PIC24FV32KA304 FAMILY

20/28/44/48-Pin, General Purpose, 16-Bit Flash Microcontrollers with XLP Technology

Power Management Modes

- Run – CPU, Flash, SRAM and Peripherals On
- Doze – CPU Clock Runs Slower than Peripherals
- Idle – CPU Off, Flash, SRAM and Peripherals On
- Sleep – CPU, Flash and Peripherals Off, and SRAM On
- Deep Sleep – CPU, Flash, SRAM and Most Peripherals Off; Multiple Autonomous Wake-up Sources
- Low-Power Consumption:
  - Run mode currents down to 8 μA, typical
  - Idle mode currents down to 2.2 μA, typical
  - Deep Sleep mode currents down to 20 nA, typical
  - Real-Time Clock/Calendar currents down to 700 nA, 32 kHz, 1.8V
  - Watchdog Timer is 500 nA, 1.8V typical

High-Performance CPU

- Modified Harvard Architecture
- Up to 16 MIPS Operation @ 32 MHz
- 8 MHz Internal Oscillator with 4x PLL Option and Multiple Divide Options
- 17-Bit by 17-Bit Single-Cycle Hardware Multiplier
- 32-Bit by 16-Bit Hardware Divider, 16-Bit x 16-Bit Working Register Array
- C Compiler Optimized Instruction Set Architecture

Peripheral Features

- Hardware Real-Time Clock and Calendar (RTCC):
  - Provides clock, calendar and alarm functions
  - Can run in Deep Sleep mode
  - Can use 50/60 Hz power line input as clock source
- Programmable 32-Bit Cyclic Redundancy Check (CRC)
- Multiple Serial Communication modules:
  - Two 3/4-wire SPI modules
  - Two I^{2}C modules with multi-master/slave support
  - Two UART modules, supporting RS-485, RS-232, LIN/J2602, IrDA®
- Five 16-Bit Timers/Counters with Programmable Prescaler:
  - Can be paired as 32-bit timers/counters
- Three 16-Bit Capture Inputs with Dedicated Timers
- Three 16-Bit Compare/PWM Outputs with Dedicated Timers
- Configurable Open-Drain Outputs on Digital I/O Pins
- Up to Three External Interrupt Sources

Analog Features

- 12-Bit, Up to 16-Channel Analog-to-Digital Converter:
  - 100 ksps conversion rate
  - Conversion available during Sleep and Idle
  - Auto-sampling, timer-based option for Sleep and Idle modes
  - Wake on auto-compare option
- Dual Rail-to-Rail Analog Comparators with Programmable Input/Output Configuration
- On-Chip Voltage Reference
- Internal Temperature Sensor
- Charge Time Measurement Unit (CTMU):
  - Used for capacitance sensing, 16 channels
  - Time measurement, down to 200 ps resolution
  - Delay/pulse generation, down to 1 ns resolution

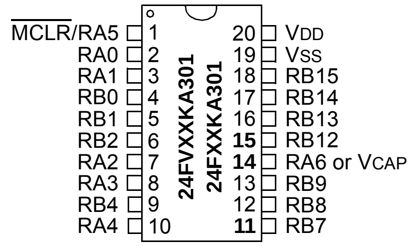
Special Microcontroller Features

- Wide Operating Voltage Range:
  - 1.8V to 3.6V (PIC24F devices)
  - 2.0V to 5.5V (PIC24FV devices)
- Low-Power Wake-up Sources and Supervisors:
  - Ultra Low-Power Wake-up (ULPWU) for Sleep/Deep Sleep
  - Low-Power Watchdog Timer (DSWDT) for Deep Sleep
  - Extreme Low-Power Brown-out Reset (DSBOR) for Deep Sleep, LPBOR for all other modes
- System Frequency Range Declaration bits:
  - Declaring the frequency range optimizes the current consumption.
- Standard Watchdog Timer (WDT) with On-Chip, Low-Power RC Oscillator for Reliable Operation
- Programmable High/Low-Voltage Detect (HLVD)
- Standard Brown-out Reset (BOR) with 3 Programmable Trip Points that can be Disabled in Sleep
- High-Current Sink/Source (18 mA/18 mA) on All I/O Pins
- Flash Program Memory:
  - Erase/write cycles: 10,000 minimum
  - 40 years' data retention minimum
- Data EEPROM:
  - Erase/write cycles: 100,000 minimum
  - 40 years' data retention minimum
- Fail-Safe Clock Monitor (FSCM)
- Programmable Reference Clock Output
- Self-Programmable under Software Control
- In-Circuit Serial Programming™ (ICSP™) and In-Circuit Debug (ICD) via 2 Pins

PIC24F Device	Pins	Memory			Timers 16-Bit	Capture Input	Compare/PWM Output	UART w/ IrDA®	SPI	I^2C	12-Bit A/D (ch)	Comparators	CTMU (ch)	RTCC
		Flash Program (bytes)	SRAM (bytes)	EE Data (bytes)										
PIC24FV16KA301/ PIC24F16KA301	20	16K	2K	512	5	3	3	2	2	2	12	3	12	Y
PIC24FV32KA301/ PIC24F32KA301	20	32K	2K	512	5	3	3	2	2	2	12	3	12	Y
PIC24FV16KA302/ PIC24F16KA302	28	16K	2K	512	5	3	3	2	2	2	13	3	13	Y
PIC24FV32KA302/ PIC24F32KA302	28	32K	2K	512	5	3	3	2	2	2	13	3	13	Y
PIC24FV16KA304/ PIC24F16KA304	44	16K	2K	512	5	3	3	2	2	2	16	3	16	Y
PIC24FV32KA304/ PIC24F32KA304	44	32K	2K	512	5	3	3	2	2	2	16	3	16	Y

Pin Diagrams

20-Pin SPDIP/SSOP/SOIC(1)



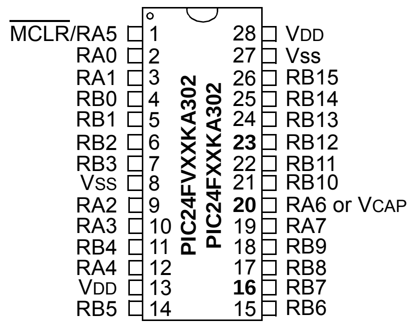
Pin	Pin Features	
	PIC24FVXXKA301	**PIC24FXXKA301**
1	$\overline{\text{MCLR}}$/VPP/RA5	$\overline{\text{MCLR}}$/VPP/RA5
2	PGEC2/VREF+/CVREF+/AN0/C3INC/SCK2/CN2/RA0	PGEC2/VREF+/CVREF+/AN0/C3INC/SCK2/CN2/RA0
3	PGED2/CVREF-/VREF-/AN1/SDO2/CN3/RA1	PGED2/CVREF-/VREF-/AN1/SDO2/CN3/RA1
4	PGED1/AN2/ULPWU/CTCMP/C1IND/C2INB/C3IND/U2TX/SDI2/OC2/CN4/RB0	PGED1/AN2/ULPWU/CTCMP/C1IND/C2INB/C3IND/U2TX/SDI2/OC2/CN4/RB0
5	PGEC1/AN3/C1INC/C2INA/U2RX/OC3/CTED12/CN5/RB1	PGEC1/AN3/C1INC/C2INA/U2RX/OC3/CTED12/CN5/RB1
6	AN4/SDA2/T5CK/T4CK/U1RX/CTED13/CN6/RB2	AN4/SDA2/T5CK/T4CK/U1RX/CTED13/CN6/RB2
7	OSCI/AN13/C1INB/C2IND/CLKI/CN30/RA2	OSCI/AN13/C1INB/C2IND/CLKI/CN30/RA2
8	OSCO/AN14/C1INA/C2INC/CLKO/CN29/RA3	OSCO/AN14/C1INA/C2INC/CLKO/CN29/RA3
9	PGED3/SOSCI/AN15/$\overline{\text{U2RTS}}$/CN1/RB4	PGED3/SOSCI/AN15/$\overline{\text{U2RTS}}$/CN1/RB4
10	PGEC3/SOSCO/SCLKI/$\overline{\text{U2CTS}}$/CN0/RA4	PGEC3/SOSCO/SCLKI/$\overline{\text{U2CTS}}$/CN0/RA4
11	**U1TX/C2OUT/OC1/IC1/CTED1/INT0/CN23/RB7**	**U1TX/INT0/CN23/RB7**
12	SCL1/$\overline{\text{U1CTS}}$/C3OUT/CTED10/CN22/RB8	SCL1/$\overline{\text{U1CTS}}$/C3OUT/CTED10/CN22/RB8
13	SDA1/T1CK/$\overline{\text{U1RTS}}$/IC2/CTED4/CN21/RB9	SDA1/T1CK/$\overline{\text{U1RTS}}$/IC2/CTED4/CN21/RB9
14	**VCAP**	**C2OUT/OC1/IC1/CTED1/INT2/CN8/RA6**
15	**AN12/HLVDIN/SCK1/$\overline{\text{SS2}}$/IC3/CTED2/INT2/CN14/RB12**	**AN12/HLVDIN/SCK1/$\overline{\text{SS2}}$/IC3/CTED2/CN14/RB12**
16	AN11/SDO1/OCFB/CTPLS/CN13/RB13	AN11/SDO1/OCFB/CTPLS/CN13/RB13
17	CVREF/AN10/C3INB/RTCC/SDI1/C1OUT/OCFA/CTED5/INT1/CN12/RB14	CVREF/AN10/C3INB/RTCC/SDI1/C1OUT/OCFA/CTED5/INT1/CN12/RB14
18	AN9/C3INA/SCL2/T3CK/T2CK/REFO/$\overline{\text{SS1}}$/CTED6/CN11/RB15	AN9/C3INA/SCL2/T3CK/T2CK/REFO/$\overline{\text{SS1}}$/CTED6/CN11/RB15
19	VSS/AVSS	VSS/AVSS
20	VDD/AVDD	VDD/AVDD

Legend: Pin numbers in **bold** indicate pin function differences between PIC24FV and PIC24F devices.

Note 1: PIC24F32KA304 device pins have a maximum voltage of 3.6V and are not 5V tolerant.

Pin Diagrams

28-Pin SPDIP/SSOP/SOIC(2)



Pin	Pin Features	
	PIC24FVXXKA302	**PIC24FXXKA302**
1	$\overline{\text{MCLR}}$/VPP/RA5	$\overline{\text{MCLR}}$/VPP/RA5
2	VREF+/CVREF+/AN0/C3INC/CTED1/CN2/RA0	VREF+/CVREF+/AN0/C3INC/CTED1/CN2/RA0
3	CVREF-/VREF-/AN1/CN3/RA1	CVREF-/VREF-/AN1/CN3/RA1
4	PGED1/AN2/ULPWU/CTCMP/C1IND/C2INB/C3IND/U2TX/CN4/RB0	PGED1/AN2/ULPWU/CTCMP/C1IND/C2INB/C3IND/U2TX/CN4/RB0
5	PGEC1/AN3/C1INC/C2INA/U2RX/CTED12/CN5/RB1	PGEC1/AN3/C1INC/C2INA/U2RX/CN5/RB1
6	AN4/C1INB/C2IND/SDA2/T5CK/T4CK/U1RX/CTED13/CN6/RB2	AN4/C1INB/C2IND/SDA2/T5CK/T4CK/U1RX/CTED13/CN6/RB2
7	AN5/C1INA/C2INC/SCL2/CN7/RB3	AN5/C1INA/C2INC/SCL2/CN7/RB3
8	VSS	VSS
9	OSCI/AN13/CLKI/CN30/RA2	OSCI/AN13/CLKI/CN30/RA2
10	OSCO/AN14/CLKO/CN29/RA3	OSCO/AN14/CLKO/CN29/RA3
11	SOSCI/AN15/$\overline{\text{U2RTS}}$/CN1/RB4	SOSCI/AN15/$\overline{\text{U2RTS}}$/CN1/RB4
12	SOSCO/SCLKI/$\overline{\text{U2CTS}}$/CN0/RA4	SOSCO/SCLKI/$\overline{\text{U2CTS}}$/CN0/RA4
13	VDD	VDD
14	PGED3/ASDA(1)/SCK2/CN27/RB5	PGED3/ASDA(1)/SCK2/CN27/RB5
15	PGEC3/ASCL(1)/SDO2/CN24/RB6	PGEC3/ASCL(1)/SDO2/CN24/RB6
16	**U1TX/C2OUT/OC1/INT0/CN23/RB7**	**U1TX/INT0/CN23/RB7**
17	SCL1/$\overline{\text{U1CTS}}$/C3OUT/CTED10/CN22/RB8	SCL1/$\overline{\text{U1CTS}}$/C3OUT/CTED10/CN22/RB8
18	SDA1/T1CK/$\overline{\text{U1RTS}}$/IC2/CTED4/CN21/RB9	SDA1/T1CK/$\overline{\text{U1RTS}}$/IC2/CTED4/CN21/RB9
19	SDI2/IC1/CTED3/CN9/RA7	SDI2/IC1/CTED3/CN9/RA7
20	**VCAP**	**C2OUT/OC1/CTED1/INT2/CN8/RA6**
21	PGED2/SDI1/OC3/CTED11/CN16/RB10	PGED2/SDI1/OC3/CTED11/CN16/RB10
22	PGEC2/SCK1/OC2/CTED9/CN15/RB11	PGEC2/SCK1/OC2/CTED9/CN15/RB11
23	**AN12/HLVDIN/$\overline{\text{SS2}}$/IC3/CTED2/INT2/CN14/RB12**	**AN12/HLVDIN/$\overline{\text{SS2}}$/IC3/CTED2/CN14/RB12**
24	AN11/SDO1/OCFB/CTPLS/CN13/RB13	AN11/SDO1/OCFB/CTPLS/CN13/RB13
25	CVREF/AN10/C3INB/RTCC/C1OUT/OCFA/CTED5/**INT1/**CN12/RB14	CVREF/AN10/C3INB/RTCC/C1OUT/OCFA/CTED5/**INT1/**CN12/RB14
26	AN9/C3INA/T3CK/T2CK/REFO/$\overline{\text{SS1}}$/CTED6/CN11/RB15	AN9/C3INA/T3CK/T2CK/REFO/$\overline{\text{SS1}}$/CTED6/CN11/RB15
27	VSS/AVSS	VSS/AVSS
28	VDD/AVDD	VDD/AVDD

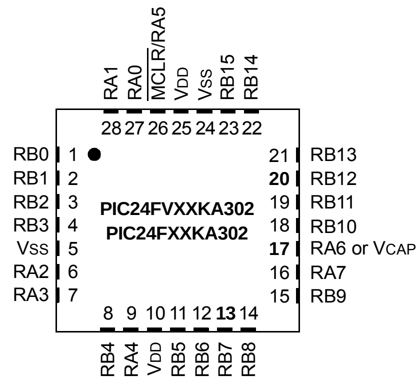
Legend: Pin numbers in **bold** indicate pin function differences between PIC24FV and PIC24F devices.

Note 1: Alternative multiplexing for SDA1 (ASDA1) and SCL1 (ASCL1) when the I2CSEL Configuration bit is set.

2: PIC24F32KA304 device pins have a maximum voltage of 3.6V and are not 5V tolerant.

Pin Diagrams

28-Pin QFN(1,3)



Pin	Pin Features	
	PIC24FVXXKA302	PIC24FXXKA302
1	PGED1/AN2/ULPWU/CTCMP/C1IND/C2INB/C3IND/U2TX/CN4/RB0	PGED1/AN2/ULPWU/CTCMP/C1IND/C2INB/C3IND/U2TX/CN4/RB0
2	PGEC1/AN3/C1INC/C2INA/U2RX/CTED12/CN5/RB1	PGEC1/AN3/C1INC/C2INA/U2RX/CTED12/CN5/RB1
3	AN4/C1INB/C2IND/SDA2/T5CK/T4CK/U1RX/CTED13/CN6/RB2	AN4/C1INB/C2IND/SDA2/T5CK/T4CK/U1RX/CTED13/CN6/RB2
4	AN5/C1INA/C2INC/SCL2/CN7/RB3	AN5/C1INA/C2INC/SCL2/CN7/RB3
5	VSS	VSS
6	OSCI/AN13/CLKI/CN30/RA2	OSCI/AN13/CLKI/CN30/RA2
7	OSCO/AN14/CLKO/CN29/RA3	OSCO/AN14/CLKO/CN29/RA3
8	SOSCI/AN15/U2RTS/CN1/RB4	SOSCI/AN15/U2RTS/CN1/RB4
9	SOSCO/SCLKI/U2CTS/CN0/RA4	SOSCO/SCLKI/U2CTS/CN0/RA4
10	VDD	VDD
11	PGED3/ASDA1(2)/SCK2/CN27/RB5	PGED3/ASDA1(2)/SCK2/CN27/RB5
12	PGEC3/ASCL1(2)/SDO2/CN24/RB6	PGEC3/ASCL1(2)/SDO2/CN24/RB6
13	**U1TX/C2OUT/OC1/INT0/CN23/RB7**	**U1TX/INT0/CN23/RB7**
14	SCL1/U1CTS/C3OUT/CTED10/CN22/RB8	SCL1/U1CTS/C3OUT/CTED10/CN22/RB8
15	SDA1/T1CK/U1RTS/IC2/CTED4/CN21/RB9	SDA1/T1CK/U1RTS/IC2/CTED4/CN21/RB9
16	SDI2/IC1/CTED3/CN9/RA7	SDI2/IC1/CTED3/CN9/RA7
17	**VCAP**	**C2OUT/OC1/CTED1/INT2/CN8/RA6**
18	PGED2/SDI1/OC3/CTED11/CN16/RB10	PGED2/SDI1/OC3/CTED11/CN16/RB10
19	PGEC2/SCK1/OC2/CTED9/CN15/RB11	PGEC2/SCK1/OC2/CTED9/CN15/RB11
20	**AN12/HLVDIN/SS2/IC3/CTED2/INT2/CN14/RB12**	**AN12/HLVDIN/SS2/IC3/CTED2/CN14/RB12**
21	AN11/SDO1/OCFB/CTPLS/CN13/RB13	AN11/SDO1/OCFB/CTPLS/CN13/RB13
22	CVREF/AN10/C3INB/RTCC/C1OUT/OCFA/CTED5/INT1/CN12/RB14	CVREF/AN10/C3INB/RTCC/C1OUT/OCFA/CTED5/INT1/CN12/RB14
23	AN9/C3INA/T3CK/T2CK/REFO/SS1/CTED6/CN11/RB15	AN9/C3INA/T3CK/T2CK/REFO/SS1/CTED6/CN11/RB15
24	VSS/AVSS	VSS/AVSS
25	VDD/AVDD	VDD/AVDD
26	MCLR/VPP/RA5	MCLR/VPP/RA5
27	VREF+/CVREF+/AN0/C3INC/CTED1/CN2/RA0	VREF+/CVREF+/AN0/C3INC/CN2/RA0
28	CVREF-/VREF-/AN1/CN3/RA1	CVREF-/VREF-/AN1/CN3/RA1

Legend: Pin numbers in **bold** indicate pin function differences between PIC24FV and PIC24F devices.

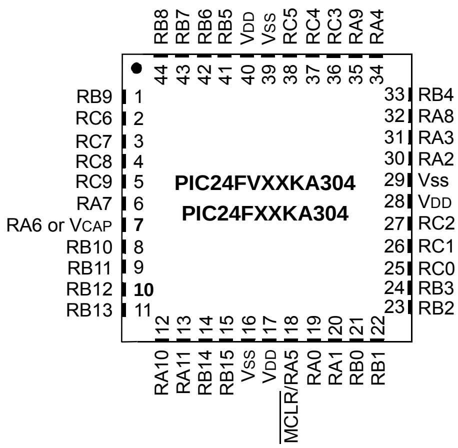
Note 1: Exposed pad on underside of device is connected to VSS.

2: Alternative multiplexing for SDA1 (ASDA1) and SCL1 (ASCL1) when the I2CSEL Configuration bit is set.

3: PIC24F32KA304 device pins have a maximum voltage of 3.6V and are not 5V tolerant.

Pin Diagrams

44-Pin TQFP/QFN(1,3)



Legend: Pin numbers in **bold** indicate pin function differences between PIC24FV and PIC24F devices.

Note 1: Exposed pad on underside of device is connected to VSS.

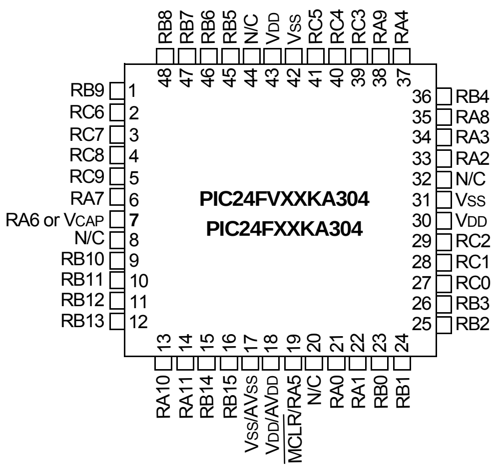
2: Alternative multiplexing for SDA1 (ASDA1) and SCL1 (ASCL1) when the I2CSEL Configuration bit is set.

3: PIC24F32KA304 device pins have a maximum voltage of 3.6V and are not 5V tolerant.

Pin	Pin Features	
	PIC24FVXXKA304	PIC24FXXKA304
1	SDA1/T1CK/U1RTS/CTED4/CN21/RB9	SDA1/T1CK/U1RTS/CTED4/CN21/RB9
2	U1RX/CN18/RC6	U1RX/CN18/RC6
3	U1TX/CN17/RC7	U1TX/CN17/RC7
4	OC2/CN20/RC8	OC2/CN20/RC8
5	IC2/CTED7/CN19/RC9	IC2/CTED7/CN19/RC9
6	IC1/CTED3/CN9/RA7	IC1/CTED3/CN9/RA7
7	**VCAP**	**C2OUT/OC1/CTED1/INT2/CN8/RA6**
8	PGED2/SDI1/CTED11/CN16/RB10	PGED2/SDI1/CTED11/CN16/RB10
9	PGEC2/SCK1/CTED9/CN15/RB11	PGEC2/SCK1/CTED9/CN15/RB11
10	**AN12/HLVDIN/CTED2/INT2/CN14/RB12**	**AN12/HLVDIN/CTED2/CN14/RB12**
11	AN11/SDO1/CTPLS/CN13/RB13	AN11/SDO1/CTPLS/CN13/RB13
12	OC3/CN35/RA10	OC3/CN35/RA10
13	IC3/CTED8/CN36/RA11	IC3/CTED8/CN36/RA11
14	CVREF/AN10/C3INB/RTCC/C1OUT/OCFA/CTED5/INT1/CN12/RB14	CVREF/AN10/C3INB/RTCC/C1OUT/OCFA/CTED5/INT1/CN12/RB14
15	AN9/C3INA/T3CK/T2CK/REFO/SS1/CTED6/CN11/RB15	AN9/C3INA/T3CK/T2CK/REFO/SS1/CTED6/CN11/RB15
16	VSS/AVSS	VSS/AVSS
17	VDD/AVDD	VDD/AVDD
18	MCLR/VPP/RA5	MCLR/VPP/RA5
19	VREF+/CVREF+/AN0/C3INC/CTED1/CN2/RA0	VREF+/CVREF+/AN0/C3INC/CN2/RA0
20	CVREF-/VREF-/AN1/CN3/RA1	CVREF-/VREF-/AN1/CN3/RA1
21	PGED1/AN2/ULPWU/CTCMP/C1IND/C2INB/C3IND/U2TX/CN4/RB0	PGED1/AN2/ULPWU/CTCMP/C1IND/C2INB/C3IND/U2TX/CN4/RB0
22	PGEC1/AN3/C1INC/C2INA/U2RX/CTED12/CN5/RB1	PGEC1/AN3/C1INC/C2INA/U2RX/CTED12/CN5/RB1
23	AN4/C1INB/C2IND/SDA2/T5CK/T4CK/CTED13/CN6/RB2	AN4/C1INB/C2IND/SDA2/T5CK/T4CK/CTED13/CN6/RB2
24	AN5/C1INA/C2INC/SCL2/CN7/RB3	AN5/C1INA/C2INC/SCL2/CN7/RB3
25	AN6/CN32/RC0	AN6/CN32/RC0
26	AN7/CN31/RC1	AN7/CN31/RC1
27	AN8/CN10/RC2	AN8/CN10/RC2
28	VDD	VDD
29	VSS	VSS
30	OSCI/AN13/CLKI/CN30/RA2	OSCI/AN13/CLKI/CN30/RA2
31	OSCO/AN14/CLKO/CN29/RA3	OSCO/AN14/CLKO/CN29/RA3
32	OCFB/CN33/RA8	OCFB/CN33/RA8
33	SOSCI/AN15/U2RTS/CN1/RB4	SOSCI/AN15/U2RTS/CN1/RB4
34	SOSCO/SCLKI/U2CTS/CN0/RA4	SOSCO/SCLKI/U2CTS/CN0/RA4
35	SS2/CN34/RA9	SS2/CN34/RA9
36	SDI2/CN28/RC3	SDI2/CN28/RC3
37	SDO2/CN25/RC4	SDO2/CN25/RC4
38	SCK2/CN26/RC5	SCK2/CN26/RC5
39	VSS	VSS
40	VDD	VDD
41	PGED3/ASDA1(2)/CN27/RB5	PGED3/ASDA1(2)/CN27/RB5
42	PGEC3/ASCL1(2)/CN24/RB6	PGEC3/ASCL1(2)/CN24/RB6
43	C2OUT/OC1/INT0/CN23/RB7	INT0/CN23/RB7
44	SCL1/U1CTS/C3OUT/CTED10/CN22/RB8	SCL1/U1CTS/C3OUT/CTED10/CN22/RB8

Pin Diagrams

48-Pin UQFN(1,3)



Legend: Pin numbers in **bold** indicate pin function differences between PIC24FV and PIC24F devices.

Note 1: Exposed pad on underside of device is connected to VSS.

2: Alternative multiplexing for SDA1 (ASDA1) and SCL1 (ASCL1) when the I2CSEL Configuration bit is set.

3: PIC24F32KA3XX device pins have a maximum voltage of 3.6V and are not 5V tolerant.

Pin	Pin Features	
	PIC24FVXXKA304	**PIC24FXXKA304**
1	SDA1/T1CK/$\overline{\text{U1RTS}}$/CTED4/CN21/RB9	SDA1/T1CK/$\overline{\text{U1RTS}}$/CTED4/CN21/RB9
2	U1RX/CN18/RC6	U1RX/CN18/RC6
3	U1TX/CN17/RC7	U1TX/CN17/RC7
4	OC2/CN20/RC8	OC2/CN20/RC8
5	IC2/CTED7/CN19/RC9	IC2/CTED7/CN19/RC9
6	IC1/CTED3/CN9/RA7	IC1/CTED3/CN9/RA7
7	**VCAP**	**C20UT/OC1/CTED1/INT2CN8/RA6**
8	N/C	N/C
9	PGED2/SDI1/CTED11/CN16/RB10	PGED2/SDI1/CTED11/CN16/RB10
10	PGEC2/SCK1/CTED9/CN15/RB11	PGEC2/SCK1/CTED9/CN15/RB11
11	AN12/HLVDIN/CTED2/INT2/CN14/RB12	AN12/HLVDIN/CTED2/INT2/CN14/RB12
12	AN11/SDO1/CTPLS/CN13/RB13	AN11/SDO1/CTPLS/CN13/RB13
13	OC3/CN35/RA10	OC3/CN35/RA10
14	IC3/CTED8/CN36/RA11	IC3/CTED8/CN36/RA11
15	CVREF/AN10/C3INB/RTCC/C1OUT/OCFA/CTED5/INT1/CN12/RB14	CVREF/AN10/C3INB/RTCC/C1OUT/OCFA/CTED5/INT1/CN12/RB14
16	AN9/C3INA/T3CK/T2CK/REFO/$\overline{\text{SS1}}$/CTED6/CN11/RB15	AN9/C3INA/T3CK/T2CK/REFO/$\overline{\text{SS1}}$/CTED6/CN11/RB15
17	VSS/AVSS	VSS/AVSS
18	VDD/AVDD	VDD/AVDD
19	$\overline{\text{MCLR}}$/RA5	$\overline{\text{MCLR}}$/RA5
20	N/C	N/C
21	VREF+/CVREF+/AN0/C3INC/CTED1/CN2/RA0	VREF+/CVREF+/AN0/C3INC/CN2/RA0
22	CVREF-/VREF-/AN1/CN3/RA1	CVREF-/VREF-/AN1/CN3/RA1
23	PGED1/AN2/ULPWU/CTCMP/C1IND/C2INB/C3IND/U2TX/CN4/RB0	PGED1/AN2/ULPWU/CTCMP/C1IND/C2INB/C3IND/U2TX/CN4/RB0
24	PGEC1/AN3/C1INC/C2INA/U2RX/CTED12/CN5/RB1	PGEC1/AN3/C1INC/C2INA/U2RX/CTED12/CN5/RB1
25	AN4/C1INB/C2IND/SDA2/T5CK/T4CK/CTED13/CN6/RB2	AN4/C1INB/C2IND/SDA2/T5CK/T4CK/CTED13/CN6/RB2
26	AN5/C1INA/C2INC/SCL2/CN7/RB3	AN5/C1INA/C2INC/SCL2/CN7/RB3
27	AN6/CN32/RC0	AN6/CN32/RC0
28	AN7/CN31/RC1	AN7/CN31/RC1
29	AN8/CN10/RC2	AN8/CN10/RC2
30	VDD	VDD
31	VSS	VSS
32	N/C	N/C
33	OSCI/AN13/CLKI/CN30/RA2	OSCI/AN13/CLKI/CN30/RA2
34	OSCO/AN14/CLKO/CN29/RA3	OSCO/AN14/CLKO/CN29/RA3
35	OCFB/CN33/RA8	OCFB/CN33/RA8
36	SOSCI/AN15/$\overline{\text{U2RTS}}$/CN1/RB4	SOSCI/AN15/$\overline{\text{U2RTS}}$/CN1/RB4
37	SOSCO/SCLKI/$\overline{\text{U2CTS}}$/CN0/RA4	SOSCO/SCLKI/$\overline{\text{U2CTS}}$/CN0/RA4
38	$\overline{\text{SS2}}$/CN34/RA9	$\overline{\text{SS2}}$/CN34/RA9
39	SDI2/CN28/RC3	SDI2/CN28/RC3
40	SDO2/CN25/RC4	SDO2/CN25/RC4
41	SCK2/CN26/RC5	SCK2/CN26/RC5
42	VSS	VSS
43	VDD	VDD
44	N/C	N/C
45	PGED3/ASDA1(2)/CN27/RB5	PGED3/ASDA1(2)/CN27/RB5
46	PGEC3/ASCL1(2)/CN24/RB6	PGEC3/ASCL1(2)/CN24/RB6
47	C2OUT/OC1/INT0/CN23/RB7	INT0/CN23/RB7
48	SCL1/$\overline{\text{U1CTS}}$/C3OUT/CTED10/CN22/RB8	SCL1/$\overline{\text{U1CTS}}$/C3OUT/CTED10/CN22/RB8

Table of Contents

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**. 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., DS30000000A is version A of document DS30000000).

Errata

An errata sheet, describing minor operational differences from the data sheet and recommended work arounds, 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.

NOTES:

1.0 DEVICE OVERVIEW

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

- PIC24FV16KA301, PIC24F16KA301
- PIC24FV16KA302, PIC24F16KA302
- PIC24FV16KA304, PIC24F16KA304
- PIC24FV32KA301, PIC24F32KA301
- PIC24FV32KA302, PIC24F32KA302
- PIC24FV32KA304, PIC24F32KA304

The PIC24FV32KA304 family introduces a new line of extreme low-power Microchip devices. This is a 16-bit microcontroller family with a broad peripheral feature set and enhanced computational performance. This family also offers a new migration option for those high-performance applications which may be outgrowing their 8-bit platforms, but do not require the numerical processing power of a Digital Signal Processor (DSP).

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. 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 64 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-bit 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 PIC24FV32KA304 family incorporate a range of features that can significantly reduce power consumption during operation. Key features 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 users 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:** There are three instruction-based power-saving modes:
  - Idle Mode: The core is shut down while leaving the peripherals active.
  - Sleep Mode: The core and peripherals that require the system clock are shut down, leaving the peripherals that use their own clock, or the clock from other devices, active.
  - Deep Sleep Mode: The core, peripherals (except RTCC and DSWDT), Flash and SRAM are shut down.

1.1.3 OSCILLATOR OPTIONS AND FEATURES

The PIC24FV32KA304 family offers 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.
- Two Fast Internal oscillators (FRCs): One with a nominal 8 MHz output and the other with a nominal 500 kHz output. These outputs can also be divided under software control to provide clock speed as low as 31 kHz or 2 kHz.
- A Phase Locked Loop (PLL) frequency multiplier, available to the external Oscillator modes and the 8 MHz FRC oscillator, which allows clock speeds of up to 32 MHz.
- A separate 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 the 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 helps in migrating to the next larger device. This is true when moving between devices with the same pin count, or even jumping from 20-pin or 28-pin devices to 44-pin/48-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.

1.2 Other Special Features

- **Communications:** The PIC24FV32KA304 family incorporates a range of serial communication peripherals to handle a range of application requirements. There is an I^{2}C module that supports both the Master and Slave modes of operation. It also comprises UARTs with built-in IrDA® encoders/decoders and an SPI module.
- **Real-Time Clock/Calendar:** 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.
- **12-Bit A/D Converter:** This 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 speed. The 16-deep result buffer can be used either in Sleep to reduce power, or in Active mode to improve throughput.
- **Charge Time Measurement Unit (CTMU) Interface:** The PIC24FV32KA304 family includes the new CTMU interface module, which can be used for capacitive touch sensing, proximity sensing, and also for precision time measurement and pulse generation.

1.3 Details on Individual Family Members

Devices in the PIC24FV32KA304 family are available in 20-pin, 28-pin, 44-pin and 48-pin packages. The general block diagram for all devices is shown in Figure 1-1.

The devices are different from each other in four ways:

1. Flash program memory (16 Kbytes for PIC24FV16KA devices, 32 Kbytes for PIC24FV32KA devices).
2. Available I/O pins and ports (18 pins on two ports for 20-pin devices, 22 pins on two ports for 28-pin devices and 38 pins on three ports for 44/48-pin devices).
3. Alternate SCLx and SDAx pins are available only in 28-pin, 44-pin and 48-pin devices and not in 20-pin devices.
4. Members of the PIC24FV32KA301 family are available as both standard and high-voltage devices. High-voltage devices, designated with an “FV” in the part number (such as PIC24FV32KA304), accommodate an operating VDD range of 2.0V to 5.5V, and have an on-board Voltage Regulator that powers the core. Peripherals operate at VDD. Standard devices, designated by “F” (such as PIC24F32KA304), function over a lower VDD range of 1.8V to 3.6V. These parts do not have an internal regulator, and both the core and peripherals operate directly from VDD.

All other features for devices in this family are identical; these are summarized in Table 1-1.

A list of the pin features available on the PIC24FV32KA304 family devices, sorted by function, is provided in Table 1-3.

Note:	Table 1-1 provides 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 on pages 3, 4, 5, 6 and 7 of the data sheet. Multiplexed features are sorted by the priority given to a feature, with the highest priority peripheral being listed first.

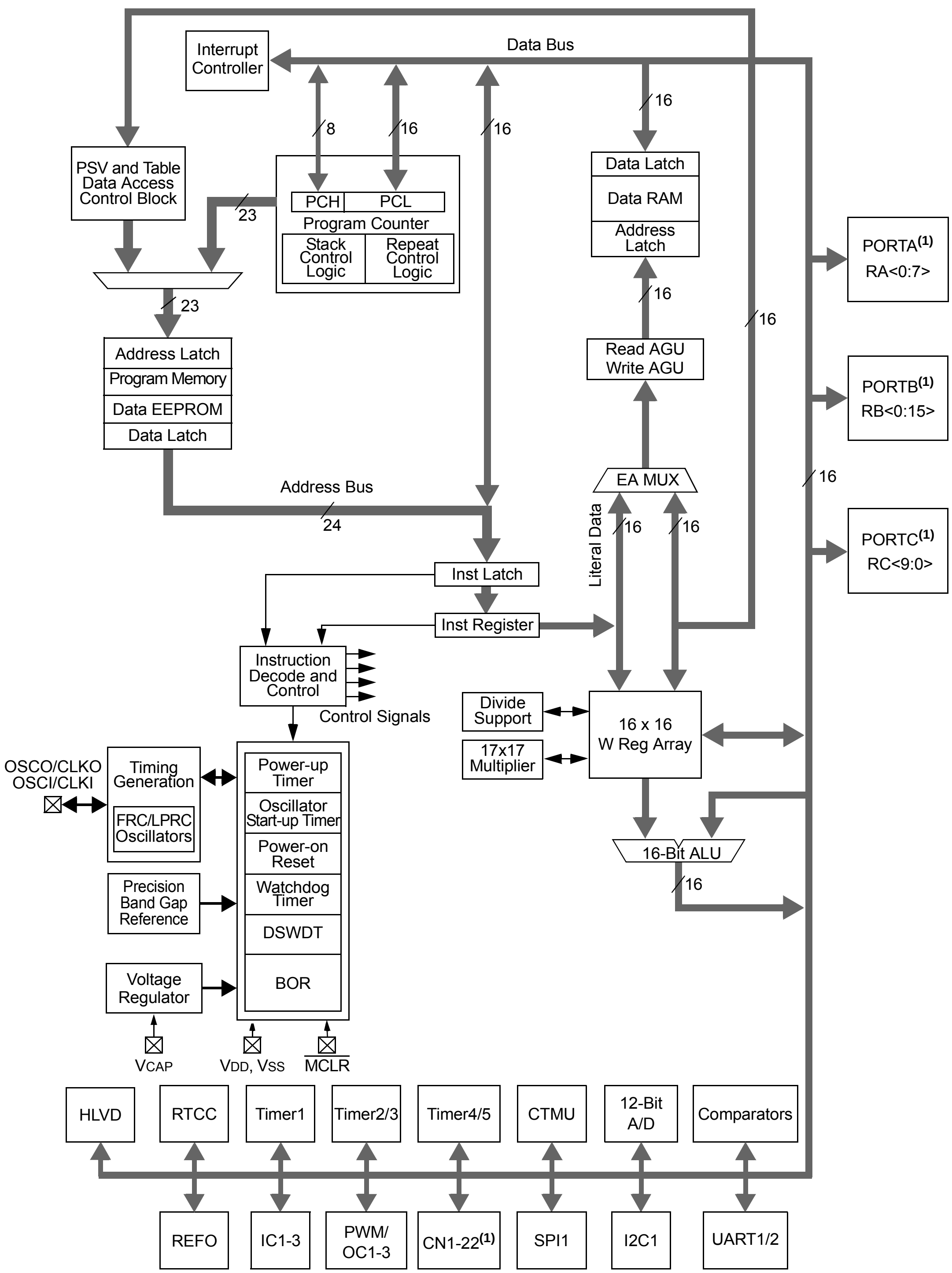
TABLE 1-1: DEVICE FEATURES FOR THE PIC24FV32KA304 FAMILY

Features	PIC24FV16KA301	PIC24FV32KA301	PIC24FV16KA302	PIC24FV32KA302	PIC24FV16KA304	PIC24FV32KA304
Operating Frequency	DC – 32 MHz					
Program Memory (bytes)	16K	32K	16K	32K	16K	32K
Program Memory (instructions)	5632	11264	5632	11264	5632	11264
Data Memory (bytes)	2048					
Data EEPROM Memory (bytes)	512					
Interrupt Sources (soft vectors/ NMI traps)	30 (26/4)					
I/O Ports	PORTA<5:0> PORTB<15:12,9:7,4,2:0>		PORTA<7,5:0> PORTB<15:0>		PORTA<11:7,5:0> PORTB<15:0> PORTC<9:0>	
Total I/O Pins	17		23		38	
Timers: Total Number (16-bit)	5					
32-Bit (from paired 16-bit timers)	2					
Input Capture Channels	3					
Output Compare/PWM Channels	3					
Input Change Notification Interrupt	16		22		37	
Serial Communications: UART SPI (3-wire/4-wire)	2					
I^{2}C	2					
12-Bit Analog-to-Digital Module (input channels)	12		13		16	
Analog Comparators	3					
Resets (and delays)	POR, BOR, RESET Instruction, $\overline{\text{MCLR}}$, WDT, Illegal Opcode, REPEAT Instruction, Hardware Traps, Configuration Word Mismatch (PWRT, OST, PLL Lock)					
Instruction Set	76 Base Instructions, Multiple Addressing Mode Variations					
Packages	20-Pin PDIP/SSOP/SOIC		28-Pin SPDIP/SSOP/SOIC/QFN		44-Pin QFN/TQFP 48-Pin UQFN	

TABLE 1-2: DEVICE FEATURES FOR THE PIC24F32KA304 FAMILY

Features	PIC24F16KA301	PIC24F32KA301	PIC24F16KA302	PIC24F32KA302	PIC16F16KA304	PIC24F32KA304
Operating Frequency	DC – 32 MHz					
Program Memory (bytes)	16K	32K	16K	32K	16K	32K
Program Memory (instructions)	5632	11264	5632	11264	5632	11264
Data Memory (bytes)	2048					
Data EEPROM Memory (bytes)	512					
Interrupt Sources (soft vectors/ NMI traps)	30 (26/4)					
I/O Ports	PORTA<6:0>, PORTB<15:12, 9:7, 4, 2:0>		PORTA<7:0>, PORTB<15:0>		PORTA<11:0>, PORTB<15:0>, PORTC<9:0>	
Total I/O Pins	18		24		39	
Timers: Total Number (16-bit)	5					
32-Bit (from paired 16-bit timers)	2					
Input Capture Channels	3					
Output Compare/PWM Channels	3					
Input Change Notification Interrupt	17		23		38	
Serial Communications: UART SPI (3-wire/4-wire)	2					
I^2C	2					
12-Bit Analog-to-Digital Module (input channels)	12		13		16	
Analog Comparators	3					
Resets (and delays)	POR, BOR, RESET Instruction, $\overline{\text{MCLR}}$, WDT, Illegal Opcode, REPEAT Instruction, Hardware Traps, Configuration Word Mismatch (PWRT, OST, PLL Lock)					
Instruction Set	76 Base Instructions, Multiple Addressing Mode Variations					
Packages	20-Pin PDIP/SSOP/SOIC		28-Pin SPDIP/SSOP/SOIC/QFN		44-Pin QFN/TQFP 48-Pin UQFN	

FIGURE 1-1: PIC24FV32KA304 FAMILY GENERAL BLOCK DIAGRAM



Note 1: All pins or features are not implemented on all device pinout configurations. See Table 1-3 for I/O port pin descriptions.

TABLE 1-3: PIC24FV32KA304 FAMILY PINOUT DESCRIPTIONS

Function	F					FV					I/O	Buffer	Description
	Pin Number					Pin Number							
	20-Pin PDIP/ SSOP/ SOIC	28-Pin SPDIP/ SSOP/ SOIC	28-Pin QFN	44-Pin QFN/ TQFP	48-Pin UQFN	20-Pin PDIP/ SSOP/ SOIC	28-Pin SPDIP/ SSOP/ SOIC	28-Pin QFN	44-Pin QFN/ TQFP	48-Pin UQFN			
AN0	2	2	27	19	21	2	2	27	19	21	I	ANA	A/D Analog Inputs
AN1	3	3	28	20	22	3	3	28	20	22	I	ANA	
AN2	4	4	1	21	23	4	4	1	21	23	I	ANA	
AN3	5	5	2	22	24	5	5	2	22	24	I	ANA	
AN4	6	6	3	23	25	6	6	3	23	25	I	ANA	
AN5	—	7	4	24	26	—	7	4	24	26	I	ANA	
AN6	—	—	—	25	27	—	—	—	25	27	I	ANA	
AN7	—	—	—	26	28	—	—	—	26	28	I	ANA	
AN8	—	—	—	27	29	—	—	—	27	29	I	ANA	
AN9	18	26	23	15	16	18	26	23	15	16	I	ANA	
AN10	17	25	22	14	15	17	25	22	14	15	I	ANA	
AN11	16	24	21	11	12	16	24	21	11	12	I	ANA	
AN12	15	23	20	10	11	15	23	20	10	11	I	ANA	
AN13	7	9	6	30	33	7	9	6	30	33	I	ANA	
AN14	8	10	7	31	34	8	10	7	31	34	I	ANA	
AN15	9	11	8	33	36	9	11	8	33	36	I	ANA	
ASCL1	—	15	12	42	46	—	15	12	42	46	I/O	I^2C	Alternate I2C1 Clock Input/Output
ASDA1	—	14	11	41	45	—	14	11	41	45	I/O	I^2C	Alternate I2C1 Data Input/Output
AVDD	20	28	25	17	18	20	28	25	17	18	I	ANA	A/D Supply Pins
AVSS	19	27	24	16	17	19	27	24	16	17	I	ANA	
C1INA	8	7	4	24	26	8	7	4	24	26	I	ANA	Comparator 1 Input A (+)
C1INB	7	6	3	23	25	7	6	3	23	25	I	ANA	Comparator 1 Input B (-)
C1INC	5	5	2	22	24	5	5	2	22	24	I	ANA	Comparator 1 Input C (+)
C1IND	4	4	1	21	23	4	4	1	21	23	I	ANA	Comparator 1 Input D (-)
C1OUT	17	25	22	14	15	17	25	22	14	15	O	—	Comparator 1 Output
C2INA	5	5	2	22	24	5	5	2	22	24	I	ANA	Comparator 2 Input A (+)
C2INB	4	4	1	21	23	4	4	1	21	23	I	ANA	Comparator 2 Input B (-)
C2INC	8	7	4	24	26	8	7	4	24	26	I	ANA	Comparator 2 Input C (+)
C2IND	7	6	3	23	25	7	6	3	23	25	I	ANA	Comparator 2 Input D (-)
C2OUT	14	20	17	7	7	11	16	13	43	47	O	—	Comparator 2 Output

TABLE 1-3: PIC24FV32KA304 FAMILY PINOUT DESCRIPTIONS (CONTINUED)

Function	F					FV					I/O	Buffer	Description
	Pin Number					Pin Number							
	20-Pin PDIP/ SSOP/ SOIC	28-Pin SPDIP/ SSOP/ SOIC	28-Pin QFN	44-Pin QFN/ TQFP	48-Pin UQFN	20-Pin PDIP/ SSOP/ SOIC	28-Pin SPDIP/ SSOP/ SOIC	28-Pin QFN	44-Pin QFN/ TQFP	48-Pin UQFN			
C3INA	18	26	23	15	16	18	26	23	15	16	I	ANA	Comparator 3 Input A (+)
C3INB	17	25	22	14	15	17	25	22	14	15	I	ANA	Comparator 3 Input B (-)
C3INC	2	2	27	19	21	2	2	27	19	21	I	ANA	Comparator 3 Input C (+)
C3IND	4	4	1	21	23	4	4	1	21	23	I	ANA	Comparator 3 Input D (-)
C3OUT	12	17	14	44	48	12	17	14	44	48	O	—	Comparator 3 Output
CLK I	7	9	6	30	33	7	9	6	30	33	I	ANA	Main Clock Input
CLKO	8	10	7	31	34	8	10	7	31	34	O	—	System Clock Output
CN0	10	12	9	34	37	10	12	9	34	37	I	ST	Interrupt-on-Change Inputs
CN1	9	11	8	33	36	9	11	8	33	36	I	ST	
CN2	2	2	27	19	21	2	2	27	19	21	I	ST	
CN3	3	3	28	20	22	3	3	28	20	22	I	ST	
CN4	4	4	1	21	23	4	4	1	21	23	I	ST	
CN5	5	5	2	22	24	5	5	2	22	24	I	ST	
CN6	6	6	3	23	25	6	6	3	23	25	I	ST	
CN7	—	7	4	24	26	—	7	4	24	26	I	ST	
CN8	14	20	17	7	7	—	—	—	—	—	I	ST	
CN9	—	19	16	6	6	—	19	16	6	6	I	ST	
CN10	—	—	—	27	29	—	—	—	27	29	I	ST	
CN11	18	26	23	15	16	18	26	23	15	16	I	ST	
CN12	17	25	22	14	15	17	25	22	14	15	I	ST	
CN13	16	24	21	11	12	16	24	21	11	12	I	ST	
CN14	15	23	20	10	11	15	23	20	10	11	I	ST	
CN15	—	22	19	9	10	—	22	19	9	10	I	ST	
CN16	—	21	18	8	9	—	21	18	8	9	I	ST	
CN17	—	—	—	3	3	—	—	—	3	3	I	ST	
CN18	—	—	—	2	2	—	—	—	2	2	I	ST	
CN19	—	—	—	5	5	—	—	—	5	5	I	ST	
CN20	—	—	—	4	4	—	—	—	4	4	I	ST	
CN21	13	18	15	1	1	13	18	15	1	1	I	ST	
CN22	12	17	14	44	48	12	17	14	44	48	I	ST	

TABLE 1-3: PIC24FV32KA304 FAMILY PINOUT DESCRIPTIONS (CONTINUED)

Function	F					FV					I/O	Buffer	Description
	Pin Number					Pin Number							
	20-Pin PDIP/ SSOP/ SOIC	28-Pin SPDIP/ SSOP/ SOIC	28-Pin QFN	44-Pin QFN/ TQFP	48-Pin UQFN	20-Pin PDIP/ SSOP/ SOIC	28-Pin SPDIP/ SSOP/ SOIC	28-Pin QFN	44-Pin QFN/ TQFP	48-Pin UQFN			
CN23	11	16	13	43	47	11	16	13	43	47	I	ST	Interrupt-on-Change Inputs
CN24	—	15	12	42	46	—	15	12	42	46	I	ST	
CN25	—	—	—	37	40	—	—	—	37	40	I	ST	
CN26	—	—	—	38	41	—	—	—	38	41	I	ST	
CN27	—	14	11	41	45	—	14	11	41	45	I	ST	
CN28	—	—	—	36	39	—	—	—	36	39	I	ST	
CN29	8	10	7	31	34	8	10	7	31	34	I	ST	
CN30	7	9	6	30	33	7	9	6	30	33	I	ST	
CN31	—	—	—	26	28	—	—	—	26	28	I	ST	
CN32	—	—	—	25	27	—	—	—	25	27	I	ST	
CN33	—	—	—	32	35	—	—	—	32	35	I	ST	
CN34	—	—	—	35	38	—	—	—	35	38	I	ST	
CN35	—	—	—	12	13	—	—	—	12	13	I	ST	
CN36	—	—	—	13	14	—	—	—	13	14	I	ST	
CVREF	17	25	22	14	15	17	25	22	14	15	I	ANA	Comparator Voltage Reference Output
CVREF+	2	2	27	19	21	2	2	27	19	21	I	ANA	Comparator Reference Positive Input Voltage
CVREF-	3	3	28	20	22	3	3	28	20	22	I	ANA	Comparator Reference Negative Input Voltage
CTCMP	4	4	1	21	23	4	4	1	21	23	I	ANA	CTMU Comparator Input
CTED1	14	20	17	7	7	11	2	27	19	21	I	ST	CTMU Trigger Edge Inputs
CTED2	15	23	20	10	11	15	23	20	10	11	I	ST	
CTED3	—	19	16	6	6	—	19	16	6	6	I	ST	
CTED4	13	18	15	1	1	13	18	15	1	1	I	ST	
CTED5	17	25	22	14	15	17	25	22	14	15	I	ST	
CTED6	18	26	23	15	16	18	26	23	15	16	I	ST	
CTED7	—	—	—	5	5	—	—	—	5	5	I	ST	
CTED8	—	—	—	13	14	—	—	—	13	14	I	ST	
CTED9	—	22	19	9	10	—	22	19	9	10	I	ST	
CTED10	12	17	14	44	48	12	17	14	44	48	I	ST	
CTED11	—	21	18	8	9	—	21	18	8	9	I	ST	
CTED12	5	5	2	22	24	5	5	2	22	24	I	ST	
CTED13	6	6	3	23	25	6	6	3	23	25	I	ST	

TABLE 1-3: PIC24FV32KA304 FAMILY PINOUT DESCRIPTIONS (CONTINUED)

Function	F					FV					I/O	Buffer	Description
	Pin Number					Pin Number							
	20-Pin PDIP/ SSOP/ SOIC	28-Pin SPDIP/ SSOP/ SOIC	28-Pin QFN	44-Pin QFN/ TQFP	48-Pin UQFN	20-Pin PDIP/ SSOP/ SOIC	28-Pin SPDIP/ SSOP/ SOIC	28-Pin QFN	44-Pin QFN/ TQFP	48-Pin UQFN			
CTPLS	16	24	21	11	12	16	24	21	11	12	O	—	CTMU Pulse Output
HLVDIN	15	23	20	10	11	15	23	20	10	11	I	ST	High/Low-Voltage Detect Input
IC1	14	19	16	6	6	11	19	16	6	6	I	ST	Input Capture 1 Input
IC2	13	18	15	5	5	13	18	15	5	5	I	ST	Input Capture 2 Input
IC3	15	23	20	13	14	15	23	20	13	14	I	ST	Input Capture 3 Input
INT0	11	16	13	43	47	11	16	13	43	47	I	ST	Interrupt 0 Input
INT1	17	25	22	14	15	17	25	22	14	15	I	ST	Interrupt 1 Input
INT2	14	20	17	7	7	15	23	20	10	11	I	ST	Interrupt 2 Input
$\overline{\text{MCLR}}$	1	1	26	18	19	1	1	26	18	19	I	ST	Master Clear (Device Reset) Input (active-low)
OC1	14	20	17	7	7	11	16	13	43	47	O	—	Output Compare/PWM1 Output
OC2	4	22	19	4	4	4	22	19	4	4	O	—	Output Compare/PWM2 Output
OC3	5	21	18	12	13	5	21	18	12	13	O	—	Output Compare/PWM3 Output
OCFA	17	25	22	14	15	17	25	22	14	15	O	—	Output Compare Fault A
OFCB	16	24	21	32	35	16	24	21	32	35	O	—	Output Compare Fault B
OSCI	7	9	6	30	33	7	9	6	30	33	I	ANA	Main Oscillator Input
OSCO	8	10	7	31	34	8	10	7	31	34	O	ANA	Main Oscillator Output
PGEC1	5	5	2	22	24	5	5	2	22	24	I/O	ST	ICSP™ Clock 1
PCED1	4	4	1	21	23	4	4	1	21	23	I/O	ST	ICSP Data 1
PGEC2	2	22	19	9	10	2	22	19	9	10	I/O	ST	ICSP Clock 2
PGED2	3	21	18	8	9	3	21	18	8	9	I/O	ST	ICSP Data 2
PGEC3	10	15	12	42	46	10	15	12	42	46	I/O	ST	ICSP Clock 3
PGED3	9	14	11	41	45	9	14	11	41	45	I/O	ST	ICSP Data 3

TABLE 1-3: PIC24FV32KA304 FAMILY PINOUT DESCRIPTIONS (CONTINUED)

Function	F					FV					I/O	Buffer	Description
	Pin Number					Pin Number							
	20-Pin PDIP/ SSOP/ SOIC	28-Pin SPDIP/ SSOP/ SOIC	28-Pin QFN	44-Pin QFN/ TQFP	48-Pin UQFN	20-Pin PDIP/ SSOP/ SOIC	28-Pin SPDIP/ SSOP/ SOIC	28-Pin QFN	44-Pin QFN/ TQFP	48-Pin UQFN			
RA0	2	2	27	19	21	2	2	27	19	21	I/O	ST	PORTA Pins
RA1	3	3	28	20	22	3	3	28	20	22	I/O	ST	
RA2	7	9	6	30	33	7	9	6	30	33	I/O	ST	
RA3	8	10	7	31	34	8	10	7	31	34	I/O	ST	
RA4	10	12	9	34	37	10	12	9	34	37	I/O	ST	
RA5	1	1	26	18	19	1	1	26	18	19	I/O	ST	
RA6	14	20	17	7	7	—	—	—	—	—	I/O	ST	
RA7	—	19	16	6	6	—	19	16	6	6	I/O	ST	
RA8	—	—	—	32	35	—	—	—	32	35	I/O	ST	
RA9	—	—	—	35	38	—	—	—	35	38	I/O	ST	
RA10	—	—	—	12	13	—	—	—	12	13	I/O	ST	
RA11	—	—	—	13	14	—	—	—	13	14	I/O	ST	
RB0	4	4	1	21	23	4	4	1	21	23	I/O	ST	PORTB Pins
RB1	5	5	2	22	24	5	5	2	22	24	I/O	ST	
RB2	6	6	3	23	25	6	6	3	23	25	I/O	ST	
RB3	—	7	4	24	26	—	7	4	24	26	I/O	ST	
RB4	9	11	8	33	36	9	11	8	33	36	I/O	ST	
RB5	—	14	11	41	45	—	14	11	41	45	I/O	ST	
RB6	—	15	12	42	46	—	15	12	42	46	I/O	ST	
RB7	11	16	13	43	47	11	16	13	43	47	I/O	ST	
RB8	12	17	14	44	48	12	17	14	44	48	I/O	ST	
RB9	13	18	15	1	1	13	18	15	1	1	I/O	ST	
RB10	—	21	18	8	9	—	21	18	8	9	I/O	ST	
RB11	—	22	19	9	10	—	22	19	9	10	I/O	ST	
RB12	15	23	20	10	11	15	23	20	10	11	I/O	ST	
RB13	16	24	21	11	12	16	24	21	11	12	I/O	ST	
RB14	17	25	22	14	15	17	25	22	14	15	I/O	ST	
RB15	18	26	23	15	16	18	26	23	15	16	I/O	ST	

TABLE 1-3: PIC24FV32KA304 FAMILY PINOUT DESCRIPTIONS (CONTINUED)

Function	F					FV					I/O	Buffer	Description
	Pin Number					Pin Number							
	20-Pin PDIP/ SSOP/ SOIC	28-Pin SPDIP/ SSOP/ SOIC	28-Pin QFN	44-Pin QFN/ TQFP	48-Pin UQFN	20-Pin PDIP/ SSOP/ SOIC	28-Pin SPDIP/ SSOP/ SOIC	28-Pin QFN	44-Pin QFN/ TQFP	48-Pin UQFN			
RC0	—	—	—	25	27	—	—	—	25	27	I/O	ST	PORTC Pins
RC1	—	—	—	26	28	—	—	—	26	28	I/O	ST	
RC2	—	—	—	27	29	—	—	—	27	29	I/O	ST	
RC3	—	—	—	36	39	—	—	—	36	39	I/O	ST	
RC4	—	—	—	37	40	—	—	—	37	40	I/O	ST	
RC5	—	—	—	38	41	—	—	—	38	41	I/O	ST	
RC6	—	—	—	2	2	—	—	—	2	2	I/O	ST	
RC7	—	—	—	3	3	—	—	—	3	3	I/O	ST	
RC8	—	—	—	4	4	—	—	—	4	4	I/O	ST	
RC9	—	—	—	5	5	—	—	—	5	5	I/O	ST	
REFO	18	26	23	15	16	18	26	23	15	16	O	—	Reference Clock Output
RTCC	17	25	22	14	15	17	25	22	14	15	O	—	Real-Time Clock/Calendar Output
SCK1	15	22	19	9	10	15	22	19	9	10	I/O	ST	SPI1 Serial Input/Output Clock
SCK2	2	14	11	38	41	2	14	11	38	41	I/O	ST	SPI2 Serial Input/Output Clock
SCL1	12	17	14	44	48	12	17	14	44	48	I/O	I^2C	I2C1 Clock Input/Output
SCL2	18	7	4	24	26	18	7	4	24	26	I/O	I^2C	I2C2 Clock Input/Output
SCLKI	10	12	9	34	37	10	12	9	34	37	I	ST	Digital Secondary Clock Input
SDA1	13	18	15	1	1	13	18	15	1	1	I/O	I^2C	I2C1 Data Input/Output
SDA2	6	6	3	23	25	6	6	3	23	25	I/O	I^2C	I2C2 Data Input/Output
SDI1	17	21	18	8	9	17	21	18	8	9	I	ST	SPI1 Serial Data Input
SDI2	4	19	16	36	39	4	19	16	36	39	I	ST	SPI2 Serial Data Input
SDO1	16	24	21	11	12	16	24	21	11	12	O	—	SPI1 Serial Data Output
SDO2	3	15	12	37	40	3	15	12	37	40	O	—	SPI2 Serial Data Output
SOSCI	9	11	8	33	36	9	11	8	33	36	I	ANA	Secondary Oscillator Input
SOSCO	10	12	9	34	37	10	12	9	34	37	O	ANA	Secondary Oscillator Output
$\overline{SS1}$	18	26	23	15	16	18	26	23	15	16	O	—	SPI1 Slave Select
$\overline{SS2}$	15	23	20	35	38	15	23	20	35	38	O	—	SPI2 Slave Select

TABLE 1-3: PIC24FV32KA304 FAMILY PINOUT DESCRIPTIONS (CONTINUED)

Function	F					FV					I/O	Buffer	Description
	Pin Number					Pin Number							
	20-Pin PDIP/ SSOP/ SOIC	28-Pin SPDIP/ SSOP/ SOIC	28-Pin QFN	44-Pin QFN/ TQFP	48-Pin UQFN	20-Pin PDIP/ SSOP/ SOIC	28-Pin SPDIP/ SSOP/ SOIC	28-Pin QFN	44-Pin QFN/ TQFP	48-Pin UQFN			
T1CK	13	18	15	1	1	13	18	15	1	1	I	ST	Timer1 Clock
T2CK	18	26	23	15	16	18	26	23	15	16	I	ST	Timer2 Clock
T3CK	18	26	23	15	16	18	26	23	15	16	I	ST	Timer3 Clock
T4CK	6	6	3	23	25	6	6	3	23	25	I	ST	Timer4 Clock
T5CK	6	6	3	23	25	6	6	3	23	25	I	ST	Timer5 Clock
$\overline{\text{U1CTS}}$	12	17	14	44	48	12	17	14	44	48	I	ST	UART1 Clear-to-Send Input
$\overline{\text{U1RTS}}$	13	18	15	1	1	13	18	15	1	1	O	—	UART1 Request-to-Send Output
U1RX	6	6	3	2	2	6	6	3	2	2	I	ST	UART1 Receive
U1TX	11	16	13	3	3	11	16	13	3	3	O	—	UART1 Transmit
$\overline{\text{U2CTS}}$	10	12	9	34	37	10	12	9	34	37	I	ST	UART2 Clear-to-Send Input
$\overline{\text{U2RTS}}$	9	11	8	33	36	9	11	8	33	36	O	—	UART2 Request-to-Send Output
U2RX	5	5	2	22	24	5	5	2	22	24	I	ST	UART2 Receive
U2TX	4	4	1	21	23	4	4	1	21	23	O	—	UART2 Transmit
ULPWU	4	4	1	21	23	4	4	1	21	23	I	ANA	Ultra Low-Power Wake-up Input
VCAP	—	—	—	—	—	14	20	17	7	7	P	—	Core Power
VDD	20	28,13	25,10	17,28,40	18,30,43	20	28,13	25,10	17,28,40	18,30,43	P	—	Device Digital Supply Voltage
VREF+	2	2	27	19	21	2	2	27	19	21	I	ANA	A/D Reference Voltage Input (+)
VREF-	3	3	28	20	22	3	3	28	20	22	I	ANA	A/D Reference Voltage Input (-)
VSS	19	27,8	24,5	16,29,39	17,31,42	19	27,8	24,5	16,29,39	17,31,42	P	—	Device Digital Ground Return

2.0 GUIDELINES FOR GETTING STARTED WITH 16-BIT MICROCONTROLLERS

2.1 Basic Connection Requirements

Getting started with the PIC24FV32KA304 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"**)
- $\overline{\text{MCLR}}$ pin (see **Section 2.3 "Master Clear (MCLR) Pin"**)
- VCAP pins (see **Section 2.4 "Voltage Regulator Pin (VCAP)"**)

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"**)

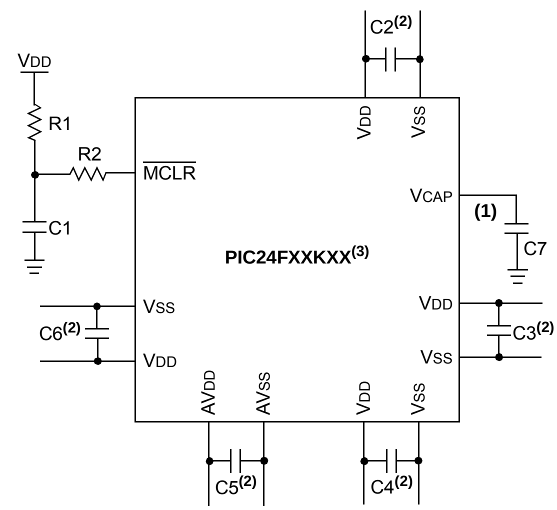
Additionally, the following pins may be required:

- VREF+/VREF- pins are 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, 16V tantalum or ceramic

R1: 10 kΩ

R2: 100Ω to 470Ω

Note 1: See **Section 2.4 "Voltage Regulator Pin (VCAP)"** for explanation of VCAP 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.

3: Some PIC24F K parts do not have a regulator.

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.

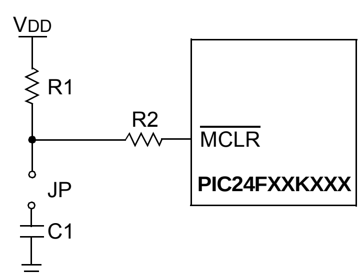
2.3 Master Clear ($\overline{\text{MCLR}}$) Pin

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 Pin (VCAP)

Note:	This section applies only to PIC24F K devices with an On-Chip Voltage Regulator.

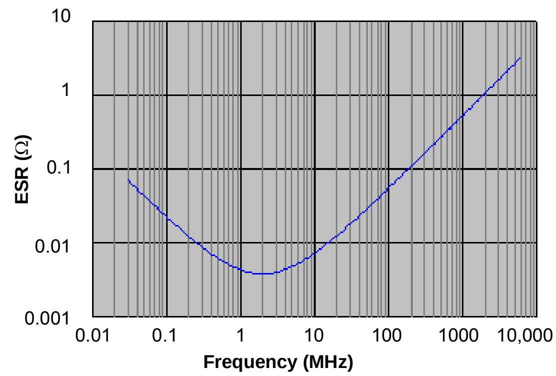
Some of the PIC24F K devices have an internal Voltage Regulator. These devices have the Voltage Regulator output brought out on the VCAP pin. On the PIC24F K devices with regulators, a low-ESR (< 5Ω) capacitor is required on the VCAP pin to stabilize the Voltage Regulator output. The VCAP pin must not be connected to VDD and must use a capacitor of 10 µF connected to ground. The type can be ceramic or tantalum. Suitable examples of capacitors are shown in Table 2-1. Capacitors with equivalent specifications can be used.

Designers may use Figure 2-3 to evaluate ESR equivalence of candidate devices.

The placement of this capacitor should be close to VCAP. It is recommended that the trace length not exceed 0.25 inch (6 mm). Refer to **Section 29.0 "Electrical Characteristics"** for additional information.

Refer to **Section 29.0 "Electrical Characteristics"** for information on VDD and VDDCORE.

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



Note: Typical data measurement at 25°C, 0V DC bias.

TABLE 2-1: SUITABLE CAPACITOR EQUIVALENTS

Make	Part #	Nominal Capacitance	Base Tolerance	Rated Voltage	Temp. Range
TDK	C3216X7R1C106K	10 µF	±10%	16V	-55 to 125ºC
TDK	C3216X5R1C106K	10 µF	±10%	16V	-55 to 85ºC
Panasonic	ECJ-3YX1C106K	10 µF	±10%	16V	-55 to 125ºC
Panasonic	ECJ-4YB1C106K	10 µF	±10%	16V	-55 to 85ºC
Murata	GRM32DR71C106KA01L	10 µF	±10%	16V	-55 to 125ºC
Murata	GRM31CR61C106KC31L	10 µF	±10%	16V	-55 to 85ºC

2.4.1 CONSIDERATIONS FOR CERAMIC CAPACITORS

In recent years, large value, low-voltage, surface-mount ceramic capacitors have become very cost effective in sizes up to a few tens of microfarad. The low-ESR, small physical size and other properties make ceramic capacitors very attractive in many types of applications.

Ceramic capacitors are suitable for use with the internal Voltage Regulator of this microcontroller. However, some care is needed in selecting the capacitor to ensure that it maintains sufficient capacitance over the intended operating range of the application.

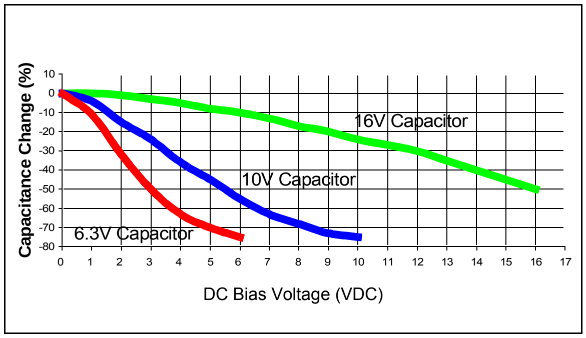
Typical low-cost, 10 μF ceramic capacitors are available in X5R, X7R and Y5V dielectric ratings (other types are also available, but are less common). The initial tolerance specifications for these types of capacitors are often specified as ±10% to ±20% (X5R and X7R), or -20%/+80% (Y5V). However, the effective capacitance that these capacitors provide in an application circuit will also vary based on additional factors, such as the applied DC bias voltage and the temperature. The total in-circuit tolerance is, therefore, much wider than the initial tolerance specification.

The X5R and X7R capacitors typically exhibit satisfactory temperature stability (ex: ±15% over a wide temperature range, but consult the manufacturer's data sheets for exact specifications). However, Y5V capacitors typically have extreme temperature tolerance specifications of +22%/-82%. Due to the extreme temperature tolerance, a 10 μF nominal rated Y5V type capacitor may not deliver enough total capacitance to meet minimum internal Voltage Regulator stability and transient response requirements. Therefore, Y5V capacitors are not recommended for use with the internal regulator if the application must operate over a wide temperature range.

In addition to temperature tolerance, the effective capacitance of large value ceramic capacitors can vary substantially, based on the amount of DC voltage applied to the capacitor. This effect can be very significant, but is often overlooked or is not always documented.

A typical DC bias voltage vs. capacitance graph for X7R type capacitors is shown in Figure 2-4.

FIGURE 2-4: DC BIAS VOLTAGE vs. CAPACITANCE CHARACTERISTICS



When selecting a ceramic capacitor to be used with the internal Voltage Regulator, it is suggested to select a high-voltage rating, so that the operating voltage is a small percentage of the maximum rated capacitor voltage. For example, choose a ceramic capacitor rated at 16V for the 3.3V or 2.5V core voltage. Suggested capacitors are shown in Table 2-1.

2.5 ICSP Pins

The PGC and PGD 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 PGC and PGD 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 Voltage Input High (V_{IH}) pin and Voltage Input Low (V_{IL}) pin requirements.

For device emulation, ensure that the "Communication Channel Select" (i.e., PGCx/PGDx 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 27.0 "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 9.0 "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 2-5. 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).

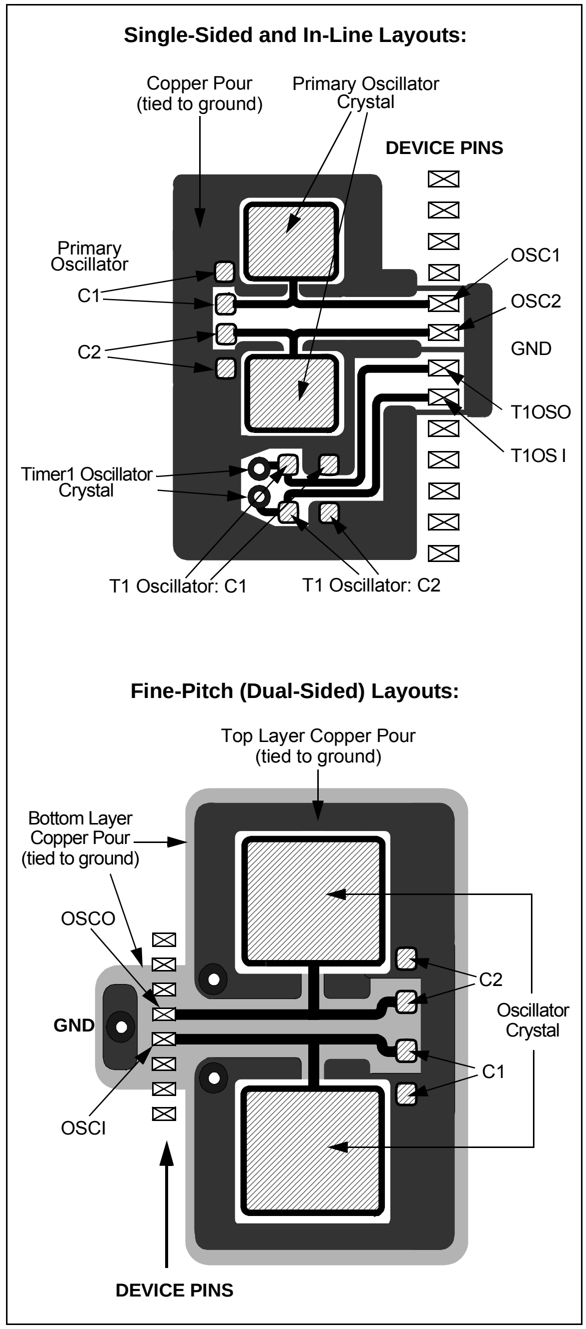
For additional information and design guidance on oscillator circuits, please refer to these Microchip Application Notes, available at the corporate web site (www.microchip.com):

- *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"*

2.7 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.

FIGURE 2-5: SUGGESTED PLACEMENT OF THE OSCILLATOR CIRCUIT



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 on the CPU, refer to the *"dsPIC33/PIC24 Family Reference Manual"*, **"CPU"** (DS39703).

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 16^{th} Working register (W15) operates as a Software Stack Pointer (SSP) for interrupts and calls.

The upper 32 Kbytes of the data space memory map can optionally be mapped into program space at any 16K word boundary of either program memory or data EEPROM memory, defined by the 8-bit Program Space Visibility Page Address (PSVPAG) register. The program to data space mapping feature lets any instruction access program space as if it were data space.

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 and 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 (i.e., A + B = C) to be executed in a single cycle.

A high-speed, 17-bit by 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 by 16-bit or 8-bit by 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 eight 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 illustrated in Figure 3-1.

3.1 Programmer's Model

Figure 3-2 displays the programmer's model for the PIC24F. All registers in the programmer's model are memory mapped and can be manipulated directly by instructions.

Table 3-1 provides a description of each register. All registers associated with the programmer's model are memory mapped.

FIGURE 3-1: PIC24F CPU CORE BLOCK DIAGRAM

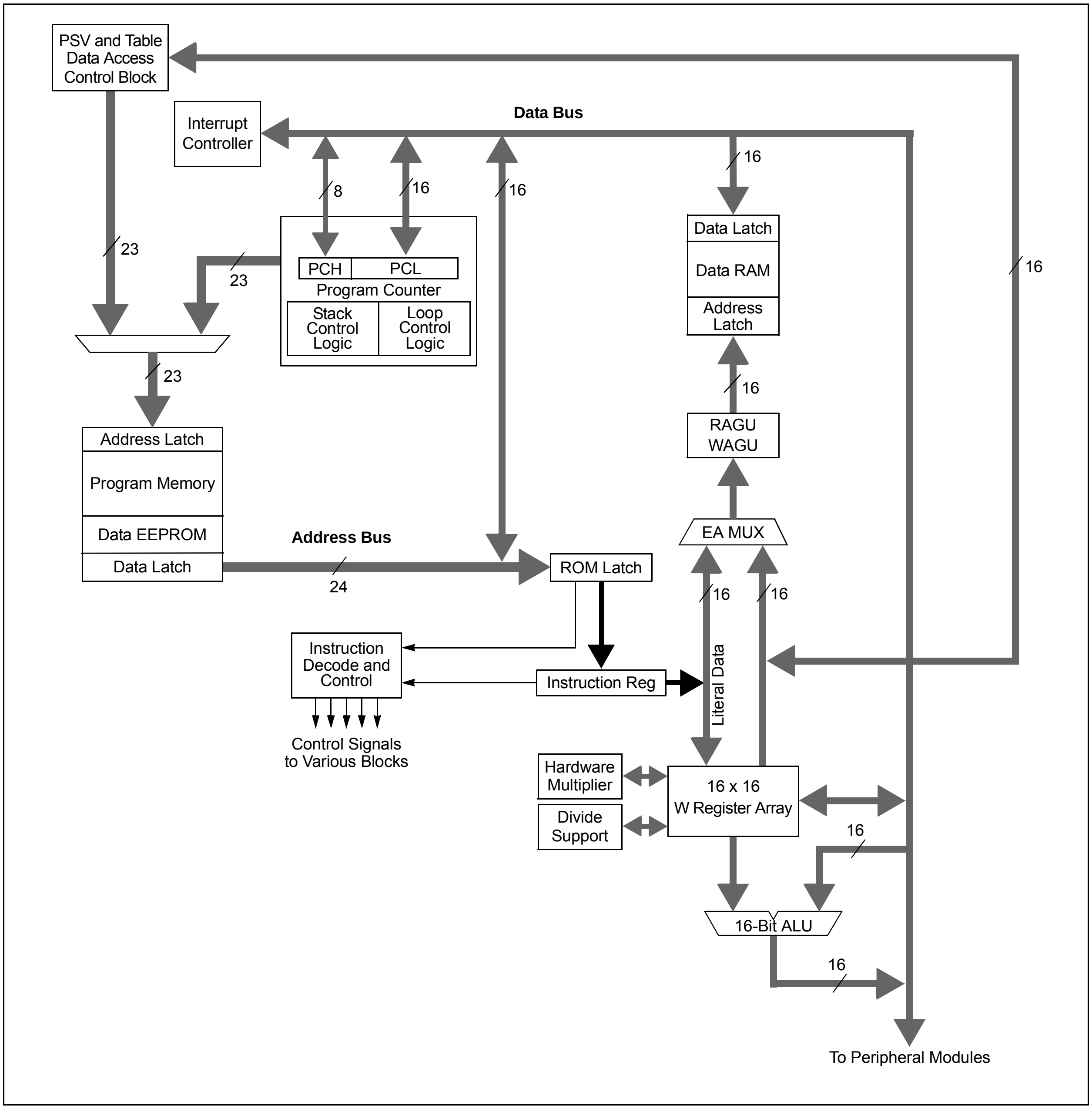
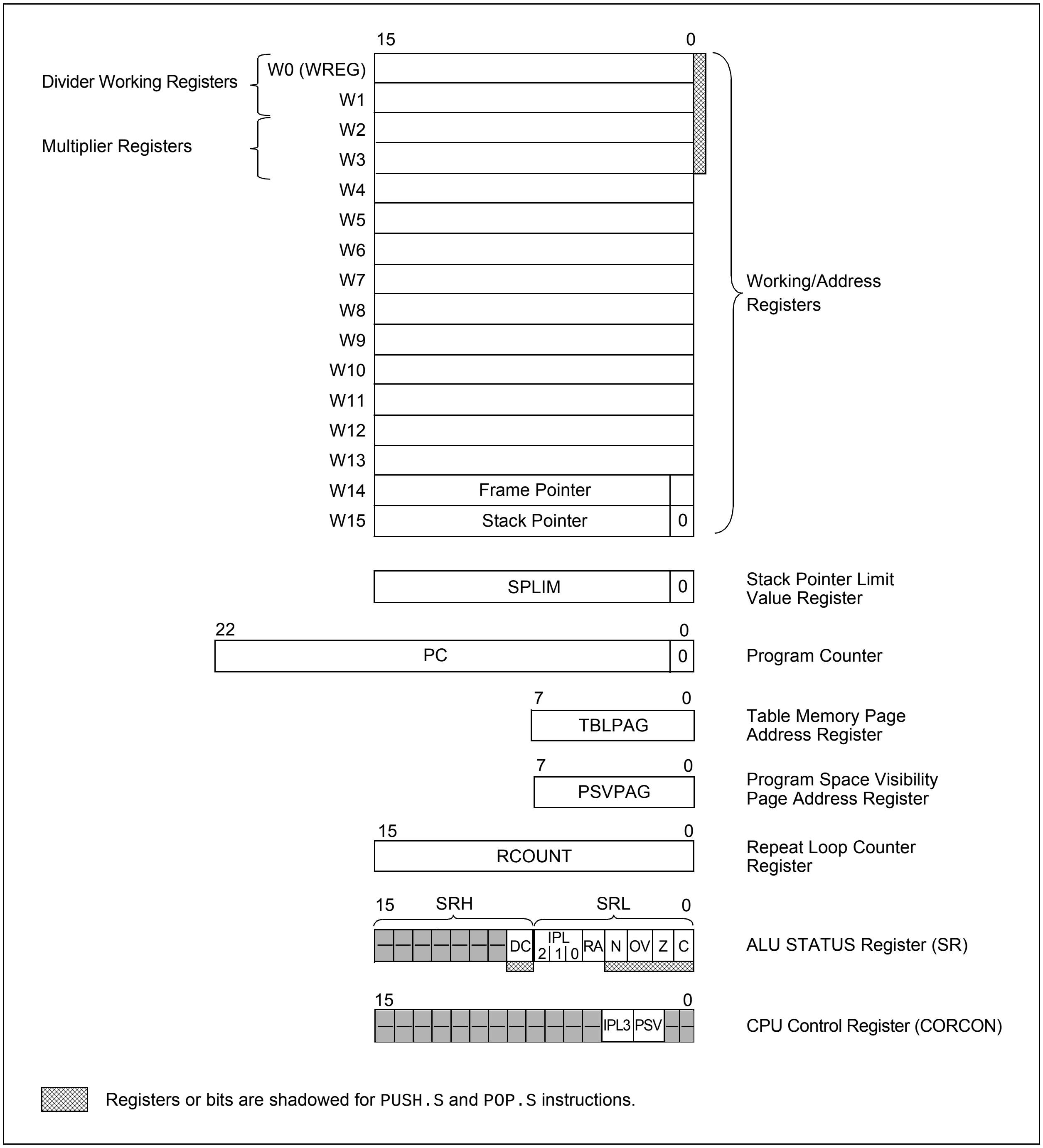


TABLE 3-1: CPU CORE REGISTERS

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
PSVPAG	Program Space Visibility Page Address Register
RCOUNT	Repeat Loop Counter Register
CORCON	CPU Control 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 non-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 effects the Z bit, has set it at some time in the past
0 = The most recent operation, which effects 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 (MSb) of the result occurred
0 = No carry-out from the Most Significant bit (MSb) of the result occurred

Note 1: The IPLx Status bits are read-only when NSTDIS (INTCON1<15>) = 1.
2: The IPL<2:0> Status bits are concatenated with the IPL3 bit (CORCON<3>) 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/W-0	U-0	U-0
—	—	—	—	IPL3(1)	PSV	—	—
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

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	**PSV:** Program Space Visibility in Data Space Enable bit 1 = Program space is visible in data space 0 = Program space is not visible in data space
bit 1-0	**Unimplemented:** Read as ‘0’

Note 1: User interrupts are disabled when IPL3 = 1.

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 division for 16-bit divisor.

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:

- 16-bit x 16-bit signed
- 16-bit x 16-bit unsigned
- 16-bit signed x 5-bit (literal) unsigned
- 16-bit unsigned x 16-bit unsigned
- 16-bit unsigned x 5-bit (literal) unsigned
- 16-bit unsigned x 16-bit signed
- 8-bit unsigned x 8-bit unsigned

3.3.2 DIVIDER

The divide block supports 32-bit/16-bit and 16-bit/16-bit 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. Sixteen-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 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, the PIC24F microcontrollers feature separate program and data memory space and busing. This architecture also allows the direct access of program memory from the data space during code execution.

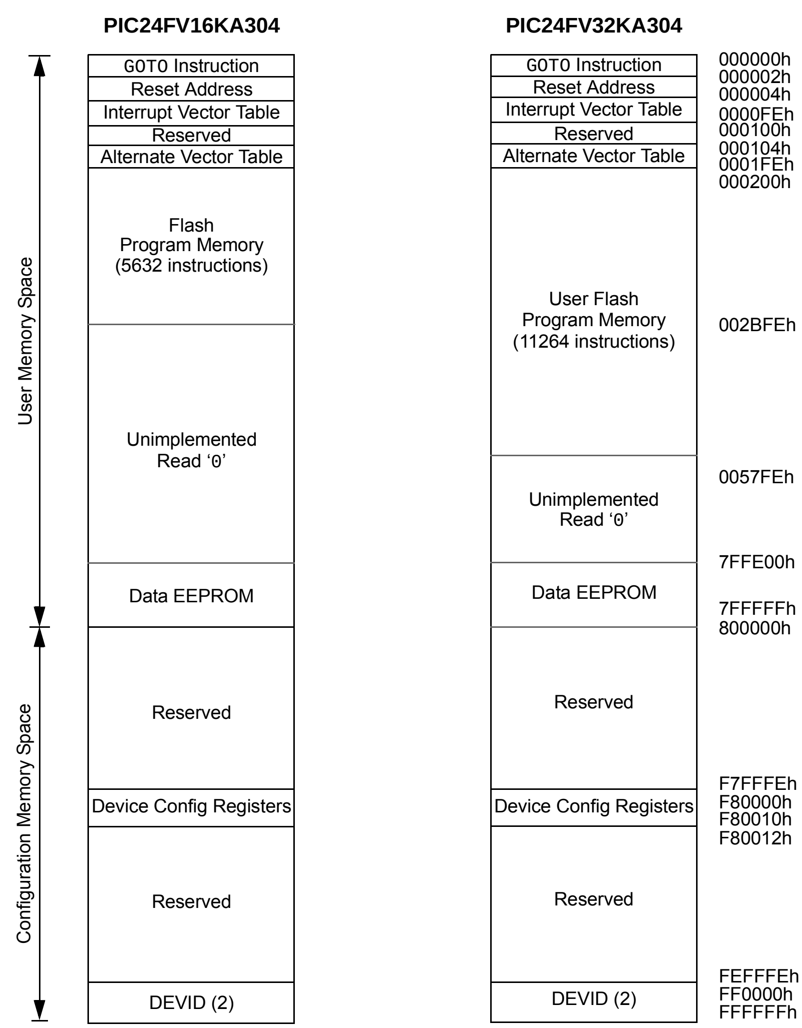
4.1 Program Address Space

The program address memory space of the PIC24FV32KA304 family 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 a 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 PIC24FV32KA304 family of devices are shown in Figure 4-1.

FIGURE 4-1: PROGRAM SPACE MEMORY MAP FOR PIC24FV32KA304 FAMILY DEVICES



Note: Memory areas are not displayed 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, as shown in 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 00000h and 000200h 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 000000h, with the actual address for the start of code at 000002h.

PIC24F devices also have two Interrupt Vector Tables, located from 000004h to 0000FFh and 000104h to 0001FFh. These vector tables allow each of the many device interrupt sources to be handled by separate Interrupt Service Routines (ISRs). A more detailed discussion of the Interrupt Vector Tables (IVT) is provided in **Section 8.1 "Interrupt Vector Table (IVT)"**.

4.1.3 DATA EEPROM

In the PIC24FV32KA304 family, the data EEPROM is mapped to the top of the user program memory space, starting at address, 7FFE00, and expanding up to address, 7FFFFF.

The data EEPROM is organized as 16-bit wide memory and 256 words deep. This memory is accessed using Table Read and write operations similar to the user code memory.

4.1.4 DEVICE CONFIGURATION WORDS

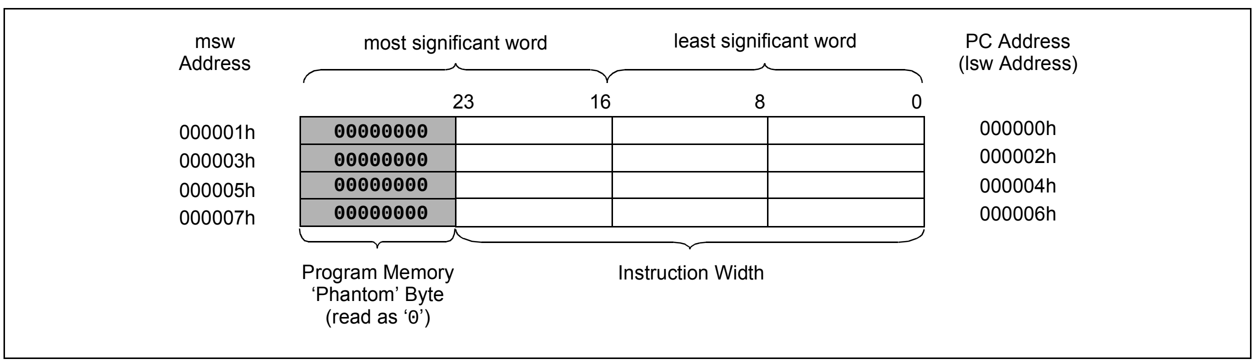
Table 4-1 provides the addresses of the device Configuration Words for the PIC24FV32KA304 family. Their location in the memory map is shown in Figure 4-1.

For more information on device Configuration Words, see **Section 26.0 "Special Features"**.

TABLE 4-1: DEVICE CONFIGURATION WORDS FOR PIC24FV32KA304 FAMILY DEVICES

Configuration Words	Configuration Word Addresses
FBS	F80000
FGS	F80004
FOSCSEL	F80006
FOSC	F80008
FWDT	F8000A
FPOR	F8000C
FICD	F8000E
FDS	F80010

FIGURE 4-2: PROGRAM MEMORY ORGANIZATION



4.2 Data Address Space

The PIC24F core has a separate, 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.

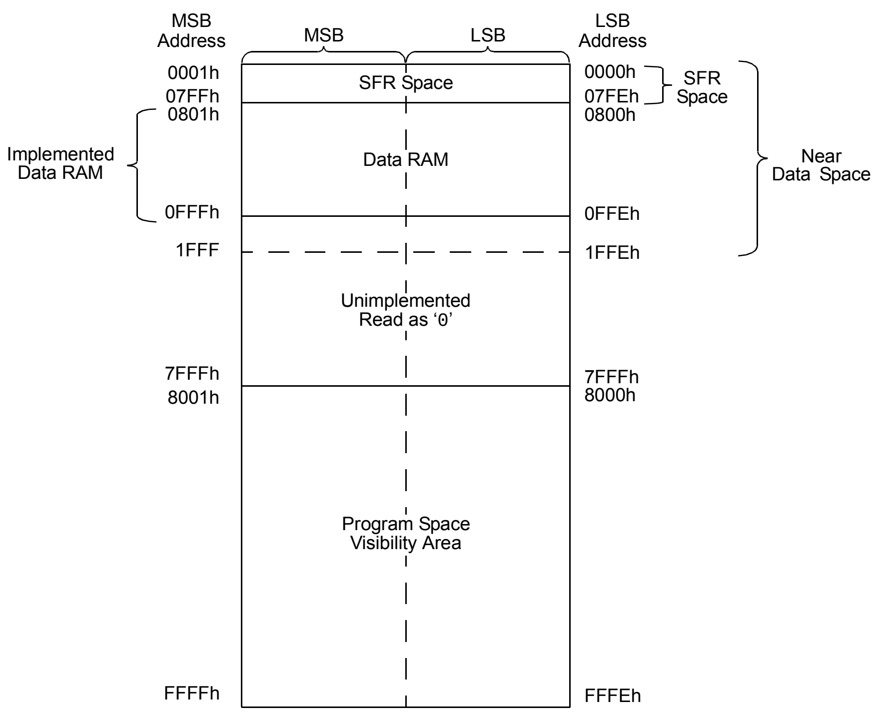
All Effective Addresses (EAs) in the data memory space are 16 bits wide and point to bytes within the data space. This gives a data space address range of 64 Kbytes or 32K words. The lower half of the data memory space (that is, when EA<15> = 0) is used for implemented memory addresses, while the upper half (EA<15> = 1) is reserved for the Program Space Visibility (PSV) area (see **Section 4.3.3 "Reading Data from Program Memory Using Program Space Visibility"**).

PIC24FV32KA304 family devices implement a total of 1024 words of data memory. If an EA points to a location outside of this area, an all zero word or byte will be returned.

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 the 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 PIC24FV32KA304 FAMILY DEVICES



Note: Data memory areas are not shown to scale.

4.2.2 DATA MEMORY ORGANIZATION AND ALIGNMENT

To maintain backward compatibility with PIC® MCU devices and improve data space memory usage efficiency, the PIC24F instruction set supports both word and byte operations. As a consequence of byte accessibility, all Effective Address (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 the 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 MSB is not modified.

A Sign-Extend (`SE`) instruction is provided to allow the 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 (NDS). 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 (MDA) with a 16-bit address field. For PIC24FV32KA304 family devices, the entire implemented data memory lies in Near Data Space.

4.2.4 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 the module. Much of the SFR space contains unused addresses; these are read as '`0`'. The SFR space, where the SFRs are actually implemented, is provided in Table 4-2. Each implemented area indicates a 32-byte region, where at least one address is implemented as an SFR. A complete listing of implemented SFRs, including their addresses, is provided in Table 4-3 through Table 4-25.

TABLE 4-2: 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	I^2C	UART	SPI		—	—	I/O	
300h	A/D/CMTU				—	—	—	—
400h	—	—	—	—	—	—	—	—
500h	—	—	—	—	—	—	—	—
600h	—	RTC/Comp	CRC	—	—			
700h	—	—	System/DS/HLVD	NVM/PMD	—	—	—	—

Legend: — = No implemented SFRs in this block.

TABLE 4-3: CPU CORE REGISTERS MAP

File Name	Start 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	WREG0																0000
WREG1	0002	WREG1																0000
WREG2	0004	WREG2																0000
WREG3	0006	WREG3																0000
WREG4	0008	WREG4																0000
WREG5	000A	WREG5																0000
WREG6	000C	WREG6																0000
WREG7	000E	WREG7																0000
WREG8	0010	WREG8																0000
WREG9	0012	WREG9																0000
WREG10	0014	WREG10																0000
WREG11	0016	WREG11																0000
WREG12	0018	WREG12																0000
WREG13	001A	WREG13																0000
WREG14	001C	WREG14																0000
WREG15	001E	WREG15																0000
SPLIM	0020	SPLIM																xxxx
PCL	002E	PCL																0000
PCH	0030	—	—	—	—	—	—	—	—	—	PCH							0000
TBLPAG	0032	—	—	—	—	—	—	—	—	TBLPAG								0000
PSVPAG	0034	—	—	—	—	—	—	—	—	PSVPAG								0000
RCOUNT	0036	RCOUNT																xxxxx
SR	0042	—	—	—	—	—	—	—	DC	IPL2	IPL1	IPL0	RA	N	OV	Z	C	0000
CORCON	0044	—	—	—	—	—	—	—	—	—	—	—	—	IPL3	PSV	—	—	0000
DISICNT	0052	—	—	DISICNT														xxxx

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

TABLE 4-4: 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[1]	CN14PDE	CN13PDE	CN12PDE	CN11PDE	CN10PDE[1,2]	CN9PDE[1]	CN8PDE[3]	CN7PDE[1]	CN6PDE	CN5PDE	CN4PDE	CN3PDE	CN2PDE	CN1PDE	CN0PDE	0000
CNPD2	0058	CN31PDE[1,2]	CN30PDE	CN29PDE	CN28PDE[1,2]	CN27PDE[1]	CN26PDE[1,2]	CN25PDE[1,2]	CN24PDE[1]	CN23PDE	CN22PDE	CN21PDE	CN20PDE[1,2]	CN19PDE[1,2]	CN18PDE[1,2]	CN17PDE[1,2]	CN16PDE[1]	0000
CNPD3	005A	—	—	—	—	—	—	—	—	—	—	—	CN36PDE[1,2]	CN35PDE[1,2]	CN34PDE[1,2]	CN33PDE[1,2]	CN32PDE[1,2]	0000
CNEN1	0062	CN15IE[1]	CN14IE	CN13IE	CN12IE	CN11IE	CN10IE[1,2]	CN9IE[1]	CN8IE[3]	CN7IE[1]	CN6IE	CN5IE	CN4IE	CN3IE	CN2IE	CN1IE	CN0IE	0000
CNEN2	0064	CN31IE[1,2]	CN30IE	CN29IE	CN28IE[1,2]	CN27IE[1]	CN26IE[1,2]	CN25IE[1,2]	CN24IE[1]	CN23IE	CN22IE	CN21IE	CN20IE[1,2]	CN19IE[1,2]	CN18IE[1,2]	CN17IE[1,2]	CN16IE[1]	0000
CNEN3	0066	—	—	—	—	—	—	—	—	—	—	—	CN36IE[1,2]	CN35IE[1,2]	CN34IE[1,2]	CN33IE[1,2]	CN32IE[1,2]	0000
CNPU1	006E	CN15PUE[1]	CN14PUE	CN13PUE	CN12PUE	CN11PUE	CN10PUE[1,2]	CN9PUE[1]	CN8PUE[3]	CN7PUE[1]	CN6PUE	CN5PUE	CN4PUE	CN3PUE	CN2PUE	CN1PUE	CN0PUE	0000
CNPU2	0070	CN31PUE[1,2]	CN30PUE	CN29PUE	CN28PUE[1,2]	CN27PUE[1]	CN26PUE[1,2]	CN25PUE[1,2]	CN24PUE[1]	CN23PUE	CN22PUE	CN21PUE	CN20PUE[1,2]	CN19PUE[1,2]	CN18PUE[1,2]	CN17PUE[1,2]	CN16PUE[1]	0000
CNPU3	0072	—	—	—	—	—	—	—	—	—	—	—	CN36PUE[1,2]	CN35PUE[1,2]	CN34PUE[1,2]	CN33PUE[1,2]	CN32PUE[1,2]	0000

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

Note 1: These bits are not implemented in 20-pin devices.

2: These bits are not implemented in 28-pin devices.

3: These bits are not implemented in FV devices.

TABLE 4-5: 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	—	—	—	—	—	—	—	—	—	—	—	INT2EP	INT1EP	INT0EP	0000
IFS0	0084	NVMIF	—	AD1IF	U1TXIF	U1RXIF	SPI1IF	SPF1IF	T3IF	T2IF	OC2IF	IC2IF	—	T1IF	OC1IF	IC1IF	INT0IF	0000
IFS1	0086	U2TXIF	U2RXIF	INT2IF	T5IF	T4IF	—	OC3IF	—	—	—	—	INT1IF	CNIF	CMIF	MI2C1IF	SI2C1IF	0000
IFS2	0088	—	—	—	—	—	—	—	—	—	—	IC3IF	—	—	—	SPI2IF	SPF2IF	0000
IFS3	008A	—	RTCIF	—	—	—	—	—	—	—	—	—	—	—	MI2C2IF	SI2C2IF	—	0000
IFS4	008C	—	—	CTMUIF	—	—	—	—	HLVDIF	—	—	—	—	CRCIF	U2ERIF	U1ERIF	—	0000
IFS5	008E	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ULPWUIF	0000
IEC0	0094	NVMIE	—	AD1IE	U1TXIE	U1RXIE	SPI1IE	SPF1IE	T3IE	T2IE	OC2IE	IC2IE	—	T1IE	OC1IE	IC1IE	INT0IE	0000
IEC1	0096	U2TXIE	U2RXIE	INT2IE	T5IE	T4IE	—	OC3IE	—	—	—	—	INT1IE	CNIE	CMIE	MI2C1IE	SI2C1IE	0000
IEC2	0098	—	—	—	—	—	—	—	—	—	—	IC3IE	—	—	—	SPI2IE	SPF2IE	0000
IEC3	009A	—	RTCIE	—	—	—	—	—	—	—	—	—	—	—	MI2C2IE	SI2C2IE	—	0000
IEC4	009C	—	—	CTMUIE	—	—	—	—	HLVDIE	—	—	—	—	CRCIE	U2ERIE	U1ERIE	—	0000
IEC5	009E	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ULPWUIE	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	—	—	—	—	4444
IPC2	00A8	—	U1RXIP2	U1RXIP1	U1RXIP0	—	SPI1IP2	SPI1IP1	SPI1IP0	—	SPF1IP2	SPF1IP1	SPF1IP0	—	T3IP2	T3IP1	T3IP0	4444
IPC3	00AA	—	NVMIP2	NVMIP1	NVMIP0	—	—	—	—	—	AD1IP2	AD1IP1	AD1IP0	—	U1TXIP2	U1TXIP1	U1TXIP0	4044
IPC4	00AC	—	CNIP2	CNIP1	CNIP0	—	CMIP2	CMIP1	CMIP0	—	MI2C1P2	MI2C1P1	MI2C1P0	—	SI2C1P2	SI2C1P1	SI2C1P0	4444
IPC5	00AE	—	—	—	—	—	—	—	—	—	—	—	—	—	INT1IP2	INT1IP1	INT1IP0	0004
IPC6	00B0	—	T4IP2	T4IP1	T4IP0	—	—	—	—	—	OC3IP2	OC3IP1	OC3IP0	—	—	—	—	4040
IPC7	00B2	—	U2TXIP2	U2TXIP1	U2TXIP0	—	U2RXIP2	U2RXIP1	U2RXIP0	—	INT2IP2	INT2IP1	INT2IP0	—	T5IP2	T5IP1	T5IP0	4440
IPC8	00B4	—	—	—	—	—	—	—	—	—	SPI2IP2	SPI2IP1	SPI2IP0	—	SPF2IP2	SPF2IP1	SPF2IP0	0044
IPC9	00B6	—	—	—	—	—	—	—	—	—	IC3IP2	IC3IP1	IC3IP0	—	—	—	—	0040
IPC12	00BC	—	—	—	—	—	MI2C2IP2	MI2C2IP1	MI2C2IP0	—	SI2C2IP2	SI2C2IP1	SI2C2IP0	—	—	—	—	0440
IPC15	00C2	—	—	—	—	—	RTCIP2	RTCIP1	RTCIP0	—	—	—	—	—	—	—	—	0400
IPC16	00C4	—	CRCIP2	CRCIP1	CRCIP0	—	U2ERIP2	U2ERIP1	U2ERIP0	—	U1ERIP2	U1ERIP1	U1ERIP0	—	—	—	—	4440
IPC18	00C8	—	—	—	—	—	—	—	—	—	—	—	—	—	HLVDIP2	HLVDIP1	HLVDIP0	0004
IPC19	00CA	—	—	—	—	—	—	—	—	—	CTMUIP2	CTMUIP1	CTMUIP0	—	—	—	—	0040
IPC20	00CC	—	—	—	—	—	—	—	—	—	—	—	—	—	ULPWUIP2	ULPWUIP1	ULPWUIP0	0000
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.

TABLE 4-6: 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	TMR1																0000
PR1	0102	PR1																FFFF
T1CON	0104	TON	—	TSIDL	—	—	—	T1ECS1	T1ECS0	—	TGATE	TCKPS1	TCKPS0	—	TSYNC	TCS	—	0000
TMR2	0106	TMR2																0000
TMR3HLD	0108	TMR3HLD																0000
TMR3	010A	TMR3																0000
PR2	010C	PR2																0000
PR3	010E	PR3																FFFF
T2CON	0110	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	T32	—	TCS	—	FFFF
T3CON	0112	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	—	TCS	—	0000
TMR4	0114	TMR4																0000
TMR5HLD	0116	TMR5HLD																0000
TMR5	0118	TMR5																0000
PR4	011A	PR4																FFFF
PR5	011C	PR5																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 4-7: 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	IC1BUF																0000
IC1TMR	0146	IC1TMR																xxxx
IC2CON1	0148	—	—	ICSIDL	IC2TSEL2	IC2TSEL1	IC2TSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC2CON2	014A	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D
IC2BUF	014C	IC2BUF																0000
IC2TMR	014E	IC2TMR																xxxx
IC3CON1	0150	—	—	ICSIDL	IC3TSEL2	IC3TSEL1	IC3TSEL0	—	—	—	ICI1	ICI0	ICOV	ICBNE	ICM2	ICM1	ICM0	0000
IC3CON2	0152	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000D
IC3BUF	0154	IC3BUF																0000
IC3TMR	0156	IC3TMR																xxxx

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

TABLE 4-8: 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	OC1RS																0000
OC1R	0196	OC1R																0000
OC1TMR	0198	OC1TMR																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	OC2RS																0000
OC2R	01A0	OC2R																0000
OC2TMR	01A2	OC2TMR																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	OC3RS																0000
OC3R	01AA	OC3R																0000
OC3TMR	01AC	OC3TMR																xxxx

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

TABLE 4-9: I2Cx 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	—	—	—	—	—	—	—	—	I2CRCV								0000
I2C1TRN	0202	—	—	—	—	—	—	—	—	I2CTRN								00FF
I2C1BRG	0204	—	—	—	—	—	—	—	—	I2CBRG								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	—	—	—	—	—	—	I2CADD										0000
I2C1MSK	020C	—	—	—	—	—	—	AMSK<9:0>										0000
I2C2RCV	0210	—	—	—	—	—	—	—	—	I2CRCV								0000
I2C2TRN	0212	—	—	—	—	—	—	—	—	I2CTRN								00FF
I2C2BRG	0214	—	—	—	—	—	—	—	—	I2CBRG								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	—	—	—	—	—	—	I2CADD										0000
I2C2MSK	021C	—	—	—	—	—	—	AMSK<9:0>										0000

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

TABLE 4-10: UARTx 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
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	—	—	—	—	—	—	—	U1TXREG									xxxx
U1RXREG	0226	—	—	—	—	—	—	—	U1RXREG									0000
U1BRG	0228	BRG																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	—	—	—	—	—	—	—	U2TXREG									xxxx
U2RXREG	0236	—	—	—	—	—	—	—	U2RXREG									0000
U2BRG	0238	BRG																0000

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

TABLE 4-11: SPIx 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
SPI1STAT	0240	SPIEN	—	SPISIDL	—	—	SPIBEC2	SPIBEC1	SPIBEC0	SRMPT	SPIROV	SR1MPT	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	SPI1BUF																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	SPI2BUF																0000

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

TABLE 4-12: PORTA REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11[2,3]	Bit 10[2,3]	Bit 9[2,3]	Bit 8[2,3]	Bit 7[2]	Bit 6[4]	Bit 5[1]	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
TRISA	02C0	—	—	—	—	TRISA<11:6>						—	TRISA<4:0>					00DF
PORTA	02C2	—	—	—	—	RA<11:0>												xxxx
LATA	02C4	—	—	—	—	LATA<11:6>						—	LATA<4:0>					xxxx
ODCA	02C6	—	—	—	—	ODA<11:6>						—	ODA<4:0>					0000

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

Note 1: This bit is available only when MCLRE = 1.
2: These bits are not implemented in 20-pin devices.
3: These bits are not implemented in 28-pin devices.
4: These bits are not implemented in FV devices.

TABLE 4-13: PORTB REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11[1]	Bit 10[1]	Bit 9	Bit 8	Bit 7	Bit 6[1]	Bit 5[1]	Bit 4	Bit 3[1]	Bit 2	Bit 1	Bit 0	All Resets
TRISB	02C8	TRISB<15:0>																FFFF
PORTB	02CA	RB<15:0>																xxxx
LATB	02CC	LATB<15:0>																xxxx
ODCB	02CE	ODB<15:0>																0000

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

Note 1: These bits are not implemented in 20-pin devices.

TABLE 4-14: PORTC 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
TRISC	02D0	—	—	—	—	—	—	TRISC<9:0>										03FF
PORTC	02D2	—	—	—	—	—	—	RC<9:0>										xxxx
LATC	02D4	—	—	—	—	—	—	LATC<9:0>										xxxx
ODCC	02D6	—	—	—	—	—	—	ODC<9:0>										0000

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

Note 1: PORTC is not implemented in 20-pin devices or 28-pin devices.

TABLE 4-15: 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	—	—	—	—	—	—	—	—	—	—	SMBUSDEL<2:1>		—	—	—	—	0000

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

TABLE 4-16: A/D 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	ADC1BUF0																xxxx
ADC1BUF1	0302	ADC1BUF1																xxxx
ADC1BUF2	0304	ADC1BUF2																xxxx
ADC1BUF3	0306	ADC1BUF3																xxxx
ADC1BUF4	0308	ADC1BUF4																xxxx
ADC1BUF5	030A	ADC1BUF5																xxxx
ADC1BUF6	030C	ADC1BUF6																xxxx
ADC1BUF7	030E	ADC1BUF7																xxxx
ADC1BUF8	0310	ADC1BUF8																xxxx
ADC1BUF9	0312	ADC1BUF9																xxxx
ADC1BUF10	0314	ADC1BUF10																xxxx
ADC1BUF11	0316	ADC1BUF11																xxxx
ADC1BUF12	0318	ADC1BUF12																xxxx
ADC1BUF13	031A	ADC1BUF13																xxxx
ADC1BUF14	031C	ADC1BUF14																xxxx
ADC1BUF15	031E	ADC1BUF15																xxxx
ADC1BUF16	0320	ADC1BUF16																xxxx
ADC1BUF17	0322	ADC1BUF17																xxxx
AD1CON1	0340	ADON	—	ADSIDL	—	—	MODE12	FORM1	FORM0	SSRC3	SSRC2	SSRC1	SSRC0	—	ASAM	SAMP	DONE	0000
AD1CON2	0342	PVCFG1	PVCFG0	NVCFG0	—	BUFREGEN	CSCNA	—	—	BUFS	SMPI4	SMPI3	SMPI2	SMPI1	SMPI0	BUFM	ALTS	0000
AD1CON3	0344	ADRC	EXTSAM	—	SAMC4	SAMC3	SAMC2	SAMC1	SAMC0	ADCS7	ADCS6	ADCS5	ADCS4	ADCS3	ADCS2	ADCS1	ADCS0	0000
AD1CHS	0348	CH0NB2	CH0NB1	CH0NB0	CH0SB4	CH0SB3	CH0SB2	CH0SB1	CH0SB0	CH0NA2	CH0NA1	CH0NA0	CH0SA4	CH0SA3	CH0SA2	CH0SA1	CH0SA0	0000
AD1CSSH	034E	—	CSSL<30:26>					—	—	—	—	—	—	—	—	CSSL<17:16>		0000
AD1CSSL	0350	CSSL<15:0>																0000
AD1CON5	0354	ASEN	LPEN	CTMUREQ	BGREQ	r	—	ASINT1	ASINT0	—	—	—	—	WM1	WM0	CM1	CM0	0000
AD1CHITH	0356	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CHH<17:16>		0000
AD1CHITL	0358	CHH<15:0>																0000

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

TABLE 4-17: 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
CTMUCON1	035A	CTMUEN	—	CTMUSIDL	TGEN	EDGEN	EDGSEQEN	IDISSEN	CTTRIG	—	—	—	—	—	—	—	—	0000
CTMUCON2	035C	EDG1MOD	EDG1POL	EDG1SEL3	EDG1SEL2	EDG1SEL1	EDG1SEL0	EDG2STAT	EDG1STAT	EDG2EMOD	EDG2POL	EDG2SEL3	EDG2SEL2	EDG2SEL1	EDG2SEL0	—	—	0000
CTMUICON	035E	ITRIM5	ITRIM4	ITRIM3	ITRIM2	ITRIM1	ITRIM0	IRNG1	IRNG0	—	—	—	—	—	—	—	—	0000
AD1CTMUENH	0360	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CTMEN<17:16>		0000
AD1CTMUENL	0362	CTMEN<15:0>																0000

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

TABLE 4-18: ANALOG 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
ANSA	04E0	—	—	—	—	—	—	—	—	—	—	—	—	ANSA<3:0>				000F
ANSB	04E2	ANSB<15:12>				—	—	—	—	—	—	—	ANSB<4:0>[1]					F01F
ANSC	04E4	—	—	—	—	—	—	—	—	—	—	—	—	—	ANSC<2:0>[1,2]			0007

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

Note 1: ANSB3 and ANSC<2:0> bits are not implemented in 20-pin devices.

2: These bits are not implemented in 28-pin devices.

TABLE 4-19: 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	ALRMVAL																xxxx
ALCFGRPT	0622	ALRMEN	CHIME	AMASK3	AMASK2	AMASK1	AMASK0	ALRMPTR1	ALRMPTR0	ARPT7	ARPT6	ARPT5	ARPT4	ARPT3	ARPT2	ARPT1	ARPT0	0000
RTCVAL	0624	RTCVAL																xxxx
RCFGCAL	0626	RTCEN	—	RTCWREN	RTCSYNC	HALFSEC	RTCOE	RTCPTR1	RTCPTR0	CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0	0000
RTCPWC	0628	PWCEN	PWCPOL	PWCCPRE	PWCSPRE	RTCCLK1	RTCCLK0	RTCOUT1	RTCOUT0	—	—	—	—	—	—	—	—	xxxx

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

TABLE 4-20: TRIPLE COMPARATOR 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	xxxx
CVRCON	0632	—	—	—	—	—	—	—	—	CVREN	CVROE	CVRSS	CVR4	CVR3	CVR2	CVR1	CVR0	0000
CM1CON	0634	CON	COE	CPOL	CLPWR	—	—	CEVT	COUT	EVPOL1	EVPOL0	—	CREF	—	—	CCH1	CCH0	xxxx
CM2CON	0636	CON	COE	CPOL	CLPWR	—	—	CEVT	COUT	EVPOL1	EVPOL0	—	CREF	—	—	CCH1	CCH0	0000
CM3CON	0638	CON	COE	CPOL	CLPWR	—	—	CEVT	COUT	EVPOL1	EVPOL0	—	CREF	—	—	CCH1	CCH0	0000

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

TABLE 4-21: 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	—	—	—	0000
CRCCON2	0642	—	—	—	DWIDTH4	DWIDTH3	DWIDTH2	DWIDTH1	DWIDTH0	—	—	—	PLEN4	PLEN3	PLEN2	PLEN1	PLEN0	0000
CRCXORL	0644	X<15:1>															—	0000
CRCXORH	0646	X<31:16>																0000
CRCDATL	0648	CRCDATL																xxxx
CRCDATH	064A	CRCDATH																xxxx
CRCWDATL	064C	CRCWDATL																xxxx
CRCWDATH	064E	CRCWDATH																xxxx

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

TABLE 4-22: CLOCK CONTROL 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	SBOREN	RETEN	—	DPSLP	CM	PMSLP	EXTR	SWR	SWDTEN	WDTO	SLEEP	IDLE	BOR	POR	(Note 1)
OSCCON	0742	—	COSC2	COSC1	COSC0	—	NOSC2	NOSC1	NOSC0	CLKLOCK	—	LOCK	—	CF	SOSCDRV	SOSCEN	OSWEN	(Note 2)
CLKDIV	0744	ROI	DOZE2	DOZE1	DOZE0	DOZEN	RCDIV2	RCDIV1	RCDIV0	—	—	—	—	—	—	—	—	3140
OSCTUN	0748	—	—	—	—	—	—	—	—	—	—	TUN<5:0>						0000
REFOCON	074E	ROEN	—	ROSSLP	ROSEL	RODIV3	RODIV2	RODIV1	RODIV0	—	—	—	—	—	—	—	—	0000
HLVDCON	0756	HLVDEN	—	HLSIDL	—	—	—	—	—	VDIR	BGVST	IRVST	—	HLVDL3	HLVDL2	HLVDL1	HLVDL0	0000

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

Note 1: RCON register Reset values are dependent on the type of Reset.
2: OSCCON register Reset values are dependent on the Configuration fuses and by type of Reset.

TABLE 4-23: DEEP SLEEP 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
DSCON	0758	DSEN	—	—	—	—	—	—	RTCCWDIS	—	—	—		—	ULPWDIS	DSBOR	RELEASE	0000
DSWAKE	075A	—	—	—	—	—	—	—	DSINT0	DSFLT	—	—	DSWDT	DSRTCC	DSMCLR	—	DSPOR	0000
DSGPR0(1)	075C	DSGPR0																0000
DSGPR1(1)	075E	DSGPR1																0000

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

Note 1: The Deep Sleep registers, DSGPR0 and DSGPR1, are only reset on a VDD POR event.

TABLE 4-24: 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[1]
NVMCON	0760	WR	WREN	WRERR	PGMONLY	—	—	—	—	—	ERASE	NVMOP5	NVMOP4	NVMOP3	NVMOP2	NVMOP1	NVMOP0	0000
NVMKEY	0766	—	—	—	—	—	—	—	—	NVMKEY								0000

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

TABLE 4-25: ULTRA LOW-POWER WAKE-UP 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
ULPWCON	0768	ULPEN	—	ULPSIDL	—	—	—	—	ULPSINK	—	—	—	—	—	—	—	—	0000

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

TABLE 4-26: 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	—	—	—	—	—	IC3MD	IC2MD	IC1MD	—	—	—	—	—	OC3MD	OC2MD	OC1MD	0000
PMD3	0774	—	—	—	—	—	CMPMD	RTCCMD	—	CRCPMD	—	—	—	—	—	I2C2MD	—	0000
PMD4	0776	—	—	—	—	—	—	—	—	ULPWUMD	—	—	EEMD	REFOMD	CTMUMD	HLVDMD	—	0000

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

4.2.5 SOFTWARE STACK

In addition to its use as a Working register, the W15 register in PIC24F devices is also used as a Software Stack Pointer. The pointer always points to the first available free word and grows from lower to higher addresses. It predecrements for stack pops and post-increments for stack pushes, as shown in Figure 4-4.

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.

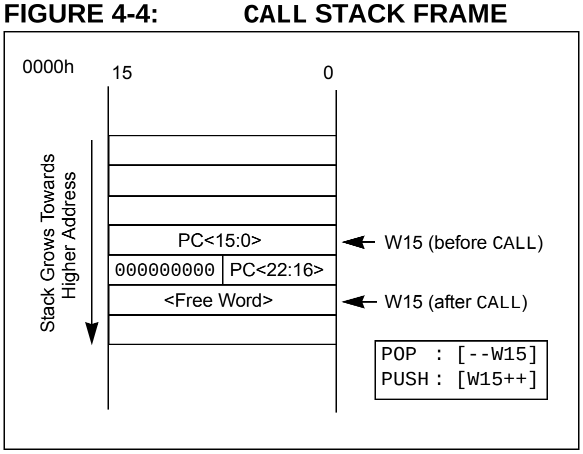
The Stack Pointer Limit Value (SPLIM) register, associated with the Stack Pointer, sets an upper address boundary for the stack. SPLIM is uninitialized at Reset. As is the case for the Stack Pointer, SPLIM<0> is forced to '`0`' as all stack operations must be word-aligned. Whenever an EA is generated, 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, 0DF6 in RAM, initialize the SPLIM with the value, 0DF4.

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 Special Function Register (SFR) space.

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

FIGURE 4-4: 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.

Apart from the normal execution, the PIC24F architecture provides two methods by which the 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, PSV

Table instructions allow an application to read or write 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 (lsw) 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 Most Significant bit (MSb) 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 8-bit Program Space Visibility Page Address register (PSVPAG) is used to define a 16K word page in the program space. When the MSb of the EA is '`1`', PSVPAG is concatenated with the lower 15 bits of the EA to form a 23-bit program space address. Unlike the table operations, this limits remapping operations strictly to the user memory area.

Table 4-27 and Figure 4-5 show how the program EA is created for table operations and remapping accesses from the data EA. Here, the P<23:0> bits refer to a program space word, whereas the D<15:0> bits refer to a data space word.

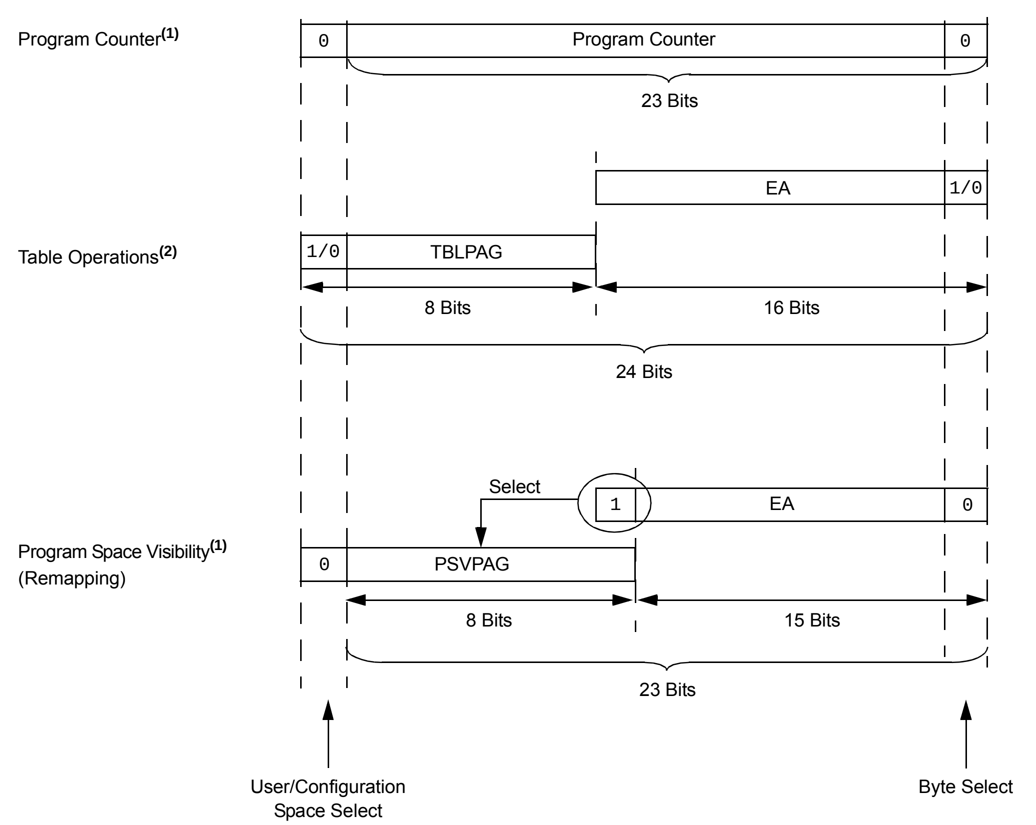
TABLE 4-27: PROGRAM SPACE ADDRESS CONSTRUCTION

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	PSVPAG<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 PSVPAG<0>.

2: PSVPAG can have only two values (‘00’ to access program memory and FF to access data EEPROM) in the PIC24FV32KA304 family.

FIGURE 4-5: DATA ACCESS FROM PROGRAM SPACE ADDRESS GENERATION



Note 1: The LSb of program space addresses is always fixed as ‘0’ in order to maintain word alignment of data in the program and data spaces.

2: Table operations are not required to be word-aligned. Table Read operations are permitted in the configuration memory space.

4.3.2 DATA ACCESS FROM PROGRAM MEMORY AND DATA EEPROM 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 memory without going through data space. It also offers a direct method of reading or writing a word of any address within data EEPROM memory. The TBLRDH and TBLWTH instructions are the only method to read or write the upper 8 bits of a program space word as data.

> **Note:** The TBLRDH and TBLWTH instructions are not used while accessing data EEPROM memory.

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'.

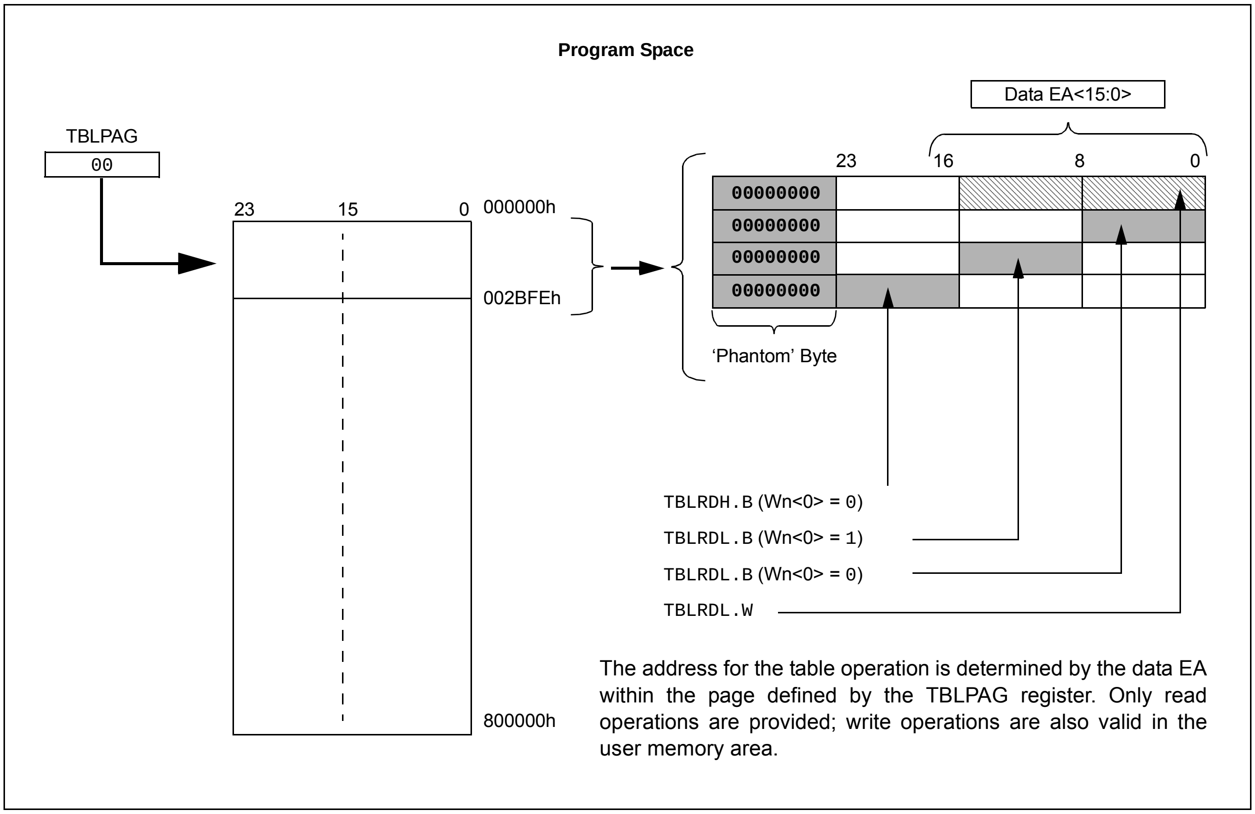
   In Byte mode, it maps the upper or lower byte of the program word to D<7:0> of the data address, as above. Note that the data will always be '0' when the upper 'phantom' byte is selected (Byte Select = 1).

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 explained 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, and only then, in implemented areas, such as the Device ID. Table Write operations are not allowed.

FIGURE 4-6: ACCESSING PROGRAM MEMORY WITH TABLE INSTRUCTIONS



4.3.3 READING DATA FROM PROGRAM MEMORY USING PROGRAM SPACE VISIBILITY

The upper 32 Kbytes of data space may optionally be mapped into a 16K word page (in PIC24FV16KA3XX devices) and a 32K word page (in PIC24FV32KA3XX devices) 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 if the MSb of the data space EA is '`1`' and PSV is enabled by setting the PSV bit in the CPU Control (CORCON<2>) register. The location of the program memory space to be mapped into the data space is determined by the Program Space Visibility Page Address (PSVPAG) register. This 8-bit register defines any one of 256 possible pages of 16K words in program space. In effect, PSVPAG functions as the upper 8 bits of the program memory address, with the 15 bits of the EA functioning as the lower bits.

By incrementing the PC by 2 for each program memory word, the lower 15 bits of data space addresses directly map to the lower 15 bits in the corresponding program space addresses.

Data reads from this area add an additional cycle to the instruction being executed, since two program memory fetches are required.

Although each data space address, 8000h and higher, maps directly into a corresponding program memory address (see Figure 4-7), only the lower 16 bits of the 24-bit program word are used to contain the data. The upper 8 bits of any program space location used as data should be programmed with '`1111 1111`' or '`0000 0000`' to force a `NOP`. This prevents possible issues should the area of code ever be accidentally executed.

Note:	PSV access is temporarily disabled during Table Reads/Writes.

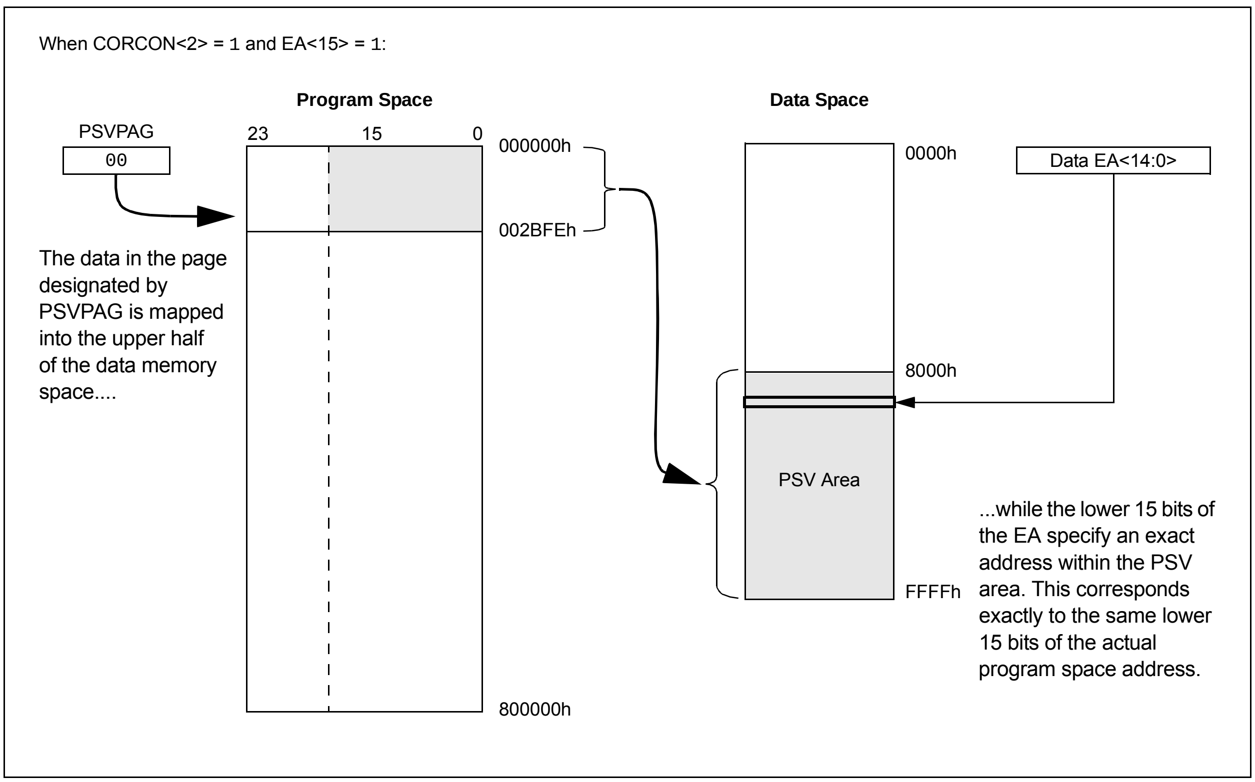
For operations that use PSV and are executed outside a `REPEAT` loop, the `MOV` and `MOV.D` instructions will require one instruction cycle in addition to the specified execution time. All other instructions will require two instruction cycles in addition to the specified execution time.

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.

FIGURE 4-7: PROGRAM SPACE VISIBILITY OPERATION



NOTES:

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 on Flash programming, refer to the *"dsPIC33/PIC24 Family Reference Manual"*, **"PIC24F Flash Program Memory"** (DS30009715).

The PIC24FV32KA304 family of devices contains internal Flash program memory for storing and executing application code. The memory is readable, writable and erasable when operating with VDD over 1.8V.

Flash memory can be programmed in three ways:

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

ICSP allows a PIC24FV32KA304 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 (which are named PGECx and PGEDx, respectively), and three other lines for power (VDD), ground (VSS) and Master Clear/Program mode Entry voltage (MCLR/VPP). 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 custom firmware to be programmed.

Run-Time Self Programming (RTSP) is accomplished using TBLRD (Table Read) and TBLWT (Table Write) instructions. With RTSP, the user may write program memory data in blocks of 32 instructions (96 bytes) at a time, and erase program memory in blocks of 32, 64 and 128 instructions (96,192 and 384 bytes) at a time.

The NVMOP<1:0> (NVMCON<1:0>) bits decide the erase block size.

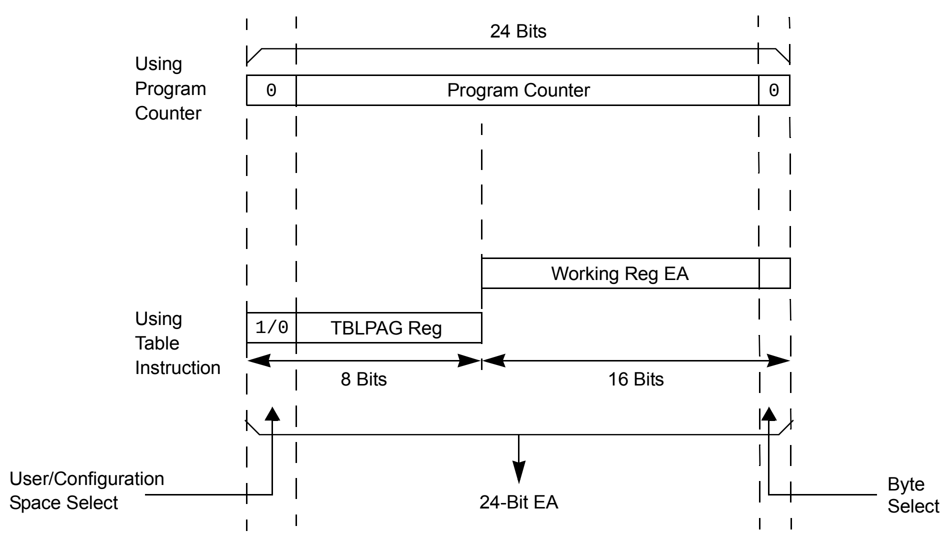
5.1 Table Instructions and Flash Programming

Regardless of the method used, Flash memory programming 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 depicted 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 32 instructions or 96 bytes. RTSP allows the user to erase blocks of 1 row, 2 rows and 4 rows (32, 64 and 128 instructions) at a time, and to program one row at a time. It is also possible to program single words.

The 1-row (96 bytes), 2-row (192 bytes) and 4-row (384 bytes) erase blocks, and single row write block (96 bytes) are edge-aligned from the beginning of program memory.

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, 32 `TBLWT` instructions are required to write the full row of memory.

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 it is not recommended.

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

5.3 Enhanced In-Circuit Serial Programming

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

5.4 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 the blocks that need 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. For more information, refer to **Section 5.5 "Programming Operations"**.

5.5 Programming Operations

A complete programming sequence is necessary for programming or erasing the internal Flash in RTSP mode. During a program 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/SO-0, HC	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
WR	WREN	WRERR	PGMONLY[4]	—	—	—	—
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
—	ERASE	NVMOP5[1]	NVMOP4[1]	NVMOP3[1]	NVMOP2[1]	NVMOP1[1]	NVMOP0[1]
bit 7							bit 0

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

bit 15 **WR:** Write Control bit
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 = Enables Flash program/erase operations
0 = Inhibits Flash program/erase operations

bit 13 **WRERR:** Write Sequence Error Flag bit
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 **PGMONLY:** Program Only Enable bit[4]

bit 11-7 **Unimplemented:** Read as '0'

bit 6 **ERASE:** Erase/Program Enable bit
1 = Performs the erase operation specified by NVMOP<5:0> on the next WR command
0 = Performs the program operation specified by NVMOP<5:0> on the next WR command

bit 5-0 **NVMOP<5:0>:** Programming Operation Command Byte bits[1]
Erase Operations (when ERASE bit is '1'):
1010xx = Erases entire boot block (including code-protected boot block)[2]
1001xx = Erases entire memory (including boot block, configuration block, general block)[2]
011010 = Erases 4 rows of Flash memory[3]
011001 = Erases 2 rows of Flash memory[3]
011000 = Erases 1 row of Flash memory[3]
0101xx = Erases entire configuration block (except code protection bits)
0100xx = Erases entire data EEPROM[4]
0011xx = Erases entire general memory block programming operations
0001xx = Writes 1 row of Flash memory (when ERASE bit is '0')[3]

Note 1: All other combinations of NVMOP<5:0> are no operation.
2: These values are available in ICSP™ mode only. Refer to the device programming specification.
3: The address in the Table Pointer decides which rows will be erased.
4: This bit is used only while accessing data EEPROM.

5.5.1 PROGRAMMING ALGORITHM FOR FLASH PROGRAM MEMORY

The user can program one row of Flash program memory at a time by erasing the programmable row. The general process is as follows:

1. Read a row of program memory (32 instructions) and store in data RAM.
2. Update the program data in RAM with the desired new data.
3. Erase a row (see Example 5-1):
   a) Set the NVMOPx bits (NVMCON<5:0>) to '011000' to configure for row 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 32 instructions from data RAM into the program memory buffers (see Example 5-1).
5. Write the program block to Flash memory:
   a) Set the NVMOPx bits to '011000' 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.

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 5-5.

EXAMPLE 5-1: ERASING A PROGRAM MEMORY ROW – ASSEMBLY LANGUAGE CODE

```
; Set up NVMCON for row erase operation
       MOV     #0x4058, W0                ;
       MOV     W0, NVMCON                 ; Initialize NVMCON
; Init pointer to row to be ERASED
       MOV     #tblpage(PROG_ADDR), W0    ;
       MOV     W0, TBLPAG                 ; Initialize 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
                                            for next 5 instructions
       MOV     #0x55, W0
       MOV     W0, NVMKEY                 ; Write the 55 key
       MOV     #0xAA, W1                  ;
       MOV     W1, NVMKEY                 ; Write the AA 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 ROW – ‘C’ LANGUAGE CODE

```
// C example using MPLAB C30

    int __attribute__ ((space(auto_psv))) progAddr = 0x1234; // Global variable located in Pgm Memory
    unsigned int offset;

//Set up pointer to the first memory location to be written

    TBLPAG = __builtin_tblpage(&progAddr);                  // Initialize PM Page Boundary SFR
    offset = __builtin_tbloffset(&progAddr);                // Initialize lower word of address

    __builtin_tblwtl(offset, 0x0000);                       // Set base address of erase block
                                                            // with dummy latch write

    NVMCON = 0x4058;                                        // Initialize NVMCON

    asm("DISI #5");                                         // Block all interrupts for next 5
                                                            // instructions
    __builtin_write_NVM();                                  // C30 function to perform unlock
                                                            // sequence and set WR
```

EXAMPLE 5-3: LOADING THE WRITE BUFFERS – ASSEMBLY LANGUAGE CODE

```
; Set up NVMCON for row programming operations
        MOV     #0x4004, 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
        •
        •
        •
; 32nd_program_word
        MOV     #LOW_WORD_31, W2            ;
        MOV     #HIGH_BYTE_31, W3           ;
        TBLWTL  W2, [W0]                    ; Write PM low word into program latch
        TBLWTH  W3, [W0]                    ; Write PM high byte into program latch
```

EXAMPLE 5-4: LOADING THE WRITE BUFFERS – ‘C’ LANGUAGE CODE

```
// C example using MPLAB C30

   #define NUM_INSTRUCTION_PER_ROW 64
   int __attribute__ ((space(auto_psv))) progAddr = 0x1234;  // Global variable located in Pgm Memory
   unsigned int offset;
   unsigned int i;
   unsigned int progData[2*NUM_INSTRUCTION_PER_ROW];         // Buffer of data to write

   //Set up NVMCON for row programming
   NVMCON = 0x4001;                                          // Initialize NVMCON

   //Set up pointer to the first memory location to be written
   TBLPAG = __builtin_tblpage(&progAddr);                    // Initialize PM Page Boundary SFR
   offset = __builtin_tbloffset(&progAddr);                  // Initialize lower word of address

   //Perform TBLWT instructions to write necessary number of latches
   for(i=0; i < 2*NUM_INSTRUCTION_PER_ROW; i++)
   {
       __builtin_tblwtl(offset, progData[i++]);              // Write to address low word
       __builtin_tblwth(offset, progData[i]);                // Write to upper byte
       offset = offset + 2;                                  // Increment address
   }
```

EXAMPLE 5-5: INITIATING A PROGRAMMING SEQUENCE – ASSEMBLY LANGUAGE CODE

```
        DISI    #5                          ; Block all interrupts
                                              for next 5 instructions
        MOV     #0x55, W0
        MOV     W0, NVMKEY                  ; Write the 55 key
        MOV     #0xAA, W1                   ;
        MOV     W1, NVMKEY                  ; Write the AA key
        BSET    NVMCON, #WR                 ; Start the erase sequence
        NOP                                 ; 2 NOPs required after setting WR
        NOP                                 ;
        BTSC    NVMCON, #15                 ; Wait for the sequence to be completed
        BRA     $-2                         ;
```

EXAMPLE 5-6: INITIATING A PROGRAMMING SEQUENCE – ‘C’ LANGUAGE CODE

```
// C example using MPLAB C30

asm("DISI #5");              // Block all interrupts for next 5 instructions

__builtin_write_NVM();       // Perform unlock sequence and set WR
```

6.0 DATA EEPROM 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 on Data EEPROM, refer to the *"dsPIC33/PIC24 Family Reference Manual"*, **"Data EEPROM"** (DS39720).

The data EEPROM memory is a Nonvolatile Memory (NVM), separate from the program and volatile data RAM. Data EEPROM memory is based on the same Flash technology as program memory, and is optimized for both long retention and a higher number of erase/write cycles.

The data EEPROM is mapped to the top of the user program memory space, with the top address at program memory address, 7FFE00h to 7FFFFFh. The size of the data EEPROM is 256 words in the PIC24FV32KA304 family devices.

The data EEPROM is organized as 16-bit wide memory. Each word is directly addressable, and is readable and writable during normal operation over the entire VDD range.

Unlike the Flash program memory, normal program execution is not stopped during a data EEPROM program or erase operation.

The data EEPROM programming operations are controlled using the three NVM Control registers:

- NVMCON: Nonvolatile Memory Control Register
- NVMKEY: Nonvolatile Memory Key Register
- NVMADR: Nonvolatile Memory Address Register

6.1 NVMCON Register

The NVMCON register (Register 6-1) is also the primary control register for data EEPROM program/erase operations. The upper byte contains the control bits used to start the program or erase cycle, and the flag bit to indicate if the operation was successfully performed. The lower byte of NVMCOM configures the type of NVM operation that will be performed.

6.2 NVMKEY Register

The NVMKEY is a write-only register that is used to prevent accidental writes or erasures of data EEPROM locations.

To start any programming or erase sequence, the following instructions must be executed first, in the exact order provided:

1. Write 55h to NVMKEY.
2. Write AAh to NVMKEY.

After this sequence, a write will be allowed to the NVMCON register for one instruction cycle. In most cases, the user will simply need to set the WR bit in the NVMCON register to start the program or erase cycle. Interrupts should be disabled during the unlock sequence.

The MPLAB® C30 C compiler provides a defined library procedure (`builtin_write_NVM`) to perform the unlock sequence. Example 6-1 illustrates how the unlock sequence can be performed with in-line assembly.

EXAMPLE 6-1: DATA EEPROM UNLOCK SEQUENCE

```
//Disable Interrupts For 5 instructions
asm volatile   ("disi #5");
//Issue Unlock Sequence
asm volatile   ("mov   #0x55, W0   \n"
                "mov   W0, NVMKEY  \n"
                "mov   #0xAA, W1   \n"
                "mov   W1, NVMKEY  \n");
// Perform Write/Erase operations
asm volatile   ("bset NVMCON, #WR \n"
                "nop               \n"
                "nop               \n");
```

REGISTER 6-1: NVMCON: NONVOLATILE MEMORY CONTROL REGISTER

R/S-0, HC	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
WR	WREN	WRERR	PGMONLY	—	—	—	—
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
—	ERASE	NVMOP5	NVMOP4	NVMOP3	NVMOP2	NVMOP1	NVMOP0
bit 7							bit 0

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

bit 15 **WR:** Write Control bit (program or erase)
1 = Initiates a data EEPROM erase or write cycle (can be set, but not cleared in software)
0 = Write cycle is complete (cleared automatically by hardware)

bit 14 **WREN:** Write Enable bit (erase or program)
1 = Enables an erase or program operation
0 = No operation allowed (device clears this bit on completion of the write/erase operation)

bit 13 **WRERR:** Write Flash Error Flag bit
1 = A write operation is prematurely terminated (any $\overline{\text{MCLR}}$ or WDT Reset during programming operation)
0 = The write operation completed successfully

bit 12 **PGMONLY:** Program Only Enable bit
1 = Write operation is executed without erasing target address(es) first
0 = Automatic erase-before-write
Write operations are preceded automatically by an erase of the target address(es).

bit 11-7 **Unimplemented:** Read as ‘0’

bit 6 **ERASE:** Erase Operation Select bit
1 = Performs an erase operation when WR is set
0 = Performs a write operation when WR is set

bit 5-0 **NVMOP<5:0>:** Programming Operation Command Byte bits
Erase Operations (when ERASE bit is ‘1’):
011010 = Erases 8 words
011001 = Erases 4 words
011000 = Erases 1 word
0100xx = Erases entire data EEPROM
Programming Operations (when ERASE bit is ‘0’):
0001xx = Writes 1 word

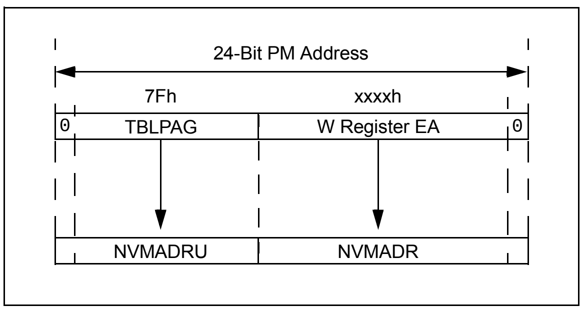
6.3 NVM Address Register

As with Flash program memory, the NVM Address registers, NVMADRU and NVMADR, form the 24-bit Effective Address (EA) of the selected row or word for data EEPROM operations. The NVMADRU register is used to hold the upper 8 bits of the EA, while the NVMADR register is used to hold the lower 16 bits of the EA. These registers are not mapped into the Special Function Register (SFR) space. Instead, they directly capture the EA<23:0> of the last Table Write instruction that has been executed and selects the data EEPROM row to erase. Figure 6-1 depicts the program memory EA that is formed for programming and erase operations.

Like program memory operations, the Least Significant bit (LSb) of NVMADR is restricted to even addresses. This is because any given address in the data EEPROM space consists of only the lower word of the program memory width; the upper word, including the uppermost "phantom byte", are unavailable. This means that the LSb of a data EEPROM address will always be '0'.

Similarly, the Most Significant bit (MSb) of NVMADRU is always '0', since all addresses lie in the user program space.

FIGURE 6-1: DATA EEPROM ADDRESSING WITH TBLPAG AND NVM ADDRESS REGISTERS



6.4 Data EEPROM Operations

The EEPROM block is accessed using Table Read and Write operations similar to those used for program memory. The TBLWTH and TBLRDH instructions are not required for data EEPROM operations since the memory is only 16 bits wide (data on the lower address is valid only). The following programming operations can be performed on the data EEPROM:

- Erase one, four or eight words
- Bulk erase the entire data EEPROM
- Write one word
- Read one word

Note 1: Unexpected results will be obtained if the user attempts to read the EEPROM while a programming or erase operation is underway.

2: The C30 C compiler includes library procedures to automatically perform the Table Read and Table Write operations, manage the Table Pointer and write buffers, and unlock and initiate memory write sequences. This eliminates the need to create assembler macros or time critical routines in C for each application.

The library procedures are used in the code examples detailed in the following sections. General descriptions of each process are provided for users who are not using the C30 compiler libraries.

6.4.1 ERASE DATA EEPROM

The data EEPROM can be fully erased, or can be partially erased, at three different sizes: one word, four words or eight words. The bits, NVMOP<1:0> (NVMCON<1:0>), decide the number of words to be erased. To erase partially from the data EEPROM, the following sequence must be followed:

1. Configure NVMCON to erase the required number of words: one, four or eight.
2. Load TBLPAG and WREG with the EEPROM address to be erased.
3. Clear the NVMIF status bit and enable the NVM interrupt (optional).
4. Write the key sequence to NVMKEY.
5. Set the WR bit to begin the erase cycle.
6. Either poll the WR bit or wait for the NVM interrupt (NVMIF is set).

A typical erase sequence is provided in Example 6-2. This example shows how to do a one-word erase. Similarly, a four-word erase and an eight-word erase can be done. This example uses C library procedures to manage the Table Pointer (`builtin_tblpage` and `builtin_tbloffset`) and the Erase Page Pointer (`builtin_tblwtl`). The memory unlock sequence (`builtin_write_NVM`) also sets the WR bit to initiate the operation and returns control when complete.

EXAMPLE 6-2: SINGLE-WORD ERASE

```
int __attribute__ ((space(eedata))) eeData = 0x1234;
/*--------------------------------------------------------------------------------------------
The variable eeData must be a Global variable declared outside of any method

the code following this comment can be written inside the method that will execute the erase
----------------------------------------------------------------------------------------------
*/
    unsigned int offset;

    // Set up NVMCON to erase one word of data EEPROM
    NVMCON = 0x4058;

    // Set up a pointer to the EEPROM location to be erased
    TBLPAG = __builtin_tblpage(&eeData);           // Initialize EE Data page pointer
    offset = __builtin_tbloffset(&eeData);         // Initizlize lower word of address
    __builtin_tblwtl(offset, 0);                   // Write EEPROM data to write latch

    asm volatile ("disi #5");                      // Disable Interrupts For 5 Instructions
    __builtin_write_NVM();                         // Issue Unlock Sequence & Start Write Cycle
    while(NVMCONbits.WR=1);                        // Optional: Poll WR bit to wait for
                                                   // write sequence to complete
```

6.4.1.1 Data EEPROM Bulk Erase

To erase the entire data EEPROM (bulk erase), the address registers do not need to be configured because this operation affects the entire data EEPROM. The following sequence helps in performing a bulk erase:

1. Configure NVMCON to Bulk Erase mode.
2. Clear the NVMIF status bit and enable the NVM interrupt (optional).
3. Write the key sequence to NVMKEY.
4. Set the WR bit to begin the erase cycle.
5. Either poll the WR bit or wait for the NVM interrupt (NVMIF is set).

A typical bulk erase sequence is provided in Example 6-3.

6.4.2 SINGLE-WORD WRITE

To write a single word in the data EEPROM, the following sequence must be followed:

1. Erase one data EEPROM word (as mentioned in the previous section) if the PGMONLY bit (NVMCON<12>) is set to '1'.
2. Write the data word into the data EEPROM latch.
3. Program the data word into the EEPROM:
   - Configure the NVMCON register to program one EEPROM word (NVMCON<5:0> = `0001xx`).
   - Clear the NVMIF status bit and enable the NVM interrupt (optional).
   - Write the key sequence to NVMKEY.
   - Set the WR bit to begin the erase cycle.
   - Either poll the WR bit or wait for the NVM interrupt (NVMIF is set).
   - To get cleared, wait until NVMIF is set.

A typical single-word write sequence is provided in Example 6-4.

EXAMPLE 6-3: DATA EEPROM BULK ERASE

```
    // Set up NVMCON to bulk erase the data EEPROM
    NVMCON = 0x4050;

    // Disable Interrupts For 5 Instructions
    asm volatile ("disi #5");

    // Issue Unlock Sequence and Start Erase Cycle
    __builtin_write_NVM();
```

EXAMPLE 6-4: SINGLE-WORD WRITE TO DATA EEPROM

```
int __attribute__ ((space(eedata))) eeData = 0x1234;
   int newData;                                     // New data to write to EEPROM
/*--------------------------------------------------------------------------------------------------
The variable eeData must be a Global variable declared outside of any method

the code following this comment can be written inside the method that will execute the write
----------------------------------------------------------------------------------------------------
*/
   unsigned int offset;

   // Set up NVMCON to erase one word of data EEPROM
   NVMCON = 0x4004;

   // Set up a pointer to the EEPROM location to be erased
   TBLPAG = __builtin_tblpage(&eeData);             // Initialize EE Data page pointer
   offset = __builtin_tbloffset(&eeData);           // Initizlize lower word of address
   __builtin_tblwtl(offset, newData);               // Write EEPROM data to write latch

   asm volatile ("disi #5");                        // Disable Interrupts For 5 Instructions
   __builtin_write_NVM();                           // Issue Unlock Sequence & Start Write Cycle
   while(NVMCONbits.WR=1);                          // Optional: Poll WR bit to wait for
                                                    // write sequence to complete
```

6.4.3 READING THE DATA EEPROM

To read a word from data EEPROM, the Table Read instruction is used. Since the EEPROM array is only 16 bits wide, only the TBLRDL instruction is needed. The read operation is performed by loading TBLPAG and WREG with the address of the EEPROM location, followed by a TBLRDL instruction.

A typical read sequence, using the Table Pointer management (builtin_tblpage and builtin_tbloffset) and Table Read procedures (builtin_tblrdl) from the C30 compiler library, is provided in Example 6-5.

Program Space Visibility (PSV) can also be used to read locations in the data EEPROM.

EXAMPLE 6-5: READING THE DATA EEPROM USING THE TBLRD COMMAND

```
int __attribute__ ((space(eedata))) eeData = 0x1234;
int data;                                           // Data read from EEPROM
/*--------------------------------------------------------------------------------------------
The variable eeData must be a Global variable declared outside of any method

the code following this comment can be written inside the method that will execute the read
--------------------------------------------------------------------------------------------
*/
    unsigned int offset;

    // Set up a pointer to the EEPROM location to be erased
    TBLPAG = __builtin_tblpage(&eeData);            // Initialize EE Data page pointer
    offset = __builtin_tbloffset(&eeData);          // Initizlize lower word of address
    data = __builtin_tblrdl(offset);                // Write EEPROM data to write latch
```


7.0 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 on Resets, refer to the *"dsPIC33/PIC24 Family Reference Manual"*, **"Reset with Programmable Brown-out Reset"** (DS39728).

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
- WDTR: Watchdog Timer Reset
- BOR: Brown-out Reset
- Low-Power BOR/Deep Sleep BOR
- 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 7-1.

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 Power-on Reset (POR) and unchanged by all other Resets.

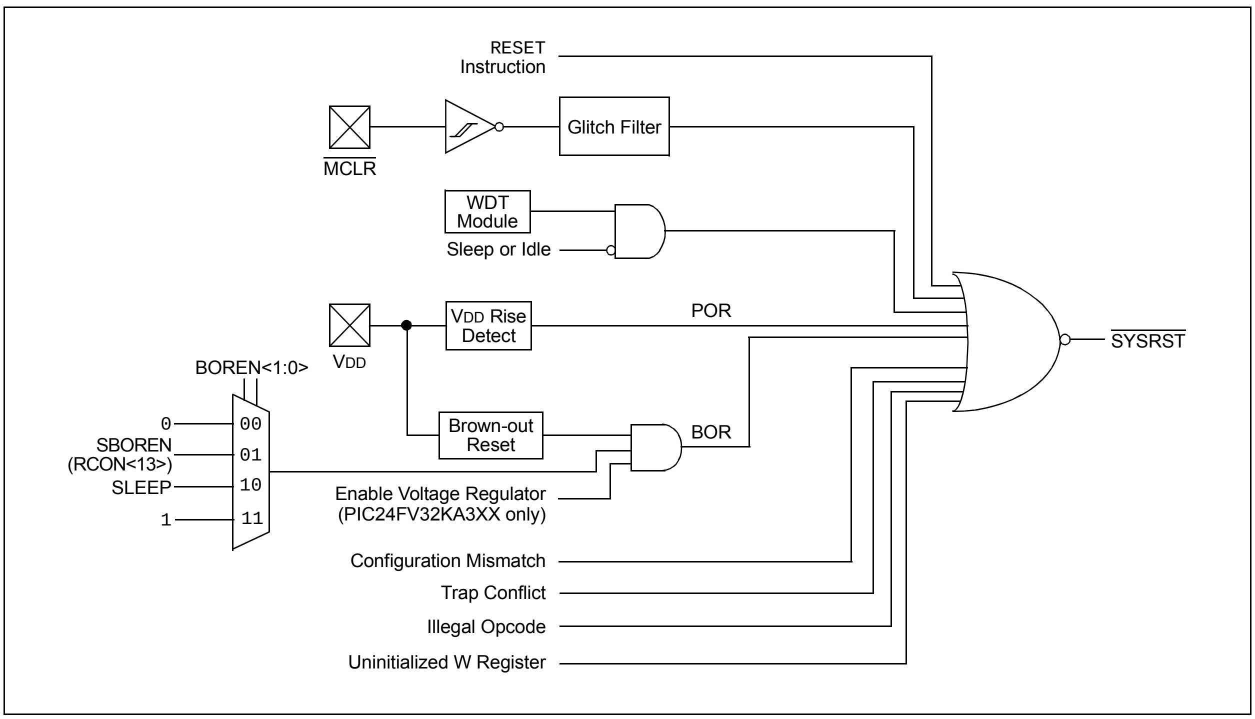
> **Note:** Refer to the specific peripheral or **Section 3.0 "CPU"** of this data sheet for register Reset states.

All types of device Reset will set a corresponding status bit in the RCON register to indicate the type of Reset (see Register 7-1). A Power-on Reset will clear all bits except for the BOR and POR bits (RCON<1:0>) 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 (WDT) and device power-saving states. The function of these bits is discussed in other sections of this manual.

> **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 7-1: RESET SYSTEM BLOCK DIAGRAM



REGISTER 7-1: RCON: RESET CONTROL REGISTER[1]

R/W-0, HS	R/W-0, HS	R/W-0	R/W-0	U-0	R/C-0, HS	R/W-0	R/W-0
TRAPR	IOPUWR	SBOREN	RETEN[3]	—	DPSLP	CM	PMSLP
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:	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

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 used as an Address Pointer caused a Reset
0 = An illegal opcode or Uninitialized W Reset has not occurred

bit 13 **SBOREN:** Software Enable/Disable of BOR bit
1 = BOR is turned on in software
0 = BOR is turned off in software

bit 12 **RETEN:** Retention Sleep Mode control bit[3]
1 = Regulated voltage supply provided solely by the Retention Regulator (RETEN) during Sleep
0 = Regulated voltage supply provided by the main Voltage Regulator (VREG) during Sleep

bit 11 **Unimplemented:** Read as '0'

bit 10 **DPSLP:** Deep Sleep Mode Flag bit
1 = Deep Sleep has occurred
0 = Deep Sleep has not occurred

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 **PMSLP:** Program Memory Power During Sleep bit
1 = Program memory bias voltage remains powered during Sleep
0 = Program memory bias voltage is powered down during Sleep and the Voltage Regulator enters Standby mode

bit 7 **EXTR:** External Reset ($\overline{\text{MCLR}}$) 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

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 FWDTENx Configuration bit is '1' (unprogrammed), the WDT is always enabled regardless of the SWDTEN bit setting.
3: This is implemented on PIC24FV32KA3XX parts only; not used on PIC24F32KA3XX devices.

REGISTER 7-1: RCON: RESET CONTROL REGISTER[1] (CONTINUED)

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-up from Sleep Flag bit
1 = Device has been in Sleep mode
0 = Device has not been in Sleep mode

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 (the BOR is also set after a POR)
0 = A Brown-out Reset has not occurred

bit 0 **POR:** Power-on Reset Flag bit
1 = A Power-up Reset has occurred
0 = A Power-up 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 FWDTENx Configuration bit is ‘1’ (unprogrammed), the WDT is always enabled regardless of the SWDTEN bit setting.
3: This is implemented on PIC24FV32KA3XX parts only; not used on PIC24F32KA3XX devices.

TABLE 7-1: 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	PWRSAV Instruction, POR
SLEEP (RCON<3>)	PWRSAV #SLEEP Instruction	POR
IDLE (RCON<2>)	PWRSAV #IDLE Instruction	POR
BOR (RCON<1>)	POR, BOR	—
POR (RCON<0>)	POR	—
DPSLP (RCON<10>)	PWRSAV #SLEEP Instruction with DSEN (DSCON<15>) Set	POR

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

7.1 Clock Source Selection at Reset

If clock switching is enabled, the system clock source at device Reset is chosen, as shown in Table 7-2. If clock switching is disabled, the system clock source is always selected according to the Oscillator Configuration bits. For more information, see **Section 9.0 "Oscillator Configuration"**.

TABLE 7-2: OSCILLATOR SELECTION vs. TYPE OF RESET (CLOCK SWITCHING ENABLED)

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

7.2 Device Reset Times

The Reset times for various types of device Reset are summarized in Table 7-3. Note that the System Reset Signal, $\overline{\text{SYSRST}}$, is released after the POR and PWRT delay times expire.

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 FSCM delay determines the time at which the FSCM begins to monitor the system clock source after the $\overline{\text{SYSRST}}$ signal is released.

TABLE 7-3: DELAY TIMES FOR VARIOUS DEVICE RESETS

Reset Type	Clock Source	$\overline{\text{SYSRST}}$ Delay	System Clock Delay	Notes
POR(6)	EC	TPOR + TPWRT	—	1, 2
	FRC, FRCDIV	TPOR + TPWRT	TFRC	1, 2, 3
	LPRC	TPOR + TPWRT	TLPRC	1, 2, 3
	ECPLL	TPOR + TPWRT	TLOCK	1, 2, 4
	FRCPLL	TPOR + TPWRT	TFRC + TLOCK	1, 2, 3, 4
	XT, HS, SOSC	TPOR+ TPWRT	TOST	1, 2, 5
	XTPLL, HSPLL	TPOR + TPWRT	TOST + TLOCK	1, 2, 4, 5
BOR	EC	TPWRT	—	2
	FRC, FRCDIV	TPWRT	TFRC	2, 3
	LPRC	TPWRT	TLPRC	2, 3
	ECPLL	TPWRT	TLOCK	2, 4
	FRCPLL	TPWRT	TFRC + TLOCK	2, 3, 4
	XT, HS, SOSC	TPWRT	TOST	2, 5
	XTPLL, HSPLL	TPWRT	TFRC + TLOCK	2, 3, 4
All Others	Any Clock	—	—	None

Note 1: TPOR = Power-on Reset delay.
2: TPWRT = 64 ms nominal if the Power-up Timer (PWRT) is enabled; otherwise, it is zero.
3: TFRC and TLPRC = RC oscillator start-up times.
4: TLOCK = PLL lock time.
5: TOST = Oscillator Start-up Timer (OST). A 10-bit counter waits 1024 oscillator periods before releasing the oscillator clock to the system.
6: If Two-Speed Start-up is enabled, regardless of the primary oscillator selected, the device starts with FRC, and in such cases, FRC start-up time is valid.

Note: For detailed operating frequency and timing specifications, see **Section 29.0 "Electrical Characteristics"**.

7.2.1 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 (OST) 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.

7.2.2 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).

7.3 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 FNOSCx bits in the Flash Configuration Word (FOSCSEL<2:0>); see Table 7-2. The RCFGCAL and NVMCON registers are only affected by a POR.

7.4 Deep Sleep BOR (DSBOR)

Deep Sleep BOR is a very low-power BOR circuitry, used when the device is in Deep Sleep mode. Due to low current consumption, accuracy may vary.

The DSBOR trip point is around 2.0V. DSBOR is enabled by configuring DSLPBOR (FDS<6>) = `1`. DSLPBOR will re-arm the POR to ensure the device will reset if V_{DD} drops below the POR threshold.

7.5 Brown-out Reset (BOR)

The PIC24FV32KA304 family devices implement a BOR circuit, which provides the user several configuration and power-saving options. The BOR is controlled by the BORV<1:0> and BOREN<1:0> Configuration bits (FPOR<6:5,1:0>). There are a total of four BOR configurations, which are provided in Table 7-3.

The BOR threshold is set by the BORV<1:0> bits. If BOR is enabled (any values of BOREN<1:0>, except '`00`'), any drop of V_{DD} below the set threshold point will reset the device. The chip will remain in BOR until V_{DD} rises above the threshold.

If the Power-up Timer is enabled, it will be invoked after V_{DD} rises above the threshold. Then, it will keep the chip in Reset for an additional time delay, T_{PWRT}, if V_{DD} drops below the threshold while the Power-up Timer is running. The chip goes back into a BOR and the Power-up Timer will be initialized. Once V_{DD} rises above the threshold, the Power-up Timer will execute the additional time delay.

BOR and the Power-up Timer (PWRT) are independently configured. Enabling the Brown-out Reset does not automatically enable the PWRT.

7.5.1 SOFTWARE ENABLED BOR

When BOREN<1:0> = `01`, the BOR can be enabled or disabled by the user in software. This is done with the control bit, SBOREN (RCON<13>). Setting SBOREN enables the BOR to function as previously described. Clearing the SBOREN disables the BOR entirely. The SBOREN bit operates only in this mode; otherwise, it is read as '`0`'.

Placing BOR under software control gives the user the additional flexibility of tailoring the application to its environment without having to reprogram the device to change the BOR configuration. It also allows the user to tailor the incremental current that the BOR consumes. While the BOR current is typically very small, it may have some impact in low-power applications.

Note:	Even when the BOR is under software control, the Brown-out Reset voltage level is still set by the BORV<1:0> Configuration bits; it cannot be changed in software.

7.5.2 DETECTING BOR

When BOR is enabled, the BOR bit (RCON<1>) is always reset to '1' on any BOR or POR event. This makes it difficult to determine if a BOR event has occurred just by reading the state of BOR alone. A more reliable method is to simultaneously check the state of both POR and BOR. This assumes that the POR and BOR bits are reset to '0' in the software immediately after any POR event. If the BOR bit is '1' while POR is '0', it can be reliably assumed that a BOR event has occurred.

Note:	Even when the device exits from Deep Sleep mode, both the POR and BOR bits are set.

7.5.3 DISABLING BOR IN SLEEP MODE

When BOREN<1:0> = 10, BOR remains under hardware control and operates as previously described. However, whenever the device enters Sleep mode, BOR is automatically disabled. When the device returns to any other operating mode, BOR is automatically re-enabled.

This mode allows for applications to recover from brown-out situations, while actively executing code when the device requires BOR protection the most. At the same time, it saves additional power in Sleep mode by eliminating the small incremental BOR current.

Note:	BOR levels differ depending on device type; PIC24FV32KA3XX devices are at different levels than those of PIC24F32KA3XX devices. See **Section 29.0 "Electrical Characteristics"** for BOR voltage levels.

8.0 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 on the Interrupt Controller, refer to the *"dsPIC33/PIC24 Family Reference Manual"*, **"Interrupts"** (DS70000600).

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

- Up to Eight 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

8.1 Interrupt Vector Table (IVT)

The IVT is shown in Figure 8-1. The IVT resides in the program memory, starting at location, 000004h. The IVT contains 126 vectors, consisting of eight 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.

PIC24FV32KA304 family devices implement non-maskable traps and unique interrupts; these are summarized in Table 8-1 and Table 8-2.

8.1.1 ALTERNATE INTERRUPT VECTOR TABLE (AIVT)

The Alternate Interrupt Vector Table (AIVT) is located after the IVT, as shown in Figure 8-1. Access to the AIVT is provided by the ALTIVT control bit (INTCON2<15>). 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.

8.2 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 Program Counter (PC) to zero. The microcontroller then begins program execution at location, 000000h. The user programs a `GOTO` instruction at the Reset address, which redirects the 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 8-1: PIC24F INTERRUPT VECTOR TABLE

Decreasing Natural Order Priority

Reset – GOTO Instruction	000000h	
Reset – GOTO Address	000002h	
Reserved	000004h	
Oscillator Fail Trap Vector		
Address Error Trap Vector		
Stack Error Trap Vector		
Math Error Trap Vector		
Reserved		
Reserved		
Reserved		
Interrupt Vector 0	000014h	
Interrupt Vector 1		
—		
—		
—		
Interrupt Vector 52	00007Ch	**Interrupt Vector Table (IVT)(1)**
Interrupt Vector 53	00007Eh	
Interrupt Vector 54	000080h	
—		
—		
—		
Interrupt Vector 116	0000FCh	
Interrupt Vector 117	0000FEh	
Reserved	000100h	
Reserved	000102h	
Reserved		
Oscillator Fail Trap Vector		
Address Error Trap Vector		
Stack Error Trap Vector		
Math Error Trap Vector		
Reserved		
Reserved		
Reserved		
Interrupt Vector 0	000114h	
Interrupt Vector 1		
—		
—		
—		**Alternate Interrupt Vector Table (AIVT)(1)**
Interrupt Vector 52	00017Ch	
Interrupt Vector 53	00017Eh	
Interrupt Vector 54	000180h	
—		
—		
—		
Interrupt Vector 116		
Interrupt Vector 117	0001FEh	
Start of Code	000200h	

Note 1: See Table 8-2 for the interrupt vector list.

TABLE 8-1: 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 8-2: 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	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>
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>
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 Change Notification	19	00003Ah	00013Ah	IFS1<3>	IEC1<3>	IPC4<14:12>
HLVD (High/Low-Voltage Detect)	72	0000A4h	0001A4h	IFS4<8>	IEC4<8>	IPC17<2:0>
NVM – NVM Write Complete	15	000032h	000132h	IFS0<15>	IEC0<15>	IPC3<14:12>
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>
Real-Time Clock/Calendar	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<2>	IPC8<2:0>
SPI2 Event	33	000056h	000156h	IFS2<1>	IEC2<1>	IPC8<6:4>
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	00015Ch	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>
Ultra Low-Power Wake-up	80	0000B4h	0001B4h	IFS5<0>	IEC5<0>	IPC20<2:0>

8.3 Interrupt Control and Status Registers

The PIC24FV32KA304 family of devices implements a total of 23 registers for the interrupt controller:

- INTCON1
- INTCON2
- IFS0, IFS1, IFS3 and IFS4
- IEC0, IEC1, IEC3 and IEC4
- IPC0 through IPC5, IPC7 and IPC15 through IPC19
- INTTREG

Global Interrupt Enable (GIE) 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 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 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 (IPL) 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 same sequence listed in Table 8-2. For example, the INT0 (External Interrupt 0) is depicted as having a vector number and a natural order priority of 0. The INT0IF status bit is found in IFS0<0>, the INT0IE enable bit in IEC0<0> and the INT0IP<2:0> priority bits are 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 may change the current CPU Interrupt Priority Level by writing to the IPLx bits.

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

All Interrupt registers are described in Register 8-1 through Register 8-33, in the following sections.

REGISTER 8-1: SR: ALU STATUS REGISTER

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 these bits, which are not dedicated to interrupt control functions.
2: The IPL<2:0> bits are concatenated with the IPL3 bit (CORCON<3>) to form the CPU Interrupt Priority Level. The value in parentheses indicates the Interrupt Priority Level if IPL3 = 1.
3: The IPLx Status bits are read-only when NSTDIS (INTCON1<15>) = 1.

Note: Bit 8 and bits 4 through 0 are described in **Section 3.0 “CPU”**.

REGISTER 8-2: 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/W-0	U-0	U-0
—	—	—	—	IPL3(2)	PSV(1)	—	—
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

bit 15-4 **Unimplemented:** Read as ‘0’

bit 3 **IPL3:** CPU Interrupt Priority Level Status bit(2)
1 = CPU Interrupt Priority Level is greater than 7
0 = CPU Interrupt Priority Level is 7 or less

bit 1-0 **Unimplemented:** Read as ‘0’

Note 1: See Register 3-2 for the description of this bit, which is not dedicated to interrupt control functions.
2: The IPL3 bit is concatenated with the IPL<2:0> bits (SR<7:5>) to form the CPU Interrupt Priority Level.

Note: Bit 2 is described in **Section 3.0 “CPU”**.

REGISTER 8-3: 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 8-4: INTCON2: INTERRUPT CONTROL REGISTER2

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	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	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 = Uses Alternate Interrupt Vector Table (AIVT)
0 = Uses standard (default) Interrupt Vector Table (IVT)

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

bit 13-3 **Unimplemented:** Read as ‘0’

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

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

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

REGISTER 8-5: IFS0: INTERRUPT FLAG STATUS REGISTER 0

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	R/W-0, HS
NVMIF	—	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 **NVMIF:** NVM Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 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

REGISTER 8-5: IFS0: INTERRUPT FLAG STATUS REGISTER 0 (CONTINUED)

bit 2	**OC1IF:** Output Compare Channel 1 Interrupt Flag Status bit 1 = Interrupt request has occurred 0 = Interrupt request has not occurred
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 8-6: 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	U-0	R/W-0, HS	U-0
U2TXIF	U2RXIF	INT2IF	T5IF	T4IF	—	OC3IF	—
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	R/W-0
—	—	—	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 **Unimplemented:** Read as ‘0’

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

bit 8-5 **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 8-7: IFS2: INTERRUPT FLAG STATUS 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, HS	U-0	U-0	U-0	R/W-0, HS	R/W-0, HS
—	—	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 15-6 **Unimplemented:** Read as '0'

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

REGISTER 8-8: 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	U-0	U-0	U-0	U-0	R/W-0, HS	R/W-0, HS	U-0
—	—	—	—	—	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 and Calendar Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 13-3 **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 8-9: 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	—	—	—	—	HLVDIF
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 **HLVDIF:** High/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 8-10: IFS5: INTERRUPT FLAG STATUS REGISTER 5

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-0, HS
—	—	—	—	—	—	—	ULPWUIF
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 15-1 **Unimplemented:** Read as ‘0’

bit 0 **ULPWUIF:** Ultra Low-Power Wake-up Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

REGISTER 8-11: IEC0: INTERRUPT ENABLE CONTROL REGISTER 0

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
NVMIE	—	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 **NVMIE:** NVM Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 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 8-11: 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 8-12: IEC1: INTERRUPT ENABLE CONTROL REGISTER 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	U-0
U2TXIE	U2RXIE	INT2IE	T5IE	T4IE	—	OC3IE	—
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
—	—	—	INT1IE	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 = 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 **Unimplemented:** Read as ‘0’

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

bit 8-5 **Unimplemented:** Read as ‘0’

bit 4 **INT1IE:** External Interrupt 1 Enable bit
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

REGISTER 8-13: IEC2: INTERRUPT ENABLE CONTROL 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	U-0	U-0	U-0	R/W-0	R/W-0
—	—	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 15-6 **Unimplemented:** Read as ‘0’

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

REGISTER 8-14: 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	U-0	U-0	U-0	U-0	R/W-0	R/W-0	U-0
—	—	—	—	—	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 and Calendar Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 13-3 **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’

REGISTER 8-15: 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	—	—	—	—	HLVDIE
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 **HLVDIE:** High/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 8-16: IEC5: INTERRUPT ENABLE CONTROL REGISTER 5

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-0
—	—	—	—	—	—	—	ULPWUIE
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 **ULPWUIE:** Ultra Low-Power Wake-up Interrupt Enable Bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

REGISTER 8-17: 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 8-18: 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 8-19: 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 8-20: IPC3: INTERRUPT PRIORITY CONTROL REGISTER 3

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	NVMIP2	NVMIP1	NVMIP0	—	—	—	—
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 **Unimplemented:** Read as ‘0’

bit 14-12 **NVMIP<2:0>:** NVM Interrupt Priority bits
111 = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is Priority 1
000 = Interrupt source is disabled

bit 11-7 **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 8-21: 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
—	MI2C1P2	MI2C1P1	MI2C1P0	—	SI2C1P2	SI2C1P1	SI2C1P0
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 **MI2C1P<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 **SI2C1P<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 8-22: IPC5: INTERRUPT PRIORITY CONTROL REGISTER 5

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
—	—	—	—	—	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 15-3 **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 8-23: IPC6: INTERRUPT PRIORITY CONTROL REGISTER 6

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	T4IP2	T4IP1	T4IP0	—	—	—	—
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 11-7 **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 8-24: 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 8-25: 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 8-26: IPC9: INTERRUPT PRIORITY CONTROL REGISTER 9

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
—	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 15-7 **Unimplemented:** Read as ‘0’

bit 6-4 **IC3IP<2:0>:** Input Capture Channel 3 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 8-27: 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 8-28: 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 8-29: 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 8-30: 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
—	—	—	—	—	HLVDIP2	HLVDIP1	HLVDIP0
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 **HLVDIP<2:0>:** High/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 8-31: 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 8-32: IPC20: INTERRUPT PRIORITY CONTROL REGISTER 20

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
—	—	—	—	—	ULPWUIP2	ULPWUIP1	ULPWUIP0
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 6-4 **ULPWUIP<2:0>:** Ultra Low-Power Wake-up Interrupt Priority bits

111 = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is Priority 1
000 = Interrupt source is disabled

REGISTER 8-33: INTTREG: INTERRUPT CONTROL AND STATUS REGISTER

R-0	U-0	R/W-0	U-0	R-0	R-0	R-0	R-0
CPUIRQ	—	VHOLD	—	ILR3	ILR2	ILR1	ILR0
bit 15							bit 8

U-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
—	VECNUM6	VECNUM5	VECNUM4	VECNUM3	VECNUM2	VECNUM1	VECNUM0
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 **CPUIRQ:** Interrupt Request from Interrupt Controller CPU bit
1 = An interrupt request has occurred but has not yet been Acknowledged by the CPU (this will happen when the CPU priority is higher than the interrupt priority)
0 = No interrupt request is left unacknowledged

bit 14 **Unimplemented:** Read as ‘0’

bit 13 **VHOLD:** Vector Hold bit
Allows Vector Number Capture and Changes which Interrupt is Stored in the VECNUM bit:
1 = VECNUM will contain the value of the highest priority pending interrupt, instead of the current interrupt
0 = VECNUM will contain the value of the last Acknowledged interrupt (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<6:0>:** Vector Number of Pending Interrupt bits
0111111 = Interrupt vector pending is Number 135
•
•
•
0000001 = Interrupt vector pending is Number 9
0000000 = Interrupt vector pending is Number 8

8.4 Interrupt Setup Procedures

8.4.1 INITIALIZATION

To configure an interrupt source:

1. Set the NSTDIS control bit (INTCON1<15>) 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.

8.4.2 INTERRUPT SERVICE ROUTINE

The method that is used to declare an ISR (Interrupt Service Routine) and initialize the IVT with the correct vector address depends on the programming language (i.e., C or assembly) 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.

8.4.3 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.

8.4.4 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, OEh with SRL.

To enable user interrupts, the `POP` instruction may be used to restore the previous SR value.

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. Level 7 interrupt sources are not disabled by the `DISI` instruction.

NOTES:

9.0 OSCILLATOR CONFIGURATION

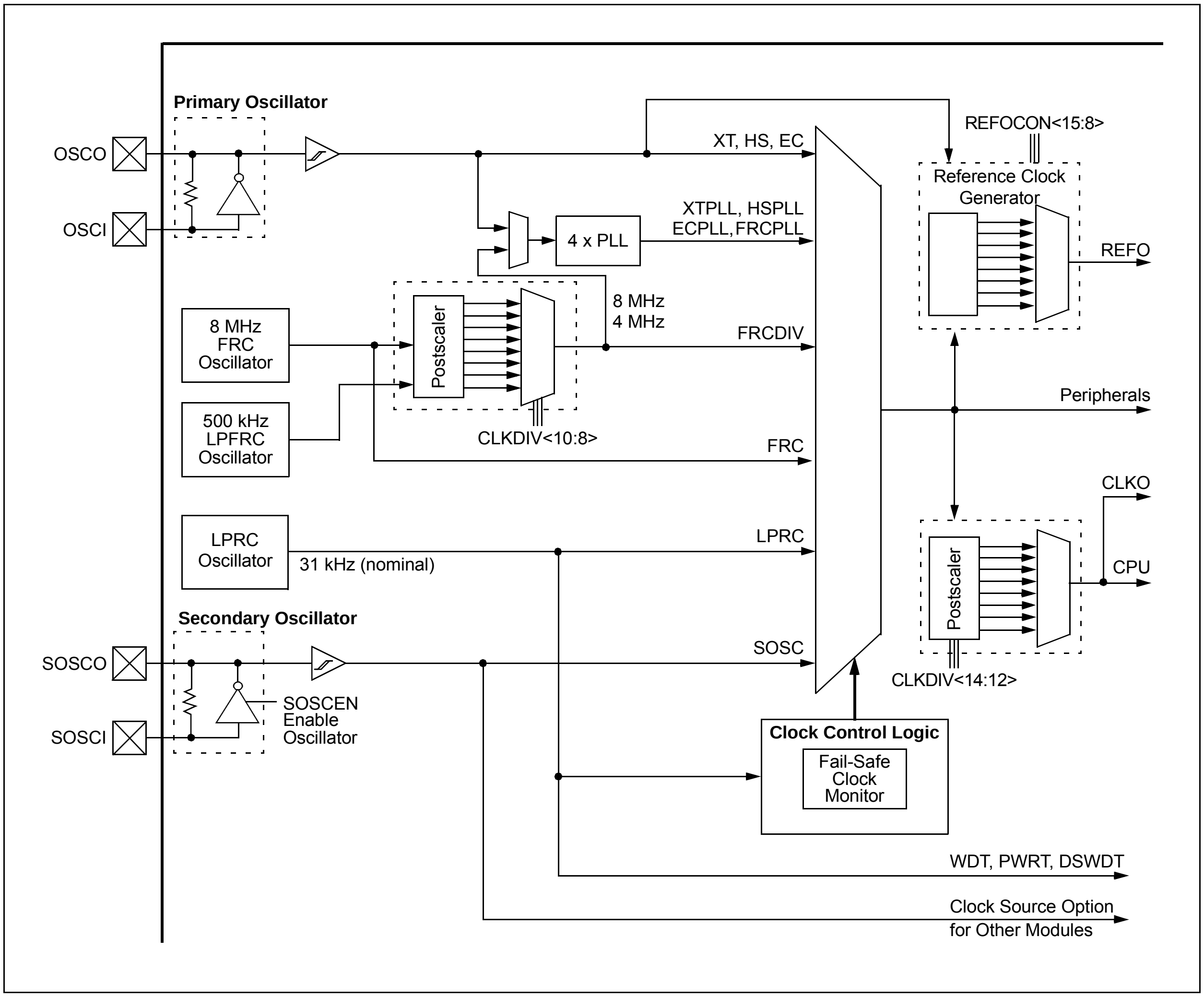
Note: This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information on Oscillator Configuration, refer to the *"dsPIC33/PIC24 Family Reference Manual"*, **"Oscillator with 500 kHz Low-Power FRC"** (DS39726).

The oscillator system for the PIC24FV32KA304 family of devices has the following features:

- A total of five external and internal oscillator options as clock sources, providing 11 different clock modes.
- On-chip 4x Phase Locked Loop (PLL) to boost internal operating frequency on select internal and external oscillator sources.
- Software-controllable switching between various clock sources.
- Software-controllable postscaler for selective clocking of CPU for system power savings.
- System frequency range declaration bits for EC mode. When using an external clock source, the current consumption is reduced by setting the declaration bits to the expected frequency range.
- A Fail-Safe Clock Monitor (FSCM) that detects clock failure and permits safe application recovery or shutdown.

A simplified diagram of the oscillator system is shown in Figure 9-1.

FIGURE 9-1: PIC24FV32KA304 FAMILY CLOCK DIAGRAM



9.1 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
  The PIC24FV32KA304 family devices consist of two types of secondary oscillator:
  - High-Power Secondary Oscillator
  - Low-Power Secondary Oscillator
  These can be selected by using the SOSCSEL (FOSC<5>) bit.
- Fast Internal RC (FRC) Oscillator
  - 8 MHz FRC Oscillator
  - 500 kHz Lower Power FRC Oscillator
- Low-Power Internal RC (LPRC) Oscillator with two modes:
  - High-Power/High Accuracy mode
  - Low-Power/Low Accuracy mode

The primary oscillator and 8 MHz FRC sources have the option of using the internal 4x PLL. The frequency of the FRC clock source can optionally be reduced by the programmable clock divider. 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.

9.2 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 (for more information, see **Section 26.1 "Configuration Bits"**). The Primary Oscillator Configuration bits, POSCMD<1:0> (FOSC<1:0>), and the Initial Oscillator Select Configuration bits, FNOSC<2:0> (FOSCSEL<2:0>), 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 EC mode Frequency Range Configuration bits, POSCFREQ<1:0> (FOSC<4:3>), optimize power consumption when running in EC mode. The default configuration is "frequency range is greater than 8 MHz".

The Configuration bits allow users to choose between the various clock modes, shown in Table 9-1.

9.2.1 CLOCK SWITCHING MODE CONFIGURATION BITS

The FCKSMx Configuration bits (FOSC<7:6>) are used jointly to configure device clock switching and the 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 9-1: CONFIGURATION BIT VALUES FOR CLOCK SELECTION

Oscillator Mode	Oscillator Source	POSCMD<1:0>	FNOSC<2:0>	Notes
8 MHz FRC Oscillator with Postscaler (FRCDIV)	Internal	11	111	1, 2
500 kHz FRC Oscillator with Postscaler (LPFRCDIV)	Internal	11	110	1
Low-Power RC Oscillator (LPRC)	Internal	11	101	1
Secondary (Timer1) Oscillator (SOSC)	Secondary	00	100	1
Primary Oscillator (HS) with PLL Module (HSPLL)	Primary	10	011	
Primary Oscillator (EC) with PLL Module (ECPLL)	Primary	00	011	
Primary Oscillator (HS)	Primary	10	010	
Primary Oscillator (XT)	Primary	01	010	
Primary Oscillator (EC)	Primary	00	010	
8 MHz FRC Oscillator with PLL Module (FRCPLL)	Internal	11	001	1
8 MHz FRC Oscillator (FRC)	Internal	11	000	1

Note 1: The OSCO pin function is determined by the OSCIOFNC Configuration bit.
2: This is the default oscillator mode for an unprogrammed (erased) device.

9.3 Control Registers

The operation of the oscillator is controlled by three Special Function Registers (SFRs):

- OSCCON
- CLKDIV
- OSCTUN

The OSCCON register (Register 9-1) is the main control register for the oscillator. It controls clock source switching and allows the monitoring of clock sources.

The Clock Divider register (Register 9-2) controls the features associated with Doze mode, as well as the postscaler for the FRC oscillator.

The FRC Oscillator Tune register (Register 9-3) allows the user to fine tune the FRC oscillator over a range of approximately ±5.25%. Each bit increment or decrement changes the factory calibrated frequency of the FRC oscillator by a fixed amount.

REGISTER 9-1: OSCCON: OSCILLATOR CONTROL REGISTER

U-0	R-0, HSC	R-0, HSC	R-0, HSC	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/SO-0, HSC	U-0	R-0, HSC[2]	U-0	R/CO-0, HS	R/W-0[3]	R/W-0	R/W-0
CLKLOCK	—	LOCK	—	CF	SOSCDRV	SOSCEN	OSWEN
bit 7							bit 0

Legend:	HSC = Hardware Settable/Clearable bit		
HS = Hardware Settable bit	CO = Clearable Only bit	SO = Settable Only 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 **COSC<2:0>:** Current Oscillator Selection bits
111 = 8 MHz Fast RC Oscillator with Postscaler (FRCDIV)
110 = 500 kHz Low-Power Fast RC Oscillator (FRC) with Postscaler (LPFRCDIV)
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 = 8 MHz FRC Oscillator with Postscaler and PLL module (FRCPLL)
000 = 8 MHz FRC Oscillator (FRC)

bit 11 **Unimplemented:** Read as ‘0’

bit 10-8 **NOSC<2:0>:** New Oscillator Selection bits[1]
111 = 8 MHz Fast RC Oscillator with Postscaler (FRCDIV)
110 = 500 kHz Low-Power Fast RC Oscillator (FRC) with Postscaler (LPFRCDIV)
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 = 8 MHz FRC Oscillator with Postscaler and PLL module (FRCPLL)
000 = 8 MHz FRC Oscillator (FRC)

Note 1: Reset values for these bits are determined by the FNOSCx Configuration bits.
2: This bit also resets to ‘0’ during any valid clock switch or whenever a non-PLL Clock mode is selected.
3: When SOSC is selected to run from a digital clock input, rather than an external crystal (SOSCSRC = 0), this bit has no effect.

REGISTER 9-1: OSCCON: OSCILLATOR CONTROL REGISTER (CONTINUED)

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	**Unimplemented:** Read as '0'
bit 5	**LOCK:** PLL Lock Status bit[2] 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	**SOSCDRV**: Secondary Oscillator Drive Strength bit[3] 1 = High-power SOSC circuit is selected 0 = Low/high-power select is done via the SOSCSRC Configuration bit
bit 1	**SOSCEN:** 32 kHz Secondary Oscillator (SOSC) Enable bit 1 = Enables the secondary oscillator 0 = Disables the secondary oscillator
bit 0	**OSWEN:** Oscillator Switch Enable bit 1 = Initiates 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 FNOSCx Configuration bits.

2: This bit also resets to '0' during any valid clock switch or whenever a non-PLL Clock mode is selected.

3: When SOSC is selected to run from a digital clock input, rather than an external crystal (SOSCSRC = 0), this bit has no effect.

REGISTER 9-2: CLKDIV: CLOCK DIVIDER REGISTER

R/W-0	R/W-0	R/W-1	R/W-1	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

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 **ROI:** Recover on Interrupt bit
1 = Interrupts clear the DOZEN bit, and reset the CPU and peripheral clock ratio to 1:1
0 = Interrupts have no effect on the DOZEN bit

bit 14-12 **DOZE<2:0>:** CPU and 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 and peripheral clock ratio
0 = CPU and peripheral clock ratio are set to 1:1

bit 10-8 **RCDIV<2:0>:** FRC Postscaler Select bits
When COSC<2:0> (OSCCON<14:12>) = 111:
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) (default)
000 = 8 MHz (divide-by-1)
When COSC<2:0> (OSCCON<14:12>) = 110:
111 = 1.95 kHz (divide-by-256)
110 = 7.81 kHz (divide-by-64)
101 = 15.62 kHz (divide-by-32)
100 = 31.25 kHz (divide-by-16)
011 = 62.5 kHz (divide-by-8)
010 = 125 kHz (divide-by-4)
001 = 250 kHz (divide-by-2) (default)
000 = 500 kHz (divide-by-1)

bit 7-0 **Unimplemented:** Read as ‘0’

Note 1: This bit is automatically cleared when the ROI bit is set and an interrupt occurs.

REGISTER 9-3: 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
—	—	TUN<5:0>(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.

9.4 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.

9.4.1 ENABLING CLOCK SWITCHING

To enable clock switching, the FCKSM1 Configuration bit in the FOSC Configuration register must be programmed to ‘0’. (Refer to **Section 26.0 “Special Features”** for further details.) If the FCKSM1 Configuration bit is unprogrammed (‘1’), the clock switching function and FSCM function are disabled. This is the default setting.

The NOSCx control bits (OSCCON<10:8>) do not control the clock selection when clock switching is disabled. However, the COSCx bits (OSCCON<14:12>) will reflect the clock source selected by the FNOSCx Configuration bits.

The OSWEN control bit (OSCCON<0>) has no effect when clock switching is disabled; it is held at ‘0’ at all times.

9.4.2 OSCILLATOR SWITCHING SEQUENCE

At a minimum, performing a clock switch requires this basic sequence:

1. If desired, read the COSCx bits (OSCCON<14:12>) 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 bits (OSCCON<10:8>) 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 bits value is transferred to the COSCx bits.
6. The old clock source is turned off at this time, with the exception of LPRC (if WDT, FSCM or RTCC with LPRC as a clock source is enabled) or SOSC (if SOSCEN remains enabled).

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 mode 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.

The following code sequence for a clock switch is recommended:

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 9-1.

EXAMPLE 9-1: BASIC CODE SEQUENCE FOR CLOCK SWITCHING

```
;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
```

9.5 Reference Clock Output

In addition to the CLKO output (FOSC/2) available in certain oscillator modes, the device clock in the PIC24FV32KA304 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 9-4). Setting the ROEN bit (REFOCON<15>) makes the clock signal available on the REFO pin. The RODIVx 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 OSC1 and OSC2, 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 ROSEL bit is not also set, the oscillator on OSC1 and OSC2 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 9-4: 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	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 is enabled on REFO pin
0 = Reference oscillator is 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 = Primary oscillator is used as the base clock(1)
0 = System clock is 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: The crystal oscillator must be enabled using the FOSC<2:0> bits; the crystal maintains the operation in Sleep mode.

NOTES:

10.0 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 *"dsPIC33/PIC24 Family Reference Manual"*, **"Power-Saving Features with Deep Sleep"** (DS39727).

The PIC24FV32KA304 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, Idle and Deep Sleep 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.

10.1 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 NOSCx bits. The process of changing a system clock during operation, as well as limitations to the process, are discussed in more detail in **Section 9.0 "Oscillator Configuration"**.

10.2 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. Deep Sleep mode stops clock operation, code execution and all peripherals, except RTCC and DSWDT. It also freezes I/O states and removes power to SRAM and Flash memory.

The assembly syntax of the PWRSAV instruction is shown in Example 10-1.

Note:	SLEEP_MODE and IDLE_MODE are constants defined in the assembler include file for the selected device.

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".

10.2.1 SLEEP MODE

Sleep mode includes 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 I/O pin directions and states are frozen.
- The Fail-Safe Clock Monitor 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 or RTCC with LPRC as the clock source 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.

The device will wake-up from Sleep mode on any of 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 10-1: PWRSAV INSTRUCTION SYNTAX

```
PWRSAV     #SLEEP_MODE    ; Put the device into SLEEP mode
PWRSAV     #IDLE_MODE     ; Put the device into IDLE mode
BSET       DSCON, #DSEN   ; Enable Deep Sleep
PWRSAV     #SLEEP_MODE    ; Put the device into Deep SLEEP mode
```

10.2.2 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 10.6 "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 re-applied to the CPU and instruction execution begins immediately, starting with the instruction following the PWRSAV instruction or the first instruction in the ISR.

10.2.3 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.

10.2.4 DEEP SLEEP MODE

In PIC24FV32KA304 family devices, Deep Sleep mode is intended to provide the lowest levels of power consumption available without requiring the use of external switches to completely remove all power from the device. Entry into Deep Sleep mode is completely under software control. Exit from Deep Sleep mode can be triggered from any of the following events:

- POR Event
- $\overline{\text{MCLR}}$ Event
- RTCC Alarm (if the RTCC is present)
- External Interrupt 0
- Deep Sleep Watchdog Timer (DSWDT) Time-out
- Ultra Low-Power Wake-up (ULPWU) Event

In Deep Sleep mode, it is possible to keep the device Real-Time Clock and Calendar (RTCC) running without the loss of clock cycles.

The device has a dedicated Deep Sleep Brown-out Reset (DSBOR) and a Deep Sleep Watchdog Timer Reset (DSWDT) for monitoring voltage and time-out events. The DSBOR and DSWDT are independent of the standard BOR and WDT used with other power-managed modes (Sleep, Idle and Doze).

10.2.4.1 Entering Deep Sleep Mode

Deep Sleep mode is entered by setting the DSEN bit in the DSCON register and then executing a Sleep command (PWRSAV #SLEEP_MODE). An unlock sequence is required to set the DSEN bit. Once the DSEN bit has been set, there is no time limit before the SLEEP command can be executed. The DSEN bit is automatically cleared when exiting the Deep Sleep mode.

Note:	To re-enter Deep Sleep after a Deep Sleep wake-up, allow a delay of at least 3 TCY after clearing the RELEASE bit.

The sequence to enter Deep Sleep mode is:

1. If the application requires the Deep Sleep WDT, enable it and configure its clock source. For more information on Deep Sleep WDT, see **Section 10.2.4.5 "Deep Sleep WDT"**.
2. If the application requires Deep Sleep BOR, enable it by programming the DSLPBOR Configuration bit (FDS<6>).
3. If the application requires wake-up from Deep Sleep on RTCC alarm, enable and configure the RTCC module For more information on RTCC, see **Section 19.0 "Real-Time Clock and Calendar (RTCC)"**.
4. If needed, save any critical application context data by writing it to the DSGPR0 and DSGPR1 registers (optional).
5. Enable Deep Sleep mode by setting the DSEN bit (DSCON<15>).

Note:	An unlock sequence is required to set the DSEN bit.

6. Enter Deep Sleep mode by issuing a PWRSAV #0 instruction.

Any time the DSEN bit is set, all bits in the DSWAKE register will be automatically cleared.

To set the DSEN bit, the unlock sequence in Example 10-2 is required:

EXAMPLE 10-2: THE UNLOCK SEQUENCE

```
//Disable Interrupts For 5 instructions
asm   volatile("disi #5");
//Issue Unlock Sequence
asm   volatile
mov   #0x55, W0;
mov   W0, NVMKEY;
mov   #0xAA, W1;
mov   W1, NVMKEY;
bset  DSCON, #DSEN
```


10.2.4.2 Exiting Deep Sleep Mode

Deep Sleep mode exits on any one of the following events:

- A POR event on V_{DD} supply. If there is no DSBOR circuit to re-arm the V_{DD} supply POR circuit, the external V_{DD} supply must be lowered to the natural arming voltage of the POR circuit.
- A DSWDT time-out. When the DSWDT timer times out, the device exits Deep Sleep.
- An RTCC alarm (if RTCEN = `1`).
- An assertion (‘`0`’) of the $\overline{\text{MCLR}}$ pin.
- An assertion of the INT0 pin (if the interrupt was enabled before Deep Sleep mode was entered). The polarity configuration is used to determine the assertion level (‘`0`’ or ‘`1`’) of the pin that will cause an exit from Deep Sleep mode. Exiting from Deep Sleep mode requires a change on the INT0 pin while in Deep Sleep mode.

Note:	Any interrupt pending when entering Deep Sleep mode is cleared.

Exiting Deep Sleep mode generally does not retain the state of the device and is equivalent to a Power-on Reset (POR) of the device. Exceptions to this include the RTCC (if present), which remains operational through the wake-up, the DSGPRx registers and DSWDT.

Wake-up events that occur after Deep Sleep exits, but before the POR sequence completes, are ignored and are not be captured in the DSWAKE register.

The sequence for exiting Deep Sleep mode is:

1. After a wake-up event, the device exits Deep Sleep and performs a POR. The DSEN bit is cleared automatically. Code execution resumes at the Reset vector.
2. To determine if the device exited Deep Sleep, read the Deep Sleep bit, DPSLP (RCON<10>). This bit will be set if there was an exit from Deep Sleep mode; if the bit is set, clear it.
3. Determine the wake-up source by reading the DSWAKE register.
4. Determine if a DSBOR event occurred during Deep Sleep mode by reading the DSBOR bit (DSCON<1>).
5. If application context data has been saved, read it back from the DSGPR0 and DSGPR1 registers.
6. Clear the RELEASE bit (DSCON<0>).

10.2.4.3 Saving Context Data with the DSGPR0/DSGPR1 Registers

As exiting Deep Sleep mode causes a POR, most Special Function Registers reset to their default POR values. In addition, because V_{CORE} power is not supplied in Deep Sleep mode, information in data RAM may be lost when exiting this mode.

Applications which require critical data to be saved prior to Deep Sleep may use the Deep Sleep General Purpose registers, DSGPR0 and DSGPR1 or data EEPROM (if available). Unlike other SFRs, the contents of these registers are preserved while the device is in Deep Sleep mode. After exiting Deep Sleep, software can restore the data by reading the registers and clearing the RELEASE bit (DSCON<0>).

10.2.4.4 I/O Pins During Deep Sleep

During Deep Sleep, the general purpose I/O pins retain their previous states and the Secondary Oscillator (SOSC) will remain running, if enabled. Pins that are configured as inputs (TRISx bit is set), prior to entry into Deep Sleep, remain high-impedance during Deep Sleep. Pins that are configured as outputs (TRISx bit is clear), prior to entry into Deep Sleep, remain as output pins during Deep Sleep. While in this mode, they continue to drive the output level determined by their corresponding LATx bit at the time of entry into Deep Sleep.

Once the device wakes back up, all I/O pins continue to maintain their previous states, even after the device has finished the POR sequence and is executing application code again. Pins configured as inputs during Deep Sleep remain high-impedance and pins configured as outputs continue to drive their previous value. After waking up, the TRIS and LAT registers, and the SOSCEN bit (OSCCON<1>) are reset. If firmware modifies any of these bits or registers, the I/O will not immediately go to the newly configured states. Once the firmware clears the RELEASE bit (DSCON<0>), the I/O pins are “released”. This causes the I/O pins to take the states configured by their respective TRISx and LATx bit values.

This means that keeping the SOSC running after waking up requires the SOSCEN bit to be set before clearing RELEASE.

If the Deep Sleep BOR (DSBOR) is enabled, and a DSBOR or a true POR event occurs during Deep Sleep, the I/O pins will be immediately released, similar to clearing the RELEASE bit. All previous state information will be lost, including the general purpose DSGPR0 and DSGPR1 contents.

If a $\overline{\text{MCLR}}$ Reset event occurs during Deep Sleep, the DSGPRx, DSCON and DSWAKE registers will remain valid, and the RELEASE bit will remain set. The state of the SOSC will also be retained. The I/O pins, however, will be reset to their $\overline{\text{MCLR}}$ Reset state. Since RELEASE is still set, changes to the SOSCEN bit (OSCCON<1>) cannot take effect until the RELEASE bit is cleared.

In all other Deep Sleep wake-up cases, application firmware must clear the RELEASE bit in order to reconfigure the I/O pins.

10.2.4.5 Deep Sleep WDT

To enable the DSWDT in Deep Sleep mode, program the Configuration bit, DSWDTEN (FDS<7>). The device Watchdog Timer (WDT) need not be enabled for the DSWDT to function. Entry into Deep Sleep mode automatically resets the DSWDT.

The DSWDT clock source is selected by the DSWDTOSC Configuration bit (FDS<4>). The postscaler options are programmed by the DSWDTPS<3:0> Configuration bits (FDS<3:0>). The minimum time-out period that can be achieved is 2.1 ms and the maximum is 25.7 days. For more details on the FDS Configuration register and DSWDT configuration options, refer to **Section 26.0 "Special Features"**.

10.2.4.6 Switching Clocks in Deep Sleep Mode

Both the RTCC and the DSWDT may run from either SOSC or the LPRC clock source. This allows both the RTCC and DSWDT to run without requiring both the LPRC and SOSC to be enabled together, reducing power consumption.

Running the RTCC from LPRC will result in a loss of accuracy in the RTCC of approximately 5 to 10%. If a more accurate RTCC is required, it must be run from the SOSC clock source. The RTCC clock source is selected with the RTCOSC Configuration bit (FDS<5>).

Under certain circumstances, it is possible for the DSWDT clock source to be off when entering Deep Sleep mode. In this case, the clock source is turned on automatically (if DSWDT is enabled), without the need for software intervention; however, this can cause a delay in the start of the DSWDT counters. In order to avoid this delay when using SOSC as a clock source, the application can activate SOSC prior to entering Deep Sleep mode.

10.2.4.7 Checking and Clearing the Status of Deep Sleep

Upon entry into Deep Sleep mode, the status bit, DPSLP (RCON<10>), becomes set and must be cleared by the software.

On power-up, the software should read this status bit to determine if the Reset was due to an exit from Deep Sleep mode and clear the bit if it is set. Of the four possible combinations of DPSLP and POR bit states, three cases can be considered:

- Both the DPSLP and POR bits are cleared. In this case, the Reset was due to some event other than a Deep Sleep mode exit.
- The DPSLP bit is clear, but the POR bit is set; this is a normal POR.
- Both the DPSLP and POR bits are set. This means that Deep Sleep mode was entered, the device was powered down and Deep Sleep mode was exited.

10.2.4.8 Power-on Resets (PORs)

V_{DD} voltage is monitored to produce PORs. Since exiting from Deep Sleep functionally looks like a POR, the technique described in **Section 10.2.4.7 "Checking and Clearing the Status of Deep Sleep"** should be used to distinguish between Deep Sleep and a true POR event.

When a true POR occurs, the entire device, including all Deep Sleep logic (Deep Sleep registers: RTCC, DSWDT, etc.) is reset.

10.2.4.9 Summary of Deep Sleep Sequence

To review, these are the necessary steps involved in invoking and exiting Deep Sleep mode:

1. The device exits Reset and begins to execute its application code.
2. If DSWDT functionality is required, program the appropriate Configuration bit.
3. Select the appropriate clock(s) for the DSWDT and RTCC (optional).
4. Enable and configure the DSWDT (optional).
5. Enable and configure the RTCC (optional).
6. Write context data to the DSGPRx registers (optional).
7. Enable the INT0 interrupt (optional).
8. Set the DSEN bit in the DSCON register.
9. Enter Deep Sleep by issuing a `PWRSV #SLEEP_MODE` command.
10. The device exits Deep Sleep when a wake-up event occurs.
11. The DSEN bit is automatically cleared.
12. Read and clear the DPSLP status bit in RCON, and the DSWAKE status bits.
13. Read the DSGPRx registers (optional).
14. Once all state related configurations are complete, clear the RELEASE bit.
15. The application resumes normal operation.

REGISTER 10-1: DSCON: DEEP SLEEP CONTROL REGISTER[1]

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
DSEN	—	—	—	—	—	—	RTCCWDIS
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/C-0, HS
—	—	—	—	—	ULPWUDIS	DSBOR[2]	RELEASE
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

bit 15 **DSEN:** Deep Sleep Enable bit
1 = Enters Deep Sleep on execution of `PWRSAV #0`
0 = Enters normal Sleep on execution of `PWRSAV #0`

bit 14-9 **Unimplemented:** Read as ‘0’

bit 8 **RTCCWDIS:** RTCC Wake-up Disable bit
1 = Wake-up from Deep Sleep with RTCC disabled
0 = Wake-up from Deep Sleep with RTCC enabled

bit 7-3 **Unimplemented:** Read as ‘0’

bit 2 **ULPWUDIS:** ULPWU Wake-up Disable bit
1 = Wake-up from Deep Sleep with ULPWU disabled
0 = Wake-up from Deep Sleep with ULPWU enabled

bit 1 **DSBOR:** Deep Sleep BOR Event bit[2]
1 = The DSBOR was active and a BOR event was detected during Deep Sleep
0 = The DSBOR was not active or was active but did not detect a BOR event during Deep Sleep

bit 0 **RELEASE:** I/O Pin State Release bit
1 = Upon waking from Deep Sleep, I/O pins maintain their previous states to Deep Sleep entry
0 = Release I/O pins from their state previous to Deep Sleep entry, and allow their respective TRISx and LATx bits to control their states

Note 1: All register bits are only reset in the case of a POR event outside of Deep Sleep mode.
2: Unlike all other events, a Deep Sleep BOR event will NOT cause a wake-up from Deep Sleep; this re-arms POR.

REGISTER 10-2: DSWAKE: DEEP SLEEP WAKE-UP SOURCE REGISTER[1]

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0, HS
—	—	—	—	—	—	—	DSINT0
bit 15							bit 8

R/W-0, HS	U-0	U-0	R/W-0, HS	R/W-0, HS	R/W-0, HS	U-0	R/W-0, HS
DSFLT	—	—	DSWDT	DSRTCC	DSMCLR	—	DSPOR[2,3]
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 **DSINT0:** Deep Sleep Interrupt-on-Change bit
1 = Interrupt-on-change was asserted during Deep Sleep
0 = Interrupt-on-change was not asserted during Deep Sleep

bit 7 **DSFLT:** Deep Sleep Fault Detect bit
1 = A Fault occurred during Deep Sleep and some Deep Sleep configuration settings may have been corrupted
0 = No Fault was detected during Deep Sleep

bit 6-5 **Unimplemented:** Read as '0'

bit 4 **DSWDT:** Deep Sleep Watchdog Timer Time-out bit
1 = The Deep Sleep Watchdog Timer timed out during Deep Sleep
0 = The Deep Sleep Watchdog Timer did not time out during Deep Sleep

bit 3 **DSRTCC:** Deep Sleep Real-Time Clock and Calendar (RTCC) Alarm bit
1 = The Real-Time Clock and Calendar triggered an alarm during Deep Sleep
0 = The Real-Time Clock and Calendar did not trigger an alarm during Deep Sleep

bit 2 **DSMCLR:** Deep Sleep $\overline{\text{MCLR}}$ Event bit
1 = The $\overline{\text{MCLR}}$ pin was active and was asserted during Deep Sleep
0 = The $\overline{\text{MCLR}}$ pin was not active, or was active, but not asserted during Deep Sleep

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

bit 0 **DSPOR:** Deep Sleep Power-on Reset Event bit[2,3]
1 = The V_{DD} supply POR circuit was active and a POR event was detected
0 = The V_{DD} supply POR circuit was not active, or was active, but did not detect a POR event

Note 1: All register bits are cleared when the DSEN (DSCON<15>) bit is set.
2: All register bits are reset only in the case of a POR event outside of Deep Sleep mode, except bit, DSPOR, which does not reset on a POR event that is caused due to a Deep Sleep exit.
3: Unlike the other bits in this register, this bit can be set outside of Deep Sleep.

10.3 Ultra Low-Power Wake-up

The Ultra Low-Power Wake-up (ULPWU) on pin, RB0, allows a slow falling voltage to generate an interrupt without excess current consumption.

To use this feature:

1. Charge the capacitor on RB0 by configuring the RB0 pin to an output and setting it to '1'.
2. Stop charging the capacitor by configuring RB0 as an input.
3. Discharge the capacitor by setting the ULPEN and ULPSINK bits in the ULPWCON register.
4. Configure Sleep mode.
5. Enter Sleep mode.

When the voltage on RB0 drops below V_{IL}, the device wakes up and executes the next instruction.

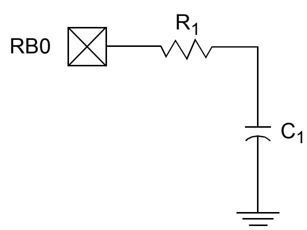
This feature provides a low-power technique for periodically waking up the device from Sleep mode.

The time-out is dependent on the discharge time of the RC circuit on RB0.

When the ULPWU module wakes the device from Sleep mode, the ULPWUIF bit (IFS5<0>) is set. Software can check this bit upon wake-up to determine the wake-up source. See Example 10-3 for initializing the ULPWU module.

A series resistor, between RB0 and the external capacitor, provides overcurrent protection for the RB0/AN0/ULPWU pin and enables software calibration of the time-out (see Figure 10-1).

FIGURE 10-1: SERIES RESISTOR



A timer can be used to measure the charge time and discharge time of the capacitor. The charge time can then be adjusted to provide the desired delay in Sleep. This technique compensates for the affects of temperature, voltage and component accuracy. The peripheral can also be configured as a simple, programmable Low-Voltage Detect (LVD) or temperature sensor.

EXAMPLE 10-3: ULTRA LOW-POWER WAKE-UP INITIALIZATION

```
//********************************
// 1. Charge the capacitor on RB0
//********************************
    TRISBbits.TRISB0 = 0;
    LATBbits.LATB0 = 1;
    for(i = 0; i < 10000; i++) Nop();
//******************************
//2. Stop Charging the capacitor
//   on RB0
//******************************
    TRISBbits.TRISB0 = 1;
//******************************
//3. Enable ULPWU Interrupt
//******************************
IFS5bits.ULPWUIF = 0;
IEC5bits.ULPWUIE = 1;
IPC21bits.ULPWUIP = 0x7;
//******************************
//4. Enable the Ultra Low Power
//   Wakeup module and allow
//   capacitor discharge
//******************************
    ULPWCONbits.ULPEN = 1;
    ULPWCONbit.ULPSINK = 1;
//******************************
//5. Enter Sleep Mode
//******************************
    Sleep();
//for sleep, execution will
//resume here
```

REGISTER 10-3: ULPWCON: ULPWU CONTROL REGISTER

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0
ULPEN	—	ULPSIDL	—	—	—	—	ULPSINK
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 **ULPEN:** ULPWU Module Enable bit
1 = Module is enabled
0 = Module is disabled

bit 14 **Unimplemented:** Read as ‘0’

bit 13 **ULPSIDL:** ULPWU Stop in Idle Select bit
1 = Discontinues module operation when the device enters Idle mode
0 = Continues module operation in Idle mode

bit 12-9 **Unimplemented:** Read as ‘0’

bit 8 **ULPSINK:** ULPWU Current Sink Enable bit
1 = Current sink is enabled
0 = Current sink is disabled

bit 7-0 **Unimplemented:** Read as ‘0’

10.4 Voltage Regulator-Based Power-Saving Features

The PIC24FV32KA304 series devices have a Voltage Regulator that has the ability to alter functionality to provide power savings. The on-board regulator is made up of two basic modules: the Voltage Regulator (VREG) and the Retention Regulator (RETREG). With the combination of VREG and RETREG, the following power modes are available:

10.4.1 RUN MODE

In Run mode, the main VREG is providing a regulated voltage with enough current to supply a device running at full speed, and the device is not in Sleep or Deep Sleep Mode. The Retention Regulator may or may not be running, but is unused.

10.4.2 SLEEP (STANDBY) MODE

In Sleep mode, the device is in Sleep and the main VREG is providing a regulated voltage at a reduced (standby) supply current. This mode provides for limited functionality due to the reduced supply current. It requires a longer time to wake-up from Sleep.

10.4.3 RETENTION SLEEP MODE

In Retention Sleep mode, the device is in Sleep and all regulated voltage is provided solely by the Retention Regulator. Consequently, this mode has lower power consumption than regular Sleep mode, but is also limited in terms of how much functionality can be enabled. Retention Sleep wake-up time is longer than Sleep mode due to the extra time required to raise the VCORE supply rail back to normal regulated levels.

> **Note:** PIC24F32KA30X family devices do not use an On-Chip Voltage Regulator, so they do not support Retention Sleep mode.

10.4.4 DEEP SLEEP MODE

In Deep Sleep mode, both the main Voltage Regulator and Retention Regulator are shut down, providing the lowest possible device power consumption. However, this mode provides no retention or functionality of the device and has the longest wake-up time.

TABLE 10-1: VOLTAGE REGULATION CONFIGURATION SETTINGS FOR PIC24FV32KA304 FAMILY DEVICES

RETCGF Bit (FPOR<2>)	RETEN Bit (RCON<12>)	PMSLP Bit (RCON<8>)	Power Mode During Sleep	Description
0	0	1	Sleep	VREG mode (normal) is unchanged during Sleep. RETREG is unused.
0	0	0	Sleep (Standby)	VREG goes to Low-Power Standby mode during Sleep. RETREG is unused.
0	1	0	Retention Sleep	VREG is off during Sleep. RETREG is enabled and provides Sleep voltage regulation.
1	X	1	Sleep	VREG mode (normal) is unchanged during Sleep. RETREG is disabled at all times.
1	X	0	Sleep (Standby)	VREG goes to Low-Power Standby mode during Sleep. RETREG is disabled at all times

10.5 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. Meanwhile, 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.

10.6 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 PMDx bits 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 PMDx 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. Power consumption is reduced, but not by as much as the PMDx bits are used. Most peripheral modules have an enable bit; exceptions include capture, 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 disables the module while in Idle mode, allowing further reduction of power consumption during Idle mode, enhancing power savings for extremely critical power applications.

11.0 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 on the I/O Ports, refer to the *"dsPIC33/PIC24 Family Reference Manual"*, **"I/O Ports with Peripheral Pin Select (PPS)"** (DS39711). Note that the PIC24FV32KA304 family devices do not support Peripheral Pin Select features.

All of the device pins (except VDD and VSS) are shared between the peripherals and the parallel I/O ports. All I/O input ports feature Schmitt Trigger inputs for improved noise immunity.

11.1 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 11-1 illustrates 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.

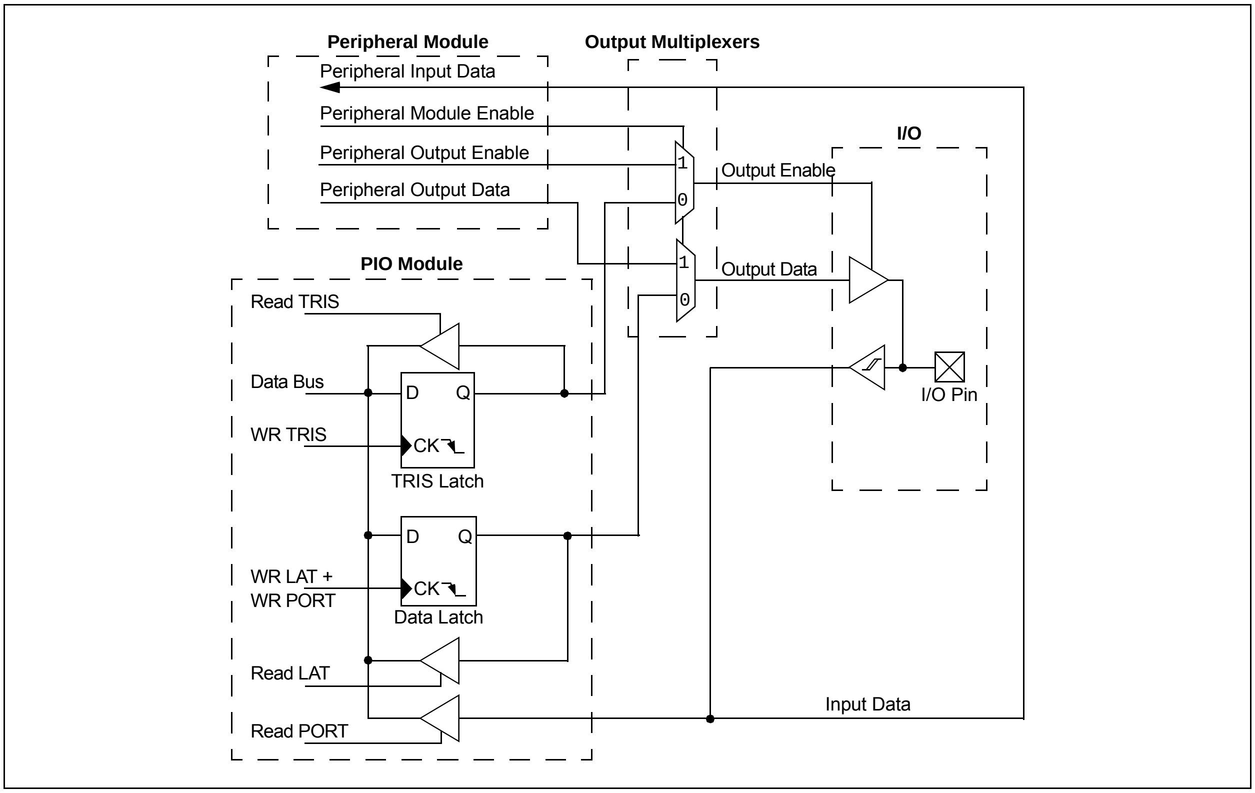
All port pins have three registers directly associated with their operation as digital I/O. The Data Direction register (TRISx) determines whether the pin is an input or an output. If the data direction bit is a '1', then the pin is an input. All port pins are defined as inputs after a Reset. Reads from the Data Latch register (LAT), read the latch. Writes to the latch, write the latch. Reads from the port (PORT), 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 nevertheless regarded as a dedicated port because there is no other competing source of outputs.

Note:	The I/O pins retain their state during Deep Sleep. They will retain this state at wake-up until the software restore bit (RELEASE) is cleared.

FIGURE 11-1: BLOCK DIAGRAM OF A TYPICAL SHARED PORT STRUCTURE



11.1.1 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 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 maximum open-drain voltage allowed is the same as the maximum V_{IH} specification.

11.2 Configuring Analog Port Pins

The use of the ANS and TRIS registers control the operation of the A/D port pins. The port pins that are desired as analog inputs must have their corresponding TRISx bit set (input). If the TRISx bit is cleared (output), the digital output level (V_{OH} or V_{OL}) will be converted.

When reading the PORT register, all pins configured as analog input channels will read as cleared (a low level). Analog levels on any pin that is defined as a digital input (including the ANx pins) may cause the input buffer to consume current that exceeds the device specifications.

11.2.1 ANALOG SELECTION REGISTERS

I/O pins with shared analog functionality, such as A/D inputs and comparator inputs, must have their digital inputs shut off when analog functionality is used. Note that analog functionality includes an analog voltage being applied to the pin externally.

To allow for analog control, the ANSx registers are provided. There is one ANS register for each port (ANSA, ANSB and ANSC). Within each ANSx register, there is a bit for each pin that shares analog functionality with the digital I/O functionality.

If a particular pin does not have an analog function, that bit is unimplemented. See Register 11-1 to Register 11-3 for implementation.

REGISTER 11-1: ANSA: ANALOG SELECTION (PORTA)

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/W-1	R/W-1	R/W-1	R/W-1
—	—	—	—	ANSA<3:0>			
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 15-4 **Unimplemented:** Read as ‘0’

bit 3-0 **ANSA<3:0>:** Analog Select Control bits
1 = Digital input buffer is not active (use for analog input)
0 = Digital input buffer is active

REGISTER 11-2: ANSB: ANALOG SELECTION (PORTB)

R/W-1	R/W-1	R/W-1	R/W-1	U-0	U-0	U-0	U-0
ANSB<15:12>				—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	ANSB<4:0>(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 **ANSB<15:12>:** Analog Select Control bits
1 = Digital input buffer is not active (use for analog input)
0 = Digital input buffer is active

bit 11-5 **Unimplemented:** Read as ‘0’

bit 4-0 **ANSB<4:0>:** Analog Select Control bits(1)
1 = Digital input buffer is not active (use for analog input)
0 = Digital input buffer is active

Note 1: The ANSB3 bit is not available on 20-pin devices.

REGISTER 11-3: ANSC ANALOG SELECTION (PORTC)

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-1	R/W-1
—	—	—	—	—	ANSC<2:0>(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 15-3 **Unimplemented:** Read as ‘0’

bit 2-0 **ANSC<2:0>:** Analog Select Control bits(1)
1 = Digital input buffer is not active (use for analog input)
0 = Digital input buffer is active

Note 1: These bits are not available on 20-pin or 28-pin devices.

11.2.2 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.

11.3 Input Change Notification (ICN)

The Input Change Notification function of the I/O ports allows the PIC24FV32KA304 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 23 external signals (CN0 through CN22) that may be selected (enabled) for generating an interrupt request on a Change-of-State.

There are six control registers associated with the ICN module. The CNEN1 and CNEN2 registers 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 also has a weak pull-up/pull-down connected to it. The pull-ups act as a current source that is connected to the pin. The pull-downs act as a current sink to eliminate the need for external resistors when push button or keypad devices are connected.

On any pin, only the pull-up resistor or the pull-down resistor should be enabled, but not both of them. If the push button or the keypad is connected to V_{DD}, enable the pull-down, or if they are connected to V_{SS}, enable the pull-up resistors. The pull-ups are enabled separately, using the CNPU1 and CNPU2 registers, which contain the control bits for each of the CN pins.

Setting any of the control bits enables the weak pull-ups for the corresponding pins. The pull-downs are enabled separately, using the CNPD1 and CNPD2 registers, which contain the control bits for each of the CN pins. Setting any of the control bits enables the weak pull-downs for the corresponding pins.

When the internal pull-up is selected, the pin uses V_{DD} as the pull-up source voltage. When the internal pull-down is selected, the pins are pulled down to V_{SS} by an internal resistor. Make sure that there is no external pull-up source/pull-down sink when the internal pull-ups/pull-downs are enabled.

Note:	Pull-ups and pull-downs on Change Notification pins should always be disabled whenever the port pin is configured as a digital output.

EXAMPLE 11-1: PORT WRITE/READ EXAMPLE

```
MOV   0xFF00, W0;          //Configure PORTB<15:8> as inputs and PORTB<7:0> as outputs
MOV   W0, TRISB;
NOP;                       //Delay 1 cycle
BTSS  PORTB, #13;          //Next Instruction

Equivalent ‘C’ Code
TRISB = 0xFF00;            //Configure PORTB<15:8> as inputs and PORTB<7:0> as outputs
NOP();                     //Delay 1 cycle
if(PORTBbits.RB13 == 1)    // execute following code if PORTB pin 13 is set.
{
}
```

12.0 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 on Timers, refer to the *"dsPIC33/PIC24 Family Reference Manual"*, **"Timers"** (DS39704).

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:

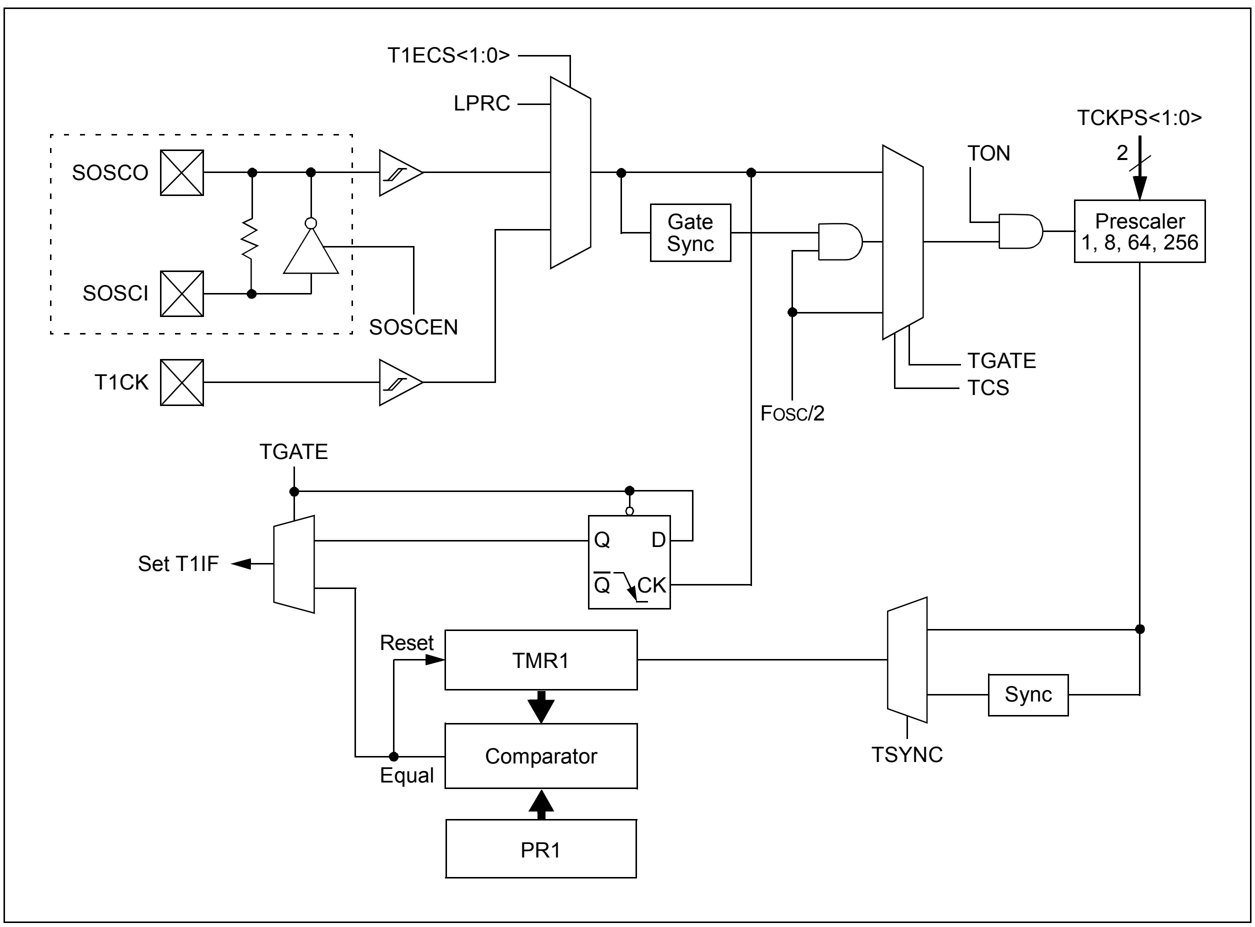
- Timer Gate Operation
- Selectable Prescaler Settings
- Timer Operation During CPU Idle and Sleep modes
- Interrupt on 16-Bit Period Register Match or Falling Edge of External Gate Signal

Figure 12-1 illustrates a block diagram of the 16-bit Timer1 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 Timer1 Interrupt Enable bit, T1IE. Use the Timer1 Interrupt Priority bits, T1IP<2:0>, to set the interrupt priority.

FIGURE 12-1: 16-BIT TIMER1 MODULE BLOCK DIAGRAM



REGISTER 12-1: T1CON: TIMER1 CONTROL REGISTER

R/W-0	U-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
TON	—	TSIDL	—	—	—	T1ECS1[1]	T1ECS0[1]
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:** Timer1 Stop in Idle Mode bit
1 = Discontinues module operation when device enters Idle mode
0 = Continues module operation in Idle mode

bit 12-10 **Unimplemented:** Read as '0'

bit 9-8 **T1ECS<1:0>:** Timer1 Extended Clock Select bits[1]
11 = Reserved; do not use
10 = Timer1 uses the LPRC as the clock source
01 = Timer1 uses the external clock from T1CK
00 = Timer1 uses the Secondary Oscillator (SOSC) as the clock source

bit 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 is enabled
0 = Gated time accumulation is 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 = Synchronizes external clock input
0 = Does not synchronize external clock input
When TCS = 0:
This bit is ignored.

bit 1 **TCS:** Timer1 Clock Source Select bit
1 = Timer1 clock source is selected by T1ECS<1:0>
0 = Internal clock (FOSC/2)

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

Note 1: The T1ECSx bits are valid only when TCS = 1.

13.0 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 on Timers, refer to the *"dsPIC33/PIC24 Family Reference Manual"*, **"Timers"** (DS39704).

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 a 32-bit timer, Timer2/3 or Timer4/5 operate in three modes:

- Two Independent 16-Bit Timers (Timer2 and Timer3) 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 mode
- Interrupt on a 32-Bit Period Register Match
- A/D Event Trigger

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 A/D event trigger (this is implemented only with Timer3). The operating modes and enabled features are determined by setting the appropriate bit(s) in the T2CON, T3CON, T4CON and T5CON registers. The T2CON,T3CON, T4CON and T5CON registers are provided in generic form in Register 13-1 and Register 13-2, respectively.

For 32-bit timer/counter operation, Timer2/Timer4 is the least significant word (lsw) and Timer3/Timer5 is the most significant word (msw) of the 32-bit timer.

Note: For 32-bit operation, T3CON or T5CON control bits are ignored. Only T2CON or T4CON control bits are used for setup and control. Timer2 or 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:

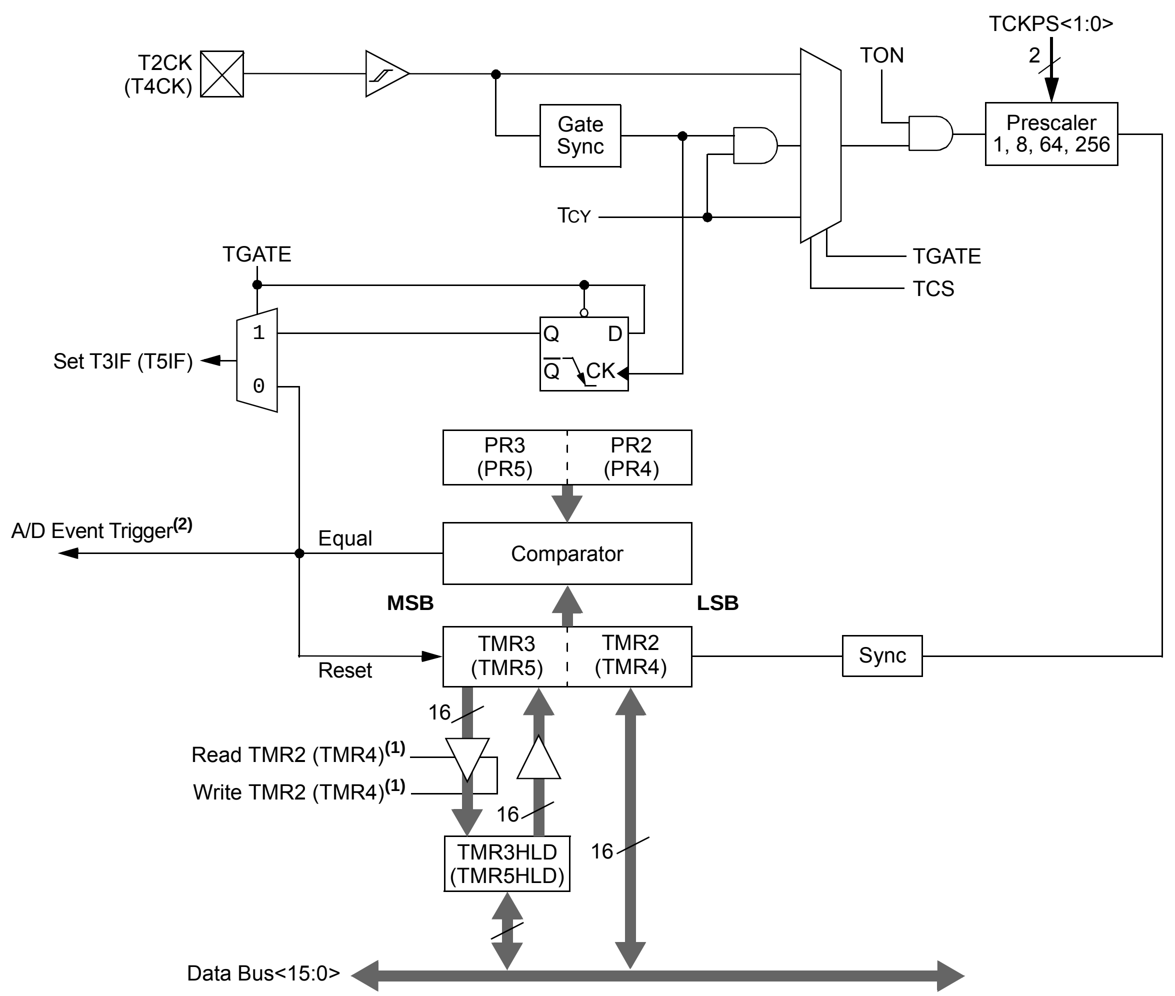
1. Set the T32 bit (T2CON<3> or T4CON<3> = `1`).
2. Select the prescaler ratio for Timer2 or Timer4 using the TCKPS<1:0> bits.
3. Set the Clock and Gating modes using the TCS and TGATE bits.
4. Load the timer period value. PR3 (or PR5) will contain the most significant word of the value, while PR2 (or PR4) contains the least significant word.
5. If interrupts are required, set the Timerx Interrupt Enable bit, TxIE. Use the Timerx Interrupt Priority bits, TxIP<2:0>, to set the interrupt priority.
6. Set the TON bit (TxCON<15> = `1`).

The timer value, at any point, is stored in the register pair, TMR3:TMR2 (or TMR5:TMR4). 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.
4. Load the timer period value into the PRx register.
5. If interrupts are required, set the Timerx Interrupt Enable bit, TxIE; use the Timerx Interrupt Priority bits, TxIP<2:0>, to set the interrupt priority.
6. Set the TON bit (TxCON<15> = `1`).

FIGURE 13-1: 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 A/D event trigger is available only on Timer2/3 and Timer4/5 in 32-bit mode, and Timer3 and Timer5 in 16-bit mode.

FIGURE 13-2: TIMER2 AND TIMER4 (16-BIT SYNCHRONOUS) BLOCK DIAGRAM

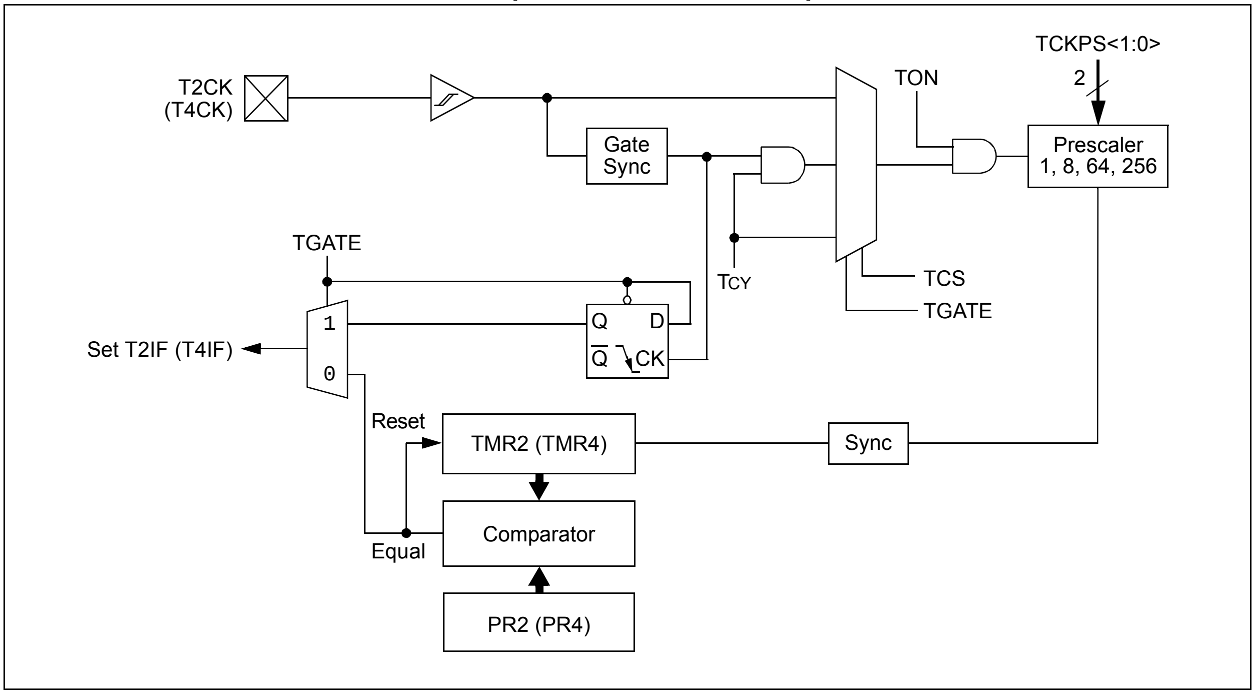
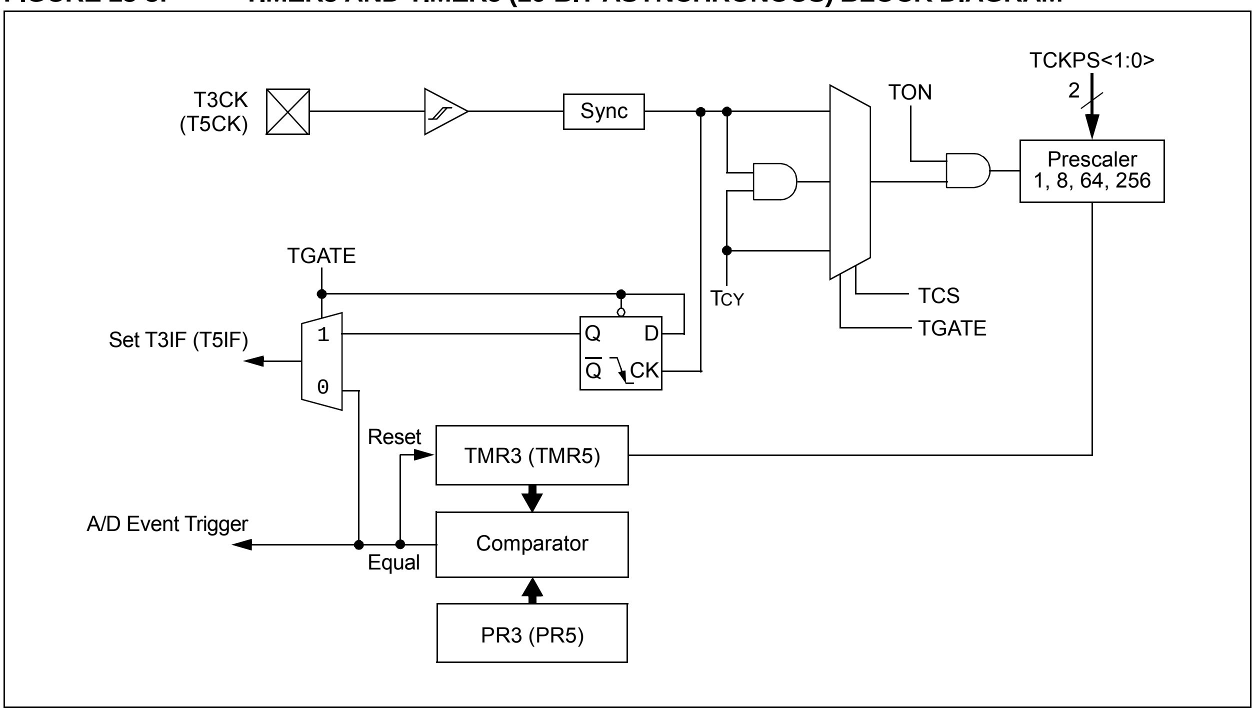


FIGURE 13-3: TIMER3 AND TIMER5 (16-BIT ASYNCHRONOUS) BLOCK DIAGRAM



REGISTER 13-1: TxCON: TIMER2 AND TIMER4 CONTROL REGISTER

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	—
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:** Timerx Stop in Idle Mode bit
1 = Discontinues module operation when device enters Idle mode
0 = Continues 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 is enabled
0 = Gated time accumulation is 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 = Timer2 and Timer3 or Timer4 and Timer5 form a single 32-bit timer
0 = Timer2 and Timer3 or Timer4 and Timer5 act as two 16-bit timers

bit 2 **Unimplemented:** Read as ‘0’

bit 1 **TCS:** Timerx Clock Source Select bit
1 = External clock from pin, TxCK (on the rising edge)
0 = Internal clock (FOSC/2)

bit 0 **Unimplemented:** Read as ‘0’

Note 1: In 32-bit mode, the T3CON or T5CON control bits do not affect 32-bit timer operation.

REGISTER 13-2: TyCON: TIMER3 AND TIMER5 CONTROL REGISTER

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]	—
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:** Timery Stop in Idle Mode bit[1]
1 = Discontinues module operation when device enters Idle mode
0 = Continues 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 is enabled
0 = Gated time accumulation is 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]
1 = External clock is from the T3CK pin (on the rising edge)
0 = Internal clock (FOSC/2)

bit 0 **Unimplemented:** Read as ‘0’

Note 1: When 32-bit operation is enabled (TxCON<3> = 1), these bits have no effect on Timery operation. All timer functions are set through the TxCON register.

NOTES:

14.0 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 *"dsPIC33/PIC24 Family Reference Manual"*, **"Input Capture with Dedicated Timer"** (DS70000352).

All devices in the PIC24FV32KA304 family feature three 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 20 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 14-1) and ICxCON2 (Register 14-2). A general block diagram of the module is shown in Figure 14-1.

14.1 General Operating Modes

14.1.1 SYNCHRONOUS AND TRIGGER MODES

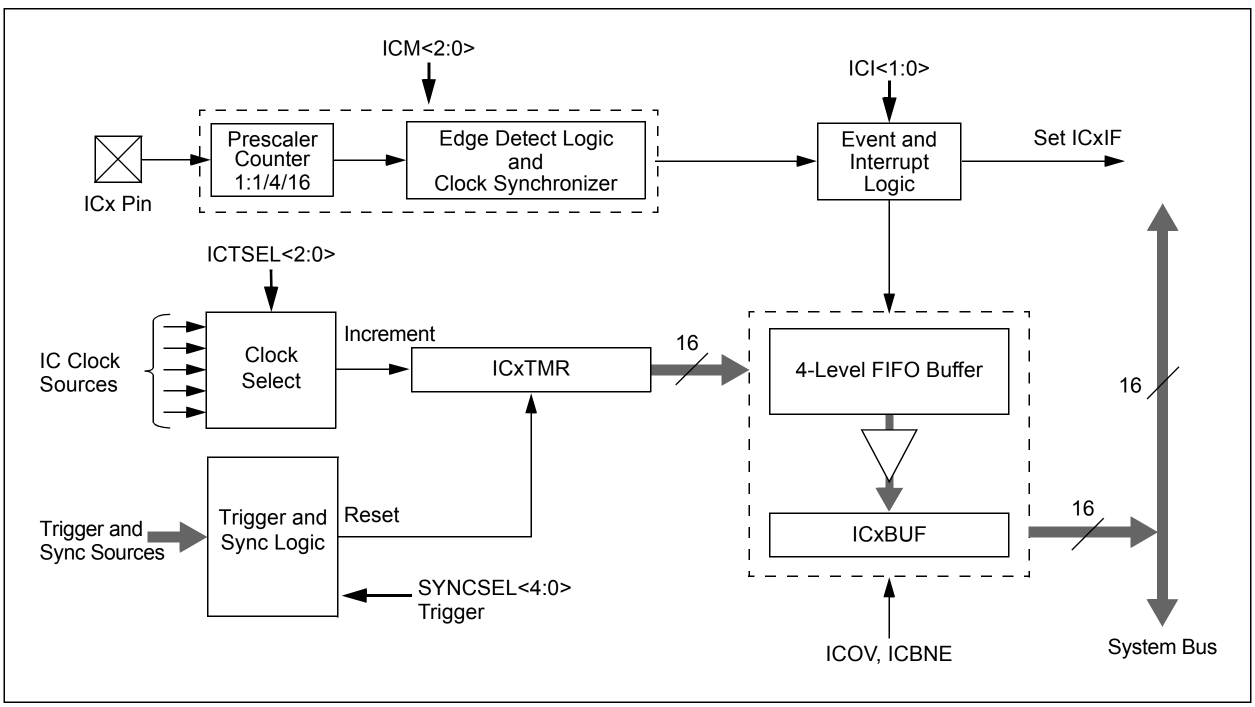
By default, the input capture module operates in a Free-Running mode. The internal 16-bit counter, ICxTMR, counts up continuously, wrapping around from FFFFh to 0000h on each overflow, with its period synchronized to the selected external clock source. When a capture event occurs, the current 16-bit 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 SYNCSELx bits to '`00000`' and clearing the ICTRIG bit (ICxCON2<7>). Synchronous and Trigger modes are selected any time the SYNCSELx 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 SYNCSELx bits determine the Sync/trigger source.

When the SYNCSELx 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 14-1: INPUT CAPTURE x BLOCK DIAGRAM



14.1.2 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 numbered module (ICy) provides the Most Significant 16 bits. Wraparounds of the ICx registers cause an increment of their corresponding ICy registers.

Cascaded operation is configured in hardware by setting the IC32 bit (ICxCON2<8>) for both modules.

14.2 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. If Synchronous mode is to be used, disable the Sync source before proceeding.
2. Make sure that any previous data has been removed from the FIFO by reading ICxBUF until the ICBNE bit (ICxCON1<3>) is cleared.
3. Set the SYNCSELx bits (ICxCON2<4:0>) to the desired Sync/trigger source.
4. Set the ICTSELx bits (ICxCON1<12:10>) for the desired clock source. If the desired clock source is running, set the ICTSELx bits before the input capture module is enabled, for proper synchronization with the desired clock source.
5. Set the ICIx bits (ICxCON1<6:5>) to the desired interrupt frequency.
6. Select Synchronous or Trigger mode operation:
   a) Check that the SYNCSELx 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>).
7. Set the ICMx bits (ICxCON1<2:0>) to the desired operational mode.
8. 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 ICTSELx and SYNCSELx 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 ICTSELx and SYNCSELx bit 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 ICIx bits (ICxCON1<6:5>) to 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 ICMx 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 14-1: 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	ICM1	ICM0
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 x Timer Select bits
111 = System clock (FOSC/2)
110 = Reserved
101 = Reserved
100 = Timer1
011 = Timer5
010 =T imer4
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
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 is turned off

REGISTER 14-2: 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:** Input Capture x Sync/Trigger Select bit
1 = Triggers ICx from source designated by the SYNCSELx bits
0 = Synchronizes ICx with 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>:** Trigger/Synchronization Source Selection bits
11111 = Reserved
11110 = Reserved
11101 = Reserved
11100 = CTMU(1)
11011 = A/D(1)
11010 = Comparator 3(1)
11001 = Comparator 2(1)
11000 = Comparator 1(1)
10111 = Input Capture 4
10110 = Input Capture 3
10101 = Input Capture 2
10100 = Input Capture 1
10011 = Reserved
10010 = Reserved
1000x = Reserved
01111 = Timer5
01110 = Timer4
01101 = Timer3
01100 = Timer2
01011 = Timer1
01010 = Input Capture 5
01001 = Reserved
01000 = Reserved
00111 = Reserved
00110 = Reserved
00101 = Output Compare 5
00100 = Output Compare 4
00011 = Output Compare 3
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.

15.0 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 *"dsPIC33/PIC24 Family Reference Manual"*, **"Output Compare with Dedicated Timer"** (DS70005159).

All devices in the PIC24FV32KA304 family feature 3 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. Also, the modules can produce Pulse-Width Modulated (PWM) 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 21 user-selectable Sync/trigger 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

15.1 General Operating Modes

15.1.1 SYNCHRONOUS AND TRIGGER MODES

By default, the output compare module operates in a Free-Running mode. The internal 16-bit counter, OCxTMR, counts up continuously, wrapping around from FFFFh to 0000h 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.

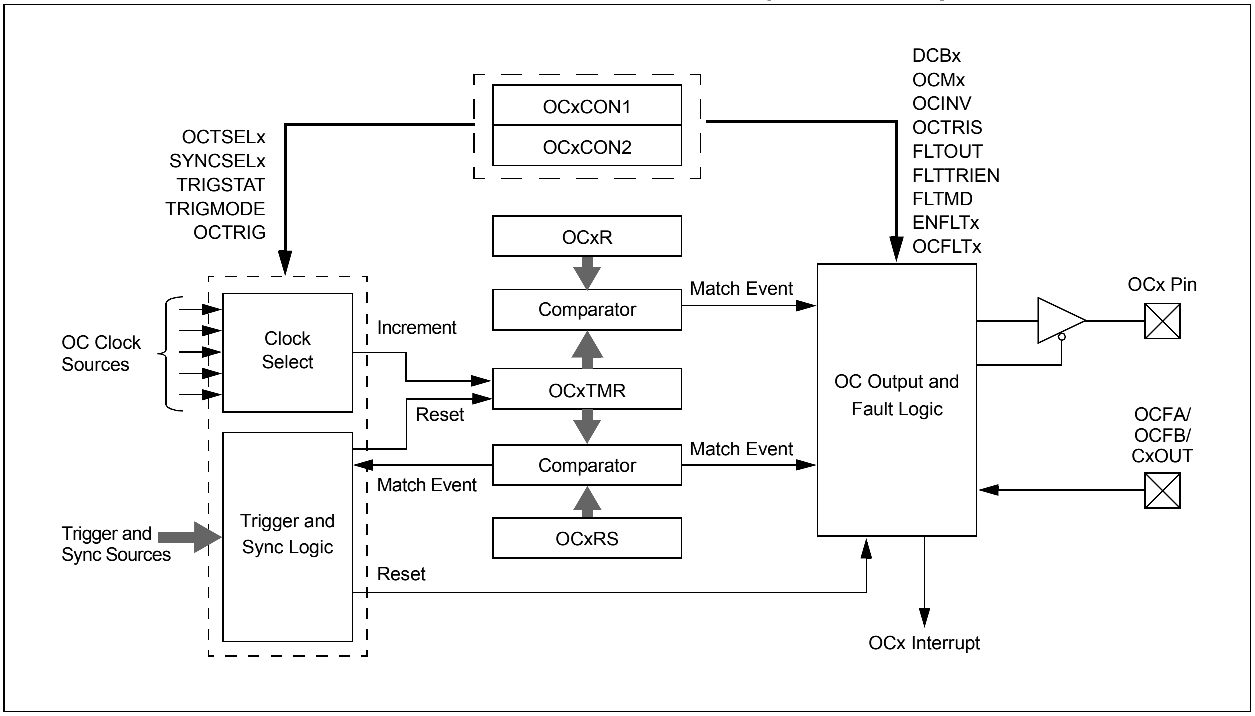
Free-Running mode is selected by default or any time that the SYNCSELx bits (OCxCON2<4:0>) are set to '`00000`'. Synchronous or Trigger modes are selected any time the SYNCSELx bits are set to any value except '`00000`'. The OCTRIG bit (OCxCON2<7>) selects either Synchronous or Trigger mode. Setting this bit selects Trigger mode operation. In both modes, the SYNCSELx bits determine the Sync/trigger source.

15.1.2 CASCADED (32-BIT) MODE

By default, each module operates independently with its own set of 16-bit Timer and Duty Cycle registers. To increase the range, adjacent even and odd numbered 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 numbered module (OCy) provides the Most Significant 16 bits. Wraparounds 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.

FIGURE 15-1: OUTPUT COMPARE x BLOCK DIAGRAM (16-BIT MODE)



15.2 Compare Operations

In Compare mode (Figure 15-1), 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. 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 the 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.
2. Write the rising edge value to OCxR and the falling edge value to OCxRS.
3. For Trigger mode operations, set OCTRIG to enable Trigger mode. Set or clear TRIGMODE to configure the trigger operation and TRIGSTAT to select a hardware or software trigger. For Synchronous mode, clear OCTRIG.
4. Set the SYNCSEL<4:0> bits to configure the trigger or synchronization source. If free-running timer operation is required, set the SYNCSELx bits to ‘`00000`’ (no Sync/trigger source).
5. Select the time base source with the OCTSEL<2:0> bits. If the desired clock source is running, set the OCTSEL<2:0> bits before the output compare module is enabled for proper synchronization with the desired clock source. If necessary, set the TON bit for the selected timer which enables the compare time base to count. Synchronous mode operation starts as soon as the synchronization source is enabled; Trigger mode operation starts after a trigger source event occurs.
6. Set the OCM<2:0> bits for the appropriate compare operation (‘`0xx`’).

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>), TRIGSTAT (OCxCON2<6>) and SYNCSELx (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.

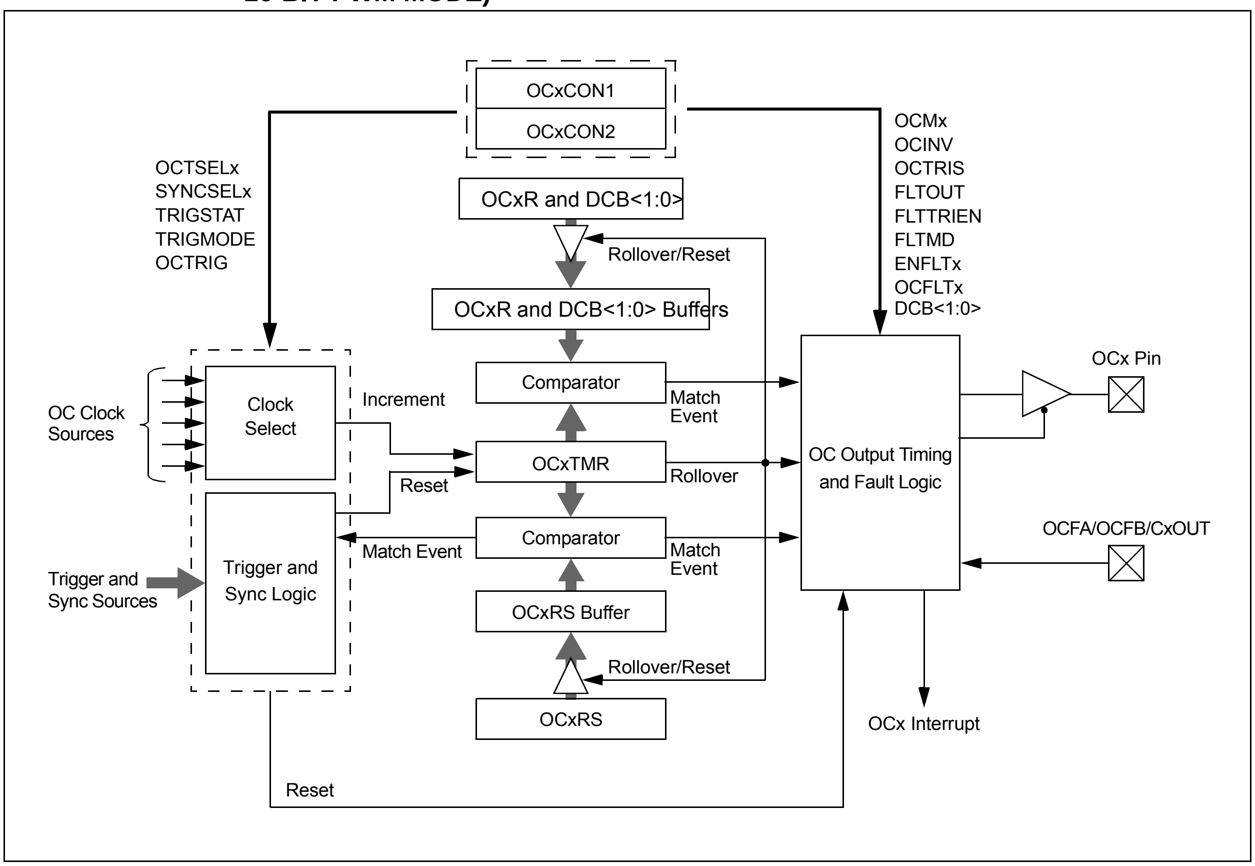
15.3 Pulse-Width Modulation (PWM) Mode

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 edge-aligned PWM operation:

1. Calculate the desired ON time and load it into the OCxR register.
2. Calculate the desired period and load it into the OCxRS register.
3. Select the current OCx as the synchronization source by writing 0x1F to SYNCSEL<4:0> (OCxCON2<4:0>) and '0' to OCTRIG (OCxCON2<7>).
4. Select a clock source by writing the OCTSEL2<2:0> (OCxCON<12:10>) bits.
5. Enable interrupts, if required, for the timer and output compare modules. The output compare interrupt is required for PWM Fault pin utilization.
6. Select the desired PWM mode in the OCM<2:0> (OCxCON1<2:0>) bits.
7. If a timer is selected as a clock source, set the TMRy prescale value and enable the time base by setting the TON (TxCON<15>) bit.

FIGURE 15-2: OUTPUT COMPARE x BLOCK DIAGRAM (DOUBLE-BUFFERED, 16-BIT PWM MODE)



15.3.1 PWM PERIOD

In Edge-Aligned PWM mode, the period is specified by the value of the OCxRS register. In Center-Aligned PWM mode, the period of the synchronization source, such as the Timers' PRy, specifies the period. The period in both cases can be calculated using Equation 15-1.

EQUATION 15-1: CALCULATING THE PWM PERIOD[(1)]

PWM Period = [Value + 1] x TCY x (Prescaler Value)

Where:

Value = OCxRS in Edge-Aligned PWM mode and can be PRy in Center-Aligned PWM mode (if TMRy is the Sync source).

Note 1: Based on TCY = TOSC * 2; Doze mode and PLL are disabled.

15.3.2 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 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:

- Edge-Aligned PWM:
  - If OCxR and OCxRS are loaded with 0000h, the OCx pin will remain low (0% duty cycle).
  - If OCxRS is greater than OCxR, the pin will remain high (100% duty cycle).
- Center-Aligned PWM (with TMRy as the Sync source):
  - 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 go high (100% duty cycle).

See Example 15-3 for PWM mode timing details. Table 15-1 and Table 15-2 show example PWM frequencies and resolutions for a device operating at 4 MIPS and 10 MIPS, respectively.

EQUATION 15-2: CALCULATION FOR MAXIMUM PWM RESOLUTION[(1)]

$$\text{Maximum PWM Resolution (bits)} = \frac{\log_{10}\left(\dfrac{F_{CY}}{F_{PWM} \bullet (\text{Prescale Value})}\right)}{\log_{10}(2)} \text{ bits}$$

Note 1: Based on FCY = FOSC/2; Doze mode and PLL are disabled.

EQUATION 15-3: PWM PERIOD AND DUTY CYCLE CALCULATIONS[(1)]

1. Find the OCxRS register value for a desired PWM frequency of 52.08 kHz, where FOSC = 8 MHz with PLL (32 MHz device clock rate) and a prescaler setting of 1:1 using Edge-Aligned PWM mode:

$$\begin{aligned} T_{CY} = 2 \bullet T_{OSC} &= 62.5\text{ ns} \\ \text{PWM Period} &= 1/\text{PWM Frequency} = 1/52.08\text{ kHz} = 19.2\ \mu\text{s} \\ \text{PWM Period} &= (\text{OCxRS} + 1) \bullet T_{CY} \bullet (\text{OCx Prescale Value}) \\ 19.2\ \mu\text{s} &= (\text{OCxRS} + 1) \bullet 62.5\text{ ns} \bullet 1 \\ \text{OCxRS} &= 306 \end{aligned}$$

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:

$$\begin{aligned} \text{PWM Resolution} &= \log_{10}(F_{CY}/F_{PWM})/\log_{10}2)\text{ bits} \\ &= (\log_{10}(16\text{ MHz}/52.08\text{ kHz})/\log_{10}2)\text{ bits} \\ &= 8.3\text{ bits} \end{aligned}$$

Note 1: Based on TCY = 2 * TOSC; Doze mode and PLL are disabled.

15.4 Subcycle Resolution

The DCBx bits (OCxCON2<10:9>) provide for resolution better than one instruction cycle. When used, they delay the falling edge generated from a match event by a portion of an instruction cycle.

For example, setting DCB<1:0> = `10` causes the falling edge to occur halfway through the instruction cycle in which the match event occurs, instead of at the beginning. These bits cannot be used when OCM<2:0> = `001`. When operating the module in PWM mode (OCM<2:0> = `110` or `111`), the DCBx bits will be double-buffered.

The DCBx bits are intended for use with a clock source identical to the system clock. When an OCx module with enabled prescaler is used, the falling edge delay caused by the DCBx bits will be referenced to the system clock period, rather than the OCx module's period.

TABLE 15-1: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 4 MIPS (F_{CY} = 4 MHz)(1)

PWM Frequency	7.6 Hz	61 Hz	122 Hz	977 Hz	3.9 kHz	31.3 kHz	125 kHz
Prescaler Ratio	8	1	1	1	1	1	1
Period Value	FFFFh	FFFFh	7FFFh	0FFFh	03FFh	007Fh	001Fh
Resolution (bits)	16	16	15	12	10	7	5

Note 1: Based on F_{CY} = $F_{OSC}/2$; Doze mode and PLL are disabled.

TABLE 15-2: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 16 MIPS (F_{CY} = 16 MHz)(1)

PWM Frequency	30.5 Hz	244 Hz	488 Hz	3.9 kHz	15.6 kHz	125 kHz	500 kHz
Prescaler Ratio	8	1	1	1	1	1	1
Period Value	FFFFh	FFFFh	7FFFh	0FFFh	03FFh	007Fh	001Fh
Resolution (bits)	16	16	15	12	10	7	5

Note 1: Based on F_{CY} = $F_{OSC}/2$; Doze mode and PLL are disabled.

REGISTER 15-1: 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	ENFLT1
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	OCFLT2	OCFLT1	OCFLT0	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:** Output Compare x Stop 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 = System clock
110 = Reserved
101 = Reserved
100 = Timer1
011 = Timer5
010 = Timer4
001 = Timer3
000 = Timer2

bit 9 **ENFLT2:** Comparator Fault Input Enable bit
1 = Comparator Fault input is enabled
0 = Comparator Fault input is disabled

bit 8 **ENFLT1:** OCFB Fault Input Enable bit
1 = OCFB Fault input is enabled
0 = OCFB Fault input is disabled

bit 7 **ENFLT0:** OCFA Fault Input Enable bit
1 = OCFA Fault input is enabled
0 = OCFA Fault input is disabled

bit 6 **OCFLT2:** PWM Comparator Fault Condition Status bit
1 = PWM comparator Fault condition has occurred (this is cleared in hardware only)
0 = PWM comparator Fault condition has not occurred (this bit is used only when OCM<2:0> = 111)

bit 5 **OCFLT1:** PWM OCFB Fault Input Enable bit
1 = PWM OCFB Fault condition has occurred (this is cleared in hardware only)
0 = PWM OCFB Fault condition has not occurred (this bit is used only when OCM<2:0> = 111)

bit 4 **OCFLT0:** PWM OCFA Fault Condition Status bit
1 = PWM OCFA Fault condition has occurred (this is cleared in hardware only)
0 = PWM OCFA Fault condition has not occurred (this bit is used only when OCM<2:0> = 111)

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

Note 1: The comparator module used for Fault input varies with the OCx module. OC1 and OC2 use Comparator 1; OC3 and OC4 use Comparator 2; OC5 uses Comparator 3.

REGISTER 15-1: OCxCON1: OUTPUT COMPARE x CONTROL REGISTER 1 (CONTINUED)

bit 2-0	**OCM<2:0>:** Output Compare x Mode Select bits[1] `111` = Center-Aligned PWM mode on OCx `110` = Edge-Aligned PWM mode on OCx `101` = Double Compare Continuous Pulse mode: Initialize OCx pin low; toggle OCx state continuously on alternate matches of OCxR and OCxRS `100` = Double Compare Single-Shot mode: Initialize OCx pin low; toggle 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 comparator module used for Fault input varies with the OCx module. OC1 and OC2 use Comparator 1; OC3 and OC4 use Comparator 2; OC5 uses Comparator 3.

REGISTER 15-2: 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 OCFLTx 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:** Output Compare x Invert bit
1 = OCx output is inverted
0 = OCx output is not inverted

bit 11 **Unimplemented:** Read as ‘0’

bit 10-9 **DCB<1:0>:** Output Compare x Pulse-Width Least Significant bits(3)
11 = Delays OCx falling edge by 3/4 of the instruction cycle
10 = Delays OCx falling edge by 1/2 of the instruction cycle
01 = Delays OCx falling edge by 1/4 of the instruction cycle
00 = OCx falling edge occurs at the start of the instruction cycle

bit 8 **OC32:** Cascade Two Output Compare Modules Enable bit (32-bit operation)
1 = Cascade module operation is enabled
0 = Cascade module operation is disabled

bit 7 **OCTRIG:** Output Compare x Sync/Trigger Select bit
1 = Triggers OCx from source designated by the SYNCSELx bits
0 = Synchronizes OCx with 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:** Output Compare x Output Pin Direction Select bit
1 = OCx pin is tri-stated
0 = Output Compare x peripheral is connected to the OCx pin

Note 1: Do not use an output compare module as its own trigger source, either by selecting this mode or another equivalent SYNCSELx setting.
2: Use these inputs as trigger sources only and never as Sync sources.
3: These bits affect the rising edge when OCINV = 1. The bits have no effect when the OCMx bits (OCxCON1<2:0>) = 001.

REGISTER 15-2: OCxCON2: OUTPUT COMPARE x CONTROL REGISTER 2 (CONTINUED)

bit 4-0 **SYNCSEL<4:0>:** Trigger/Synchronization Source Selection bits

`11111` = This output compare module[1]
`11110` = Reserved
`11101` = Reserved
`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]
`100xx` = Reserved
`01111` = Timer5
`01110` = Timer4
`01101` = Timer3
`01100` = Timer2
`01011` = Timer1
`01010` = Input Capture 5[2]
`01001` = Reserved
`01000` = Reserved
`00111` = Reserved
`00110` = Reserved
`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: Do not use an output compare module as its own trigger source, either by selecting this mode or another equivalent SYNCSELx setting.

2: Use these inputs as trigger sources only and never as Sync sources.

3: These bits affect the rising edge when OCINV = `1`. The bits have no effect when the OCMx bits (OCxCON1<2:0>) = `001`.

16.0 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 on the Serial Peripheral Interface, refer to the *"dsPIC33/PIC24 Family Reference Manual"*, **"Serial Peripheral Interface (SPI)"** (DS70005185).

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 data EEPROMs, shift registers, display drivers, A/D Converters, etc. The SPI module is compatible with Motorola® SPI and SIOP interfaces.

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 SPI1BUF 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:

- SDI1: Serial Data Input
- SDO1: Serial Data Output
- SCK1: Shift Clock Input or Output
- $\overline{\text{SS1}}$: 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, $\overline{\text{SS1}}$ is not used. In the 2-pin mode, both SDO1 and $\overline{\text{SS1}}$ are not used.

Block diagrams of the module, in Standard and Enhanced Buffer modes, are shown in Figure 16-1 and Figure 16-2.

The devices of the PIC24FV32KA304 family offer two SPI modules on a device.

Note:	In this section, the SPI modules are referred to as SPIx. Special Function Registers (SFRs) will follow a similar notation. For example, SPI1CON1 or SPI1CON2 refers to the control register for the SPI1 module.

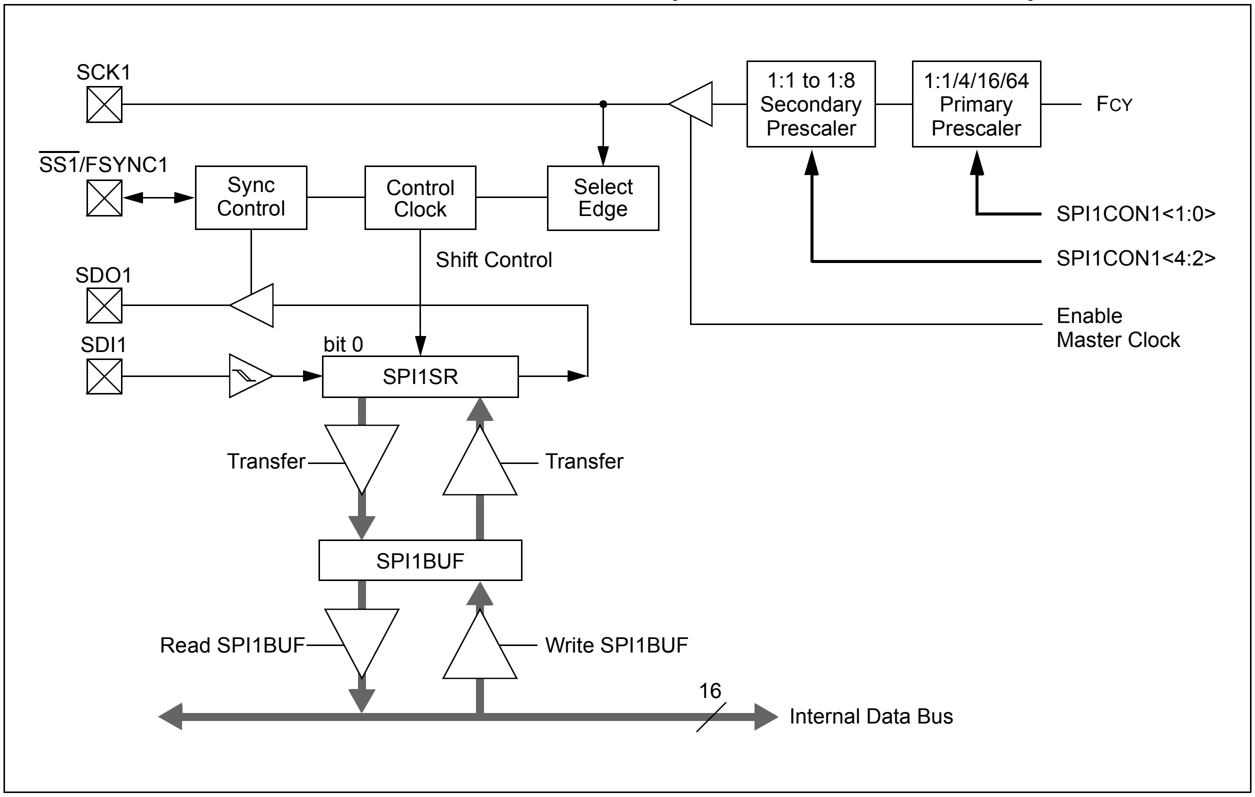
To set up the SPI1 module for the Standard Master mode of operation:

1. If using interrupts:
   a) Clear the SPI1IF bit in the IFS0 register.
   b) Set the SPI1IE bit in the IEC0 register.
   c) Write the respective SPI1IPx bits in the IPC2 register to set the interrupt priority.
2. Write the desired settings to the SPI1CON1 and SPI1CON2 registers with the MSTEN bit (SPI1CON1<5>) = `1`.
3. Clear the SPIROV bit (SPI1STAT<6>).
4. Enable SPI operation by setting the SPIEN bit (SPI1STAT<15>).
5. Write the data to be transmitted to the SPI1BUF register. Transmission (and reception) will start as soon as data is written to the SPI1BUF register.

To set up the SPI1 module for the Standard Slave mode of operation:

1. Clear the SPI1BUF register.
2. If using interrupts:
   a) Clear the SPI1IF bit in the IFS0 register.
   b) Set the SPI1IE bit in the IEC0 register.
   c) Write the respective SPI1IPx bits in the IPC2 register to set the interrupt priority.
3. Write the desired settings to the SPI1CON1 and SPI1CON2 registers with the MSTEN bit (SPI1CON1<5>) = `0`.
4. Clear the SMP bit.
5. If the CKE bit is set, then the SSEN bit (SPI1CON1<7>) must be set to enable the $\overline{\text{SS1}}$ pin.
6. Clear the SPIROV bit (SPI1STAT<6>).
7. Enable SPI operation by setting the SPIEN bit (SPI1STAT<15>).

FIGURE 16-1: SPI1 MODULE BLOCK DIAGRAM (STANDARD BUFFER MODE)



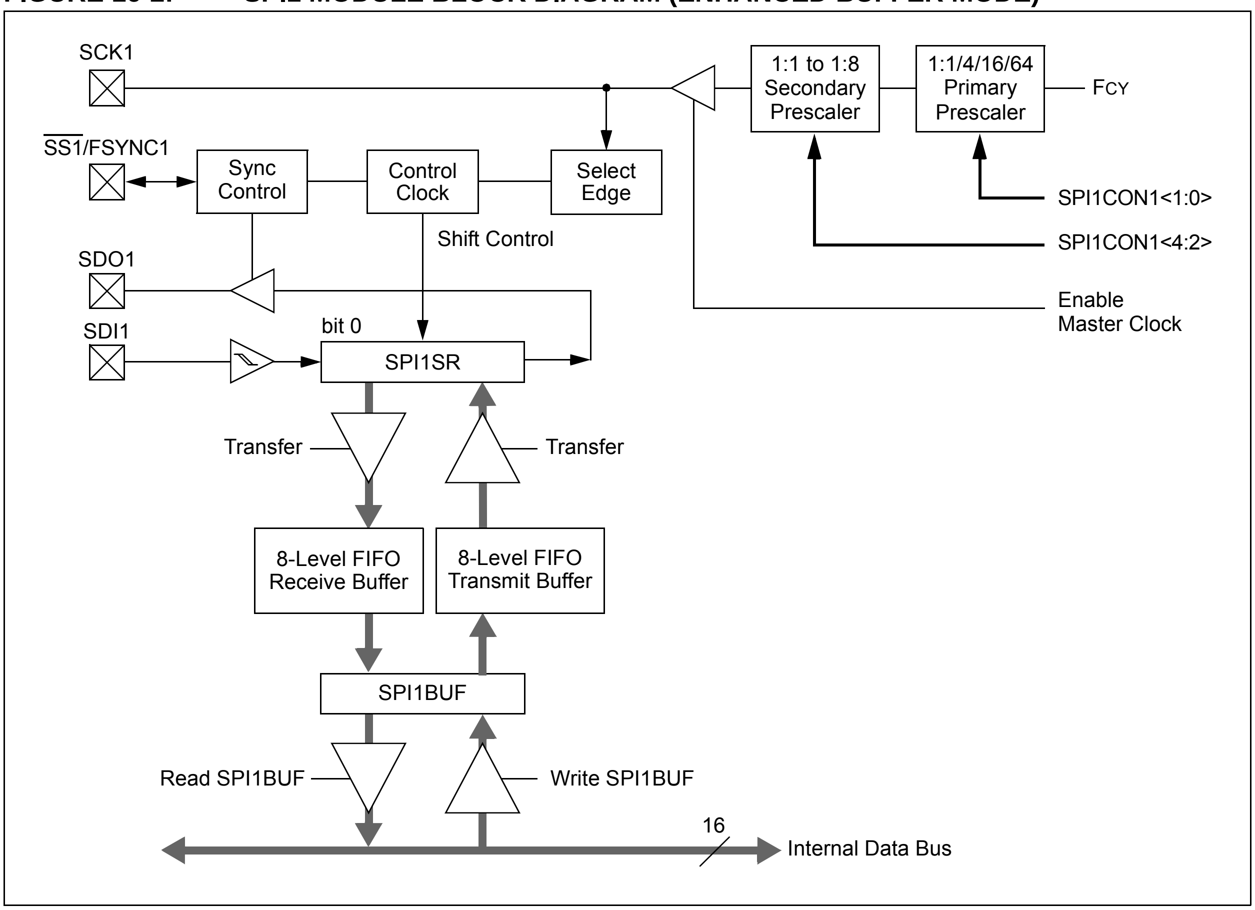
To set up the SPI1 module for the Enhanced Buffer Master (EBM) mode of operation:

1. If using interrupts:
   a) Clear the SPI1IF bit in the IFS0 register.
   b) Set the SPI1IE bit in the IEC0 register.
   c) Write the respective SPI1IPx bits in the IPC2 register.
2. Write the desired settings to the SPI1CON1 and SPI1CON2 registers with the MSTEN bit (SPI1CON1<5>) = `1`.
3. Clear the SPIROV bit (SPI1STAT<6>).
4. Select Enhanced Buffer mode by setting the SPIBEN bit (SPI1CON2<0>).
5. Enable SPI operation by setting the SPIEN bit (SPI1STAT<15>).
6. Write the data to be transmitted to the SPI1BUF register. Transmission (and reception) will start as soon as data is written to the SPI1BUF register.

To set up the SPI1 module for the Enhanced Buffer Slave mode of operation:

1. Clear the SPI1BUF register.
2. If using interrupts:
   a) Clear the SPI1IF bit in the IFS0 register.
   b) Set the SPI1IE bit in the IEC0 register.
   c) Write the respective SPI1IPx bits in the IPC2 register to set the interrupt priority.
3. Write the desired settings to the SPI1CON1 and SPI1CON2 registers with the MSTEN bit (SPI1CON1<5>) = `0`.
4. Clear the SMP bit.
5. If the CKE bit is set, then the SSEN bit must be set, thus enabling the $\overline{\text{SS1}}$ pin.
6. Clear the SPIROV bit (SPI1STAT<6>).
7. Select Enhanced Buffer mode by setting the SPIBEN bit (SPI1CON2<0>).
8. Enable SPI operation by setting the SPIEN bit (SPI1STAT<15>).

FIGURE 16-2: SPI1 MODULE BLOCK DIAGRAM (ENHANCED BUFFER MODE)



REGISTER 16-1: 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	—	SPISIDL	—	—	SPIBEC2	SPIBEC1	SPIBEC0
bit 15							bit 8

R-0, HSC	R/C-0, HS	R/W-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	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 **SPIEN:** SPIx Enable bit
1 = Enables module and configures SCKx, SDOx, SDIx and $\overline{\text{SSx}}$ as serial port pins
0 = Disables module

bit 14 **Unimplemented:** Read as ‘0’

bit 13 **SPISIDL:** SPIx Stop in Idle Mode bit
1 = Discontinues module operation when device enters Idle mode
0 = Continues 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:** SPIx 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:** SPIx 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 SPI1BUF register)
0 = No overflow has occurred

bit 5 **SRXMPT:** SPIx 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 SPIx transmit buffer is full (SPITBF bit is set)
110 = Interrupt when 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 byte is shifted into the SPIxSR; as a result, the TX FIFO has one open spot
011 = Interrupt when SPIx receive buffer is full (SPIRBF bit is set)
010 = Interrupt when SPIx receive buffer is 3/4 or more full
001 = Interrupt when data is available in 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 is set)

REGISTER 16-1: SPIxSTAT: SPIx STATUS AND CONTROL REGISTER (CONTINUED)

bit 1 **SPITBF:** SPIx Transmit Buffer Full Status bit

1 = Transmit has not yet started, SPIxTXB is full
0 = Transmit has started, SPIxTXB is empty

In Standard Buffer mode:
Automatically set in hardware when the CPU writes 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 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 is complete, SPIxRXB is full
0 = Receive is not complete, SPIxRXB is empty

In Standard Buffer mode:
Automatically set in hardware when the SPIx transfers data from the 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 SPIxSR to buffer, filling the last unread buffer location. Automatically cleared in hardware when a buffer location is available for a transfer from SPIxSR.

REGISTER 16-2: 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	DISSDO	MODE16	SMP	CKE[1]
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	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 (SPIx Master modes only)
1 = Internal SPIx clock is disabled, pin functions as an I/O
0 = Internal SPIx clock is enabled

bit 11 **DISSDO:** Disables SDOx pin bit
1 = SDOx pin is not used by the module; pin functions as an 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 is sampled at the end of data output time
0 = Input data is sampled at the middle of data output time
Slave mode:
SMP must be cleared when SPIx is used in Slave mode.

bit 8 **CKE:** Clock Edge Select bit[1]
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 bit (Slave mode)
1 = $\overline{\text{SSx}}$ pin is used for Slave mode
0 = $\overline{\text{SSx}}$ pin is not used by the module; pin is controlled by port function

bit 6 **CKP:** Clock Polarity Select bit
1 = Idle state for clock is a high level; active state is a low level
0 = Idle state for clock is a low level; active state is a high level

bit 5 **MSTEN:** Master Mode Enable bit
1 = Master mode
0 = Slave mode

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

Note 1: 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).

REGISTER 16-2: SPIxCON1: SPIx CONTROL REGISTER 1 (CONTINUED)

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: 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).

REGISTER 16-3: 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:** SPIx Frame Sync Pulse Direction Control on $\overline{\text{SSx}}$ Pin bit
1 = Frame Sync pulse input (slave)
0 = Frame Sync pulse output (master)

bit 13 **SPIFPOL:** SPIx 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:** SPIx 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:** SPIx Enhanced Buffer Enable bit
1 = Enhanced buffer is enabled
0 = Enhanced buffer is disabled (Legacy mode)

EQUATION 16-1: RELATIONSHIP BETWEEN DEVICE AND SPIx CLOCK SPEED[1]

$$F_{SCK} = \frac{F_{CY}}{\text{Primary Prescaler * Secondary Prescaler}}$$

Note 1: Based on FCY = FOSC/2; Doze mode and PLL are disabled.

TABLE 16-1: 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 are indicated in kHz.

17.0 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 on the Inter-Integrated Circuit, refer to the *"dsPIC33/PIC24 Family Reference Manual"*, **"Inter-Integrated Circuit (I^2C)"** (DS70000195).

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 data EEPROMs, display drivers, A/D Converters, etc.

The I^2C modules support 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 17-1.

17.1 Pin Remapping Options

The I^2C modules are tied to a fixed pin. To allow flexibility with peripheral multiplexing, the I2C1 module in 28-pin devices, can be reassigned to the alternate pins. These alternate pins are designated as SCL1 and SDA1 during device configuration.

Pin assignment is controlled by the I2CxSEL Configuration bit. Programming this bit (= `0`) multiplexes the module to the SCL1 and SDA1 pins.

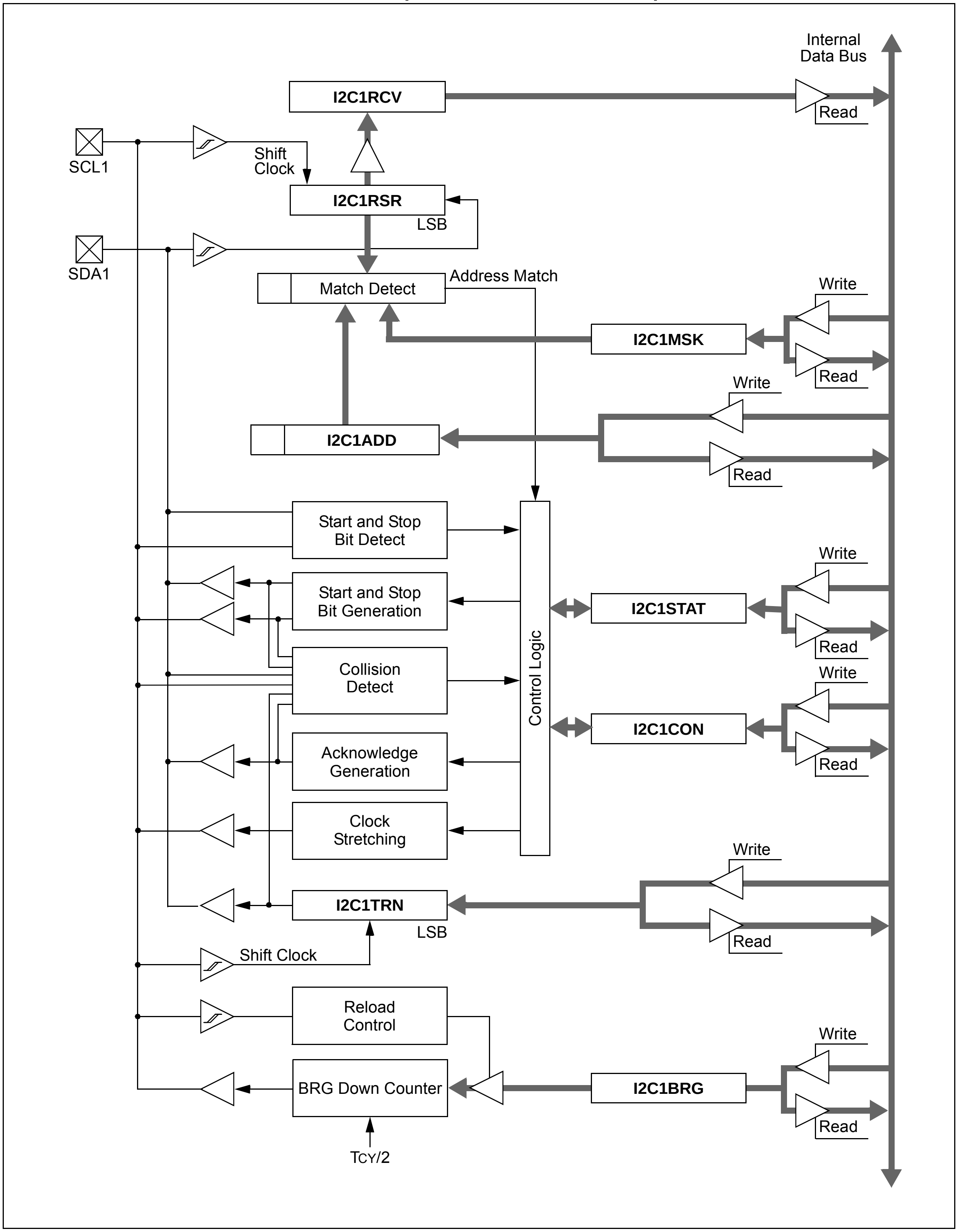
Note:	Throughout this section, references to register and bit names that may be associated with a specific I^2C module are referred to generically by the use of 'x' in place of the specific module number. Thus, "I2CxSTAT" might refer to the Receive Status register for either I2C1 or I2C2.

17.2 Communicating as a Master in a Single Master Environment

The details of sending a message in Master mode depends on the communication protocols 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 17-1: I^{2}C BLOCK DIAGRAM (I2C1 MODULE IS SHOWN)



17.3 Setting Baud Rate When Operating as a Bus Master

To compute the Baud Rate Generator (BRG) reload value, use Equation 17-1.

EQUATION 17-1: COMPUTING BAUD RATE RELOAD VALUE(1)

$$F_{SCL} = \frac{F_{CY}}{I2CxBRG + 1 + \frac{F_{CY}}{10,000,000}}$$

or

$$I2CxBRG = \left(\frac{F_{CY}}{F_{SCL}} - \frac{F_{CY}}{10,000,000}\right) - 1$$

Note 1: Based on FCY = FOSC/2; Doze mode and PLL are disabled.

17.4 Slave Address Masking

The I2CxMSK register (Register 17-3) 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 '`0`' or '`1`'. For example, when I2CxMSK is set to '`00100000`', the slave module will detect both addresses: '`0000000`' and '`00100000`'.

To enable address masking, the Intelligent Peripheral Management Interface (IPMI) must be disabled by clearing the IPMIEN bit (I2C1CON<11>).

Note:	As a result of changes in the I^{2}C protocol, the addresses in Table 17-2 are reserved and will not be Acknowledged in Slave mode. This includes any address mask settings that include any of these addresses.

TABLE 17-1: I^{2}C CLOCK RATES(1)

Required System FSCL	FCY	I2CxBRG Value (Decimal)	I2CxBRG Value (Hexadecimal)	Actual FSCL
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.

TABLE 17-2: I^{2}C 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 010`	`x`	Reserved
`0000 011`	`x`	Reserved
`0000 1xx`	`x`	HS Mode Master Code
`1111 1xx`	`x`	Reserved
`1111 0xx`	`x`	10-Bit Slave Upper Byte(3)

Note 1: The address bits listed here will never cause an address match, independent of the address mask settings.

2: This 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 17-1: 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^{2}C pins are controlled by port functions

bit 14 **Unimplemented:** Read as ‘0’

bit 13 **I2CSIDL:** I2Cx Stop in Idle Mode bit
1 = Discontinues module operation when the device enters an Idle mode
0 = Continues module operation in Idle mode

bit 12 **SCLREL:** SCLx Release Control bit (when operating as I^{2}C slave)
1 = Releases SCLx clock
0 = Holds SCLx clock low (clock stretch)

If STREN = 1:
The 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 the slave transmission. Hardware is clear at the end of slave reception.

If STREN = 0:
The bit is R/S (i.e., software may only write ‘1’ to release clock). Hardware is clear at the beginning of slave transmission.

bit 11 **IPMIEN:** Intelligent Peripheral Management Interface (IPMI) Enable bit
1 = IPMI Support mode is enabled; all addresses are Acknowledged
0 = IPMI Support 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 is disabled
0 = Slew rate control is enabled

bit 8 **SMEN:** SMBus Input Levels bit
1 = Enables I/O pin thresholds compliant with the SMBus specification
0 = Disables the SMBus input thresholds

bit 7 **GCEN:** General Call Enable bit (when operating as I^{2}C slave)
1 = Enables interrupt when a general call address is received in the I2CxRSR (module is enabled for reception)
0 = General call address is disabled

bit 6 **STREN:** SCLx Clock Stretch Enable bit (when operating as I^{2}C slave)
Used in conjunction with the SCLREL bit.
1 = Enables software or receives clock stretching
0 = Disables software or receives clock stretching

REGISTER 17-1: I2CxCON: I2Cx CONTROL REGISTER (CONTINUED)

bit 5 **ACKDT:** Acknowledge Data bit (when operating as I^{2}C 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^{2}C master; applicable during master receive)
1 = Initiates the Acknowledge sequence on the 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^{2}C master)
1 = Enables Receive mode for I^{2}C; 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^{2}C master)
1 = Initiates Stop condition on SDAx and SCLx pins; hardware is clear at end of master Stop sequence
0 = Stop condition is not in progress

bit 1 **RSEN:** Repeated Start Condition Enable bit (when operating as I^{2}C master)
1 = Initiates Repeated Start condition on 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 Enable bit (when operating as I^{2}C master)
1 = Initiates Start condition on SDAx and SCLx pins; hardware is clear at the end of the master Start sequence
0 = Start condition is not in progress

REGISTER 17-2: 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/$\overline{\text{A}}$	P	S	R/$\overline{\text{W}}$	RBF	TBF
bit 7							bit 0

Legend:	C = Clearable bit	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 **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^{2}C 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 the 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 an 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 a match of the 2nd byte of the matched 10-bit address; hardware is clear at Stop detection.

bit 7 **IWCOL:** I2Cx Write Collision Detect bit
1 = An attempt to write to the I2CxTRN register failed because the I^{2}C module is busy
0 = No collision
Hardware is set at an occurrence of a write to I2CxTRN while busy (cleared by software).

bit 6 **I2COV:** I2Cx 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/$\overline{\text{A}}$:** Data/$\overline{\text{Address}}$ bit (when operating as I^{2}C slave)
1 = Indicates that the last byte received was data
0 = Indicates that the last byte received was the device address
Hardware is clear at a device address match; hardware is set by a write to I2CxTRN or by reception of a slave byte.

REGISTER 17-2: 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 cleared when a 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 a Start, Repeated Start or Stop is detected.
bit 2	**R/$\overline{\text{W}}$:** Read/$\overline{\text{Write}}$ Information bit (when operating as I^{2}C slave) `1` = Read – indicates data transfer is output from the slave `0` = Write – indicates data transfer is input to the slave Hardware is set or clear after the reception of an I^{2}C device address byte.
bit 1	**RBF:** Receive Buffer Full Status bit `1` = Receive is complete, I2CxRCV is full `0` = Receive is not complete, I2CxRCV is empty Hardware is set when I2CxRCV is written with a 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 the software writes to I2CxTRN; hardware is clear at the completion of data transmission.

REGISTER 17-3: 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
—	—	—	—	—	—	AMSK<9:8>	
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
AMSK<7:0>							
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 9-0 **AMSK<9:0>:** Mask for Address Bit x Select bits
1 = Enables masking for bit x of an incoming message address; bit match is not required in this position
0 = Disables masking for bit x; bit match is required in this position

REGISTER 17-4: 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	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	—	SMBUSDEL2	SMBUSDEL1	—	—	—	—
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 **SMBUSDEL2:** SMBus SDA2 Input Delay Select bit
1 = The I2C2 module is configured for a longer SMBus input delay (nominal 300 ns delay)
0 = The I2C2 module is configured for a legacy input delay (nominal 150 ns delay)

bit 4 **SMBUSDEL1:** SMBus SDA1 Input Delay Select bit
1 = The I2C1 module is configured for a longer SMBus input delay (nominal 300 ns delay)
0 = The I2C1 module is configured for a legacy input delay (nominal 150 ns delay)

bit 3-0 **Unimplemented:** Read as ‘0’

18.0 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 on the Universal Asynchronous Receiver Transmitter, refer to the *“dsPIC33/PIC24 Family Reference Manual”*, **“Universal Asynchronous Receiver Transmitter (UART)”** (DS70000582).

The Universal Asynchronous Receiver Transmitter (UART) module is one of the serial I/O modules available in this 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. This module also supports a hardware flow control option with the $\overline{\text{UxCTS}}$ and $\overline{\text{UxRTS}}$ pins, and also includes an IrDA® encoder and decoder.

The primary features of the UART module are:

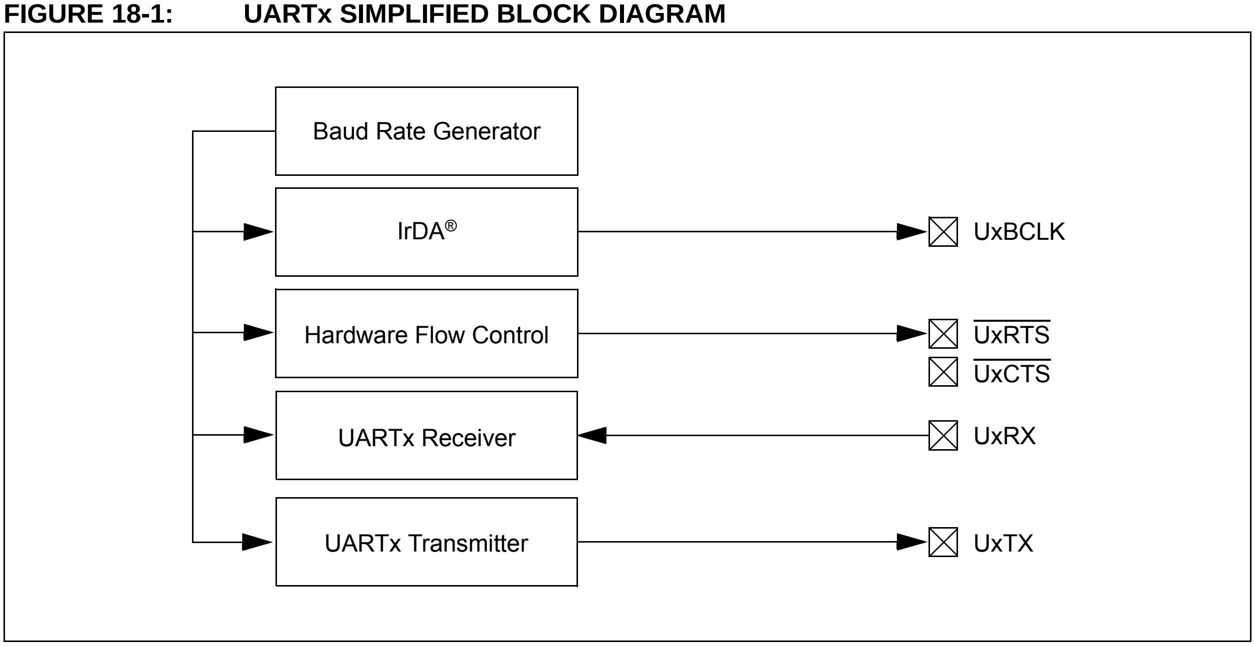
- Full-Duplex, 8-Bit or 9-Bit Data Transmission through the UxTX and UxRX Pins
- Even, Odd or No Parity Options (for 8-bit data)
- One or Two Stop bits
- Hardware Flow Control Option with $\overline{\text{UxCTS}}$ and $\overline{\text{UxRTS}}$ pins
- Fully Integrated Baud Rate Generator (IBRG) with 16-Bit Prescaler
- Baud Rates Ranging from 1 Mbps to 15 bps 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 UARTx is shown in Figure 18-1. The UARTx module consists of these important hardware elements:

- Baud Rate Generator
- Asynchronous Transmitter
- Asynchronous Receiver

> **Note:** Throughout this section, references to register and bit names that may be associated with a specific USART module are referred to generically by the use of ‘x’ in place of the specific module number. Thus, “UxSTA” might refer to the USART Status register for either USART1 or USART2.

FIGURE 18-1: UARTx SIMPLIFIED BLOCK DIAGRAM



18.1 UARTx Baud Rate Generator (BRG)

The UARTx module includes a dedicated 16-bit Baud Rate Generator (BRG). The UxBRG register controls the period of a free-running, 16-bit timer. Equation 18-1 provides the formula for computation of the baud rate with BRGH = 0.

EQUATION 18-1: UARTx BAUD RATE WITH BRGH = 0[1]

$$\text{Baud Rate} = \frac{F_{CY}}{16 \bullet (\text{UxBRG} + 1)}$$

$$\text{UxBRG} = \frac{F_{CY}}{16 \bullet \text{Baud Rate}} - 1$$

Note 1: Based on $F_{CY} = F_{OSC}/2$; Doze mode and PLL are disabled.

Example 18-1 provides the calculation of the baud rate error for the following conditions:

- F_{CY} = 4 MHz
- Desired Baud Rate = 9600

The maximum baud rate (BRGH = 0) possible is $F_{CY}/16$ (for UxBRG = 0) and the minimum baud rate possible is $F_{CY}/(16 * 65536)$.

Equation 18-2 shows the formula for computation of the baud rate with BRGH = 1.

EQUATION 18-2: UARTx BAUD RATE WITH BRGH = 1[1]

$$\text{Baud Rate} = \frac{F_{CY}}{4 \bullet (\text{UxBRG} + 1)}$$

$$\text{UxBRG} = \frac{F_{CY}}{4 \bullet \text{Baud Rate}} - 1$$

Note 1: Based on $F_{CY} = F_{OSC}/2$; Doze mode and PLL are disabled.

The maximum baud rate (BRGH = 1) possible is $F_{CY}/4$ (for UxBRG = 0) and the minimum baud rate possible is $F_{CY}/(4 * 65536)$.

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 18-1: BAUD RATE ERROR CALCULATION (BRGH = 0)[1]

Desired Baud Rate = F_{CY}/(16 (UxBRG + 1))

Solving for UxBRG value:

UxBRG = ((F_{CY}/Desired Baud Rate)/16) – 1
UxBRG = ((4000000/9600)/16) – 1
UxBRG = 25

Calculated Baud Rate = 4000000/(16 (25 + 1))
= 9615

Error = (Calculated Baud Rate – Desired Baud Rate)
Desired Baud Rate
= (9615 – 9600)/9600
= 0.16%

Note 1: Based on $F_{CY} = F_{OSC}/2$; Doze mode and PLL are disabled.

18.2 Transmitting in 8-Bit Data Mode

1. Set up the UARTx:
   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 UARTx.
3. Set the UTXEN bit (causes a transmit interrupt, two cycles after being set).
4. Write the data byte to the lower byte of the 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.

18.3 Transmitting in 9-Bit Data Mode

1. Set up the UARTx (as described in **Section 18.2 "Transmitting in 8-Bit Data Mode"**).
2. Enable the UARTx.
3. Set the UTXEN bit (causes a transmit interrupt, two cycles after being set).
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.

18.4 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 UARTx for the desired mode.
2. Set UTXEN and UTXBRK – this sets up the Break character.
3. Load the UxTXREG with a dummy character to initiate transmission (value is ignored).
4. Write '55h' to UxTXREG – 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.

18.5 Receiving in 8-Bit or 9-Bit Data Mode

1. Set up the UARTx (as described in **Section 18.2 "Transmitting in 8-Bit Data Mode"**).
2. Enable the UARTx.
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.

18.6 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 UARTx module. These two pins allow the UARTx to operate in Simplex and Flow Control modes. 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.

18.7 Infrared Support

The UARTx module provides two types of infrared UARTx 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.

As the IrDA modes require a 16x baud clock, they will only work when the BRGH bit (UxMODE<3>) is '`0`'.

18.7.1 EXTERNAL IrDA SUPPORT – IrDA CLOCK OUTPUT

To support external IrDA encoder and decoder devices, the UxBCLK pin (same as the $\overline{\text{UxRTS}}$ pin) can be configured to generate the 16x baud clock. When UEN<1:0> = `11`, the UxBCLK pin will output the 16x baud clock if the UARTx module is enabled; it can be used to support the IrDA codec chip.

18.7.2 BUILT-IN IrDA ENCODER AND DECODER

The UARTx has full implementation of the IrDA encoder and decoder as part of the UARTx 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 18-1: UxMODE: UARTx MODE REGISTER

R/W-0	U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0[2]	R/W-0[2]
UARTEN	—	USIDL	IREN[1]	RTSMD	—	UEN1	UEN0
bit 15							bit 8

R/C-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:	C = Clearable 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 **UARTEN:** UARTx Enable bit
- 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:** UARTx Stop in Idle Mode bit
- 1 = Discontinues module operation when the device enters Idle mode
- 0 = Continues module operation in Idle mode

bit 12 **IREN:** IrDA® Encoder and Decoder Enable bit[1]
- 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[2]
- 11 = UxTX, UxRX and UxBCLK 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 $\overline{\text{UxRTS}}$/UxBCLK 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 = Enables Loopback mode
- 0 = Loopback mode is disabled

bit 5 **ABAUD:** Auto-Baud Enable bit
- 1 = Enables 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

bit 4 **RXINV:** Receive Polarity Inversion bit
- 1 = UxRX Idle state is '0'
- 0 = UxRX Idle state is '1'

Note 1: This feature is is only available for the 16x BRG mode (BRGH = 0).
2: The bit availability depends on the pin availability.

REGISTER 18-1: UxMODE: UARTx MODE REGISTER (CONTINUED)

bit 3	**BRGH:** High Baud Rate Enable bit 1 = BRG generates 4 clocks per bit period (4x baud clock, High-Speed mode) 0 = BRG generates 16 clocks per bit period (16x baud clock, Standard mode)
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: This feature is is only available for the 16x BRG mode (BRGH = 0).

2: The bit availability depends on the pin availability.

REGISTER 18-2: 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	UTXISEL0	—	UTXBRK	UTXEN	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:			
HS = Hardware Settable 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	HC = Hardware Clearable bit

bit 15,13 **UTXISEL<1:0>:** UARTx 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
If IREN = 0:
1 = UxTX Idle '0'
0 = UxTX Idle '1'
If IREN = 1:
1 = UxTX Idle '1'
0 = UxTX Idle '0'

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

bit 11 **UTXBRK:** UARTx Transmit Break bit
1 = Sends 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:** UARTx Transmit Enable bit
1 = Transmit is enabled; UxTX pin is controlled by UARTx
0 = Transmit is disabled; any pending transmission is aborted and the buffer is reset. UxTX pin is controlled by the PORT register.

bit 9 **UTXBF:** UARTx 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 the transmit buffer is empty (the last transmission has completed)
0 = Transmit Shift Register is not empty; a transmission is in progress or queued

bit 7-6 **URXISEL<1:0>:** UARTx Receive Interrupt Mode Selection bits
11 = Interrupt is set on a RSR transfer, making the receive buffer full (i.e., has 4 data characters)
10 = Interrupt is set on a 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.

REGISTER 18-2: UxSTA: UARTx STATUS AND CONTROL REGISTER (CONTINUED)

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:** UARTx Receive Buffer Data Available bit (read-only) 1 = Receive buffer has data; at least one more characters can be read 0 = Receive buffer is empty

REGISTER 18-3: UxTXREG: UARTx TRANSMIT REGISTER

U-x	U-x	U-x	U-x	U-x	U-x	U-x	W-x
—	—	—	—	—	—	—	UTX8
bit 15							bit 8

W-x	W-x	W-x	W-x	W-x	W-x	W-x	W-x
UTX<7:0>							
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 15-9	**Unimplemented:** Read as ‘0’
bit 8	**UTX8:** UARTx Data of the Transmitted Character bit (in 9-bit mode)
bit 7-0	**UTX<7:0>:** UARTx Data of the Transmitted Character bits

REGISTER 18-4: UxRXREG: UARTx RECEIVE REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R-0, HSC
—	—	—	—	—	—	—	URX8
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
URX<7:0>							
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	**URX8:** UARTx Data of the Received Character bit (in 9-bit mode)
bit 7-0	**URX<7:0>:** UARTx Data of the Received Character bits

19.0 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 on the Real-Time Clock and Calendar, refer to the *"dsPIC33/PIC24 Family Reference Manual"*, **"RTCC with External Power Control"** (DS39745).

The RTCC provides the user with a Real-Time Clock and Calendar (RTCC) function that can be calibrated.

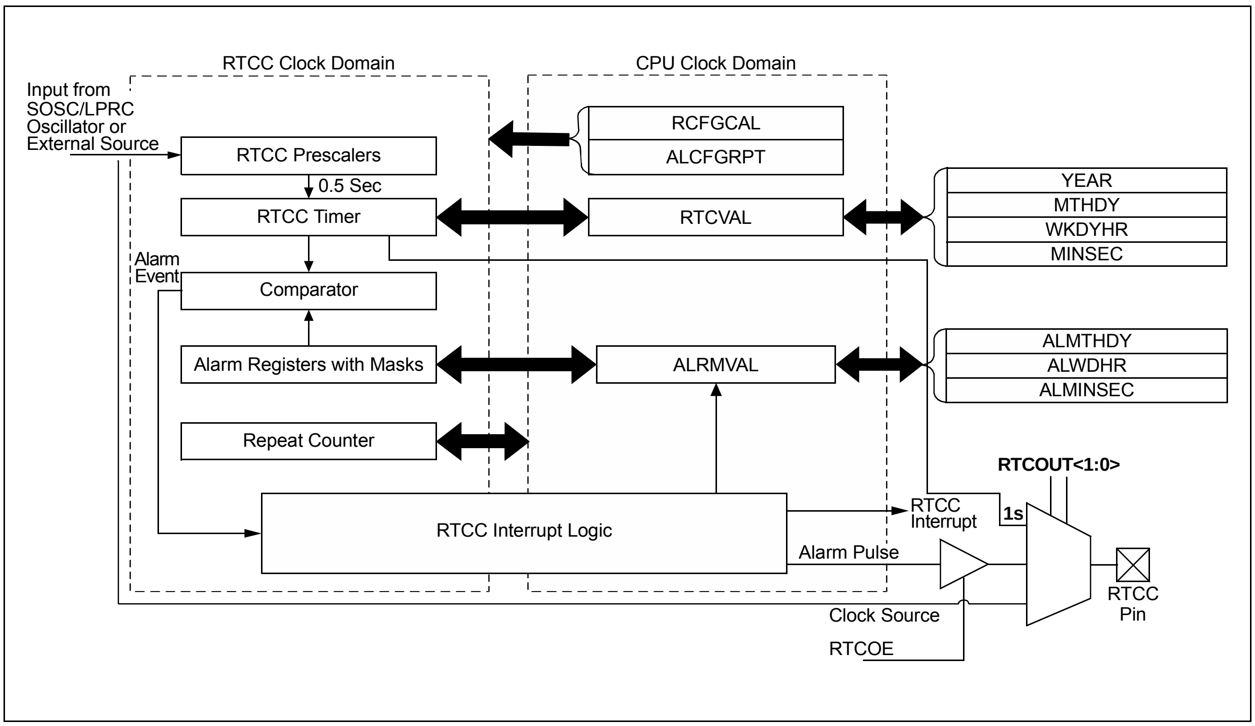
Key features of the RTCC module are:

- Operates in Deep Sleep mode
- Selectable clock source
- Provides hours, minutes and seconds using 24-hour format
- Visibility of one half 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
- Optimized for long-term battery operation
- Fractional second synchronization
- Calibration to within ±2.64 seconds error per month
- Calibrates up to 260 ppm of crystal error
- Ability to periodically wake-up external devices without CPU intervention (external power control)
- Power control output for external circuit control
- Calibration takes effect every 15 seconds
- Runs from any one of the following:
  - External Real-Time Clock of 32.768 kHz
  - Internal 31.25 kHz LPRC Clock
  - 50 Hz or 60 Hz External Input

19.1 RTCC Source Clock

The user can select between the SOSC crystal oscillator, LPRC internal oscillator or an external 50 Hz/60 Hz power line input as the clock reference for the RTCC module. This gives the user an option to trade off system cost, accuracy and power consumption, based on the overall system needs.

FIGURE 19-1: RTCC BLOCK DIAGRAM



19.2 RTCC Module Registers

The RTCC module registers are organized into three categories:

- RTCC Control Registers
- RTCC Value Registers
- Alarm Value Registers

19.2.1 REGISTER MAPPING

To limit the register interface, the RTCC Timer and Alarm Time registers are accessed through 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 19-1).

By writing the RTCVALH byte, the RTCC Pointer value, the 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 19-1: 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 ALRMPTRx bits (ALCFGRPT<9:8>) to select the desired Alarm register pair (see Table 19-2).

By writing the ALRMVALH byte, the Alarm Pointer value (ALRMPTR<1:0> bits) decrements 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 19-2: 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	PWCSTAB	PWCSAMP

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, the ALRMPTR<1:0> value will be decremented. 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.

19.2.2 WRITE LOCK

In order to perform a write to any of the RTCC Timer registers, the RTCWREN bit (RTCPWC<13>) must be set (see Example 19-1).

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 one instruction cycle time window allowed between the 55h/AA sequence and the setting of RTCWREN. Therefore, it is recommended that code follow the procedure in Example 19-1.

19.2.3 SELECTING RTCC CLOCK SOURCE

There are four reference source clock options that can be selected for the RTCC using the RTCCSEL<1:0> bits: 00 = Secondary Oscillator, 01 = LPRC, 10 = 50 Hz External Clock and 11 = 60 Hz External Clock.

EXAMPLE 19-1: SETTING THE RTCWREN BIT

```
asm  volatile ("push w7");
asm  volatile ("push w8");
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
asm  volatile ("pop  w8");
asm  volatile ("pop  w7");
```

19.2.4 RTCC CONTROL REGISTERS

REGISTER 19-1: 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	R/W-0
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>:
00 = MINUTES
01 = WEEKDAY
10 = MONTH
11 = Reserved
RTCVAL<7:0>:
00 = SECONDS
01 = HOURS
10 = DAY
11 = YEAR

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 19-1: RCFGCAL: RTCC CALIBRATION AND CONFIGURATION REGISTER[1] (CONTINUED)

bit 7-0 **CAL<7:0>:** RTC Drift Calibration bits

`01111111` = Maximum positive adjustment; adds 508 RTC clock pulses every one minute
•
•
•
`00000001` = Minimum positive adjustment; adds 4 RTC clock pulses every one minute
`00000000` = No adjustment
`11111111` = Minimum negative adjustment; subtracts 4 RTC clock pulses every one minute
•
•
•
`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 19-2: RTCPWC: RTCC CONFIGURATION REGISTER 2[1]

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PWCEN	PWCPOL	PWCCPRE	PWCSPRE	RTCCLK1[2]	RTCCLK0[2]	RTCOUT1	RTCOUT0
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 **PWCEN:** Power Control Enable bit
1 = Power control is enabled
0 = Power control is disabled

bit 14 **PWCPOL:** Power Control Polarity bit
1 = Power control output is active-high
0 = Power control output is active-low

bit 13 **PWCCPRE:** Power Control Control/Stability Prescaler bits
1 = PWC stability window clock is divide-by-2 of source RTCC clock
0 = PWC stability window clock is divide-by-1 of source RTCC clock

bit 12 **PWCSPRE:** Power Control Sample Prescaler bits
1 = PWC sample window clock is divide-by-2 of source RTCC clock
0 = PWC sample window clock is divide-by-1 of source RTCC clock

bit 11-10 **RTCCLK<1:0>:** RTCC Clock Select bits[2]
Determines the source of the internal RTCC clock, which is used for all RTCC timer operations.
00 = External Secondary Oscillator (SOSC)
01 = Internal LPRC Oscillator
10 = External power line source – 50 Hz
11 = External power line source – 60 Hz

bit 9-8 **RTCOUT<1:0>:** RTCC Output Select bits
Determines the source of the RTCC pin output.
00 = RTCC alarm pulse
01 = RTCC seconds clock
10 = RTCC clock
11 = Power control

bit 7-0 **Unimplemented:** Read as ‘0’

Note 1: The RTCPWC register is only affected by a POR.
2: When a new value is written to these register bits, the Seconds Value register should also be written to properly reset the clock prescalers in the RTCC.

REGISTER 19-3: 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	R/W-0
ALRMEN	CHIME	AMASK3	AMASK2	AMASK1	AMASK0	ALRMPTR1	ALRMPTR0
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
ARPT7	ARPT6	ARPT5	ARPT4	ARPT3	ARPT2	ARPT1	ARPT0
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 **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
0000 = Every half second
0001 = Every second
0010 = Every 10 seconds
0011 = Every minute
0100 = Every 10 minutes
0101 = Every hour
0110 = Once a day
0111 = Once a week
1000 = Once a month
1001 = Once a year (except when configured for February 29th, once every 4 years)
101x = Reserved – do not use
11xx = Reserved – do not use

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>:
00 = ALRMMIN
01 = ALRMWD
10 = ALRMMNTH
11 = Unimplemented

ALRMVAL<7:0>:
00 = ALRMSEC
01 = ALRMHR
10 = ALRMDAY
11 = Unimplemented

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; it is prevented from rolling over from 00h to FFh unless CHIME = 1.

19.2.5 RTCVAL REGISTER MAPPINGS

REGISTER 19-4: 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	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
YRTEN3	YRTEN2	YRTEN2	YRTEN1	YRONE3	YRONE2	YRONE1	YRONE0
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 **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 19-5: MTHDY: 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 19-6: WKDYHR: 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 19-7: MINSEC: 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.

19.2.6 ALRMVAL REGISTER MAPPINGS

REGISTER 19-8: 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 19-9: 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 19-10: 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.

REGISTER 19-11: RTCCSWT: RTCC CONTROL/SAMPLE WINDOW TIMER REGISTER[1]

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
PWCSTAB7	PWCSTAB6	PWCSTAB5	PWCSTAB4	PWCSTAB3	PWCSTAB2	PWCSTAB1	PWCSTAB0
bit 15							bit 8

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
PWCSAMP7	PWCSAMP6	PWCSAMP5	PWCSAMP4	PWCSAMP3	PWCSAMP2	PWCSAMP1	PWCSAMP0
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 **PWCSTAB<7:0>:** PWM Stability Window Timer bits

11111111 = Stability window is 255 T_{PWCCLK} clock periods

•

•

•

00000000 = Stability window is 0 T_{PWCCLK} clock periods

The sample window starts when the alarm event triggers. The stability window timer starts counting from every alarm event when PWCEN = 1.

bit 7-0 **PWCSAMP<7:0>:** PWM Sample Window Timer bits

11111111 = Sample window is always enabled, even when PWCEN = 0

11111110 = Sample window is 254 T_{PWCCLK} clock periods

•

•

•

00000000 = Sample window is 0 T_{PWCCLK} clock periods

The sample window timer starts counting at the end of the stability window when PWCEN = 1. If PWCSTAB<7:0> = 00000000, the sample window timer starts counting from every alarm event when PWCEN = 1.

Note 1: A write to this register is only allowed when RTCWREN = 1.

19.3 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 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 be either added or subtracted from the RTCC timer, once every minute. Refer to the steps below 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.
3. a) If the oscillator is faster than ideal (negative result from Step 2), the RCFGCAL register value must 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 than ideal (positive result from Step 2), the RCFGCAL register value must be positive. This causes the specified number of clock pulses to be subtracted from the timer counter, once every minute.

EQUATION 19-1:

(Ideal Frequency† – Measured Frequency) * 60 = Clocks per Minute

† Ideal Frequency = 32,768 Hz

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, except when SECONDS = 00, 15, 30 or 45. This is due to the auto-adjust of the RTCC at 15 second intervals.

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.

19.4 Alarm

- Configurable from half second to one year
- Enabled using the ALRMEN bit (ALCFGRPT<15>)
- One-time alarm and repeat alarm options are available

19.4.1 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 19-2, the interval selection of the alarm is configured through the AMASKx 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<7:0> bits (ALCFGRPT<7:0>). When the value of the ARPTx 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 ARPTx 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 ARPTx bits reaches 00h, it rolls over to FFh and continues counting indefinitely while CHIME is set.

19.4.2 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 than 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 19-2: 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.

19.5 POWER CONTROL

The RTCC includes a power control feature that allows the device to periodically wake-up an external device, wait for the device to be stable before sampling wake-up events from that device and then shut down the external device. This can be done completely autonomously by the RTCC, without the need to wake from the current low-power mode (Sleep, Deep Sleep, etc.).

To enable this feature, the RTCC must be enabled (RTCEN = `1`), the PWCEN register bit must be set and the RTCC pin must be driving the PWC control signal (RTCOE = `1` and RTCOUT<1:0> = `11`).

The polarity of the PWC control signal may be chosen using the PWCPOL register bit. Active-low or active-high may be used with the appropriate external switch to turn on or off the power to one or more external devices. The active-low setting may also be used in conjunction with an open-drain setting on the RTCC pin. This setting is able to drive the GND pin(s) of the external device directly (with the appropriate external VDD pull-up device), without the need for external switches. Finally, the CHIME bit should be set to enable the PWC periodicity.

NOTES:

20.0 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 *"dsPIC33/PIC24 Family Reference Manual"*, **"32-Bit Programmable Cyclic Redundancy Check (CRC)"** (DS30009729).

The 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

A simplified block diagram of the CRC generator is shown in Figure 20-1. A simple version of the CRC shift engine is shown in Figure 20-2.

FIGURE 20-1: CRC BLOCK DIAGRAM

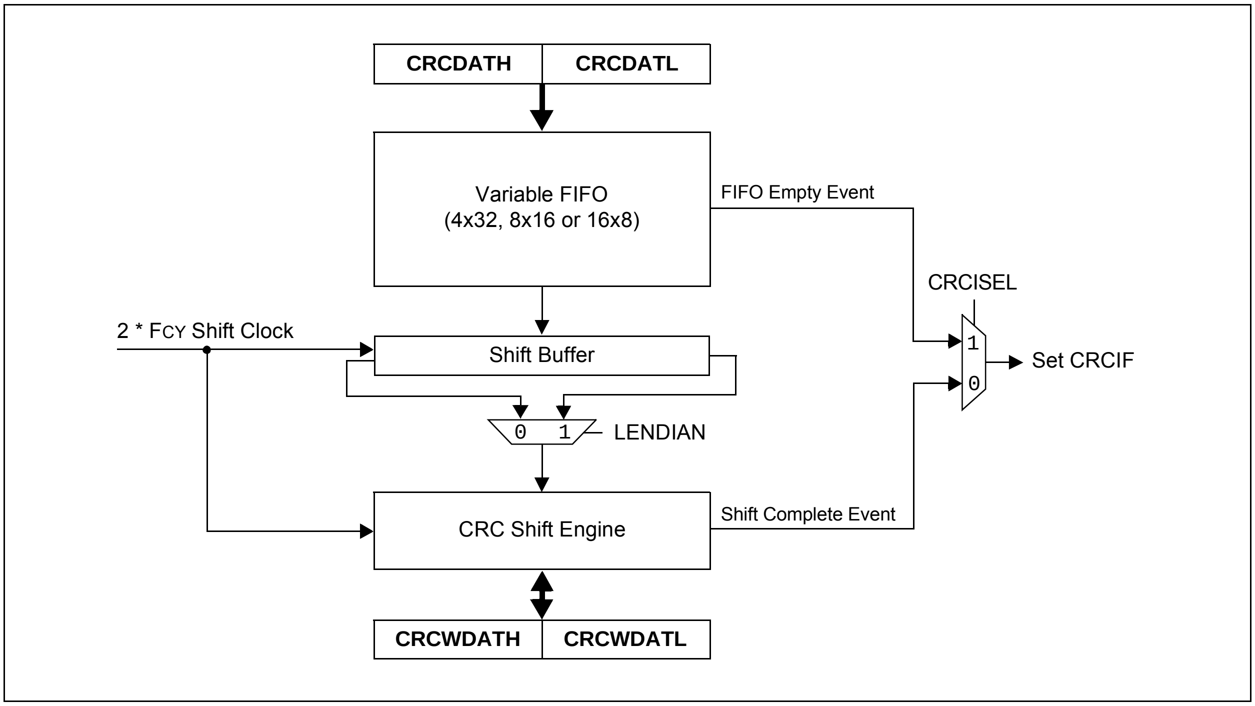
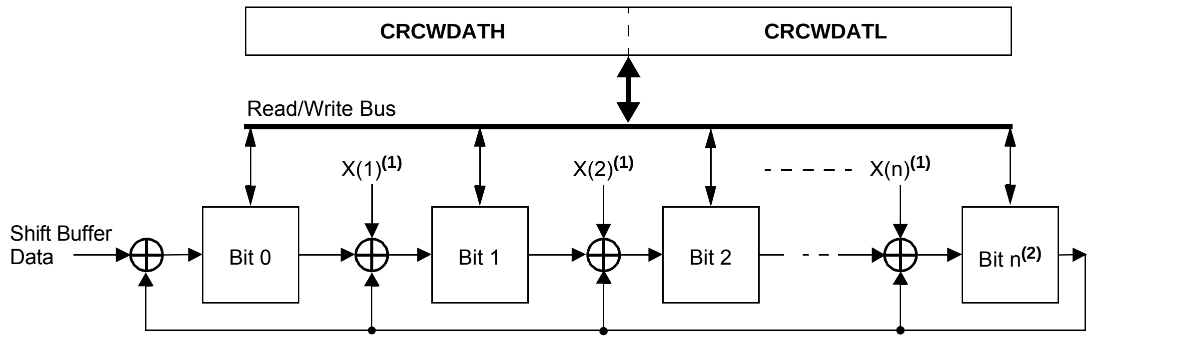


FIGURE 20-2: CRC SHIFT ENGINE DETAIL



Note 1: Each XOR stage of the shift engine is programmable; see text for details.

2: Polynomial Length n is determined by ([PLEN<4:0>] + 1).

20.1 User Interface

20.1.1 POLYNOMIAL INTERFACE

The CRC module can be programmed for CRC polynomials of up to 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 this bit disables the XOR.

For example, consider two CRC polynomials, one is a 16-bit equation and the other is a 32-bit equation:

$$x16 + x12 + x5 + 1$$

and

$$x32 + x26 + x23 + x22 + x16 + x12 + x11 + x10 + x8 + x7 + x5 + x4 + x2 + x + 1$$

To program these polynomials into the CRC generator, set the register bits, as shown in Table 20-1.

Note that the appropriate positions are set to '1' to indicate that 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, N, it is assumed that the *N*th bit will always be used, regardless of the bit setting. Therefore, for a polynomial length of 32, there is no 32nd bit in the CRCXOR register.

20.1.2 DATA INTERFACE

The module incorporates a FIFO that works with a variable data width. Input data width can be configured to any value, between 1 and 32 bits, using the DWIDTH<4:0> bits (CRCCON2<12:8>). When the data width is greater than 15, the FIFO is 4 words deep. When the DWIDTHx value is between 15 and 8, the FIFO is 8 words deep. When the DWIDTHx value is 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 the DWIDTHx value is 5, then the size of the data is DWIDTHx + 1 or 6. 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 the DWIDTHx value is 24, the VWORDx bits will increment when bit 7 of CRCDATH is written. Therefore, CRCDATL must always be written before CRCDATH.

The CRC engine starts shifting data when the CRCGO bit is set and the value of the VWORDx bits is greater than zero. Each word is copied out of the FIFO into a buffer register, which decrements VWORDx. The data is then shifted out of the buffer. The CRC engine continues shifting at a rate of two bits per instruction cycle until the VWORDx value 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 the VWORDx value reaches the maximum value for the configured value of DWIDTHx (4, 8 or 16), the CRCFUL bit becomes set. When the VWORDx value 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 CRCDAT, before a read of the VWORDx bits is done.

TABLE 20-1: CRC SETUP EXAMPLES FOR 16 AND 32-BIT POLYNOMIAL

CRC Control Bits	Bit Values	
	16-Bit Polynomial	32-Bit Polynomial
PLEN<4:0>	01111	11111
X<31:16>	0000 0000 0000 000x	0000 0100 1100 0001
X<15:0>	0001 0000 0010 000x	0001 1101 1011 011x

20.1.3 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 of the data that is shifted into the engine. The result of the CRC calculation will still be a normal CRC result, not a reverse CRC result.

20.1.4 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.

20.1.5 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 the 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 DWIDTHx 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. CRCWDAT can also be left unchanged to resume a previously halted calculation.
5. Set the CRCGO bit to start calculation.
6. Write the 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.

20.2 Registers

There are eight registers associated with the module:

- CRCCON1
- CRCCON2
- CRCXORL
- CRCXORH
- CRCDATL
- CRCDATH
- CRCWDATL
- CRCWDATH

The CRCCON1 and CRCCON2 registers (Register 20-1 and Register 20-2) control the operation of the module, and configure the various settings. The CRCXOR registers (Register 20-3 and Register 20-4) 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 20-1: CRCCON1: CRC CONTROL REGISTER 1

R/W-0	U-0	R/W-0	R-0	R-0	R-0	R-0	R-0
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:	HC = Hardware 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 **CRCEN**: CRC Enable bit
1 = Module is enabled
0 = Module is enabled
All state machines, pointers and CRCWDAT/CRCDAT registers are reset; other SFRs are NOT reset.

bit 14 **Unimplemented:** Read as ‘0’

bit 13 **CSIDL:** CRC Stop in Idle Mode bit
1 = Discontinues module operation when device enters Idle mode
0 = Continues module operation in Idle mode

bit 12-8 **VWORD<4:0>:** Pointer Value bits
Indicates the number of valid words in the FIFO, which has a maximum value of 8 when PLEN<4:0> > 7 or 16 when PLEN<4:0> ≤ 7.

bit 7 **CRCFUL**: CRC FIFO Full bit
1 = FIFO is full
0 = FIFO is not full

bit 6 **CRCMPT:** CRC 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; CRC calculation is not complete
0 = Interrupt on shift is complete and CRCWDAT result is ready

bit 4 **CRCGO:** Start CRC bit
1 = Starts 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 20-2: CRCCON2: CRC CONTROL REGISTER 2

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 Width Select bits
Defines the width of the data word (Data Word Width = (DWIDTH<4:0>) + 1).

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **PLEN<4:0>:** Polynomial Length Select bits
Defines the length of the CRC polynomial (Polynomial Length = (PLEN<4:0>) + 1).

REGISTER 20-3: 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
X<15:8>							
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
X<7:1>							—
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 15-1 **X<15:1>:** XOR of Polynomial Term X^n Enable bits

bit 0 **Unimplemented:** Read as ‘0’

REGISTER 20-4: CRCXORH: CRC XOR POLYNOMIAL REGISTER, HIGH 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
X<31:24>							
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
X<23:16>							
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

21.0 HIGH/LOW-VOLTAGE DETECT (HLVD)

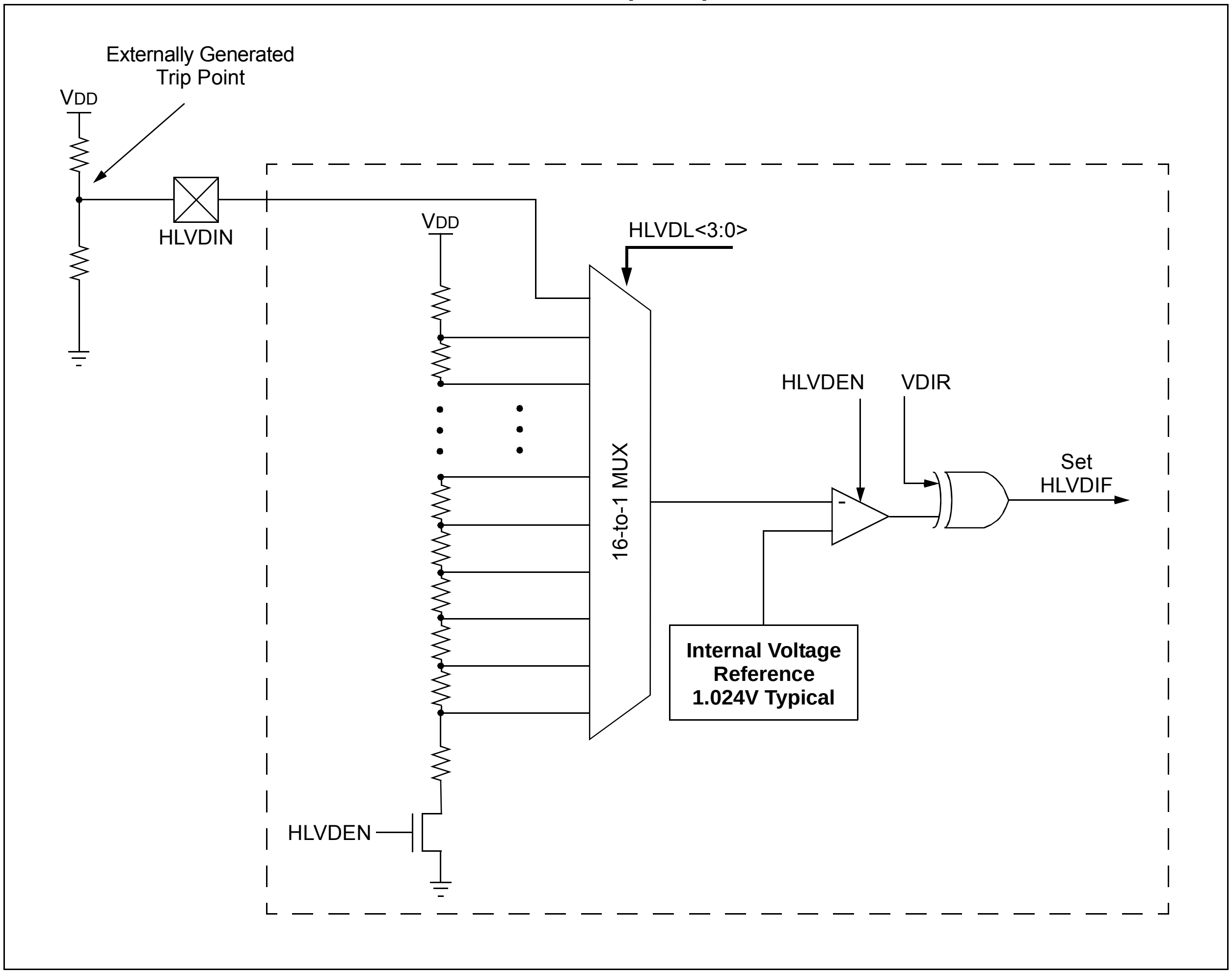
Note:	This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information on the High/Low-Voltage Detect, refer to the *"dsPIC33/PIC24 Family Reference Manual"*, **Section 36. "High-Level Integration with Programmable High/Low-Voltage Detect (HLVD)"** (DS39725).

The High/Low-Voltage Detect module (HLVD) is a programmable circuit that allows the user to specify both the device voltage trip point and the direction of change.

An interrupt flag is set if the device experiences an excursion past the trip point in the direction of change. If the interrupt is enabled, the program execution will branch to the interrupt vector address and the software can then respond to the interrupt.

The HLVD Control register (see Register 21-1) completely controls the operation of the HLVD module. This allows the circuitry to be "turned off" by the user under software control, which minimizes the current consumption for the device.

FIGURE 21-1: HIGH/LOW-VOLTAGE DETECT (HLVD) MODULE BLOCK DIAGRAM



REGISTER 21-1: HLVDCON: HIGH/LOW-VOLTAGE DETECT CONTROL REGISTER

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
HLVDEN	—	HLSIDL	—	—	—	—	—
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
VDIR	BGVST	IRVST	—	HLVDL3	HLVDL2	HLVDL1	HLVDL0
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 **HLVDEN:** High/Low-Voltage Detect Power Enable bit
1 = HLVD is enabled
0 = HLVD is disabled

bit 14 **Unimplemented:** Read as ‘0’

bit 13 **HLSIDL:** HLVD Stop in Idle Mode bit
1 = Discontinues module operation when device enters Idle mode
0 = Continues module operation in Idle mode

bit 12-8 **Unimplemented:** Read as ‘0’

bit 7 **VDIR:** Voltage Change Direction Select bit
1 = Event occurs when voltage equals or exceeds trip point (HLVDL<3:0>)
0 = Event occurs when voltage equals or falls below trip point (HLVDL<3:0>)

bit 6 **BGVST:** Band Gap Voltage Stable Flag bit
1 = Indicates that the band gap voltage is stable
0 = Indicates that the band gap voltage is unstable

bit 5 **IRVST:** Internal Reference Voltage Stable Flag bit
1 = Indicates that the internal reference voltage is stable and the high-voltage detect logic generates the interrupt flag at the specified voltage range
0 = Indicates that the internal reference voltage is unstable and the high-voltage detect logic will not generate the interrupt flag at the specified voltage range, and the HLVD interrupt should not be enabled

bit 4 **Unimplemented:** Read as ‘0’

bit 3-0 **HLVDL<3:0>:** High/Low-Voltage Detection Limit bits
1111 = External analog input is used (input comes from the HLVDIN pin)
1110 = Trip Point 1[(1)]
1101 = Trip Point 2[(1)]
1100 = Trip Point 3[(1)]
•
•
•
0000 = Trip Point 15[(1)]

Note 1: For the actual trip point, see **Section 29.0 “Electrical Characteristics”**.

22.0 12-BIT A/D CONVERTER WITH THRESHOLD DETECT

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 on the 12-Bit A/D Converter with Threshold Detect, refer to the *"dsPIC33/PIC24 Family Reference Manual"*, **"12-Bit A/D Converter with Threshold Detect"** (DS39739).

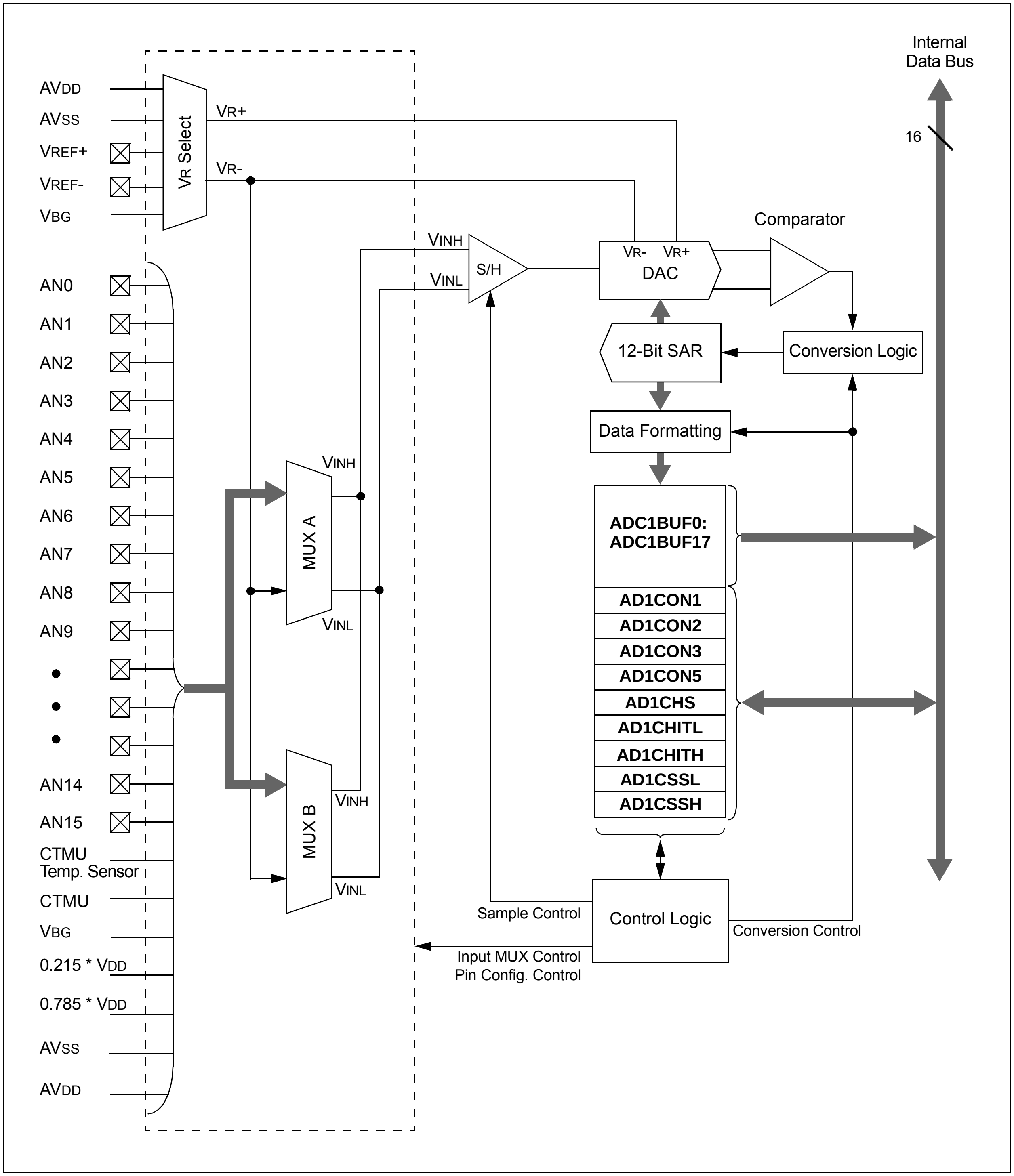
The PIC24F 12-bit A/D Converter has the following key features:

- Successive Approximation Register (SAR) Conversion
- Conversion Speeds of up to 100 ksps
- Up to 32 Analog Input Channels (Internal and External)
- Multiple Internal Reference Input Channels
- External Voltage Reference Input Pins
- Unipolar Differential Sample-and-Hold (S/H) Amplifier
- Automated Threshold Scan and Compare Operation to Pre-Evaluate Conversion Results
- Selectable Conversion Trigger Source
- Fixed-Length (one word per channel), Configurable Conversion Result Buffer
- Four Options for Results Alignment
- Configurable Interrupt Generation
- Operation During CPU Sleep and Idle modes

The 12-bit A/D Converter module is an enhanced version of the 10-bit module offered in some PIC24 devices. Both modules are Successive Approximation Register (SAR) converters at their cores, surrounded by a range of hardware features for flexible configuration. This version of the module extends functionality by providing 12-bit resolution, a wider range of automatic sampling options and tighter integration with other analog modules, such as the CTMU and a configurable results buffer. This module also includes a unique Threshold Detect feature that allows the module itself to make simple decisions based on the conversion results.

A simplified block diagram for the module is illustrated in Figure 22-1.

FIGURE 22-1: 12-BIT A/D CONVERTER BLOCK DIAGRAM



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 (ANS<12:10>, ANS<5:0>).
   b) Select voltage reference source to match the expected range on the 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<7:4> and AD1CON3<12:8>).
   e) Select how 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.

To perform an A/D sample and conversion using Threshold Detect scanning:

1. Configure the A/D module:
   a) Configure the port pins as analog inputs (ANS<12:10>, ANS<5,0>).
   b) Select the voltage reference source to match the expected range on the 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<7:4>, 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>).
2. Configure the threshold compare channels:
   a) Enable auto-scan – ASEN bit (AD1CON5<15>).
   b) Select the Compare mode, “Greater Than, Less Than or Windowed” – CMx bits (AD1CON5<1:0>).
   c) Select the threshold compare channels to be scanned (ADCSSH, ADCSSL).
   d) If the CTMU is required as a current source for a threshold compare channel, enable the corresponding CTMU channel (ADCCTMUENH, ADCCTMUENL).
   e) Write the threshold values into the corresponding ADC1BUFn registers.
   f) Turn on the A/D module (AD1CON1<15>).

Note:	If performing an A/D sample and conversion using Threshold Detect in Sleep Mode, the RC A/D clock source must be selected before entering into Sleep mode.

3. Configure the A/D interrupt (OPTIONAL):
   a) Clear the AD1IF bit.
   b) Select the A/D interrupt priority.

22.1 A/D Control Registers

The 12-bit A/D Converter module uses up to 43 registers for its operation. All registers are mapped in the data memory space.

22.1.1 CONTROL REGISTERS

Depending on the specific device, the module has up to eleven control and status registers:

- AD1CON1: A/D Control Register 1
- AD1CON2: A/D Control Register 2
- AD1CON3: A/D Control Register 3
- AD1CON5: A/D Control Register 5
- AD1CHS: A/D Sample Select Register
- AD1CHITH and AD1CHITL: A/D Scan Compare Hit Registers
- AD1CSSL and AD1CSSH: A/D Input Scan Select Registers
- AD1CTMUENH and AD1CTMUENL: CTMU Enable Registers

The AD1CON1, AD1CON2 and AD1CON3 registers (Register 22-1, Register 22-2 and Register 22-3) control the overall operation of the A/D module. This includes enabling the module, configuring the conversion clock and voltage reference sources, selecting the sampling and conversion triggers, and manually controlling the sample/convert sequences. The AD1CON5 register (Register 22-4) specifically controls features of the Threshold Detect operation, including its function in power-saving modes.

The AD1CHS register (Register 22-5) selects the input channels to be connected to the S/H amplifier. It also allows the choice of input multiplexers and the selection of a reference source for differential sampling.

The AD1CHITH and AD1CHITL registers (Register 22-6 and Register 22-7) are semaphore registers used with Threshold Detect operations. The status of individual bits, or bit pairs in some cases, indicate if a match condition has occurred. AD1CHITL is always implemented, whereas AD1CHITH may not be implemented in devices with 16 or fewer channels.

The AD1CSSH/L registers (Register 22-8 and Register 22-9) select the channels to be included for sequential scanning.

The AD1CTMUENH/L registers (Register 22-10 and Register 22-11) select the channel(s) to be used by the CTMU during conversions. Selecting a particular channel allows the A/D Converter to control the CTMU (particularly, its current source) and read its data through that channel. AD1CTMUENL is always implemented, whereas AD1CTMUENH may not be implemented in devices with 16 or fewer channels.

22.1.2 A/D RESULT BUFFERS

The module incorporates a multi-word, dual port RAM, called ADC1BUF. The buffer is composed of at least the same number of word locations as there are external analog channels for a particular device, with a maximum number of 32. The number of buffer addresses is always even. Each of the locations is mapped into the data memory space and is separately addressable. The buffer locations are referred to as ADC1BUF0 through ADC1BUFn (up to 31).

The A/D result buffers are both readable and writable. When the module is active (AD1CON<15> = 1), the buffers are read-only, and store the results of A/D conversions. When the module is inactive (AD1CON<15> = 0), the buffers are both readable and writable. In this state, writing to a buffer location programs a conversion threshold for Threshold Detect operations.

Buffer contents are not cleared when the module is deactivated with the ADON bit (AD1CON1<15>). Conversion results and any programmed threshold values are maintained when ADON is set or cleared.

REGISTER 22-1: AD1CON1: A/D CONTROL REGISTER 1

R/W-0	U-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0
ADON	—	ADSIDL	—	—	MODE12	FORM1	FORM0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0, HSC	R/C-0, HSC
SSRC3	SSRC2	SSRC1	SSRC0	—	ASAM	SAMP	DONE
bit 7							bit 0

Legend:	C = Clearable bit	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 **ADON:** A/D Operating Mode bit
1 = A/D Converter module is operating
0 = A/D Converter is off

bit 14 **Unimplemented:** Read as ‘0’

bit 13 **ADSIDL:** A/D Stop in Idle Mode bit
1 = Discontinues module operation when device enters Idle mode
0 = Continues module operation in Idle mode

bit 12-11 **Unimplemented:** Read as ‘0’

bit 10 **MODE12:** 12-Bit Operation Mode bit
1 = 12-bit A/D operation
0 = 10-bit A/D operation

bit 9-8 **FORM<1:0>:** Data Output Format bits (see the following formats)
11 = Fractional result, signed, left-justified
10 = Absolute fractional result, unsigned, left-justified
01 = Decimal result, signed, right-justified
00 = Absolute decimal result, unsigned, right-justified

bit 7-4 **SSRC<3:0>:** Sample Clock Source Select bits
1111 = Not available; do not use
•
•
•
1000 = Not available; do not use
0111 = Internal counter ends sampling and starts conversion (auto-convert)
0110 = Not available; do not use
0101 = Timer1 event ends sampling and starts conversion
0100 = CTMU event ends sampling and starts conversion
0011 = Timer5 event ends sampling and starts conversion
0010 = Timer3 event ends sampling and starts conversion
0001 = INT0 event ends sampling and starts conversion
0000 = Clearing the SAMP bit in software ends sampling and begins conversion

bit 3 **Unimplemented:** Read as ‘0’

bit 2 **ASAM:** A/D Sample Auto-Start bit
1 = Sampling begins immediately after the last conversion; SAMP bit is auto-set
0 = Sampling begins when the SAMP bit is manually set

bit 1 **SAMP:** A/D Sample Enable bit
1 = A/D Sample-and-Hold amplifiers are sampling
0 = A/D Sample-and-Hold amplifiers are holding

bit 0 **DONE:** A/D Conversion Status bit
1 = A/D conversion cycle has completed
0 = A/D conversion cycle has not started or is in progress

REGISTER 22-2: AD1CON2: A/D CONTROL REGISTER 2

R/W-0	R/W-0	R/W-0	r-0	R/W-0	R/W-0	U-0	U-0
PVCFG1	PVCFG0	NVCFG0	—	BUFREGEN	CSCNA	—	—
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
BUFS[1]	SMPI4	SMPI3	SMPI2	SMPI1	SMPI0	BUFM[1]	ALTS
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 15-14 **PVCFG<1:0>:** Converter Positive Voltage Reference Configuration bits
`11` = 4 * Internal VBG[2]
`10` = 2 * Internal VBG[3]
`01` = External VREF+
`00` = AVDD

bit 13 **NVCFG0:** Converter Negative Voltage Reference Configuration bits
`1` = External VREF-
`0` = AVSS

bit 12 **Reserved:** Maintain as ‘`0`’

bit 11 **BUFREGEN:** A/D Buffer Register Enable bit
`1` = Conversion result is loaded into a buffer location determined by the converted channel
`0` = A/D result buffer is treated as a FIFO

bit 10 **CSCNA:** Scan Input Selections for CH0+ S/H Input for MUX A Setting bit
`1` = Scans inputs
`0` = Does not scan inputs

bit 9-8 **Unimplemented:** Read as ‘`0`’

bit 7 **BUFS:** Buffer Fill Status bit[1]
`1` = A/D is filling the upper half of the buffer; user should access data in the lower half
`0` = A/D is filling the lower half of the buffer; user should access data in the upper half

bit 6-2 **SMPI<4:0>:** Sample Rate Interrupt Select bits
`11111` = Interrupts at the completion of the conversion for each 32nd sample
`11110` = Interrupts at the completion of the conversion for each 31st sample
•
•
•
`00001` = Interrupts at the completion of the conversion for every other sample
`00000` = Interrupts at the completion of the conversion for each sample

bit 1 **BUFM:** Buffer Fill Mode Select bit[1]
`1` = Starts filling the buffer at address, AD1BUF0, on the first interrupt and AD1BUF(n/2) on the next interrupt (Split Buffer mode)
`0` = Starts filling the buffer at address, ADCBUF0, and each sequential address on successive interrupts (FIFO mode)

Note 1: This is only applicable when the buffer is used in FIFO mode (BUFREGEN = `0`). In addition, BUFS is only used when BUFM = `1`.
2: The voltage reference setting will not be within the specification with VDD below 4.5V.
3: The voltage reference setting will not be within the specification with VDD below 2.3V.

REGISTER 22-2: AD1CON2: A/D CONTROL REGISTER 2 (CONTINUED)

bit 0 **ALTS:** Alternate Input Sample Mode Select bit

1 = Uses channel input selects for Sample A on the first sample and Sample B on the next sample
0 = Always uses channel input selects for Sample A

Note 1: This is only applicable when the buffer is used in FIFO mode (BUFREGEN = 0). In addition, BUFS is only used when BUFM = 1.

2: The voltage reference setting will not be within the specification with V_{DD} below 4.5V.

3: The voltage reference setting will not be within the specification with V_{DD} below 2.3V.

REGISTER 22-3: 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	EXTSAM	—	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 = RC clock
0 = Clock is derived from the system clock

bit 14 **EXTSAM:** Extended Sampling Time bit
1 = A/D is still sampling after SAMP = 0
0 = A/D is finished sampling

bit 13 **Reserved:** Maintain as ‘0’

bit 12-8 **SAMC<4:0>:** Auto-Sample Time Select bits
11111 = 31 TAD
•
•
•
00001 = 1 TAD
00000 = 0 TAD

bit 7-0 **ADCS<7:0>:** A/D Conversion Clock Select bits
11111111-01000000 = Reserved
00111111 = 64 · TCY = TAD
•
•
•
00000001 = 2 · TCY = TAD
00000000 = TCY = TAD

REGISTER 22-4: AD1CON5: A/D CONTROL REGISTER 5

R/W-0	R/W-0	R/W-0	R/W-0	r-0	U-0	R/W-0	R/W-0
ASEN[1]	LPEN	CTMREQ	BGREQ	—	—	ASINT1	ASINT0
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	WM1	WM0	CM1	CM0
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 **ASEN:** Auto-Scan Enable bit[1]
1 = Auto-scan is enabled
0 = Auto-scan is disabled

bit 14 **LPEN:** Low-Power Enable bit
1 = Returns to Low-Power mode after scan
0 = Remains in Full-Power mode after scan

bit 13 **CTMREQ:** CTMU Request bit
1 = CTMU is enabled when the A/D is enabled and active
0 = CTMU is not enabled by the A/D

bit 12 **BGREQ:** Band Gap Request bit
1 = Band gap is enabled when the A/D is enabled and active
0 = Band gap is not enabled by the A/D

bit 11 **Reserved:** Maintain as ‘0’

bit 10 **Unimplemented:** Read as ‘0’

bit 9-8 **ASINT<1:0>:** Auto-Scan (Threshold Detect) Interrupt Mode bits
11 = Interrupt after a Threshold Detect sequence completed and a valid compare has occurred
10 = Interrupt after a valid compare has occurred
01 = Interrupt after a Threshold Detect sequence completed
00 = No interrupt

bit 7-4 **Unimplemented:** Read as ‘0’

bit 3-2 **WM<1:0>:** Write Mode bits
11 = Reserved
10 = Auto-compare only (conversion results are not saved, but interrupts are generated when a valid match, as defined by the CMx and ASINTx bits, occurs)
01 = Convert and save (conversion results are saved to locations as determined by the register bits when a match, as defined by the CMx bits, occurs)
00 = Legacy operation (conversion data is saved to a location determined by the buffer register bits)

bit 1-0 **CM<1:0>:** Compare Mode bits
11 = Outside Window mode (valid match occurs if the conversion result is outside of the window defined by the corresponding buffer pair)
10 = Inside Window mode (valid match occurs if the conversion result is inside the window defined by the corresponding buffer pair)
01 = Greater Than mode (valid match occurs if the result is greater than the value in the corresponding buffer register)
00 = Less Than mode (valid match occurs if the result is less than the value in the corresponding buffer register)

Note 1: When using auto-scan with Threshold Detect (ASEN = 1), do not configure the sample clock source to Auto-Convert mode (SSRCx = 7). Any other available SSRCx selection is valid. To use auto-convert as the sample clock source (SSRCx = 7), make sure ASEN is cleared.

REGISTER 22-5: AD1CHS: A/D SAMPLE SELECT 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
CH0NB2	CH0NB1	CH0NB0	CH0SB4	CH0SB3	CH0SB2	CH0SB1	CH0SB0
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
CH0NA2	CH0NA1	CH0NA0	CH0SA4	CH0SA3	CH0SA2	CH0SA1	CH0SA0
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 **CH0NB<2:0>:** Sample B Channel 0 Negative Input Select bits
111 = AN6[(1)]
110 = AN5[(2)]
101 = AN4
100 = AN3
011 = AN2
010 = AN1
001 = AN0
000 = AVSS

bit 12-8 **CH0SB<4:0>:** S/H Amplifier Positive Input Select for MUX B Multiplexer Setting bits
11111 = Unimplemented, do not use
11110 = AVDD
11101 = AVSS
11100 = Upper guardband rail (0.785 * VDD)
11011 = Lower guardband rail (0.215 * VDD)
11010 = Internal Band Gap Reference (VBG)[(3)]
11001-10010 = Unimplemented, do not use
10001 = No channels are connected, all inputs are floating (used for CTMU)
10000 = No channels are connected, all inputs are floating (used for CTMU temperature sensor input)
01111 = AN15
01110 = AN14
01101 = AN13
01100 = AN12
01011 = AN11
01010 = AN10
01001 = AN9
01000 = AN8[(1)]
00111 = AN7[(1)]
00110 = AN6[(1)]
00101 = AN5[(2)]
00100 = AN4
00011 = AN3
00010 = AN2
00001 = AN1
00000 = AN0

bit 7-5 **CH0NA<2:0>:** Sample A Channel 0 Negative Input Select bits
The same definitions as for CHONB<2:0>.

bit 4-0 **CH0SA<4:0>:** Sample A Channel 0 Positive Input Select bits
The same definitions as for CHONA<4:0>.

Note 1: This is implemented on 44-pin devices only.
2: This is implemented on 28-pin and 44-pin devices only.
3: The band gap value used for this input is 2x or 4x the internal VBG, which is selected when PVCFG<1:0> = 1x.

REGISTER 22-6: AD1CHITH: A/D SCAN COMPARE HIT REGISTER (HIGH WORD)[1]

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
—	—	—	—	—	—	CHH<17:16>	
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 1-0 **CHH<17:16>:** A/D Compare Hit bits

If CM<1:0> = 11:
1 = A/D Result Buffer x has been written with data or a match has occurred
0 = A/D Result Buffer x has not been written with data

For All Other Values of CM<1:0>:
1 = A match has occurred on A/D Result Channel x
0 = No match has occurred on A/D Result Channel x

Note 1: Unimplemented channels are read as ‘0’.

REGISTER 22-7: AD1CHITL: A/D SCAN COMPARE HIT REGISTER (LOW WORD)[1]

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CHH<15:8>							
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
CHH<7:0>							
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 15-0 **CHH<15:0>:** A/D Compare Hit bits

If CM<1:0> = 11:
1 = A/D Result Buffer x has been written with data or a match has occurred
0 = A/D Result Buffer x has not been written with data

For all other values of CM<1:0>:
1 = A match has occurred on A/D Result Channel x
0 = No match has occurred on A/D Result Channel x

Note 1: Unimplemented channels are read as ‘0’.

REGISTER 22-8: AD1CSSH: A/D INPUT SCAN SELECT REGISTER (HIGH WORD)[1]

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
—	CSS<30:26>					—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	CSS<17:16>	
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 14-10 **CSS<30:26>:** A/D Input Scan Selection bits
1 = Includes corresponding channel for input scan
0 = Skips channel for input scan

bit 9-2 **Unimplemented:** Read as ‘0’

bit 1-0 **CSS<17:16>:** A/D Input Scan Selection bits
1 = Includes corresponding channel for input scan
0 = Skips channel for input scan

Note 1: Unimplemented channels are read as ‘0’. Do not select unimplemented channels for sampling as indeterminate results may be produced.

REGISTER 22-9: AD1CSSL: A/D INPUT SCAN SELECT REGISTER (LOW WORD)[1]

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CSS<15:8>							
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
CSS<7:0>							
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 15-0 **CSS<15:0>:** A/D Input Scan Selection bits
1 = Includes corresponding ANx input for scan
0 = Skips channel for input scan

Note 1: Unimplemented channels are read as ‘0’. Do not select unimplemented channels for sampling as indeterminate results may be produced.

REGISTER 22-10: AD1CTMUENH: A/D CTMU ENABLE REGISTER (HIGH WORD)(1)

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
—	—	—	—	—	—	CTMEN<17:16>	
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 1-0 **CTMEN<17:16>:** CTMU Enabled During Conversion bits
1 = CTMU is enabled and connected to the selected channel during conversion
0 = CTMU is not connected to this channel

Note 1: Unimplemented channels are read as ‘0’.

REGISTER 22-11: AD1CTMUENL: A/D CTMU ENABLE REGISTER (LOW WORD)(1)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CTMEN<15:8>							
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
CTMEN<7:0>							
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 15-0 **CTMEN<15:0>:** CTMU Enabled During Conversion bits
1 = CTMU is enabled and connected to the selected channel during conversion
0 = CTMU is not connected to this channel

Note 1: Unimplemented channels are read as ‘0’.

22.2 A/D Sampling Requirements

The analog input model of the 12-bit A/D Converter is shown in Figure 22-2. The total sampling time for the A/D is a function of the holding capacitor charge time.

For the A/D Converter to meet its specified accuracy, the Charge Holding Capacitor (C_{HOLD}) must be allowed to fully charge to the voltage level on the analog input pin. The Source (R_S) impedance, the Interconnect (R_{IC}) impedance and the internal Sampling Switch (R_{SS}) impedance combine to directly affect the time required to charge C_{HOLD}. The combined impedance of the analog sources must, therefore, be small enough to fully charge the holding capacitor within the chosen sample time. To minimize the effects of pin leakage currents on the accuracy of the A/D Converter, the maximum recommended Source impedance, R_S, is 2.5 kΩ. After the analog input channel is selected (changed), this sampling function must be completed prior to starting the conversion. The internal holding capacitor will be in a discharged state prior to each sample operation.

At least 1 T_{AD} time period should be allowed between conversions for the sample time. For more details, see **Section 29.0 "Electrical Characteristics"**.

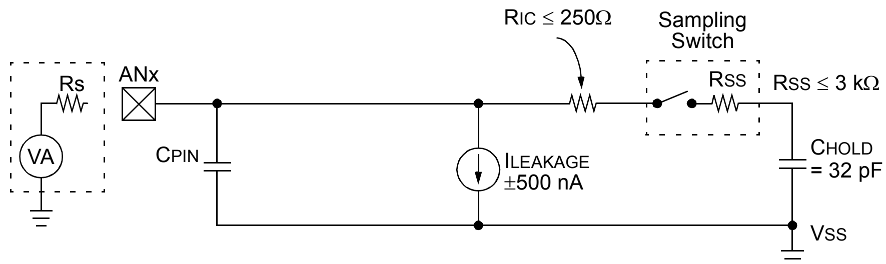
EQUATION 22-1: A/D CONVERSION CLOCK PERIOD

$$T_{AD} = T_{CY}(ADCS + 1)$$

$$ADCS = \frac{T_{AD}}{T_{CY}} - 1$$

Note: Based on $T_{CY} = 2/F_{OSC}$; Doze mode and PLL are disabled.

FIGURE 22-2: 12-BIT A/D CONVERTER ANALOG INPUT MODEL



Legend:

C_{PIN}	=	Input Capacitance
V_T	=	Threshold Voltage
$I_{LEAKAGE}$	=	Leakage Current at the Pin Due to Various Junctions
R_{IC}	=	Interconnect Resistance
R_{SS}	=	Sampling Switch Resistance
C_{HOLD}	=	Sample-and-Hold Capacitance (from DAC)

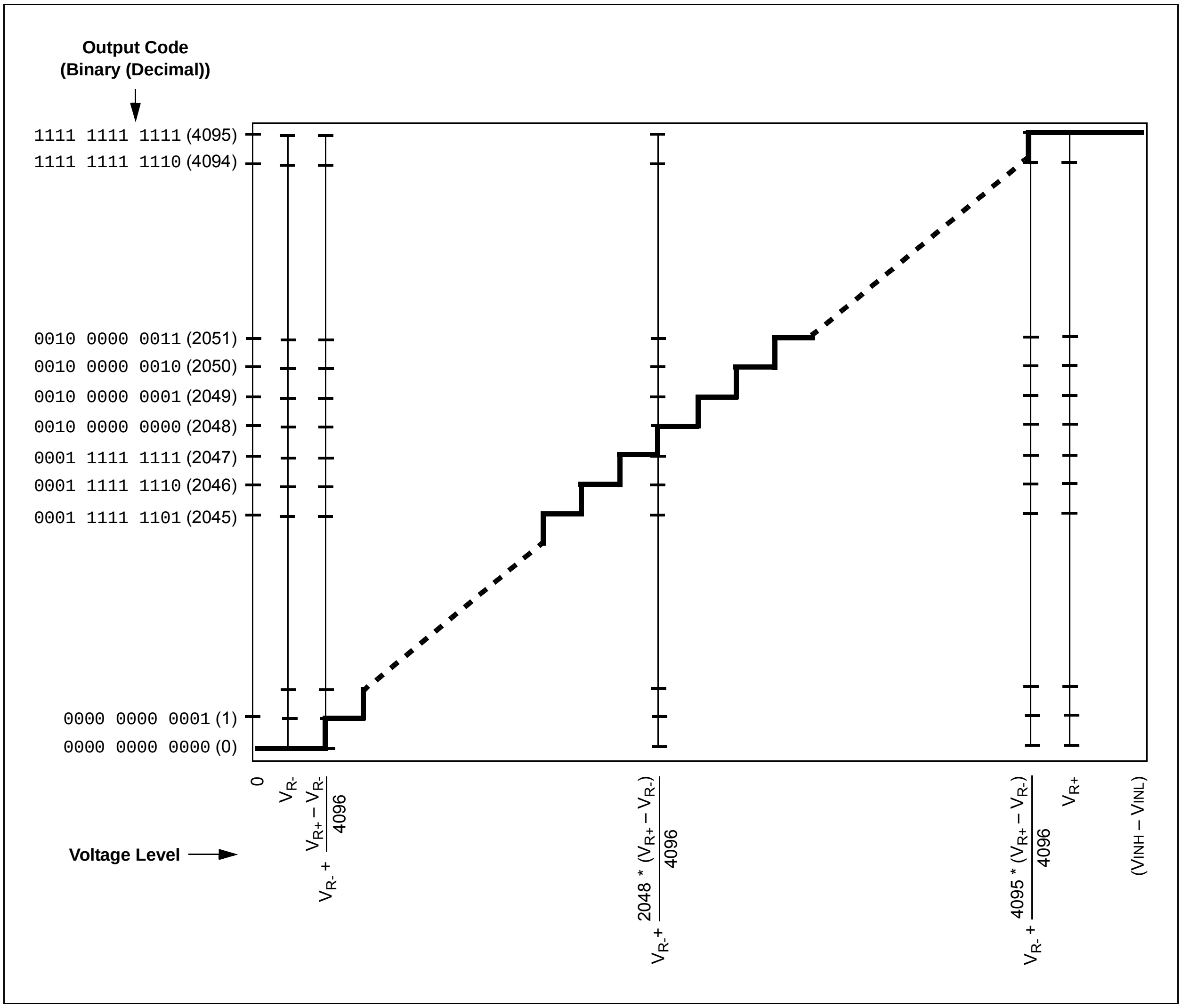
Note: The C_{PIN} value depends on the device package and is not tested. The effect of C_{PIN} is negligible if $R_S \le 5$ kΩ.

22.3 Transfer Function

The transfer functions of the A/D Converter in 12-bit resolution are shown in Figure 22-3. The difference of the input voltages, (V_{INH} – V_{INL}), is compared to the reference, ((V_{R+}) – (V_{R-})).

- The first code transition occurs when the input voltage is ((V_{R+}) – (V_{R-}))/4096 or 1.0 LSb.
- The '`0000 0000 0001`' code is centered at V_{R-} + (1.5 * ((V_{R+}) – (V_{R-}))/4096).
- The '`0010 0000 0000`' code is centered at V_{REFL} + (2048.5 * ((V_{R+}) – (V_{R-}))/4096).
- An input voltage less than V_{R-} + (((V_{R-}) – (V_{R-}))/4096) converts as '`0000 0000 0000`'.
- An input voltage greater than (V_{R-}) + (4095 ((V_{R+}) – (V_{R-}))/4096) converts as '`1111 1111 1111`'.

FIGURE 22-3: 12-BIT A/D TRANSFER FUNCTION



22.4 Buffer Data Formats

The A/D conversions are fully differential 12-bit values when MODE12 = 1 (AD1CON1<10>) and 10-bit values when MODE12 = 0. When absolute fractional or absolute integer formats are used, the results are 12 or 10 bits wide, respectively. When signed decimal formatting is used, the conversion also includes a sign bit, making 12-bit conversions 13 bits wide, and 10-bit conversions 11 bits wide. The signed decimal format yields 12-bit and 10-bit values, respectively. The sign bit (bit 12 or bit 10) is sign-extended to fill the buffer. The FORM<1:0> bits (AD1CON1<9:8>) select the format. Figure 22-4 and Figure 22-5 show the data output formats that can be selected. Table 22-1 through Table 22-4 show the numerical equivalents for the various conversion result codes.

FIGURE 22-4: A/D OUTPUT DATA FORMATS (12-BIT)

RAM Contents:					d11	d10	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00
Read to Bus:																
Integer	0	0	0	0	d11	d10	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00
Signed Integer	s0	s0	s0	s0	d11	d10	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00
Fractional (1.15)	d11	d10	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00	0	0	0	0
Signed Fractional (1.15)	s0	d11	d10	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00	0	0	0

TABLE 22-1: NUMERICAL EQUIVALENTS OF VARIOUS RESULT CODES: 12-BIT INTEGER FORMATS

VIN/VREF	12-Bit Differential Output Code (13-bit result)	16-Bit Integer Format/ Equivalent Decimal Value		16-Bit Signed Integer Format/ Equivalent Decimal Value	
+4095/4096	0 1111 1111 1111	0000 1111 1111 1111	+4095	0000 1111 1111 1111	+4095
+4094/4096	0 1111 1111 1110	0000 1111 1111 1110	+4094	0000 1111 1111 1110	+4094
• • •					
+1/4096	0 1000 0000 0001	0000 0000 0000 0001	+1	0000 0000 0000 0001	+1
0/4096	0 0000 0000 0000	0000 0000 0000 0000	0	0000 0000 0000 0000	0
-1/4096	1 0111 1111 1111	0000 0000 0000 0000	0	1111 1111 1111 1111	-1
• • •					
-4095/4096	1 0000 0000 0001	0000 0000 0000 0000	0	1111 0000 0000 0001	-4095
-4096/4096	1 0000 0000 0000	0000 0000 0000 0000	0	1111 0000 0000 0000	-4096

TABLE 22-2: NUMERICAL EQUIVALENTS OF VARIOUS RESULT CODES: 12-BIT FRACTIONAL FORMATS

VIN/VREF	12-Bit Output Code	16-Bit Fractional Format/ Equivalent Decimal Value		16-Bit Signed Fractional Format/ Equivalent Decimal Value	
+4095/4096	0 1111 1111 1111	1111 1111 1111 0000	0.999	0111 1111 1111 1000	0.999
+4094/4096	0 1111 1111 1110	1111 1111 1110 0000	0.998	0111 1111 1110 1000	0.998
• • •					
+1/4096	0 0000 0000 0001	0000 0000 0001 0000	0.001	0000 0000 0000 1000	0.001
0/4096	0 0000 0000 0000	0000 0000 0000 0000	0.000	0000 0000 0000 0000	0.000
-1/4096	1 0111 1111 1111	0000 0000 0000 0000	0.000	1111 1111 1111 1000	-0.001
• • •					
-4095/4096	1 0000 0000 0001	0000 0000 0000 0000	0.000	1000 0000 0000 1000	-0.999
-4096/4096	1 0000 0000 0000	0000 0000 0000 0000	0.000	1000 0000 0000 0000	-1.000

FIGURE 22-5: A/D OUTPUT DATA FORMATS (10-BIT)

RAM Contents:							d09	d08	d07	d06	d05	d04	d03	d02	d01	d00
Read to Bus:																
Integer	0	0	0	0	0	0	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00
Signed Integer	s0	s0	s0	s0	s0	s0	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00
Fractional (1.15)	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00	0	0	0	0	0	0
Signed Fractional (1.15)	s0	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00	0	0	0	0	0

TABLE 22-3: NUMERICAL EQUIVALENTS OF VARIOUS RESULT CODES: 10-BIT INTEGER FORMATS

VIN/VREF	10-Bit Differential Output Code (11-bit result)	16-Bit Integer Format/ Equivalent Decimal Value		16-Bit Signed Integer Format/ Equivalent Decimal Value	
+1023/1024	011 1111 1111	0000 0011 1111 1111	1023	0000 0001 1111 1111	1023
+1022/1024	011 1111 1110	0000 0011 1111 1110	1022	0000 0001 1111 1110	1022
• • •					
+1/1024	000 0000 0001	0000 0000 0000 0001	1	0000 0000 0000 0001	1
0/1024	000 0000 0000	0000 0000 0000 0000	0	0000 0000 0000 0000	0
-1/1024	101 1111 1111	0000 0000 0000 0000	0	1111 1111 1111 1111	-1
• • •					
-1023/1024	100 0000 0001	0000 0000 0000 0000	0	1111 1110 0000 0001	-1023
-1024/1024	100 0000 0000	0000 0000 0000 0000	0	1111 1110 0000 0000	-1024

TABLE 22-4: NUMERICAL EQUIVALENTS OF VARIOUS RESULT CODES: 10-BIT FRACTIONAL FORMATS

VIN/VREF	10-Bit Differential Output Code (11-bit result)	16-Bit Fractional Format/ Equivalent Decimal Value		16-Bit Signed Fractional Format/ Equivalent Decimal Value	
+1023/1024	011 1111 1111	1111 1111 1100 0000	0.999	0111 1111 1110 0000	0.999
+1022/1024	011 1111 1110	1111 1111 1000 0000	0.998	0111 1111 1000 0000	0.998
• • •					
+1/1024	000 0000 0001	0000 0000 0100 0000	0.001	0000 0000 0010 0000	0.001
0/1024	000 0000 0000	0000 0000 0000 0000	0.000	0000 0000 0000 0000	0.000
-1/1024	101 1111 1111	0000 0000 0000 0000	0.000	1111 1111 1110 0000	-0.001
• • •					
-1023/1024	100 0000 0001	0000 0000 0000 0000	0.000	1000 0000 0010 0000	-0.999
-1024/1024	100 0000 0000	0000 0000 0000 0000	0.000	1000 0000 0000 0000	-1.000

23.0 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 on the Comparator module, refer to the *"dsPIC33/PIC24 Family Reference Manual"*, **"Scalable Comparator Module"** (DS39734).

The comparator module provides three dual input comparators. The inputs to the comparator can be configured to use any one of four external analog inputs, as well as a voltage reference input from either the internal band gap reference, divided by 2 (VBG/2), or the comparator voltage reference generator.

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 is shown in Figure 23-1. Diagrams of the possible individual comparator configurations are shown in Figure 23-2.

Each comparator has its own control register, CMxCON (Register 23-1), for enabling and configuring its operation. The output and event status of all three comparators is provided in the CMSTAT register (Register 23-2).

FIGURE 23-1: COMPARATOR x MODULE BLOCK DIAGRAM

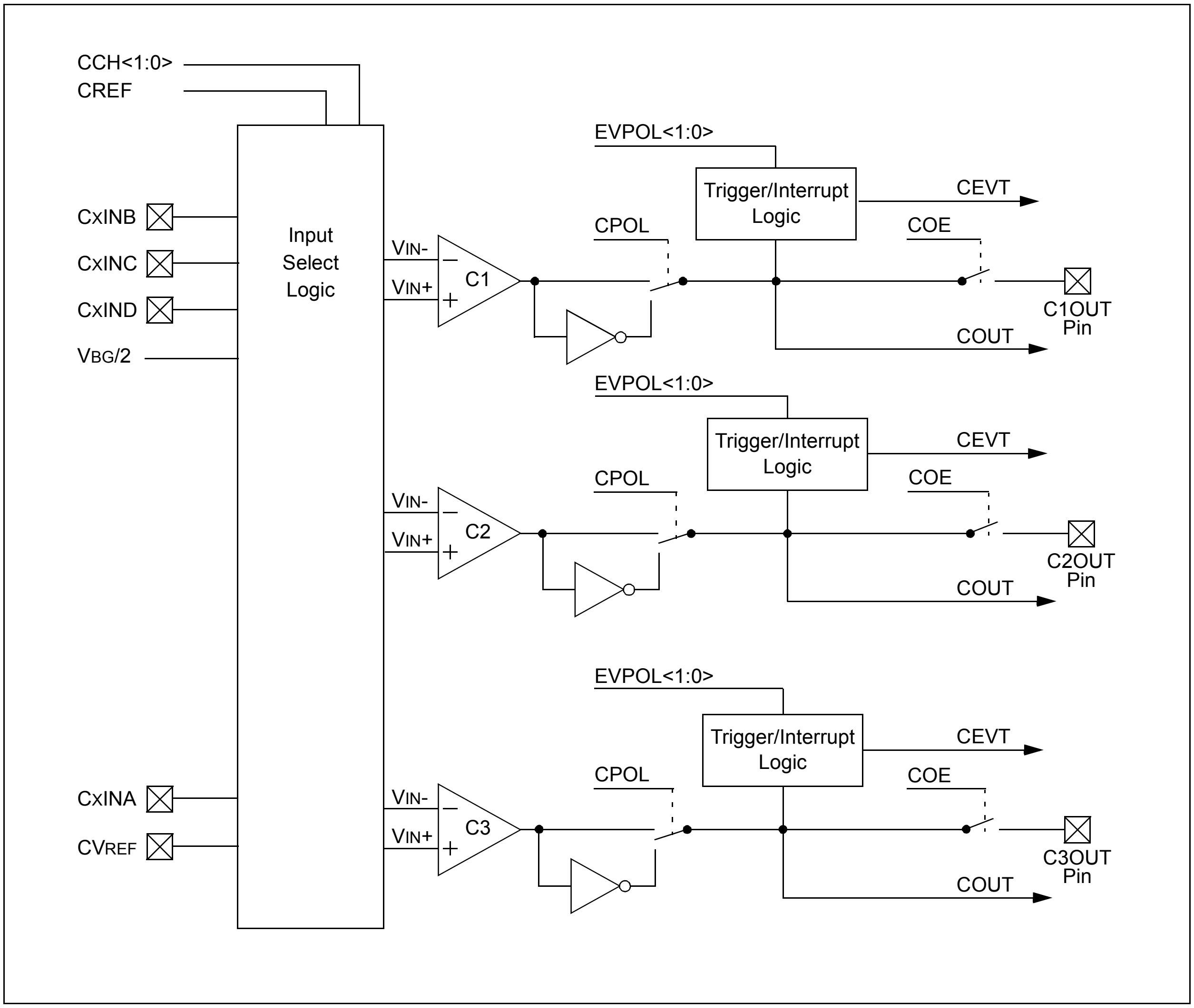
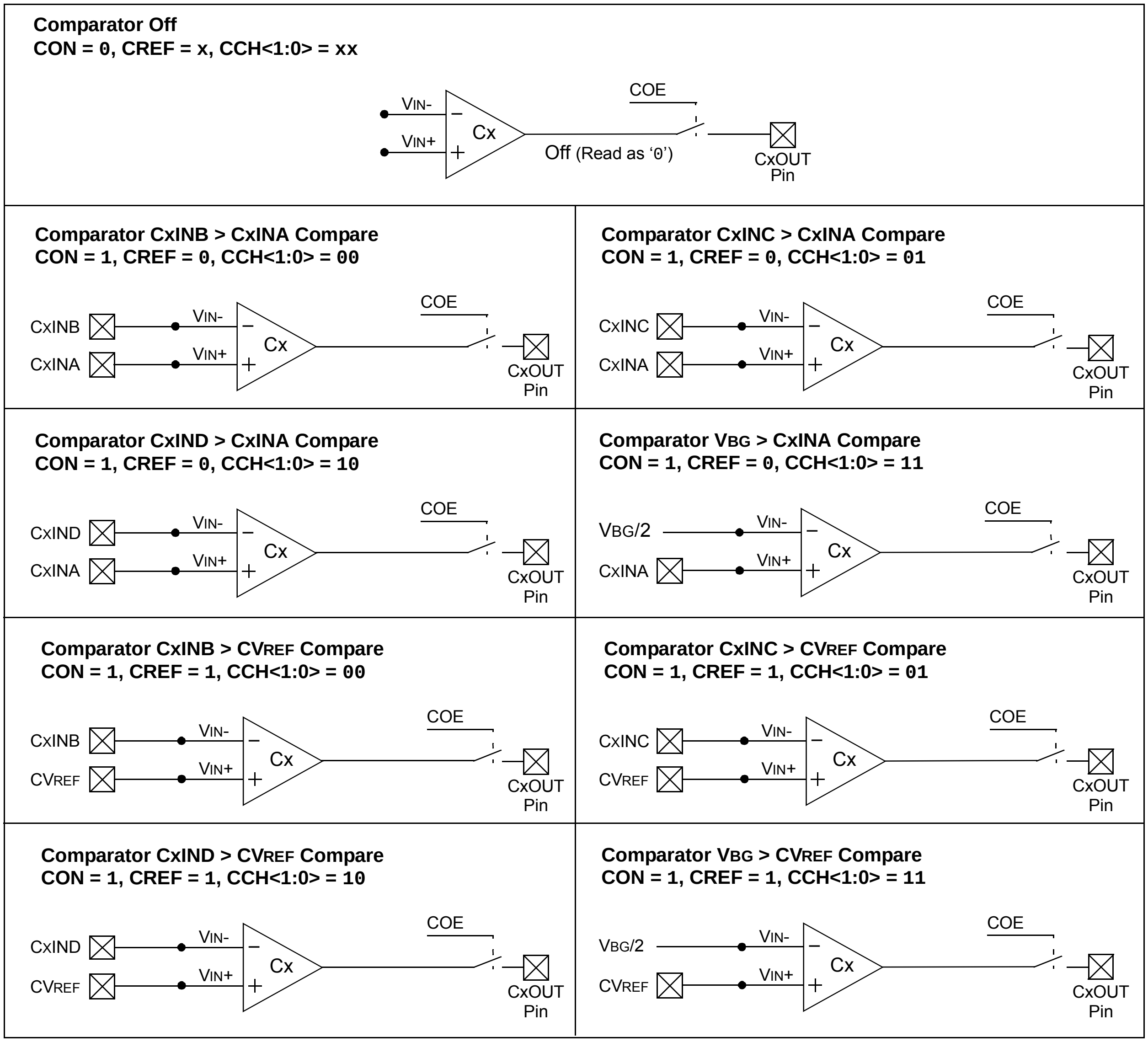


FIGURE 23-2: INDIVIDUAL COMPARATOR CONFIGURATIONS



REGISTER 23-1: CMxCON: COMPARATOR x CONTROL REGISTERS

R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R-0
CON	COE	CPOL	CLPWR	—	—	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:			
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 **CON:** Comparator x Enable bit
1 = Comparator is enabled
0 = Comparator is disabled

bit 14 **COE:** Comparator x Output Enable bit
1 = Comparator output is present on the CxOUT pin
0 = Comparator output is internal only

bit 13 **CPOL:** Comparator x Output Polarity Select bit
1 = Comparator output is inverted
0 = Comparator output is not inverted

bit 12 **CLPWR:** Comparator x Low-Power Mode Select bit
1 = Comparator operates in Low-Power mode, transient response is reduced
0 = Comparator does not operate in Low-Power mode

bit 11-10 **Unimplemented:** Read as ‘0’

bit 9 **CEVT:** Comparator x Event bit
1 = Comparator event 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 x 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 the 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 the transition of the 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 23-1: CMxCON: COMPARATOR x CONTROL REGISTERS (CONTINUED)

bit 4	**CREF:** Comparator x 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 x Channel Select bits 11 = Inverting input of the comparator connects to VBG 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 23-2: CMSTAT: COMPARATOR x 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 x Stop in Idle Mode bit 1 = Comparator interrupts are disabled in Idle mode; enabled comparators remain operational 0 = Continues 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>).

24.0 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 on the Comparator Voltage Reference, refer to the *"dsPIC33/PIC24 Family Reference Manual"*, **"Comparator Voltage Reference Module"** (DS39709).

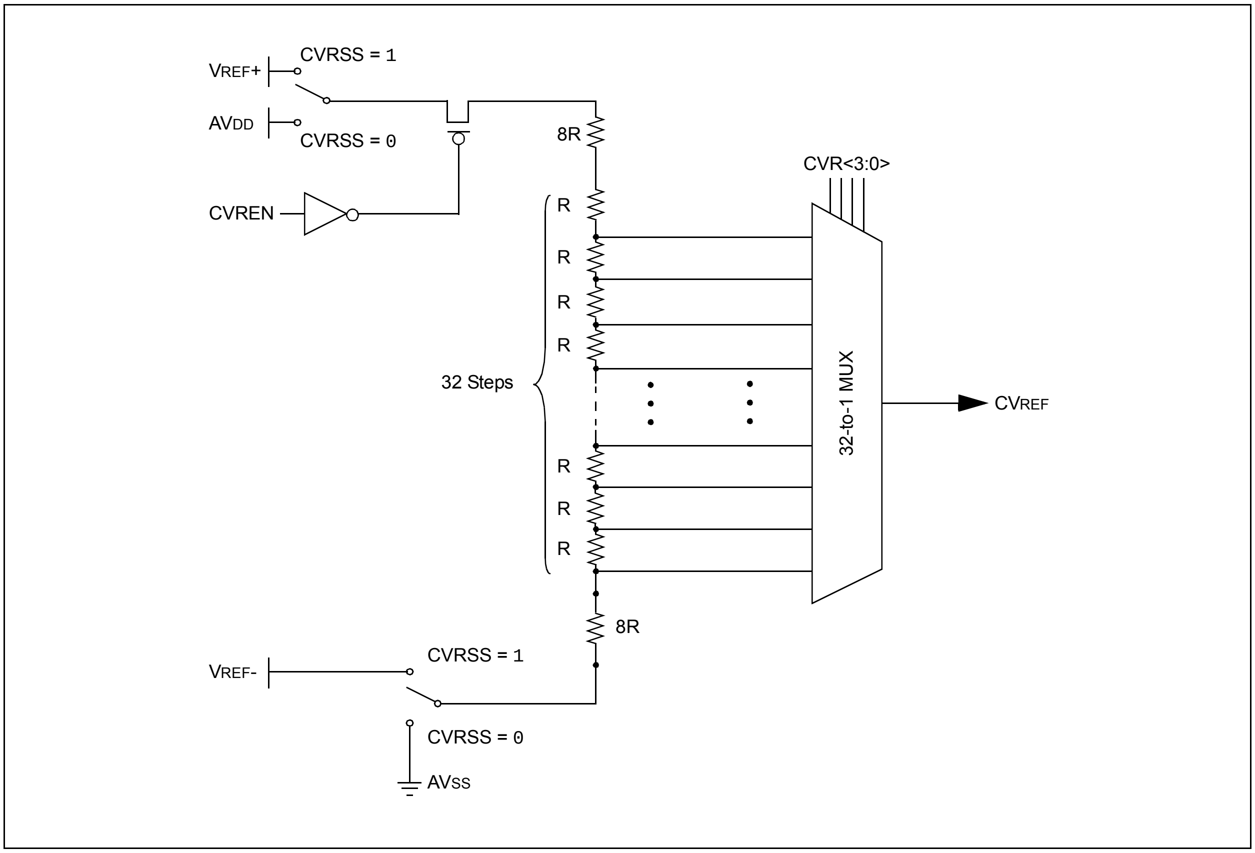
24.1 Configuring the Comparator Voltage Reference

The comparator voltage reference module is controlled through the CVRCON register (Register 24-1). The comparator voltage reference provides a range of output voltages, with 32 distinct levels.

The comparator voltage 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<5>).

The settling time of the comparator voltage reference must be considered when changing the CVREF output.

FIGURE 24-1: COMPARATOR VOLTAGE REFERENCE BLOCK DIAGRAM



REGISTER 24-1: CVRCON: COMPARATOR VOLTAGE REFERENCE 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
CVREN	CVROE	CVRSS	CVR4	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 15-8 **Unimplemented:** Read as ‘0’

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 **CVRSS:** Comparator VREF Source Selection bit
1 = Comparator reference source, CVRSRC = VREF+ – VREF-
0 = Comparator reference source, CVRSRC = AVDD – AVSS

bit 4-0 **CVR<4:0>:** Comparator VREF Value Selection 0 ≤ CVR<4:0> ≤ 31 bits
When CVRSS = 1:
CVREF = (VREF-) + (CVR<4:0>/32) • (VREF+ – VREF-)
When CVRSS = 0:
CVREF = (AVSS) + (CVR<4:0>/32) • (AVDD – AVSS)

25.0 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 on the Charge Measurement Unit, refer to the *"dsPIC33/PIC24 Family Reference Manual"*, **"Charge Time Measurement Unit (CTMU) and CTMU Operation with Threshold Detect"** (DS30009743).

The Charge Time Measurement Unit (CTMU) is a flexible analog module that provides charge measurement, accurate differential time measurement between pulse sources and asynchronous pulse generation. Its key features include:

- Thirteen external edge input trigger sources
- Polarity control for each edge source
- Control of edge sequence
- Control of response to edge levels or edge transitions
- Time measurement resolution of one 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 touch sensors.

The CTMU is controlled through three registers: CTMUCON1, CTMUCON2 and CTMUICON. CTMUCON1 enables the module and controls the mode of operation of the CTMU, as well as controlling edge sequencing. CTMUCON2 controls edge source selection and edge source polarity selection. The CTMUICON register selects the current range of current source and trims the current.

25.1 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 several internal peripheral modules (OC1, Timer1, any input capture or comparator module) and up to 13 external pins (CTED1 through CTED13). This pulse is used with the module's precision current source to calculate capacitance according to the relationship:

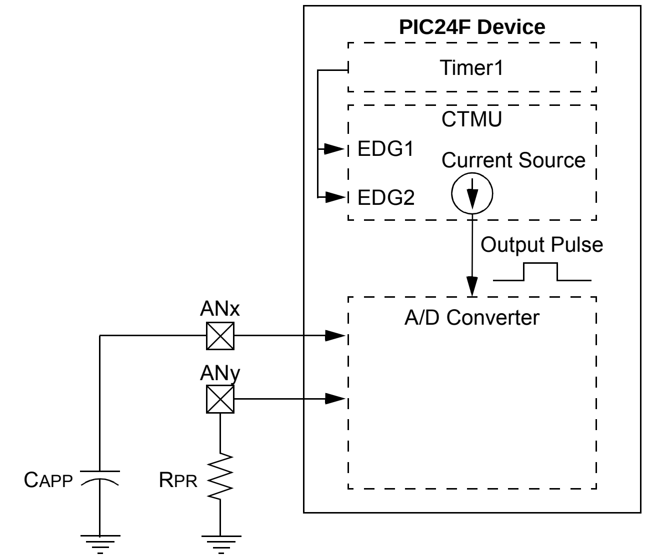
EQUATION 25-1:

$$I = C \cdot \frac{dV}{dT}$$

For capacitance measurements, the A/D Converter samples an external capacitor (C_{APP}) on one of its input channels after the CTMU output's pulse. A Precision Resistor (R_{PR}) 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 25-1 illustrates 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 *"dsPIC33/PIC24 Family Reference Manual"*.

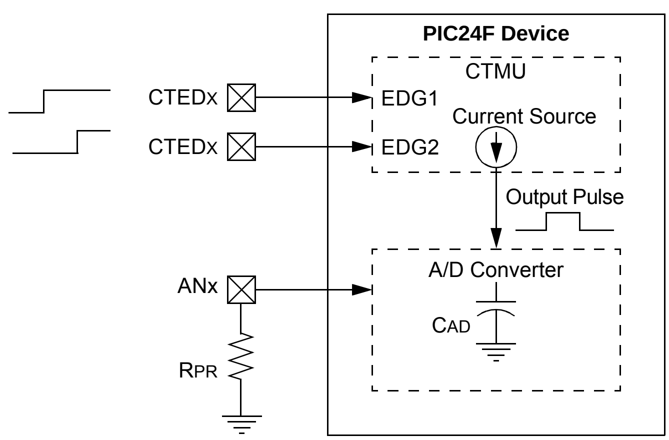
FIGURE 25-1: TYPICAL CONNECTIONS AND INTERNAL CONFIGURATION FOR CAPACITANCE MEASUREMENT



25.2 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 25-2 displays 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 CTEDx pins, but other configurations using internal edge sources are possible.

FIGURE 25-2: TYPICAL CONNECTIONS AND INTERNAL CONFIGURATION FOR TIME MEASUREMENT



25.3 Pulse Generation and Delay

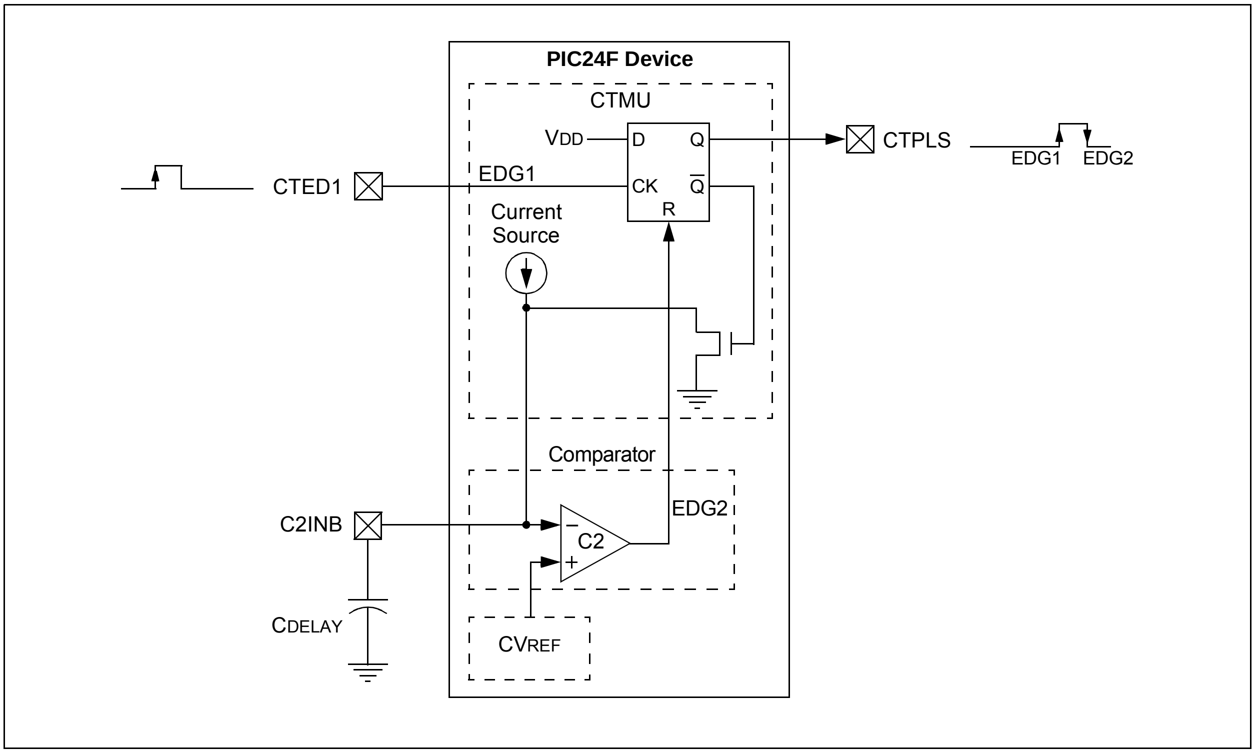
The CTMU module can also generate an output pulse with edges that are not synchronous with the device's system clock. More specifically, it can generate a pulse with a programmable delay from an edge event input to the module.

When the module is configured for pulse generation delay by setting the TGEN bit (CTMUCON<12>), 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. While CVREF is greater than the voltage on CDELAY, CTPLS is high.

When the voltage on CDELAY equals CVREF, CTPLS goes low. With Comparator 2 configured as the second edge, this stops the CTMU from charging. In this state event, the CTMU automatically connects to ground. The IDISSEN bit doesn't need to be set and cleared before the next CTPLS cycle.

Figure 25-3 illustrates the external connections for pulse generation, as well as the relationship of the different analog modules required. While CTED1 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 *"dsPIC33/PIC24 Family Reference Manual"*.

FIGURE 25-3: TYPICAL CONNECTIONS AND INTERNAL CONFIGURATION FOR PULSE DELAY GENERATION



REGISTER 25-1: CTMUCON1: CTMU 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
CTMUEN	—	CTMUSIDL	TGEN	EDGEN	EDGSEQEN	IDISSEN	CTTRIG
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 **CTMUEN:** CTMU Enable bit
1 = Module is enabled
0 = Module is disabled

bit 14 **Unimplemented:** Read as ‘0’

bit 13 **CTMUSIDL:** CTMU Stop in Idle Mode bit
1 = Discontinues module operation when device enters Idle mode
0 = Continues module operation in Idle mode

bit 12 **TGEN:** Time Generation Enable bit
1 = Enables edge delay generation
0 = Disables edge delay generation

bit 11 **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: CTMU** Trigger Control bit
1 = Trigger output is enabled
0 = Trigger output is disabled

bit 7-0 **Unimplemented:** Read as ‘0’

REGISTER 25-2: CTMUCON2: CTMU CONTROL REGISTER 2

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EDG1MOD	EDG1POL	EDG1SEL3	EDG1SEL2	EDG1SEL1	EDG1SEL0	EDG2STAT	EDG1STAT
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
EDG2MOD	EDG2POL	EDG2SEL3	EDG2SEL2	EDG2SEL1	EDG2SEL0	—	—
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 **EDG1MOD:** Edge 1 Edge-Sensitive Select bit
1 = Input is edge-sensitive
0 = Input is level-sensitive

bit 14 **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

bit 13-10 **EDG1SEL<3:0>:** Edge 1 Source Select bits
1111 = Edge 1 source is Comparator 3 output
1110 = Edge 1 source is Comparator 2 output
1101 = Edge 1 source is Comparator 1 output
1100 = Edge 1 source is IC3
1011 = Edge 1 source is IC2
1010 = Edge 1 source is IC1
1001 = Edge 1 source is CTED8
1000 = Edge 1 source is CTED7
0111 = Edge 1 source is CTED6
0110 = Edge 1 source is CTED5
0101 = Edge 1 source is CTED4
0100 = Edge 1 source is CTED3(1)
0011 = Edge 1 source is CTED1
0010 = Edge 1 source is CTED2
0001 = Edge 1 source is OC1
0000 = Edge 1 source is Timer1

bit 9 **EDG2STAT:** Edge 2 Status bit
Indicates the status of Edge 2 and can be written to control the current source.
1 = Edge 2 has occurred
0 = Edge 2 has not occurred

bit 8 **EDG1STAT:** Edge 1 Status bit
Indicates the status of Edge 1 and can be written to control the current source.
1 = Edge 1 has occurred
0 = Edge 1 has not occurred

bit 7 **EDG2MOD:** Edge 2 Edge-Sensitive Select bit
1 = Input is edge-sensitive
0 = Input is level-sensitive

bit 6 **EDG2POL:** Edge 2 Polarity Select bit
1 = Edge 2 is programmed for a positive edge
0 = Edge 2 is programmed for a negative edge

Note 1: Edge sources, CTED3 and CTED11, are not available on PIC24FV32KA301 devices.

REGISTER 25-2: CTMUCON2: CTMU CONTROL REGISTER 2 (CONTINUED)

bit 5-2	**EDG2SEL<3:0>:** Edge 2 Source Select bits 1111 = Edge 2 source is Comparator 3 output 1110 = Edge 2 source is Comparator 2 output 1101 = Edge 2 source is Comparator 1 output 1100 = Unimplemented; do not use 1011 = Edge 2 source is IC3 1010 = Edge 2 source is IC2 1001 = Edge 2 source is IC1 1000 = Edge 2 source is CTED13[(1)] 0111 = Edge 2 source is CTED12[(1)] 0110 = Edge 2 source is CTED11[(1)] 0101 = Edge 2 source is CTED10 0100 = Edge 2 source is CTED9 0011 = Edge 2 source is CTED1 0010 = Edge 2 source is CTED2 0001 = Edge 2 source is OC1 0000 = Edge 2 source is Timer1
bit 1-0	**Unimplemented:** Read as ‘0’

Note 1: Edge sources, CTED3 and CTED11, are not available on PIC24FV32KA301 devices.

REGISTER 25-3: 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 = 1000 × Base Current

bit 7-0 **Unimplemented:** Read as ‘0’

NOTES:

26.0 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 on the Watchdog Timer, High-Level Device Integration and Programming Diagnostics, refer to the individual sections of the *"dsPIC33/PIC24 Family Reference Manual"* provided below:
> - **"Watchdog Timer (WDT)"** (DS39697)
> - **"High-Level Integration with Programmable High/Low-Voltage Detect (HLVD)"** (DS39725)
> - **Section 33. "Programming and Diagnostics"** (DS39716)

PIC24FV32KA304 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
- In-Circuit Serial Programming™ (ICSP™)
- In-Circuit Emulation

26.1 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 complete list of Configuration register locations is provided in Table 26-1. A detailed explanation of the various bit functions is provided in Register 26-1 through Register 26-8.

The 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.

TABLE 26-1: CONFIGURATION REGISTERS LOCATIONS

Configuration Register	Address
FBS	F80000
FGS	F80004
FOSCSEL	F80006
FOSC	F80008
FWDT	F8000A
FPOR	F8000C
FICD	F8000E
FDS	F80010

REGISTER 26-1: FBS: BOOT SEGMENT CONFIGURATION REGISTER

U-0	U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	—	BSS2	BSS1	BSS0	BWRP
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 7-4 **Unimplemented:** Read as '0'

bit 3-1 **BSS<2:0>:** Boot Segment Program Flash Code Protection bits
- 111 = No boot program Flash segment
- 011 = Reserved
- 110 = Standard security, boot program Flash segment starts at 200h, ends at 000AFEh
- 010 = High-security boot program Flash segment starts at 200h, ends at 000AFEh
- 101 = Standard security, boot program Flash segment starts at 200h, ends at 0015FEh(1)
- 001 = High-security, boot program Flash segment starts at 200h, ends at 0015FEh(1)
- 100 = Standard security; boot program Flash segment starts at 200h, ends at 002BFEh(1)
- 000 = High-security; boot program Flash segment starts at 200h, ends at 002BFEh(1)

bit 0 **BWRP:** Boot Segment Program Flash Write Protection bit
- 1 = Boot Segment may be written
- 0 = Boot Segment is write-protected

Note 1: This selection should not be used in PIC24FV16KA3XX devices.

REGISTER 26-2: FGS: GENERAL SEGMENT CONFIGURATION REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R/C-1	R/C-1
—	—	—	—	—	—	GSS0	GWRP
bit 7							bit 0

Legend:			
R = Readable bit	C = Clearable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7-2 **Unimplemented:** Read as ‘0’

bit 1 **GSS0:** General Segment Code Flash Code Protection bit
1 = No protection
0 = Standard security is enabled

bit 0 **GWRP:** General Segment Code Flash Write Protection bit
1 = General Segment may be written
0 = General Segment is write-protected

REGISTER 26-3: FOSCSEL: OSCILLATOR SELECTION CONFIGURATION REGISTER

R/P-1	R/P-1	R/P-1	U-0	U-0	R/P-1	R/P-1	R/P-1
IESO	LPRCSEL	SOSCSRC	—	—	FNOSC2	FNOSC1	FNOSC0
bit 7							bit 0

Legend:			
R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7 **IESO:** Internal External Switchover bit
1 = Internal External Switchover mode is enabled (Two-Speed Start-up is enabled)
0 = Internal External Switchover mode is disabled (Two-Speed Start-up is disabled)

bit 6 **LPRCSEL:** Internal LPRC Oscillator Power Select bit
1 = High-Power/High-Accuracy mode
0 = Low-Power/Low-Accuracy mode

bit 5 **SOSCSRC:** Secondary Oscillator Clock Source Configuration bit
1 = SOSC analog crystal function is available on the SOSCI/SOSCO pins
0 = SOSC crystal is disabled; digital SCLKI function is selected on the SOSCO pin

bit 4-3 **Unimplemented:** Read as ‘0’

bit 2-0 **FNOSC<2:0>:** Oscillator Selection bits
000 = Fast RC Oscillator (FRC)
001 = Fast RC Oscillator with Divide-by-N with PLL module (FRCDIV+PLL)
010 = Primary Oscillator (XT, HS, EC)
011 = Primary Oscillator with PLL module (HS+PLL, EC+PLL)
100 = Secondary Oscillator (SOSC)
101 = Low-Power RC Oscillator (LPRC)
110 = 500 kHz Low-Power FRC Oscillator with Divide-by-N (LPFRCDIV)
111 = 8 MHz FRC Oscillator with Divide-by-N (FRCDIV)

REGISTER 26-4: FOSC: OSCILLATOR CONFIGURATION REGISTER

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
FCKSM1	FCKSM0	SOSCSEL	POSCFREQ1	POSCFREQ0	OSCIOFNC	POSCMD1	POSCMD0
bit 7							bit 0

Legend:			
R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7-6 **FCKSM<1:0>:** Clock Switching and Fail-Safe Clock Monitor Selection Configuration bits
- 1x = Clock switching is disabled, Fail-Safe Clock Monitor is 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 **SOSCSEL:** Secondary Oscillator Power Selection Configuration bit
- 1 = Secondary oscillator is configured for high-power operation
- 0 = Secondary oscillator is configured for low-power operation

bit 4-3 **POSCFREQ<1:0>:** Primary Oscillator Frequency Range Configuration bits
- 11 = Primary oscillator/external clock input frequency is greater than 8 MHz
- 10 = Primary oscillator/external clock input frequency is between 100 kHz and 8 MHz
- 01 = Primary oscillator/external clock input frequency is less than 100 kHz
- 00 = Reserved; do not use

bit 2 **OSCIOFNC:** CLKO Enable Configuration bit
- 1 = CLKO output signal is active on the OSCO pin; primary oscillator must be disabled or configured for the External Clock mode (EC) for the CLKO to be active (POSCMD<1:0> = 11 or 00)
- 0 = CLKO output is disabled

bit 1-0 **POSCMD<1:0>:** Primary Oscillator Configuration bits
- 11 = Primary Oscillator mode is disabled
- 10 = HS Oscillator mode is selected
- 01 = XT Oscillator mode is selected
- 00 = External Clock mode is selected

REGISTER 26-5: FWDT: WATCHDOG TIMER CONFIGURATION REGISTER

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
FWDTEN1	WINDIS	FWDTEN0	FWPSA	WDTPS3	WDTPS2	WDTPS1	WDTPS0
bit 7							bit 0

Legend:			
R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7,5 **FWDTEN<1:0>:** Watchdog Timer Enable bits

11 = WDT is enabled in hardware
10 = WDT is controlled with the SWDTEN bit setting
01 = WDT is enabled only while device is active; WDT is disabled in Sleep, SWDTEN bit is disabled
00 = WDT is disabled in hardware; SWDTEN bit is disabled

bit 6 **WINDIS:** Windowed Watchdog Timer Disable bit

1 = Standard WDT is selected; windowed WDT is disabled
0 = Windowed WDT is enabled; note that executing a CLRWDT instruction while the WDT is disabled in hardware and software (FWDTEN<1:0> = 00 and SWDTEN (RCON<5>) = 0) will not cause a device Reset

bit 4 **FWPSA:** WDT Prescaler bit

1 = WDT prescaler ratio of 1:128
0 = WDT prescaler ratio of 1:32

bit 3-0 **WDTPS<3:0>:** Watchdog Timer Postscale 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

REGISTER 26-6: FPOR: RESET CONFIGURATION REGISTER

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
MCLRE(2)	BORV1(3)	BORV0(3)	I2C1SEL(1)	PWRTEN	RETCFG(1)	BOREN1	BOREN0
bit 7							bit 0

Legend:			
R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7 **MCLRE:** $\overline{\text{MCLR}}$ Pin Enable bit(2)
1 = $\overline{\text{MCLR}}$ pin is enabled; RA5 input pin is disabled
0 = RA5 input pin is enabled; $\overline{\text{MCLR}}$ is disabled

bit 6-5 **BORV<1:0>:** Brown-out Reset Enable bits(3)
11 = Brown-out Reset is set to the lowest voltage
10 = Brown-out Reset
01 = Brown-out Reset is set to the highest voltage
00 = Downside protection on POR is enabled – “zero power” is selected

bit 4 **I2C1SEL:** Alternate I2C1 Pin Mapping bit(1)
1 = Default location for SCL1/SDA1 pins
0 = Alternate location for SCL1/SDA1 pins

bit 3 **PWRTEN:** Power-up Timer Enable bit
1 = PWRT is enabled
0 = PWRT is disabled

bit 2 **RETCFG:** Retention Regulator Configuration bit(1)
1 = Retention Regulator is not available
0 = Retention Regulator is available and controlled by the RETEN bit (RCON<12>) during Sleep

bit 1-0 **BOREN<1:0>:** Brown-out Reset Enable bits
11 = Brown-out Reset is enabled in hardware; SBOREN bit is disabled
10 = Brown-out Reset is enabled only while device is active and disabled in Sleep; SBOREN bit is disabled
01 = Brown-out Reset is controlled with the SBOREN bit setting
00 = Brown-out Reset is disabled in hardware; SBOREN bit is disabled

Note 1: This setting only applies to the “FV” devices. This bit is reserved and should be maintained as ‘1’ on “F” devices.
2: The MCLRE fuse can only be changed when using the VPP-based ICSP™ mode entry. This prevents a user from accidentally locking out the device from the low-voltage test entry.
3: Refer to **Section 29.0 “Electrical Characteristics”** for BOR voltages.

REGISTER 26-7: FICD: IN-CIRCUIT DEBUGGER CONFIGURATION REGISTER

R/P-1	U-0	U-0	U-0	U-0	U-0	R/P-1	R/P-1
$\overline{\text{DEBUG}}$	—	—	—	—	—	FICD1	FICD0
bit 7							bit 0

Legend:			
R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7 **$\overline{\text{DEBUG}}$:** Background Debugger Enable bit
1 = Background debugger is disabled
0 = Background debugger functions are enabled

bit 6-2 **Unimplemented:** Read as ‘0’

bit 1-0 **FICD<1:0:>:** ICD Pin Select bits
11 = PGEC1/PGED1 are used for programming and debugging the device
10 = PGEC2/PGED2 are used for programming and debugging the device
01 = PGEC3/PGED3 are used for programming and debugging the device
00 = Reserved; do not use

REGISTER 26-8: FDS: DEEP SLEEP CONFIGURATION REGISTER

R/P-1	R/P-1	U-0	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
DSWDTEN	DSBOREN	—	DSWDTOSC	DSWDTPS3	DSWDTPS2	DSWDTPS1	DSWDTPS0
bit 7							bit 0

Legend:			
R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7 **DSWDTEN:** Deep Sleep Watchdog Timer Enable bit
1 = DSWDT is enabled
0 = DSWDT is disabled

bit 6 **DSBOREN:** Deep Sleep/Low-Power BOR Enable bit
(does not affect operation in non Deep Sleep modes)
1 = Deep Sleep BOR is enabled in Deep Sleep
0 = Deep Sleep BOR is disabled in Deep Sleep

bit 5 **Unimplemented:** Read as ‘0’

bit 4 **DSWDTOSC:** DSWDT Reference Clock Select bit
1 = DSWDT uses LPRC as the reference clock
0 = DSWDT uses SOSC as the reference clock

bit 3-0 **DSWDTPS<3:0>:** Deep Sleep Watchdog Timer Postscale Select bits
The DSWDT prescaler is 32; this creates an approximate base time unit of 1 ms.
1111 = 1:2,147,483,648 (25.7 days) nominal
1110 = 1:536,870,912 (6.4 days) nominal
1101 = 1:134,217,728 (38.5 hours) nominal
1100 = 1:33,554,432 (9.6 hours) nominal
1011 = 1:8,388,608 (2.4 hours) nominal
1010 = 1:2,097,152 (36 minutes) nominal
1001 = 1:524,288 (9 minutes) nominal
1000 = 1:131,072 (135 seconds) nominal
0111 = 1:32,768 (34 seconds) nominal
0110 = 1:8,192 (8.5 seconds) nominal
0101 = 1:2,048 (2.1 seconds) nominal
0100 = 1:512 (528 ms) nominal
0011 = 1:128 (132 ms) nominal
0010 = 1:32 (33 ms) nominal
0001 = 1:8 (8.3 ms) nominal
0000 = 1:2 (2.1 ms) nominal

REGISTER 26-9: 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	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 15-8 **FAMID<7:0>:** Device Family Identifier bits

01000101 = PIC24FV32KA304 family

bit 7-0 **DEV<7:0>:** Individual Device Identifier bits

00010111 = PIC24FV32KA304
00000111 = PIC24FV16KA304
00010011 = PIC24FV32KA302
00000011 = PIC24FV16KA302
00011001 = PIC24FV32KA301
00001001 = PIC24FV16KA301

00010110 = PIC24F32KA304
00000110 = PIC24F16KA304
00010010 = PIC24F32KA302
00000010 = PIC24F16KA302
00011000 = PIC24F32KA301
00001000 = PIC24F16KA301

REGISTER 26-10: 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
—	—	—	—	REV<3:0>			
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 23-4 **Unimplemented:** Read as ‘0’

bit 3-0 **REV<3:0>:** Minor Revision Identifier bits

26.2 On-Chip Voltage Regulator

All of the PIC24FV32KA304 family devices power their core digital logic at a nominal 3.0V. This may create an issue for designs that are required to operate at a higher typical voltage, as high as 5.0V. To simplify system design, all devices in the "FV" family incorporate an on-chip regulator that allows the device to run its core logic from VDD.

The regulator is always enabled and provides power to the core from the other VDD pins. A low-ESR capacitor (such as ceramic) must be connected to the VCAP pin (Figure 26-1). This helps to maintain the stability of the regulator. The recommended value for the filter capacitor is discussed in **Section 2.4 "Voltage Regulator Pin (VCAP)"**, and in **Section 29.1 "DC Characteristics"**.

For "F" devices, the regulator is disabled. Instead, core logic is powered directly from VDD. This allows the devices to operate at an overall lower allowable voltage range (1.8V-3.6V).

26.2.1 VOLTAGE REGULATOR TRACKING MODE AND LOW-VOLTAGE DETECTION

For all PIC24FV32KA304 devices, the on-chip regulator provides a constant voltage of 3.2V nominal to the digital core logic. The regulator can provide this level from a VDD of about 3.2V, all the way up to the device's VDDMAX. It does not have the capability to boost VDD levels below 3.2V. In order to prevent "brown-out" conditions when the voltage drops too low for the regulator, the regulator enters Tracking mode. In Tracking mode, the regulator output follows VDD with a typical voltage drop of 150 mV.

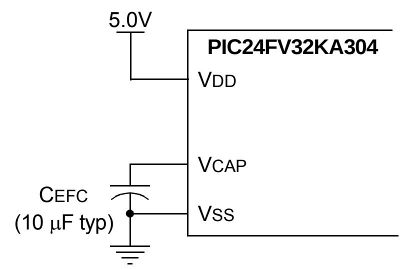
When the device enters Tracking mode, it is no longer possible to operate at full speed. To provide information about when the device enters Tracking mode, the on-chip regulator includes a simple, High/Low-Voltage Detect (HLVD) circuit. When VDD drops below full-speed operating voltage, the circuit sets the High/Low-Voltage Detect Interrupt Flag, HLVDIF (IFS4<8>). This can be used to generate an interrupt and put the application into a low-power operational mode or trigger an orderly shutdown. Maximum device speeds as a function of VDD are shown in **Section 29.1 "DC Characteristics"**, in Figure 29-1 and Figure 29-1.

26.2.2 ON-CHIP REGULATOR AND POR

For PIC24FV32KA304 devices, it takes a brief time, designated as TPM, for the Voltage Regulator to generate a stable output. During this time, code execution is disabled. TPM (DC Specification SY71) is applied every time the device resumes operation after any power-down, including Sleep mode.

FIGURE 26-1: CONNECTIONS FOR THE ON-CHIP REGULATOR

Regulator Enabled(1):



Note 1: These are typical operating voltages. Refer to **Section 29.0 "Electrical Characteristics"** for the full operating ranges of VDD.

26.3 Watchdog Timer (WDT)

For the PIC24FV32KA304 family of 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 Configuration bits, WDTPS<3:0> (FWDT<3:0>), which allow 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 NOSCx 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 in hardware (FWDTEN<1:0> = 11), it will continue to run during Sleep or Idle modes. When the WDT time-out occurs, the device will wake and code execution will continue from where the PWRSAV instruction was executed. The corresponding SLEEP or IDLE bit (RCON<3:2>) 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.

26.3.1 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.

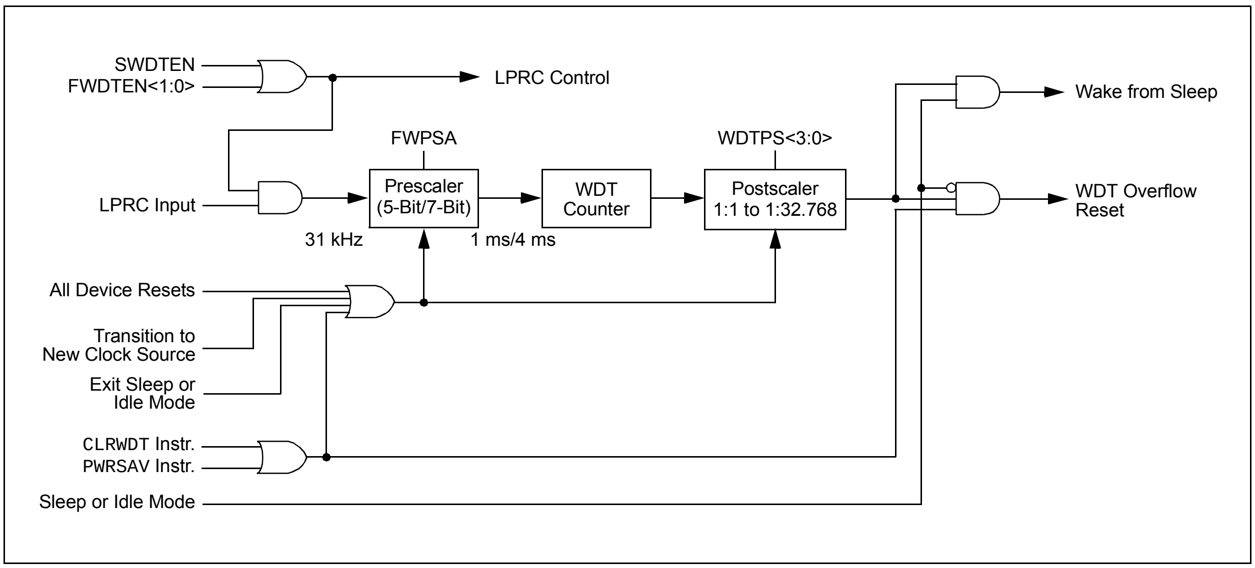
Windowed WDT mode is enabled by programming the Configuration bit, WINDIS (FWDT<6>), to '0'.

26.3.2 CONTROL REGISTER

The WDT is enabled or disabled by the FWDTEN<1:0> Configuration bits. When both the FWDTEN<1:0> Configuration bits are set, the WDT is always enabled.

The WDT can be optionally controlled in software when the FWDTEN<1:0> Configuration bits have been programmed to '10'. 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. When the FWTEN<1:0> bits are set to '01', the WDT is only enabled in Run and Idle modes, and is disabled in Sleep. Software control of the SWDTEN bit (RCON<5>) is disabled with this setting.

FIGURE 26-2: WDT BLOCK DIAGRAM



26.4 Deep Sleep Watchdog Timer (DSWDT)

In PIC24FV32KA304 family devices, in addition to the WDT module, a DSWDT module is present which runs while the device is in Deep Sleep, if enabled. It is driven by either the SOSC or LPRC oscillator. The clock source is selected by the Configuration bit, DSWDTOSC (FDS<4>).

The DSWDT can be configured to generate a time-out, at 2.1 ms to 25.7 days, by selecting the respective postscaler. The postscaler can be selected by the Configuration bits, DSWDTPS<3:0> (FDS<3:0>). When the DSWDT is enabled, the clock source is also enabled.

DSWDT is one of the sources that can wake-up the device from Deep Sleep mode.

26.5 Program Verification and Code Protection

For all devices in the PIC24FV32KA304 family, code protection for the Boot Segment (BS) is controlled by the Configuration bit, BSS0, and the General Segment (GS) by the Configuration bit, GSS0. These bits inhibit external reads and writes to the program memory space This has no direct effect in normal execution mode.

Write protection is controlled by bit, BWRP, for the Boot Segment and bit, GWRP, for the General Segment in the Configuration Word. When these bits are programmed to '0', internal write and erase operations to program memory are blocked.

26.6 In-Circuit Serial Programming

PIC24FV32KA304 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, ground and the programming voltage. 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.

26.7 In-Circuit Debugger

When MPLAB® ICD 3, MPLAB REAL ICE™ or PICkit™ 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 and PGEDx pins.

To use the in-circuit debugger function of the device, the design must implement ICSP connections to $\overline{\text{MCLR}}$, V_{DD}, V_{SS}, PGECx, PGEDx and the pin pair. 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.

27.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers (MCU) and dsPIC® digital signal controllers (DSC) are supported with a full range of software and hardware development tools:

- Integrated Development Environment
  - MPLAB® X IDE Software
- Compilers/Assemblers/Linkers
  - MPLAB XC Compiler
  - MPASM™ Assembler
  - MPLINK™ Object Linker/ MPLIB™ Object Librarian
  - MPLAB Assembler/Linker/Librarian for Various Device Families
- Simulators
  - MPLAB X SIM Software Simulator
- Emulators
  - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debuggers/Programmers
  - MPLAB ICD 3
  - PICkit™ 3
- Device Programmers
  - MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards, Evaluation Kits and Starter Kits
- Third-party development tools

27.1 MPLAB X Integrated Development Environment Software

The MPLAB X IDE is a single, unified graphical user interface for Microchip and third-party software, and hardware development tool that runs on Windows®, Linux and Mac OS® X. Based on the NetBeans IDE, MPLAB X IDE is an entirely new IDE with a host of free software components and plug-ins for high-performance application development and debugging. Moving between tools and upgrading from software simulators to hardware debugging and programming tools is simple with the seamless user interface.

With complete project management, visual call graphs, a configurable watch window and a feature-rich editor that includes code completion and context menus, MPLAB X IDE is flexible and friendly enough for new users. With the ability to support multiple tools on multiple projects with simultaneous debugging, MPLAB X IDE is also suitable for the needs of experienced users.

Feature-Rich Editor:

- Color syntax highlighting
- Smart code completion makes suggestions and provides hints as you type
- Automatic code formatting based on user-defined rules
- Live parsing

User-Friendly, Customizable Interface:

- Fully customizable interface: toolbars, toolbar buttons, windows, window placement, etc.
- Call graph window

Project-Based Workspaces:

- Multiple projects
- Multiple tools
- Multiple configurations
- Simultaneous debugging sessions

File History and Bug Tracking:

- Local file history feature
- Built-in support for Bugzilla issue tracker

27.2 MPLAB XC Compilers

The MPLAB XC Compilers are complete ANSI C compilers for all of Microchip's 8, 16 and 32-bit MCU and DSC devices. These compilers provide powerful integration capabilities, superior code optimization and ease of use. MPLAB XC Compilers run on Windows, Linux or MAC OS X.

For easy source level debugging, the compilers provide debug information that is optimized to the MPLAB X IDE.

The free MPLAB XC Compiler editions support all devices and commands, with no time or memory restrictions, and offer sufficient code optimization for most applications.

MPLAB XC Compilers include an assembler, linker and utilities. 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. MPLAB XC Compiler uses the assembler to produce its object 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 X IDE compatibility

27.3 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 X IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multipurpose source files
- Directives that allow complete control over the assembly process

27.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler. 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

27.5 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC DSC devices. MPLAB XC 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 X IDE compatibility

27.6 MPLAB X SIM Software Simulator

The MPLAB X 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 X SIM Software Simulator fully supports symbolic debugging using the MPLAB XC 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.

27.7 MPLAB REAL ICE In-Circuit Emulator System

The 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 all 8, 16 and 32-bit MCU, and DSC devices with the easy-to-use, powerful graphical user interface of the MPLAB X IDE.

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 (RJ-11) 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 X IDE. MPLAB REAL ICE offers significant advantages over competitive emulators including full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, logic probes, a ruggedized probe interface and long (up to three meters) interconnection cables.

27.8 MPLAB ICD 3 In-Circuit Debugger System

The MPLAB ICD 3 In-Circuit Debugger System is Microchip's most cost-effective, high-speed hardware debugger/programmer for Microchip Flash DSC and MCU devices. It debugs and programs PIC Flash microcontrollers and dsPIC DSCs with the powerful, yet easy-to-use graphical user interface of the MPLAB 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.

27.9 PICkit 3 In-Circuit Debugger/ Programmer

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 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 a 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™ (ICSP™).

27.10 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at VDDMIN and VDDMAX 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.

27.11 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.

27.12 Third-Party Development Tools

Microchip also offers a great collection of tools from third-party vendors. These tools are carefully selected to offer good value and unique functionality.

- Device Programmers and Gang Programmers from companies, such as SoftLog and CCS
- Software Tools from companies, such as Gimpel and Trace Systems
- Protocol Analyzers from companies, such as Saleae and Total Phase
- Demonstration Boards from companies, such as MikroElektronika, Digilent® and Olimex
- Embedded Ethernet Solutions from companies, such as EZ Web Lynx, WIZnet and IPLogika®

28.0 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 28-1 lists the general symbols used in describing the instructions. The PIC24F instruction set summary in Table 28-2 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 of 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 (PC) 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 28-1: 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...16384}
lit16	16-bit unsigned literal ∈ {0...65535}
lit23	23-bit unsigned literal ∈ {0...8388608}; 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 28-2: 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 28-2: 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	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 28-2: 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	None
	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 28-2: 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 – ($\overline{\text{C}}$)	1	1	C, DC, N, OV, Z
	SUBB f,WREG	WREG = f – WREG – ($\overline{\text{C}}$)	1	1	C, DC, N, OV, Z
	SUBB #lit10,Wn	Wn = Wn – lit10 – ($\overline{\text{C}}$)	1	1	C, DC, N, OV, Z
	SUBB Wb,Ws,Wd	Wd = Wb – Ws – ($\overline{\text{C}}$)	1	1	C, DC, N, OV, Z
	SUBB Wb,#lit5,Wd	Wd = Wb – lit5 – ($\overline{\text{C}}$)	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 – ($\overline{\text{C}}$)	1	1	C, DC, N, OV, Z
	SUBBR f,WREG	WREG = WREG – f – ($\overline{\text{C}}$)	1	1	C, DC, N, OV, Z
	SUBBR Wb,Ws,Wd	Wd = Ws – Wb – ($\overline{\text{C}}$)	1	1	C, DC, N, OV, Z
	SUBBR Wb,#lit5,Wd	Wd = lit5 – Wb – ($\overline{\text{C}}$)	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 28-2: 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:

29.0 ELECTRICAL CHARACTERISTICS

This section provides an overview of the PIC24FV32KA304 family electrical characteristics. Additional information will be provided in future revisions of this document as it becomes available.

Absolute maximum ratings for the PIC24FV32KA304 family devices 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(†)

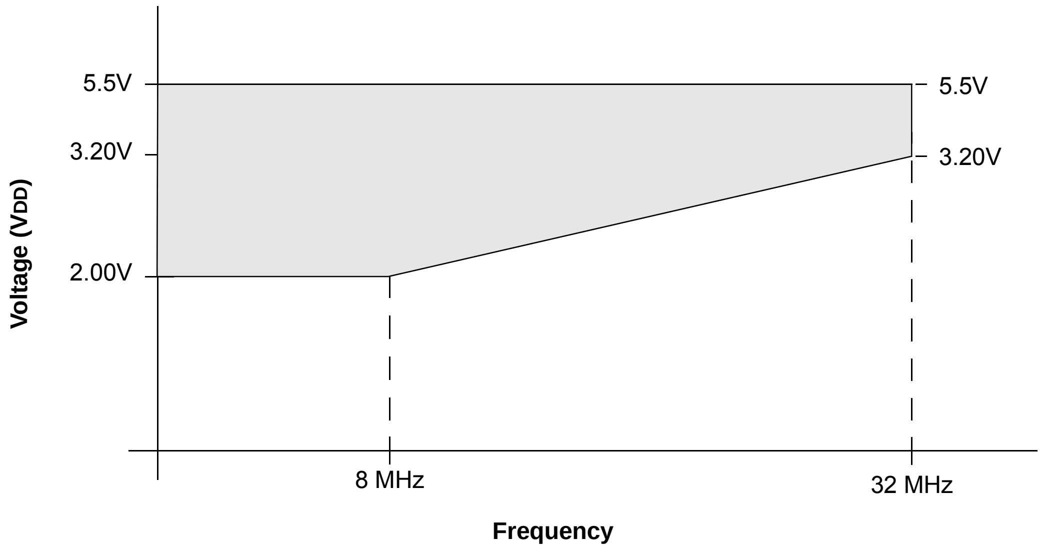
Ambient temperature under bias .... -40°C to +125°C
Storage temperature .... -65°C to +150°C
Voltage on VDD with respect to VSS (PIC24FVXXKA30X) .... -0.3V to +6.5V
Voltage on VDD with respect to VSS (PIC24FXXKA30X) .... -0.3V to +4.5V
Voltage on any combined analog and digital pin with respect to VSS .... -0.3V to (VDD + 0.3V)
Voltage on any digital only pin with respect to VSS .... -0.3V to (VDD + 0.3V)
Voltage on $\overline{MCLR}$/VPP pin with respect to VSS .... -0.3V to +9.0V
Maximum current out of VSS pin .... 300 mA
Maximum current into VDD pin[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[1] .... 200 mA

Note 1: Maximum allowable current is a function of the device maximum power dissipation (see Table 29-1).

† **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.

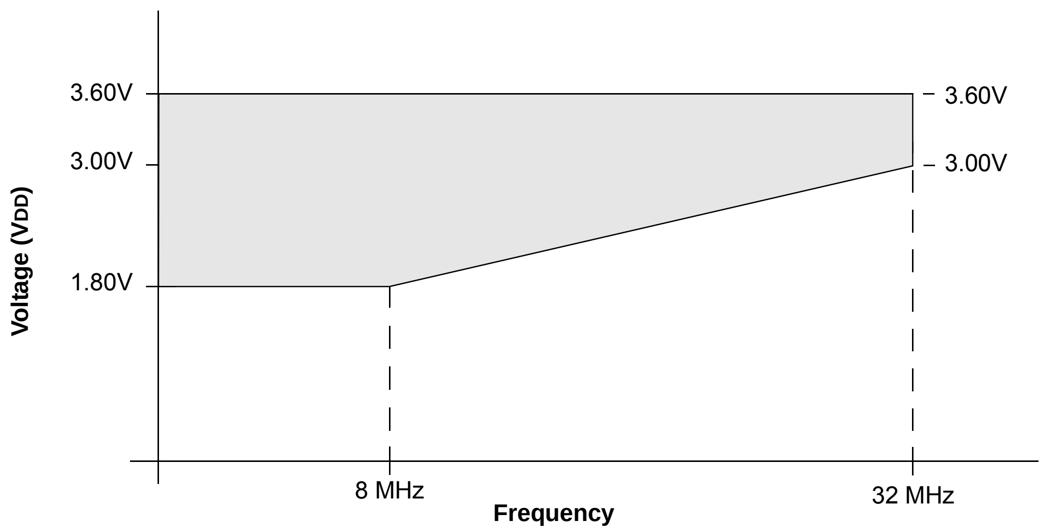
29.1 DC Characteristics

FIGURE 29-1: PIC24FV32KA304 VOLTAGE-FREQUENCY GRAPH (INDUSTRIAL AND EXTENDED)



Note: For frequencies between 8 MHz and 32 MHz, F_{MAX} = 20 MHz * (V_{DD} – 2.0) + 8 MHz.

FIGURE 29-2: PIC24F32KA304 FAMILY VOLTAGE-FREQUENCY GRAPH (INDUSTRIAL AND EXTENDED)



Note: For frequencies between 8 MHz and 32 MHz, F_{MAX} = 20 MHz * (V_{DD} – 1.8) + 8 MHz.

TABLE 29-1: THERMAL OPERATING CONDITIONS

Rating	Symbol	Min	Typ	Max	Unit
Operating Junction Temperature Range	T_J	-40	—	+140	°C
Operating Ambient Temperature Range	T_A	-40	—	+125	°C
Power Dissipation: Internal Chip Power Dissipation: $P_{INT} = V_{DD} \times (I_{DD} - \Sigma I_{OH})$ I/O Pin Power Dissipation: $P_{I/O} = \Sigma (\{V_{DD} - V_{OH}\} \times I_{OH}) + \Sigma (V_{OL} \times I_{OL})$	P_D	$P_{INT} + P_{I/O}$			W
Maximum Allowed Power Dissipation	P_{DMAX}	$(T_J - T_A)/\theta_{JA}$			W

TABLE 29-2: THERMAL PACKAGING CHARACTERISTICS

Characteristic	Symbol	Typ	Max	Unit	Notes
Package Thermal Resistance, 20-Pin SPDIP	θ_{JA}	62.4	—	°C/W	1
Package Thermal Resistance, 28-Pin SPDIP	θ_{JA}	60	—	°C/W	1
Package Thermal Resistance, 20-Pin SSOP	θ_{JA}	108	—	°C/W	1
Package Thermal Resistance, 28-Pin SSOP	θ_{JA}	71	—	°C/W	1
Package Thermal Resistance, 20-Pin SOIC	θ_{JA}	75	—	°C/W	1
Package Thermal Resistance, 28-Pin SOIC	θ_{JA}	80.2	—	°C/W	1
Package Thermal Resistance, 28-Pin QFN	θ_{JA}	32	—	°C/W	1
Package Thermal Resistance, 44-Pin QFN	θ_{JA}	29	—	°C/W	1
Package Thermal Resistance, 48-Pin UQFN	θ_{JA}	—	—	°C/W	1

Note 1: Junction to ambient thermal resistance, Theta-JA (θ_{JA}) numbers are achieved by package simulations.

TABLE 29-3: DC CHARACTERISTICS: TEMPERATURE AND VOLTAGE SPECIFICATIONS

DC CHARACTERISTICS			**Standard Operating Conditions: 1.8V to 3.6V PIC24F32KA3XX 2.0V to 5.5V PIC24FV32KA3XX** Operating temperature -40°C ≤ T_A ≤ +85°C for Industrial -40°C ≤ T_A ≤ +125°C for Extended				
Param No.	**Symbol**	**Characteristic**	**Min**	**Typ[1]**	**Max**	**Units**	**Conditions**
DC10	V_{DD}	**Supply Voltage**	1.8	—	3.6	V	For F devices
			2.0	—	5.5	V	For FV devices
DC12	V_{DR}	**RAM Data Retention Voltage[2]**	1.5	—	—	V	For F devices
			1.7	—	—	V	For FV devices
DC16	V_{POR}	**V_{DD} Start Voltage** to Ensure Internal Power-on Reset Signal	V_{SS}	—	50	mV	V_{DD} must be maintained in this range for at least 64ms.
DC17	SV_{DD}	**V_{DD} Rise Rate** to Ensure Internal Power-on Reset Signal	0.05	—	—	V/ms	0-3.3V in 0.1s 0-2.5V in 60 ms

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: This is the limit to which V_{DD} can be lowered without losing RAM data.

TABLE 29-4: HIGH/LOW–VOLTAGE DETECT CHARACTERISTICS

Standard Operating Conditions: 1.8V to 3.6V PIC24F32KA3XX 2.0V to 5.5V PIC24FV32KA3XX Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended								
Param No.	**Symbol**	**Characteristic**		**Min**	**Typ**	**Max**	**Units**	**Conditions**
DC18	VHLVD	HLVD Voltage on VDD Transition	HLVDL<3:0> = 0000[2]	—	—	1.90	V	
			HLVDL<3:0> = 0001	1.86	—	2.13	V	
			HLVDL<3:0> = 0010	2.08	—	2.35	V	
			HLVDL<3:0> = 0011	2.22	—	2.53	V	
			HLVDL<3:0> = 0100	2.30	—	2.62	V	
			HLVDL<3:0> = 0101	2.49	—	2.84	V	
			HLVDL<3:0> = 0110	2.73	—	3.10	V	
			HLVDL<3:0> = 0111	2.86	—	3.25	V	
			HLVDL<3:0> = 1000	3.00	—	3.41	V	
			HLVDL<3:0> = 1001	3.16	—	3.59[1]	V	
			HLVDL<3:0> = 1010[1]	3.33	—	3.79	V	
			HLVDL<3:0> = 1011[1]	3.53	—	4.01	V	
			HLVDL<3:0> = 1100[1]	3.74	—	4.26	V	
			HLVDL<3:0> = 1101[1]	4.00	—	4.55	V	
			HLVDL<3:0> = 1110[1]	4.28	—	4.87	V	

Note 1: These trip points should not be used on PIC24FXXKA30X devices.
2: This trip point should not be used on PIC24FVXXKA30X devices.

TABLE 29-5: BOR TRIP POINTS

Standard Operating Conditions: 1.8V to 3.6V PIC24F32KA3XX 2.0V to 5.5V PIC24FV32KA3XX Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended								
Param No.	**Sym**	**Characteristic**		**Min**	**Typ**	**Max**	**Units**	**Conditions**
DC15		BOR Hysteresis		—	5	—	mV	
DC19		BOR Voltage on VDD Transition	BORV<1:0> = 00	—	—	—	—	Valid for LPBOR and DSBOR (**Note 1**)
			BORV<1:0> = 01	2.90	3	3.38	V	
			BORV<1:0> = 10	2.53	2.7	3.07	V	
			BORV<1:0> = 11	1.75	1.85	2.05	V	**(Note 2)**
			BORV<1:0> = 11	1.95	2.05	2.16	V	**(Note 3)**

Note 1: LPBOR re-arms the POR circuit but does not cause a BOR.
2: This is valid for PIC24F (3.3V) devices.
3: This is valid for PIC24FV (5V) devices.

TABLE 29-6: DC CHARACTERISTICS: OPERATING CURRENT (I_{DD})

DC CHARACTERISTICS		Standard Operating Conditions: 1.8V to 3.6V PIC24F32KA3XX 2.0V to 5.5V PIC24FV32KA3XX Operating temperature: $-40°C \le T_A \le +85°C$ for Industrial $-40°C \le T_A \le +125°C$ for Extended				
Parameter No.	**Device**	**Typical**	**Max**	**Units**	**Conditions**	
I_{DD} Current						
D20	PIC24FV32KA3XX	269	450	µA	2.0V	0.5 MIPS, F_{OSC} = 1 MHz(1)
		465	830	µA	5.0V	
	PIC24F32KA3XX	200	330	µA	1.8V	
		410	750	µA	3.3V	
DC22	PIC24FV32KA3XX	490	—	µA	2.0V	1 MIPS, F_{OSC} = 2 MHz(1)
		880	—	µA	5.0V	
	PIC24F32KA3XX	407	—	µA	1.8V	
		800	—	µA	3.3V	
DC24	PIC24FV32KA3XX	13.0	20.0	mA	5.0V	16 MIPS, F_{OSC} = 32 MHz(1)
	PIC24F32KA3XX	12.0	18.0	mA	3.3V	
DC26	PIC24FV32KA3XX	2.0	—	mA	2.0V	FRC (4 MIPS), F_{OSC} = 8 MHz
		3.5	—	mA	5.0V	
	PIC24F32KA3XX	1.80	—	mA	1.8V	
		3.40	—	mA	3.3V	
DC30	PIC24FV32KA3XX	48.0	250	µA	2.0V	LPRC (15.5 KIPS), F_{OSC} = 31 kHz
		75.0	450	µA	5.0V	
	PIC24F32KA3XX	8.1	28	µA	1.8V	
		13.50	150	µA	3.3V	

Legend: Unshaded rows represent PIC24F32KA3XX devices and shaded rows represent PIC24FV32KA3XX devices.

Note 1: Oscillator is in External Clock mode (FOSCSEL<2:0> = `010`, FOSC<1:0> = `00`).

TABLE 29-7: DC CHARACTERISTICS: IDLE CURRENT (IIDLE)

DC CHARACTERISTICS		Standard Operating Conditions: 1.8V to 3.6V PIC24F32KA3XX 2.0V to 5.5V PIC24FV32KA3XX Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Parameter No.	**Device**	**Typical**	**Max**	**Units**	**Conditions**	
Idle Current (IIDLE)						
DC40	PIC24FV32KA3XX	120	200	µA	2.0V	0.5 MIPS, FOSC = 1 MHz(1)
		160	430	µA	5.0V	
	PIC24F32KA3XX	50	100	µA	1.8V	
		90	370	µA	3.3V	
DC42	PIC24FV32KA3XX	165	—	µA	2.0V	1 MIPS, FOSC = 2 MHz(1)
		260	—	µA	5.0V	
	PIC24F32KA3XX	95	—	µA	1.8V	
		180	—	µA	3.3V	
DC44	PIC24FV32KA3XX	3.1	6.5	mA	5.0V	16 MIPS, FOSC = 32 MHz(1)
	PIC24F32KA3XX	2.9	6.0	mA	3.3V	
DC46	PIC24FV32KA3XX	0.65	—	mA	2.0V	FRC (4 MIPS), FOSC = 8 MHz
		1.0	—	mA	5.0V	
	PIC24F32KA3XX	0.55	—	mA	1.8V	
		1.0	—	mA	3.3V	
DC50	PIC24FV32KA3XX	60	200	µA	2.0V	LPRC (15.5 KIPS), FOSC = 31 kHz
		70	350	µA	5.0V	
	PIC24F32KA3XX	2.2	18	µA	1.8V	
		4.0	60	µA	3.3V	

Legend: Unshaded rows represent PIC24F32KA3XX devices and shaded rows represent PIC24FV32KA3XX devices.

Note 1: Oscillator is in External Clock mode (FOSCSEL<2:0> = `010`, FOSC<1:0> = `00`).

TABLE 29-8: DC CHARACTERISTICS: POWER-DOWN CURRENT (IPD)

DC CHARACTERISTICS		Standard Operating Conditions: 1.8V to 3.6V PIC24F32KA3XX 2.0V to 5.5V PIC24FV32KA3XX Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended					
Parameter No.	**Device**	**Typical[1]**	**Max**	**Units**	**Conditions**		
Power-Down Current (IPD)							
DC60	PIC24FV32KA3XX	6.0	—	µA	-40°C	2.0V	Sleep Mode[2]
			8.0		+25°C		
			8.5		+60°C		
			9.0		+85°C		
		—	15		+125°C		
		6.0	—	µA	-40°C	5.0V	
			8.0		+25°C		
			9.0		+60°C		
			10.0		+85°C		
		—	15		+125°C		
	PIC24F32KA3XX	0.025	—	µA	-40°C	1.8V	
			0.80		+25°C		
			1.5		+60°C		
			2.0		+85°C		
		—	7.5		+125°C		
		0.040	—	µA	-40°C	3.3V	
			1.0		+25°C		
			2.0		+60°C		
			3.0		+85°C		
		—	7.5		+125°C		
DC61	PIC24FV32KA3XX	0.25	—	µA	-40°C	2.0V	Low-Voltage Sleep Mode[2]
		0.35	3.0	µA	+85°C	5.0V	
		—	7.5	µA	+125°C	5.0V	
DC70	PIC24FV32KA3XX	0.03	—	µA	-40°C	2.0V	Deep Sleep Mode
		0.10	2.0	µA	+85°C	5.0V	
		—	6.0	µA	+125°C	5.0V	
	PIC24F32KA3XX	0.02	—	µA	-40°C	1.8V	
		0.08	1.2	µA	+85°C	3.3V	
		—	1.2	µA	+125°C	3.3V	

Legend: Unshaded rows represent PIC24F32KA3XX devices and shaded rows represent PIC24FV32KA3XX devices.

Note 1: Data in the Typical column is at 3.3V, +25°C (PIC24F32KA3XX) or 5.0V, +25°C (PIC24FV32KA3XX) unless otherwise stated. Parameters are for design guidance only and are not tested.

2: Base IPD is measured with all peripherals and clocks shut down. All I/Os are configured as outputs and set low, PMSLP is set to ‘0’ and WDT, etc., are all switched off.

3: The Δ current is the additional current consumed when the module is enabled. This current should be added to the base IPD current.

4: This current applies to Sleep only.

5: This current applies to Sleep and Deep Sleep.

6: This current applies to Deep Sleep only.

TABLE 29-8: DC CHARACTERISTICS: POWER-DOWN CURRENT (IPD) (CONTINUED)

DC CHARACTERISTICS		Standard Operating Conditions: 1.8V to 3.6V PIC24F32KA3XX 2.0V to 5.5V PIC24FV32KA3XX Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended					
Parameter No.	**Device**	**Typical(1)**	**Max**	**Units**	**Conditions**		
Module Differential Current (ΔIPD)(3)							
DC71	PIC24FV32KA3XX	0.50	—	μA	-40°C	2.0V	Watchdog Timer Current: ΔIWDT(4)
		0.70	1.5	μA	+85°C	5.0V	
		—	1.5	μA	+125°C	5.0V	
	PIC24F32KA3XX	0.50	—	μA	-40°C	1.8V	
		0.70	1.5	μA	+85°C	3.3V	
		—	1.5	μA	+125°C	3.3V	
DC72	PIC24FV32KA3XX	0.80	—	μA	-40°C	2.0V	32 kHz Crystal with RTCC, DSWDT or Timer1: ΔISOSC (SOSCSEL = 0)(5)
		1.50	2.0	μA	+85°C	5.0V	
		—	2.0	μA	+125°C	5.0V	
	PIC24F32KA3XX	0.70	—	μA	-40°C	1.8V	
		1.0	1.5	μA	+85°C	3.3V	
		—	1.5	μA	+125°C	3.3V	
DC75	PIC24FV32KA3XX	5.4	—	μA	-40°C	2.0V	ΔIHLVD(4)
		8.1	14.0	μA	+85°C	5.0V	
		—	14.0	μA	+125°C	5.0V	
	PIC24F32KA3XX	4.9	—	μA	-40°C	1.8V	
		7.5	14.0	μA	+85°C	3.3V	
		—	14.0	μA	+125°C	3.3V	
DC76	PIC24FV32KA3XX	5.6	—	μA	-40°C	2.0V	ΔIBOR(4)
		6.5	11.2	μA	-40°C	5.0V	
		—	11.2	μA	+125°C	5.0V	
	PIC24F32KA3XX	5.6	—	μA	-40°C	1.8V	
		6.0	11.2	μA	+85°C	3.3V	
		—	11.2	μA	+125°C	3.3V	

Legend: Unshaded rows represent PIC24F32KA3XX devices and shaded rows represent PIC24FV32KA3XX devices.

Note 1: Data in the Typical column is at 3.3V, +25°C (PIC24F32KA3XX) or 5.0V, +25°C (PIC24FV32KA3XX) unless otherwise stated. Parameters are for design guidance only and are not tested.

2: Base IPD is measured with all peripherals and clocks shut down. All I/Os are configured as outputs and set low, PMSLP is set to '0' and WDT, etc., are all switched off.

3: The Δ current is the additional current consumed when the module is enabled. This current should be added to the base IPD current.

4: This current applies to Sleep only.

5: This current applies to Sleep and Deep Sleep.

6: This current applies to Deep Sleep only.

TABLE 29-8: DC CHARACTERISTICS: POWER-DOWN CURRENT (IPD) (CONTINUED)

DC CHARACTERISTICS		Standard Operating Conditions: 1.8V to 3.6V PIC24F32KA3XX 2.0V to 5.5V PIC24FV32KA3XX Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended					
Parameter No.	**Device**	**Typical[1]**	**Max**	**Units**	**Conditions**		
Module Differential Current (ΔIPD)[3]							
DC78	PIC24FV32KA3XX	0.03	—	μA	-40°C	2.0V	Deep Sleep BOR: ΔILPBOR[5]
		0.05	0.20	μA	+85°C	5.0V	
		—	0.30	μA	+125°C	5.0V	
	PIC24F32KA3XX	0.03	—	μA	-40°C	1.8V	
		0.05	0.20	μA	+85°C	3.3V	
		—	0.30	μA	+125°C	3.3V	
DC80	PIC24FV32KA3XX	0.20	—	μA	-40°C	2.0V	Deep Sleep WDT: ΔIDSWDT (LPRC)[6]
		0.70	1.5	μA	+85°C	5.0V	
		—	1.5	μA	+125°C	5.0V	
	PIC24F32KA3XX	0.20	—	μA	-40°C	1.8V	
		0.35	0.8	μA	+85°C	3.3V	
		—	1.5	μA	+125°C	3.3V	

Legend: Unshaded rows represent PIC24F32KA3XX devices and shaded rows represent PIC24FV32KA3XX devices.

Note 1: Data in the Typical column is at 3.3V, +25°C (PIC24F32KA3XX) or 5.0V, +25°C (PIC24FV32KA3XX) unless otherwise stated. Parameters are for design guidance only and are not tested.

2: Base IPD is measured with all peripherals and clocks shut down. All I/Os are configured as outputs and set low, PMSLP is set to '0' and WDT, etc., are all switched off.

3: The Δ current is the additional current consumed when the module is enabled. This current should be added to the base IPD current.

4: This current applies to Sleep only.

5: This current applies to Sleep and Deep Sleep.

6: This current applies to Deep Sleep only.

TABLE 29-9: DC CHARACTERISTICS: I/O PIN INPUT SPECIFICATIONS

DC CHARACTERISTICS			Standard Operating Conditions: 1.8V to 3.6V PIC24F32KA3XX 2.0V to 5.5V PIC24FV32KA3XX Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	**Sym**	**Characteristic**	**Min**	**Typ[1]**	**Max**	**Units**	**Conditions**
	VIL	**Input Low Voltage[4]**					
DI10		I/O Pins	VSS	—	0.2 VDD	V	
DI15		$\overline{\text{MCLR}}$	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 I^2C Buffer	VSS	—	0.3 VDD	V	SMBus is disabled
DI19		I/O Pins with SMBus Buffer	VSS	—	0.8	V	SMBus is enabled
	VIH	**Input High Voltage[4]**					
DI20		I/O Pins: with Analog Functions Digital Only	 0.8 VDD 0.8 VDD	 — —	 VDD VDD	 V V	
DI25		$\overline{\text{MCLR}}$	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 I^2C Buffer: with Analog Functions Digital Only	 0.7 VDD 0.7 VDD	 — —	 VDD VDD	 V V	
DI29		I/O Pins with SMBus	2.1	—	VDD	V	2.5V ≤ VPIN ≤ VDD
DI30	ICNPU	**CNx Pull-up Current**	50	250	500	μA	VDD = 3.3V, VPIN = VSS
	IIL	**Input Leakage Current[2,3]**					
DI50		I/O Ports	—	0.05	0.1	μA	VSS ≤ VPIN ≤ VDD, Pin at high-impedance
DI55		$\overline{\text{MCLR}}$	—	—	0.1	μA	VSS ≤ VPIN ≤ VDD
DI56		OSCI	—	—	5	μA	VSS ≤ VPIN ≤ VDD, 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: The leakage current on the $\overline{\text{MCLR}}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

3: Negative current is defined as current sourced by the pin.

4: Refer to Table 1-3 for I/O pin buffer types.

TABLE 29-10: DC CHARACTERISTICS: I/O PIN OUTPUT SPECIFICATIONS

DC CHARACTERISTICS			Standard Operating Conditions: 1.8V to 3.6V PIC24F32KA3XX 2.0V to 5.5V PIC24FV32KA3XX Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended					
Param No.	**Sym**	**Characteristic**	**Min**	**Typ(1)**	**Max**	**Units**	**Conditions**	
	VOL	**Output Low Voltage**						
DO10		All I/O Pins	—	—	0.4	V	IOL = 8.0 mA	VDD = 4.5V
			—	—	0.4	V	IOL = 4.0 mA	VDD = 3.6V
			—	—	0.4	V	IOL = 3.5 mA	VDD = 2.0V
DO16		OSC2/CLKO	—	—	0.4	V	IOL = 2.0 mA	VDD = 4.5V
			—	—	0.4	V	IOL = 1.2 mA	VDD = 3.6V
			—	—	0.4	V	IOL = 0.4 mA	VDD = 2.0V
	VOH	**Output High Voltage**						
DO20		All I/O Pins	3.8	—	—	V	IOH = -3.5 mA	VDD = 4.5V
			3	—	—	V	IOH = -3.0 mA	VDD = 3.6V
			1.6	—	—	V	IOH = -1.0 mA	VDD = 2.0V
DO26		OSC2/CLKO	3.8	—	—	V	IOH = -2.0 mA	VDD = 4.5V
			3	—	—	V	IOH = -1.0 mA	VDD = 3.6V
			1.6	—	—	V	IOH = -0.5 mA	VDD = 2.0V

Note 1: Data in “Typ” column is at +25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

TABLE 29-11: DC CHARACTERISTICS: PROGRAM MEMORY

DC CHARACTERISTICS			Standard Operating Conditions: 1.8V to 3.6V PIC24F32KA3XX 2.0V to 5.5V PIC24FV32KA3XX Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	**Sym**	**Characteristic**	**Min**	**Typ(1)**	**Max**	**Units**	**Conditions**
		Program Flash Memory					
D130	EP	Cell Endurance	10,000(2)	—	—	E/W	
D131	VPR	VDD for Read	VMIN	—	3.6	V	VMIN = Minimum operating voltage
D133A	TIW	Self-Timed Write Cycle Time	—	2	—	ms	
D134	TRETD	Characteristic Retention	40	—	—	Year	Provided no other specifications are violated
D135	IDDP	Supply Current During Programming	—	10	—	mA	

Note 1: Data in “Typ” column is at 3.3V, +25°C unless otherwise stated.
2: Self-write and block erase.

TABLE 29-12: DC CHARACTERISTICS: DATA EEPROM MEMORY

DC CHARACTERISTICS			Standard Operating Conditions: 1.8V to 3.6V PIC24F32KA3XX 2.0V to 5.5V PIC24FV32KA3XX Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	**Sym**	**Characteristic**	**Min**	**Typ[1]**	**Max**	**Units**	**Conditions**
		Data EEPROM Memory					
D140	EPD	Cell Endurance	100,000	—	—	E/W	
D141	VPRD	VDD for Read	VMIN	—	3.6	V	VMIN = Minimum operating voltage
D143A	TIWD	Self-Timed Write Cycle Time	—	4	—	ms	
D143B	TREF	Number of Total Write/Erase Cycles Before Refresh	—	10M	—	E/W	
D144	TRETDD	Characteristic Retention	40	—	—	Year	Provided no other specifications are violated
D145	IDDPD	Supply Current During Programming	—	7	—	mA	

Note 1: Data in "Typ" column is at 3.3V, +25°C unless otherwise stated.

TABLE 29-13: DC CHARACTERISTICS: COMPARATOR SPECIFICATIONS

Operating Conditions: 2.0V < VDD < 3.6V, -40°C < TA < +125°C (unless otherwise stated)							
Param No.	**Symbol**	**Characteristic**	**Min**	**Typ**	**Max**	**Units**	**Comments**
D300	VIOFF	Input Offset Voltage	—	20	40	mV	
D301	VICM	Input Common-Mode Voltage	0	—	VDD	V	
D302	CMRR	Common-Mode Rejection Ratio	55	—	—	dB	

TABLE 29-14: DC CHARACTERISTICS: COMPARATOR VOLTAGE REFERENCE SPECIFICATIONS

Operating Conditions: 2.0V < VDD < 3.6V, -40°C < TA < +125°C (unless otherwise stated)							
Param No.	**Symbol**	**Characteristic**	**Min**	**Typ**	**Max**	**Units**	**Comments**
VRD310	CVRES	Resolution	—	—	VDD/32	LSb	
VRD311	CVRAA	Absolute Accuracy	—	—	AVDD – 1.5	LSb	
VRD312	CVRUR	Unit Resistor Value (R)	—	2k	—	Ω	

TABLE 29-15: INTERNAL VOLTAGE REGULATOR SPECIFICATIONS

Operating Conditions: -40°C < TA < +125°C (unless otherwise stated)							
Param No.	**Symbol**	**Characteristics**	**Min**	**Typ**	**Max**	**Units**	**Comments**
DVR10	VBG	Band Gap Reference Voltage	0.973	1.024	1.075	V	
DVR11	TBG	Band Gap Reference Start-up Time	—	1	—	ms	
DVR20	VRGOUT	Regulator Output Voltage	3.1	3.3	3.6	V	-40°C < TA < +85°C
			3.0	3.19	3.6	V	-40°C < TA < +125°C
DVR21	CEFC	External Filter Capacitor Value	4.7	10	—	μF	Series resistance < 3 Ohm recommended; < 5 Ohm is required.
DVR30	VLVR	Retention Regulator Output Voltage	—	2.6	—	V	

TABLE 29-16: CTMU CURRENT SOURCE SPECIFICATIONS

DC CHARACTERISTICS			Standard Operating Conditions: 1.8V to 3.6V PIC24F32KA3XX 2.0V to 5.5V PIC24FV32KA3XX Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended					
Param No.	**Sym**	**Characteristic**	**Min**	**Typ[1]**	**Max**	**Units**	**Comments**	**Conditions**
DCT10	IOUT1	CTMU Current Source, Base Range	—	550	—	nA	CTMUICON<9:8> = 01	2.5V < VDD < VDDMAX
DCT11	IOUT2	CTMU Current Source, 10x Range	—	5.5	—	μA	CTMUICON<9:8> = 10	
DCT12	IOUT3	CTMU Current Source, 100x Range	—	55	—	μA	CTMUICON<9:8> = 11	
DCT13	IOUT4	CTMU Current Source, 1000x Range	—	550	—	μA	CTMUICON<9:8> = 00 (Note 2)	
DCT20	VF	Temperature Diode Forward Voltage	—	.76	—	V		
DCT21	VΔ	Voltage Change per Degree Celsius	—	1.6	—	mV/°C		

Note 1: Nominal value at the center point of the current trim range (CTMUICON<7:2> = 000000). On PIC24F32KA parts, the current output is limited to the typical current value when IOUT4 is chosen.

2: Do not use this current range with a temperature sensing diode.

29.2 AC Characteristics and Timing Parameters

The information contained in this section defines the PIC24FV32KA304 family AC characteristics and timing parameters.

TABLE 29-17: TEMPERATURE AND VOLTAGE SPECIFICATIONS – AC

AC CHARACTERISTICS	**Standard Operating Conditions: 1.8V to 3.6V PIC24F32KA3XX** **2.0V to 5.5V PIC24FV32KA3XX** **Operating temperature: -40°C ≤ TA ≤ +85°C for Industrial** **-40°C ≤ TA ≤ +125°C for Extended** Operating voltage VDD range as described in **Section 29.1 "DC Characteristics"**.

FIGURE 29-3: LOAD CONDITIONS FOR DEVICE TIMING SPECIFICATIONS

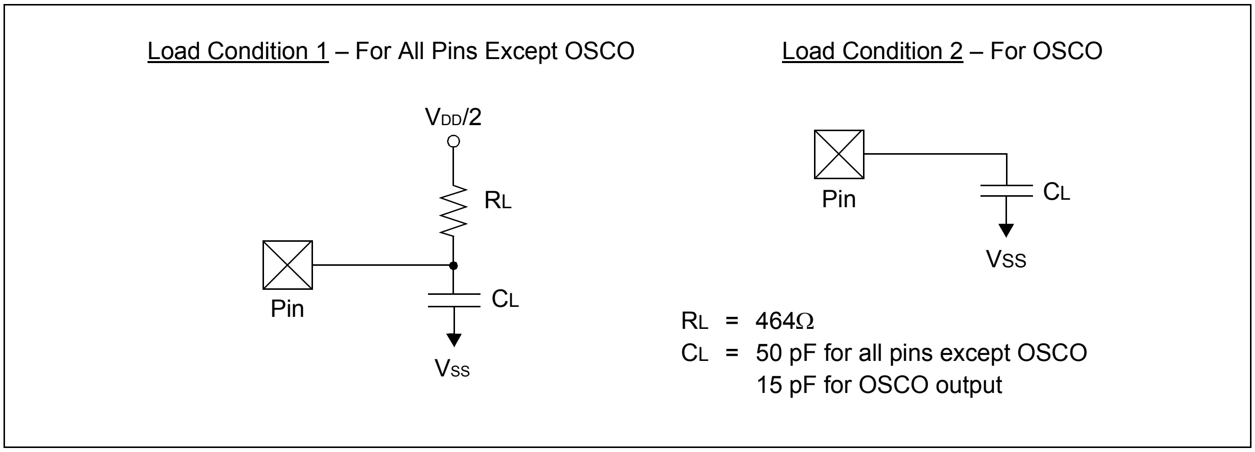


TABLE 29-18: CAPACITIVE LOADING REQUIREMENTS ON OUTPUT PINS

Param No.	Symbol	Characteristic	Min	Typ[1]	Max	Units	Conditions
DO50	COSC2	OSCO/CLKO Pin	—	—	15	pF	In XT and HS modes when the 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 29-4: EXTERNAL CLOCK TIMING

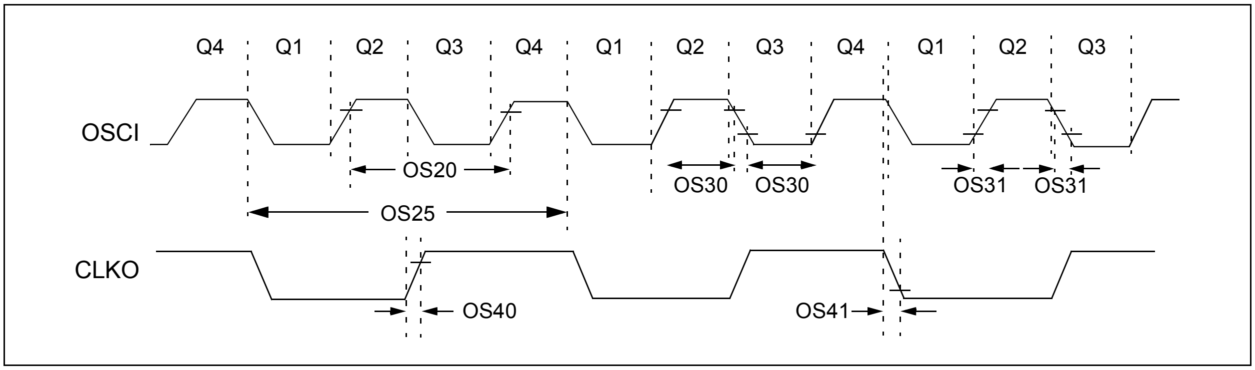


TABLE 29-19: EXTERNAL CLOCK TIMING REQUIREMENTS

AC CHARACTERISTICS			Standard Operating Conditions: 1.8V to 3.6V PIC24F32KA3XX 2.0V to 5.5V PIC24FV32KA3XX Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	**Sym**	**Characteristic**	**Min**	**Typ[(1)]**	**Max**	**Units**	**Conditions**
OS10	FOSC	External CLKI Frequency (External clocks allowed only in EC mode)	DC 4	— —	32 8	MHz MHz	EC ECPLL
OS15		Oscillator Frequency	0.2 4 4 31	— — — —	4 25 8 33	MHz MHz MHz kHz	XT HS XTPLL 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: The 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 29-20: PLL CLOCK TIMING SPECIFICATIONS

AC CHARACTERISTICS			Standard Operating Conditions: 1.8V to 3.6V PIC24F32KA3XX 2.0V to 5.5V PIC24FV32KA3XX Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	**Sym**	**Characteristic(1)**	**Min**	**Typ(2)**	**Max**	**Units**	**Conditions**
OS50	FPLLI	PLL Input Frequency Range	4	—	8	MHz	ECPLL, HSPLL modes, -40°C ≤ TA ≤ +85°C
OS51	FSYS	PLL Output Frequency Range	16	—	32	MHz	-40°C ≤ TA ≤ +85°C
OS52	TLOCK	PLL Start-up Time (Lock Time)	—	1	2	ms	
OS53	DCLK	CLKO Stability (Jitter)	-2	1	2	%	Measured over a 100 ms period

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 29-21: AC CHARACTERISTICS: INTERNAL RC ACCURACY

AC CHARACTERISTICS		Standard Operating Conditions: 1.8V to 3.6V PIC24F32KA3XX 2.0V to 5.5V PIC24FV32KA3XX Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended					
Param No.	**Characteristic**	**Min**	**Typ**	**Max**	**Units**	**Conditions**	
F20	**Internal FRC Accuracy @ 8 MHz(1)**						
	FRC	-2	—	+2	%	+25°C	3.0V ≤ VDD ≤ 3.6V, F device 3.2V ≤ VDD ≤ 5.5V, FV device
		-6	—	+6	%	-40°C ≤ TA ≤ +85°C	1.8V ≤ VDD ≤ 3.6V, F device 2.0V ≤ VDD ≤ 5.5V, FV device
F21	**LPRC @ 31 kHz(2)**						
		-15	—	15	%		

Note 1: Frequency is calibrated at +25°C and 3.3V. The OSCTUN bits can be used to compensate for temperature drift.

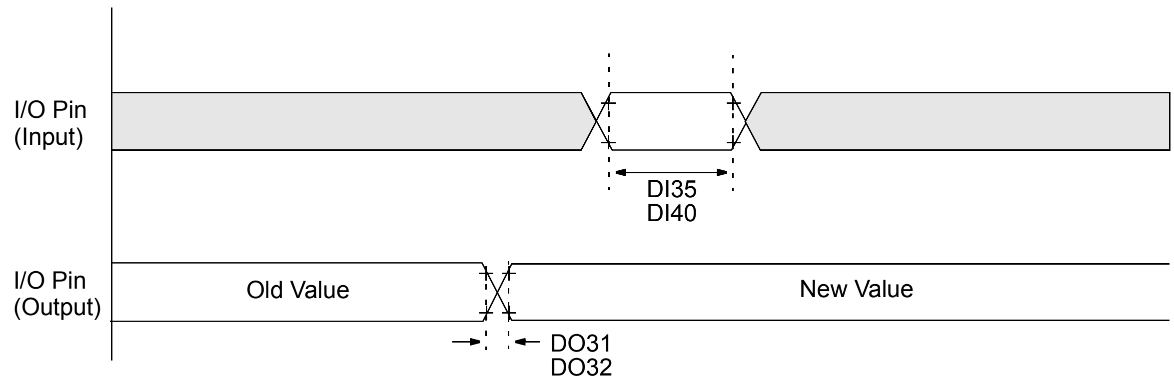
2: The change of LPRC frequency as VDD changes.

TABLE 29-22: INTERNAL RC OSCILLATOR SPECIFICATIONS

AC CHARACTERISTICS			Standard Operating Conditions: 1.8V to 3.6V PIC24F32KA3XX 2.0V to 5.5V PIC24FV32KA3XX Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	**Sym**	**Characteristic(1)**	**Min**	**Typ**	**Max**	**Units**	**Conditions**
	TFRC	FRC Start-up Time	—	5	—	μs	
	TLPRC	LPRC Start-up Time	—	70	—	μs	

Note 1: These parameters are characterized but not tested in manufacturing.

FIGURE 29-5: CLKO AND I/O TIMING CHARACTERISTICS



Note: Refer to Figure 29-3 for load conditions.

TABLE 29-23: CLKO AND I/O TIMING REQUIREMENTS

AC CHARACTERISTICS			Standard Operating Conditions: 1.8V to 3.6V PIC24F32KA3XX 2.0V to 5.5V PIC24FV32KA3XX Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	**Sym**	**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 (output)	20	—	—	ns	
DI40	TRBP	CNx High or Low Time (input)	2	—	—	TCY	

Note 1: Data in “Typ” column is at 3.3V, +25°C (PIC24F32KA3XX); 5.0V, +25°C (PIC24FV32KA3XX), unless otherwise stated.

TABLE 29-24: COMPARATOR TIMINGS

Param No.	Symbol	Characteristic	Min	Typ	Max	Units	Comments
300	TRESP	Response Time*(1)	—	150	400	ns	
301	TMC2OV	Comparator Mode Change to Output Valid*	—	—	10	µs	

* Parameters are characterized but not tested.

Note 1: Response time is measured with one comparator input at (VDD – 1.5)/2, while the other input transitions from VSS to VDD.

TABLE 29-25: COMPARATOR VOLTAGE REFERENCE SETTLING TIME SPECIFICATIONS

Param No.	Symbol	Characteristic	Min	Typ	Max	Units	Comments
VR310	TSET	Settling Time(1)	—	—	10	µs	

Note 1: Settling time is measured while CVRSS = `1` and the CVR<3:0> bits transition from ‘`0000`’ to ‘`1111`’.

FIGURE 29-6: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING CHARACTERISTICS

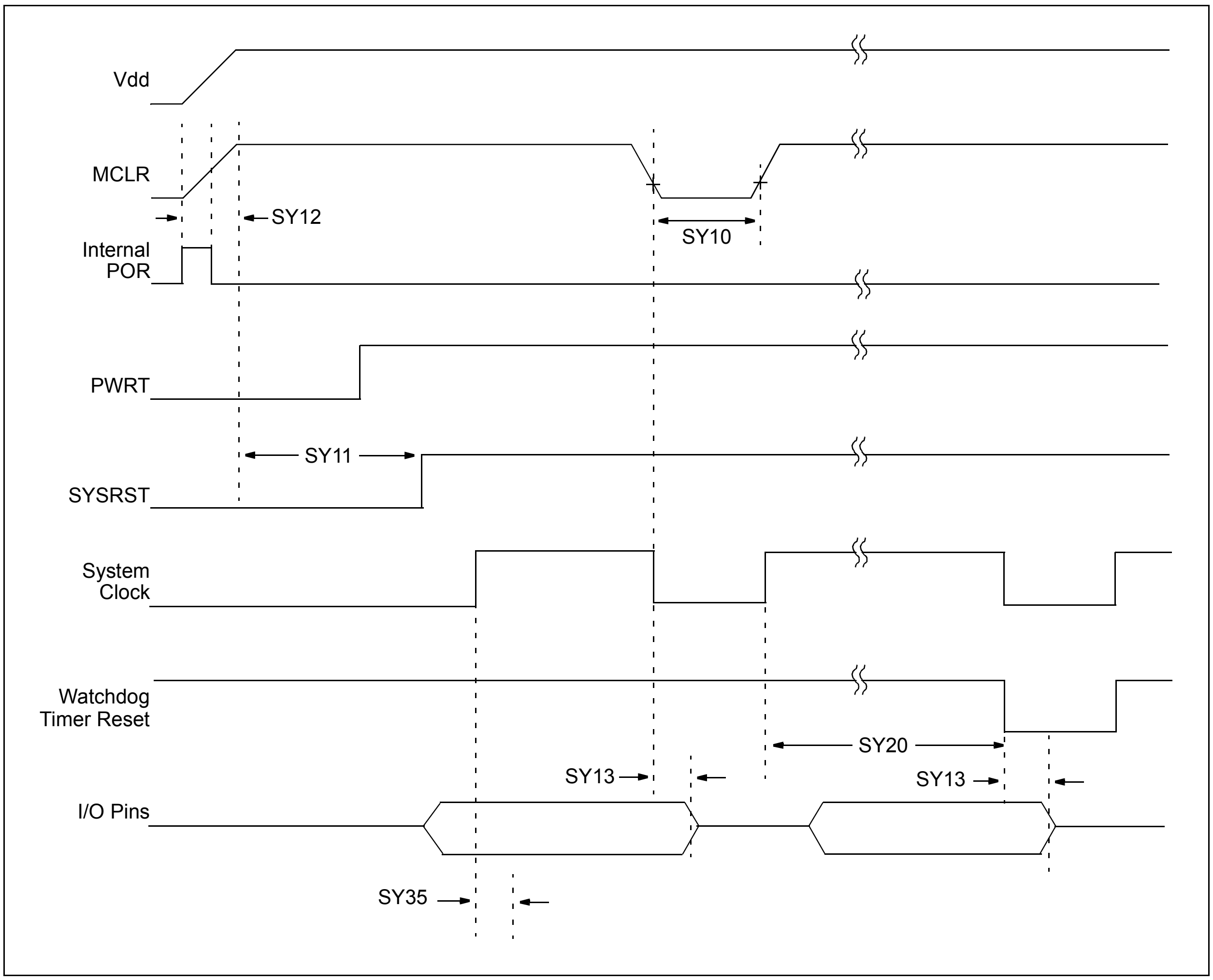


FIGURE 29-7: BROWN-OUT RESET CHARACTERISTICS

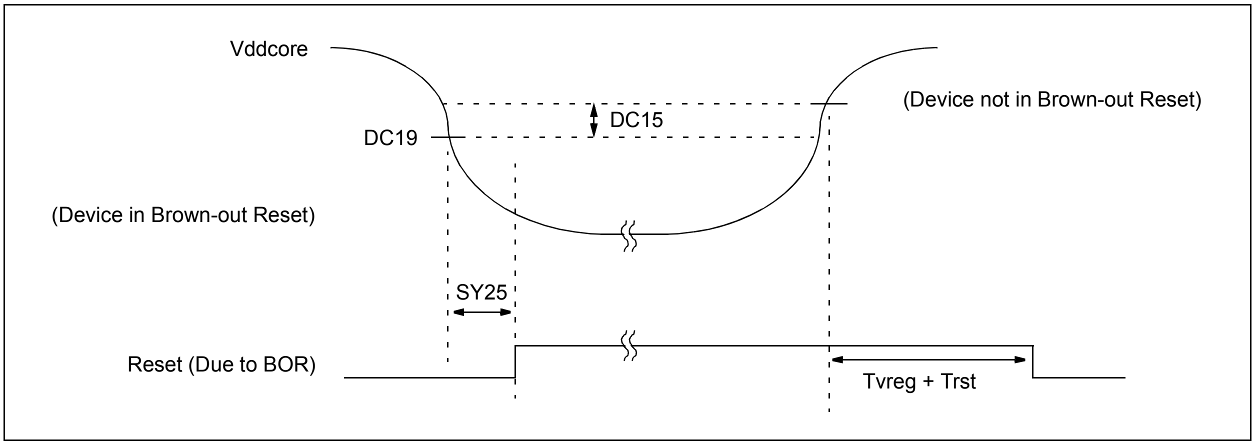


TABLE 29-26: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER, AND BROWN-OUT RESET TIMING REQUIREMENTS

AC CHARACTERISTICS			Standard Operating Conditions: 1.8V to 3.6V PIC24F32KA3XX 2.0V to 5.5V PIC24FV32KA3XX Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	**Symbol**	**Characteristic**	**Min.**	**Typ(1)**	**Max.**	**Units**	**Conditions**
SY10	TmcL	MCLR Pulse Width (low)	2	—	—	µs	
SY11	TPWRT	Power-up Timer Period	50	64	90	ms	
SY12	TPOR	Power-on Reset Delay	1	5	10	µs	
SY13	TIOZ	I/O High-Impedance from MCLR Low or Watchdog Timer Reset	—	—	100	ns	
SY20	TWDT	Watchdog Timer Time-out Period	0.85	1.0	1.15	ms	1.32 prescaler
			3.4	4.0	4.6	ms	1:128 prescaler
SY25	TBOR	Brown-out Reset Pulse Width	1	—	—	µs	
SY35	TFSCM	Fail-Safe Clock Monitor Delay	—	2.0	2.3	µs	
SY45	TRST	Internal State Reset Time	—	5	—	µs	
SY50	TVREG	On-Chip Voltage Regulator Output Delay	—	10	—	µs	(Note 2)
SY55	TLOCK	PLL Start-up Time	—	100	—	µs	
SY65	TOST	Oscillator Start-up Time	—	1024	—	TOSC	
SY70	TDSWU	Wake-up from Deep Sleep Time	—	100	—	µs	Based on full discharge of 10 µF capacitor on VCAP; includes TPOR and TRST
SY71	TPM	Program Memory Wake-up Time	—	1	—	µs	Sleep wake-up with PMSLP = 0
SY72	TLVR	Retention Regulator Wake-up Time	—	250	—	µs	
SY73	THVLD	HVLD Interrupt Response Time	—	2	—	µs	

Note 1: Data in “Typ” column is at 3.3V, +25°C unless otherwise stated.
2: This applies to PIC24FV32KA3XX devices only.

FIGURE 29-8: TIMER1/2/3/4/5 EXTERNAL CLOCK INPUT TIMING

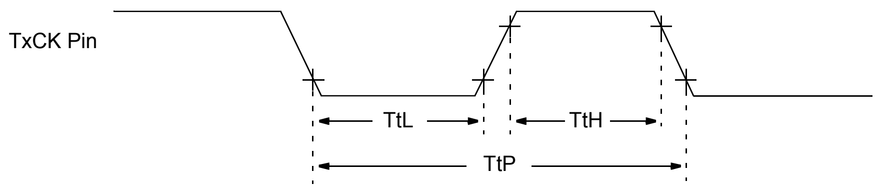


TABLE 29-27: TIMER1/2/3/4/5 EXTERNAL CLOCK INPUT REQUIREMENTS

Param. No.	Symbol	Characteristic		Min	Max	Units	Conditions
	TtH	TxCK High Pulse Time	Sync w/Prescaler	TCY + 20	—	ns	Must also meet Parameter Ttp
			Async w/Prescaler	10	—	ns	
			Async Counter	20	—	ns	
	TtL	TxCK Low Pulse Time	Sync w/Prescaler	TCY + 20	—	ns	Must also meet Parameter Ttp
			Async w/Prescaler	10	—	ns	
			Async Counter	20	—	ns	
	TtP	TxCK External Input Period	Sync w/Prescaler	2 * TCY + 40	—	ns	N = Prescale Value (1, 4, 8, 16)
			Async w/Prescaler	Greater of: 20 or (2 * TCY + 40)/N	—	ns	
			Async Counter	40	—	ns	
		Delay for Input Edge to Timer Increment	Synchronous	1	2	TCY	
			Asynchronous	—	20	ns	

FIGURE 29-9: INPUT CAPTURE x TIMINGS

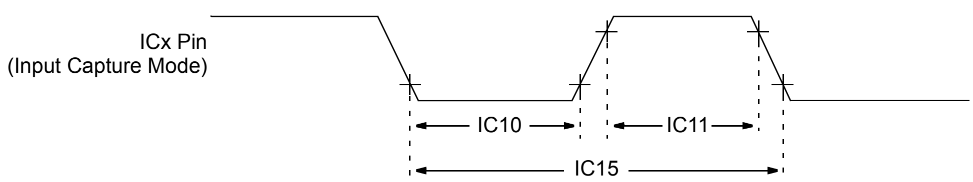


TABLE 29-28: INPUT CAPTURE x REQUIREMENTS

Param. No.	Symbol	Characteristic		Min	Max	Units	Conditions
IC10	TccL	ICx Input Low Time – Synchronous Timer	No Prescaler	TCY + 20	—	ns	Must also meet Parameter IC15
			With Prescaler	20	—	ns	
IC11	TccH	ICx Input Low Time – Synchronous Timer	No Prescaler	TCY + 20	—	ns	Must also meet Parameter IC15
			With Prescaler	20	—	ns	
IC15	TccP	ICx Input Period – Synchronous Timer		(2 * TCY/N) + 40	—	ns	N = prescale value (1, 4, 16)

FIGURE 29-10: OUTPUT COMPARE x TIMINGS

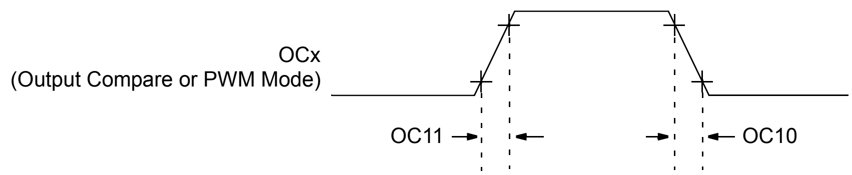


TABLE 29-29: OUTPUT CAPTURE REQUIREMENTS

Param. No.	Symbol	Characteristic	Min	Max	Units	Conditions
OC11	TCCR	OC1 Output Rise Time	—	10	ns	
			—	—	ns	
OC10	TCCF	OC1 Output Fall Time	—	10	ns	
			—	—	ns	

FIGURE 29-11: PWM MODULE TIMING REQUIREMENTS

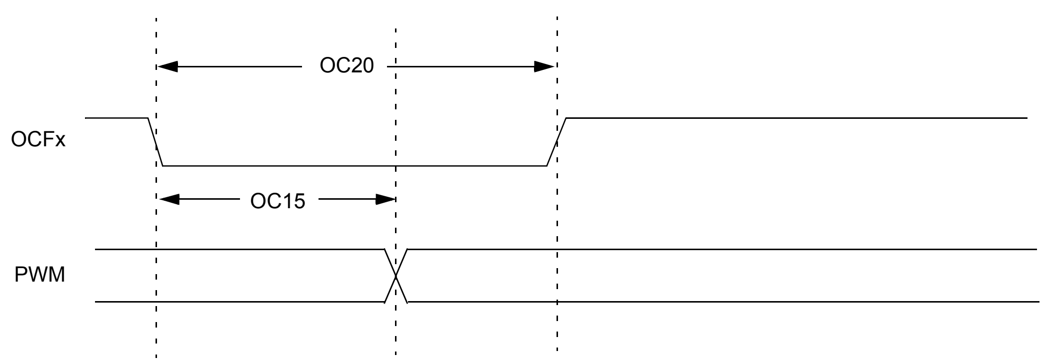
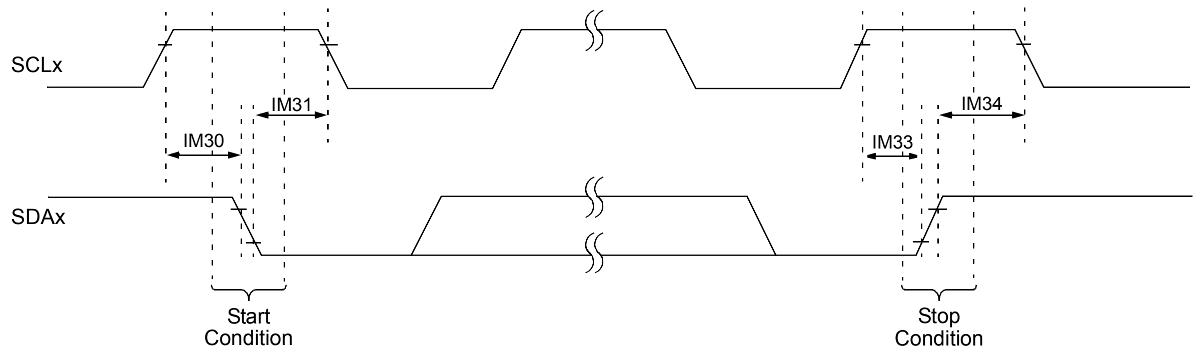


TABLE 29-30: PWM TIMING REQUIREMENTS

Param. No.	Symbol	Characteristic	Min	Typ†	Max	Units	Conditions
OC15	TFD	Fault Input to PWM I/O Change	—	—	25	ns	VDD = 3.0V, -40°C to +125°C
OC20	TFH	Fault Input Pulse Width	50	—	—	ns	VDD = 3.0V, -40°C to +125°C

† Data in “Typ” column is at 5V, +25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 29-12: I²C BUS START/STOP BITS TIMING CHARACTERISTICS (MASTER MODE)



Note: Refer to Figure 29-3 for load conditions.

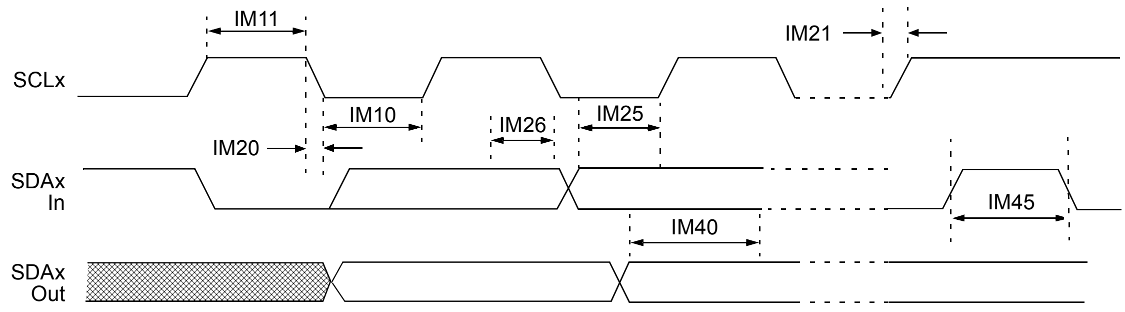
TABLE 29-31: I²C BUS START/STOP BIT TIMING REQUIREMENTS (MASTER MODE)

AC CHARACTERISTICS				Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial) -40°C ≤ TA ≤ +125°C for Extended			
Param No.	Symbol	Characteristic		Min(1)	Max	Units	Conditions
IM30	TSU:STA	Start Condition Setup Time	100 kHz mode	TCY/2 (BRG + 1)	—	μs	Only relevant for Repeated Start condition
			400 kHz mode	TCY/2 (BRG + 1)	—	μs	
			1 MHz mode(2)	TCY/2 (BRG + 1)	—	μs	
IM31	THD:STA	Start Condition Hold Time	100 kHz mode	TCY/2 (BRG + 1)	—	μs	After this period, the first clock pulse is generated
			400 kHz mode	TCY/2 (BRG + 1)	—	μs	
			1 MHz mode(2)	TCY/2 (BRG + 1)	—	μs	
IM33	TSU:STO	Stop Condition Setup Time	100 kHz mode	TCY/2 (BRG + 1)	—	μs	
			400 kHz mode	TCY/2 (BRG + 1)	—	μs	
			1 MHz mode(2)	TCY/2 (BRG + 1)	—	μs	
IM34	THD:STO	Stop Condition Hold Time	100 kHz mode	TCY/2 (BRG + 1)	—	ns	
			400 kHz mode	TCY/2 (BRG + 1)	—	ns	
			1 MHz mode(2)	TCY/2 (BRG + 1)	—	ns	

Note 1: BRG is the value of the I²C Baud Rate Generator. Refer to **Section 17.3 "Setting Baud Rate When Operating as a Bus Master"** for details.

2: Maximum pin capacitance = 10 pF for all I²C pins (for 1 MHz mode only).

FIGURE 29-13: I²C BUS DATA TIMING CHARACTERISTICS (MASTER MODE)



Note: Refer to Figure 29-3 for load conditions.

TABLE 29-32: I²C BUS DATA TIMING REQUIREMENTS (MASTER MODE)

AC CHARACTERISTICS				Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended			
Param No.	**Symbol**	**Characteristic**		**Min[1]**	**Max**	**Units**	**Conditions**
IM10	TLO:SCL	Clock Low Time	100 kHz mode	TCY/2 (BRG + 1)	—	μs	
			400 kHz mode	TCY/2 (BRG + 1)	—	μs	
			1 MHz mode[2]	TCY/2 (BRG + 1)	—	μs	
IM11	THI:SCL	Clock High Time	100 kHz mode	TCY/2 (BRG + 1)	—	μs	
			400 kHz mode	TCY/2 (BRG + 1)	—	μs	
			1 MHz mode[2]	TCY/2 (BRG + 1)	—	μs	
IM20	TF:SCL	SDAx and SCLx Fall Time	100 kHz mode	—	300	ns	CB is specified to be from 10 to 400 pF
			400 kHz mode	20 + 0.1 CB	300	ns	
			1 MHz mode[2]	—	100	ns	
IM21	TR:SCL	SDAx and SCLx Rise Time	100 kHz mode	—	1000	ns	CB is specified to be from 10 to 400 pF
			400 kHz mode	20 + 0.1 CB	300	ns	
			1 MHz mode[2]	—	300	ns	
IM25	TSU:DAT	Data Input Setup Time	100 kHz mode	250	—	ns	
			400 kHz mode	100	—	ns	
			1 MHz mode[2]	100	—	ns	
IM26	THD:DAT	Data Input Hold Time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	μs	
			1 MHz mode[2]	0	—	ns	
IM40	TAA:SCL	Output Valid from Clock	100 kHz mode	—	3500	ns	
			400 kHz mode	—	1000	ns	
			1 MHz mode[2]	—	—	ns	
IM45	TBF:SDA	Bus Free Time	100 kHz mode	4.7	—	μs	Time the bus must be free before a new transmission can start
			400 kHz mode	1.3	—	μs	
			1 MHz mode[2]	0.5	—	μs	
IM50	CB	Bus Capacitive Loading		—	400	pF	

Note 1: BRG is the value of the I²C Baud Rate Generator. Refer to **Section 17.3 "Setting Baud Rate When Operating as a Bus Master"** for details.

2: Maximum pin capacitance = 10 pF for all I²C pins (for 1 MHz mode only).

FIGURE 29-14: I²C BUS DATA TIMING CHARACTERISTICS (SLAVE MODE)

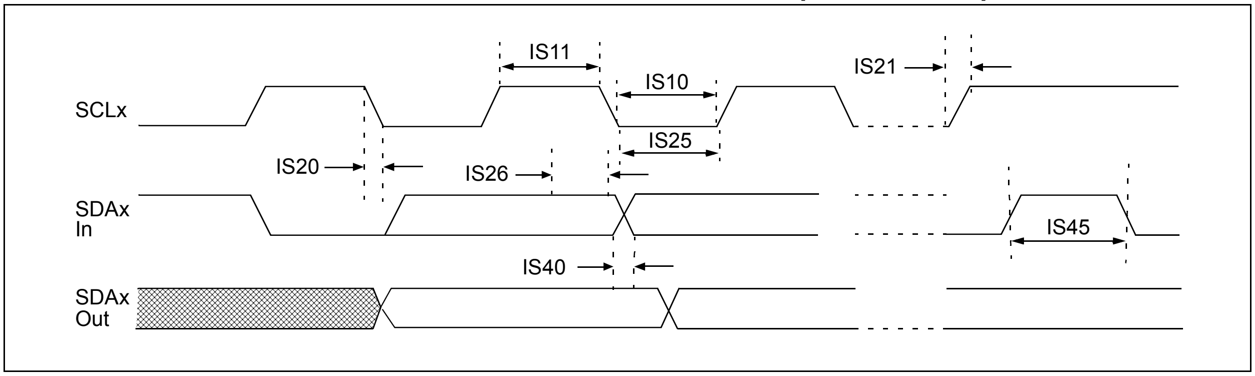


TABLE 29-33: I²C BUS DATA TIMING REQUIREMENTS (SLAVE MODE)

AC CHARACTERISTICS				Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C (Industrial) -40°C ≤ TA ≤ +125°C (Extended)			
Param No.	**Symbol**	**Characteristic**		**Min**	**Max**	**Units**	**Conditions**
IS10	TLO:SCL	Clock Low Time	100 kHz mode	4.7	—	μs	Device must operate at a minimum of 1.5 MHz
			400 kHz mode	1.3	—	μs	Device must operate at a minimum of 10 MHz
			1 MHz mode(1)	0.5	—	μs	
IS11	THI:SCL	Clock High Time	100 kHz mode	4.0	—	μs	Device must operate at a minimum of 1.5 MHz
			400 kHz mode	0.6	—	μs	Device must operate at a minimum of 10 MHz
			1 MHz mode(1)	0.5	—	μs	
IS20	TF:SCL	SDAx and SCLx Fall Time	100 kHz mode	—	300	ns	CB is specified to be from 10 to 400 pF
			400 kHz mode	20 + 0.1 CB	300	ns	
			1 MHz mode(1)	—	100	ns	
IS21	TR:SCL	SDAx and SCLx Rise Time	100 kHz mode	—	1000	ns	CB is specified to be from 10 to 400 pF
			400 kHz mode	20 + 0.1 CB	300	ns	
			1 MHz mode(1)	—	300	ns	
IS25	TSU:DAT	Data Input Setup Time	100 kHz mode	250	—	ns	
			400 kHz mode	100	—	ns	
			1 MHz mode(1)	100	—	ns	
IS26	THD:DAT	Data Input Hold Time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	μs	
			1 MHz mode(1)	0	0.3	μs	
IS40	TAA:SCL	Output Valid From Clock	100 kHz mode	0	3500	ns	
			400 kHz mode	0	1000	ns	
			1 MHz mode(1)	0	350	ns	
IS45	TBF:SDA	Bus Free Time	100 kHz mode	4.7	—	μs	Time the bus must be free before a new transmission can start
			400 kHz mode	1.3	—	μs	
			1 MHz mode(1)	0.5	—	μs	
IS50	CB	Bus Capacitive Loading		—	400	pF	

Note 1: Maximum pin capacitance = 10 pF for all I²C pins (for 1 MHz mode only).

FIGURE 29-15: I²C BUS START/STOP BITS TIMING CHARACTERISTICS (SLAVE MODE)

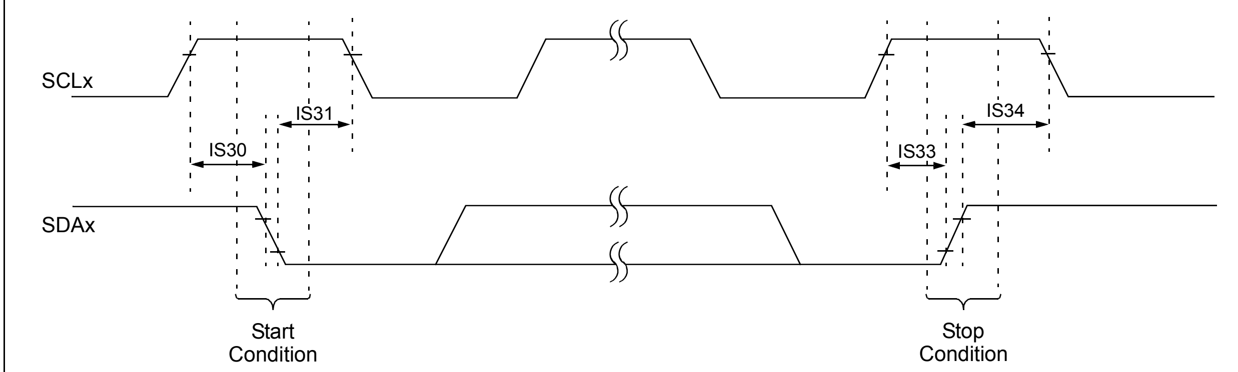


TABLE 29-34: I²C BUS START/STOP BITS TIMING REQUIREMENTS (SLAVE MODE)

AC CHARACTERISTICS				**Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C (Industrial) -40°C ≤ TA ≤ +125°C (Extended)			
Param No.	**Symbol**	**Characteristic**		**Min**	**Max**	**Units**	**Conditions**
IS30	TSU:STA	Start Condition Setup Time	100 kHz mode	4.7	—	μs	Only relevant for Repeated Start condition
			400 kHz mode	0.6	—	μs	
			1 MHz mode(1)	0.25	—	μs	
IS31	THD:STA	Start Condition Hold Time	100 kHz mode	4.0	—	μs	After this period, the first clock pulse is generated
			400 kHz mode	0.6	—	μs	
			1 MHz mode(1)	0.25	—	μs	
IS33	TSU:STO	Stop Condition Setup Time	100 kHz mode	4.7	—	μs	
			400 kHz mode	0.6	—	μs	
			1 MHz mode(1)	0.6	—	μs	
IS34	THD:STO	Stop Condition Hold Time	100 kHz mode	4000	—	ns	
			400 kHz mode	600	—	ns	
			1 MHz mode(1)	250	—	ns	

Note 1: Maximum pin capacitance = 10 pF for all I²C pins (for 1 MHz mode only).

FIGURE 29-16: UARTx BAUD RATE GENERATOR OUTPUT TIMING

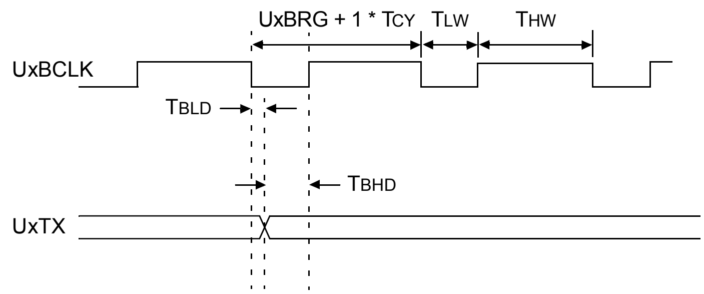


FIGURE 29-17: UARTx START BIT EDGE DETECTION

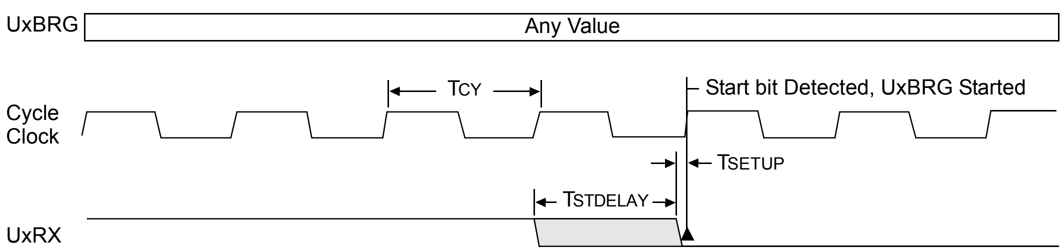


TABLE 29-35: UARTx TIMING REQUIREMENTS

AC CHARACTERISTICS		**Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C (Industrial) -40°C ≤ TA ≤ +125°C (Extended)			
Symbol	**Characteristics**	**Min**	**Typ**	**Max**	**Units**
TLW	UxBCLK High Time	20	TCY/2	—	ns
THW	UxBCLK Low Time	20	(TCY * UxBRG) + TCY/2	—	ns
TBLD	UxBCLK Falling Edge Delay from UxTX	-50	—	50	ns
TBHD	UxBCLK Rising Edge Delay from UxTX	TCY/2 – 50	—	TCY/2 + 50	ns
TWAK	Minimum Low on UxRX Line to Cause Wake-up	—	1	—	μs
TCTS	Minimum Low on $\overline{\text{UxCTS}}$ Line to Start Transmission	Tcy	—	—	ns
TSETUP	Start bit Falling Edge to System Clock Rising Edge Setup Time	3	—	—	ns
TSTDELAY	Maximum Delay in the Detection of the Start bit Falling Edge	—	—	TCY + TSETUP	ns

FIGURE 29-18: SPIx MODULE MASTER MODE TIMING CHARACTERISTICS (CKE = 0)

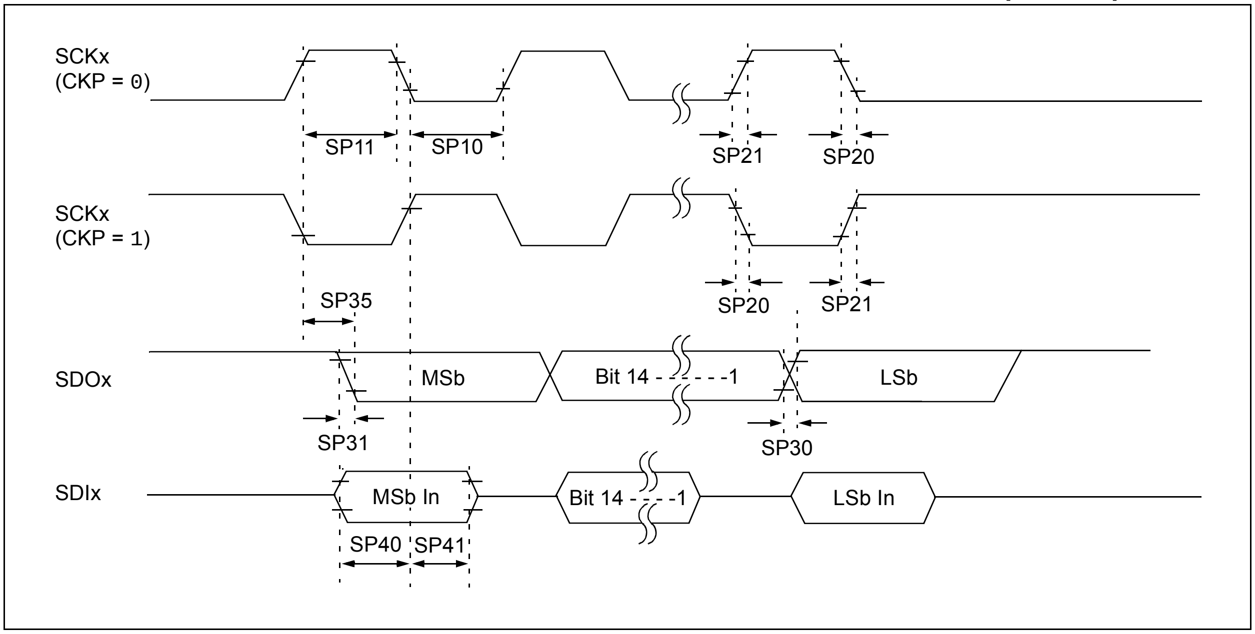


TABLE 29-36: SPIx MASTER MODE TIMING REQUIREMENTS (CKE = 0)

AC CHARACTERISTICS			Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C (Industrial) -40°C ≤ TA ≤ +125°C (Extended)				
Param No.	**Symbol**	**Characteristic**	**Min**	**Typ[(1)]**	**Max**	**Units**	**Conditions**
SP10	TscL	SCKx Output Low Time[(2)]	TCY/2	—	—	ns	
SP11	TscH	SCKx Output High Time[(2)]	TCY/2	—	—	ns	
SP20	TscF	SCKx Output Fall Time[(3)]	—	10	25	ns	
SP21	TscR	SCKx Output Rise Time[(3)]	—	10	25	ns	
SP30	TdoF	SDOx Data Output Fall Time[(3)]	—	10	25	ns	
SP31	TdoR	SDOx Data Output Rise Time[(3)]	—	10	25	ns	
SP35	TscH2doV, TscL2doV	SDOx Data Output Valid after SCKx Edge	—	—	30	ns	
SP40	TdiV2scH, TdiV2scL	Setup Time of SDIx Data Input to SCKx Edge	20	—	—	ns	
SP41	TscH2diL, TscL2diL	Hold Time of SDIx Data Input to SCKx Edge	20	—	—	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: The minimum clock period for SCKx is 100 ns; therefore, the clock generated in Master mode must not violate this specification.

3: This assumes a 50 pF load on all SPIx pins.

FIGURE 29-19: SPIx MODULE MASTER MODE TIMING CHARACTERISTICS (CKE = 1)

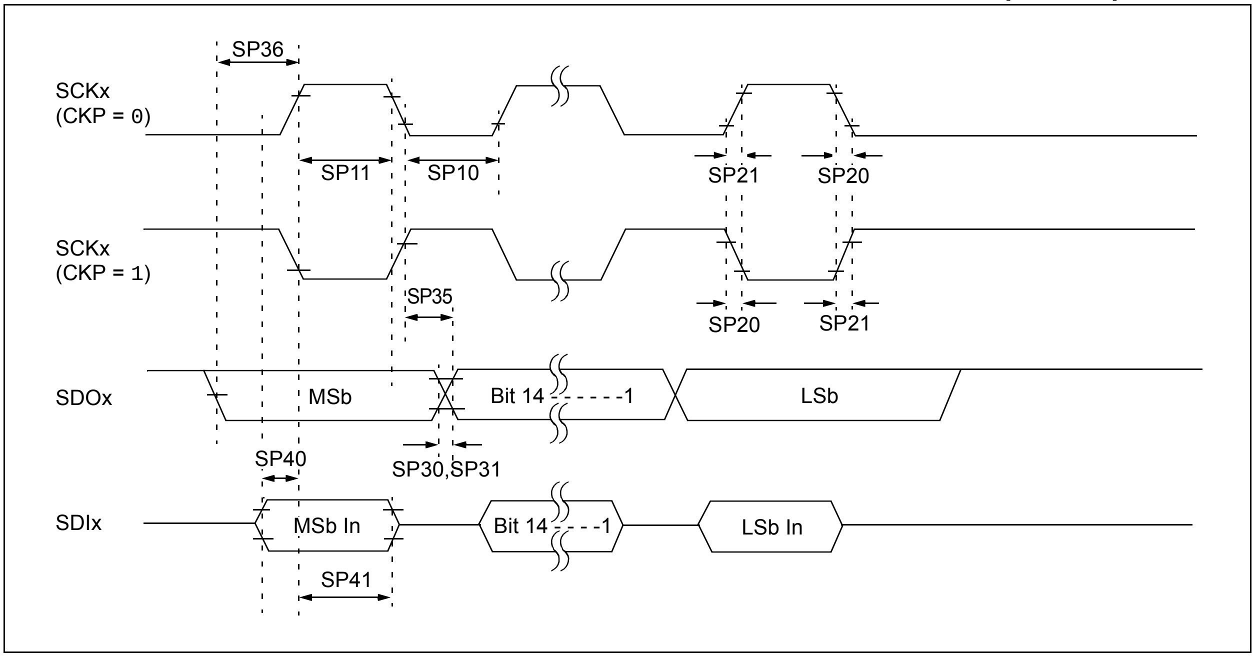


TABLE 29-37: SPIx MODULE MASTER MODE TIMING REQUIREMENTS (CKE = 1)

AC CHARACTERISTICS			Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	**Symbol**	**Characteristic**	**Min**	**Typ(1)**	**Max**	**Units**	**Conditions**
SP10	TscL	SCKx Output Low Time(2)	TCY/2	—	—	ns	
SP11	TscH	SCKx Output High Time(2)	TCY/2	—	—	ns	
SP20	TscF	SCKx Output Fall Time(3)	—	10	25	ns	
SP21	TscR	SCKx Output Rise Time(3)	—	10	25	ns	
SP30	TdoF	SDOx Data Output Fall Time(3)	—	10	25	ns	
SP31	TdoR	SDOx Data Output Rise Time(3)	—	10	25	ns	
SP35	TscH2doV, TscL2doV	SDOx Data Output Valid after SCKx Edge	—	—	30	ns	
SP36	TdoV2sc, TdoV2scL	SDOx Data Output Setup to First SCKx Edge	30	—	—	ns	
SP40	TdiV2scH, TdiV2scL	Setup Time of SDIx Data Input to SCKx Edge	20	—	—	ns	
SP41	TscH2diL, TscL2diL	Hold Time of SDIx Data Input to SCKx Edge	20	—	—	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: The minimum clock period for SCKx is 100 ns; therefore, the clock generated in Master mode must not violate this specification.

3: This assumes a 50 pF load on all SPIx pins.

FIGURE 29-20: SPIx MODULE SLAVE MODE TIMING CHARACTERISTICS (CKE = 0)

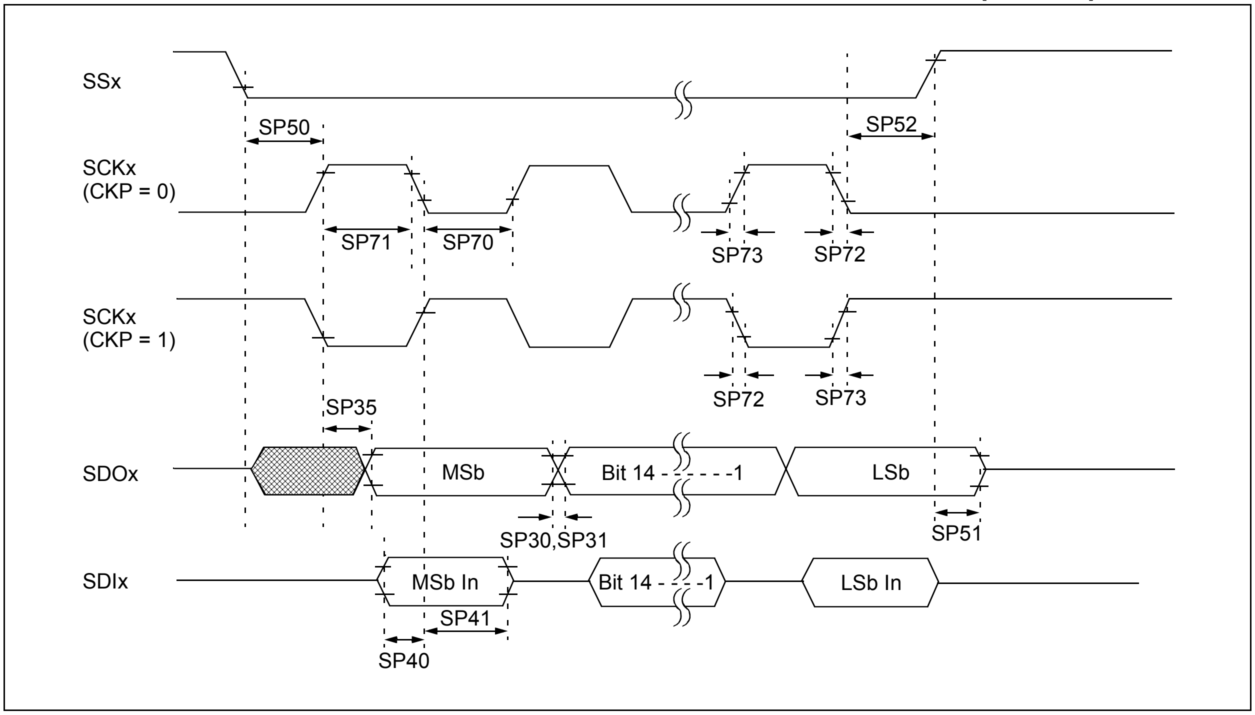


TABLE 29-38: SPIx MODULE SLAVE MODE TIMING REQUIREMENTS (CKE = 0)

AC CHARACTERISTICS			Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	**Symbol**	**Characteristic**	**Min**	**Typ(1)**	**Max**	**Units**	**Conditions**
SP70	TscL	SCKx Input Low Time	30	—	—	ns	
SP71	TscH	SCKx Input High Time	30	—	—	ns	
SP72	TscF	SCKx Input Fall Time(2)	—	10	25	ns	
SP73	TscR	SCKx Input Rise Time(2)	—	10	25	ns	
SP30	TdoF	SDOx Data Output Fall Time(2)	—	10	25	ns	
SP31	TdoR	SDOx Data Output Rise Time(2)	—	10	25	ns	
SP35	TscH2doV, TscL2doV	SDOx Data Output Valid after SCKx Edge	—	—	30	ns	
SP40	TdiV2scH, TdiV2scL	Setup Time of SDIx Data Input to SCKx Edge	20	—	—	ns	
SP41	TscH2diL, TscL2diL	Hold Time of SDIx Data Input to SCKx Edge	20	—	—	ns	
SP50	TssL2scH, TssL2scL	$\overline{SSx}$ to SCKx ↑ or SCKx Input	120	—	—	ns	
SP51	TssH2doZ	$\overline{SSx}$ ↑ to SDOx Output High-Impedance(3)	10	—	50	ns	
SP52	TscH2ssH TscL2ssH	$\overline{SSx}$ after SCKx Edge	1.5 TCY + 40	—	—	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: The minimum clock period for SCKx is 100 ns; therefore, the clock generated in Master mode must not violate this specification.

3: This assumes a 50 pF load on all SPIx pins.

FIGURE 29-21: SPIx MODULE SLAVE MODE TIMING CHARACTERISTICS (CKE = 1)

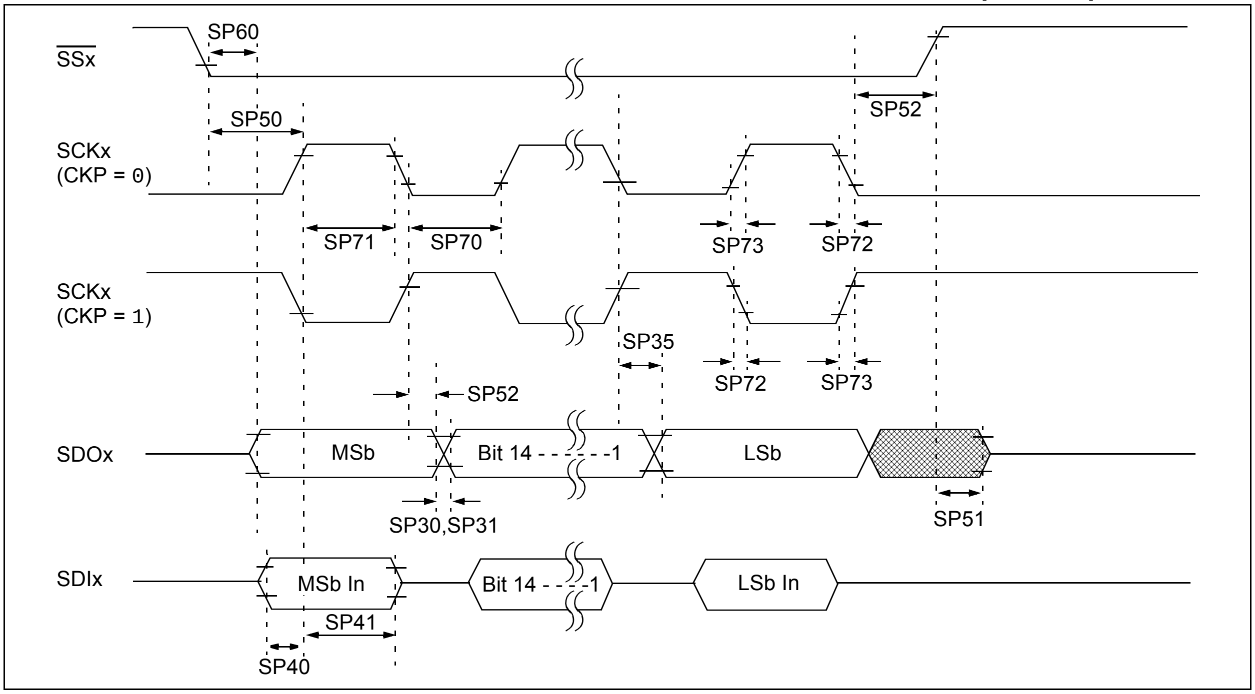


TABLE 29-39: SPIx MODULE SLAVE MODE TIMING REQUIREMENTS (CKE = 1)

AC CHARACTERISTICS			Standard Operating Conditions: 2.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	Symbol	Characteristic	Min	Typ(1)	Max	Units	Conditions
SP70	TscL	SCKx Input Low Time	30	—	—	ns	
SP71	TscH	SCKx Input High Time	30	—	—	ns	
SP72	TscF	SCKx Input Fall Time(2)	—	10	25	ns	
SP73	TscR	SCKx Input Rise Time(2)	—	10	25	ns	
SP30	TdoF	SDOx Data Output Fall Time(2)	—	10	25	ns	
SP31	TdoR	SDOx Data Output Rise Time(2)	—	10	25	ns	
SP35	TscH2doV, TscL2doV	SDOx Data Output Valid after SCKx Edge	—	—	30	ns	
SP40	TdiV2scH, TdiV2scL	Setup Time of SDIx Data Input to SCKx Edge	20	—	—	ns	
SP41	TscH2diL, TscL2diL	Hold Time of SDIx Data Input to SCKx Edge	20	—	—	ns	
SP50	TssL2scH, TssL2scL	$\overline{\text{SSx}}$ ↓ to SCKx ↓ or SCKx ↑ Input	120	—	—	ns	
SP51	TssH2doZ	$\overline{\text{SSx}}$ ↑ to SDOx Output High-Impedance(3)	10	—	50	ns	
SP52	TscH2ssH TscL2ssH	$\overline{\text{SSx}}$ ↑ after SCKx Edge	1.5 TCY + 40	—	—	ns	
SP60	TssL2doV	SDOx Data Output Valid after $\overline{\text{SSx}}$ Edge	—	—	50	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: The minimum clock period for SCKx is 100 ns; therefore, the clock generated in Master mode must not violate this specification.

3: This assumes a 50 pF load on all SPIx pins.

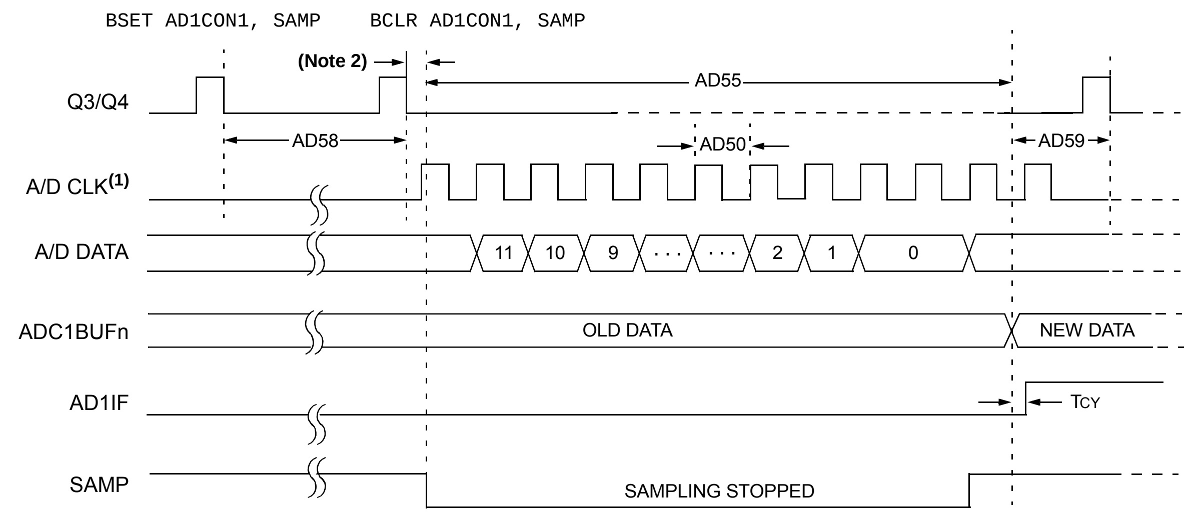
TABLE 29-40: A/D MODULE SPECIFICATIONS

AC CHARACTERISTICS			Standard Operating Conditions: 1.8V to 3.6V PIC24F32KA3XX 2.0V to 5.5V PIC24FV32KA3XX Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	**Symbol**	**Characteristic**	**Min.**	**Typ**	**Max.**	**Units**	**Conditions**
Device Supply							
AD01	AVDD	Module VDD Supply	Greater of: VDD – 0.3 or 1.8	—	Lesser of: VDD + 0.3 or 3.6	V	PIC24FXXKA30X devices
			Greater of: VDD – 0.3 or 2.0	—	Lesser of: VDD + 0.3 or 5.5	V	PIC24FVXXKA30X devices
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	
AD08	IVREF	Reference Voltage Input Current	—	1.25	—	mA	
AD09	ZVREF	Reference Input Impedance	—	10k	—	Ω	
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	
AD17	RIN	Recommended Impedance of Analog Voltage Source	—	—	1k	Ω	12-bit
A/D Accuracy							
AD20b	NR	Resolution	—	12	—	bits	
AD21b	INL	Integral Nonlinearity	—	±1	±9	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 5V
AD22b	DNL	Differential Nonlinearity	—	±1	±5	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 5V
AD23b	GERR	Gain Error	—	±1	±9	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 5V
AD24b	EOFF	Offset Error	—	±1	±5	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 5V
AD25b		Monotonicity[1]	—	—	—	—	Guaranteed

Note 1: The A/D conversion result never decreases with an increase in the input voltage.

2: Measurements are taken with external VREF+ and VREF- used as the A/D voltage reference.

FIGURE 29-22: A/D CONVERSION TIMING



Note 1: If the A/D clock source is selected as RC, a time of TCY is added before the A/D clock starts. This allows the SLEEP instruction to be executed.

2: This is a minimal RC delay (typically 100 ns) which also disconnects the holding capacitor from the analog input.

TABLE 29-41: A/D CONVERSION TIMING REQUIREMENTS[(1)]

AC CHARACTERISTICS			Standard Operating Conditions: 1.8V to 3.6V PIC24F32KA3XX 2.0V to 5.5V PIC24FV32KA3XX Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	**Symbol**	**Characteristic**	**Min.**	**Typ**	**Max.**	**Units**	**Conditions**
		Clock Parameters					
AD50	TAD	A/D Clock Period	600	—	—	ns	TCY = 75 ns, AD1CON3 in default state
AD51	TRC	A/D Internal RC Oscillator Period	—	1.67	—	µs	
		Conversion Rate					
AD55	TCONV	Conversion Time	— —	12 14	— —	TAD TAD	10-bit results 12-bit results
AD56	FCNV	Throughput Rate	—	—	100	ksps	
AD57	TSAMP	Sample Time	—	1	—	TAD	
AD58	TACQ	Acquisition Time	750	—	—	ns	(Note 2)
AD59	TSWC	Switching Time from Convert to Sample	—	—	(Note 3)		
AD60	TDIS	Discharge Time	12	—	—	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.

2: The time for the holding capacitor to acquire the "New" input voltage when the voltage changes full scale after the conversion (VDD to VSS or VSS to VDD).

3: On the following cycle of the device clock.

30.0 DC AND AC CHARACTERISTICS GRAPHS AND TABLES

Note: The graphs and tables provided following this note are a statistical summary based on a limited number of samples and are provided for informational purposes only. The performance characteristics listed herein are not tested or guaranteed. In some graphs or tables, the data presented may be outside the specified operating range (e.g., outside specified power supply range) and therefore, outside the warranted range.

Data for VDD levels greater than 3.3V are applicable to PIC24FV32KA304 family devices only.

30.1 Characteristics for Industrial Temperature Devices (-40°C to +85°C)

FIGURE 30-1: TYPICAL AND MAXIMUM IDD vs. FOSC (EC MODE, 2 MHz TO 32 MHz, -40°C TO +85°C)

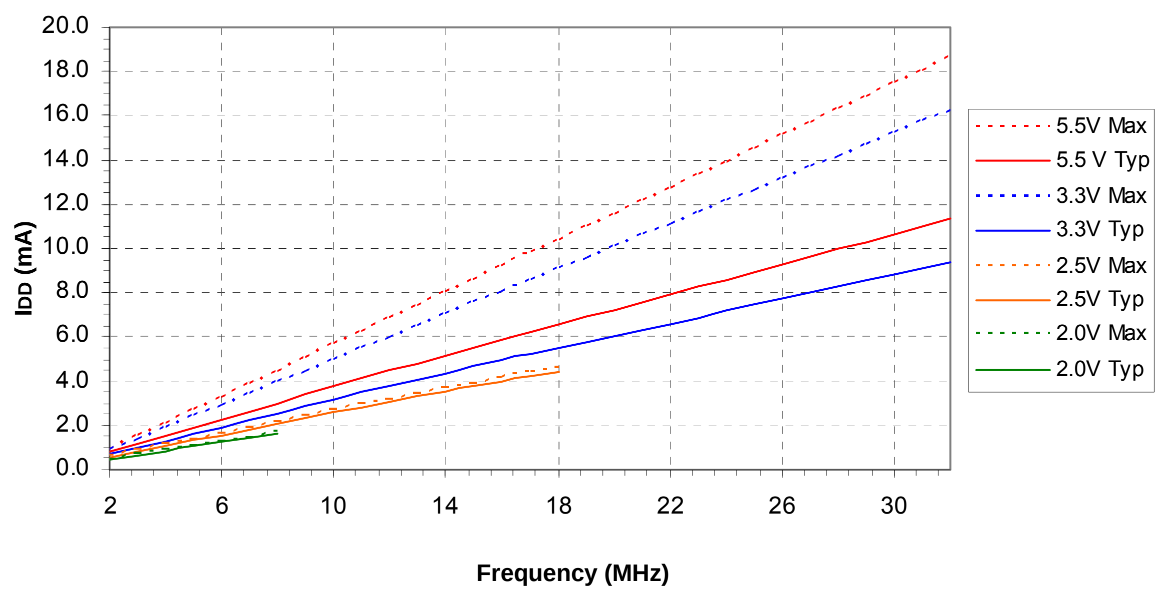


FIGURE 30-2: TYPICAL AND MAXIMUM IDD vs. FOSC (EC MODE, 1.95 kHz TO 1 MHz, +25°C)

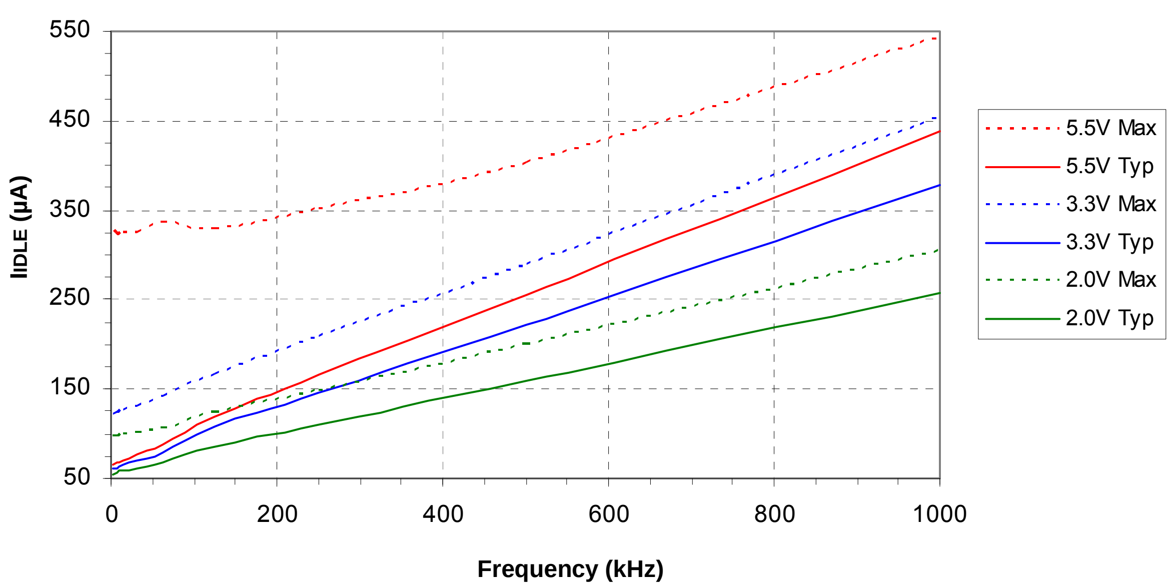


FIGURE 30-3: TYPICAL AND MAXIMUM IIDLE vs. FREQUENCY (EC MODE, 2 MHz TO 32 MHz)

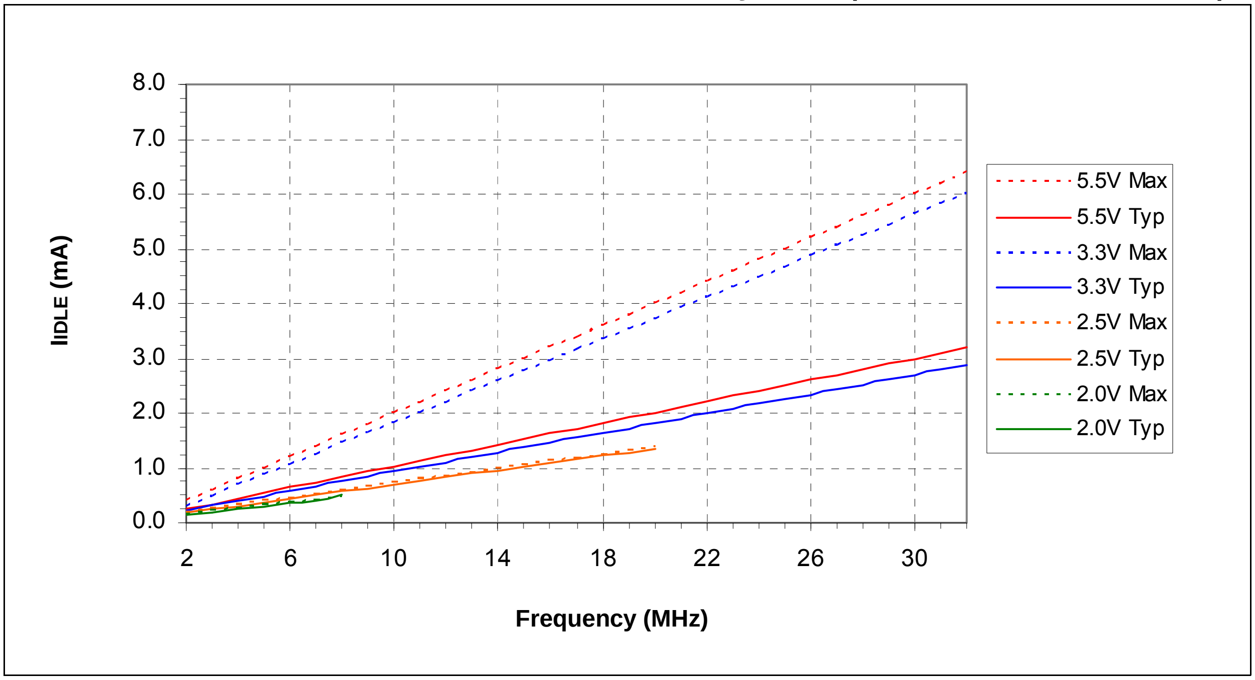


FIGURE 30-4: TYPICAL AND MAXIMUM IIDLE vs. FREQUENCY (EC MODE, 1.95 kHz TO 1 MHz)

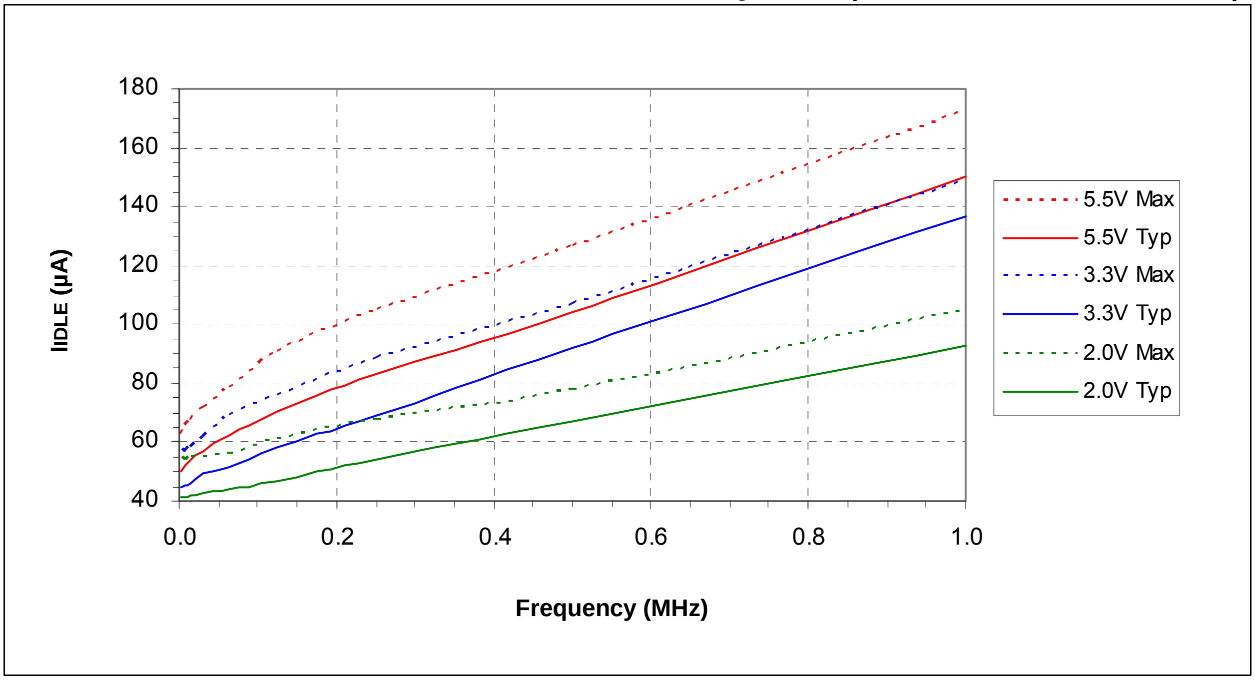


FIGURE 30-5: TYPICAL IDD vs. VDD (8 MHz, EC MODE)

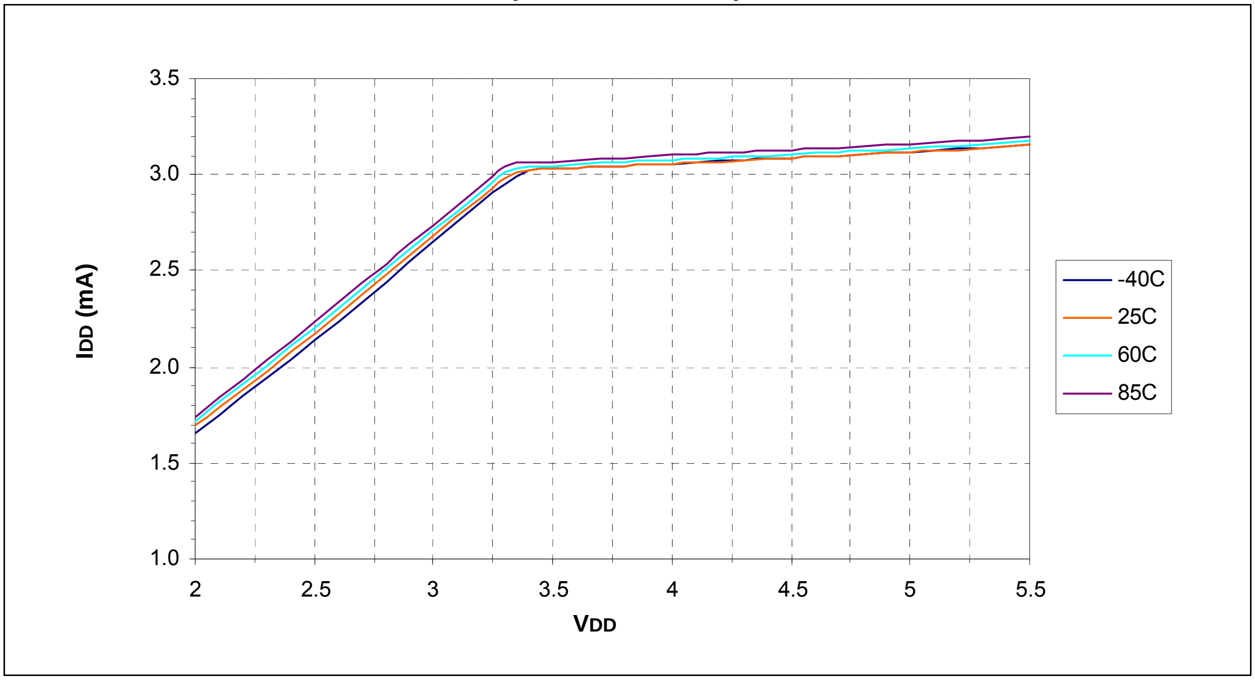


FIGURE 30-6: TYPICAL IDD vs. VDD (FRC MODE)

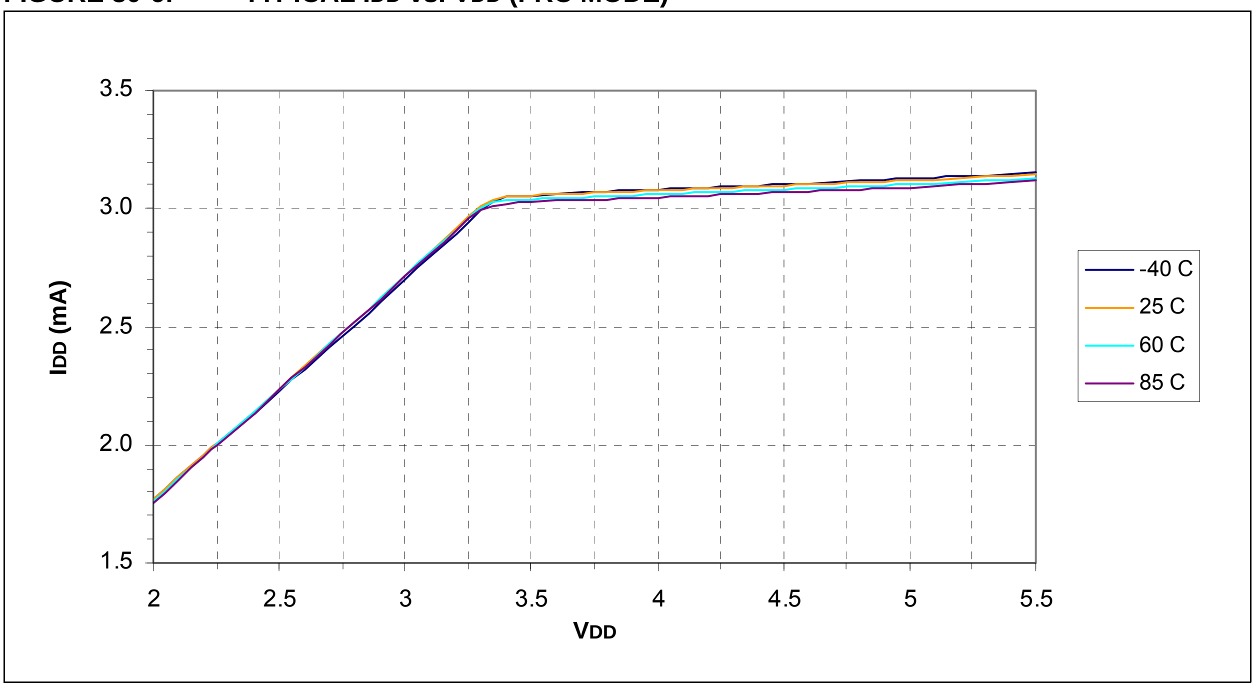


FIGURE 30-7: TYPICAL AND MAXIMUM IDD vs. TEMPERATURE (FRC MODE)

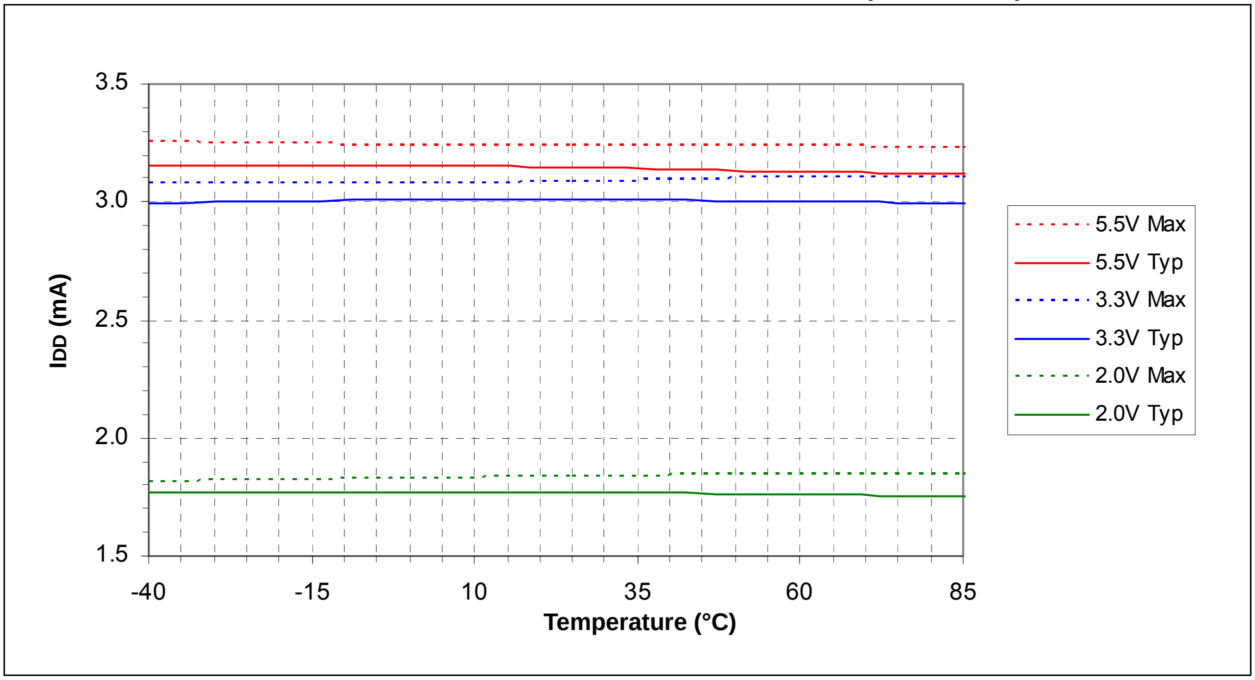


FIGURE 30-8: TYPICAL AND MAXIMUM IIDLE vs. VDD (FRC MODE)

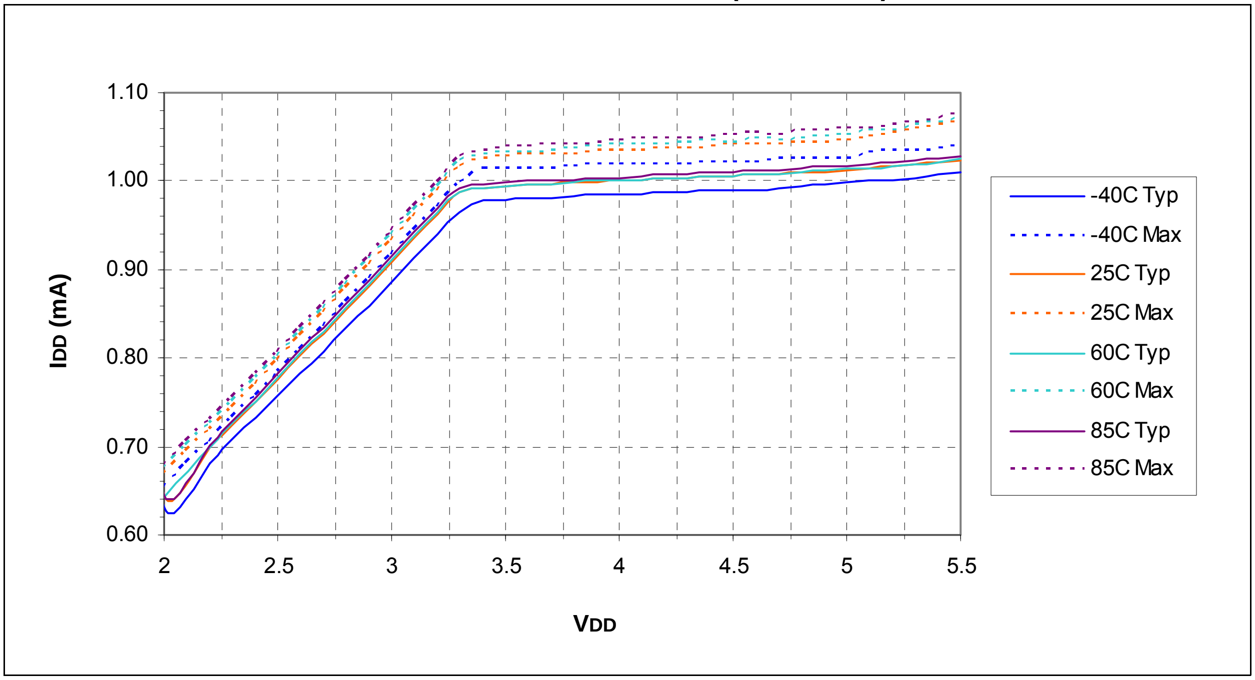


FIGURE 30-9: TYPICAL AND MAXIMUM I_{IDLE} vs. TEMPERATURE (FRC MODE)

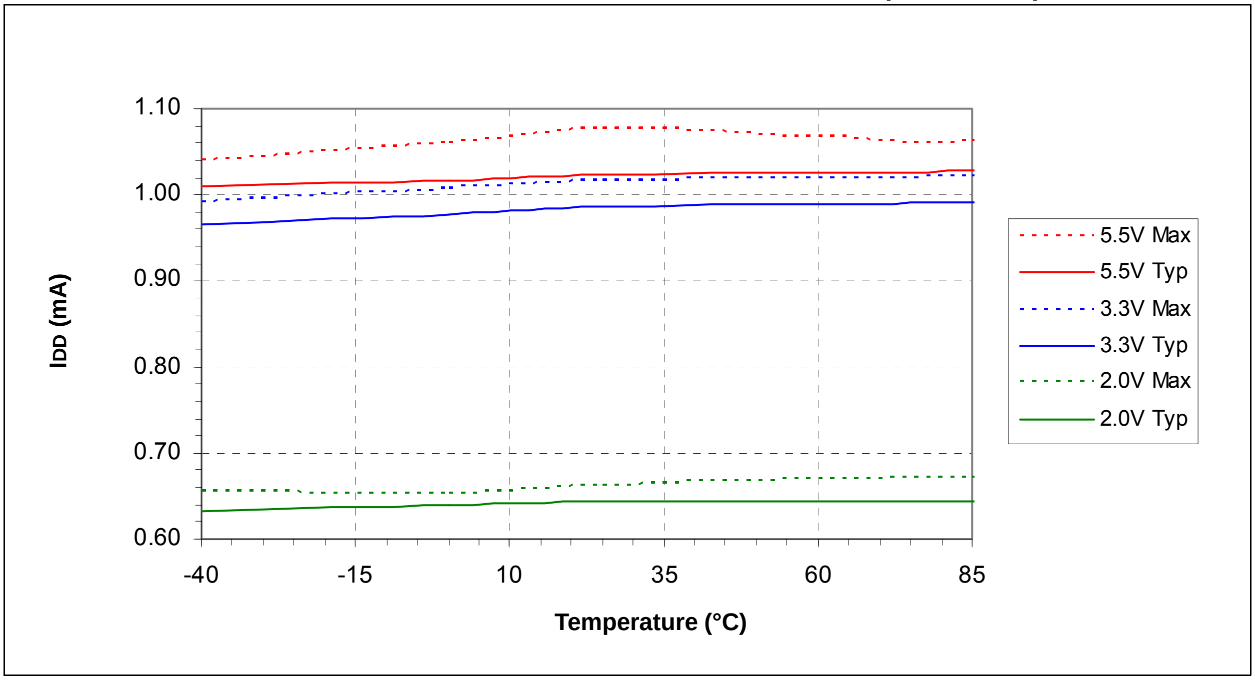


FIGURE 30-10: FRC FREQUENCY ACCURACY vs. V_{DD}

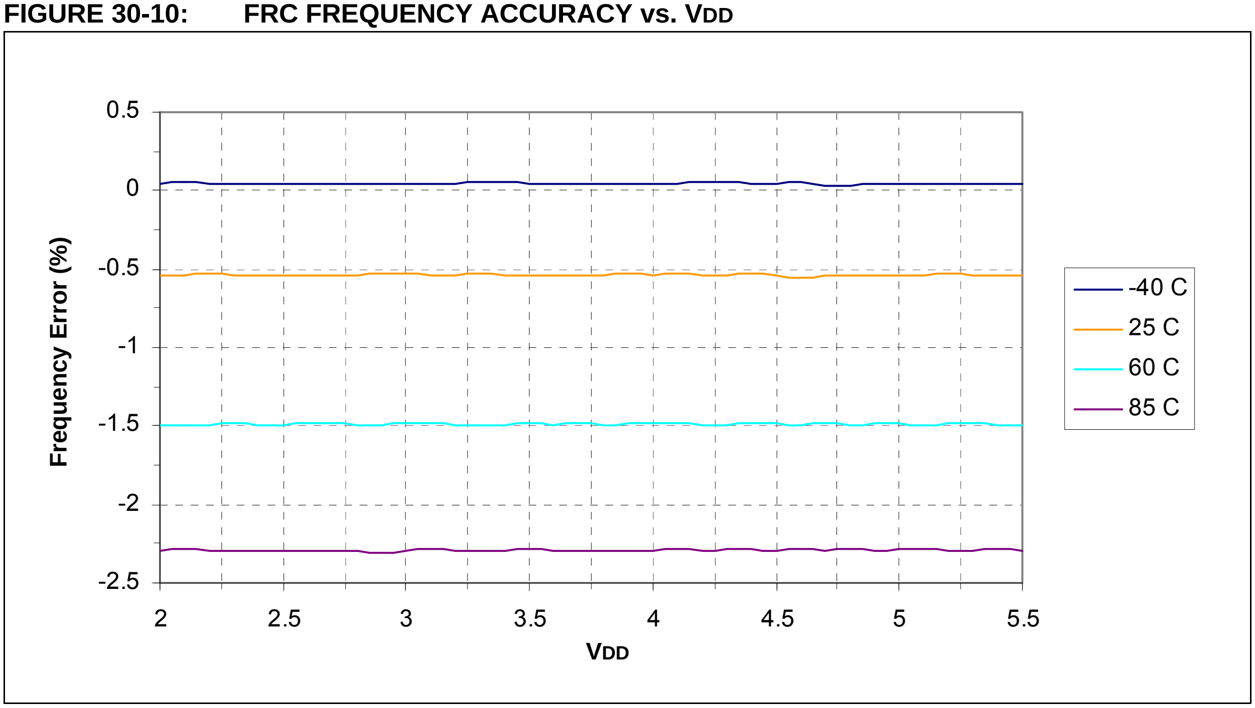


FIGURE 30-11: FRC FREQUENCY ACCURACY vs. TEMPERATURE (2.0V ≤ VDD ≤ 5.5V)

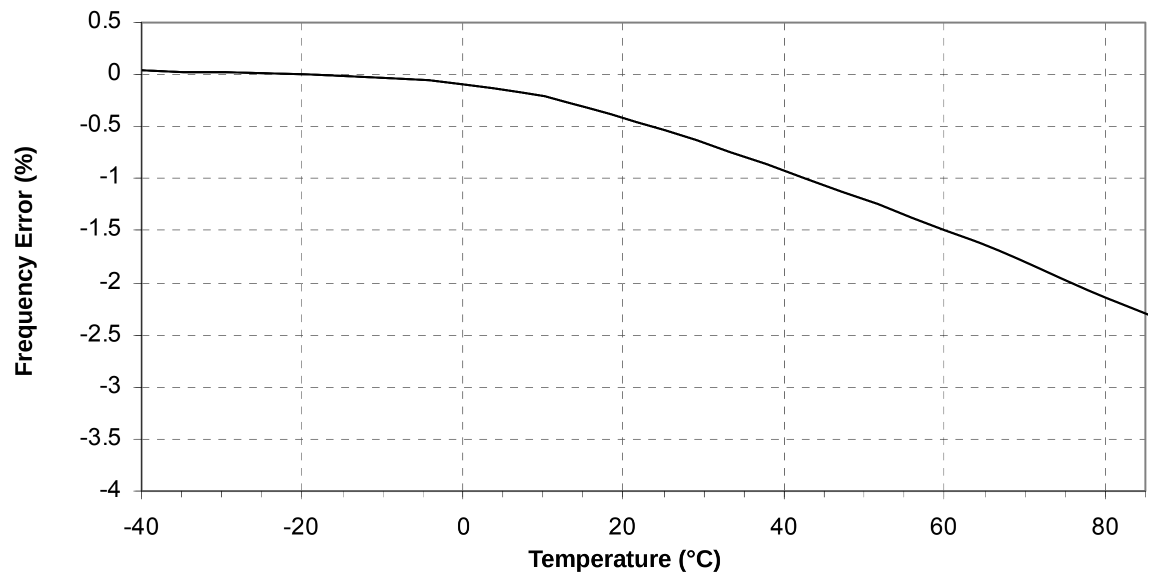


FIGURE 30-12: LPRC FREQUENCY ACCURACY vs. VDD

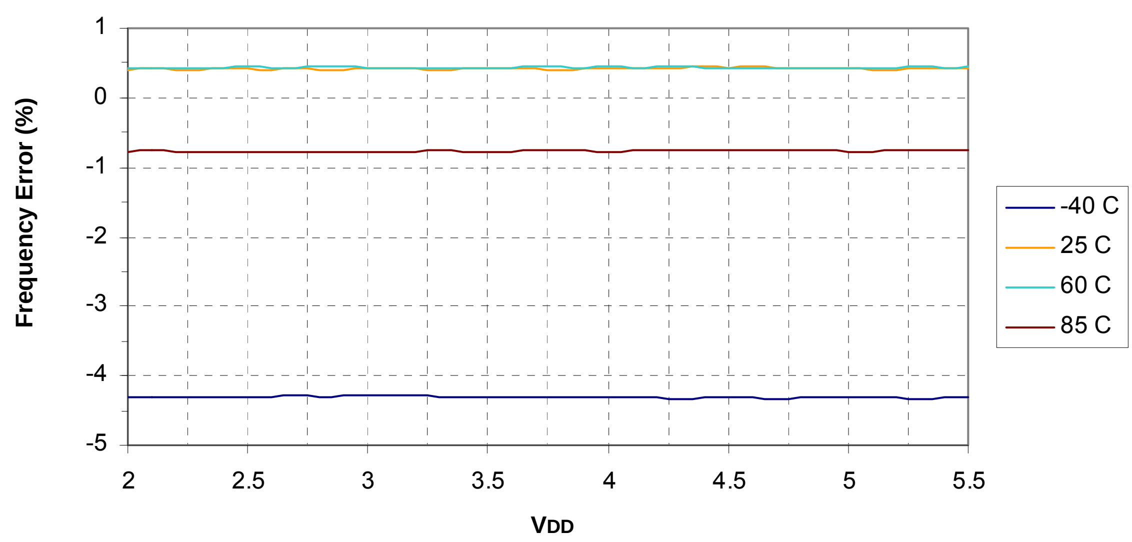


FIGURE 30-13: LPRC FREQUENCY ACCURACY vs. TEMPERATURE (2.0V ≤ VDD ≤ 5.5V)

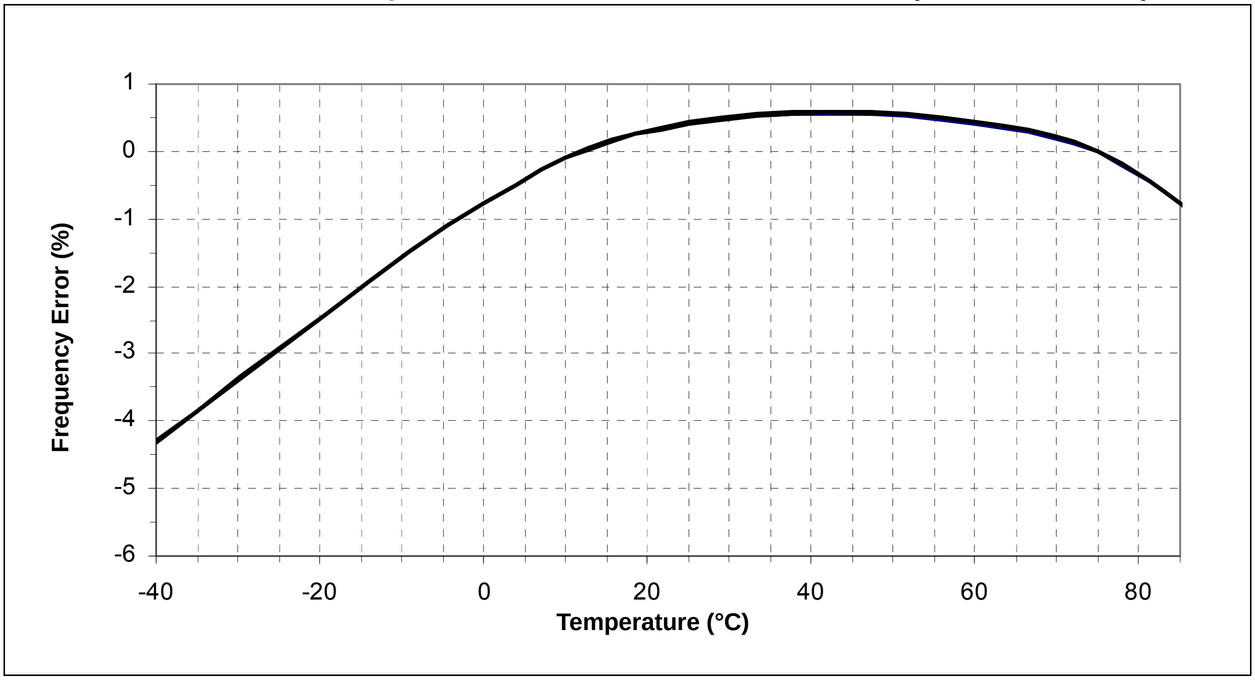


FIGURE 30-14: TYPICAL AND MAXIMUM IPD vs. VDD

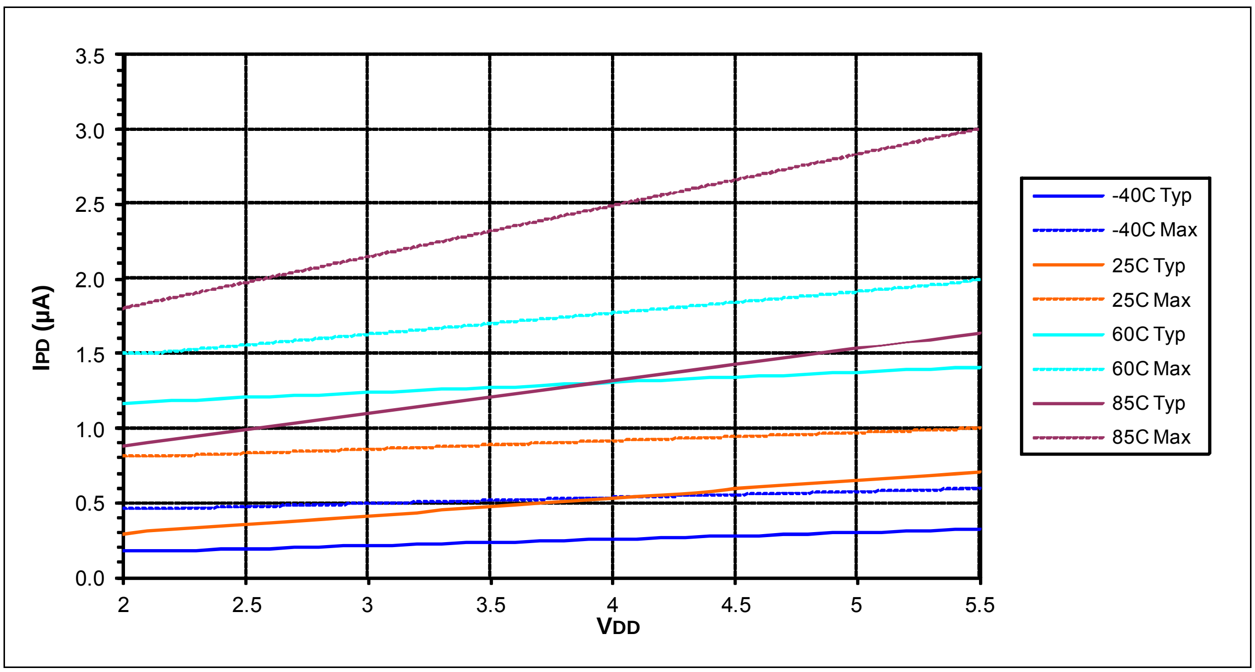


FIGURE 30-15: TYPICAL AND MAXIMUM IPD vs. TEMPERATURE

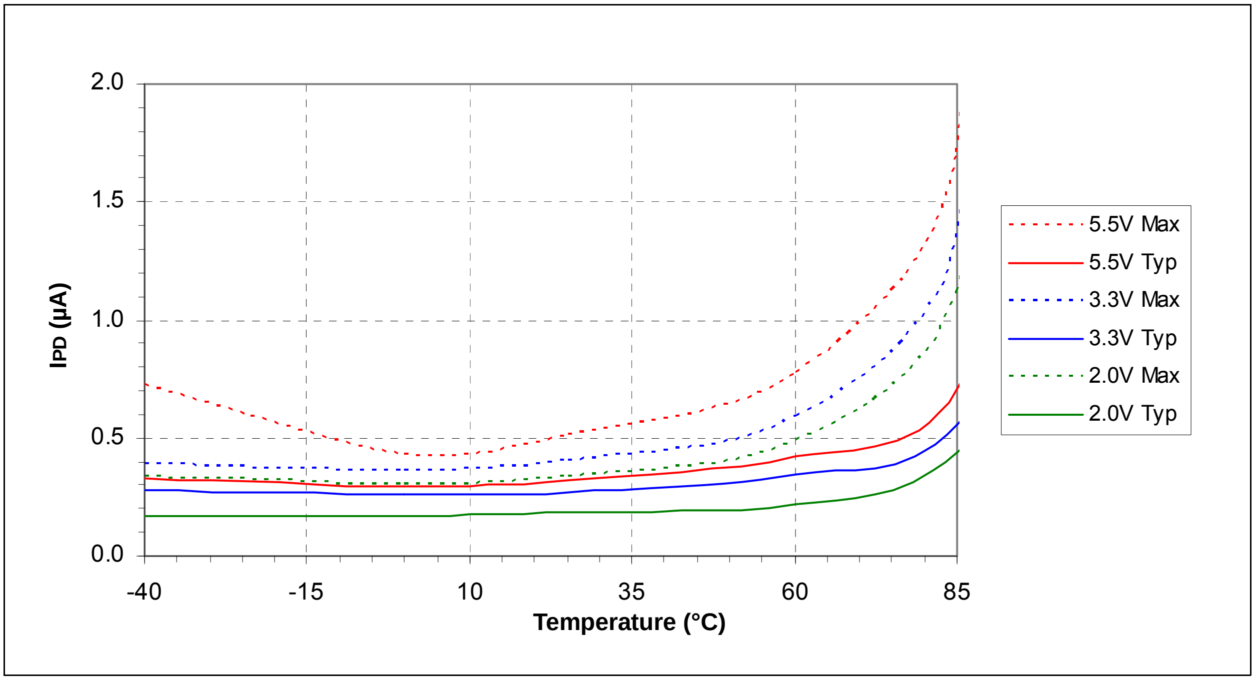


FIGURE 30-16: TYPICAL AND MAXIMUM IPD vs. VDD (DEEP SLEEP MODE)

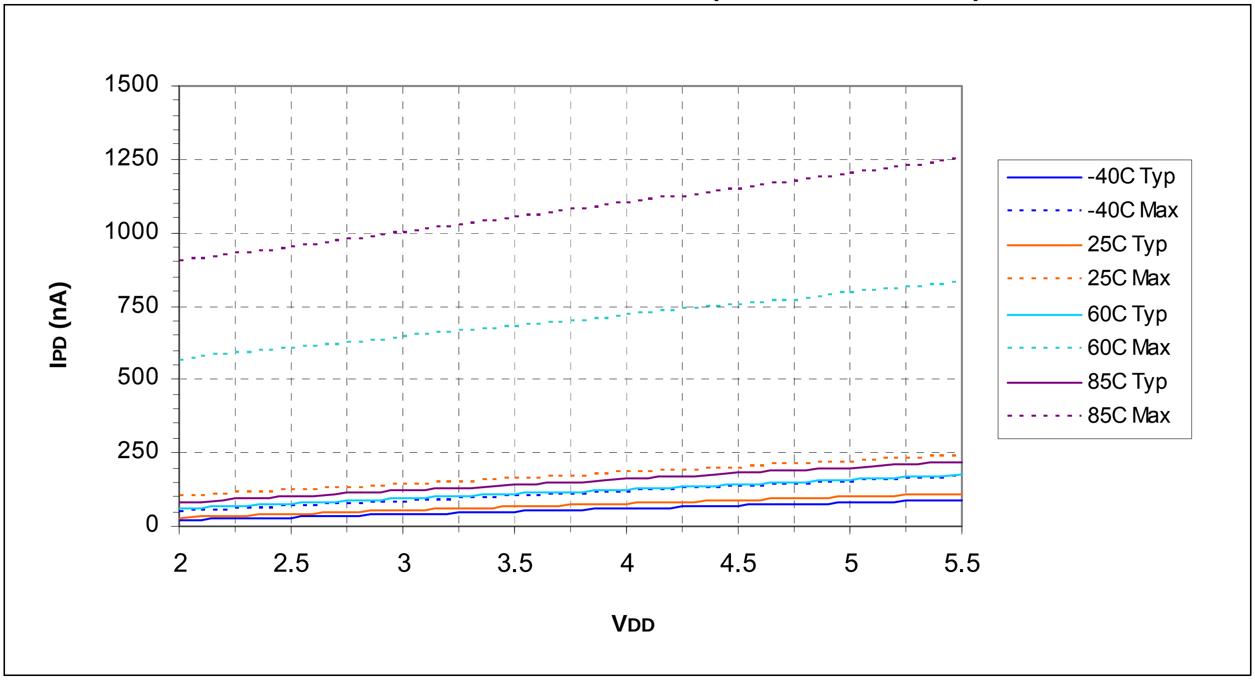


FIGURE 30-17: TYPICAL AND MAXIMUM IPD vs. TEMPERATURE (DEEP SLEEP MODE)

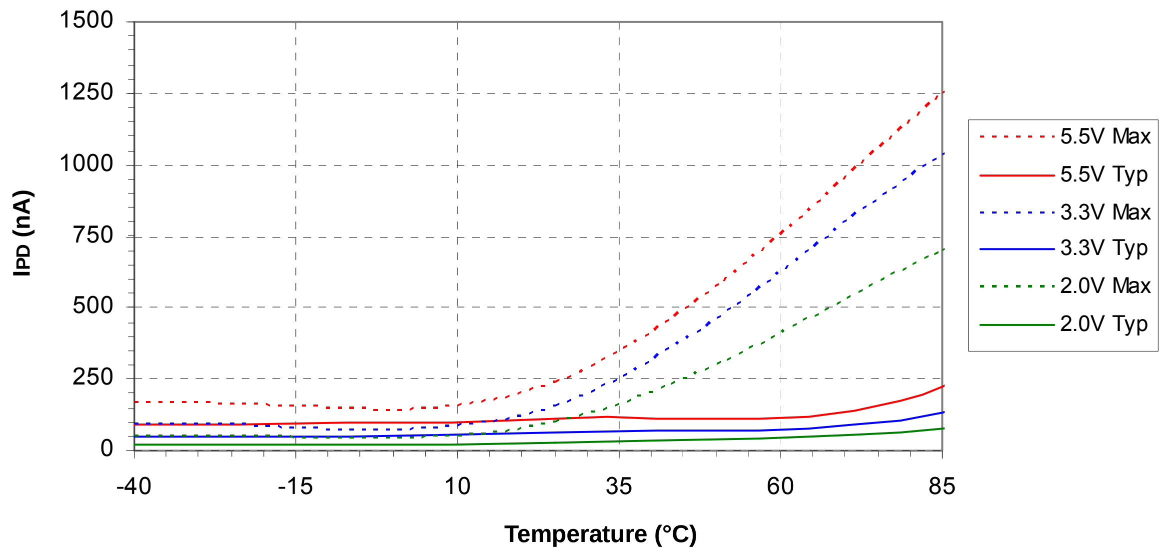


FIGURE 30-18: TYPICAL ΔIBOR vs. VDD

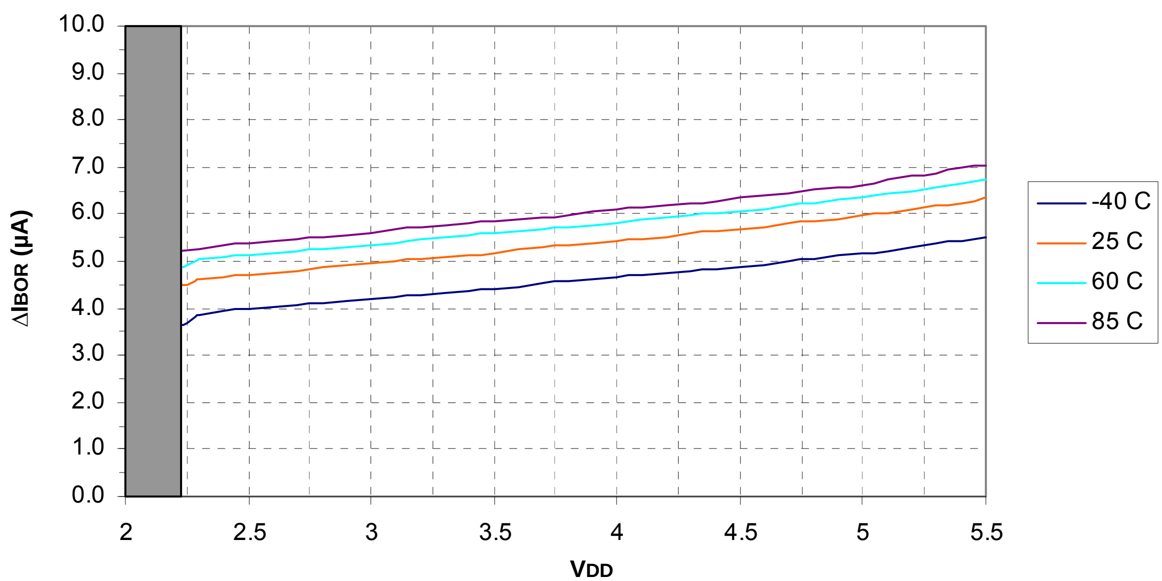


FIGURE 30-19: TYPICAL ΔIWDT vs. VDD

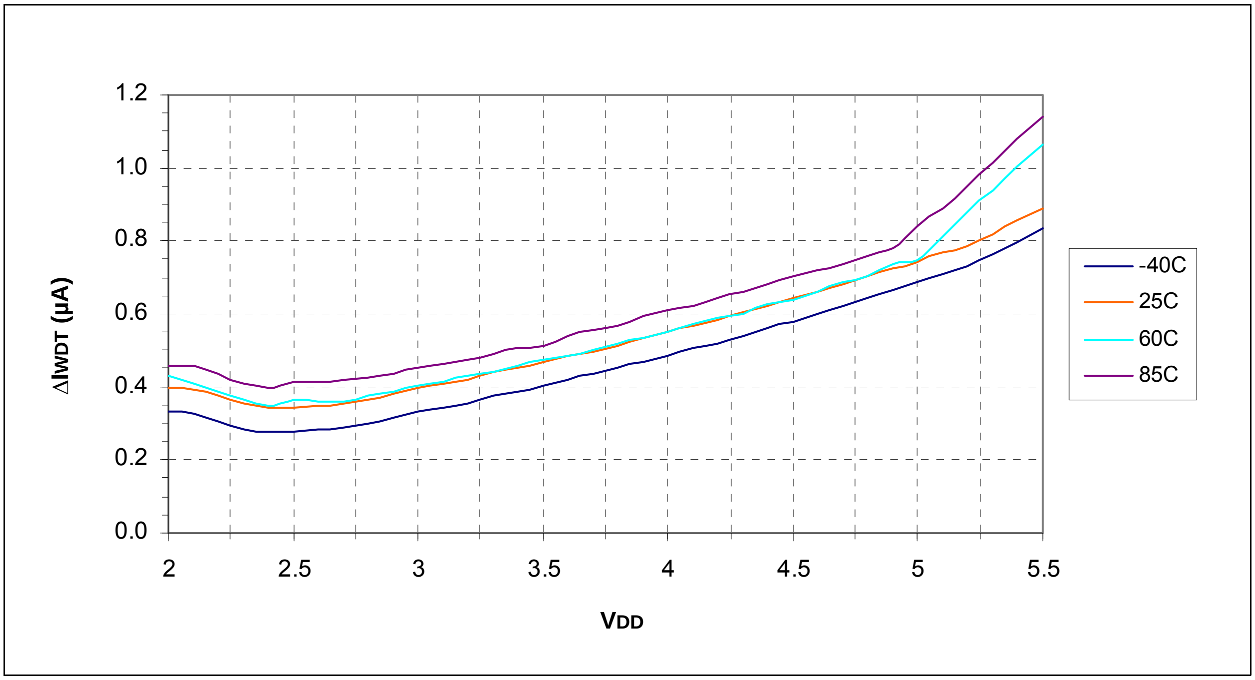


FIGURE 30-20: TYPICAL ΔIDSBOR vs. VDD

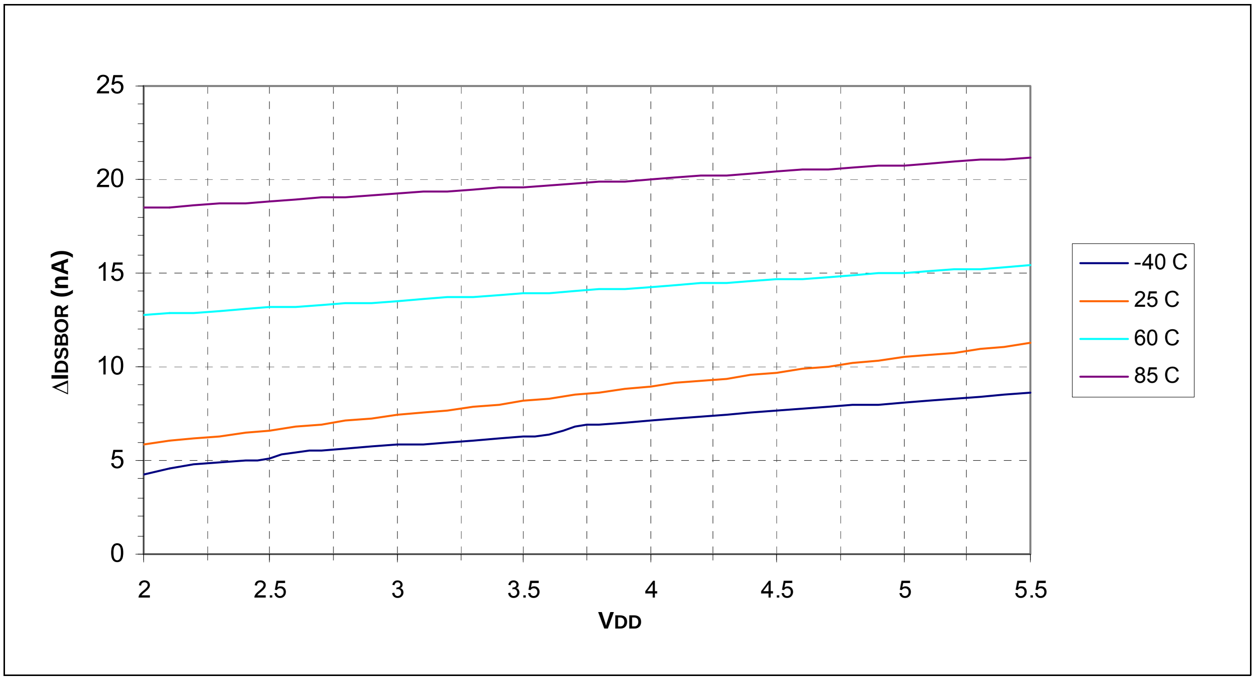


FIGURE 30-21: TYPICAL ΔIHLVD vs. VDD

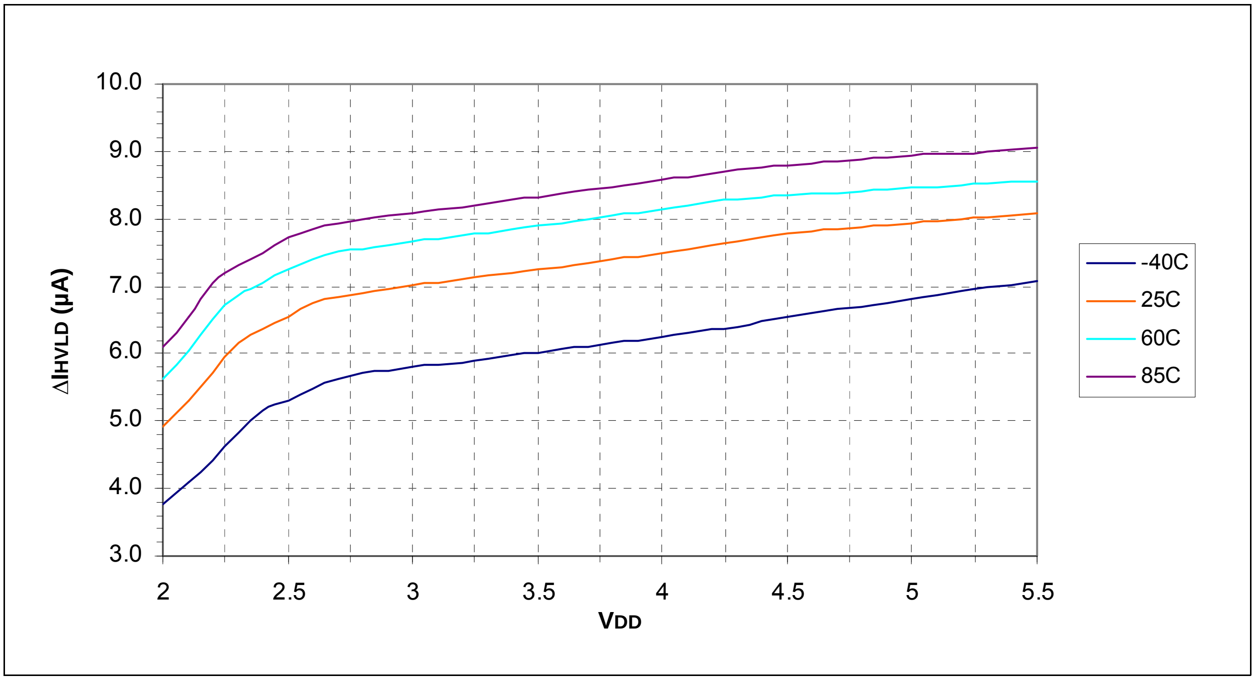


FIGURE 30-22: TYPICAL ΔIDSWDT vs. VDD

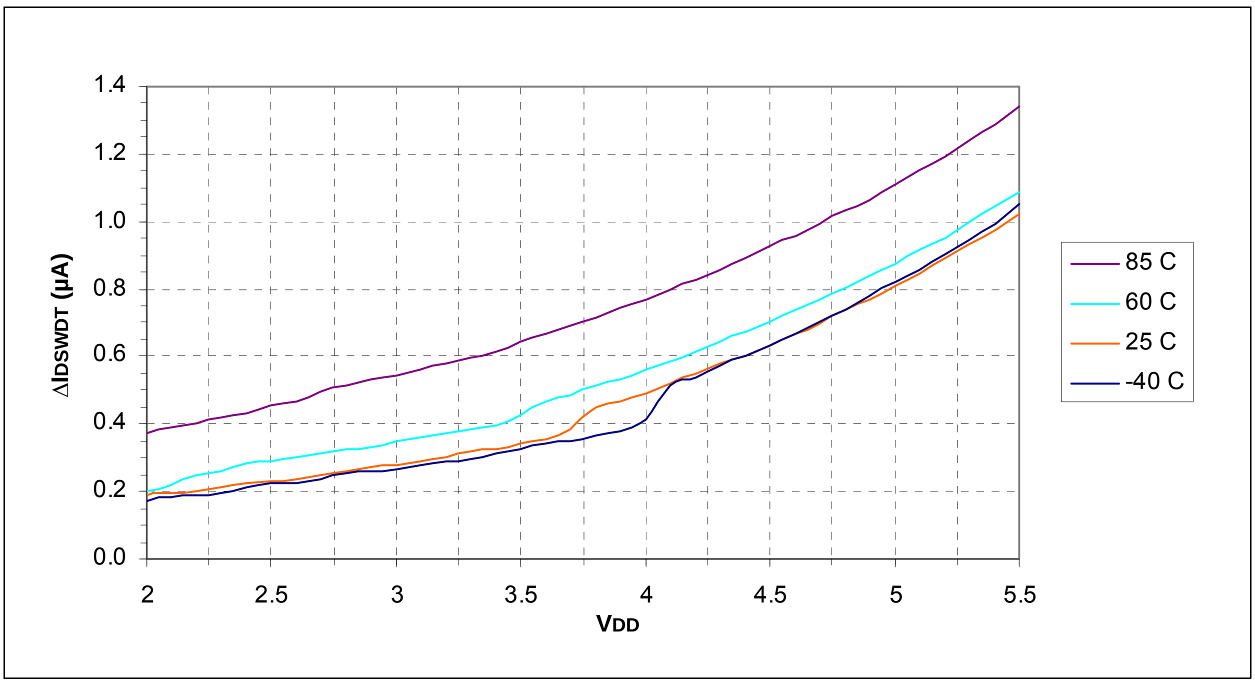


FIGURE 30-23: TYPICAL VBOR vs. TEMPERATURE (BOR TRIP POINT 3)

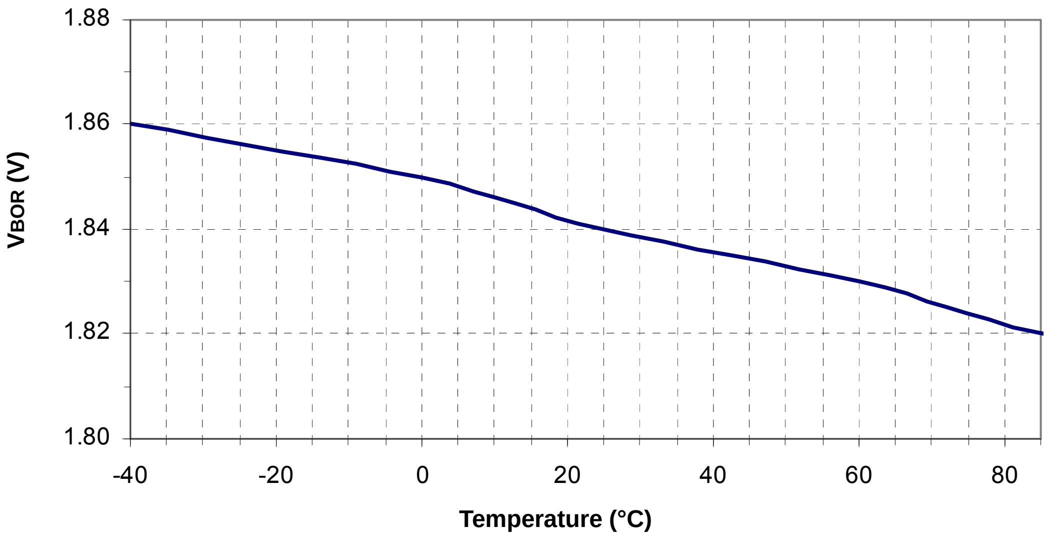


FIGURE 30-24: TYPICAL VOH vs. IOH (GENERAL PURPOSE I/O, AS A FUNCTION OF VDD)

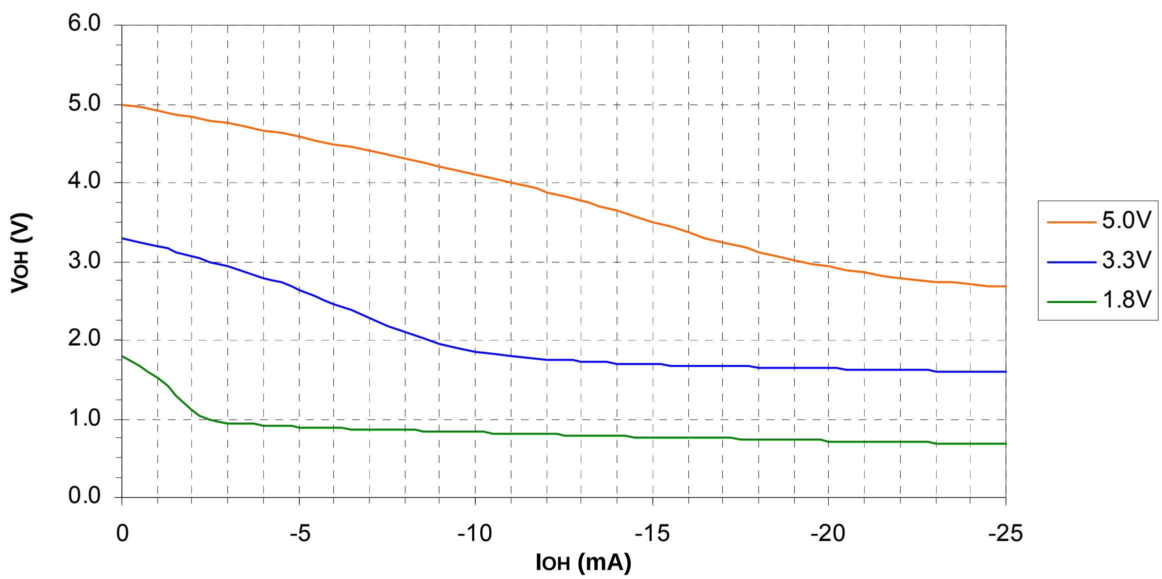


FIGURE 30-25: TYPICAL VOH vs. IOH (GENERAL PURPOSE I/O, AS A FUNCTION OF TEMPERATURE, 2.0V ≤ VDD ≤ 5.5V)

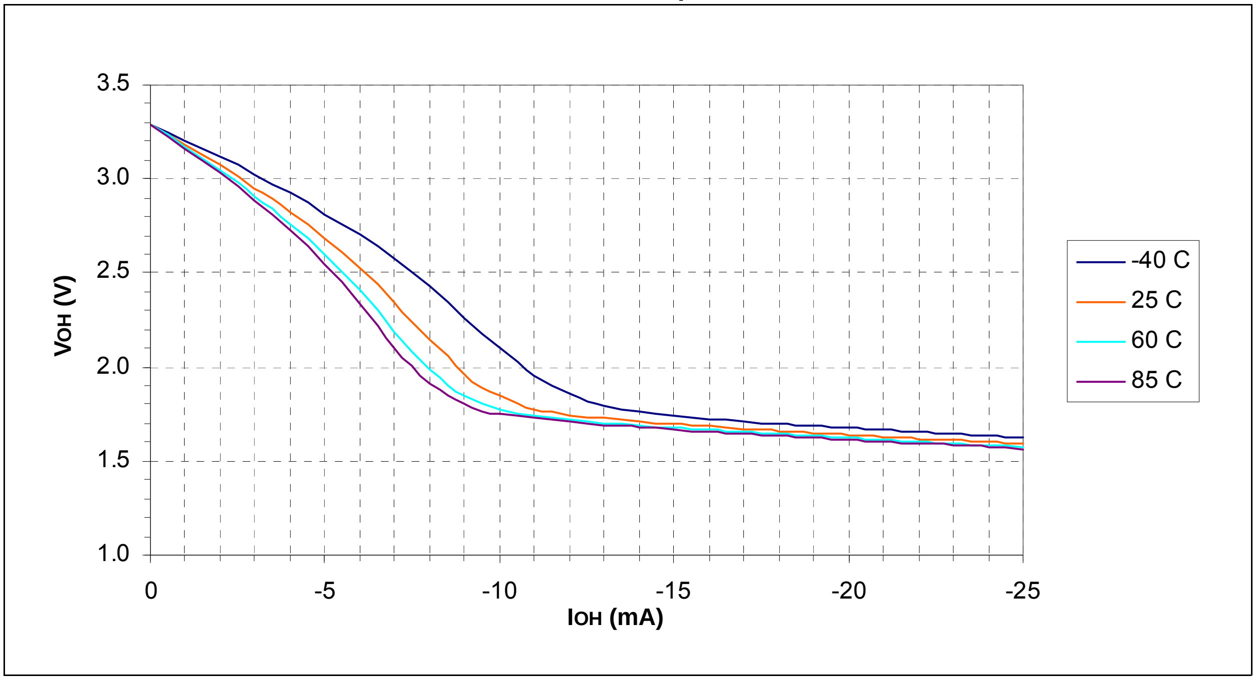


FIGURE 30-26: TYPICAL VOL vs. IOL (GENERAL PURPOSE I/O, AS A FUNCTION OF VDD)

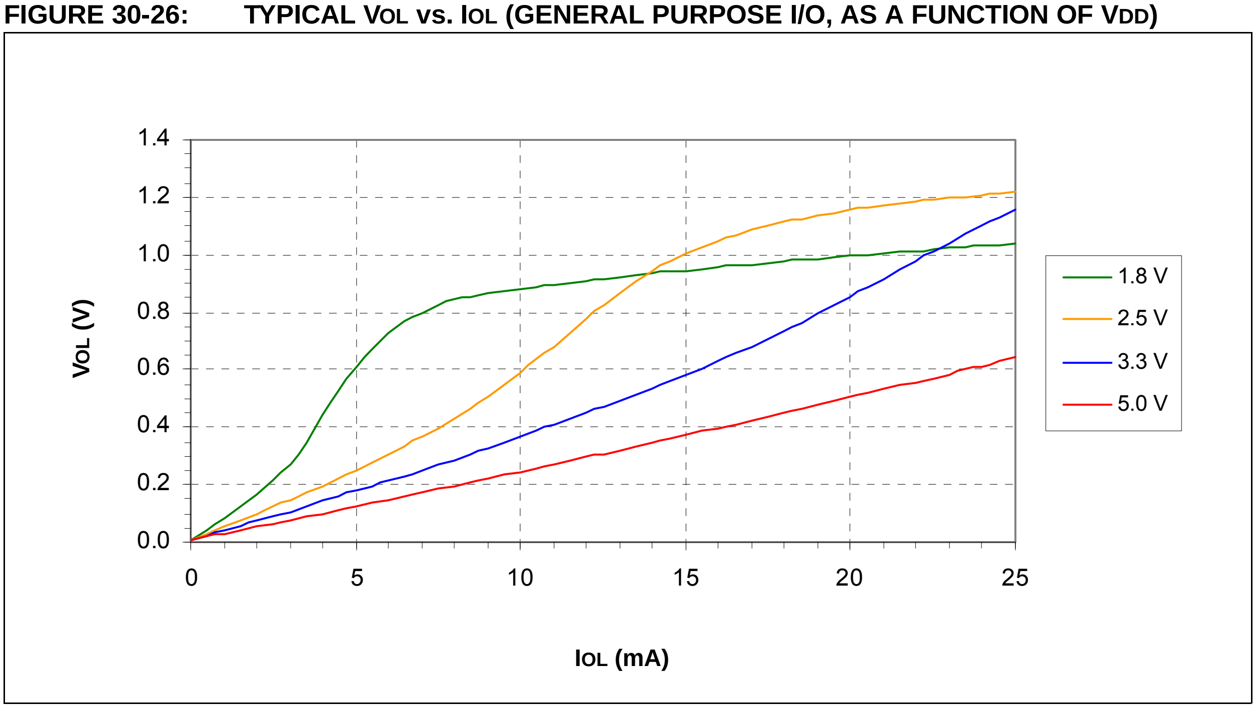


FIGURE 30-27: TYPICAL V_{OL} vs. I_{OL} (GENERAL PURPOSE I/O, AS A FUNCTION OF TEMPERATURE, 2.0V ≤ VDD ≤ 5.5V)

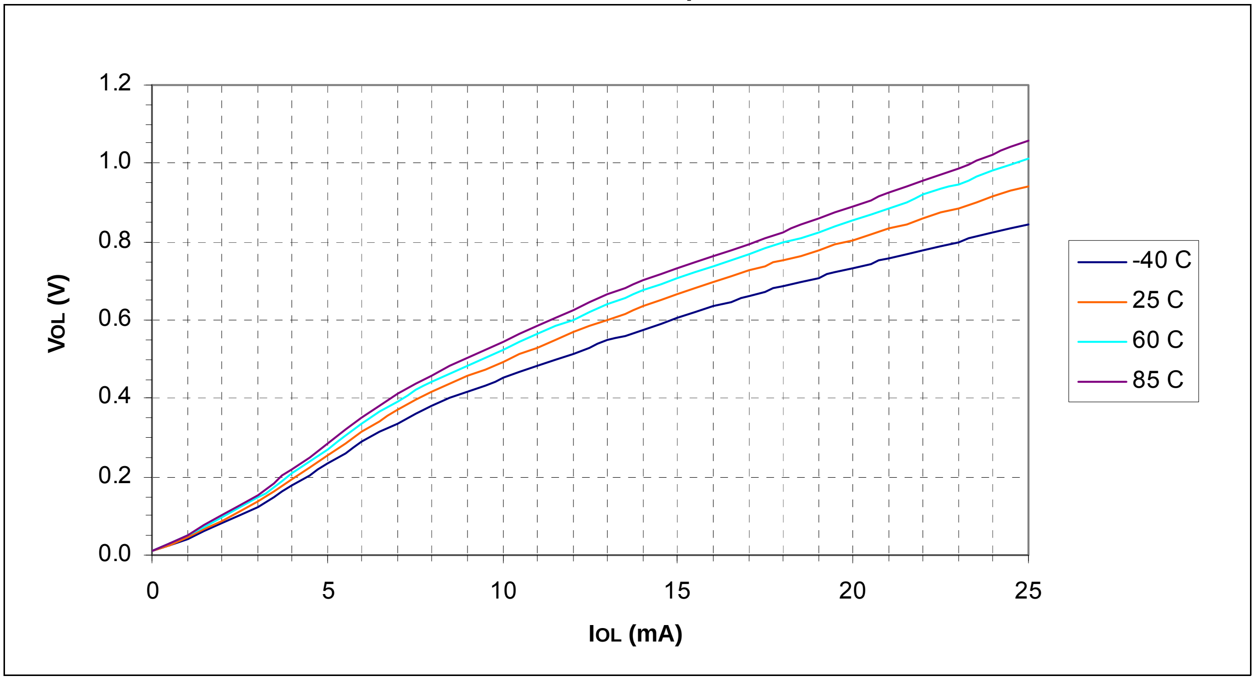


FIGURE 30-28: V_{IL}/V_{IH} vs. VDD (GENERAL PURPOSE I/O, TEMPERATURES AS NOTED)

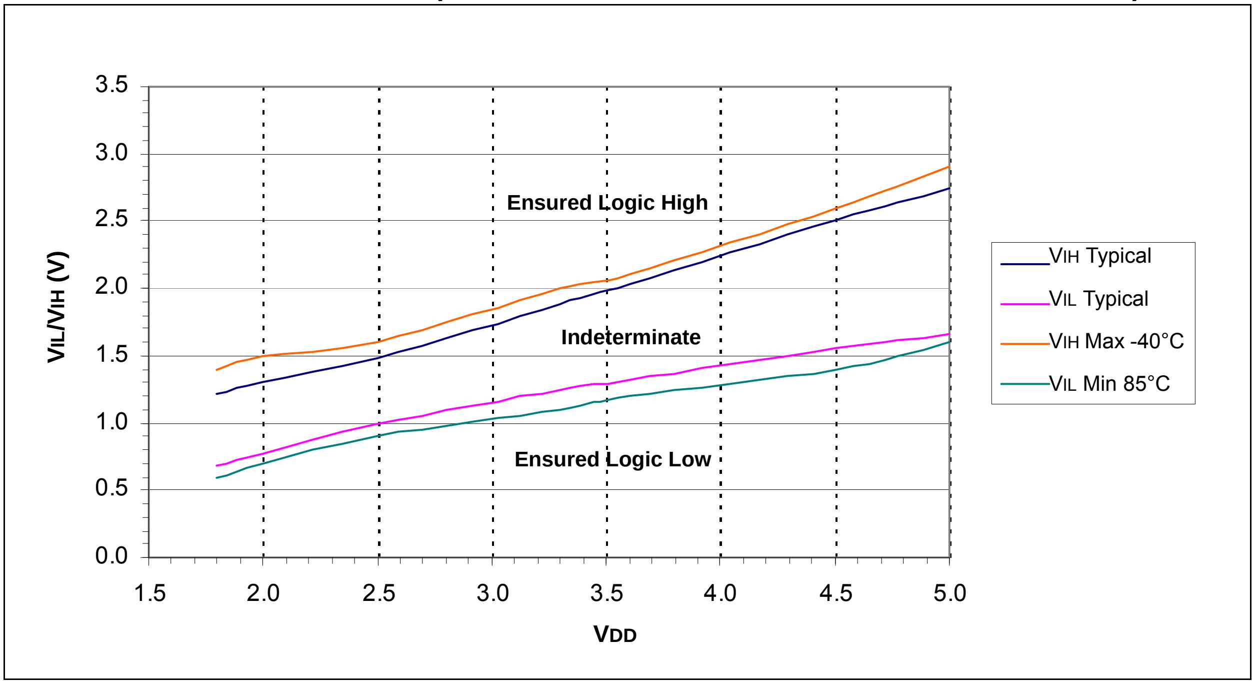


FIGURE 30-29: VIL/VIH vs. VDD (I^2C, TEMPERATURES AS NOTED)

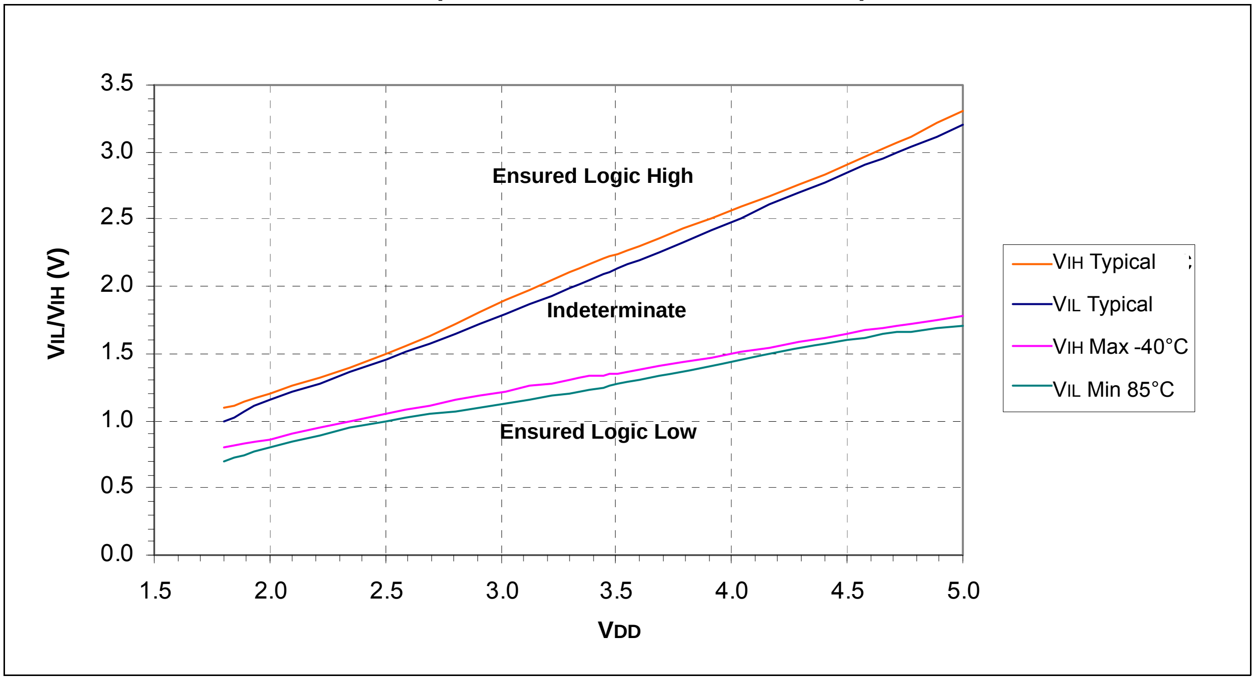


FIGURE 30-30: VIL/VIH vs. VDD (OSCO, TEMPERATURES AS NOTED)

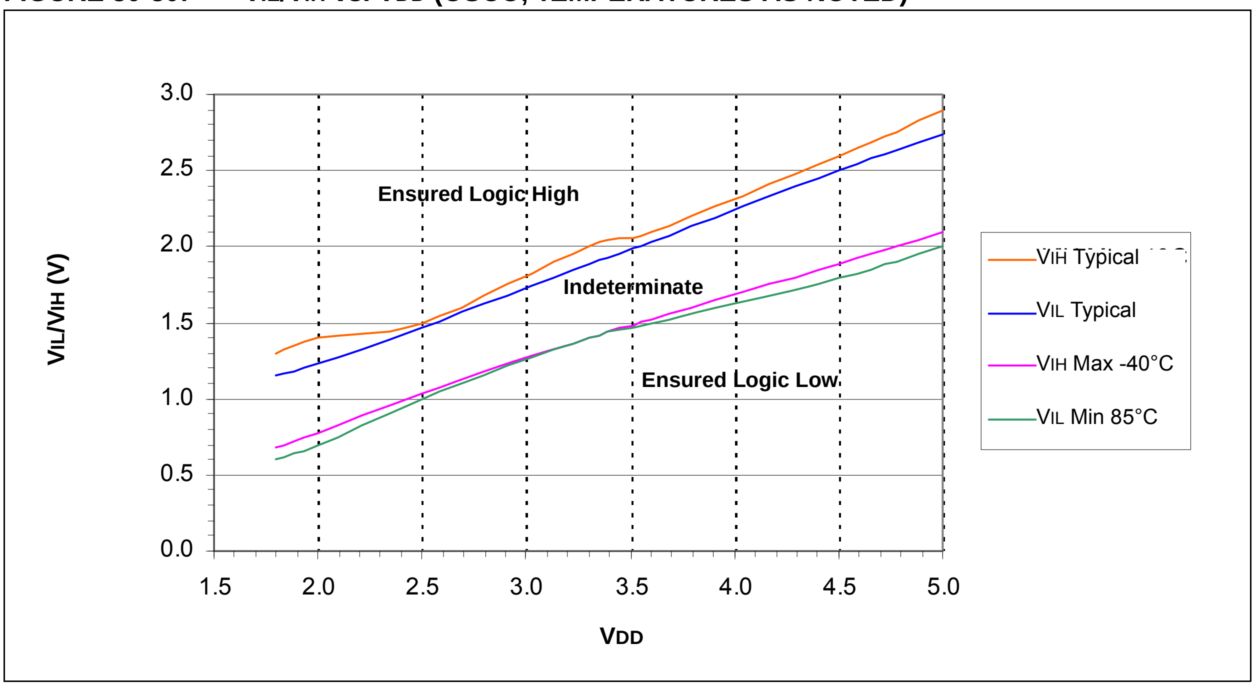


FIGURE 30-31: VIL/VIH vs. VDD ($\overline{MCLR}$, TEMPERATURES AS NOTED)

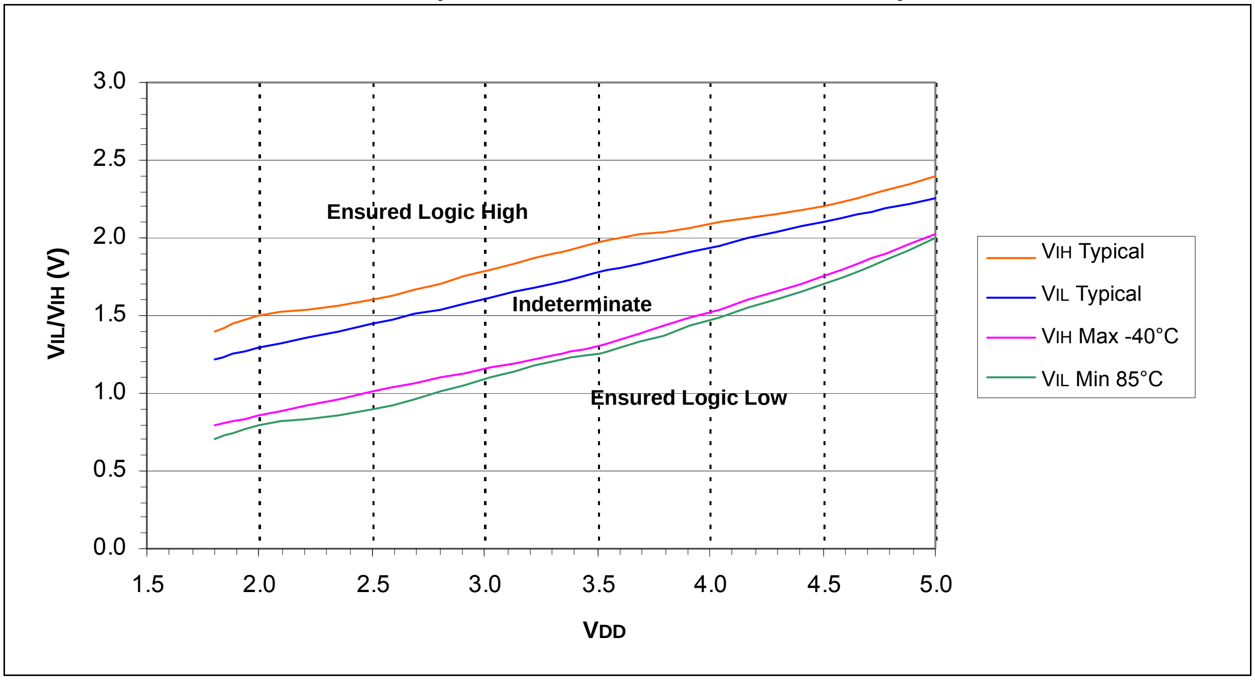


FIGURE 30-32: TYPICAL BAND GAP VOLTAGE vs. VDD

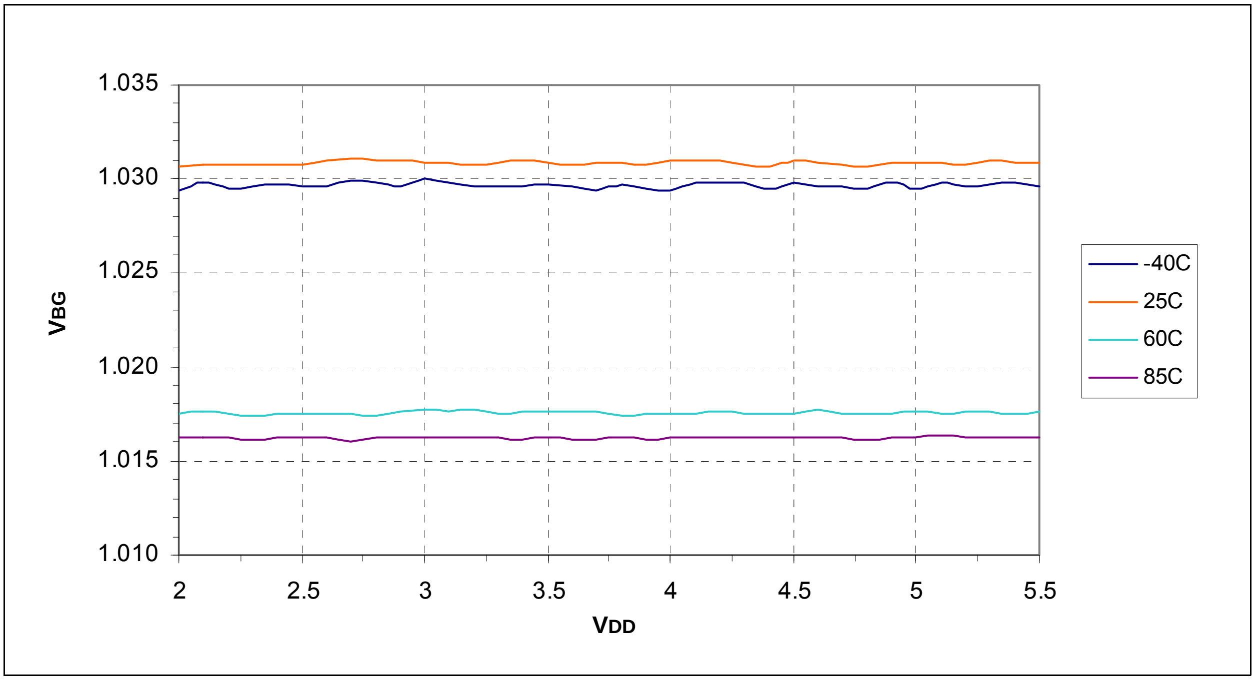


FIGURE 30-33: TYPICAL BAND GAP VOLTAGE vs. TEMPERATURE (2.0V ≤ VDD ≤ 5.5V)

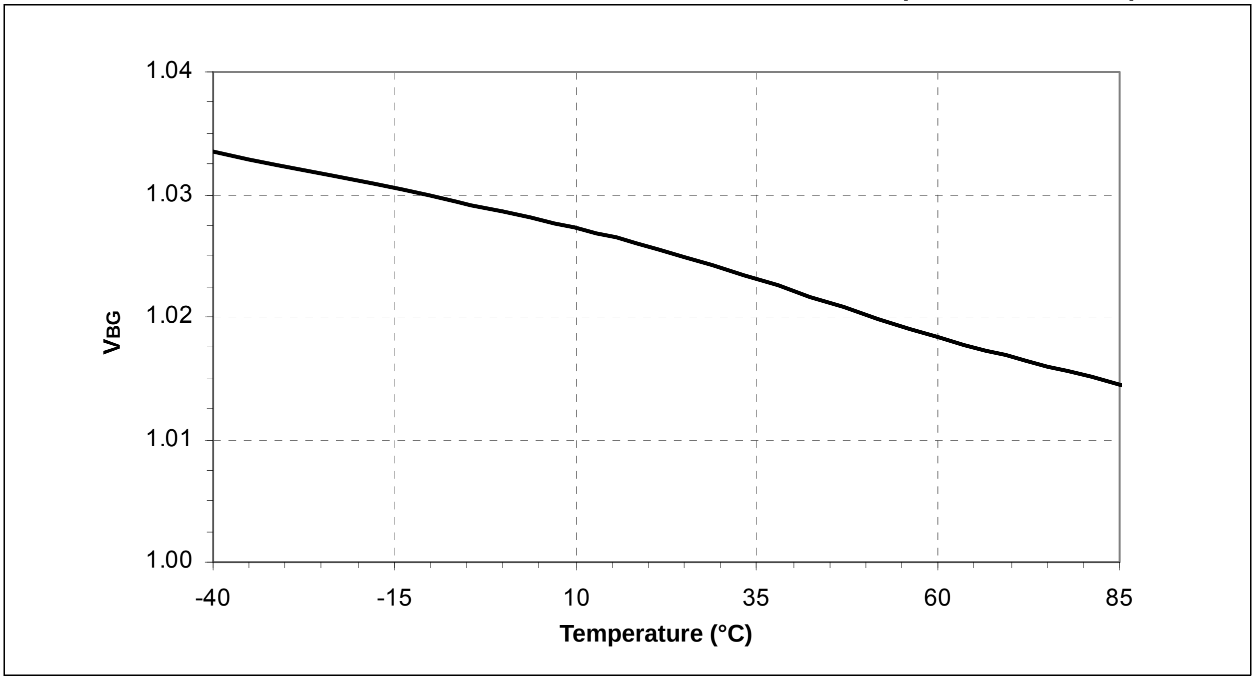


FIGURE 30-34: TYPICAL VOLTAGE REGULATOR OUTPUT vs. VDD

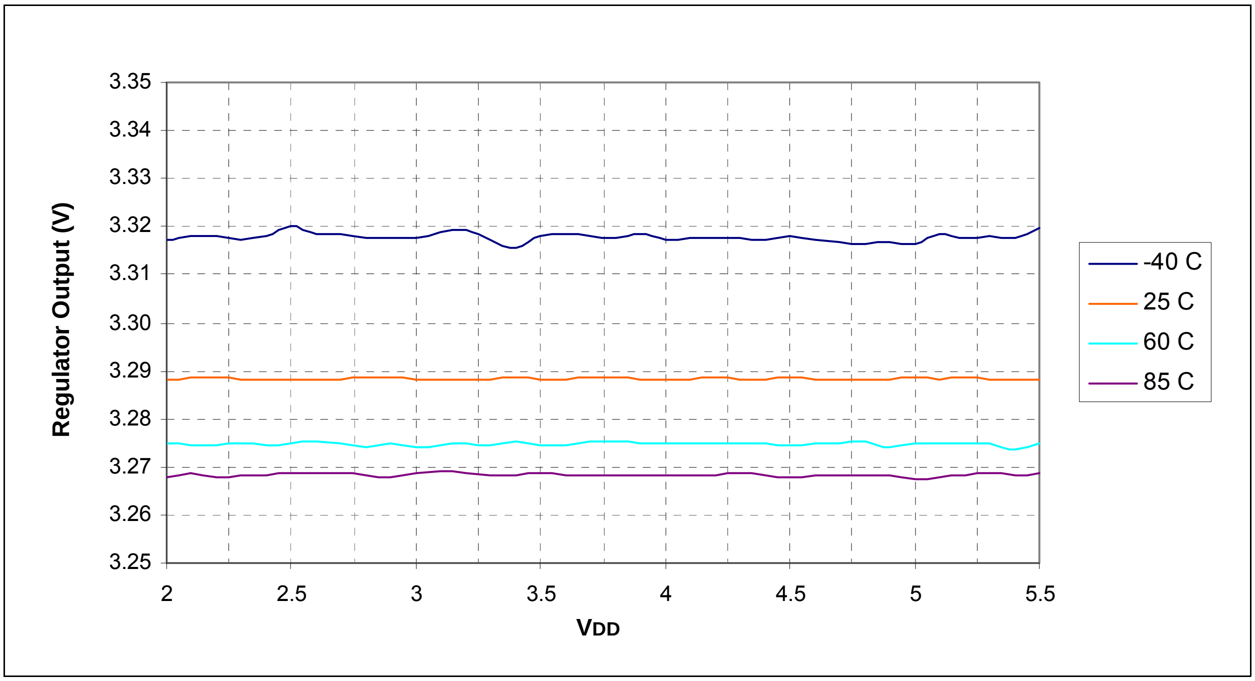


FIGURE 30-35: TYPICAL VOLTAGE REGULATOR OUTPUT vs. TEMPERATURE

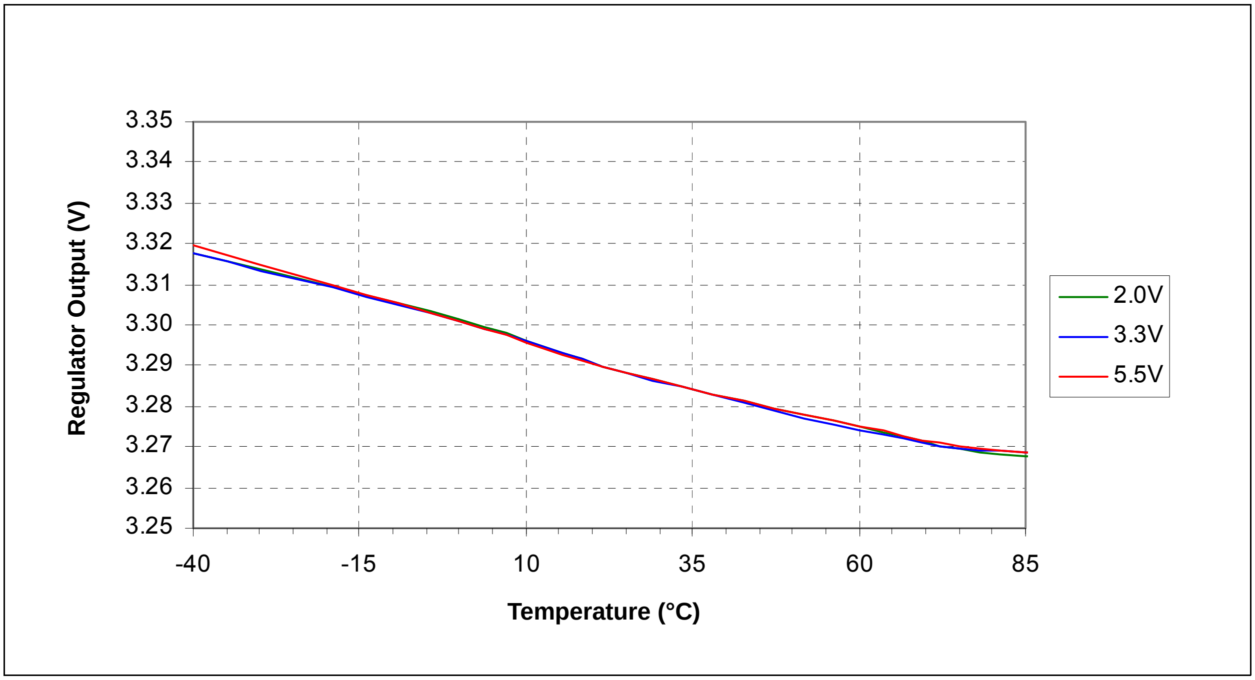


FIGURE 30-36: HLVD TRIP POINT VOLTAGE vs. TEMPERATURE (HLVDL<3:0> = 0000, PIC24F32KA304 FAMILY DEVICES ONLY

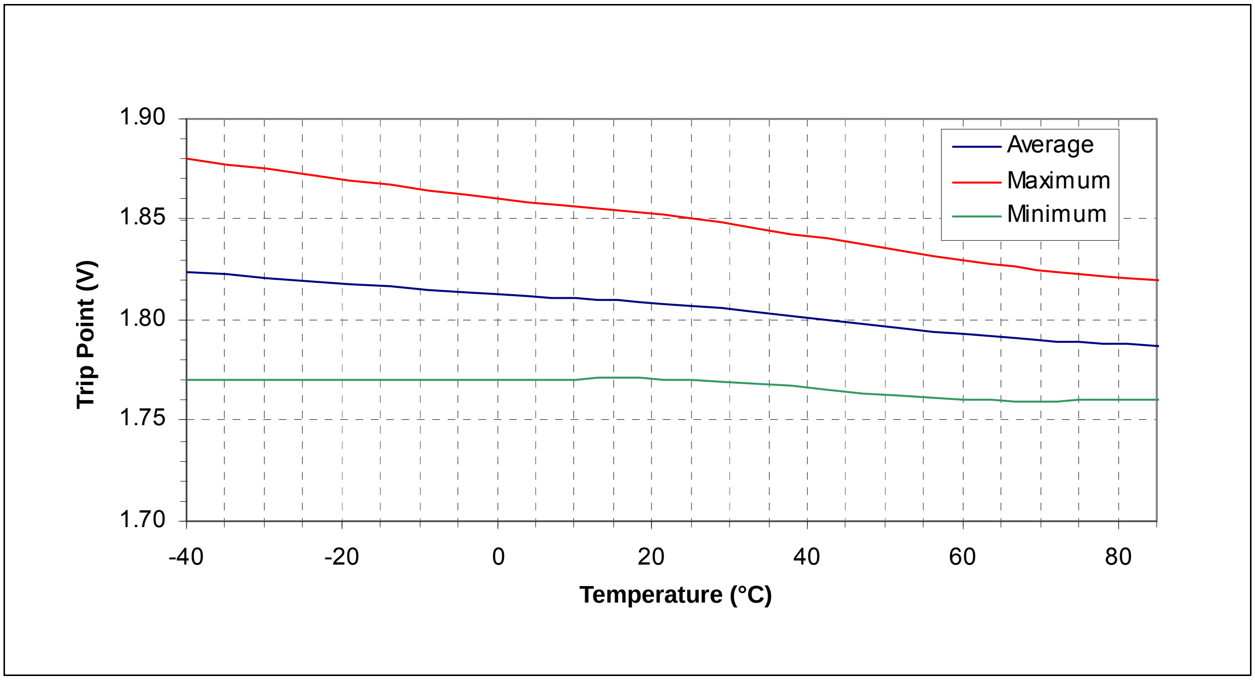


FIGURE 30-37: TEMPERATURE SENSOR DIODE VOLTAGE vs. TEMPERATURE ($2.0V \leq V_{DD} \leq 5.5V$)

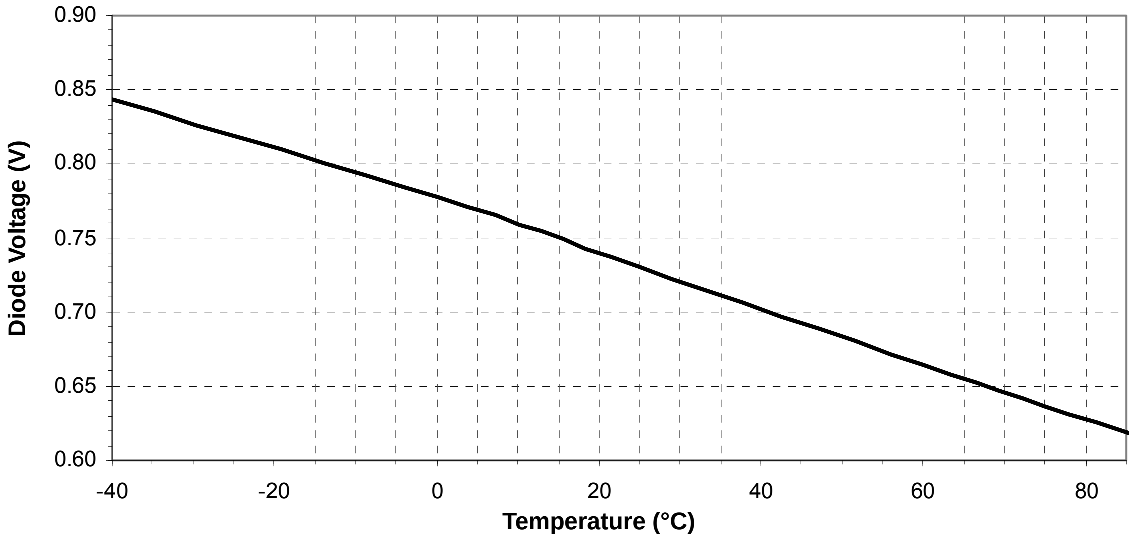


FIGURE 30-38: CTMU OUTPUT CURRENT vs. TEMPERATURE (IRNG<1:0> = 01, $2.0V \leq V_{DD} \leq 5.5V$)

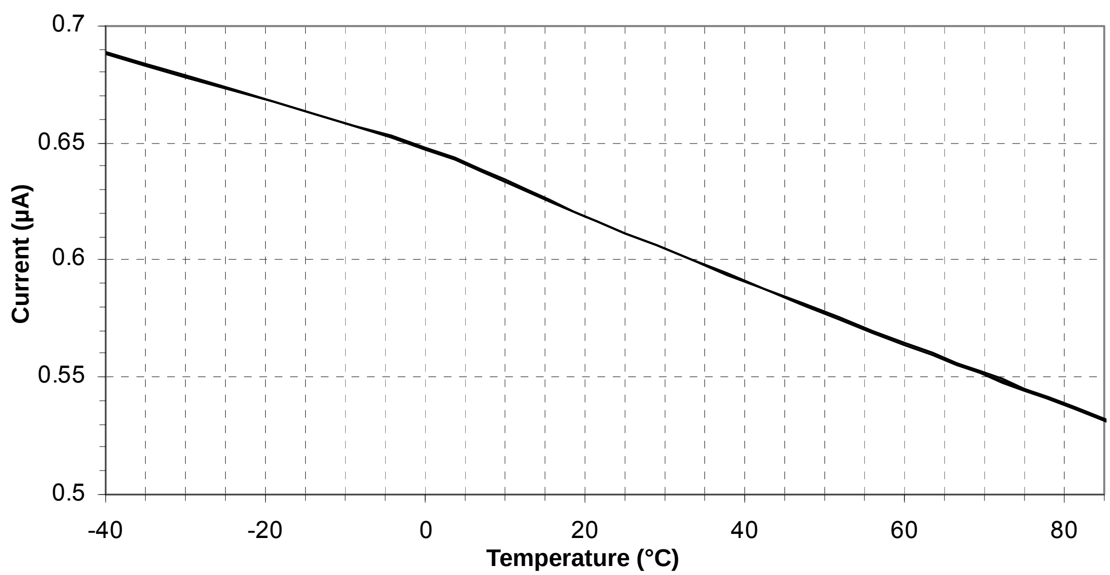
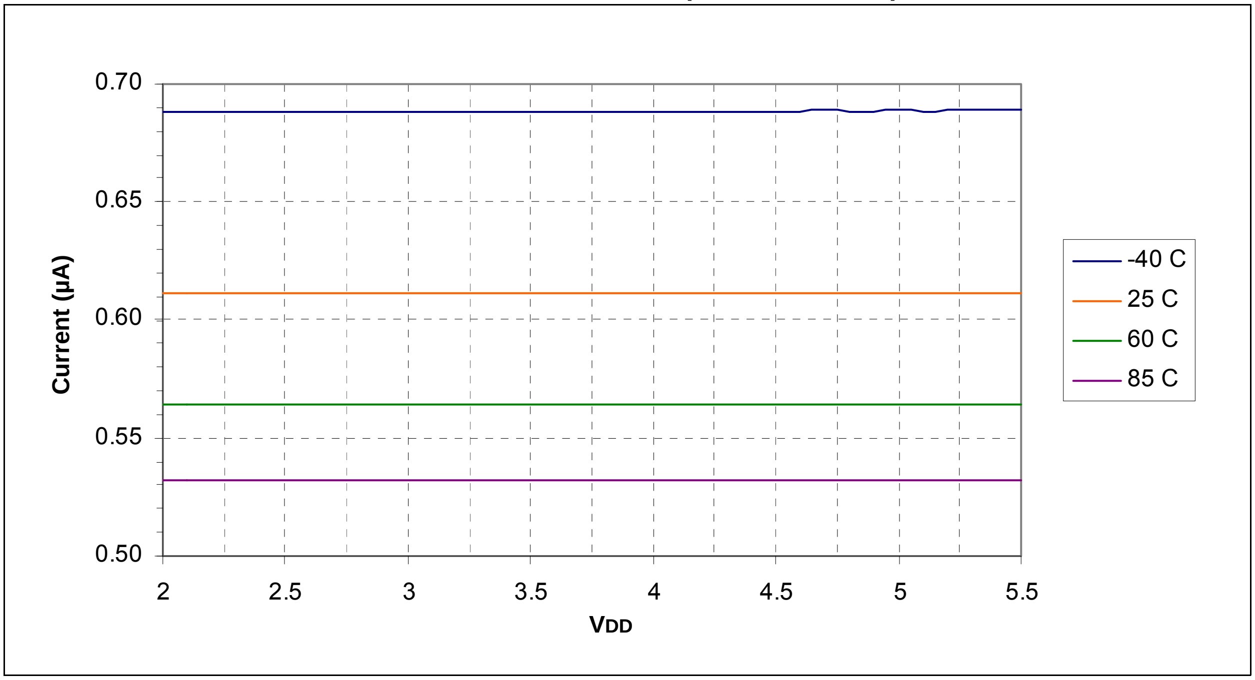


FIGURE 30-39: CTMU OUTPUT CURRENT vs. VDD (IRNG<1:0> = 01)



30.2 Characteristics for Extended Temperature Devices (-40°C to +125°C)

Note:	Data for VDD levels greater than 3.3V are applicable to PIC24FV32KA304 family devices only.

FIGURE 30-40: TYPICAL AND MAXIMUM IIDLE vs. VDD (FRC MODE)

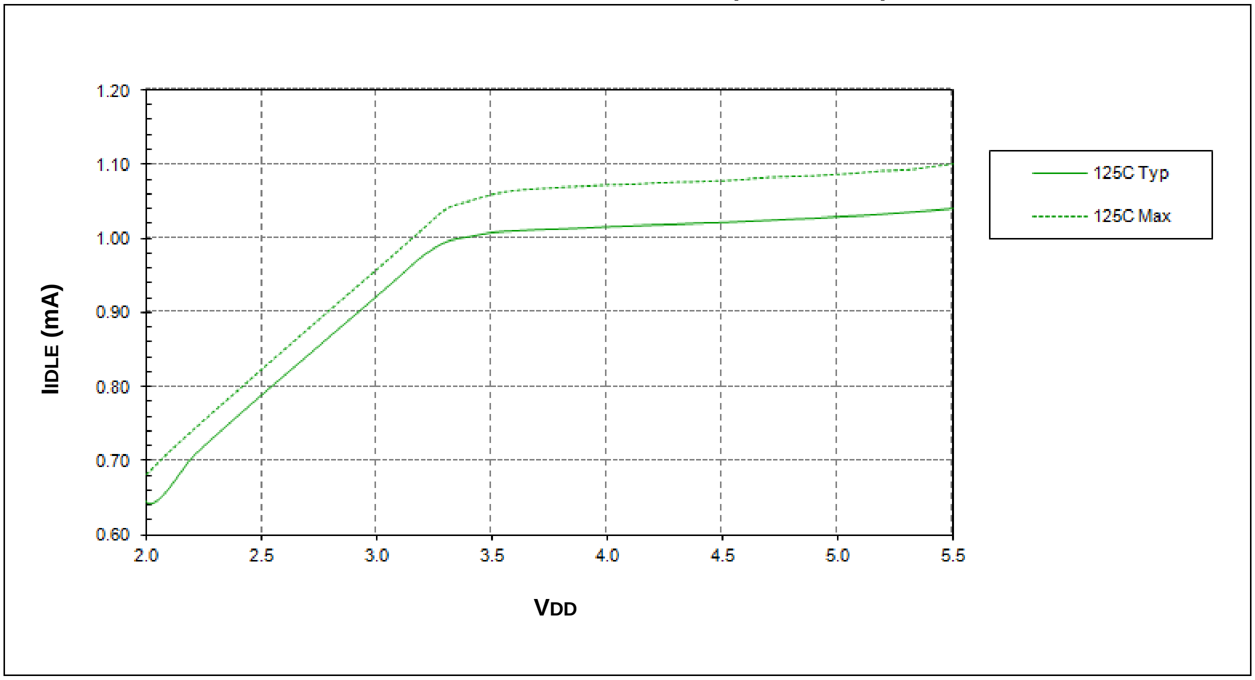


FIGURE 30-41: TYPICAL AND MAXIMUM IIDLE vs. TEMPERATURE (FRC MODE)

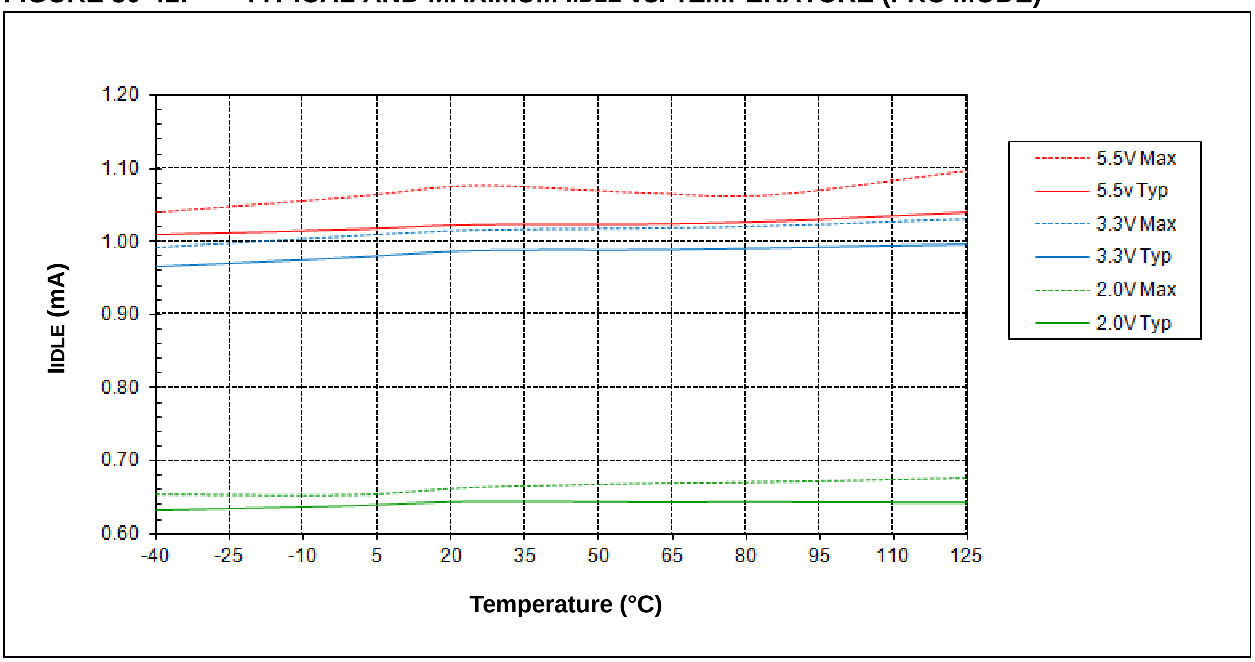


FIGURE 30-42: TYPICAL AND MAXIMUM IPD vs. VDD

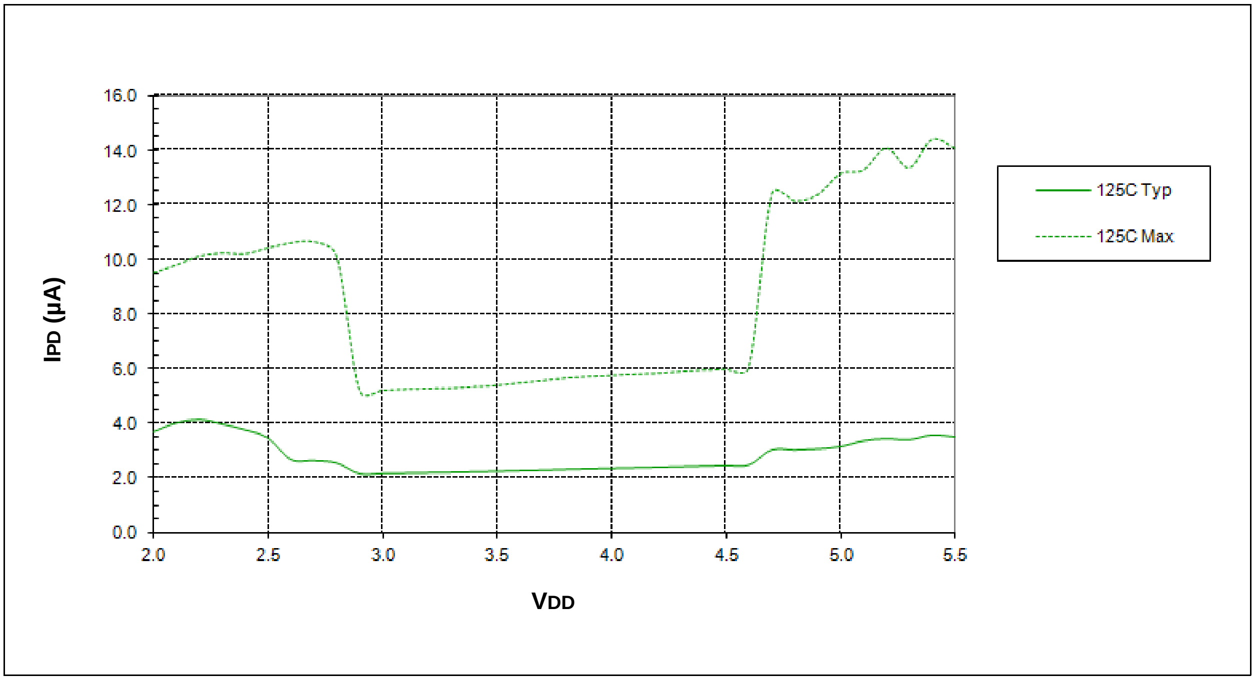


FIGURE 30-43: TYPICAL AND MAXIMUM IPD vs. TEMPERATURE

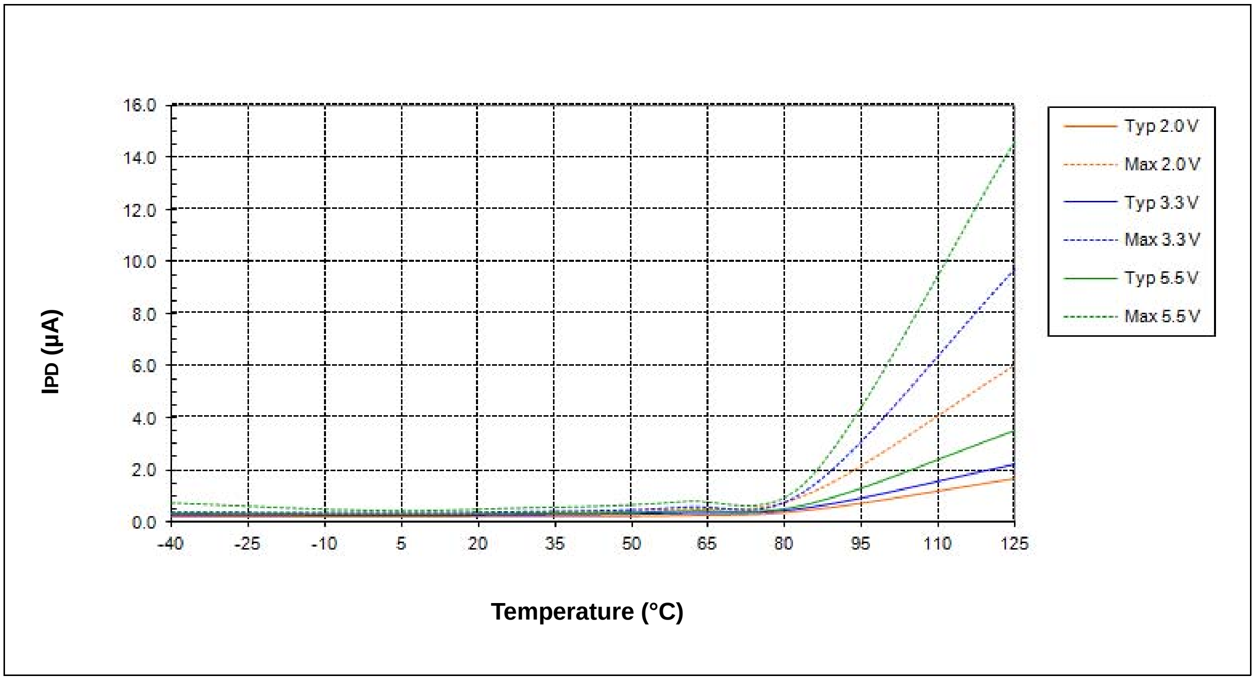


FIGURE 30-44: TYPICAL AND MAXIMUM IPD vs. VDD (DEEP SLEEP MODE)

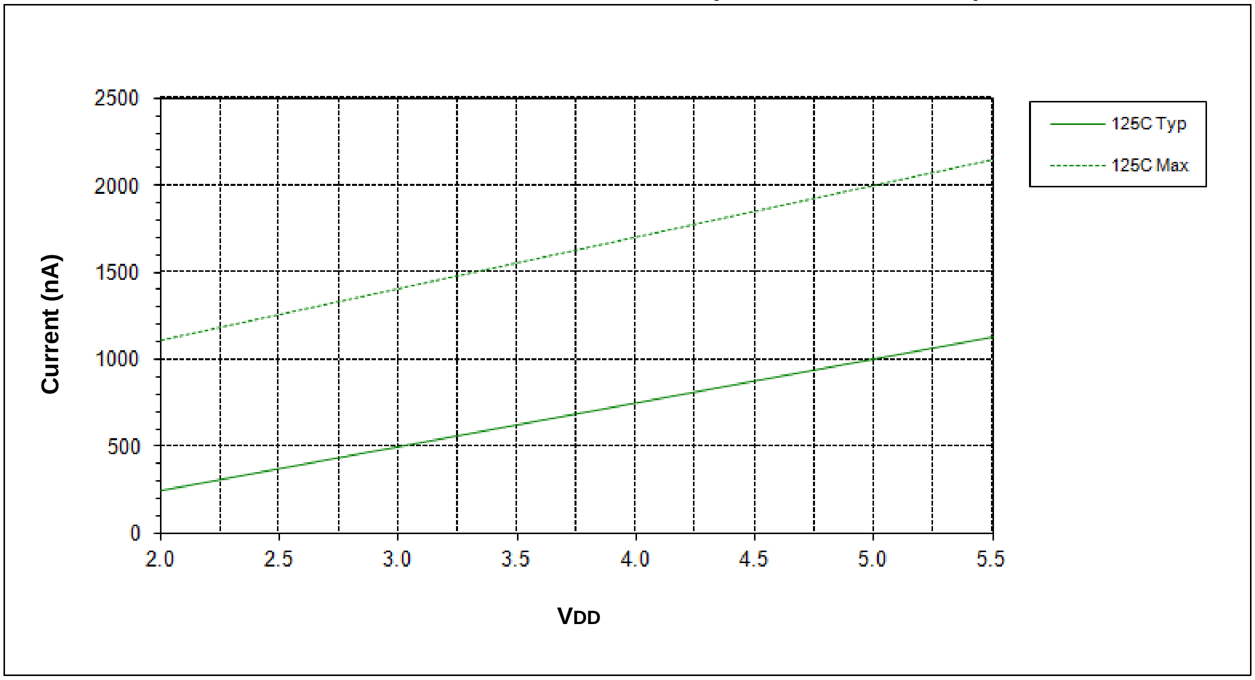


FIGURE 30-45: TYPICAL AND MAXIMUM IPD vs. TEMPERATURE (DEEP SLEEP MODE)

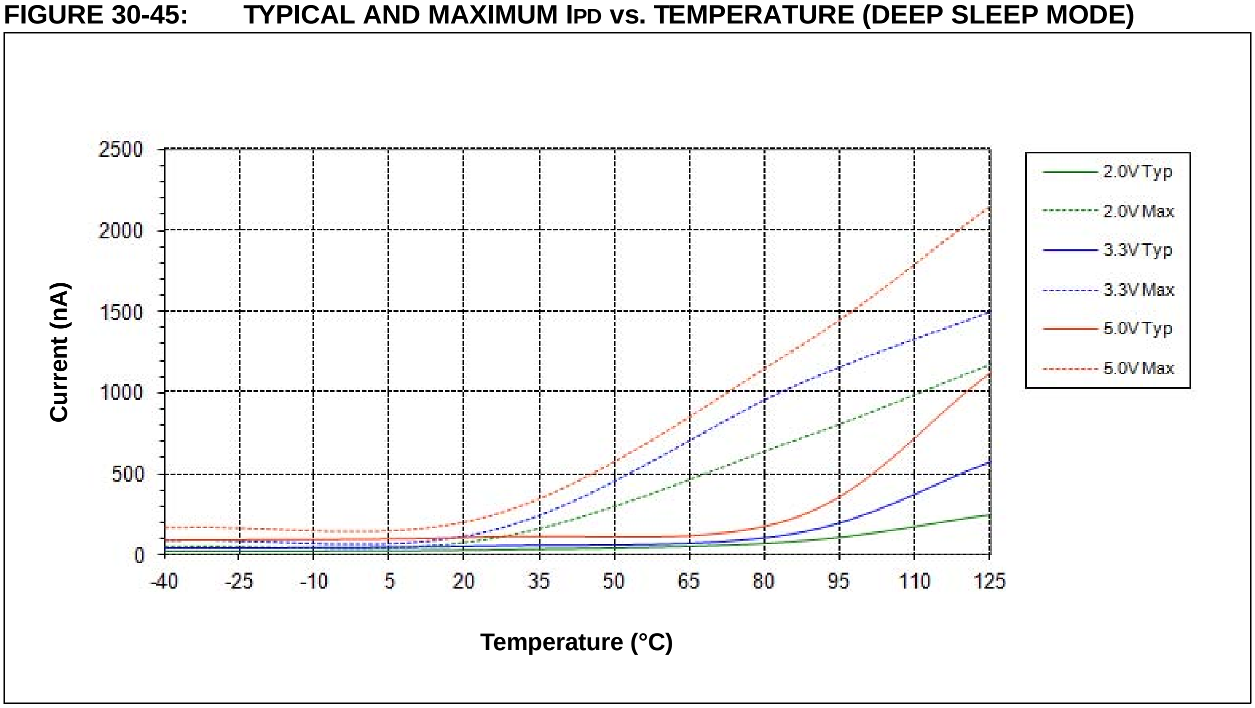


FIGURE 30-46: TYPICAL ΔIWDT vs. VDD

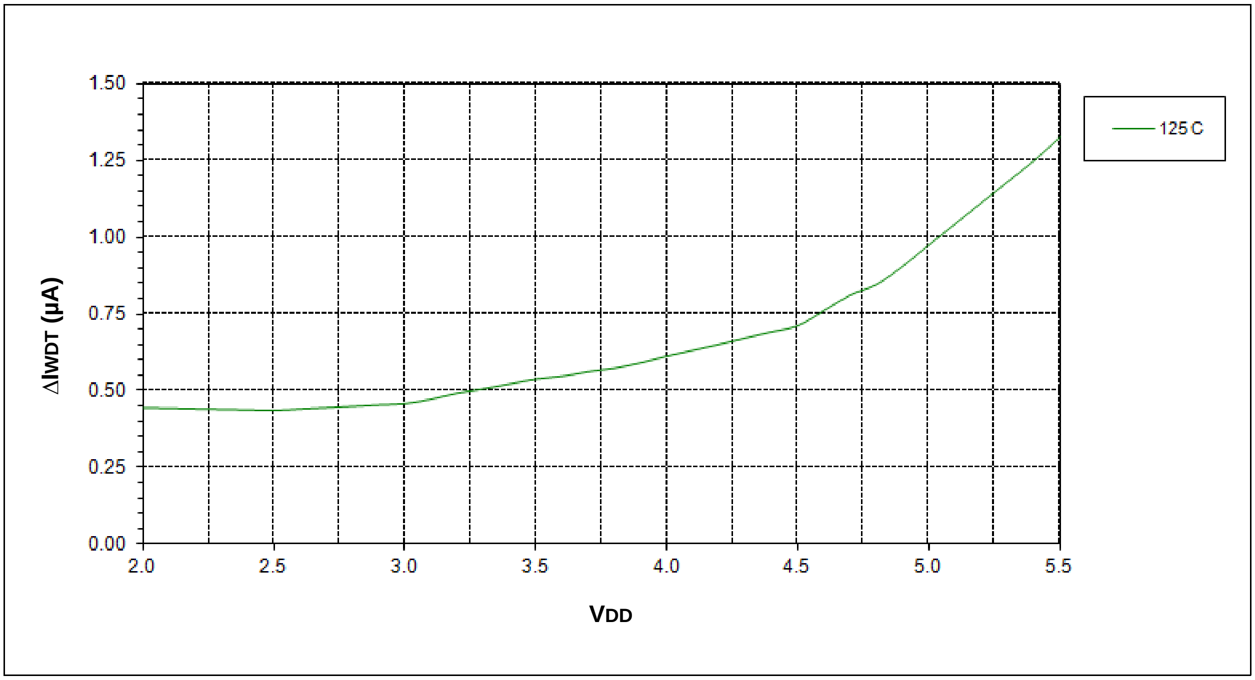


FIGURE 30-47: TYPICAL ΔIDSBOR vs. VDD

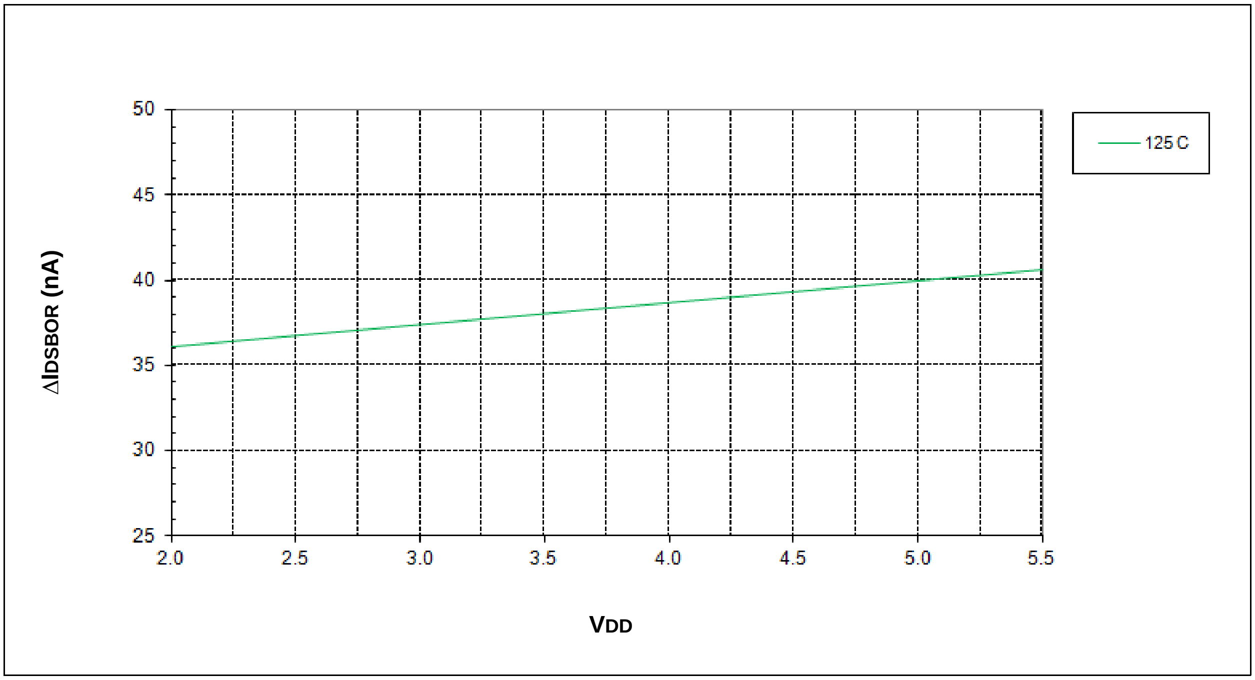


FIGURE 30-48: TYPICAL ΔIHLVD vs. VDD

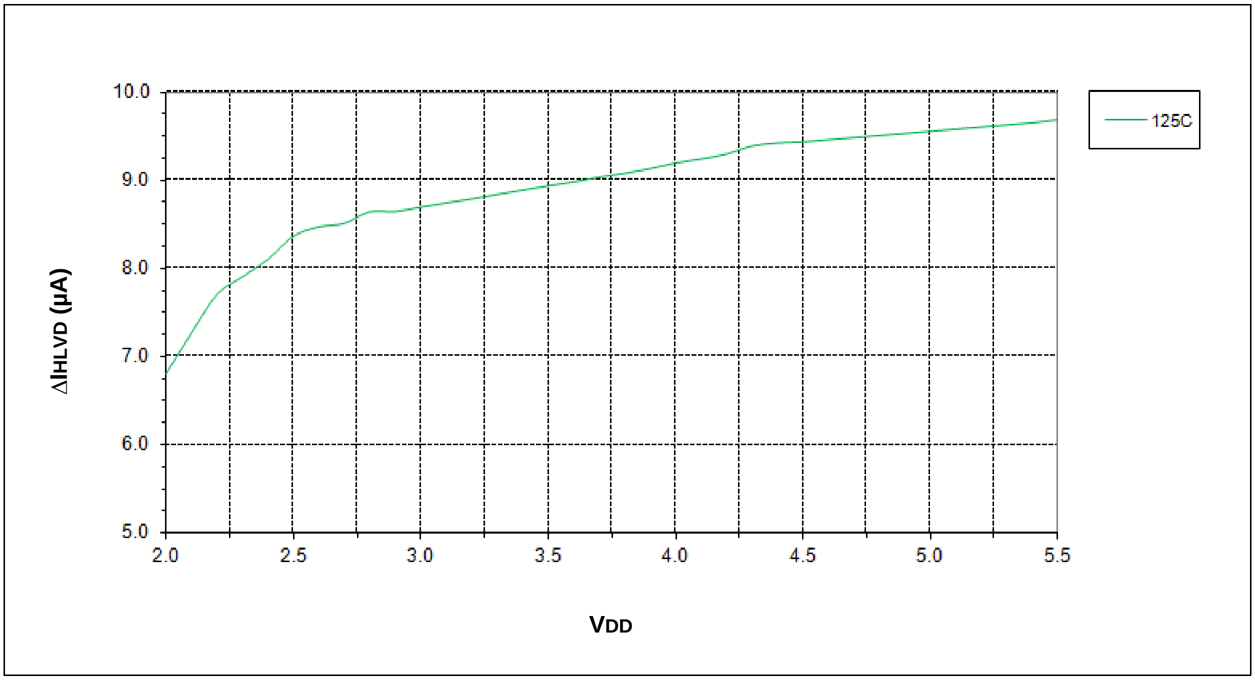


FIGURE 30-49: TYPICAL VOL vs. IOL (GENERAL I/O, 2.0V ≤ VDD ≤ 5.5V)

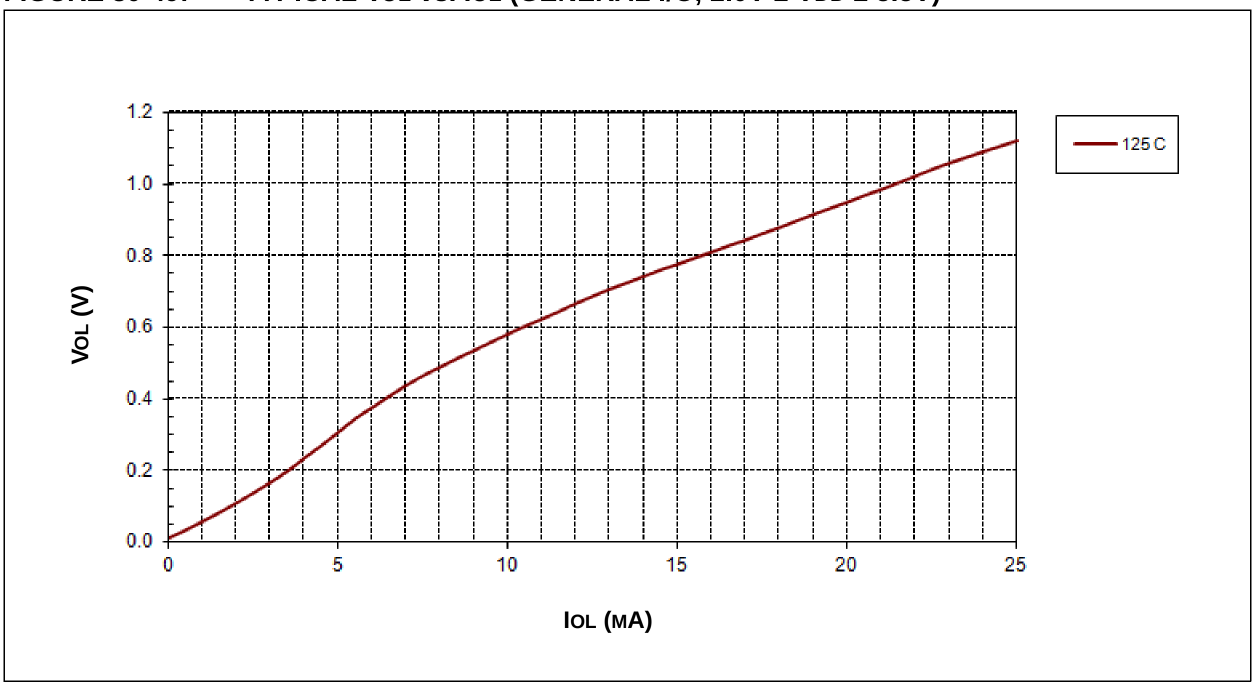


FIGURE 30-50: TYPICAL V_{OH} vs. I_{OH} (GENERAL I/O, AS A FUNCTION OF TEMPERATURE, 2.0V $\leq$ V_{DD} $\leq$ 5.5V)

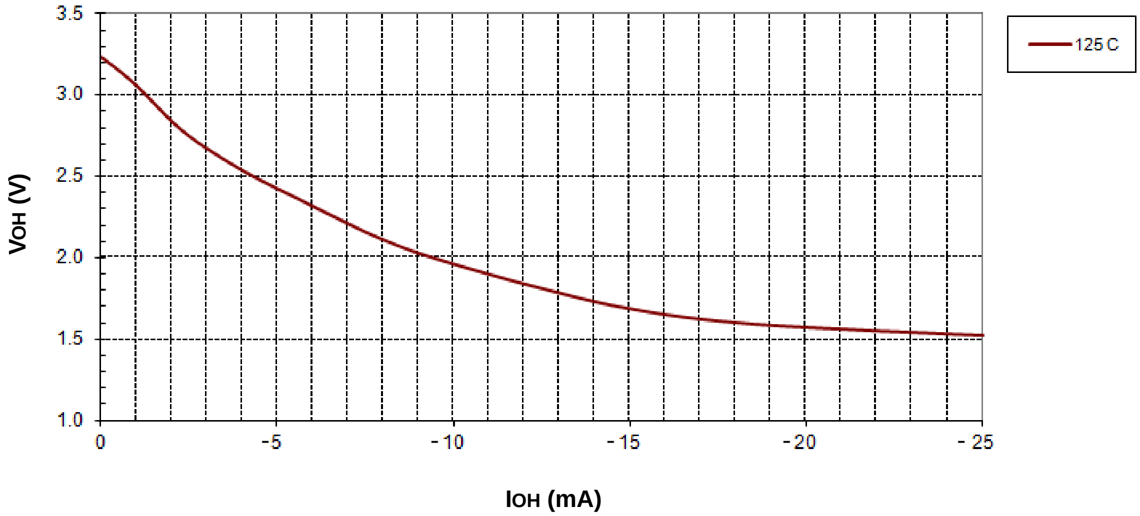


FIGURE 30-51: V_{IL}/V_{IH} vs. V_{DD} (GENERAL PURPOSE I/O, TEMPERATURES AS NOTED)

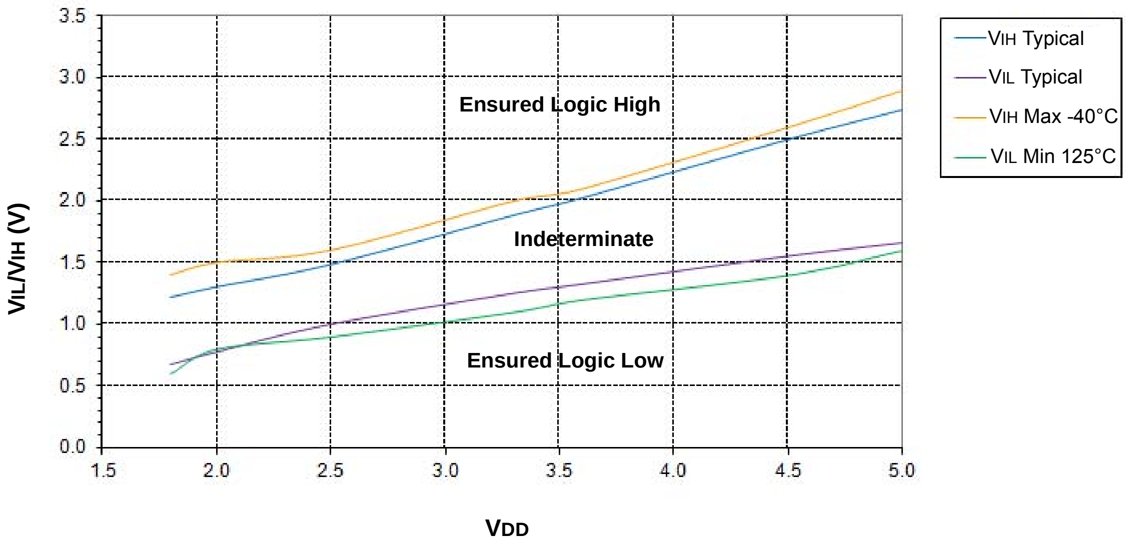


FIGURE 30-52: VIL/VIH vs. VDD (I²C, TEMPERATURES AS NOTED)

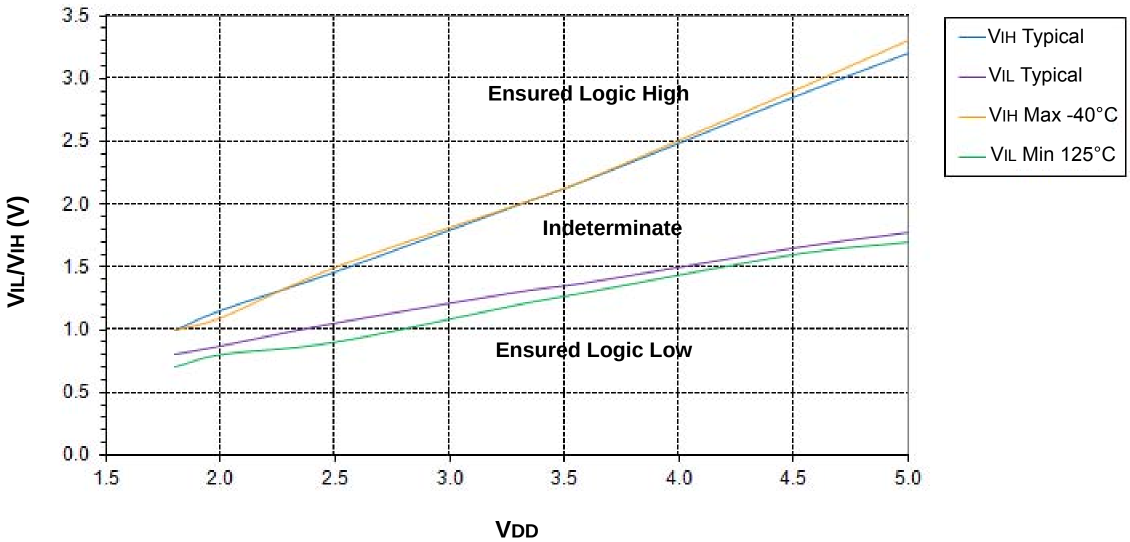


FIGURE 30-53: VIL/VIH vs. VDD (OSCO, TEMPERATURES AS NOTED)

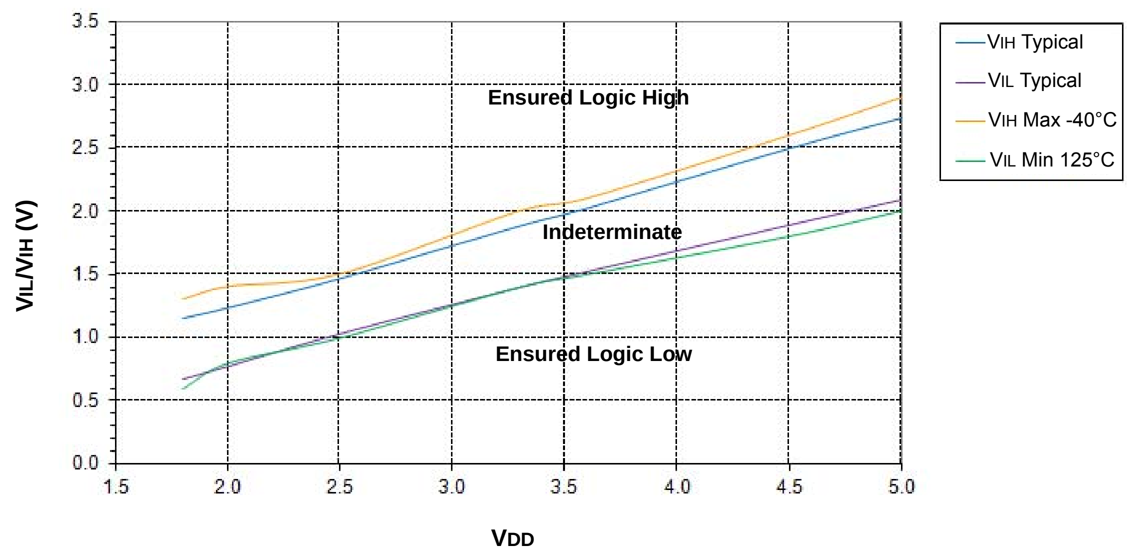


FIGURE 30-54: V_{IL}/V_{IH} vs. V_{DD} ($\overline{MCLR}$, TEMPERATURES AS NOTED)

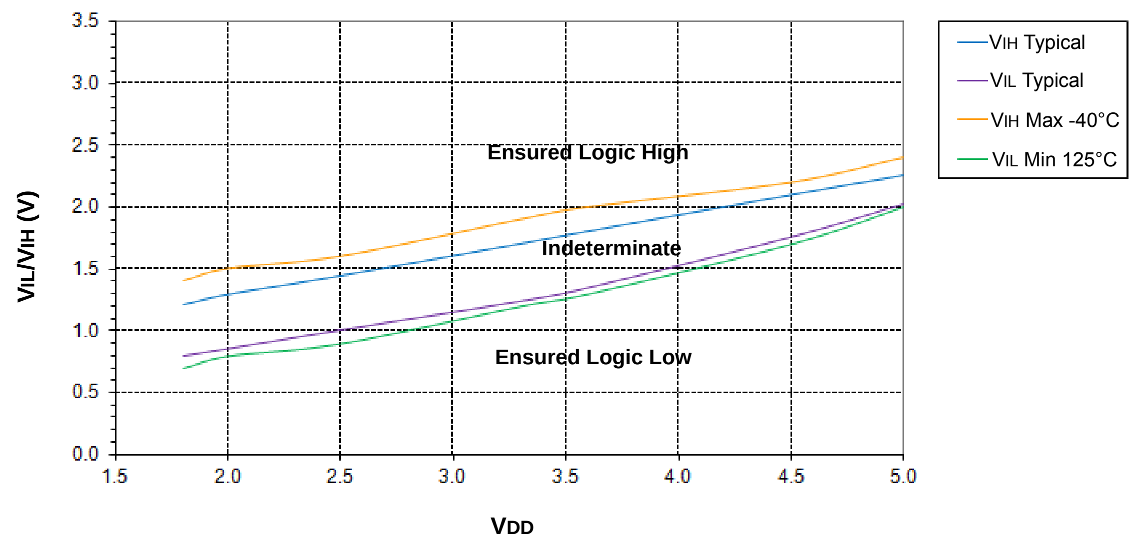


FIGURE 30-55: TYPICAL BAND GAP VOLTAGE vs. TEMPERATURE (2.0V ≤ V_{DD} ≤ 5.5V)

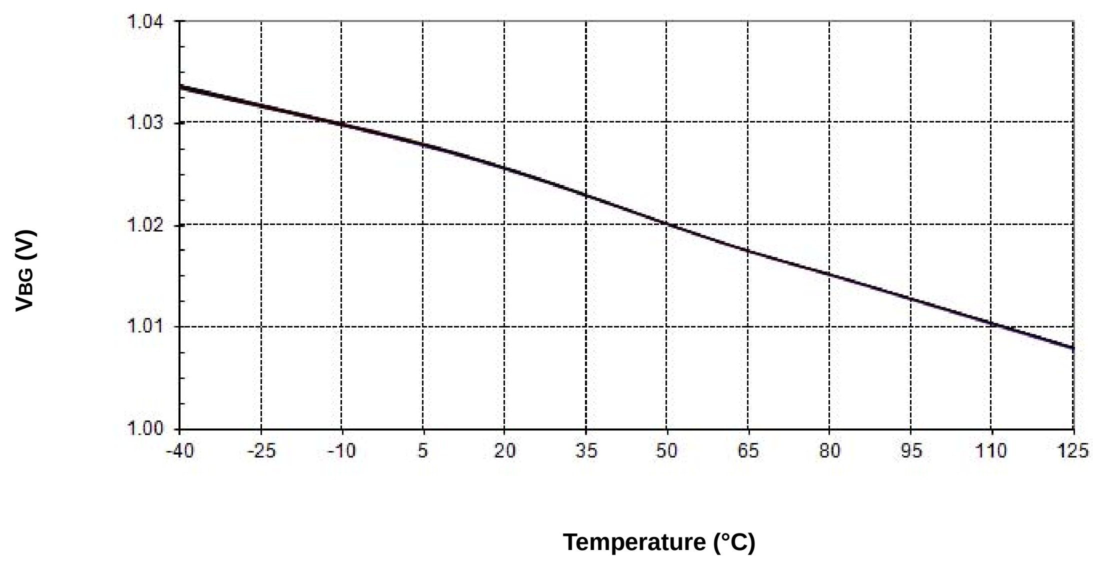
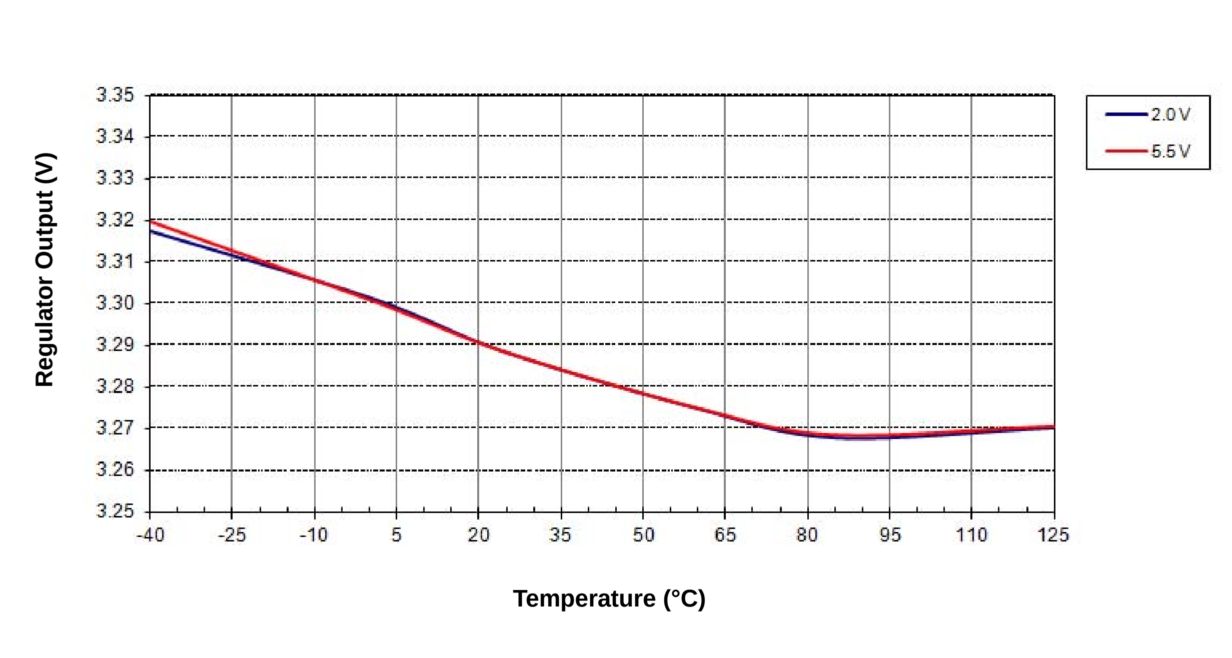


FIGURE 30-56: TYPICAL VOLTAGE REGULATOR OUTPUT vs. TEMPERATURE

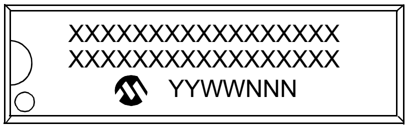


NOTES:

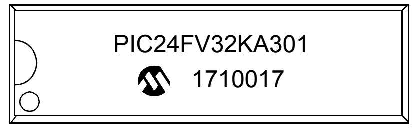
31.0 PACKAGING INFORMATION

31.1 Package Marking Information

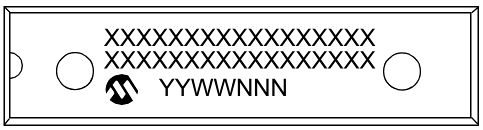
20-Lead PDIP (300 mil)



Example

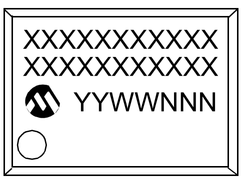


28-Lead SPDIP (.300")

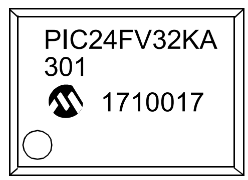


Example

20-Lead SSOP (5.30 mm)



Example



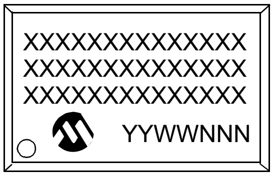
28-Lead SSOP (5.30 mm)

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

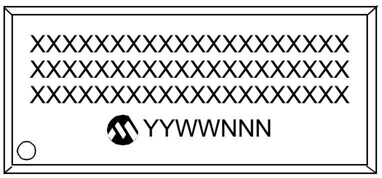
Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

20-Lead SOIC (7.50 mm)



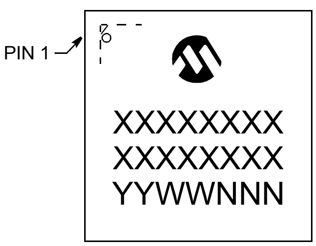
Example

28-Lead SOIC (7.50 mm)

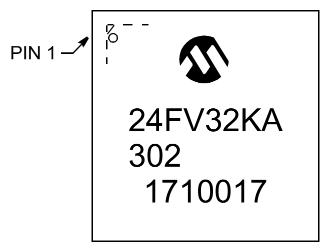


Example

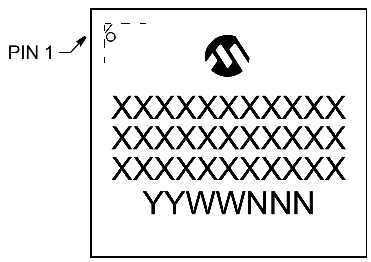
28-Lead QFN (6x6 mm)



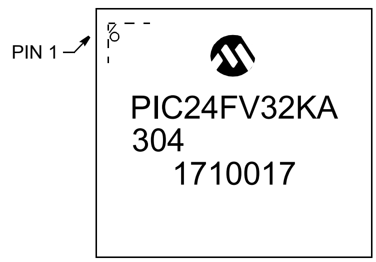
Example



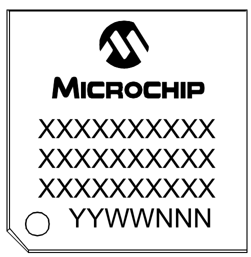
44-Lead QFN (8x8x0.9 mm)



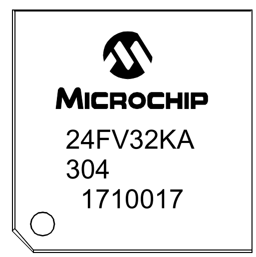
Example



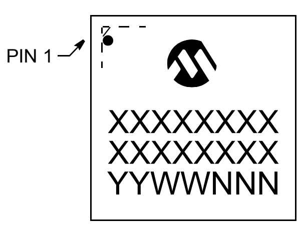
44-Lead TQFP (10x10x1 mm)



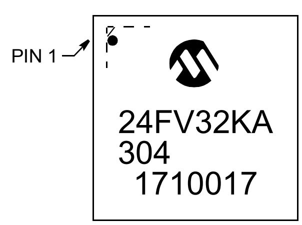
Example



48-Lead UQFN (6x6x0.5 mm)



Example

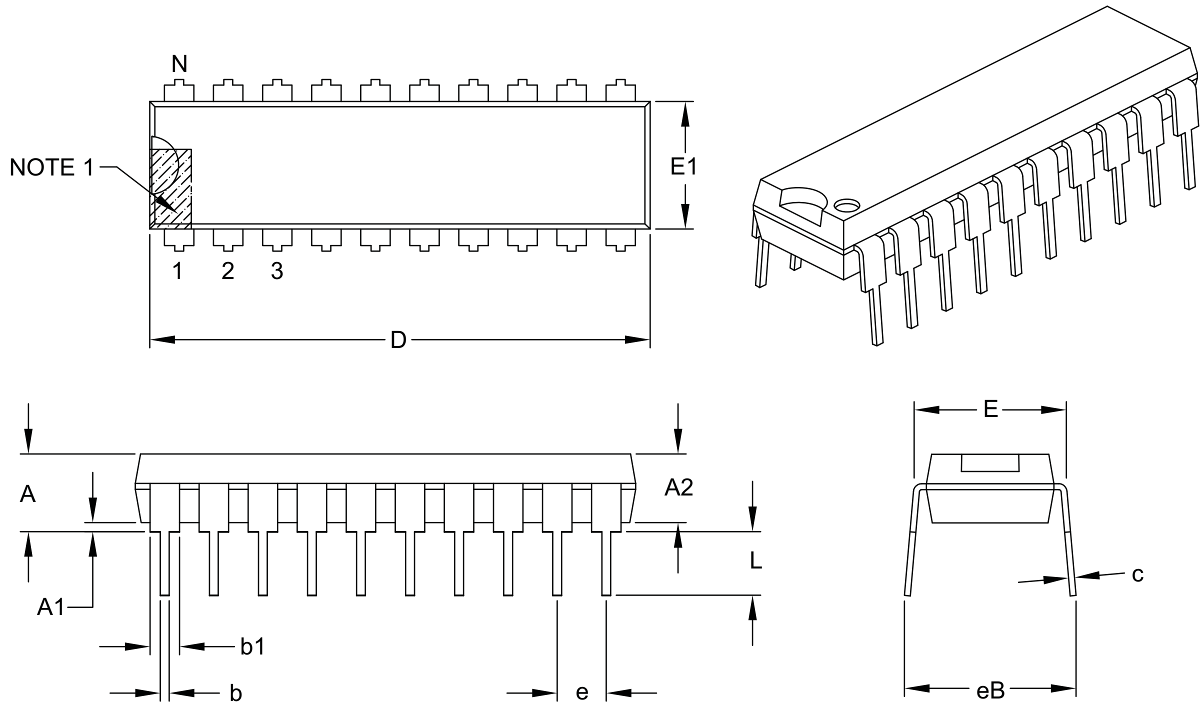


31.2 Package Details

The following sections give the technical details of the packages.

20-Lead Plastic Dual In-Line (P) – 300 mil Body [PDIP]

Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Units		INCHES		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		20	
Pitch	e		.100 BSC	
Top to Seating Plane	A	–	–	.210
Molded Package Thickness	A2	.115	.130	.195
Base to Seating Plane	A1	.015	–	–
Shoulder to Shoulder Width	E	.300	.310	.325
Molded Package Width	E1	.240	.250	.280
Overall Length	D	.980	1.030	1.060
Tip to Seating Plane	L	.115	.130	.150
Lead Thickness	c	.008	.010	.015
Upper Lead Width	b1	.045	.060	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	–	–	.430

Notes:

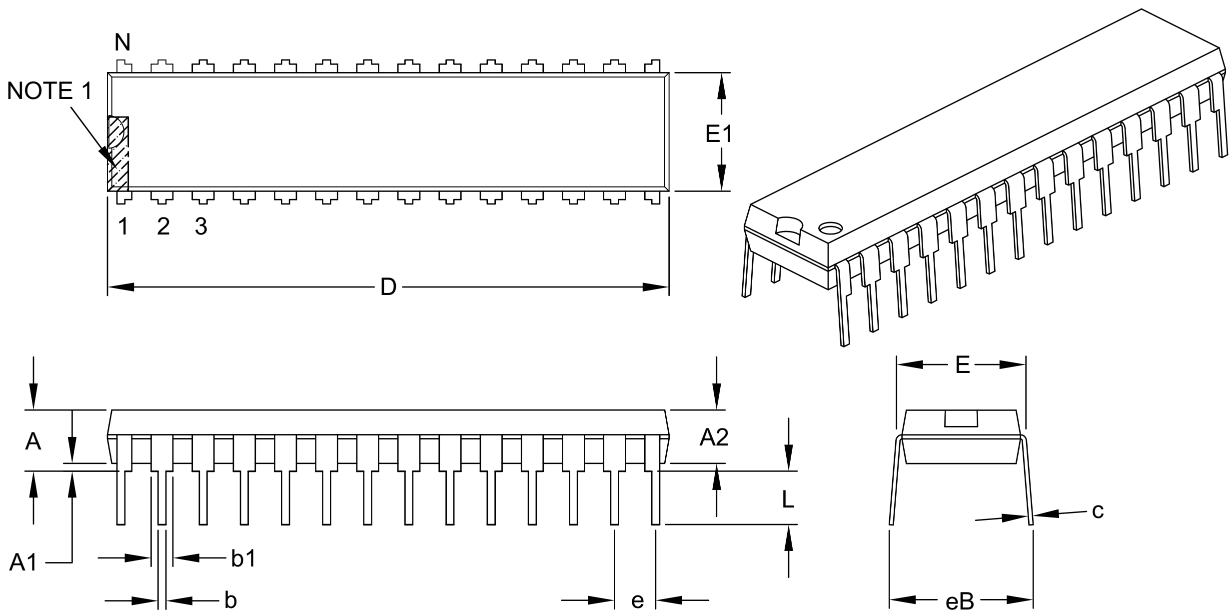
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-019B

28-Lead Skinny Plastic Dual In-Line (SP) – 300 mil Body [SPDIP]

Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Units		INCHES		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N	28		
Pitch	e	.100 BSC		
Top to Seating Plane	A	–	–	.200
Molded Package Thickness	A2	.120	.135	.150
Base to Seating Plane	A1	.015	–	–
Shoulder to Shoulder Width	E	.290	.310	.335
Molded Package Width	E1	.240	.285	.295
Overall Length	D	1.345	1.365	1.400
Tip to Seating Plane	L	.110	.130	.150
Lead Thickness	c	.008	.010	.015
Upper Lead Width	b1	.040	.050	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	–	–	.430

Notes:

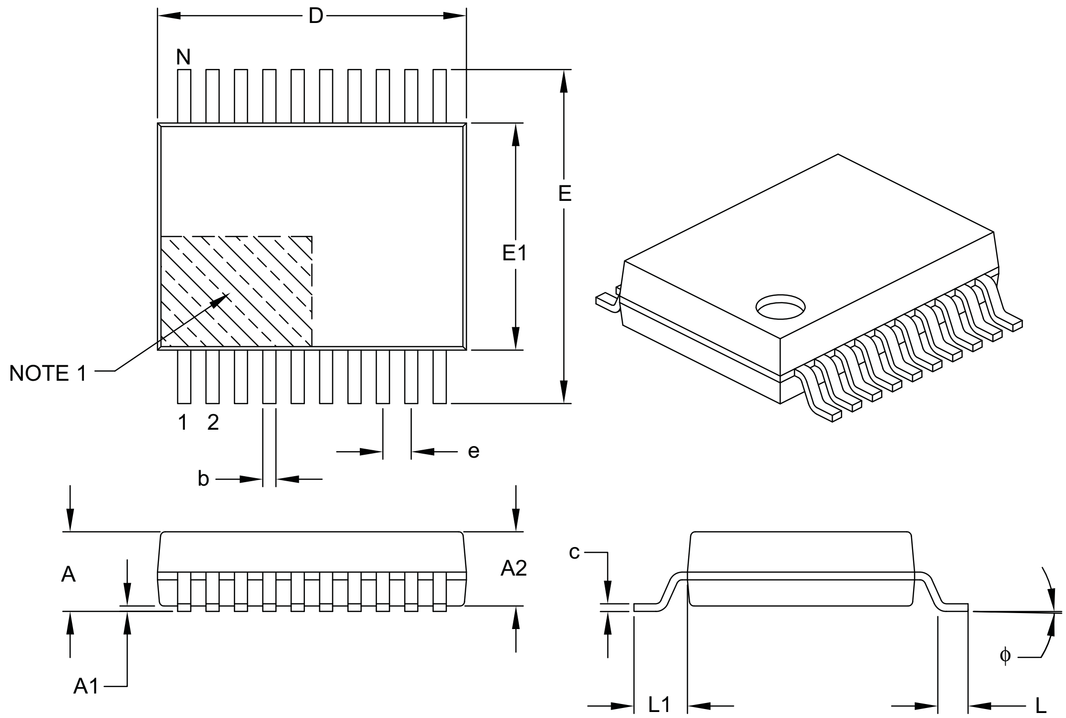
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-070B

20-Lead Plastic Shrink Small Outline (SS) – 5.30 mm Body [SSOP]

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		20	
Pitch	e		0.65 BSC	
Overall Height	A	–	–	2.00
Molded Package Thickness	A2	1.65	1.75	1.85
Standoff	A1	0.05	–	–
Overall Width	E	7.40	7.80	8.20
Molded Package Width	E1	5.00	5.30	5.60
Overall Length	D	6.90	7.20	7.50
Foot Length	L	0.55	0.75	0.95
Footprint	L1		1.25 REF	
Lead Thickness	c	0.09	–	0.25
Foot Angle	φ	0°	4°	8°
Lead Width	b	0.22	–	0.38

Notes:

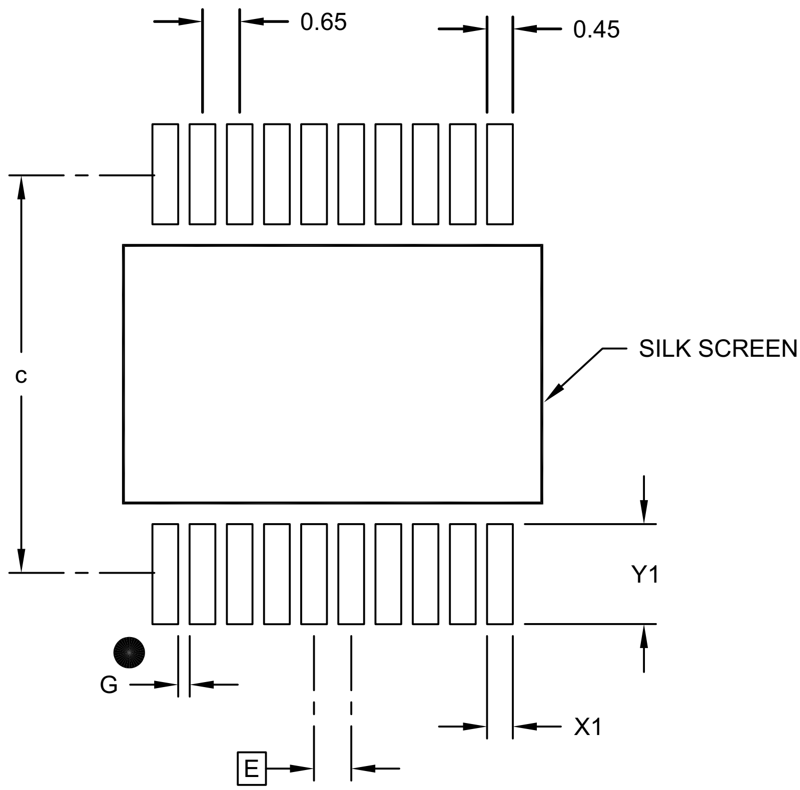
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.20 mm per side.
3. Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-072B

20-Lead Plastic Shrink Small Outline (SS) - 5.30 mm Body [SSOP]

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	0.65 BSC		
Contact Pad Spacing	C		7.20	
Contact Pad Width (X20)	X1			0.45
Contact Pad Length (X20)	Y1			1.75
Distance Between Pads	G	0.20		

Notes:

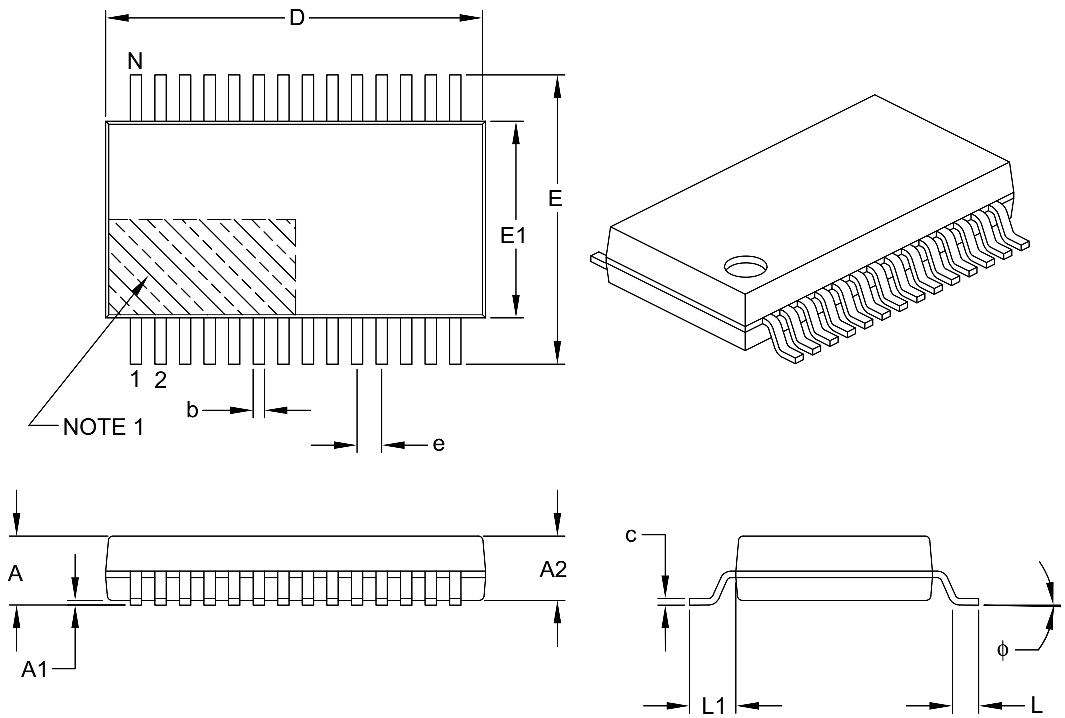
1. Dimensioning and tolerancing per ASME Y14.5M

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2072B

28-Lead Plastic Shrink Small Outline (SS) – 5.30 mm Body [SSOP]

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		28	
Pitch	e		0.65 BSC	
Overall Height	A	–	–	2.00
Molded Package Thickness	A2	1.65	1.75	1.85
Standoff	A1	0.05	–	–
Overall Width	E	7.40	7.80	8.20
Molded Package Width	E1	5.00	5.30	5.60
Overall Length	D	9.90	10.20	10.50
Foot Length	L	0.55	0.75	0.95
Footprint	L1		1.25 REF	
Lead Thickness	c	0.09	–	0.25
Foot Angle	φ	0°	4°	8°
Lead Width	b	0.22	–	0.38

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.20 mm per side.
3. Dimensioning and tolerancing per ASME Y14.5M.

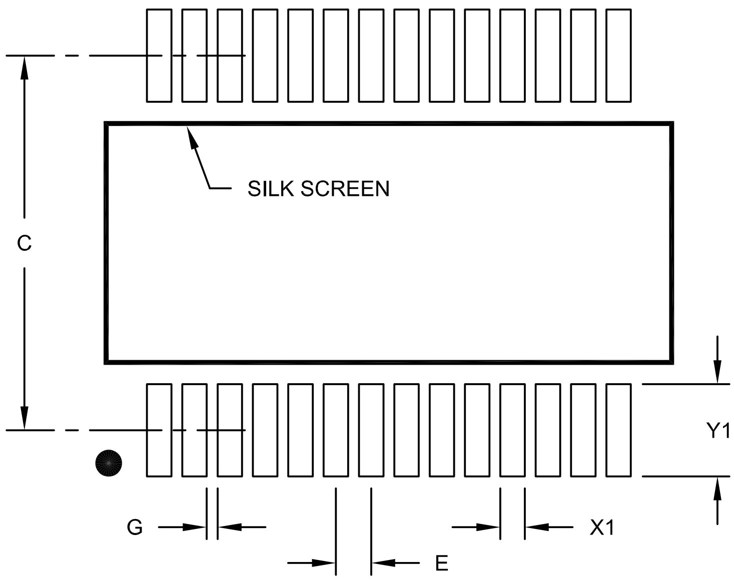
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-073B

28-Lead Plastic Shrink Small Outline (SS) - 5.30 mm Body [SSOP]

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	0.65 BSC		
Contact Pad Spacing	C		7.20	
Contact Pad Width (X28)	X1			0.45
Contact Pad Length (X28)	Y1			1.75
Distance Between Pads	G	0.20		

Notes:

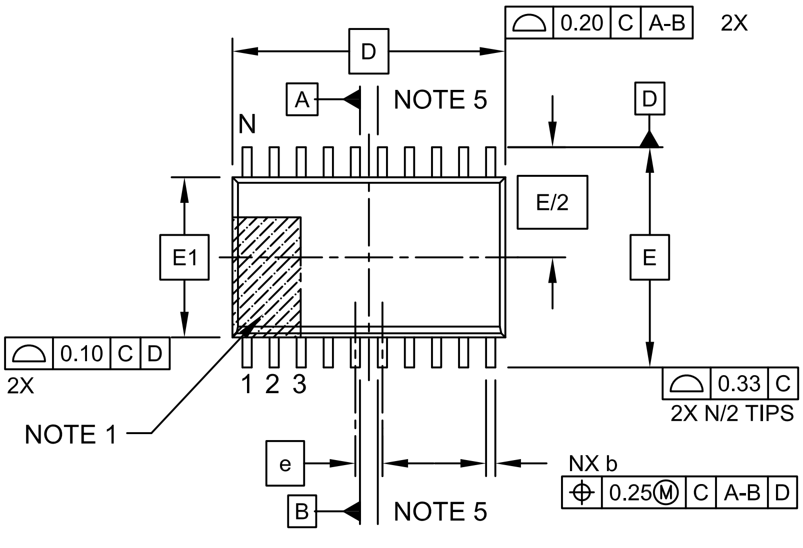
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

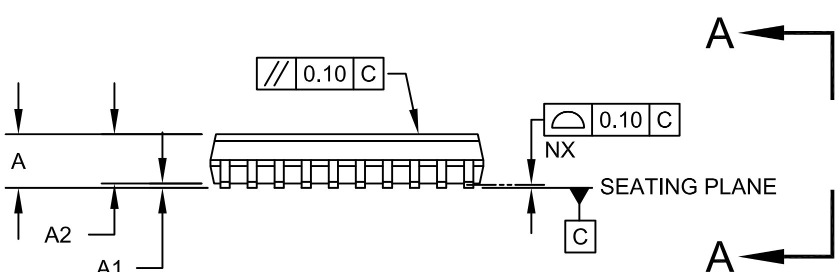
Microchip Technology Drawing No. C04-2073A

20-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

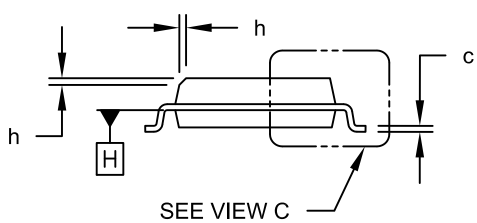
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



TOP VIEW



SIDE VIEW

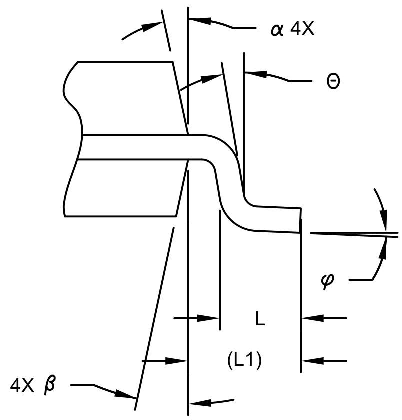


VIEW A-A

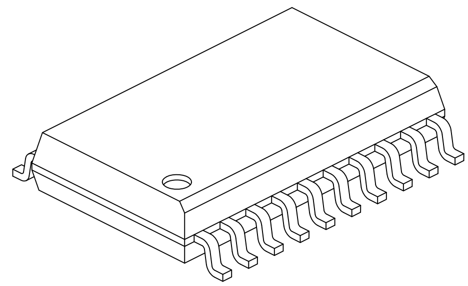
Microchip Technology Drawing C04-094C Sheet 1 of 2

20-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



VIEW C



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N	20		
Pitch	e	1.27 BSC		
Overall Height	A	-	-	2.65
Molded Package Thickness	A2	2.05	-	-
Standoff §	A1	0.10	-	0.30
Overall Width	E	10.30 BSC		
Molded Package Width	E1	7.50 BSC		
Overall Length	D	12.80 BSC		
Chamfer (Optional)	h	0.25	-	0.75
Foot Length	L	0.40	-	1.27
Footprint	L1	1.40 REF		
Lead Angle	Θ	0°	-	-
Foot Angle	φ	0°	-	8°
Lead Thickness	c	0.20	-	0.33
Lead Width	b	0.31	-	0.51
Mold Draft Angle Top	α	5°	-	15°
Mold Draft Angle Bottom	β	5°	-	15°

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic
3. Dimension D does not include mold flash, protrusions or gate burrs, which shall not exceed 0.15 mm per end. Dimension E1 does not include interlead flash or protrusion, which shall not exceed 0.25 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M

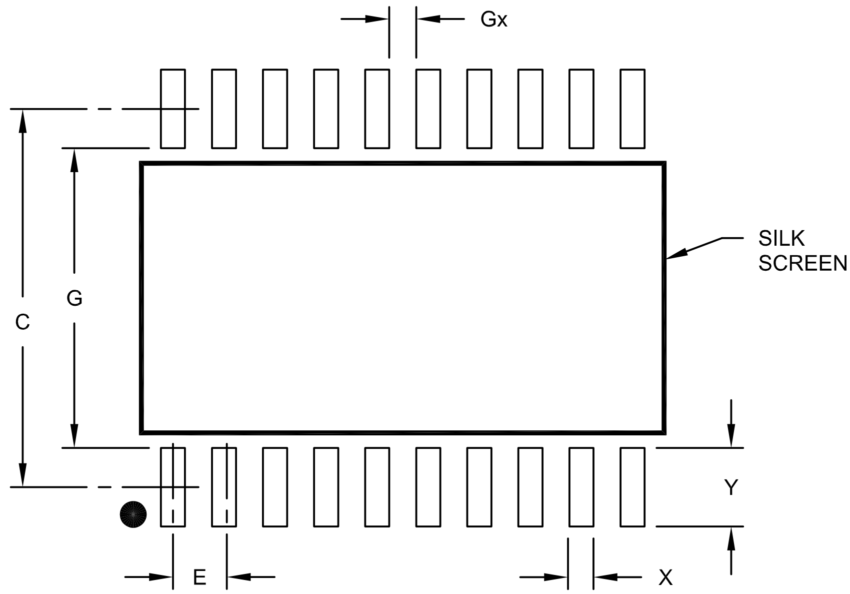
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.
5. Datums A & B to be determined at Datum H.

Microchip Technology Drawing No. C04-094C Sheet 2 of 2

20-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

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		1.27 BSC	
Contact Pad Spacing	C		9.40	
Contact Pad Width (X20)	X			0.60
Contact Pad Length (X20)	Y			1.95
Distance Between Pads	Gx	0.67		
Distance Between Pads	G	7.45		

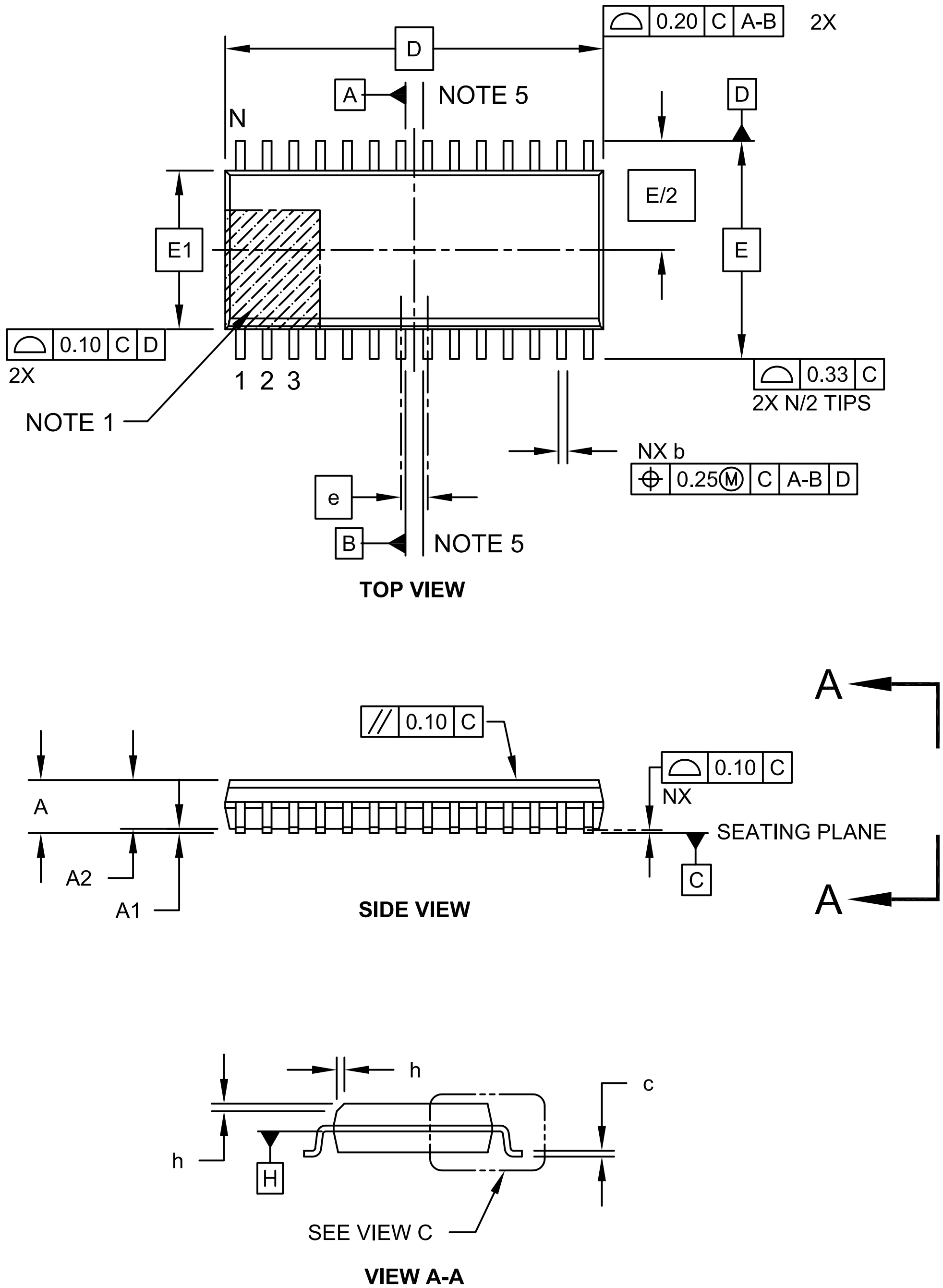
Notes:

1. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2094A

28-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

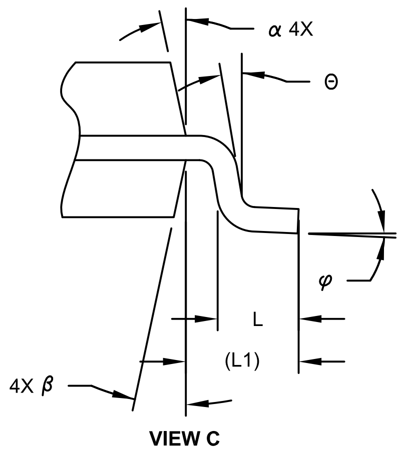
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging

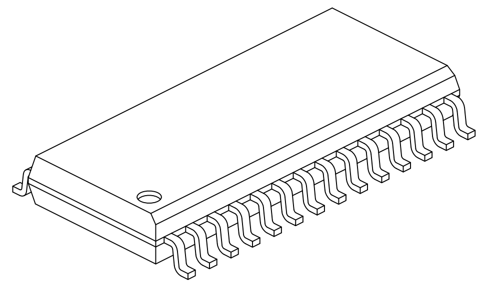


Microchip Technology Drawing C04-052C Sheet 1 of 2

28-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

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		28	
Pitch	e		1.27 BSC	
Overall Height	A	-	-	2.65
Molded Package Thickness	A2	2.05	-	-
Standoff §	A1	0.10	-	0.30
Overall Width	E		10.30 BSC	
Molded Package Width	E1		7.50 BSC	
Overall Length	D		17.90 BSC	
Chamfer (Optional)	h	0.25	-	0.75
Foot Length	L	0.40	-	1.27
Footprint	L1		1.40 REF	
Lead Angle	Θ	0°	-	-
Foot Angle	φ	0°	-	8°
Lead Thickness	c	0.18	-	0.33
Lead Width	b	0.31	-	0.51
Mold Draft Angle Top	α	5°	-	15°
Mold Draft Angle Bottom	β	5°	-	15°

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic
3. Dimension D does not include mold flash, protrusions or gate burrs, which shall not exceed 0.15 mm per end. Dimension E1 does not include interlead flash or protrusion, which shall not exceed 0.25 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M

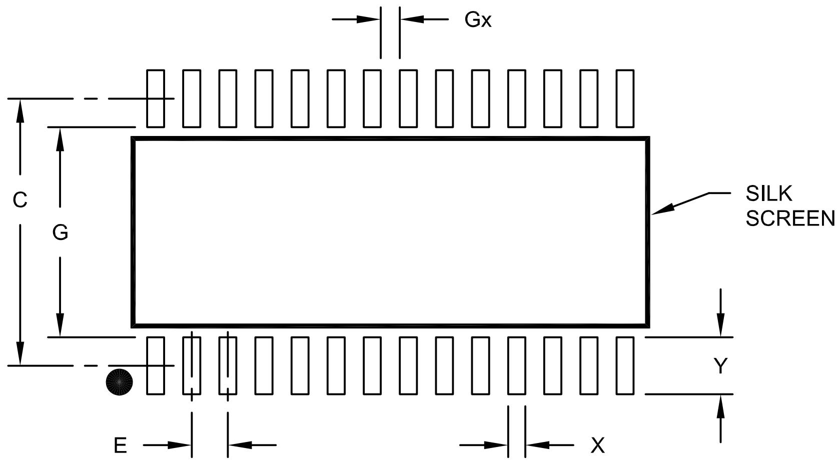
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.
5. Datums A & B to be determined at Datum H.

Microchip Technology Drawing C04-052C Sheet 2 of 2

28-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

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		1.27 BSC	
Contact Pad Spacing	C		9.40	
Contact Pad Width (X28)	X			0.60
Contact Pad Length (X28)	Y			2.00
Distance Between Pads	Gx	0.67		
Distance Between Pads	G	7.40		

Notes:

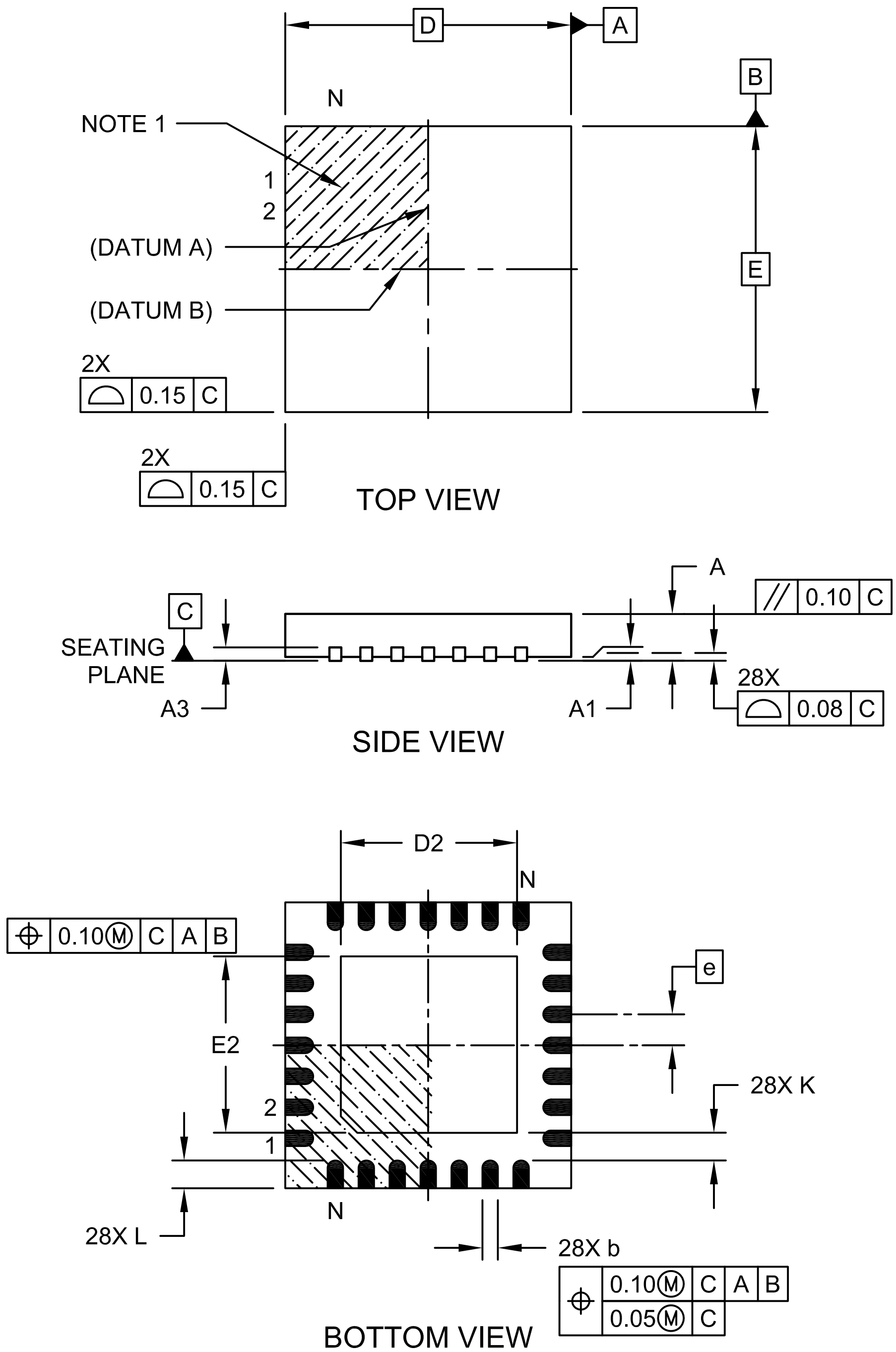
1. Dimensioning and tolerancing per ASME Y14.5M

    BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2052A

28-Lead Plastic Quad Flat, No Lead Package (ML) - 6x6 mm Body [QFN] With 0.55 mm Terminal Length

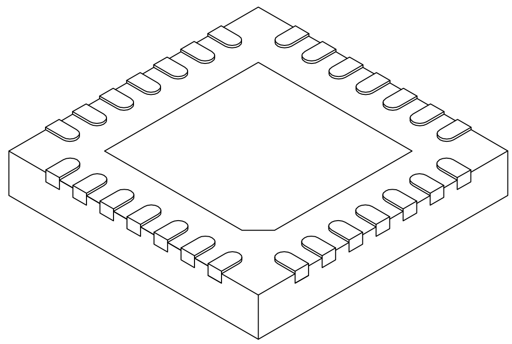
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Microchip Technology Drawing C04-105C Sheet 1 of 2

28-Lead Plastic Quad Flat, No Lead Package (ML) - 6x6 mm Body [QFN] With 0.55 mm Terminal Length

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		28	
Pitch	e		0.65 BSC	
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Terminal Thickness	A3		0.20 REF	
Overall Width	E		6.00 BSC	
Exposed Pad Width	E2	3.65	3.70	4.20
Overall Length	D		6.00 BSC	
Exposed Pad Length	D2	3.65	3.70	4.20
Terminal Width	b	0.23	0.30	0.35
Terminal Length	L	0.50	0.55	0.70
Terminal-to-Exposed Pad	K	0.20	-	-

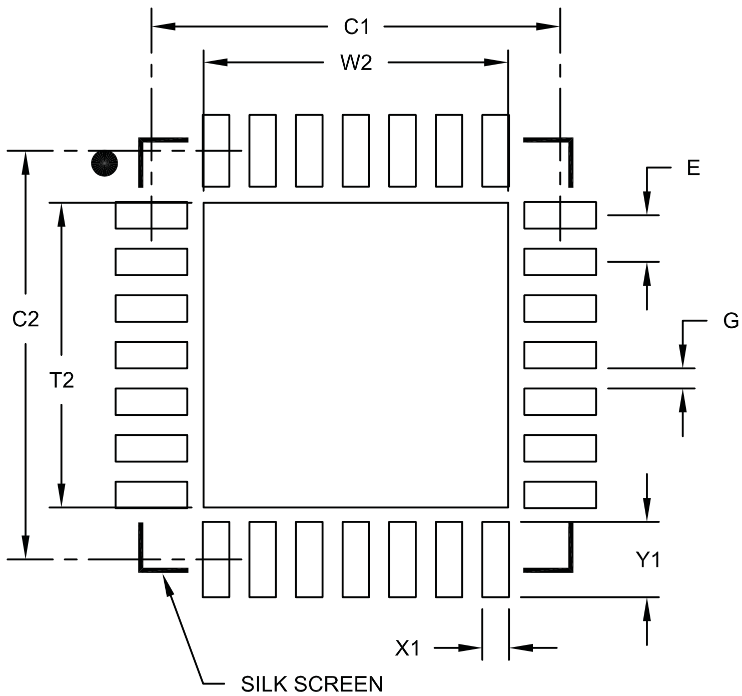
Notes:

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 C04-105C Sheet 2 of 2

28-Lead Plastic Quad Flat, No Lead Package (ML) – 6x6 mm Body [QFN] with 0.55 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		0.65 BSC	
Optional Center Pad Width	W2			4.25
Optional Center Pad Length	T2			4.25
Contact Pad Spacing	C1		5.70	
Contact Pad Spacing	C2		5.70	
Contact Pad Width (X28)	X1			0.37
Contact Pad Length (X28)	Y1			1.00
Distance Between Pads	G	0.20		

Notes:

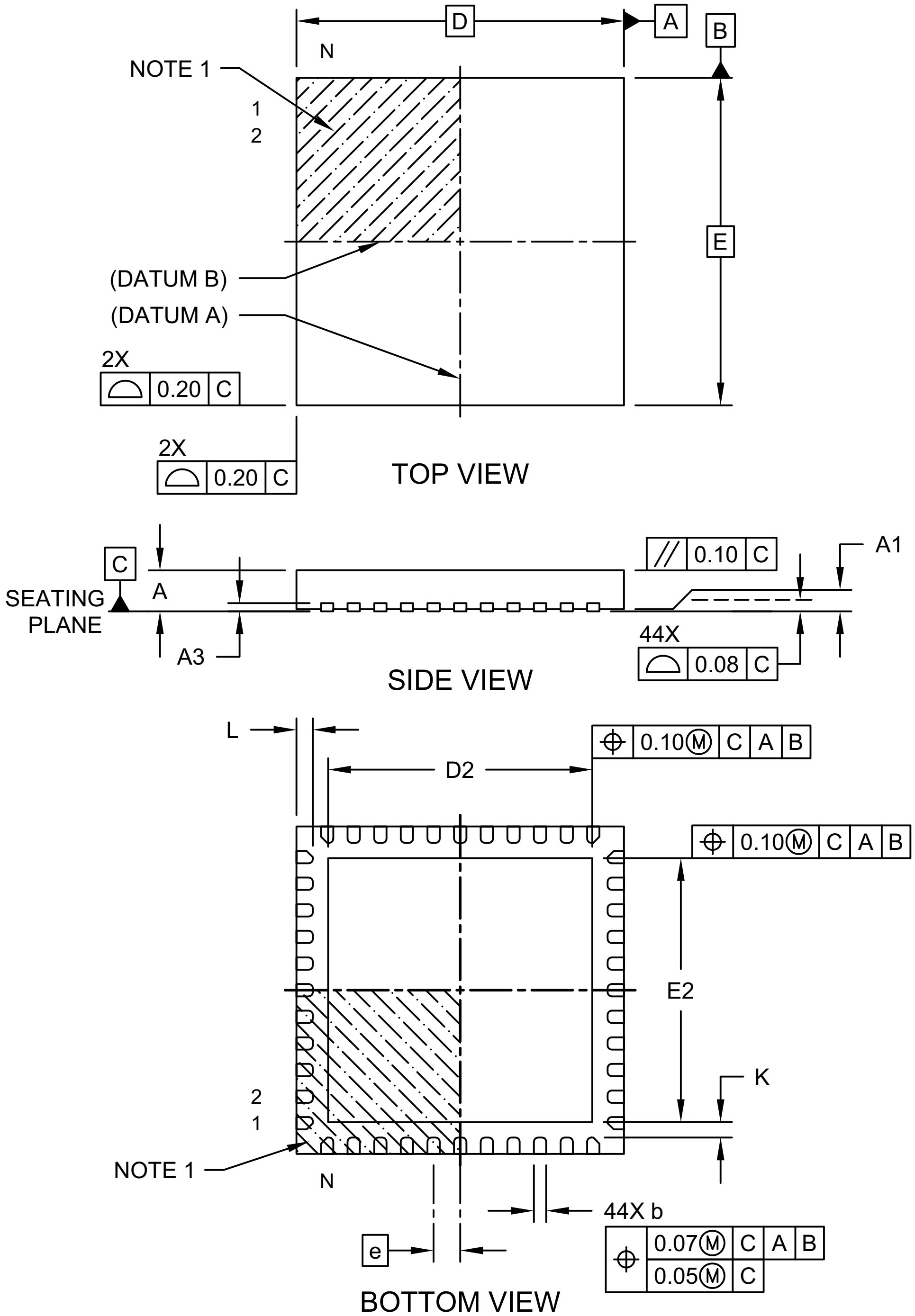
1. Dimensioning and tolerancing per ASME Y14.5M

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2105A

44-Lead Plastic Quad Flat, No Lead Package (ML) - 8x8 mm Body [QFN or VQFN]

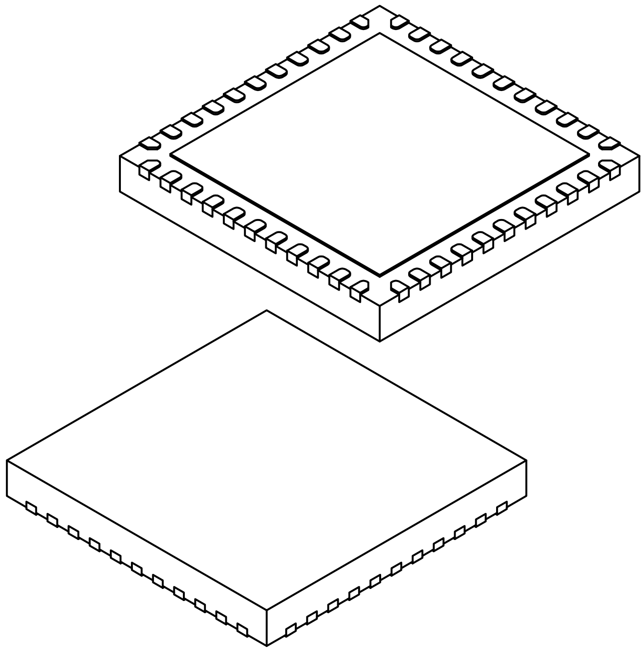
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Microchip Technology Drawing C04-103D Sheet 1 of 2

44-Lead Plastic Quad Flat, No Lead Package (ML) - 8x8 mm Body [QFN or VQFN]

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		44	
Pitch	e		0.65 BSC	
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Terminal Thickness	A3		0.20 REF	
Overall Width	E		8.00 BSC	
Exposed Pad Width	E2	6.25	6.45	6.60
Overall Length	D		8.00 BSC	
Exposed Pad Length	D2	6.25	6.45	6.60
Terminal Width	b	0.20	0.30	0.35
Terminal Length	L	0.30	0.40	0.50
Terminal-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

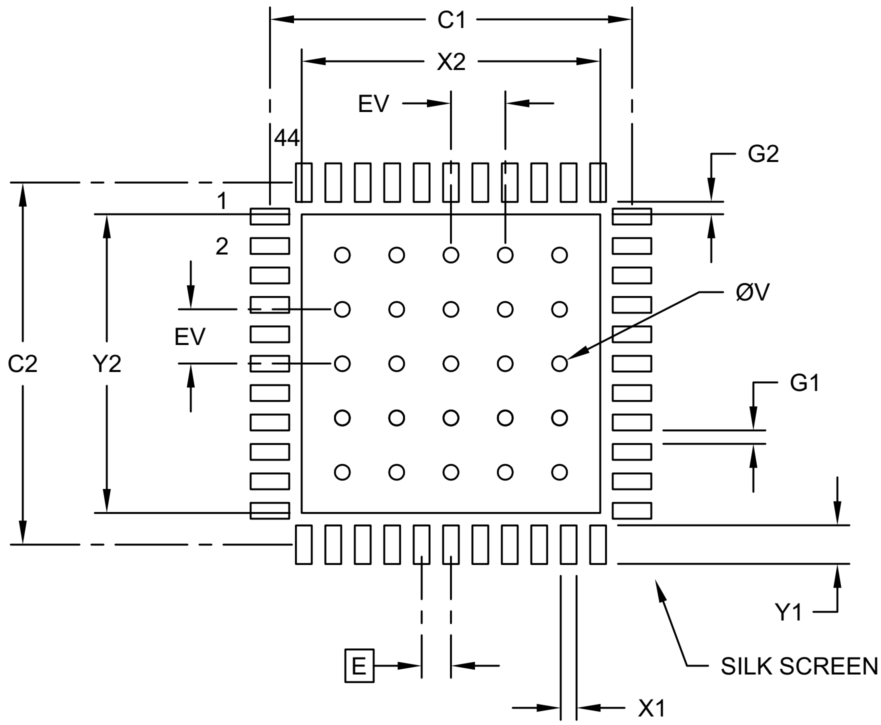
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-103D Sheet 2 of 2

44-Lead Plastic Quad Flat, No Lead Package (ML) - 8x8 mm Body [QFN or VQFN]

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		0.65 BSC	
Optional Center Pad Width	X2			6.60
Optional Center Pad Length	Y2			6.60
Contact Pad Spacing	C1		8.00	
Contact Pad Spacing	C2		8.00	
Contact Pad Width (X44)	X1			0.35
Contact Pad Length (X44)	Y1			0.85
Contact Pad to Contact Pad (X40)	G1	0.30		
Contact Pad to Center Pad (X44)	G2	0.28		
Thermal Via Diameter	V		0.33	
Thermal Via Pitch	EV		1.20	

Notes:

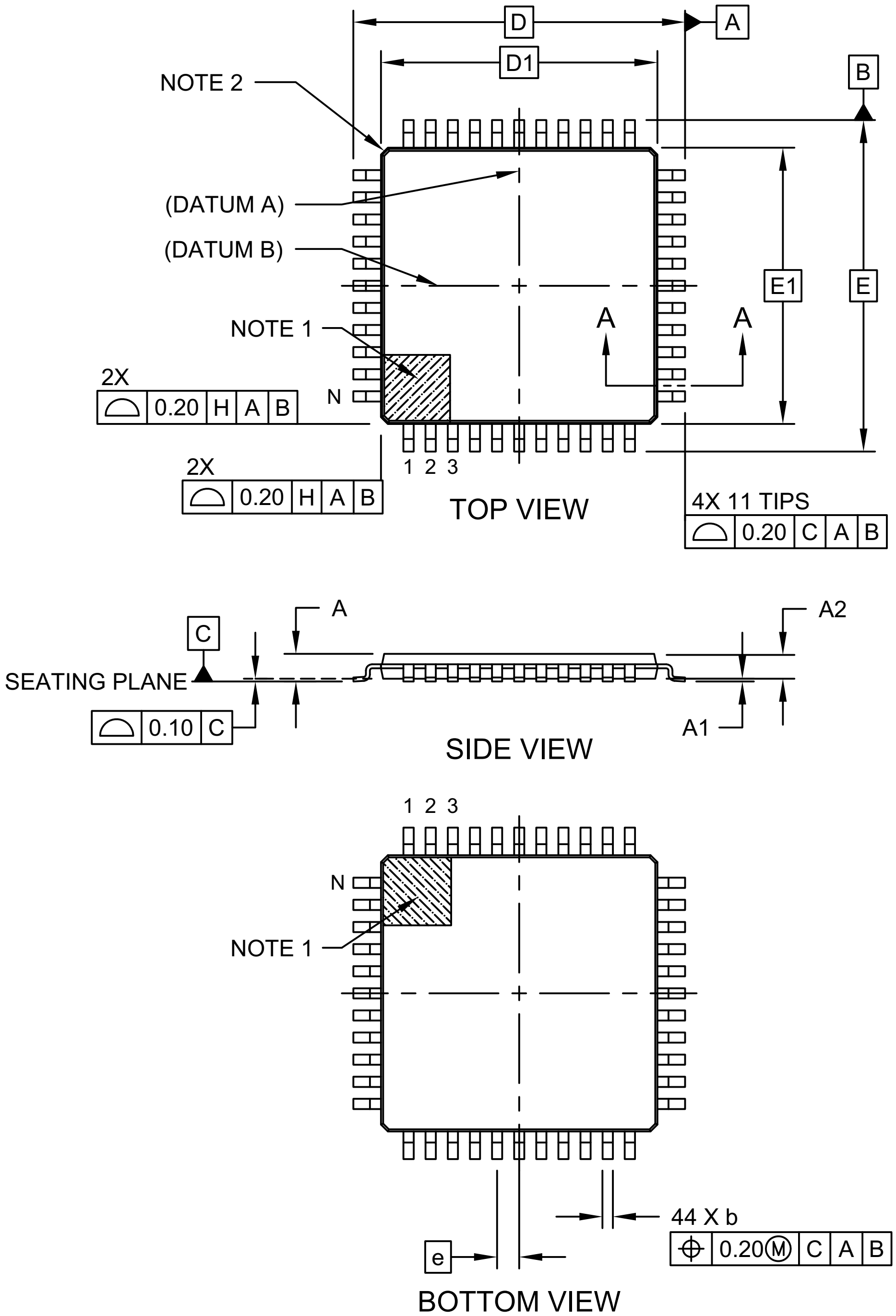
1. Dimensioning and tolerancing per ASME Y14.5M

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
2. For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during reflow process

Microchip Technology Drawing No. C04-2103C

44-Lead Plastic Thin Quad Flatpack (PT) - 10x10x1.0 mm Body [TQFP]

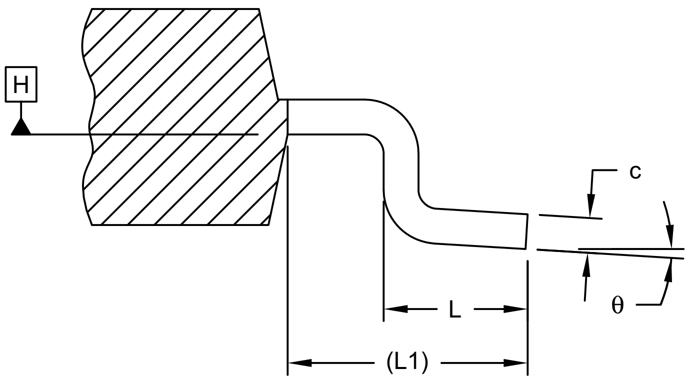
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



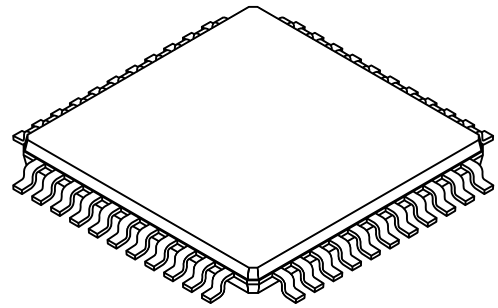
Microchip Technology Drawing C04-076C Sheet 1 of 2

44-Lead Plastic Thin Quad Flatpack (PT) - 10x10x1.0 mm Body [TQFP]

Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



SECTION A-A



	Units	MILLIMETERS		
	Dimension Limits	MIN	NOM	MAX
Number of Leads	N		44	
Lead Pitch	e		0.80 BSC	
Overall Height	A	-	-	1.20
Standoff	A1	0.05	-	0.15
Molded Package Thickness	A2	0.95	1.00	1.05
Overall Width	E		12.00 BSC	
Molded Package Width	E1		10.00 BSC	
Overall Length	D		12.00 BSC	
Molded Package Length	D1		10.00 BSC	
Lead Width	b	0.30	0.37	0.45
Lead Thickness	c	0.09	-	0.20
Lead Length	L	0.45	0.60	0.75
Footprint	L1		1.00 REF	
Foot Angle	θ	0°	3.5°	7°

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Exact shape of each corner is optional.
3. Dimensioning and tolerancing per ASME Y14.5M

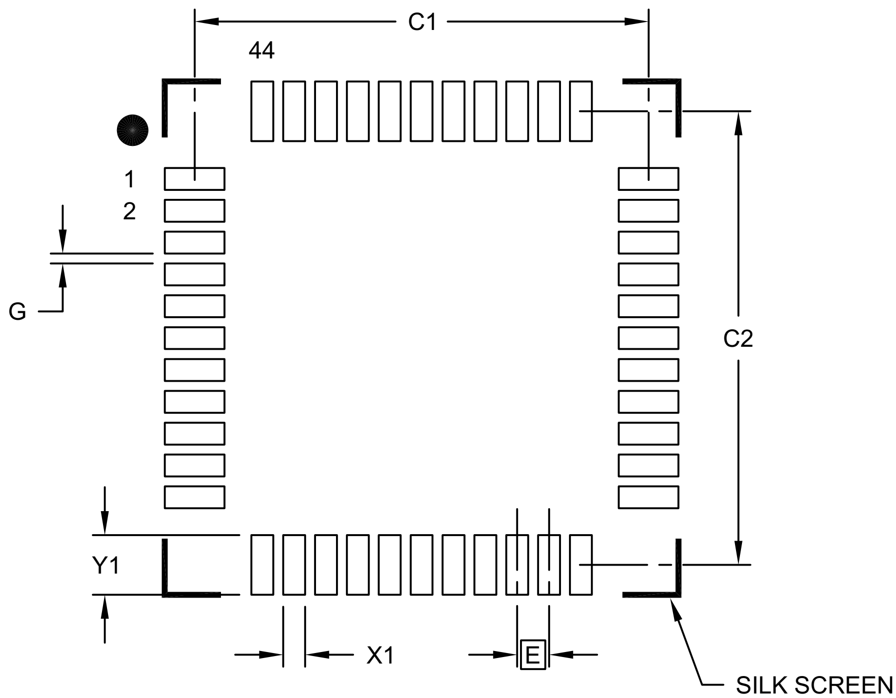
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-076C Sheet 2 of 2

44-Lead Plastic Thin Quad Flatpack (PT) - 10X10X1 mm Body, 2.00 mm Footprint [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	0.80 BSC		
Contact Pad Spacing	C1		11.40	
Contact Pad Spacing	C2		11.40	
Contact Pad Width (X44)	X1			0.55
Contact Pad Length (X44)	Y1			1.50
Distance Between Pads	G	0.25		

Notes:

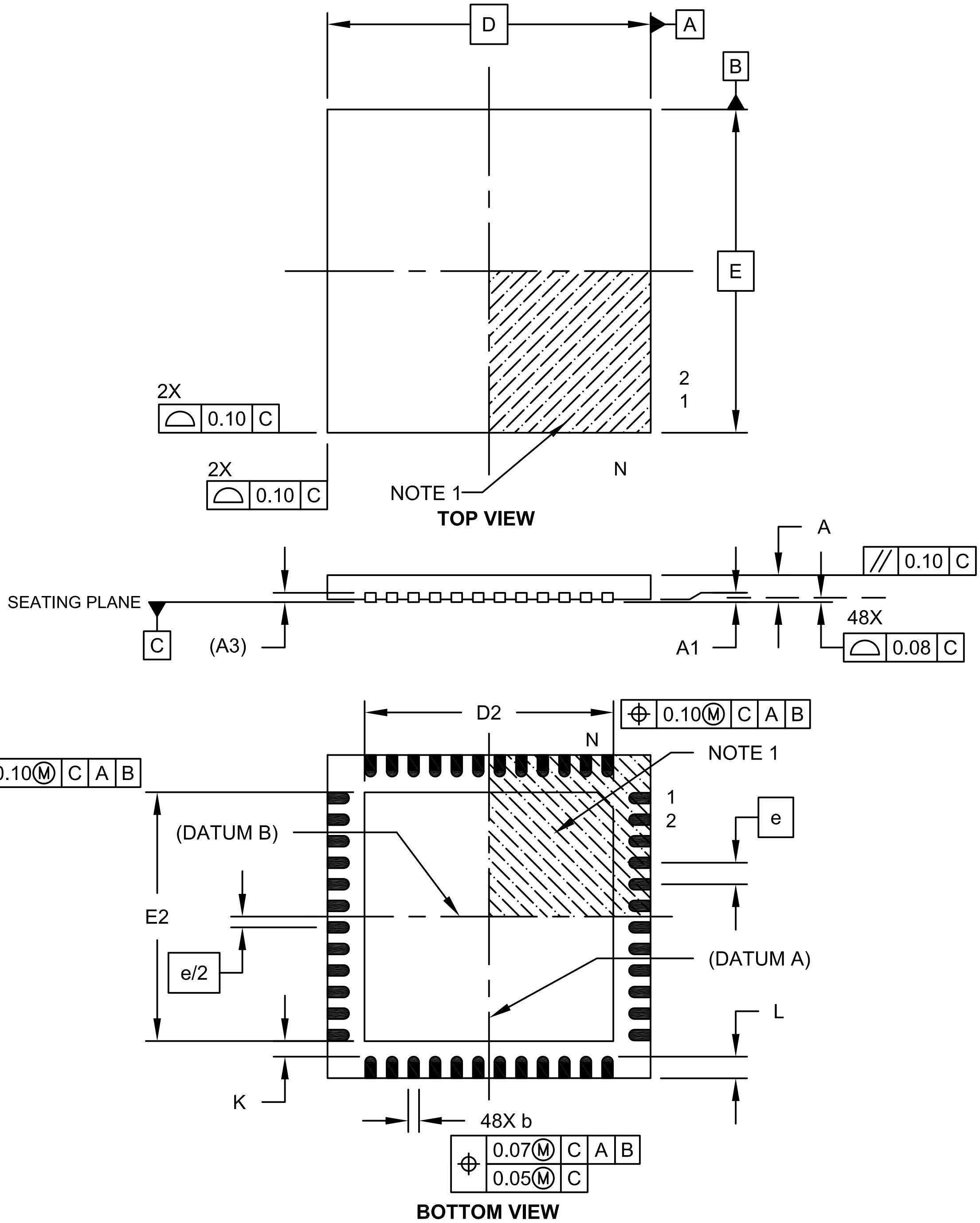
1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2076B

48-Lead Plastic Ultra Thin Quad Flat, No Lead Package (MV) – 6x6x0.5 mm Body [UQFN]

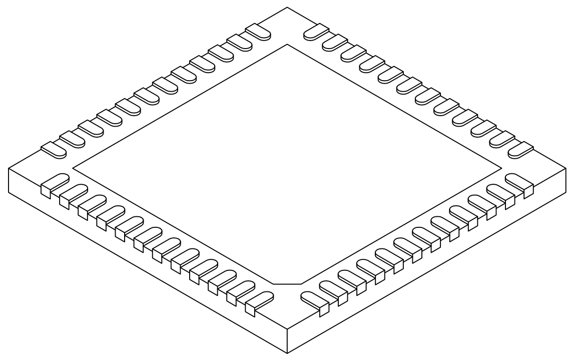
Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Microchip Technology Drawing C04-153A Sheet 1 of 2

48-Lead Plastic Ultra Thin Quad Flat, No Lead Package (MV) – 6x6x0.5 mm Body [UQFN]

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		48	
Pitch	e		0.40 BSC	
Overall Height	A	0.45	0.50	0.55
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3		0.127 REF	
Overall Width	E		6.00 BSC	
Exposed Pad Width	E2	4.45	4.60	4.75
Overall Length	D		6.00 BSC	
Exposed Pad Length	D2	4.45	4.60	4.75
Contact Width	b	0.15	0.20	0.25
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.

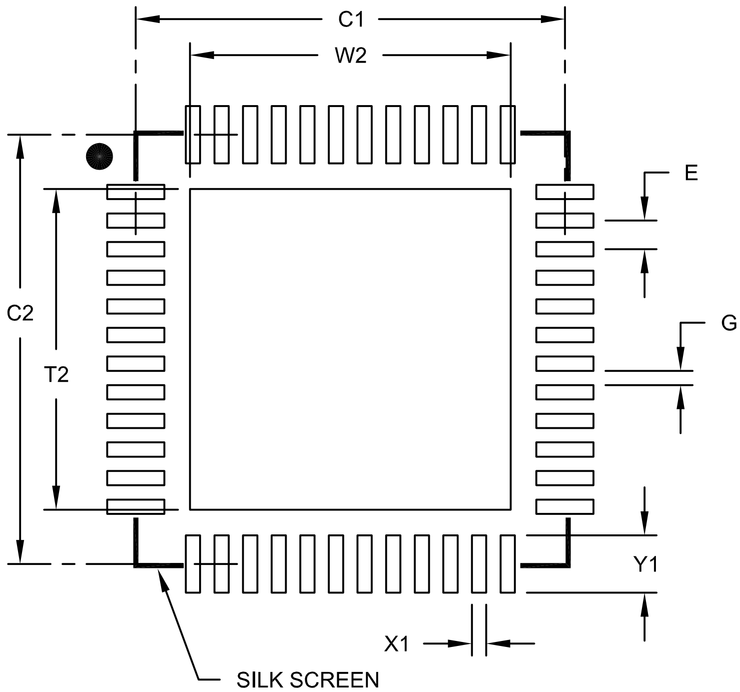
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-153A Sheet 2 of 2

48-Lead Ultra Thin Plastic Quad Flat, No Lead Package (MV) - 6x6 mm Body [UQFN]
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		0.40 BSC	
Optional Center Pad Width	W2			4.45
Optional Center Pad Length	T2			4.45
Contact Pad Spacing	C1		6.00	
Contact Pad Spacing	C2		6.00	
Contact Pad Width (X28)	X1			0.20
Contact Pad Length (X28)	Y1			0.80
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. C04-2153A

NOTES:

APPENDIX A: REVISION HISTORY

Revision A (March 2011)

Original data sheet for the PIC24FV32KA304 family of devices.

Revision B (April 2011)

Section 25.0 "Charge Time Measurement Unit (CTMU)" was revised to change the description of the IRNGx bits in CTMUICON (Register 25-3). Setting '01' is the base current level (0.55 µA nominal) and setting '00' is 1000x base current.

Section 29.0 "Electrical Characteristics" was revised to change the following typical IPD specifications:

- DC20h/i/j/k from 204 µA to 200 µA
- DC60h/i/j/k from 0.15 µA to 0.025 µA
- DC60l/m/n/o from 0.25 µA to 0.040 µA
- DC72h/i/j/k from 0.80 µA to 0.70 µA

Revision C (April 2012)

Updated the Pin Diagrams on Pages 3 through 7, to change "LVDIN" to "HLVDIN" in all occurrences, and correct the placement of certain functions.

Updated Table 1-3 to remove references to unimplemented package types, corrected several erroneous pin assignments and removed other alternate but unimplemented assignments.

For **Section 5.0 "Flash Program Memory"**, updated Example 5-2, Example 5-3 and Example 5-4 with new table offset functions.

Updated Figure 12-1 to correctly show the implemented Timer1 input options.

For **Section 22.0 "12-Bit A/D Converter with Threshold Detect"**:

- Updated Register 22-1 to add the MODE12 bit
- Updated the descriptions of the PVCFGx and CSCNA bits in Register 22-2
- Updated Register 22-4 to change the VRSREQ bit to a reserved bit position
- Modified footnote text in Register 22-5
- Corrected CHOLD in Figure 22-2

For **Section 25.0 "Charge Time Measurement Unit (CTMU)"**:

- Updated the text in **Section 25.1 "Measuring Capacitance"** and **Section 25.3 "Pulse Generation and Delay"** to better reflect the module's implementation
- Updated Figure 25-3 to show additional detail in pulse generation

Added the following timing diagrams and timing requirement tables to **Section 29.0 "Electrical Characteristics"**:

- Figure 29-6 (Reset, Watchdog Timer, Oscillator Start-up Timer and Power-up Timer Timing Characteristics)
- Figure 29-7 (Brown-out Reset Characteristics)
- Figure 29-9 (Input Capture x Timings) through Figure 29-21 (SPIx Module Slave Mode Timing Characteristics (CKE = 1))
- Table 29-28 (Input Capture x Requirements) through Table 29-39 (SPIx Module Slave Mode Timing Requirements (CKE = 1))
- Figure 29-22 (A/D Conversion Timing)

Updated Table 29-5 to add specification, DC15.

Replaced Table 29-6, Table 29-7 and Table 29-8 with new, shorter versions that remove unimplemented temperature options. (No existing specification values have been changed in this process.)

Updated Table 29-16 with correct values for CTMUICON bit settings.

Combined previous Table 29-21 and Table 29-22 to create a new Table 29-21 (AC Characteristics: Internal RC Accuracy). All existing subsequent tables are renumbered accordingly.

Updated Table 29-26 to add specifications, SY35 and SY55.

Updated Table 29-40:

- Split AD01 into separate entries for "F" and "FV" device families
- Added specifications, AD08 (IVREF) and AD09 (ZVREF)
- Changed AD17 (2.5 kΩ max. to 1 kΩ max.)

Updated Table 29-41:

- Changed AD50 (75 ns min. to 600 ns min.)
- Changed AD51 (250 ns typ. to 1.67 µs typ.)
- Changed AD60 (0.5 TAD min. to 2 TAD min.)
- Split AD55 into separate entries for 10-bit and 12-bit conversions

Added **Section 30.0 "DC and AC Characteristics Graphs and Tables"**, with Figure 30-1 through Figure 30-39.

Replaced some of the packaging diagrams in **Section 31.0 "Packaging Information"** with the newly revised diagrams.

Other minor typographic corrections throughout.

Revision D (March 2013)

Throughout the data sheet: corrected the name of RCON register bit 12 as RETEN, to maintain consistency with other PIC24F devices (was previously LVREN). In addition, changed the description of the bit in the RCON register (Register 7-1) to clarify its function in controlling the Retention Regulator.

Throughout the data sheet: corrected the name of FPOR Configuration register bit 2 as RETCFG, to maintain consistency with other PIC24F devices (was previously LVRCFG). In addition, changed the description of the bit in the FPOR Configuration register (Register 26-6) to clarify its function in enabling the Retention Regulator.

For **Section 10.4 "Voltage Regulator-Based Power-Saving Features"**:

- Removed all references to Fast Wake-up Sleep mode, not implemented in this device
- Changed all references of the High-Voltage Regulator to On-Chip Voltage Regulator
- Removed all references to the Low-Voltage Regulator, which was replaced in most cases with Retention Regulator
- Clarified the Retention Regulator's operation in **Section 10.4.3 "Retention Sleep Mode"** (formerly "Low-Voltage Sleep Mode")
- Modified Table 10-1 for consistency with the above changes

Corrects **Section 26.2 "On-Chip Voltage Regulator"** to clarify the operation of the on-chip regulator in "F" and "FV" families, and include DC parameters and specifications.

For **Section 29.0 "Electrical Characteristics"**:

- Updated captioning on all specification tables to include extended temperature data
- Amended Table 29-8 to include +125°C data for all existing specifications
- Added new Table 29-27 and Figure 29-8 to characterize external clock input specifications for general purpose timers (all subsequent tables and figures are renumbered accordingly)
- Added parameter numbers to several existing but previous unnumbered parameters in multiple tables

Updated **Section 30.0 "DC and AC Characteristics Graphs and Tables"**:

- Added additional graphs for Extended temperature devices (**Section 30.2 "Characteristics for Extended Temperature Devices (-40°C to +125°C)"**, Figure 30-40 through Figure 30-56)
- Replaced Figure 30-32 with an updated graph

Replaced some of the packaging diagrams in **Section 31.0 "Packaging Information"** with the newly revised diagrams.

Updates Product Information System to include extended temperature devices in the information key.

Other minor typographic corrections throughout.

Revision E (October 2017)

Changed the PGEC2/PGED2 44-Pin value to 9 in Table 1-3.

Removed the OFFCAL bit from Table 4-16 and Register 22-2.

Updated Register 6-1 to include the correct Programming Operations for NVMP<5:0>.

Updated Figure 22-1.

Updated footnotes in Register 25-2.

Updated ambient temperature under bias range in **Section 29.0 "Electrical Characteristics"**, updated the DC16 electrical specs in Table 29-21, updated the F20 FRC electrical specs in Table 29-3 and changed the Minimum Input Capture x requirement for IC15 TccP.

Other minor typographic and formatting corrections throughout.

INDEX

NOTES:

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. Under "Support", 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

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://microchip.com/support

NOTES:

PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PIC 24 FV 32 KA3 04 T - I / PT - XXX

- **Microchip Trademark**
- **Architecture**
- **Flash Memory Family**
- **Program Memory Size (KB)**
- **Product Group**
- **Pin Count**
- **Tape and Reel Flag (if applicable)**
- **Temperature Range**
- **Package**
- **Pattern**

Architecture	24	= 16-bit modified Harvard without DSP
Flash Memory Family	F	= Standard voltage range Flash program memory
	FV	= Wide voltage range Flash program memory
Product Group	KA3	= General purpose microcontrollers
Pin Count	01	= 20-pin
	02	= 28-pin
	04	= 44-pin
Temperature Range	I	= -40°C to +85°C (Industrial)
	E	= -40°C to +125°C (Industrial)
Package	SP	= SPDIP
	SO	= SOIC
	SS	= SSOP
	ML	= QFN
	P	= PDIP
	PT	= TQFP
	MV	= UQFN
Pattern		Three-digit QTP, SQTP, Code or Special Requirements (blank otherwise)
	ES	= Engineering Sample

Examples:

a) PIC24FV32KA304-I/ML: Wide voltage range, General Purpose, 32-Kbyte program memory, 44-pin, Industrial temp., QFN package

b) PIC24F16KA302-I/SS: Standard voltage range, General Purpose, 16-Kbyte program memory, 28-pin, Industrial temp., SSOP package

NOTES:

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.

Microchip received ISO/TS-16949:2009 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

ISBN: 978-1-5224-2236-5

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://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453
Tel: 317-536-2380

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608
Tel: 951-273-7800

Raleigh, NC
Tel: 919-844-7510

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110
Tel: 408-436-4270

Canada - Toronto
Tel: 905-695-1980
Fax: 905-695-2078

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon

Hong Kong
Tel: 852-2943-5100
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
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 - Dongguan
Tel: 86-769-8702-9880

China - Guangzhou
Tel: 86-20-8755-8029

China - Hangzhou
Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

China - Hong Kong SAR
Tel: 852-2943-5100
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: 86-21-3326-8000
Fax: 86-21-3326-8021

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8864-2200
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

ASIA/PACIFIC

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

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: 91-20-3019-1500

Japan - Osaka
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

Japan - Tokyo
Tel: 81-3-6880- 3770
Fax: 81-3-6880-3771

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: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-213-7830

Taiwan - Taipei
Tel: 886-2-2508-8600
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

Finland - Espoo
Tel: 358-9-4520-820

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Garching
Tel: 49-8931-9700

Germany - Haan
Tel: 49-2129-3766400

Germany - Heilbronn
Tel: 49-7131-67-3636

Germany - Karlsruhe
Tel: 49-721-625370

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Germany - Rosenheim
Tel: 49-8031-354-560

Israel - Ra'anana
Tel: 972-9-744-7705

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Padova
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Norway - Trondheim
Tel: 47-7289-7561

Poland - Warsaw
Tel: 48-22-3325737

Romania - Bucharest
Tel: 40-21-407-87-50

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Gothenberg
Tel: 46-31-704-60-40

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

Microchip:

PIC24F16KA301-I/P PIC24F16KA301-I/SO PIC24F16KA301-I/SS PIC24F16KA301T-I/SO PIC24F16KA301T-I/SS PIC24F16KA302-I/ML PIC24F16KA302-I/SO PIC24F16KA302-I/SP PIC24F16KA302-I/SS PIC24F16KA302T-I/ML PIC24F16KA302T-I/SO PIC24F16KA302T-I/SS PIC24F16KA304-I/ML PIC24F16KA304-I/MV PIC24F16KA304-I/PT PIC24F16KA304T-I/ML PIC24F16KA304T-I/MV PIC24F16KA304T-I/PT PIC24F32KA301-I/P PIC24F32KA301-I/SO PIC24F32KA301-I/SS PIC24F32KA301T-I/SO PIC24F32KA301T-I/SS PIC24F32KA302-I/ML PIC24F32KA302-I/SO PIC24F32KA302-I/SP PIC24F32KA302-I/SS PIC24F32KA302T-I/ML PIC24F32KA302T-I/SO PIC24F32KA302T-I/SS PIC24F32KA304-I/ML PIC24F32KA304-I/MV PIC24F32KA304-I/PT PIC24F32KA304T-I/ML PIC24F32KA304T-I/MV PIC24F32KA304T-I/PT PIC24LF16KA301-I/P PIC24LF16KA301-I/SO PIC24LF16KA301-I/SS PIC24LF16KA301T-I/SO PIC24LF16KA301T-I/SS PIC24LF16KA302-I/ML PIC24LF16KA302-I/SO PIC24LF16KA302-I/SP PIC24LF16KA302-I/SS PIC24LF16KA302T-I/ML PIC24LF16KA302T-I/SO PIC24LF16KA302T-I/SS PIC24LF16KA304-I/ML PIC24LF16KA304-I/MV PIC24LF16KA304-I/PT PIC24LF16KA304T-I/ML PIC24LF16KA304T-I/MV PIC24LF16KA304T-I/PT PIC24LF32KA301-I/P PIC24LF32KA301-I/SO PIC24LF32KA301-I/SS PIC24LF32KA301T-I/SO PIC24LF32KA301T-I/SS PIC24LF32KA302-I/ML PIC24LF32KA302-I/SO PIC24LF32KA302-I/SP PIC24LF32KA302-I/SS PIC24LF32KA302T-I/ML PIC24LF32KA302T-I/SO PIC24LF32KA302T-I/SS PIC24LF32KA304-I/ML PIC24LF32KA304-I/MV PIC24LF32KA304-I/PT PIC24LF32KA304T-I/ML PIC24LF32KA304T-I/MV PIC24LF32KA304T-I/PT PIC24F16KA301-E/P PIC24F16KA301-E/SO PIC24F16KA301-E/SS PIC24F16KA302-E/ML PIC24F16KA302-E/SO PIC24F16KA302-E/SP PIC24F16KA302-E/SS PIC24F16KA304-E/ML PIC24F16KA304-E/MV PIC24F16KA304-E/PT PIC24F32KA301-E/P PIC24F32KA301-E/SO PIC24F32KA301-E/SS PIC24F32KA302-E/ML PIC24F32KA302-E/SO PIC24F32KA302-E/SP PIC24F32KA302-E/SS PIC24F32KA304-E/ML PIC24F32KA304-E/MV PIC24FV16KA301-E/P PIC24FV16KA301-E/SO PIC24FV16KA301-E/SS PIC24FV16KA301-I/P PIC24FV16KA301-I/SO PIC24FV16KA301-I/SS PIC24FV16KA301T-I/SO PIC24FV16KA301T-I/SS PIC24FV16KA302-E/ML

OCEAN CHIPS

Океан Электроники

Поставка электронных компонентов

Компания «Океан Электроники» предлагает заключение долгосрочных отношений при поставках импортных электронных компонентов на взаимовыгодных условиях!

Наши преимущества:

- Поставка оригинальных импортных электронных компонентов напрямую с производств Америки, Европы и Азии, а так же с крупнейших складов мира;
- Широкая линейка поставок активных и пассивных импортных электронных компонентов (более 30 млн. наименований);
- Поставка сложных, дефицитных, либо снятых с производства позиций;
- Оперативные сроки поставки под заказ (от 5 рабочих дней);
- Экспресс доставка в любую точку России;
- Помощь Конструкторского Отдела и консультации квалифицированных инженеров;
- Техническая поддержка проекта, помощь в подборе аналогов, поставка прототипов;
- Поставка электронных компонентов под контролем ВП;
- Система менеджмента качества сертифицирована по Международному стандарту ISO 9001;
- При необходимости вся продукция военного и аэрокосмического назначения проходит испытания и сертификацию в лаборатории (по согласованию с заказчиком);
- Поставка специализированных компонентов военного и аэрокосмического уровня качества (Xilinx, Altera, Analog Devices, Intersil, Interpoint, Microsemi, Actel, Aeroflex, Peregrine, VPT, Syfer, Eurofarad, Texas Instruments, MS Kennedy, Miteq, Cobham, E2V, MA-COM, Hittite, Mini-Circuits, General Dynamics и др.);

Компания «Океан Электроники» является официальным дистрибьютором и эксклюзивным представителем в России одного из крупнейших производителей разъемов военного и аэрокосмического назначения **«JONHON»**, а так же официальным дистрибьютором и эксклюзивным представителем в России производителя высокотехнологичных и надежных решений для передачи СВЧ сигналов **«FORSTAR»**.

AVIC JONHON

«JONHON» (основан в 1970 г.)

Разъемы специального, военного и аэрокосмического назначения:

(Применяются в военной, авиационной, аэрокосмической, морской, железнодорожной, горно- и нефтедобывающей отраслях промышленности)

«FORSTAR» (основан в 1998 г.)

ВЧ соединители, коаксиальные кабели, кабельные сборки и микроволновые компоненты:

(Применяются в телекоммуникациях гражданского и специального назначения, в средствах связи, РЛС, а так же военной, авиационной и аэрокосмической отраслях промышленности).

FORSTAR®

Телефон: 8 (812) 309-75-97 (многоканальный)
Факс: 8 (812) 320-03-32
Электронная почта: ocean@oceanchips.ru
Web: http://oceanchips.ru/
Адрес: 198099, г. Санкт-Петербург, ул. Калинина, д. 2, корп. 4, лит. А