Welcome to E-XFL.COM

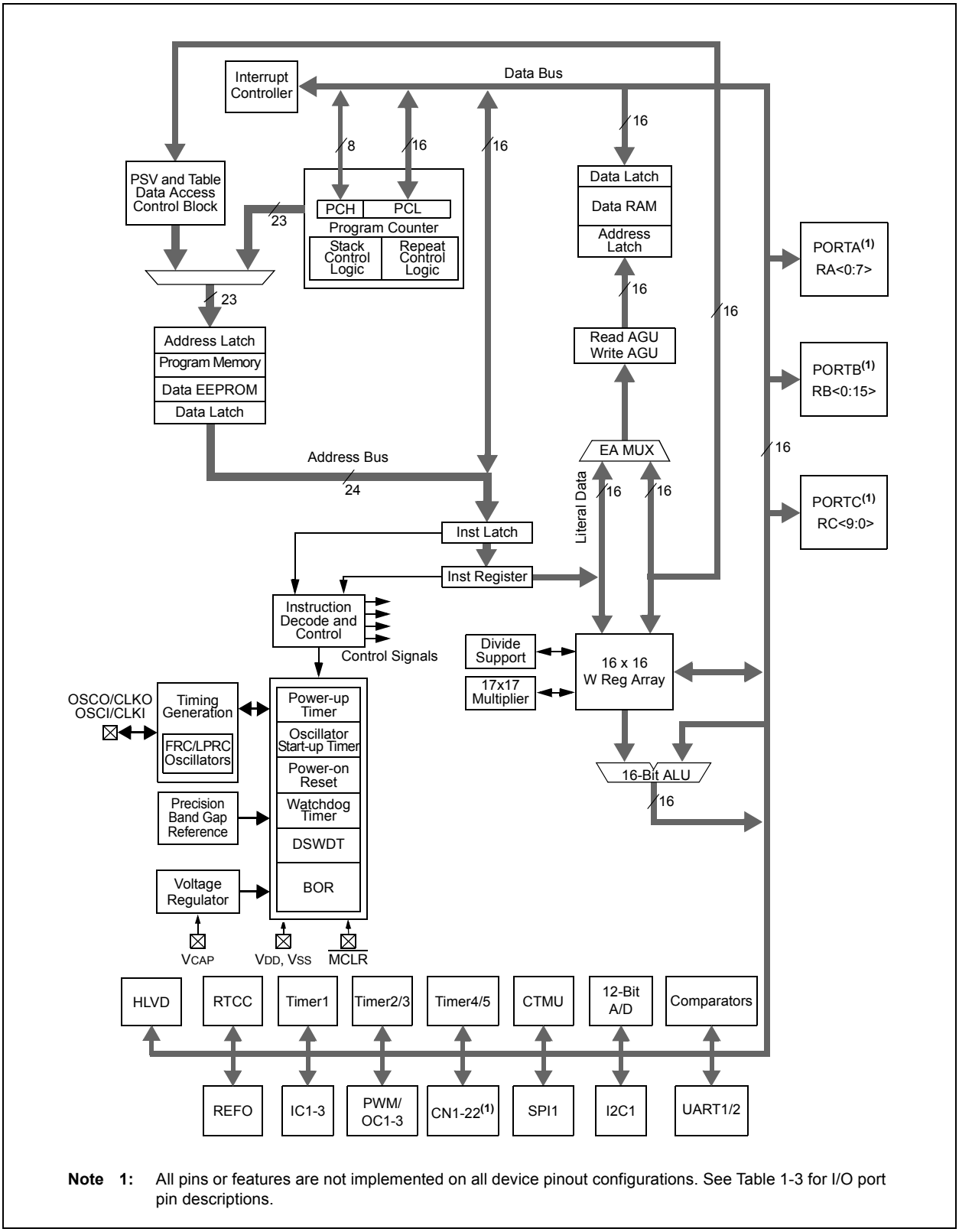
What is "Embedded - Microcontrollers"?

"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	24
Program Memory Size	32KB (11K x 24)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 13x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SSOP (0.209", 5.30mm Width)
Supplier Device Package	28-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24f32ka302-i-ss

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
AV_{DD}	20	28	25	17	18	20	28	25	17	18	I	ANA	A/D Supply Pins
AV_{SS}	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	

NOTES:

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
```

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 8-22: 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 8-28: 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'

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.

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.

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 *"PIC24F Family Reference Manual"*, **Section 51. "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.

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

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

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**
IDD Current						
D20	PIC24FV32KA3XX	269	450	µA	2.0V	0.5 MIPS, FOSC = 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, FOSC = 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, FOSC = 32 MHz(1)
	PIC24F32KA3XX	12.0	18.0	mA	3.3V	
DC26	PIC24FV32KA3XX	2.0	—	mA	2.0V	FRC (4 MIPS), FOSC = 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), FOSC = 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 29-10: 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 29-24: 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

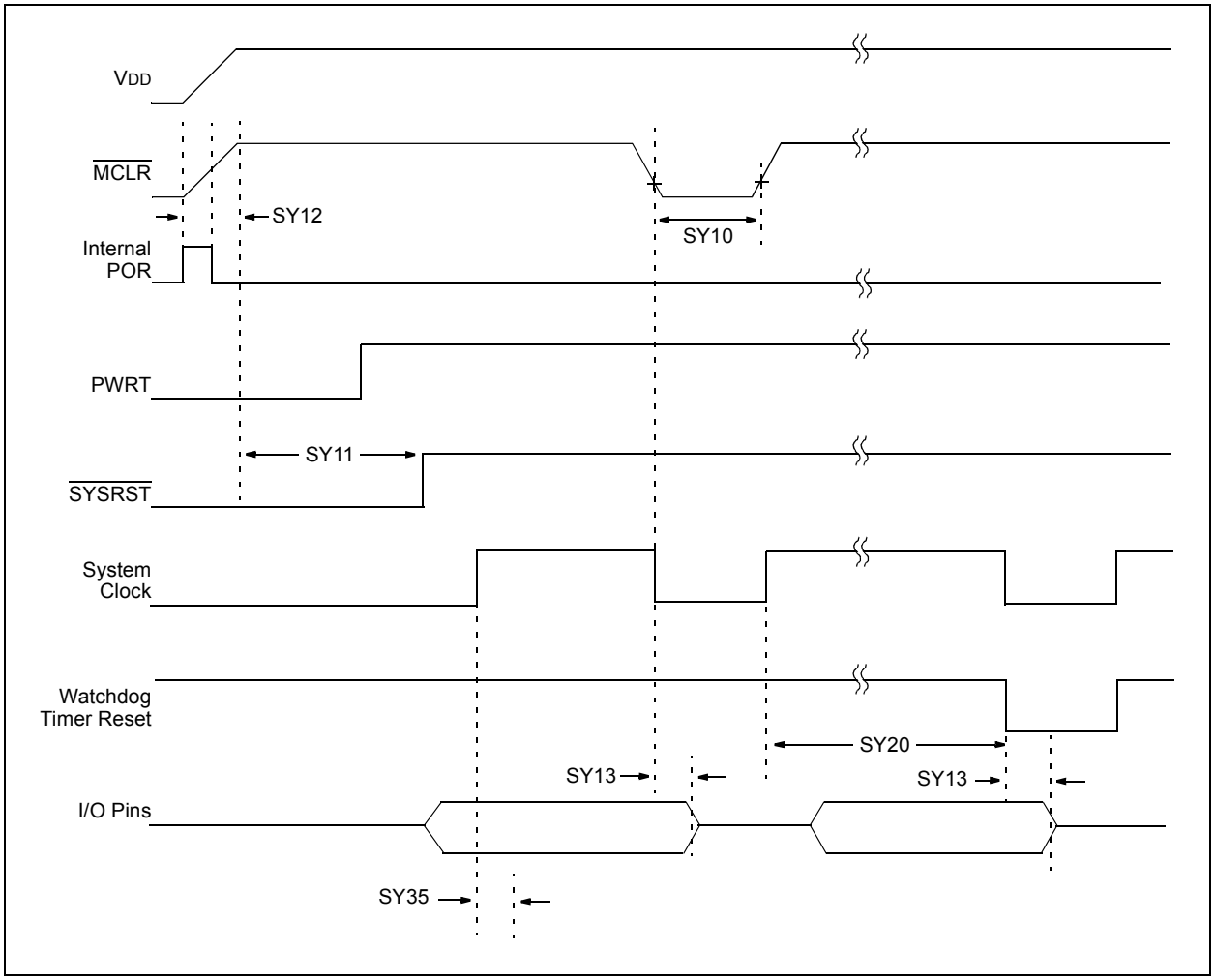


FIGURE 30-38: CTMU OUTPUT CURRENT vs. TEMPERATURE (IRNG<1:0> = 01, 2.0V ≤ VDD ≤ 5.5V)

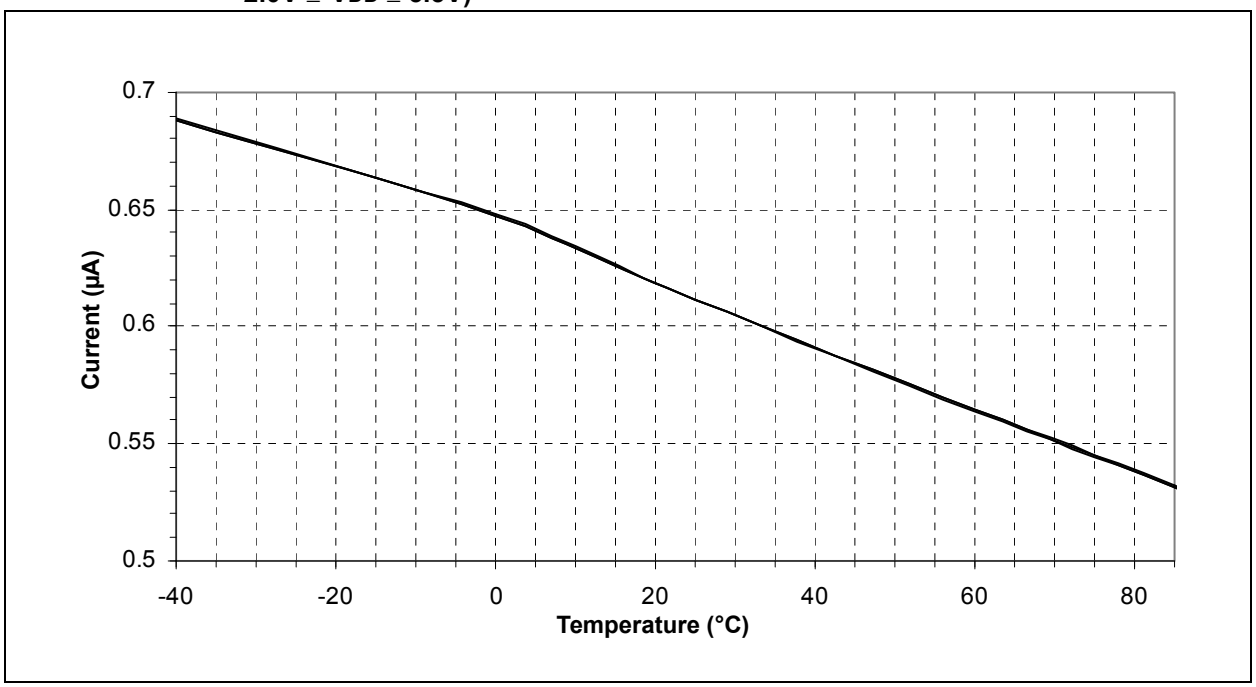


FIGURE 30-39: CTMU OUTPUT CURRENT vs. VDD (IRNG<1:0> = 01)

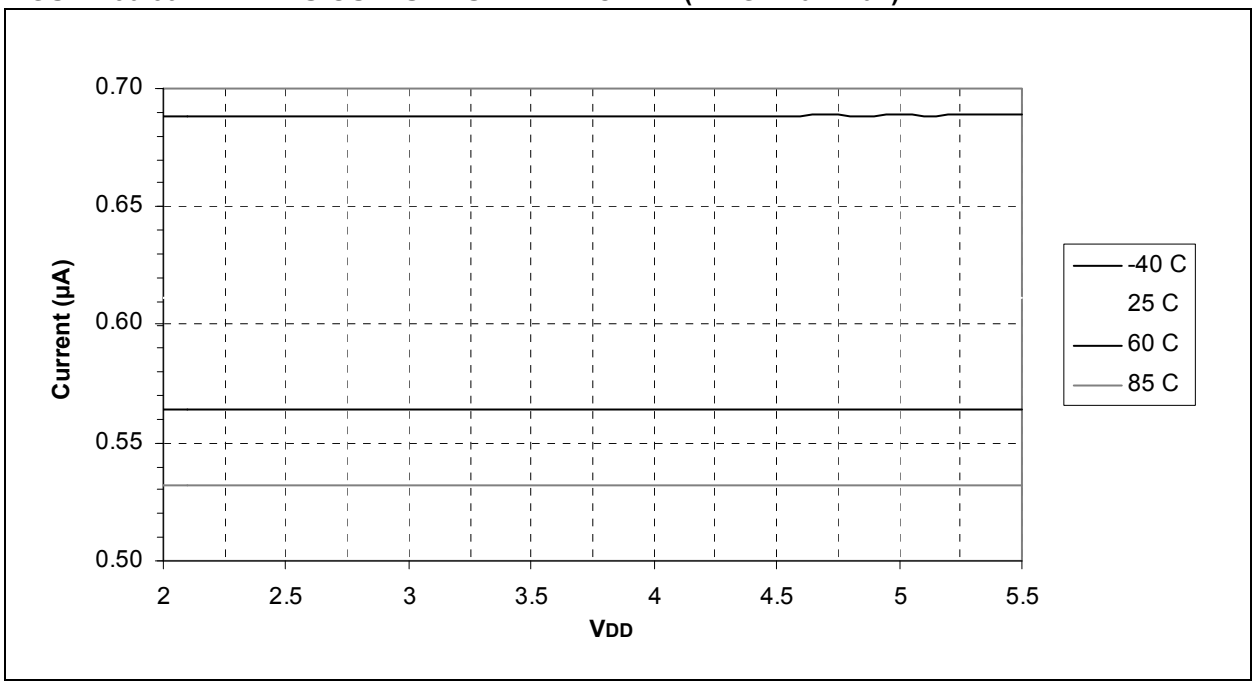


FIGURE 30-46: TYPICAL ΔIWDT vs. VDD

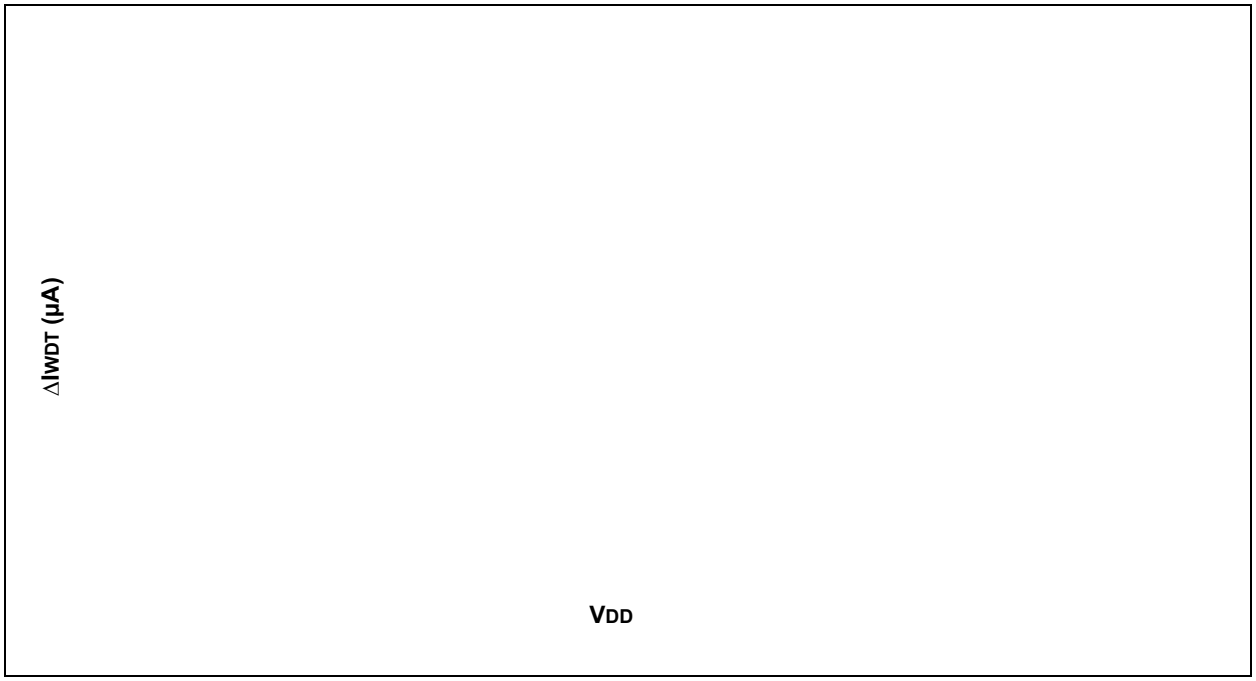


FIGURE 30-47: TYPICAL ΔIDSBOR vs. VDD

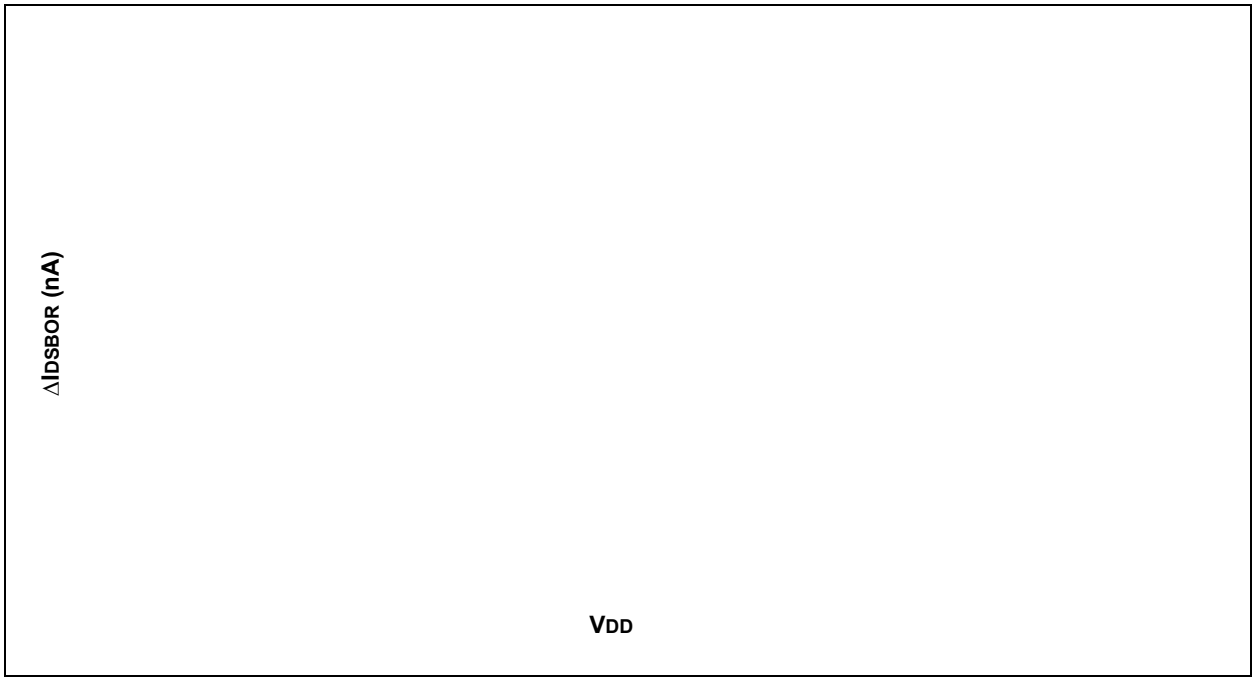


FIGURE 30-53: VIL/VIH vs. VDD (OSCO, TEMPERATURES AS NOTED)

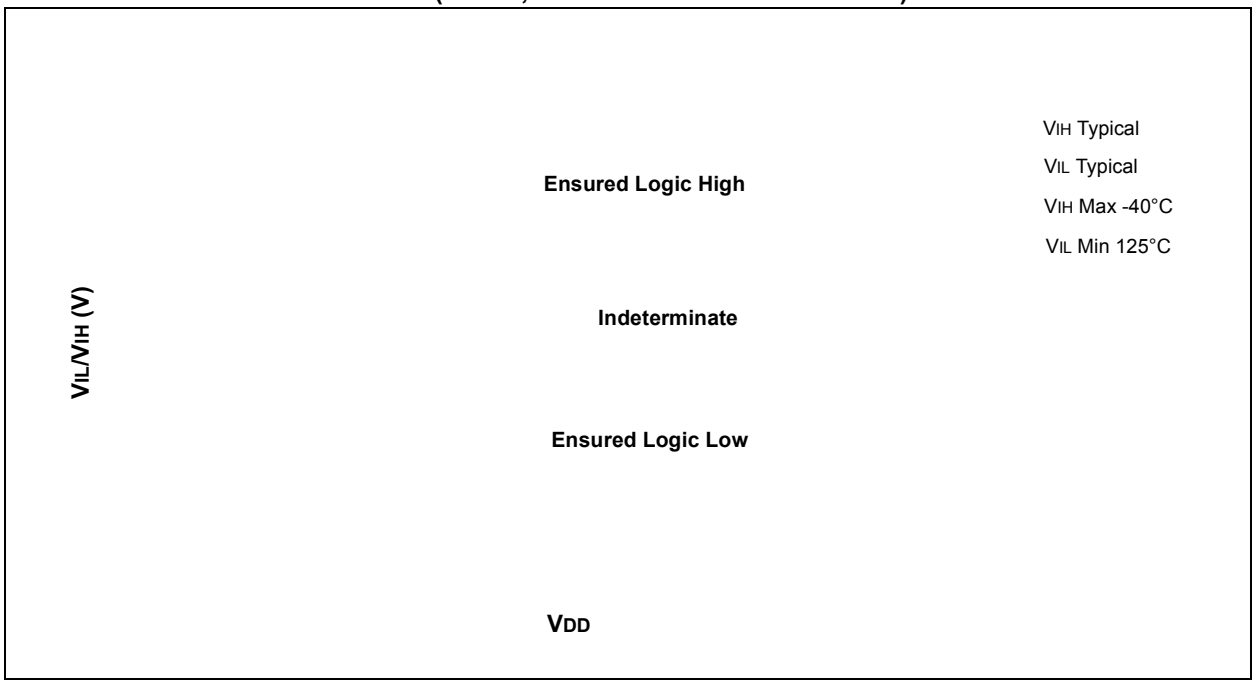
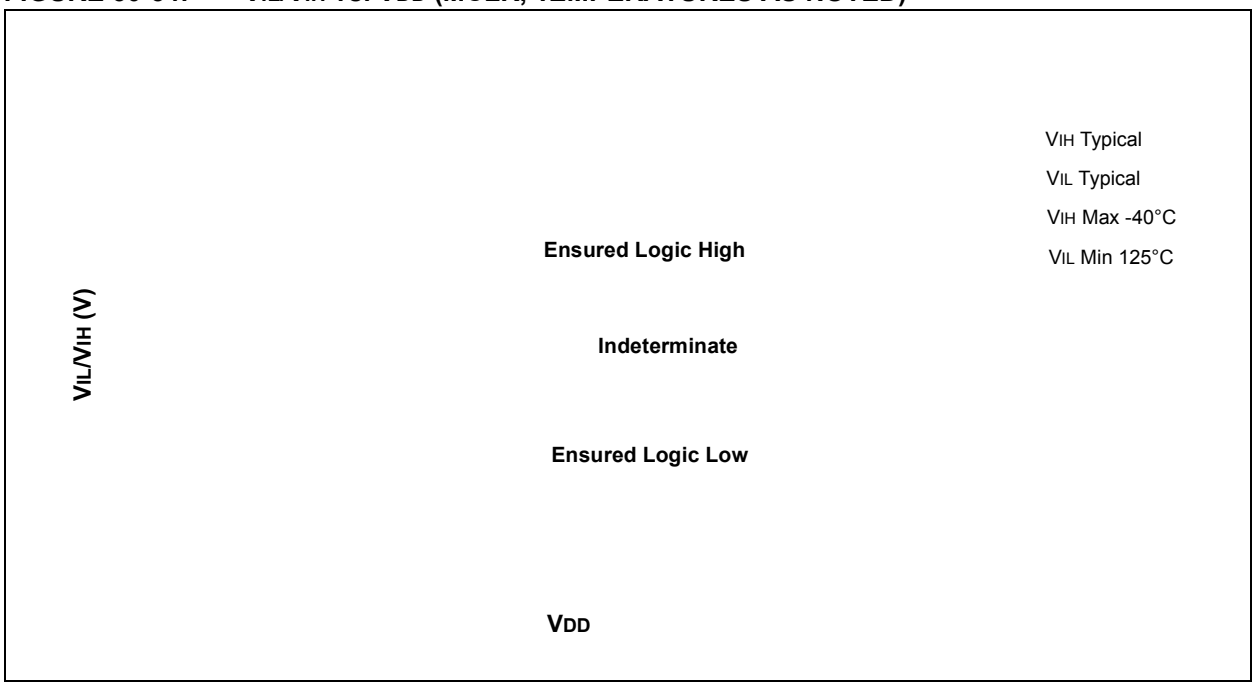


FIGURE 30-54: VIL/VIH vs. VDD ($\overline{\text{MCLR}}$, TEMPERATURES AS NOTED)



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.