Microchip Technology - PIC16F1769T-I/SS 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	32MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	18
Program Memory Size	14KB (8K x 14)
Program Memory Type	FLASH
EEPROM Size	128 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 12x10b; D/A 2x5b, 2x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SSOP (0.209", 5.30mm Width)
Supplier Device Package	20-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f1769t-i-ss

EXAMPLE 3-2: ACCESSING PROGRAM MEMORY VIA FSRn

```
constants
   DW DATA0              ;First constant
   DW DATA1              ;Second constant
   DW DATA2
   DW DATA3
my_function
   ;… LOTS OF CODE…
   MOVLW   DATA_INDEX
   ADDLW   LOW constants
   MOVWF   FSR1L
   MOVLW   HIGH constants ;MSb sets
                           automatically
   MOVWF   FSR1H
   BTFSC   STATUS, C      ;carry from ADDLW?
   INCF    FSR1H, f       ;yes
   MOVIW   0[FSR1]
;THE PROGRAM MEMORY IS IN W
```

3.3 Data Memory Organization

The data memory is partitioned in 32 memory banks with 128 bytes in a bank. Each bank consists of (Figure 3-2):

- 12 core registers
- 20 Special Function Registers (SFRs)
- Up to 80 bytes of General Purpose RAM (GPR)
- 16 bytes of common RAM

The active bank is selected by writing the bank number into the Bank Select Register (BSR). Unimplemented memory will read as '0'. All data memory can be accessed either directly (via instructions that use the file registers) or indirectly via the two File Select Registers (FSRs). See **Section 3.6 "Indirect Addressing"** for more information.

Data memory uses a 12-bit address. The upper five bits of the address define the Bank address and the lower seven bits select the registers/RAM in that bank.

3.3.1 CORE REGISTERS

The core registers contain the registers that directly affect the basic operation. The core registers occupy the first 12 addresses of every data memory bank (addresses, x00h/x08h through x0Bh/x8Bh). These registers are listed below in Table 3-2. For detailed information, see Table 3-15.

TABLE 3-2: CORE REGISTERS

Addresses	BANKx
x00h or x80h	INDF0
x01h or x81h	INDF1
x02h or x82h	PCL
x03h or x83h	STATUS
x04h or x84h	FSR0L
x05h or x85h	FSR0H
x06h or x86h	FSR1L
x07h or x87h	FSR1H
x08h or x88h	BSR
x09h or x89h	WREG
x0Ah or x8Ah	PCLATH
x0Bh or x8Bh	INTCON

3.3.1.1 STATUS Register

The STATUS register, shown in Register 3-1, contains:

- The arithmetic status of the ALU
- The Reset status

The STATUS register can be the destination for any instruction, like any other register. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, the $\overline{TO}$ and $\overline{PD}$ bits are not writable. Therefore, the result of an instruction with the STATUS register as destination may be different than intended.

For example, `CLRF STATUS` will clear the upper three bits and set the Z bit. This leaves the STATUS register as '000u u1uu' (where u = unchanged).

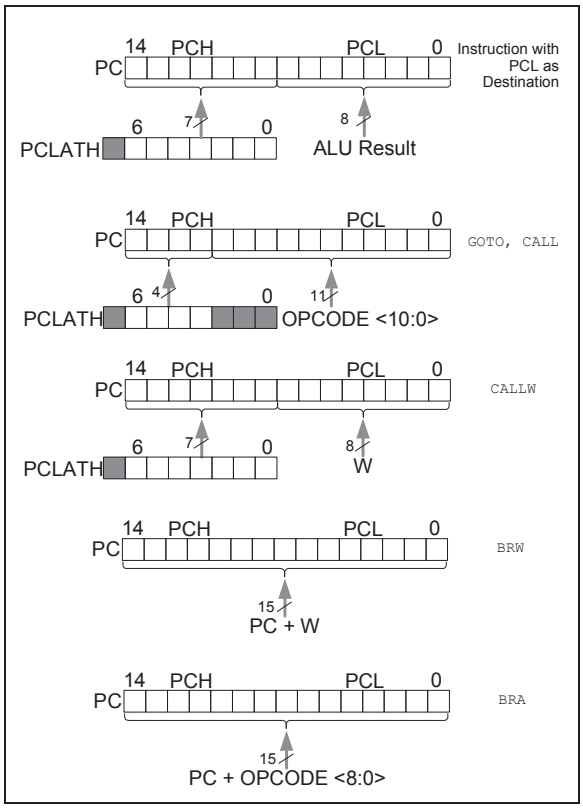
It is recommended, therefore, that only `BCF`, `BSF`, `SWAPF` and `MOVWF` instructions are used to alter the STATUS register, because these instructions do not affect any Status bits. For other instructions not affecting any Status bits, refer to **Section 35.0 "Instruction Set Summary"**.

Note:	The C and DC bits operate as $\overline{Borrow}$ and $\overline{Digit\ Borrow}$ out bits, respectively, in subtraction.

3.4 PCL and PCLATH

The Program Counter (PC) is 15 bits wide. The low byte comes from the PCL register, which is a readable and writable register. The high byte (PC<14:8>) is not directly readable or writable and comes from PCLATH. On any Reset, the PC is cleared. Figure 3-3 shows the five situations for the loading of the PC.

FIGURE 3-3: LOADING OF PC IN DIFFERENT SITUATIONS



3.4.1 MODIFYING PCL

Executing any instruction with the PCL register as the destination simultaneously causes the Program Counter PC<14:8> bits (PCH) to be replaced by the contents of the PCLATH register. This allows the entire contents of the Program Counter to be changed by writing the desired upper seven bits to the PCLATH register. When the lower eight bits are written to the PCL register, all 15 bits of the Program Counter will change to the values contained in the PCLATH register and those being written to the PCL register.

3.4.2 COMPUTED GOTO

A computed GOTO is accomplished by adding an offset to the Program Counter (ADDWF PCL). When performing a table read using a computed GOTO method, care should be exercised if the table location crosses a PCL memory boundary (each 256-byte block). Refer to Application Note AN556, *"Implementing a Table Read"* (DS00556).

3.4.3 COMPUTED FUNCTION CALLS

A computed function CALL allows programs to maintain tables of functions and provide another way to execute state machines or look-up tables. When performing a table read using a computed function CALL, care should be exercised if the table location crosses a PCL memory boundary (each 256-byte block).

If using the CALL instruction, the PCH<2:0> and PCL registers are loaded with the operand of the CALL instruction. PCH<6:3> is loaded with PCLATH<6:3>.

The CALLW instruction enables computed calls by combining PCLATH and W to form the destination address. A computed CALLW is accomplished by loading the W register with the desired address and executing CALLW. The PCL register is loaded with the value of W and PCH is loaded with PCLATH.

3.4.4 BRANCHING

The branching instructions add an offset to the PC. This allows relocatable code and code that crosses page boundaries. There are two forms of branching, BRW and BRA. The PC will have incremented to fetch the next instruction in both cases. When using either branching instruction, a PCL memory boundary may be crossed.

If using BRW, load the W register with the desired unsigned address and execute BRW. The entire PC will be loaded with the address PC + 1 + W.

If using BRA, the entire PC will be loaded with PC + 1 +, the signed value of the operand of the BRA instruction.

TABLE 6-5: SUMMARY OF REGISTERS ASSOCIATED WITH RESETS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BORCON	SBOREN	BORFS	—	—	—	—	—	BORRDY	90
PCON	STKOVF	STKUNF	—	$\overline{\text{RWDT}}$	$\overline{\text{RMCLR}}$	$\overline{\text{RI}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	94
STATUS	—	—	—	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	27
WDTCON	—	—	WDTPS<4:0>					SWDTEN	115

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by Resets.

10.2.3 ERASING FLASH PROGRAM MEMORY

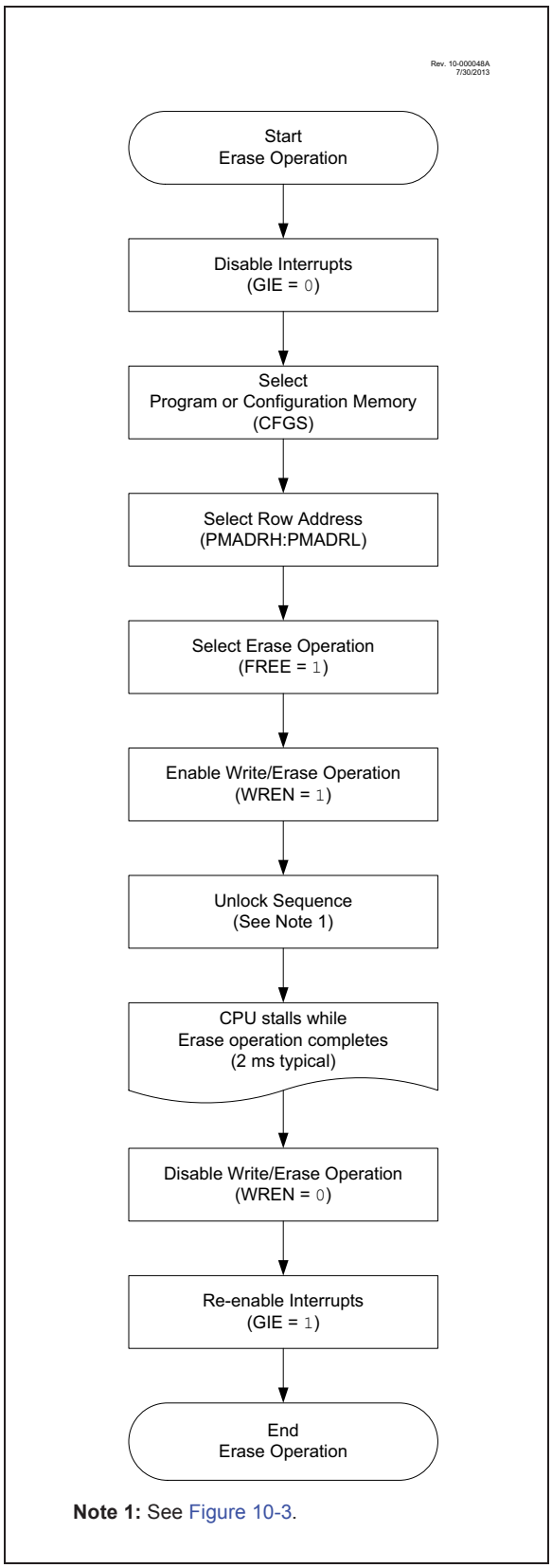
While executing code, program memory can only be erased by rows. To erase a row:

1. Load the PMADRH:PMADRL register pair with any address within the row to be erased.
2. Clear the CFGS bit of the PMCON1 register.
3. Set the FREE and WREN bits of the PMCON1 register.
4. Write 55h, then AAh, to PMCON2 (Flash programming unlock sequence).
5. Set control bit WR of the PMCON1 register to begin the erase operation.

See Example 10-2.

After the "`BSF PMCON1,WR`" instruction, the processor requires two cycles to set up the erase operation. The user must place two `NOP` instructions immediately following the WR bit set instruction. The processor will halt internal operations for the typical 2 ms erase time. This is not Sleep mode as the clocks and peripherals will continue to run. After the erase cycle, the processor will resume operation with the third instruction after the PMCON1 write instruction.

FIGURE 10-4: FLASH PROGRAM MEMORY ERASE FLOWCHART



Note 1: See Figure 10-3.

10.6 Register Definitions: Flash Program Memory Control

REGISTER 10-1: PMDATL: PROGRAM MEMORY DATA LOW BYTE REGISTER

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
PMDAT<7:0>							
bit 7							bit 0

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

bit 7-0 **PMDAT<7:0>**: Read/Write Value for Least Significant bits of Program Memory bits

REGISTER 10-2: PMDATH: PROGRAM MEMORY DATA HIGH BYTE 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
—	—	PMDAT<13:8>					
bit 7							bit 0

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

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

bit 5-0 **PMDAT<13:8>**: Read/Write Value for Most Significant bits of Program Memory bits

REGISTER 10-3: PMADRL: PROGRAM MEMORY ADDRESS LOW BYTE 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
PMADR<7:0>							
bit 7							bit 0

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

bit 7-0 **PMADR<7:0>**: Specifies the Least Significant bits for Program Memory Address bits

REGISTER 10-4: PMADRH: PROGRAM MEMORY ADDRESS HIGH BYTE REGISTER

U-1	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
—(1)	PMADR<14:8>						
bit 7							bit 0

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

bit 7 **Unimplemented:** Read as '1'

bit 6-0 **PMADR<14:8>**: Specifies the Most Significant bits for Program Memory Address bits

Note 1: Unimplemented, read as '1'.

16.2.6 ADC CONVERSION PROCEDURE

This is an example procedure for using the ADC to perform an Analog-to-Digital conversion:

1. Configure the PORT:
   - Disable the pin output driver (refer to the TRISx register)
   - Configure pin as an analog (refer to the ANSELx register)
   - Disable weak pull-ups either globally (Refer to the OPTION_REG register) or individually (Refer to the appropriate WPUx register)
2. Configure the ADC module:
   - Select ADC conversion clock
   - Configure voltage reference
   - Select ADC input channel
   - Turn on ADC module
3. Configure ADC interrupt (optional):
   - Clear ADC interrupt flag
   - Enable ADC interrupt
   - Enable peripheral interrupt
   - Enable global interrupt(1)
4. Wait the required acquisition time.(2)
5. Start conversion by setting the GO/$\overline{\text{DONE}}$ bit.
6. Wait for ADC conversion to complete by one of the following:
   - Polling the GO/$\overline{\text{DONE}}$ bit
   - Waiting for the ADC interrupt (interrupts enabled)
7. Read ADC result.
8. Clear the ADC interrupt flag (required if interrupt is enabled).

Note 1: The global interrupt can be disabled if the user is attempting to wake-up from Sleep and resume in-line code execution.

2: Refer to **Section 16.4 "ADC Acquisition Requirements"**.

EXAMPLE 16-1: ADC CONVERSION

```
;This code block configures the ADC
;for polling, Vdd and Vss references, FRC
;oscillator and AN0 input.
;
;Conversion start & polling for completion
; are included.
;
BANKSEL   ADCON1       ;
MOVLW     B'11110000'  ;Right justify, FRC
                       ;oscillator
MOVWF     ADCON1       ;Vdd and Vss Vref
BANKSEL   TRISA        ;
BSF       TRISA,0      ;Set RA0 to input
BANKSEL   ANSEL        ;
BSF       ANSEL,0      ;Set RA0 to analog
BANKSEL   WPUA
BCF       WPUA,0       ;Disable weak
                       ;pull-up on RA0
BANKSEL   ADCON0       ;
MOVLW     B'00000001'  ;Select channel AN0
MOVWF     ADCON0       ;Turn ADC On
CALL      SampleTime   ;Acquisiton delay
BSF       ADCON0,ADGO  ;Start conversion
BTFSC     ADCON0,ADGO  ;Is conversion done?
GOTO      $-1          ;No, test again
BANKSEL   ADRESH       ;
MOVF      ADRESH,W     ;Read upper 2 bits
MOVWF     RESULTHI     ;store in GPR space
BANKSEL   ADRESL       ;
MOVF      ADRESL,W     ;Read lower 8 bits
MOVWF     RESULTLO     ;Store in GPR space
```

19.2 Comparator Control

Each comparator has two control registers: CMxCON0 and CMxCON1.

The CMxCON0 register (see Register 19-1) contains control and status bits for the following:

- Enable
- Output
- Output polarity
- Zero latency filter
- Speed/power selection
- Hysteresis enable
- Output synchronization

The CMxCON1 register (see Register 19-2) contains control bits for the following:

- Interrupt enable
- Interrupt edge polarity
- Positive input channel selection
- Negative input channel selection

19.2.1 COMPARATOR ENABLE

Setting the ON bit of the CMxCON0 register enables the comparator for operation. Clearing the ON bit disables the comparator, resulting in minimum current consumption.

19.2.2 COMPARATOR OUTPUT SELECTION

The output of the comparator can be monitored by reading either the OUT bit of the CMxCON0 register or the MCxOUT bit of the CMOUT register. In order to make the output available for an external connection, the following conditions must be true:

- Desired pin PPS control
- Corresponding TRISx bit must be cleared
- ON bit of the CMxCON0 register must be set

Note 1: The internal output of the comparator is latched with each instruction cycle. Unless otherwise specified, external outputs are not latched.

19.2.3 COMPARATOR OUTPUT POLARITY

Inverting the output of the comparator is functionally equivalent to swapping the comparator inputs. The polarity of the comparator output can be inverted by setting the POL bit of the CMxCON0 register. Clearing the POL bit results in a non-inverted output.

Table 19-2 shows the output state versus input conditions, including polarity control.

TABLE 19-2: COMPARATOR OUTPUT STATE vs. INPUT CONDITIONS

Input Condition	CxPOL	CxOUT
CxVN > CxVP	0	0
CxVN < CxVP	0	1
CxVN > CxVP	1	1
CxVN < CxVP	1	0

22.3 Timer1 Prescaler

Timer1 has four prescaler options, allowing 1, 2, 4 or 8 divisions of the clock input. The CKPS<1:0> bits of the T1CON register control the prescale counter. The prescale counter is not directly readable or writable; however, the prescaler counter is cleared upon a write to TMR1H or TMR1L.

22.4 Timer1 (Secondary) Oscillator

A dedicated low-power 32.768 kHz oscillator circuit is built in between pins, SOSCI (input) and SOSCO (amplifier output). This internal circuit is to be used in conjunction with an external 32.768 kHz crystal.

The oscillator circuit is enabled by setting the OSCEN bit of the T1CON register. The oscillator will continue to run during Sleep.

Note:	The oscillator requires a start-up and stabilization time before use. Thus, OSCEN should be set and a suitable delay observed prior to using Timer1. A suitable delay similar to the OST delay can be implemented in software by clearing the TMR1IF bit then presetting the TMR1H:TMR1L register pair to FC00h. The TMR1IF flag will be set when 1024 clock cycles have elapsed, thereby indicating that the oscillator is running and reasonably stable.

22.5 Timer1 Operation in Asynchronous Counter Mode

If the control bit, $\overline{\text{SYNC}}$ of the T1CON register, is set, the external clock input is not synchronized. The timer increments asynchronously to the internal phase clocks. If the external clock source is selected then the timer will continue to run during Sleep and can generate an interrupt-on-overflow, which will wake-up the processor. However, special precautions in software are needed to read/write the timer (see **Section 22.5.1 "Reading and Writing Timer1 in Asynchronous Counter Mode"**).

Note:	When switching from synchronous to asynchronous operation, it is possible to skip an increment. When switching from asynchronous to synchronous operation, it is possible to produce an additional increment.

22.5.1 READING AND WRITING TIMER1 IN ASYNCHRONOUS COUNTER MODE

Reading TMR1H or TMR1L while the timer is running from an external asynchronous clock will ensure a valid read (taken care of in hardware). However, the user should keep in mind that reading the 16-bit timer in two 8-bit values itself, poses certain problems, since the timer may overflow between the reads.

For writes, it is recommended that the user simply stop the timer and write the desired values. A write contention may occur by writing to the timer registers, while the register is incrementing. This may produce an unpredictable value in the TMR1H:TMR1L register pair.

22.6 Timer1 Gate

Timer1 can be configured to count freely or the count can be enabled and disabled using Timer1 gate circuitry. This is also referred to as Timer1 Gate Enable.

Timer1 gate can also be driven by multiple selectable sources.

22.6.1 TIMER1 GATE ENABLE

The Timer1 Gate Enable mode is enabled by setting the GE bit of the T1GCON register. The polarity of the Timer1 Gate Enable mode is configured using the GPOL bit of the T1GCON register.

When Timer1 Gate Enable mode is enabled, Timer1 will increment on the rising edge of the Timer1 clock source. When Timer1 Gate Enable mode is disabled, no incrementing will occur and Timer1 will hold the current count. See Figure 22-3 for timing details.

TABLE 22-3: TIMER1 GATE ENABLE SELECTIONS

T1CLK	T1GPOL	T1G	Timer1 Operation
↑	0	0	Counts
↑	0	1	Holds Count
↑	1	0	Holds Count
↑	1	1	Counts

23.3 External Reset Sources

In addition to the clock source, the Timer2 also takes in an external Reset source. This external Reset source is selected for Timer2, Timer4 and Timer6, with the T2RST, T4RST and T6RST registers, respectively. This source can control starting and stopping of the timer, as well as resetting the timer, depending on which mode the timer is in. The mode of the timer is controlled by the MODE<4:0> bits of the TMRxHLT register. Edge-Triggered modes require six timer clock periods between external triggers. Level-Triggered modes require the triggering level to be at least three timer clock periods long. External triggers are ignored while in Debug Freeze mode.

23.4 Operating Modes

TABLE 23-1: TIMER2 OPERATING MODES

Mode	MODE<4:0>		Output Operation	Operation	Timer Control		
	<4:3>	<2:0>			Start	Reset	Stop
Free-Running Period	00	000	Period Pulse	Software gate (Figure 23-4)	ON = 1	—	ON = 0
		001		Hardware gate, active-high (Figure 23-5)	ON = 1 and TMRx_ers = 1	—	ON = 0 or TMRx_ers = 0
		010		Hardware gate, active-low	ON = 1 and TMRx_ers = 0	—	ON = 0 or TMRx_ers = 1
		011	Period Pulse with Hardware Reset	Rising or falling edge Reset	ON = 1	TMRx_ers ↕	ON = 0
		100		Rising edge Reset (Figure 23-6)		TMRx_ers ↑	
		101		Falling edge Reset		TMRx_ers ↓	
		110		Low-level Reset		TMRx_ers = 0	ON = 0 or TMRx_ers = 0
		111		High-level Reset (Figure 23-7)		TMRx_ers = 1	ON = 0 or TMRx_ers = 1
One-Shot	01	000	One-Shot	Software start (Figure 23-8)	ON = 1	—	ON = 0 or next clock after TMRx = PRx **(Note 2)**
		001	Edge-Triggered Start **(Note 1)**	Rising edge start (Figure 23-9)	ON = 1 and TMRx_ers ↑	—	
		010		Falling edge start	ON = 1 and TMRx_ers ↓	—	
		011		Any edge start	ON = 1 and TMRx_ers ↕	—	
		100	Edge-Triggered Start and Hardware Reset **(Note 1)**	Rising edge start and rising edge Reset (Figure 23-10)	ON = 1 and TMRx_ers ↑	TMRx_ers ↑	
		101		Falling edge start and falling edge Reset	ON = 1 and TMRx_ers ↓	TMRx_ers ↓	
		110		Rising edge start and low-level Reset (Figure 23-11)	ON = 1 and TMRx_ers ↑	TMRx_ers = 0	
		111		Falling edge start and high-level Reset	ON = 1 and TMRx_ers ↓	TMRx_ers = 1	
Mono-stable	10	000	Reserved				
		001	Edge-Triggered Start **(Note 1)**	Rising edge start (Figure 23-12)	ON = 1 and TMRx_ers ↑	—	ON = 0 or next clock after TMRx = PRx **(Note 3)**
		010		Falling edge start	ON = 1 and TMRx_ers ↓	—	
		011		Any edge start	ON = 1 and TMRx_ers ↕	—	
Reserved		100	Reserved				
Reserved		101	Reserved				
One-shot		110	Level-Triggered Start and Hardware Reset	High-level start and low-level Reset (Figure 23-13)	ON = 1 and TMRx_ers = 1	TMRx_ers = 0	ON = 0 or held in Reset **(Note 2)**
		111		Low-level start and high-level Reset	ON = 1 and TMRx_ers = 0	TMRx_ers = 1	
Reserved	11	xxx	Reserved				

24.3.1 STANDARD PWM OPERATION

The standard PWM function described in this section is available and identical for all CCP modules.

The standard PWM mode generates a Pulse-Width Modulation (PWM) signal on the CCPx pin with up to 10 bits of resolution. The period, duty cycle and resolution are controlled by the following registers:

- T2PR/T4PR/T6PR registers
- T2CON/T4CON/T6CON registers
- CCPRxH:CCPRxL register pair

Figure 24-3 shows a simplified block diagram of PWM operation.

Note 1:	The corresponding TRISx bit must be cleared to enable the PWM output on the CCPx pin.
2:	Clearing the CCPxCON register will relinquish control of the CCPx pin.

24.3.2 SETUP FOR PWM OPERATION

The following steps should be taken when configuring the CCP module for standard PWM operation:

1. Disable the CCPx pin output driver by setting the associated TRISx bit.
2. Select the timer associated with the PWM by setting the CCPTMRS register.
3. Load the associated T2PR/T4PR/T6PR register with the PWM period value.
4. Configure the CCP module for the PWM mode by loading the CCPxCON register with the appropriate values.
5. Load the CCPRxH:CCPRxL register pair with the PWM duty cycle value.
6. Configure and start the timer selected in Step 2:
   - Clear the timer interrupt flag bit of the PIRx register. See Note below.
   - Configure the CKPSx bits of the TxCON register with the timer prescale value.
   - Enable the timer by setting the ON bit of the TxCON register.
7. Enable PWM output pin:
   - Wait until the timer overflows and the timer interrupt bit of the PIRx register is set. See Note below.
   - Enable the CCPx pin output driver by clearing the associated TRISx bit.

Note:	In order to send a complete duty cycle and period on the first PWM output, the above steps must be included in the setup sequence. If it is not critical to start with a complete PWM signal on the first output, then Step 6 may be ignored.

The maximum PWM resolution is ten bits when T2PR is 255. The resolution is a function of the T2PR register value as shown by Equation 25-4.

Note: If the pulse-width value is greater than the period the assigned PWM pin(s) will remain unchanged.

EQUATION 25-4: PWM RESOLUTION

$$Resolution = \frac{log[4(T2PR + 1)]}{log(2)} \ bits$$

TABLE 25-1: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (FOSC = 20 MHz)

PWM Frequency	0.31 kHz	4.88 kHz	19.53 kHz	78.12 kHz	156.3 kHz	208.3 kHz
Timer Prescale	64	4	1	1	1	1
T2PR Value	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
Maximum Resolution (bits)	10	10	10	8	7	6.6

TABLE 25-2: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS (FOSC = 8 MHz)

PWM Frequency	0.31 kHz	4.90 kHz	19.61 kHz	76.92 kHz	153.85 kHz	200.0 kHz
Timer Prescale	64	4	1	1	1	1
T2PR Value	0x65	0x65	0x65	0x19	0x0C	0x09
Maximum Resolution (bits)	8	8	8	6	5	5

25.7 Operation in Sleep Mode

In Sleep mode, the TMR2 register will not increment and the state of the module will not change. If the PWMx pin is driving a value, it will continue to drive that value. When the device wakes up, TMR2 will continue from its previous state.

25.8 Changes in System Clock Frequency

The PWM frequency is derived from the system clock frequency (FOSC). Any changes in the system clock frequency will result in changes to the PWM frequency. Refer to **Section 5.0 "Oscillator Module (with Fail-Safe Clock Monitor)"** for additional details.

25.9 Effects of Reset

Any Reset will force all ports to Input mode and the PWM registers to their Reset states.

REGISTER 26-2: PWMxINTE: PWMx INTERRUPT ENABLE 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
—	—	—	—	OFIE	PHIE	DCIE	PRIE
bit 7							bit 0

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

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

bit 3 **OFIE**: Offset Interrupt Enable bit
1 = Interrupts CPU on offset match
0 = Does not interrupt CPU on offset match

bit 2 **PHIE:** Phase Interrupt Enable bit
1 = Interrupts CPU on phase match
0 = Does not interrupt CPU on phase match

bit 1 **DCIE:** Duty Cycle Interrupt Enable bit
1 = Interrupts CPU on duty cycle match
0 = Does not interrupt CPU on duty cycle match

bit 0 **PRIE:** Period Interrupt Enable bit
1 = Interrupts CPU on period match
0 = Does not interrupt CPU on period match

REGISTER 29-4: OPAxPCHS: OP AMP x POSITIVE CHANNEL SOURCE SELECT 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
—	—	—	—	PCH<3:0>			
bit 7							bit 0

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

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

bit 3-0 **PCH<3:0>:** Op Amp Non-Inverting Input Channel Selection bits

1111 = Reserved; do not use
•
•
•
1010 = Reserved; do not use
1001 = Programmable Ramp Generator PRG2_out(1)
1000 = Programmable Ramp Generator PRG1_out
0111 = Reserved. Do not use.
0110 = FVR_Buffer2
0101 = DAC4_out(1)
0100 = DAC3_out
0011 = DAC2_out(1)
0010 = DAC1_out
0001 = OPAxIN1+ pin(1)
0000 = OPAxIN0+ pin

Note 1: PIC16(L)F1768/9 only

TABLE 29-2: SUMMARY OF REGISTERS ASSOCIATED WITH OP AMPS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELB(2)	ANSB<7:6>		ANSB<5:4>		—	—	—	—	143
ANSELC(2)	ANSC<7:6>(2)		—	—	ANSC<3:2>		ANSC<1:0>		148
DACxCON0	EN	FM	OE1	—	PSS<1:0>		NSS<1:0>		188
DACxREF	---	---	---	REF<4:0>					189
DACxREFL(2)	REF<7:0>								194
DACxREFH(2)	REF<15:8>								194
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		169
OPAxCON	EN	—	—	UG	—	ORPOL	ORM<1:0>		348
OPAxNCHS	—	—	—	—	NCH<3:0>				350
OPAxPCHS	—	—	—	—	PCH<3:0>				351
OPAxORS	—	—	—	ORS<4:0>					349
TRISB(2)	TRISB<7:6>		TRISB<5:4>		—	—	—	—	142
TRISC(2)	TRISC<7:6>(2)		TRISC<5:4>		TRISC<3:2>		TRISC<1:0>		147

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by op amps.

Note 1: Unimplemented, read as '1'.

2: PIC16(L)F1768/9 only

32.0 MASTER SYNCHRONOUS SERIAL PORT (MSSP) MODULE

32.1 MSSP Module Overview

The Master Synchronous Serial Port (MSSP) module is a serial interface useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be serial EEPROMs, shift registers, display drivers, A/D converters, etc. The MSSP module can operate in one of two modes:

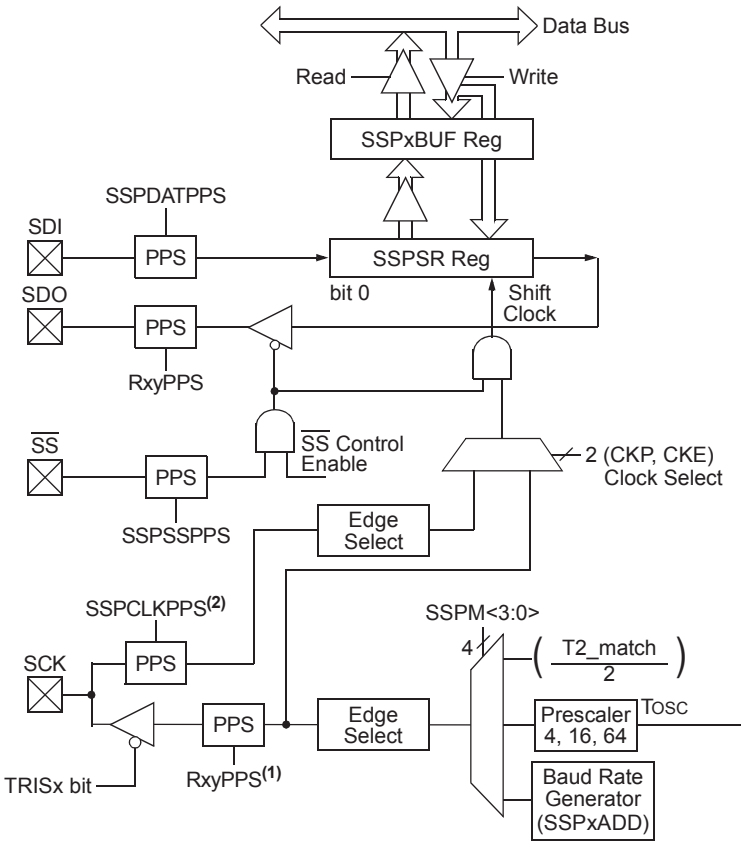
- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I^2C)

The SPI interface supports the following modes and features:

- Master mode
- Slave mode
- Clock Parity
- Slave Select Synchronization (Slave mode only)
- Daisy-Chain Connection of Slave Devices

Figure 32-1 is a block diagram of the SPI interface module.

FIGURE 32-1: MSSP BLOCK DIAGRAM (SPI MODE)



Note 1: Output selection for Master mode.
2: Input selection for Slave and Master modes.

FIGURE 32-29: I²C MASTER MODE WAVEFORM (RECEPTION, 7-BIT ADDRESS)

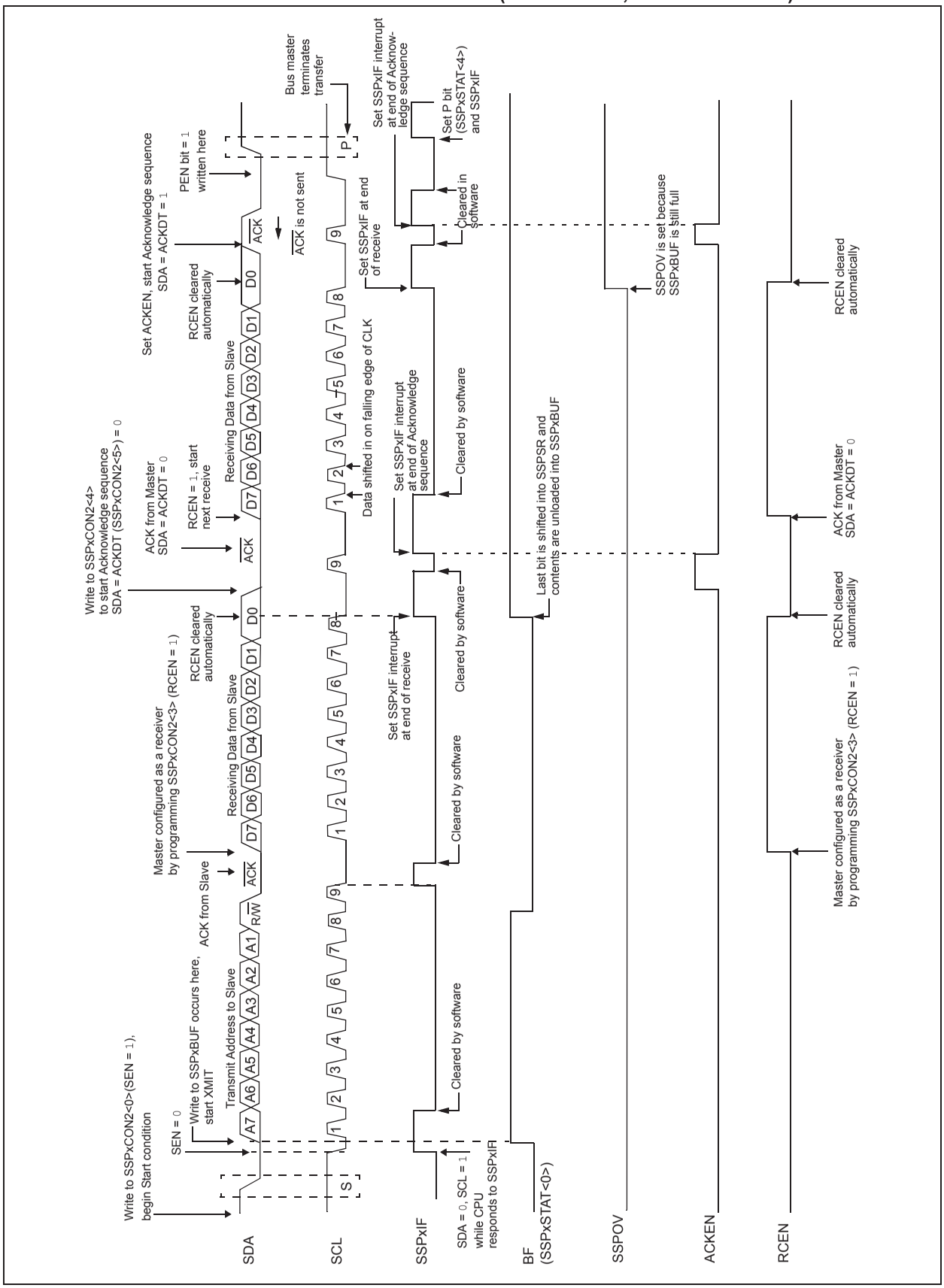


TABLE 33-9: SUMMARY OF REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE TRANSMISSION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELA	—	—	—	ANSA4	—	ANSA<2:0>			137
ANSELB(1)	ANSB<7:4>				—	—	—	—	143
ANSELC	ANSC<7:6>(1)		—	—	ANSC<3:0>				148
BAUD1CON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	442
CKPPS	—	—	—	CKPPS<4:0>					154, 156
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	101
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	102
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	105
RC1STA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	441
RXPPS	—	—	—	RXPPS<4:0>					154, 156
RxyPPS	—	—	—	RxyPPS<4:0>					154
TRISA	—	—	TRISA<5:4>		—(2)	TRISA<2:0>			136
TRISB(1)	TRISB<7:4>				—	—	—	—	142
TRISC	TRISC<7:6>(1)		TRISC<5:0>						147
TX1REG	EUSART Transmit Data Register								433*
TX1STA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	440

Legend: — = unimplemented location, read as '0'. Shaded cells are not used for synchronous slave transmission.

* Page provides register information.

Note 1: PIC16(L)F1768/9 only.

2: Unimplemented, read as '1'.

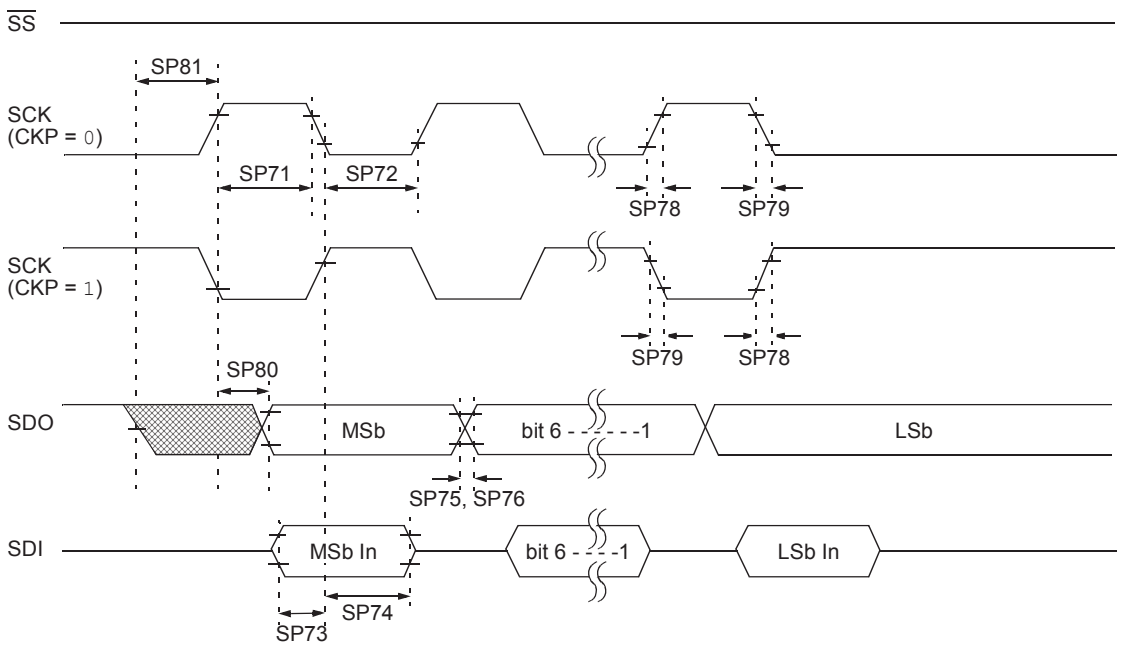
TABLE 36-12: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS

Standard Operating Conditions (unless otherwise stated) Operating Temperature -40°C ≤ TA ≤ +125°C								
Param No.	**Sym.**	**Characteristic**		**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
40*	TT0H	T0CKI High Pulse Width	No Prescaler	0.5 TCY + 20	—	—	ns	
			With Prescaler	10	—	—	ns	
41*	TT0L	T0CKI Low Pulse Width	No Prescaler	0.5 TCY + 20	—	—	ns	
			With Prescaler	10	—	—	ns	
42*	TT0P	T0CKI Period		Greater of: 20 or TCY + 40 / N	—	—	ns	N = prescale value
45*	TT1H	T1CKI High Time	Synchronous, No Prescaler	0.5 TCY + 20	—	—	ns	
			Synchronous, with Prescaler	15	—	—	ns	
			Asynchronous	30	—	—	ns	
46*	TT1L	T1CKI Low Time	Synchronous, No Prescaler	0.5 TCY + 20	—	—	ns	
			Synchronous, with Prescaler	15	—	—	ns	
			Asynchronous	30	—	—	ns	
47*	TT1P	T1CKI Input Period	Synchronous	Greater of: 30 or TCY + 40 / N	—	—	ns	N = prescale value
			Asynchronous	60	—	—	ns	
48	FT1	Secondary Oscillator Input Frequency Range (oscillator enabled by setting bit, T1OSCEN)		32.4	32.768	33.1	kHz	
49*	TCKEZTMR1	Delay from External Clock Edge to Timer Increment		2 TOSC	—	7 TOSC	—	Timers in Sync mode

* These parameters are characterized but not tested.

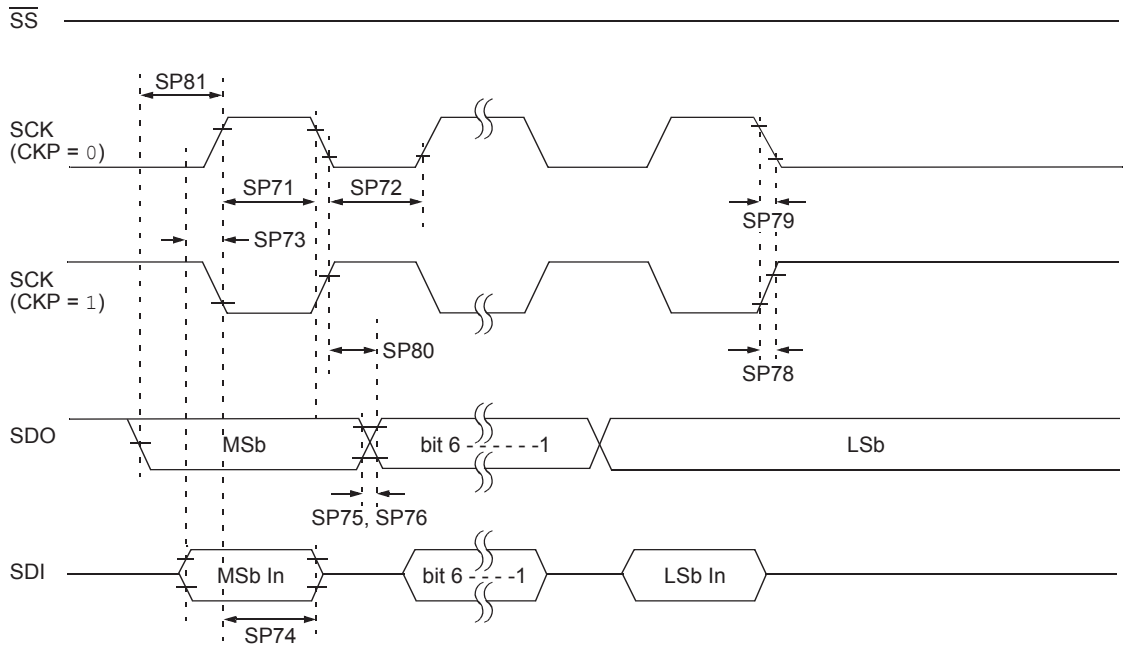
† Data in "Typ" column is at 3.0V, +25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 36-17: SPI MASTER MODE TIMING (CKE = 0, SMP = 0)



Note: Refer to Figure 36-4 for load conditions.

FIGURE 36-18: SPI MASTER MODE TIMING (CKE = 1, SMP = 1)



Note: Refer to Figure 36-4 for load conditions.

Note: Unless otherwise noted, VIN = 5V, FOSC = 300 kHz, CIN = 0.1 µF, TA = 25°C.

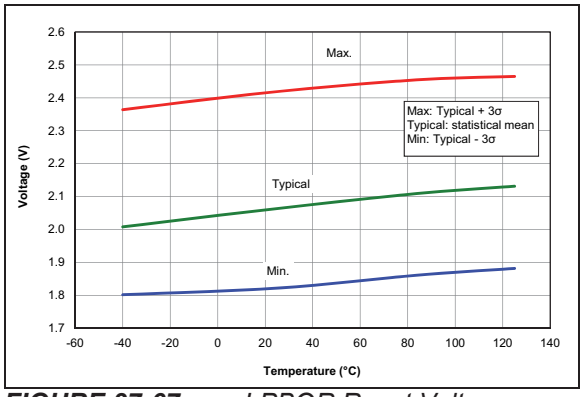


FIGURE 37-67: *LPBOR Reset Voltage.*

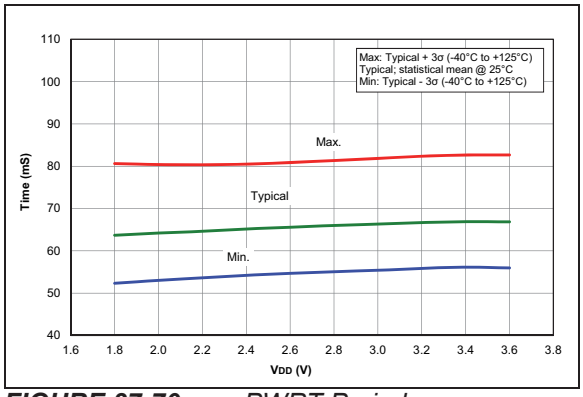


FIGURE 37-70: *PWRT Period, PIC16LF1764/5/8/9 Only.*

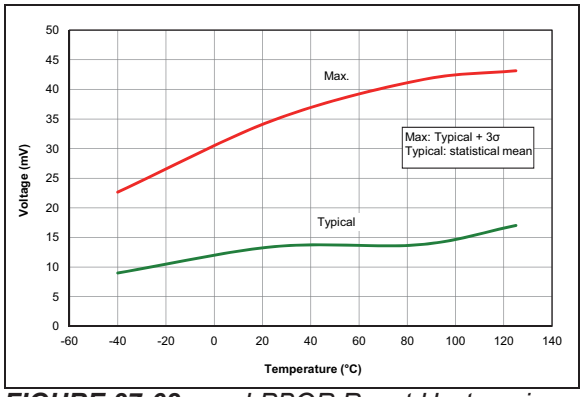


FIGURE 37-68: *LPBOR Reset Hysteresis.*

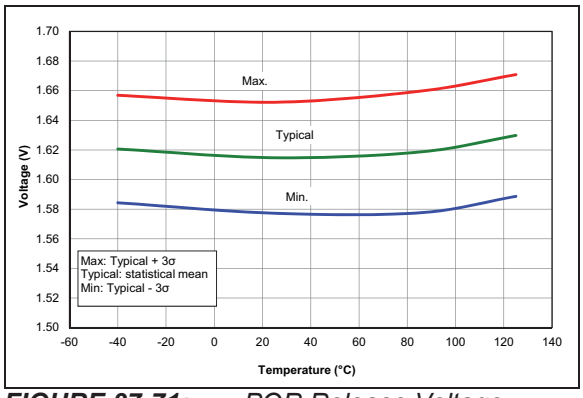


FIGURE 37-71: *POR Release Voltage.*

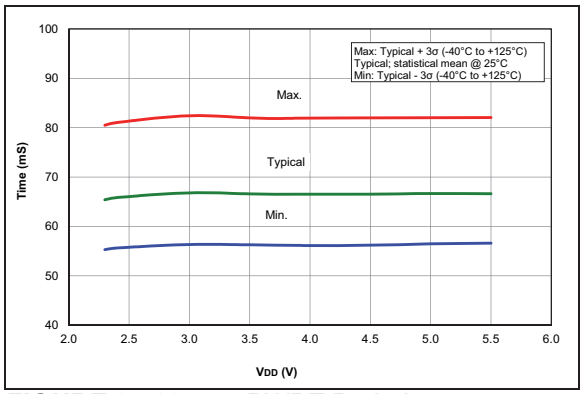


FIGURE 37-69: *PWRT Period, PIC16F1764/5/8/9 Only.*

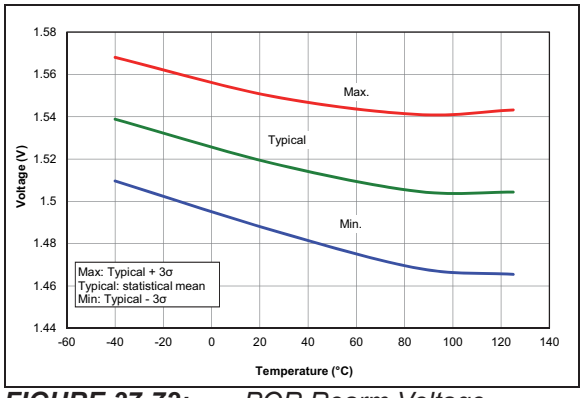


FIGURE 37-72: *POR Rearm Voltage, NP Mode (VREGPM = 0), PIC16F1764/5/8/9 Only.*