Microchip Technology - PIC24F08KA102-E/SP Datasheet

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	PIC
Core Size	16-Bit
Speed	32MHz
Connectivity	I²C, IrDA, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	24
Program Memory Size	8KB (2.75K x 24)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	1.5K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 9x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Through Hole
Package / Case	28-DIP (0.300", 7.62mm)
Supplier Device Package	28-SPDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24f08ka102-e-sp

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 to 28-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 PIC24F16KA102 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.
- **10-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 PIC24F16KA102 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 PIC24F16KA102 family are available in 20-pin and 28-pin packages. The general block diagram for all devices is displayed in Figure 1-1.

The devices are different from each other in two ways:

1. Flash program memory (8 Kbytes for PIC24F08KA devices, 16 Kbytes for PIC24F16KA devices).
2. Available I/O pins and ports (18 pins on two ports for 20-pin devices and 24 pins on two ports for 28-pin devices).
3. Alternate SCLx and SDAx pins are available only in 28-pin devices and not in 20-pin devices.

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 PIC24F16KA102 family devices, sorted by function, is provided in Table 1-2.

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 4, 5 and 6 of the data sheet. Multiplexed features are sorted by the priority given to a feature, with the highest priority peripheral being listed first.

2.4 Voltage Regulator Pin (VCAP)

Note: This section applies only to PIC24F K devices with an on-chip voltage regulator.

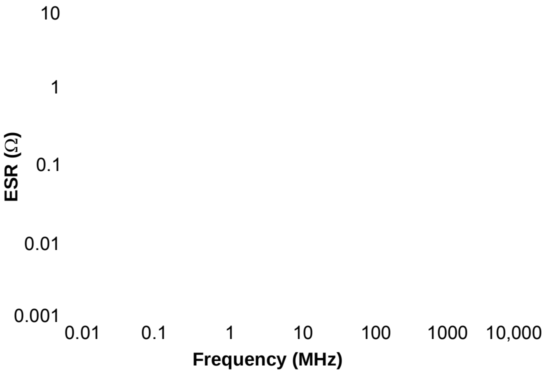
Some of the PIC24F K devices have an internal voltage regulator. These devices have the voltage regulator output brought out on the VCAP pin. On the PIC24F K devices with regulators, a low-ESR (< 5Ω) capacitor is required on the VCAP pin to stabilize the voltage regulator output. The VCAP pin must not 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

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:	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 $\overline{\text{Borrow}}$ and $\overline{\text{Digit Borrow}}$ 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

TABLE 4-9: I^{2}C™ REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
I2C1RCV	0200	—	—	—	—	—	—	—	—	I2C1 Receive Register								0000
I2C1TRN	0202	—	—	—	—	—	—	—	—	I2C1 Transmit Register								00FF
I2C1BRG	0204	—	—	—	—	—	—	—	I2C1 Baud Rate Generator Register									0000
I2C1CON	0206	I2CEN	—	I2CSIDL	SCLREL	IPMIEN	A10M	DISSLW	SMEN	GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	1000
I2C1STAT	0208	ACKSTAT	TRSTAT	—	—	—	BCL	GCSTAT	ADD10	IWCOL	I2COV	$D/\overline{A}$	P	S	$R/\overline{W}$	RBF	TBF	0000
I2C1ADD	020A	—	—	—	—	—	—	I2C1 Address Register										0000
I2C1MSK	020C	—	—	—	—	—	—	AMSK9	AMSK8	AMSK7	AMSK6	AMSK5	AMSK4	AMSK3	AMSK2	AMSK1	AMSK0	0000

Legend: — = unimplemented, read as ‘0’. Reset values are shown in h.5adecimal.

TABLE 4-10: UART 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	—	—	—	—	—	—	—	UART1 Transmit Register									0000
U1RXREG	0226	—	—	—	—	—	—	—	UART1 Receive Register									0000
U1BRG	0228	Baud Rate Generator Prescaler Register																0000
U2MODE	0230	UARTEN	—	USIDL	IREN	RTSMD	—	UEN1	UEN0	WAKE	LPBACK	ABAUD	RXINV	BRGH	PDSEL1	PDSEL0	STSEL	0000
U2STA	0232	UTXISEL1	UTXINV	UTXISEL0	—	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA	0110
U2TXREG	0234	—	—	—	—	—	—	—	UART2 Transmit Register									0000
U2RXREG	0236	—	—	—	—	—	—	—	UART2 Receive Register									0000
U2BRG	0238	Baud Rate Generator Prescaler																0000

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

TABLE 4-11: SPI 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	SRXMPT	SISEL2	SISEL1	SISEL0	SPITBF	SPIRBF	0000
SPI1CON1	0242	—	—	—	DISSCK	DISSDO	MODE16	SMP	CKE	SSEN	CKP	MSTEN	SPRE2	SPRE1	SPRE0	PPRE1	PPRE0	0000
SPI1CON2	0244	FRMEN	SPIFSD	SPIFPOL	—	—	—	—	—	—	—	—	—	—	—	SPIFE	SPIBEN	0000
SPI1BUF	0248	SPI1 Transmit/Receive Buffer																0000

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

TABLE 4-20: 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	—	—	DPSLP	—	PMSLP	EXTR	SWR	SWDTEN	WDTO	SLEEP	IDLE	BOR	POR	**(Note 1)**
OSCCON	0742	—	COSC2	COSC1	COSC0	—	NOSC2	NOSC1	NOSC0	CLKLOCK	—	LOCK	—	CF	—	SOSCEN	OSWEN	**(Note 2)**
CLKDIV	0744	ROI	DOZE2	DOZE1	DOZE0	DOZEN	RCDIV2	RCDIV1	RCDIV0	—	—	—	—	—	—	—	—	3140
OSCTUN	0748	—	—	—	—	—	—	—	—	—	—	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0	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 configuration fuses and by type of Reset.

TABLE 4-21: 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[(1)]
DSCON	0758	DSEN	—	—	—	—	—	—	—	—	—	—	—	—	—	DSBOR	RELEASE	0000
DSWAKE	075A	—	—	—	—	—	—	—	DSINT0	DSFLT	—	—	DSWDT	DSRTCC	DSMCLR	—	DSPOR	0000
DSGPR0	075C	Deep Sleep General Purpose Register 0																0000
DSGPR1	075E	Deep Sleep General Purpose Register 1																0000

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

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

TABLE 4-22: NVM REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
NVMCON	0760	WR	WREN	WRERR	PGMONLY	—	—	—	—	—	ERASE	NVMOP5	NVMOP4	NVMOP3	NVMOP2	NVMOP1	NVMOP0	0000[(1)]
NVMKEY	0766	—	—	—	—	—	—	—	—	NVMKEY7	NVMKEY6	NVMKEY5	NVMKEY4	NVMKEY3	NVMKEY2	NVMKEY1	NVMKEY0	0000

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

Note 1: Reset value shown is for POR only. Value on other Reset states is dependent on the state of memory write or erase operations at the time of Reset.

TABLE 4-23: 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	—	—	T3MD	T2MD	T1MD	—	—	—	I2C1MD	U2MD	U1MD	—	SPI1MD	—	—	ADC1MD	0000
PMD2	0772	—	—	—	—	—	—	—	IC1MD	—	—	—	—	—	—	—	OC1MD	0000
PMD3	0774	—	—	—	—	—	CMPMD	RTCCMD	—	CRCPMD	—	—	—	—	—	—	—	0000
PMD4	0776	—	—	—	—	—	—	—	—	—	—	—	EEMD	REFOMD	CTMUMD	HLVDMD	—	0000

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

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 *“PIC24F Family Reference Manual”*, **Section 5. “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 PIC24F16KA102 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 V_{DD} 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 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	U-0	U-0	U-0	R/W-0, HS	R/W-0, HS	R/W-0, HS	R/W-0, HS
T2IF	—	—	—	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 6-4 **Unimplemented:** Read as ‘0’

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

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

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 8-16: 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

NOTES:

13.0 TIMER2/3

> **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 *"PIC24F Family Reference Manual"*, **Section 14. "Timers"** (DS39704).

The Timer2/3 module is a 32-bit timer, which can also be configured as two independent 16-bit timers with selectable operating modes.

As a 32-bit timer, Timer2/3 operates 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 and Sleep modes
- Interrupt on a 32-bit Period register match
- A/D Event Trigger

Individually, both 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 and T3CON registers. T2CON and T3CON are provided in generic form in Register 13-1 and Register 13-2, respectively.

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

> **Note:** For 32-bit operation, T3CON control bits are ignored. Only T2CON control bits are used for setup and control. Timer2 clock and gate inputs are utilized for the 32-bit timer modules, but an interrupt is generated with the Timer3 interrupt flags.

To configure Timer2/3 for 32-bit operation:

1. Set the T32 bit (T2CON<3> = `1`).
2. Select the prescaler ratio for Timer2 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 will contain the msw of the value while PR2 contains the lsw.
5. If interrupts are required, set the interrupt enable bit, T3IE. Use the priority bits, T3IP<2:0>, to set the interrupt priority.

   While Timer2 controls the timer, the interrupt appears as a Timer3 interrupt.
6. Set the TON bit (= `1`).

The timer value, at any point, is stored in the register pair, TMR<3:2>. TMR3 always contains the msw of the count, while TMR2 contains the lsw.

To configure any of the timers for individual 16-bit operation:

1. Clear the T32 bit in T2CON<3>.
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 interrupt enable bit, TxIE; use the priority bits, TxIP<2:0>, to set the interrupt priority.
6. Set the TON bit (TxCON<15> = `1`).

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 *"PIC24F Family Reference Manual"*, **Section 29. "Real-Time Clock and Calendar (RTCC)"** (DS39696).

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

19.1 RTCC Source Clock

The user can select between the SOSC crystal oscillator or the LPRC internal oscillator as the clock reference for the RTCC module. This is configured using the RTCOSC (FDS<5>) Configuration bit. This gives the user an option to trade off system cost, accuracy and power consumption, based on the overall system needs.

The RTCC will continue to run, along with its chosen clock source, while the device is held in Reset with $\overline{\text{MCLR}}$ and will continue running after $\overline{\text{MCLR}}$ is released.

FIGURE 19-1: RTCC BLOCK DIAGRAM

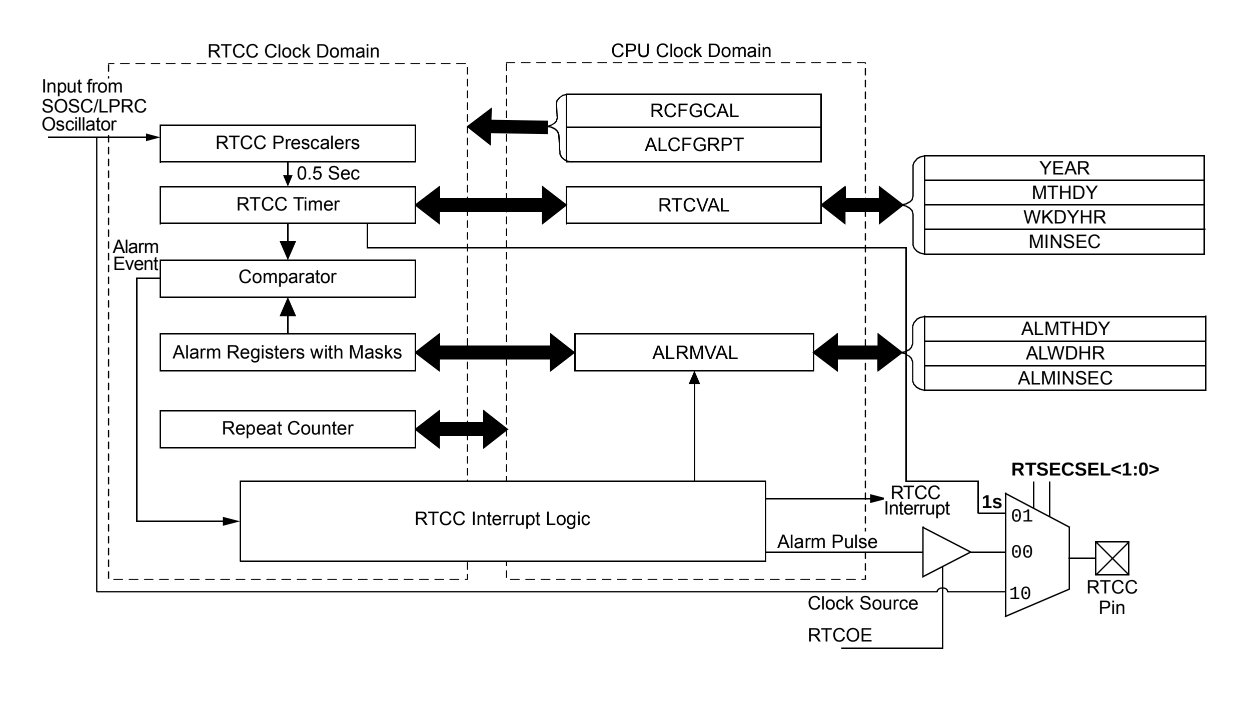
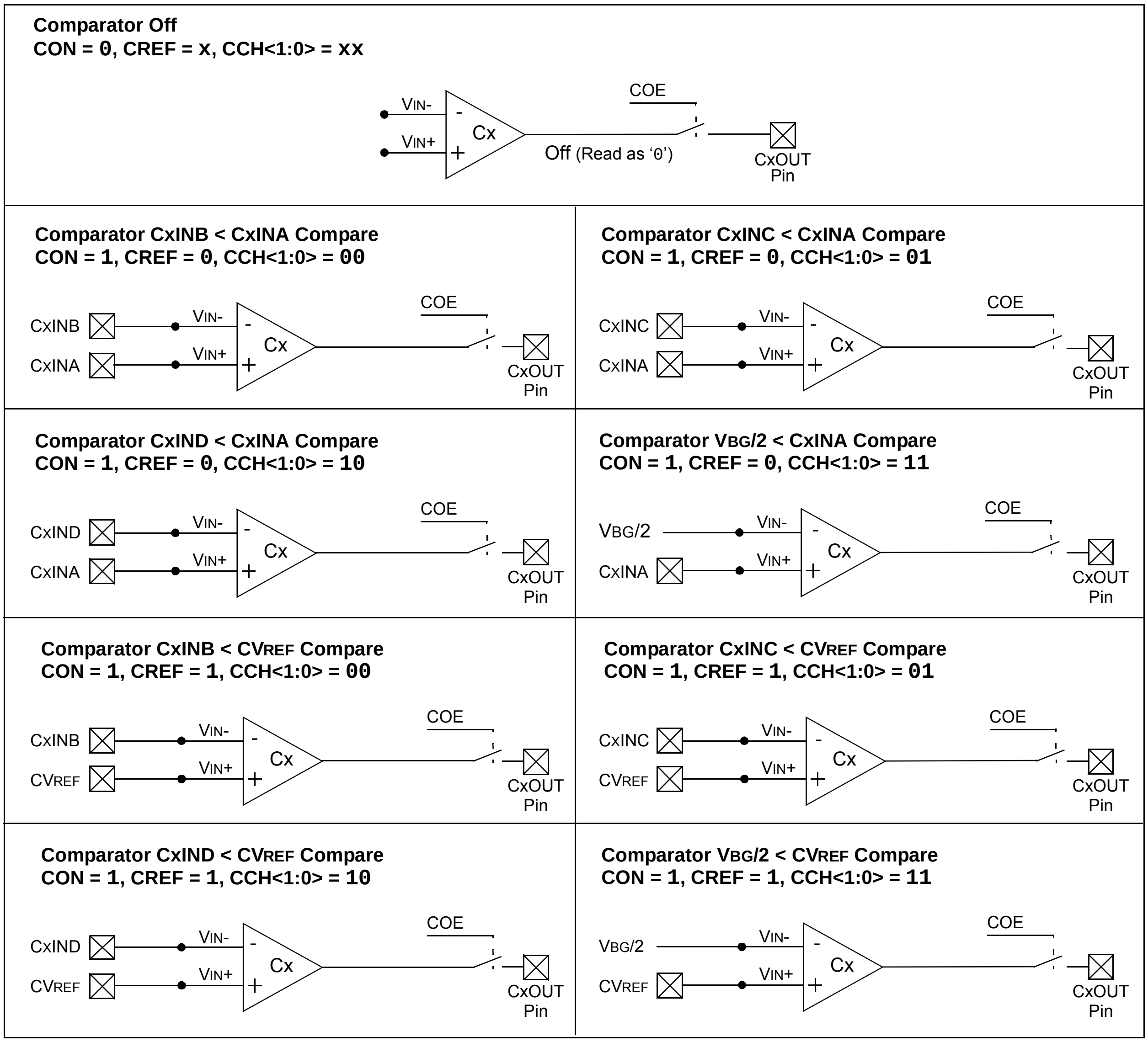


FIGURE 23-2: INDIVIDUAL COMPARATOR CONFIGURATIONS



Comparator Off
CON = 0, CREF = x, CCH<1:0> = xx

Comparator CxINB < CxINA Compare
CON = 1, CREF = 0, CCH<1:0> = 00

Comparator CxINC < CxINA Compare
CON = 1, CREF = 0, CCH<1:0> = 01

Comparator CxIND < CxINA Compare
CON = 1, CREF = 0, CCH<1:0> = 10

Comparator VBG/2 < CxINA Compare
CON = 1, CREF = 0, CCH<1:0> = 11

Comparator CxINB < CVREF Compare
CON = 1, CREF = 1, CCH<1:0> = 00

Comparator CxINC < CVREF Compare
CON = 1, CREF = 1, CCH<1:0> = 01

Comparator CxIND < CVREF Compare
CON = 1, CREF = 1, CCH<1:0> = 10

Comparator VBG/2 < CVREF Compare
CON = 1, CREF = 1, CCH<1:0> = 11

27.7 MPLAB SIM Software Simulator

The MPLAB SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC® DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB SIM Software Simulator fully supports symbolic debugging using the MPLAB C Compilers, and the MPASM and MPLAB Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

27.8 MPLAB REAL ICE In-Circuit Emulator System

MPLAB REAL ICE In-Circuit Emulator System is Microchip’s next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs PIC® Flash MCUs and dsPIC® Flash DSCs with the easy-to-use, powerful graphical user interface of the MPLAB Integrated Development Environment (IDE), included with each kit.

The emulator is connected to the design engineer’s PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with in-circuit debugger systems (RJ11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

The emulator is field upgradable through future firmware downloads in MPLAB IDE. In upcoming releases of MPLAB IDE, new devices will be supported, and new features will be added. MPLAB REAL ICE offers significant advantages over competitive emulators including low-cost, full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, a ruggedized probe interface and long (up to three meters) interconnection cables.

27.9 MPLAB ICD 3 In-Circuit Debugger System

MPLAB ICD 3 In-Circuit Debugger System is Microchip's most cost effective high-speed hardware debugger/programmer for Microchip Flash Digital Signal Controller (DSC) and microcontroller (MCU) devices. It debugs and programs PIC® Flash microcontrollers and dsPIC® DSCs with the powerful, yet easy-to-use graphical user interface of MPLAB Integrated Development Environment (IDE).

The MPLAB ICD 3 In-Circuit Debugger probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with a connector compatible with the MPLAB ICD 2 or MPLAB REAL ICE systems (RJ-11). MPLAB ICD 3 supports all MPLAB ICD 2 headers.

27.10 PICkit 3 In-Circuit Debugger/ Programmer and PICkit 3 Debug Express

The MPLAB PICkit 3 allows debugging and programming of PIC® and dsPIC® Flash microcontrollers at a most affordable price point using the powerful graphical user interface of the MPLAB Integrated Development Environment (IDE). The MPLAB PICkit 3 is connected to the design engineer's PC using a full speed USB interface and can be connected to the target via an Microchip debug (RJ-11) connector (compatible with MPLAB ICD 3 and MPLAB REAL ICE). The connector uses two device I/O pins and the reset line to implement in-circuit debugging and In-Circuit Serial Programming™.

The PICkit 3 Debug Express include the PICkit 3, demo board and microcontroller, hookup cables and CDROM with user’s guide, lessons, tutorial, compiler and MPLAB IDE software.

28.0 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.

FIGURE 29-3: BROWN-OUT RESET CHARACTERISTICS

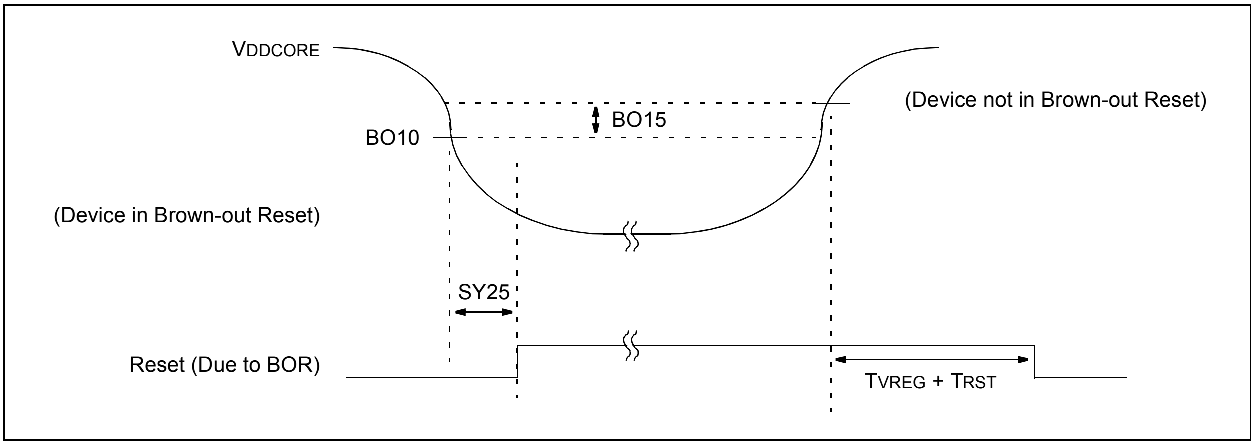


TABLE 29-5: BOR TRIP POINTS

Standard Operating Conditions (unless otherwise stated)
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
DC19	VBOR	BOR Voltage on VDD Transition	BOR = 00	—	—	—	—	LPBOR[1]
			BOR = 01	2.92	3	3.08	V	
			BOR = 10	2.63	2.7	2.77	V	
			BOR = 11	1.75	1.82	1.85	V	
DC14	VBHYS	BOR Hysteresis		—	5	—	mV	

Note 1: LPBOR re-arms the POR circuit, but does not cause a BOR. LPBOR can be used to ensure a POR after the supply voltage rises to a safe operating level. It does not stop code execution after the supply voltage falls below a chosen trip point.

TABLE 29-6: DC CHARACTERISTICS: OPERATING CURRENT (IDD)

DC CHARACTERISTICS			**Standard Operating Conditions: 1.8V to 3.6V (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended			
Parameter No.	**Typical(1)**	**Max**	**Units**	**Conditions**		
IDD Current(2)						
DC20	195	330	µA	-40°C	1.8V	0.5 MIPS, FOSC = 1 MHz
DS20a		330		+25°C		
DC20b		330		+60°C		
DC20c		330		+85°C		
DC20d		500		+125°C		
DC20e	365	590	µA	-40°C	3.3V	
DC20f		590		+25°C		
DC20g		645		+60°C		
DC20h		720		+85°C		
DC20i		800		+125°C		
DC22	363	600	µA	-40°C	1.8V	1 MIPS, FOSC = 2 MHz
DC22a		600		+25°C		
DC22b		600		+60°C		
DC22c		600		+85°C		
DC22d		800		+125°C		
DC22e	695	1100	µA	-40°C	3.3V	
DC22f		1100		+25°C		
DC22g		1100		+60°C		
DC22h		1100		+85°C		
DC22i		1500		+125°C		
DC23	11	18	mA	-40°C	3.3V	16 MIPS, FOSC = 32 MHz
DC23a		18		+25°C		
DC23b		18		+60°C		
DC23c		18		+85°C		
DC23d		18		+125°C		
DC27	2.25	3.40	mA	-40°C	2.5V	FRC (4 MIPS), FOSC = 8 MHz
DC27a		3.40		+25°C		
DC27b		3.40		+60°C		
DC27c		3.40		+85°C		
DC27d		3.40		+125°C		
DC27e	3.05	4.60	mA	-40°C	3.3V	
DC27f		4.60		+25°C		
DC27g		4.60		+60°C		
DC27h		4.60		+85°C		
DC27i		5.40		+125°C		

Note 1: Data in “Typical” column is at 3.3V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

2: Operating Parameters:
- EC mode with clock input driven with a square wave rail-to-rail
- I/Os are configured as outputs, driven low
- $\overline{\text{MCLR}}$ – VDD
- WDT FSCM is disabled
- SRAM, program and data memory are active
- All PMD bits are set except for modules being measured

TABLE 29-22: AC SPECIFICATIONS

Symbol	Characteristics	Min	Typ	Max	Units
TLW	BCLKx High Time	20	TCY/2	—	ns
THW	BCLKx Low Time	20	(TCY * BRGx) + TCY/2	—	ns
TBLD	BCLKx Falling Edge Delay from UxTX	-50	—	50	ns
TBHD	BCLKx Rising Edge Delay from UxTX	TCY/2 – 50	—	TCY/2 + 50	ns
TWAK	Min. Low on UxRX Line to Cause Wake-up	—	1	—	μs
TCTS	Min. 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

TABLE 29-23: A/D CONVERSION TIMING REQUIREMENTS(1)

A/D CHARACTERISTICS			**Standard Operating Conditions: 1.8V 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**	**Max.**	**Units**	**Conditions**
Clock Parameters							
AD50	TAD	A/D Clock Period	75	—	—	ns	TCY = 75 ns, AD1CON3 is in the default state
AD51	TRC	A/D Internal RC Oscillator Period	—	250	—	ns	
Conversion Rate							
AD55	TCONV	Conversion Time	—	12	—	TAD	
AD56	FCNV	Throughput Rate	—	—	500	ksps	AVDD ≥ 2.7V
AD57	TSAMP	Sample Time	—	1	—	TAD	
AD58	TACQ	Acquisition Time	750	—	—	ns	**(Note 2)**
AD59	TSWC	Switching Time from Convert to Sample	—	—	**(Note 3)**		
AD60	TDIS	Discharge Time	0.5	—	—	TAD	
Clock Parameters							
AD61	TPSS	Sample Start Delay from Setting Sample bit (SAMP)	2	—	3	TAD	

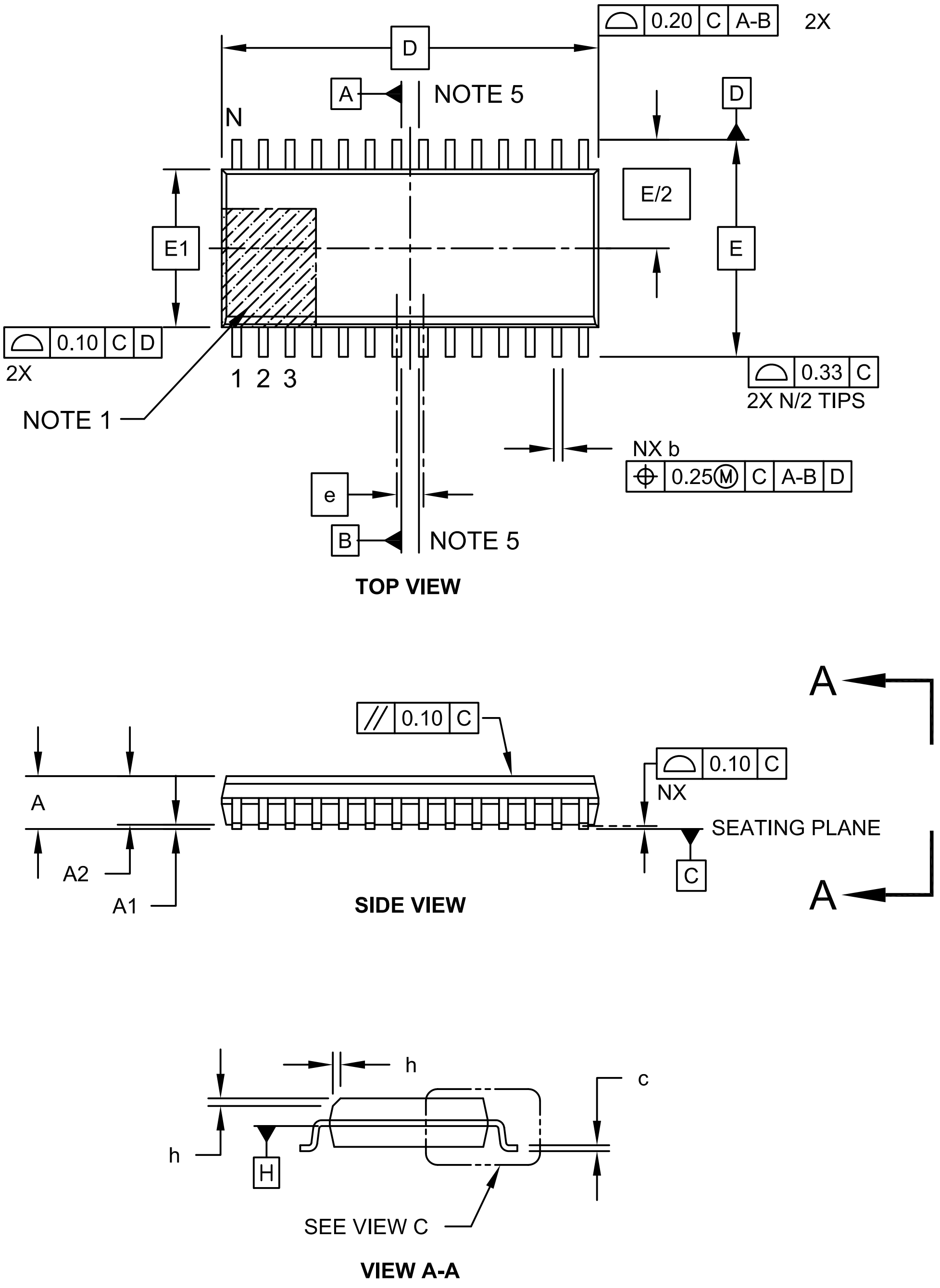
Note 1: Because the sample caps will eventually lose charge, clock rates below 10 kHz can affect linearity performance, especially at elevated temperatures.

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.

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

APPENDIX A: REVISION HISTORY

Revision A (November 2008)

Original data sheet for the PIC24F16KA102 family of devices.

Revision B (March 2009)

Section 29.0 "Electrical Characteristics" was revised and minor text edits were made throughout the document.

Revision C (October 2011)

- Changed all instances of DSWSRC to DSWAKE.
- Corrected Example 5-2.
- Corrected Example 5-4.
- Corrected Example 6-1.
- Corrected Example 6-3.
- Added a comment to Example 6-5.
- Corrected Figure 9-1 to connect the SOSCI and SOSCO pins to the Schmitt trigger correctly.
- Added register descriptions for PMD1, PMD2, PMD3 and PMD4.
- Added note that RTCC will run in Reset.
- Corrected time values of ADCS (AD1CON3<5:0>).
- Corrected CH0SB and CH0SA (AD1CHS<11:8> and AD1CHS<3:0>) to correctly reference AVDD and AN3.
- Added description of PGCF15 and PGCF14 (AD1PCFG<15:14>).
- Edited Figure 22-2 to correctly reference RIC and the A/D capacitance.
- Changed all references from CTEDG1 to CTED1.
- Changed all references from CTEDG2 to CTED2.
- Changed description of CMIDL: it used to say it disables all comparators in Idle, now only disables interrupts in Idle mode.
- Changed all references of RTCCKSEL to RTCOSC.
- Changed all references of DSLPBOR to DSBOREN.
- Changed all references of DSWCKSEL to DSWDTOSC
- Imported Figure 40-9 from PIC24F FRM, Section 40.
- Added spec for BOR hysteresis.
- Edited Note 1 for Table 29-5 to further describe LPBOR.
- Edited max values of DC20d and DC20e on Table 29-6.
- Edited typical value for DC61-DC61c in Table 29-8.
- Edited Note 2 of Table 29-8.
- Added Note 5 to Table 29-9.
- Added Table 29-15.
- Added AD08 and AD09 in Table 29-26.
- Added Note 3 to Table 29-26.
- Imported Figure 40.10 from PIC24F FRM, Section 40.
- Deleted TVREG spec.
- Imported Figure 15-5 from PIC24F FRM, Section 15.
- Imported Table 15-4 from PIC24F FRM, Section 15.
- Imported Figure 16-22 from PIC24F FRM, Section 16.
- Imported Table 16-9 from PIC24F FRM, Section 16.
- Imported Figure 16-23 from PIC24F FRM, Section 16.
- Imported Table 16-10 from PIC24F FRM, Section 16.
- Imported Figure 21-24 from PIC24F FRM. Section 21.
- Imported Figure 21-25 from PIC24F FRM, Section 21.
- Imported Table 21-5 from PIC24F FRM, Section 21.
- Imported Figure 23-17 from PIC24F FRM, Section 23.
- Imported Table 23-3 from PIC24F FRM, Section 23.
- Imported Figure 23-18 from PIC24F FRM, Section 23.
- Imported Table 23-4 from PIC24F FRM, Section 23.
- Imported Figure 23-19 from PIC24F FRM, Section 23.
- Imported Table 23-5 from PIC24F FRM, Section 23.
- Imported Figure 23-20 from PIC24F FRM, Section 23.
- Imported Table 23-6 from PIC24F FRM, Section 23.
- Imported Figure 24-33 from PIC24F FRM, Section 24.
- Imported Table 24-6 from PIC24F FRM, Section 24.
- Imported Figure 24-34 from PIC24F FRM, Section 24.
- Imported Table 24-7 from PIC24F FRM, Section 24.
- Imported Figure 24-35 from PIC24F FRM, Section 24.
- Imported Table 24-8 from PIC24F FRM, Section 24.
- Imported Figure 24-36 from PIC24F FRM, Section 24.
- Imported Table 24-9 from PIC24F FRM, Section 24.

INDEX