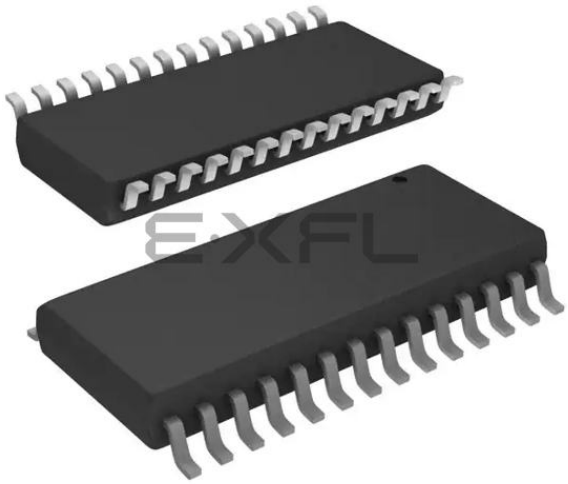


Welcome to E-XFL.COM

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, PSMC, PWM, WDT
Number of I/O	24
Program Memory Size	28KB (16K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 11x12b; D/A 1x8b, 3x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf1788-i-so

TABLE 3-12: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
Bank 30											
F0Ch — F10h	—	Unimplemented								—	—
F11h	PSMC4CON	PSMC4EN	PSMC4LD	P4DBFE	P4DBRE	P4MODE<3:0>				0000 0000	0000 0000
F12h	PSMC4MDL	P4MDLEN	P4MDLPOL	P4MDLBIT	—	P4MSRC<3:0>				000- 0000	000- 0000
F13h	PSMC4SYNC	P4POFST	P4PRPOL	P4DCPOL	—	—	P4SYNC<2:0>			000- -000	000- -000
F14h	PSMC4CLK	—	—	P4CPRE<1:0>		—	—	P4CSRC<1:0>		--00 --00	--00 --00
F15h	PSMC4OEN	—	—	—	—	—	—	P4OEB	P4OEA	---- --00	---- --00
F16h	PSMC4POL	—	P4INPOL	—	—	—	—	P4POLB	P4POLA	-0-- --00	-0-- --00
F17h	PSMC4BLNK	—	—	P4FEBM<1:0>		—	—	P4REBM<1:0>		--00 --00	--00 --00
F18h	PSMC4REBS	P4REBSIN	—	—	P4REBSC4	P4REBSC3	P4REBSC2	P4REBSC1	—	0--0 000-	0--0 000-
F19h	PSMC4FEBS	P4FEBSIN	—	—	P4FEBSC4	P4FEBSC3	P4FEBSC2	P4FEBSC1	—	0--0 000-	0--0 000-
F1Ah	PSMC4PHS	P4PHSIN	—	—	P4PHSC4	P4PHSC3	P4PHSC2	P4PHSC1	P4PHST	0--0 0000	0--0 0000
F1Bh	PSMC4DCS	P4DCSIN	—	—	P4DCSC4	P4DCSC3	P4DCSC2	P4DCSC1	P4DCST	0--0 0000	0--0 0000
F1Ch	PSMC4PRS	P4PRSIN	—	—	P4PRSC4	P4PRSC3	P4PRSC2	P4PRSC1	P4PRST	0--0 0000	0--0 0000
F1Dh	PSMC4ASDC	P4ASE	P4ASDEN	P4ARSEN	—	—	—	—	P4ASDOV	000- ---0	000- ---0
F1Eh	PSMC4ASDL	—	—	—	—	—	—	P4ASDLB	P4ASDLA	---- --00	---- --00
F1Fh	PSMC4ASDS	P4ASDSIN	—	—	P4ASDSC4	P4ASDSC3	P4ASDSC2	P4ASDSC1	—	0--0 000-	0--0 000-
F20h	PSMC4INT	P4TOVIE	P4TPHIE	P4TDCIE	P4TPRIE	P4TOVIF	P4TPHIF	P4TDCIF	P4TPRIF	0000 0000	0000 0000
F21h	PSMC4PHL	Phase Low Count								0000 0000	0000 0000
F22h	PSMC4PHH	Phase High Count								0000 0000	0000 0000
F23h	PSMC4DCL	Duty Cycle Low Count								0000 0000	0000 0000
F24h	PSMC4DCH	Duty Cycle High Count								0000 0000	0000 0000
F25h	PSMC4PRL	Period Low Count								0000 0000	0000 0000
F26h	PSMC4PRH	Period High Count								0000 0000	0000 0000
F27h	PSMC4TMRL	Time base Low Counter								0000 0001	0000 0001
F28h	PSMC4TMRH	Time base High Counter								0000 0000	0000 0000
F29h	PSMC4DBR	Rising Edge Dead-band Counter								0000 0000	0000 0000
F2Ah	PSMC4DBF	Falling Edge Dead-band Counter								0000 0000	0000 0000
F2Bh	PSMC4BLKR	Rising Edge Blanking Counter								0000 0000	0000 0000
F2Ch	PSMC4BLKF	Falling Edge Blanking Counter								0000 0000	0000 0000
F2Dh	PSMC4FFA	—	—	—	—	Fractional Frequency Adjust Register				---- 0000	---- 0000
F2Eh	PSMC4STR0	—	—	—	—	—	—	P4STRB	P4STRA	---- --01	---- --01
F2Fh	PSMC4STR1	P4SSYNC	—	—	—	—	—	P4LSMEN	P4HSMEN	0--- --00	0--- --00

Legend: x = unknown, u = unchanged, q = value depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations are unimplemented, read as '0'.

Note 1: These registers can be addressed from any bank.
2: Unimplemented, read as '1'.
3: PIC16(L)F1789 only.
4: PIC16F1788/9 only.

4.3 Code Protection

Code protection allows the device to be protected from unauthorized access. Program memory protection and data EEPROM protection are controlled independently. Internal access to the program memory and data EEPROM are unaffected by any code protection setting.

4.3.1 PROGRAM MEMORY PROTECTION

The entire program memory space is protected from external reads and writes by the $\overline{CP}$ bit in Configuration Words. When $\overline{CP}$ = 0, external reads and writes of program memory are inhibited and a read will return all '0's. The CPU can continue to read program memory, regardless of the protection bit settings. Writing the program memory is dependent upon the write protection setting. See **Section 4.4 "Write Protection"** for more information.

4.3.2 DATA EEPROM PROTECTION

The entire data EEPROM is protected from external reads and writes by the $\overline{CPD}$ bit. When $\overline{CPD}$ = 0, external reads and writes of data EEPROM are inhibited. The CPU can continue to read and write data EEPROM regardless of the protection bit settings.

4.4 Write Protection

Write protection allows the device to be protected from unintended self-writes. Applications, such as bootloader software, can be protected while allowing other regions of the program memory to be modified.

The WRT<1:0> bits in Configuration Words define the size of the program memory block that is protected.

4.5 User ID

Four memory locations (8000h-8003h) are designated as ID locations where the user can store checksum or other code identification numbers. These locations are readable and writable during normal execution. See **Section 12.5 "User ID, Device ID and Configuration Word Access"**for more information on accessing these memory locations. For more information on checksum calculation, see the *"PIC16(L)F178X Memory Programming Specification"* (DS41457).

5.4 Low-Power Brown-Out Reset (LPBOR)

The Low-Power Brown-Out Reset (LPBOR) is an essential part of the Reset subsystem. Refer to Figure 5-1 to see how the BOR interacts with other modules.

The LPBOR is used to monitor the external VDD pin. When too low of a voltage is detected, the device is held in Reset. When this occurs, a register bit ($\overline{BOR}$) is changed to indicate that a BOR Reset has occurred. The same bit is set for both the BOR and the LPBOR. Refer to Register 5-2.

5.4.1 ENABLING LPBOR

The LPBOR is controlled by the $\overline{LPBOR}$ bit of Configuration Words. When the device is erased, the LPBOR module defaults to disabled.

5.4.1.1 LPBOR Module Output

The output of the LPBOR module is a signal indicating whether or not a Reset is to be asserted. This signal is OR'd together with the Reset signal of the BOR module to provide the generic $\overline{BOR}$ signal, which goes to the PCON register and to the power control block.

5.5 $\overline{MCLR}$

The $\overline{MCLR}$ is an optional external input that can reset the device. The $\overline{MCLR}$ function is controlled by the MCLRE bit of Configuration Words and the LVP bit of Configuration Words (Table 5-2).

TABLE 5-2: $\overline{MCLR}$ CONFIGURATION

MCLRE	LVP	$\overline{MCLR}$
0	0	Disabled
1	0	Enabled
x	1	Enabled

5.5.1 $\overline{MCLR}$ ENABLED

When $\overline{MCLR}$ is enabled and the pin is held low, the device is held in Reset. The $\overline{MCLR}$ pin is connected to VDD through an internal weak pull-up.

The device has a noise filter in the $\overline{MCLR}$ Reset path. The filter will detect and ignore small pulses.

Note:	A Reset does not drive the $\overline{MCLR}$ pin low.

5.5.2 $\overline{MCLR}$ DISABLED

When $\overline{MCLR}$ is disabled, the pin functions as a general purpose input and the internal weak pull-up is under software control. See **Section 13.11 "PORTE Registers"** for more information.

5.6 Watchdog Timer (WDT) Reset

The Watchdog Timer generates a Reset if the firmware does not issue a CLRWDT instruction within the time-out period. The $\overline{TO}$ and $\overline{PD}$ bits in the STATUS register are changed to indicate the WDT Reset. See **Section 11.0 "Watchdog Timer (WDT)"** for more information.

5.7 RESET Instruction

A RESET instruction will cause a device Reset. The $\overline{RI}$ bit in the PCON register will be set to '0'. See Table 5-4 for default conditions after a RESET instruction has occurred.

5.8 Stack Overflow/Underflow Reset

The device can reset when the Stack Overflows or Underflows. The STKOVF or STKUNF bits of the PCON register indicate the Reset condition. These Resets are enabled by setting the STVREN bit in Configuration Words. See **Section 5.8 "Stack Overflow/Underflow Reset"** for more information.

5.9 Programming Mode Exit

Upon exit of Programming mode, the device will behave as if a POR had just occurred.

5.10 Power-Up Timer

The Power-up Timer optionally delays device execution after a BOR or POR event. This timer is typically used to allow VDD to stabilize before allowing the device to start running.

The Power-up Timer is controlled by the $\overline{PWRTE}$ bit of Configuration Words.

5.11 Start-up Sequence

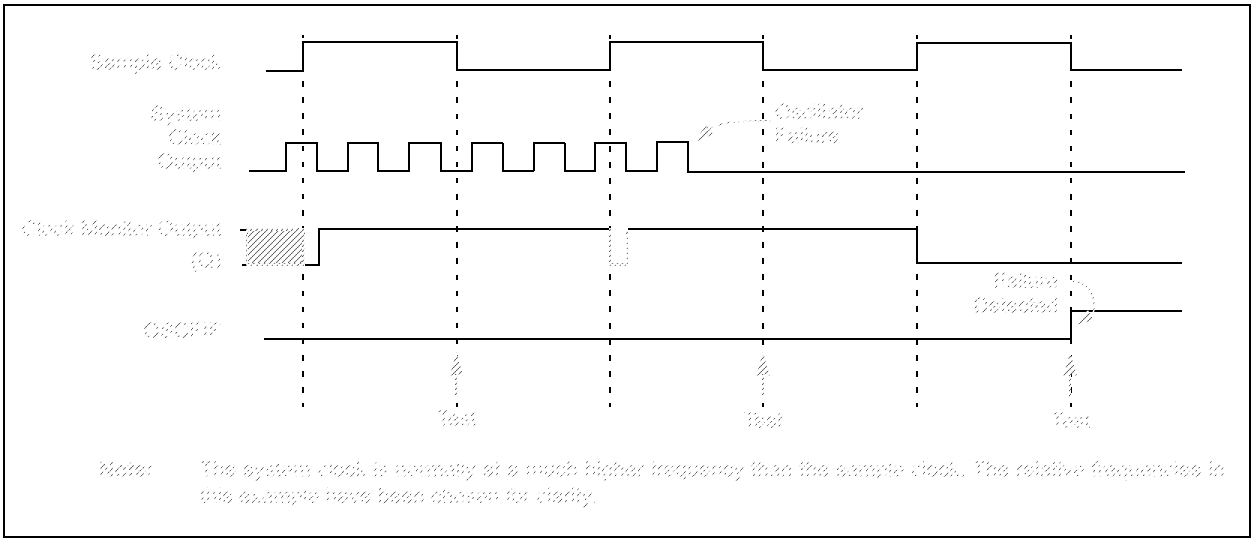
Upon the release of a POR or BOR, the following must occur before the device will begin executing:

1. Power-up Timer runs to completion (if enabled).
2. Oscillator start-up timer runs to completion (if required for oscillator source).
3. $\overline{MCLR}$ must be released (if enabled).

The total time-out will vary based on oscillator configuration and Power-up Timer configuration. See **Section 6.0 "Oscillator Module (with Fail-Safe Clock Monitor)"** for more information.

The Power-up Timer and oscillator start-up timer run independently of $\overline{MCLR}$ Reset. If $\overline{MCLR}$ is kept low long enough, the Power-up Timer and oscillator start-up timer will expire. Upon bringing $\overline{MCLR}$ high, the device will begin execution immediately (see Figure 5-3). This is useful for testing purposes or to synchronize more than one device operating in parallel.

FIGURE 6-10: FSCM TIMING DIAGRAM



Note: The system clock is normally at a much higher frequency than the sample clock. The relative frequencies in this example have been chosen for clarity.

21.0 COMPARATOR MODULE

Comparators are used to interface analog circuits to a digital circuit by comparing two analog voltages and providing a digital indication of their relative magnitudes. Comparators are very useful mixed signal building blocks because they provide analog functionality independent of program execution. The analog comparator module includes the following features:

- Independent comparator control
- Programmable input selection
- Comparator output is available internally/externally
- Programmable output polarity
- Interrupt-on-change
- Wake-up from Sleep
- Programmable Speed/Power optimization
- PWM shutdown
- Programmable and fixed voltage reference

21.1 Comparator Overview

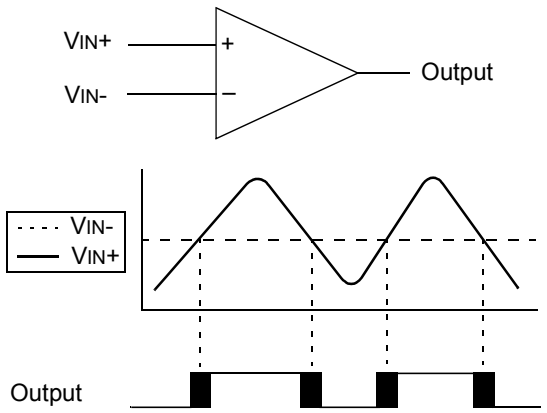
A single comparator is shown in Figure 21-1 along with the relationship between the analog input levels and the digital output. When the analog voltage at VIN+ is less than the analog voltage at VIN-, the output of the comparator is a digital low level. When the analog voltage at VIN+ is greater than the analog voltage at VIN-, the output of the comparator is a digital high level.

The comparators available for this device are located in Table 21-1.

TABLE 21-1: COMPARATOR AVAILABILITY PER DEVICE

Device	C1	C2	C3	C4
PIC16(L)F1788/9	●	●	●	●

FIGURE 21-1: SINGLE COMPARATOR



Note: The black areas of the output of the comparator represents the uncertainty due to input offsets and response time.

22.1.3 SOFTWARE PROGRAMMABLE PRESCALER

A software programmable prescaler is available for exclusive use with Timer0. The prescaler is enabled by clearing the PSA bit of the OPTION_REG register.

Note:	The Watchdog Timer (WDT) uses its own independent prescaler.

There are eight prescaler options for the Timer0 module ranging from 1:2 to 1:256. The prescale values are selectable via the PS<2:0> bits of the OPTION_REG register. In order to have a 1:1 prescaler value for the Timer0 module, the prescaler must be disabled by setting the PSA bit of the OPTION_REG register.

The prescaler is not readable or writable. All instructions writing to the TMR0 register will clear the prescaler.

22.1.4 TIMER0 INTERRUPT

Timer0 will generate an interrupt when the TMR0 register overflows from FFh to 00h. The TMR0IF interrupt flag bit of the INTCON register is set every time the TMR0 register overflows, regardless of whether or not the Timer0 interrupt is enabled. The TMR0IF bit can only be cleared in software. The Timer0 interrupt enable is the TMR0IE bit of the INTCON register.

Note:	The Timer0 interrupt cannot wake the processor from Sleep since the timer is frozen during Sleep.

22.1.5 8-BIT COUNTER MODE SYNCHRONIZATION

When in 8-Bit Counter mode, the incrementing edge on the T0CKI pin must be synchronized to the instruction clock. Synchronization can be accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the instruction clock. The high and low periods of the external clocking source must meet the timing requirements as shown in **Section 31.0 "Electrical Specifications"**.

22.1.6 OPERATION DURING SLEEP

Timer0 cannot operate while the processor is in Sleep mode. The contents of the TMR0 register will remain unchanged while the processor is in Sleep mode.

23.3 Timer1 Prescaler

Timer1 has four prescaler options allowing 1, 2, 4 or 8 divisions of the clock input. The T1CKPS 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.

23.4 Timer1 Oscillator

A dedicated low-power 32.768 kHz oscillator circuit is built-in between pins T1OSI (input) and T1OSO (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 T1OSCEN 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, T1OSCEN 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.

23.5 Timer1 Operation in Asynchronous Counter Mode

If the control bit $\overline{\text{T1SYNC}}$ 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 23.5.1 "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.

23.5.1 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.

23.6 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.

23.6.1 TIMER1 GATE ENABLE

The Timer1 Gate Enable mode is enabled by setting the TMR1GE bit of the T1GCON register. The polarity of the Timer1 Gate Enable mode is configured using the T1GPOL 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 23-3 for timing details.

TABLE 23-3: TIMER1 GATE ENABLE SELECTIONS

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

26.2.7 ASYNCHRONOUS INPUTS

The PSMC module supports asynchronous inputs alone or in combination with the synchronous inputs. asynchronous inputs include:

- Analog
  - sync_C1OUT
  - sync_C2OUT
  - sync_C3OUT
  - sync_C4OUT
- Digital
  - PSMCxIN pin

26.2.7.1 Comparator Inputs

The outputs of any combination of the synchronized comparators may be used to trigger any of the three events as well as auto-shutdown.

The event triggers on the rising edge of the comparator output. Except for auto-shutdown, the event input is not level sensitive.

26.2.7.2 PSMCxIN Pin Input

The PSMCxIN pin may be used to trigger PSMC events. Data is passed through straight to the PSMC module without any synchronization to a system clock. This is so that input blanking may be applied to any external circuit using the module.

The event triggers on the rising edge of the PSMCxIN signal.

26.2.7.3 Asynchronous Polarity

Polarity control is available for the period and duty-cycle asynchronous event inputs. Polarity control is necessary when the same signal is used as the source for both events. Inverting the polarity of one event relative to the other enables starting the period on one edge of the signal and terminating the duty-cycle on the opposite edge. Polarity is controlled with the PxPRPOL and PxDCPOL bits of the PSMCxSYNC register. Inverting the asynchronous input with these controls inverts all enabled asynchronous inputs for the corresponding event.

26.2.8 INPUT BLANKING

Input blanking is a function whereby the inputs from any selected asynchronous input may be driven inactive for a short period of time. This is to prevent electrical transients from the turn-on/off of power components from generating a false event.

Blanking is initiated by either or both:

- Rising event
- Falling event

Blanked inputs are suppressed from causing all asynchronous events, including:

- Rising
- Falling
- Period
- Shutdown

Rising edge and falling edge blanking are controlled independently. The following features are available for blanking:

- Blanking enable
- Blanking time counters
- Blanking mode

The following Blanking modes are available:

- Blanking disabled
- Immediate blanking

The Falling Edge Blanking mode is set with the PxFEBM<1:0> bits of the PSMCx Blanking Control (PSMCxBLNK) register (Register 26-10).

The Rising Edge Blanking mode is set with the PxREBM<1:0> bits of the PSMCx Blanking Control (PSMCxBLNK) register (Register 26-10).

26.2.8.1 Blanking Disabled

With blanking disabled, the asynchronous inputs are passed to the PSMC module without any intervention.

26.2.8.2 Immediate Blanking

With Immediate blanking, a counter is used to determine the blanking period. The desired blanking time is measured in psmc_clk periods. A rising edge event will start incrementing the rising edge blanking counter. A falling edge event will start incrementing the falling edge blanking counter.

The rising edge blanking time is set with the PSMC Rising Edge Blanking Time (PSMCxBLKR) register (Register 26-30). The inputs to be blanked are selected with the PSMC Rising Edge Blanked Source (PSMCxREBS) register (Register 26-11). During rising edge blanking, the selected blanked sources are suppressed for falling edge as well as rising edge, auto-shutdown and period events.

The falling edge blanking time is set with the PSMC Falling Edge Blanking Time (PSMCxBLKF) register (Register 26-31). The inputs to be blanked are selected with the PSMC Falling Edge Blanked Source (PSMCxFEBS) register (Register 26-12). During falling edge blanking, the selected blanked sources are suppressed for rising edge, as well as falling edge, auto-shutdown, and period events.

The blanking counters are incremented on the rising edge of psmc_clk. Blanked sources are suppressed until the counter value equals the blanking time register causing the blanking to terminate.

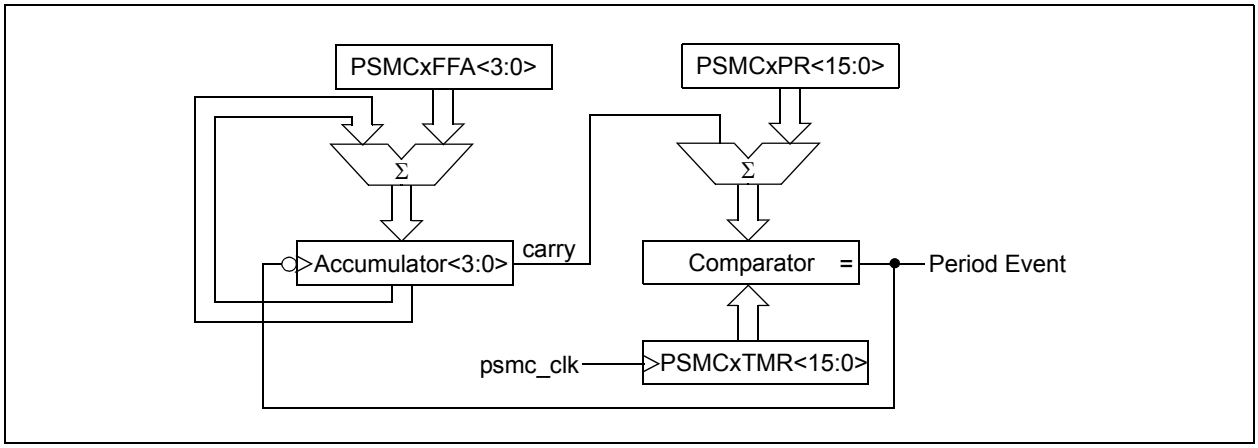
As the rising and falling edge events are from asynchronous inputs, there may be some uncertainty in the actual blanking time implemented in each cycle. The maximum uncertainty is equal to one psmc_clk period.

26.9 Fractional Frequency Adjust (FFA)

FFA is a method by which PWM resolution can be improved on 50% fixed duty cycle signals. Higher resolution is achieved by altering the PWM period by a single count for calculated intervals. This increased resolution is based upon the PWM frequency averaged over a large number of PWM periods. For example, if the period event time is increased by one psmc_clk period (TPSMC_CLK) every N events, then the effective resolution of the average event period is TPSMC_CLK/N.

When active, after every period event the FFA hardware adds the PSMCxFFA value with the previously accumulated result. Each time the addition causes an overflow, the period event time is increased by one. Refer to Figure 26-22.

FIGURE 26-22: FFA BLOCK DIAGRAM.



The FFA function is only available when using one of the two Fixed Duty Cycle modes of operation. In fixed duty cycle operation each PWM period is comprised of two period events. That is why the PWM periods in Table 26-3 example calculations are multiplied by two as opposed to the normal period calculations for normal mode operation.

The extra resolution gained by the FFA is based upon the number of bits in the FFA register and the psmc_clk frequency. The parameters of interest are:

- TPWM – this is the lower bound of the PWM period that will be adjusted
- TPWM+1 – this is the upper bound of the PWM period that will be adjusted. This is used to help determine the step size for each increment of the FFA register
- TRESOLUTION – each increment of the FFA register will add this amount of period to average PWM frequency

TABLE 26-3: FRACTIONAL FREQUENCY ADJUST CALCULATIONS

Parameter	Value
FPSMC_CLK	64 MHz
TPSMC_CLK	15.625 ns
PSMCxPR<15:0>	00FFh = 255
TPWM	= (PSMCxPR<15:0>+1)*2*TPSMC_CLK = 256*2*15.625ns = 8 us
FPWM	125 kHz
TPWM+1	= (PSMCxPR<15:0>+2)*2*TPSMC_CLK = 257*2*15.625ns = 8.03125 us
FPWM+1	= 124.513 kHz
TRESOLUTION	= (TPWM+1-TPWM)/$2^{FFA\text{-}Bits}$ = (8.03125us - 8.0 us)/16 = 0.03125us/16 ~ 1.95 ns
FRESOLUTION	(FPWM+1-FPWM)/$2^{FFA\text{-}Bits}$ ~ -30.4 Hz

REGISTER 26-18: PSMCxASDS: PSMC AUTO-SHUTDOWN SOURCE REGISTER

R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0
PxASDSIN	—	—	PxASDSC4	PxASDSC3	PxASDSC2	PxASDSC1	—
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 **PxASDSIN:** Auto-shutdown occurs on PSMCxIN pin
1 = Auto-shutdown will occur when PSMCxIN pin goes true
0 = PSMCxIN pin will not cause auto-shutdown

bit 6-5 **Unimplemented:** Read as '0'

bit 4 **PxASDSC4:** Auto-shutdown occurs on sync_C4OUT output
1 = Auto-shutdown will occur when sync_C4OUT output goes true
0 = sync_C4OUT will not cause auto-shutdown

bit 3 **PxASDSC3:** Auto-shutdown occurs on sync_C3OUT output
1 = Auto-shutdown will occur when sync_C3OUT output goes true
0 = sync_C3OUT will not cause auto-shutdown

bit 2 **PxASDSC2:** Auto-shutdown occurs on sync_C2OUT output
1 = Auto-shutdown will occur when sync_C2OUT output goes true
0 = sync_C2OUT will not cause auto-shutdown

bit 1 **PxASDSC1:** Auto-shutdown occurs on sync_C1OUT output
1 = Auto-shutdown will occur when sync_C1OU output goes true
0 = sync_C1OU will not cause auto-shutdown

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

27.5.2 SLAVE RECEPTION

When the R/$\overline{W}$ bit of a matching received address byte is clear, the R/$\overline{W}$ bit of the SSPSTAT register is cleared. The received address is loaded into the SSPBUF register and acknowledged.

When the overflow condition exists for a received address, then not Acknowledge is given. An overflow condition is defined as either bit BF of the SSPSTAT register is set, or bit SSPOV of the SSPCON1 register is set. The BOEN bit of the SSPCON3 register modifies this operation. For more information see Register 27-4.

An MSSP interrupt is generated for each transferred data byte. Flag bit, SSP1IF, must be cleared by software.

When the SEN bit of the SSPCON2 register is set, SCL will be held low (clock stretch) following each received byte. The clock must be released by setting the CKP bit of the SSPCON1 register, except sometimes in 10-bit mode. See **Section 27.2.3 "SPI Master Mode"** for more detail.

27.5.2.1 7-bit Addressing Reception

This section describes a standard sequence of events for the MSSP module configured as an I^2C Slave in 7-bit Addressing mode. All decisions made by hardware or software and their effect on reception. Figure 27-13 and Figure 27-14 is used as a visual reference for this description.

This is a step by step process of what typically must be done to accomplish I^2C communication.

1. Start bit detected.
2. S bit of SSPSTAT is set; SSP1IF is set if interrupt on Start detect is enabled.
3. Matching address with R/$\overline{W}$ bit clear is received.
4. The slave pulls SDA low sending an $\overline{ACK}$ to the master, and sets SSP1IF bit.
5. Software clears the SSP1IF bit.
6. Software reads received address from SSPBUF clearing the BF flag.
7. If SEN = 1; Slave software sets CKP bit to release the SCL line.
8. The master clocks out a data byte.
9. Slave drives SDA low sending an $\overline{ACK}$ to the master, and sets SSP1IF bit.
10. Software clears SSP1IF.
11. Software reads the received byte from SSPBUF clearing BF.
12. Steps 8-12 are repeated for all received bytes from the master.
13. Master sends Stop condition, setting P bit of SSPSTAT, and the bus goes idle.

27.5.2.2 7-bit Reception with AHEN and DHEN

Slave device reception with AHEN and DHEN set operate the same as without these options with extra interrupts and clock stretching added after the 8th falling edge of SCL. These additional interrupts allow the slave software to decide whether it wants to $\overline{ACK}$ the receive address or data byte, rather than the hardware. This functionality adds support for PMBus™ that was not present on previous versions of this module.

This list describes the steps that need to be taken by slave software to use these options for I^2C communication. Figure 27-15 displays a module using both address and data holding. Figure 27-16 includes the operation with the SEN bit of the SSPCON2 register set.

1. S bit of SSPSTAT is set; SSP1IF is set if interrupt on Start detect is enabled.
2. Matching address with R/$\overline{W}$ bit clear is clocked in. SSP1IF is set and CKP cleared after the 8th falling edge of SCL.
3. Slave clears the SSP1IF.
4. Slave can look at the ACKTIM bit of the SSPCON3 register to determine if the SSP1IF was after or before the $\overline{ACK}$.
5. Slave reads the address value from SSPBUF, clearing the BF flag.
6. Slave sets $\overline{ACK}$ value clocked out to the master by setting ACKDT.
7. Slave releases the clock by setting CKP.
8. SSP1IF is set after an $\overline{ACK}$, not after a NACK.
9. If SEN = 1 the slave hardware will stretch the clock after the $\overline{ACK}$.
10. Slave clears SSP1IF.

Note: SSP1IF is still set after the 9th falling edge of SCL even if there is no clock stretching and BF has been cleared. Only if NACK is sent to master is SSP1IF not set

11. SSP1IF set and CKP cleared after 8th falling edge of SCL for a received data byte.
12. Slave looks at ACKTIM bit of SSPCON3 to determine the source of the interrupt.
13. Slave reads the received data from SSPBUF clearing BF.
14. Steps 7-14 are the same for each received data byte.
15. Communication is ended by either the slave sending an $\overline{ACK}$ = 1, or the master sending a Stop condition. If a Stop is sent and Interrupt on Stop Detect is disabled, the slave will only know by polling the P bit of the SSTSTAT register.

27.6.6 I²C MASTER MODE TRANSMISSION

Transmission of a data byte, a 7-bit address or the other half of a 10-bit address is accomplished by simply writing a value to the SSPBUF register. This action will set the Buffer Full flag bit, BF and allow the Baud Rate Generator to begin counting and start the next transmission. Each bit of address/data will be shifted out onto the SDA pin after the falling edge of SCL is asserted. SCL is held low for one Baud Rate Generator rollover count (TBRG). Data should be valid before SCL is released high. When the SCL pin is released high, it is held that way for TBRG. The data on the SDA pin must remain stable for that duration and some hold time after the next falling edge of SCL. After the eighth bit is shifted out (the falling edge of the eighth clock), the BF flag is cleared and the master releases SDA. This allows the slave device being addressed to respond with an $\overline{ACK}$ bit during the ninth bit time if an address match occurred, or if data was received properly. The status of $\overline{ACK}$ is written into the ACKSTAT bit on the rising edge of the ninth clock. If the master receives an Acknowledge, the Acknowledge Status bit, ACKSTAT, is cleared. If not, the bit is set. After the ninth clock, the SSP1IF bit is set and the master clock (Baud Rate Generator) is suspended until the next data byte is loaded into the SSPBUF, leaving SCL low and SDA unchanged (Figure 27-27).

After the write to the SSPBUF, each bit of the address will be shifted out on the falling edge of SCL until all seven address bits and the R/$\overline{W}$ bit are completed. On the falling edge of the eighth clock, the master will release the SDA pin, allowing the slave to respond with an Acknowledge. On the falling edge of the ninth clock, the master will sample the SDA pin to see if the address was recognized by a slave. The status of the $\overline{ACK}$ bit is loaded into the ACKSTAT Status bit of the SSPCON2 register. Following the falling edge of the ninth clock transmission of the address, the SSP1IF is set, the BF flag is cleared and the Baud Rate Generator is turned off until another write to the SSPBUF takes place, holding SCL low and allowing SDA to float.

27.6.6.1 BF Status Flag

In Transmit mode, the BF bit of the SSPSTAT register is set when the CPU writes to SSPBUF and is cleared when all eight bits are shifted out.

27.6.6.2 WCOL Status Flag

If the user writes the SSPBUF when a transmit is already in progress (i.e., SSPSR is still shifting out a data byte), the WCOL is set and the contents of the buffer are unchanged (the write does not occur).

WCOL must be cleared by software before the next transmission.

27.6.6.3 ACKSTAT Status Flag

In Transmit mode, the ACKSTAT bit of the SSPCON2 register is cleared when the slave has sent an Acknowledge ($\overline{ACK}$ = 0) and is set when the slave does not Acknowledge ($\overline{ACK}$ = 1). A slave sends an Acknowledge when it has recognized its address (including a general call), or when the slave has properly received its data.

27.6.6.4 Typical transmit sequence:

1. The user generates a Start condition by setting the SEN bit of the SSPCON2 register.
2. SSP1IF is set by hardware on completion of the Start.
3. SSP1IF is cleared by software.
4. The MSSP module will wait the required start time before any other operation takes place.
5. The user loads the SSPBUF with the slave address to transmit.
6. Address is shifted out the SDA pin until all eight bits are transmitted. Transmission begins as soon as SSPBUF is written to.
7. The MSSP module shifts in the $\overline{ACK}$ bit from the slave device and writes its value into the ACKSTAT bit of the SSPCON2 register.
8. The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSP1IF bit.
9. The user loads the SSPBUF with eight bits of data.
10. Data is shifted out the SDA pin until all eight bits are transmitted.
11. The MSSP module shifts in the $\overline{ACK}$ bit from the slave device and writes its value into the ACKSTAT bit of the SSPCON2 register.
12. Steps 8-11 are repeated for all transmitted data bytes.
13. The user generates a Stop or Restart condition by setting the PEN or RSEN bits of the SSPCON2 register. Interrupt is generated once the Stop/Restart condition is complete.

28.1.2 EUSART ASYNCHRONOUS RECEIVER

The Asynchronous mode is typically used in RS-232 systems. The receiver block diagram is shown in Figure 28-2. The data is received on the RX/DT pin and drives the data recovery block. The data recovery block is actually a high-speed shifter operating at 16 times the baud rate, whereas the serial Receive Shift Register (RSR) operates at the bit rate. When all eight or nine bits of the character have been shifted in, they are immediately transferred to a two character First-In-First-Out (FIFO) memory. The FIFO buffering allows reception of two complete characters and the start of a third character before software must start servicing the EUSART receiver. The FIFO and RSR registers are not directly accessible by software. Access to the received data is via the RCREG register.

28.1.2.1 Enabling the Receiver

The EUSART receiver is enabled for asynchronous operation by configuring the following three control bits:

- CREN = 1
- SYNC = 0
- SPEN = 1

All other EUSART control bits are assumed to be in their default state.

Setting the CREN bit of the RCSTA register enables the receiver circuitry of the EUSART. Clearing the SYNC bit of the TXSTA register configures the EUSART for asynchronous operation. Setting the SPEN bit of the RCSTA register enables the EUSART. The programmer must set the corresponding TRIS bit to configure the RX/DT I/O pin as an input.

Note:	If the RX/DT function is on an analog pin, the corresponding ANSEL bit must be cleared for the receiver to function.

28.1.2.2 Receiving Data

The receiver data recovery circuit initiates character reception on the falling edge of the first bit. The first bit, also known as the Start bit, is always a zero. The data recovery circuit counts one-half bit time to the center of the Start bit and verifies that the bit is still a zero. If it is not a zero then the data recovery circuit aborts character reception, without generating an error, and resumes looking for the falling edge of the Start bit. If the Start bit zero verification succeeds then the data recovery circuit counts a full bit time to the center of the next bit. The bit is then sampled by a majority detect circuit and the resulting '0' or '1' is shifted into the RSR. This repeats until all data bits have been sampled and shifted into the RSR. One final bit time is measured and the level sampled. This is the Stop bit, which is always a '1'. If the data recovery circuit samples a '0' in the Stop bit position then a framing error is set for this character, otherwise the framing error is cleared for this character. See **Section 28.1.2.4 "Receive Framing Error"** for more information on framing errors.

Immediately after all data bits and the Stop bit have been received, the character in the RSR is transferred to the EUSART receive FIFO and the RCIF interrupt flag bit of the PIR1 register is set. The top character in the FIFO is transferred out of the FIFO by reading the RCREG register.

Note:	If the receive FIFO is overrun, no additional characters will be received until the overrun condition is cleared. See **Section 28.1.2.5 "Receive Overrun Error"** for more information on overrun errors.

28.1.2.3 Receive Interrupts

The RCIF interrupt flag bit of the PIR1 register is set whenever the EUSART receiver is enabled and there is an unread character in the receive FIFO. The RCIF interrupt flag bit is read-only, it cannot be set or cleared by software.

RCIF interrupts are enabled by setting all of the following bits:

- RCIE, Interrupt Enable bit of the PIE1 register
- PEIE, Peripheral Interrupt Enable bit of the INTCON register
- GIE, Global Interrupt Enable bit of the INTCON register

The RCIF interrupt flag bit will be set when there is an unread character in the FIFO, regardless of the state of interrupt enable bits.

REGISTER 28-2: RCSTA: RECEIVE STATUS AND CONTROL REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R-0/0	R-0/0	R-0/0
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
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 **SPEN:** Serial Port Enable bit
1 = Serial port enabled (configures RX/DT and TX/CK pins as serial port pins)
0 = Serial port disabled (held in Reset)

bit 6 **RX9:** 9-bit Receive Enable bit
1 = Selects 9-bit reception
0 = Selects 8-bit reception

bit 5 **SREN:** Single Receive Enable bit
Asynchronous mode:
Don't care
Synchronous mode – Master:
1 = Enables single receive
0 = Disables single receive
This bit is cleared after reception is complete.
Synchronous mode – Slave
Don't care

bit 4 **CREN:** Continuous Receive Enable bit
Asynchronous mode:
1 = Enables receiver
0 = Disables receiver
Synchronous mode:
1 = Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN)
0 = Disables continuous receive

bit 3 **ADDEN:** Address Detect Enable bit
Asynchronous mode 9-bit (RX9 = 1):
1 = Enables address detection, enable interrupt and load the receive buffer when RSR<8> is set
0 = Disables address detection, all bytes are received and ninth bit can be used as parity bit
Asynchronous mode 8-bit (RX9 = 0):
Don't care

bit 2 **FERR:** Framing Error bit
1 = Framing error (can be updated by reading RCREG register and receive next valid byte)
0 = No framing error

bit 1 **OERR:** Overrun Error bit
1 = Overrun error (can be cleared by clearing bit CREN)
0 = No overrun error

bit 0 **RX9D:** Ninth bit of Received Data
This can be address/data bit or a parity bit and must be calculated by user firmware.

TABLE 31-17: 5-BIT DIGITAL-TO-ANALOG CONVERTER (DAC) SPECIFICATIONS

Operating Conditions: V_{DD} = 3V, Temperature = 25°C (unless otherwise stated).							
Param No.	**Sym.**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Comments**
DAC01*	C_{LSB}5	Step Size	—	V_{DD}/32	—	V	
DAC02*	C_{ACC}5	Absolute Accuracy	—	—	± 1/2	LSb	
DAC03*	C_R5	Unit Resistor Value (R)	—	5K	—	Ω	
DAC04*	C_{ST}5	Settling Time(2)	—	—	10	μs	

* These parameters are characterized but not tested.

Note 1: See **Section 32.0 "DC and AC Characteristics Graphs and Charts"** for operating characterization.

2: Settling time measured while DACR<7:0> transitions from '00000' to '01111'.

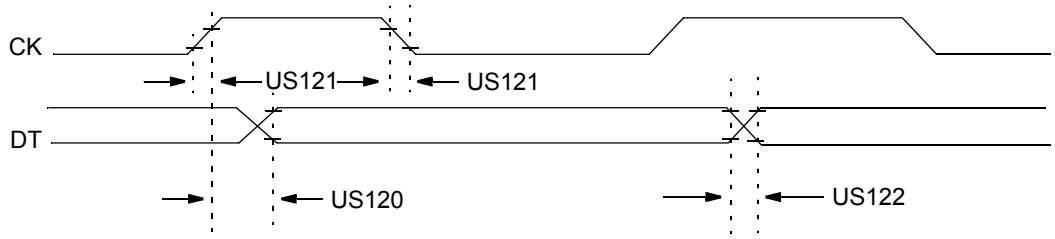
TABLE 31-18: 8-BIT DIGITAL-TO-ANALOG CONVERTER (DAC) SPECIFICATIONS

Operating Conditions: V_{DD} = 3V, Temperature = 25°C (unless otherwise stated).							
Param No.	**Sym.**	**Characteristics**	**Min.**	**Typ.**	**Max.**	**Units**	**Comments**
DAC05*	C_{LSB}8	Step Size	—	V_{DD}/256	—	V	
DAC06*	C_{ACC}8	Absolute Accuracy	—	—	± 1.5	LSb	
DAC07*	C_R8	Unit Resistor Value (R)	—	600	—	Ω	
DAC08*	C_{ST}8	Settling Time(1)	—	—	10	μs	

* These parameters are characterized but not tested.

Note 1: Settling time measured while DACR<7:0> transitions from '0x00' to '0xFF'.

FIGURE 31-14: EUSART SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING



Note: Refer to Figure 31-4 for load conditions.

TABLE 31-19: EUSART SYNCHRONOUS TRANSMISSION REQUIREMENTS

Standard Operating Conditions (unless otherwise stated)							
Param. No.	**Symbol**	**Characteristic**		**Min.**	**Max.**	**Units**	**Conditions**
US120	$T_{CK}H2_{DT}V$	SYNC XMIT (Master and Slave) Clock high to data-out valid	3.0-5.5V	—	80	ns	
			1.8-5.5V	—	100	ns	
US121	T_{CKRF}	Clock out rise time and fall time (Master mode)	3.0-5.5V	—	45	ns	
			1.8-5.5V	—	50	ns	
US122	T_{DTRF}	Data-out rise time and fall time	3.0-5.5V	—	45	ns	
			1.8-5.5V	—	50	ns	

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

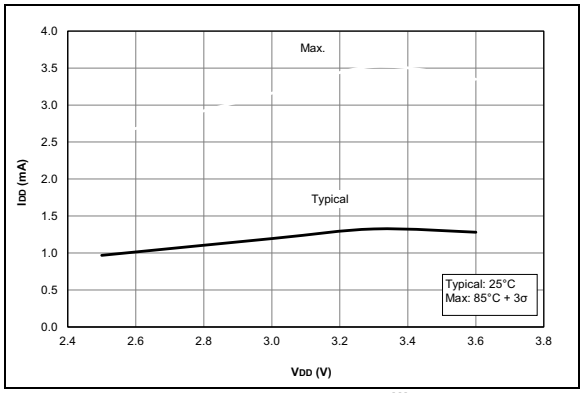


FIGURE 32-31: *IDD, HS Oscillator, 32 MHz (8 MHz + 4x PLL), PIC16LF1788/9 Only.*

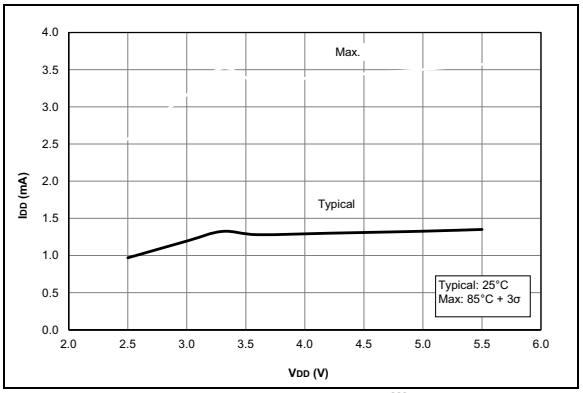


FIGURE 32-32: *IDD, HS Oscillator, 32 MHz (8 MHz + 4x PLL), PIC16F1788/9 Only.*

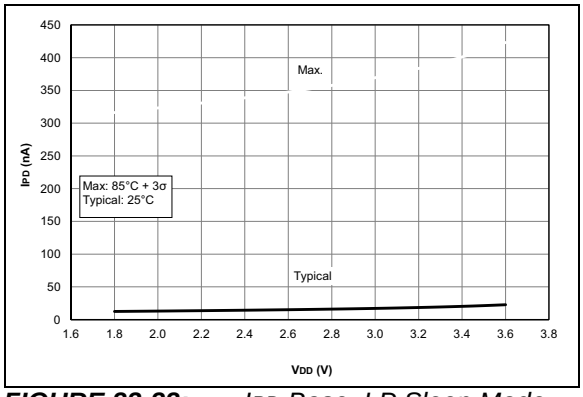


FIGURE 32-33: *IPD Base, LP Sleep Mode, PIC16LF1788/9 Only.*

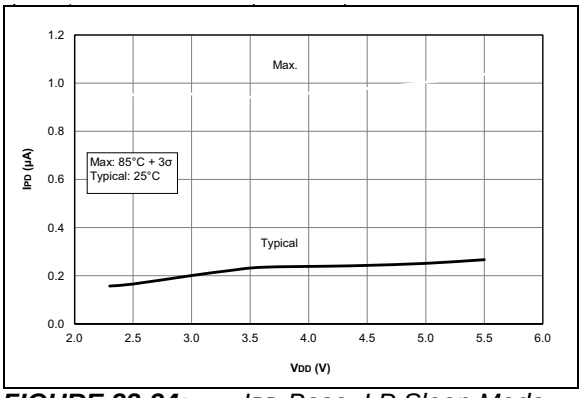


FIGURE 32-34: *IPD Base, LP Sleep Mode (VREGPM = 1), PIC16F1788/9 Only.*

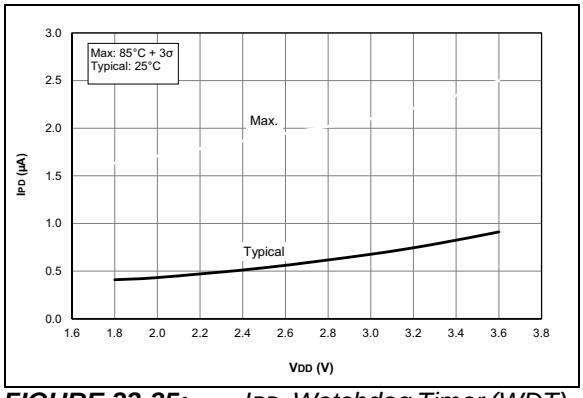


FIGURE 32-35: *IPD, Watchdog Timer (WDT), PIC16LF1788/9 Only.*

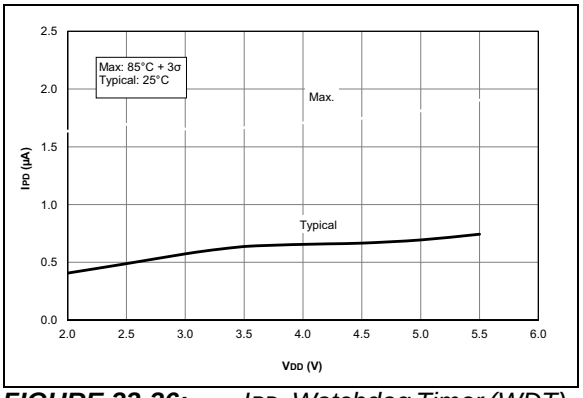


FIGURE 32-36: *IPD, Watchdog Timer (WDT), PIC16F1788/9 Only.*

Note: Unless otherwise noted, V_{IN} = 5V, F_{OSC} = 300 kHz, C_{IN} = 0.1 µF, T_A = 25°C.

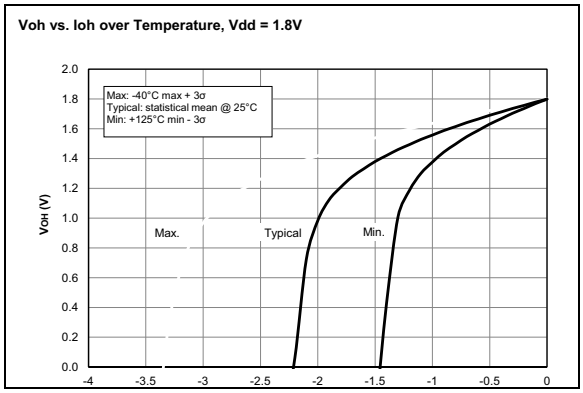


FIGURE 32-55: *V_{OH} vs. I_{OH} Over Temperature, V_{DD} = 1.8V, PIC16LF1788/9 Only.*

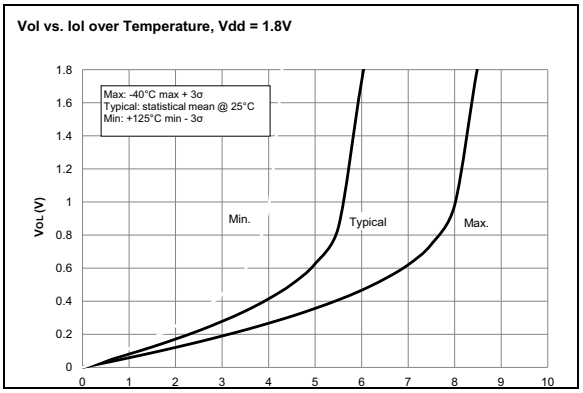


FIGURE 32-56: *V_{OL} vs. I_{OL} Over Temperature, V_{DD} = 1.8V, PIC16LF1788/9 Only.*

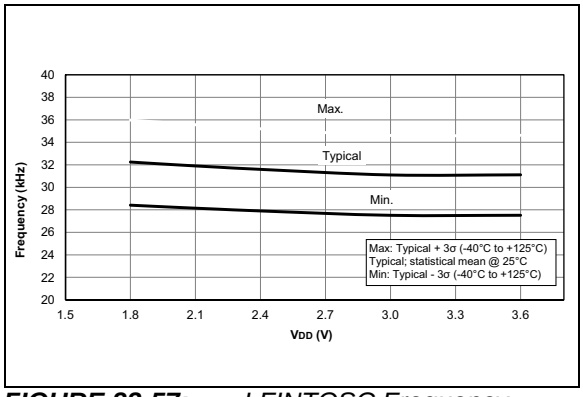


FIGURE 32-57: *LFINTOSC Frequency, PIC16LF1788/9 Only.*

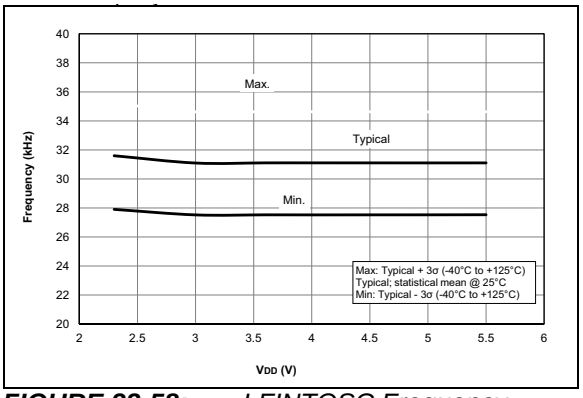


FIGURE 32-58: *LFINTOSC Frequency, PIC16F1788/9 Only.*

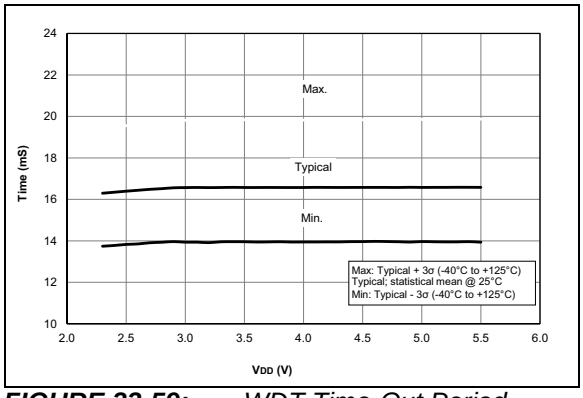


FIGURE 32-59: *WDT Time-Out Period, PIC16F1788/9 Only.*

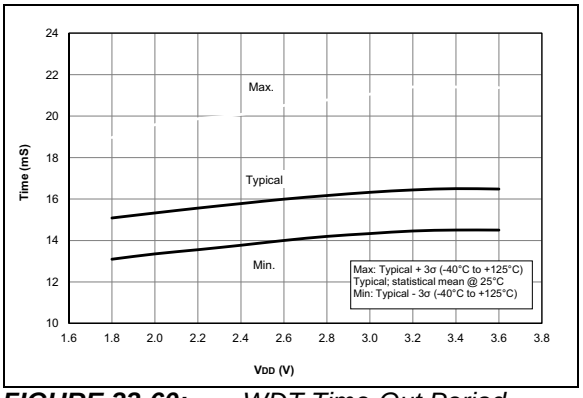


FIGURE 32-60: *WDT Time-Out Period, PIC16LF1788/9 Only.*

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

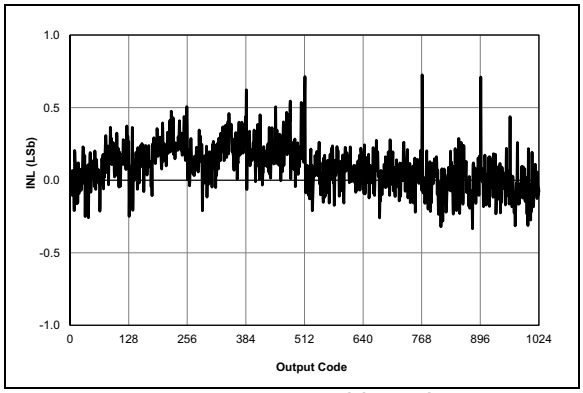


FIGURE 32-79: *ADC 10-bit Mode, Single-Ended INL, VDD = 3.0V, TAD = 1 µS, 25°C.*

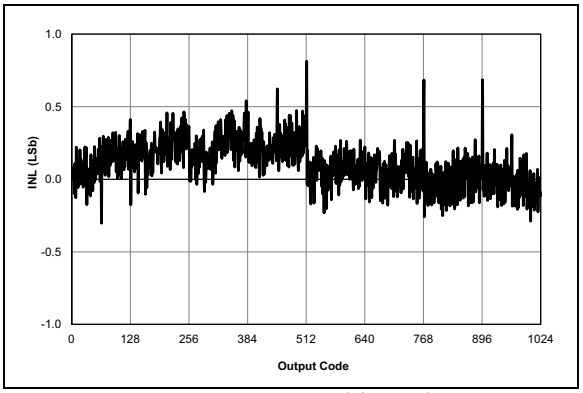


FIGURE 32-80: *ADC 10-bit Mode, Single-Ended INL, VDD = 3.0V, TAD = 4 µS, 25°C.*

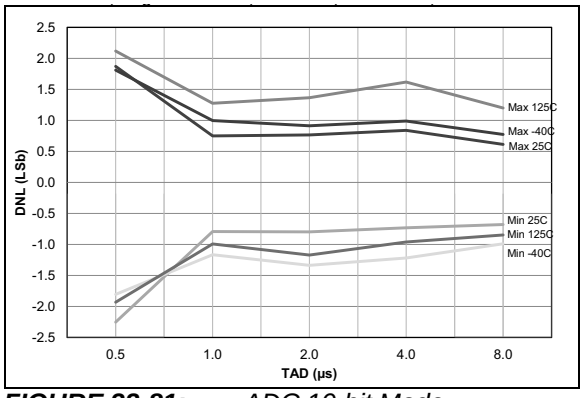


FIGURE 32-81: *ADC 10-bit Mode, Single-Ended DNL, VDD = 3.0V, VREF = 3.0V.*

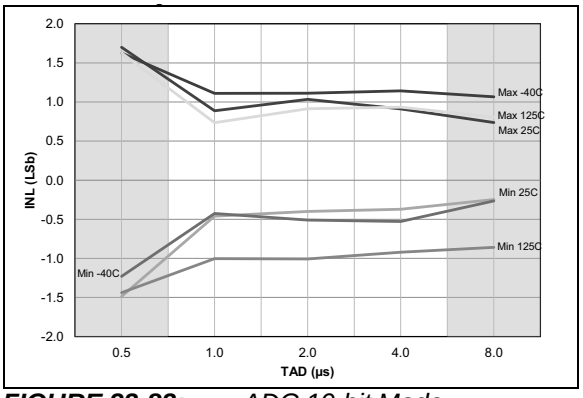


FIGURE 32-82: *ADC 10-bit Mode, Single-Ended INL, VDD = 3.0V, VREF = 3.0V.*

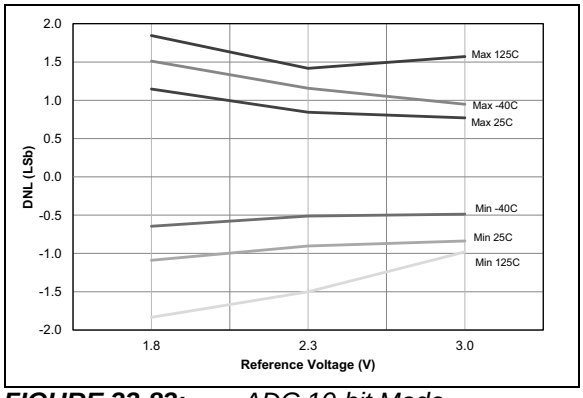


FIGURE 32-83: *ADC 10-bit Mode, Single-Ended DNL, VDD = 3.0V, TAD = 1 µS.*

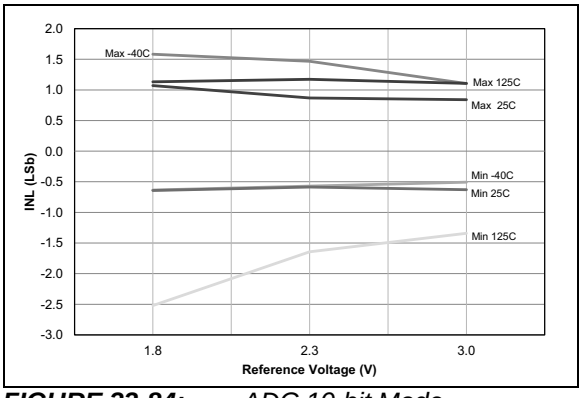


FIGURE 32-84: *ADC 10-bit Mode, Single-Ended INL, VDD = 3.0V, TAD = 1 µS.*

33.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers (MCU) and dsPIC® digital signal controllers (DSC) are supported with a full range of software and hardware development tools:

- Integrated Development Environment
  - MPLAB® X IDE Software
- Compilers/Assemblers/Linkers
  - MPLAB XC Compiler
  - MPASM™ Assembler
  - MPLINK™ Object Linker/ MPLIB™ Object Librarian
  - MPLAB Assembler/Linker/Librarian for Various Device Families
- Simulators
  - MPLAB X SIM Software Simulator
- Emulators
  - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debuggers/Programmers
  - MPLAB ICD 3
  - PICkit™ 3
- Device Programmers
  - MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards, Evaluation Kits and Starter Kits
- Third-party development tools

33.1 MPLAB X Integrated Development Environment Software

The MPLAB X IDE is a single, unified graphical user interface for Microchip and third-party software, and hardware development tool that runs on Windows®, Linux and Mac OS® X. Based on the NetBeans IDE, MPLAB X IDE is an entirely new IDE with a host of free software components and plug-ins for high-performance application development and debugging. Moving between tools and upgrading from software simulators to hardware debugging and programming tools is simple with the seamless user interface.

With complete project management, visual call graphs, a configurable watch window and a feature-rich editor that includes code completion and context menus, MPLAB X IDE is flexible and friendly enough for new users. With the ability to support multiple tools on multiple projects with simultaneous debugging, MPLAB X IDE is also suitable for the needs of experienced users.

Feature-Rich Editor:

- Color syntax highlighting
- Smart code completion makes suggestions and provides hints as you type
- Automatic code formatting based on user-defined rules
- Live parsing

User-Friendly, Customizable Interface:

- Fully customizable interface: toolbars, toolbar buttons, windows, window placement, etc.
- Call graph window

Project-Based Workspaces:

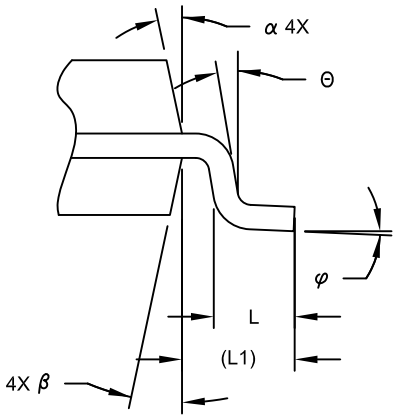
- Multiple projects
- Multiple tools
- Multiple configurations
- Simultaneous debugging sessions

File History and Bug Tracking:

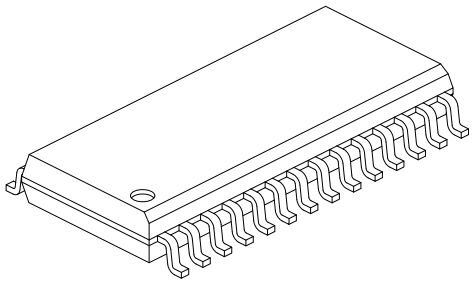
- Local file history feature
- Built-in support for Bugzilla issue tracker

28-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

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



VIEW C



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		28	
Pitch	e		1.27 BSC	
Overall Height	A	-	-	2.65
Molded Package Thickness	A2	2.05	-	-
Standoff §	A1	0.10	-	0.30
Overall Width	E		10.30 BSC	
Molded Package Width	E1		7.50 BSC	
Overall Length	D		17.90 BSC	
Chamfer (Optional)	h	0.25	-	0.75
Foot Length	L	0.40	-	1.27
Footprint	L1		1.40 REF	
Lead Angle	Θ	0°	-	-
Foot Angle	φ	0°	-	8°
Lead Thickness	c	0.18	-	0.33
Lead Width	b	0.31	-	0.51
Mold Draft Angle Top	α	5°	-	15°
Mold Draft Angle Bottom	β	5°	-	15°

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic
3. Dimension D does not include mold flash, protrusions or gate burrs, which shall not exceed 0.15 mm per end. Dimension E1 does not include interlead flash or protrusion, which shall not exceed 0.25 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M

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

   REF: Reference Dimension, usually without tolerance, for information purposes only.
5. Datums A & B to be determined at Datum H.

Microchip Technology Drawing C04-052C Sheet 2 of 2