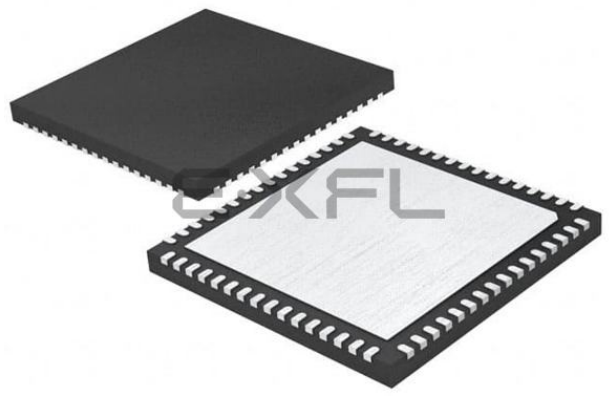
Microchip Technology - PIC18F65K40-I/MR Datasheet



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, POR, PWM, WDT
Number of I/O	60
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 45x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-VFQFN Exposed Pad
Supplier Device Package	64-VQFN (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f65k40-i-mr

TABLE 1: 64-PIN ALLOCATION TABLE (PIC18(L)F6XK40) (CONTINUED)

I/O(2)	64-Pin TQFP, QFN	A/D	DAC	Comparator	Timers	CCP and PWM	CWG	ZCD	SMT	Clock Reference (CLKR)	Interrupt	EUSART	DSM	MSSP	Basic
RE5	61	ANE5	—	—	—	CCP1(1)	—	—	—	—	IOCE5	—	—	—	—
RE6	60	ANE6	—	—	—	CCP3(1)	—	—	SMT1WIN1(1)	—	IOCE6	—	—	—	—
RE7	59	ANE7	—	—	—	—	—	—	SMT1SIG1(1)	—	IOCE7	—	—	—	—
RF0	18	—	—	C1IN0- C2IN0-	—	—	—	—	—	—	—	—	—	—	—
RF1	17	ANF1	—	—	—	—	—	—	—	—	—	—	—	—	—
RF2	16	ANF2	—	—	—	—	—	—	—	—	—	—	—	—	—
RF3	15	—	—	C1IN2- C2IN2- C3IN2-	—	—	—	—	—	—	—	—	—	—	—
RF4	14	ANF4	—	C2IN0+	—	—	—	—	—	—	—	—	—	—	—
RF5	13	ANF5	DAC1OUT1	C1IN1- C2IN1-	—	—	—	—	—	—	—	—	—	—	—
RF6	12	ANF6	—	C1IN0+	—	—	—	—	—	—	—	—	—	—	—
RF7	11	—	—	C1IN3- C2IN3- C3IN3-	—	—	—	—	—	—	—	—	—	$\overline{\text{SS1}}$(1)	—
RG0	3	ANG0	—	—	—	—	—	—	—	—	—	—	—	—	—
RG1	4	ANG1	—	—	—	—	—	—	—	—	—	CK2(1)	—	—	—
RG2	5	ANG2	—	C3IN0+	—	—	—	—	—	—	—	RX2(1) DT2(3)	—	—	—
RG3	6	ANG3	—	C3IN0-	—	CCP4(1)	—	—	—	—	—	—	—	—	—
RG4	8	ANG4	—	C3IN1-	T5G(1) T7CKI(1)	CCP5(1)	—	—	—	—	—	—	—	—	—
RG5	7	—	—	—	—	—	—	—	—	—	—	IOCG5	—	—	$\overline{\text{MCLR}}$, VPP
RG6	20	ANG6	—	—	—	—	—	—	SMT2WIN1(1)	—	—	—	—	—	—
RG7	19	ANG7	—	—	—	—	—	—	SMT2SIG1(1)	—	—	—	—	—	—
RH0	26	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RH1	25	ADCACT(1)	—	—	—	—	—	—	—	—	—	—	—	—	—

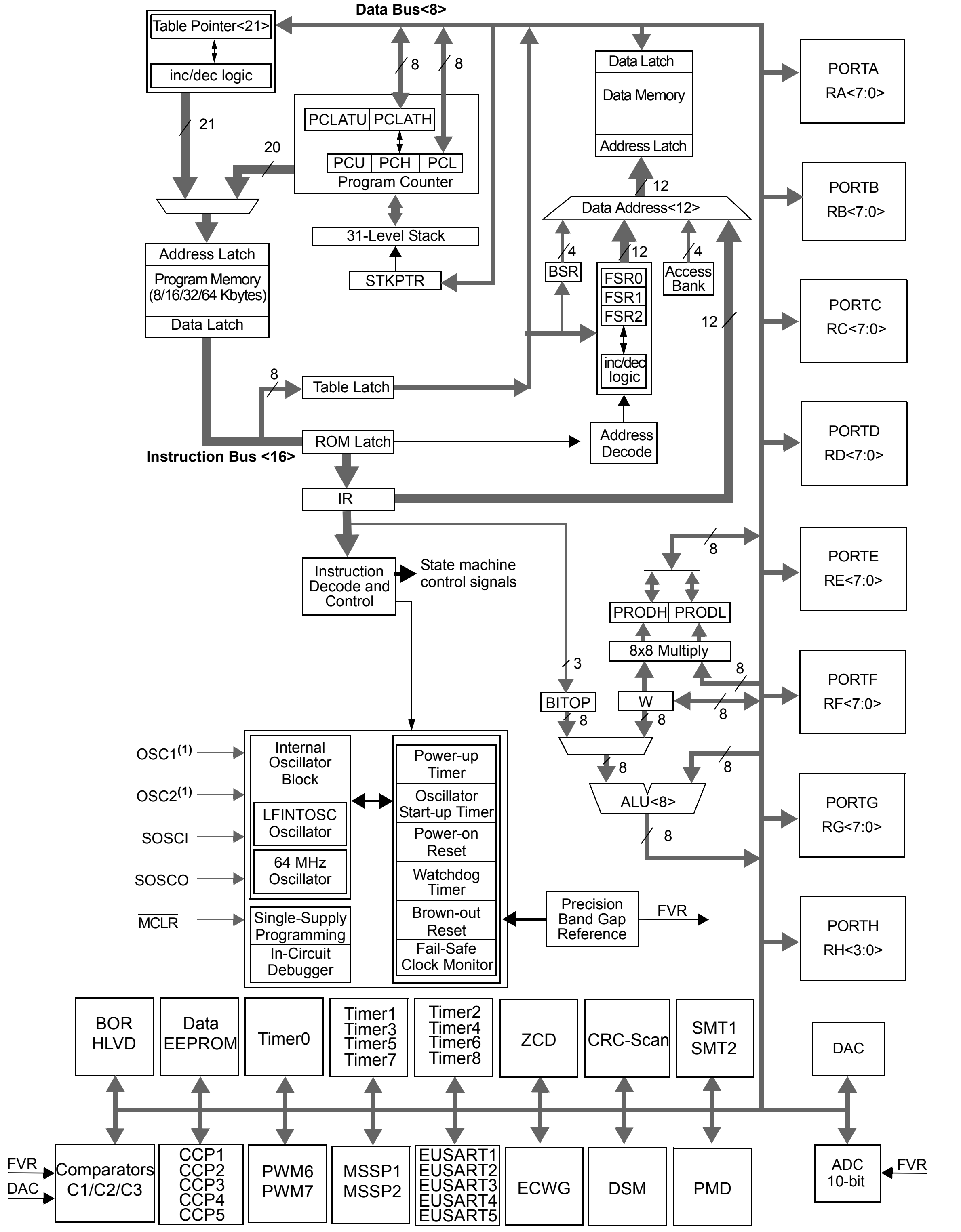
Note 1: This is a PPS remappable input signal. The input function may be moved from the default location shown to one of several other PORTx pins. Refer to Table 17-1 for details on which PORT pins may be used for this signal.

2: All output signals shown in this row are PPS remappable. These signals may be mapped to output onto one of several PORTx pin options as described in Register 17-2

3: This is a bidirectional signal. For normal module operation, the firmware should map this signal to the same pin in both the PPS input and PPS output registers.

4: These pins are configured for I^{2}C™ logic levels; The SCLx/SDAx signals may be assigned to any of these pins. PPS assignments to the other pins (e.g., RB1) will operate, but input logic levels will be standard TTL/ST as selected by the INLVL register, instead of the I^{2}C specific or SMBus input buffer thresholds.

FIGURE 1-1: PIC18(L)F6XK40 FAMILY BLOCK DIAGRAM



Note 1: OSC1/CLKIN and OSC2/CLKOUT are only available in select oscillator modes and when these pins are not being used as digital I/O. Refer to **Section 4.0 "Oscillator Module (with Fail-Safe Clock Monitor)"** for additional information.

5.1 Clock Source

The input to the reference clock output can be selected using the CLKRCLK register.

5.1.1 CLOCK SYNCHRONIZATION

Once the reference clock enable (EN) is set, the module is ensured to be glitch-free at start-up.

When the reference clock output is disabled, the output signal will be disabled immediately.

Clock dividers and clock duty cycles can be changed while the module is enabled, but glitches may occur on the output. To avoid possible glitches, clock dividers and clock duty cycles should be changed only when the CLKREN is clear.

5.2 Programmable Clock Divider

The module takes the clock input and divides it based on the value of the DIV<2:0> bits of the CLKRCON register (Register 5-1).

The following configurations can be made based on the DIV<2:0> bits:

- Base FOSC value
- FOSC divided by 2
- FOSC divided by 4
- FOSC divided by 8
- FOSC divided by 16
- FOSC divided by 32
- FOSC divided by 64
- FOSC divided by 128

The clock divider values can be changed while the module is enabled; however, in order to prevent glitches on the output, the DIV<2:0> bits should only be changed when the module is disabled (EN = 0).

5.3 Selectable Duty Cycle

The DC<1:0> bits of the CLKRCON register can be used to modify the duty cycle of the output clock. A duty cycle of 25%, 50%, or 75% can be selected for all clock rates, with the exception of the undivided base FOSC value.

The duty cycle can be changed while the module is enabled; however, in order to prevent glitches on the output, the DC<1:0> bits should only be changed when the module is disabled (EN = 0).

Note: The DC1 bit is reset to ‘1’. This makes the default duty cycle 50% and not 0%.

5.4 Operation in Sleep Mode

The reference clock output module clock is based on the system clock. When the device goes to Sleep, the module outputs will remain in their current state. This will have a direct effect on peripherals using the reference clock output as an input signal. No change should occur in the module from entering or exiting from Sleep.

6.0 POWER-SAVING OPERATION MODES

The purpose of the Power-Down modes is to reduce power consumption. There are three Power-Down modes:

- Doze mode
- Sleep mode
- Idle mode

6.1 Doze Mode

Doze mode allows for power saving by reducing CPU operation and program memory (PFM) access, without affecting peripheral operation. Doze mode differs from Sleep mode because the bandgap and system oscillators continue to operate, while only the CPU and PFM are affected. The reduced execution saves power by eliminating unnecessary operations within the CPU and memory.

When the Doze Enable (DOZEN) bit is set (DOZEN = 1), the CPU executes only one instruction cycle out of every N cycles as defined by the DOZE<2:0> bits of the CPUDOZE register. For example, if DOZE<2:0> = 001, the instruction cycle ratio is 1:4. The CPU and memory execute for one instruction cycle and then lay idle for three instruction cycles. During the unused cycles, the peripherals continue to operate at the system clock speed.

6.1.1 DOZE OPERATION

The Doze operation is illustrated in Figure 6-1. For this example:

- Doze enable (DOZEN) bit set (DOZEN = 1)
- DOZE<2:0> = 001 (1:4) ratio
- Recover-on-Interrupt (ROI) bit set (ROI = 1)

As with normal operation, the PFM fetches for the next instruction cycle. The Q-clocks to the peripherals continue throughout.

8.3 Power-on Reset (POR)

The POR circuit holds the device in Reset until VDD has reached an acceptable level for minimum operation. Slow rising VDD, fast operating speeds or analog performance may require greater than minimum VDD. The PWRT, BOR or MCLR features can be used to extend the start-up period until all device operation conditions have been met.

8.4 Brown-out Reset (BOR)

The BOR circuit holds the device in Reset when VDD reaches a selectable minimum level. Between the POR and BOR, complete voltage range coverage for execution protection can be implemented.

The Brown-out Reset module has four operating modes controlled by the BOREN<1:0> bits in Configuration Words. The four operating modes are:

- BOR is always on
- BOR is off when in Sleep
- BOR is controlled by software
- BOR is always off

Refer to Table 8-1 for more information.

The Brown-out Reset voltage level is selectable by configuring the BORV<1:0> bits in Configuration Words.

A VDD noise rejection filter prevents the BOR from triggering on small events. If VDD falls below VBOR for a duration greater than parameter TBORDC, the device will reset. See Table 37-11 for more information.

8.4.1 BOR IS ALWAYS ON

When the BOREN bits of Configuration Words are programmed to '11', the BOR is always on. The device start-up will be delayed until the BOR is ready and VDD is higher than the BOR threshold.

BOR protection is active during Sleep. The BOR does not delay wake-up from Sleep.

8.4.2 BOR IS OFF IN SLEEP

When the BOREN bits of Configuration Words are programmed to '10', the BOR is on, except in Sleep. The device start-up will be delayed until the BOR is ready and VDD is higher than the BOR threshold.

BOR protection is not active during Sleep. The device wake-up will be delayed until the BOR is ready.

8.4.3 BOR CONTROLLED BY SOFTWARE

When the BOREN bits of Configuration Words are programmed to '01', the BOR is controlled by the SBOREN bit of the BORCON register. The device start-up is not delayed by the BOR ready condition or the VDD level.

BOR protection begins as soon as the BOR circuit is ready. The status of the BOR circuit is reflected in the BORRDY bit of the BORCON register.

BOR protection is unchanged by Sleep.

8.4.4 BOR AND BULK ERASE

BOR is forced ON during PFM Bulk Erase operations to make sure that the system code protection cannot be compromised by reducing VDD.

During Bulk Erase, the BOR is enabled at 2.45V for F and LF devices, even if it is configured to some other value. If VDD falls, the erase cycle will be aborted, but the device will not be reset.

10.2.3 LOOK-UP TABLES IN PROGRAM MEMORY

There may be programming situations that require the creation of data structures, or look-up tables, in program memory. For PIC18 devices, look-up tables can be implemented in two ways:

- Computed `GOTO`
- Table Reads

10.2.3.1 Computed `GOTO`

A computed `GOTO` is accomplished by adding an offset to the program counter. An example is shown in Example 10-2.

A look-up table can be formed with an `ADDWF PCL` instruction and a group of `RETLW nn` instructions. The W register is loaded with an offset into the table before executing a call to that table. The first instruction of the called routine is the `ADDWF PCL` instruction. The next instruction executed will be one of the `RETLW nn` instructions that returns the value ‘nn’ to the calling function.

The offset value (in WREG) specifies the number of bytes that the program counter should advance and should be multiples of two (LSb = `0`).

In this method, only one data byte may be stored in each instruction location and room on the return address stack is required.

EXAMPLE 10-2: COMPUTED `GOTO` USING AN OFFSET VALUE

```
        MOVF    OFFSET, W
        CALL    TABLE
ORG     nn00h
TABLE   ADDWF   PCL
        RETLW   nnh
        RETLW   nnh
        RETLW   nnh
        .
        .
        .
```

10.2.3.2 Table Reads and Table Writes

A better method of storing data in program memory allows two bytes of data to be stored in each instruction location.

Look-up table data may be stored two bytes per program word by using table reads and writes. The Table Pointer (TBLPTR) register specifies the byte address and the Table Latch (TABLAT) register contains the data that is read from or written to program memory. Data is transferred to or from program memory one byte at a time.

Table read and table write operations are discussed further in **Section 11.1.1 “Table Reads and Table Writes”**.

11.1.1 TABLE READS AND TABLE WRITES

In order to read and write program memory, there are two operations that allow the processor to move bytes between the program memory space and the data RAM:

- Table Read (TBLRD)
- Table Write (TBLWT)

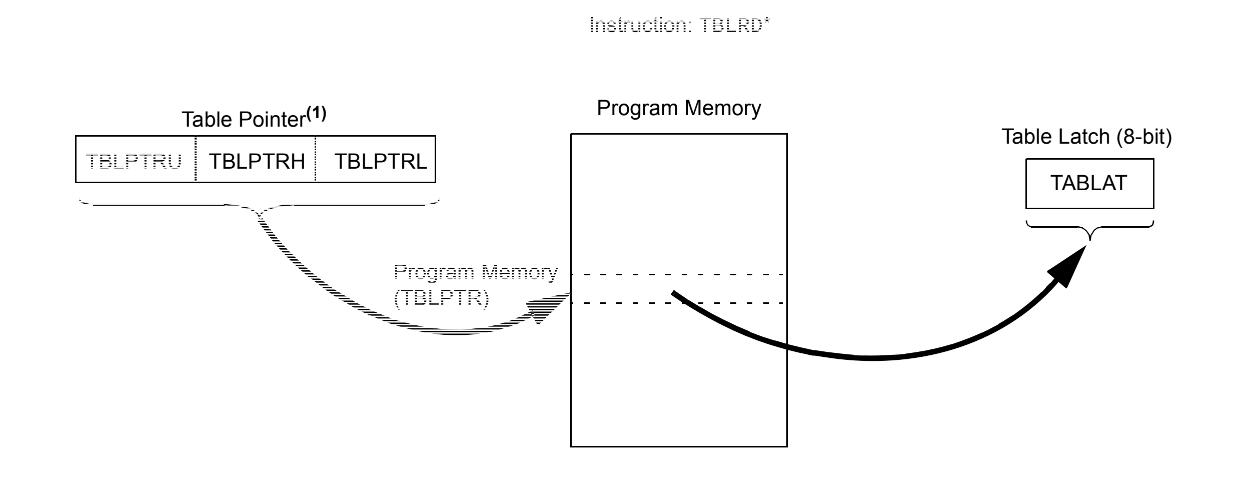
The program memory space is 16 bits wide, while the data RAM space is eight bits wide. Table reads and table writes move data between these two memory spaces through an 8-bit register (TABLAT).

The table read operation retrieves one byte of data directly from program memory and places it into the TABLAT register. Figure 11-1 shows the operation of a table read.

The table write operation stores one byte of data from the TABLAT register into a write block holding register. The procedure to write the contents of the holding registers into program memory is detailed in **Section 11.1.6 "Writing to Program Flash Memory"**. Figure 11-2 shows the operation of a table write with program memory and data RAM.

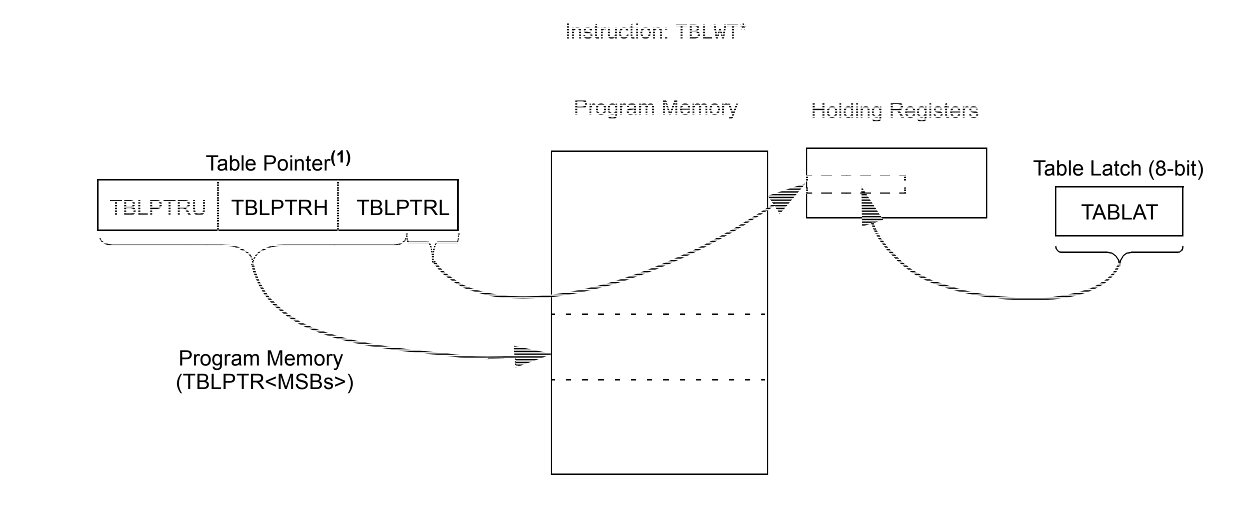
Table operations work with byte entities. Tables containing data, rather than program instructions, are not required to be word aligned. Therefore, a table can start and end at any byte address. If a table write is being used to write executable code into program memory, program instructions will need to be word aligned.

FIGURE 11-1: TABLE READ OPERATION



Note 1: Table Pointer register points to a byte in program memory.

FIGURE 11-2: TABLE WRITE OPERATION



Note 1: During table writes the Table Pointer does not point directly to program memory. The LSBs of TBLPRTL actually point to an address within the write block holding registers. The MSBs of the Table Pointer determine where the write block will eventually be written. The process for writing the holding registers to the program memory array is discussed in **Section 11.1.6 "Writing to Program Flash Memory".**

EXAMPLE 11-4: WRITING TO PROGRAM FLASH MEMORY (CONTINUED)

```
WRITE_BYTE_TO_HREGS
             MOVF     POSTINC0, W                ; get low byte of buffer data
             MOVWF    TABLAT                     ; present data to table latch
             TBLWT+*                             ; write data, perform a short write
                                                 ; to internal TBLWT holding register.
             DECFSZ   COUNTER                    ; loop until holding registers are full
             BRA      WRITE_WORD_TO_HREGS
PROGRAM_MEMORY
             BCF      NVMCON1, NVMREG0           ; point to Program Flash Memory
             BSF      NVMCON1, NVMREG1           ; point to Program Flash Memory
             BSF      NVMCON1, WREN              ; enable write to memory
             BCF      NVMCON1, FREE              ; enable write to memory
             BCF      INTCON, GIE                ; disable interrupts
             MOVLW    55h
Required     MOVWF    NVMCON2                    ; write 55h
Sequence     MOVLW    0AAh
             MOVWF    NVMCON2                    ; write 0AAh
             BSF      NVMCON1, WR                ; start program (CPU stall)
             DCFSZ    COUNTER2                   ; repeat for remaining write blocks
             BRA      WRITE_BYTE_TO_HREGS
             BSF      INTCON, GIE                ; re-enable interrupts
             BCF      NVMCON1, WREN              ; disable write to memory
```

13.11.7 IN-CIRCUIT DEBUG (ICD) INTERACTION

The scanner freezes when an ICD halt occurs, and remains frozen until user-mode operation resumes. The debugger may inspect the SCANCON0 and SCANLADR registers to determine the state of the scan.

The ICD interaction with each operating mode is summarized in Table 13-4.

TABLE 13-4: ICD AND SCANNER INTERACTIONS

ICD Halt	Scanner Operating Mode		
	Peek	**Concurrent Triggered**	**Burst**
External Halt	If scanner would peek an instruction that is not executed (because of ICD entry), the peek will occur after ICD exit, when the instruction executes.	If external halt is asserted during a scan cycle, the instruction (delayed by scan) may or may not execute before ICD entry, depending on external halt timing.	If external halt is asserted during the `BSF(SCANCON.GO)`, ICD entry occurs, and the burst is delayed until ICD exit. Otherwise, the current NVM-access cycle will complete, and then the scanner will be interrupted for ICD entry.
		If external halt is asserted during the cycle immediately prior to the scan cycle, both scan and instruction execution happen after the ICD exits.	If external halt is asserted during the burst, the burst is suspended and will resume with ICD exit.
PC Breakpoint		Scan cycle occurs before ICD entry and instruction execution happens after the ICD exits.	If PCPB (or single step) is on `BSF(SCANCON.GO)`, the ICD is entered before execution; execution of the burst will occur at ICD exit, and the burst will run to completion. Note that the burst can be interrupted by an external halt.
Data Breakpoint		The instruction with the dataBP executes and ICD entry occurs immediately after. If scan is requested during that cycle, the scan cycle is postponed until the ICD exits.	
Single Step		If a scan cycle is ready after the debug instruction is executed, the scan will read PFM and then the ICD is re-entered.	
SWBP and ICDINST		If scan would stall a SWBP, the scan cycle occurs and the ICD is entered.	If SWBP replaces `BSF(SCANCON.GO)`, the ICD will be entered; instruction execution will occur at ICD exit (from ICDINSTR register), and the burst will run to completion.

13.11.8 PERIPHERAL MODULE DISABLE

Both the CRC and scanner module can be disabled individually by setting the CRCMD and SCANMD bits of the PMD0 register (Register 7-1). The SCANMD can be used to enable or disable to the scanner module only if the SCANE bit of Configuration Word 4 is set. If the SCANE bit is cleared, then the scanner module is not available for use and the SCANMD bit is ignored.

14.0 INTERRUPTS

The PIC18(L)F6xK40 devices have multiple interrupt sources and an interrupt priority feature that allows most interrupt sources to be assigned a high or low priority level. The high priority interrupt vector is at 0008h and the low priority interrupt vector is at 0018h. A high priority interrupt event will interrupt a low priority interrupt that may be in progress.

The registers for controlling interrupt operation are:

- INTCON
- PIR1, PIR2, PIR3, PIR4, PIR5, PIR6, PIR7, PIR8, PIR9
- PIE1, PIE2, PIE3, PIE4, PIE5, PIE6, PIE7, PIE8, PIE9
- IPR1, IPR2, IPR3, IPR4, IPR5, IPR6, IPR7, IPR8, IPR9

It is recommended that the Microchip header files supplied with MPLAB® IDE be used for the symbolic bit names in these registers. This allows the assembler/ compiler to automatically take care of the placement of these bits within the specified register.

In general, interrupt sources have three bits to control their operation. They are:

- **Flag bit** to indicate that an interrupt event occurred
- **Enable bit** that allows program execution to branch to the interrupt vector address when the flag bit is set
- **Priority bit** to select high priority or low priority

14.1 Mid-Range Compatibility

When the IPEN bit is cleared (default state), the interrupt priority feature is disabled and interrupts are compatible with PIC® microcontroller mid-range devices. In Compatibility mode, the interrupt priority bits of the IPRx registers have no effect. The PEIE/GIEL bit of the INTCON register is the global interrupt enable for the peripherals. The PEIE/GIEL bit disables only the peripheral interrupt sources and enables the peripheral interrupt sources when the GIE/GIEH bit is also set. The GIE/GIEH bit of the INTCON register is the global interrupt enable which enables all non-peripheral interrupt sources and disables all interrupt sources, including the peripherals. All interrupts branch to address 0008h in Compatibility mode.

14.2 Interrupt Priority

The interrupt priority feature is enabled by setting the IPEN bit of the INTCON register. When interrupt priority is enabled the GIE/GIEH and PEIE/GIEL Global Interrupt Enable bits of Compatibility mode are replaced by the GIEH high priority, and GIEL low priority, global interrupt enables. When the IPEN bit is set, the GEIH bit of the INTCON register enables all interrupts which have their associated bit in the IPRx register set. When the GEIH bit is cleared, then all interrupt sources including those selected as low priority in the IPRx register are disabled.

When both GIEH and GIEL bits are set, all interrupts selected as low priority sources are enabled.

A high priority interrupt will vector immediately to address 00 0008h and a low priority interrupt will vector to address 00 0018h.

14.3 Interrupt Response

When an interrupt is responded to, the Global Interrupt Enable bit is cleared to disable further interrupts. The GIE/GIEH bit is the Global Interrupt Enable when the IPEN bit is cleared. When the IPEN bit is set, enabling interrupt priority levels, the GIEH bit is the high priority Global Interrupt Enable and the GIEL bit is the low priority Global Interrupt Enable. High priority interrupt sources can interrupt a low priority interrupt. Low priority interrupts are not processed while high priority interrupts are in progress.

The return address is pushed onto the stack and the PC is loaded with the interrupt vector address (0008h or 0018h). Once in the Interrupt Service Routine, the source(s) of the interrupt can be determined by polling the interrupt flag bits in the INTCONx and PIRx registers. The interrupt flag bits must be cleared by software before re-enabling interrupts to avoid repeating the same interrupt.

The “return from interrupt” instruction, `RETFIE`, exits the interrupt routine and sets the GIE/GIEH bit (GIEH or GIEL if priority levels are used), which re-enables interrupts.

For external interrupt events, such as the INT pins or the Interrupt-on-change pins, the interrupt latency will be three to four instruction cycles. The exact latency is the same for one-cycle or two-cycle instructions. Individual interrupt flag bits are set, regardless of the status of their corresponding enable bits or the Global Interrupt Enable bit.

Note:	Do not use the `MOVFF` instruction to modify any of the interrupt control registers while **any** interrupt is enabled. Doing so may cause erratic microcontroller behavior.

REGISTER 14-31: IPR9: PERIPHERAL INTERRUPT PRIORITY REGISTER 9

U-0	U-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
—	—	SMT2PWAIP	SMT2PRAIP	SMT2IP	SMT1PWAIP	SMT1PRAIP	SMT1IP
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 **SMT2PWAIP:** SMT2 Pulse Width Acquisition Interrupt Priority bit
1 = High priority
0 = Low priority

bit 4 **SMT2PRAIP:** SMT2 Period Acquisition Interrupt Priority bit
1 = High priority
0 = Low priority

bit 3 **SMT2IP:** SMT2 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 2 **SMT1PWAIP:** SMT1 Pulse Width Acquisition Interrupt Priority bit
1 = High priority
0 = Low priority

bit 1 **SMT1PRAIP:** SMT1 Period Acquisition Interrupt Priority bit
1 = High priority
0 = Low priority

bit 0 **SMT1IP:** SMT1 Interrupt Priority bit
1 = High priority
0 = Low priority

REGISTER 15-6: ODCONx: OPEN-DRAIN CONTROL REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
ODCx7	ODCx6	ODCx5	ODCx4	ODCx3	ODCx2	ODCx1	ODCx0
bit 7							bit 0

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

bit 7-0 **ODCx<7:0>**: Open-Drain Configuration on Pins Rx<7:0>
1 = Output drives only low-going signals (sink current only)
0 = Output drives both high-going and low-going signals (source and sink current)

TABLE 15-6: OPEN-DRAIN CONTROL REGISTERS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ODCONA	ODCONA7	ODCONA6	ODCONA5	ODCONA4	ODCONA3	ODCONA2	ODCONA1	ODCONA0
ODCONB	ODCONB7	ODCONB6	ODCONB5	ODCONB4	ODCONB3	ODCONB2	ODCONB1	ODCONB0
ODCONC	ODCONC7	ODCONC6	ODCONC5	ODCONC4	ODCONC3	ODCONC2	ODCONC1	ODCONC0
ODCOND	ODCOND7	ODCOND6	ODCOND5	ODCOND4	ODCOND3	ODCOND2	ODCOND1	ODCOND0
ODCONE	ODCONE7	ODCONE6	ODCONE5	ODCONE4	ODCONE3	ODCONE2	ODCONE1	ODCONE0
ODCONF	ODCONF7	ODCONF6	ODCONF5	ODCONF4	ODCONF3	ODCONF2	ODCONF1	ODCONF0
ODCONG	ODCONG7	ODCONG6	—	ODCONG4	ODCONG3	ODCONG2	ODCONG1	ODCONG0
ODCONH	—	—	—	—	ODCONH3	ODCONH2	ODCONH1	ODCONH0

TABLE 20-3: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER2

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE/GIEH	PEIE/GIEL	IPEN	—	INT3EDG	INT2EDG	INT1EDG	INT0EDG	173
PIE5	TMR8IE	TMR7IE	TMR6IE	TMR5IE	TMR4IE	TMR3IE	TMR2IE	TMR1IE	190
PIR5	TMR8IF	TMR7IF	TMR6IF	TMR5IF	TMR4IF	TMR3IF	TMR2IF	TMR1IF	178
IPR5	TMR8IP	TMR7IP	TMR6IP	TMR5IP	TMR4IP	TMR3IP	TMR2IP	TMR1IP	200
PMD1	TMR7MD	TMR6MD	TMR5MD	TMR4MD	TMR3MD	TMR2MD	TMR1MD	TMR0MD	67
PMD2	—	—	CWGMD	—	DSMMD	SMT2MD	SMT1MD	TMR8MD	68
PR2	Timer2 Module Period Register								255*
TMR2	Holding Register for the 8-bit TMR2 Register								255*
T2CON	ON	CKPS<2:0>			OUTPS<3:0>				273
T2CLKCON	—	—	—	—	—	CS<2:0>			275
T2RST	—	—	—	—	RSEL<3:0>				276
T2HLT	PSYNC	CPOL	CSYNC	MODE<4:0>					274
PR4	Timer4 Module Period Register								255*
TMR4	Holding Register for the 8-bit TMR4 Register								255*
T4CON	ON	CKPS<2:0>			OUTPS<3:0>				273
T4CLKCON	—	—	—	—	—	CS<2:0>			275
T4RST	—	—	—	—	RSEL<3:0>				276
T4HLT	PSYNC	CPOL	CSYNC	MODE<4:0>					274
PR6	Timer6 Module Period Register								255*
TMR6	Holding Register for the 8-bit TMR6 Register								255*
T6CON	ON	CKPS<2:0>			OUTPS<3:0>				273
T6CLKCON	—	—	—	—	—	CS<2:0>			275
T6RST	—	—	—	—	RSEL<3:0>				276
T6HLT	PSYNC	CPOL	CSYNC	MODE<4:0>					274
PR8	Timer8 Module Period Register								255*
TMR8	Holding Register for the 8-bit TMR8 Register								255*
T8CON	ON	CKPS<2:0>			OUTPS<3:0>				273
T8CLKCON	—	—	—	—	—	CS<2:0>			275
T8RST	—	—	—	—	RSEL<3:0>				276
T8HLT	$\overline{\text{PSYNC}}$	CPOL	$\overline{\text{CSYNC}}$	MODE<4:0>					274
T2INPPS	—	—	T2INPPS<5:0>						225
T4INPPS	—	—	T4INPPS<5:0>						225
T6INPPS	—	—	T6INPPS<5:0>						225
T8INPPS	—	—	T8INPPS<5:0>						225

Legend: — = unimplemented location, read as ‘0’. Shaded cells are not used for Timer2 module.

* Page provides register information.

21.0 CAPTURE/COMPARE/PWM MODULE

The Capture/Compare/PWM module is a peripheral that allows the user to time and control different events, and to generate Pulse-Width Modulation (PWM) signals. In Capture mode, the peripheral allows the timing of the duration of an event. The Compare mode allows the user to trigger an external event when a predetermined amount of time has expired. The PWM mode can generate Pulse-Width Modulated signals of varying frequency and duty cycle.

This family of devices contains five standard Capture/Compare/PWM modules (CCP1, CCP2, CCP3, CCP4, and CCP5). Each individual CCP module can select the timer source that controls the module. Each module has an independent timer selection which can be accessed using the CxTSEL bits in the CCPTMRS0 register (Register 21-2). The default timer selection is TMR1 when using Capture/Compare mode and TMR2 when using PWM mode in the CCPx module.

Please note that the Capture/Compare mode operation is described with respect to TMR1 and the PWM mode operation is described with respect to TMR2 in the following sections.

The Capture and Compare functions are identical for all CCP modules.

Note 1: In devices with more than one CCP module, it is very important to pay close attention to the register names used. A number placed after the module acronym is used to distinguish between separate modules. For example, the CCP1CON and CCP2CON control the same operational aspects of two completely different CCP modules.

2: Throughout this section, generic references to a CCP module in any of its operating modes may be interpreted as being equally applicable to CCPx module. Register names, module signals, I/O pins, and bit names may use the generic designator ‘x’ to indicate the use of a numeral to distinguish a particular module, when required.

21.1 CCP Module Configuration

Each Capture/Compare/PWM module is associated with a control register (CCPxCON), a capture input selection register (CCPxCAP) and a data register (CCPRx). The data register, in turn, is comprised of two 8-bit registers: CCPRxL (low byte) and CCPRxH (high byte).

21.1.1 CCP MODULES AND TIMER RESOURCES

The CCP modules utilize Timers 1 through 8 that vary with the selected mode. Various timers are available to the CCP modules in Capture, Compare or PWM modes, as shown in Table 21-1.

TABLE 21-1: CCP MODE – TIMER RESOURCE

CCP Mode	Timer Resource
Capture	Timer1, Timer3, Timer5, or Timer7
Compare	
PWM	Timer2, Timer4, Timer6, or Timer8

The assignment of a particular timer to a module is determined by the timer to CCP enable bits in the CCPTMRS0/1 registers (see Register 21-2 and Register 21-3) All of the modules may be active at once and may share the same timer resource if they are configured to operate in the same mode (Capture/Compare or PWM) at the same time.

21.1.2 OPEN-DRAIN OUTPUT OPTION

When operating in Output mode (the Compare or PWM modes), the drivers for the CCPx pins can be optionally configured as open-drain outputs. This feature allows the voltage level on the pin to be pulled to a higher level through an external pull-up resistor and allows the output to communicate with external circuits without the need for additional level shifters.

24.7 Rising Edge and Reverse Dead Band

In Half-Bridge mode, the rising edge dead band delays the turn-on of the CWG1A output after the rising edge of the CWG data input. In Full-Bridge mode, the reverse dead-band delay is only inserted when changing directions from Forward mode to Reverse mode, and only the modulated output CWG1B is affected.

The CWG1DBR register determines the duration of the dead-band interval on the rising edge of the input source signal. This duration is from 0 to 64 periods of the CWG clock.

Dead band is always initiated on the edge of the input source signal. A count of zero indicates that no dead band is present.

If the input source signal reverses polarity before the dead-band count is completed, then no signal will be seen on the respective output.

The CWG1DBR register value is double-buffered. When EN = `0` (Register 24-1), the buffer is loaded when CWG1DBR is written. If EN = `1`, then the buffer will be loaded at the rising edge following the first falling edge of the data input, after the LD bit (Register 24-1) is set. Refer to Figure 24-12 for an example.

24.8 Falling Edge and Forward Dead Band

In Half-Bridge mode, the falling edge dead band delays the turn-on of the CWG1B output at the falling edge of the CWG data input. In Full-Bridge mode, the forward dead-band delay is only inserted when changing directions from Reverse mode to Forward mode, and only the modulated output CWG1D is affected.

The CWG1DBF register determines the duration of the dead-band interval on the falling edge of the input source signal. This duration is from zero to 64 periods of CWG clock.

Dead-band delay is always initiated on the edge of the input source signal. A count of zero indicates that no dead band is present.

If the input source signal reverses polarity before the dead-band count is completed, then no signal will be seen on the respective output.

The CWG1DBF register value is double-buffered. When EN = `0` (Register 24-1), the buffer is loaded when CWG1DBF is written. If EN = `1`, then the buffer will be loaded at the rising edge following the first falling edge of the data input after the LD (Register 24-1) is set. Refer to Figure 24-13 for an example.

27.9.8 GENERAL CALL ADDRESS SUPPORT

The addressing procedure for the I^2C bus is such that the first byte after the Start condition usually determines which device will be the slave addressed by the master device. The exception is the general call address which can address all devices. When this address is used, all devices should, in theory, respond with an acknowledge.

The general call address is a reserved address in the I^2C protocol, defined as address 0x00. When the GCEN bit of the SSPxCON2 register is set, the slave module will automatically $\overline{\text{ACK}}$ the reception of this address regardless of the value stored in SSPxADD. After the slave clocks in an address of all zeros with the R/$\overline{\text{W}}$ bit clear, an interrupt is generated and slave software can read SSPxBUF and respond. Figure 27-24 shows a general call reception sequence.

In 10-bit Address mode, the UA bit will not be set on the reception of the general call address. The slave will prepare to receive the second byte as data, just as it would in 7-bit mode.

If the AHEN bit of the SSPxCON3 register is set, just as with any other address reception, the slave hardware will stretch the clock after the eighth falling edge of SCL. The slave must then set its ACKDT value and release the clock with communication progressing as it would normally.

27.9.9 SSP MASK REGISTER

An SSP Mask (SSPxMSK) register (Register 27-12) is available in I^2C Slave mode as a mask for the value held in the SSPSR register during an address comparison operation. A zero (‘0’) bit in the SSPxMSK register has the effect of making the corresponding bit of the received address a “don’t care”.

This register is reset to all ‘1’s upon any Reset condition and, therefore, has no effect on standard SSP operation until written with a mask value.

The SSP Mask register is active during:

- 7-bit Address mode: address compare of A<7:1>.

10-bit Address mode: address compare of A<7:0> only. The SSP mask has no effect during the reception of the first (high) byte of the address.

FIGURE 27-24: SLAVE MODE GENERAL CALL ADDRESS SEQUENCE

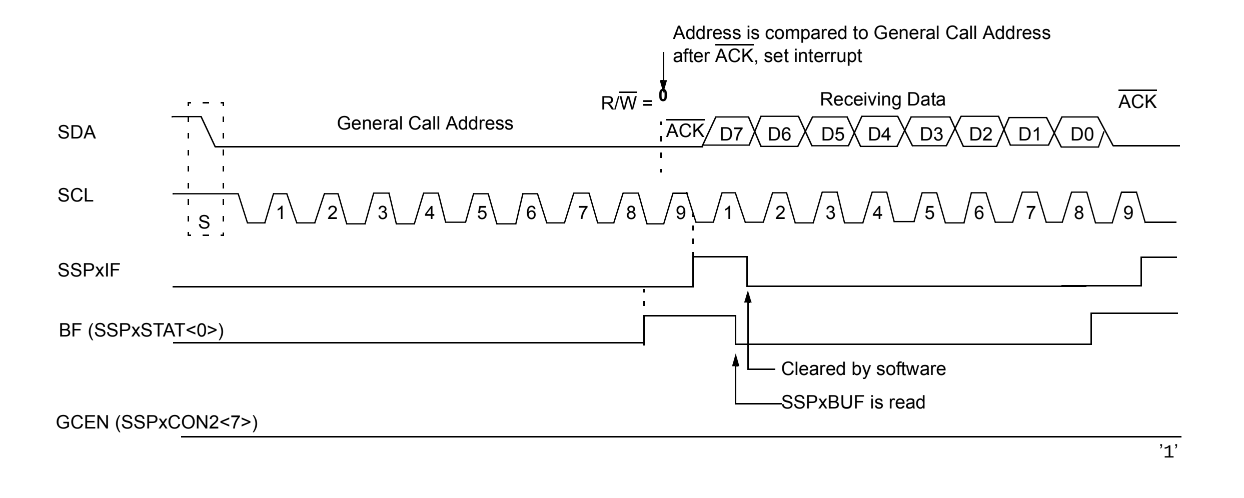


TABLE 28-5: SAMPLE BAUD RATES FOR ASYNCHRONOUS MODES (CONTINUED)

BAUD RATE	SYNC = 0, BRGH = 1, BRG16 = 0											
	Fosc = 8.000 MHz			Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 1.000 MHz		
	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	—	—	—	—	—	—	—	—	—	300	0.16	207
1200	—	—	—	1202	0.16	207	1200	0.00	191	1202	0.16	51
2400	2404	0.16	207	2404	0.16	103	2400	0.00	95	2404	0.16	25
9600	9615	0.16	51	9615	0.16	25	9600	0.00	23	—	—	—
10417	10417	0.00	47	10417	0.00	23	10473	0.53	21	10417	0.00	5
19.2k	19231	0.16	25	19.23k	0.16	12	19.2k	0.00	11	—	—	—
57.6k	55556	-3.55	8	—	—	—	57.60k	0.00	3	—	—	—
115.2k	—	—	—	—	—	—	115.2k	0.00	1	—	—	—

BAUD RATE	SYNC = 0, BRGH = 0, BRG16 = 1											
	Fosc = 32.000 MHz			Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 11.0592 MHz		
	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	300.0	0.00	6666	300.0	-0.01	4166	300.0	0.00	3839	300.0	0.00	2303
1200	1200	-0.02	3332	1200	-0.03	1041	1200	0.00	959	1200	0.00	575
2400	2401	-0.04	832	2399	-0.03	520	2400	0.00	479	2400	0.00	287
9600	9615	0.16	207	9615	0.16	129	9600	0.00	119	9600	0.00	71
10417	10417	0.00	191	10417	0.00	119	10378	-0.37	110	10473	0.53	65
19.2k	19.23k	0.16	103	19.23k	0.16	64	19.20k	0.00	59	19.20k	0.00	35
57.6k	57.14k	-0.79	34	56.818	-1.36	21	57.60k	0.00	19	57.60k	0.00	11
115.2k	117.6k	2.12	16	113.636	-1.36	10	115.2k	0.00	9	115.2k	0.00	5

BAUD RATE	SYNC = 0, BRGH = 0, BRG16 = 1											
	Fosc = 8.000 MHz			Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 1.000 MHz		
	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	299.9	-0.02	1666	300.1	0.04	832	300.0	0.00	767	300.5	0.16	207
1200	1199	-0.08	416	1202	0.16	207	1200	0.00	191	1202	0.16	51
2400	2404	0.16	207	2404	0.16	103	2400	0.00	95	2404	0.16	25
9600	9615	0.16	51	9615	0.16	25	9600	0.00	23	—	—	—
10417	10417	0.00	47	10417	0.00	23	10473	0.53	21	10417	0.00	5
19.2k	19.23k	0.16	25	19.23k	0.16	12	19.20k	0.00	11	—	—	—
57.6k	55556	-3.55	8	—	—	—	57.60k	0.00	3	—	—	—
115.2k	—	—	—	—	—	—	115.2k	0.00	1	—	—	—

REGISTER 31-2: DAC1CON1: DAC DATA 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
—	—	—	DAC1R<4:0>				
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 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **DAC1R<4:0>:** Data Input Register for DAC bits

TABLE 31-2: SUMMARY OF REGISTERS ASSOCIATED WITH THE DAC MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
DAC1CON0	EN	—	OE1	OE2	PSS<1:0>		—	NSS	484
DAC1CON1	—	—	—	DAC1R<4:0>					485
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		479

Legend: — = Unimplemented location, read as ‘0’. Shaded cells are not used with the DAC module.

TABLE 36-1: OPCODE FIELD DESCRIPTIONS

Field	Description
a	RAM access bit a = 0: RAM location in Access RAM (BSR register is ignored) a = 1: RAM bank is specified by BSR register
bbb	Bit address within an 8-bit file register (0 to 7).
BSR	Bank Select Register. Used to select the current RAM bank.
C, DC, Z, OV, N	ALU Status bits: **C**arry, **D**igit **C**arry, **Z**ero, **Ov**erflow, **N**egative.
d	Destination select bit d = 0: store result in WREG d = 1: store result in file register f
dest	Destination: either the WREG register or the specified register file location.
f	8-bit Register file address (00h to FFh) or 2-bit FSR designator (0h to 3h).
f_s	12-bit Register file address (000h to FFFh). This is the source address.
f_d	12-bit Register file address (000h to FFFh). This is the destination address.
GIE	Global Interrupt Enable bit.
k	Literal field, constant data or label (may be either an 8-bit, 12-bit or a 20-bit value).
label	Label name.
mm	The mode of the TBLPTR register for the table read and table write instructions. Only used with table read and table write instructions:
*	No change to register (such as TBLPTR with table reads and writes)
*+	Post-Increment register (such as TBLPTR with table reads and writes)
*-	Post-Decrement register (such as TBLPTR with table reads and writes)
+*	Pre-Increment register (such as TBLPTR with table reads and writes)
n	The relative address (2’s complement number) for relative branch instructions or the direct address for CALL/BRANCH and RETURN instructions.
PC	Program Counter.
PCL	Program Counter Low Byte.
PCH	Program Counter High Byte.
PCLATH	Program Counter High Byte Latch.
PCLATU	Program Counter Upper Byte Latch.
$\overline{\text{PD}}$	Power-down bit.
PRODH	Product of Multiply High Byte.
PRODL	Product of Multiply Low Byte.
s	Fast Call/Return mode select bit s = 0: do not update into/from shadow registers s = 1: certain registers loaded into/from shadow registers (Fast mode)
TBLPTR	21-bit Table Pointer (points to a Program Memory location).
TABLAT	8-bit Table Latch.
$\overline{\text{TO}}$	Time-out bit.
TOS	Top-of-Stack.
u	Unused or unchanged.
WDT	Watchdog Timer.
WREG	Working register (accumulator).
x	Don’t care (‘0’ or ‘1’). The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
z_s	7-bit offset value for indirect addressing of register files (source).
z_d	7-bit offset value for indirect addressing of register files (destination).
{ }	Optional argument.
[text]	Indicates an indexed address.
(text)	The contents of text.
[expr]<n>	Specifies bit n of the register indicated by the pointer expr.
→	Assigned to.
< >	Register bit field.
∈	In the set of.
italics	User defined term (font is Courier).

FIGURE 38-7: CLKOUT AND I/O TIMING

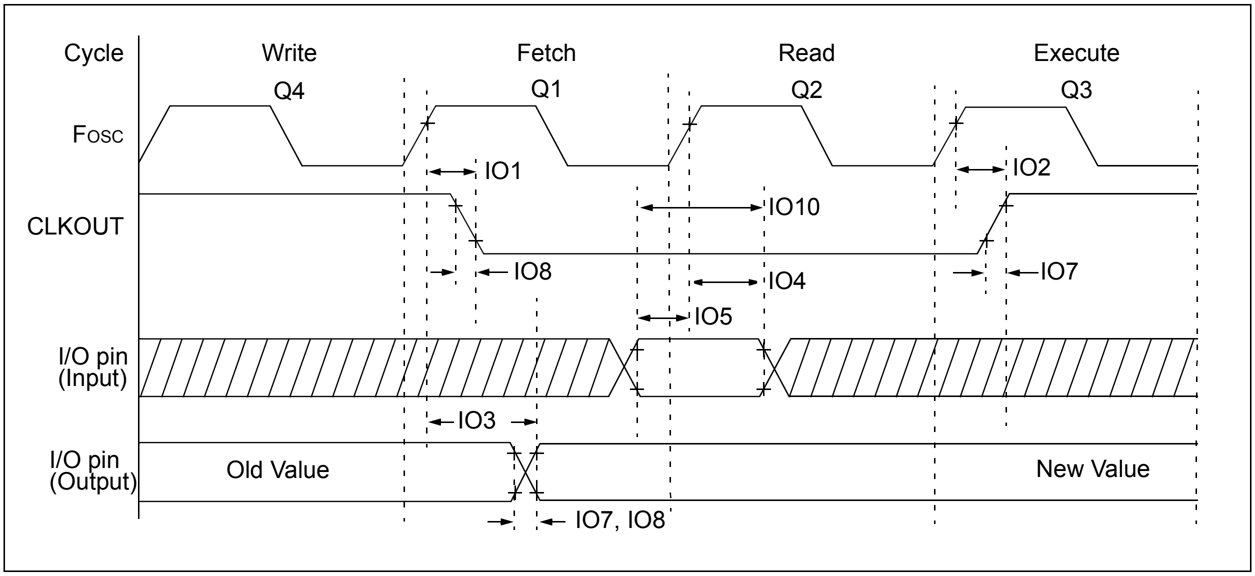


TABLE 38-10: I/O AND CLKOUT TIMING SPECIFICATIONS

Standard Operating Conditions (unless otherwise stated)							
Param No.	Sym.	Characteristic	Min.	Typ†	Max.	Units	Conditions
IO1*	$T_{CLKOUTH}$	CLKOUT rising edge delay (rising edge Fosc (Q1 cycle) to falling edge CLKOUT	—	—	70	ns	
IO2*	$T_{CLKOUTL}$	CLKOUT falling edge delay (rising edge Fosc (Q3 cycle) to rising edge CLKOUT	—	—	72	ns	
IO3*	T_{IO_VALID}	Port output valid time (rising edge Fosc (Q1 cycle) to port valid)	—	50	70	ns	
IO4*	T_{IO_SETUP}	Port input setup time (Setup time before rising edge Fosc – Q2 cycle)	20	—	—	ns	
IO5*	T_{IO_HOLD}	Port input hold time (Hold time after rising edge Fosc – Q2 cycle)	50	—	—	ns	
IO6*	T_{IOR_SLREN}	Port I/O rise time, slew rate enabled	—	25	—	ns	VDD = 3.0V
IO7*	T_{IOR_SLRDIS}	Port I/O rise time, slew rate disabled	—	5	—	ns	VDD = 3.0V
IO8*	T_{IOF_SLREN}	Port I/O fall time, slew rate enabled	—	25	—	ns	VDD = 3.0V
IO9*	T_{IOF_SLRDIS}	Port I/O fall time, slew rate disabled	—	5	—	ns	VDD = 3.0V
IO10*	T_{INT}	INT pin high or low time to trigger an interrupt	25	—	—	ns	
IO11*	T_{IOC}	Interrupt-on-Change minimum high or low time to trigger interrupt	25	—	—	ns	

*These parameters are characterized but not tested.