Welcome to E-XFL.COM

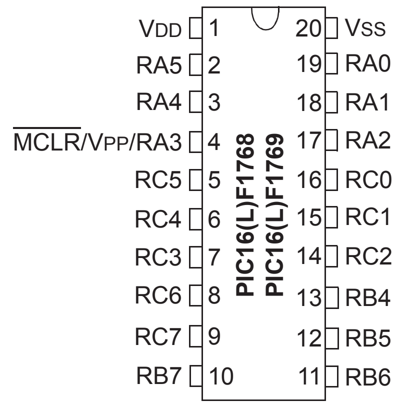
What is "Embedded - Microcontrollers"?

"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"

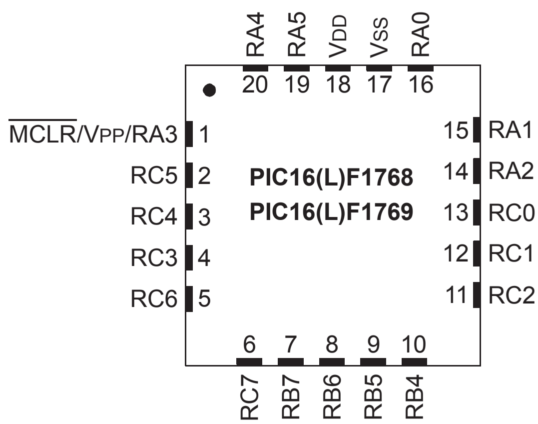
Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	32MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	18
Program Memory Size	7KB (4K x 14)
Program Memory Type	FLASH
EEPROM Size	128 x 8
RAM Size	512 x 8
Voltage - Supply (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/pic16f1768t-i-ss

FIGURE 3: 20-PIN PDIP, SOIC, SSOP



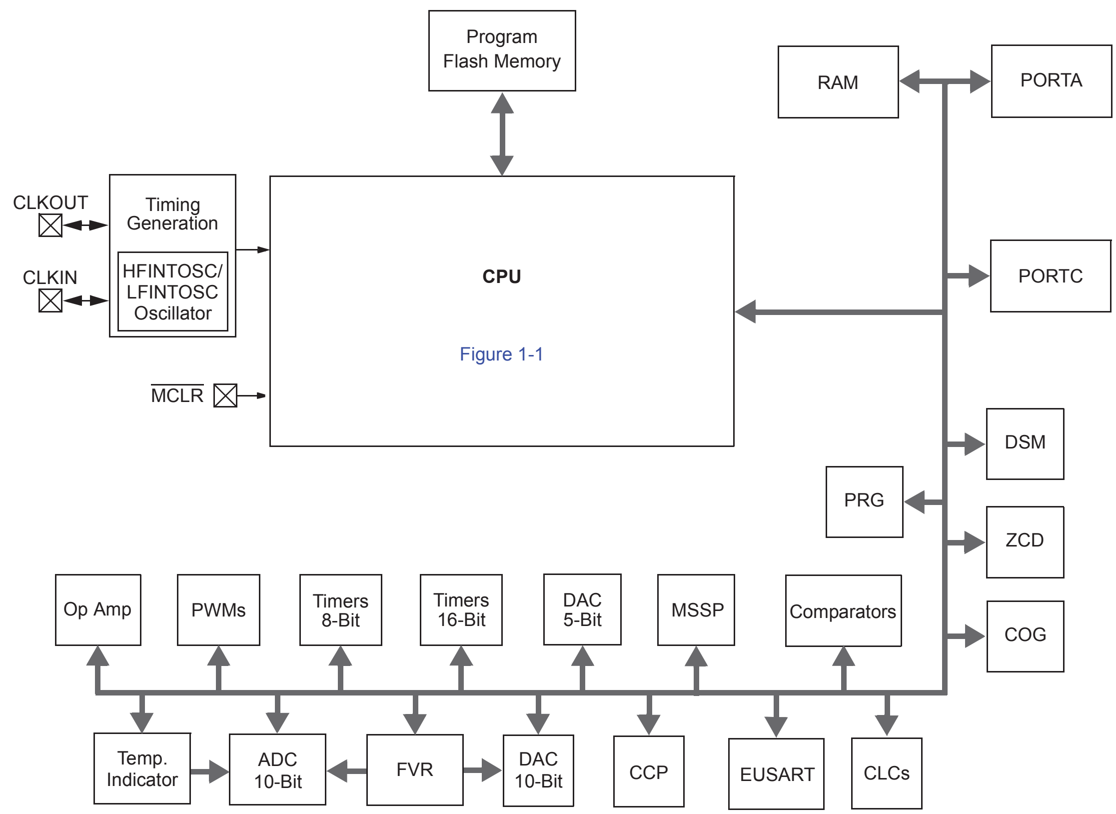
Note: See Table 4 for location of all peripheral functions.

FIGURE 4: 20-PIN QFN (4x4)



Note: See Table 4 for location of all peripheral functions.

FIGURE 1-1: PIC16(L)F1764/5 BLOCK DIAGRAM



Note 1: See applicable chapters for more information on peripherals.

1.2 Peripheral Connection Matrix

Input selection multiplexers on many of the peripherals enable selecting the output of another peripheral, such that the signal path is contained entirely within the device. Although the peripheral output can also be routed to a pin with the PPS selection feature, it is not necessary to do so. Table 1-4 shows all the possible inter-peripheral signal connections. Please refer to the corresponding peripheral section to obtain the multiplexer selection codes for the desired connection.

TABLE 1-4: PERIPHERAL CONNECTION MATRIX

	Peripheral Input																								
Peripheral Output	**ADC Trigger**	**COG Clock**	**COG Rising/Falling**	**COG Shutdown**	**10-Bit DAC**	**5-Bit DAC**	**PRG Analog Input**	**PRG Rising/Falling**	**Comparator +**	**Comparator -**	**CLC**	**DSM CH**	**DSM CL**	**DSM Mod**	**Op Amp +**	**Op Amp -**	**Op Amp Override**	**10-Bit PWM**	**16-Bit PWM**	**CCP Capture**	**CCP Clock**	**Timer2/4/6 Clock**	**Timer2/4/6 Reset**	**Timer1/3/5 Gate**	**Timer0 Clock**
FVR					•	•	•		•	•					•	•									
ZCD											•						•					•			
PRG									•						•	•									
10-Bit DAC							•		•						•	•									
5-Bit DAC							•		•						•	•									
CCP	•		•					•			•	•	•	•			•						•		
Comparator (sync)	•							•			•						•			•			•	•	
Comparator (async)			•	•										•											
CLC	•		•	•							•	•	•	•			•			•		•	•		
DSM																									
COG																	•								
EUSART TX/CK											•			•											
EUSART DT											•			•											
MSSP SCK/SCL											•			•											
MSSP SDO/SDA											•			•											
Op Amp							•																		
10-Bit PWM	•		•					•			•	•	•	•			•						•		
16-Bit PWM	•		•					•			•	•	•	•			•						•		
Timer0 Overflow	•										•													•	
Timer2 = T2PR				•							•							•			•		•		
Timer4 = T4PR				•							•							•			•		•		
Timer6 = T6PR				•							•							•			•		•		
Timer2 Postscale	•			•							•							•			•		•		
Timer4 Postscale	•			•							•							•			•		•		
Timer6 Postscale	•			•							•							•			•		•		
Timer1 Overflow	•										•							•			•				
Timer3 Overflow	•										•							•			•				
Timer5 Overflow	•										•							•			•				
SOSC																			•			•			
FOSC/4		•																				•			
FOSC		•									•	•	•						•			•			
HFINTOSC		•									•	•	•						•			•			
LFINTOSC											•								•			•			
MFINTOSC																						•			
IOCIF											•									•	•				
PPS Input Pin			•	•				•				•	•	•						•	•	•	•	•	•

5.3 Clock Switching

The system clock source can be switched between external and internal clock sources via software using the System Clock Select (SCS) bits of the OSCCON register. The following clock sources can be selected using the SCS<1:0> bits:

- Default system oscillator determined by the FOSCx bits in the Configuration Words
- Timer1 32 kHz crystal oscillator
- Internal Oscillator Block (INTOSC)

5.3.1 SYSTEM CLOCK SELECT (SCS) BITS

The System Clock Select (SCS<1:0>) bits of the OSCCON register select the system clock source that is used for the CPU and peripherals.

- When SCS<1:0> = `00`, the system clock source is determined by the value of the FOSC<2:0> bits in the Configuration Words.
- When SCS<1:0> = `01`, the system clock source is the secondary oscillator.
- When SCS<1:0> = `1x`, the system clock source is chosen by the internal oscillator frequency selected by the IRCF<3:0> bits of the OSCCON register. After a Reset, the SCSx bits of the OSCCON register are always cleared.

Note:	Any automatic clock switch, which may occur from Two-Speed Start-up or Fail-Safe Clock Monitor, does not update the SCS bits of the OSCCON register. The user can monitor the OSTS bit of the OSCSTAT register to determine the current system clock source.

When switching between clock sources, a delay is required to allow the new clock to stabilize. These oscillator delays are shown in Table 5-1.

5.3.2 OSCILLATOR START-UP TIMER STATUS (OSTS) BIT

The Oscillator Start-up Timer Status (OSTS) bit of the OSCSTAT register indicates whether the system clock is running from the external clock source, as defined by the FOSC<2:0> bits in the Configuration Words, or from the internal clock source. In particular, OSTS indicates that the Oscillator Start-up Timer (OST) has timed out for LP, XT or HS modes. The OST does not reflect the status of the secondary oscillator.

5.3.3 SECONDARY OSCILLATOR

The secondary oscillator is a separate crystal oscillator associated with the Timer1 peripheral. It is optimized for timekeeping operations with a 32.768 kHz crystal connected between the SOSCO and SOSCI device pins.

The secondary oscillator is enabled using the OSCEN control bit in the T1CON register. See **Section 22.0 “Timer1/3/5 Module with Gate Control”** for more information about the Timer1 peripheral.

5.3.4 SECONDARY OSCILLATOR READY (SOSCR) BIT

The user must ensure that the secondary oscillator is ready to be used before it is selected as a system clock source. The Secondary Oscillator Ready (SOSCR) bit of the OSCSTAT register indicates whether the secondary oscillator is ready to be used. After the SOSCR bit is set, the SCSx bits can be configured to select the secondary oscillator.

5.3.5 CLOCK SWITCH BEFORE SLEEP

When a clock switch from an old clock to a new clock is requested just prior to entering Sleep mode, it is necessary to confirm that the switch is complete before the sleep instruction is executed. Failure to do so may result in an incomplete switch and consequential loss of the system clock altogether. Clock switching is confirmed by monitoring the clock status bits in the OSCSTAT register. Switch confirmation can be accomplished by sensing that the ready bit for the new clock is set or the ready bit for the old clock is cleared. For example, when switching between the internal oscillator with the PLL and the internal oscillator without the PLL, monitor the PLLR bit. When PLLR is set, the switch to 32 MHz operation is complete. Conversely, when PLLR is cleared, the switch from 32 MHz operation to the selected internal clock is complete.

6.3 Register Definitions: BOR Control

REGISTER 6-1: BORCON: BROWN-OUT RESET CONTROL REGISTER

R/W-1/u	R/W-0/u	U-0	U-0	U-0	U-0	U-0	R-q/u
SBOREN	BORFS(1)	—	—	—	—	—	BORRDY
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	q = Value depends on condition
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 **SBOREN:** Software Brown-out Reset Enable bit

If BOREN <1:0> in Configuration Words ≠ 01:
SBOREN is read/write, but has no effect on the BOR.

If BOREN <1:0> in Configuration Words = 01:
1 = BOR is enabled
0 = BOR is disabled

bit 6 **BORFS:** Brown-out Reset Fast Start bit(1)

If BOREN<1:0> = 11 (always on) or BOREN<1:0> = 00 (always off):
BORFS is Read/Write, but has no effect.

If BOREN <1:0> = 10 (disabled in Sleep) or BOREN<1:0> = 01 (under software control):
1 = Band gap is forced on always (covers Sleep/wake-up/operating cases)
0 = Band gap operates normally and may turn off

bit 5-1 **Unimplemented:** Read as ‘0’

bit 0 **BORRDY:** Brown-out Reset Circuit Ready Status bit
1 = The Brown-out Reset circuit is active
0 = The Brown-out Reset circuit is inactive

Note 1: The BOREN<1:0> bits are located in the Configuration Words.

TABLE 10-1: FLASH MEMORY ORGANIZATION BY DEVICE

Device	Row Erase (words)	Write Latches (words)
PIC16(L)F1764	32	32
PIC16(L)F1765		
PIC16(L)F1768		
PIC16(L)F1769		

10.2.1 READING THE FLASH PROGRAM MEMORY

To read a program memory location, the user must:

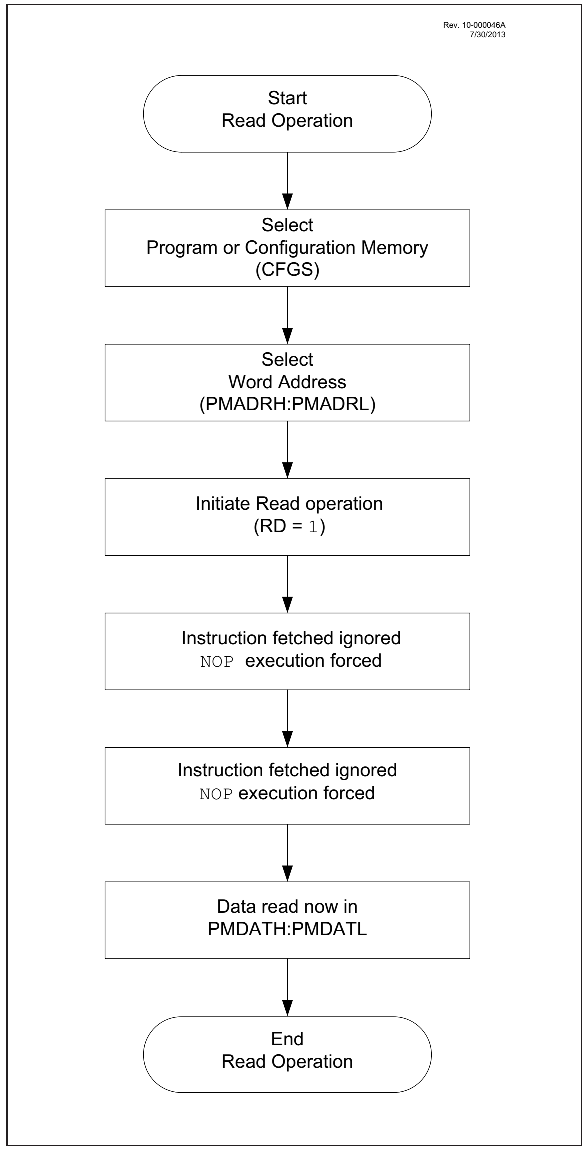
1. Write the desired address to the PMADRH:PMADRL register pair.
2. Clear the CFGS bit of the PMCON1 register.
3. Then, set control bit, RD, of the PMCON1 register.

Once the read control bit is set, the Program Flash Memory controller will use the second instruction cycle to read the data. This causes the second instruction, immediately following the "`BSF PMCON1,RD`" instruction, to be ignored. The data is available in the very next cycle, in the PMDATH:PMDATL register pair; therefore, it can be read as two bytes in the following instructions.

The PMDATH:PMDATL register pair will hold this value until another read or until it is written to by the user.

> **Note:** The two instructions following a program memory read are required to be `NOP`s. This prevents the user from executing a 2-cycle instruction on the next instruction after the RD bit is set.

FIGURE 10-1: FLASH PROGRAM MEMORY READ FLOWCHART

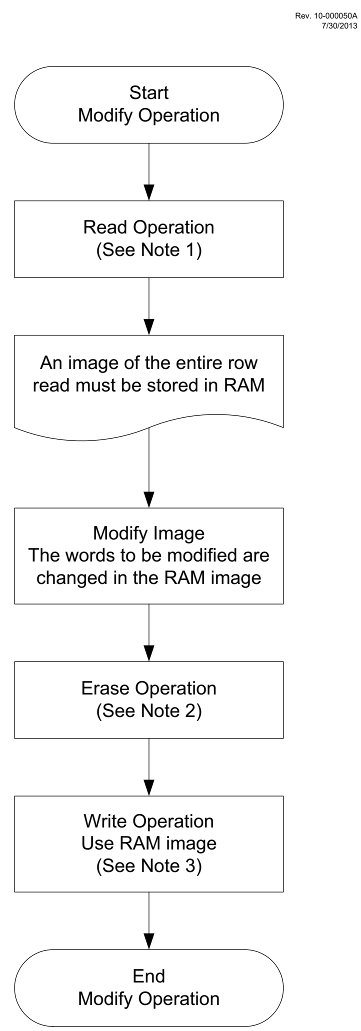


10.3 Modifying Flash Program Memory

When modifying existing data in a program memory row, and data within that row must be preserved, it must first be read and saved in a RAM image. Program memory is modified using the following steps:

1. Load the starting address of the row to be modified.
2. Read the existing data from the row into a RAM image.
3. Modify the RAM image to contain the new data to be written into program memory.
4. Load the starting address of the row to be rewritten.
5. Erase the program memory row.
6. Load the write latches with data from the RAM image.
7. Initiate a programming operation.

FIGURE 10-7: FLASH PROGRAM MEMORY MODIFY FLOWCHART



Note 1: See Figure 10-1.
2: See Figure 10-4.
3: See Figure 10-6.

REGISTER 11-7: SLRCONA: PORTA SLEW RATE CONTROL REGISTER

U-0	U-0	R/W-1/1	R/W-1/1	U-0	R/W-1/1	R/W-1/1	R/W-1/1
—	—	SLRA<5:4>		—	SLRA<2:0>		
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 7-6 **Unimplemented:** Read as '0'

bit 5-4 **SLRA<5:4>:** PORTA Slew Rate Enable bits
For RA<5:4> Pins:
1 = Port pin slew rate is limited
0 = Port pin slews at maximum rate

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

bit 2-0 **SLRA<2:0>:** PORTA Slew Rate Enable bits
For RA<2:0> Pins:
1 = Port pin slew rate is limited
0 = Port pin slews at maximum rate

REGISTER 11-8: INLVLA: PORTA INPUT LEVEL CONTROL REGISTER

U-0	U-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
—	—	INLVLA<5:0>					
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 7-6 **Unimplemented:** Read as '0'

bit 5-0 **INLVLA<5:0>:** PORTA Input Level Select bits
For RA<5:0> Pins:
1 = Port pin digital input operates with ST thresholds
0 = Port pin digital input operates with TTL thresholds

TABLE 12-2: AVAILABLE PORTS FOR OUTPUT BY PERIPHERAL[(2)]

RxyPPS<4:0>	Output Signal	PIC16(L)F1768/9			PIC16(L)F1764/5	
		PORTA	PORTB	PORTC	PORTA	PORTC
1111x	Reserved	—	—	—	—	—
11101	MD2_out	●	●	●	—	—
11100	MD1_out	●	●	●	●	●
11011	sync_C4OUT	●	●	●	—	—
11010	sync_C3OUT	●	●	●	—	—
11001	sync_C2OUT	●	●	●	●	●
11000	sync_C1OUT	●	●	●	●	●
10111	DT[(1)]	●	●	●	●	●
10110	TX/CK[(1)]	●	●	●	●	●
10101	Reserved	—	—	—	—	—
10100	SDO	●	●	●	●	●
10011	SDA	●	●	●	●	●
10010	SCK/SCL[(1)]	●	●	●	●	●
10001	PWM6_out	●	●	●	—	—
10000	PWM5_out	●	●	●	●	●
01111	PWM4_out	●	●	●	—	—
01110	PWM3_out	●	●	●	●	●
01101	CCP2_out	●	●	●	●	●
01100	CCP1_out	●	●	●	●	●
01011	COG2D[(1)]	●	●	●	—	—
01010	COG2C[(1)]	●	●	●	—	—
01001	COG2B[(1)]	●	●	●	—	—
01000	COG2A[(1)]	●	●	●	—	—
00111	COG1D[(1)]	●	●	●	●	●
00110	COG1C[(1)]	●	●	●	●	●
00101	COG1B[(1)]	●	●	●	●	●
00100	COG1A[(1)]	●	●	●	●	●
00011	LC3_out	●	●	●	●	●
00010	LC2_out	●	●	●	●	●
00001	LC1_out	●	●	●	●	●
00000	LATxy	●	●	●	●	●

Note 1: TRISx control is overridden by the peripheral as required.
2: Unsupported peripherals will output a 0.

REGISTER 16-2: ADCON1: ADC CONTROL REGISTER 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
ADFM	ADCS<2:0>			—	ADNREF	ADPREF<1:0>	
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 **ADFM:** ADC Result Format Select bit
- 1 = Right justified; six Most Significant bits of ADRESH are set to ‘0’ when the conversion result is loaded
- 0 = Left justified; six Least Significant bits of ADRESL are set to ‘0’ when the conversion result is loaded

bit 6-4 **ADCS<2:0>:** ADC Conversion Clock Select bits
- 111 = FRC (clock supplied from an internal RC oscillator)
- 110 = FOSC/64
- 101 = FOSC/16
- 100 = FOSC/4
- 011 = FRC (clock supplied from an internal RC oscillator)
- 010 = FOSC/32
- 001 = FOSC/8
- 000 = FOSC/2

bit 3 **Unimplemented:** Read as ‘0’

bit 2 **ADNREF:** ADC Negative Voltage Reference Configuration bit
- 1 = VREF- is connected to external VREF- pin
- 0 = VREF- is connected to VSS

bit 1-0 **ADPREF<1:0>:** ADC Positive Voltage Reference Configuration bits
- 11 = VREF+ is connected to the internal Fixed Voltage Reference (FVR) module(1)
- 10 = VREF+ is connected to the external VREF+ pin(1)
- 01 = Reserved
- 00 = VREF+ is connected to VDD

Note 1: When selecting the VREF+ pin as the source of the positive reference, be aware that a minimum voltage specification exists. See Table 36-16 for details.

17.0 5-BIT DIGITAL-TO-ANALOG CONVERTER (DAC) MODULE

The Digital-to-Analog Converter supplies a variable voltage reference, ratiometric with the input source, with 32 selectable output levels.

The input of the DAC can be connected to:

- External VREF pins
- VDD supply voltage
- FVR (Fixed Voltage Reference)

The output of the DAC can be configured to supply a reference voltage to the following:

- Comparator positive input
- Operational amplifier inverting and non-inverting inputs
- ADC input channel
- DACxOUT1 pin

TABLE 17-1: AVAILABLE 5-BIT DACs

Device	D3	D4
PIC16(L)F1764	●	
PIC16(L)F1765	●	
PIC16(L)F1768	●	●
PIC16(L)F1769	●	●

The Digital-to-Analog Converter (DAC) is enabled by setting the EN bit of the DACxCON0 register.

17.1 Output Voltage Selection

The DAC has 32 voltage level ranges. The 32 levels are set with the REF<4:0> bits of the DACxREF register.

The DAC output voltage is determined by Equation 17-1.

EQUATION 17-1: DAC OUTPUT VOLTAGE

IF DACxEN = 1:

$$V_{OUT} = \left((V_{SOURCE+} - V_{SOURCE-}) \times \frac{DACxR[4:0]}{2^5}\right) + V_{SOURCE-}$$

$$V_{SOURCE+} = V_{DD},\ V_{REF},\ \text{or FVR Buffer2}$$

$$V_{SOURCE-} = V_{SS}$$

17.2 Ratiometric Output Level

The DAC output value is derived using a resistor ladder with each end of the ladder tied to a positive and negative voltage reference input source. If the voltage of either input source fluctuates, a similar fluctuation will result in the DAC output value.

The value of the individual resistors within the ladder can be found in Table 36-20.

17.3 DAC Voltage Reference Output

The DAC voltage can be output to the DACxOUT1 pin by setting the OE1 bit of the DACxCON0 register. Selecting the DAC voltage for output on the DACxOUT1 pin automatically overrides the digital output buffer and digital input threshold detector functions of that pin. Reading the DACxOUT1 pin when it has been configured for DAC voltage output will always return a '0'.

Due to the limited current drive capability, a buffer must be used on the DAC voltage output for external connections to the DACxOUT1 pin. Figure 17-2 shows an example buffering technique.

REGISTER 27-3: COGxRIS0: COGx RISING EVENT INPUT SELECTION REGISTER 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	R/W-0/0
RIS7	RIS6	RIS5	RIS4	RIS3	RIS2	RIS1	RIS0
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 **RIS7:** COGx Rising Event Input Source 7 Enable bit
1 = PWM3 output is enabled as a rising event input
0 = PWM3 output has no effect on the rising event

bit 6 **RIS6:** COGx Rising Event Input Source 6 Enable bit
1 = CCP2 output is enabled as a rising event input
0 = CCP2 output has no effect on the rising event

bit 5 **RIS5:** COGx Rising Event Input Source 5 Enable bit
1 = CCP1 output is enabled as a rising event input
0 = CCP1 output has no effect on the rising event

bit 4 **RIS4:** COGx Rising Event Input Source 4 Enable bit
1 = Comparator 4 output is enabled as a rising event input
0 = Comparator 4 output has no effect on the rising event

bit 3 **RIS3:** COGx Rising Event Input Source 3 Enable bit
1 = Comparator 3 output is enabled as a rising event input
0 = Comparator 3 output has no effect on the rising event

bit 2 **RIS2:** COGx Rising Event Input Source 2 Enable bit
1 = Comparator 2 output is enabled as a rising event input
0 = Comparator 2 output has no effect on the rising event

bit 1 **RIS1:** COGx Rising Event Input Source 1 Enable bit
1 = Comparator 1 output is enabled as a rising event input
0 = Comparator 1 output has no effect on the rising event

bit 0 **RIS0:** COGx Rising Event Input Source 0 Enable bit
1 = Pin selected with COGxINPPS register is enabled as rising event input
0 = Pin selected with COGxINPPS register has no effect on the rising event

FIGURE 31-2: ON-OFF KEYING (OOK) SYNCHRONIZATION

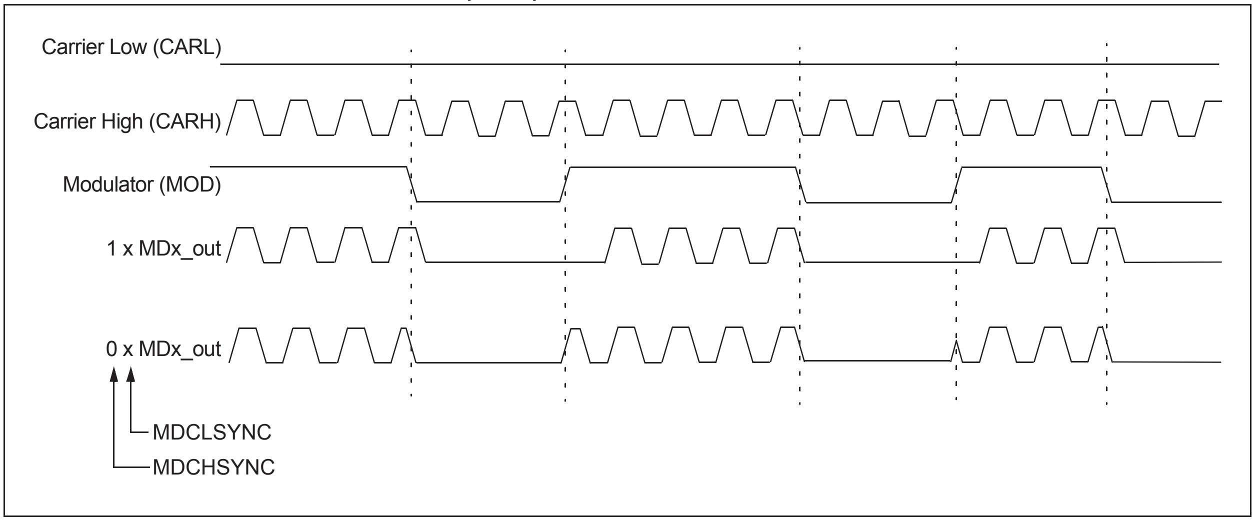


FIGURE 31-3: NO SYNCHRONIZATION (MDCHSYNC = 0, MDCLSYNC = 0)

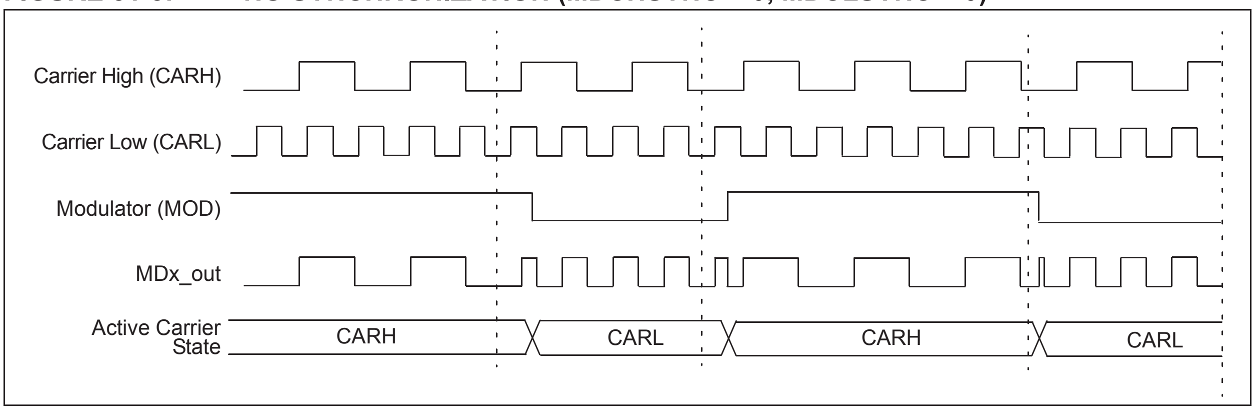
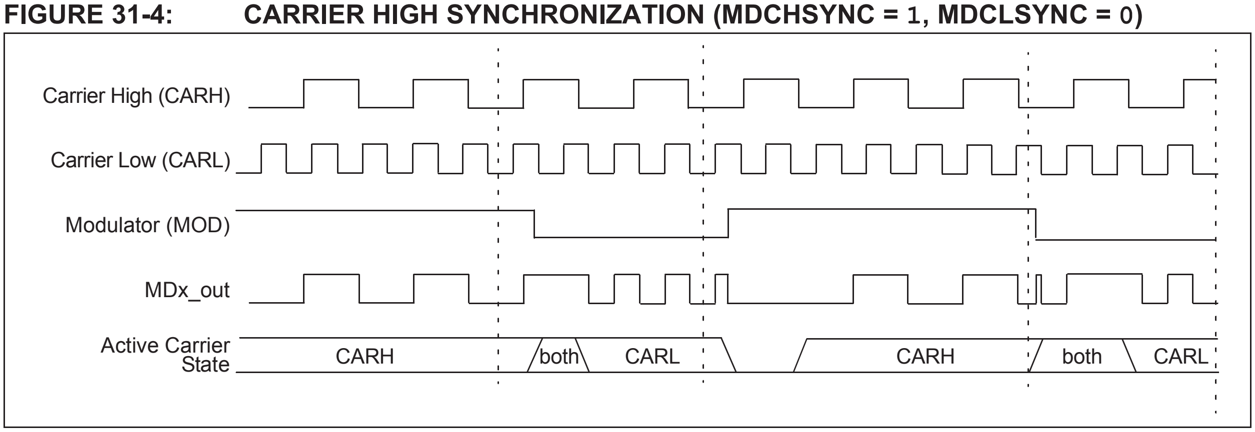


FIGURE 31-4: CARRIER HIGH SYNCHRONIZATION (MDCHSYNC = 1, MDCLSYNC = 0)



REGISTER 31-3: MDxSRC: MODULATION x SOURCE CONTROL REGISTER

U-0	U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	—	MS<4:0>				
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 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **MS<4:0>** Modulation Source Selection bits
See Table 31-3.

TABLE 31-3: MODULATION SOURCE

MS<4:0>	Modulation Source PIC16(L)F1764/5	Modulation Source PIC16(L)F1768/9
11111-10100	Fixed Low	Fixed Low
10011	Fixed Low	sync_C4OUT
10010	Fixed Low	sync_C3OUT
10001	sync_C2OUT	sync_C2OUT
10000	sync_C1OUT	sync_C1OUT
01111	LC3_out	LC3_out
01110	LC2_out	LC2_out
01101	LC1_out	LC1_out
01100	Fixed Low	PWM6_out
01011	PWM5_out	PWM5_out
01010	Fixed Low	PWM4_out
01001	PWM3_out	PWM3_out
01000	Fixed low	CCP2_out
00111	CCP1_out	CCP1_out
00110	SDO_out	SDO_out
00101	Fixed Low	COG2A
00100	DT	DT
00011	TX_out	TX_out
00010	COG1A	COG1A
00001	MDxBIT	MDxBIT
00000	MDxMODPPS Pin Selection	MDxMODPPS Pin Selection

32.5.6 CLOCK STRETCHING

Clock stretching occurs when a device on the bus holds the SCL line low, effectively pausing communication. The slave may stretch the clock to allow more time to handle data or prepare a response for the master device. A master device is not concerned with stretching, as anytime it is active on the bus and not transferring data, it is stretching. Any stretching done by a slave is invisible to the master software and handled by the hardware that generates SCL.

The CKP bit of the SSPxCON1 register is used to control stretching in software. Any time the CKP bit is cleared, the module will wait for the SCL line to go low and then hold it. Setting CKP will release SCL and allow more communication.

32.5.6.1 Normal Clock Stretching

Following an $\overline{\text{ACK}}$, if the R/$\overline{\text{W}}$ bit of SSPxSTAT is set and there is a read request, the slave hardware will clear CKP. This allows the slave time to update SSPxBUF with data to transfer to the master. If the SEN bit of SSPxCON2 is set, the slave hardware will always stretch the clock after the $\overline{\text{ACK}}$ sequence. Once the slave is ready, CKP is set by software and communication resumes.

> **Note 1:** The BF bit has no effect on if the clock will be stretched or not. This is different than previous versions of the module that would not stretch the clock, and cleared CKP if SSPxBUF was read before the 9th falling edge of SCL.
>
> **2:** Previous versions of the module did not stretch the clock for a transmission if SSPxBUF was loaded before the 9th falling edge of SCL; it is now always cleared for read requests.

32.5.6.2 10-Bit Addressing Mode

In 10-Bit Addressing mode when the UA bit is set, the clock is always stretched. This is the only time the SCL is stretched without CKP being cleared. SCL is released immediately after a write to SSPxADD.

> **Note:** Previous versions of the module did not stretch the clock if the second address byte did not match.

32.5.6.3 Byte NACKing

When the AHEN bit of SSPxCON3 is set, CKP is cleared by hardware after the eighth falling edge of SCL for a received matching address byte. When the DHEN bit of SSPxCON3 is set, CKP is cleared after the eighth falling edge of SCL for received data.

Stretching after the eighth falling edge of SCL allows the slave to look at the received address or data and decide if it wants to ACK the received data.

32.5.6.4 Clock Synchronization and the CKP Bit

Any time the CKP bit is cleared, the module will wait for the SCL line to go low and then hold it. However, clearing the CKP bit will not assert the SCL output low until the SCL output is already sampled low. Therefore, the CKP bit will not assert the SCL line until an external I^{2}C master device has already asserted the SCL line. The SCL output will remain low until the CKP bit is set and all other devices on the I^{2}C bus have released SCL. This ensures that a write to the CKP bit will not violate the minimum high time requirement for SCL (see Figure 32-23).

FIGURE 32-23: CLOCK SYNCHRONIZATION TIMING

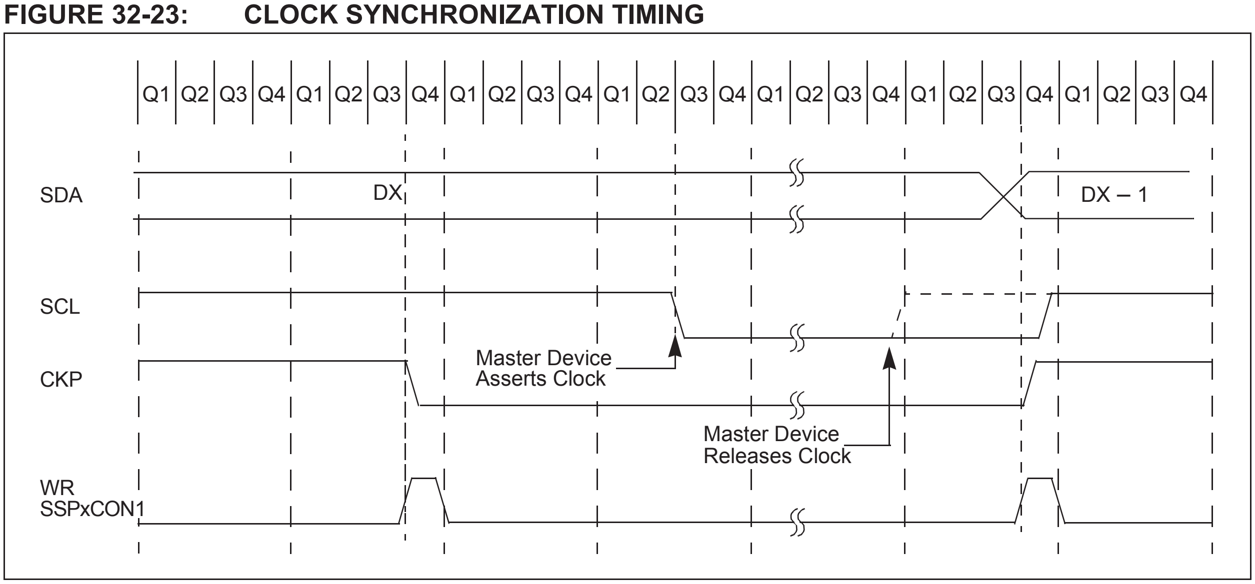


TABLE 32-3: SUMMARY OF REGISTERS ASSOCIATED WITH I²C OPERATION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
ANSELA	—	—	—	ANSA4	—	ANSA<2:0>			137
ANSELB(1)	ANSB<7:4>				—	—	—	—	143
ANSELC	ANSC<7:6>(1)		—	—	ANSC<3:0>				148
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	101
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	102
PIE2	OSFIE	C2IE	C1IE	—	BCL1IE	C4IE(1)	C3IE(1)	CCP2IE(1)	103
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	105
PIR2	OSFIF	C2IF	C1IF	—	BCL1IF	C4IF(1)	C3IF(1)	CCP2IF(1)	106
RxyPPS	—	—	—	RxyPPS<4:0>					154
SSPCLKPPS	—	—	—	SSPCLKPPS<4:0>					154, 156
SSPDATPPS	—	—	—	SSPDATPPS<4:0>					154, 156
SSPSSPPS	—	—	—	SSPSSPPS<4:0>					154, 156
SSP1ADD	ADD<7:0>								430
SSP1BUF	Synchronous Serial Port Receive Buffer/Transmit Register								380*
SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>				426
SSP1CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	428
SSP1CON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	429
SSP1MSK	MSK<7:0>								430
SSP1STAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	424
TRISA	—	—	TRISA<5:4>		—(2)	TRISA<2:0>			136
TRISB(1)	TRISB<7:4>				—	—	—	—	142
TRISC	TRISC<7:6>(1)		TRISC<5:0>						147

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by the MSSP module in I²C mode.
* Page provides register information.

Note 1: PIC16(L)F1768/9 only.
2: Unimplemented, read as '1'.

REGISTER 32-2: SSP1CON1: MSSP CONTROL REGISTER 1 (CONTINUED)

bit 4 **CKP:** Clock Polarity Select bit

In SPI mode:
1 = Idle state for clock is a high level
0 = Idle state for clock is a low level

In I^{2}C Slave mode:
SCL release control.
1 = Enables clock
0 = Holds clock low (clock stretch), used to ensure data setup time

In I^{2}C Master mode:
Unused in this mode.

bit 3-0 **SSPM<3:0>:** Synchronous Serial Port Mode Select bits

1111 = I^{2}C Slave mode, 10-bit address with Start and Stop bit interrupts enabled
1110 = I^{2}C Slave mode, 7-bit address with Start and Stop bit interrupts enabled
1101 = Reserved
1100 = Reserved
1011 = I^{2}C Firmware Controlled Master mode (slave Idle)
1010 = SPI Master mode, clock = FOSC/(4 * (SSP1ADD + 1))**(5)**
1001 = Reserved
1000 = I^{2}C Master mode, clock = FOSC/(4 * (SSP1ADD + 1))**(4)**
0111 = I^{2}C Slave mode, 10-bit address
0110 = I^{2}C Slave mode, 7-bit address
0101 = SPI Slave mode, clock = SCK pin, $\overline{SS}$ pin control is disabled, $\overline{SS}$ can be used as I/O pin
0100 = SPI Slave mode, clock = SCK pin, $\overline{SS}$ pin control is enabled
0011 = SPI Master mode, clock = T2_match/2
0010 = SPI Master mode, clock = FOSC/64
0001 = SPI Master mode, clock = FOSC/16
0000 = SPI Master mode, clock = FOSC/4

Note 1: In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPxBUF register.

2: When enabled, these pins must be properly configured as input or output. Use SSPSSPPS, SSPCLKPPS, SSPDATPPS and RxyPPS to select the pins.

3: When enabled, the SDA and SCL pins must be configured as inputs. Use SSPCLKPPS, SSPDATPPS and RxyPPS to select the pins.

4: SSPxADD values of 0, 1 or 2 are not supported for I^{2}C mode.

5: SSPxADD value of 0 is not supported; use SSPM<3:0> = 0000 instead.

TABLE 33-5: BAUD RATES FOR ASYNCHRONOUS MODES (CONTINUED)

BAUD RATE	SYNC = 0, BRGH = 1, BRG16 = 1 or SYNC = 1, BRG16 = 1											
	Fosc = 32.000 MHz			Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 11.0592 MHz		
	Actual Rate	% Error	SPxBRG value (decimal)	Actual Rate	% Error	SPxBRG value (decimal)	Actual Rate	% Error	SPxBRG value (decimal)	Actual Rate	% Error	SPxBRG value (decimal)
300	300.0	0.00	26666	300.0	0.00	16665	300.0	0.00	15359	300.0	0.00	9215
1200	1200	0.00	6666	1200	-0.01	4166	1200	0.00	3839	1200	0.00	2303
2400	2400	0.01	3332	2400	0.02	2082	2400	0.00	1919	2400	0.00	1151
9600	9604	0.04	832	9597	-0.03	520	9600	0.00	479	9600	0.00	287
10417	10417	0.00	767	10417	0.00	479	10425	0.08	441	10433	0.16	264
19.2k	19.18k	-0.08	416	19.23k	0.16	259	19.20k	0.00	239	19.20k	0.00	143
57.6k	57.55k	-0.08	138	57.47k	-0.22	86	57.60k	0.00	79	57.60k	0.00	47
115.2k	115.9k	0.64	68	116.3k	0.94	42	115.2k	0.00	39	115.2k	0.00	23

BAUD RATE	SYNC = 0, BRGH = 1, BRG16 = 1 or SYNC = 1, BRG16 = 1											
	Fosc = 8.000 MHz			Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 1.000 MHz		
	Actual Rate	% Error	SPxBRG value (decimal)	Actual Rate	% Error	SPxBRG value (decimal)	Actual Rate	% Error	SPxBRG value (decimal)	Actual Rate	% Error	SPxBRG value (decimal)
300	300.0	0.00	6666	300.0	0.01	3332	300.0	0.00	3071	300.1	0.04	832
1200	1200	-0.02	1666	1200	0.04	832	1200	0.00	767	1202	0.16	207
2400	2401	0.04	832	2398	0.08	416	2400	0.00	383	2404	0.16	103
9600	9615	0.16	207	9615	0.16	103	9600	0.00	95	9615	0.16	25
10417	10417	0	191	10417	0.00	95	10473	0.53	87	10417	0.00	23
19.2k	19.23k	0.16	103	19.23k	0.16	51	19.20k	0.00	47	19.23k	0.16	12
57.6k	57.14k	-0.79	34	58.82k	2.12	16	57.60k	0.00	15	—	—	—
115.2k	117.6k	2.12	16	111.1k	-3.55	8	115.2k	0.00	7	—	—	—

LSLF	**Logical Left Shift**
Syntax:	[*label*] LSLF f {,d}
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f<7>) → C (f<6:0>) → dest<7:1> 0 → dest<0>
Status Affected:	C, Z
Description:	The contents of register ‘f’ are shifted one bit to the left through the Carry flag. A ‘0’ is shifted into the LSb. If ‘d’ is ‘0’, the result is placed in W. If ‘d’ is ‘1’, the result is stored back in register ‘f’.

LSRF	**Logical Right Shift**
Syntax:	[*label*] LSRF f {,d}
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	0 → dest<7> (f<7:1>) → dest<6:0>, (f<0>) → C,
Status Affected:	C, Z
Description:	The contents of register ‘f’ are shifted one bit to the right through the Carry flag. A ‘0’ is shifted into the MSb. If ‘d’ is ‘0’, the result is placed in W. If ‘d’ is ‘1’, the result is stored back in register ‘f’.

MOVF	**Move f**
Syntax:	[*label*] MOVF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) → (dest)
Status Affected:	Z
Description:	The contents of register f are moved to a destination dependent upon the status of d. If d = 0, destination is W register. If d = 1, the destination is file register f itself. d = 1 is useful to test a file register since status flag Z is affected.
Words:	1
Cycles:	1
Example:	`MOVF FSR, 0`
	After Instruction W = value in FSRn register Z = 1

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

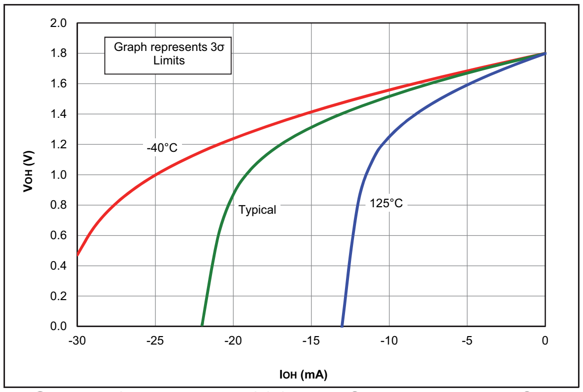


FIGURE 37-55: *100mA IO VOH vs. IOH Over Temperature, VDD = 1.8V, PIC16LF1764/5/8/9 Only.*

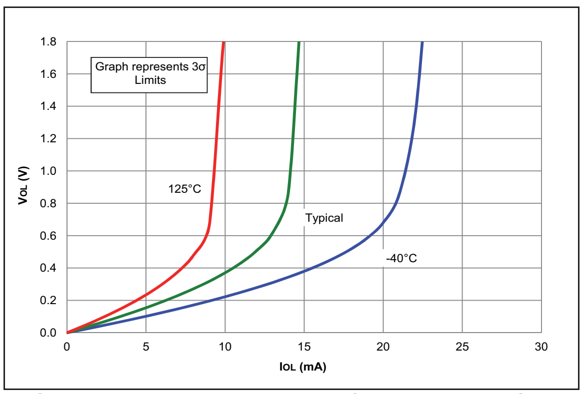


FIGURE 37-56: *100mA IO VOL vs. IOL Over Temperature, VDD = 1.8V, PIC16LF1764/5/8/9 Only.*

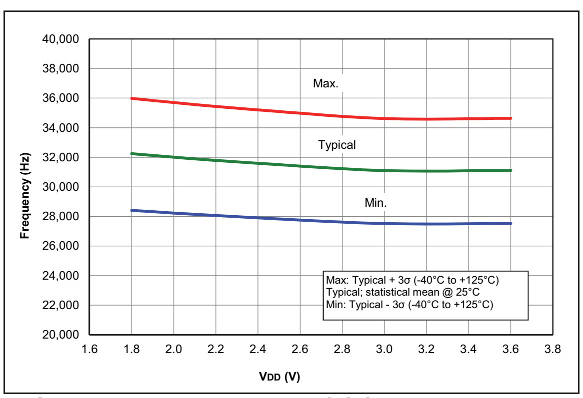


FIGURE 37-57: *LFINTOSC Frequency, PIC16LF1764/5/8/9 Only.*

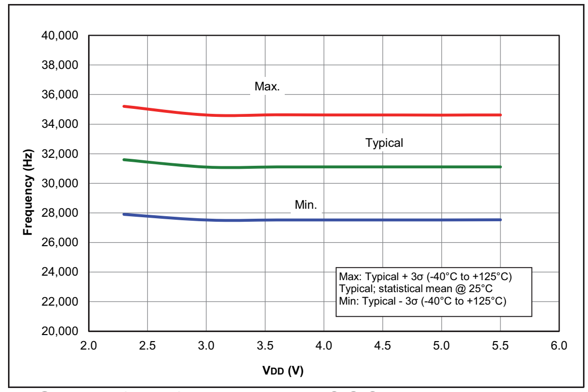


FIGURE 37-58: *LFINTOSC Frequency, PIC16F1764/5/8/9 Only.*

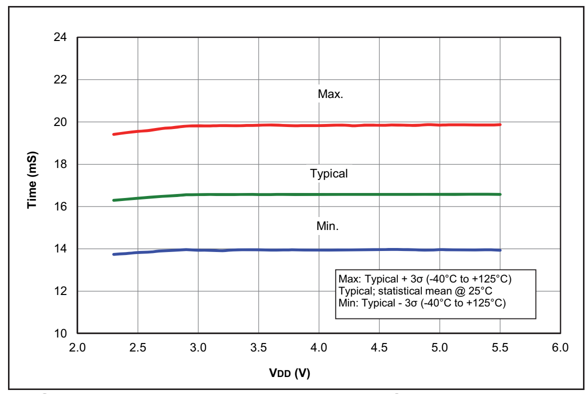


FIGURE 37-59: *WDT Time-Out Period, PIC16F1764/5/8/9 Only.*

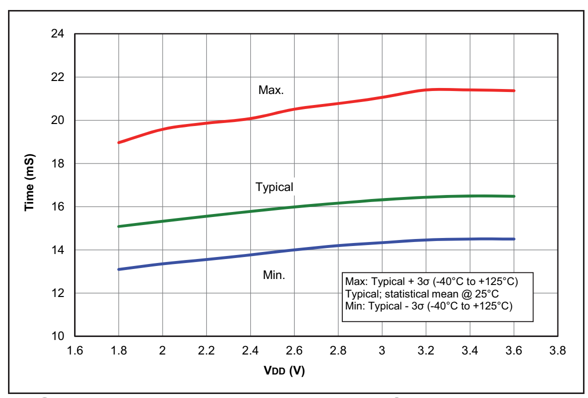


FIGURE 37-60: *WDT Time-Out Period, PIC16LF1764/5/8/9 Only.*