry and the buffer memory are “owned” by the USB peripheral. The core should not modify the BD or its corresponding data buffer during this time. Note that the microcontroller core can still read BDnSTAT while the SIE owns the buffer and vice versa.

The Buffer Descriptors have a different meaning based on the source of the register update. Register 18-1 and Register 18-2 show the differences in BDnSTAT depending on its current “ownership”.

When UOWN is set, the user can no longer depend on the values that were written to the BDs. From this point, the USB module updates the BDs as necessary, overwriting the original BD values. The BDnSTAT register is updated by the SIE with the token PID and the transfer count is updated.

18.2.2 DMA INTERFACE

The USB OTG module uses a dedicated DMA to access both the BDT and the endpoint data buffers. Since part of the address space of the DMA is dedicated to the Buffer Descriptors, a portion of the memory connected to the DMA must comprise a contiguous address space properly mapped for the access by the module.

TABLE 18-2: ASSIGNMENT OF BUFFER DESCRIPTORS FOR THE DIFFERENT BUFFERING MODES

Endpoint	BDs Assigned to Endpoint							
	Mode 0 (No Ping-Pong)		Mode 1 (Ping-Pong on EP0 OUT)		Mode 2 (Ping-Pong on all EPs)		Mode 3 (Ping-Pong on all other EPs, except EP0)	
	Out	In	Out	In	Out	In	Out	In
0	0	1	0 (E), 1 (O)	2	0 (E), 1 (O)	2 (E), 3 (O)	0	1
1	2	3	3	4	4 (E), 5 (O)	6 (E), 7 (O)	2 (E), 3 (O)	4 (E), 5 (O)
2	4	5	5	6	8 (E), 9 (O)	10 (E), 11 (O)	6 (E), 7 (O)	8 (E), 9 (O)
3	6	7	7	8	12 (E), 13 (O)	14 (E), 15 (O)	10 (E), 11 (O)	12 (E), 13 (O)
4	8	9	9	10	16 (E), 17 (O)	18 (E), 19 (O)	14 (E), 15 (O)	16 (E), 17 (O)
5	10	11	11	12	20 (E), 21 (O)	22 (E), 23 (O)	18 (E), 19 (O)	20 (E), 21 (O)
6	12	13	13	14	24 (E), 25 (O)	26 (E), 27 (O)	22 (E), 23 (O)	24 (E), 25 (O)
7	14	15	15	16	28 (E), 29 (O)	30 (E), 31 (O)	26 (E), 27 (O)	28 (E), 29 (O)
8	16	17	17	18	32 (E), 33 (O)	34 (E), 35 (O)	30 (E), 31 (O)	32 (E), 33 (O)
9	18	19	19	20	36 (E), 37 (O)	38 (E), 39 (O)	34 (E), 35 (O)	36 (E), 37 (O)
10	20	21	21	22	40 (E), 41 (O)	42 (E), 43 (O)	38 (E), 39 (O)	40 (E), 41 (O)
11	22	23	23	24	44 (E), 45 (O)	46 (E), 47 (O)	42 (E), 43 (O)	44 (E), 45 (O)
12	24	25	25	26	48 (E), 49 (O)	50 (E), 51 (O)	46 (E), 47 (O)	48 (E), 49 (O)
13	26	27	27	28	52 (E), 53 (O)	54 (E), 55 (O)	50 (E), 51 (O)	52 (E), 53 (O)
14	28	29	29	30	56 (E), 57 (O)	58 (E), 59 (O)	54 (E), 55 (O)	56 (E), 57 (O)
15	30	31	31	32	60 (E), 61 (O)	62 (E), 63 (O)	58 (E), 59 (O)	60 (E), 61 (O)

Legend: (E) = Even transaction buffer, (O) = Odd transaction buffer

REGISTER 18-1: BDnSTAT: BUFFER DESCRIPTOR n STATUS REGISTER PROTOTYPE, USB MODE (BD0STAT THROUGH BD63STAT)

R/W-x	R/W-x	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC
UOWN	DTS	PID3	PID2	PID1	PID0	BC9	BC8
bit 15							bit 8

R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC
BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0
bit 7							bit 0

Legend:	HSC = Hardware Settable/Clearable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **UOWN:** USB Own bit
1 = The USB module owns the BD and its corresponding buffer; the CPU must not modify the BD or the buffer

bit 14 **DTS:** Data Toggle Packet bit
1 = Data 1 packet
0 = Data 0 packet

bit 13-10 **PID<3:0>:** Packet Identifier bits (written by the USB module)
In Device mode:
Represents the PID of the received token during the last transfer.
In Host mode:
Represents the last returned PID or the transfer status indicator.

bit 9-0 **BC<9:0>:** Byte Count bits
This represents the number of bytes to be transmitted or the maximum number of bytes to be received during a transfer. Upon completion, the byte count is updated by the USB module with the actual number of bytes transmitted or received.

REGISTER 18-2: BDnSTAT: BUFFER DESCRIPTOR n STATUS REGISTER PROTOTYPE, CPU MODE (BD0STAT THROUGH BD63STAT)

R/W-x	R/W-x	r-0	r-0	R/W-x	R/W-x	R/W-x, HSC	R/W-x, HSC
UOWN	DTS(1)	Reserved	Reserved	DTSEN	BSTALL	BC9	BC8
bit 15							bit 8

R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC
BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0
bit 7							bit 0

Legend:	HSC = Hardware Settable/Clearable bit		r = Reserved bit
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'r' = Reserved bit	x = Bit is unknown

bit 15 **UOWN:** USB Own bit
- 0 = The microcontroller core owns the BD and its corresponding buffer; the USB module ignores all other fields in the BD

bit 14 **DTS:** Data Toggle Packet bit(1)
- 1 = Data 1 packet
- 0 = Data 0 packet

bit 13-12 **Reserved:** Maintain as '0'

bit 11 **DTSEN:** Data Toggle Synchronization Enable bit
- 1 = Data toggle synchronization is enabled; data packets with incorrect sync value will be ignored
- 0 = No data toggle synchronization is performed

bit 10 **BSTALL:** Buffer Stall Enable bit
- 1 = Buffer STALL enabled; STALL handshake issued if a token is received that would use the BD in the given location (UOWN bit remains set, BD value is unchanged); corresponding EPSTALL bit will get set on any STALL handshake
- 0 = Buffer STALL disabled

bit 9-0 **BC<9:0>:** Byte Count bits

This represents the number of bytes to be transmitted or the maximum number of bytes to be received during a transfer. Upon completion, the byte count is updated by the USB module with the actual number of bytes transmitted or received.

Note 1: This bit is ignored unless DTSEN = 1.

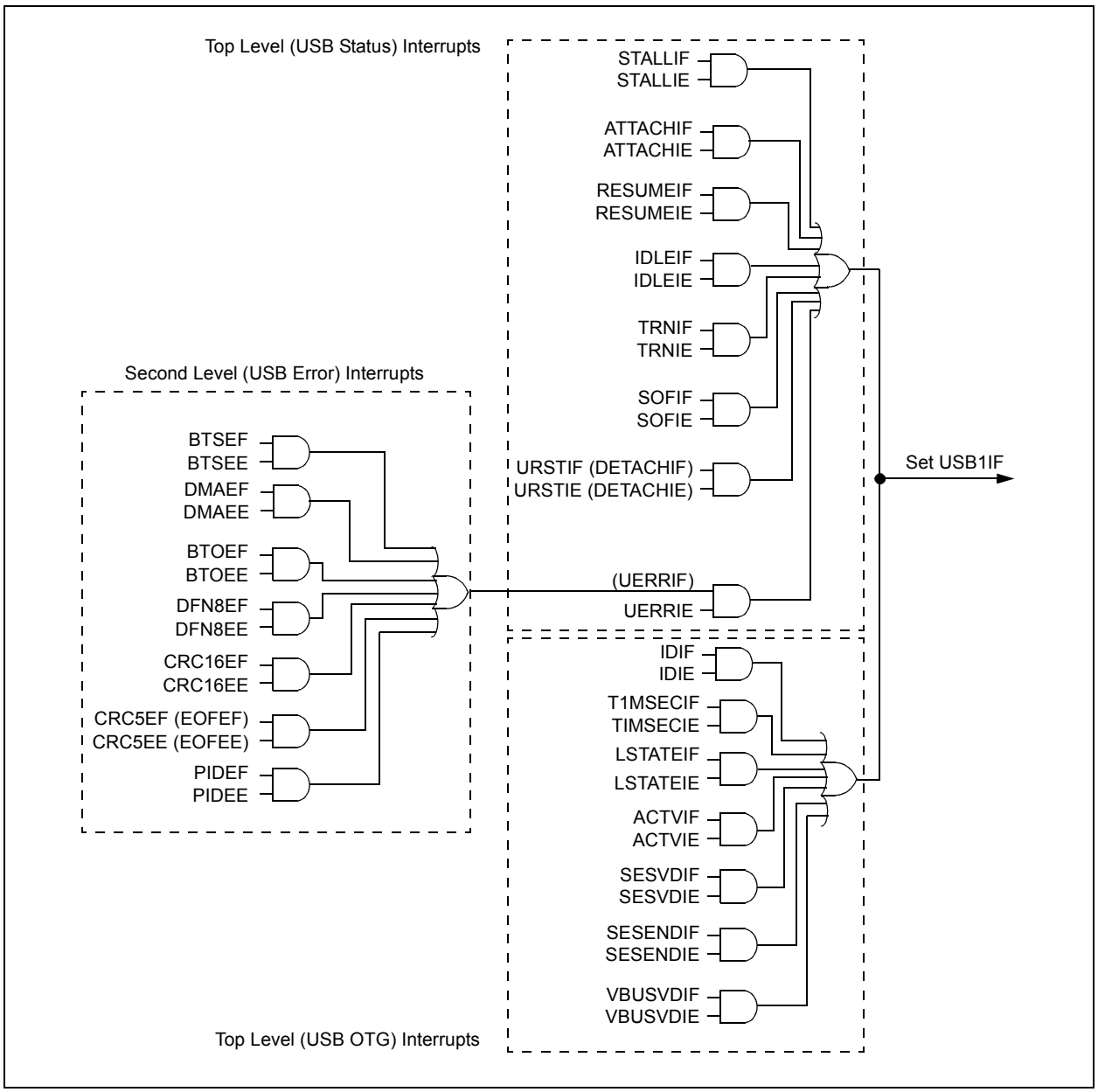
18.3 USB Interrupts

The USB OTG module has many conditions that can be configured to cause an interrupt. All interrupt sources use the same interrupt vector.

Figure 18-9 shows the interrupt logic for the USB module. There are two layers of interrupt registers in the USB module. The top level consists of overall USB status interrupts; these are enabled and flagged in the U1IE and U1IR registers, respectively. The second level consists of USB error conditions, which are enabled and flagged in the U1EIR and U1EIE registers. An interrupt condition in any of these triggers a USB Error Interrupt Flag (UERRIF) in the top level.

Interrupts may be used to trap routine events in a USB transaction. Figure 18-10 provides some common events within a USB frame and their corresponding interrupts.

FIGURE 18-9: USB OTG INTERRUPT FUNNEL

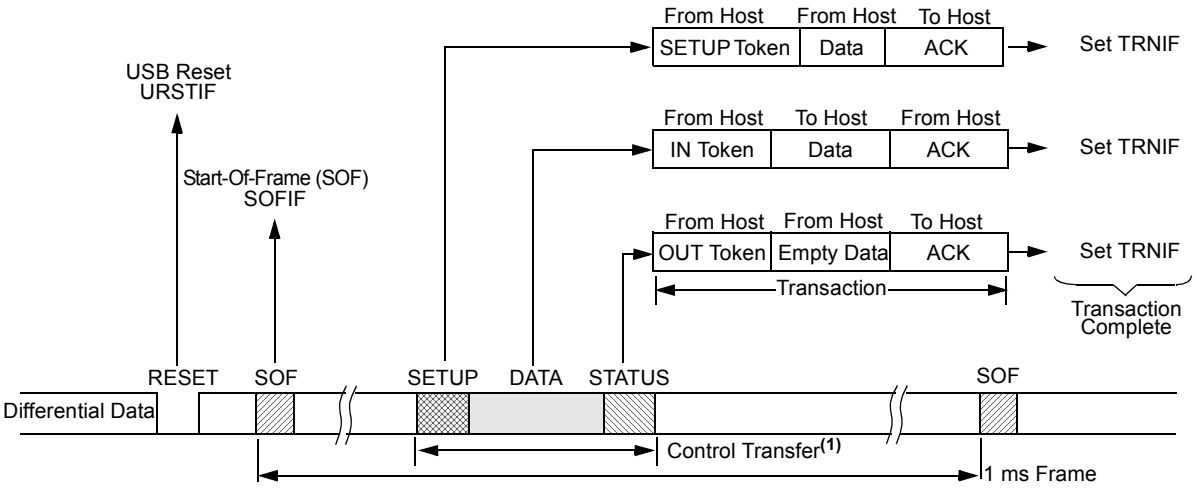


18.3.1 CLEARING USB OTG INTERRUPTS

Unlike device level interrupts, the USB OTG interrupt status flags are not freely writable in software. All USB OTG flag bits are implemented as hardware set only bits. Additionally, these bits can only be cleared in software by writing a '1' to their locations (i.e., performing a MOV type instruction). Writing a '0' to a flag bit (i.e., a BCLR instruction) has no effect.

Note:	Throughout this data sheet, a bit that can only be cleared by writing a '1' to its location is referred to as "Write 1 to clear". In register descriptions, this function is indicated by the descriptor, "K".

FIGURE 18-10: EXAMPLE OF A USB TRANSACTION AND INTERRUPT EVENTS



Note 1: The control transfer shown here is only an example showing events that can occur for every transaction. Typical control transfers will spread across multiple frames.

18.4 Device Mode Operation

The following section describes how to perform a common Device mode task. In Device mode, USB transfers are performed at the transfer level. The USB module automatically performs the status phase of the transfer.

18.4.1 ENABLING DEVICE MODE

1. Reset the Ping-Pong Buffer Pointers by setting, then clearing, the Ping-Pong Buffer Reset bit, PPBRST (U1CON<1>).
2. Disable all interrupts (U1IE and U1EIE = 00h).
3. Clear any existing interrupt flags by writing FFh to U1IR and U1EIR.
4. Verify that VBUS is present (non OTG devices only).
5. Enable the USB module by setting the USBEN bit (U1CON<0>).
6. Set the OTGEN bit (U1OTGCON<2>) to enable OTG operation.
7. Enable the endpoint zero buffer to receive the first setup packet by setting the EPRXEN and EPHSHK bits for Endpoint 0 (U1EP0<3,0> = 1).
8. Power up the USB module by setting the USBPWR bit (U1PWRC<0>).
9. Enable the D+ pull-up resistor to signal an attach by setting DPPULUP bit (U1OTGCON<7>).

18.4.2 RECEIVING AN IN TOKEN IN DEVICE MODE

1. Attach to a USB host and enumerate as described in *"Chapter 9 of the USB 2.0 Specification"*.
2. Create a data buffer and populate it with the data to send to the host.
3. In the appropriate (even or odd) TX BD for the desired endpoint:
   a) Set up the status register (BDnSTAT) with the correct data toggle (DATA0/1) value and the byte count of the data buffer.
   b) Set up the address register (BDnADR) with the starting address of the data buffer.
   c) Set the UOWN bit of the status register to '1'.
4. When the USB module receives an IN token, it automatically transmits the data in the buffer. Upon completion, the module updates the status register (BDnSTAT) and sets the Transfer Complete Interrupt Flag, TRNIF (U1IR<3>).

18.4.3 RECEIVING AN OUT TOKEN IN DEVICE MODE

1. Attach to a USB host and enumerate as described in *"Chapter 9 of the USB 2.0 Specification"*.
2. Create a data buffer with the amount of data you are expecting from the host.
3. In the appropriate (even or odd) TX BD for the desired endpoint:
   a) Set up the status register (BDnSTAT) with the correct data toggle (DATA0/1) value and the byte count of the data buffer.
   b) Set up the address register (BDnADR) with the starting address of the data buffer.
   c) Set the UOWN bit of the status register to '1'.
4. When the USB module receives an OUT token, it automatically receives the data sent by the host to the buffer. Upon completion, the module updates the status register (BDnSTAT) and sets the Transfer Complete Interrupt Flag, TRNIF (U1IR<3>).

18.5 Host Mode Operation

The following sections describe how to perform common Host mode tasks. In Host mode, USB transfers are invoked explicitly by the host software. The host software is responsible for the Acknowledge portion of the transfer. Also, all transfers are performed using the Endpoint 0 Control register (U1EP0) and Buffer Descriptors.

18.5.1 ENABLE HOST MODE AND DISCOVER A CONNECTED DEVICE

1. Enable Host mode by setting the HOSTEN bit (U1CON<3>). This causes the Host mode control bits in other USB OTG registers to become available.
2. Enable the D+ and D- pull-down resistors by setting the DPPULDWN and DMPULDWN bits (U1OTGCON<5:4>). Disable the D+ and D- pull-up resistors by clearing the DPPULUP and DMPULUP bits (U1OTGCON<7:6>).
3. At this point, SOF generation begins with the SOF counter loaded with 12,000. Eliminate noise on the USB by clearing the SOFEN bit (U1CON<0>) to disable Start-Of-Frame packet generation.
4. Enable the device attached interrupt by setting the ATTACHIE bit (U1IE<6>).
5. Wait for the device attached interrupt (U1IR<6> = 1). This is signaled by the USB device changing the state of D+ or D- from '0' to '1' (SE0 to J state). After it occurs, wait 100 ms for the device power to stabilize.
6. Check the state of the JSTATE and SE0 bits in U1CON. If the JSTATE bit (U1CON<7>) is '0', the connecting device is low speed. If the connecting device is low speed, set the low LSPDEN and LSPD bits (U1ADDR<7> and U1EP0<7>) to enable low-speed operation.
7. Reset the USB device by setting the USBRST bit (U1CON<4>) for at least 50 ms, sending Reset signaling on the bus. After 50 ms, terminate the Reset by clearing USBRST.
8. In order to keep the connected device from going into suspend, enable the SOF packet generation by setting the SOFEN bit.
9. Wait 10 ms for the device to recover from Reset.
10. Perform enumeration as described by *"Chapter 9 of the USB 2.0 Specification"*.

18.5.2 COMPLETE A CONTROL TRANSACTION TO A CONNECTED DEVICE

1. Follow the procedure described in **Section 18.5.1 "Enable Host Mode and Discover a Connected Device"** to discover a device.
2. Set up the Endpoint Control register for bidirectional control transfers by writing 0Dh to U1EP0 (this sets the EPCONDIS, EPTXEN and EPHSHK bits).
3. Place a copy of the device framework setup command in a memory buffer. See *"Chapter 9 of the USB 2.0 Specification"* for information on the device framework command set.
4. Initialize the Buffer Descriptor (BD) for the current (even or odd) TX EP0 to transfer the eight bytes of command data for a device framework command (i.e., `GET DEVICE DESCRIPTOR`):
   a) Set the BD data buffer address (BD0ADR) to the starting address of the 8-byte memory buffer containing the command.
   b) Write 8008h to BD0STAT (this sets the UOWN bit and sets a byte count of 8).
5. Set the USB device address of the target device in the address register (U1ADDR<6:0>). After a USB bus Reset, the device USB address will be zero. After enumeration, it will be set to another value between 1 and 127.
6. Write D0h to U1TOK; this is a SETUP token to Endpoint 0, the target device's default control pipe. This initiates a SETUP token on the bus, followed by a data packet. The device handshake is returned in the PID field of BD0STAT after the packets are complete. When the USB module updates BD0STAT, a transfer done interrupt is asserted (the TRNIF flag is set). This completes the setup phase of the setup transaction as referenced in *"Chapter 9 of the USB Specification"*.
7. To initiate the data phase of the setup transaction (i.e., get the data for the `GET DEVICE DESCRIPTOR` command), set up a buffer in memory to store the received data.
8. Initialize the current (even or odd) RX or TX (RX for IN, TX for OUT) EP0 BD to transfer the data.
   a) Write C040h to BD0STAT. This sets the UOWN, configures Data Toggle (DTS) to DATA1 and sets the byte count to the length of the data buffer (64 or 40h in this case).
   b) Set BD0ADR to the starting address of the data buffer.
9. Write the Token register with the appropriate IN or OUT token to Endpoint 0, the target device's default control pipe (e.g., write 90h to U1TOK for an IN token for a `GET DEVICE DESCRIPTOR` command). This initiates an IN token on the bus followed by a data packet from the device to the host. When the data packet completes, the BD0STAT is written and a transfer done interrupt is asserted (the TRNIF flag is set). For control transfers with a single packet data phase, this completes the data phase of the setup transaction as referenced in *"Chapter 9 of the USB Specification"*. If more data needs to be transferred, return to step 8.
10. To initiate the status phase of the setup transaction, set up a buffer in memory to receive or send the zero length status phase data packet.
11. Initialize the current (even or odd) TX EP0 BD to transfer the status data:
    a) Set the BDT buffer address field to the start address of the data buffer.
    b) Write 8000h to BD0STAT (set UOWN bit, configure DTS to DATA0 and set byte count to 0).
12. Write the Token register with the appropriate IN or OUT token to Endpoint 0, the target device's default control pipe (e.g., write 01h to U1TOK for an OUT token for a `GET DEVICE DESCRIPTOR` command). This initiates an OUT token on the bus followed by a zero length data packet from the host to the device. When the data packet completes, the BD is updated with the handshake from the device and a transfer done interrupt is asserted (the TRNIF flag is set). This completes the status phase of the setup transaction as described in *"Chapter 9 of the USB Specification"*.

Note:	Only one control transaction can be performed per frame.

18.5.3 SEND A FULL-SPEED BULK DATA TRANSFER TO A TARGET DEVICE

1. Follow the procedure described in **Section 18.5.1 "Enable Host Mode and Discover a Connected Device"** and **Section 18.5.2 "Complete a Control Transaction to a Connected Device"** to discover and configure a device.
2. To enable transmit and receive transfers with handshaking enabled, write 1Dh to U1EP0. If the target device is a low-speed device, also set the LSPD (U1EP0<7>) bit. If you want the hardware to automatically retry indefinitely if the target device asserts a NAK on the transfer, clear the Retry Disable bit, RETRYDIS (U1EP0<6>).
3. Set up the BD for the current (even or odd) TX EP0 to transfer up to 64 bytes.
4. Set the USB device address of the target device in the address register (U1ADDR<6:0>).
5. Write an OUT token to the desired endpoint to U1TOK. This triggers the module's transmit state machines to begin transmitting the token and the data.
6. Wait for the Transfer Done Interrupt Flag, TRNIF. This indicates that the BD has been released back to the microprocessor and the transfer has completed. If the retry disable bit is set, the handshake (ACK, NAK, STALL or ERROR (0Fh)) is returned in the BD PID field. If a STALL interrupt occurs, the pending packet must be dequeued and the error condition in the target device cleared. If a detach interrupt occurs (SE0 for more than 2.5 µs), then the target has detached (U1IR<0> is set).
7. Once the transfer done interrupt occurs (TRNIF is set), the BD can be examined and the next data packet queued by returning to step 2.

Note:	USB speed, transceiver and pull-ups should only be configured during the module setup phase. It is not recommended to change these settings while the module is enabled.

18.6 OTG Operation

18.6.1 SESSION REQUEST PROTOCOL (SRP)

An OTG A-device may decide to power down the VBUS supply when it is not using the USB link through the Session Request Protocol (SRP). Software may do this by clearing VBUSON (U1OTGCON<3>). When the VBUS supply is powered down, the A-device is said to have ended a USB session.

An OTG A-device or embedded host may repower the VBUS supply at any time (initiate a new session). An OTG B-device may also request that the OTG A-device repower the VBUS supply (initiate a new session). This is accomplished via Session Request Protocol (SRP).

Prior to requesting a new session, the B-device must first check that the previous session has definitely ended. To do this, the B-device must check for two conditions:

1. VBUS supply is below the session valid voltage, and

2. Both D+ and D- have been low for at least 2 ms.

The B-device will be notified of Condition 1 by the SESENDIF (U1OTGIR<2>) interrupt. Software will have to manually check for Condition 2.

Note:	When the A-device powers down the VBUS supply, the B-device must disconnect its pull-up resistor from power. If the device is self-powered, it can do this by clearing DPPULUP (U1OTGCON<7>) and DMPULUP (U1OTGCON<6>).

The B-device may aid in achieving Condition 1 by discharging the VBUS supply through a resistor. Software may do this by setting VBUSDIS (U1OTGCON<0>).

After these initial conditions are met, the B-device may begin requesting the new session. The B-device begins by pulsing the D+ data line. Software should do this by setting DPPULUP (U1OTGCON<7>). The data line should be held high for 5 to 10 ms.

The B-device then proceeds by pulsing the VBUS supply. Software should do this by setting PUVBUS (U1CNFG2<4>). When an A-device detects SRP signaling (either via the ATTACHIF (U1IR<6>) interrupt or via the SESVDIF (U1OTGIR<3>) interrupt), the A-device must restore the VBUS supply by either setting VBUSON (U1OTGCON<3>) or by setting the I/O port controlling the external power source.

The B-device should not monitor the state of the VBUS supply while performing VBUS supply pulsing. When the B-device does detect that the VBUS supply has been restored (via the SESVDIF (U1OTGIR<3>) interrupt), the B-device must reconnect to the USB link by pulling up D+ or D- (via the DPPULUP or DMPULUP).

The A-device must complete the SRP by driving USB Reset signaling.

18.6.2 HOST NEGOTIATION PROTOCOL (HNP)

In USB OTG applications, a Dual Role Device (DRD) is a device that is capable of being either a host or a peripheral. Any OTG DRD must support Host Negotiation Protocol (HNP).

HNP allows an OTG B-device to temporarily become the USB host. The A-device must first enable the B-device to follow HNP. Refer to the *"On-The-Go Supplement to the USB 2.0 Specification"* for more information regarding HNP. HNP may only be initiated at full speed.

After being enabled for HNP by the A-device, the B-device requests being the host any time that the USB link is in suspend state, by simply indicating a disconnect. This can be done in software by clearing DPPULUP and DMPULUP. When the A-device detects the disconnect condition (via the URSTIF (U1IR<0>) interrupt), the A-device may allow the B-device to take over as host. The A-device does this by signaling connect as a full-speed function. Software may accomplish this by setting DPPULUP.

If the A-device responds instead with resume signaling, the A-device remains as host. When the B-device detects the connect condition (via ATTACHIF (U1IR<6>), the B-device becomes host. The B-device drives Reset signaling prior to using the bus.

When the B-device has finished in its role as host, it stops all bus activity and turns on its D+ pull-up resistor by setting DPPULUP. When the A-device detects a suspend condition (Idle for 3 ms), the A-device turns off its D+ pull-up. The A-device may also power-down the VBUS supply to end the session. When the A-device detects the connect condition (via ATTACHIF), the A-device resumes host operation and drives Reset signaling.

18.6.3 EXTERNAL VBUS COMPARATORS

The external VBUS comparator option is enabled by setting the UVCMPDIS bit (U1CNFG2<1>). This disables the internal VBUS comparators, removing the need to attach VBUS to the microcontroller's VBUS pin.

The external comparator interface uses either the VCMPST1 and VCMPST2 pins, or the VBUSVLD, SESSVLD and SESSEND pins, based upon the setting of the UVCMPSEL bit (U1CNFG2<5>). These pins are digital inputs and should be set in the following patterns (see Table 18-3), based on the current level of the VBUS voltage.

TABLE 18-3: EXTERNAL VBUS COMPARATOR STATES

If UVCMPSEL = 0			
VCMPST1	VCMPST2	Bus Condition	
0	0	VBUS < VB_SESS_END	
1	0	VB_SESS_END < VBUS < VA_SESS_VLD	
0	1	VA_SESS_VLD < VBUS < VA_VBUS_VLD	
1	1	VBUS > VBUS_VLD	
If UVCMPSEL = 1			
VBUSVLD	SESSVLD	SESSEND	Bus Condition
0	0	1	VBUS < VB_SESS_END
0	0	0	VB_SESS_END < VBUS < VA_SESS_VLD
0	1	0	VA_SESS_VLD < VBUS < VA_VBUS_VLD
1	1	0	VBUS > VBUS_VLD

18.7 USB OTG Module Registers

There are a total of 37 memory mapped registers associated with the USB OTG module. They can be divided into four general categories:

- USB OTG Module Control (12)
- USB Interrupt (7)
- USB Endpoint Management (16)
- USB VBUS Power Control (2)

This total does not include the (up to) 128 BD registers in the BDT. Their prototypes, described in Register 18-1 and Register 18-2, are shown separately in **Section 18.2 “USB Buffer Descriptors and the BDT”**.

With the exception of U1PWMCON and U1PWMRRS, all USB OTG registers are implemented in the Least Significant Byte of the register. Bits in the upper byte are unimplemented and have no function. Note that some registers are instantiated only in Host mode, while other registers have different bit instantiations and functions in Device and Host modes.

The registers described in the following sections are those that have bits with specific control and configuration features. The following registers are used for data or address values only:

- U1BDTP1: Specifies the 256-word page in data RAM used for the BDT; 8-bit value with bit 0 fixed as ‘0’ for boundary alignment.
- U1FRML and U1FRMH: Contains the 11-bit byte counter for the current data frame.
- U1PWMRRS: Contains the 8-bit value for PWM duty cycle bits<15:8> and PWM period bits<7:0> for the VBUS boost assist PWM module.

18.7.1 USB OTG MODULE CONTROL REGISTERS

REGISTER 18-3: U1OTGSTAT: USB OTG STATUS REGISTER (HOST MODE ONLY)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R-0, HSC	U-0	R-0, HSC	U-0	R-0, HSC	R-0, HSC	U-0	R-0, HSC
ID	—	LSTATE	—	SESVD	SESEND	—	VBUSVD
bit 7							bit 0

Legend:	U = Unimplemented bit, read as '0'		
R = Readable bit	W = Writable bit	HSC = Hardware Settable/Clearable bit	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as '0'

bit 7 **ID:** ID Pin State Indicator bit
1 = No plug is attached, or a type B cable has been plugged into the USB receptacle
0 = A type A plug has been plugged into the USB receptacle

bit 6 **Unimplemented:** Read as '0'

bit 5 **LSTATE:** Line State Stable Indicator bit
1 = The USB line state (as defined by SE0 and JSTATE) has been stable for the previous 1 ms
0 = The USB line state has not been stable for the previous 1 ms

bit 4 **Unimplemented:** Read as '0'

bit 3 **SESVD:** Session Valid Indicator bit
1 = The VBUS voltage is above VA_SESS_VLD (as defined in the USB OTG Specification) on the A or B-device
0 = The VBUS voltage is below VA_SESS_VLD on the A or B-device

bit 2 **SESEND:** B Session End Indicator bit
1 = The VBUS voltage is below VB_SESS_END (as defined in the USB OTG Specification) on the B-device
0 = The VBUS voltage is above VB_SESS_END on the B-device

bit 1 **Unimplemented:** Read as '0'

bit 0 **VBUSVD:** A VBUS Valid Indicator bit
1 = The VBUS voltage is above VA_VBUS_VLD (as defined in the USB OTG Specification) on the A-device
0 = The VBUS voltage is below VA_VBUS_VLD on the A-device

REGISTER 18-4: U1OTGCON: USB ON-THE-GO CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DPPULUP	DMPULUP	DPPULDWN(1)	DMPULDWN(1)	VBUSON(1)	OTGEN(1)	VBUSCHG(1)	VBUSDIS(1)
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as '0'

bit 7 **DPPULUP:** D+ Pull-up Enable bit
1 = D+ data line pull-up resistor is enabled
0 = D+ data line pull-up resistor is disabled

bit 6 **DMPULUP:** D- Pull-up Enable bit
1 = D- data line pull-up resistor is enabled
0 = D- data line pull-up resistor is disabled

bit 5 **DPPULDWN:** D+ Pull-Down Enable bit(1)
1 = D+ data line pull-down resistor is enabled
0 = D+ data line pull-down resistor is disabled

bit 4 **DMPULDWN:** D- Pull-Down Enable bit(1)
1 = D- data line pull-down resistor is enabled
0 = D- data line pull-down resistor is disabled

bit 3 **VBUSON:** VBUS Power-on bit(1)
1 = VBUS line is powered
0 = VBUS line is not powered

bit 2 **OTGEN:** OTG Features Enable bit(1)
1 = USB OTG is enabled; all D+/D- pull-up and pull-down bits are enabled
0 = USB OTG is disabled; D+/D- pull-up and pull-down bits are controlled in hardware by the settings of the HOSTEN and USBEN (U1CON<3,0>) bits

bit 1 **VBUSCHG:** VBUS Charge Select bit(1)
1 = VBUS line is set to charge to 3.3V
0 = VBUS line is set to charge to 5V

bit 0 **VBUSDIS:** VBUS Discharge Enable bit(1)
1 = VBUS line is discharged through a resistor
0 = VBUS line is not discharged

Note 1: These bits are only used in Host mode; do not use in Device mode.

REGISTER 18-5: U1PWRC: USB POWER CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0, HS	U-0	U-0	R/W-0	U-0	U-0	R/W-0, HC	R/W-0
UACTPND	—	—	USLPGRD	—	—	USUSPND	USBPWR
bit 7							bit 0

Legend:	HS = Hardware Settable bit	HC = Hardware Clearable bit	
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as '0'

bit 7 **UACTPND:** USB Activity Pending bit
1 = Module should not be suspended at the moment (requires the USLPGRD bit to be set)
0 = Module may be suspended or powered down

bit 6-5 **Unimplemented:** Read as '0'

bit 4 **USLPGRD:** Sleep/Suspend Guard bit
1 = Indicate to the USB module that it is about to be suspended or powered down
0 = No suspend

bit 3-2 **Unimplemented:** Read as '0'

bit 1 **USUSPND:** USB Suspend Mode Enable bit
1 = USB OTG module is in Suspend mode; USB clock is gated and the transceiver is placed in a low-power state
0 = Normal USB OTG operation

bit 0 **USBPWR:** USB Operation Enable bit
1 = USB OTG module is enabled
0 = USB OTG module is disabled(1)

Note 1: Do not clear this bit unless the HOSTEN, USBEN and OTGEN bits (U1CON<3,0> and U1OTGCON<2>) are all cleared.

REGISTER 18-6: U1STAT: USB STATUS REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	U-0	U-0
ENDPT3	ENDPT2	ENDPT1	ENDPT0	DIR	PPBI(1)	—	—
bit 7							bit 0

Legend:	U = Unimplemented bit, read as '0'		
R = Readable bit	W = Writable bit	HSC = Hardware Settable/Clearable bit	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as '0'

bit 7-4 **ENDPT<3:0>:** Number of the Last Endpoint Activity bits
(Represents the number of the BDT updated by the last USB transfer.)
1111 = Endpoint 15
1110 = Endpoint 14
.
.
.
0001 = Endpoint 1
0000 = Endpoint 0

bit 3 **DIR:** Last BD Direction Indicator bit
1 = The last transaction was a transmit transfer (TX)
0 = The last transaction was a receive transfer (RX)

bit 2 **PPBI:** Ping-Pong BD Pointer Indicator bit(1)
1 = The last transaction was to the odd BD bank
0 = The last transaction was to the even BD bank

bit 1-0 **Unimplemented:** Read as '0'

Note 1: This bit is only valid for endpoints with available even and odd BD registers.

REGISTER 18-7: U1CON: USB CONTROL REGISTER (DEVICE MODE)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	R-x, HSC	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	SE0	PKTDIS	—	HOSTEN	RESUME	PPBRST	USBEN
bit 7							bit 0

Legend:	U = Unimplemented bit, read as '0'		
R = Readable bit	W = Writable bit	HSC = Hardware Settable/Clearable bit	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-7 **Unimplemented:** Read as '0'

bit 6 **SE0:** Live Single-Ended Zero Flag bit
1 = Single-ended zero is active on the USB bus
0 = No single-ended zero is detected

bit 5 **PKTDIS:** Packet Transfer Disable bit
1 = SIE token and packet processing are disabled; automatically set when a SETUP token is received
0 = SIE token and packet processing are enabled

bit 4 **Unimplemented:** Read as '0'

bit 3 **HOSTEN:** Host Mode Enable bit
1 = USB host capability is enabled; pull-downs on D+ and D- are activated in hardware
0 = USB host capability is disabled

bit 2 **RESUME:** Resume Signaling Enable bit
1 = Resume signaling is activated
0 = Resume signaling is disabled

bit 1 **PPBRST:** Ping-Pong Buffers Reset bit
1 = Reset all Ping-Pong Buffer Pointers to the even BD banks
0 = Ping-Pong Buffer Pointers are not reset

bit 0 **USBEN:** USB Module Enable bit
1 = USB module and supporting circuitry are enabled (device attached); D+ pull-up is activated in hardware
0 = USB module and supporting circuitry are disabled (device detached)

REGISTER 18-8: U1CON: USB CONTROL REGISTER (HOST MODE ONLY)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R-x, HSC	R-x, HSC	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
JSTATE	SE0	TOKBUSY	USBRST	HOSTEN	RESUME	PPBRST	SOFEN
bit 7							bit 0

Legend:	U = Unimplemented bit, read as '0'		
R = Readable bit	W = Writable bit	HSC = Hardware Settable/Clearable bit	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as '0'

bit 7 **JSTATE:** Live Differential Receiver J State Flag bit
1 = J state (differential '0' in low speed, differential '1' in full speed) is detected on the USB
0 = No J state is detected

bit 6 **SE0:** Live Single-Ended Zero Flag bit
1 = Single-ended zero is active on the USB bus
0 = No single-ended zero is detected

bit 5 **TOKBUSY:** Token Busy Status bit
1 = Token is being executed by the USB module in On-The-Go state
0 = No token is being executed

bit 4 **USBRST:** Module Reset bit
1 = USB Reset has been generated; for software Reset, application must set this bit for 50 ms, then clear it
0 = USB Reset is terminated

bit 3 **HOSTEN:** Host Mode Enable bit
1 = USB host capability is enabled; pull-downs on D+ and D- are activated in hardware
0 = USB host capability is disabled

bit 2 **RESUME:** Resume Signaling Enable bit
1 = Resume signaling is activated; software must set bit for 10 ms and then clear to enable remote wake-up
0 = Resume signaling is disabled

bit 1 **PPBRST:** Ping-Pong Buffers Reset bit
1 = Reset all Ping-Pong Buffer Pointers to the even BD banks
0 = Ping-Pong Buffer Pointers are not reset

bit 0 **SOFEN:** Start-Of-Frame Enable bit
1 = Start-Of-Frame token is sent every one 1 ms
0 = Start-Of-Frame token is disabled

REGISTER 18-9: U1ADDR: USB ADDRESS REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
LSPDEN(1)	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	ADDR0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as '0'

bit 7 **LSPDEN:** Low-Speed Enable Indicator bit(1)
1 = USB module operates at low speed
0 = USB module operates at full speed

bit 6-0 **ADDR<6:0>:** USB Device Address bits

Note 1: Host mode only. In Device mode, this bit is unimplemented and read as '0'.

REGISTER 18-10: U1TOK: USB TOKEN REGISTER (HOST MODE ONLY)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PID3	PID2	PID1	PID0	EP3	EP2	EP1	EP0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as '0'

bit 7-4 **PID<3:0>:** Token Type Identifier bits
1101 = SETUP (TX) token type transaction(1)
1001 = IN (RX) token type transaction(1)
0001 = OUT (TX) token type transaction(1)

bit 3-0 **EP<3:0>:** Token Command Endpoint Address bits
This value must specify a valid endpoint on the attached device.

Note 1: All other combinations are reserved and are not to be used.

REGISTER 18-11: U1SOF: USB OTG START-OF-TOKEN THRESHOLD REGISTER (HOST MODE ONLY)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CNT7	CNT6	CNT5	CNT4	CNT3	CNT2	CNT1	CNT0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as '0'

bit 7-0 **CNT<7:0>:** Start-Of-Frame Size bits
Value represents 10 + (packet size of n bytes). For example:
0100 1010 = 64-byte packet
0010 1010 = 32-byte packet
0001 0010 = 8-byte packet

REGISTER 18-12: U1CNFG1: USB CONFIGURATION REGISTER 1

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	U-0	R/W-0	U-0	U-0	R/W-0	R/W-0
UTEYE	UOEMON(1)	—	USBSIDL	—	—	PPB1	PPB0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as '0'

bit 7 **UTEYE:** USB Eye Pattern Test Enable bit
1 = Eye pattern test is enabled
0 = Eye pattern test is disabled

bit 6 **UOEMON:** USB $\overline{OE}$ Monitor Enable bit(1)
1 = $\overline{OE}$ signal is active; it indicates intervals during which the D+/D- lines are driving
0 = $\overline{OE}$ signal is inactive

bit 5 **Unimplemented:** Read as '0'

bit 4 **USBSIDL:** USB OTG Stop in Idle Mode bit
1 = Discontinue module operation when the device enters Idle mode
0 = Continue module operation in Idle mode

bit 3-2 **Unimplemented:** Read as '0'

bit 1-0 **PPB<1:0>:** Ping-Pong Buffers Configuration bits
11 = Even/Odd ping-pong buffers are enabled for Endpoints 1 to 15
10 = Even/Odd ping-pong buffers are enabled for all endpoints
01 = Even/Odd ping-pong buffers are enabled for OUT Endpoint 0
00 = Even/Odd ping-pong buffers are disabled

Note 1: This bit is only active when the UTRDIS bit (U1CNFG2<0>) is set.

REGISTER 18-13: U1CNFG2: USB CONFIGURATION REGISTER 2

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	UVCMPSEL	PUVBUS	EXTI2CEN	UVBUSDIS(1)	UVCMPDIS(1)	UTRDIS(1)
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-6 **Unimplemented:** Read as '0'

bit 5 **UVCMPSEL:** VBUS Comparator External Interface Selection bit
1 = Use VBUSVLD, SESSVLD and SESSEND as comparator interface pins
0 = Use VCMPST1 and VCMPST2 as comparator interface pins

bit 4 **PUVBUS:** VBUS Pull-Up Enable bit
1 = Pull-up on VBUS pin is enabled
0 = Pull-up on VBUS pin is disabled

bit 3 **EXTI2CEN:** I2C™ Interface For External Module Control Enable bit
1 = External module(s) is controlled via the I2C™ interface
0 = External module(s) controlled via the dedicated pins

bit 2 **UVBUSDIS:** On-Chip 5V Boost Regulator Builder Disable bit(1)
1 = On-chip boost regulator builder is disabled; digital output control interface is enabled
0 = On-chip boost regulator builder is active

bit 1 **UVCMPDIS:** On-Chip VBUS Comparator Disable bit(1)
1 = On-chip charge VBUS comparator is disabled; digital input status interface is enabled
0 = On-chip charge VBUS comparator is active

bit 0 **UTRDIS:** On-Chip Transceiver Disable bit(1)
1 = On-chip transceiver is disabled; digital transceiver interface is enabled
0 = On-chip transceiver is active

Note 1: Never change these bits while the USBPWR bit is set (U1PWRC<0> = 1).

18.7.2 USB INTERRUPT REGISTERS

REGISTER 18-14: U1OTGIR: USB OTG INTERRUPT STATUS REGISTER (HOST MODE ONLY)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/K-0, HS	R/K-0, HS	R/K-0, HS	R/K-0, HS	R/K-0, HS	R/K-0, HS	U-0	R/K-0, HS
IDIF	T1MSECIF	LSTATEIF	ACTVIF	SESVDIF	SESENDIF	—	VBUSVDIF
bit 7							bit 0

Legend:	U = Unimplemented bit, read as '0'		
R = Readable bit	K = Write '1' to clear bit	HS = Hardware Settable bit	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as '0'

bit 7 **IDIF:** ID State Change Indicator bit
1 = Change in ID state is detected
0 = No ID state change is detected

bit 6 **T1MSECIF:** 1 Millisecond Timer bit
1 = The 1 millisecond timer has expired
0 = The 1 millisecond timer has not expired

bit 5 **LSTATEIF:** Line State Stable Indicator bit
1 = USB line state (as defined by the SE0 and JSTATE bits) has been stable for 1 ms, but different from the last time
0 = USB line state has not been stable for 1 ms

bit 4 **ACTVIF:** Bus Activity Indicator bit
1 = Activity on the D+/D- lines or VBUS is detected
0 = No activity on the D+/D- lines or VBUS is detected

bit 3 **SESVDIF:** Session Valid Change Indicator bit
1 = VBUS has crossed VA_SESS_END (as defined in the USB OTG Specification)(1)
0 = VBUS has not crossed VA_SESS_END

bit 2 **SESENDIF:** B-Device VBUS Change Indicator bit
1 = VBUS change on B-device detected; VBUS has crossed VB_SESS_END (as defined in the USB OTG Specification)(1)
0 = VBUS has not crossed VA_SESS_END

bit 1 **Unimplemented:** Read as '0'

bit 0 **VBUSVDIF:** A-Device VBUS Change Indicator bit
1 = VBUS change on A-device is detected; VBUS has crossed VA_VBUS_VLD (as defined in the USB OTG Specification)(1)
0 = No VBUS change on A-device is detected

Note 1: VBUS threshold crossings may be either rising or falling.

Note: Individual bits can only be cleared by writing a '1' to the bit position as part of a word write operation on the entire register. Using Boolean instructions or bitwise operations to write to a single bit position will cause all set bits at the moment of the write to become cleared.

REGISTER 18-15: U1OTGIE: USB OTG INTERRUPT ENABLE REGISTER (HOST MODE ONLY)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
IDIE	T1MSECIE	LSTATEIE	ACTVIE	SESVDIE	SESENDIE	—	VBUSVDIE
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as '0'

bit 7 **IDIE:** ID Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 6 **T1MSECIE:** 1 Millisecond Timer Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 5 **LSTATEIE:** Line State Stable Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 4 **ACTVIE:** Bus Activity Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 3 **SESVDIE:** Session Valid Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 2 **SESENDIE:** B-Device Session End Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 1 **Unimplemented:** Read as '0'

bit 0 **VBUSVDIE:** A-Device VBUS Valid Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

REGISTER 18-16: U1IR: USB INTERRUPT STATUS REGISTER (DEVICE MODE ONLY)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/K-0, HS	U-0	R/K-0, HS	R/K-0, HS	R/K-0, HS	R/K-0, HS	R-0	R/K-0, HS
STALLIF	—	RESUMEIF	IDLEIF	TRNIF	SOFIF	UERRIF	URSTIF
bit 7							bit 0

Legend:	U = Unimplemented bit, read as '0'		
R = Readable bit	K = Write '1' to clear bit	HS = Hardware Settable bit	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as '0'

bit 7 **STALLIF:** STALL Handshake Interrupt bit
- 1 = A STALL handshake was sent by the peripheral during the handshake phase of the transaction in Device mode
- 0 = A STALL handshake has not been sent

bit 6 **Unimplemented:** Read as '0'

bit 5 **RESUMEIF:** Resume Interrupt bit
- 1 = A K-state is observed on the D+ or D- pin for 2.5 μs (differential '1' for low speed, differential '0' for full speed)
- 0 = No K-state is observed

bit 4 **IDLEIF:** Idle Detect Interrupt bit
- 1 = Idle condition is detected (constant Idle state of 3 ms or more)
- 0 = No Idle condition is detected

bit 3 **TRNIF:** Token Processing Complete Interrupt bit
- 1 = Processing of the current token is complete; read the U1STAT register for endpoint information
- 0 = Processing of the current token is not complete; clear the U1STAT register or load the next token from STAT (clearing this bit causes the STAT FIFO to advance)

bit 2 **SOFIF:** Start-Of-Frame Token Interrupt bit
- 1 = A Start-Of-Frame token is received by the peripheral or the Start-Of-Frame threshold is reached by the host
- 0 = No Start-Of-Frame token is received or threshold reached

bit 1 **UERRIF**: USB Error Condition Interrupt bit (read-only)
- 1 = An unmasked error condition has occurred; only error states enabled in the U1EIE register can set this bit
- 0 = No unmasked error condition has occurred

bit 0 **URSTIF:** USB Reset Interrupt bit
- 1 = Valid USB Reset has occurred for at least 2.5 μs; Reset state must be cleared before this bit can be reasserted
- 0 = No USB Reset has occurred. Individual bits can only be cleared by writing a '1' to the bit position as part of a word write operation on the entire register. Using Boolean instructions or bitwise operations to write to a single bit position will cause all set bits at the moment of the write to become cleared.

Note: Individual bits can only be cleared by writing a '1' to the bit position as part of a word write operation on the entire register. Using Boolean instructions or bitwise operations to write to a single bit position will cause all set bits at the moment of the write to become cleared.

REGISTER 18-17: U1IR: USB INTERRUPT STATUS REGISTER (HOST MODE ONLY)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/K-0, HS	R/K-0, HS	R/K-0, HS	R/K-0, HS	R/K-0, HS	R/K-0, HS	R-0	R/K-0, HS
STALLIF	ATTACHIF	RESUMEIF	IDLEIF	TRNIF	SOFIF	UERRIF	DETACHIF
bit 7							bit 0

Legend:	U = Unimplemented bit, read as '0'		
R = Readable bit	K = Write '1' to clear bit	HS = Hardware Settable bit	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as '0'

bit 7 **STALLIF:** STALL Handshake Interrupt bit
- 1 = A STALL handshake was sent by the peripheral device during the handshake phase of the transaction in Device mode
- 0 = A STALL handshake has not been sent

bit 6 **ATTACHIF:** Peripheral Attach Interrupt bit
- 1 = A peripheral attachment has been detected by the module; it is set if the bus state is not SE0 and there has been no bus activity for 2.5 μs
- 0 = No peripheral attacement has been detected

bit 5 **RESUMEIF:** Resume Interrupt bit
- 1 = A K-state is observed on the D+ or D- pin for 2.5 μs (differential '1' for low speed, differential '0' for full speed)
- 0 = No K-state is observed

bit 4 **IDLEIF:** Idle Detect Interrupt bit
- 1 = Idle condition is detected (constant Idle state of 3 ms or more)
- 0 = No Idle condition is detected

bit 3 **TRNIF:** Token Processing Complete Interrupt bit
- 1 = Processing of the current token is complete; read the U1STAT register for endpoint information
- 0 = Processing of the current token not complete; clear the U1STAT register or load the next token from U1STAT

bit 2 **SOFIF:** Start-Of-Frame Token Interrupt bit
- 1 = A Start-Of-Frame token received by the peripheral or the Start-Of-Frame threshold reached by the host
- 0 = No Start-Of-Frame token received or threshold reached

bit 1 **UERRIF:** USB Error Condition Interrupt bit
- 1 = An unmasked error condition has occurred; only error states enabled in the U1EIE register can set this bit
- 0 = No unmasked error condition has occurred

bit 0 **DETACHIF:** Detach Interrupt bit
- 1 = A peripheral detachment has been detected by the module; Reset state must be cleared before this bit can be reasserted
- 0 = No peripheral detachment is detected. Individual bits can only be cleared by writing a '1' to the bit position as part of a word write operation on the entire register. Using Boolean instructions or bit-wise operations to write to a single bit position will cause all set bits at the moment of the write to become cleared.

Note: Individual bits can only be cleared by writing a '1' to the bit position as part of a word write operation on the entire register. Using Boolean instructions or bitwise operations to write to a single bit position will cause all set bits at the moment of the write to become cleared.

REGISTER 18-18: U1IE: USB INTERRUPT ENABLE REGISTER (ALL USB MODES)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
STALLIE	ATTACHIE(1)	RESUMEIE	IDLEIE	TRNIE	SOFIE	UERRIE	URSTIE
							DETACHIE
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as '0'

bit 7 **STALLIE:** STALL Handshake Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 6 **ATTACHIE:** Peripheral Attach Interrupt bit (Host mode only)(1)
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 5 **RESUMEIE:** Resume Interrupt bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 4 **IDLEIE:** Idle Detect Interrupt bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 3 **TRNIE:** Token Processing Complete Interrupt bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 2 **SOFIE:** Start-Of-Frame Token Interrupt bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 1 **UERRIE:** USB Error Condition Interrupt bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 0 **URSTIE or DETACHIE:** USB Reset Interrupt (Device mode) or USB Detach Interrupt (Host mode) Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

Note 1: Unimplemented in Device mode, read as '0'.

REGISTER 18-19: U1EIR: USB ERROR INTERRUPT STATUS REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/K-0, HS	U-0	R/K-0, HS	R/K-0, HS	R/K-0, HS	R/K-0, HS	R/K-0, HS	R/K-0, HS
BTSEF	—	DMAEF	BTOEF	DFN8EF	CRC16EF	CRC5EF EOFEF	PIDEF
bit 7							bit 0

Legend:	U = Unimplemented bit, read as '0'		
R = Readable bit	K = Write '1' to clear bit	HS = Hardware Settable bit	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as '0'

bit 7 **BTSEF:** Bit Stuff Error Flag bit
1 = Bit stuff error has been detected
0 = No bit stuff error has been detected

bit 6 **Unimplemented:** Read as '0'

bit 5 **DMAEF:** DMA Error Flag bit
1 = A USB DMA error condition is detected; the data size indicated by the BD byte count field is less than the number of received bytes, the received data is truncated
0 = No DMA error

bit 4 **BTOEF:** Bus Turnaround Time-out Error Flag bit
1 = Bus turnaround time-out has occurred
0 = No bus turnaround time-out

bit 3 **DFN8EF:** Data Field Size Error Flag bit
1 = Data field was not an integral number of bytes
0 = Data field was an integral number of bytes

bit 2 **CRC16EF:** CRC16 Failure Flag bit
1 = CRC16 failed
0 = CRC16 passed

bit 1 For Device mode:
CRC5EF: CRC5 Host Error Flag bit
1 = Token packet is rejected due to CRC5 error
0 = Token packet is accepted (no CRC5 error)
For Host mode:
EOFEF: End-Of-Frame Error Flag bit
1 = End-Of-Frame error has occurred
0 = End-Of-Frame interrupt is disabled

bit 0 **PIDEF:** PID Check Failure Flag bit
1 = PID check failed
0 = PID check passed

Note: Individual bits can only be cleared by writing a '1' to the bit position as part of a word write operation on the entire register. Using Boolean instructions or bitwise operations to write to a single bit position will cause all set bits at the moment of the write to become cleared.

REGISTER 18-20: U1EIE: USB ERROR INTERRUPT ENABLE REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BTSEE	—	DMAEE	BTOEE	DFN8EE	CRC16EE	CRC5EE EOFEE	PIDEE
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as '0'

bit 7 **BTSEE:** Bit Stuff Error Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 6 **Unimplemented:** Read as '0'

bit 5 **DMAEE:** DMA Error Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 4 **BTOEE:** Bus Turnaround Time-out Error Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 3 **DFN8EE:** Data Field Size Error Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 2 **CRC16EE:** CRC16 Failure Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 1 For Device mode:
CRC5EE: CRC5 Host Error Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled
For Host mode:
EOFEE: End-of-Frame Error interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

bit 0 **PIDEE:** PID Check Failure Interrupt Enable bit
1 = Interrupt is enabled
0 = Interrupt is disabled

18.7.3 USB ENDPOINT MANAGEMENT REGISTERS

REGISTER 18-21: U1EPn: USB ENDPOINT n CONTROL REGISTERS (n = 0 TO 15)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
LSPD(1)	RETRYDIS(1)	—	EPCONDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as '0'

bit 7 **LSPD:** Low-Speed Direct Connection Enable bit (U1EP0 only)(1)
1 = Direct connection to a low-speed device is enabled
0 = Direct connection to a low-speed device is disabled

bit 6 **RETRYDIS:** Retry Disable bit (U1EP0 only)(1)
1 = Retry NAK transactions is disabled
0 = Retry NAK transactions is enabled; retry is done in hardware

bit 5 **Unimplemented:** Read as '0'

bit 4 **EPCONDIS:** Bidirectional Endpoint Control bit
If EPTXEN and EPRXEN = 1:
1 = Disable Endpoint n from control transfers; only TX and RX transfers are allowed
0 = Enable Endpoint n for control (SETUP) transfers; TX and RX transfers are also allowed
For all other combinations of EPTXEN and EPRXEN:
This bit is ignored.

bit 3 **EPRXEN:** Endpoint Receive Enable bit
1 = Endpoint n receive is enabled
0 = Endpoint n receive is disabled

bit 2 **EPTXEN:** Endpoint Transmit Enable bit
1 = Endpoint n transmit is enabled
0 = Endpoint n transmit is disabled

bit 1 **EPSTALL:** Endpoint Stall Status bit
1 = Endpoint n was stalled
0 = Endpoint n was not stalled

bit 0 **EPHSHK:** Endpoint Handshake Enable bit
1 = Endpoint handshake is enabled
0 = Endpoint handshake is disabled (typically used for isochronous endpoints)

Note 1: These bits are available only for U1EP0 and only in Host mode. For all other U1EPn registers, these bits are always unimplemented and read as '0'.

18.7.4 USB VBUS POWER CONTROL REGISTER

REGISTER 18-22: U1PWMCON: USB VBUS PWM GENERATOR CONTROL REGISTER

R/W-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
PWMEN	—	—	—	—	—	PWMPOL	CNTEN
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **PWMEN:** PWM Enable bit
1 = PWM generator is enabled
0 = PWM generator is disabled; output is held in the Reset state specified by PWMPOL

bit 14-10 **Unimplemented:** Read as '0'

bit 9 **PWMPOL:** PWM Polarity bit
1 = PWM output is active-low and resets high
0 = PWM output is active-high and resets low

bit 8 **CNTEN:** PWM Counter Enable bit
1 = Counter is enabled
0 = Counter is disabled

bit 7-0 **Unimplemented:** Read as '0'

19.0 ENHANCED PARALLEL MASTER PORT (EPMP)

Note: This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the "*PIC24F Family Reference Manual*", **Section 42. "Enhanced Parallel Master Port (EPMP)"** (DS39730). The information in this data sheet supersedes the information in the FRM.

The Enhanced Parallel Master Port (EPMP) module is present in PIC24FJXXXDAX10 devices and not in PIC24FJXXXDAX06 devices. The EPMP provides a parallel 4-bit (Master mode only), 8-bit (Master and Slave modes) or 16-bit (Master mode only) data bus interface to communicate with off-chip modules, such as memories, FIFOs, LCD controllers and other microcontrollers. This module can serve as either the master or the slave on the communication bus. For EPMP Master modes, all external addresses are mapped into the internal Extended Data Space (EDS). This is done by allocating a region of the EDS for each chip select, and then assigning each chip select to a particular external resource, such as a memory or external controller. This region should not be assigned to another device resource, such as RAM or SFRs. To perform a write or read on an external resource, the CPU should simply perform a write or read within the address range assigned for EPMP.

Note: The EPMP module is not present in 64-pin devices (PIC24FJXXXDAX06).

The EPMP has an alternative master feature. The graphics controller module can control the EPMP directly in Alternate Master mode to access an external graphics buffer.

Key features of the EPMP module are:

- Extended Data Space (EDS) interface allows Direct Access from the CPU
- Up to 23 Programmable Address Lines
- Up to 2 Chip Select Lines
- Up to 2 Acknowledgement Lines (one per chip select)
- 4-bit, 8-bit or 16-bit wide Data Bus
- Programmable Strobe Options (per chip select)
  - Individual Read and Write Strobes or;
  - Read/$\overline{\text{Write}}$ Strobe with Enable Strobe
- Programmable Address/Data Multiplexing
- Programmable Address Wait States
- Programmable Data Wait States (per chip select)
- Programmable Polarity on Control Signals (per chip select)
- Legacy Parallel Slave Port Support
- Enhanced Parallel Slave Support
  - Address Support
  - 4-Byte Deep Auto-Incrementing Buffer
- Alternate Master feature

19.1 $\overline{\text{ALTPMP}}$ Setting

Many of the lower order EPMP address pins are shared with ADC inputs. This is an untenable situation for users that need both the ADC channels and the EPMP bus. If the user does not need to use all the address bits, then by clearing the $\overline{\text{ALTPMP}}$ (CW3<12>) Configuration bit, the lower order address bits can be mapped to higher address pins, which frees the ADC channels.

TABLE 19-1: ALTERNATE EPMP PINS

Pin	$\overline{\text{ALTPMP}}$ = 0	$\overline{\text{ALTPMP}}$ = 1
RA14	PMCS2	PMA22
RC4	PMA22	PMCS2
RF12	PMA5	PMA18
RG6	PMA18	PMA5
RG7	PMA20	PMA4
RA3	PMA4	PMA20
RG8	PMA21	PMA3
RA4	PMA3	PMA21

TABLE 19-2: PARALLEL MASTER PORT PIN DESCRIPTION

Pin Name	Type	Description
PMA<22:16>	O	Address bus bits<22-16>
PMA<15>, PMCS2	O	Address bus bit<15>
	O	Chip Select 2 (alternate location)
	I/O	Data bus bit<15> when port size is 16 bits and address is multiplexed
PMA<14>, PMCS1	O	Address bus bit<14>
	O	Chip Select 1 (alternate location)
	I/O	Data bus bit 14 when port size is 16-bit and address is multiplexed
PMA<13:8>	O	Address bus bit< 13-8>
	I/O	Data bus bits<13-8> when port size is 16 bits and address is multiplexed
PMA<7:3>	O	Address bus bit< 7-3>
PMA<2>, PMALU	O	Address bus bit<2>
	O	Address latch upper strobe for multiplexed address
PMA<1>, PMALH	I/O	Address bus bit<1>
	O	Address latch high strobe for multiplexed address
PMA<0>, PMALL	I/O	Address bus bit<0>
	O	Address latch low strobe for multiplexed address
PMD<15:8>	I/O	Data bus bits<15-8> when address is not multiplexed
PMD<7:4>	I/O	Data bus bits<7-4>
	O	Address bus bits<7-4> when port size is 4 bits and address is multiplexed with 1 address phase
PMD<3:0>	I/O	Data bus bits<3-0>
PMCS1	I/O	Chip Select 1
PMCS2	O	Chip Select 2
PMWR, PMENB	I/O	Write strobe or Enable signal depending on Strobe mode
PMRD, PMRD/$\overline{\text{PMWR}}$	I/O	Read strobe or Read/$\overline{\text{Write}}$ signal depending on Strobe mode
PMBE1	O	Byte indicator
PMBE0	O	Nibble or byte indicator
PMACK1	I	Acknowledgment 1
PMACK2	I	Acknowledgment 2

REGISTER 19-1: PMCON1: EPMP CONTROL REGISTER 1

R/W-0	U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
PMPEN	—	PSIDL	ADRMUX1	ADRMUX0	—	MODE1	MODE0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
CSF1	CSF0	ALP	ALMODE	—	BUSKEEP	IRQM1	IRQM0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **PMPEN:** Parallel Master Port Enable bit
1 = EPMP is enabled
0 = EPMP is disabled

bit 14 **Unimplemented:** Read as '0'

bit 13 **PSIDL:** Stop in Idle Mode bit
1 = Discontinue module operation when device enters Idle mode
0 = Continue module operation in Idle mode

bit 12-11 **ADRMUX<1:0>:** Address/Data Multiplexing Selection bits
11 = Lower address bits are multiplexed with data bits using 3 address phases
10 = Lower address bits are multiplexed with data bits using 2 address phases
01 = Lower address bits are multiplexed with data bits using 1 address phase
00 = Address and data appear on separate pins

bit 10 **Unimplemented:** Read as '0'

bit 9-8 **MODE<1:0>:** Parallel Port Mode Select bits
11 = Master mode
10 = Enhanced PSP; pins used are PMRD, PMWR, PMCS, PMD<7:0> and PMA<1:0>
01 = Buffered PSP; pins used are PMRD, PMWR, PMCS and PMD<7:0>
00 = Legacy Parallel Slave Port; PMRD, PMWR, PMCS and PMD<7:0> pins are used

bit 7-6 **CSF<1:0>:** Chip Select Function bits
11 = Reserved
10 = PMA<15> used for Chip Select 2, PMA<14> used for Chip Select 1
01 = PMA<15> used for Chip Select 2, PMCS1 used for Chip Select 1
00 = PMCS2 used for Chip Select 2, PMCS1 used for Chip Select 1

bit 5 **ALP:** Address Latch Polarity bit
1 = Active-high (PMALL, PMALH and PMALU)
0 = Active-low ($\overline{\text{PMALL}}$, $\overline{\text{PMALH}}$ and $\overline{\text{PMALU}}$)

bit 4 **ALMODE:** Address Latch Strobe Mode bit
1 = Enable "smart" address strobes (each address phase is only present if the current access would cause a different address in the latch than the previous address)
0 = Disable "smart" address strobes

bit 3 **Unimplemented:** Read as '0'

bit 2 **BUSKEEP:** Bus Keeper bit
1 = Data bus keeps its last value when not actively being driven
0 = Data bus is in high-impedance state when not actively being driven

bit 1-0 **IRQM<1:0>:** Interrupt Request Mode bits
11 = Interrupt generated when Read Buffer 3 is read or Write Buffer 3 is written (Buffered PSP mode), or on a read or write operation when PMA<1:0> = 11 (Addressable PSP mode only)
10 = Reserved
01 = Interrupt generated at the end of a read/write cycle
00 = No interrupt is generated

REGISTER 19-2: PMCON2: EPMP CONTROL REGISTER 2

R-0, HSC	U-0	R/C-0, HS	R/C-0, HS	R-0, HSC	R-1, HSC	R/W-0	R/W-0
BUSY	—	ERROR	TIMEOUT	AMREQ	CURMST	MSTSEL1	MSTSEL0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RADDR23	RADDR22	RADDR21	RADDR20	RADDR19	RADDR18	RADDR17	RADDR16
bit 7							bit 0

Legend:	HS = Hardware Settable bit	HSC = Hardware Settable/Clearable bit	
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	C = Clearable bit
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **BUSY:** Busy bit (Master mode only)
1 = Port is busy
0 = Port is not busy

bit 14 **Unimplemented:** Read as '0'

bit 13 **ERROR:** Error bit
1 = Transaction error (illegal transaction was requested)
0 = Transaction completed successfully

bit 12 **TIMEOUT:** Time-Out bit
1 = Transaction timed out
0 = Transaction completed successfully

bit 11 **AMREQ:** Alternate Master Request bit
1 = The Alternate Master is requesting use of EPMP
0 = The Alternate Master is not requesting use of EPMP

bit 10 **CURMST:** Current Master bit
1 = EPMP access is granted to CPU
0 = EPMP access is granted to alternate master

bit 9-8 **MSTSEL<1:0>:** Parallel Port Master Select bits
11 = Alternate master I/Os direct access (EPMP Bypass mode)
10 = Reserved
01 = Alternate master
00 = CPU

bit 7-0 **RADDR<23:16>:** Parallel Master Port Reserved Address Space bits(1)

Note 1: If RADDR<23:16> = 00000000, then the last EDS address for Chip Select 2 will be 0xFFFFFF.

REGISTER 19-3: PMCON3: EPMP CONTROL REGISTER 3

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
PTWREN	PTRDEN	PTBE1EN	PTBE0EN	—	AWAITM1	AWAITM0	AWAITE
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	PTEN22	PTEN21	PTEN20	PTEN19	PTEN18	PTEN17	PTEN16
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **PTWREN:** Write/Enable Strobe Port Enable bit
1 = PMWR/PMENB port is enabled
0 = PMWR/PMENB port is disabled

bit 14 **PTRDEN:** Read/Write Strobe Port Enable bit
1 = PMRD/$\overline{\text{PMWR}}$ port is enabled
0 = PMRD/$\overline{\text{PMWR}}$ port is disabled

bit 13 **PTBE1EN:** High Nibble/Byte Enable Port Enable bit
1 = PMBE1 port is enabled
0 = PMBE1 port is disabled

bit 12 **PTBE0EN:** Low Nibble/Byte Enable Port Enable bit
1 = PMBE0 port is enabled
0 = PMBE0 port is disabled

bit 11 **Unimplemented:** Read as '0'

bit 10-9 **AWAITM<1:0>:** Address Latch Strobe Wait States bits
11 = Wait of 3½ TCY
10 = Wait of 2½ TCY
01 = Wait of 1½ TCY
00 = Wait of ½ TCY

bit bit 8 **AWAITE:** Address Hold After Address Latch Strobe Wait States bits
1 = Wait of 1¼ TCY
0 = Wait of ¼ TCY

bit 7 **Unimplemented:** Read as '0'

bit 6-0 **PTEN<22:16>:** EPMP Address Port Enable bits
1 = PMA<22:16> function as EPMP address lines
0 = PMA<22:16> function as port I/Os

REGISTER 19-4: PMCON4: EPMP CONTROL REGISTER 4

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTEN15	PTEN14	PTEN13	PTEN12	PTEN11	PTEN10	PTEN9	PTEN8
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTEN7	PTEN6	PTEN5	PTEN4	PTEN3	PTEN2	PTEN1	PTEN0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **PTEN15:** PMA15 Port Enable bit
1 = PMA15 functions as either Address Line 15 or Chip Select 2
0 = PMA15 functions as port I/O

bit 14 **PTEN14:** PMA14 Port Enable bit
1 = PMA14 functions as either Address Line 14 or Chip Select 1
0 = PMA14 functions as port I/O

bit 13-3 **PTEN<13:3>:** EPMP Address Port Enable bits
1 = PMA<13:3> function as EPMP address lines
0 = PMA<13:3> function as port I/Os

bit 2-0 **PTEN<2:0>:** PMALU/PMALH/PMALL Strobe Enable bits
1 = PMA<2:0> function as either address lines or address latch strobes
0 = PMA<2:0> function as port I/Os

REGISTER 19-5: PMCSxCF: CHIP SELECT x CONFIGURATION REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
CSDIS	CSP	CSPTEN	BEP	—	WRSP	RDSP	SM
bit 15							bit 8

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0
ACKP	PTSZ1	PTSZ0	—	—	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **CSDIS:** Chip Select x Disable bit
1 = Disable the Chip Select x functionality
0 = Enable the Chip Select x functionality

bit 14 **CSP:** Chip Select x Polarity bit
1 = Active-high (PMCSx)
0 = Active-low ($\overline{\text{PMCSx}}$)

bit 13 **CSPTEN:** PMCSx Port Enable bit
1 = PMCSx port is enabled
0 = PMCSx port is disabled

bit 12 **BEP:** Chip Select x Nibble/Byte Enable Polarity bit
1 = Nibble/Byte enable active-high (PMBE0, PMBE1)
0 = Nibble/Byte enable active-low ($\overline{\text{PMBE0}}$, $\overline{\text{PMBE1}}$)

bit 11 **Unimplemented:** Read as '0'

bit 10 **WRSP:** Chip Select x Write Strobe Polarity bit
For Slave modes and Master mode when SM = 0:
1 = Write strobe active-high (PMWR)
0 = Write strobe active-low ($\overline{\text{PMWR}}$)
For Master mode when SM = 1:
1 = Enable strobe active-high (PMENB)
0 = Enable strobe active-low ($\overline{\text{PMENB}}$)

bit 9 **RDSP:** Chip Select x Read Strobe Polarity bit
For Slave modes and Master mode when SM = 0:
1 = Read strobe active-high (PMRD)
0 = Read strobe active-low ($\overline{\text{PMRD}}$)
For Master mode when SM = 1:
1 = Read/write strobe active-high (PMRD/$\overline{\text{PMWR}}$)
0 = Read/Write strobe active-low (PMRD/PMWR)

bit 8 **SM:** Chip Select x Strobe Mode bit
1 = Read/Write and enable strobes (PMRD/$\overline{\text{PMWR}}$ and PMENB)
0 = Read and write strobes (PMRD and PMWR)

bit 7 **ACKP:** Chip Select x Acknowledge Polarity bit
1 = ACK active-high (PMACK1)
0 = ACK active-low ($\overline{\text{PMACK1}}$)

bit 6-5 **PTSZ<1:0>:** Chip Select x Port Size bits
11 = Reserved
10 = 16-bit port size (PMD<15:0>)
01 = 4-bit port size (PMD<3:0>)
00 = 8-bit port size (PMD<7:0>)

bit 4-0 **Unimplemented:** Read as '0'

REGISTER 19-6: PMCSxBS: CHIP SELECT x BASE ADDRESS REGISTER

R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)	R/W(1)
BASE23	BASE22	BASE21	BASE20	BASE19	BASE18	BASE17	BASE16
bit 15							bit 8

R/W(1)	U-0	U-0	U-0	R/W(1)	U-0	U-0	U-0
BASE15	—	—	—	BASE11	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-7 **BASE<23:15>:** Chip Select x Base Address bits(2)

bit 6-4 **Unimplemented:** Read as '0'

bit 3 **BASE<11>:** Chip Select x Base Address bits(2)

bit 2-0 **Unimplemented:** Read as '0'

Note 1: Value at POR is 0x0200 for PMCS1BS and 0x0600 for PMCS2BS.

2: If the whole PMCS2BS register is written together as 0x0000, then the last EDS address for the Chip Select 1 will be 0xFFFFFF. In this case, the Chip Select 2 should not be used. PMCS1BS has no such feature.

REGISTER 19-7: PMCSxMD: CHIP SELECT x MODE REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0
ACKM1	ACKM0	AMWAIT2	AMWAIT1	AMWAIT0	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DWAITB1	DWAITB0	DWAITM3	DWAITM2	DWAITM1	DWAITM0	DWAITE1	DWAITE0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **ACKM<1:0>:** Chip Select x Acknowledge Mode bits
11 = Reserved
10 = PMACKx is used to determine when a read/write operation is complete
01 = PMACKx is used to determine when a read/write operation is complete with time-out
If DWAITM<3:0> = 0000, the maximum time-out is 255 TCY, else it is DWAITM<3:0> cycles.
00 = PMACKx is not used

bit 13-11 **AMWAIT<2:0>:** Chip Select x Alternate Master Wait States bits
111 = Wait of 10 alternate master cycles
...
001 = Wait of 4 alternate master cycles
000 = Wait of 3 alternate master cycles

bit 10-8 **Unimplemented:** Read as '0'

bit 7-6 **DWAITB<1:0>:** Chip Select x Data Setup Before Read/Write Strobe Wait States bits
11 = Wait of 3¼ TCY
10 = Wait of 2¼ TCY
01 = Wait of 1¼ TCY
00 = Wait of ¼ TCY

bit 5-2 **DWAITM<3:0>:** Chip Select x Data Read/Write Strobe Wait States bits
For Write operations:
1111 = Wait of 15½ TCY
...
0001 = Wait of 1½ TCY
0000 = Wait of ½ TCY
For Read operations:
1111 = Wait of 15¾ TCY
...
0001 = Wait of 1¾ TCY
0000 = Wait of ¾ TCY

bit 1-0 **DWAITE<1:0>:** Chip Select x Data Hold After Read/Write Strobe Wait States bits
For Write operations:
11 = Wait of 3¼ TCY
10 = Wait of 2¼ TCY
01 = Wait of 1¼ TCY
00 = Wait of ¼ TCY
For Read operations:
11 = Wait of 3 TCY
10 = Wait of 2 TCY
01 = Wait of 1 TCY
00 = Wait of 0 TCY

REGISTER 19-8: PMSTAT: EPMP STATUS REGISTER (SLAVE MODE ONLY)

R-0, HSC	R/W-0 HS	U-0	U-0	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC
IBF	IBOV	—	—	IB3F	IB2F	IB1F	IB0F
bit 15							bit 8

R-1, HSC	R/W-0 HS	U-0	U-0	R-1, HSC	R-1, HSC	R-1, HSC	R-1, HSC
OBE	OBUF	—	—	OB3E	OB2E	OB1E	OB0E
bit 7							bit 0

Legend:	HS = Hardware Settable bit	HSC = Hardware Settable/Clearable bit	
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **IBF:** Input Buffer Full Status bit
1 = All writable input buffer registers are full
0 = Some or all of the writable input buffer registers are empty

bit 14 **IBOV:** Input Buffer Overflow Status bit
1 = A write attempt to a full input register occurred (must be cleared in software)
0 = No overflow occurred

bit 13-12 **Unimplemented:** Read as '0'

bit 11-8 **IBxF:** Input Buffer x Status Full bit(1)
1 = Input buffer contains unread data (reading buffer will clear this bit)
0 = Input buffer does not contain unread data

bit 7 **OBE:** Output Buffer Empty Status bit
1 = All readable output buffer registers are empty
0 = Some or all of the readable output buffer registers are full

bit 6 **OBUF:** Output Buffer Underflow Status bit
1 = A read occurred from an empty output register (must be cleared in software)
0 = No underflow occurred

bit 5-4 **Unimplemented:** Read as '0'

bit 3-0 **OBxE:** Output Buffer x Status Empty bit
1 = Output buffer is empty (writing data to the buffer will clear this bit)
0 = Output buffer contains untransmitted data

Note 1: Even though an individual bit represents the byte in the buffer, the bits corresponding to the Word (byte 0 and 1, or byte 2 and 3) gets cleared even on byte reading.

REGISTER 19-9: PADCFG1: PAD CONFIGURATION CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	RTSECSEL(1)	PMPTTL(2)
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-2 **Unimplemented:** Read as '0'

bit 1 **RTSECSEL:** RTCC Seconds Clock Output Select bit(1)
- 1 = RTCC seconds clock is selected for the RTCC pin
- 0 = RTCC alarm pulse is selected for the RTCC pin

bit 0 **PMPTTL:** EPMP Module TTL Input Buffer Select bit(2)
- 1 = EPMP module inputs (PMDx, PMCS1) use TTL input buffers
- 0 = EPMP module inputs use Schmitt Trigger input buffers

Note 1: To enable the actual RTCC output, the RTCOE (RCFGCAL<10>) bit must also be set.

2: Unimplemented in 64-pin devices (PIC24FJXXXDAX06); maintain as '0'.

NOTES:

20.0 REAL-TIME CLOCK AND CALENDAR (RTCC)

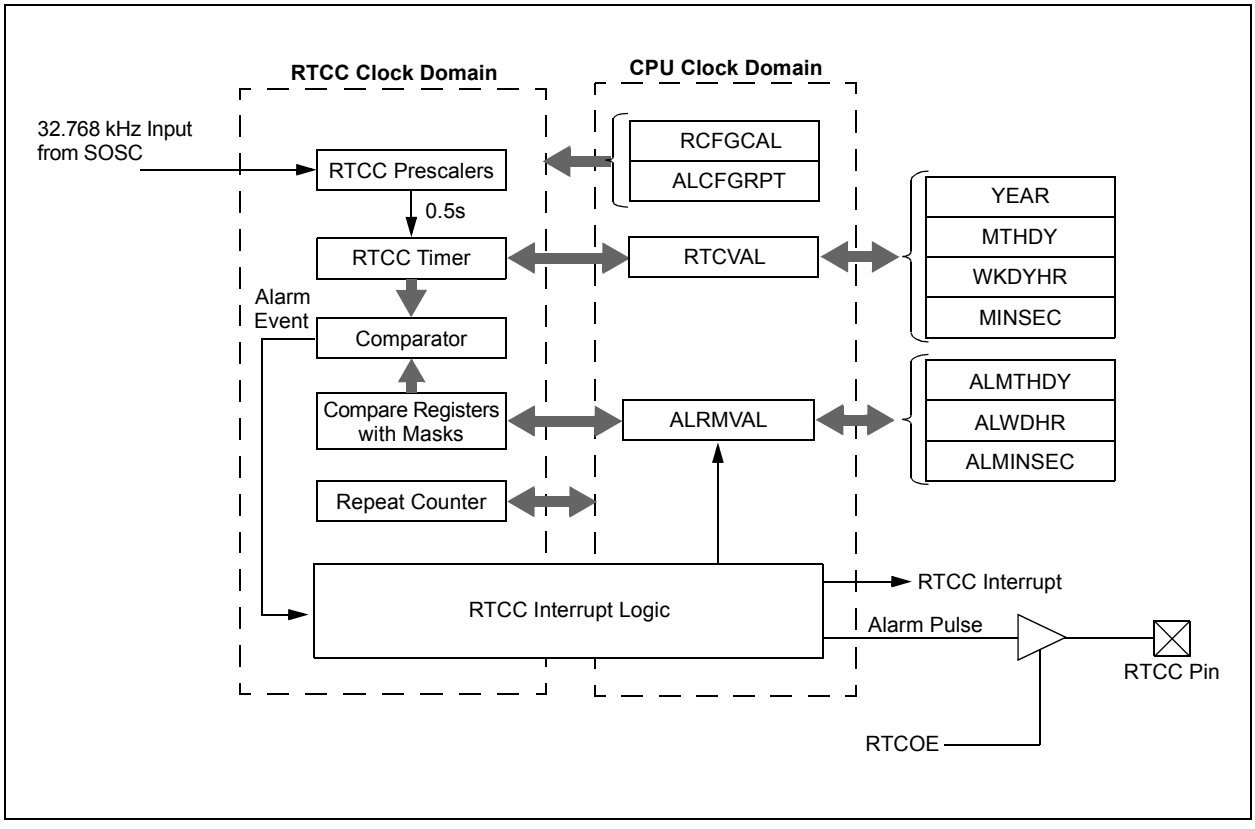
Note: This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the "*PIC24F Family Reference Manual*", **Section 29. "Real-Time Clock and Calendar (RTCC)"** (DS39696). The information in this data sheet supersedes the information in the FRM.

The Real-Time Clock and Calendar (RTCC) provides a function that can be calibrated.

Key features of the RTCC module are:

- Operates in Sleep mode
- Provides hours, minutes and seconds using 24-hour format
- Visibility of half of one second period
- Provides calendar – weekday, date, month and year
- Alarm configurable for half a second, one second,10 seconds, one minute, 10 minutes, one hour, one day, one week, one month or one year
- Alarm repeat with decrementing counter
- Alarm with indefinite repeat chime
- Year, 2000 to 2099, leap year correction
- BCD format for smaller software overhead
- Optimized for long-term battery operation
- User calibration of the 32.768 kHz clock crystal/32K INTRC frequency with periodic auto-adjust
  - Calibration to within ±2.64 seconds error per month
  - Calibrates up to 260 ppm of crystal error

FIGURE 20-1: RTCC BLOCK DIAGRAM



20.1 RTCC Module Registers

The RTCC module registers are organized into three categories:

- RTCC Control Registers
- RTCC Value Registers
- Alarm Value Registers

20.1.1 REGISTER MAPPING

To limit the register interface, the RTCC Timer and Alarm Time registers are accessed through the corresponding register pointers. The RTCC Value register window (RTCVALH and RTCVALL) uses the RTCPTR bits (RCFGCAL<9:8>) to select the desired Timer register pair (see Table 20-1).

By writing the RTCVALH byte, the RTCC Pointer value, RTCPTR<1:0> bits, decrement by one until they reach '00'. Once they reach '00', the MINUTES and SECONDS value will be accessible through RTCVALH and RTCVALL until the pointer value is manually changed.

TABLE 20-1: RTCVAL REGISTER MAPPING

RTCPTR <1:0>	RTCC Value Register Window	
	RTCVAL<15:8>	RTCVAL<7:0>
00	MINUTES	SECONDS
01	WEEKDAY	HOURS
10	MONTH	DAY
11	—	YEAR

The Alarm Value register window (ALRMVALH and ALRMVALL) uses the ALRMPTR bits (ALCFGRPT<9:8>) to select the desired Alarm register pair (see Table 20-2).

By writing the ALRMVALH byte, the Alarm Pointer value bits, ALRMPTR<1:0>, decrement by one until they reach '00'. Once they reach '00', the ALRMMIN and ALRMSEC value will be accessible through ALRMVALH and ALRMVALL until the pointer value is manually changed.

TABLE 20-2: ALRMVAL REGISTER MAPPING

ALRMPTR <1:0>	Alarm Value Register Window	
	ALRMVAL<15:8>	ALRMVAL<7:0>
00	ALRMMIN	ALRMSEC
01	ALRMWD	ALRMHR
10	ALRMMNTH	ALRMDAY
11	—	—

Considering that the 16-bit core does not distinguish between 8-bit and 16-bit read operations, the user must be aware that when reading either the ALRMVALH or ALRMVALL bytes, they will decrement the ALRMPTR<1:0> value. The same applies to the RTCVALH or RTCVALL bytes with the RTCPTR<1:0> being decremented.

Note: This only applies to read operations and not write operations.

20.1.2 WRITE LOCK

In order to perform a write to any of the RTCC Timer registers, the RTCWREN (RCFGCAL<13>) bit must be set (refer to Example 20-1).

Note: To avoid accidental writes to the timer, it is recommended that the RTCWREN bit (RCFGCAL<13>) is kept clear at any other time. For the RTCWREN bit to be set, there is only 1 instruction cycle time window allowed between the unlock sequence and the setting of RTCWREN; therefore, it is recommended that code follow the procedure in Example 20-1.

For applications written in C, the unlock sequence should be implemented using in-line assembly.

EXAMPLE 20-1: SETTING THE RTCWREN BIT

```
asm  volatile("disi #5");
asm  volatile("mov #0x55, w7");
asm  volatile("mov w7, _NVMKEY");
asm  volatile("mov #0xAA, w8");
asm  volatile("mov w8, _NVMKEY");
asm  volatile("bset _RCFGCAL, #13");      //set the RTCWREN bit
```

20.1.3 RTCC CONTROL REGISTERS

REGISTER 20-1: RCFGCAL: RTCC CALIBRATION AND CONFIGURATION REGISTER(1)

R/W-0	U-0	R/W-0	R-0, HSC	R-0, HSC	R/W-0	R/W-0, HSC	R/W-0, HSC
RTCEN(2)	—	RTCWREN	RTCSYNC	HALFSEC(3)	RTCOE	RTCPTR1	RTCPTR0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0
bit 7							bit 0

Legend:	HSC = Hardware Settable/Clearable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **RTCEN:** RTCC Enable bit(2)
1 = RTCC module is enabled
0 = RTCC module is disabled

bit 14 **Unimplemented:** Read as '0'

bit 13 **RTCWREN:** RTCC Value Registers Write Enable bit
1 = RTCVALH and RTCVALL registers can be written to by the user
0 = RTCVALH and RTCVALL registers are locked out from being written to by the user

bit 12 **RTCSYNC:** RTCC Value Registers Read Synchronization bit
1 = RTCVALH, RTCVALL and ALCFGRPT registers can change while reading due to a rollover ripple resulting in an invalid data read. If the register is read twice and results in the same data, the data can be assumed to be valid.
0 = RTCVALH, RTCVALL or ALCFGRPT registers can be read without concern over a rollover ripple

bit 11 **HALFSEC:** Half-Second Status bit(3)
1 = Second half period of a second
0 = First half period of a second

bit 10 **RTCOE:** RTCC Output Enable bit
1 = RTCC output is enabled
0 = RTCC output is disabled

bit 9-8 **RTCPTR<1:0>:** RTCC Value Register Window Pointer bits
Points to the corresponding RTCC Value registers when reading the RTCVALH and RTCVALL registers. The RTCPTR<1:0> value decrements on every read or write of RTCVALH until it reaches '00'.
RTCVAL<15:8>:
11 = Reserved
10 = MONTH
01 = WEEKDAY
00 = MINUTES
RTCVAL<7:0>:
11 = YEAR
10 = DAY
01 = HOURS
00 = SECONDS

Note 1: The RCFGCAL register is only affected by a POR.
2: A write to the RTCEN bit is only allowed when RTCWREN = 1.
3: This bit is read-only. It is cleared to '0' on a write to the lower half of the MINSEC register.

REGISTER 20-1: RCFGCAL: RTCC CALIBRATION AND CONFIGURATION REGISTER(1)

bit 7-0 **CAL<7:0>:** RTC Drift Calibration bits

```
01111111 = Maximum positive adjustment; adds 508 RTC clock pulses every one minute
.
.
.
11111111 = Minimum negative adjustment; subtracts 4 RTC clock pulses every one minute
00000001 = Minimum positive adjustment; adds 4 RTC clock pulses every one minute
00000000 = No adjustment
.
.
.
10000000 = Maximum negative adjustment; subtracts 512 RTC clock pulses every one minute
```

Note 1: The RCFGCAL register is only affected by a POR.
2: A write to the RTCEN bit is only allowed when RTCWREN = 1.
3: This bit is read-only. It is cleared to '0' on a write to the lower half of the MINSEC register.

REGISTER 20-2: PADCFG1: PAD CONFIGURATION CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	RTSECSEL(1)	PMPTTL
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-2 **Unimplemented:** Read as '0'

bit 1 **RTSECSEL:** RTCC Seconds Clock Output Select bit(1)
1 = RTCC seconds clock is selected for the RTCC pin
0 = RTCC alarm pulse is selected for the RTCC pin

bit 0 **PMPTTL:** EPMP Module TTL Input Buffer Select bit
1 = EPMP module inputs (PMDx, PMCS1) use TTL input buffers
0 = EPMP module inputs use Schmitt Trigger input buffers

Note 1: To enable the actual RTCC output, the RTCOE (RCFGCAL<10>) bit must also be set.

REGISTER 20-3: ALCFGRPT: ALARM CONFIGURATION REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0, HSC	R/W-0, HSC
ALRMEN	CHIME	AMASK3	AMASK2	AMASK1	AMASK0	ALRMPTR1	ALRMPTR0
bit 15							bit 8

R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC
ARPT7	ARPT6	ARPT5	ARPT4	ARPT3	ARPT2	ARPT1	ARPT0
bit 7							bit 0

Legend:	HSC = Hardware Settable/Clearable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **ALRMEN:** Alarm Enable bit
1 = Alarm is enabled (cleared automatically after an alarm event whenever ARPT<7:0> = 00h and CHIME = 0)
0 = Alarm is disabled

bit 14 **CHIME:** Chime Enable bit
1 = Chime is enabled; ARPT<7:0> bits are allowed to roll over from 00h to FFh
0 = Chime is disabled; ARPT<7:0> bits stop once they reach 00h

bit 13-10 **AMASK<3:0>:** Alarm Mask Configuration bits
11xx = Reserved – do not use
101x = Reserved – do not use
1001 = Once a year (except when configured for February 29th, once every 4 years)
1000 = Once a month
0111 = Once a week
0110 = Once a day
0101 = Every hour
0100 = Every 10 minutes
0011 = Every minute
0010 = Every 10 seconds
0001 = Every second
0000 = Every half second

bit 9-8 **ALRMPTR<1:0>:** Alarm Value Register Window Pointer bits
Points to the corresponding Alarm Value registers when reading the ALRMVALH and ALRMVALL registers. The ALRMPTR<1:0> value decrements on every read or write of ALRMVALH until it reaches '00'.
ALRMVAL<15:8>:
11 = Unimplemented
10 = ALRMMNTH
01 = ALRMWD
00 = ALRMMIN
ALRMVAL<7:0>:
11 = Unimplemented
10 = ALRMDAY
01 = ALRMHR
00 = ALRMSEC

bit 7-0 **ARPT<7:0>:** Alarm Repeat Counter Value bits
11111111 = Alarm will repeat 255 more times
...
00000000 = Alarm will not repeat
The counter decrements on any alarm event. The counter is prevented from rolling over from 00h to FFh unless CHIME = 1.

20.1.4 RTCVAL REGISTER MAPPINGS

REGISTER 20-4: YEAR: YEAR VALUE REGISTER(1)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC
YRTEN3	YRTEN2	YRTEN1	YRTEN0	YRONE3	YRONE2	YRONE1	YRONE0
bit 7							bit 0

Legend:	HSC = Hardware Settable/Clearable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as '0'

bit 7-4 **YRTEN<3:0>:** Binary Coded Decimal Value of Year's Tens Digit bits
Contains a value from 0 to 9.

bit 3-0 **YRONE<3:0>:** Binary Coded Decimal Value of Year's Ones Digit bits
Contains a value from 0 to 9.

Note 1: A write to the YEAR register is only allowed when RTCWREN = 1.

REGISTER 20-5: MTHDY: MONTH AND DAY VALUE REGISTER(1)

U-0	U-0	U-0	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC
—	—	—	MTHTEN0	MTHONE3	MTHONE2	MTHONE1	MTHONE0
bit 15							bit 8

U-0	U-0	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC
—	—	DAYTEN1	DAYTEN0	DAYONE3	DAYONE2	DAYONE1	DAYONE0
bit 7							bit 0

Legend:	HSC = Hardware Settable/Clearable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-13 **Unimplemented:** Read as '0'

bit 12 **MTHTEN0:** Binary Coded Decimal Value of Month's Tens Digit bit
Contains a value of 0 or 1.

bit 11-8 **MTHONE<3:0>:** Binary Coded Decimal Value of Month's Ones Digit bits
Contains a value from 0 to 9.

bit 7-6 **Unimplemented:** Read as '0'

bit 5-4 **DAYTEN<1:0>:** Binary Coded Decimal Value of Day's Tens Digit bits
Contains a value from 0 to 3.

bit 3-0 **DAYONE<3:0>:** Binary Coded Decimal Value of Day's Ones Digit bits
Contains a value from 0 to 9.

Note 1: A write to this register is only allowed when RTCWREN = 1.

REGISTER 20-6: WKDYHR: WEEKDAY AND HOURS VALUE REGISTER(1)

U-0	U-0	U-0	U-0	U-0	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC
—	—	—	—	—	WDAY2	WDAY1	WDAY0
bit 15							bit 8

U-0	U-0	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC
—	—	HRTEN1	HRTEN0	HRONE3	HRONE2	HRONE1	HRONE0
bit 7							bit 0

Legend:	HSC = Hardware Settable/Clearable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-11 **Unimplemented:** Read as '0'

bit 10-8 **WDAY<2:0>:** Binary Coded Decimal Value of Weekday Digit bits
Contains a value from 0 to 6.

bit 7-6 **Unimplemented:** Read as '0'

bit 5-4 **HRTEN<1:0>:** Binary Coded Decimal Value of Hour's Tens Digit bits
Contains a value from 0 to 2.

bit 3-0 **HRONE<3:0>:** Binary Coded Decimal Value of Hour's Ones Digit bits
Contains a value from 0 to 9.

Note 1: A write to this register is only allowed when RTCWREN = 1.

REGISTER 20-7: MINSEC: MINUTES AND SECONDS VALUE REGISTER

U-0	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC
—	MINTEN2	MINTEN1	MINTEN0	MINONE3	MINONE2	MINONE1	MINONE0
bit 15							bit 8

U-0	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC	R/W-x, HSC
—	SECTEN2	SECTEN1	SECTEN0	SECONE3	SECONE2	SECONE1	SECONE0
bit 7							bit 0

Legend:	HSC = Hardware Settable/Clearable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as '0'

bit 14-12 **MINTEN<2:0>:** Binary Coded Decimal Value of Minute's Tens Digit bits
Contains a value from 0 to 5.

bit 11-8 **MINONE<3:0>:** Binary Coded Decimal Value of Minute's Ones Digit bits
Contains a value from 0 to 9.

bit 7 **Unimplemented:** Read as '0'

bit 6-4 **SECTEN<2:0>:** Binary Coded Decimal Value of Second's Tens Digit bits
Contains a value from 0 to 5.

bit 3-0 **SECONE<3:0>:** Binary Coded Decimal Value of Second's Ones Digit bits
Contains a value from 0 to 9.

20.1.5 ALRMVAL REGISTER MAPPINGS

REGISTER 20-8: ALMTHDY: ALARM MONTH AND DAY VALUE REGISTER(1)

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	MTHTEN0	MTHONE3	MTHONE2	MTHONE1	MTHONE0
bit 15							bit 8

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	DAYTEN1	DAYTEN0	DAYONE3	DAYONE2	DAYONE1	DAYONE0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-13 **Unimplemented:** Read as '0'

bit 12 **MTHTEN0:** Binary Coded Decimal Value of Month's Tens Digit bit
Contains a value of 0 or 1.

bit 11-8 **MTHONE<3:0>:** Binary Coded Decimal Value of Month's Ones Digit bits
Contains a value from 0 to 9.

bit 7-6 **Unimplemented:** Read as '0'

bit 5-4 **DAYTEN<1:0>:** Binary Coded Decimal Value of Day's Tens Digit bits
Contains a value from 0 to 3.

bit 3-0 **DAYONE<3:0>:** Binary Coded Decimal Value of Day's Ones Digit bits
Contains a value from 0 to 9.

Note 1: A write to this register is only allowed when RTCWREN = 1.

REGISTER 20-9: ALWDHR: ALARM WEEKDAY AND HOURS VALUE REGISTER(1)

U-0	U-0	U-0	U-0	U-0	R/W-x	R/W-x	R/W-x
—	—	—	—	—	WDAY2	WDAY1	WDAY0
bit 15							bit 8

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	HRTEN1	HRTEN0	HRONE3	HRONE2	HRONE1	HRONE0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-11 **Unimplemented:** Read as '0'

bit 10-8 **WDAY<2:0>:** Binary Coded Decimal Value of Weekday Digit bits
Contains a value from 0 to 6.

bit 7-6 **Unimplemented:** Read as '0'

bit 5-4 **HRTEN<1:0>:** Binary Coded Decimal Value of Hour's Tens Digit bits
Contains a value from 0 to 2.

bit 3-0 **HRONE<3:0>:** Binary Coded Decimal Value of Hour's Ones Digit bits
Contains a value from 0 to 9.

Note 1: A write to this register is only allowed when RTCWREN = 1.

REGISTER 20-10: ALMINSEC: ALARM MINUTES AND SECONDS VALUE REGISTER

U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	MINTEN2	MINTEN1	MINTEN0	MINONE3	MINONE2	MINONE1	MINONE0
bit 15							bit 8

U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	SECTEN2	SECTEN1	SECTEN0	SECONE3	SECONE2	SECONE1	SECONE0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **Unimplemented:** Read as '0'

bit 14-12 **MINTEN<2:0>:** Binary Coded Decimal Value of Minute's Tens Digit bits
Contains a value from 0 to 5.

bit 11-8 **MINONE<3:0>:** Binary Coded Decimal Value of Minute's Ones Digit bits
Contains a value from 0 to 9.

bit 7 **Unimplemented:** Read as '0'

bit 6-4 **SECTEN<2:0>:** Binary Coded Decimal Value of Second's Tens Digit bits
Contains a value from 0 to 5.

bit 3-0 **SECONE<3:0>:** Binary Coded Decimal Value of Second's Ones Digit bits
Contains a value from 0 to 9.

20.2 Calibration

The real-time crystal input can be calibrated using the periodic auto-adjust feature. When properly calibrated, the RTCC can provide an error of less than 3 seconds per month. This is accomplished by finding the number of error clock pulses for one minute and storing the value into the lower half of the RCFGCAL register. The 8-bit signed value loaded into the lower half of RCFGCAL is multiplied by four and will either be added or subtracted from the RTCC timer, once every minute. Refer to the following steps for RTCC calibration:

1. Using another timer resource on the device, the user must find the error of the 32.768 kHz crystal.
2. Once the error is known, it must be converted to the number of error clock pulses per minute and loaded into the RCFGCAL register.

EQUATION 20-1: RTCC CALIBRATION

Error (clocks per minute) = (Ideal Frequency† – Measured Frequency) x 60

†Ideal Frequency = 32,768H

3. a) If the oscillator is faster then ideal (negative result form Step 2), the RCFGCAL register value needs to be negative. This causes the specified number of clock pulses to be subtracted from the timer counter, once every minute.

   b) If the oscillator is slower then ideal (positive result from Step 2), the RCFGCAL register value needs to be positive. This causes the specified number of clock pulses to be added to the timer counter, once every minute.
4. Divide the number of error clocks per minute by 4 to get the correct CAL value and load the RCFGCAL register with the correct value.

   (Each 1-bit increment in CAL adds or subtracts 4 pulses).

Writes to the lower half of the RCFGCAL register should only occur when the timer is turned off or immediately after the rising edge of the seconds pulse.

Note:	It is up to the user to include in the error value the initial error of the crystal, drift due to temperature and drift due to crystal aging.

20.3 Alarm

- Configurable from half second to one year
- Enabled using the ALRMEN bit (ALCFGRPT<15>, Register 20-3)
- One-time alarm and repeat alarm options available

20.3.1 CONFIGURING THE ALARM

The alarm feature is enabled using the ALRMEN bit. This bit is cleared when an alarm is issued. Writes to ALRMVAL should only take place when ALRMEN = 0.

As shown in Figure 20-2, the interval selection of the alarm is configured through the AMASK bits (ALCFGRPT<13:10>). These bits determine which and how many digits of the alarm must match the clock value for the alarm to occur.

The alarm can also be configured to repeat based on a preconfigured interval. The amount of times this occurs, once the alarm is enabled, is stored in the ARPT bits, ARPT<7:0> (ALCFGRPT<7:0>). When the value of the ARPT bits equals 00h and the CHIME bit (ALCFGRPT<14>) is cleared, the repeat function is disabled and only a single alarm will occur. The alarm can be repeated up to 255 times by loading ARPT<7:0> with FFh.

After each alarm is issued, the value of the ARPT bits is decremented by one. Once the value has reached 00h, the alarm will be issued one last time, after which the ALRMEN bit will be cleared automatically and the alarm will turn off.

Indefinite repetition of the alarm can occur if the CHIME bit = 1. Instead of the alarm being disabled when the value of the ARPT bits reaches 00h, it rolls over to FFh and continues counting indefinitely while CHIME is set.

20.3.2 ALARM INTERRUPT

At every alarm event, an interrupt is generated. In addition, an alarm pulse output is provided that operates at half the frequency of the alarm. This output is completely synchronous to the RTCC clock and can be used as a trigger clock to other peripherals.

Note:	Changing any of the registers, other then the RCFGCAL and ALCFGRPT registers and the CHIME bit while the alarm is enabled (ALRMEN = 1), can result in a false alarm event leading to a false alarm interrupt. To avoid a false alarm event, the timer and alarm values should only be changed while the alarm is disabled (ALRMEN = 0). It is recommended that the ALCFGRPT register and CHIME bit be changed when RTCSYNC = 0.

FIGURE 20-2: ALARM MASK SETTINGS

Alarm Mask Setting (AMASK<3:0>)	Day of the Week	Month	Day	Hours	Minutes	Seconds
0000 – Every half second 0001 – Every second						
0010 – Every 10 seconds						s
0011 – Every minute						s s
0100 – Every 10 minutes					m	s s
0101 – Every hour					m m	s s
0110 – Every day				h h	m m	s s
0111 – Every week	d			h h	m m	s s
1000 – Every month			d d	h h	m m	s s
1001 – Every year(1)		m m	d d	h h	m m	s s

Note 1: Annually, except when configured for February 29.

NOTES:

21.0 32-BIT PROGRAMMABLE CYCLIC REDUNDANCY CHECK (CRC) GENERATOR

Note: This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the "*PIC24F Family Reference Manual*", **Section 41. "32-Bit Programmable Cyclic Redundancy Check (CRC)"** (DS39729). The information in this data sheet supersedes the information in the FRM.

The 32-bit programmable CRC generator provides a hardware implemented method of quickly generating checksums for various networking and security applications. It offers the following features:

- User-programmable CRC polynomial equation, up to 32 bits
- Programmable shift direction (little or big-endian)
- Independent data and polynomial lengths
- Configurable interrupt output
- Data FIFO

Figure 21-1 displays a simplified block diagram of the CRC generator. A simple version of the CRC shift engine is displayed in Figure 21-2.

FIGURE 21-1: CRC BLOCK DIAGRAM

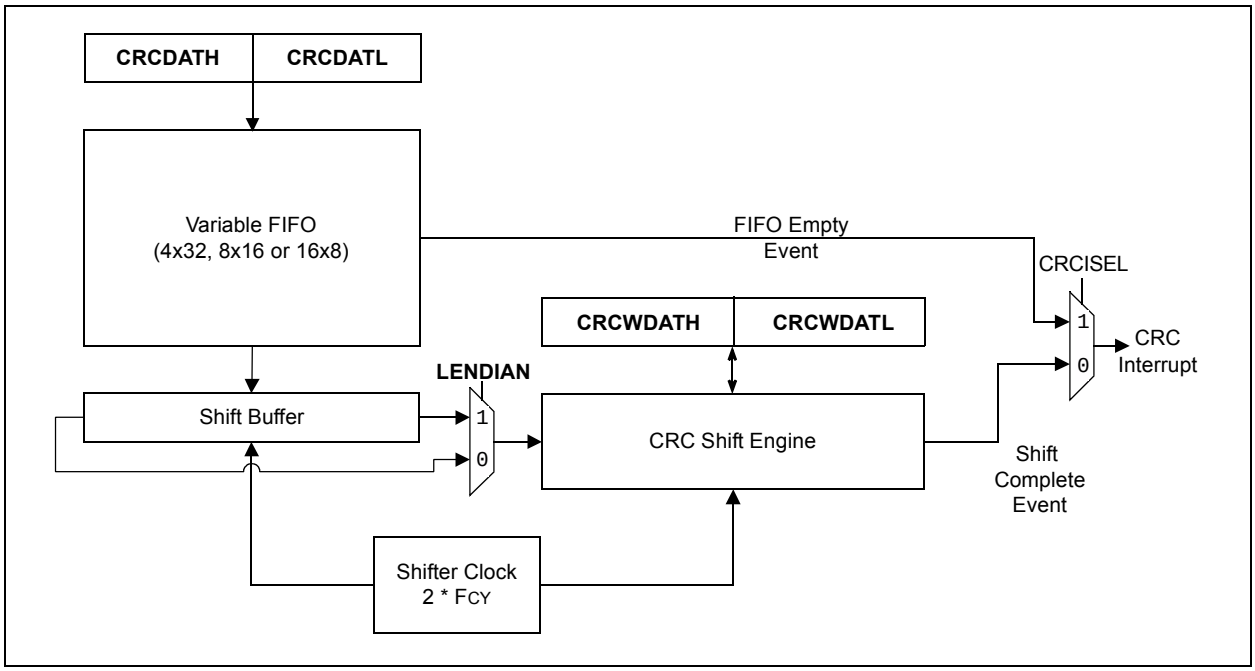
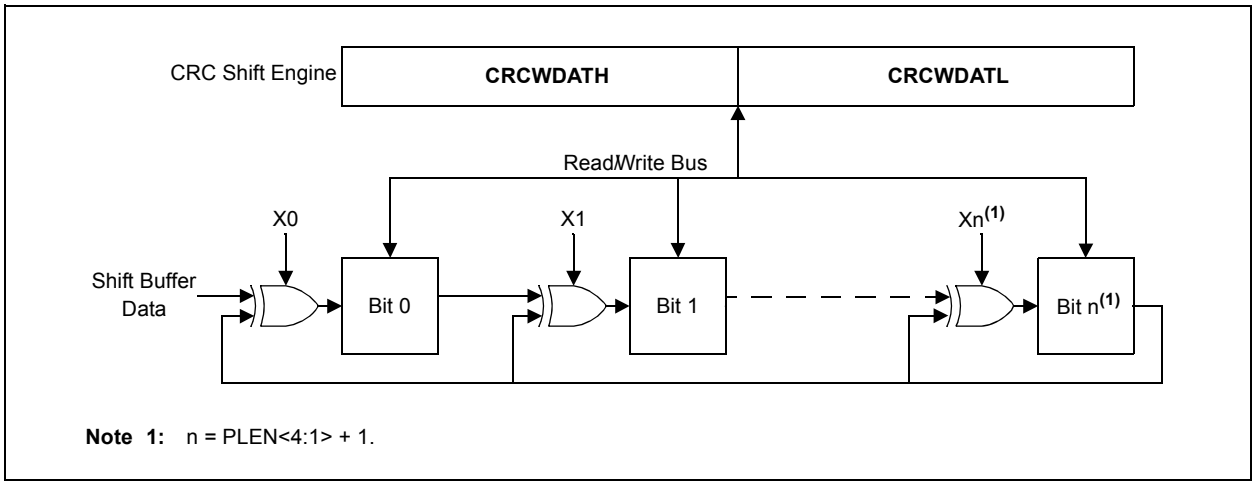


FIGURE 21-2: CRC SHIFT ENGINE DETAIL



Note 1: n = PLEN<4:1> + 1.

21.1 User Interface

21.1.1 POLYNOMIAL INTERFACE

The CRC module can be programmed for CRC polynomials of up of up the 32nd order, using up to 32 bits.

Polynomial length, which reflects the highest exponent in the equation, is selected by the PLEN<4:0> bits (CRCCON2<4:0>).

The CRCXORL and CRCXORH registers control which exponent terms are included in the equation. Setting a particular bit includes that exponent term in the equation; functionally, this includes an XOR operation on the corresponding bit in the CRC engine. Clearing the bit disables the XOR.

For example, consider two CRC polynomials, one a 16-bit and the other a 32-bit equation.

EQUATION 21-1: 16-BIT, 32-BIT CRC POLYNOMIALS

$$X16 + X12 + X5 + 1$$

and

$$X32+X26 + X23 + X22 + X16 + X12 + X11 + X10 + X8 + X7 + X5 + X4 + X2 + X + 1$$

To program these polynomial into the CRC generator, set the register bits as shown in Table 21-1.

Note that the appropriate positions are set to '1' to indicate they are used in the equation (for example, X26 and X23). The '0' bit required by the equation is always XORed; thus, X0 is a don't care. For a polynomial of length 32, it is assumed that the 32nd bit will be used. Therefore, the X<31:1> bits do not have the 32nd bit.

21.1.2 DATA INTERFACE

The module incorporates a FIFO that works with a variable data width. Input data width can be configured to any value between one and 32 bits using the DWIDTH<4:0> bits (CRCCON2<12:8>). When the data width is greater than 15, the FIFO is four words deep. When the DWITDH bits are between 15 and 8, the FIFO is 8 words deep. When the DWIDTH bits are less than 8, the FIFO is 16 words deep.

The data for which the CRC is to be calculated must first be written into the FIFO. Even if the data width is less than 8, the smallest data element that can be written into the FIFO is one byte. For example, if DWIDTH is five, then the size of the data is DWIDTH + 1 or six. The data is written as a whole byte; the two unused upper bits are ignored by the module.

Once data is written into the MSb of the CRCDAT registers (that is, MSb as defined by the data width), the value of the VWORD<4:0> bits (CRCCON1<12:8>) increments by one. For example, if DWIDTH is 24, the VWORD bits will increment when bit 7 of CRCDATH is written. Therefore, CRCDATL must always be written to before CRCDATH.

The CRC engine starts shifting data when the CRCGO bit is set and the value of VWORD is greater than zero.

Each word is copied out of the FIFO into a buffer register, which decrements VWORD. The data is then shifted out of the buffer. The CRC engine continues shifting at a rate of two bits per instruction cycle, until VWORD reaches zero. This means that for a given data width, it takes half that number of instructions for each word to complete the calculation. For example, it takes 16 cycles to calculate the CRC for a single word of 32-bit data.

When VWORD reaches the maximum value for the configured value of DWIDTH (4, 8 or 16), the CRCFUL bit becomes set. When VWORD reaches zero, the CRCMPT bit becomes set. The FIFO is emptied and the VWORD<4:0> bits are set to '00000' whenever CRCEN is '0'.

At least one instruction cycle must pass after a write to CRCWDAT before a read of the VWORD bits is done.

TABLE 21-1: CRC SETUP EXAMPLES FOR 16 AND 32-BIT POLYNOMIALS

CRC Control Bits	Bit Values	
	16-Bit Polynomial	32-Bit Polynomial
PLEN<4:0>	01111	11111
X<31:16>	0000 0000 0000 0001	0000 0100 1100 0001
X<15:0>	0001 0000 0010 000X	0001 1101 1011 011x

21.1.3 DATA SHIFT DIRECTION

The LENDIAN bit (CRCCON1<3>) is used to control the shift direction. By default, the CRC will shift data through the engine, MSb first. Setting LENDIAN (= 1) causes the CRC to shift data, LSb first. This setting allows better integration with various communication schemes and removes the overhead of reversing the bit order in software. Note that this only changes the direction the data is shifted into the engine. The result of the CRC calculation will still be a normal CRC result, not a reverse CRC result.

21.1.4 INTERRUPT OPERATION

The module generates an interrupt that is configurable by the user for either of two conditions.

If CRCISEL is '0', an interrupt is generated when the VWORD<4:0> bits make a transition from a value of '1' to '0'. If CRCISEL is '1', an interrupt will be generated after the CRC operation finishes and the module sets the CRCGO bit to '0'. Manually setting CRCGO to '0' will not generate an interrupt. Note that when an interrupt occurs, the CRC calculation would not yet be complete. The module will still need (PLEN + 1)/2 clock cycles after the interrupt is generated until the CRC calculation is finished.

21.1.5 TYPICAL OPERATION

To use the module for a typical CRC calculation:

1. Set the CRCEN bit to enable the module.
2. Configure the module for desired operation:
   a) Program the desired polynomial using the CRCXORL and CRCXORH registers, and the PLEN<4:0> bits.
   b) Configure the data width and shift direction using the DWIDTH and LENDIAN bits.
   c) Select the desired interrupt mode using the CRCISEL bit.
3. Preload the FIFO by writing to the CRCDATL and CRCDATH registers until the CRCFUL bit is set or no data is left.
4. Clear old results by writing 00h to CRCWDATL and CRCWDATH. The CRCWDAT registers can also be left unchanged to resume a previously halted calculation.
5. Set the CRCGO bit to start calculation.
6. Write remaining data into the FIFO as space becomes available.
7. When the calculation completes, CRCGO is automatically cleared. An interrupt will be generated if CRCISEL = 1.
8. Read CRCWDATL and CRCWDATH for the result of the calculation.

There are eight registers used to control programmable CRC operation:

- CRCCON1
- CRCCON2
- CRCXORL
- CRCXORH
- CRCDATL
- CRCDATH
- CRCWDATL
- CRCWDATH

The CRCCON1 and CRCCON2 registers (Register 21-1 and Register 21-2) control the operation of the module and configure the various settings.

The CRCXOR registers (Register 21-3 and Register 21-4) select the polynomial terms to be used in the CRC equation. The CRCDAT and CRCWDAT registers are each register pairs that serve as buffers for the double-word input data, and CRC processed output, respectively.

REGISTER 21-1: CRCCON1: CRC CONTROL 1 REGISTER

R/W-0	U-0	R/W-0	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC
CRCEN	—	CSIDL	VWORD4	VWORD3	VWORD2	VWORD1	VWORD0
bit 15							bit 8

R-0, HSC	R-1, HSC	R/W-0	R/W-0, HC	R/W-0	U-0	U-0	U-0
CRCFUL	CRCMPT	CRCISEL	CRCGO	LENDIAN	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown
HC = Hardware Cleared	HSC = Hardware Settable/Clearable bit		

bit 15 **CRCEN:** CRC Enable bit
1 = Enables module
0 = Disables module; all state machines, pointers and CRCWDAT/CRCDATH reset; other SFRs are NOT reset

bit 14 **Unimplemented:** Read as '0'

bit 13 **CSIDL:** CRC Stop in Idle Mode bit
1 = Discontinue module operation when device enters Idle mode
0 = Continue module operation in Idle mode

bit 12-8 **VWORD<4:0>:** Pointer Value bits
Indicates the number of valid words in the FIFO. Has a maximum value of 8 when PLEN<4:0> ≥ 7 or 16 when PLEN<4:0> ≤ 7.

bit 7 **CRCFUL:** FIFO Full bit
1 = FIFO is full
0 = FIFO is not full

bit 6 **CRCMPT:** FIFO Empty bit
1 = FIFO is empty
0 = FIFO is not empty

bit 5 **CRCISEL:** CRC Interrupt Selection bit
1 = Interrupt on FIFO is empty; the final word of data is still shifting through the CRC
0 = Interrupt on shift is complete and results are ready

bit 4 **CRCGO:** Start CRC bit
1 = Start CRC serial shifter
0 = CRC serial shifter is turned off

bit 3 **LENDIAN:** Data Shift Direction Select bit
1 = Data word is shifted into the CRC, starting with the LSb (little endian)
0 = Data word is shifted into the CRC, starting with the MSb (big endian)

bit 2-0 **Unimplemented:** Read as '0'

REGISTER 21-2: CRCCON2: CRC CONTROL 2 REGISTER

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	DWIDTH4	DWIDTH3	DWIDTH2	DWIDTH1	DWIDTH0
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	PLEN4	PLEN3	PLEN2	PLEN1	PLEN0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-13 **Unimplemented:** Read as ‘0’

bit 12-8 **DWIDTH<4:0>:** Data Word Width Configuration bits
Configures the width of the data word (data word width – 1).

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **PLEN<4:0>:** Polynomial Length Configuration bits
Configures the length of the polynomial (polynomial length – 1).

REGISTER 21-3: CRCXORL: CRC XOR POLYNOMIAL REGISTER, LOW BYTE

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
X15	X14	X13	X12	X11	X10	X9	X8
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
X7	X6	X5	X4	X3	X2	X1	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-1 **X<15:1>:** XOR of Polynomial Term x^n Enable bits

bit 0 **Unimplemented:** Read as ‘0’

REGISTER 21-4: CRCXORH: CRC XOR HIGH REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
X31	X30	X29	X28	X27	X26	X25	X24
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
X23	X22	X21	X20	X19	X18	X17	X16
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-0 **X<31:16>:** XOR of Polynomial Term x^n Enable bits

REGISTER 21-5: CRCDATL: CRC DATA LOW REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DATA15	DATA14	DATA13	DATA12	DATA11	DATA10	DATA9	DATA8
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-0 **DATA<15:0>:** CRC Input Data bits
Writing to this register fills the FIFO; reading from this register returns '0'.

REGISTER 21-6: CRCDATH: CRC DATA HIGH REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DATA15	DATA14	DATA13	DATA12	DATA11	DATA10	DATA9	DATA8
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-0 **DATA<15:0>:** CRC Input Data bits
Writing to this register fills the FIFO; reading from this register returns '0'.

REGISTER 21-7: CRCWDATL: CRC SHIFT LOW REGISTER

R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC
SDATA15	SDATA14	SDATA13	SDATA12	SDATA11	SDATA10	SDATA9	SDATA8
bit 15							bit 8

R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC
SDATA7	SDATA6	SDATA5	SDATA4	SDATA3	SDATA2	SDATA1	SDATA0
bit 7							bit 0

Legend:	HSC = Hardware Settable/Clearable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-0 **SDATA<15:0>:** CRC Shift Register bits
Writing to this register writes to the CRC Shift register through the CRC write bus. Reading from this register reads the CRC read bus.

REGISTER 21-8: CRCWDATH: CRC SHIFT HIGH REGISTER

R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC
SDATA31	SDATA30	SDATA29	SDATA28	SDATA27	SDATA26	SDATA25	SDATA24
bit 15							bit 8

R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC
SDATA23	SDATA22	SDATA21	SDATA20	SDATA19	SDATA18	SDATA17	SDATA16
bit 7							bit 0

Legend:	HSC = Hardware Settable/Clearable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-0 **SDATA<31:16>:** CRC Input Data bits
Writing to this register writes to the CRC Shift register through the CRC write bus. Reading from this register reads the CRC read bus.

NOTES:

22.0 GRAPHICS CONTROLLER MODULE (GFX)

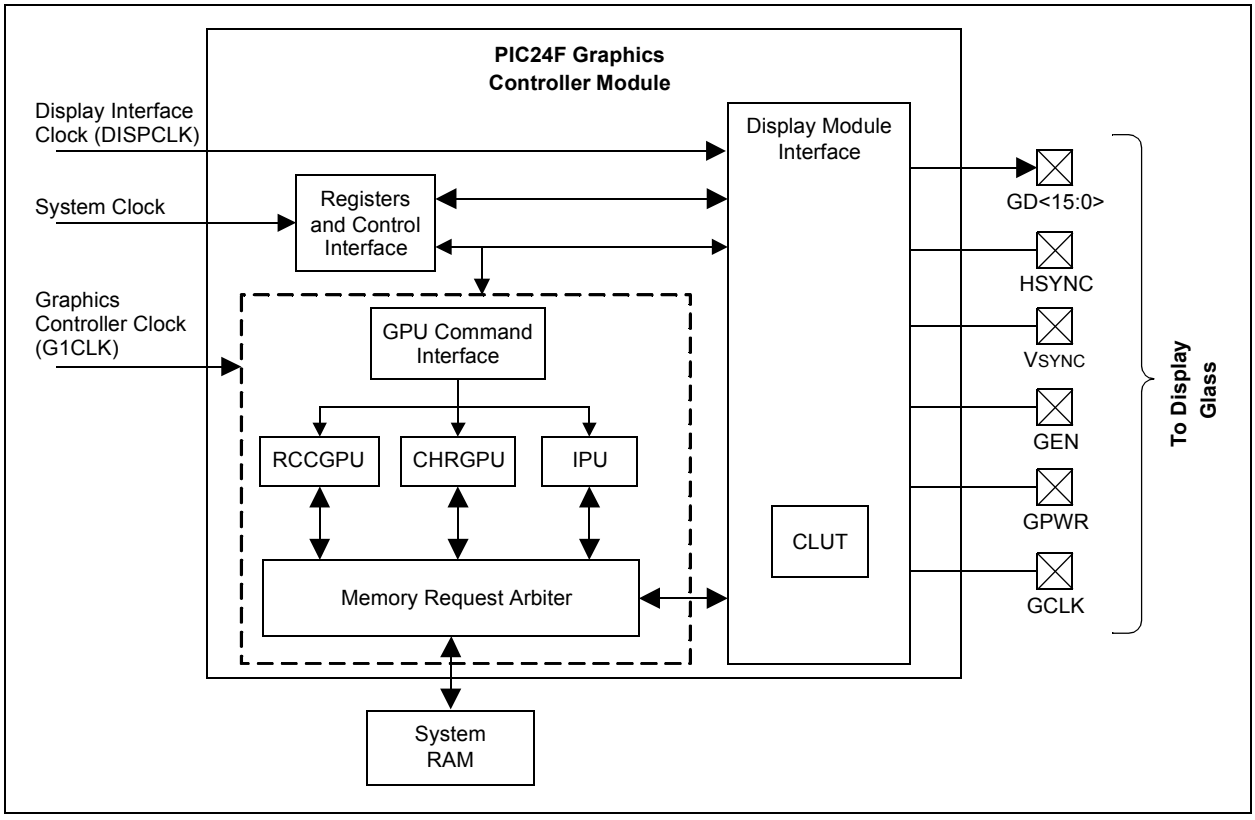
Note: This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the "*PIC24F Family Reference Manual*", **Section 43. "Graphics Controller Module (GFX)"** (DS39731). The information in this data sheet supersedes the information in the FRM.

The Graphics Controller (GFX) module is specifically designed to directly interface with the display glasses, with a built-in analog drive, to individually control pixels in the screen. The module also provides an accelerated rendering of vertical and horizontal lines, rectangles, copying of rectangular area between different locations on the screen, drawing texts and decompressing packed data. The use of the accelerated rendering is performed using command registers. Once initiated, the hardware will perform the rendering, and the software can either poll the status, or use the interrupts to continue rendering of the succeeding shapes.

Key features of the GFX module include:

- Direct interface to three categories of display glasses:
  - Monochrome STN
  - Color STN
  - Color TFT
- Programmable vertical and horizontal synchronization signals' timing and display clock frequency to meet display's frame rates
- Optional internal or external display buffer to accommodate different types of display resolution
- Graphic hardware accelerators:
  - Character Graphics Processing Unit (CHRGPU)
  - Rectangle Copy Graphics Processing Unit (RCCGPU)
  - Inflate Processing Unit (IPU)
- 256 Entries Color Look-up Table (CLUT)
- Supports 1/2/4/8/16 bits-per-pixel (bpp) color depth
- Programmable display resolution
- Supports multiple display interfaces:
  - 4/8/16-bit Monochrome STN
  - 4/8/16-bit Color STN
  - 9/12/18/24-bit color TFT (18 and 24-bit displays are connected as 16-bit 5-6-5 RGB color format)

FIGURE 22-1: GRAPHICS MODULE OVERVIEW



22.1 GFX Module Registers

REGISTER 22-1: G1CMDL: GPU COMMAND LOW REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GCMD15	GCMD14	GCMD13	GCMD12	GCMD11	GCMD10	GCMD9	GCMD8
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GCMD7	GCMD6	GCMD5	GCMD4	GCMD3	GCMD2	GCMD1	GCMD0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-0 **GCMD<15:0>:** Low GPU Command bits

The full 32-bit command is defined by G1CMDH and G1CMDL (GCMD<31:0>). Writes to this register will not trigger the loading of GCMD <31:0> to the command FIFO. For command FIFO loading, see the G1CMDH register description.

REGISTER 22-2: G1CMDH: GPU COMMAND HIGH REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GCMD31	GCMD30	GCMD29	GCMD28	GCMD27	GCMD26	GCMD25	GCMD24
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GCMD23	GCMD22	GCMD21	GCMD20	GCMD19	GCMD18	GCMD17	GCMD16
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-0 **GCMD<31:16>:** High GPU Command bits

The full 32-bit command is defined by G1CMDH and G1CMDL (GCMD<31:0>). A word write to the G1CMDH register triggers the loading of GCMD<31:0> to the command FIFO. Byte writes to the G1CMDH are allowed but only a high byte write will trigger the command loading to the FIFO. Low byte write to this register will only update the G1CMDH<7:0> bits.

REGISTER 22-3: G1CON1: DISPLAY CONTROL REGISTER 1

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
G1EN	—	G1SIDL	GCMDWMK4	GCMDWMK3	GCMDWMK2	GCMDWMK1	GCMDWMK0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC
PUBPP2	PUBPP1	PUBPP0	GCMDCNT4	GCMDCNT3	GCMDCNT2	GCMDCNT1	GCMDCNT0
bit 7							bit 0

Legend:	HSC = Hardware Settable/Clearable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **G1EN:** Module Enable bit
1 = Display module is enabled
0 = Display module is disabled

bit 14 **Unimplemented:** Read as '0'

bit 13 **G1SIDL:** Stop in Idle bit
1 = Display module stops in Idle mode
0 = Display module does not stop in Idle mode

bit 12-8 **GCMDWMK<4:0>:** Command FIFO Watermark bits
Sets the command watermark level that triggers the CMDLVIF interrupt and sets the CMDLV flag; GCMDWMK<4:0> (10000 = Reserved)
10000 = If the number of commands present in the FIFO goes from 16 to 15 commands, the CMDLVIF interrupt will trigger and the CMDLV flag will be set
01111 = f the number of commands present in the FIFO goes from 15 to 14 commands, CMDLVIF interrupt will trigger and CMDLV flag will be set
.
.
.
00001 = If the number of commands present in the FIFO goes from 1 to 0 commands, the CMDLVIF interrupt will trigger and the CMDLV flag will be set
00000 = CMDLVIF interrupt will not trigger and the CMDLV flag will not be set

bit 7-5 **PUBPP<2:0>:** GPU bits-per-pixel (bpp) Setting bits
Other = Reserved
100 = 16 bits-per-pixel
011 = 8 bits-per-pixel
010 = 4 bits-per-pixel
001 = 2 bits-per -pixel
000 = 1-bit -per-pixel

bit 4-0 **GCMDCNT<4:0>:** Command FIFO Occupancy Status bits
When the FIFO is full, any additional commands written to the FIFO are discarded.
10000 = 16 commands are present in the FIFO
01111 = 15 commands are present in the FIFO
.
.
.
0001 = 1 command is present in the FIFO
0000 = 0 command is present in the FIFO

REGISTER 22-4: G1CON2: DISPLAY CONTROL REGISTER 2

R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0
DPGWDTH1	DPGWDTH0	DPSTGER1	DPSTGER0	—	—	DPTEST1	DPTEST0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0
DPBPP2	DPBPP1	DPBPP0	—	—	DPMODE2	DPMODE1	DPMODE0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **DPGWDTH<1:0>:** STN Display Glass Data Width bits
11 = Reserved
10 = 16 bits wide
01 = 8 bits wide
00 = 4 bits wide
These bits have no effect on TFT mode. TFT display glass data width is always assumed to be 16 bits wide.

bit 13-12 **DPSTGER<1:0>:** Display Data Timing Stagger bits
11 = Delays of the display data are staggered in groups:
Bit group 0: 0 4 8 12 – not delayed
Bit group 1: 1 5 9 13 – delayed by ½ GPUCLK cycle
Bit group 2: 2 6 10 14 – delayed by full GPUCLK cycle
Bit group 3: 3 7 11 15 – delayed by 1 ½ GPUCLK cycle
10 = Even bits of the display data are delayed by 1 full GPUCLK cycle; odd bits are not delayed
01 = Odd bits of the display data are delayed by ½ GPUCLK cycle; even bits are not delayed
00 = Display data timing is all synchronized on one clock GPUCLK edge

bit 11-10 **Unimplemented:** Read as '0'

bit 9-8 **DPTEST<1:0>:** Display Test Pattern Generator bits
11 = Borders
10 = Bars
01 = Black screen
00 = Normal Display mode; test patterns are off

bit 7-5 **DPBPP<2:0>:** Display bits-per-pixel Setting bits
This setting must match the GPU bits-per-pixel set in PUBPP<2:0> (G1CON1<7:5>).
100 = 16 bits-per-pixel
011 = 8 bits-per-pixel
010 = 4 bits-per-pixel
001 = 2 bits-per-pixel
000 = 1 bit-per-pixel
Other = Reserved

bit 4-3 **Unimplemented:** Read as '0'

bit 2-0 **DPMODE<2:0>:** Display Glass Type bits
011 = Color STN type
010 = Mono STN type
001 = TFT type
000 = Display off
Other = Reserved

REGISTER 22-5: G1CON3: DISPLAY CONTROL REGISTER 3

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	DPPINOE	DPPOWER
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DPCLKPOL	DPENPOL	DPVSPOL	DPHSPOL	DPPWROE	DPENOE	DPVSOE	DPHSOE
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-10 **Unimplemented:** Read as '0'

bit 9 **DPPINOE:** Display Pin Output Pad Enable bit

DPPINOE is the master output enable and must be set to allow GDBEN<15:0>, DPENOE, DPPWROE, DPVSOE and DPHSOE to enable the associated pads

1 = Enable display output pads

0 = Disable display output signals as set by GDBEN<15:0>

Pins used by the signals are assigned to the next enabled module that uses the same pins.

For data signals, GDBEN<15:0> can be used to disable or enable specific data signals while DPPINOE is set.

bit 8 **DPPOWER:** Display Power-up Power-Down Sequencer Control bit

Refer to the "*PIC24F Family Reference Manual*", **Section 43. "Graphics Controller Module (GFX)"** for details.

1 = Set Display Power Sequencer Control port (GPWR) to '1'

0 = Set Power Control Sequencer signal (GPWR) '0'

bit 7 **DPCLKPOL:** Display Glass Clock (GCLK) Polarity bit

1 = Display latches data on the positive edge of GCLK

0 = Display latches data on the negative edge of GCLK

bit 6 **DPENPOL:** Display Enable Signal (GEN) Polarity bit

For TFT mode (DPMODE (G1CON2<2:0>) = 001):

1 = Active-high (GEN)

0 = Active-low ($\overline{\text{GEN}}$)

For STN mode (DPMODE (G1CON2<2:0>) = 010 or 011):

1 = GEN connects to the shift clock input of the display (Shift Clock mode)

0 = GEN connects to the MOD input of the display (Line/Frame Toggle mode)

bit 5 **DPVSPOL:** Display Vertical Synchronization (VSYNC) Polarity bit

1 = Active-high (VSYNC)

0 = Active-low ($\overline{\text{VSYNC}}$)

bit 4 **DPHSPOL:** Display Horizontal Synchronization (HSYNC) Polarity bit

1 = Active-high (HSYNC)

0 = Active-low ($\overline{\text{HSYNC}}$)

bit 3 **DPPWROE:** Display Power-up/Power-Down Sequencer Control port (GPWR) enable bit

1 = GPWR port is enabled (pin controlled by the DPPOWER bit (G1CON3<8>))

0 = GPWR port is disabled (pin can be used as an ordinary I/O)

bit 2 **DPENOE:** Display Enable Port Enable bit

1 = GEN port is enabled

0 = GEN port is disabled

REGISTER 22-5: G1CON3: DISPLAY CONTROL REGISTER 3 (CONTINUED)

bit 1 **DPVSOE:** Display Vertical Synchronization Port Enable bit
1 = VSYNC port is enabled
0 = VSYNC port is disabled

bit 0 **DPHSOE:** Display Horizontal Synchronization Port Enable bit
1 = HSYNC port is enabled
0 = HSYNC port is disabled

REGISTER 22-6: G1STAT: GFX STATUS REGISTER

R-0, HSC	U-0	U-0	U-0	U-0	U-0	U-0	U-0
PUBUSY	—	—	—	—	—	—	—
bit 15							bit 8

R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC
IPUBUSY	RCCBUSY	CHRBUSY	VMRGN	HMRGN	CMDLV	CMDFUL	CMDMPT
bit 7							bit 0

Legend:	HSC = Hardware Settable/Clearable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **PUBUSY:** Processing Units are Busy Status bit
This bit is logically equivalent to the ORed combination of IPUBUSY, RCCBUSY or CHRBUSY.
1 = At least one processing unit is busy
0 = None of the processing units are busy

bit 14-8 **Unimplemented:** Read as '0'

bit 7 **IPUBUSY:** Inflate Processing Unit Busy Status bit
1 = IPU is busy
0 = IPU is not busy

bit 6 **RCCBUSY:** Rectangle Copy Graphics Processing Unit Busy Status bit
1 = RCCGPU is busy
0 = RCCGPU is not busy

bit 5 **CHRBUSY:** Character Graphics Processing Unit Busy Status bit
1 = CHRGPU is busy
0 = CHRGPU is not busy

bit 4 **VMRGN:** Vertical Blanking Status bit
1 = Display interface is in the vertical blanking period
0 = Display interface is not in the vertical blanking period

bit 3 **HMRGN:** Horizontal Blanking Status bit
1 = Display interface is in the horizontal blanking period
0 = Display interface is not in the horizontal blanking period

bit 2 **CMDLV:** Command Watermark Level Status bit
The number of commands in the command FIFO changed from equal (=) to the command watermark value to less than (<) the Command Watermark value set in CMDWMK (G1CON1<12:8>) register bits.
1 = Command in FIFO is less than the set CMDWMK value
0 = Command in FIFO is equal to or greater than the set CMDWMK value

bit 1 **CMDFUL:** Command FIFO Full Status bit
1 = Command FIFO is full
0 = Command FIFO is not full

bit 0 **CMDMPT:** Command FIFO Empty Status bit
1 = Command FIFO is empty
0 = Command FIFO is not empty

REGISTER 22-7: G1IE: GFX INTERRUPT ENABLE REGISTER

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
PUIE	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
IPUIE	RCCIE	CHRIE	VMRGNIE	HMRGNIE	CMDLVIE	CMDFULIE	CMDMPTIE
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **PUIE:** Processing Units Complete Interrupt Enable bit
1 = Enables the PU complete interrupt
0 = Disables the PU complete interrupt

bit 14-8 **Unimplemented:** Read as '0'

bit 7 **IPUIE:** Inflate Processing Unit Complete Interrupt Enable bit
1 = Enables the IPU complete interrupt
0 = Disables the IPU complete interrupt

bit 6 **RCCIE:** Rectangle Copy Graphics Processing Unit Complete Interrupt bit
1 = Enables the RCCGPU complete interrupt
0 = Disables the RCCGPU complete interrupt

bit 5 **CHRIE:** Character Graphics Processing Unit Busy Interrupt bit
1 = Enables the CHRGPU busy interrupt
0 = Disables the CHRGPU busy interrupt

bit 4 **VMRGNIE:** Vertical Blanking Interrupt Enable bit
1 = Enables the vertical blanking period interrupt
0 = Disables the vertical blanking period interrupt

bit 3 **HMRGNIE:** Horizontal Blanking Interrupt Enable bit
1 = Enables the horizontal blanking period interrupt
0 = Disables the horizontal blanking period interrupt

bit 2 **CMDLVIE:** Command Watermark Interrupt Enable bit
1 = Enables the command watermark interrupt bit
0 = Disables the command watermark interrupt bit

bit 1 **CMDFULIE:** Command FIFO Full Interrupt Enable bit
1 = Enables the command FIFO full interrupt
0 = Disables the command FIFO full interrupt

bit 0 **CMDMPTIE:** Command FIFO Empty Interrupt Enable bit
1 = Enables the command FIFO empty interrupt
0 = Disables the command FIFO empty interrupt

REGISTER 22-8: G1IR: GFX INTERRUPT STATUS REGISTER

R/W-0, HS	U-0	U-0	U-0	U-0	U-0	U-0	U-0
PUIF	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS
IPUIF(1)	RCCIF(1)	CHRIF(1)	VMRGNIF	HMRGNIF	CMDLVIF	CMDFULIF	CMDMPTIF
bit 7							bit 0

Legend:	HS = Hardware Settable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **PUIF:** Processing Units Complete Interrupt Flag bit
PUIF is an ORed combination of IPUIF, RCCIF and CHRIF.
1 = One or more PUs completed command execution (must be cleared in software)
0 = All PUs are Idle or busy completing command execution

bit 14-8 **Unimplemented:** Read as '0'

bit 7 **IPUIF:** Inflate Processing Unit Complete Interrupt Flag bit(1)
1 = IPU completed command execution (must be cleared in software)
0 = IPU is Idle or busy completing command execution

bit 6 **RCCIF:** Rectangle Copy Graphics Processing Unit Complete Interrupt Flag bit(1)
1 = RCCGPU completed command execution (must be cleared in software)
0 = RCCGPU is Idle or busy completing command execution

bit 5 **CHRIF:** Character Graphics Processing Unit Complete Interrupt Flag bit(1)
1 = CHRGPU completed command execution (must be cleared in software)
0 = CHRGPU is Idle or busy completing command execution

bit 4 **VMRGNIF:** Vertical Blanking Interrupt Flag bit
1 = Display interface is in the vertical blanking period (must be cleared in software)
0 = Display interface is not in the vertical blanking period

bit 3 **HMRGNIF:** Horizontal Blanking Interrupt Flag bit
1 = Display interface is in the horizontal blanking period (must be cleared in software)
0 = Display interface is not in the horizontal blanking period

bit 2 **CMDLVIF:** Command Watermark Interrupt Flag bit
1 = Command watermark level is reached (must be cleared in software)
0 = Command watermark level is not yet reached

bit 1 **CMDFULIF:** Command FIFO Full Interrupt Flag bit
1 = Command FIFO is full (must be cleared in software)
0 = Command FIFO is not full

bit 0 **CMDMPTIF:** Command FIFO Empty Interrupt Flag bit
1 = Command FIFO is empty (must be cleared in software)
0 = Command FIFO is not empty

Note 1: The logic of each Processing Unit Status bit is the reverse of the PUIF. This provides flexibility on software code utilizing the processing units.

REGISTER 22-9: G1W1ADRL: GPU WORK AREA 1 START ADDRESS REGISTER LOW

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
W1ADR15	W1ADR14	W1ADR13	W1ADR12	W1ADR11	W1ADR10	W1ADR9	W1ADR8
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
W1ADR7	W1ADR6	W1ADR5	W1ADR4	W1ADR3	W1ADR2	W1ADR1	W1ADR0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-0 **W1ADR<15:0>:** GPU Work Area 1 Start Address Low bits
Work area address must point to an even byte address in memory.

REGISTER 22-10: G1W1ADRH: GPU WORK AREA 1 START ADDRESS REGISTER HIGH

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
W1ADR23	W1ADR22	W1ADR21	W1ADR20	W1ADR19	W1ADR18	W1ADR17	W1ADR16
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as '0'
bit 7-0 **W1ADR<23:16>:** GPU Work Area 1 Start Address High bits
Work area address must point to an even byte address in memory.

REGISTER 22-11: G1W2ADRL: GPU WORK AREA 2 START ADDRESS REGISTER LOW

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
W2ADR15	W2ADR14	W2ADR13	W2ADR12	W2ADR11	W2ADR10	W2ADR9	W2ADR8
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
W2ADR7	W2ADR6	W2ADR5	W2ADR4	W2ADR3	W2ADR2	W2ADR1	W2ADR0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-0 **W2ADR<15:0>:** GPU Work Area 2 Start Address Low bits
Work area address must point to an even byte address in memory.

REGISTER 22-12: G1W2ADRH: GPU WORK AREA 2 START ADDRESS REGISTER HIGH

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
W2ADR23	W2ADR22	W2ADR21	W2ADR20	W2ADR19	W2ADR18	W2ADR17	W2ADR16
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as '0'

bit 7-0 **W2ADR<23:16>:** GPU Work Area 2 Start Address High bits
Work area address must point to an even byte address in memory.

REGISTER 22-13: G1PUW: GPU WORK AREA WIDTH REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	PUW10	PUW9	PUW8
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PUW7	PUW6	PUW5	PUW4	PUW3	PUW2	PUW1	PUW0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-11 **Unimplemented:** Read as '0'

bit 10-0 **PUW<10:0>:** GPU Work Area Width bits (in pixels)

REGISTER 22-14: G1PUH: GPU WORK AREA HEIGHT REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	PUH10	PUH9	PUH8
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PUH7	PUH6	PUH5	PUH4	PUH3	PUH2	PUH1	PUH0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-11 **Unimplemented:** Read as '0'

bit 10-0 **PUH<10:0>:** GPU Work Area Height bits (in pixels)

REGISTER 22-15: G1DPADRL: DISPLAY BUFFER START ADDRESS REGISTER LOW

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DPADR15	DPADR14	DPADR13	DPADR12	DPADR11	DPADR10	DPADR9	DPADR8
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DPADR7	DPADR6	DPADR5	DPADR4	DPADR3	DPADR2	DPADR1	DPADR0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-0 **DPADR<15:0>:** Display Buffer Start Address Low bits
Display buffer start address must point to an even byte address in memory.

REGISTER 22-16: G1DPADRH: DISPLAY BUFFER START ADDRESS REGISTER HIGH

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DPADR23	DPADR22	DPADR21	DPADR20	DPADR19	DPADR18	DPADR17	DPADR16
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8 **Unimplemented:** Read as '0'
bit 7-0 **DPADR<23:16>:** Display Buffer Start Address High bits
Display buffer start address must point to an even byte address in memory.

REGISTER 22-17: G1DPW: DISPLAY BUFFER WIDTH REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	DPW10	DPW9	DPW8
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DPW7	DPW6	DPW5	DPW4	DPW3	DPW2	DPW1	DPW0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-11 **Unimplemented:** Read as '0'
bit 10-0 **DPW<10:0>:** Display Frame Width bits (in pixels)

REGISTER 22-18: G1DPH: DISPLAY BUFFER HEIGHT REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	DPH10	DPH9	DPH8
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DPH7	DPH6	DPH5	DPH4	DPH3	DPH2	DPH1	DPH0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-11 **Unimplemented:** Read as '0'

bit 10-0 **DPH<10:0>:** Display Frame Height bits (in pixels)

REGISTER 22-19: G1DPWT: DISPLAY TOTAL WIDTH REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	DPWT10	DPWT9	DPWT8
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DPWT7	DPWT6	DPWT5	DPWT4	DPWT3	DPWT2	DPWT1	DPWT0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-11 **Unimplemented:** Read as '0'

bit 10-0 **DPWT<10:0>:** Display Total Width bits (in pixels)

REGISTER 22-20: G1DPHT: DISPLAY TOTAL HEIGHT REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	DPHT10	DPHT9	DPHT8
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DPHT7	DPHT6	DPHT5	DPHT4	DPHT3	DPHT2	DPHT1	DPHT0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-11 **Unimplemented:** Read as '0'

bit 10-0 **DPHT<10:0>:** Display Total Height bits (in pixels)

REGISTER 22-21: G1ACTDA: ACTIVE DISPLAY AREA REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ACTLINE7	ACTLINE6	ACTLINE5	ACTLINE4	ACTLINE3	ACTLINE2	ACTLINE1	ACTLINE0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ACTPIX7	ACTPIX6	ACTPIX5	ACTPIX4	ACTPIX3	ACTPIX2	ACTPIX1	ACTPIX0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8 **ACTLINE<7:0>:** Number of Lines Before the First Active (Displayed) Line bits
Typically, ACTLINEx = VENSTx (G1DBLCON<15:8>).
This register is added for versatility in the timing of the active lines.
For TFT mode, DPMODE bits (G1CON2<2:0>) = 001; the minimum value is 2.
For STN mode, DPMODE bits (G1CON2<2:0>) = 010,011,100; the minimum value is 0.

bit 7-0 **ACTPIX<7:0>:** Number of Pixels Before the First Active (Displayed) Pixel bits (in DISPCLKs)
Typically, ACTPIXx = HENSTx (G1DBLCON<7:0>).
This register is added for versatility in the timing of the active pixels. Note that the programmed value is computed in DISPCLK cycles. This value is dependent on the DPGWDTH bit (G1CON2<15:14>). Refer to the "*PIC24F Family Reference Manual*", **Section 43. "Graphics Controller Module (GFX)"** for details.

REGISTER 22-22: G1HSYNC: HORIZONTAL SYNCHRONIZATION CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
HSLEN7	HSLEN6	HSLEN5	HSLEN4	HSLEN3	HSLEN2	HSLEN1	HSLEN0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
HSST7	HSST6	HSST5	HSST4	HSST3	HSST2	HSST1	HSST0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8 **HSLEN<7:0>:** HSYNC Pulse-Width Configuration bits (in DISPCLKs)
DPHSOE bit (G1CON3<0>) must be set for the HSYNC signal to toggle; minimum value is 1.

bit 7-0 **HSST<7:0>:** HSYNC Start Delay Configuration bits (in DISPCLKs)
This is the number of DISPCLK cycles from the start of horizontal blanking to the start of HSYNC active.

REGISTER 22-23: G1VSYNC: VERTICAL SYNCHRONIZATION CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
VSLEN7	VSLEN6	VSLEN5	VSLEN4	VSLEN3	VSLEN2	VSLEN1	VSLEN0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
VSST7	VSST6	VSST5	VSST4	VSST3	VSST2	VSST1	VSST0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8 **VSLEN<7:0>:** VSYNC Pulse-Width Configuration bits (in lines)
The DPVSOE bit (G1CON3<1>) must be set for the VSYNC signal to toggle; minimum value is 1.

bit 7-0 **VSST<7:0>:** VSYNC Start Delay Configuration bits (in lines)
This is the number of lines from the start of vertical blanking to the start of VSYNC active.

REGISTER 22-24: G1DBLCON: DISPLAY BLANKING CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
VENST7	VENST6	VENST5	VENST4	VENST3	VENST2	VENST1	VENST0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
HENST7	HENST6	HENST5	HENST4	HENST3	HENST2	HENST1	HENST0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8 **VENST<7:0>:** Vertical Blanking Start to First Displayed Line Configuration bits (in lines)
This is the number of lines from the start of vertical blanking to the first displayed line of a frame.

bit 7-0 **HENST<7:0>:** Horizontal Blanking Start to First Displayed Pixel Configuration bits (in DISPCLKs)
This is the number of GCLK cycles from the start of horizontal blanking to the first displayed pixel of each displayed line.

REGISTER 22-25: G1CLUT: COLOR LOOK-UP TABLE CONTROL REGISTER

R/W-0	R-0, HSC	U-0	U-0	U-0	U-0	R/W-0	R/W-0
CLUTEN	CLUTBUSY	—	—	—	—	CLUTTRD	CLUTRWEN
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CLUTADR7	CLUTADR6	CLUTADR5	CLUTADR4	CLUTADR3	CLUTADR2	CLUTADR1	CLUTADR0
bit 7							bit 0

Legend:	HSC = Hardware Settable/Clearable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **CLUTEN:** Color Look-up Table Enable Control bit
1 = Color look-up table is enabled
0 = Color look-up table is disabled

bit 14 **CLUTBUSY:** Color Look-up Table Busy Status bit
1 = A CLUT entry read/write access is being executed
0 = No CLUT entry read/write access is being executed

bit 13-10 **Unimplemented:** Read as '0'

bit 9 **CLUTTRD:** Color Look-up Table Read Trigger bit
Enabling this bit will trigger a read to the CLUT location determined by the CLUTADR bits (G1CLUT<7:0>) with CLUTRWEN enabled.
1 = CLUT read trigger is enabled (must be cleared in software after reading data in the G1CLUTRD register)
0 = CLUT read trigger is disabled

bit 8 **CLUTRWEN:** Color Look-up Table Read/Write Enable Control bit
This bit must be set when reading or modifying entries on the CLUT and it must also be cleared when CLUT is used by the display controller.
1 = Color look-up table read/write enabled; display controller cannot access the CLUT
0 = Color look-up table read/write disabled; display controller can access the CLUT

bit 7-0 **CLUTADR<7:0>:** Color Look-up Table Memory Address bits

REGISTER 22-26: G1CLUTWR: COLOR LOOK-UP TABLE (CLUT) MEMORY WRITE DATA REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CLUTWR15	CLUTWR14	CLUTWR13	CLUTWR12	CLUTWR11	CLUTWR10	CLUTWR9	CLUTWR8
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CLUTWR7	CLUTWR6	CLUTWR5	CLUTWR4	CLUTWR3	CLUTWR2	CLUTWR1	CLUTWR0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-0 **CLUTWR<15:0>:** Color Look-up Table Memory Write Data bits

A write to this register triggers a write to the CLUT memory at the address pointed to by the CLUTADR bits. A word write or a high byte write to this register triggers a write to the CLUT memory at the address pointed to by CLUTADR. Low byte write to this register will only update the G1CLUTWR<7:0> and no write to CLUT memory will be triggered. During power-up and power-down of the display, the most recent data written to this register will be used to control the timing of the GPWR signal. Refer to the "*PIC24F Family Reference Manual*", **Section 43. "Graphics Controller Module (GFX)"** for details on writing entries to the CLUT.

REGISTER 22-27: G1CLUTRD: COLOR LOOK-UP TABLE (CLUT) MEMORY READ DATA REGISTER

R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC
CLUTRD15	CLUTRD14	CLUTRD13	CLUTRD12	CLUTRD11	CLUTRD10	CLUTRD9	CLUTRD8
bit 15							bit 8

R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC
CLUTRD7	CLUTRD6	CLUTRD5	CLUTRD4	CLUTRD3	CLUTRD2	CLUTRD1	CLUTRD0
bit 7							bit 0

Legend:	HSC = Hardware Settable/Clearable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-0 **CLUTRD<15:0>:** Color Look-up Table Memory Read Data bits

This register contains the most recent read from the CLUT memory pointed to by the CLUTADR bits (G1CLUT<7:0>). Reading of the CLUT memory is triggered when the CLUTTRD bit (G1CLUT<9>) goes from '0' to '1'. Refer to the "*PIC24F Family Reference Manual*", **Section 43. "Graphics Controller Module (GFX)"** for details on reading entries from the CLUT.

REGISTER 22-28: G1MRGN: INTERRUPT ADVANCE REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
VBAMGN7	VBAMGN6	VBAMGN5	VBAMGN4	VBAMGN3	VBAMGN2	VBAMGN1	VBAMGN0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
HBAMGN7	HBAMGN6	HBAMGN5	HBAMGN4	HBAMGN3	HBAMGN2	HBAMGN1	HBAMGN0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-8 **VBAMGN<7:0>:** Vertical Blanking Advance bits
The number of DISPCLK cycles in advance that the vertical blanking interrupt will assert ahead of the actual start of the vertical blanking.

bit 7-0 **HBAMGN<7:0>:** Horizontal Blanking Advance bits
The number of DISPCLK cycles in advance that the horizontal blanking interrupt will assert ahead of the actual start of the horizontal blanking.

REGISTER 22-29: G1CHRX: CHARACTER-X COORDINATE PRINT POSITION REGISTER

U-0	U-0	U-0	U-0	U-0	R-0, HSC	R-0, HSC	R-0, HSC
—	—	—	—	—	CURPOSX10	CURPOSX9	CURPOSX8
bit 15							bit 8

R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC
CURPOSX7	CURPOSX6	CURPOSX5	CURPOSX4	CURPOSX3	CURPOSX2	CURPOSX1	CURPOSX0
bit 7							bit 0

Legend:	HSC = Hardware Settable/Clearable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-11 **Unimplemented:** Read as '0'

bit 10-0 **CURPOSX<10:0>:** Current Character Position in the X-Coordinate bits

REGISTER 22-30: G1CHRY: CHARACTER Y-COORDINATE PRINT POSITION REGISTER

U-0	U-0	U-0	U-0	U-0	R-0, HSC	R-0, HSC	R-0, HSC
—	—	—	—	—	CURPOSY10	CURPOSY9	CURPOSY8
bit 15							bit 8

R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC
CURPOSY7	CURPOSY6	CURPOSY5	CURPOSY4	CURPOSY3	CURPOSY2	CURPOSY1	CURPOSY0
bit 7							bit 0

Legend:	HSC = Hardware Settable/Clearable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-11 **Unimplemented:** Read as '0'

bit 10-0 **CURPOSY<10:0>:** Current Character Position in the Y-Coordinate bits

REGISTER 22-31: G1IPU: INFLATE PROCESSOR STATUS REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC
—	—	HUFFERR(2)	BLCKERR(2)	LENERR(2)	WRAPERR(2)	IPUDONE(1,2)	BFINAL(1,2)
bit 7							bit 0

Legend:	HSC = Hardware Settable/Clearable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-6 **Unimplemented:** Read as '0'

bit 5 **HUFFERR:** Undefined Huffmann Code Encountered Status bit(2)
- 1 = Undefined code is encountered
- 0 = No undefined code is encountered

bit 4 **BLCKERR:** Undefined Block Code Encountered Status bit(2)
- 1 = Undefined block is encountered
- 0 = No undefined block is encountered

bit 3 **LENERR:** Mismatch in Expected Block Length Status bit(2)
- 1 = Mismatch in block length is detected
- 0 = No mismatch in block length is detected

bit 2 **WRAPERR:** Wrap-Around Error Status bit(2)
- 1 = Wrap-around error is encountered
- 0 = No wrap-around error is encountered

bit 1 **IPUDONE:** IPU Decompression Status bit(1,2)
- 1 = Decompression is done
- 0 = Decompression is not yet done

bit 0 **BFINAL:** Final Block Encountered Status bit(1,2)
- 1 = Final block is encountered
- 0 = Final block is not encountered

Note 1: IPUDONE and BFINAL status bits are set after successful decompression.

2: All IPU status bits are available after each decompression. All status bits are automatically cleared every time a decompression command (IPU_DECOMPRESS) is issued.

REGISTER 22-32: G1DBEN: DATA I/O PAD ENABLE REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GDBEN15	GDBEN14	GDBEN13	GDBEN12	GDBEN11	GDBEN10	GDBEN9	GDBEN8
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GDBEN7	GDBEN6	GDBEN5	GDBEN4	GDBEN3	GDBEN2	GDBEN1	GDBEN0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-0 **GDBEN<15:0>:** Display Data Pads Output Enable bits

1 = Corresponding display data (GD<x>) pin is enabled
0 = Corresponding display data (GD<x>) pin is disabled

GDBEN<15:0> can be used to disable or enable specific data signals while the DPPINOE bit (G1CON3<9>) is set.

DPPINOE	GDBENx (where x = 0 to 15)	
1	1	Display data signal (GD) associated with GDBENx is enabled.
1	0	Display data signal (GD) associated with GDBENx is disabled.
0	x	Display data signal (GD) associated with GDBENx is disabled.

22.2 Display Resolution and Memory Requirements

The PIC24FJ256DA210 family of devices has two variants in terms of on-board RAM (24-Kbyte and 96-Kbyte variants). The 24-Kbyte variant supports monochrome displays while the 96-Kbyte variant supports Quarter VGA (QVGA) color displays, up to 256 colors. Support of higher resolution displays with higher color depth requirements are available by extending the data space through external memory. Table 22-1 provides the summary of image buffer memory requirements of different display resolutions and color depth requirements.

22.3 Display Clock (GCLK) Source

Frequency of the Graphics Controller Display Clock (GCLK) signal is determined by programming the GCLKDIV bits (CLKDIV2<15:9>). For more information, refer to the "*PIC24F Family Reference Manual*", **Section 6. "Oscillator" (**DS39700**)**.

22.4 Display Buffer and Work Areas Memory Locations

The PIC24FJ256DA210 family of devices has variants with two on-board RAM sizes. These are the 24-Kbyte and 96-Kbyte variants. These two RAM variants are further divided in terms of pin counts. The 100-pin count device will have the EPMP module available for extending RAM for applications. The 64-pin count device will not have the EPMP modules. Extending the RAM size is necessary for applications that require larger display buffers and work areas. It is recommended that the display buffers and work areas are **not** mapped into an area that overlaps the internal RAM and the external RAM. The external RAM can be interfaced using the EPMP module. For details, refer to the "*PIC24F Family Reference Manual*", **Section 42. "Enhanced Parallel Master Port (EPMP)"** (DS39730).

TABLE 22-1: BUFFER MEMORY REQUIREMENTS vs. DISPLAY CONFIGURATION

Display Resolution	Display Buffer Memory Requirements (Bytes)				
	1 Bpp	2 Bpp	4 Bpp	8 Bpp	16 Bpp
480x272 (WQVGA)	16320	32640	65280	130560	261120
320x240 (QVGA)	9600	19200	38400	76800	153600
240x160 (HQVGA)	4800	9600	19200	38400	76800
160x160	3200	6400	12800	25600	51200
160x120 (QQVGA)	2400	4800	9600	19200	38400
128x64	1024	2048	4096	8192	16384

Legend:

Less than 24-Kbyte RAM variants (PIC24FJXXXDA106)

Less than 96-Kbyte RAM variants (PIC24FJXXXDA2XX)

External Memory with 96 Kbytes/24 Kbytes of RAM variants (PIC24FJXXXDAX10)

23.0 10-BIT HIGH-SPEED A/D CONVERTER

Note: This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the "*PIC24F Family Reference Manual*", **Section 17. "10-Bit A/D Converter"** (DS39705). The information in this data sheet supersedes the information in the FRM.

The 10-bit A/D Converter has the following key features:

- Successive Approximation (SAR) conversion
- Conversion speeds of up to 500 ksps
- 24 analog input pins (PIC24FJXXXDAX10 devices) and 16 analog input pins (PIC24FJXXXDAX06 devices)
- External voltage reference input pins
- Internal band gap reference inputs
- Automatic Channel Scan mode
- Selectable conversion trigger source
- 32-word conversion result buffer
- Selectable Buffer Fill modes
- Four result alignment options
- Operation during CPU Sleep and Idle modes

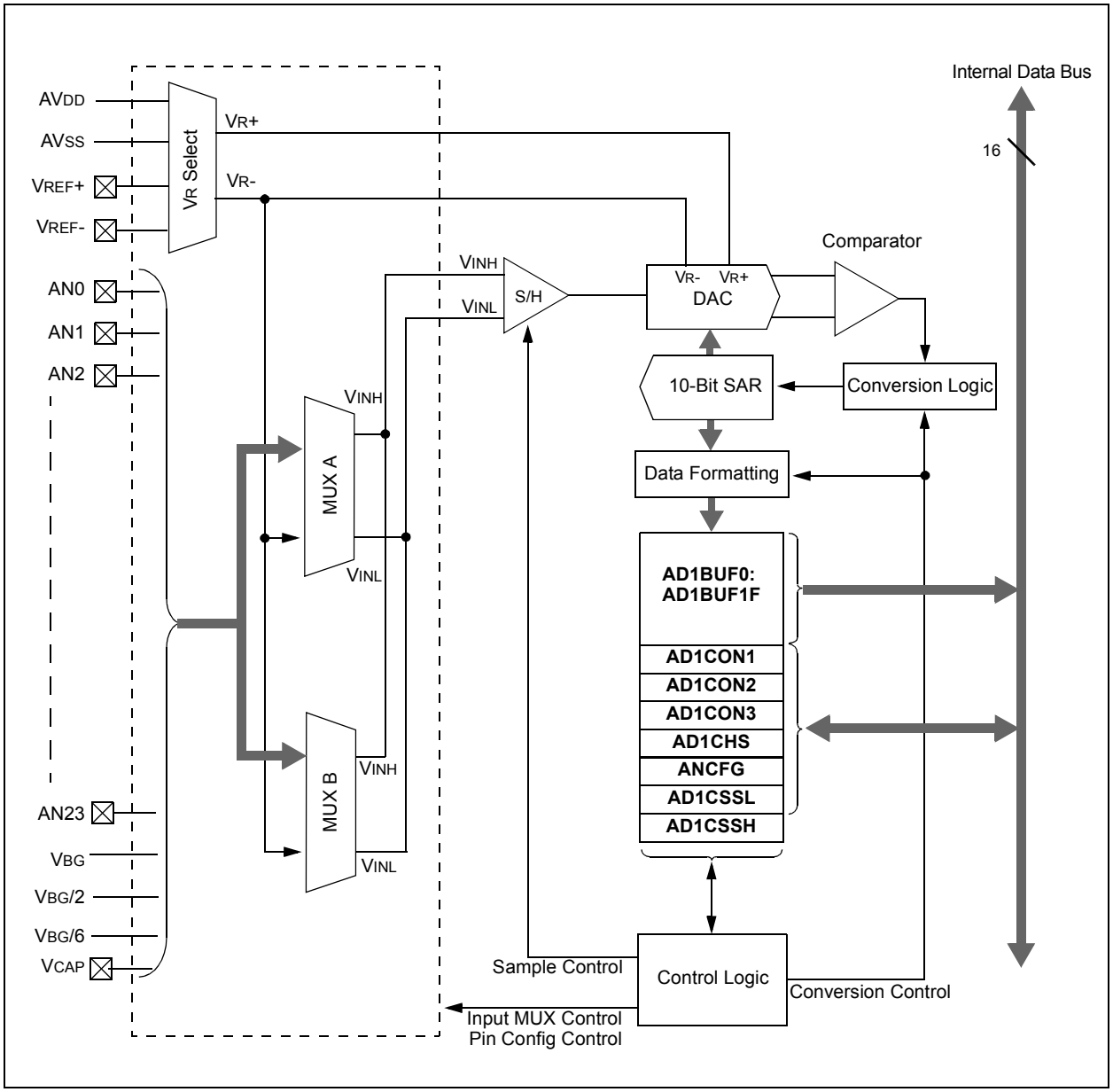
On all PIC24FJ256DA210 family devices, the 10-bit A/D Converter has 24 analog input pins, designated AN0 through AN23. In addition, there are two analog input pins for external voltage reference connections (V_{REF}+ and V_{REF}-). These voltage reference inputs may be shared with other analog input pins.

A block diagram of the A/D Converter is shown in Figure 23-1.

To perform an A/D conversion:

1. Configure the A/D module:
   a) Configure the port pins as analog inputs and/or select band gap reference inputs (ANCFG registers).
   b) Select the voltage reference source to match the expected range on analog inputs (AD1CON2<15:13>).
   c) Select the analog conversion clock to match the desired data rate with the processor clock (AD1CON3<7:0>).
   d) Select the appropriate sample/conversion sequence (AD1CON1<7:5> and AD1CON3<12:8>).
   e) Select how the conversion results are presented in the buffer (AD1CON1<9:8>).
   f) Select the interrupt rate (AD1CON2<6:2>).
   g) Turn on the A/D module (AD1CON1<15>).
2. Configure the A/D interrupt (if required):
   a) Clear the AD1IF bit.
   b) Select the A/D interrupt priority.

FIGURE 23-1: 10-BIT HIGH-SPEED A/D CONVERTER BLOCK DIAGRAM



REGISTER 23-1: AD1CON1: A/D CONTROL REGISTER 1

R/W-0	U-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
ADON(1)	—	ADSIDL	—	—	—	FORM1	FORM0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R-0, HSC	R-0, HSC
SSRC2	SSRC1	SSRC0	—	—	ASAM	SAMP	DONE
bit 7							bit 0

Legend:	HSC = Hardware Settable/Clearable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **ADON:** A/D Operating Mode bit(1)
1 = A/D Converter module is operating
0 = A/D Converter is off

bit 14 **Unimplemented:** Read as '0'

bit 13 **ADSIDL:** Stop in Idle Mode bit
1 = Discontinue module operation when device enters Idle mode
0 = Continue module operation in Idle mode

bit 12-10 **Unimplemented:** Read as '0'

bit 9-8 **FORM<1:0>:** Data Output Format bits
11 = Signed fractional (sddd dddd dd00 0000)
10 = Fractional (dddd dddd dd00 0000)
01 = Signed integer (ssss sssd dddd dddd)
00 = Integer (0000 00dd dddd dddd)

bit 7-5 **SSRC<2:0>:** Conversion Trigger Source Select bits
111 = Internal counter ends sampling and starts conversion (auto-convert)
110 = CTMU event ends sampling and starts conversion
101 = Reserved
100 = Timer5 compare ends sampling and starts conversion
011 = Reserved
010 = Timer3 compare ends sampling and starts conversion
001 = Active transition on INT0 pin ends sampling and starts conversion
000 = Clearing SAMP bit ends sampling and starts conversion

bit 4-3 **Unimplemented:** Read as '0'

bit 2 **ASAM:** A/D Sample Auto-Start bit
1 = Sampling begins immediately after the last conversion completes. The SAMP bit is auto-set.
0 = Sampling begins when the SAMP bit is set

bit 1 **SAMP:** A/D Sample Enable bit
1 = A/D sample/hold amplifier is sampling input
0 = A/D sample/hold amplifier is holding

bit 0 **DONE:** A/D Conversion Status bit
1 = A/D conversion is done
0 = A/D conversion is NOT done

Note 1: The values of the ADC1BUFx registers will not retain their values once the ADON bit is cleared. Read out the conversion values from the buffer before disabling the module.

REGISTER 23-2: AD1CON2: A/D CONTROL REGISTER 2

R/W-0	R/W-0	R/W-0	r-0	U-0	R/W-0	U-0	U-0
VCFG2	VCFG1	VCFG0	r	—	CSCNA	—	—
bit 15							bit 8

R-0, HSC	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BUFS	SMPI4	SMPI3	SMPI2	SMPI1	SMPI0	BUFM	ALTS
bit 7							bit 0

Legend:	r = Reserved bit	HSC = Hardware Settable/Clearable bit	
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-13 **VCFG<2:0>:** Voltage Reference Configuration bits

VCFG<2:0>	VR+	VR-
000	AVDD	AVSS
001	External VREF+ pin	AVSS
010	AVDD	External VREF- pin
011	External VREF+ pin	External VREF- pin
1xx	AVDD	AVSS

bit 12 **Reserved:** Maintain as '0'

bit 11 **Unimplemented:** Read as '0'

bit 10 **CSCNA:** Scan Input Selections for the CH0+ S/H Input for MUX A Input Multiplexer Setting bit
1 = Scan inputs
0 = Do not scan inputs

bit 9-8 **Unimplemented:** Read as '0'

bit 7 **BUFS:** Buffer Fill Status bit (valid only when BUFM = 1)
1 = A/D is currently filling buffer, 10-1F, user should access data in 00-0F
0 = A/D is currently filling buffer, 00-0F, user should access data in 10-1F

bit 6-2 **SMPI<4:0>:** Sample/Convert Sequences Per Interrupt Selection bits
11111 = Interrupts at the completion of conversion for each 32nd sample/convert sequence
11110 = Interrupts at the completion of conversion for each 31st sample/convert sequence
.
.
.
00001 = Interrupts at the completion of conversion for each 2nd sample/convert sequence
00000 = Interrupts at the completion of conversion for each sample/convert sequence

bit 1 **BUFM:** Buffer Mode Select bit
1 = Buffer is configured as two 16-word buffers (ADC1BUFn<31:16> and ADC1BUFn<15:0>)
0 = Buffer is configured as one 32-word buffer (ADC1BUFn<31:0>)

bit 0 **ALTS:** Alternate Input Sample Mode Select bit
1 = Uses MUX A input multiplexer settings for the first sample, then alternates between MUX B and MUX A input multiplexer settings for all subsequent samples
0 = Always uses the MUX A input multiplexer settings

REGISTER 23-3: AD1CON3: A/D CONTROL REGISTER 3

R/W-0	r-0	r-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADRC	r	r	SAMC4	SAMC3	SAMC2	SAMC1	SAMC0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADCS7	ADCS6	ADCS5	ADCS4	ADCS3	ADCS2	ADCS1	ADCS0
bit 7							bit 0

Legend:	r = Reserved bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **ADRC:** A/D Conversion Clock Source bit
1 = A/D internal RC clock
0 = Clock is derived from the system clock

bit 14-13 **Reserved**: Maintain as '0'

bit 12-8 **SAMC<4:0>:** Auto-Sample Time bits
11111 = 31 TAD
.
.
.
00001 = 1 TAD
00000 = 0 TAD (not recommended)

bit 7-0 **ADCS<7:0>:** A/D Conversion Clock Select bits
11111111 = 256 * TCY
.....
00000001 = 2 * TCY
00000000 = TCY

REGISTER 23-4: AD1CHS: A/D INPUT SELECT REGISTER

R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CH0NB	—	—	CH0SB4(1)	CH0SB3(1)	CH0SB2(1)	CH0SB1(1)	CH0SB0(1)
bit 15							bit 8

R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CH0NA	—	—	CH0SA4(1)	CH0SA3(1)	CH0SA2(1)	CH0SA1(1)	CH0SA0(1)
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **CH0NB:** Channel 0 Negative Input Select for MUX B Multiplexer Setting bit
1 = Channel 0 negative input is AN1
0 = Channel 0 negative input is VR-

bit 14-13 **Unimplemented:** Read as '0'

bit 12-8 **CH0SB<4:0>:** Channel 0 Positive Input Select for MUX B(1)
Other = Not available; do not use
11111 = No channel used; all inputs are floating; used for CTMU
11011 = Channel 0 positive input is the band gap divided-by-six reference (VBG/6)
11010 = Channel 0 positive input is the core voltage (VCAP)
11001 = Channel 0 positive input is the band gap reference (VBG)
11000 = Channel 0 positive input is the band gap divided-by-two reference (VBG/2)
10111 = Channel 0 positive input is AN23(2)
.
.
.
00001 = Channel 0 positive input is AN1
00000 = Channel 0 positive input is AN0

bit 7 **CH0NA:** Channel 0 Negative Input Select for MUX A Multiplexer Setting bit
1 = Channel 0 negative input is AN1
0 = Channel 0 negative input is VR-

bit 6-5 **Unimplemented:** Read as '0'

bit 4-0 **CH0SA<4:0>:** Channel 0 Positive Input Select for MUX(1)
Other = Not available; do not use
11111 = No Channel used; all inputs are floating; used for CTMU
11011 = Channel 0 positive input is the band gap divided-by-six reference (VBG/6)
11010 = Channel 0 positive input is the core voltage (VCAP)
11001 = Channel 0 positive input is the band gap reference (VBG)
11000 = Channel 0 positive input is the band gap divided-by-two reference (VBG/2)
10111 = Channel 0 positive input is AN23(2)
.
.
.
00001 = Channel 0 positive input is AN1
00000 = Channel 0 positive input is AN0

Note 1: Combinations not shown here (11100 to 11110) are unimplemented; do not use.
2: Channel 0 positive inputs, AN16 through AN23, are not available on 64-pin devices (PIC24FJXXXDAX06).

REGISTER 23-5: ANCFG: A/D BAND GAP REFERENCE CONFIGURATION REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	VBG6EN	VBG2EN	VBGEN
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-3 **Unimplemented:** Read as '0'

bit 2 **VBG6EN:** A/D Input VBG/6 Enable bit
1 = Band gap voltage divided-by-six reference (VBG/6) is enabled
0 = Band gap divided-by-six reference (VBG/6) is disabled

bit 1 **VBG2EN:** A/D Input VBG/2 Enable bit
1 = Band gap voltage divided-by-two reference (VBG/2) is enabled
0 = Band gap divided-by-two reference (VBG/2) is disabled

bit 0 **VBGEN:** A/D Input VBG Enable bit
1 = Band gap voltage reference (VBG) is enabled
0 = Band gap reference (VBG) is disabled

REGISTER 23-6: AD1CSSL: A/D INPUT SCAN SELECT REGISTER (LOW)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CSSL15	CSSL14	CSSL13	CSSL12	CSSL11	CSSL10	CSSL9	CSSL8
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CSSL7	CSSL6	CSSL5	CSSL4	CSSL3	CSSL2	CSSL1	CSSL0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-0 **CSSL<15:0>:** A/D Input Pin Scan Selection bits
1 = Corresponding analog channel is selected for input scan
0 = Analog channel is omitted from input scan

REGISTER 23-7: AD1CSSH: A/D INPUT SCAN SELECT REGISTER (HIGH)

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	CSSL27	CSSL26	CSSL25	CSSL24
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CSSL23(1)	CSSL22(1)	CSSL21(1)	CSSL20(1)	CSSL19(1)	CSSL18(1)	CSSL17(1)	CSSL16(1)
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-12 **Unimplemented:** Read as '0'

bit 11 **CSSL27:** A/D Input Band Gap Scan Selection bit
1 = Band gap divided-by-six reference (VBG/6) is selected for input scan
0 = Analog channel is omitted from input scan

bit 10 **CSSL26:** A/D Input Band Gap Scan Selection bit
1 = Internal core voltage (VCAP) is selected for input scan
0 = Analog channel is omitted from input scan

bit 9 **CSSL25:** A/D Input Half Band Gap Scan Selection bit
1 = Band gap reference (VBG) is selected for input scan
0 = Analog channel is omitted from input scan

bit 8 **CSSL24:** A/D Input Band Gap Scan Selection bit
1 = Band gap divided-by-two reference (VBG/2) is selected for input scan
0 = Analog channel is omitted from input scan

bit 7-0 **CSSL<23:16>:** Analog Input Pin Scan Selection bits(1)
1 = Corresponding analog channel selected for input scan
0 = Analog channel is omitted from input scan

Note 1: Unimplemented in 64-pin devices, read as '0'.

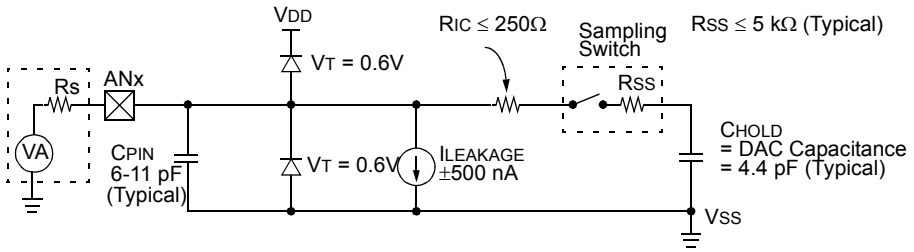
EQUATION 23-1: A/D CONVERSION CLOCK PERIOD(1)

$$ADCS = \frac{TAD}{TCY} - 1$$

$$TAD = TCY \bullet (ADCS = 1)$$

Note 1: Based on TCY = 2 * TOSC; Doze mode and PLL are disabled.

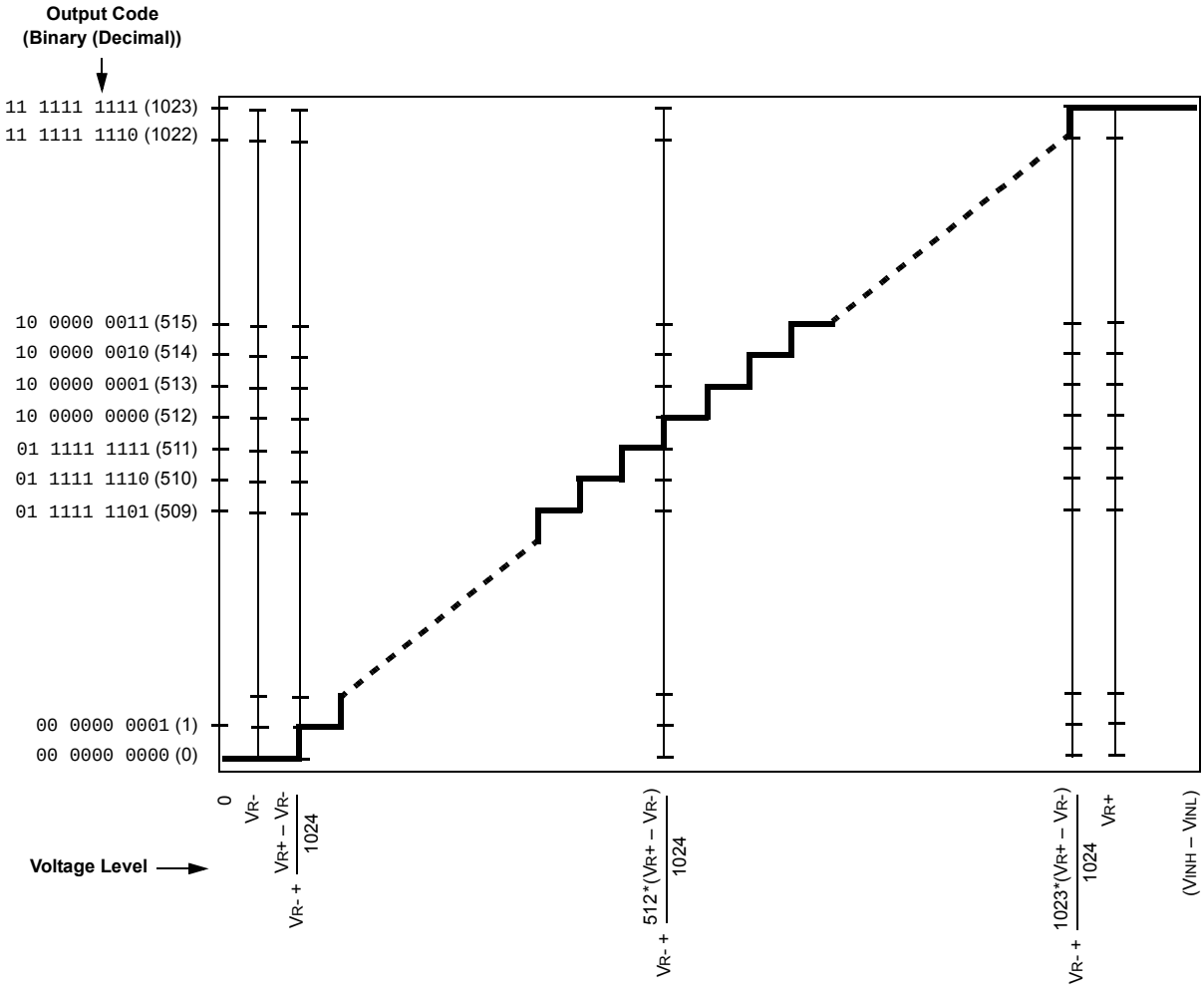
FIGURE 23-2: 10-BIT A/D CONVERTER ANALOG INPUT MODEL



Legend: CPIN = Input Capacitance
VT = Threshold Voltage
ILEAKAGE = Leakage Current at the pin due to various junctions
RIC = Interconnect Resistance
RSS = Sampling Switch Resistance
CHOLD = Sample/Hold Capacitance (from DAC)

Note: CPIN value depends on the device package and is not tested. The effect of CPIN IS negligible if Rs ≤ 5 kΩ.

FIGURE 23-3: A/D TRANSFER FUNCTION



NOTES:

24.0 TRIPLE COMPARATOR MODULE

Note: This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the associated "*PIC24F Family Reference Manual*".

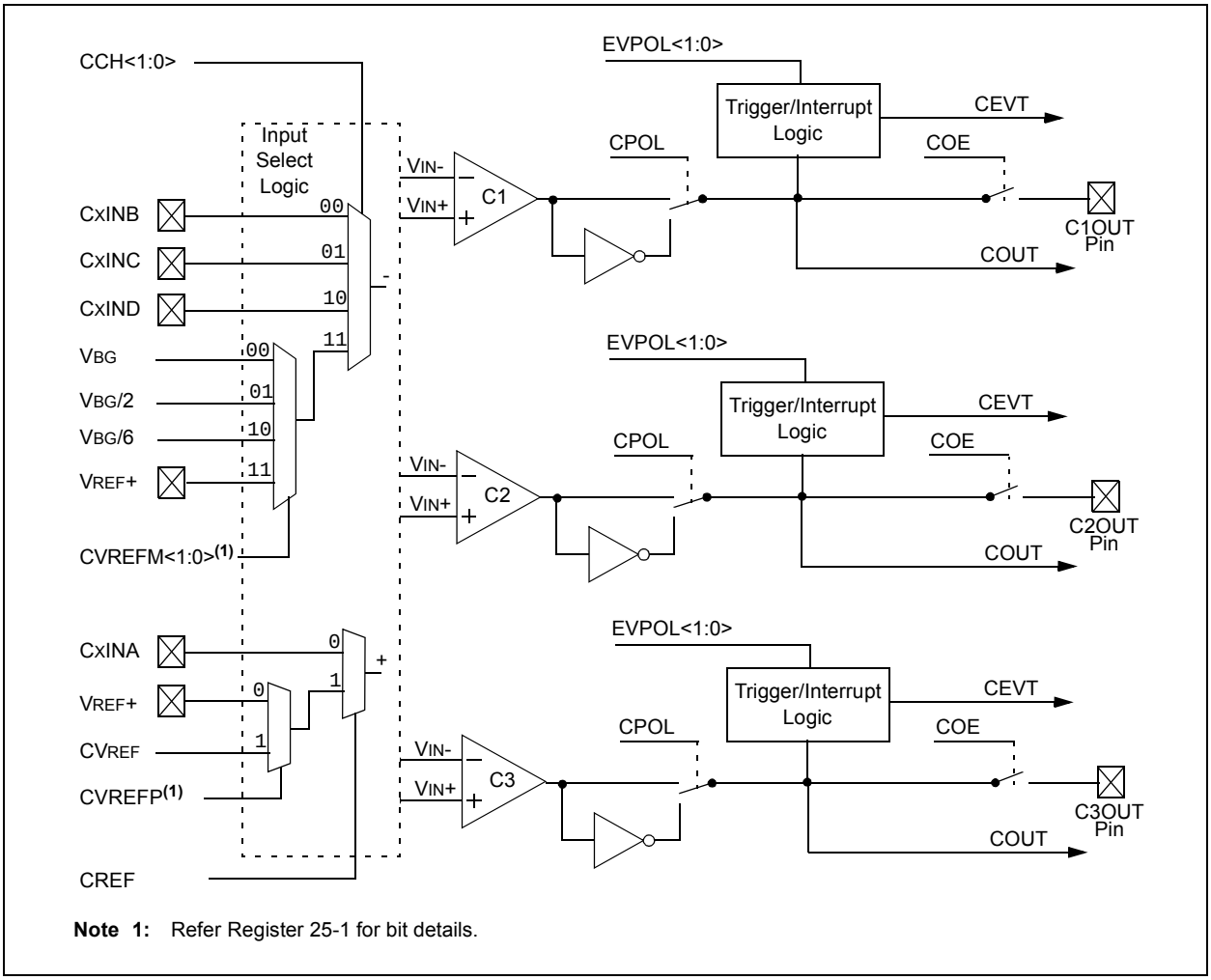
The triple comparator module provides three dual input comparators. The inputs to the comparator can be configured to use any one of five external analog inputs (CxINA, CxINB, CxINC, CxIND and VREF+) and a voltage reference input from one of the internal band gap references or the comparator voltage reference generator (VBG, VBG/2, VBG/6 and CVREF).

The comparator outputs may be directly connected to the CxOUT pins. When the respective COE equals '1', the I/O pad logic makes the unsynchronized output of the comparator available on the pin.

A simplified block diagram of the module in shown in Figure 24-1. Diagrams of the possible individual comparator configurations are shown in Figure 24-2.

Each comparator has its own control register, CMxCON (Register 24-1), for enabling and configuring its operation. The output and event status of all three comparators is provided in the CMSTAT register (Register 24-2).

FIGURE 24-1: TRIPLE COMPARATOR MODULE BLOCK DIAGRAM



Note 1: Refer Register 25-1 for bit details.

FIGURE 24-2: INDIVIDUAL COMPARATOR CONFIGURATIONS WHEN CREF = 0

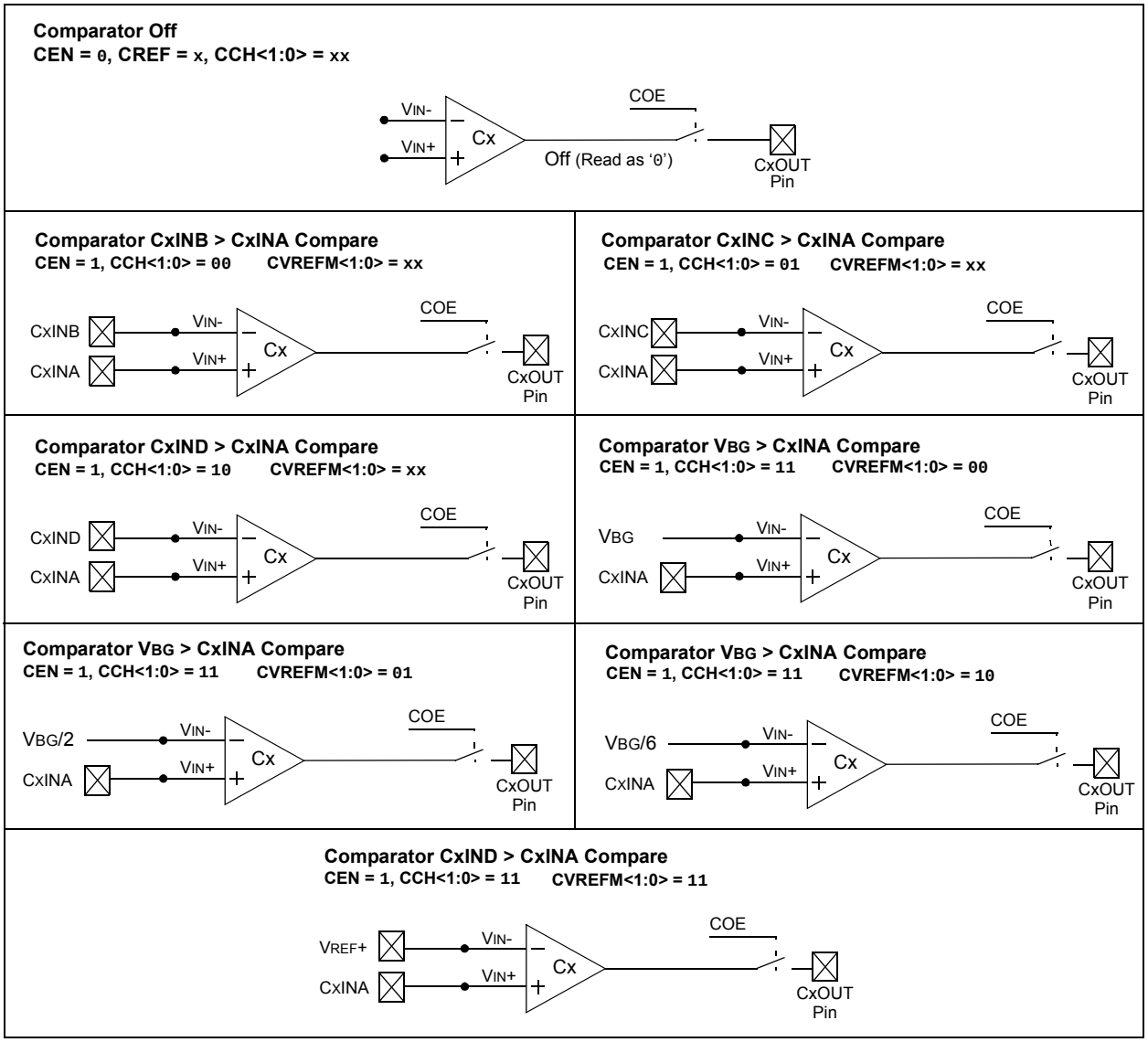


FIGURE 24-3: INDIVIDUAL COMPARATOR CONFIGURATIONS WHEN CREF = 1 AND CVREFP = 0

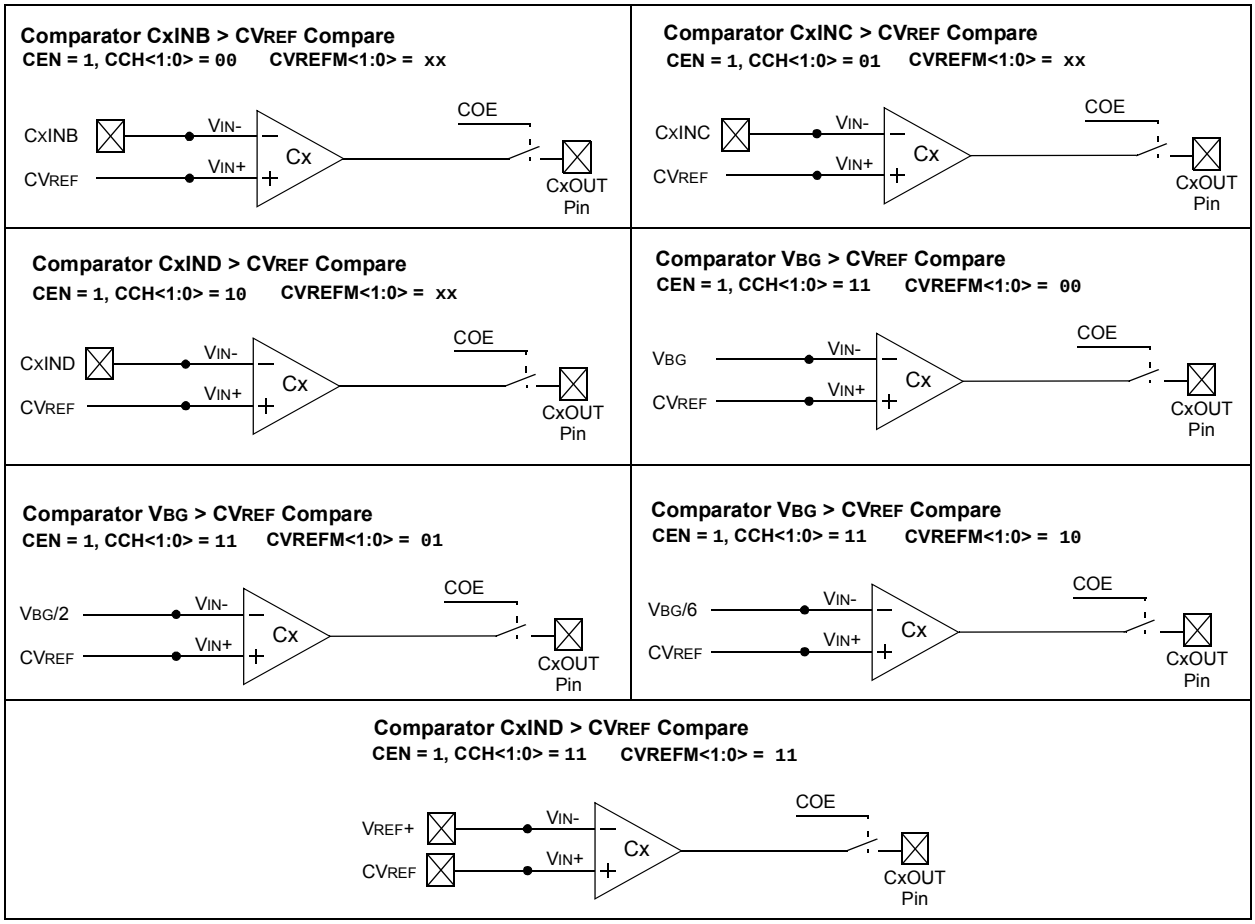
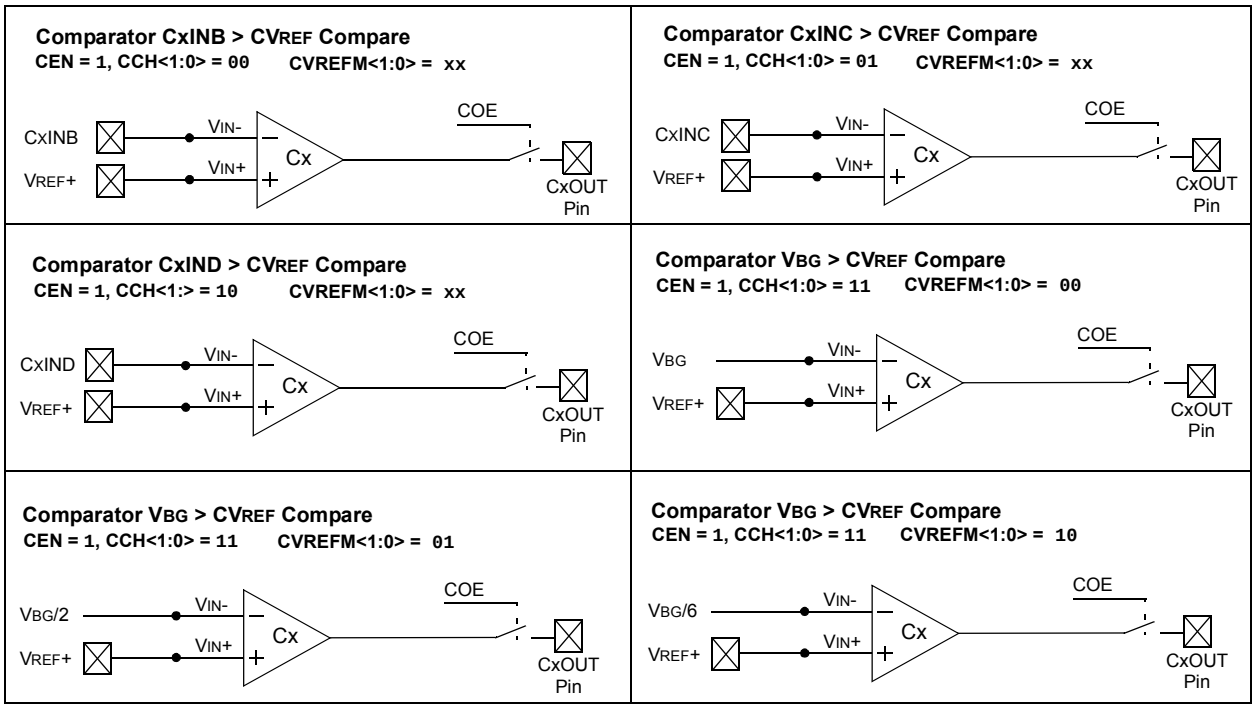


FIGURE 24-4: INDIVIDUAL COMPARATOR CONFIGURATIONS WHEN CREF = 1 AND CVREFP = 1



REGISTER 24-1: CMxCON: COMPARATOR x CONTROL REGISTERS (COMPARATORS 1 THROUGH 3)

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0, HS	R-0, HSC
CEN	COE	CPOL	—	—	—	CEVT	COUT
bit 15							bit 8

R/W-0	R/W-0	U-0	R/W-0	U-0	U-0	R/W-0	R/W-0
EVPOL1	EVPOL0	—	CREF	—	—	CCH1	CCH0
bit 7							bit 0

Legend:	HS = Hardware Settable bit	HSC = Hardware Settable/Clearable bit	
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **CEN:** Comparator Enable bit
1 = Comparator is enabled
0 = Comparator is disabled

bit 14 **COE:** Comparator Output Enable bit
1 = Comparator output is present on the CxOUT pin
0 = Comparator output is internal only

bit 13 **CPOL:** Comparator Output Polarity Select bit
1 = Comparator output is inverted
0 = Comparator output is not inverted

bit 12-10 **Unimplemented:** Read as '0'

bit 9 **CEVT:** Comparator Event bit
1 = Comparator event that is defined by EVPOL<1:0> has occurred; subsequent triggers and interrupts are disabled until the bit is cleared
0 = Comparator event has not occurred

bit 8 **COUT:** Comparator Output bit
When CPOL = 0:
1 = VIN+ > VIN-
0 = VIN+ < VIN-
When CPOL = 1:
1 = VIN+ < VIN-
0 = VIN+ > VIN-

bit 7-6 **EVPOL<1:0>:** Trigger/Event/Interrupt Polarity Select bits
11 = Trigger/event/interrupt is generated on any change of the comparator output (while CEVT = 0)
10 = Trigger/event/interrupt is generated on transition of the comparator output:
If CPOL = 0 (non-inverted polarity):
High-to-low transition only.
If CPOL = 1 (inverted polarity):
Low-to-high transition only.
01 = Trigger/event/interrupt is generated on transition of comparator output:
If CPOL = 0 (non-inverted polarity):
Low-to-high transition only.
If CPOL = 1 (inverted polarity):
High-to-low transition only.
00 = Trigger/event/interrupt generation is disabled

bit 5 **Unimplemented:** Read as '0'

REGISTER 24-1: CMxCON: COMPARATOR x CONTROL REGISTERS (COMPARATORS 1 THROUGH 3) (CONTINUED)

bit 4 **CREF:** Comparator Reference Select bits (non-inverting input)
1 = Non-inverting input connects to the internal CVREF voltage
0 = Non-inverting input connects to the CxINA pin

bit 3-2 **Unimplemented:** Read as '0'

bit 1-0 **CCH<1:0>:** Comparator Channel Select bits
11 = Inverting input of the comparator connects to the internal selectable reference voltage specified by the CVREFM<1:0> bits in the CVRCON register
10 = Inverting input of the comparator connects to the CxIND pin
01 = Inverting input of the comparator connects to the CxINC pin
00 = Inverting input of the comparator connects to the CxINB pin

REGISTER 24-2: CMSTAT: COMPARATOR MODULE STATUS REGISTER

R/W-0	U-0	U-0	U-0	U-0	R-0, HSC	R-0, HSC	R-0, HSC
CMIDL	—	—	—	—	C3EVT	C2EVT	C1EVT
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R-0, HSC	R-0, HSC	R-0, HSC
—	—	—	—	—	C3OUT	C2OUT	C1OUT
bit 7							bit 0

Legend:	HSC = Hardware Settable/Clearable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **CMIDL:** Comparator Stop in Idle Mode bit
1 = Discontinue operation of all comparators when device enters Idle mode
0 = Continue operation of all enabled comparators in Idle mode

bit 14-11 **Unimplemented:** Read as '0'

bit 10 **C3EVT:** Comparator 3 Event Status bit (read-only)
Shows the current event status of Comparator 3 (CM3CON<9>).

bit 9 **C2EVT:** Comparator 2 Event Status bit (read-only)
Shows the current event status of Comparator 2 (CM2CON<9>).

bit 8 **C1EVT:** Comparator 1 Event Status bit (read-only)
Shows the current event status of Comparator 1 (CM1CON<9>).

bit 7-3 **Unimplemented:** Read as '0'

bit 2 **C3OUT:** Comparator 3 Output Status bit (read-only)
Shows the current output of Comparator 3 (CM3CON<8>).

bit 1 **C2OUT:** Comparator 2 Output Status bit (read-only)
Shows the current output of Comparator 2 (CM2CON<8>).

bit 0 **C1OUT:** Comparator 1 Output Status bit (read-only)
Shows the current output of Comparator 1 (CM1CON<8>).

NOTES:

25.0 COMPARATOR VOLTAGE REFERENCE

Note: This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the "*PIC24F Family Reference Manual*", **Section 19. "Comparator Module"** (DS39710). The information in this data sheet supersedes the information in the FRM.

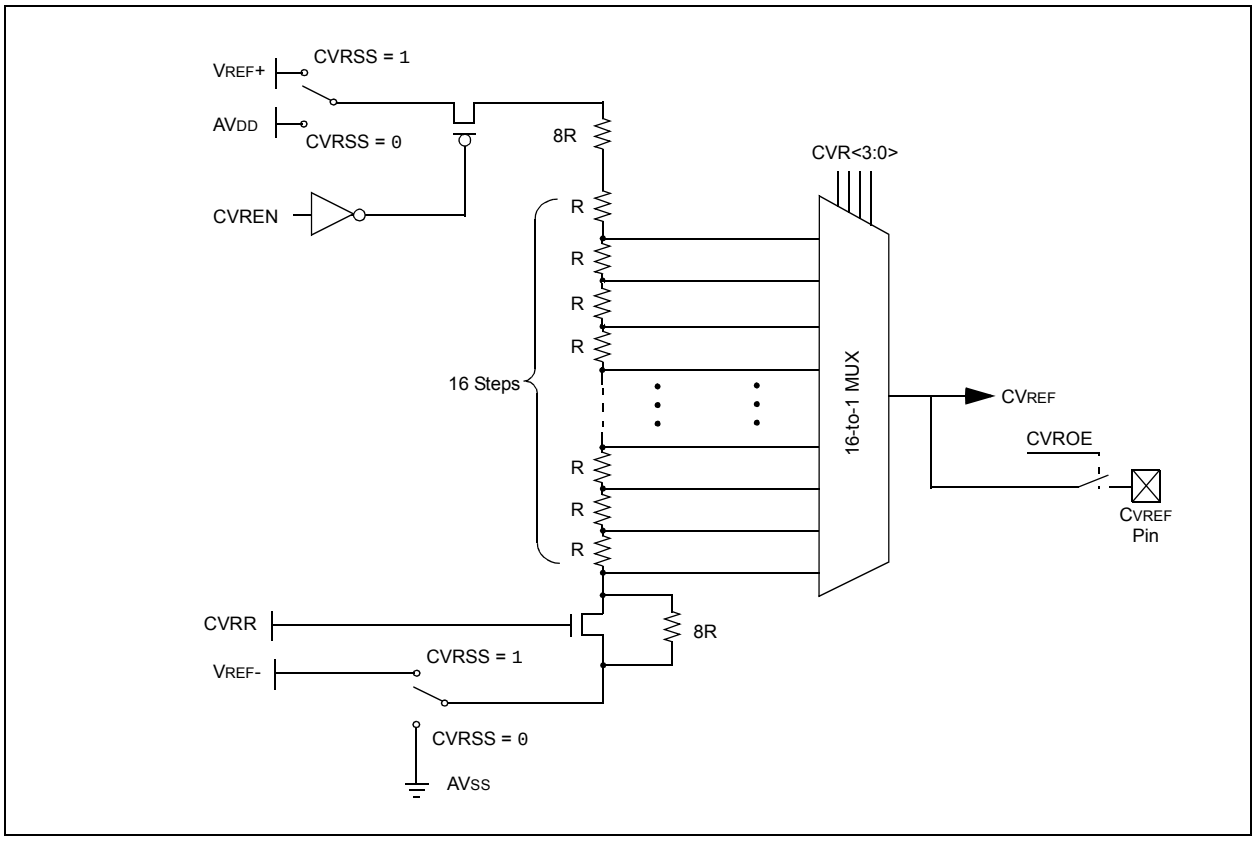
25.1 Configuring the Comparator Voltage Reference

The voltage reference module is controlled through the CVRCON register (Register 25-1). The comparator voltage reference provides two ranges of output voltage, each with 16 distinct levels. The range to be used is selected by the CVRR bit (CVRCON<5>). The primary difference between the ranges is the size of the steps selected by the CVREF Selection bits (CVR<3:0>), with one range offering finer resolution.

The comparator reference supply voltage can come from either VDD and VSS, or the external VREF+ and VREF-. The voltage source is selected by the CVRSS bit (CVRCON<4>).

The settling time of the comparator voltage reference must be considered when changing the CVREF output.

FIGURE 25-1: COMPARATOR VOLTAGE REFERENCE BLOCK DIAGRAM



REGISTER 25-1: CVRCON: COMPARATOR VOLTAGE REFERENCE CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	CVREFP	CVREFM1	CVREFM0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-11 **Unimplemented:** Read as '0'

bit 10 **CVREFP:** Voltage Reference Select bit (valid only when CREF is '1')
1 = VREF+ is used as a reference voltage to the comparators
0 = The CVR (4-bit DAC) within this module provides the the reference voltage to the comparators

bit 9-8 **CVREFM<1:0>:** Band Gap Reference Source Select bits (valid only when CCH<1:0> = 11)
00 = Band gap voltage is provided as an input to the comparators
01 = Band gap voltage divided-by-two is provided as an input to the comparators
10 = Band gap voltage divided-by-six is provided as an input to the comparators
11 = VREF+ pin is provided as an input the comparators

bit 7 **CVREN:** Comparator Voltage Reference Enable bit
1 = CVREF circuit is powered on
0 = CVREF circuit is powered down

bit 6 **CVROE:** Comparator VREF Output Enable bit
1 = CVREF voltage level is output on the CVREF pin
0 = CVREF voltage level is disconnected from the CVREF pin

bit 5 **CVRR:** Comparator VREF Range Selection bit
1 = CVRSRC range should be 0 to 0.625 CVRSRC with CVRSRC/24 step size
0 = CVRSRC range should be 0.25 to 0.719 CVRSRC with CVRSRC/32 step size

bit 4 **CVRSS:** Comparator VREF Source Selection bit
1 = Comparator reference source CVRSRC = VREF+ – VREF-
0 = Comparator reference source CVRSRC = AVDD – AVSS

bit 3-0 **CVR<3:0>:** Comparator VREF Value Selection 0 ≤ CVR<3:0> ≤ 15 bits
When CVRR = 1:
CVREF = (CVR<3:0>/ 24) • (CVRSRC)
When CVRR = 0:
CVREF = 1/4 • (CVRSRC) + (CVR<3:0>/32) • (CVRSRC)

26.0 CHARGE TIME MEASUREMENT UNIT (CTMU)

Note: This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the associated "*PIC24F Family Reference Manual*", **Section 11. "Charge Time Measurement Unit (CTMU)"** (DS39724). The information in this data sheet supersedes the information in the FRM.

The Charge Time Measurement Unit (CTMU) is a flexible analog module that provides accurate differential time measurement between pulse sources, as well as asynchronous pulse generation. Its key features include:

- Four edge input trigger sources
- Polarity control for each edge source
- Control of edge sequence
- Control of response to edges
- Time measurement resolution of 1 nanosecond
- Accurate current source suitable for capacitive measurement

Together with other on-chip analog modules, the CTMU can be used to precisely measure time, measure capacitance, measure relative changes in capacitance or generate output pulses that are independent of the system clock. The CTMU module is ideal for interfacing with capacitive-based sensors.

The CTMU is controlled through two registers: CTMUCON and CTMUICON. CTMUCON enables the module, and controls edge source selection, edge source polarity selection, and edge sequencing. The CTMUICON register controls the selection and trim of the current source.

26.1 Measuring Capacitance

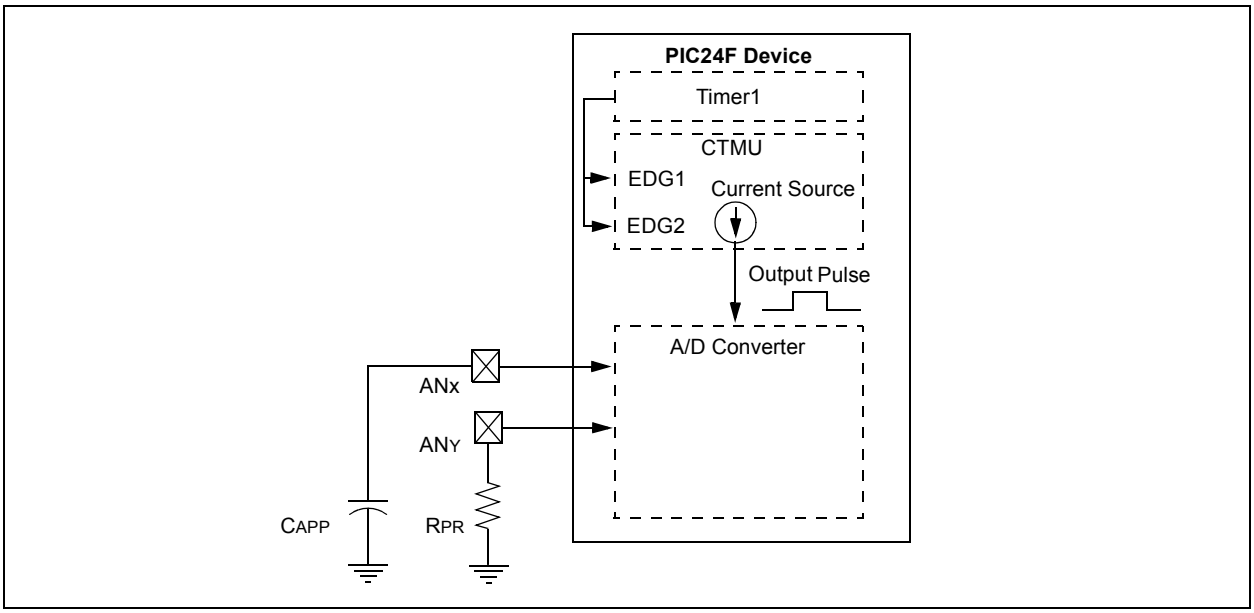
The CTMU module measures capacitance by generating an output pulse with a width equal to the time between edge events on two separate input channels. The pulse edge events to both input channels can be selected from four sources: two internal peripheral modules (OC1 and Timer1) and two external pins (CTEDG1 and CTEDG2). This pulse is used with the module's precision current source to calculate capacitance according to the relationship:

$$C = I \cdot \frac{dV}{dT}$$

For capacitance measurements, the A/D Converter samples an external capacitor (CAPP) on one of its input channels after the CTMU output's pulse. A precision resistor (RPR) provides current source calibration on a second A/D channel. After the pulse ends, the converter determines the voltage on the capacitor. The actual calculation of capacitance is performed in software by the application.

Figure 26-1 shows the external connections used for capacitance measurements, and how the CTMU and A/D modules are related in this application. This example also shows the edge events coming from Timer1, but other configurations using external edge sources are possible. A detailed discussion on measuring capacitance and time with the CTMU module is provided in the "*PIC24F Family Reference Manual*".

FIGURE 26-1: TYPICAL CONNECTIONS AND INTERNAL CONFIGURATION FOR CAPACITANCE MEASUREMENT



26.2 Measuring Time

Time measurements on the pulse width can be similarly performed using the A/D module's internal capacitor (CAD) and a precision resistor for current calibration. Figure 26-2 shows the external connections used for time measurements, and how the CTMU and A/D modules are related in this application. This example also shows both edge events coming from the external CTEDG pins, but other configurations using internal edge sources are possible. A detailed discussion on measuring capacitance and time with the CTMU module is provided in the "*PIC24F Family Reference Manual*".

26.3 Pulse Generation and Delay

The CTMU module can also generate an output pulse with edges that are not synchronous with the device's system clock. More specifically, it can generate a pulse with a programmable delay from an edge event input to the module.

When the module is configured for pulse generation delay by setting the TGEN (CTMUCON<12>) bit, the internal current source is connected to the B input of Comparator 2. A capacitor (CDELAY) is connected to the Comparator 2 pin, C2INB, and the comparator voltage reference, CVREF, is connected to C2INA. CVREF is then configured for a specific trip point. The module begins to charge CDELAY when an edge event is detected. When CDELAY charges above the CVREF trip point, a pulse is output on CTPLS. The length of the pulse delay is determined by the value of CDELAY and the CVREF trip point.

Figure 26-3 shows the external connections for pulse generation, as well as the relationship of the different analog modules required. While CTEDG1 is shown as the input pulse source, other options are available. A detailed discussion on pulse generation with the CTMU module is provided in the "*PIC24F Family Reference Manual*".

FIGURE 26-2: TYPICAL CONNECTIONS AND INTERNAL CONFIGURATION FOR TIME MEASUREMENT TIME

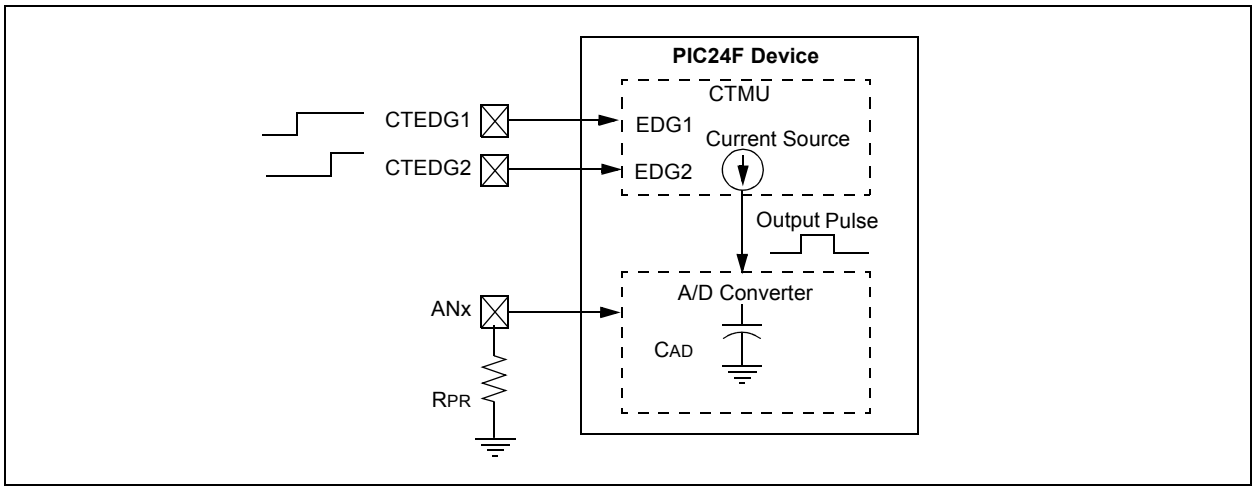
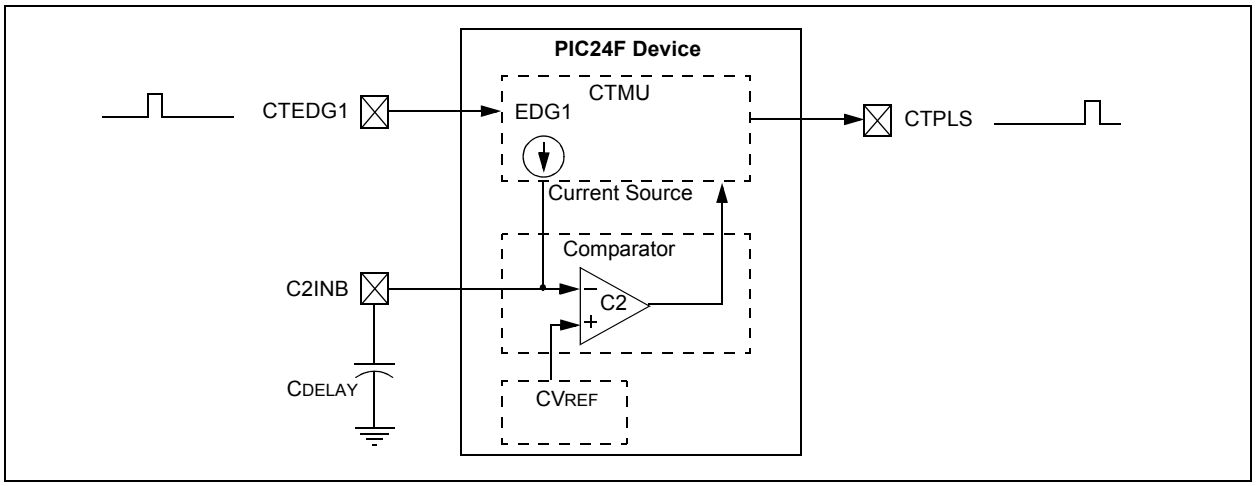


FIGURE 26-3: TYPICAL CONNECTIONS AND INTERNAL CONFIGURATION FOR PULSE DELAY GENERATION



REGISTER 26-1: CTMUCON: CTMU CONTROL REGISTER

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CTMUEN	—	CTMUSIDL	TGEN(1)	EDGEN	EDGSEQEN	IDISSEN	CTTRIG
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0, HSC	R/W-0, HSC
EDG2POL	EDG2SEL1	EDG2SEL0	EDG1POL	EDG1SEL1	EDG1SEL0	EDG2STAT	EDG1STAT
bit 7							bit 0

Legend:	HSC = Hardware Settable/Clearable bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **CTMUEN:** CTMU Enable bit
1 = Module is enabled
0 = Module is disabled

bit 14 **Unimplemented:** Read as '0'

bit 13 **CTMUSIDL:** Stop in Idle Mode bit
1 = Discontinue module operation when the device enters Idle mode
0 = Continue module operation in Idle mode

bit 12 **TGEN:** Time Generation Enable bit(1)
1 = Enables edge delay generation
0 = Disables edge delay generation

bit 10 **EDGEN:** Edge Enable bit
1 = Edges are not blocked
0 = Edges are blocked

bit 10 **EDGSEQEN:** Edge Sequence Enable bit
1 = Edge 1 event must occur before Edge 2 event can occur
0 = No edge sequence is needed

bit 9 **IDISSEN:** Analog Current Source Control bit
1 = Analog current source output is grounded
0 = Analog current source output is not grounded

bit 8 **CTTRIG:** Trigger Control bit
1 = Trigger output is enabled
0 = Trigger output is disabled

bit 7 **EDG2POL:** Edge 2 Polarity Select bit
1 = Edge 2 is programmed for a positive edge response
0 = Edge 2 is programmed for a negative edge response

bit 6-5 **EDG2SEL<1:0>:** Edge 2 Source Select bits
11 = CTEDG1 pin
10 = CTEDG2 pin
01 = OC1 module
00 = Timer1 module

bit 4 **EDG1POL:** Edge 1 Polarity Select bit
1 = Edge 1 is programmed for a positive edge response
0 = Edge 1 is programmed for a negative edge response

Note 1: If TGEN = 1, the peripheral inputs and outputs must be configured to an available RPn/RPIn pin. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

REGISTER 26-1: CTMUCON: CTMU CONTROL REGISTER (CONTINUED)

bit 3-2 **EDG1SEL<1:0>:** Edge 1 Source Select bits
11 = CTEDG1 pin
10 = CTEDG2 pin
01 = OC1 module
00 = Timer1 module

bit 1 **EDG2STAT:** Edge 2 Status bit
1 = Edge 2 event has occurred
0 = Edge 2 event has not occurred

bit 0 **EDG1STAT:** Edge 1 Status bit
1 = Edge 1 event has occurred
0 = Edge 1 event has not occurred

Note 1: If TGEN = 1, the peripheral inputs and outputs must be configured to an available RPn/RPIn pin. See **Section 10.4 "Peripheral Pin Select (PPS)"** for more information.

REGISTER 26-2: CTMUICON: CTMU CURRENT CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ITRIM5	ITRIM4	ITRIM3	ITRIM2	ITRIM1	ITRIM0	IRNG1	IRNG0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-10 **ITRIM<5:0>:** Current Source Trim bits
011111 = Maximum positive change from nominal current
011110
.
.
.
000001 = Minimum positive change from nominal current
000000 = Nominal current output specified by IRNG<1:0>
111111 = Minimum negative change from nominal current
.
.
.
100010
100001 = Maximum negative change from nominal current

bit 9-8 **IRNG<1:0>:** Current Source Range Select bits
11 = 100 × Base Current
10 = 10 × Base Current
01 = Base current level (0.55 μA nominal)
00 = Current source is disabled

bit 7-0 **Unimplemented:** Read as '0'

27.0 SPECIAL FEATURES

Note: This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the following sections of the "*PIC24F Family Reference Manual*". The information in this data sheet supersedes the information in the FRMs.

- **Section 9. "Watchdog Timer (WDT)"** (DS39697)
- **Section 32. "High-Level Device Integration"** (DS39719)
- **Section 33. "Programming and Diagnostics"** (DS39716)

PIC24FJ256DA210 family devices include several features intended to maximize application flexibility and reliability, and minimize cost through elimination of external components. These are:

- Flexible Configuration
- Watchdog Timer (WDT)
- Code Protection
- JTAG Boundary Scan Interface
- In-Circuit Serial Programming™
- In-Circuit Emulation

27.1 Configuration Bits

The Configuration bits can be programmed (read as '0'), or left unprogrammed (read as '1'), to select various device configurations. These bits are mapped starting at program memory location F80000h. A detailed explanation of the various bit functions is provided in Register 27-1 through Register 27-6.

Note that address F80000h is beyond the user program memory space. In fact, it belongs to the configuration memory space (800000h-FFFFFFh) which can only be accessed using table reads and table writes.

27.1.1 CONSIDERATIONS FOR CONFIGURING PIC24FJ256DA210 FAMILY DEVICES

In PIC24FJ256DA210 family devices, the configuration bytes are implemented as volatile memory. This means that configuration data must be programmed each time the device is powered up. Configuration data is stored in the three words at the top of the on-chip program memory space, known as the Flash Configuration Words. Their specific locations are shown in Table 27-1. These are packed representations of the actual device Configuration bits, whose actual locations are distributed among several locations in configuration space. The configuration data is automatically loaded from the Flash Configuration Words to the proper Configuration registers during device Resets.

Note: Configuration data is reloaded on all types of device Resets.

When creating applications for these devices, users should always specifically allocate the location of the Flash Configuration Word for configuration data. This is to make certain that program code is not stored in this address when the code is compiled.

The upper byte of all Flash Configuration Words in program memory should always be '0000 0000'. This makes them appear to be NOP instructions in the remote event that their locations are ever executed by accident. Since Configuration bits are not implemented in the corresponding locations, writing '0's to these locations has no effect on device operation.

Note: Performing a page erase operation on the last page of program memory clears the Flash Configuration Words, enabling code protection as a result. Therefore, users should avoid performing page erase operations on the last page of program memory.

TABLE 27-1: FLASH CONFIGURATION WORD LOCATIONS FOR PIC24FJ256DA210 FAMILY DEVICES

Device	Configuration Word Addresses			
	1	2	3	4
PIC24FJ128DAXXX	157FEh	157FCh	157FAh	157F8h
PIC24FJ256DAXXX	2ABFEh	2ABFCh	2ABFAh	2ABF8h

REGISTER 27-1: CW1: FLASH CONFIGURATION WORD 1

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 23							bit 16

r-x	R/PO-1	R/PO-1	R/PO-1	R/PO-1	r-1	R/PO-1	R/PO-1
reserved	JTAGEN	GCP	GWRP	$\overline{\text{DEBUG}}$	reserved	ICS1	ICS0
bit 15							bit 8

R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
FWDTEN	WINDIS	$\overline{\text{ALTVREF}}$(1)	FWPSA	WDTPS3	WDTPS2	WDTPS1	WDTPS0
bit 7							bit 0

Legend:	r = Reserved bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 23-16 **Unimplemented:** Read as '1'

bit 15 **Reserved:** The value is unknown; program as '0'

bit 14 **JTAGEN:** JTAG Port Enable bit
1 = JTAG port is enabled
0 = JTAG port is disabled

bit 13 **GCP:** General Segment Program Memory Code Protection bit
1 = Code protection is disabled
0 = Code protection is enabled for the entire program memory space

bit 12 **GWRP:** General Segment Code Flash Write Protection bit
1 = Writes to program memory are allowed
0 = Writes to program memory are not allowed

bit 11 **$\overline{\text{DEBUG}}$:** Background Debugger Enable bit
1 = Device resets into Operational mode
0 = Device resets into Debug mode

bit 10 **Reserved:** Always maintain as '1'

bit 9-8 **ICS<1:0>:** Emulator Pin Placement Select bits
11 = Emulator functions are shared with PGEC1/PGED1
10 = Emulator functions are shared with PGEC2/PGED2
01 = Emulator functions are shared with PGEC3/PGED3
00 = Reserved; do not use

bit 7 **FWDTEN:** Watchdog Timer Enable bit
1 = Watchdog Timer is enabled
0 = Watchdog Timer is disabled

bit 6 **WINDIS:** Windowed Watchdog Timer Disable bit
1 = Standard Watchdog Timer is enabled
0 = Windowed Watchdog Timer is enabled; FWDTEN must be '1'

bit 5 **$\overline{\text{ALTVREF}}$:** Alternate VREF Pin Selection bit(1)
1 = VREF is on a default pin (VREF+ on RA10 and VREF- on RA9)
0 = VREF is on an alternate pin (VREF+ on RB0 and VREF- on RB1)

Note 1: Unimplemented in 64-pin devices, maintain at '1' (VREF+ on RB0 and VREF- on RB1).

REGISTER 27-1: CW1: FLASH CONFIGURATION WORD 1 (CONTINUED)

bit 4 **FWPSA:** WDT Prescaler Ratio Select bit
1 = Prescaler ratio of 1:128
0 = Prescaler ratio of 1:32

bit 3-0 **WDTPS<3:0>:** Watchdog Timer Postscaler Select bits
1111 = 1:32,768
1110 = 1:16,384
1101 = 1:8,192
1100 = 1:4,096
1011 = 1:2,048
1010 = 1:1,024
1001 = 1:512
1000 = 1:256
0111 = 1:128
0110 = 1:64
0101 = 1:32
0100 = 1:16
0011 = 1:8
0010 = 1:4
0001 = 1:2
0000 = 1:1

Note 1: Unimplemented in 64-pin devices, maintain at '1' (VREF+ on RB0 and VREF- on RB1).

REGISTER 27-2: CW2: FLASH CONFIGURATION WORD 2

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 23							bit 16

R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
IESO	PLLDIV2	PLLDIV1	PLLDIV0	PLL96MHZ	FNOSC2	FNOSC1	FNOSC0
bit 15							bit 8

R/PO-1	R/PO-1	R/PO-1	R/PO-1	r-1	r-1	R/PO-1	R/PO-1
FCKSM1	FCKSM0	OSCIOFCN	IOL1WAY	reserved	reserved	POSCMD1	POSCMD0
bit 7							bit 0

Legend:	r = Reserved bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 23-16 **Unimplemented:** Read as '1'

bit 15 **IESO:** Internal External Switchover bit
1 = IESO mode (Two-Speed Start-up) is enabled
0 = IESO mode (Two-Speed Start-up) is disabled

bit 14-12 **PLLDIV<2:0>:** 96 MHz PLL Prescaler Select bits
111 = Oscillator input is divided by 12 (48 MHz input)
110 = Oscillator input is divided by 8 (32 MHz input)
101 = Oscillator input is divided by 6 (24 MHz input)
100 = Oscillator input is divided by 5 (20 MHz input)
011 = Oscillator input is divided by 4 (16 MHz input)
010 = Oscillator input is divided by 3 (12 MHz input)
001 = Oscillator input is divided by 2 (8 MHz input)
000 = Oscillator input is used directly (4 MHz input)

bit 11 **PLL96MHZ:** 96 MHz PLL Start-Up Enable bit
1 = 96 MHz PLL is enabled automatically on start-up
0 = 96 MHz PLL is software controlled (can be enabled by setting the PLLEN bit in CLKDIV<5>)

bit 10-8 **FNOSC<2:0>:** Initial Oscillator Select bits
111 = Fast RC Oscillator with Postscaler (FRCDIV)
110 = Reserved
101 = Low-Power RC Oscillator (LPRC)
100 = Secondary Oscillator (SOSC)
011 = Primary Oscillator with PLL module (XTPLL, HSPLL, ECPLL)
010 = Primary Oscillator (XT, HS, EC)
001 = Fast RC Oscillator with Postscaler and PLL module (FRCPLL)
000 = Fast RC Oscillator (FRC)

bit 7-6 **FCKSM<1:0>:** Clock Switching and Fail-Safe Clock Monitor Configuration bits
1x = Clock switching and Fail-Safe Clock Monitor are disabled
01 = Clock switching is enabled, Fail-Safe Clock Monitor is disabled
00 = Clock switching is enabled, Fail-Safe Clock Monitor is enabled

bit 5 **OSCIOFCN:** OSCO Pin Configuration bit
If POSCMD<1:0> = 11 or 00:
1 = OSCO/CLKO/RC15 functions as CLKO (FOSC/2)
0 = OSCO/CLKO/RC15 functions as port I/O (RC15)
If POSCMD<1:0> = 10 or 01:
OSCIOFCN has no effect on OSCO/CLKO/RC15.

REGISTER 27-2: CW2: FLASH CONFIGURATION WORD 2 (CONTINUED)

bit 4 **IOL1WAY:** IOLOCK One-Way Set Enable bit
- 1 = The IOLOCK bit (OSCCON<6>) can be set once, provided the unlock sequence has been completed. Once set, the Peripheral Pin Select registers cannot be written to a second time.
- 0 = The IOLOCK bit can be set and cleared as needed, provided the unlock sequence has been completed

bit 3-2 **Reserved:** Always maintain as '1'

bit 1-0 **POSCMD<1:0>:** Primary Oscillator Configuration bits
- 11 = Primary oscillator is disabled
- 10 = HS Oscillator mode is selected
- 01 = XT Oscillator mode is selected
- 00 = EC Oscillator mode is selected

REGISTER 27-3: CW3: FLASH CONFIGURATION WORD 3

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 23							bit 16

R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
WPEND	WPCFG	WPDIS	$\overline{\text{ALTPMP}}$(1)	WUTSEL1	WUTSEL0	SOSCSEL1	SOSCSEL0
bit 15							bit 8

R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
WPFP7	WPFP6	WPFP5	WPFP4	WPFP3	WPFP2	WPFP1	WPFP0
bit 7							bit 0

Legend:	PO = Program-Once bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 23-16 **Unimplemented:** Read as '1'

bit 15 **WPEND:** Segment Write Protection End Page Select bit
- 1 = Protected code segment upper boundary is at the last page of program memory; the lower boundary is the code page specified by WPFP<7:0>
- 0 = Protected code segment lower boundary is at the bottom of the program memory (000000h); upper boundary is the code page specified by WPFP<7:0>

bit 14 **WPCFG:** Configuration Word Code Page Write Protection Select bit
- 1 = Last page (at the top of program memory) and Flash Configuration Words are not write-protected(3)
- 0 = Last page and Flash Configuration Words are write-protected, provided WPDIS = '0'

bit 13 **WPDIS:** Segment Write Protection Disable bit
- 1 = Segmented code protection is disabled
- 0 = Segmented code protection is enabled; protected segment is defined by the WPEND, WPCFG and WPFPx Configuration bits

bit 12 **$\overline{\text{ALTPMP}}$:** Alternate EPMP Pin Mapping bit(1)
- 1 = EPMP pins are in default location mode
- 0 = EPMP pins are in alternate location mode

Note 1: Unimplemented in 64-pin devices, maintain at '1'.

2: Ensure that the SCLKI pin is made a digital input while using this configuration, see Table 10-1.

3: Regardless of WPCFG status, if WPEND = 1 or if WPFP corresponds to the Configuration Word's page, the Configuration Word's page is protected.

REGISTER 27-3: CW3: FLASH CONFIGURATION WORD 3 (CONTINUED)

bit 11-10 **WUTSEL<1:0>:** Voltage Regulator Standby Mode Wake-up Time Select bits
11 = Default regulator start-up time is used
01 = Fast regulator start-up time is used
x0 = Reserved; do not use

bit 9-8 **SOSCSEL<1:0>:** SOSC Selection Configuration bits
11 = Secondary oscillator is in Default (high drive strength) Oscillator mode
10 = Reserved; do not use
01 = Secondary oscillator is in Low-Power (low drive strength) Oscillator mode
00 = External clock (SCLKI) or Digital I/O mode(2)

bit 7-0 **WPFP<7:0>:** Write Protected Code Segment Boundary Page bits
Designates the 512 instruction words page boundary of the protected code segment.
If WPEND = 1:
Specifies the lower page boundary of the code-protected segment; the last page being the last implemented page in the device.
If WPEND = 0:
Specifies the upper page boundary of the code-protected segment; Page 0 being the lower boundary.

Note 1: Unimplemented in 64-pin devices, maintain at '1'.
2: Ensure that the SCLKI pin is made a digital input while using this configuration, see Table 10-1.
3: Regardless of WPCFG status, if WPEND = 1 or if WPFP corresponds to the Configuration Word's page, the Configuration Word's page is protected.

REGISTER 27-4: CW4: FLASH CONFIGURATION WORD 4

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 23							bit 16

r-1	r-1	r-1	r-1	r-1	r-1	r-1	r-1
reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
bit 15							bit 8

r-1	r-1	r-1	r-1	r-1	r-1	r-1	r-1
reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
bit 7							bit 0

Legend:	r = Reserved bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 23-16 **Unimplemented:** Read as '0'
bit 15-0 **Reserved:** Always maintain as '1'

REGISTER 27-5: DEVID: DEVICE ID REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 23							bit 16

R	R	R	R	R	R	R	R
FAMID7	FAMID6	FAMID5	FAMID4	FAMID3	FAMID2	FAMID1	FAMID0
bit 15							bit 8

R	R	R	R	R	R	R	R
DEV7	DEV6	DEV5	DEV4	DEV3	DEV2	DEV1	DEV0
bit 7							bit 0

Legend:	R = Readable bit	U = Unimplemented bit

bit 23-16 **Unimplemented:** Read as '1'

bit 15-8 **FAMID<7:0>:** Device Family Identifier bits
01000001 = PIC24FJ256DA210 family

bit 7-0 **DEV<7:0>:** Individual Device Identifier bits
00001000 = PIC24FJ128DA206
00001001 = PIC24FJ128DA106
00001010 = PIC24FJ128DA210
00001011 = PIC24FJ128DA110
00001100 = PIC24FJ256DA206
00001101 = PIC24FJ256DA106
00001110 = PIC24FJ256DA210
00001111 = PIC24FJ256DA110

REGISTER 27-6: DEVREV: DEVICE REVISION REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 23							bit 16

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	R	R	R	R
—	—	—	—	REV3	REV2	REV1	REV0
bit 7							bit 0

Legend:	R = Readable bit	U = Unimplemented bit

bit 23-4 **Unimplemented:** Read as '0'

bit 3-0 **REV<3:0>:** Device revision identifier bits

27.2 On-Chip Voltage Regulator

All PIC24FJ256DA210 family devices power their core digital logic at a nominal 1.8V. This may create an issue for designs that are required to operate at a higher typical voltage, such as 3.3V. To simplify system design, all devices in the PIC24FJ256DA210 family incorporate an on-chip regulator that allows the device to run its core logic from VDD.

The regulator is controlled by the ENVREG pin. Tying VDD to the pin enables the regulator, which in turn, provides power to the core from the other VDD pins. When the regulator is enabled, a low-ESR capacitor (such as ceramic) must be connected to the VCAP pin (Figure 27-1). This helps to maintain the stability of the regulator. The recommended value for the filter capacitor (CEFC) is provided in **Section 30.1 "DC Characteristics"**.

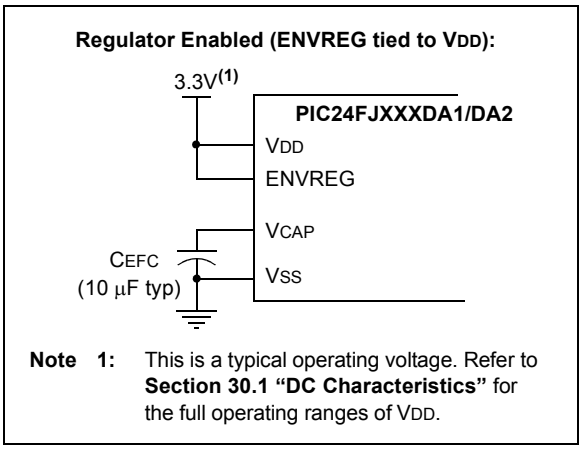
27.2.1 VOLTAGE REGULATOR LOW-VOLTAGE DETECTION

When the on-chip regulator is enabled, it provides a constant voltage of 1.8V nominal to the digital core logic.

The regulator can provide this level from a VDD of about 2.1V, all the way up to the device's VDDMAX. It does not have the capability to boost VDD levels. In order to prevent "brown-out" conditions when the voltage drops too low for the regulator, the Brown-out Reset occurs. Then the regulator output follows VDD with a typical voltage drop of 300 mV.

To provide information about when the regulator voltage starts reducing, the on-chip regulator includes a simple Low-Voltage Detect circuit, which sets the Low-Voltage Detect Interrupt Flag, LVDIF (IFS4<8>). This can be used to generate an interrupt to trigger an orderly shutdown.

FIGURE 27-1: CONNECTIONS FOR THE ON-CHIP REGULATOR



Note 1: This is a typical operating voltage. Refer to **Section 30.1 "DC Characteristics"** for the full operating ranges of VDD.

27.2.2 ON-CHIP REGULATOR AND POR

When the voltage regulator is enabled, it takes approximately 10 µs for it to generate output. During this time, designated as TVREG, code execution is disabled. TVREG is applied every time the device resumes operation after any power-down, including Sleep mode. TVREG is determined by the status of the VREGS bit (RCON<8>) and the WUTSEL Configuration bits (CW3<11:10>). Refer to **Section 30.0 "Electrical Characteristics"** for more information on TVREG.

27.2.3 ON-CHIP REGULATOR AND BOR

When the on-chip regulator is enabled, PIC24FJ256DA210 family devices also have a simple brown-out capability. If the voltage supplied to the regulator is inadequate to maintain the output level, the regulator Reset circuitry will generate a Brown-out Reset. This event is captured by the BOR (RCON<1>) flag bit. The brown-out voltage specifications are provided in **Section 7. "Reset"** (DS39712) in the "*PIC24F Family Reference Manual*".

Note:	For more information, see **Section 30.0 "Electrical Characteristics"**. The information in this data sheet supersedes the information in the FRM.

27.2.4 VOLTAGE REGULATOR STANDBY MODE

When enabled, the on-chip regulator always consumes a small incremental amount of current over IDD/IPD, including when the device is in Sleep mode, even though the core digital logic does not require power. To provide additional savings in applications where power resources are critical, the regulator can be made to enter Standby mode on its own whenever the device goes into Sleep mode. This feature is controlled by the VREGS bit (RCON<8>). Clearing the VREGS bit enables the Standby mode. When waking up from Standby mode, the regulator needs to wait for TVREG to expire before wake-up.

The regulator wake-up time required for Standby mode is controlled by the WUTSEL<1:0> (CW3<11:10>) Configuration bits. The regulator wake-up time is lower when WUTSEL<1:0> = 01, and higher when WUTSEL<1:0> = 11. Refer to the TVREG specification in Table 30-10 for regulator wake-up time.

When the regulator's Standby mode is turned off (VREGS = 1), the device wakes up without waiting for TVREG. However, with the VREGS bit set, the power consumption while in Sleep mode will be approximately 40 μA higher than what it would be if the regulator was allowed to enter Standby mode.

27.3 Watchdog Timer (WDT)

For PIC24FJ256DA210 family devices, the WDT is driven by the LPRC oscillator. When the WDT is enabled, the clock source is also enabled.

The nominal WDT clock source from LPRC is 31 kHz. This feeds a prescaler that can be configured for either 5-bit (divide-by-32) or 7-bit (divide-by-128) operation. The prescaler is set by the FWPSA Configuration bit. With a 31 kHz input, the prescaler yields a nominal WDT Time-out period (TWDT) of 1 ms in 5-bit mode or 4 ms in 7-bit mode.

A variable postscaler divides down the WDT prescaler output and allows for a wide range of time-out periods. The postscaler is controlled by the WDTPS<3:0> Configuration bits (CW1<3:0>), which allows the selection of a total of 16 settings, from 1:1 to 1:32,768. Using the prescaler and postscaler time-out periods, ranging from 1 ms to 131 seconds, can be achieved.

The WDT, prescaler and postscaler are reset:

- On any device Reset
- On the completion of a clock switch, whether invoked by software (i.e., setting the OSWEN bit after changing the NOSC bits) or by hardware (i.e., Fail-Safe Clock Monitor)
- When a PWRSAV instruction is executed (i.e., Sleep or Idle mode is entered)
- When the device exits Sleep or Idle mode to resume normal operation
- By a CLRWDT instruction during normal execution

If the WDT is enabled, it will continue to run during Sleep or Idle modes. When the WDT time-out occurs, the device will wake the device and code execution will continue from where the PWRSAV instruction was executed. The corresponding SLEEP or IDLE (RCON<3:2>) bits will need to be cleared in software after the device wakes up.

The WDT Flag bit, WDTO (RCON<4>), is not automatically cleared following a WDT time-out. To detect subsequent WDT events, the flag must be cleared in software.

Note:	The CLRWDT and PWRSAV instructions clear the prescaler and postscaler counts when executed.

27.3.1 WINDOWED OPERATION

The Watchdog Timer has an optional Fixed-Window mode of operation. In this Windowed mode, CLRWDT instructions can only reset the WDT during the last 1/4 of the programmed WDT period. A CLRWDT instruction executed before that window causes a WDT Reset, similar to a WDT time-out.

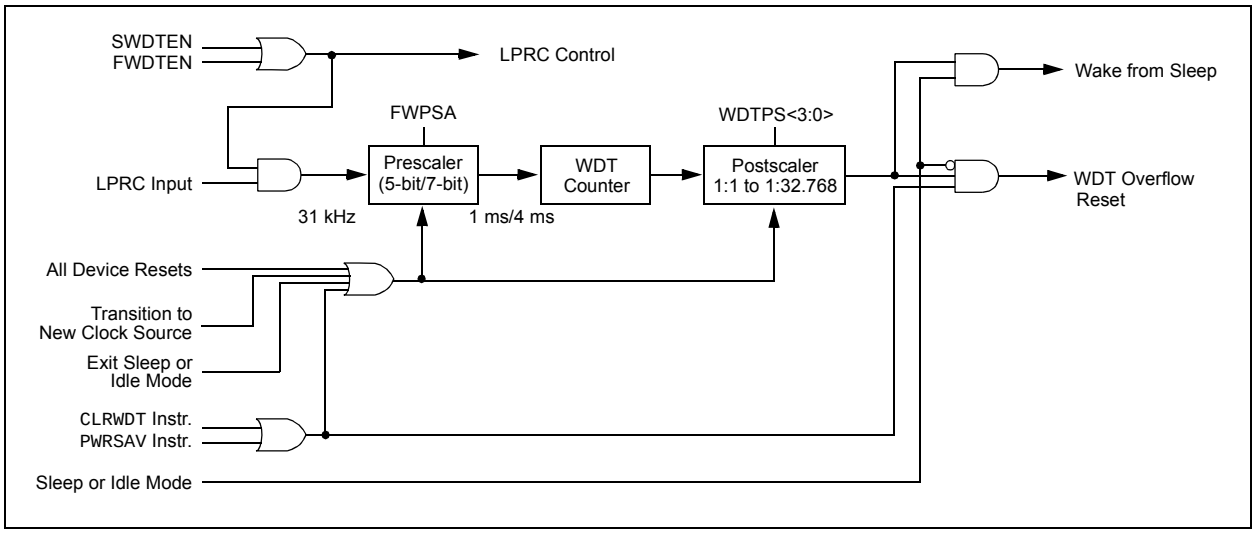
Windowed WDT mode is enabled by programming the WINDIS Configuration bit (CW1<6>) to '0'.

27.3.2 CONTROL REGISTER

The WDT is enabled or disabled by the FWDTEN Configuration bit. When the FWDTEN Configuration bit is set, the WDT is always enabled.

The WDT can be optionally controlled in software when the FWDTEN Configuration bit has been programmed to '0'. The WDT is enabled in software by setting the SWDTEN Control bit (RCON<5>). The SWDTEN control bit is cleared on any device Reset. The software WDT option allows the user to enable the WDT for critical code segments and disable the WDT during non-critical segments for maximum power savings.

FIGURE 27-2: WDT BLOCK DIAGRAM



27.4 Program Verification and Code Protection

PIC24FJ256DA210 family devices provide two complimentary methods to protect application code from overwrites and erasures. These also help to protect the device from inadvertent configuration changes during run time.

27.4.1 GENERAL SEGMENT PROTECTION

For all devices in the PIC24FJ256DA210 family, the on-chip program memory space is treated as a single block, known as the General Segment (GS). Code protection for this block is controlled by one Configuration bit, GCP. This bit inhibits external reads and writes to the program memory space. It has no direct effect in normal execution mode.

Write protection is controlled by the GWRP bit in the Configuration Word. When GWRP is programmed to '0', internal write and erase operations to program memory are blocked.

27.4.2 CODE SEGMENT PROTECTION

In addition to global General Segment protection, a separate subrange of the program memory space can be individually protected against writes and erases. This area can be used for many purposes where a separate block of write and erase-protected code is needed, such as bootloader applications. Unlike common boot block implementations, the specially protected segment in the PIC24FJ256DA210 family devices can be located by the user anywhere in the program space and configured in a wide range of sizes.

Code segment protection provides an added level of protection to a designated area of program memory by disabling the NVM safety interlock whenever a write or erase address falls within a specified range. It does not override General Segment protection controlled by the GCP or GWRP bits. For example, if GCP and GWRP are enabled, enabling segmented code protection for the bottom half of program memory does not undo General Segment protection for the top half.

The size and type of protection for the segmented code range are configured by the WPFPx, WPEND, WPCFG and WPDIS bits in Configuration Word 3. Code segment protection is enabled by programming the WPDIS bit (= 0). The WPFP bits specify the size of the segment to be protected, by specifying the 512-word code page that is the start or end of the protected segment. The specified region is inclusive, therefore, this page will also be protected.

The WPEND bit determines if the protected segment uses the top or bottom of the program space as a boundary. Programming WPEND (= 0) sets the bottom of program memory (000000h) as the lower boundary of the protected segment. Leaving WPEND unprogrammed (= 1) protects the specified page through the last page of implemented program memory, including the Configuration Word locations.

A separate bit, WPCFG, is used to protect the last page of program space, including the Flash Configuration Words. Programming WPCFG (= 0) protects the last page in addition to the pages selected by the WPEND and WPFP<7:0> bits setting. This is useful in circumstances where write protection is needed for both the code segment in the bottom of the memory and the Flash Configuration Words.

The various options for segment code protection are shown in Table 27-2.

27.4.3 CONFIGURATION REGISTER PROTECTION

The Configuration registers are protected against inadvertent or unwanted changes or reads in two ways. The primary protection method is the same as that of the RP registers – shadow registers contain a complimentary value which is constantly compared with the actual value.

To safeguard against unpredictable events, Configuration bit changes resulting from individual cell level disruptions (such as ESD events) will cause a parity error and trigger a device Reset.

The data for the Configuration registers is derived from the Flash Configuration Words in program memory. When the GCP bit is set, the source data for device configuration is also protected as a consequence. Even if General Segment protection is not enabled, the device configuration can be protected by using the appropriate code segment protection setting.

TABLE 27-2: CODE SEGMENT PROTECTION CONFIGURATION OPTIONS

Segment Configuration Bits			Write/Erase Protection of Code Segment
WPDIS	WPEND	WPCFG	
1	X	x	No additional protection is enabled; all program memory protection is configured by GCP and GWRP.
0	1	x	Addresses from the first address of the code page are defined by WPFP<7:0> through the end of implemented program memory (inclusive), write/erase protected, including Flash Configuration Words.
0	0	1	Address 000000h through the last address of the code page is defined by WPFP<7:0> (inclusive), write/erase protected.
0	0	0	Address 000000h through the last address of code page is defined by WPFP<7:0> (inclusive), write/erase protected and the last page, including Flash Configuration Words are write/erase protected.

27.5 JTAG Interface

PIC24FJ256DA210 family devices implement a JTAG interface, which supports boundary scan device testing.

27.6 In-Circuit Serial Programming™

PIC24FJ256DA210 family microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock (PGECx) and data (PGEDx), and three other lines for power (V_{DD}), ground (V_{SS}) and $\overline{MCLR}$. This allows customers to manufacture boards with unprogrammed devices and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

27.7 In-Circuit Debugger

When MPLAB® ICD 3 is selected as a debugger, the in-circuit debugging functionality is enabled. This function allows simple debugging functions when used with MPLAB IDE. Debugging functionality is controlled through the PGECx (Emulation/Debug Clock) and PGEDx (Emulation/Debug Data) pins.

To use the in-circuit debugger function of the device, the design must implement ICSP connections to $\overline{MCLR}$, V_{DD}, V_{SS} and the PGECx/PGEDx pin pair designated by the ICS Configuration bits. In addition, when the feature is enabled, some of the resources are not available for general use. These resources include the first 80 bytes of data RAM and two I/O pins.

28.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers and dsPIC® digital signal controllers are supported with a full range of software and hardware development tools:

- Integrated Development Environment
  - MPLAB® IDE Software
- Compilers/Assemblers/Linkers
  - MPLAB C Compiler for Various Device Families
  - HI-TECH C for Various Device Families
  - MPASM™ Assembler
  - MPLINK™ Object Linker/ MPLIB™ Object Librarian
  - MPLAB Assembler/Linker/Librarian for Various Device Families
- Simulators
  - MPLAB SIM Software Simulator
- Emulators
  - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debuggers
  - MPLAB ICD 3
  - PICkit™ 3 Debug Express
- Device Programmers
  - PICkit™ 2 Programmer
  - MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards, Evaluation Kits, and Starter Kits

28.1 MPLAB Integrated Development Environment Software

The MPLAB IDE software brings an ease of software development previously unseen in the 8/16/32-bit microcontroller market. The MPLAB IDE is a Windows® operating system-based application that contains:

- A single graphical interface to all debugging tools
  - Simulator
  - Programmer (sold separately)
  - In-Circuit Emulator (sold separately)
  - In-Circuit Debugger (sold separately)
- A full-featured editor with color-coded context
- A multiple project manager
- Customizable data windows with direct edit of contents
- High-level source code debugging
- Mouse over variable inspection
- Drag and drop variables from source to watch windows
- Extensive on-line help
- Integration of select third party tools, such as IAR C Compilers

The MPLAB IDE allows you to:

- Edit your source files (either C or assembly)
- One-touch compile or assemble, and download to emulator and simulator tools (automatically updates all project information)
- Debug using:
  - Source files (C or assembly)
  - Mixed C and assembly
  - Machine code

MPLAB IDE supports multiple debugging tools in a single development paradigm, from the cost-effective simulators, through low-cost in-circuit debuggers, to full-featured emulators. This eliminates the learning curve when upgrading to tools with increased flexibility and power.

28.2 MPLAB C Compilers for Various Device Families

The MPLAB C Compiler code development systems are complete ANSI C compilers for Microchip's PIC18, PIC24 and PIC32 families of microcontrollers and the dsPIC30 and dsPIC33 families of digital signal controllers. These compilers provide powerful integration capabilities, superior code optimization and ease of use.

For easy source level debugging, the compilers provide symbol information that is optimized to the MPLAB IDE debugger.

28.3 HI-TECH C for Various Device Families

The HI-TECH C Compiler code development systems are complete ANSI C compilers for Microchip's PIC family of microcontrollers and the dsPIC family of digital signal controllers. These compilers provide powerful integration capabilities, omniscient code generation and ease of use.

For easy source level debugging, the compilers provide symbol information that is optimized to the MPLAB IDE debugger.

The compilers include a macro assembler, linker, preprocessor, and one-step driver, and can run on multiple platforms.

28.4 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multi-purpose source files
- Directives that allow complete control over the assembly process

28.5 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler and the MPLAB C18 C Compiler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

28.6 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC devices. MPLAB C Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command line interface
- Rich directive set
- Flexible macro language
- MPLAB IDE compatibility

28.7 MPLAB SIM Software Simulator

The MPLAB SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC® DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB SIM Software Simulator fully supports symbolic debugging using the MPLAB C Compilers, and the MPASM and MPLAB Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

28.8 MPLAB REAL ICE In-Circuit Emulator System

MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs PIC® Flash MCUs and dsPIC® Flash DSCs with the easy-to-use, powerful graphical user interface of the MPLAB Integrated Development Environment (IDE), included with each kit.

The emulator is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with in-circuit debugger systems (RJ11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

The emulator is field upgradable through future firmware downloads in MPLAB IDE. In upcoming releases of MPLAB IDE, new devices will be supported, and new features will be added. MPLAB REAL ICE offers significant advantages over competitive emulators including low-cost, full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, a ruggedized probe interface and long (up to three meters) interconnection cables.

28.9 MPLAB ICD 3 In-Circuit Debugger System

MPLAB ICD 3 In-Circuit Debugger System is Microchip's most cost effective high-speed hardware debugger/programmer for Microchip Flash Digital Signal Controller (DSC) and microcontroller (MCU) devices. It debugs and programs PIC® Flash microcontrollers and dsPIC® DSCs with the powerful, yet easy-to-use graphical user interface of MPLAB Integrated Development Environment (IDE).

The MPLAB ICD 3 In-Circuit Debugger probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with a connector compatible with the MPLAB ICD 2 or MPLAB REAL ICE systems (RJ-11). MPLAB ICD 3 supports all MPLAB ICD 2 headers.

28.10 PICkit 3 In-Circuit Debugger/Programmer and PICkit 3 Debug Express

The MPLAB PICkit 3 allows debugging and programming of PIC® and dsPIC® Flash microcontrollers at a most affordable price point using the powerful graphical user interface of the MPLAB Integrated Development Environment (IDE). The MPLAB PICkit 3 is connected to the design engineer's PC using a full speed USB interface and can be connected to the target via an Microchip debug (RJ-11) connector (compatible with MPLAB ICD 3 and MPLAB REAL ICE). The connector uses two device I/O pins and the reset line to implement in-circuit debugging and In-Circuit Serial Programming™.

The PICkit 3 Debug Express include the PICkit 3, demo board and microcontroller, hookup cables and CDROM with user's guide, lessons, tutorial, compiler and MPLAB IDE software.

28.11 PICkit 2 Development Programmer/Debugger and PICkit 2 Debug Express

The PICkit™ 2 Development Programmer/Debugger is a low-cost development tool with an easy to use interface for programming and debugging Microchip's Flash families of microcontrollers. The full featured Windows® programming interface supports baseline (PIC10F, PIC12F5xx, PIC16F5xx), midrange (PIC12F6xx, PIC16F), PIC18F, PIC24, dsPIC30, dsPIC33, and PIC32 families of 8-bit, 16-bit, and 32-bit microcontrollers, and many Microchip Serial EEPROM products. With Microchip's powerful MPLAB Integrated Development Environment (IDE) the PICkit™ 2 enables in-circuit debugging on most PIC® microcontrollers. In-Circuit-Debugging runs, halts and single steps the program while the PIC microcontroller is embedded in the application. When halted at a breakpoint, the file registers can be examined and modified.

The PICkit 2 Debug Express include the PICkit 2, demo board and microcontroller, hookup cables and CDROM with user's guide, lessons, tutorial, compiler and MPLAB IDE software.

28.12 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at V_{DDMIN} and V_{DDMAX} for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages and a modular, detachable socket assembly to support various package types. The ICSP™ cable assembly is included as a standard item. In Stand-Alone mode, the MPLAB PM3 Device Programmer can read, verify and program PIC devices without a PC connection. It can also set code protection in this mode. The MPLAB PM3 connects to the host PC via an RS-232 or USB cable. The MPLAB PM3 has high-speed communications and optimized algorithms for quick programming of large memory devices and incorporates an MMC card for file storage and data applications.

28.13 Demonstration/Development Boards, Evaluation Kits, and Starter Kits

A wide variety of demonstration, development and evaluation boards for various PIC MCUs and dsPIC DSCs allows quick application development on fully functional systems. Most boards include prototyping areas for adding custom circuitry and provide application firmware and source code for examination and modification.

The boards support a variety of features, including LEDs, temperature sensors, switches, speakers, RS-232 interfaces, LCD displays, potentiometers and additional EEPROM memory.

The demonstration and development boards can be used in teaching environments, for prototyping custom circuits and for learning about various microcontroller applications.

In addition to the PICDEM™ and dsPICDEM™ demonstration/development board series of circuits, Microchip has a line of evaluation kits and demonstration software for analog filter design, KEELOQ® security ICs, CAN, IrDA®, PowerSmart battery management, SEEVAL® evaluation system, Sigma-Delta ADC, flow rate sensing, plus many more.

Also available are starter kits that contain everything needed to experience the specified device. This usually includes a single application and debug capability, all on one board.

Check the Microchip web page (www.microchip.com) for the complete list of demonstration, development and evaluation kits.

29.0 INSTRUCTION SET SUMMARY

Note: This chapter is a brief summary of the PIC24F instruction set architecture and is not intended to be a comprehensive reference source.

The PIC24F instruction set adds many enhancements to the previous PIC® MCU instruction sets, while maintaining an easy migration from previous PIC MCU instruction sets. Most instructions are a single program memory word. Only three instructions require two program memory locations.

Each single-word instruction is a 24-bit word divided into an 8-bit opcode, which specifies the instruction type and one or more operands, which further specify the operation of the instruction. The instruction set is highly orthogonal and is grouped into four basic categories:

- Word or byte-oriented operations
- Bit-oriented operations
- Literal operations
- Control operations

Table 29-1 shows the general symbols used in describing the instructions. The PIC24F instruction set summary in Table 29-2 lists all the instructions, along with the status flags affected by each instruction.

Most word or byte-oriented W register instructions (including barrel shift instructions) have three operands:

- The first source operand which is typically a register ‘Wb’ without any address modifier
- The second source operand which is typically a register ‘Ws’ with or without an address modifier
- The destination of the result which is typically a register ‘Wd’ with or without an address modifier

However, word or byte-oriented file register instructions have two operands:

- The file register specified by the value ‘f’
- The destination, which could either be the file register ‘f’ or the W0 register, which is denoted as ‘WREG’

Most bit-oriented instructions (including simple rotate/shift instructions) have two operands:

- The W register (with or without an address modifier) or file register (specified by the value of ‘Ws’ or ‘f’)
- The bit in the W register or file register (specified by a literal value or indirectly by the contents of register ‘Wb’)

The literal instructions that involve data movement may use some of the following operands:

- A literal value to be loaded into a W register or file register (specified by the value of ‘k’)
- The W register or file register where the literal value is to be loaded (specified by ‘Wb’ or ‘f’)

However, literal instructions that involve arithmetic or logical operations use some of the following operands:

- The first source operand which is a register ‘Wb’ without any address modifier
- The second source operand which is a literal value
- The destination of the result (only if not the same as the first source operand), which is typically a register ‘Wd’ with or without an address modifier

The control instructions may use some of the following operands:

- A program memory address
- The mode of the table read and table write instructions

All instructions are a single word, except for certain double-word instructions, which were made double-word instructions so that all the required information is available in these 48 bits. In the second word, the 8 MSbs are ‘0’s. If this second word is executed as an instruction (by itself), it will execute as a NOP.

Most single-word instructions are executed in a single instruction cycle, unless a conditional test is true or the program counter is changed as a result of the instruction. In these cases, the execution takes two instruction cycles, with the additional instruction cycle(s) executed as a NOP. Notable exceptions are the BRA (unconditional/computed branch), indirect CALL/GOTO, all table reads and writes, and RETURN/RETFIE instructions, which are single-word instructions but take two or three cycles.

Certain instructions that involve skipping over the subsequent instruction require either two or three cycles if the skip is performed, depending on whether the instruction being skipped is a single-word or two-word instruction. Moreover, double-word moves require two cycles. The double-word instructions execute in two instruction cycles.

TABLE 29-1: SYMBOLS USED IN OPCODE DESCRIPTIONS

Field	Description
#text	Means literal defined by "text"
(text)	Means "content of text"
[text]	Means "the location addressed by text"
{ }	Optional field or operation
<n:m>	Register bit field
.b	Byte mode selection
.d	Double-Word mode selection
.S	Shadow register select
.w	Word mode selection (default)
bit4	4-bit Bit Selection field (used in word addressed instructions) ∈ {0...15}
C, DC, N, OV, Z	MCU Status bits: Carry, Digit Carry, Negative, Overflow, Sticky Zero
Expr	Absolute address, label or expression (resolved by the linker)
f	File register address ∈ {0000h...1FFFh}
lit1	1-bit unsigned literal ∈ {0,1}
lit4	4-bit unsigned literal ∈ {0...15}
lit5	5-bit unsigned literal ∈ {0...31}
lit8	8-bit unsigned literal ∈ {0...255}
lit10	10-bit unsigned literal ∈ {0...255} for Byte mode, {0:1023} for Word mode
lit14	14-bit unsigned literal ∈ {0...16383}
lit16	16-bit unsigned literal ∈ {0...65535}
lit23	23-bit unsigned literal ∈ {0...8388607}; LSB must be '0'
None	Field does not require an entry, may be blank
PC	Program Counter
Slit10	10-bit signed literal ∈ {-512...511}
Slit16	16-bit signed literal ∈ {-32768...32767}
Slit6	6-bit signed literal ∈ {-16...16}
Wb	Base W register ∈ {W0..W15}
Wd	Destination W register ∈ { Wd, [Wd], [Wd++], [Wd--], [++Wd], [--Wd] }
Wdo	Destination W register ∈ { Wnd, [Wnd], [Wnd++], [Wnd--], [++Wnd], [--Wnd], [Wnd+Wb] }
Wm,Wn	Dividend, Divisor working register pair (direct addressing)
Wn	One of 16 working registers ∈ {W0..W15}
Wnd	One of 16 destination working registers ∈ {W0..W15}
Wns	One of 16 source working registers ∈ {W0..W15}
WREG	W0 (working register used in file register instructions)
Ws	Source W register ∈ { Ws, [Ws], [Ws++], [Ws--], [++Ws], [--Ws] }
Wso	Source W register ∈ { Wns, [Wns], [Wns++], [Wns--], [++Wns], [--Wns], [Wns+Wb] }

TABLE 29-2: INSTRUCTION SET OVERVIEW

Assembly Mnemonic	Assembly Syntax		Description	# of Words	# of Cycles	Status Flags Affected
ADD	ADD	f	f = f + WREG	1	1	C, DC, N, OV, Z
	ADD	f,WREG	WREG = f + WREG	1	1	C, DC, N, OV, Z
	ADD	#lit10,Wn	Wd = lit10 + Wd	1	1	C, DC, N, OV, Z
	ADD	Wb,Ws,Wd	Wd = Wb + Ws	1	1	C, DC, N, OV, Z
	ADD	Wb,#lit5,Wd	Wd = Wb + lit5	1	1	C, DC, N, OV, Z
ADDC	ADDC	f	f = f + WREG + (C)	1	1	C, DC, N, OV, Z
	ADDC	f,WREG	WREG = f + WREG + (C)	1	1	C, DC, N, OV, Z
	ADDC	#lit10,Wn	Wd = lit10 + Wd + (C)	1	1	C, DC, N, OV, Z
	ADDC	Wb,Ws,Wd	Wd = Wb + Ws + (C)	1	1	C, DC, N, OV, Z
	ADDC	Wb,#lit5,Wd	Wd = Wb + lit5 + (C)	1	1	C, DC, N, OV, Z
AND	AND	f	f = f .AND. WREG	1	1	N, Z
	AND	f,WREG	WREG = f .AND. WREG	1	1	N, Z
	AND	#lit10,Wn	Wd = lit10 .AND. Wd	1	1	N, Z
	AND	Wb,Ws,Wd	Wd = Wb .AND. Ws	1	1	N, Z
	AND	Wb,#lit5,Wd	Wd = Wb .AND. lit5	1	1	N, Z
ASR	ASR	f	f = Arithmetic Right Shift f	1	1	C, N, OV, Z
	ASR	f,WREG	WREG = Arithmetic Right Shift f	1	1	C, N, OV, Z
	ASR	Ws,Wd	Wd = Arithmetic Right Shift Ws	1	1	C, N, OV, Z
	ASR	Wb,Wns,Wnd	Wnd = Arithmetic Right Shift Wb by Wns	1	1	N, Z
	ASR	Wb,#lit5,Wnd	Wnd = Arithmetic Right Shift Wb by lit5	1	1	N, Z
BCLR	BCLR	f,#bit4	Bit Clear f	1	1	None
	BCLR	Ws,#bit4	Bit Clear Ws	1	1	None
BRA	BRA	C,Expr	Branch if Carry	1	1 (2)	None
	BRA	GE,Expr	Branch if Greater than or Equal	1	1 (2)	None
	BRA	GEU,Expr	Branch if Unsigned Greater than or Equal	1	1 (2)	None
	BRA	GT,Expr	Branch if Greater than	1	1 (2)	None
	BRA	GTU,Expr	Branch if Unsigned Greater than	1	1 (2)	None
	BRA	LE,Expr	Branch if Less than or Equal	1	1 (2)	None
	BRA	LEU,Expr	Branch if Unsigned Less than or Equal	1	1 (2)	None
	BRA	LT,Expr	Branch if Less than	1	1 (2)	None
	BRA	LTU,Expr	Branch if Unsigned Less than	1	1 (2)	None
	BRA	N,Expr	Branch if Negative	1	1 (2)	None
	BRA	NC,Expr	Branch if Not Carry	1	1 (2)	None
	BRA	NN,Expr	Branch if Not Negative	1	1 (2)	None
	BRA	NOV,Expr	Branch if Not Overflow	1	1 (2)	None
	BRA	NZ,Expr	Branch if Not Zero	1	1 (2)	None
	BRA	OV,Expr	Branch if Overflow	1	1 (2)	None
	BRA	Expr	Branch Unconditionally	1	2	None
	BRA	Z,Expr	Branch if Zero	1	1 (2)	None
	BRA	Wn	Computed Branch	1	2	None
BSET	BSET	f,#bit4	Bit Set f	1	1	None
	BSET	Ws,#bit4	Bit Set Ws	1	1	None
BSW	BSW.C	Ws,Wb	Write C bit to Ws<Wb>	1	1	None
	BSW.Z	Ws,Wb	Write Z bit to Ws<Wb>	1	1	None
BTG	BTG	f,#bit4	Bit Toggle f	1	1	None
	BTG	Ws,#bit4	Bit Toggle Ws	1	1	None
BTSC	BTSC	f,#bit4	Bit Test f, Skip if Clear	1	1 (2 or 3)	None
	BTSC	Ws,#bit4	Bit Test Ws, Skip if Clear	1	1 (2 or 3)	None

TABLE 29-2: INSTRUCTION SET OVERVIEW (CONTINUED)

Assembly Mnemonic	Assembly Syntax		Description	# of Words	# of Cycles	Status Flags Affected
BTSS	BTSS	f,#bit4	Bit Test f, Skip if Set	1	1 (2 or 3)	None
	BTSS	Ws,#bit4	Bit Test Ws, Skip if Set	1	1 (2 or 3)	None
BTST	BTST	f,#bit4	Bit Test f	1	1	Z
	BTST.C	Ws,#bit4	Bit Test Ws to C	1	1	C
	BTST.Z	Ws,#bit4	Bit Test Ws to Z	1	1	Z
	BTST.C	Ws,Wb	Bit Test Ws<Wb> to C	1	1	C
	BTST.Z	Ws,Wb	Bit Test Ws<Wb> to Z	1	1	Z
BTSTS	BTSTS	f,#bit4	Bit Test then Set f	1	1	Z
	BTSTS.C	Ws,#bit4	Bit Test Ws to C, then Set	1	1	C
	BTSTS.Z	Ws,#bit4	Bit Test Ws to Z, then Set	1	1	Z
CALL	CALL	lit23	Call Subroutine	2	2	None
	CALL	Wn	Call Indirect Subroutine	1	2	None
CLR	CLR	f	f = 0x0000	1	1	None
	CLR	WREG	WREG = 0x0000	1	1	None
	CLR	Ws	Ws = 0x0000	1	1	None
CLRWDT	CLRWDT		Clear Watchdog Timer	1	1	WDTO, Sleep
COM	COM	f	$f = \overline{f}$	1	1	N, Z
	COM	f,WREG	$WREG = \overline{f}$	1	1	N, Z
	COM	Ws,Wd	$Wd = \overline{Ws}$	1	1	N, Z
CP	CP	f	Compare f with WREG	1	1	C, DC, N, OV, Z
	CP	Wb,#lit5	Compare Wb with lit5	1	1	C, DC, N, OV, Z
	CP	Wb,Ws	Compare Wb with Ws (Wb – Ws)	1	1	C, DC, N, OV, Z
CP0	CP0	f	Compare f with 0x0000	1	1	C, DC, N, OV, Z
	CP0	Ws	Compare Ws with 0x0000	1	1	C, DC, N, OV, Z
CPB	CPB	f	Compare f with WREG, with Borrow	1	1	C, DC, N, OV, Z
	CPB	Wb,#lit5	Compare Wb with lit5, with Borrow	1	1	C, DC, N, OV, Z
	CPB	Wb,Ws	Compare Wb with Ws, with Borrow $(Wb – Ws – \overline{C})$	1	1	C, DC, N, OV, Z
CPSEQ	CPSEQ	Wb,Wn	Compare Wb with Wn, Skip if =	1	1 (2 or 3)	None
CPSGT	CPSGT	Wb,Wn	Compare Wb with Wn, Skip if >	1	1 (2 or 3)	None
CPSLT	CPSLT	Wb,Wn	Compare Wb with Wn, Skip if <	1	1 (2 or 3)	None
CPSNE	CPSNE	Wb,Wn	Compare Wb with Wn, Skip if ≠	1	1 (2 or 3)	None
DAW	DAW.B	Wn	Wn = Decimal Adjust Wn	1	1	C
DEC	DEC	f	f = f –1	1	1	C, DC, N, OV, Z
	DEC	f,WREG	WREG = f –1	1	1	C, DC, N, OV, Z
	DEC	Ws,Wd	Wd = Ws – 1	1	1	C, DC, N, OV, Z
DEC2	DEC2	f	f = f – 2	1	1	C, DC, N, OV, Z
	DEC2	f,WREG	WREG = f – 2	1	1	C, DC, N, OV, Z
	DEC2	Ws,Wd	Wd = Ws – 2	1	1	C, DC, N, OV, Z
DISI	DISI	#lit14	Disable Interrupts for k Instruction Cycles	1	1	None
DIV	DIV.SW	Wm,Wn	Signed 16/16-bit Integer Divide	1	18	N, Z, C, OV
	DIV.SD	Wm,Wn	Signed 32/16-bit Integer Divide	1	18	N, Z, C, OV
	DIV.UW	Wm,Wn	Unsigned 16/16-bit Integer Divide	1	18	N, Z, C, OV
	DIV.UD	Wm,Wn	Unsigned 32/16-bit Integer Divide	1	18	N, Z, C, OV
EXCH	EXCH	Wns,Wnd	Swap Wns with Wnd	1	1	None
FF1L	FF1L	Ws,Wnd	Find First One from Left (MSb) Side	1	1	C
FF1R	FF1R	Ws,Wnd	Find First One from Right (LSb) Side	1	1	C

TABLE 29-2: INSTRUCTION SET OVERVIEW (CONTINUED)

Assembly Mnemonic	Assembly Syntax		Description	# of Words	# of Cycles	Status Flags Affected
GOTO	GOTO	Expr	Go to Address	2	2	None
	GOTO	Wn	Go to Indirect	1	2	None
INC	INC	f	f = f + 1	1	1	C, DC, N, OV, Z
	INC	f,WREG	WREG = f + 1	1	1	C, DC, N, OV, Z
	INC	Ws,Wd	Wd = Ws + 1	1	1	C, DC, N, OV, Z
INC2	INC2	f	f = f + 2	1	1	C, DC, N, OV, Z
	INC2	f,WREG	WREG = f + 2	1	1	C, DC, N, OV, Z
	INC2	Ws,Wd	Wd = Ws + 2	1	1	C, DC, N, OV, Z
IOR	IOR	f	f = f .IOR. WREG	1	1	N, Z
	IOR	f,WREG	WREG = f .IOR. WREG	1	1	N, Z
	IOR	#lit10,Wn	Wd = lit10 .IOR. Wd	1	1	N, Z
	IOR	Wb,Ws,Wd	Wd = Wb .IOR. Ws	1	1	N, Z
	IOR	Wb,#lit5,Wd	Wd = Wb .IOR. lit5	1	1	N, Z
LNK	LNK	#lit14	Link Frame Pointer	1	1	None
LSR	LSR	f	f = Logical Right Shift f	1	1	C, N, OV, Z
	LSR	f,WREG	WREG = Logical Right Shift f	1	1	C, N, OV, Z
	LSR	Ws,Wd	Wd = Logical Right Shift Ws	1	1	C, N, OV, Z
	LSR	Wb,Wns,Wnd	Wnd = Logical Right Shift Wb by Wns	1	1	N, Z
	LSR	Wb,#lit5,Wnd	Wnd = Logical Right Shift Wb by lit5	1	1	N, Z
MOV	MOV	f,Wn	Move f to Wn	1	1	None
	MOV	[Wns+Slit10],Wnd	Move [Wns+Slit10] to Wnd	1	1	None
	MOV	f	Move f to f	1	1	N, Z
	MOV	f,WREG	Move f to WREG	1	1	N, Z
	MOV	#lit16,Wn	Move 16-bit Literal to Wn	1	1	None
	MOV.b	#lit8,Wn	Move 8-bit Literal to Wn	1	1	None
	MOV	Wn,f	Move Wn to f	1	1	None
	MOV	Wns,[Wns+Slit10]	Move Wns to [Wns+Slit10]	1	1	
	MOV	Wso,Wdo	Move Ws to Wd	1	1	None
	MOV	WREG,f	Move WREG to f	1	1	N, Z
	MOV.D	Wns,Wd	Move Double from W(ns):W(ns+1) to Wd	1	2	None
	MOV.D	Ws,Wnd	Move Double from Ws to W(nd+1):W(nd)	1	2	None
MUL	MUL.SS	Wb,Ws,Wnd	{Wnd+1, Wnd} = Signed(Wb) * Signed(Ws)	1	1	None
	MUL.SU	Wb,Ws,Wnd	{Wnd+1, Wnd} = Signed(Wb) * Unsigned(Ws)	1	1	None
	MUL.US	Wb,Ws,Wnd	{Wnd+1, Wnd} = Unsigned(Wb) * Signed(Ws)	1	1	None
	MUL.UU	Wb,Ws,Wnd	{Wnd+1, Wnd} = Unsigned(Wb) * Unsigned(Ws)	1	1	None
	MUL.SU	Wb,#lit5,Wnd	{Wnd+1, Wnd} = Signed(Wb) * Unsigned(lit5)	1	1	None
	MUL.UU	Wb,#lit5,Wnd	{Wnd+1, Wnd} = Unsigned(Wb) * Unsigned(lit5)	1	1	None
	MUL	f	W3:W2 = f * WREG	1	1	None
NEG	NEG	f	f = $\overline{f}$ + 1	1	1	C, DC, N, OV, Z
	NEG	f,WREG	WREG = $\overline{f}$ + 1	1	1	C, DC, N, OV, Z
	NEG	Ws,Wd	Wd = $\overline{Ws}$ + 1	1	1	C, DC, N, OV, Z
NOP	NOP		No Operation	1	1	None
	NOPR		No Operation	1	1	None
POP	POP	f	Pop f from Top-of-Stack (TOS)	1	1	None
	POP	Wdo	Pop from Top-of-Stack (TOS) to Wdo	1	1	None
	POP.D	Wnd	Pop from Top-of-Stack (TOS) to W(nd):W(nd+1)	1	2	None
	POP.S		Pop Shadow Registers	1	1	All
PUSH	PUSH	f	Push f to Top-of-Stack (TOS)	1	1	None
	PUSH	Wso	Push Wso to Top-of-Stack (TOS)	1	1	None
	PUSH.D	Wns	Push W(ns):W(ns+1) to Top-of-Stack (TOS)	1	2	None
	PUSH.S		Push Shadow Registers	1	1	None

TABLE 29-2: INSTRUCTION SET OVERVIEW (CONTINUED)

Assembly Mnemonic	Assembly Syntax		Description	# of Words	# of Cycles	Status Flags Affected
PWRSAV	PWRSAV	#lit1	Go into Sleep or Idle mode	1	1	WDTO, Sleep
RCALL	RCALL	Expr	Relative Call	1	2	None
	RCALL	Wn	Computed Call	1	2	None
REPEAT	REPEAT	#lit14	Repeat Next Instruction lit14 + 1 times	1	1	None
	REPEAT	Wn	Repeat Next Instruction (Wn) + 1 times	1	1	None
RESET	RESET		Software Device Reset	1	1	None
RETFIE	RETFIE		Return from Interrupt	1	3 (2)	None
RETLW	RETLW	#lit10,Wn	Return with Literal in Wn	1	3 (2)	None
RETURN	RETURN		Return from Subroutine	1	3 (2)	None
RLC	RLC	f	f = Rotate Left through Carry f	1	1	C, N, Z
	RLC	f,WREG	WREG = Rotate Left through Carry f	1	1	C, N, Z
	RLC	Ws,Wd	Wd = Rotate Left through Carry Ws	1	1	C, N, Z
RLNC	RLNC	f	f = Rotate Left (No Carry) f	1	1	N, Z
	RLNC	f,WREG	WREG = Rotate Left (No Carry) f	1	1	N, Z
	RLNC	Ws,Wd	Wd = Rotate Left (No Carry) Ws	1	1	N, Z
RRC	RRC	f	f = Rotate Right through Carry f	1	1	C, N, Z
	RRC	f,WREG	WREG = Rotate Right through Carry f	1	1	C, N, Z
	RRC	Ws,Wd	Wd = Rotate Right through Carry Ws	1	1	C, N, Z
RRNC	RRNC	f	f = Rotate Right (No Carry) f	1	1	N, Z
	RRNC	f,WREG	WREG = Rotate Right (No Carry) f	1	1	N, Z
	RRNC	Ws,Wd	Wd = Rotate Right (No Carry) Ws	1	1	N, Z
SE	SE	Ws,Wnd	Wnd = Sign-Extended Ws	1	1	C, N, Z
SETM	SETM	f	f = FFFFh	1	1	None
	SETM	WREG	WREG = FFFFh	1	1	None
	SETM	Ws	Ws = FFFFh	1	1	None
SL	SL	f	f = Left Shift f	1	1	C, N, OV, Z
	SL	f,WREG	WREG = Left Shift f	1	1	C, N, OV, Z
	SL	Ws,Wd	Wd = Left Shift Ws	1	1	C, N, OV, Z
	SL	Wb,Wns,Wnd	Wnd = Left Shift Wb by Wns	1	1	N, Z
	SL	Wb,#lit5,Wnd	Wnd = Left Shift Wb by lit5	1	1	N, Z
SUB	SUB	f	f = f – WREG	1	1	C, DC, N, OV, Z
	SUB	f,WREG	WREG = f – WREG	1	1	C, DC, N, OV, Z
	SUB	#lit10,Wn	Wn = Wn – lit10	1	1	C, DC, N, OV, Z
	SUB	Wb,Ws,Wd	Wd = Wb – Ws	1	1	C, DC, N, OV, Z
	SUB	Wb,#lit5,Wd	Wd = Wb – lit5	1	1	C, DC, N, OV, Z
SUBB	SUBB	f	f = f – WREG – ($\overline{C}$)	1	1	C, DC, N, OV, Z
	SUBB	f,WREG	WREG = f – WREG – ($\overline{C}$)	1	1	C, DC, N, OV, Z
	SUBB	#lit10,Wn	Wn = Wn – lit10 – ($\overline{C}$)	1	1	C, DC, N, OV, Z
	SUBB	Wb,Ws,Wd	Wd = Wb – Ws – ($\overline{C}$)	1	1	C, DC, N, OV, Z
	SUBB	Wb,#lit5,Wd	Wd = Wb – lit5 – ($\overline{C}$)	1	1	C, DC, N, OV, Z
SUBR	SUBR	f	f = WREG – f	1	1	C, DC, N, OV, Z
	SUBR	f,WREG	WREG = WREG – f	1	1	C, DC, N, OV, Z
	SUBR	Wb,Ws,Wd	Wd = Ws – Wb	1	1	C, DC, N, OV, Z
	SUBR	Wb,#lit5,Wd	Wd = lit5 – Wb	1	1	C, DC, N, OV, Z
SUBBR	SUBBR	f	f = WREG – f – ($\overline{C}$)	1	1	C, DC, N, OV, Z
	SUBBR	f,WREG	WREG = WREG – f – ($\overline{C}$)	1	1	C, DC, N, OV, Z
	SUBBR	Wb,Ws,Wd	Wd = Ws – Wb – ($\overline{C}$)	1	1	C, DC, N, OV, Z
	SUBBR	Wb,#lit5,Wd	Wd = lit5 – Wb – ($\overline{C}$)	1	1	C, DC, N, OV, Z
SWAP	SWAP.b	Wn	Wn = Nibble Swap Wn	1	1	None
	SWAP	Wn	Wn = Byte Swap Wn	1	1	None
TBLRDH	TBLRDH	Ws,Wd	Read Prog<23:16> to Wd<7:0>	1	2	None

TABLE 29-2: INSTRUCTION SET OVERVIEW (CONTINUED)

Assembly Mnemonic	Assembly Syntax		Description	# of Words	# of Cycles	Status Flags Affected
TBLRDL	TBLRDL	Ws,Wd	Read Prog<15:0> to Wd	1	2	None
TBLWTH	TBLWTH	Ws,Wd	Write Ws<7:0> to Prog<23:16>	1	2	None
TBLWTL	TBLWTL	Ws,Wd	Write Ws to Prog<15:0>	1	2	None
ULNK	ULNK		Unlink Frame Pointer	1	1	None
XOR	XOR	f	f = f .XOR. WREG	1	1	N, Z
	XOR	f,WREG	WREG = f .XOR. WREG	1	1	N, Z
	XOR	#lit10,Wn	Wd = lit10 .XOR. Wd	1	1	N, Z
	XOR	Wb,Ws,Wd	Wd = Wb .XOR. Ws	1	1	N, Z
	XOR	Wb,#lit5,Wd	Wd = Wb .XOR. lit5	1	1	N, Z
ZE	ZE	Ws,Wnd	Wnd = Zero-Extend Ws	1	1	C, Z, N

NOTES:

30.0 ELECTRICAL CHARACTERISTICS

This section provides an overview of the PIC24FJ256DA210 family electrical characteristics. Additional information will be provided in future revisions of this document as it becomes available.

Absolute maximum ratings for the PIC24FJ256DA210 family are listed below. Exposure to these maximum rating conditions for extended periods may affect device reliability. Functional operation of the device at these, or any other conditions above the parameters indicated in the operation listings of this specification, is not implied.

Absolute Maximum Ratings(†)

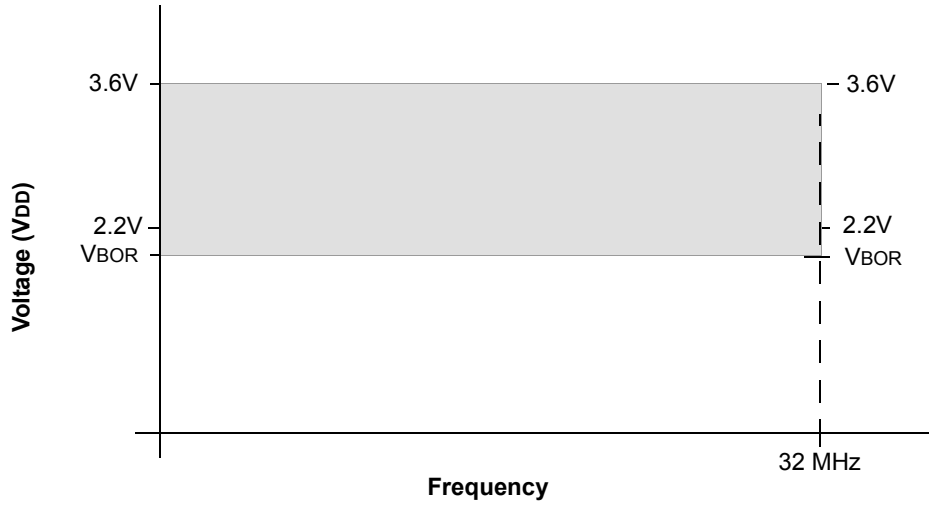
Ambient temperature under bias ..... -40°C to +100°C
Storage temperature ..... -65°C to +150°C
Voltage on V_{DD} with respect to V_{SS} ..... -0.3V to +4.0V
Voltage on any combined analog and digital pin and MCLR, with respect to V_{SS} ..... -0.3V to (V_{DD} + 0.3V)
Voltage on any digital only pin with respect to V_{SS} when V_{DD} < 3.0V ..... -0.3V to (V_{DD} + 0.3V)
Voltage on any digital only pin with respect to V_{SS} when V_{DD} > 3.0V ..... -0.3V to (+5.5V)
Voltage on V_{BUS} pin with respect to V_{SS}, independent of V_{DD} or V_{USB} ..... -0.3V to (+5.5V)
Maximum current out of V_{SS} pin ..... 300 mA
Maximum current into V_{DD} pin **(Note 1)** ..... 250 mA
Maximum output current sunk by any I/O pin ..... 25 mA
Maximum output current sourced by any I/O pin ..... 25 mA
Maximum current sunk by all ports ..... 200 mA
Maximum current sourced by all ports **(Note 1)** ..... 200 mA

Note 1: Maximum allowable current is a function of device maximum power dissipation (see Table 30-1).

†NOTICE: Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

30.1 DC Characteristics

FIGURE 30-1: PIC24FJ256DA210 FAMILY VOLTAGE-FREQUENCY GRAPH (INDUSTRIAL)



Note: VCAP (nominal On-Chip Regulator output voltage) = 1.8V.

TABLE 30-1: THERMAL OPERATING CONDITIONS

Rating	Symbol	Min	Typ	Max	Unit
PIC24FJ256DA210 family:					
Operating Junction Temperature Range	TJ	-40	—	+125	°C
Operating Ambient Temperature Range	TA	-40	—	+85	°C
Power Dissipation (with ENVREG = 1): Internal Chip Power Dissipation: PINT = VDD x (IDD – Σ IOH) I/O Pin Power Dissipation: PI/O = Σ ({VDD – VOH} x IOH) + Σ (VOL x IOL)	PD	PINT + PI/O			W
Maximum Allowed Power Dissipation	PDMAX	(TJMAX – TA)/θJA			W

TABLE 30-2: THERMAL PACKAGING CHARACTERISTICS

Characteristic	Symbol	Typ	Max	Unit	Note
Package Thermal Resistance, 12x12x1 mm TQFP	θJA	69.4	—	°C/W	**(Note 1)**
Package Thermal Resistance, 10x10x1 mm TQFP	θJA	76.6	—	°C/W	**(Note 1)**
Package Thermal Resistance, 9x9x0.9 mm QFN	θJA	28.0	—	°C/W	**(Note 1)**
Package Thermal Resistance, 10x10x1.1 mm BGA	θJA	40.2	—	°C/W	**(Note 1)**

Note 1: Junction to ambient thermal resistance, Theta-JA (θJA) numbers are achieved by package simulations.

TABLE 30-3: DC CHARACTERISTICS: TEMPERATURE AND VOLTAGE SPECIFICATIONS

DC CHARACTERISTICS			Standard Operating Conditions: 2.2V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial				
Param No.	**Symbol**	**Characteristic**	**Min**	**Typ**	**Max**	**Units**	**Conditions**
Operating Voltage							
DC10	**Supply Voltage**						
	VDD		VBOR	—	3.6	V	Regulator enabled
	VCAP(2)		—	1.8V	—	V	Regulator enabled
DC12	VDR	**RAM Data Retention Voltage(1)**	1.5	—	—	V	
DC16	VPOR	**VDD Start Voltage** to Ensure Internal Power-on Reset Signal	Vss	—	—	V	
DC17	SVDD	**VDD Rise Rate** to Ensure Internal Power-on Reset Signal	0.05	—	—	V/ms	0-3.3V in 66 ms 0-2.5V in 50 ms
	VBOR	**Brown-out Reset Voltage** on VDD Transition, High-to-Low	2.0	2.10	2.2	V	Regulator enabled
	VLVD	LVD Trip Voltage	—	VBOR + 0.10	—	V	

Note 1: This is the limit to which the RAM data can be retained, while the on-chip regulator output voltage starts following the VDD.

2: This is the on-chip regulator output voltage specification.

TABLE 30-4: DC CHARACTERISTICS: OPERATING CURRENT (IDD)

DC CHARACTERISTICS			Standard Operating Conditions: 2.2V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial			
Parameter No.	**Typical(1)**	**Max**	**Units**	**Conditions**		
Operating Current (IDD)(2)						
DC20D	0.8	1.3	mA	-40°C	3.3V(3)	1 MIPS
DC20E	0.8	1.3	mA	+25°C		
DC20F	0.8	1.3	mA	+85°C		
DC23D	3.0	4.8	mA	-40°C	3.3V(3)	4 MIPS
DC23E	3.0	4.8	mA	+25°C		
DC23F	3.0	4.8	mA	+85°C		
DC24D	12.0	18	mA	-40°C	3.3V(3)	16 MIPS
DC24E	12.0	18	mA	+25°C		
DC24F	12.0	18	mA	+85°C		
DC31D	55	95	µA	-40°C	3.3V(3)	LPRC (31 kHz)
DC31E	55	95	µA	+25°C		
DC31F	135	225	µA	+85°C		

Note 1: Data in "Typical" column is at 3.3V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors, such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature, also have an impact on the current consumption. The test conditions for all IDD measurements are as follows: OSCI driven with external square wave from rail to rail. All I/O pins are configured as inputs and pulled to VDD. MCLR = VDD; WDT and FSCM are disabled. CPU, SRAM, program memory and data memory are operational. No peripheral modules are operating and all of the Peripheral Module Disable (PMD) bits are set.

3: On-chip voltage regulator enabled (ENVREG tied to VDD). Brown-out Reset (BOR) is enabled.

TABLE 30-5: DC CHARACTERISTICS: IDLE CURRENT (IIDLE)

DC CHARACTERISTICS			Standard Operating Conditions: 2.2V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial			
Parameter No.	**Typical(1)**	**Max**	**Units**	**Conditions**		
Idle Current (IIDLE)(2)						
DC40D	170	320	µA	-40°C	3.3V(3)	1 MIPS
DC40E	170	320	µA	+25°C		
DC40F	220	380	µA	+85°C		
DC43D	0.6	1.2	mA	-40°C	3.3V(3)	4 MIPS
DC43E	0.6	1.2	mA	+25°C		
DC43F	0.7	1.2	mA	+85°C		
DC47D	2.3	4.8	mA	-40°C	3.3V(3)	16 MIPS
DC47E	2.3	4.8	mA	+25°C		
DC47F	2.4	4.8	mA	+85°C		
DC50D	0.8	1.8	mA	-40°C	3.3V(3)	FRC (4 MIPS)
DC50E	0.8	1.8	mA	+25°C		
DC50F	1.0	1.8	mA	+85°C		
DC51D	40.0	85	µA	-40°C	3.3V(3)	LPRC (31 kHz)
DC51E	40.0	85	µA	+25°C		
DC51F	120.0	210	µA	+85°C		

Note 1: Data in "Typical" column is at 3.3V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

2: Base IIDLE current is measured with the core off; OSCI driven with external square wave from rail to rail. All I/O pins are configured as inputs and pulled to VDD. $\overline{MCLR}$ = VDD; WDT and FSCM are disabled. No peripheral modules are operating and all of the Peripheral Module Disable (PMD) bits are set.

3: On-chip voltage regulator enabled (ENVREG tied to VDD). Brown-out Reset (BOR) is enabled.

TABLE 30-6: DC CHARACTERISTICS: POWER-DOWN CURRENT (IPD)

DC CHARACTERISTICS			Standard Operating Conditions: 2.2V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial			
Parameter No.	**Typical(1)**	**Max**	**Units**	**Conditions**		
Power-Down Current (IPD)(2)						
DC60D	20.0	45	µA	-40°C	3.3V(3)	Base power-down current(4)
DC60E	20.0	45	µA	+25°C		
DC60H	55.0	105	µA	+60°C		
DC60F	95.0	185	µA	+85°C		
DC61D	1.0	3.5	µA	-40°C	3.3V(3)	31 kHz LPRC oscillator with RTCC, WDT or Timer1:ΔILPRC(4)
DC61E	1.0	3.5	µA	+25°C		
DC61H	1.0	3.5	µA	+60°C		
DC61F	2.5	6.5	µA	+85°C		
DC62D	1.5	6	µA	-40°C	3.3V(3)	Low drive strength, 32 kHz crystal with RTCC or Timer1: ΔISOSC; SOSCSEL<1:0> = 01(4)
DC62E	1.5	6	µA	+25°C		
DC62H	1.5	6	µA	+60°C		
DC62F	8.0	18	µA	+85°C		
DC63D	4.0	18	µA	-40°C	3.3V(3)	32 kHz crystal with RTCC or Timer1: ΔISOSC; SOSCSEL<1:0> = 11(4)
DC63E	4.0	18	µA	+25°C		
DC63H	6.5	18	µA	+60°C		
DC63F	12.0	25	µA	+85°C		

Note 1: Data in the Typical column is at 3.3V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

2: Base IPD is measured with the device in Sleep mode (all peripherals and clocks are shut down). All I/Os are configured as inputs and pulled high. WDT, etc., are all switched off, PMSLP bit is clear and the Peripheral Module Disable (PMD) bits for all unused peripherals are set.

3: On-chip voltage regulator enabled (ENVREG tied to VDD). Brown-out Reset (BOR) is enabled.

4: The Δ current is the additional current consumed when the module is enabled. This current should be added to the base IPD current.

TABLE 30-7: DC CHARACTERISTICS: I/O PIN INPUT SPECIFICATIONS

DC CHARACTERISTICS			Standard Operating Conditions: 2.2V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial				
Param No.	**Symbol**	**Characteristic**	**Min**	**Typ(1)**	**Max**	**Units**	**Conditions**
	VIL	**Input Low Voltage(3)**					
DI10		I/O Pins with ST Buffer	VSS	—	0.2 VDD	V	
DI11		I/O Pins with TTL Buffer	VSS	—	0.15 VDD	V	
DI15		$\overline{MCLR}$	VSS	—	0.2 VDD	V	
DI16		OSCI (XT mode)	VSS	—	0.2 VDD	V	
DI17		OSCI (HS mode)	VSS	—	0.2 VDD	V	
DI18		I/O Pins with I2C™ Buffer:	VSS	—	0.3 VDD	V	
DI19		I/O Pins with SMBus Buffer:	VSS	—	0.8	V	SMBus enabled
	VIH	**Input High Voltage(3)**					
DI20		I/O Pins with ST Buffer: with Analog Functions, Digital Only	 0.8 VDD 0.8 VDD	 — —	 VDD 5.5	 V V	
DI21		I/O Pins with TTL Buffer: with Analog Functions, Digital Only	 0.25 VDD + 0.8 0.25 VDD + 0.8	 — —	 VDD 5.5	 V V	
DI25		$\overline{MCLR}$	0.8 VDD	—	VDD	V	
DI26		OSCI (XT mode)	0.7 VDD	—	VDD	V	
DI27		OSCI (HS mode)	0.7 VDD	—	VDD	V	
DI28		I/O Pins with I2C™ Buffer: with Analog Functions, Digital Only	 0.7 VDD 0.7 VDD	 — —	 VDD 5.5	 V V	
DI29		I/O Pins with SMBus Buffer: with Analog Functions, Digital Only	 2.1 2.1	 	 VDD 5.5	 V V	2.5V ≤ VPIN ≤ VDD
DI30	ICNPU	**CNxx Pull-up Current**	150	350	550	µA	VDD = 3.3V, VPIN = VSS
DI30A	ICNPD	**CNxx Pull-down Current**	15	70	150	µA	VDD = 3.3V, VPIN = VDD
	IIL	**Input Leakage Current(2)**					
DI50		I/O Ports	—	—	±1	µA	VSS ≤ VPIN ≤ VDD, pin at high-impedance
DI51		Analog Input Pins	—	—	±1	µA	VSS ≤ VPIN ≤ VDD, pin at high-impedance
DI55		$\overline{MCLR}$	—	—	±1	µA	VSS ≤ VPIN ≤ VDD
DI56		OSCI/CLKI	—	—	±1	µA	VSS ≤ VPIN ≤ VDD, EC, XT and HS modes

Note 1: Data in "Typ" column is at 3.3V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

2: Negative current is defined as current sourced by the pin.

3: Refer to Table 1-1 for I/O pins buffer types.

TABLE 30-8: DC CHARACTERISTICS: I/O PIN OUTPUT SPECIFICATIONS

DC CHARACTERISTICS			Standard Operating Conditions: 2.2V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial				
Param No.	Symbol	Characteristic	Min	Typ(1)	Max	Units	Conditions
	VOL	**Output Low Voltage**					
DO10		I/O Ports	—	—	0.4	V	IOL = 6.6 mA, VDD = 3.6V
			—	—	0.4	V	IOL = 5.0 mA, VDD = 2.2V
DO16		OSCO/CLKO	—	—	0.4	V	IOL = 6.6 mA, VDD = 3.6V
			—	—	0.4	V	IOL = 5.0 mA, VDD = 2.2V
	VOH	**Output High Voltage**					
DO20		I/O Ports	3.0	—	—	V	IOH = -3.0 mA, VDD = 3.6V
			2.4	—	—	V	IOH = -6.0 mA, VDD = 3.6V
			1.65	—	—	V	IOH = -1.0 mA, VDD = 2.2V
			1.4	—	—	V	IOH = -3.0 mA, VDD = 2.2V
DO26		OSCO/CLKO	2.4	—	—	V	IOH = -6.0 mA, VDD = 3.6V
			1.4	—	—	V	IOH = -1.0 mA, VDD = 2.2V

Note 1: Data in "Typ" column is at 3.3V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

TABLE 30-9: DC CHARACTERISTICS: PROGRAM MEMORY

DC CHARACTERISTICS			Standard Operating Conditions: 2.2V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial				
Param No.	Symbol	Characteristic	Min	Typ(1)	Max	Units	Conditions
		Program Flash Memory					
D130	EP	Cell Endurance	10000	—	—	E/W	-40°C to +85°C
D131	VPR	VDD for Read	VMIN	—	3.6	V	VMIN = Minimum operating voltage
D132B		VDD for Self-Timed Write	VMIN	—	3.6	V	VMIN = Minimum operating voltage
D133A	TIW	Self-Timed Word Write Cycle Time	—	20	—	µs	
		Self-Timed Row Write Cycle Time	—	1.5	—	ms	
D133B	TIE	Self-Timed Page Erase Time	20	—	40	ms	
D134	TRETD	Characteristic Retention	20	—	—	Year	If no other specifications are violated
D135	IDDP	Supply Current during Programming	—	16	—	mA	

Note 1: Data in "Typ" column is at 3.3V, 25°C unless otherwise stated.

TABLE 30-10: INTERNAL VOLTAGE REGULATOR SPECIFICATIONS

Operating Conditions: -40°C < TA < +85°C (unless otherwise stated)							
Param No.	**Symbol**	**Characteristics**	**Min**	**Typ**	**Max**	**Units**	**Comments**
	VRGOUT	Regulator Output Voltage	—	1.8	—	V	
	VBG	Internal Band Gap Reference	—	1.2	—	V	
	CEFC	External Filter Capacitor Value	4.7	10	—	µF	Series resistance < 3 Ohm recommended; < 5 Ohm required.
	TVREG		—	10	—	µs	VREGS = 1, VREGS = 0 with WUTSEL<1:0> = 01 or any POR or BOR
			—	190	—	µs	Sleep wake-up with VREGS = 0 and WUTSEL<1:0> = 11
	TBG	Band Gap Reference Start-up Time	—	1	—	ms	

30.2 AC Characteristics and Timing Parameters

The information contained in this section defines the PIC24FJ256DA210 family AC characteristics and timing parameters.

TABLE 30-11: TEMPERATURE AND VOLTAGE SPECIFICATIONS – AC

AC CHARACTERISTICS	**Standard Operating Conditions: 2.2V to 3.6V (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for Industrial Operating voltage VDD range as described in **Section 30.1 "DC Characteristics"**.

FIGURE 30-2: LOAD CONDITIONS FOR DEVICE TIMING SPECIFICATIONS

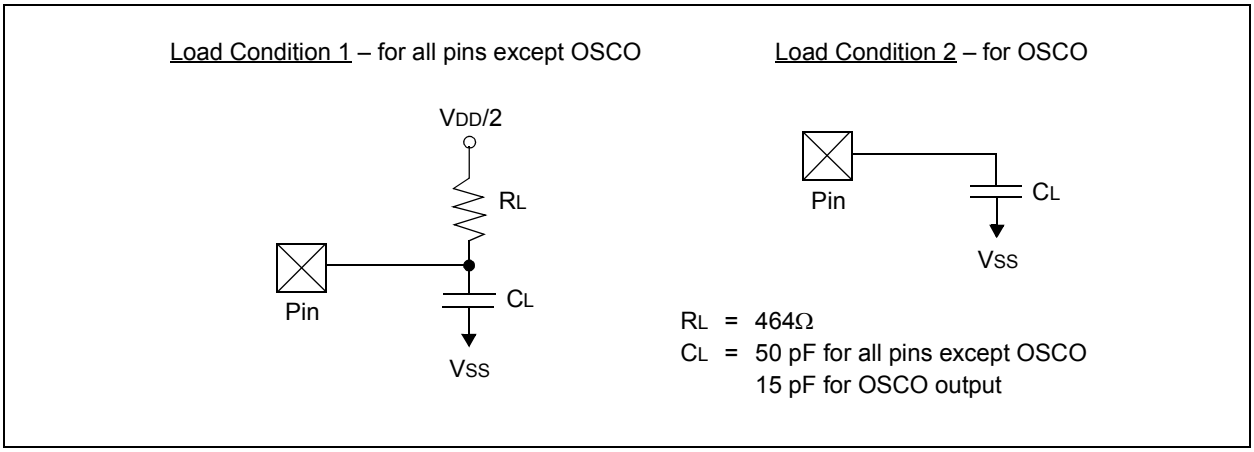


TABLE 30-12: CAPACITIVE LOADING REQUIREMENTS ON OUTPUT PINS

Param No.	Symbol	Characteristic	Min	Typ(1)	Max	Units	Conditions
DO50	COSCO	OSCO/CLKO Pin	—	—	15	pF	In XT and HS modes when external clock is used to drive OSCI
DO56	CIO	All I/O Pins and OSCO	—	—	50	pF	EC mode
DO58	CB	SCLx, SDAx	—	—	400	pF	In I^2C™ mode

Note 1: Data in "Typ" column is at 3.3V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

FIGURE 30-3: EXTERNAL CLOCK TIMING

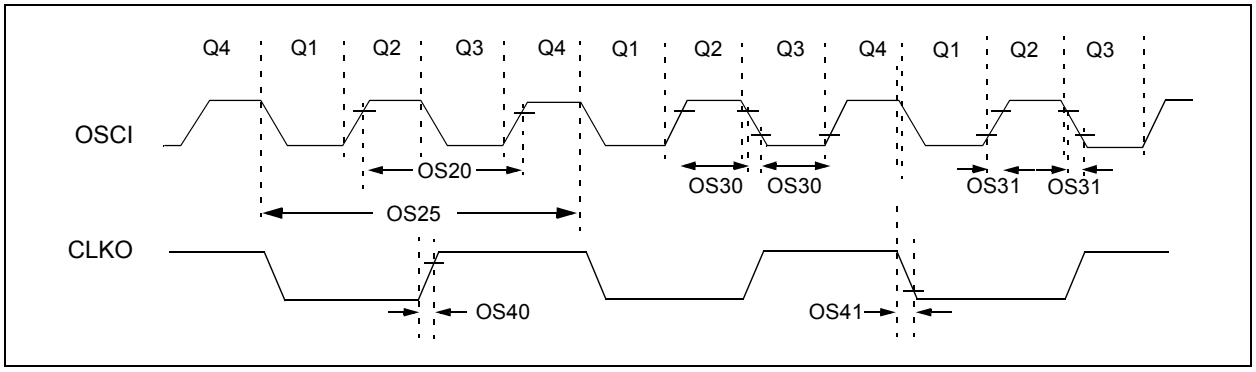


TABLE 30-13: EXTERNAL CLOCK TIMING REQUIREMENTS

AC CHARACTERISTICS			Standard Operating Conditions: 2.2V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial				
Param No.	Symbol	Characteristic	Min	Typ(1)	Max	Units	Conditions
OS10	FOSC	External CLKI Frequency (External clocks allowed only in EC mode)	DC 4	— —	32 48	MHz MHz	EC ECPLL
		Oscillator Frequency	3.5 4 10 10 31	— — — — —	10 8 32 32 33	MHz MHz MHz MHz kHz	XT XTPLL HS HSPLL SOSC
OS20	TOSC	TOSC = 1/FOSC	—	—	—	—	See parameter OS10 for FOSC value
OS25	TCY	Instruction Cycle Time(2)	62.5	—	DC	ns	
OS30	TosL, TosH	External Clock in (OSCI) High or Low Time	0.45 x TOSC	—	—	ns	EC
OS31	TosR, TosF	External Clock in (OSCI) Rise or Fall Time	—	—	20	ns	EC
OS40	TckR	CLKO Rise Time(3)	—	6	10	ns	
OS41	TckF	CLKO Fall Time(3)	—	6	10	ns	

Note 1: Data in "Typ" column is at 3.3V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

2: Instruction cycle period (TCY) equals two times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "Min." values with an external clock applied to the OSCI/CLKI pin. When an external clock input is used, the "Max." cycle time limit is "DC" (no clock) for all devices.

3: Measurements are taken in EC mode. The CLKO signal is measured on the OSCO pin. CLKO is low for the Q1-Q2 period (1/2 TCY) and high for the Q3-Q4 period (1/2 TCY).

TABLE 30-14: PLL CLOCK TIMING SPECIFICATIONS (VDD = 2.2V TO 3.6V)

AC CHARACTERISTICS			Standard Operating Conditions: 2.2V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial				
Param No.	Symbol	Characteristic(1)	Min	Typ(2)	Max	Units	Conditions
OS50	FPLLI	PLL Input Frequency Range(2)	4	—	48	MHz	ECPLL mode
			4		32	MHz	HSPLL mode
			4		8	MHz	XTPLL mode
OS51	FSYS	PLL Output Frequency Range	95.76	—	96.24	MHz	
OS52	TLOCK	PLL Start-up Time (Lock Time)	—	—	200	µs	
OS53	DCLK	CLKO Stability (Jitter)	-0.25	—	0.25	%	

Note 1: These parameters are characterized but not tested in manufacturing.

2: Data in "Typ" column is at 3.3V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

TABLE 30-15: INTERNAL RC ACCURACY

AC CHARACTERISTICS		Standard Operating Conditions: 2.2V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial					
Param No.	**Characteristic**	**Min**	**Typ**	**Max**	**Units**	**Conditions**	
F20	**FRC Accuracy @ 8 MHz(1,2)**	-1	±0.15	1	%	-40°C ≤ TA ≤ +85°C	2.2V ≤ VDD ≤ 3.6V
F21	**LPRC @ 31 kHz**	-20	—	20	%	-40°C ≤ TA ≤ +85°C	VCAP (on-chip regulator output voltage) = 1.8V

Note 1: Frequency calibrated at 25°C and 3.3V. OSCTUN bits can be used to compensate for temperature drift.

2: To achieve this accuracy, physical stress applied to the microcontroller package (ex., by flexing the PCB) must be kept to a minimum.

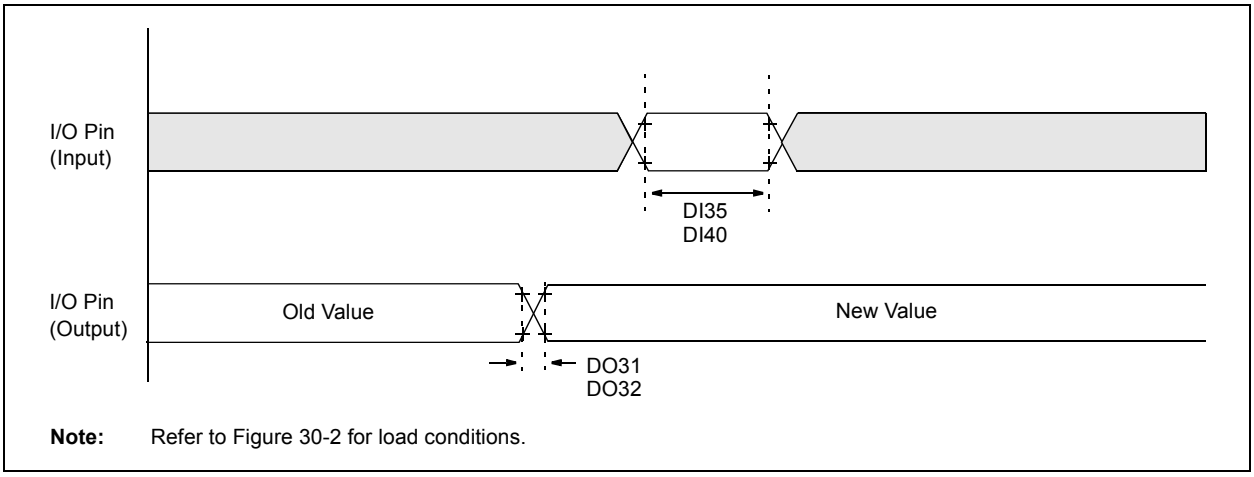
TABLE 30-16: RC OSCILLATOR START-UP TIME

AC CHARACTERISTICS		Standard Operating Conditions: 2.2V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial				
Param No.	**Characteristic**	**Min**	**Typ**	**Max**	**Units**	**Conditions**
	TFRC	—	15	—	µs	
	TLPRC	—	50	—	µs	

TABLE 30-17: RESET AND BROWN-OUT RESET REQUIREMENTS

AC CHARACTERISTICS			Standard Operating Conditions: 2.2V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial				
Param No.	**Symbol**	**Characteristic**	**Min**	**Typ**	**Max**	**Units**	**Conditions**
SY10	TMCL	$\overline{\text{MCLR}}$ Pulse width (Low)	2	—	—	µs	
SY12	TPOR	Power-on Reset Delay	—	2	—	µs	
SY13	TIOZ	I/O High-Impedance from $\overline{\text{MCLR}}$ Low or Watchdog Timer Reset	—	—	100	ns	
SY25	TBOR	Brown-out Reset Pulse Width	1	—	—	µs	VDD ≤ VBOR
	TRST	Internal State Reset Time	—	50	—	µs	

FIGURE 30-4: CLKO AND I/O TIMING CHARACTERISTICS



Note: Refer to Figure 30-2 for load conditions.

TABLE 30-18: CLKO AND I/O TIMING REQUIREMENTS

AC CHARACTERISTICS			Standard Operating Conditions: 2.2V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial				
Param No.	**Symbol**	**Characteristic**	**Min**	**Typ(1)**	**Max**	**Units**	**Conditions**
DO31	TIOR	Port Output Rise Time	—	10	25	ns	
DO32	TIOF	Port Output Fall Time	—	10	25	ns	
DI35	TINP	INTx Pin High or Low Time (input)	20	—	—	ns	
DI40	TRBP	CNx High or Low Time (input)	2	—	—	TCY	

Note 1: Data in "Typ" column is at 3.3V, 25°C unless otherwise stated.

TABLE 30-19: ADC MODULE SPECIFICATIONS

AC CHARACTERISTICS			Standard Operating Conditions: 2.2V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C				
Param No.	**Symbol**	**Characteristic**	**Min.**	**Typ**	**Max.**	**Units**	**Conditions**
Device Supply							
AD01	AVDD	Module VDD Supply	Greater of VDD – 0.3 or 2.2	—	Lesser of VDD + 0.3 or 3.6	V	
AD02	AVSS	Module VSS Supply	VSS – 0.3	—	VSS + 0.3	V	
Reference Inputs							
AD05	VREFH	Reference Voltage High	AVSS + 1.7	—	AVDD	V	
AD06	VREFL	Reference Voltage Low	AVSS	—	AVDD – 1.7	V	
AD07	VREF	Absolute Reference Voltage	AVSS – 0.3	—	AVDD + 0.3	V	
Analog Input							
AD10	VINH-VINL	Full-Scale Input Span	VREFL	—	VREFH	V	**(Note 2)**
AD11	VIN	Absolute Input Voltage	AVSS – 0.3	—	AVDD + 0.3	V	
AD12	VINL	Absolute VINL Input Voltage	AVSS – 0.3		AVDD/2	V	
AD13		Leakage Current	—	±1.0	±610	nA	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V, Source Impedance = 2.5 kΩ
AD17	RIN	Recommended Impedance of Analog Voltage Source	—	—	2.5K	Ω	10-bit
ADC Accuracy							
AD20B	Nr	Resolution	—	10	—	bits	
AD21B	INL	Integral Nonlinearity	—	±1	<±2	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD22B	DNL	Differential Nonlinearity	—	±0.5	<±1	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD23B	GERR	Gain Error	—	±1	±3	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD24B	EOFF	Offset Error	—	±1	±2	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD25B		Monotonicity(1)	—	—	—	—	Guaranteed

Note 1: The ADC conversion result never decreases with an increase in the input voltage and has no missing codes.

2: Measurements taken with external VREF+ and VREF- used as the ADC voltage reference.

TABLE 30-20: ADC CONVERSION TIMING REQUIREMENTS(1)

AC CHARACTERISTICS			Standard Operating Conditions: 2.2V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C				
Param No.	**Symbol**	**Characteristic**	**Min.**	**Typ**	**Max.**	**Units**	**Conditions**
Clock Parameters							
AD50	TAD	ADC Clock Period	75	—	—	ns	TCY = 75 ns, AD1CON3 in default state
AD51	tRC	ADC Internal RC Oscillator Period	—	250	—	ns	
Conversion Rate							
AD55	tCONV	Conversion Time	—	12	—	TAD	
AD56	FCNV	Throughput Rate	—	—	500	ksps	AVDD > 2.7V
AD57	tSAMP	Sample Time	—	1	—	TAD	
Clock Parameters							
AD61	tPSS	Sample Start Delay from Setting Sample bit (SAMP)	2	—	3	TAD	

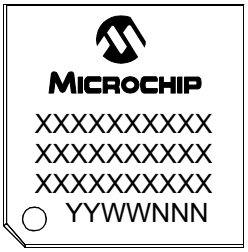
Note 1: Because the sample caps will eventually lose charge, clock rates below 10 kHz can affect linearity performance, especially at elevated temperatures.

NOTES:

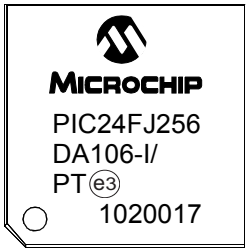
31.0 PACKAGING INFORMATION

31.1 Package Marking Information

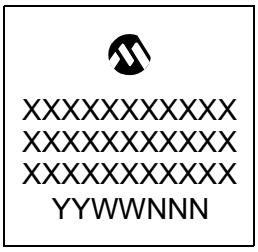
64-Lead TQFP (10x10x1 mm)



Example

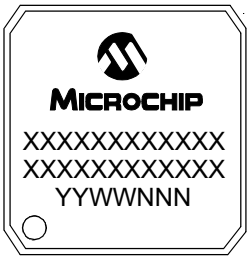


64-Lead QFN (9x9x0.9 mm)

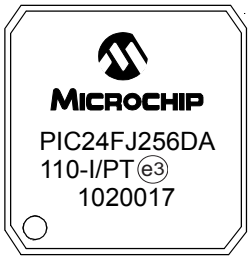


Example

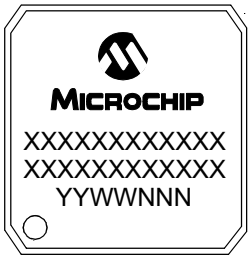
100-Lead TQFP (12x12x1 mm)



Example



121-BGA (10x10x1.1 mm)



Example

Legend:	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week '01')
	NNN	Alphanumeric traceability code
	(e3)	Pb-free JEDEC designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator ((e3)) can be found on the outer packaging for this package.

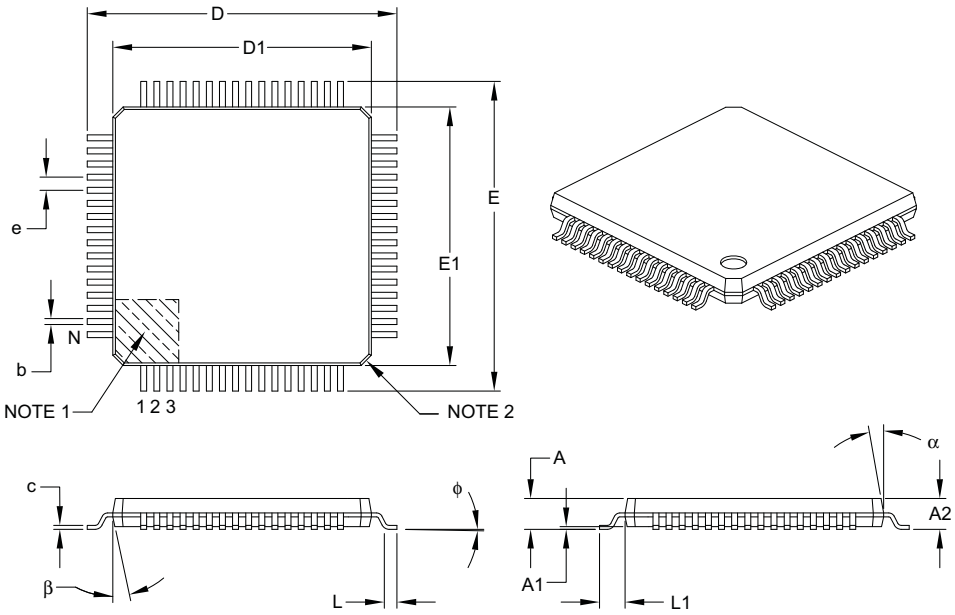
Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

31.2 Package Details

The following sections give the technical details of the packages.

64-Lead Plastic Thin Quad Flatpack (PT) – 10x10x1 mm Body, 2.00 mm [TQFP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Leads	N		64	
Lead Pitch	e		0.50 BSC	
Overall Height	A	–	–	1.20
Molded Package Thickness	A2	0.95	1.00	1.05
Standoff	A1	0.05	–	0.15
Foot Length	L	0.45	0.60	0.75
Footprint	L1		1.00 REF	
Foot Angle	ϕ	0°	3.5°	7°
Overall Width	E		12.00 BSC	
Overall Length	D		12.00 BSC	
Molded Package Width	E1		10.00 BSC	
Molded Package Length	D1		10.00 BSC	
Lead Thickness	c	0.09	–	0.20
Lead Width	b	0.17	0.22	0.27
Mold Draft Angle Top	α	11°	12°	13°
Mold Draft Angle Bottom	β	11°	12°	13°

Notes:

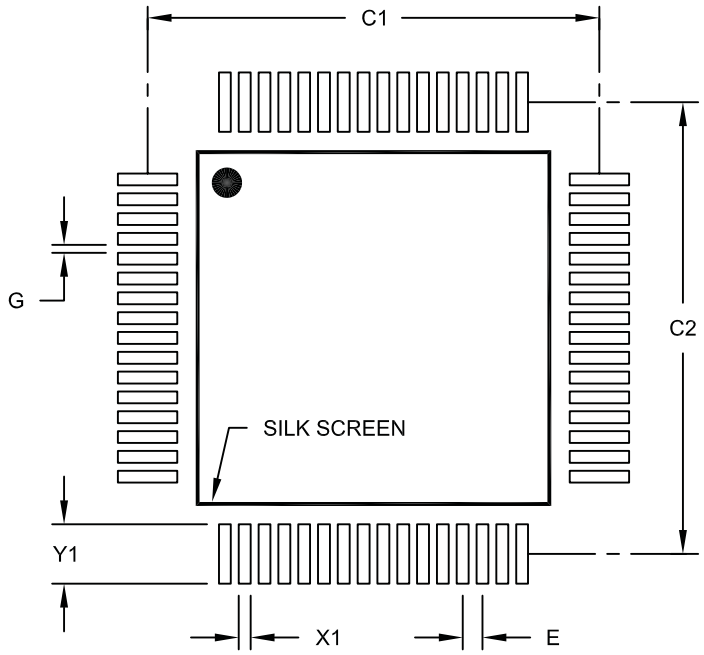
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Chamfers at corners are optional; size may vary.
3. Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.25 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-085B

64-Lead Plastic Thin Quad Flatpack (PT) – 10x10x1 mm Body, 2.00 mm [TQFP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.50 BSC	
Contact Pad Spacing	C1		11.40	
Contact Pad Spacing	C2		11.40	
Contact Pad Width (X64)	X1			0.30
Contact Pad Length (X64)	Y1			1.50
Distance Between Pads	G	0.20		

Notes:

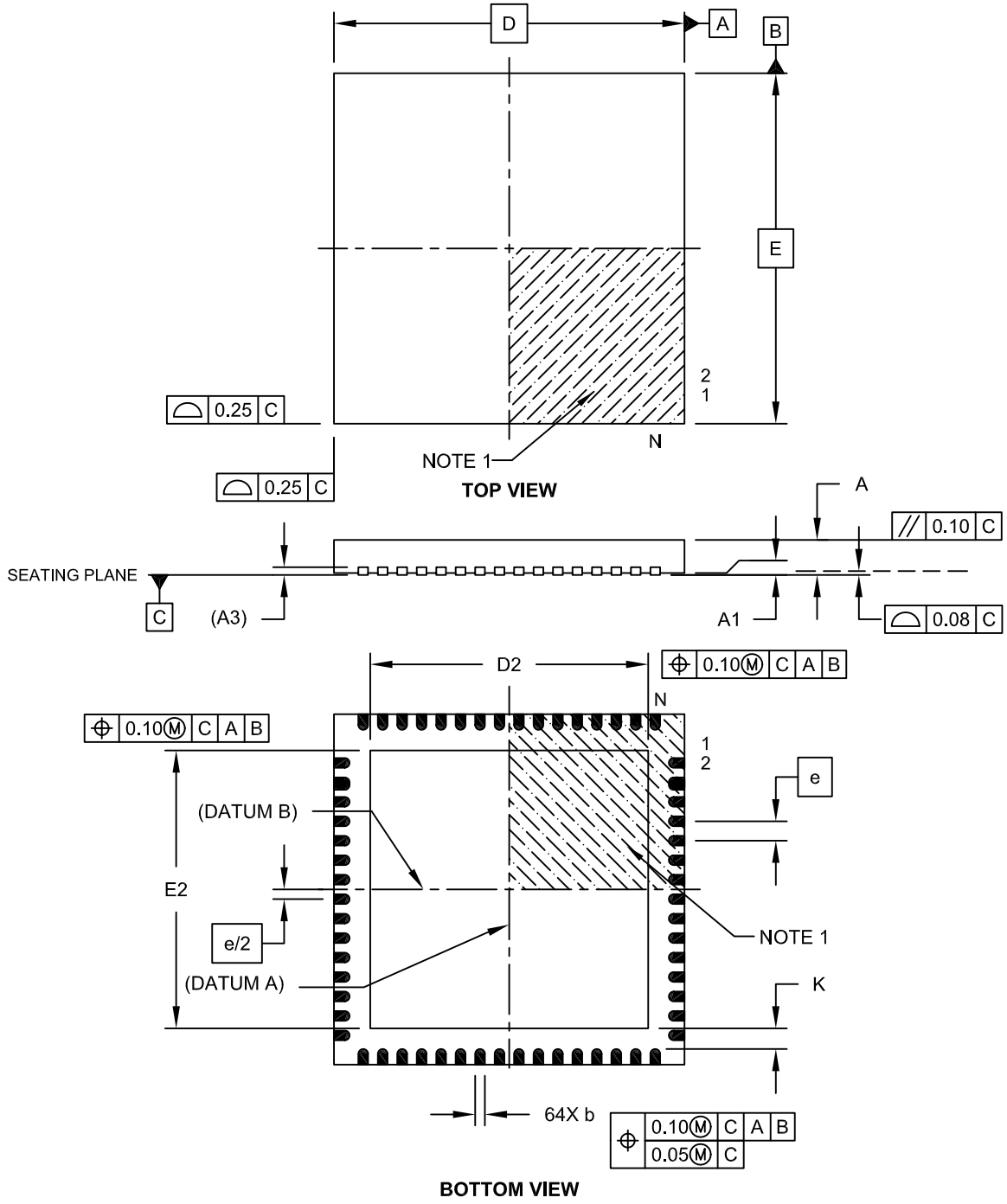
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2085A

64-Lead Plastic Quad Flat, No Lead Package (MR) – 9x9x0.9 mm Body [QFN]

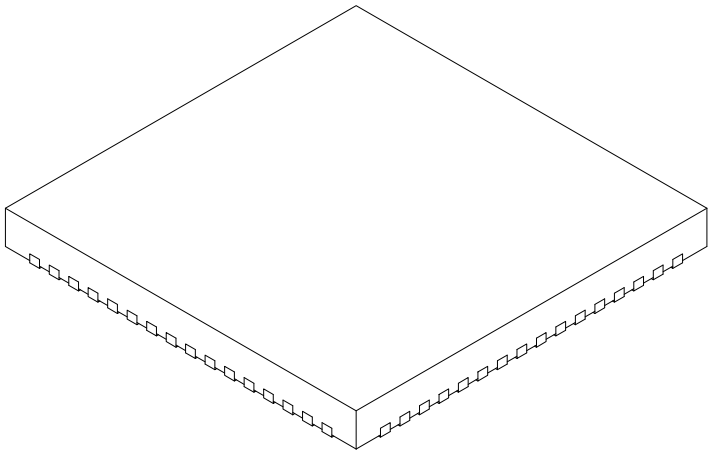
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Microchip Technology Drawing C04-149B Sheet 1 of 2

64-Lead Plastic Quad Flat, No Lead Package (MR) – 9x9x0.9 mm Body [QFN]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		64	
Pitch	e		0.50 BSC	
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3		0.20 REF	
Overall Width	E		9.00 BSC	
Exposed Pad Width	E2	7.05	7.15	7.50
Overall Length	D		9.00 BSC	
Exposed Pad Length	D2	7.05	7.15	7.50
Contact Width	b	0.18	0.25	0.30
Contact Length	L	0.30	0.40	0.50
Contact-to-Exposed Pad	K	0.20	-	-

Notes:

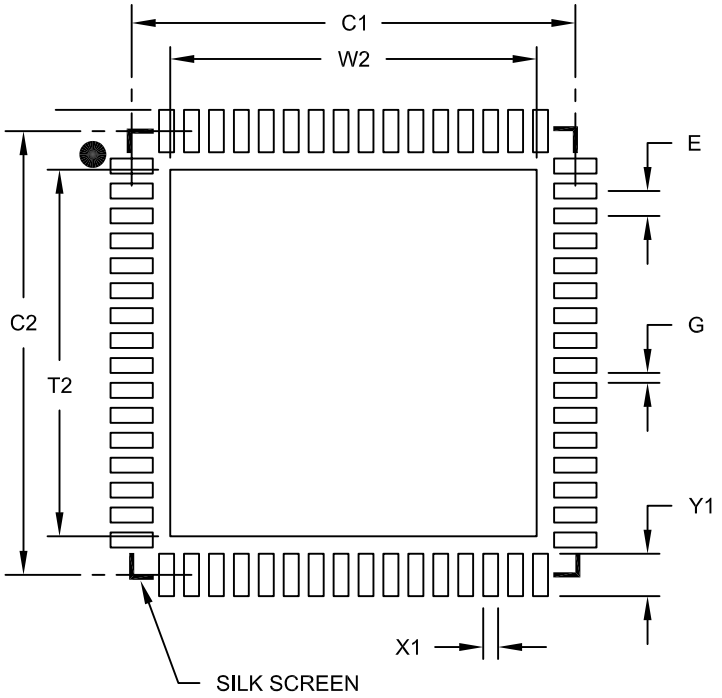
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated.
3. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-149B Sheet 2 of 2

64-Lead Plastic Quad Flat, No Lead Package (MR) – 9x9x0.9 mm Body [QFN]
With 0.40 mm Contact Length

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.50 BSC	
Optional Center Pad Width	W2			7.35
Optional Center Pad Length	T2			7.35
Contact Pad Spacing	C1		8.90	
Contact Pad Spacing	C2		8.90	
Contact Pad Width (X64)	X1			0.30
Contact Pad Length (X64)	Y1			0.85
Distance Between Pads	G	0.20		

Notes:

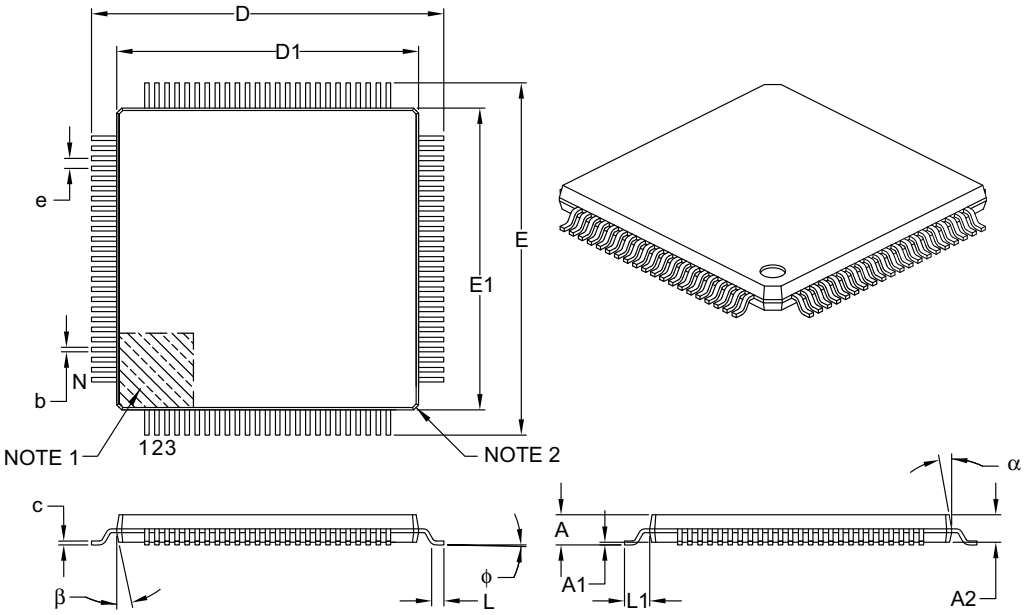
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2149A

100-Lead Plastic Thin Quad Flatpack (PT) – 12x12x1 mm Body, 2.00 mm [TQFP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Leads	N		100	
Lead Pitch	e		0.40 BSC	
Overall Height	A	–	–	1.20
Molded Package Thickness	A2	0.95	1.00	1.05
Standoff	A1	0.05	–	0.15
Foot Length	L	0.45	0.60	0.75
Footprint	L1		1.00 REF	
Foot Angle	ϕ	0°	3.5°	7°
Overall Width	E		14.00 BSC	
Overall Length	D		14.00 BSC	
Molded Package Width	E1		12.00 BSC	
Molded Package Length	D1		12.00 BSC	
Lead Thickness	c	0.09	–	0.20
Lead Width	b	0.13	0.18	0.23
Mold Draft Angle Top	α	11°	12°	13°
Mold Draft Angle Bottom	β	11°	12°	13°

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Chamfers at corners are optional; size may vary.
3. Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.25 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

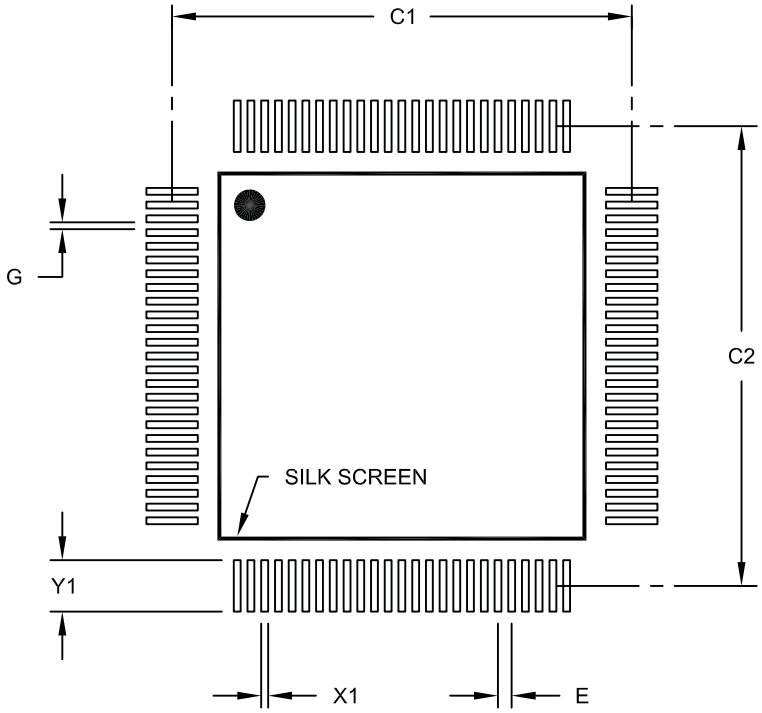
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-100B

100-Lead Plastic Thin Quad Flatpack (PT) – 12x12x1 mm Body, 2.00 mm [TQFP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.40 BSC	
Contact Pad Spacing	C1		13.40	
Contact Pad Spacing	C2		13.40	
Contact Pad Width (X100)	X1			0.20
Contact Pad Length (X100)	Y1			1.50
Distance Between Pads	G	0.20		

Notes:

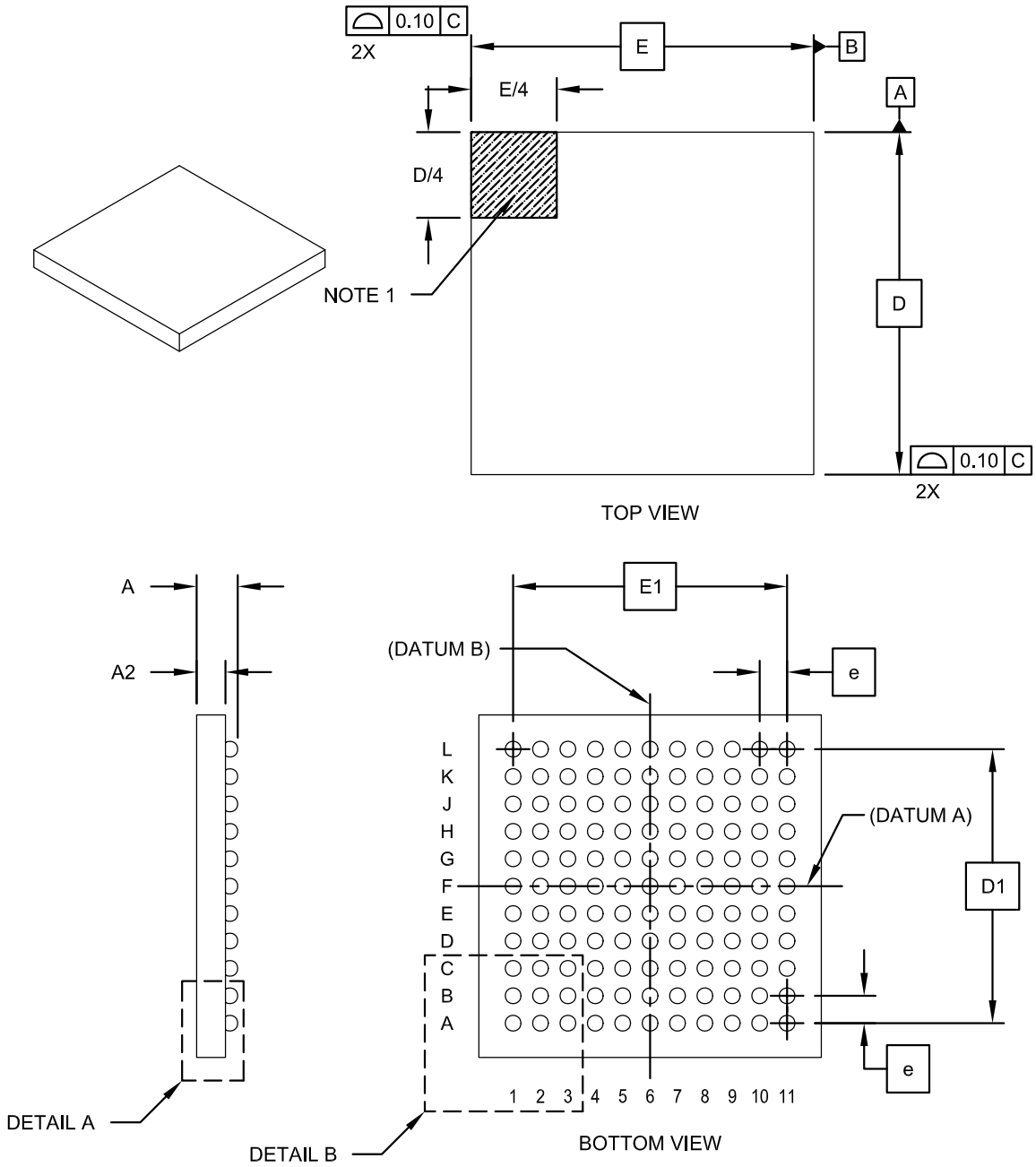
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2100A

121-Lead Plastic Thin Profile Ball Grid Array (BG) - 10x10x1.10 mm Body [XBGA]

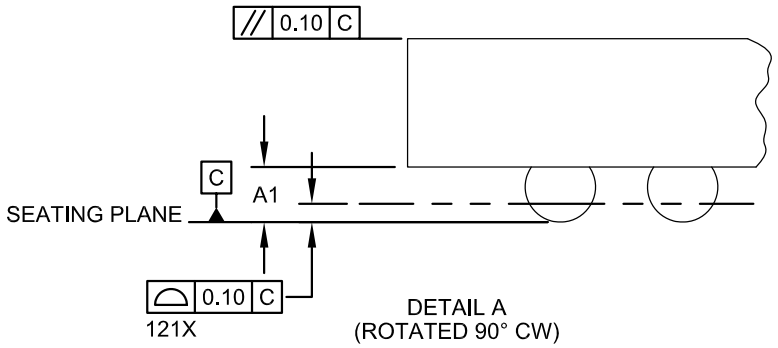
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging

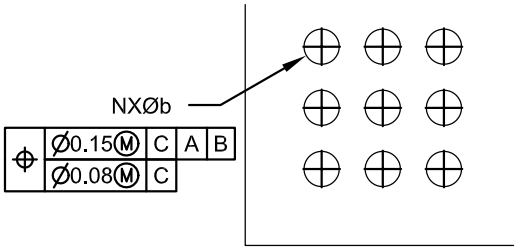


Microchip Technology Drawing C04-148A Sheet 1 of 2

121-Lead Plastic Thin Profile Ball Grid Array (BG) - 10x10x1.10 mm Body [XBGA]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging





DETAIL B

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Contacts	N		121	
Contact Pitch	e		0.80 BSC	
Overall Height	A	1.00	1.10	1.20
Standoff	A1	0.25	0.30	0.35
Molded Package Thickness	A2	0.55	0.60	0.65
Overall Width	E		10.00 BSC	
Array Width	E1		8.00 BSC	
Overall Length	D		10.00 BSC	
Array Length	D1		8.00 BSC	
Contact Diameter	b		0.40 TYP	

Notes:
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Dimensioning and tolerancing per ASME Y14.5M.
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
   REF: Reference Dimension, usually without tolerance, for information purposes only.
3. The outer rows and colums of balls are located with respect to datums A and B.

Microchip Technology Drawing C04-148 Rev A Sheet 2 of 2

APPENDIX A: REVISION HISTORY

Revision A (February 2010)

Original data sheet for the PIC24FJ256DA210 family of devices.

Revision B (May 2010)

Minor changes throughout text and the values in **Section 30.0 "Electrical Characteristics"** were updated.

NOTES:

INDEX

A

A/D Conversion
10-Bit High-Speed A/D Converter 325
A/D Converter 325
Analog Input Model 333
Transfer Function 333
AC Characteristics
ADC Conversion Timing 385
CLKO and I/O Timing 383
Internal RC Accuracy 382
Alternate Interrupt Vector Table (AIVT) 93
Alternative Master
EPMP 273
Assembler
MPASM Assembler 360

B

Block Diagram
CRC 297
Block Diagrams
10-Bit High-Speed A/D Converter 326
16-Bit Asynchronous Timer3 and Timer5 193
16-Bit Synchronous Timer2 and Timer4 193
16-Bit Timer1 Module 189
32-Bit Timer2/3 and Timer4/5 192
96 MHz PLL 150
Accessing Program Space Using Table Operations 77
Addressing for Table Registers 81
BDT Mapping for Endpoint Buffering Modes 244
CALL Stack Frame 75
Comparator Voltage Reference 341
CPU Programmer's Model 41
CRC Shift Engine Detail 297
CTMU Connections and Internal Configuration for Capacitance Measurement 343
CTMU Typical Connections and Internal Configuration for Pulse Delay Generation 344
CTMU Typical Connections and Internal Configuration for Time Measurement 344
Data Access From Program Space Address Generation 76
Graphics Module Overview 305
I^2C Module 224
Individual Comparator Configurations, CREF = 0 336
Individual Comparator Configurations, CREF = 1 and CVREFP = 0 337
Individual Comparator ConfigurationS, CREF = 1 and CVREFP = 1 337
Input Capture 197
On-Chip Regulator Connections 354
Output Compare (16-Bit Mode) 202
Output Compare (Double-Buffered, 16-Bit PWM Mode) 204
PCI24FJ256DA210 Family (General) 20
PIC24F CPU Core 40
PSV Operation (Higher Word) 79
PSV Operation (Lower Word) 79
Reset System 87
RTCC 285
Shared I/O Port Structure 157
SPI Master, Frame Master Connection 220
SPI Master, Frame Slave Connection 220
SPI Master/Slave Connection (Enhanced Buffer Modes) 219
SPI Master/Slave Connection (Standard Mode) 219
SPI Slave, Frame Master Connection 220
SPI Slave, Frame Slave Connection 220
SPIx Module (Enhanced Mode) 213
SPIx Module (Standard Mode) 212
System Clock 141
Triple Comparator Module 335
UART (Simplified) 231
USB OTG
Device Mode Power Modes 241
USB OTG Interrupt Funnel 248
USB OTG Module 240
Watchdog Timer (WDT) 356

C

C Compilers
MPLAB C18 360
Charge Time Measurement Unit (CTMU) 343
Key Features 343
Charge Time Measurement Unit. *See* CTMU.
Code Examples
Basic Sequence for Clock Switching in Assembly 149
Configuring UART1 I/O Input/Output Functions (PPS) 168
EDS Read From Program Memory in Assembly 80
EDS Read in Assembly 72
EDS Write in Assembly 73
Erasing a Program Memory Block (Assembly) 84
I/O Port Read/Write in 'C' 163
I/O Port Read/Write in Assembly 163
Initiating a Programming Sequence 85
PWRSAV Instruction Syntax 155
Setting the RTCWREN Bit 286
Single-Word Flash Programming 86
Single-Word Flash Programming ('C' Language) 86
Code Protection 357
Code Segment Protection 357
Configuration Options 358
Configuration Protection 358
Comparator Voltage Reference 341
Configuring 341
Configuration Bits 347
Core Features 15
CPU
Arithmetic Logic Unit (ALU) 43
Control Registers 42
Core Registers 40
Programmer's Model 39
CRC
32-Bit Programmable Cyclic Redundancy Check 297
Polynomials 298
Setup Examples for 16 and 32-Bit Polynomials 298
User Interface 298

CTMU
Measuring Capacitance 343
Measuring Time 344
Pulse Delay and Generation 344
Customer Change Notification Service 405
Customer Notification Service 405
Customer Support 405

D

Data Memory
Address Space 47
Memory Map 48
Near Data Space 49
SFR Space 49
Software Stack 75
Space Organization, Alignment 49
DC Characteristics
I/O Pin Input Specifications 377
I/O Pin Output Specifications 378
Program Memory 378
Development Support 359
Device Features
100/121--Pin 19
64-Pin 18
Doze Mode 156

E

EDS 273
Electrical Characteristics
A/D Specifications 384
Absolute Maximum Ratings 371
Capacitive Loading on Output Pin 380
External Clock Timing 381
Idle Current 375
Load Conditions and Requirements for Specifications 379
Operating Current 374
PLL Clock Timing Specifications 381
Power-Down Current 376
RC Oscillator Start-up Time 382
Reset and Brown-out Reset Requirements 382
Temperature and Voltage Specifications 373
Thermal Conditions 372
V/F Graph 372
Voltage Regulator Specifications 379
Enhanced Parallel Master Port. *See* EPMP. 273
ENVREG Pin 354
EPMP 273
Alternative Master 273
Key Features 273
Master Port Pins 274
Equations
16-Bit, 32-Bit CRC Polynomials 298
A/D Conversion Clock Period 332
Baud Rate Reload Calculation 225
Calculating the PWM Period 204
Calculation for Maximum PWM Resolution 205
Estimating USB Transceiver Current Consumption 243
Relationship Between Device and SPI Clock Speed 221
RTCC Calibration 294
UART Baud Rate with BRGH = 0 232
UART Baud Rate with BRGH = 1 232
Errata 14

F

Flash Configuration Words 46, 347
Flash Program Memory 81
and Table Instructions 81
Enhanced ICSP Operation 82
JTAG Operation 82
Programming Algorithm 84
RTSP Operation 82
Single-Word Programming 86

G

Graphics Controller (GFX) 305
Key Features 305
Graphics Controller Module (GFX) 305
Graphics Display Module
Display Clock (GCLK) Source 324
Display Configuration 324
Memory Locations 324
Memory Requirements 324
Module Registers 306
Graphics Display Module (GFX) 305

I

I/O Ports
Analog Port Pins Configuration 158
Analog/Digital Function of an I/O Pin 158
Input Change Notification 163
Open-Drain Configuration 158
Parallel (PIO) 157
Peripheral Pin Select 164
Pull-ups and Pull-downs 163
Selectable Input Sources 165
I^2C
Clock Rates 225
Reserved Addresses 225
Setting Baud Rate as Bus Master 225
Slave Address Masking 225
Idle Mode 156
Input Capture
32-Bit Mode 198
Operations 198
Synchronous and Trigger Modes 197
Input Capture with Dedicated Timers 197
Input Voltage Levels for Port or Pin
Tolerated Description Input 158
Instruction Set
Overview 365
Summary 363
Instruction-Based Power-Saving Modes 155, 156
Interfacing Program and Data Spaces 75
Inter-Integrated Circuit. *See* I^2C. 223
Internet Address 405
Interrupt Vector Table (IVT) 93
Interrupts
Control and Status Registers 96
Implemented Vectors 95
Reset Sequence 93
Setup and Service Procedures 140
Trap Vectors 94
Vector Table 94

J

JTAG Interface 358

K

Key Features 347
- CTMU 343
- EPMP 273
- RTCC 285

M

Memory Organization 45
Microchip Internet Web Site 405
MPLAB ASM30 Assembler, Linker, Librarian 360
MPLAB Integrated Development
- Environment Software 359

MPLAB PM3 Device Programmer 362
MPLAB REAL ICE In-Circuit Emulator System 361
MPLINK Object Linker/MPLIB Object Librarian 360

N

Near Data Space 49

O

Oscillator Configuration
- 96 MHz PLL 149
- Clock Selection 142
- Clock Switching 148
  - Sequence 148
- CPU Clocking Scheme 142
- Display Clock Frequency Division 152
- Initial Configuration on POR 142
- USB Operations 151

Output Compare
- 32-Bit Mode 201
- Synchronous and Trigger Modes 201

Output Compare with Dedicated Timers 201

P

Packaging 387
- Details 388
- Marking 387

Peripheral Enable Bits 156
Peripheral Module Disable Bits 156
Peripheral Pin Select (PPS) 164
- Available Peripherals and Pins 164
- Configuration Control 167
- Considerations for Use 168
- Input Mapping 164
- Mapping Exceptions 167
- Output Mapping 166
- Peripheral Priority 164
- Registers 169

Pin Descriptions
- 100-Pin Devices 8
- 121 (BGA)-Pin Devices 11
- 64-Pin Devices 6

Pin Diagrams
- 100-Pin TQFP 7
- 121-Pin BGA 10
- 64-Pin TQFP/QFN 5

Pinout Descriptions 21
POR
- and On-Chip Voltage Regulator 354

Power-Saving
- Clock Frequency and Clock Switching 155
- Features 155
- Instruction-Based Modes 155

Power-up Requirements 355
Product Identification System 407
Program Memory
- Access Using Table Instructions 77
- Address Construction 75
- Address Space 45
- Flash Configuration Words 46
- Memory Maps 45
- Organization 46
- Reading From Program Memory Using EDS 78

Pulse-Width Modulation (PWM) Mode 203
Pulse-Width Modulation. *See* PWM.
PWM
- Duty Cycle and Period 204

R

Reader Response 406
Reference Clock Output 152
Register Maps
- A/D Converter 61
- ANCFG 64
- ANSEL 64
- Comparators 66
- CPU Core 50
- CRC 66
- CTMU 62
- Enhanced Parallel Master/Slave Port 65
- Graphics 69
- I^2C™ 56
- ICN 51
- Input Capture 54
- Interrupt Controller 52
- NVM 70
- Output Compare 55
- Pad Configuration 60
- Peripheral Pin Select 67
- PMD 70
- PORTA 58
- PORTB 58
- PORTC 59
- PORTD 59
- PORTE 59
- PORTF 60
- PORTG 60
- RTCC 65
- SPI 58
- System 70
- Timers 53
- UART 57
- USB OTG 63

Registers
- AD1CHS (A/D Input Select) 330
- AD1CON1 (A/D Control 1) 327
- AD1CON2 (A/D Control 2) 328
- AD1CON3 (A/D Control 3) 329
- AD1CSSH (A/D Input Scan Select, High) 332
- AD1CSSL (A/D Input Scan Select, Low) 331
- ALCFGRPT (Alarm Configuration) 289
- ALMINSEC (Alarm Minutes and Seconds Value) 293
- ALMTHDY (Alarm Month and Day Value) 292
- ALWDHR (Alarm Weekday and Hours Value) 293
- ANCFG (A/D Band Gap Reference Configuration) 331
- ANSA (PORTA Analog Function Selection) 159
- ANSB (PORTB Analog Function Selection) 160
- ANSC (PORTC Analog Function Selection) 160
- ANSD (PORTD Analog Function Selection) 161
- ANSE (PORTE Analog Function Selection) 161

ANSF (PORTF Analog Function Selection) 162
ANSG (PORTG Analog Function Selection) 162
BDnSTAT Prototype (Buffer Descriptor n Status, CPU Mode) 247
BDnSTAT Prototype (Buffer Descriptor n Status, USB Mode) 246
CLKDIV (Clock Divider) 145
CLKDIV2 (Clock Divider 2) 147
CMSTAT (Comparator Status) 339
CMxCON (Comparator x Control) 338
CORCON (CPU Core Control) 43, 98
CRCCON1 (CRC Control 1) 300
CRCCON2 (CRC Control 2) 301
CRCDATH (CRC Data High) 302
CRCDATL (CRC Data Low) 302
CRCWDATH (CRC Shift High) 303
CRCWDATL (CRC Shift Low) 303
CRCXORH (CRC XOR High) 302
CRCXORL (CRC XOR Polynomial, Low Byte) 301
CTMUCON (CTMU Control) 345
CTMUICON (CTMU Current Control) 346
CVRCON (Comparator Voltage Reference Control) 342
CW1 (Flash Configuration Word 1) 348
CW2 (Flash Configuration Word 2) 350
CW3 (Flash Configuration Word 3) 351
CW4 (Flash Configuration Word 4) 352
DEVID (Device ID) 353
DEVREV (Device Revision) 354
G1ACTDA (Active Display Area) 317
G1CHRX (Character X-Coordinate Print Position) 321
G1CHRY (Character Y-Coordinate Print Position) 322
G1CLUT (Color Look-up Table Control) 319
G1CLUTRD (Color Look-up Table Memory Read Data) 320
G1CLUTWR (Color Look-up Table Memory Write Data) 320
G1CMDH (GPU Command High) 306
G1CMDL (GPU Command Low) 306
G1CON1 (Display Control 1) 307
G1CON2 (Display Control 2) 308
G1CON3 (Display Control 3) 309
G1DBEN (Data I/O Pad Enable) 323
G1DBLCON (Display Blanking Control) 318
G1DPADRH (Display Buffer Start Address High) 315
G1DPADRL (Display Buffer Start Address Low) 315
G1DPDPH (Display Buffer Height) 316
G1DPHT (Display Total Height) 316
G1DPW (Display Buffer Width) 315
G1DPWT (Display Total Width) 316
G1HSYNC (Horizontal Synchronization Control) 317
G1IE (GFX Interrupt Enable) 311
G1IPU (Inflate Processor Status) 322
G1IR (GFX Interrupt Status) 312
G1MRGN (Interrupt Advance) 321
G1PUH (GPU Work Area Height) 314
G1PUW (GPU Work Area Width) 314
G1STAT (Graphics Control Status) 310
G1VSYNC (Vertical Synchronization Control) 318
G1W1ADRH (GPU Work Area 1 Start Address High) 313
G1W1ADRL (GPU Work Area 1 Start Address Low) 313
G1W2ADRH (GPU Work Area 2 Start Address High 314
G1W2ADRL (GPU Work Area 2 Start Address Low) 313
I2CxCON (I2Cx Control) 226
I2CxMSK (I2Cx Slave Mode Address Mask) 230
I2CxSTAT (I2Cx Status) 228
ICxCON1 (Input Capture x Control 1) 199
ICxCON2 (Input Capture x Control 2) 200
IEC0 (Interrupt Enable Control 0) 109
IEC1 (Interrupt Enable Control 1) 110
IEC2 (Interrupt Enable Control 2) 112
IEC3 (Interrupt Enable Control 3) 113
IEC4 (Interrupt Enable Control 4) 114
IEC5 (Interrupt Enable Control 5) 115
IEC6 (Interrupt Enable Control 6) 116
IFS0 (Interrupt Flag Status 0) 101
IFS1 (Interrupt Flag Status 1) 102
IFS2 (Interrupt Flag Status 2) 103
IFS3 (Interrupt Flag Status 3) 105
IFS4 (Interrupt Flag Status 4) 106
IFS5 (Interrupt Flag Status 5) 107
IFS6 (Interrupt Flag Status 6) 108
INTCON1 (Interrupt Control 1) 99
INTCON2 (Interrupt Control 2) 100
INTTREG (Interrupt Controller Test 139
IPC0 (Interrupt Priority Control 0) 117
IPC1 (Interrupt Priority Control 1) 118
IPC10 (Interrupt Priority Control 10) 127
IPC11 (Interrupt Priority Control 11) 128
IPC12 (Interrupt Priority Control 12) 129
IPC13 (Interrupt Priority Control 13) 130
IPC15 (Interrupt Priority Control 15) 131
IPC16 (Interrupt Priority Control 16) 132
IPC18 (Interrupt Priority Control 18) 133
IPC19 (Interrupt Priority Control 19) 133
IPC2 (Interrupt Priority Control 2) 119
IPC20 (Interrupt Priority Control 20) 134
IPC21 (Interrupt Priority Control 21) 135
IPC22 (Interrupt Priority Control 22) 136
IPC23 (Interrupt Priority Control 23) 137
IPC25 (Interrupt Priority Control 25) 138
IPC3 (Interrupt Priority Control 3) 120
IPC4 (Interrupt Priority Control 4) 121
IPC5 (Interrupt Priority Control 5) 122
IPC6 (Interrupt Priority Control 6) 123
IPC7 (Interrupt Priority Control 7) 124
IPC8 (Interrupt Priority Control 8) 125
IPC9 (Interrupt Priority Control 9) 126
IPCn (Interrupt Priority Control 0-23) 137
MINSEC (RTCC Minutes and Seconds Value) 291
MTHDY (RTCC Month and Day Value) 290
NVMCON (Flash Memory Control) 83
OCxCON1 (Output Compare x Control 1) 206
OCxCON2 (Output Compare x Control 2) 208
OSCCON (Oscillator Control) 143

OSCTUN (FRC Oscillator Tune) ... 146
PADCFG1 (Pad Configuration Control) ... 283, 288
PMCON1 (EPMP Control 1) ... 275
PMCON2 (EPMP Control 2) ... 276
PMCON3 (EPMP Control 3) ... 277
PMCON4 (EPMP Control 4) ... 278
PMCSxBS (Chip Select x Base Address) ... 280
PMCSxCF (Chip Select x Configuration) ... 279
PMCSxMD (Chip Select x Mode) ... 281
PMSTAT (EPMP Status, Slave Mode) ... 282
RCFGCAL (RTCC Calibration and Configuration) ... 287
RCON (Reset Control) ... 88
REFOCON (Reference Oscillator Control) ... 153
RPINRn (PPS Input) ... 169–179
RPORn (PPS Output) ... 180–187
SPIxCON1 (SPIx Control 1) ... 216
SPIxCON2 (SPIx Control 2) ... 218
SPIxSTAT (SPIx Status and Control) ... 214
SR (ALU STATUS) ... 42, 97
T1CON (Timer1 Control) ... 190
TxCON (Timer2 and Timer4 Control) ... 194
TyCON (Timer3 and Timer5 Control) ... 195
U1ADDR (USB Address) ... 261
U1CNFG1 (USB Configuration 1) ... 262
U1CNFG2 (USB Configuration 2) ... 263
U1CON (USB Control, Device Mode) ... 259
U1CON (USB Control, Host Mode) ... 260
U1EIE (USB Error Interrupt Enable) ... 270
U1EIR (USB Error Interrupt Status) ... 269
U1EPn (USB Endpoint n Control) ... 271
U1IE (USB Interrupt Enable) ... 268
U1IR (USB Interrupt Status, Device Mode) ... 266
U1IR (USB Interrupt Status, Host Mode) ... 267
U1OTGCON (USB OTG Control) ... 256
U1OTGIE (USB OTG Interrupt Enable, Host Mode) ... 265
U1OTGIR (USB OTG Interrupt Status, Host Mode) ... 264
U1OTGSTAT (USB OTG Status) ... 255
U1PWMCON USB (VBUS PWM Generator Control) ... 272
U1PWRC (USB Power Control) ... 257
U1SOF (USB OTG Start-Of-Token Threshold, Host Mode) ... 262
U1STAT (USB Status) ... 258
U1TOK (USB Token, Host Mode) ... 261
UxMODE (UARTx Mode) ... 234
UxSTA (UARTx Status and Control) ... 236
WKDYHR (RTCC Weekday and Hours Value) ... 291
YEAR (RTCC Year Value) ... 290
Resets
BOR (Brown-out Reset) ... 87
Clock Source Selection ... 90
CM (Configuration Mismatch Reset) ... 87
Delay Times ... 91
Device Times ... 90
IOPUWR (Illegal Opcode Reset) ... 87
MCLR (Pin Reset) ... 87
POR (Power-on Reset) ... 87
RCON Flags Operation ... 89
SFR States ... 90
SWR (RESET Instruction) ... 87
TRAPR (Trap Conflict Reset) ... 87
UWR (Uninitialized W Register Reset) ... 87
WDT (Watchdog Timer Reset) ... 87
Revision History ... 397
RTCC
Alarm Configuration ... 294
Calibration ... 294
Key Features ... 285
Register Mapping ... 286

S

Selective Peripheral Power Control ... 156
Serial Peripheral Interface (SPI) ... 211
Serial Peripheral Interface. *See* SPI.
SFR Space ... 49
Sleep Mode ... 155
Software Simulator (MPLAB SIM) ... 361
Software Stack ... 75
Special Features ... 16
SPI ... 211

T

Timer1 ... 189
Timer2/3 and Timer4/5 ... 191
Timing Diagrams
CLKO and I/O Timing ... 383
External Clock ... 380
Triple Comparator ... 335
Triple Comparator Module ... 335

U

UART ... 231
Baud Rate Generator (BRG) ... 232
IrDA Support ... 233
Operation of UxCTS and UxRTS Pins ... 233
Receiving in 8-Bit or 9-Bit Data Mode ... 233
Transmitting
Break and Sync Sequence ... 233
in 8-Bit Data Mode ... 233
Transmitting in 9-Bit Data Mode ... 233
Universal Asynchronous Receiver Transmitter. *See* UART.
Universal Serial Bus
Buffer Descriptors
Assignment in Different Buffering Modes ... 245
Interrupts
and USB Transactions ... 249
Universal Serial Bus. *See* USB OTG.
USB On-The-Go (OTG) ... 16
USB OTG ... 239
Buffer Descriptors and BDT ... 244
Device Mode Operation ... 249
DMA Interface ... 245
Hardware
Calculating
Transceiver Power Requirements ... 243
Hardware Configuration ... 241
Device Mode ... 241
External Interface ... 243
Host and OTG Modes ... 242
VBUS Voltage Generation ... 243
Host Mode Operation ... 250
Interrupts ... 248
Operation ... 252
Registers ... 254
VBUS Voltage Generation ... 243

V

Voltage Regulator (On-Chip) 354
and BOR 355
Low Voltage Detection 354
Standby Mode 355

W

Watchdog Timer (WDT) 355
Control Register 356
Windowed Operation 356
WWW Address 405
WWW, On-Line Support 14

THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com, click on Customer Change Notification and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support
- Development Systems Information Line

Customers should contact their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://support.microchip.com

READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (480) 792-4150.

Please list the following information, and use this outline to provide us with your comments about this document.

To: Technical Publications Manager Total Pages Sent _____

RE: Reader Response

From: Name _____

Company _____

Address _____

City / State / ZIP / Country _____

Telephone: (_____) _____ - _____ FAX: (_____) _____ - _____

Application (optional):

Would you like a reply? ___Y ___N

Device: PIC24FJ256DA210 Family Literature Number: DS39969B

Questions:

1. What are the best features of this document?

2. How does this document meet your hardware and software development needs?

3. Do you find the organization of this document easy to follow? If not, why?

4. What additions to the document do you think would enhance the structure and subject?

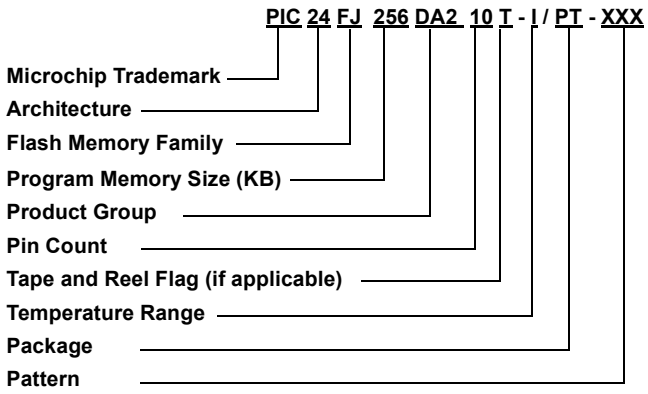
5. What deletions from the document could be made without affecting the overall usefulness?

6. Is there any incorrect or misleading information (what and where)?

7. How would you improve this document?

PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.



Architecture	24	= 16-bit modified Harvard without DSP
Flash Memory Family	FJ	= Flash program memory
Product Group	DA2	= General purpose microcontrollers with Graphics Controller and USB On-The-Go
Pin Count	06 10	= 64-pin = 100-pin (TQFP)/121-pin (BGA)
Temperature Range	I	= -40°C to +85°C (Industrial)
Package	PT = 100-lead (12x12x1 mm) TQFP (Thin Quad Flatpack) PT = 64-lead, TQFP (Thin Quad Flatpack) MR = 64-lead (9x9x0.9 mm) QFN (Quad Flatpack, No Lead) BG = 121-pin BGA package	
Pattern	Three-digit QTP, SQTP, Code or Special Requirements (blank otherwise) ES = Engineering Sample	

Examples:

a) PIC24FJ128DA206-I/PT:
PIC24F device with Graphics Controller and USB On-The-Go, 128-KB program memory, 96-KB data memory, 64-pin, Industrial temp., TQFP package.

b) PIC24FJ256DA110-I/PT:
PIC24F device with Graphics Controller and USB On-The-Go, 256-KB program memory, 24-KB data memory, 100-pin, Industrial temp., TQFP package.

c) PIC24FJ256DA210-I/BG:
PIC24F device with Graphics Controller and USB On-The-Go, 256-KB program memory, 96-KB data memory, 121-pin, Industrial temp., BGA package.

MICROCHIP

WORLDWIDE SALES AND SERVICE

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://support.microchip.com
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Kokomo
Kokomo, IN
Tel: 765-864-8360
Fax: 765-864-8387

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

Santa Clara
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

Toronto
Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8528-2100
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Hong Kong SAR
Tel: 852-2401-1200
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

Japan - Yokohama
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-6578-300
Fax: 886-3-6578-370

Taiwan - Kaohsiung
Tel: 886-7-536-4818
Fax: 886-7-536-4803

Taiwan - Taipei
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

01/05/10