Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	16-Bit
Speed	32MHz
Connectivity	I²C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, HLVD, POR, PWM, WDT
Number of I/O	17
Program Memory Size	16KB (5.5K x 24)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5.5V
Data Converters	A/D 12x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SOIC (0.295", 7.50mm Width)
Supplier Device Package	20-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24fv16ka301-i-so

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
$\overline{U1CTS}$	12	17	14	44	48	12	17	14	44	48	I	ST	UART1 Clear-to-Send Input
$\overline{U1RTS}$	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
$\overline{U2CTS}$	10	12	9	34	37	10	12	9	34	37	I	ST	UART2 Clear-to-Send Input
$\overline{U2RTS}$	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
V_{CAP}	—	—	—	—	—	14	20	17	7	7	P	—	Core Power
V_{DD}	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
V_{REF+}	2	2	27	19	21	2	2	27	19	21	I	ANA	A/D Reference Voltage Input (+)
V_{REF-}	3	3	28	20	22	3	3	28	20	22	I	ANA	A/D Reference Voltage Input (-)
V_{SS}	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

NOTES:

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.

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
```

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 8-32: 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

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.

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 *"PIC24F Family Reference Manual"*, **Section 12. "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.

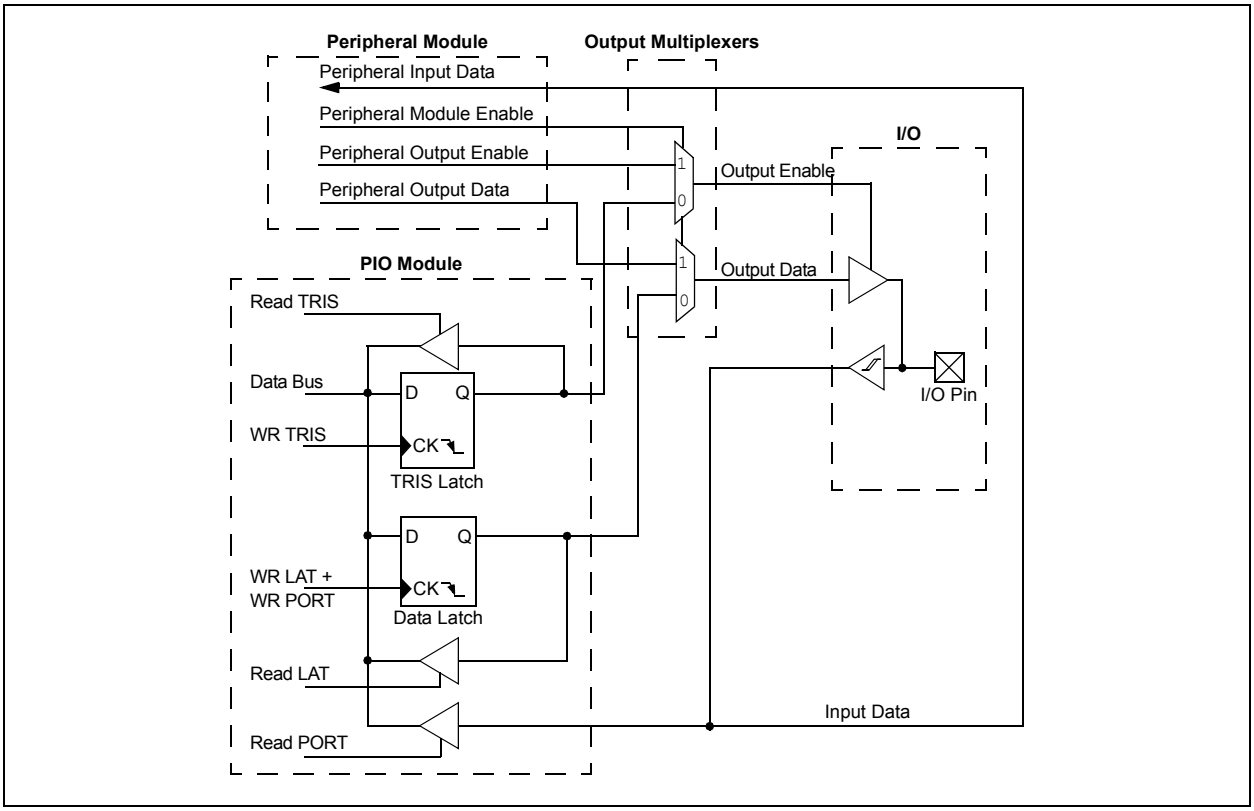
All port pins have three registers directly associated with their operation as digital I/O. The Data Direction register (TRISx) determines whether the pin is an input or an output. If the data direction bit is a '1', then the pin is an input. All port pins are defined as inputs after a Reset. Reads from 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



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");
```

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
—	—	—	—	—	—	CTMEN17	CTMEN16
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
CTMEN15	CTMEN14	CTMEN13	CTMEN12	CTMUEN11	CTMEN10	CTMEN9	CTMEN8
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
CTMEN7	CTMEN6	CTMEN5	CTMEN4	CTMEN3	CTMEN2	CTMEN1	CTMEN0
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'.

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
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 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)

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
		-5	—	+5	%	-40°C ≤ TA ≤ +85°C	1.8V ≤ VDD ≤ 3.6V, F device 2.0V ≤ VDD ≤ 5.5V, FV device
	LPRC @ 31 kHz(2)						
F21		-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 29-19: SPIx MODULE MASTER MODE TIMING CHARACTERISTICS (CKE = 1)

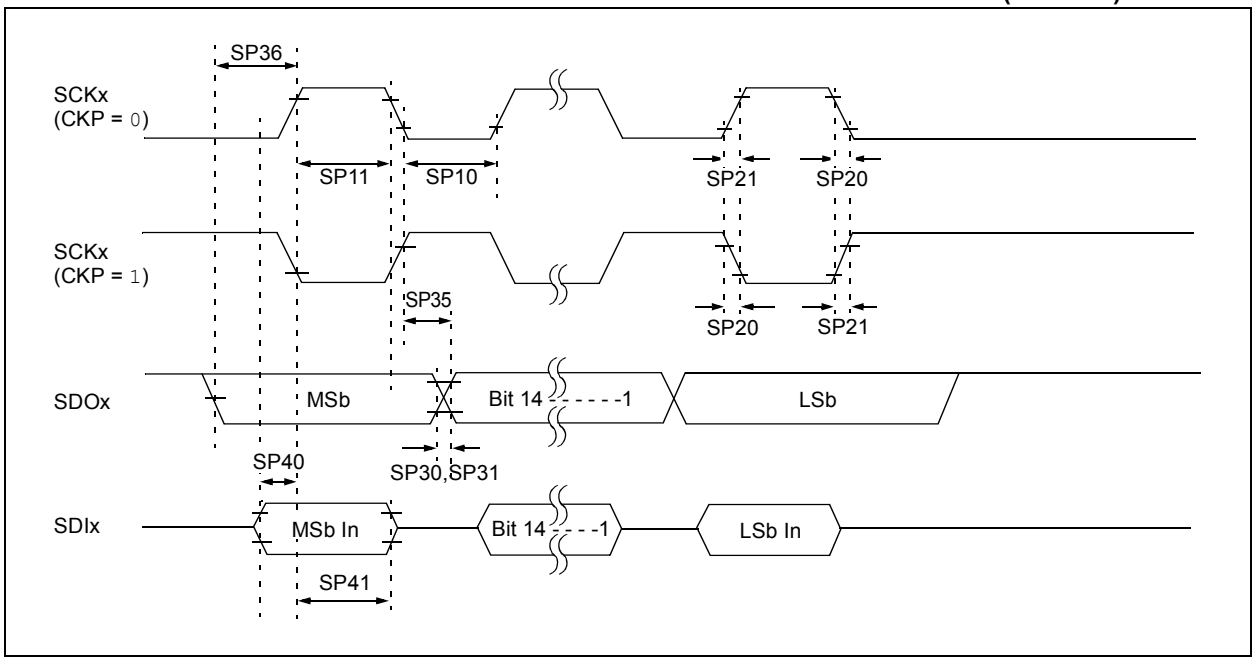


TABLE 29-37: SPIx MODULE MASTER MODE TIMING REQUIREMENTS (CKE = 1)

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 29-21: SPIx MODULE SLAVE MODE TIMING CHARACTERISTICS (CKE = 1)

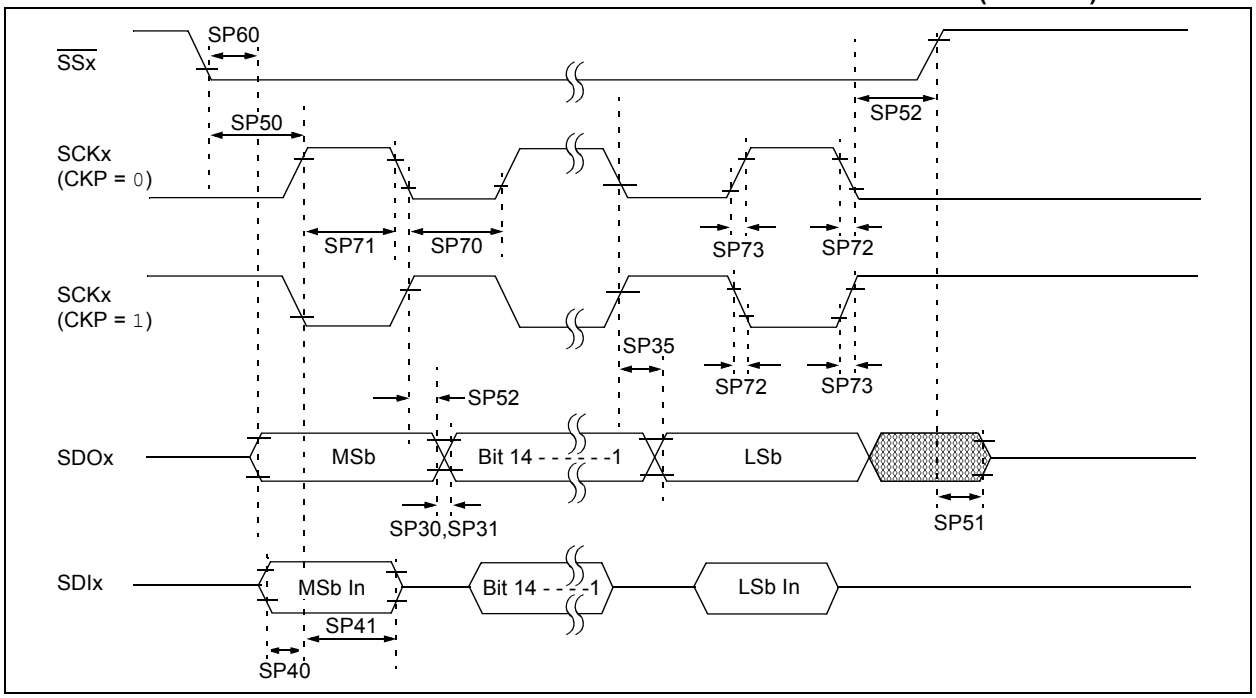


TABLE 29-39: SPIx MODULE SLAVE MODE TIMING REQUIREMENTS (CKE = 1)

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	$\overline{SSx}$ ↓ 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	
SP60	TssL2doV	SDOx Data Output Valid after $\overline{SSx}$ 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.

FIGURE 30-14: TYPICAL AND MAXIMUM IPD vs. VDD

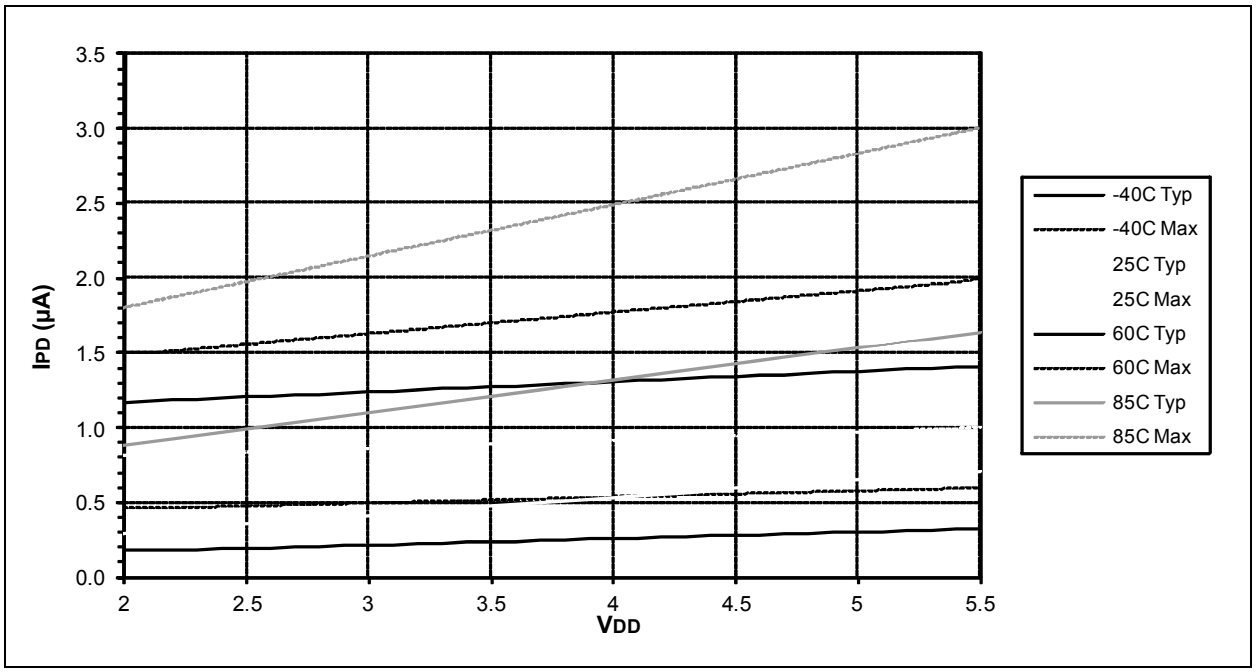
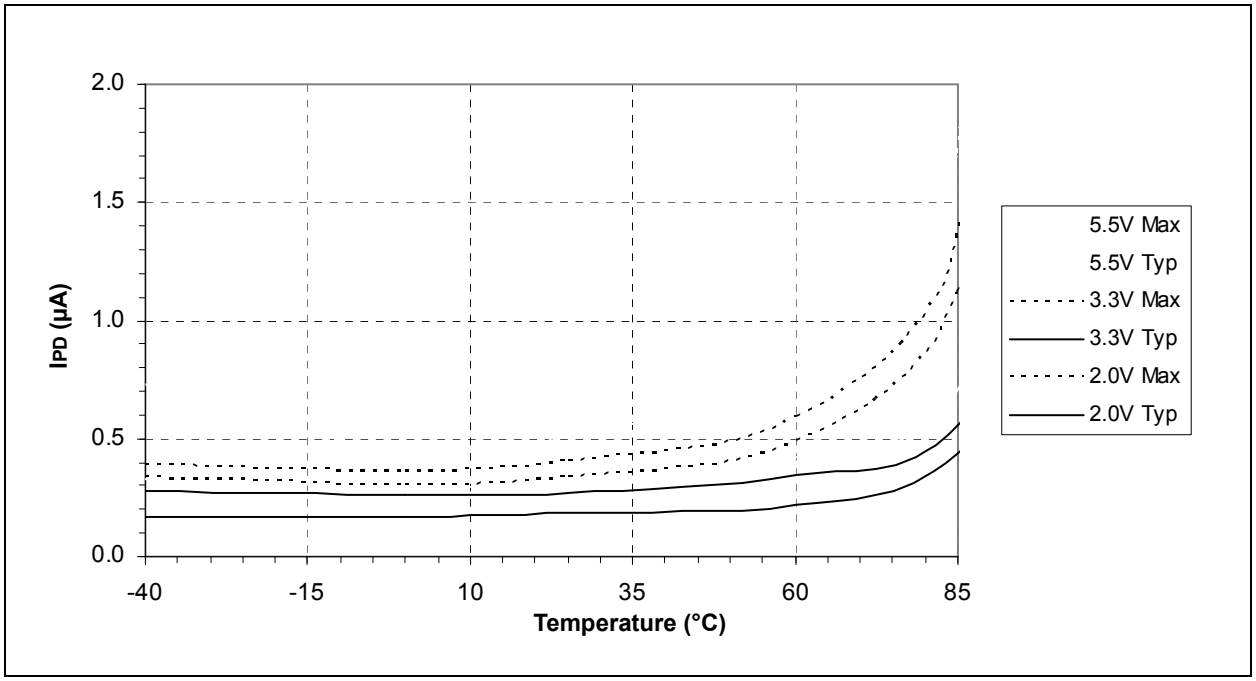


FIGURE 30-15: TYPICAL AND MAXIMUM IPD vs. TEMPERATURE

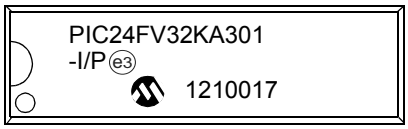


31.0 PACKAGING INFORMATION

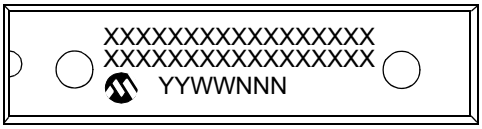
31.1 Package Marking Information

20-Lead PDIP (300 mil)

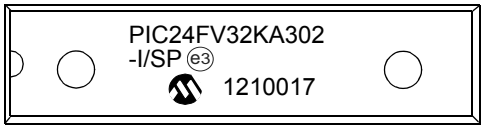
Example



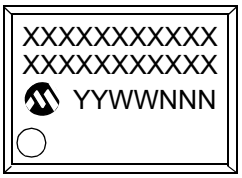
28-Lead SPDIP (.300")



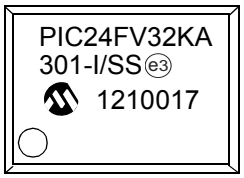
Example



20-Lead SSOP (5.30 mm)



Example



28-Lead SSOP (5.30 mm)

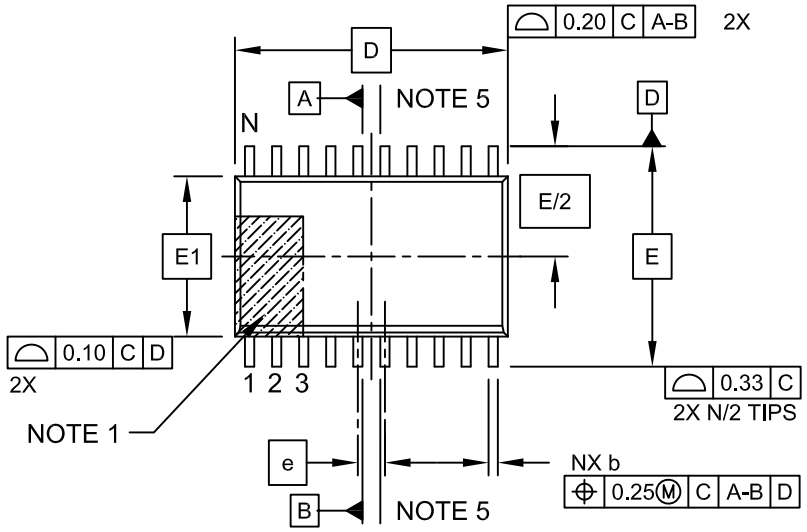
Example

Legend:	XX...X	Product-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week '01')
	NNN	Alphanumeric traceability code
	(e3)	Pb-free JEDEC designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator ((e3)) can be found on the outer packaging for this package.

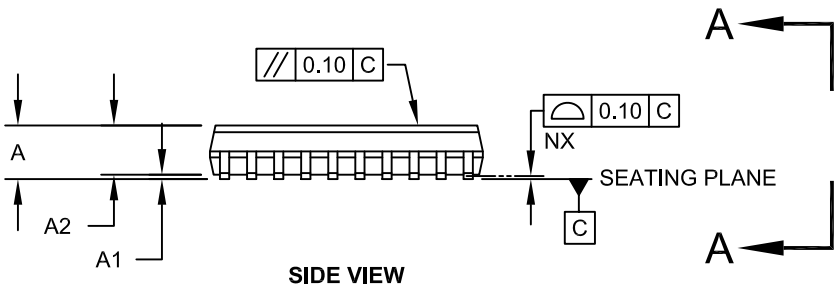
Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

20-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

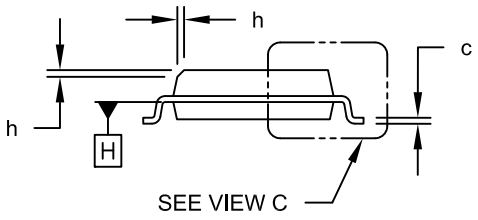
Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



TOP VIEW



SIDE VIEW

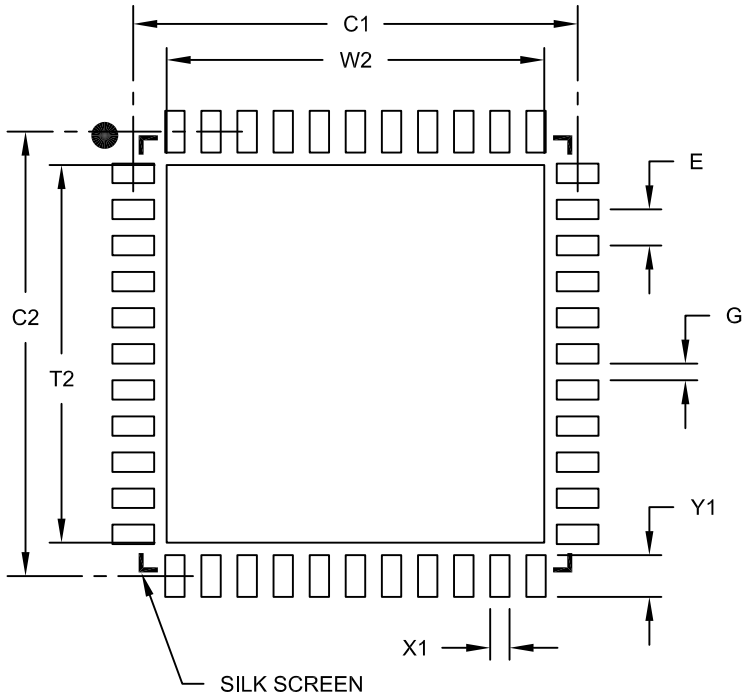


VIEW A-A

Microchip Technology Drawing C04-094C Sheet 1 of 2

44-Lead Plastic Quad Flat, No Lead Package (ML) - 8x8 mm Body [QFN]

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			6.60
Optional Center Pad Length	T2			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
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. C04-2103B