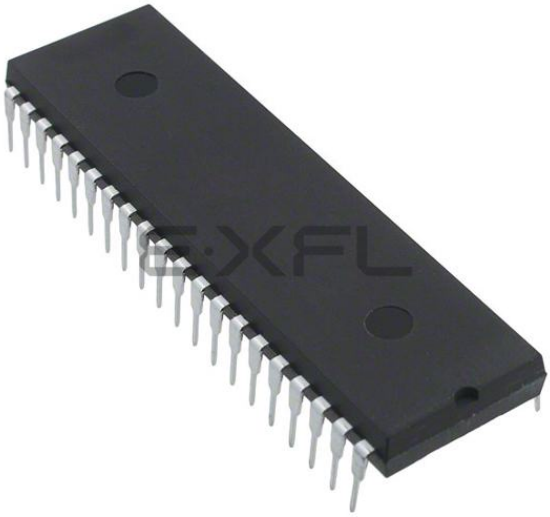


Welcome to E-XFL.COM

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	8-Bit
Speed	64MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, HLVD, POR, PWM, WDT
Number of I/O	36
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 35x12b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Through Hole
Package / Case	40-DIP (0.600", 15.24mm)
Supplier Device Package	40-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f45k42-e-p

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@microchip.com**. We welcome your feedback.

Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Website at:

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000000A is version A of document DS30000000).

Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Website; **http://www.microchip.com**
- Your local Microchip sales office (see last page)

When contacting a sales office, please specify which device, revision of silicon and data sheet (include literature number) you are using.

Customer Notification System

Register on our website at **www.microchip.com** to receive the most current information on all of our products.

4.4.2 INSTRUCTIONS IN PROGRAM MEMORY

The program memory is addressed in bytes. Instructions are stored as either two bytes or four bytes in program memory. The Least Significant Byte of an instruction word is always stored in a program memory location with an even address (LSb = 0). To maintain alignment with instruction boundaries, the PC increments in steps of two and the LSb will always read '0' (see **Section 4.2.4 "Program Counter"**).

Figure 4-2 shows an example of how instruction words are stored in the program memory.

The CALL and GOTO instructions have the absolute program memory address embedded into the instruction. Since instructions are always stored on word boundaries, the data contained in the instruction is a word address. The word address is written to PC<20:1>, which accesses the desired byte address in program memory. Instruction #2 in Figure 4-2 shows how the instruction GOTO 0006h is encoded in the program memory. Program branch instructions, which encode a relative address offset, operate in the same manner. The offset value stored in a branch instruction represents the number of single-word instructions that the PC will be offset by. **Section 41.0 "Instruction Set Summary"** provides further details of the instruction set.

4.4.3 MULTI-WORD INSTRUCTIONS

The standard PIC18 instruction set has four two-word instructions: CALL, MOVFF, GOTO and LFSR and two three-word instructions: MOVFFL and MOVSFL. In all cases, the second and the third word of the instruction always has '1111' as its four Most Significant bits; the other 12 bits are literal data, usually a data memory address.

The use of '1111' in the four MSbs of an instruction specifies a special form of NOP. If the instruction is executed in proper sequence – immediately after the first word – the data in the second word is accessed and used by the instruction sequence. If the first word is skipped for some reason and the second or third word is executed by itself, a NOP is executed instead. This is necessary for cases when the multi-word instruction is preceded by a conditional instruction that changes the PC. Example 4-4 shows how this works.

FIGURE 4-2: INSTRUCTIONS IN PROGRAM MEMORY

Program Memory Byte Locations →

Instruction 1: MOVLW 055h
Instruction 2: GOTO 0006h
Instruction 3: MOVFF 123h, 456h
Instruction 4: MOVFFL 123h, 456h

LSB = 1	LSB = 0	Word Address ↓
		000000h
		000002h
		000004h
		000006h
0Fh	55h	000008h
EFh	03h	00000Ah
F0h	00h	00000Ch
C1h	23h	00000Eh
F4h	56h	000010h
00h	60h	000012h
F4h	8Ch	000014h
F4h	56h	000016h
		000018h
		00001Ah

6.11.1 MEMORY EXECUTION VIOLATION

If the CPU executes outside the valid execution area, a memory execution violation reset occurs.

The invalid execution areas are:

1. Addresses outside implemented program memory (see Table 5-1).
2. Storage Area Flash (SAF) inside program memory, if it is enabled.

When a memory execution violation is generated, flag $\overline{\text{MEMV}}$ is cleared in PCON1 (Register 6-3) to signal the cause of Reset. It needs to be set in the user code after a memory execution violation Reset has occurred to detect further violation Resets.

6.12 Determining the Cause of a Reset

Upon any Reset, multiple bits in the STATUS and PCON0 registers are updated to indicate the cause of the Reset. Table 6-3 shows the Reset conditions of these registers.

TABLE 6-3: RESET CONDITION FOR SPECIAL REGISTERS

Condition	Program Counter	STATUS Register(1,2)	PCON0 Register	PCON1 Register
Power-on Reset	0	-110 0000	0011 110x	---- --1-
Brown-out Reset	0	-110 0000	0011 11u0	---- --1-
$\overline{\text{MCLR}}$ Reset during normal operation	0	-uuu uuuu	uuuu 0uuu	---- --u-
$\overline{\text{MCLR}}$ Reset during Sleep	0	-10u uuuu	uuuu 0uuu	---- --u-
WWDT Time-out Reset	0	-0uu uuuu	uuu0 uuuu	---- --u-
WWDT Window Violation Reset	0	-uuu uuuu	uu0u uuuu	---- --u-
RESET Instruction Executed	0	-uuu uuuu	uuuu u0uu	---- --u-
Stack Overflow Reset (STVREN = 1)	0	-uuu uuuu	1uuu uuuu	---- --u-
Stack Underflow Reset (STVREN = 1)	0	-uuu uuuu	u1uu uuuu	---- --u-
Memory Violation Reset	0	-uuu uuuu	uuuu uuuu	---- --0-

Legend: u = unchanged, x = unknown, – = unimplemented bit, reads as '0'.

Note 1: If a Status bit is not implemented, that bit will be read as '0'.

2: Status bits Z, C, DC are reset by POR/BOR, but not defined by the Resets module (Register 4-2).

REGISTER 9-25: IPR0: PERIPHERAL INTERRUPT PRIORITY REGISTER 0

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
IOCIP	CRCIP	SCANIP	NVMIP	CSWIP	OSFIP	HLVDIP	SWIP
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7 **IOCIP:** Interrupt-on-Change Priority bit
1 = High priority
0 = Low priority

bit 6 **CRCIP:** CRC Interrupt Priority bit
1 = High priority
0 = Low priority

bit 5 **SCANIP:** Memory Scanner Interrupt Priority bit
1 = High priority
0 = Low priority

bit 4 **NVMIP:** NVM Interrupt Priority bit
1 = High priority
0 = Low priority

bit 3 **CSWIP:** Clock Switch Interrupt Priority bit
1 = High priority
0 = Low priority

bit 2 **OSFIP:** Oscillator Fail Interrupt Priority bit
1 = High priority
0 = Low priority

bit 1 **HLVDIP:** HLVD Interrupt Priority bit
1 = High priority
0 = Low priority

bit 0 **SWIP:** Software Interrupt Priority bit
1 = High priority
0 = Low priority

15.10 Reset

The DMA registers are set to the default state on any Reset. The registers are also reset to the default state when the enable bit is cleared (DMA1CON1bits.EN=0).

15.11 Power Saving Mode Operation

The DMA utilizes system clocks and it is treated as a peripheral when it comes to power saving operations. Like other peripherals, the DMA also uses Peripheral Module Disable bits to further tailor its operation in low-power states.

15.11.1 SLEEP MODE

When the device enters Sleep mode, the system clock to the module is shut down, therefore no DMA operation is supported in Sleep. Once the system clock is disabled, the requisite read and write clocks are also disabled, without which the DMA cannot perform any of its tasks.

Any transfers that may be in progress are resumed on exiting from Sleep mode. Register contents are not affected by the device entering or leaving Sleep mode. It is recommended that DMA transactions be allowed to finish before entering Sleep mode.

15.11.2 IDLE MODE

In IDLE mode, all of the system clocks (including the read and write clocks) are still operating but the CPU is not using them to save power.

Therefore, every instruction cycle is available to the system arbiter and if the bubble is granted to the DMA, it may be utilized to move data.

15.11.3 DOZE MODE

Similar to the Idle mode, the CPU does not utilize all of the available instruction cycles slots that are available to it in order to save power. It only executes instructions based on its settings from the Doze settings.

Therefore, every instruction not used by the CPU is available for system arbitration and may be utilized by the DMA if granted by the arbiter.

15.11.4 PERIPHERAL MODULE DISABLE

The Peripheral Module Disable (PMD) registers provide a method to disable DMA by gating all clock sources supplied to it. The respective DMAxMD bit needs to be set in order to disable the DMA.

15.12 DMA Register Interfaces

The DMA can transfer data to any GPR or SFR location. For better user accessibility, some of the more commonly used SFR spaces have their Mirror registers placed in Bank 64 (0x4000-0x40FF). These Mirror registers can be only accessed through the DMA Source and Destination Address registers. Refer to Table 4-3 for details about Bank 64 Registers.

REGISTER 19-6: PMD5: PMD CONTROL REGISTER 5

U-0	U-0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	U2MD	U1MD	—	SPI1MD	I2C2MD	I2C1MD
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	q = Value depends on condition

bit 7-6 **Unimplemented:** Read as '0'

bit 5 **U2MD:** Disable UART2 bit
1 = UART2 module disabled
0 = UART2 module enabled

bit 4 **U1MD:** Disable UART1 bit
1 = UART1 module disabled
0 = UART1 module enabled

bit 3 **Unimplemented:** Read as '0'

bit 2 **SPI1MD:** Disable SPI1 Module bit
1 = SPI1 module disabled
0 = SPI1 module enabled

bit 1 **I2C2MD:** Disable I^2C2 Module bit
1 = I^2C2 module disabled
0 = I^2C2 module enabled

bit 0 **I2C1MD:** Disable I^2C1 Module bit
1 = I^2C1 module disabled
0 = I^2C1 module enabled

22.5.10 LEVEL-TRIGGERED HARDWARE LIMIT ONE-SHOT MODES

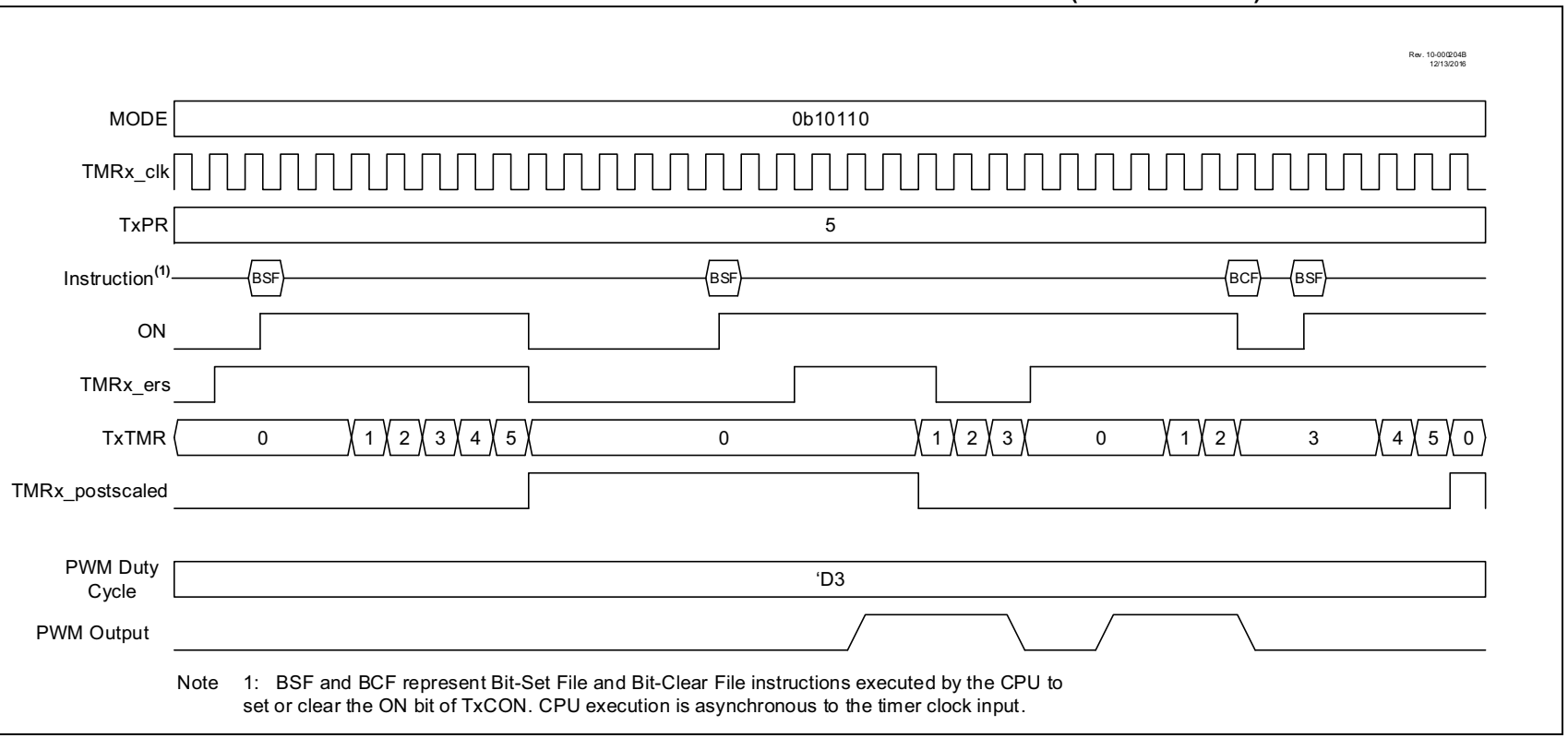
The Level Triggered Hardware Limit One-Shot modes hold the timer in Reset on an external Reset level and start counting when both the ON bit is set and the external signal is not at the Reset level. If one of either the external signal is not in reset or the ON bit is set then the other signal being set/made active will start the timer. Reset levels are selected as follows:

- Low reset level (MODE<4:0> = 10110)
- High reset level (MODE<4:0> = 10111)

When the timer count matches the T2PR period count, the timer is reset and the ON bit is cleared. When the ON bit is cleared by either a T2PR match or by software control, the timer will stay in Reset until both the ON bit is set and the external signal is not at the Reset level.

When Level Triggered Hardware Limit One-Shot modes are used in conjunction with the CCP PWM operation, the PWM drive goes active with either the external signal edge or the setting of the ON bit, whichever of the two starts the timer.

FIGURE 22-13: LEVEL-TRIGGERED HARDWARE LIMIT ONE-SHOT MODE TIMING DIAGRAM (MODE = 10110)



Note 1: BSF and BCF represent Bit-Set File and Bit-Clear File instructions executed by the CPU to set or clear the ON bit of TxCON. CPU execution is asynchronous to the timer clock input.

22.7 Register Definitions: Timer2/4/6 Control

Long bit name prefixes for the Timer2/4/6 peripherals are shown in Table 22-2. Refer to **Section 1.3.2.2 "Long Bit Names"** for more information.

TABLE 22-2: OPERATING MODES

Peripheral	Bit Name Prefix
Timer2	T2
Timer4	T4
Timer6	T6

REGISTER 22-1: TxCLK: TIMERx CLOCK SELECTION REGISTER

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	CS<3:0>			
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-4 **Unimplemented:** Read as '0'

bit 3-0 **CS<3:0>:** Timerx Clock Selection bits

CS<3:0>	T2TMR	TMR4	TMR6
	Clock Source	Clock Source	Clock Source
1111	Reserved	Reserved	Reserved
1110	CLC4_out	CLC4_out	CLC4_out
1101	CLC3_out	CLC3_out	CLC3_out
1100	CLC2_out	CLC2_out	CLC2_out
1011	CLC1_out	CLC1_out	CLC1_out
1010	ZCD_OUT	ZCD_OUT	ZCD_OUT
1001	NCO1OUT	NCO1OUT	NCO1OUT
1000	CLKREF_OUT	CLKREF_OUT	CLKREF_OUT
0111	SOSC	SOSC	SOSC
0110	MFINTOSC (32 kHz)	MFINTOSC (32 kHz)	MFINTOSC (32 kHz)
0101	MFINTOSC (500 kHz)	MFINTOSC (500 kHz)	MFINTOSC (500 kHz)
0100	LFINTOSC	LFINTOSC	LFINTOSC
0011	HFINTOSC	HFINTOSC	HFINTOSC
0010	Fosc	Fosc	Fosc
0001	Fosc/4	Fosc/4	Fosc/4
0000	Pin selected by T2INPPS	Pin selected by T4INPPS	Pin selected by T6INPPS

23.2.3 SOFTWARE INTERRUPT MODE

When the Capture mode is changed, a false capture interrupt may be generated. The user should keep the CCPxIE Interrupt Priority bit of the respective PIE register clear to avoid false interrupts. Additionally, the user should clear the CCPxIF interrupt flag bit of the respective PIR register following any change in Operating mode.

23.2.4 CAPTURE DURING SLEEP

Capture mode depends upon the Timer1 module for proper operation. There are two options for driving the Timer1 module in Capture mode. It can be driven by the instruction clock (FOSC/4), or by an external clock source.

When Timer1 is clocked by FOSC/4, Timer1 will not increment during Sleep. When the device wakes from Sleep, Timer1 will continue from its previous state.

Capture mode will operate during Sleep as long as the clock source for Timer1 is active in Sleep.

23.3 Compare Mode

Compare mode makes use of the 16-bit Timer1 resource. The 16-bit value of the CCPRxH:CCPRxL register pair is constantly compared against the 16-bit value of the TMRxH:TMRxL register pair. When a match occurs, one of the following events can occur:

- Toggle the CCPx output, clear TMRx
- Toggle the CCPx output
- Set the CCPx output
- Clear the CCPx output
- Pulse output
- Pulse output, clear TMRx

The action on the pin is based on the value of the MODE<3:0> control bits of the CCPxCON register. At the same time, the interrupt flag CCPxIF bit is set, and an ADC conversion can be triggered, if selected.

All Compare modes can generate an interrupt and trigger an ADC conversion. When MODE = `0b0001` or `0b1011`, the CCP resets the TMR register pair.

Figure 23-2 shows a simplified diagram of the compare operation.

FIGURE 23-2: COMPARE MODE OPERATION BLOCK DIAGRAM

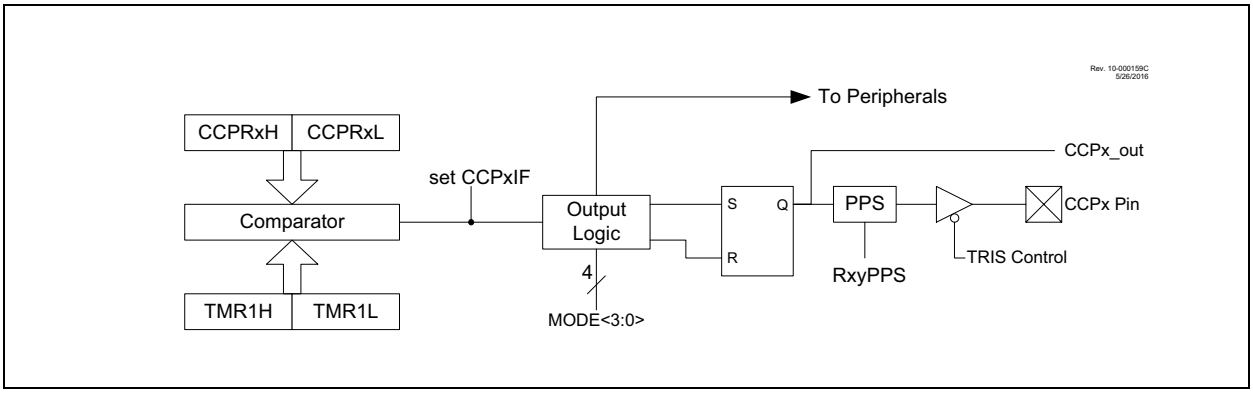
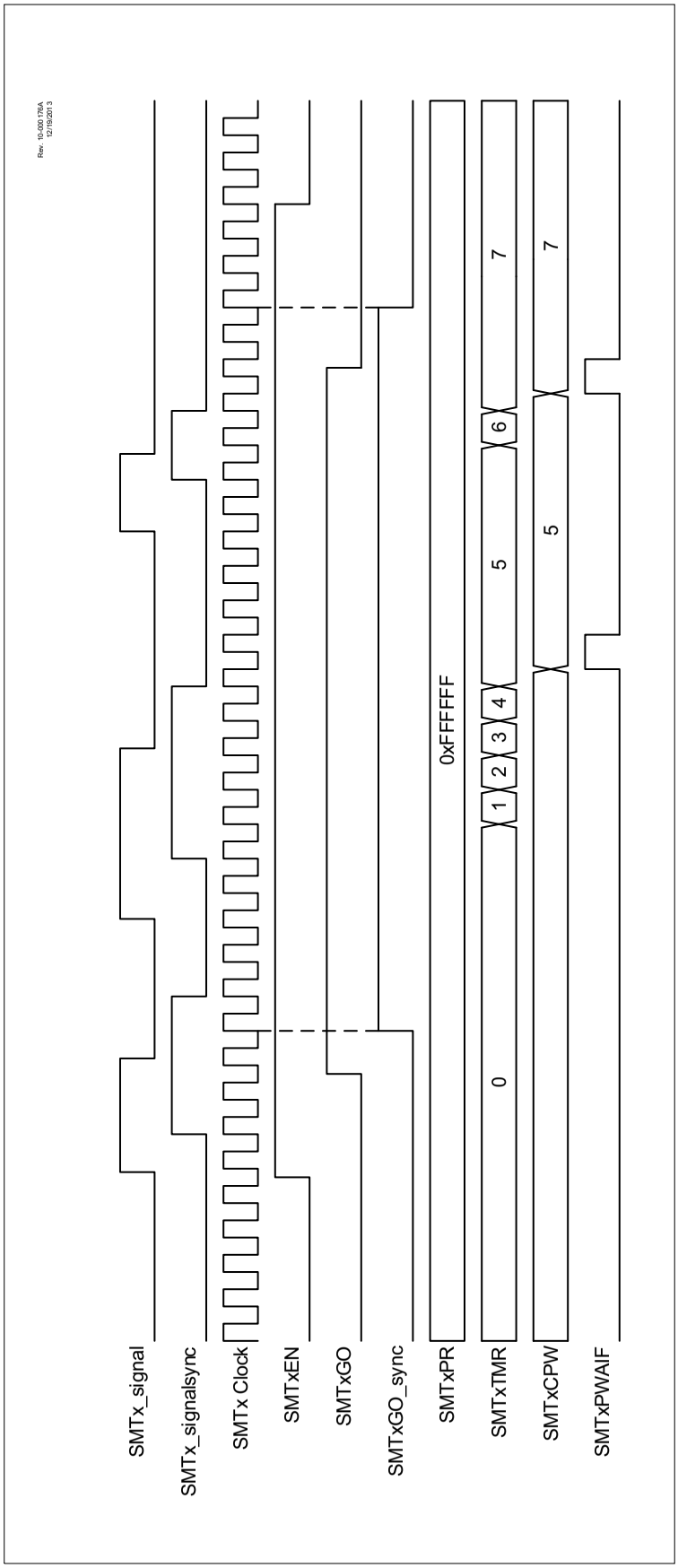


FIGURE 25-4: GATED TIMER MODE REPEAT ACQUISITION TIMING DIAGRAM



25.6.6 GATED WINDOWED MEASURE MODE

This mode measures the duty cycle of the SMT1_signal input over a known input window. It does so by incrementing the timer on each pulse of the clock signal while the SMT1_signal input is high, updating the SMT1CPR register and resetting the timer on every rising edge of the SMTWINx input after the first. See Figure 25-12 and Figure 25-13.

TABLE 26-2: SUMMARY OF REGISTERS ASSOCIATED WITH CWG

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CWGxCON0	EN	LD	—	—	—	MODE<2:0>			424
CWGxCON1	—	—	IN	—	POLD	POLC	POLB	POLA	425
CWGxCLK	—	—	—	—	—	—	—	CS	426
CWGxISM	—	—	—	ISM<4:0>					427
CWGxSTR	OVRD	OVRC	OVRB	OVRA	STRD	STRC	STRB	STRA	428
CWGxAS0	SHUTDOWN	REN	LSBD<1:0>		LSAC<1:0>		—	—	429
CWGxAS1	—	AS6E	AS5E	AS4E	AS3E	AS2E	AS1E	AS0E	430
CWGxDBR	—	—	DBR<5:0>						431
CWGxDBF	—	—	DBF<5:0>						431

Legend: – = unimplemented locations read as '0'. Shaded cells are not used by CWG.

REGISTER 27-3: CLCxSEL0: GENERIC CLCx DATA 0 SELECT REGISTER

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	D1S<5:0>					
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **D1S<5:0>:** CLCx Data1 Input Selection bits
See Table 27-1.

REGISTER 27-4: CLCxSEL1: GENERIC CLCx DATA 1 SELECT REGISTER

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	D2S<5:0>					
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **D2S<5:0>:** CLCx Data 2 Input Selection bits
See Table 27-1.

REGISTER 27-5: CLCxSEL2: GENERIC CLCx DATA 2 SELECT REGISTER

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	D3S<5:0>					
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **D3S<5:0>:** CLCx Data 3 Input Selection bits
See Table 27-1.

REGISTER 27-6: CLCxSEL3: GENERIC CLCx DATA 3 SELECT REGISTER

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	D4S<5:0>					
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **D4S<5:0>:** CLCx Data 4 Input Selection bits
See Table 27-1.

REGISTER 30-3: MD1CARH: MODULATION HIGH CARRIER CONTROL REGISTER

U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	CH<4:0>(1)				
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-5 **Unimplemented:** Read as '0'

bit 4-0 **CH<4:0>:** Modulator Carrier High Selection bits(1)
See Table 30-1 for signal list

Note 1: Unused selections provide an input value.

REGISTER 30-4: MD1CARL: MODULATION LOW CARRIER CONTROL REGISTER

U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	CL<4:0>(1)				
bit 7							bit 0

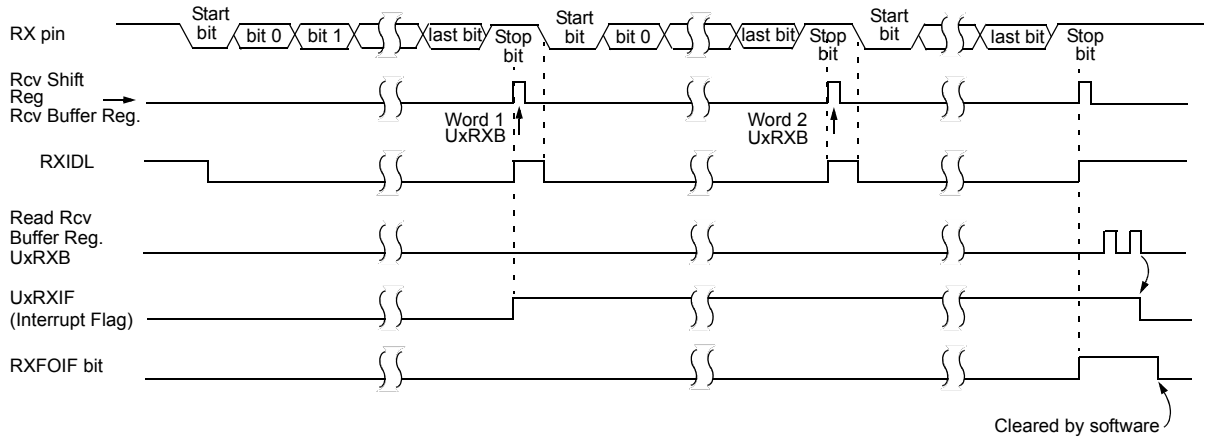
Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-5 **Unimplemented:** Read as '0'

bit 4-0 **CL<4:0>:** Modulator Carrier Low Input Selection bits(1)
See Table 30-1 for signal list

Note 1: Unused selections provide a zero as the input value.

FIGURE 31-5: ASYNCHRONOUS RECEPTION



Note: This timing diagram shows three words appearing on the RX input. The UxRXB (receive buffer) is not read before the third word is received, causing the RXFOIF (FIFO overrun) bit to be set. STPMD = 0, STP<1:0> = 00.

REGISTER 32-8: SPIxCON1: SPI CONFIGURATION REGISTER 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-1/1	R/W-0/0	R/W-0/0
SMP	CKE	CKP	FST	—	SSP	SDIP	SDOP
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'

bit 7 **SMP**: SPI Input Sample Phase Control bit

Slave mode:

1 = Reserved

0 = SDI input is sampled in the middle of data output time

Master mode:

1 = SDI input is sampled at the end of data output time

0 = SDI input is sampled in the middle of data output time

bit 6 **CKE**: Clock Edge Select bit

1 = Output data changes on transition from active to idle clock state

0 = Output data changes on transition from idle to active clock state

bit 5 **CKP**: Clock Polarity Select bit

1 = Idle state for SCK is high level

0 = Idle state for SCK is low level

bit 4 **FST**: Fast Start Enable bit

Slave mode:

This bit is ignored

Master mode:

1 = Delay to first SCK may be less than ½ baud period

0 = Delay to first SCK will be at least ½ baud period

bit 3 **Unimplemented**: Read as '0'

bit 2 **SSP**: SS Input/Output Polarity Control bit

1 = SS is active-low

0 = SS is active-high

bit 1 **SDIP**: SDI Input Polarity Control bit

1 = SDI input is active-low

0 = SDI input is active-high

bit 0 **SDOP**: SDI Output Polarity Control bit

1 = SDO output is active-low

0 = SDO output is active-high

BZ **Branch if Zero**

Syntax: BZ n

Operands: -128 ≤ n ≤ 127

Operation: if ZERO bit is '1'
(PC) + 2 + 2n → PC

Status Affected: None

Encoding:

1110	0000	nnnn	nnnn

Description: If the ZERO bit is '1', then the program will branch.
The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 2 + 2n. This instruction is then a 2-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:
If Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	Write to PC
No operation	No operation	No operation	No operation

If No Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	No operation

Example: `HERE BZ Jump`

Before Instruction
PC = address `(HERE)`
After Instruction
If ZERO = 1;
PC = address `(Jump)`
If ZERO = 0;
PC = address `(HERE + 2)`

CALL **Subroutine Call**

Syntax: CALL k {,s}

Operands: 0 ≤ k ≤ 1048575
s ∈ [0,1]

Operation: (PC) + 4 → TOS,
k → PC<20:1>,
if s = 1
(W) → WS,
(Status) → STATUSS,
(BSR) → BSRS

Status Affected: None

Encoding:

1st word (k<7:0>)	1110	110s	$k_7$kkk	kkkk$_0$
2nd word(k<19:8>)	1111	k_{19}kkk	kkkk	kkkk$_8$

Description: Subroutine call of entire 2-Mbyte memory range. First, return address (PC + 4) is pushed onto the return stack. If 's' = 1, the W, Status and BSR registers are also pushed into their respective shadow registers, WS, STATUSS and BSRS. If 's' = 0, no update occurs (default). Then, the 20-bit value 'k' is loaded into PC<20:1>. `CALL` is a 2-cycle instruction.

Words: 2

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'<7:0>,	PUSH PC to stack	Read literal 'k'<19:8>, Write to PC
No operation	No operation	No operation	No operation

Example: `HERE CALL THERE, 1`

Before Instruction
PC = address `(HERE)`
After Instruction
PC = address `(THERE)`
TOS = address `(HERE + 4)`
WS = W
BSRS = BSR
STATUSS = Status

TABLE 44-6: I/O PORTS

Standard Operating Conditions (unless otherwise stated)							
Param No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
	VIL	**Input Low Voltage**					
		I/O PORT:					
D300		with TTL buffer	—	—	0.8	V	4.5V ≤ VDD ≤ 5.5V
D301			—	—	0.15 VDD	V	1.8V ≤ VDD < 4.5V
D302		with Schmitt Trigger buffer	—	—	0.2 VDD	V	2.0V ≤ VDD ≤ 5.5V
D303		with I2C levels	—	—	0.3 VDD	V	
D304		with SMBus 2.0	—	—	0.8	V	2.7V ≤ VDD ≤ 5.5V
D305		with SMBus 3.0	—	—	0.8	V	1.8V ≤ VDD ≤ 5.5V
D306		$\overline{MCLR}$	—	—	0.2 VDD	V	
	VIH	**Input High Voltage**					
		I/O PORT:					
D320		with TTL buffer	2.0	—	—	V	4.5V ≤ VDD ≤ 5.5V
D321			0.25 VDD + 0.8	—	—	V	1.8V ≤ VDD < 4.5V
D322		with Schmitt Trigger buffer	0.8 VDD	—	—	V	2.0V ≤ VDD ≤ 5.5V
D323		with I2C levels	0.7 VDD	—	—	V	
D324		with SMBus 2.0	2.1	—	—	V	2.7V ≤ VDD ≤ 5.5V
D325		with SMBus 3.0	1.35	—	—	V	1.8V ≤ VDD ≤ 5.5V
D326		$\overline{MCLR}$	0.7 VDD	—	—	V	
	IIL	**Input Leakage Current(1)**					
D340		I/O Ports	—	± 5	± 125	nA	VSS ≤ VPIN ≤ VDD, Pin at high-impedance, 85°C
D341			—	± 5	± 1000	nA	VSS ≤ VPIN ≤ VDD, Pin at high-impedance, 125°C
D342		$\overline{MCLR}$(2)	—	± 50	± 200	nA	VSS ≤ VPIN ≤ VDD, Pin at high-impedance, 85°C
	IPUR	**Weak Pull-up Current**					
D350			25	120	200	µA	VDD = 3.0V, VPIN = VSS
	VOL	**Output Low Voltage**					
D360		I/O ports	—	—	0.6	V	IOL = 10.0mA, VDD = 3.0V
	VOH	**Output High Voltage**					
D370		I/O ports	VDD - 0.7	—	—	V	IOH = 6.0 mA, VDD = 3.0V
D380	CIO	**All I/O pins**	—	5	50	pF	

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: Negative current is defined as current sourced by the pin.

2: The leakage current on the $\overline{MCLR}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

TABLE 44-10: INTERNAL OSCILLATOR PARAMETERS(1)

Standard Operating Conditions (unless otherwise stated)							
Param No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
OS50	FHFOSC	Precision Calibrated HFINTOSC Frequency	—	4 8 12 16 48 64	—	MHz	(**Note 2**)
OS51	FHFOSCLP	Low-Power Optimized HFINTOSC Frequency	0.93 1.86	1 2	1.07 2.14	MHz MHz	
OS53*	FLFOSC	Internal LFINTOSC Frequency	—	31	—	kHz	
OS54*	THFOSCST	HFINTOSC Wake-up from Sleep Start-up Time	— —	11 50	20 —	µs µs	VREGPM = 0 VREGPM = 1
OS56	TLFOSCST	LFINTOSC Wake-up from Sleep Start-up Time	—	0.2	—	ms	

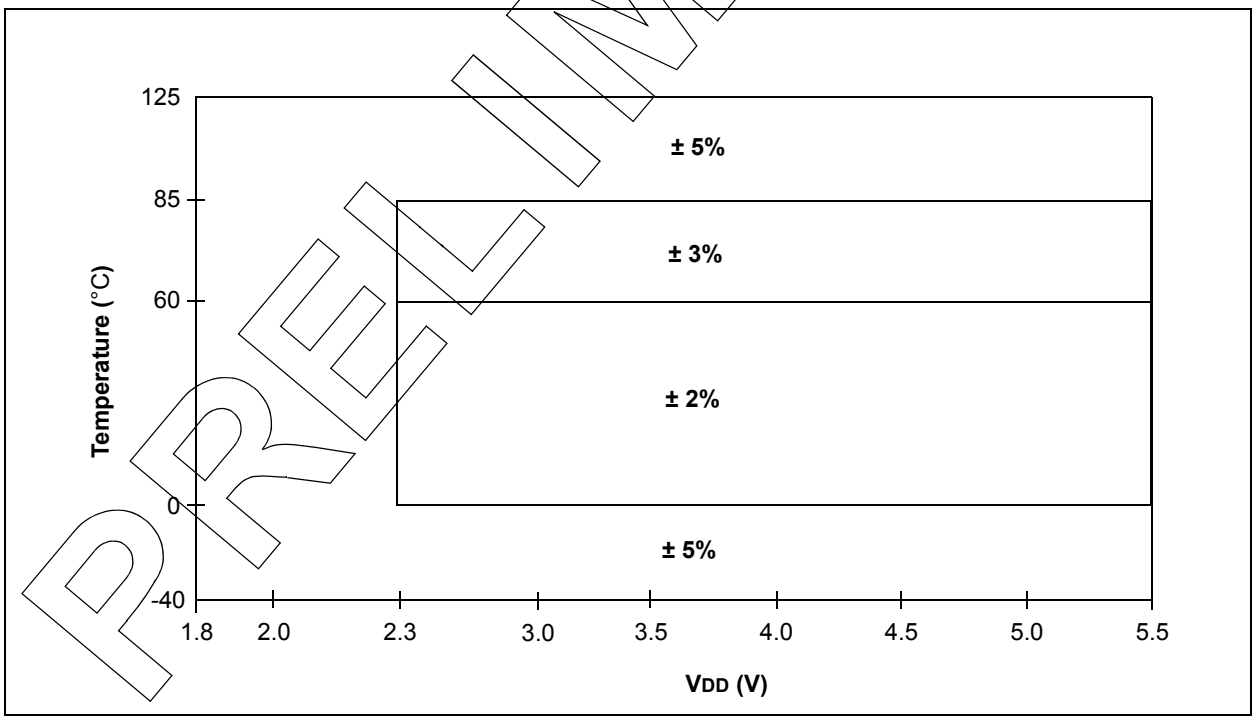
* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: To ensure these oscillator frequency tolerances, VDD and VSS must be capacitively decoupled as close to the device as possible. 0.1 µF and 0.01 µF values in parallel are recommended.

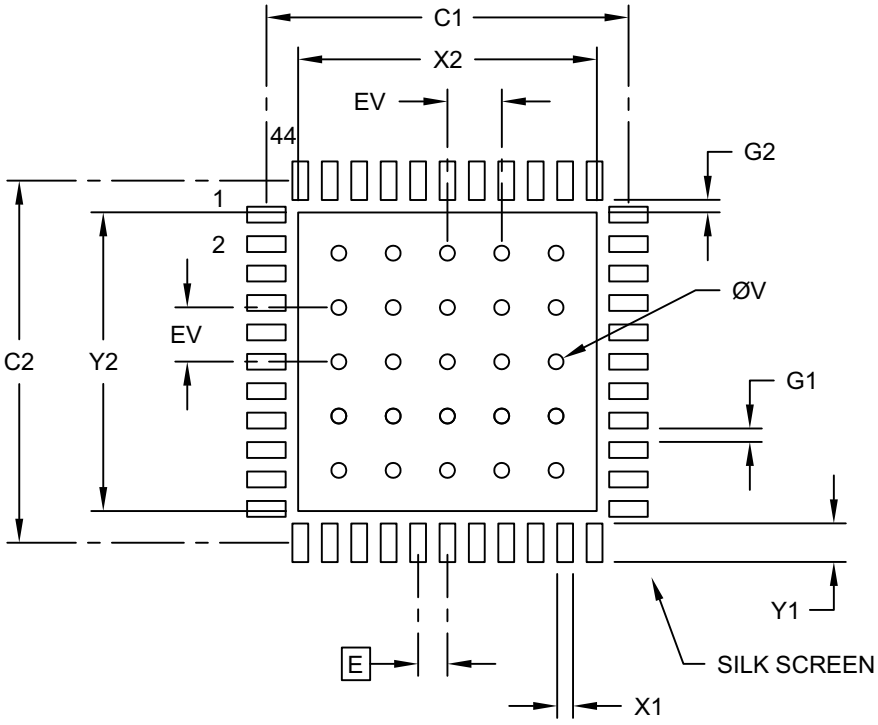
2: See Figure 44-6: Precision Calibrated HFINTOSC Frequency Accuracy Over Device VDD and Temperature.

FIGURE 44-6: PRECISION CALIBRATED HFINTOSC FREQUENCY ACCURACY OVER DEVICE VDD AND TEMPERATURE



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

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



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.65 BSC	
Optional Center Pad Width	X2			6.60
Optional Center Pad Length	Y2			6.60
Contact Pad Spacing	C1		8.00	
Contact Pad Spacing	C2		8.00	
Contact Pad Width (X44)	X1			0.35
Contact Pad Length (X44)	Y1			0.85
Contact Pad to Contact Pad (X40)	G1	0.30		
Contact Pad to Center Pad (X44)	G2	0.28		
Thermal Via Diameter	V		0.33	
Thermal Via Pitch	EV		1.20	

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

2. For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during reflow process

Microchip Technology Drawing No. C04-2103C