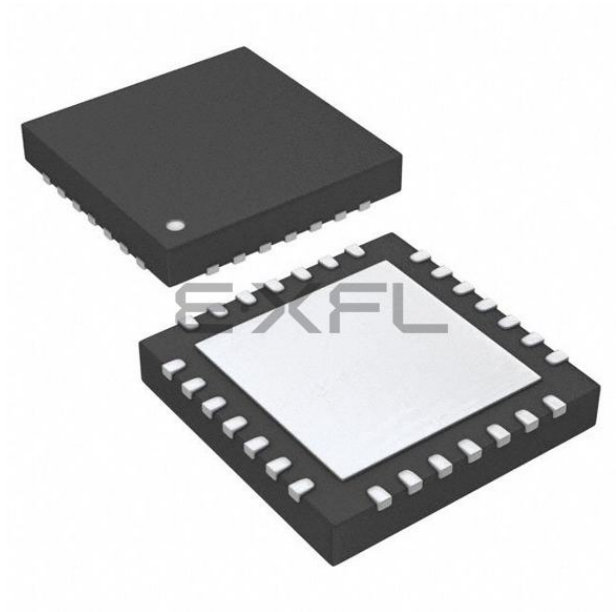


Welcome to E-XFL.COM

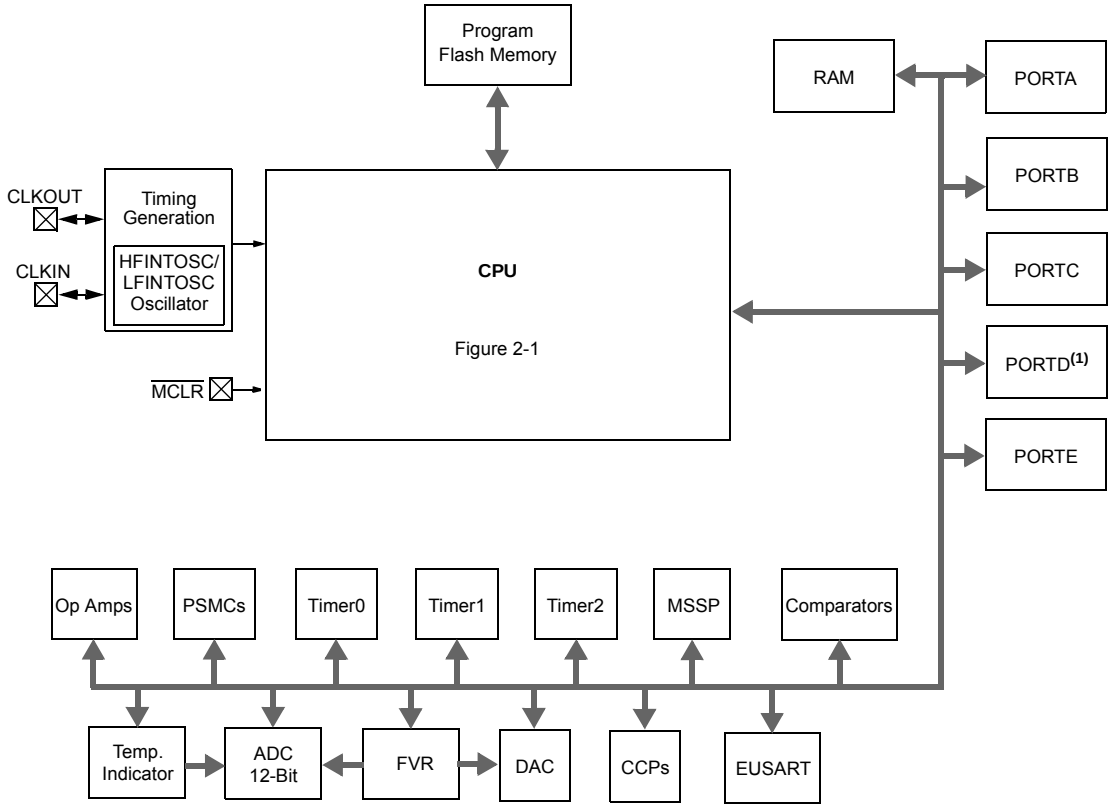
What is "Embedded - Microcontrollers"?

"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"

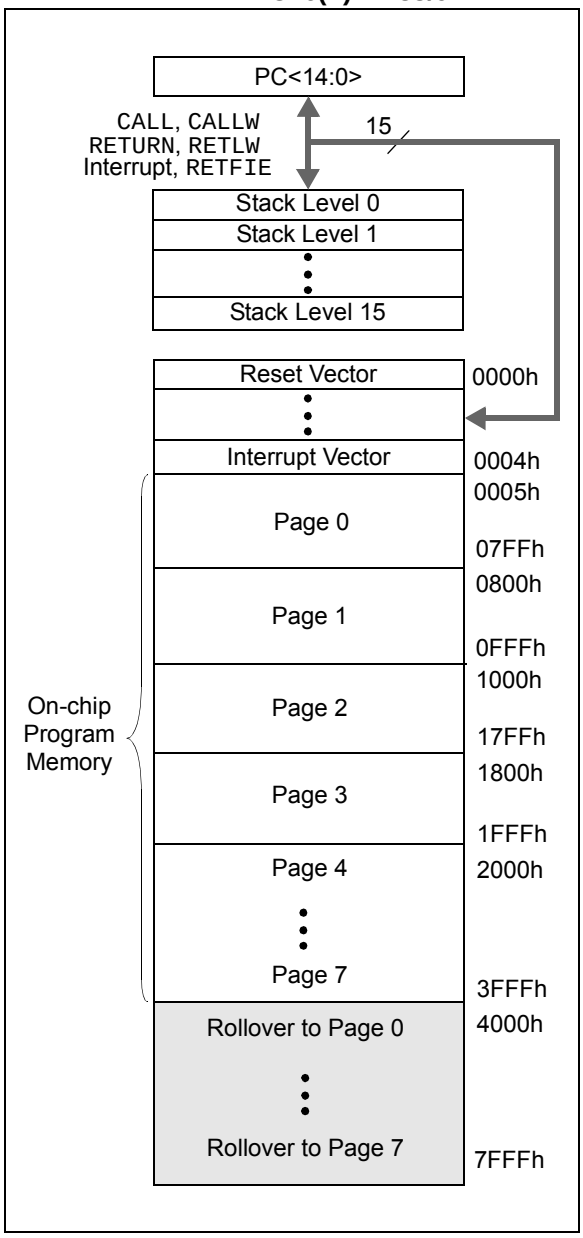
Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	32MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PSMC, PWM, WDT
Number of I/O	24
Program Memory Size	28KB (16K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	2K x 8
Voltage - 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-VQFN Exposed Pad
Supplier Device Package	28-QFN (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf1788-i-ml

FIGURE 1-1: PIC16(L)F1788/9 BLOCK DIAGRAM



Note 1: PIC16(L)F1789 only.
2: See applicable chapters for more information on peripherals.

FIGURE 3-1: PROGRAM MEMORY MAP AND STACK FOR PIC16(L)F1788/9



3.1.1 READING PROGRAM MEMORY AS DATA

There are two methods of accessing constants in program memory. The first method is to use tables of RETLW instructions. The second method is to set an FSR to point to the program memory.

3.1.1.1 RETLW Instruction

The RETLW instruction can be used to provide access to tables of constants. The recommended way to create such a table is shown in Example 3-1.

EXAMPLE 3-1: RETLW INSTRUCTION

```
constants
    BRW                 ;Add Index in W to
                        ;program counter to
                        ;select data
    RETLW DATA0         ;Index0 data
    RETLW DATA1         ;Index1 data
    RETLW DATA2
    RETLW DATA3


my_function
    ;… LOTS OF CODE…
    MOVLW       DATA_INDEX
    call constants
    ;… THE CONSTANT IS IN W
```

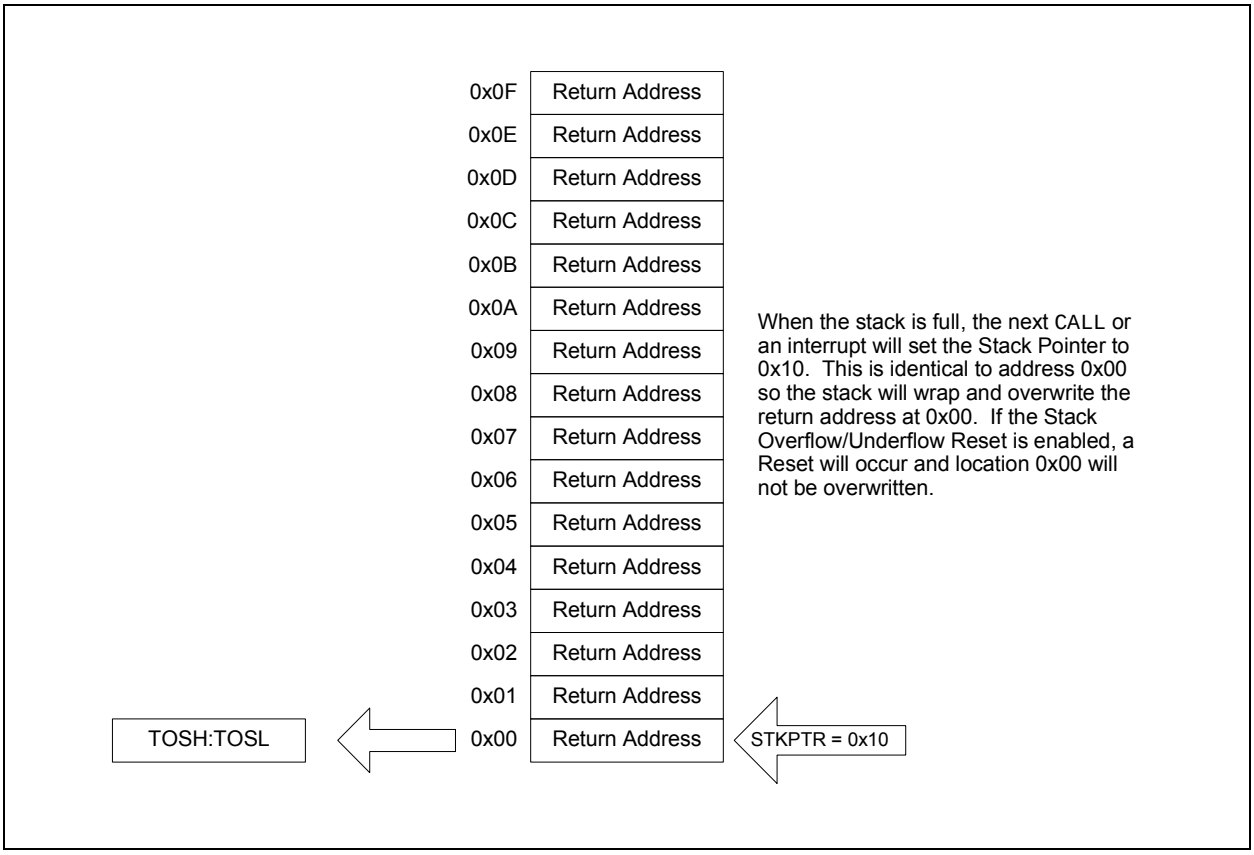
The BRW instruction makes this type of table very simple to implement. If your code must remain portable with previous generations of microcontrollers, then the BRW instruction is not available so the older table read method must be used.

3.1.1.2 Indirect Read with FSR

The program memory can be accessed as data by setting bit 7 of the FSRxH register and reading the matching INDFx register. The MOVIW instruction will place the lower eight bits of the addressed word in the W register. Writes to the program memory cannot be performed via the INDF registers. Instructions that access the program memory via the FSR require one extra instruction cycle to complete. Example 3-2 demonstrates accessing the program memory via an FSR.

The high directive will set bit<7> if a label points to a location in program memory.

FIGURE 3-7: ACCESSING THE STACK EXAMPLE 4



3.5.2 OVERFLOW/UNDERFLOW RESET

If the STVREN bit in Configuration Words is programmed to '1', the device will be reset if the stack is PUSHed beyond the sixteenth level or POPed beyond the first level, setting the appropriate bits (STKOVF or STKUNF, respectively) in the PCON register.

3.6 Indirect Addressing

The INDFn registers are not physical registers. Any instruction that accesses an INDFn register actually accesses the register at the address specified by the File Select Registers (FSR). If the FSRn address specifies one of the two INDFn registers, the read will return '0' and the write will not occur (though Status bits may be affected). The FSRn register value is created by the pair FSRnH and FSRnL.

The FSR registers form a 16-bit address that allows an addressing space with 65536 locations. These locations are divided into three memory regions:

- Traditional Data Memory
- Linear Data Memory
- Program Flash Memory

4.6 Device ID and Revision ID

The memory location 8006h is where the Device ID and Revision ID are stored. The upper nine bits hold the Device ID. The lower five bits hold the Revision ID. See **Section 12.5 "User ID, Device ID and Configuration Word Access"** for more information on accessing these memory locations.

Development tools, such as device programmers and debuggers, may be used to read the Device ID and Revision ID.

4.7 Register Definitions: Device and Revision

REGISTER 4-3: DEVID: DEVICE ID REGISTER

R	R	R	R	R	R
DEV<13:8>					
bit 13					bit 8

R	R	R	R	R	R	R	R
DEV<7:0>							
bit 7							bit 0

Legend:		
R = Readable bit		
'1' = Bit is set	'0' = Bit is cleared	

bit 13-0 **DEV<13:0>:** Device ID bits

Device	DEVID<13:0> Values
PIC16F1788	11 0000 0010 1011 (302Bh)
PIC16LF1788	11 0000 0010 1101 (302Dh)
PIC16F1789	11 0000 0010 1010 (302Ah)
PIC16LF1789	11 0000 0010 1100 (302Ch)

REGISTER 4-4: REVID: REVISION ID REGISTER

R	R	R	R	R	R
REV<13:8>					
bit 13					bit 8

R	R	R	R	R	R	R	R
REV<7:0>							
bit 7							bit 0

Legend:		
R = Readable bit		
'1' = Bit is set	'0' = Bit is cleared	

bit 13-0 **REV<13:0>:** Revision ID bits

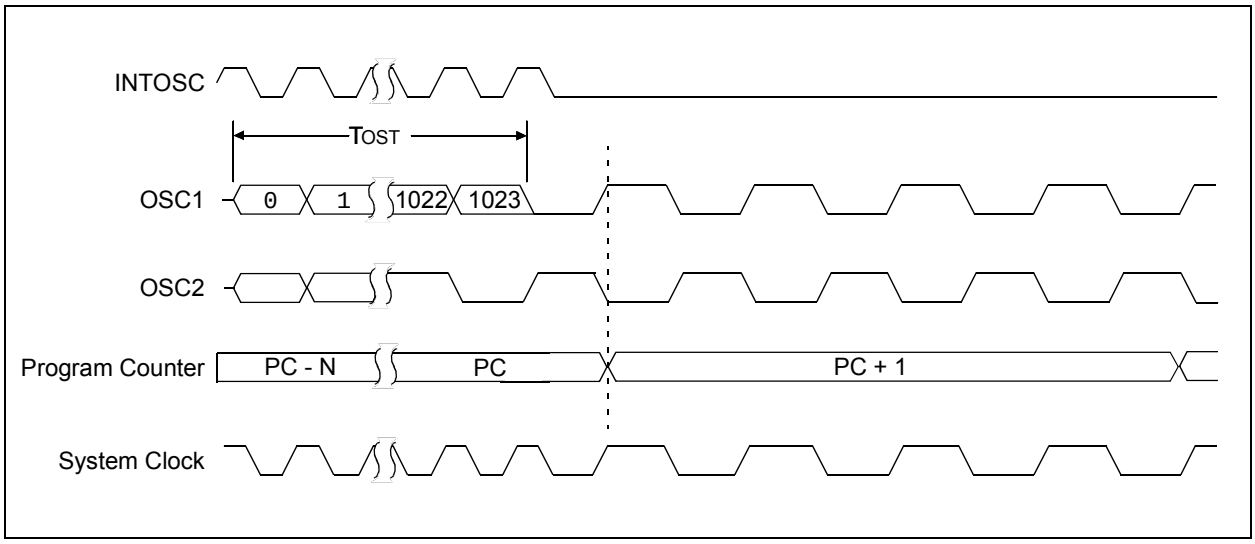
6.4.2 TWO-SPEED START-UP SEQUENCE

1. Wake-up from Power-on Reset or Sleep.
2. Instructions begin execution by the internal oscillator at the frequency set in the IRCF<3:0> bits of the OSCCON register.
3. OST enabled to count 1024 clock cycles.
4. OST timed out, wait for falling edge of the internal oscillator.
5. OSTS is set.
6. System clock held low until the next falling edge of new clock (LP, XT or HS mode).
7. System clock is switched to external clock source.

6.4.3 CHECKING TWO-SPEED CLOCK STATUS

Checking the state of the OSTS bit of the OSCSTAT register will confirm if the microcontroller is running from the external clock source, as defined by the FOSC<2:0> bits in the Configuration Words, or the internal oscillator.

FIGURE 6-8: TWO-SPEED START-UP



REGISTER 6-2: OSCSTAT: OSCILLATOR STATUS REGISTER

R-1/q	R-0/q	R-q/q	R-0/q	R-0/q	R-q/q	R-0/0	R-0/q
T1OSCR	PLLR	OSTS	HFIOFR	HFIOFL	MFIOFR	LFIOFR	HFIOFS
bit 7							bit 0

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

bit 7 **T1OSCR:** Timer1 Oscillator Ready bit
If T1OSCEN = 1:
1 = Timer1 oscillator is ready
0 = Timer1 oscillator is not ready
If T1OSCEN = 0:
1 = Timer1 clock source is always ready

bit 6 **PLLR** 4x PLL Ready bit
1 = 4x PLL is ready
0 = 4x PLL is not ready

bit 5 **OSTS:** Oscillator Start-up Timer Status bit
1 = Running from the clock defined by the FOSC<2:0> bits of the Configuration Words
0 = Running from an internal oscillator (FOSC<2:0> = 100)

bit 4 **HFIOFR:** High-Frequency Internal Oscillator Ready bit
1 = HFINTOSC is ready
0 = HFINTOSC is not ready

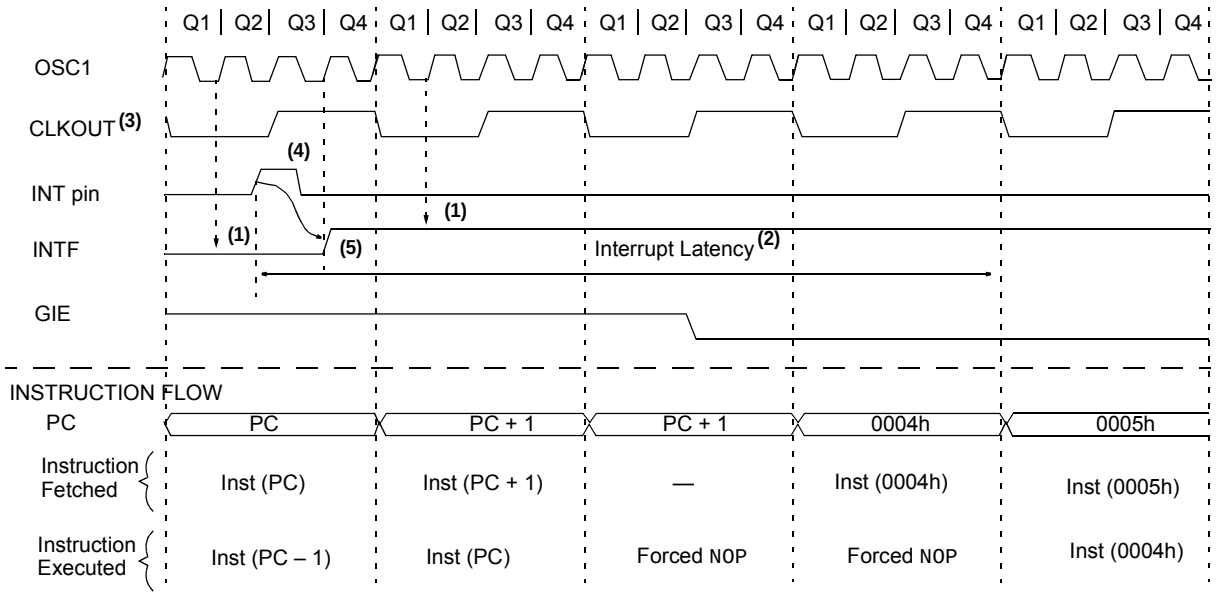
bit 3 **HFIOFL:** High-Frequency Internal Oscillator Locked bit
1 = HFINTOSC is at least 2% accurate
0 = HFINTOSC is not 2% accurate

bit 2 **MFIOFR:** Medium-Frequency Internal Oscillator Ready bit
1 = MFINTOSC is ready
0 = MFINTOSC is not ready

bit 1 **LFIOFR:** Low-Frequency Internal Oscillator Ready bit
1 = LFINTOSC is ready
0 = LFINTOSC is not ready

bit 0 **HFIOFS:** High-Frequency Internal Oscillator Stable bit
1 = HFINTOSC is at least 0.5% accurate
0 = HFINTOSC is not 0.5% accurate

FIGURE 8-3: INT PIN INTERRUPT TIMING



Note 1: INTF flag is sampled here (every Q1).

2: Asynchronous interrupt latency = 3-5 TCY. Synchronous latency = 3-4 TCY, where TCY = instruction cycle time. Latency is the same whether Inst (PC) is a single cycle or a 2-cycle instruction.

3: CLKOUT not available in all oscillator modes.

4: For minimum width of INT pulse, refer to AC specifications in **Section 31.0 "Electrical Specifications"**".

5: INTF is enabled to be set any time during the Q4-Q1 cycles.

15.0 FIXED VOLTAGE REFERENCE (FVR)

The Fixed Voltage Reference, or FVR, is a stable voltage reference, independent of VDD, with 1.024V, 2.048V or 4.096V selectable output levels. The output of the FVR can be configured to supply a reference voltage to the following:

- ADC input channel
- ADC positive reference
- Comparator positive input
- Digital-to-Analog Converter (DAC)

The FVR can be enabled by setting the FVREN bit of the FVRCON register.

15.1 Independent Gain Amplifiers

The output of the FVR supplied to the ADC, Comparators, and DAC is routed through two independent programmable gain amplifiers. Each amplifier can be programmed for a gain of 1x, 2x or 4x, to produce the three possible voltage levels.

The ADFVR<1:0> bits of the FVRCON register are used to enable and configure the gain amplifier settings for the reference supplied to the ADC module. Reference **Section 17.0 "Analog-to-Digital Converter (ADC) Module"** for additional information.

The CDAFVR<1:0> bits of the FVRCON register are used to enable and configure the gain amplifier settings for the reference supplied to the DAC and comparator module. Reference **Section 19.0 "8-Bit Digital-to-Analog Converter (DAC) Module"** and **Section 21.0 "Comparator Module"** for additional information.

15.2 FVR Stabilization Period

When the Fixed Voltage Reference module is enabled, it requires time for the reference and amplifier circuits to stabilize. Once the circuits stabilize and are ready for use, the FVRRDY bit of the FVRCON register will be set. See **Section 31.0 "Electrical Specifications"** for the minimum delay requirement.

15.3 FVR Buffer Stabilization Period

When either FVR Buffer1 or FVR Buffer 2 is enabled, the buffer amplifier circuits require 30 μs to stabilize. This stabilization time is required even when the FVR is already operating and stable.

25.3.6 PWM RESOLUTION

The resolution determines the number of available duty cycles for a given period. For example, a 10-bit resolution will result in 1024 discrete duty cycles, whereas an 8-bit resolution will result in 256 discrete duty cycles.

The maximum PWM resolution is ten bits when PR2 is 255. The resolution is a function of the PR2 register value as shown by Equation 25-4.

EQUATION 25-4: PWM RESOLUTION

$$Resolution = \frac{log[4(PR2 + 1)]}{log(2)} \ bits$$

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

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

PWM Frequency	1.22 kHz	4.88 kHz	19.53 kHz	78.12 kHz	156.3 kHz	208.3 kHz
Timer Prescale	16	4	1	1	1	1
PR2 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	1.22 kHz	4.90 kHz	19.61 kHz	76.92 kHz	153.85 kHz	200.0 kHz
Timer Prescale	16	4	1	1	1	1
PR2 Value	0x65	0x65	0x65	0x19	0x0C	0x09
Maximum Resolution (bits)	8	8	8	6	5	5

25.3.7 OPERATION IN SLEEP MODE

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

25.3.8 CHANGES IN SYSTEM CLOCK FREQUENCY

The PWM frequency is derived from the system clock frequency. Any changes in the system clock frequency will result in changes to the PWM frequency. See **Section 6.0 "Oscillator Module (with Fail-Safe Clock Monitor)"** for additional details.

25.3.9 EFFECTS OF RESET

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

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.

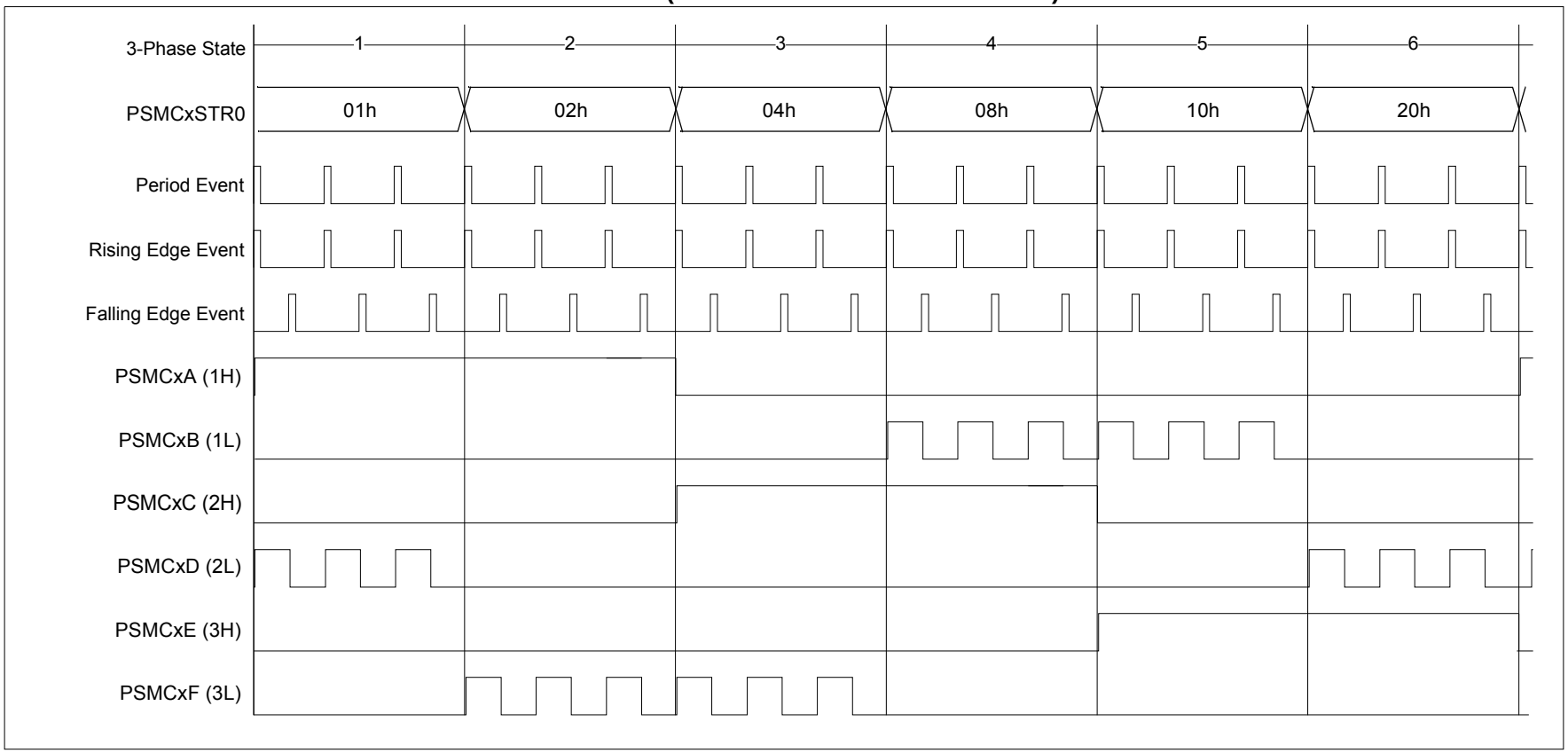
The rising edge blanking time is set with the PSMC Rising Edge Blanking Time (PSMCxBLKR) register (Register 26-30). The inputs to be blanked are selected with the PSMC Rising Edge Blanked Source (PSMCxREBS) register (Register 26-11). During rising edge blanking, the selected blanked sources are suppressed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

FIGURE 26-15: 3-PHASE PWM STEERING WAVEFORM (PXHSMEN = 0 AND PXLSMEN = 1)



REGISTER 26-13: PSMCxPHS: PSMC PHASE SOURCE REGISTER(1)

R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
PxPHSIN	—	—	PxPHSC4	PxPHSC3	PxPHSC2	PxPHSC1	PxPHST
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 **PxPHSIN:** PSMCx Rising Edge Event occurs on PSMCxIN pin
1 = Rising edge event will occur when PSMCxIN pin goes true
0 = PSMCxIN pin will not cause rising edge event

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

bit 4 **PxPHSC4:** PSMCx Rising Edge Event occurs on sync_C4OUT output
1 = Rising edge event will occur when sync_C4OUT output goes true
0 = sync_C4OUT will not cause rising edge event

bit 3 **PxPHSC3:** PSMCx Rising Edge Event occurs on sync_C3OUT output
1 = Rising edge event will occur when sync_C3OUT output goes true
0 = sync_C3OUT will not cause rising edge event

bit 2 **PxPHSC2:** PSMCx Rising Edge Event occurs on sync_C2OUT output
1 = Rising edge event will occur when sync_C2OUT output goes true
0 = sync_C2OUT will not cause rising edge event

bit 1 **PxPHSC1:** PSMCx Rising Edge Event occurs on sync_C1OUT output
1 = Rising edge event will occur when sync_C1OUT output goes true
0 = sync_C1OUT will not cause rising edge event

bit 0 **PxPHST:** PSMCx Rising Edge Event occurs on Time Base match
1 = Rising edge event will occur when PSMCxTMR = PSMCxPH
0 = Time base will not cause rising edge event

Note 1: Sources are not mutually exclusive: more than one source can cause a rising edge event.

FIGURE 27-7: SPI DAISY-CHAIN CONNECTION

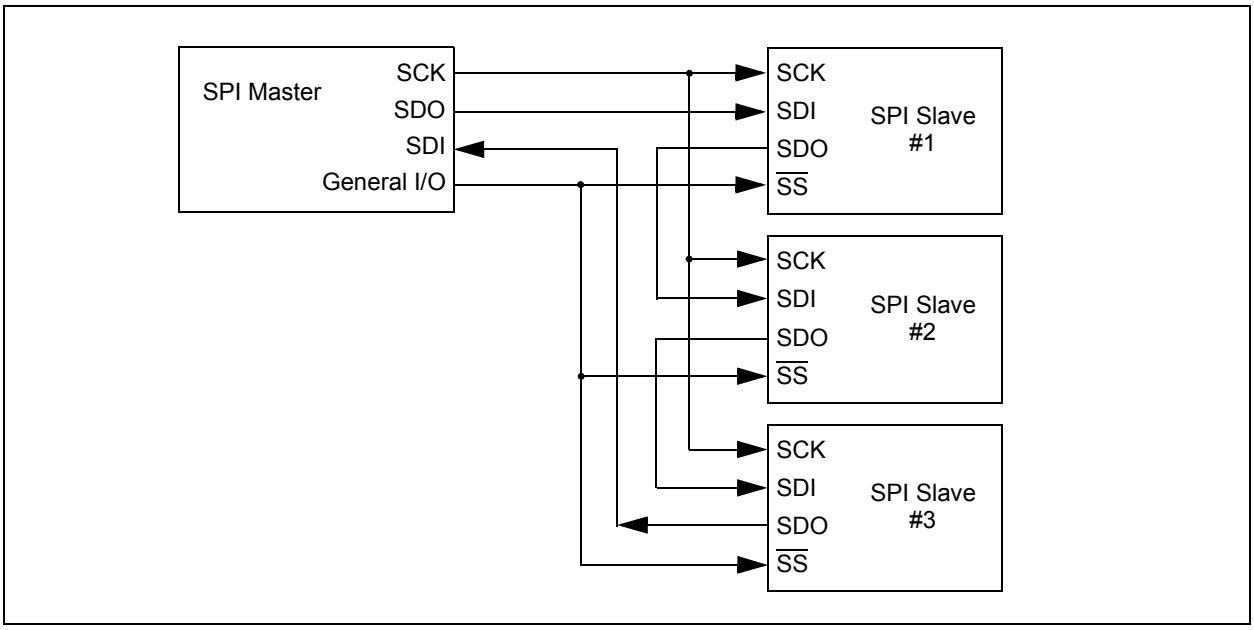


FIGURE 27-8: SLAVE SELECT SYNCHRONOUS WAVEFORM

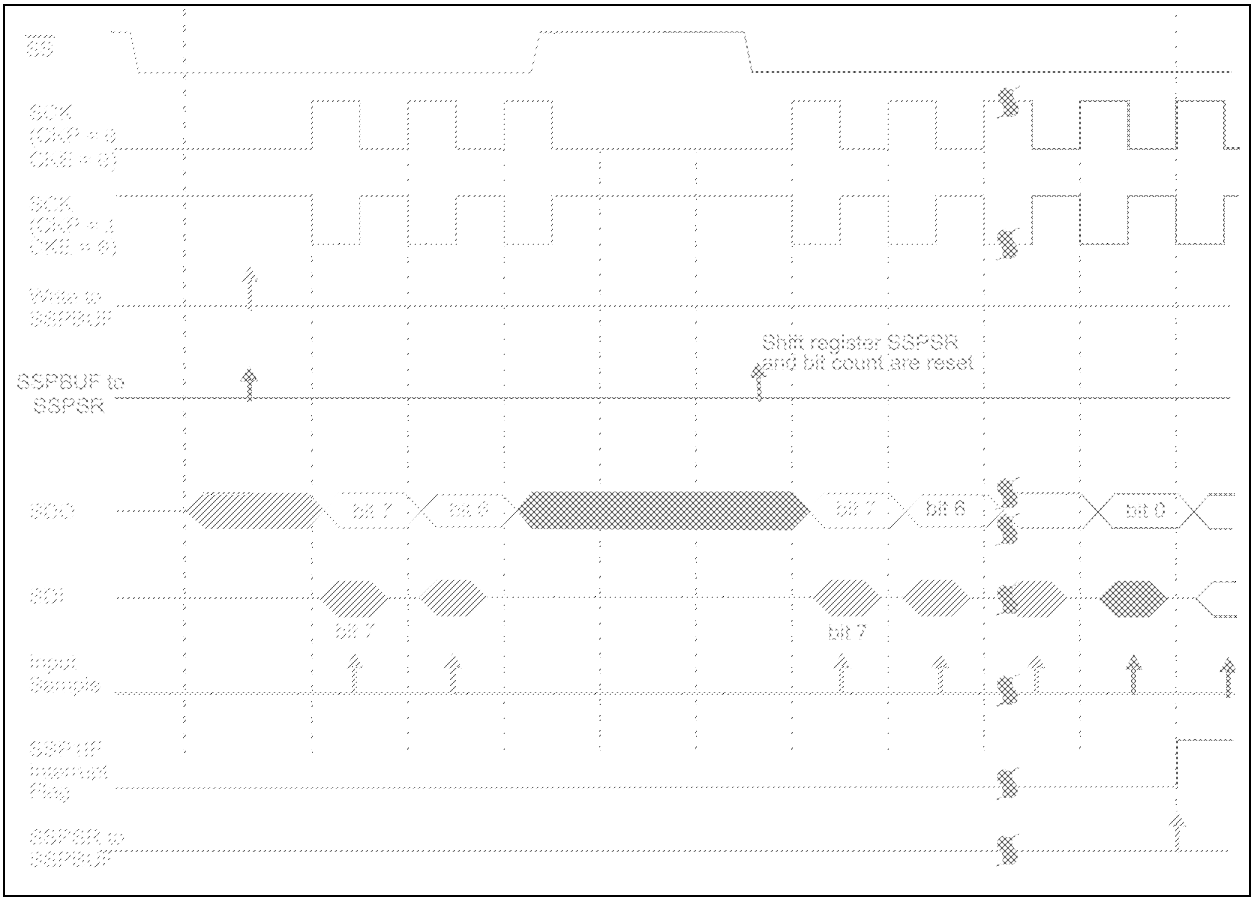


FIGURE 27-28: I²C MASTER MODE WAVEFORM (TRANSMISSION, 7 OR 10-BIT ADDRESS)

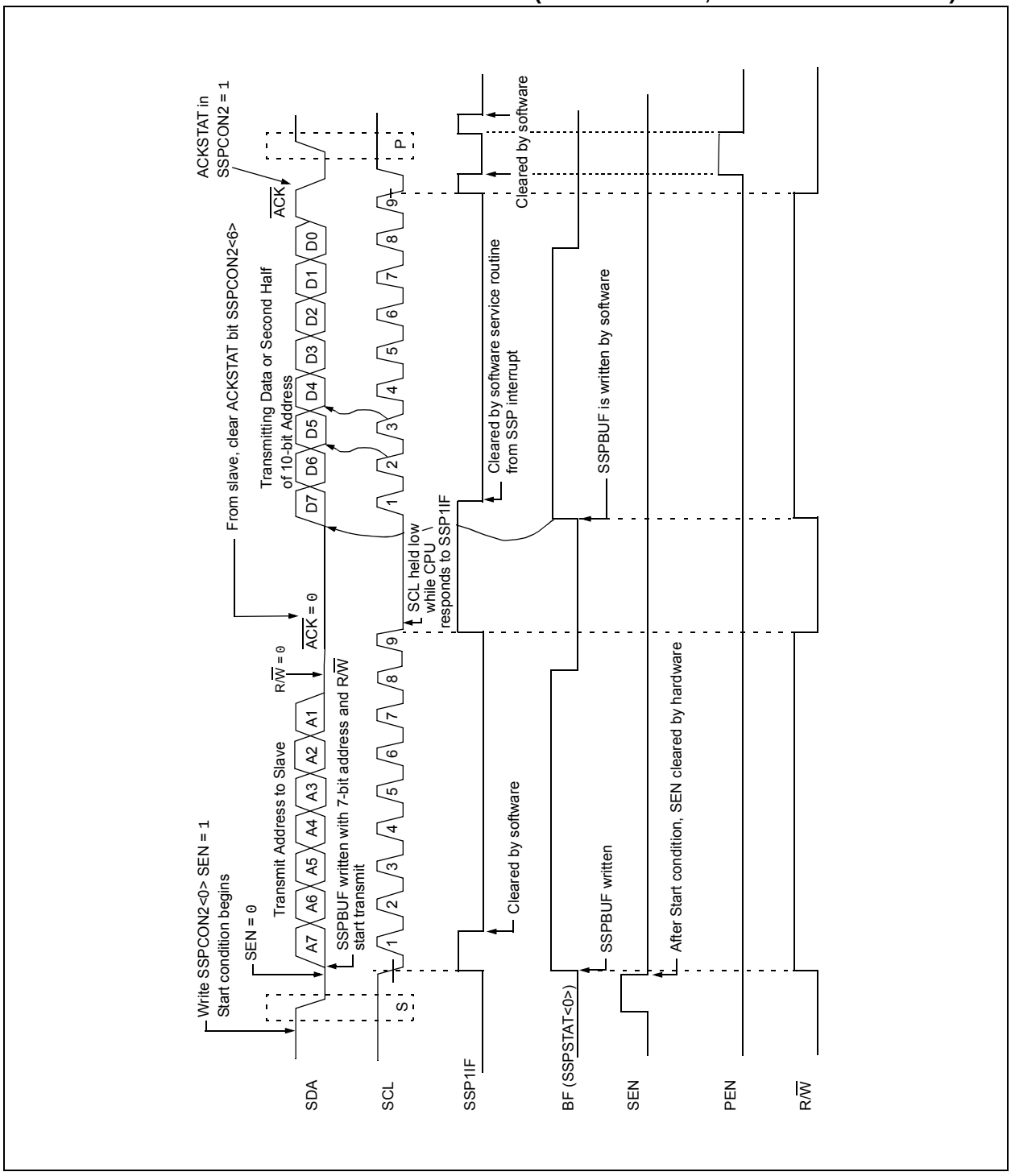


TABLE 28-2: SUMMARY OF REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
APFCON1	C2OUTSEL	CC1PSEL	SDOSEL	SCKSEL	SDISEL	TXSEL	RXSEL	CCP2SEL	132
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	356
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	97
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	98
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	102
RCREG	EUSART Receive Data Register								349*
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	355
SPBRGL	BRG<7:0>								357
SPBRGH	BRG<15:8>								357
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	147
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	354

Legend: — = unimplemented location, read as '0'. Shaded cells are not used for asynchronous reception.

* Page provides register information.

TABLE 28-5: 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	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	300.0	0.00	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	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)	Actual Rate	% Error	SPBRG value (decimal)
300	300.0	0.00	6666	300.0	0.01	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	—	—	—

MOVIW **Move INDFn to W**

Syntax: [*label*] MOVIW ++FSRn
[*label*] MOVIW --FSRn
[*label*] MOVIW FSRn++
[*label*] MOVIW FSRn--
[*label*] MOVIW k[FSRn]

Operands: n ∈ [0,1]
mm ∈ [00,01, 10, 11]
-32 ≤ k ≤ 31

Operation: INDFn → W
Effective address is determined by
- FSR + 1 (preincrement)
- FSR - 1 (predecrement)
- FSR + k (relative offset)

After the Move, the FSR value will be either:
- FSR + 1 (all increments)
- FSR - 1 (all decrements)
- Unchanged

Status Affected: Z

Mode	Syntax	mm
Preincrement	++FSRn	00
Predecrement	--FSRn	01
Postincrement	FSRn++	10
Postdecrement	FSRn--	11

Description: This instruction is used to move data between W and one of the indirect registers (INDFn). Before/ after this move, the pointer (FSRn) is updated by pre/post incrementing/decrementing it.

Note: The INDFn registers are not physical registers. Any instruction that accesses an INDFn register actually accesses the register at the address specified by the FSRn.

FSRn is limited to the range 0000h -FFFFh. Incrementing/ decrementing it beyond these bounds will cause it to wrap-around.

MOVLB **Move literal to BSR**

Syntax: [*label*] MOVLB k

Operands: 0 ≤ k ≤ 31

Operation: k → BSR

Status Affected: None

Description: The 5-bit literal 'k' is loaded into the Bank Select Register (BSR).

MOVLP **Move literal to PCLATH**

Syntax: [*label*] MOVLP k

Operands: 0 ≤ k ≤ 127

Operation: k → PCLATH

Status Affected: None

Description: The 7-bit literal 'k' is loaded into the PCLATH register.

MOVLW **Move literal to W**

Syntax: [*label*] MOVLW k

Operands: 0 ≤ k ≤ 255

Operation: k → (W)

Status Affected: None

Description: The 8-bit literal 'k' is loaded into W register. The "don't cares" will assemble as '0's.

Words: 1

Cycles: 1

Example:
```
MOVLW   0x5A
```
After Instruction
W = 0x5A

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

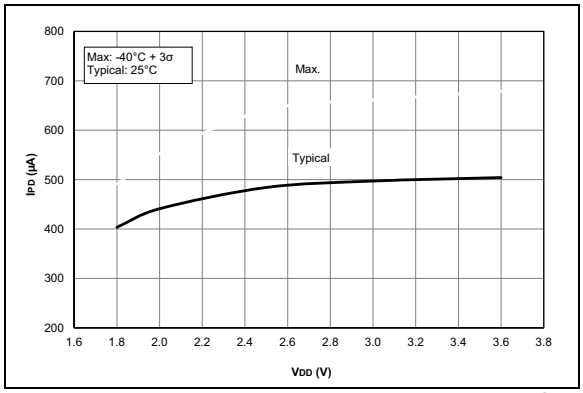


FIGURE 32-49: *IPD, Comparator, NP Mode (CxSP = 1), PIC16LF1788/9 Only.*

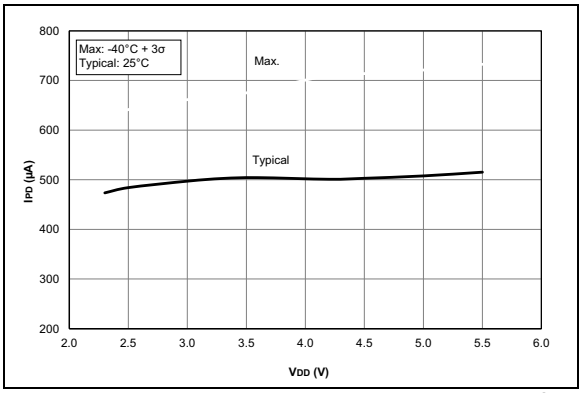


FIGURE 32-50: *IPD, Comparator, NP Mode (CxSP = 1), PIC16F1788/9 Only.*

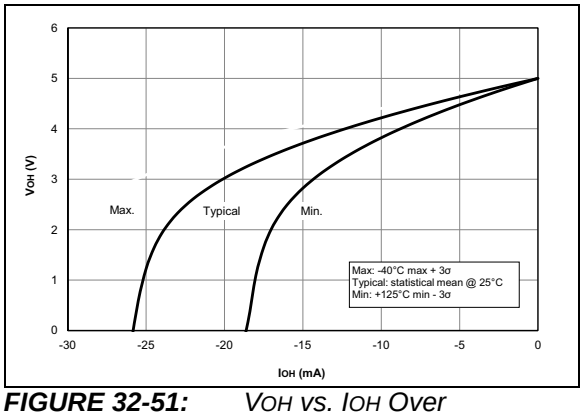


FIGURE 32-51: *VOH vs. IOH Over Temperature, VDD = 5.0V, PIC16F1788/9 Only.*

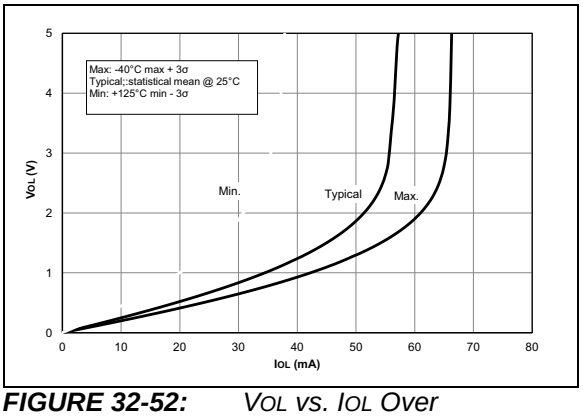


FIGURE 32-52: *VOL vs. IOL Over Temperature, VDD = 5.0V, PIC16F1788/9 Only.*

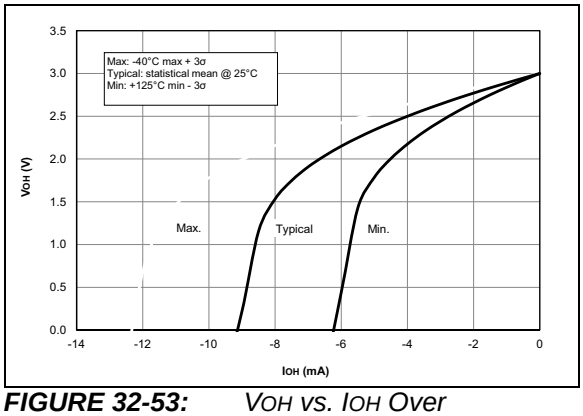


FIGURE 32-53: *VOH vs. IOH Over Temperature, VDD = 3.0V.*

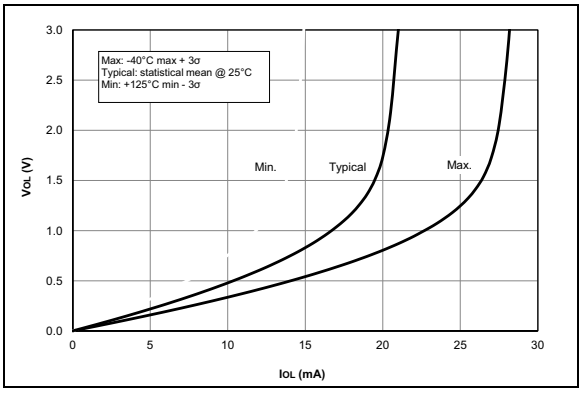


FIGURE 32-54: *VOL vs. IOL Over Temperature, VDD = 3.0V.*

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

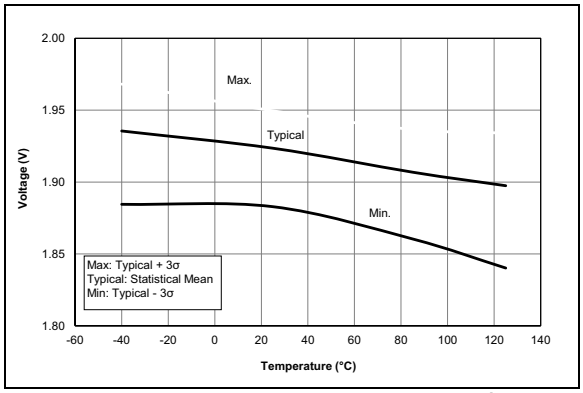


FIGURE 32-61: *Brown-Out Reset Voltage, Low Trip Point (BORV = 1), PIC16LF1788/9 Only.*

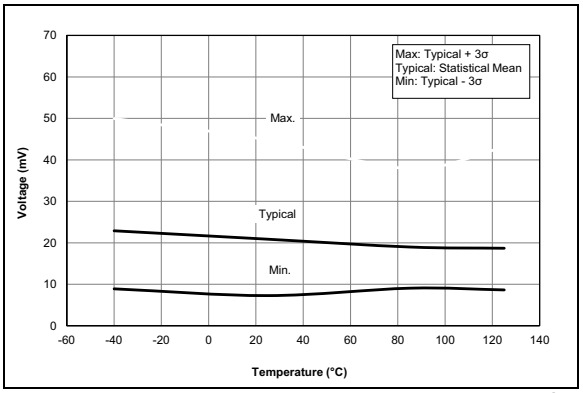


FIGURE 32-62: *Brown-Out Reset Hysteresis, Low Trip Point (BORV = 1), PIC16LF1788/9 Only.*

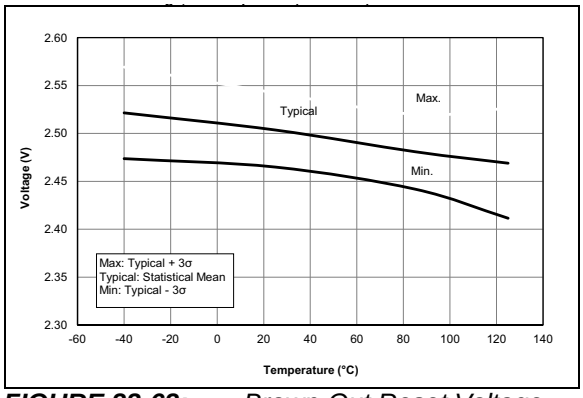


FIGURE 32-63: *Brown-Out Reset Voltage, Low Trip Point (BORV = 1), PIC16F1788/9 Only.*

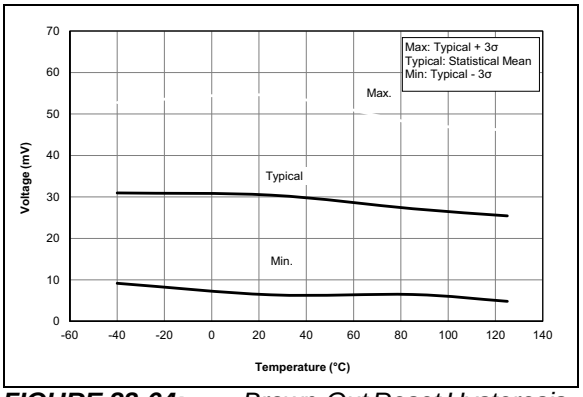


FIGURE 32-64: *Brown-Out Reset Hysteresis, Low Trip Point (BORV = 1), PIC16F1788/9 Only.*

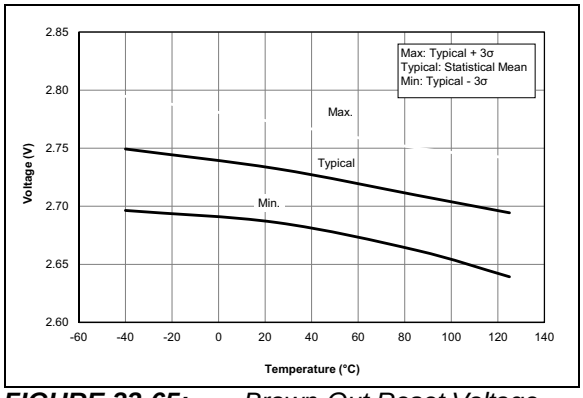


FIGURE 32-65: *Brown-Out Reset Voltage, High Trip Point (BORV = 0).*

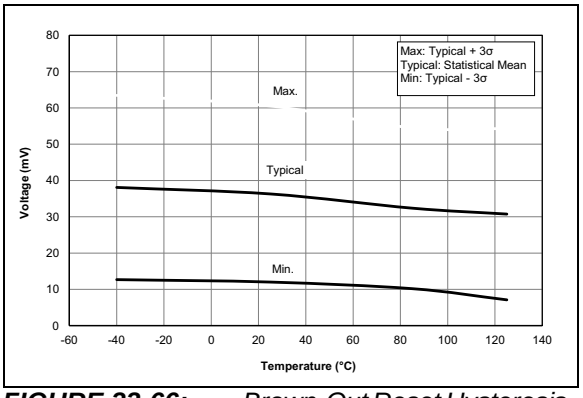


FIGURE 32-66: *Brown-Out Reset Hysteresis, High Trip Point (BORV = 0).*

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

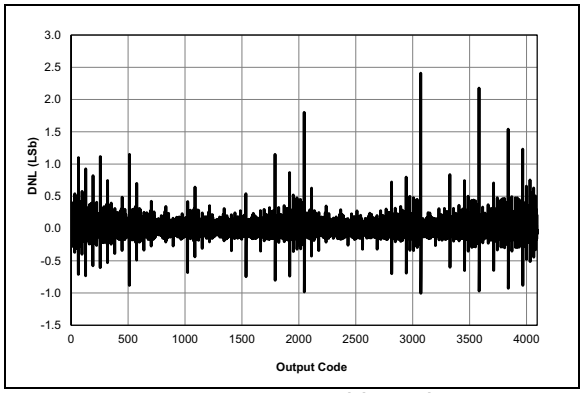


FIGURE 32-85: *ADC 12-bit Mode, Single-Ended DNL, VDD = 3.0V, TAD = 1 µS, 25°C.*

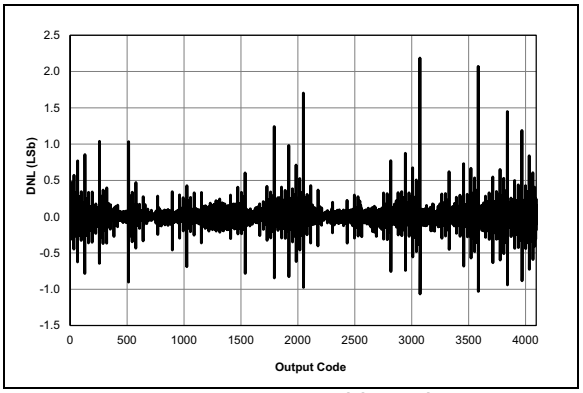


FIGURE 32-86: *ADC 12-bit Mode, Single-Ended DNL, VDD = 3.0V, TAD = 4 µS, 25°C.*

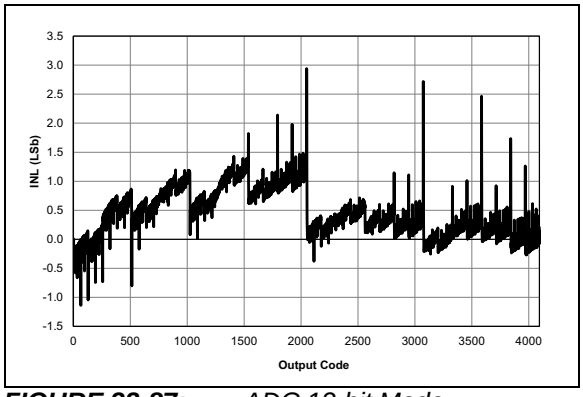


FIGURE 32-87: *ADC 12-bit Mode, Single-Ended INL, VDD = 3.0V, TAD = 1 µS, 25°C.*

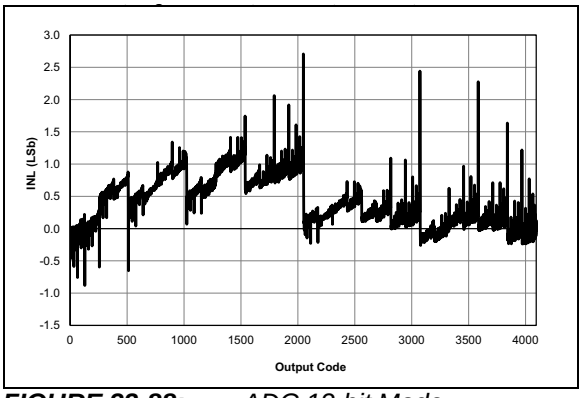


FIGURE 32-88: *ADC 12-bit Mode, Single-Ended INL, VDD = 3.0V, TAD = 4 µS, 25°C.*

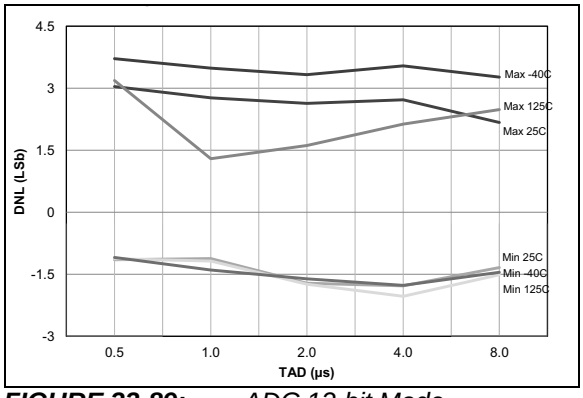


FIGURE 32-89: *ADC 12-bit Mode, Single-Ended DNL, VDD = 3.0V, VREF = 3.0V.*

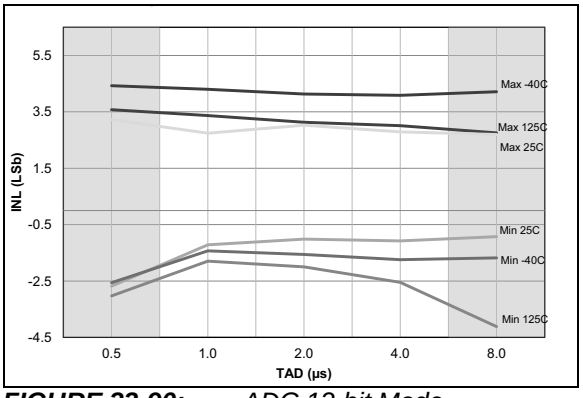


FIGURE 32-90: *ADC 12-bit Mode, Single-Ended INL, VDD = 3.0V, VREF = 3.0V.*