Microchip Technology - PIC24F16KA301-E/P Datasheet

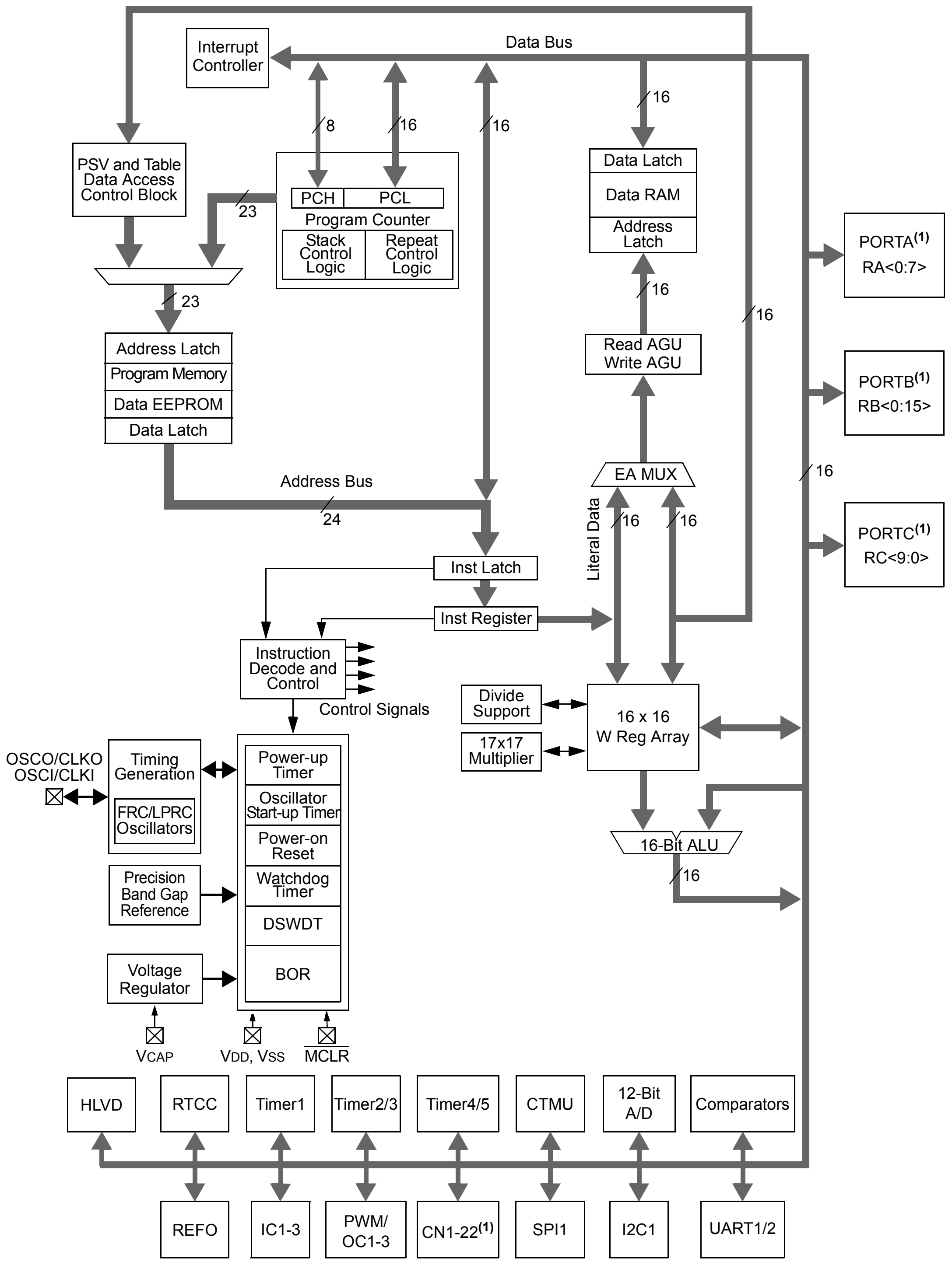
What is "Embedded - Microcontrollers"?

"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	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	18
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)	1.8V ~ 3.6V
Data Converters	A/D 12x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Through Hole
Package / Case	20-DIP (0.300", 7.62mm)
Supplier Device Package	20-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24f16ka301-e-p

FIGURE 1-1: PIC24FV32KA304 FAMILY GENERAL BLOCK DIAGRAM

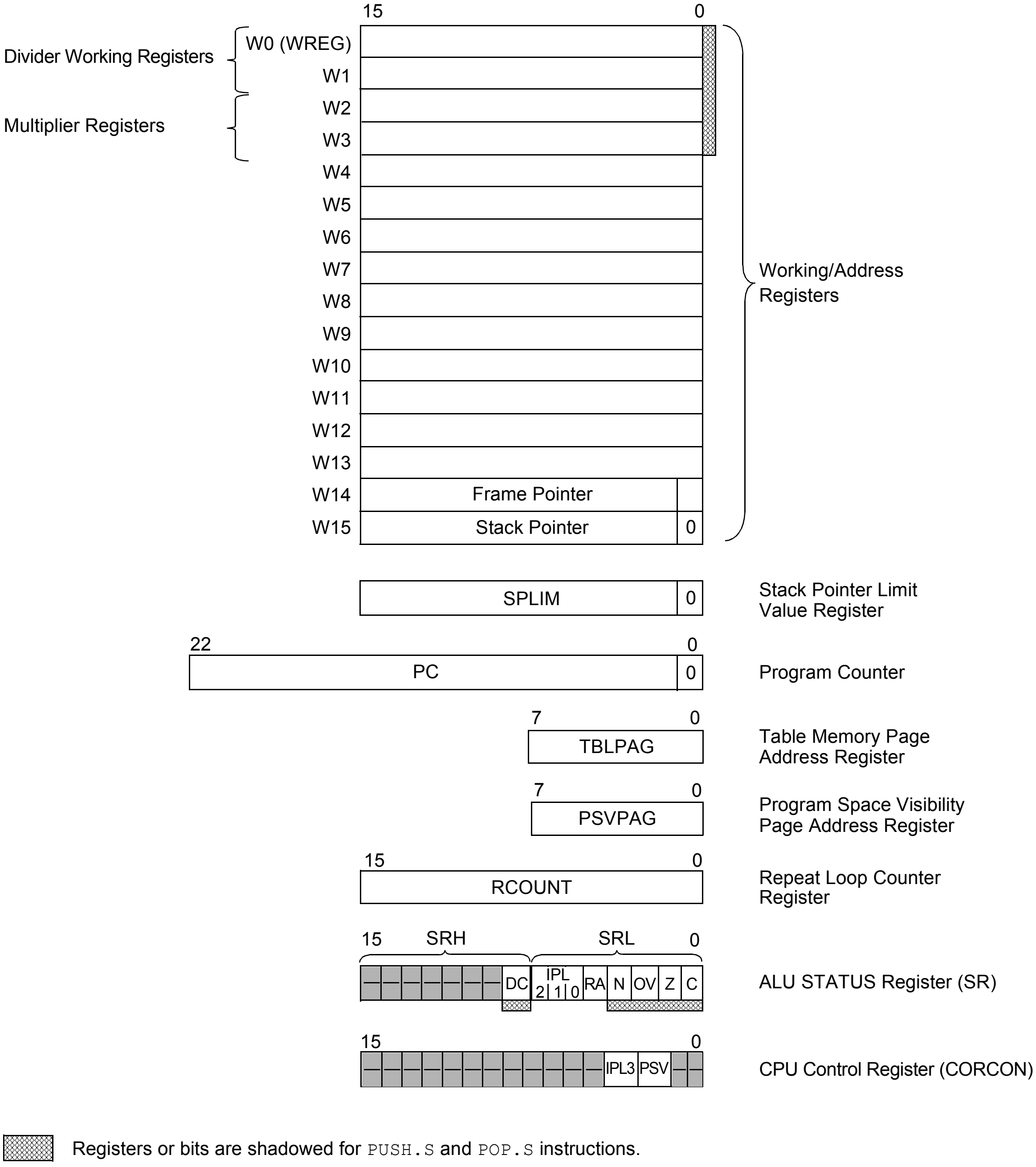


Note 1: All pins or features are not implemented on all device pinout configurations. See Table 1-3 for I/O port pin descriptions.

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^{2}C™	Alternate I2C1 Clock Input/Output
ASDA1	—	14	11	41	45	—	14	11	41	45	I/O	I^{2}C	Alternate I2C1 Data Input/Output
AVDD	20	28	25	17	18	20	28	25	17	18	I	ANA	A/D Supply Pins
AVSS	19	27	24	16	17	19	27	24	16	17	I	ANA	
C1INA	8	7	4	24	26	8	7	4	24	26	I	ANA	Comparator 1 Input A (+)
C1INB	7	6	3	23	25	7	6	3	23	25	I	ANA	Comparator 1 Input B (-)
C1INC	5	5	2	22	24	5	5	2	22	24	I	ANA	Comparator 1 Input C (+)
C1IND	4	4	1	21	23	4	4	1	21	23	I	ANA	Comparator 1 Input D (-)
C1OUT	17	25	22	14	15	17	25	22	14	15	O	—	Comparator 1 Output
C2INA	5	5	2	22	24	5	5	2	22	24	I	ANA	Comparator 2 Input A (+)
C2INB	4	4	1	21	23	4	4	1	21	23	I	ANA	Comparator 2 Input B (-)
C2INC	8	7	4	24	26	8	7	4	24	26	I	ANA	Comparator 2 Input C (+)
C2IND	7	6	3	23	25	7	6	3	23	25	I	ANA	Comparator 2 Input D (-)
C2OUT	14	20	17	7	7	11	16	13	43	47	O	—	Comparator 2 Output

FIGURE 3-2: PROGRAMMER'S MODEL



▨ Registers or bits are shadowed for `PUSH.S` and `POP.S` instructions.

TABLE 4-3: CPU CORE REGISTERS MAP

File Name	Start Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
WREG0	0000	WREG0																0000
WREG1	0002	WREG1																0000
WREG2	0004	WREG2																0000
WREG3	0006	WREG3																0000
WREG4	0008	WREG4																0000
WREG5	000A	WREG5																0000
WREG6	000C	WREG6																0000
WREG7	000E	WREG7																0000
WREG8	0010	WREG8																0000
WREG9	0012	WREG9																0000
WREG10	0014	WREG10																0000
WREG11	0016	WREG11																0000
WREG12	0018	WREG12																0000
WREG13	001A	WREG13																0000
WREG14	001C	WREG14																0000
WREG15	001E	WREG15																0000
SPLIM	0020	SPLIM																xxxx
PCL	002E	PCL																0000
PCH	0030	—	—	—	—	—	—	—	—	—	PCH							0000
TBLPAG	0032	—	—	—	—	—	—	—	—	TBLPAG								0000
PSVPAG	0034	—	—	—	—	—	—	—	—	PSVPAG								0000
RCOUNT	0036	RCOUNT																xxxxx
SR	0042	—	—	—	—	—	—	—	DC	IPL2	IPL1	IPL0	RA	N	OV	Z	C	0000
CORCON	0044	—	—	—	—	—	—	—	—	—	—	—	—	IPL3	PSV	—	—	0000
DISICNT	0052	—	—	DISICNT														xxxx

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

TABLE 4-8: OUTPUT COMPARE REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
OC1CON1	0190	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	ENFLT2	ENFLT1	ENFLT0	OCFLT2	OCFLT1	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000
OC1CON2	0192	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	DCB1	DCB0	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000C
OC1RS	0194	OC1RS																0000
OC1R	0196	OC1R																0000
OC1TMR	0198	OC1TMR																xxxx
OC2CON1	019A	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	ENFLT2	ENFLT1	ENFLT0	OCFLT2	OCFLT1	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000
OC2CON2	019C	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	DCB1	DCB0	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000C
OC2RS	019E	OC2RS																0000
OC2R	01A0	OC2R																0000
OC2TMR	01A2	OC2TMR																xxxx
OC3CON1	01A4	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	ENFLT2	ENFLT1	ENFLT0	OCFLT2	OCFLT1	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000
OC3CON2	01A6	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	DCB1	DCB0	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000C
OC3RS	01A8	OC3RS																0000
OC3R	01AA	OC3R																0000
OC3TMR	01AC	OC3TMR																xxxx

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

5.5.1 PROGRAMMING ALGORITHM FOR FLASH PROGRAM MEMORY

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

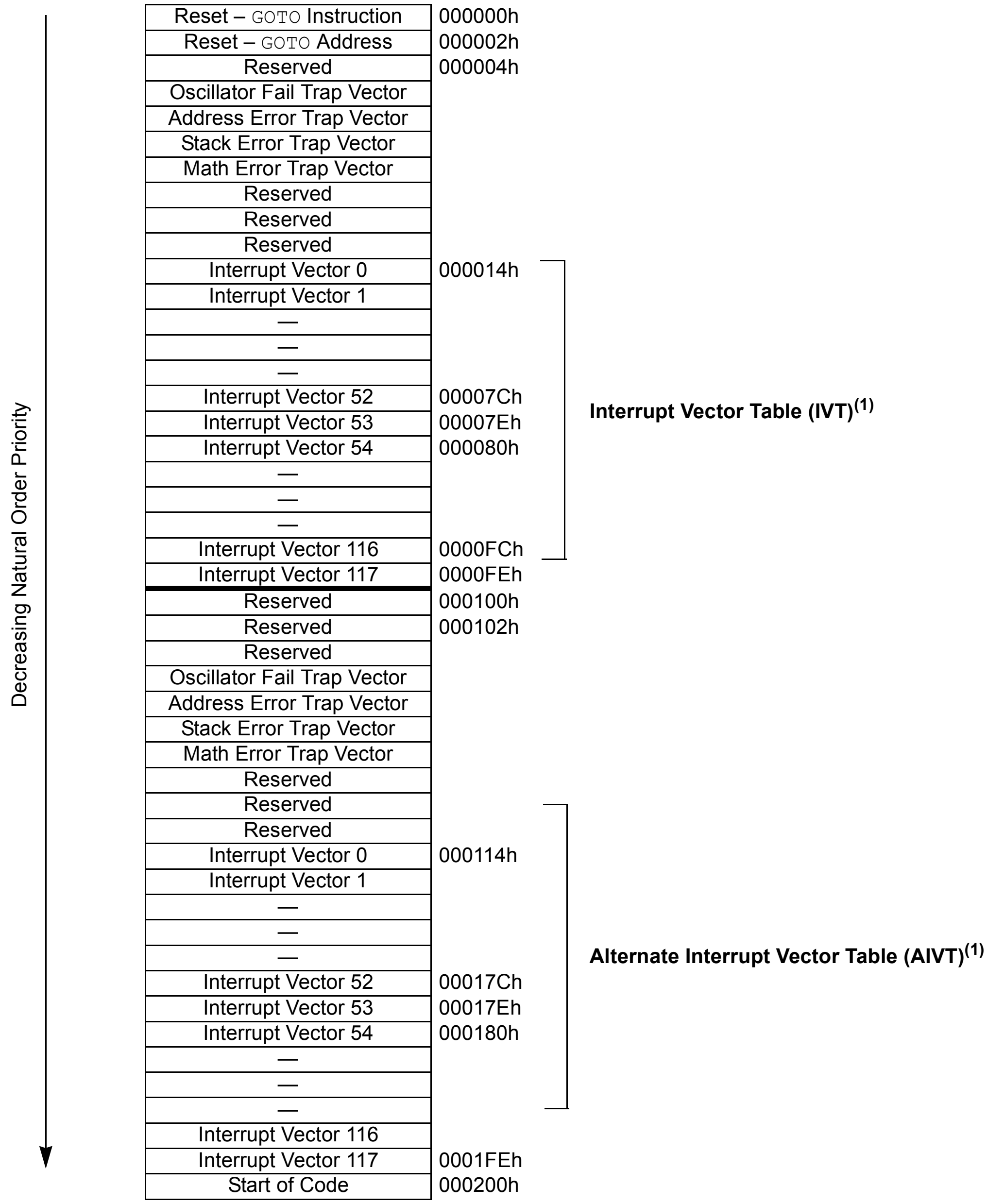
1. Read a row of program memory (32 instructions) and store in data RAM.
2. Update the program data in RAM with the desired new data.
3. Erase a row (see Example 5-1):
    a) Set the NVMOPx bits (NVMCON<5:0>) to '`011000`' to configure for row erase. Set the ERASE (NVMCON<6>) and WREN (NVMCON<14>) bits.
    b) Write the starting address of the block to be erased into the TBLPAG and W registers.
    c) Write 55h to NVMKEY.
    d) Write AAh to NVMKEY.
    e) Set the WR bit (NVMCON<15>). The erase cycle begins and the CPU stalls for the duration of the erase cycle. When the erase is done, the WR bit is cleared automatically.
4. Write the first 32 instructions from data RAM into the program memory buffers (see Example 5-1).
5. Write the program block to Flash memory:
    a) Set the NVMOPx bits to '`011000`' to configure for row programming. Clear the ERASE bit and set the WREN bit.
    b) Write 55h to NVMKEY.
    c) Write AAh to NVMKEY.
    d) Set the WR bit. The programming cycle begins and the CPU stalls for the duration of the write cycle. When the write to Flash memory is done, the WR bit is cleared automatically.

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

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

```
; Set up NVMCON for row erase operation
       MOV     #0x4058, W0                  ;
       MOV     W0, NVMCON                   ; Initialize NVMCON
; Init pointer to row to be ERASED
       MOV     #tblpage(PROG_ADDR), W0      ;
       MOV     W0, TBLPAG                   ; Initialize PM Page Boundary SFR
       MOV     #tbloffset(PROG_ADDR), W0    ; Initialize in-page EA[15:0] pointer
       TBLWTL  W0, [W0]                     ; Set base address of erase block
       DISI    #5                           ; Block all interrupts
                                              for next 5 instructions
       MOV     #0x55, W0
       MOV     W0, NVMKEY                   ; Write the 55 key
       MOV     #0xAA, W1                    ;
       MOV     W1, NVMKEY                   ; Write the AA key
       BSET    NVMCON, #WR                  ; Start the erase sequence
       NOP                                  ; Insert two NOPs after the erase
       NOP                                  ; command is asserted
```

FIGURE 8-1: PIC24F INTERRUPT VECTOR TABLE



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

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

REGISTER 8-6: IFS1: INTERRUPT FLAG STATUS REGISTER 1

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

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

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

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

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

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

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

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

bit 10 **Unimplemented:** Read as ‘0’

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

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

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

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

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

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

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

REGISTER 8-19: IPC2: INTERRUPT PRIORITY CONTROL REGISTER 2

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	U1RXIP2	U1RXIP1	U1RXIP0	—	SPI1IP2	SPI1IP1	SPI1IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	SPF1IP2	SPF1IP1	SPF1IP0	—	T3IP2	T3IP1	T3IP0
bit 7							bit 0

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

bit 15 **Unimplemented:** Read as ‘`0`’

bit 14-12 **U1RXIP<2:0>:** UART1 Receiver Interrupt Priority bits
`111` = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
`001` = Interrupt is Priority 1
`000` = Interrupt source is disabled

bit 11 **Unimplemented:** Read as ‘`0`’

bit 10-8 **SPI1IP<2:0>:** SPI1 Event Interrupt Priority bits
`111` = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
`001` = Interrupt is Priority 1
`000` = Interrupt source is disabled

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

bit 6-4 **SPF1IP<2:0>:** SPI1 Fault Interrupt Priority bits
`111` = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
`001` = Interrupt is Priority 1
`000` = Interrupt source is disabled

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

bit 2-0 **T3IP<2:0>:** Timer3 Interrupt Priority bits
`111` = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
`001` = Interrupt is Priority 1
`000` = Interrupt source is disabled

NOTES:

11.2.2 I/O PORT WRITE/READ TIMING

One instruction cycle is required between a port direction change or port write operation and a read operation of the same port. Typically, this instruction would be a `NOP`.

11.3 Input Change Notification (ICN)

The Input Change Notification function of the I/O ports allows the PIC24FV32KA304 family of devices to generate interrupt requests to the processor in response to a Change-of-State (COS) on selected input pins. This feature is capable of detecting input Change-of-States, even in Sleep mode, when the clocks are disabled. Depending on the device pin count, there are up to 23 external signals (CN0 through CN22) that may be selected (enabled) for generating an interrupt request on a Change-of-State.

There are six control registers associated with the ICN module. The CNEN1 and CNEN2 registers contain the interrupt enable control bits for each of the CN input pins. Setting any of these bits enables a CN interrupt for the corresponding pins.

Each CN pin also has a weak pull-up/pull-down connected to it. The pull-ups act as a current source that is connected to the pin. The pull-downs act as a current sink to eliminate the need for external resistors when push button or keypad devices are connected.

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

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

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

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

EXAMPLE 11-1: PORT WRITE/READ EXAMPLE

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

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

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

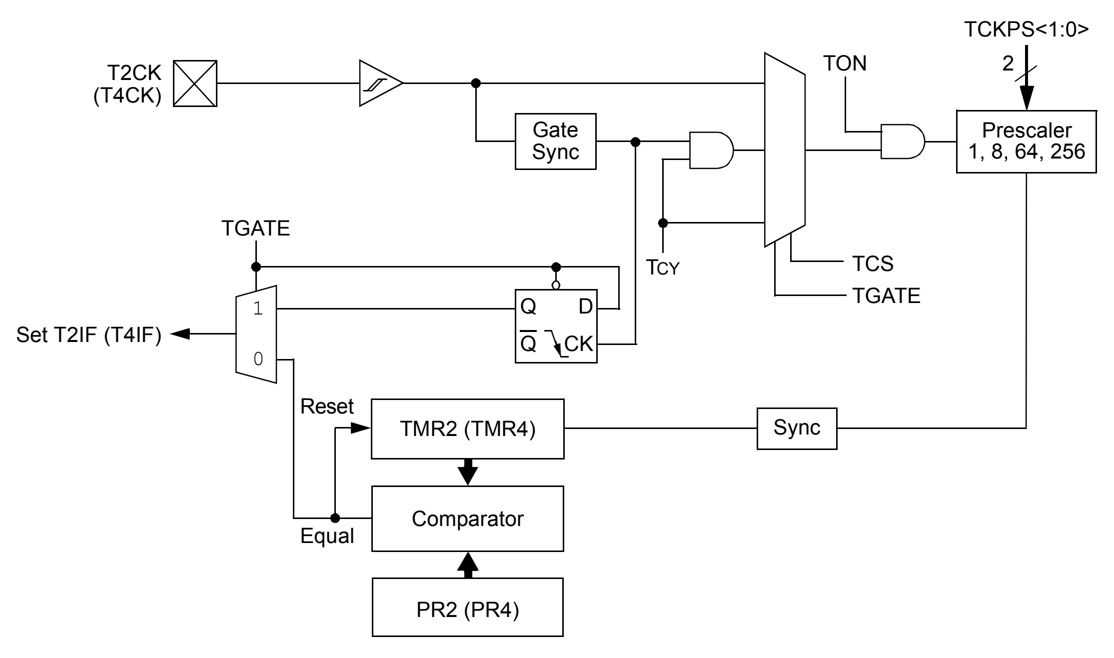
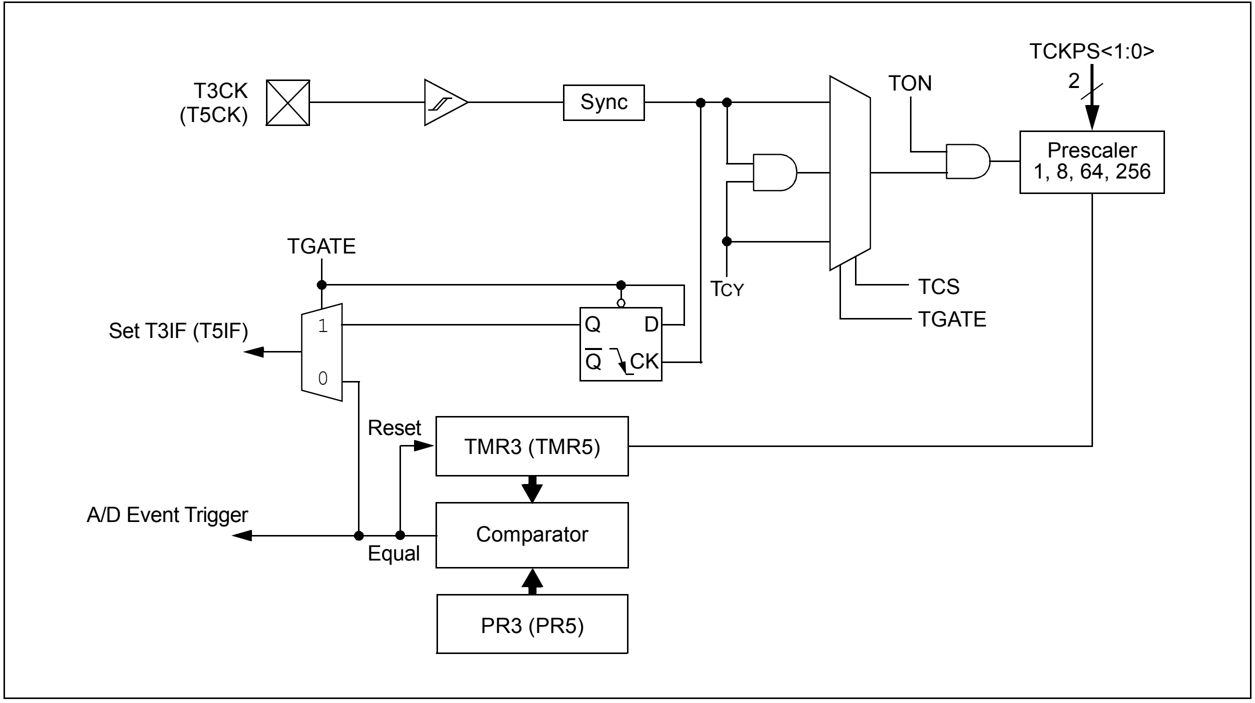


FIGURE 13-3: TIMER3 AND TIMER5 (16-BIT ASYNCHRONOUS) BLOCK DIAGRAM



18.1 UARTx Baud Rate Generator (BRG)

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Writing a new value to the UxBRG register causes the BRG timer to be reset (cleared). This ensures the BRG does not wait for a timer overflow before generating the new baud rate.

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

Desired Baud Rate = F_{CY}/(16 (UxBRG + 1))
Solving for UxBRG value:
    UxBRG = ((F_{CY}/Desired Baud Rate)/16) – 1
    UxBRG = ((4000000/9600)/16) – 1
    UxBRG = 25
Calculated Baud Rate = 4000000/(16 (25 + 1))
    = 9615
Error = (Calculated Baud Rate – Desired Baud Rate)
    Desired Baud Rate
    = (9615 – 9600)/9600
    = 0.16%

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

REGISTER 21-1: HLVDCON: HIGH/LOW-VOLTAGE DETECT CONTROL REGISTER

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
HLVDEN	—	HLSIDL	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
VDIR	BGVST	IRVST	—	HLVDL3	HLVDL2	HLVDL1	HLVDL0
bit 7							bit 0

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

bit 15 **HLVDEN:** High/Low-Voltage Detect Power Enable bit

1 = HLVD is enabled
0 = HLVD is disabled

bit 14 **Unimplemented:** Read as ‘0’

bit 13 **HLSIDL:** HLVD Stop in Idle Mode bit

1 = Discontinues module operation when device enters Idle mode
0 = Continues module operation in Idle mode

bit 12-8 **Unimplemented:** Read as ‘0’

bit 7 **VDIR:** Voltage Change Direction Select bit

1 = Event occurs when voltage equals or exceeds trip point (HLVDL<3:0>)
0 = Event occurs when voltage equals or falls below trip point (HLVDL<3:0>)

bit 6 **BGVST:** Band Gap Voltage Stable Flag bit

1 = Indicates that the band gap voltage is stable
0 = Indicates that the band gap voltage is unstable

bit 5 **IRVST:** Internal Reference Voltage Stable Flag bit

1 = Indicates that the internal reference voltage is stable and the high-voltage detect logic generates the interrupt flag at the specified voltage range
0 = Indicates that the internal reference voltage is unstable and the high-voltage detect logic will not generate the interrupt flag at the specified voltage range, and the HLVD interrupt should not be enabled

bit 4 **Unimplemented:** Read as ‘0’

bit 3-0 **HLVDL<3:0>:** High/Low-Voltage Detection Limit bits

1111 = External analog input is used (input comes from the HLVDIN pin)
1110 = Trip Point 1(1)
1101 = Trip Point 2(1)
1100 = Trip Point 3(1)
.
.
.
0000 = Trip Point 15(1)

Note 1: For the actual trip point, see **Section 29.0 “Electrical Characteristics”**.

FIGURE 30-3: TYPICAL AND MAXIMUM IIDLE vs. FREQUENCY (EC MODE, 2 MHz TO 32 MHz)

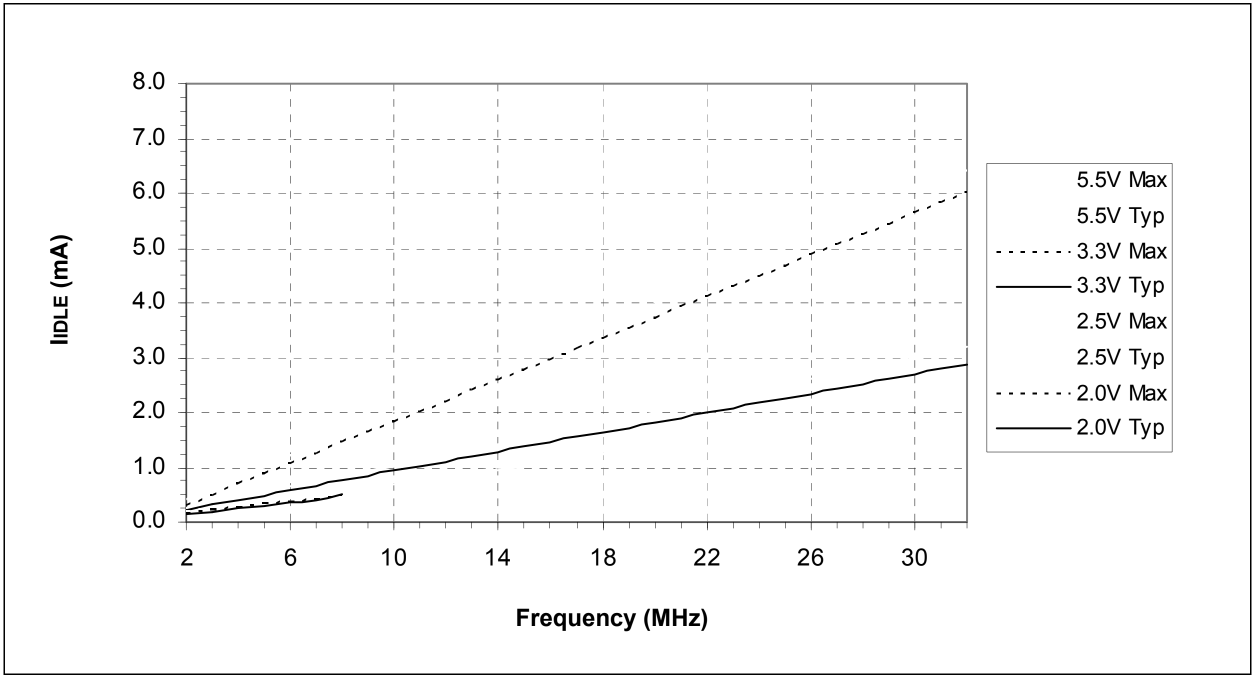


FIGURE 30-4: TYPICAL AND MAXIMUM IIDLE vs. FREQUENCY (EC MODE, 1.95 kHz TO 1 MHz)

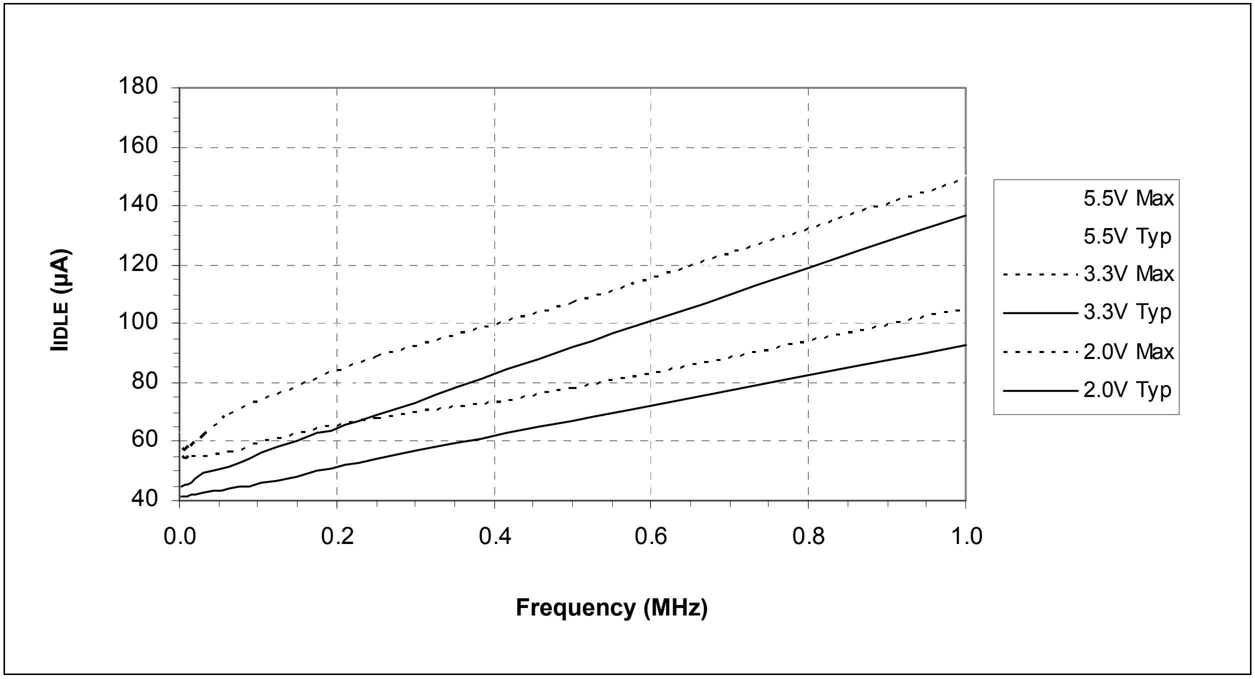


FIGURE 30-20: TYPICAL ΔIDSBOR vs. VDD

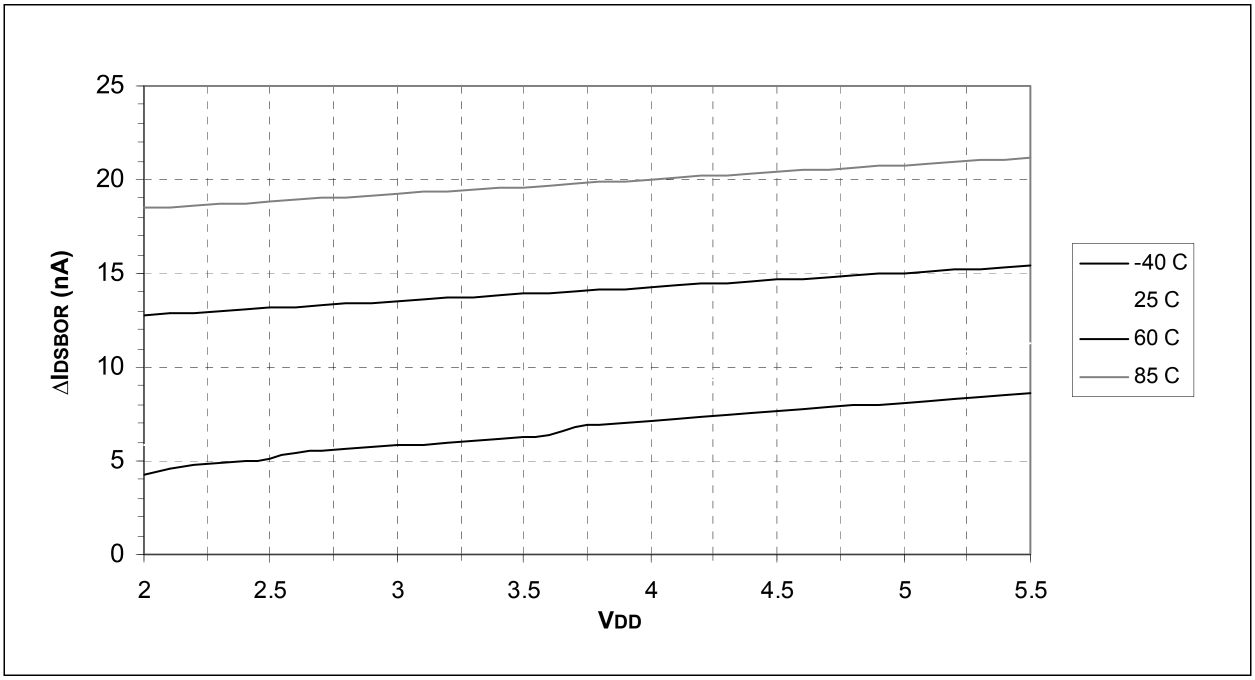


FIGURE 30-21: TYPICAL ΔIHLVD vs. VDD

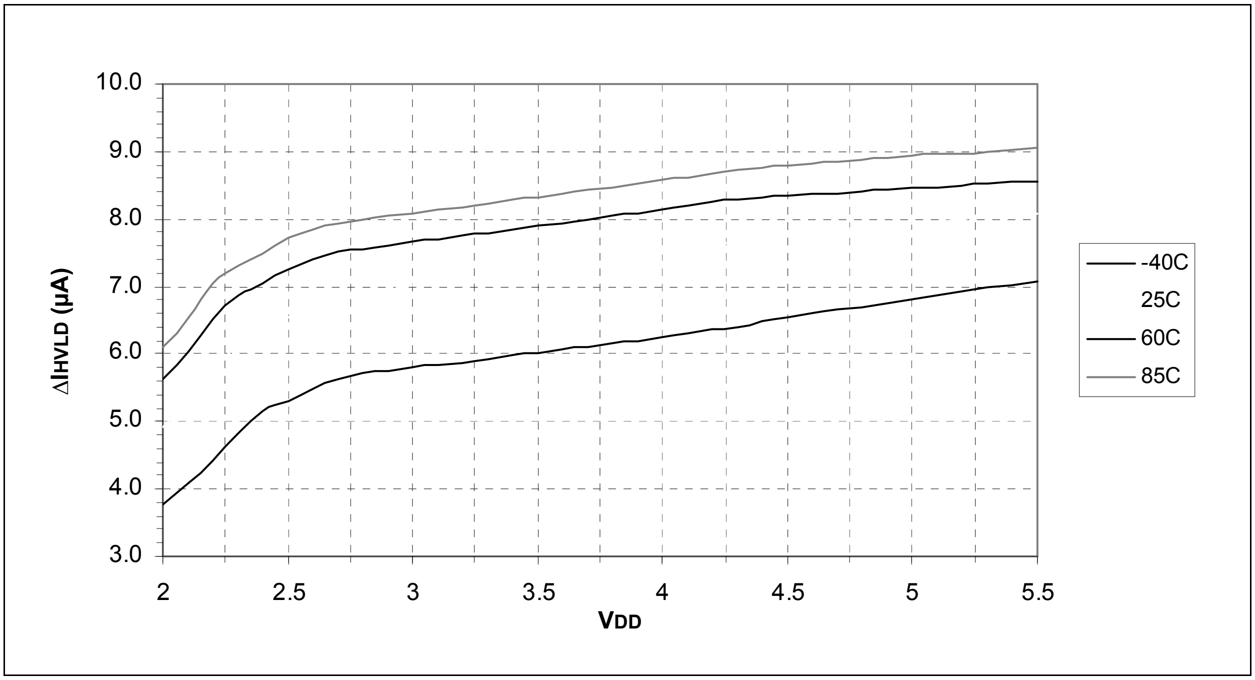


FIGURE 30-22: TYPICAL ΔIDSWDT vs. VDD

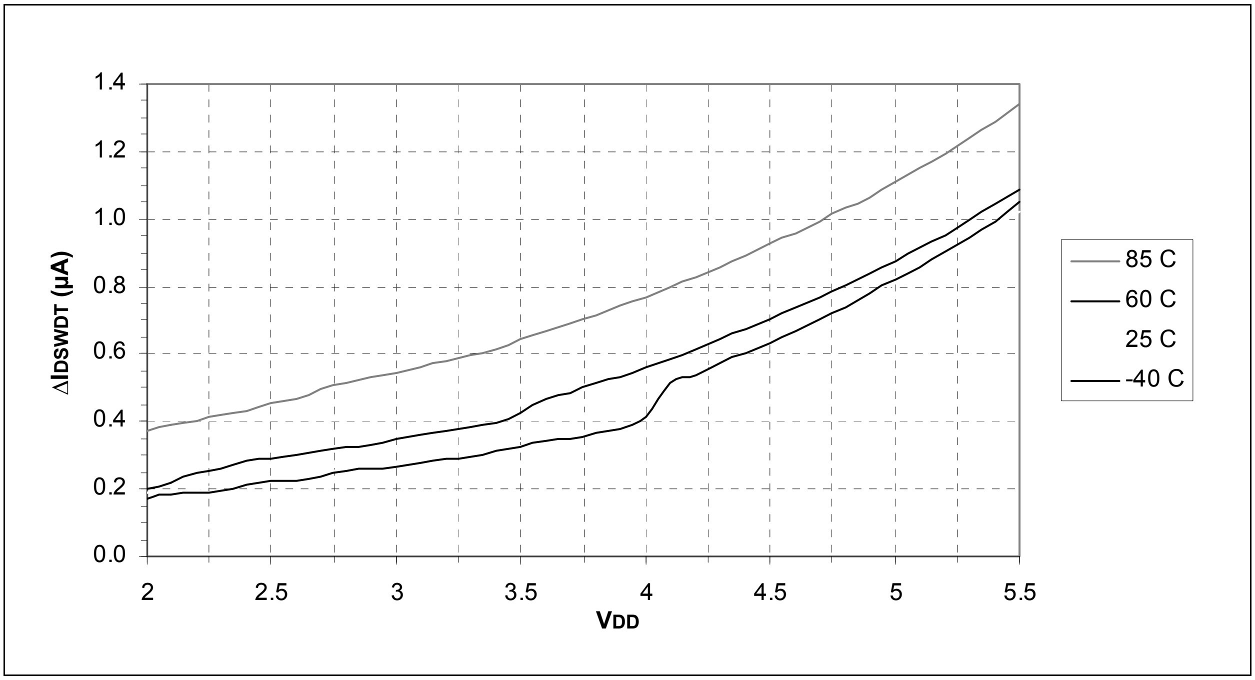


FIGURE 30-23: TYPICAL VBOR vs. TEMPERATURE (BOR TRIP POINT 3)

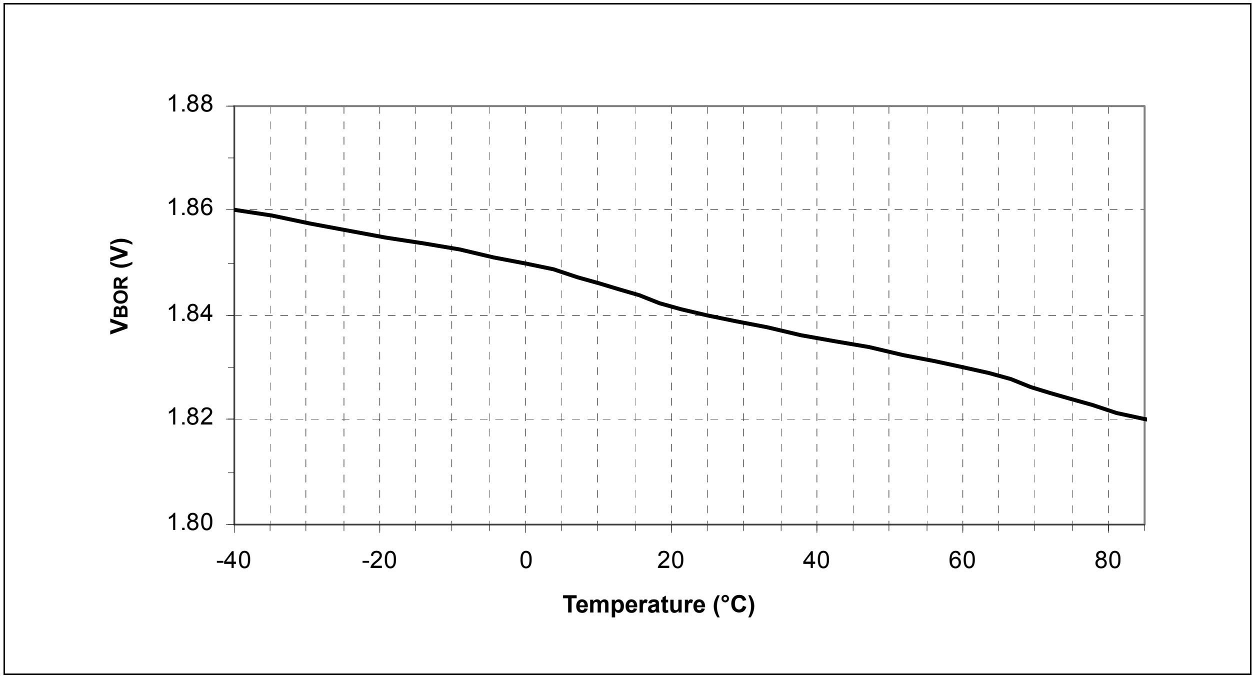


FIGURE 30-46: TYPICAL ΔI_{WDT} vs. V_{DD}

ΔI_{WDT} (µA)

V_{DD}

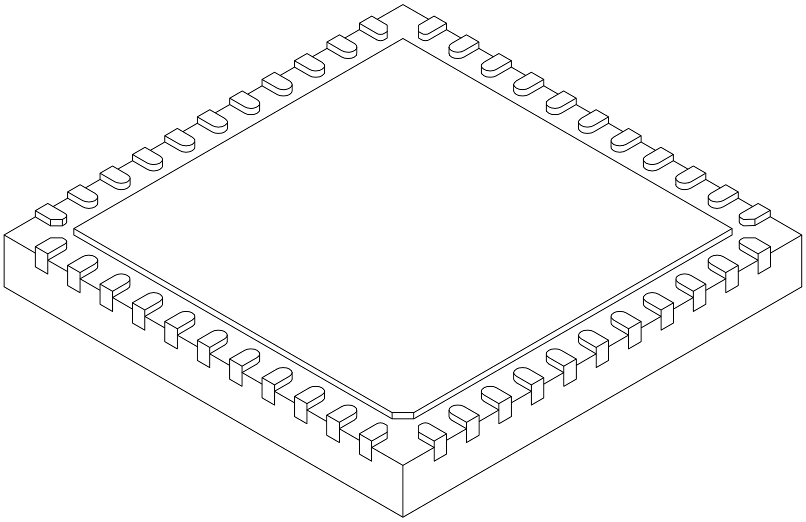
FIGURE 30-47: TYPICAL ΔI_{DSBOR} vs. V_{DD}

ΔI_{DSBOR} (nA)

V_{DD}

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



	Units	MILLIMETERS		
	Dimension Limits	MIN	NOM	MAX
Number of Pins	N	44		
Pitch	e	0.65 BSC		
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Terminal Thickness	A3	0.20 REF		
Overall Width	E	8.00 BSC		
Exposed Pad Width	E2	6.25	6.45	6.60
Overall Length	D	8.00 BSC		
Exposed Pad Length	D2	6.25	6.45	6.60
Terminal Width	b	0.20	0.30	0.35
Terminal Length	L	0.30	0.40	0.50
Terminal-to-Exposed-Pad	K	0.20	-	-

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated
3. Dimensioning and tolerancing per ASME Y14.5M

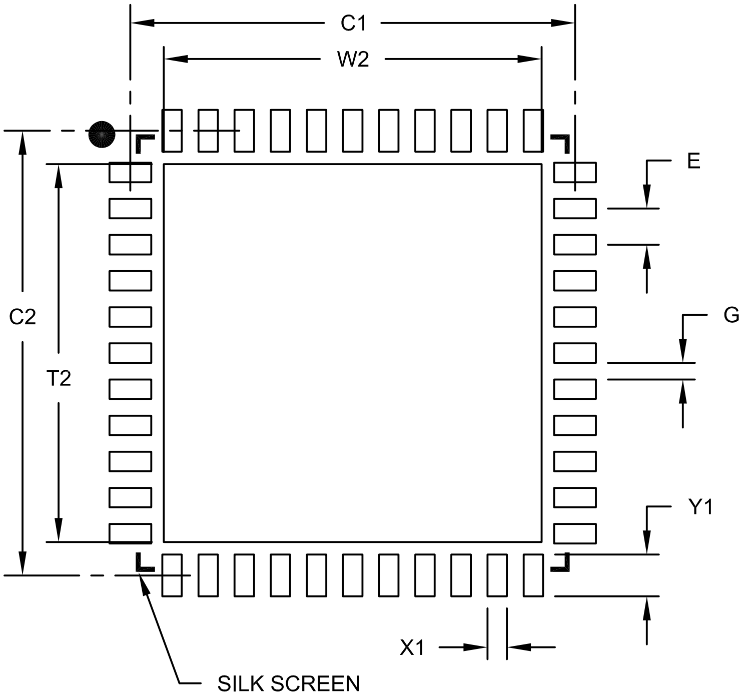
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-103C Sheet 2 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