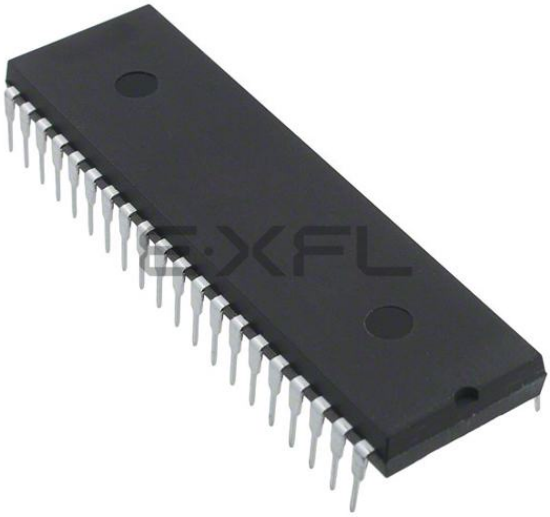


Welcome to E-XFL.COM

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	36
Program Memory Size	14KB (8K x 14)
Program Memory Type	FLASH
EEPROM Size	128 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 28x10b; D/A 4x5b, 4x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	40-DIP (0.600", 15.24mm)
Supplier Device Package	40-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f1777-i-p

3.0 MEMORY ORGANIZATION

These devices contain the following types of memory:

- Program Memory
  - Configuration Words
  - Device ID
  - User ID
  - Flash Program Memory
- Data Memory
  - Core Registers
  - Special Function Registers
  - General Purpose RAM
  - Common RAM

Note 1: The method to access Flash memory through the PMCON registers is described in **Section 10.0 "Flash Program Memory Control"**.

The following features are associated with access and control of program memory and data memory:

- PCL and PCLATH
- Stack
- Indirect Addressing

3.1 Program Memory Organization

The enhanced mid-range core has a 15-bit program counter capable of addressing a 32K x 14 program memory space. Table 3-1 shows the memory sizes implemented for the PIC16(L)F1777/8/9 family. Accessing a location above these boundaries will cause a wrap-around within the implemented memory space. The Reset vector is at 0000h and the interrupt vector is at 0004h (see Figures 3-1 and 3-2).

3.2 High-Endurance Flash

This device has a 128-byte section of high-endurance program Flash memory (PFM) in lieu of data EEPROM. This area is especially well suited for nonvolatile data storage that is expected to be updated frequently over the life of the end product. See **Section 10.2 "Flash Program Memory Overview"** for more information on writing data to PFM. See **Section 3.2.1.2 "Indirect Read with FSR"** for more information about using the FSR registers to read byte data stored in PFM.

TABLE 3-1: DEVICE SIZES AND ADDRESSES

Device	Program Memory Space (Words)	Last Program Memory Address	High-Endurance Flash Memory Address Range(1)
PIC16(L)F1777	8,192	1FFFh	1F80h-1FFFh
PIC16(L)F1778/9	16,384	3FFFh	3F80h-3FFFh

Note 1: High-endurance Flash applies to the low byte of each address in the range.

TABLE 3-18: 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 12											
60Ch to 613h	—	Unimplemented								—	—
60Ch	DAC8CON0(3)	EN	—	OE1	OE2	PSS<1:0>		NSS1	NSS0	0-00 0000	0-00 0000
60Dh	DAC8REF(3)	—	—	—	REF<4:0>					---0 0000	0000 0000
60Eh	PRG4RTSS(3)	—	—	—	—	RTSS<3:0>				---- 0000	---- 0000
60Fh	PRG4FTSS(3)	—	—	—	—	FTSS<3:0>				---- 0000	---- 0000
610h	PRG4INS(3)	—	—	—	—	INS<3:0>				---- 0000	---- 0000
611h	PRG4CON0(3)	EN	—	FEDG	REDG	MODE<1:0>		OS	GO	0-000 0000	0-00 0000
612h	PRG4CON1(3)	—	—	—	—	—	RDY	FPOL	RPOL	---- -000	---- -000
613h	PRG4CON2(3)	—	—	—	ISET<4:0>					---0 0000	---0 0000
614h	PWM3DCL	DC<1:0>		—	—	—	—	—	—	xx-- ----	uu-- ----
615h	PWM3DCH	DC<9:2>								xxxx xxxx	uuuu uuuu
616h	PWM3CON	EN	—	OUT	POL	—	—	—	—	0-00 ----	0-00 ----
617h	PWM4DCL	DC<1:0>		—	—	—	—	—	—	xx-- ----	uu-- ----
618h	PWM4DCH	DC<9:2>								xxxx xxxx	uuuu uuuu
619h	PWM4CON	EN	—	OUT	POL	—	—	—	—	0-00 ----	0-00 ----
61Ah	PWM9DCL	DC<1:0>		—	—	—	—	—	—	xx-- ----	uu-- ----
61Bh	PWM9DCH	DC<9:2>								xxxx xxxx	uuuu uuuu
61Ch	PWM9CON	EN	—	OUT	POL	—	—	—	—	0-00 ----	0-00 ----
61Dh	PWM10DCL(3)	DC<1:0>		—	—	—	—	—	—	xx-- ----	uu-- ----
61Eh	PWM10DCH(3)	DC<9:2>								xxxx xxxx	uuuu uuuu
61Fh	PWM10CON(3)	EN	—	OUT	POL	—	—	—	—	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: Unimplemented, read as '1'.
2: Unimplemented on PIC16LF1777/8/9.
3: Unimplemented on PIC16(L)F1778.

5.0 OSCILLATOR MODULE (WITH FAIL-SAFE CLOCK MONITOR)

5.1 Overview

The oscillator module has a wide variety of clock sources and selection features that allow it to be used in a wide range of applications while maximizing performance and minimizing power consumption. Figure 5-1 illustrates a block diagram of the oscillator module.

Clock sources can be supplied from external oscillators, quartz crystal resonators, ceramic resonators and Resistor-Capacitor (RC) circuits. In addition, the system clock source can be supplied from one of two internal oscillators and PLL circuits, with a choice of speeds selectable via software. Additional clock features include:

- Selectable system clock source between external or internal sources via software.
- Two-Speed Start-up mode, which minimizes latency between external oscillator start-up and code execution.
- Fail-Safe Clock Monitor (FSCM) designed to detect a failure of the external clock source (LP, XT, HS, ECH, ECM, ECL or EXTRC modes) and switch automatically to the internal oscillator.
- Oscillator Start-up Timer (OST) ensures stability of crystal oscillator sources.

The oscillator module can be configured in one of the following clock modes.

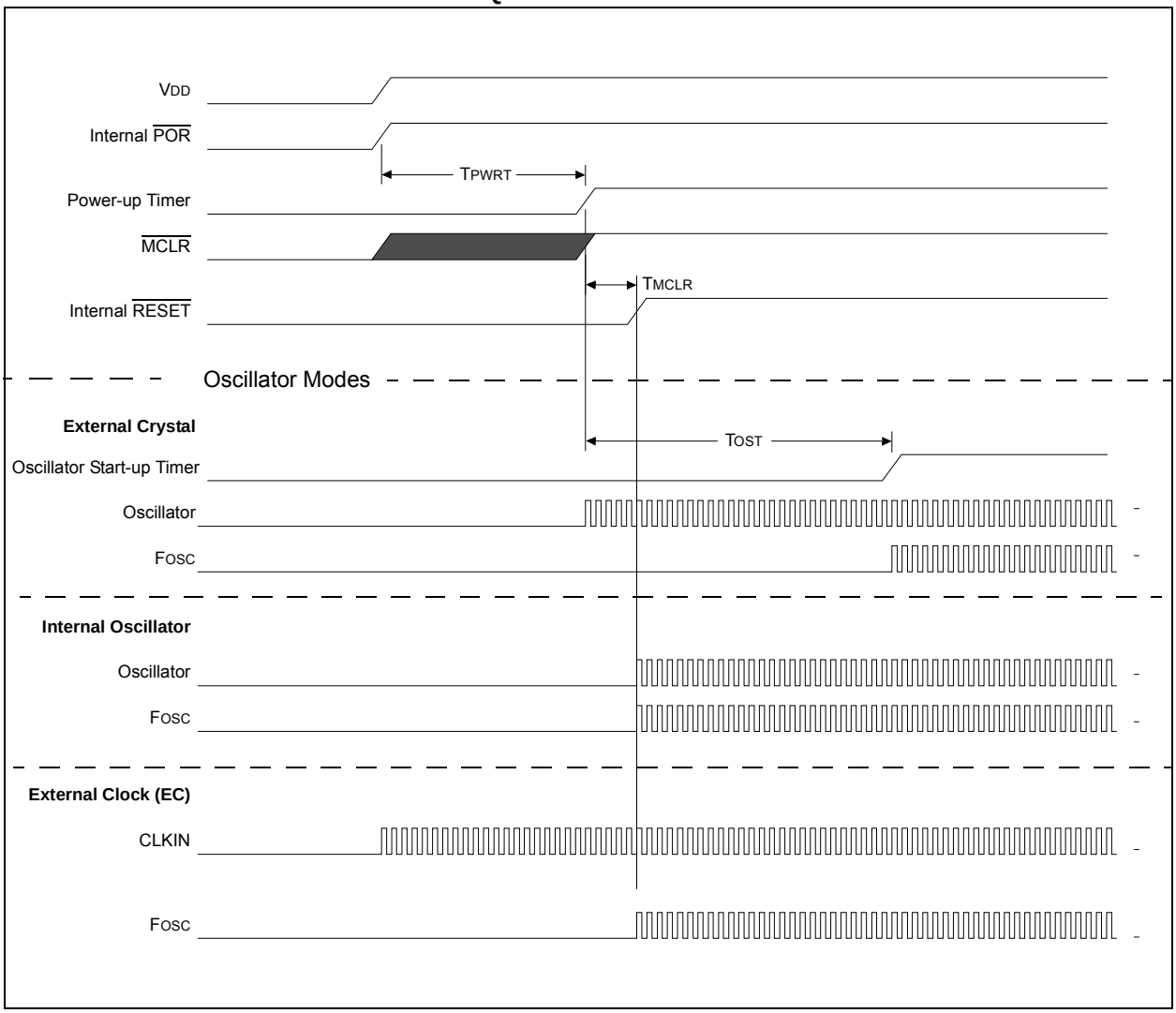
1. ECL – External Clock Low-Power mode (0 MHz to 0.5 MHz)
2. ECM – External Clock Medium Power mode (0.5 MHz to 4 MHz)
3. ECH – External Clock High-Power mode (4 MHz to 32 MHz)
4. LP – 32 kHz Low-Power Crystal mode.
5. XT – Medium Gain Crystal or Ceramic Resonator Oscillator mode (up to 4 MHz)
6. HS – High Gain Crystal or Ceramic Resonator mode (4 MHz to 20 MHz)
7. EXTRC – External Resistor-Capacitor
8. INTOSC – Internal oscillator (31 kHz to 32 MHz)

Clock Source modes are selected by the FOSC<2:0> bits in the Configuration Words. The FOSC bits determine the type of oscillator that will be used when the device is first powered.

The ECH, ECM, and ECL clock modes rely on an external logic level signal as the device clock source. The LP, XT, and HS clock modes require an external crystal or resonator to be connected to the device. Each mode is optimized for a different frequency range. The EXTRC clock mode requires an external resistor and capacitor to set the oscillator frequency.

The INTOSC internal oscillator block produces low, medium, and high-frequency clock sources, designated LFINTOSC, MFINTOSC and HFINTOSC. (see Internal Oscillator Block, Figure 5-1). A wide selection of device clock frequencies may be derived from these three clock sources.

FIGURE 6-3: RESET START-UP SEQUENCE



11.8 Register Definitions: PORTD

REGISTER 11-26: PORTD: PORTD REGISTER

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0
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 7-0 **RD<7:0>**: PORTD General Purpose I/O Pin bits(1)
1 = Port pin is ≥ V_{IH}
0 = Port pin is ≤ V_{IL}

Note 1: Writes to PORTD are actually written to corresponding LATD register. Reads from PORTD register is return of actual I/O pin values.

REGISTER 11-27: TRISD: PORTD TRI-STATE REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0
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 7-0 **TRISD<7:0>:** PORTD Tri-State Control bits
1 = PORTD pin configured as an input (tri-stated)
0 = PORTD pin configured as an output

TABLE 23-1: TIMER2 OPERATING MODES

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

Note 1: If ON = 0 then an edge is required to restart the timer after ON = 1.

2: When TMRx = PRx then the next clock clears ON and stops TMRx at 00h.

3: When TMRx = PRx then the next clock stops TMRx at 00h but does not clear ON.

24.5 Register Definitions: CCP Control

REGISTER 24-1: CCPxCON: CCPx CONTROL REGISTER

R/W-0/0	U-0	R-x	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
EN	—	OUT	FMT	MODE<3:0>			
bit 7							bit 0

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

bit 7 **EN:** CCPx Module Enable bit
1 = CCPx is enabled
0 = CCPx is disabled

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

bit 5 **OUT:** CCPx Output Data bit (read-only)

bit 4 **FMT:** CCPW (Pulse-Width) Alignment bit
If MODE = PWM Mode
1 = Left-aligned format, CCPRxH <7> is the MSB of the PWM duty cycle
0 = Right-aligned format, CCPRxL<0> is the LSB of the PWM duty cycle

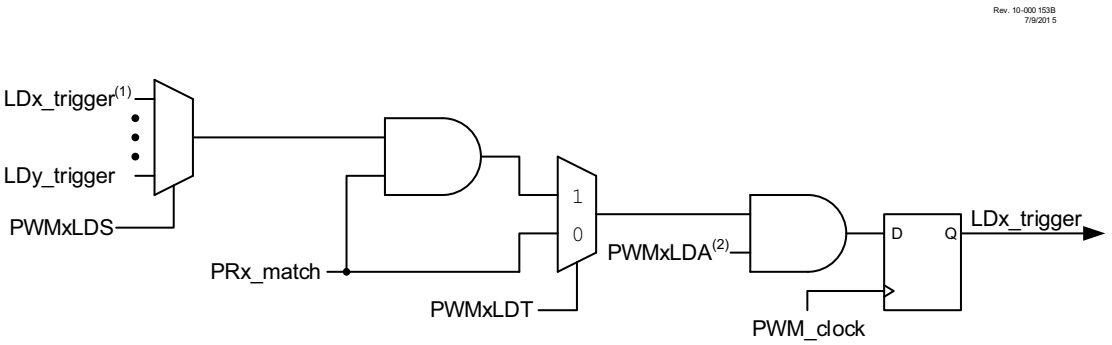
bit 3-0 **MODE<3:0>:** CCPx Mode Selection bits
11xx = PWM mode

1011 = Compare mode: Pulse output, clear TMR1
1010 = Compare mode: Pulse output (0 - 1 - 0)
1001 = Compare mode: clear output on compare match. Output is set upon selection of this mode.
1000 = Compare mode: set output on compare match. Output is set upon selection of this mode.

0111 = Capture mode: every 16th rising edge
0110 = Capture mode: every 4th rising edge
0101 = Capture mode: every rising edge
0100 = Capture mode: every falling edge

0011 = Capture mode: every rising or falling edge
0010 = Compare mode: toggle output on match
0001 = Compare mode: Toggle output and clear TMR1 on match
0000 = Capture/Compare/PWM off (resets CCPx module) (reserved for backwards compatibility)

FIGURE 26-2: LOAD TRIGGER BLOCK DIAGRAM



Note 1. The input corresponding to a PWM module's own load trigger is reserved.
2. PWMxLDA is cleared by hardware upon LDx_trigger.

26.1 Fundamental Operation

The PWM module produces a 16-bit resolution pulse-width modulated output.

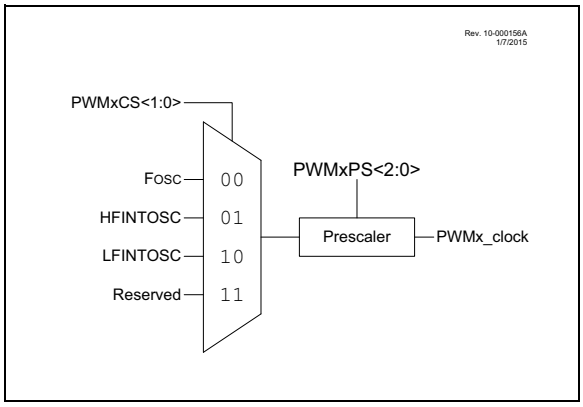
Each PWM module has an independent timer driven by a selection of clock sources determined by the PWMxCLKCON register (Register 26-4). The timer value is compared to event count registers to generate the various events of a the PWM waveform, such as the period and duty cycle. For a block diagram describing the clock sources refer to Figure 26-3.

Each PWM module can be enabled individually using the EN bit of the PWMxCON register, or several PWM modules can be enabled simultaneously using the MPWMxEN bits of the PWMEN register.

The current state of the PWM output can be read using the OUT bit of the PWMxCON register. In some modes this bit can be set and cleared by software giving additional software control over the PWM waveform. This bit is synchronized to FOSC/4 and therefore does not change in real time with respect to the PWM_clock.

Note:	If PWM_clock > FOSC/4, the OUT bit may not accurately represent the output state of the PWM.

FIGURE 26-3: PWM CLOCK SOURCE BLOCK DIAGRAM



26.1.1 PWMx PIN CONFIGURATION

This device uses the PPS control circuitry to route peripherals to any device I/O pin. Select the desired pin, or pins, for PWM output with the device pin RxyPPS control registers (Register 12-2).

All PWM outputs are multiplexed with the PORT data latch, so the pins must also be configured as outputs by clearing the associated PORT TRIS bits.

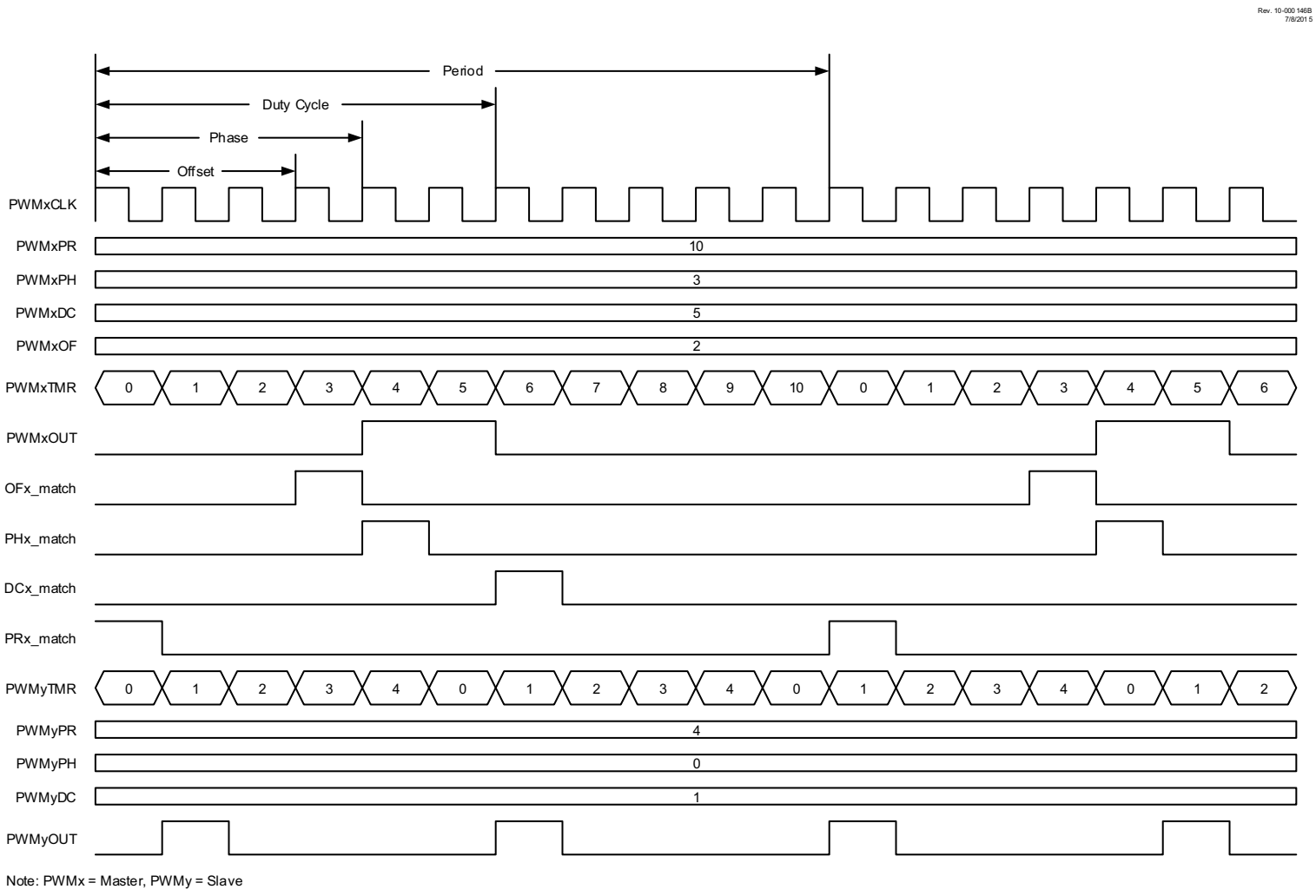
The slew rate feature may be configured to optimize the rate to be used in conjunction with the PWM outputs. High-speed output switching is attained by clearing the associated PORT SLRCON bits.

The PWM outputs can be configured to be open-drain outputs by setting the associated PORT ODCON bits.

26.1.2 PWMx Output Polarity

The output polarity is inverted by setting the POL bit of the PWMxCON register. The polarity control affects the PWM output even when the module is not enabled.

FIGURE 26-8: INDEPENDENT RUN MODE TIMING DIAGRAM



Note: PWMx = Master, PWMy = Slave

26.7 Register Definitions: PWM Control

Long bit name prefixes for the 16-bit PWM peripherals are shown in Table 26-2. Refer to **Section 1.1 "Register and Bit naming conventions"** for more information

TABLE 26-2:

Peripheral	Bit Name Prefix
PWM5	PWM5
PWM6	PWM6
PWM11	PWM11
PWM12(1)	PWM12

Note 1: PIC16(L)F1777/9 only.

REGISTER 26-1: PWMxCON: PWM CONTROL REGISTER

R/W-0/0	U-0	R/HS/HC-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0
EN	—	OUT	POL	MODE<1:0>		—	—
bit 7							bit 0

Legend:		
HC = Bit is cleared by hardware		HS = Bit is set by hardware
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 **EN:** PWM Module Enable bit
1 = Module is enabled
0 = Module is disabled

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

bit 5 **OUT:** Output State of the PWM module

bit 4 **POL:** PWM Output Polarity Control bit
1 = PWM output active state is low
0 = PWM output active state is high

bit 3-2 **MODE<1:0>:** PWM Mode Control bits
11 = Center Aligned mode
10 = Toggle On Match mode
01 = Set On Match mode
00 = Standard PWM mode

bit 1-0 **Unimplemented:** Read as '0'

FIGURE 27-5: SIMPLIFIED COG BLOCK DIAGRAM (HALF-BRIDGE MODE, MD = 4)

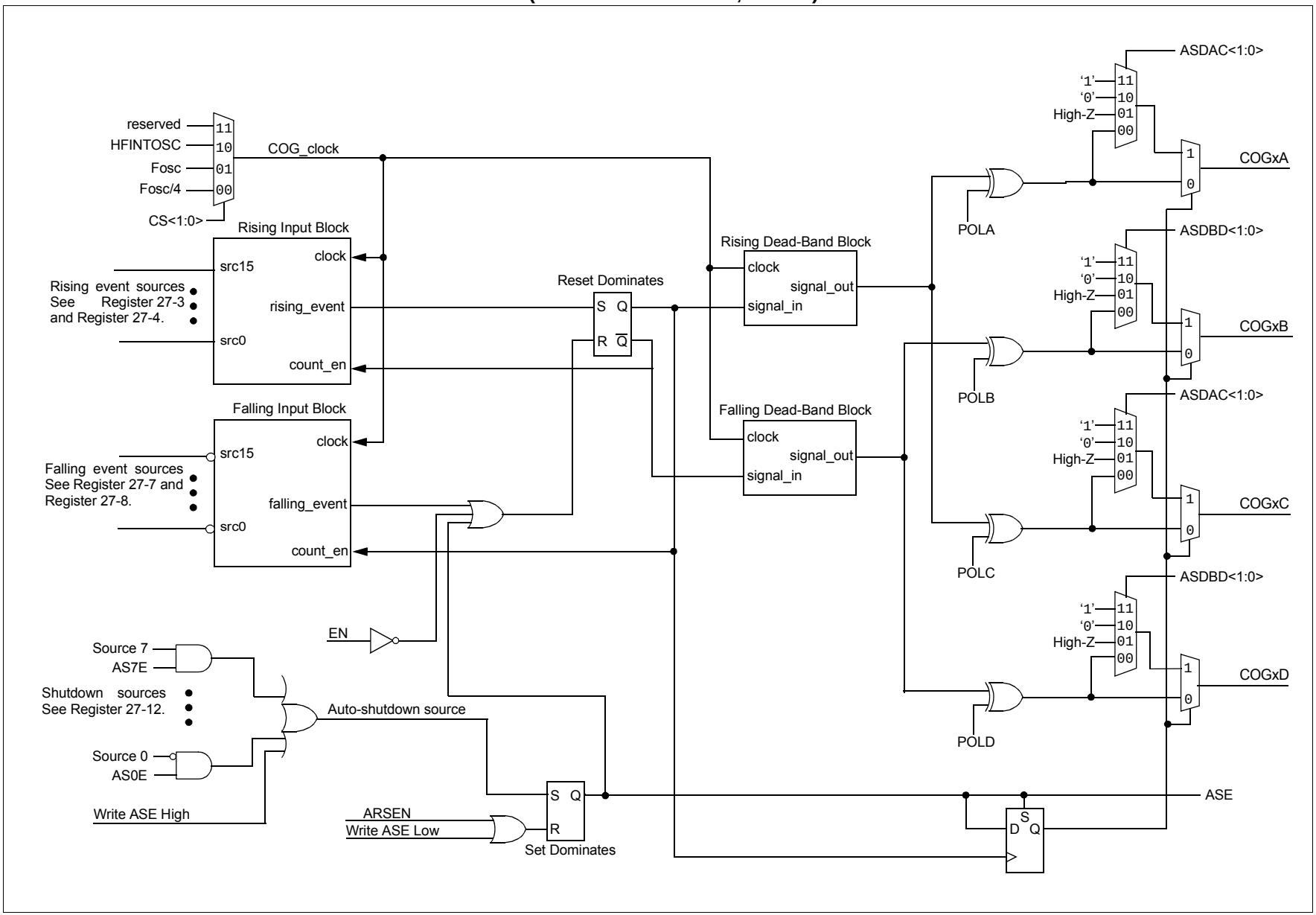
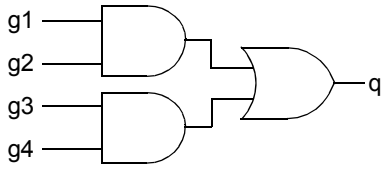


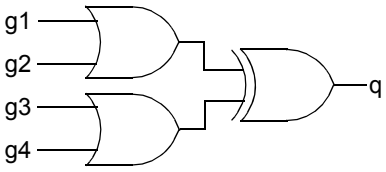
FIGURE 28-3: PROGRAMMABLE LOGIC FUNCTIONS

AND – OR



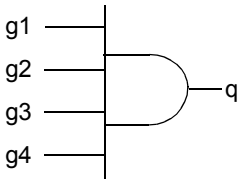
MODE<2:0>= 000

OR – XOR



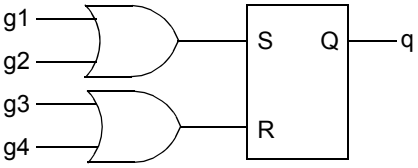
MODE<2:0>= 001

4-Input AND



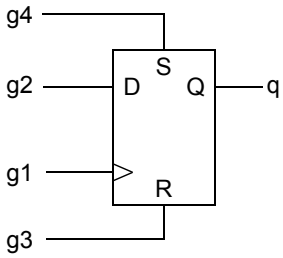
MODE<2:0>= 010

S-R Latch



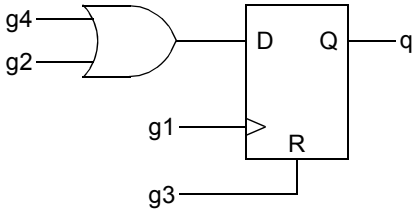
MODE<2:0>= 011

1-Input D Flip-Flop with S and R



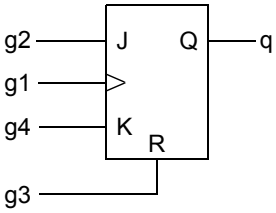
MODE<2:0>= 100

2-Input D Flip-Flop with R



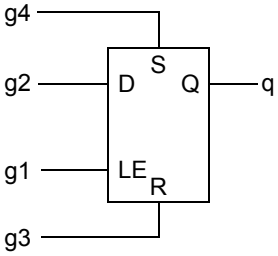
MODE<2:0>= 101

J-K Flip-Flop with R



MODE<2:0>= 110

1-Input Transparent Latch with S and R



MODE<2:0>= 111

29.6 Register Definitions: Op Amp Control

Long bit name prefixes for the op amp peripherals are shown in Table 29-2. Refer to **Section 1.1 "Register and Bit naming conventions"** for more information

TABLE 29-2:

Peripheral	Bit Name Prefix
OPA1	OPA1
OPA2	OPA2
OPA3	OPA3
OPA4(1)	OPA4

Note 1: PIC16(L)F1777/9 only.

REGISTER 29-1: OPAxCON: OPERATIONAL AMPLIFIER (OPAx) CONTROL REGISTER

R/W-0/0	U-0	U-0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
EN	—	—	UG	—	ORPOL	ORM<1:0>	
bit 7							bit 0

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

bit 7 **EN:** Op Amp Enable bit
1 = Op amp is enabled
0 = Op amp is disabled and consumes no active power

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

bit 4 **UG:** Op Amp Unity Gain Select bit
1 = OPA output is connected to inverting input. OPAxIN- pin is available for general purpose I/O.
0 = Inverting input is connected to the OPAxIN- pin

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

bit 2 **ORPOL:** Op Amp Override Source Polarity bit
1 = Override source polarity is inverted. Override occurs when source is high.
0 = Override source polarity is not inverted. Override occurs when source is low.

bit 1-0 **ORM<1:0>:** Op Amp Override Mode Selection bits
11 = Reserved. Do not use.
10 = Op amp is forced to unity gain when override source is true.
01 = Op amp output is tri-stated when override source is true.
00 = Output override function is disabled.

31.11 Register Definitions: Data Signal Modulator

Long bit name prefixes for the 10-bit DAC peripherals are shown in Table 31-3. Refer to **Section 1.1 "Register and Bit naming conventions"** for more information

TABLE 31-3:

Peripheral	Bit Name Prefix
DSM1	DSM1
DSM2	DSM2
DSM3	DSM3
DSM4(1)	DSM4

Note 1: PIC16(L)F1777/9 only.

REGISTER 31-1: MDxCON0: MODULATION CONTROL REGISTER 0

R/W-0/0	U-0	R-0/0	R/W-0/0	U-0	U-0	U-0	R/W-0/0
EN	—	OUT	OPOL	—	—	—	BIT
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 **EN:** Modulator Module Enable bit
- 1 = Modulator module is enabled and mixing input signals
- 0 = Modulator module is disabled and has no output

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

bit 5 **OUT:** Modulator Output bit
Displays the current output value of the modulator module.(1)

bit 4 **OPOL:** Modulator Output Polarity Select bit
- 1 = Modulator output signal is inverted. Idle high output.
- 0 = Modulator output signal is not inverted. Idle low output.

bit 3-1 **Unimplemented:** Read as '0'

bit 0 **BIT:** Allows direct software control of the modulation source input to module(2)
- 1 = Modulator uses High Carrier source
- 0 = Modulator uses Low Carrier source

Note 1: The modulated output frequency can be greater and asynchronous from the clock that updates this register bit, the bit value may not be valid for higher speed modulator or carrier signals.

2: BIT must be selected as the modulation source in the MDSRC register for this operation.

RRF **Rotate Right f through Carry**

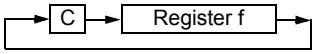
Syntax: [*label*] RRF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: See description below

Status Affected: C

Description: The contents of register 'f' are rotated one bit to the right through the Carry flag. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.



SLEEP **Enter Sleep mode**

Syntax: [*label*] SLEEP

Operands: None

Operation: 00h → WDT,
0 → WDT prescaler,
1 → $\overline{TO}$,
0 → $\overline{PD}$

Status Affected: $\overline{TO}$, $\overline{PD}$

Description: The power-down Status bit, $\overline{PD}$ is cleared. Time-out Status bit, $\overline{TO}$ is set. Watchdog Timer and its prescaler are cleared.
The processor is put into Sleep mode with the oscillator stopped.

SUBLW **Subtract W from literal**

Syntax: [*label*] SUBLW k

Operands: $0 \le k \le 255$

Operation: k - (W) → (W)

Status Affected: C, DC, Z

Description: The W register is subtracted (2's complement method) from the 8-bit literal 'k'. The result is placed in the W register.

C = 0	W > k
C = 1	W ≤ k
DC = 0	W<3:0> > k<3:0>
DC = 1	W<3:0> ≤ k<3:0>

SUBWF **Subtract W from f**

Syntax: [*label*] SUBWF f,d

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f) - (W) → (destination)

Status Affected: C, DC, Z

Description: Subtract (2's complement method) W register from register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f.

C = 0	W > f
C = 1	W ≤ f
DC = 0	W<3:0> > f<3:0>
DC = 1	W<3:0> ≤ f<3:0>

SUBWFB **Subtract W from f with Borrow**

Syntax: SUBWFB f {,d}

Operands: $0 \le f \le 127$
$d \in [0,1]$

Operation: (f) – (W) – ($\overline{B}$) → dest

Status Affected: C, DC, Z

Description: Subtract W and the BORROW flag (CARRY) from register 'f' (2's complement method). If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f'.

FIGURE 36-7: CLKOUT AND I/O TIMING

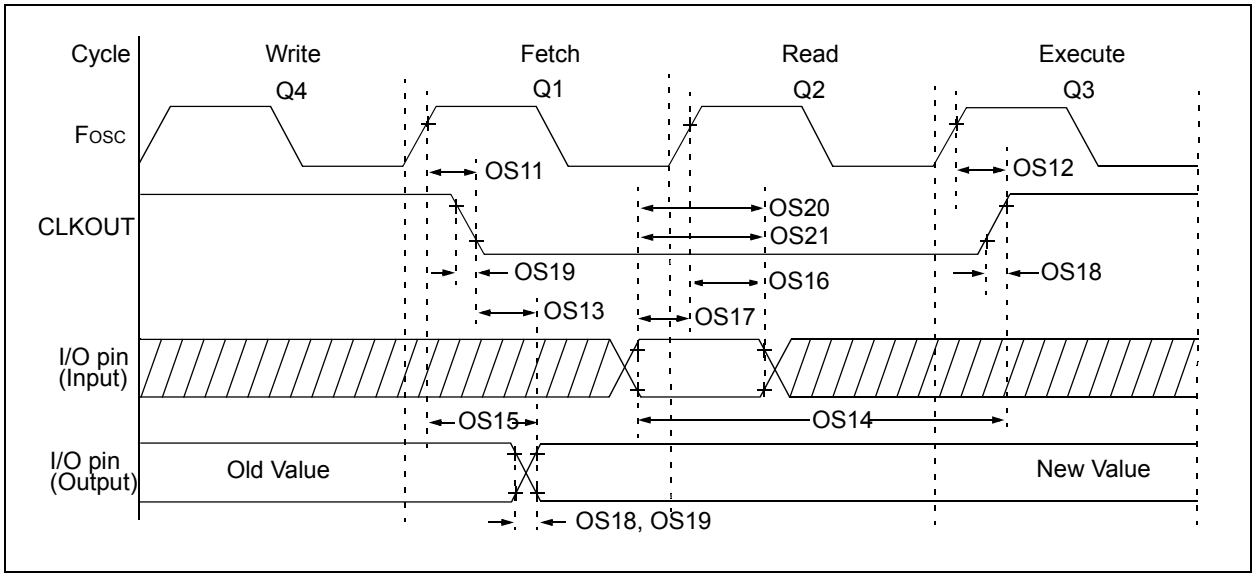


TABLE 36-10: CLKOUT AND I/O TIMING PARAMETERS

Standard Operating Conditions (unless otherwise stated)							
Param No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
OS11	TosH2ckL	FOSC↑ to CLKOUT↓ (1)	—	—	70	ns	3.3V ≤ VDD ≤ 5.0V
OS12	TosH2ckH	FOSC↑ to CLKOUT↑ (1)	—	—	72	ns	3.3V ≤ VDD ≤ 5.0V
OS13	TckL2ioV	CLKOUT↓ to Port out valid(1)	—	—	20	ns	
OS14	TioV2ckH	Port input valid before CLKOUT↑(1)	TOSC + 200 ns	—	—	ns	
OS15	TosH2ioV	Fosc↑ (Q1 cycle) to Port out valid	—	50	70*	ns	3.3V ≤ VDD ≤ 5.0V
OS16	TosH2ioI	Fosc↑ (Q2 cycle) to Port input invalid (I/O in hold time)	50	—	—	ns	3.3V ≤ VDD ≤ 5.0V
OS17	TioV2osH	Port input valid to Fosc↑ (Q2 cycle) (I/O in setup time)	20	—	—	ns	
OS18*	TioR	Port output rise time(2)	— —	40 15	72 32	ns	VDD = 1.8V 3.3V ≤ VDD ≤ 5.0V
OS19*	TioF	Port output fall time(2)	— —	28 15	55 30	ns	VDD = 1.8V 3.3V ≤ VDD ≤ 5.0V
OS20*	Tinp	INT pin input high or low time	25	—	—	ns	
OS21*	Tioc	Interrupt-on-change new input level time	25	—	—	ns	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated.

Note 1: Measurements are taken in EXTRC mode where CLKOUT output is 4 x TOSC.

2: Slew rate limited.

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

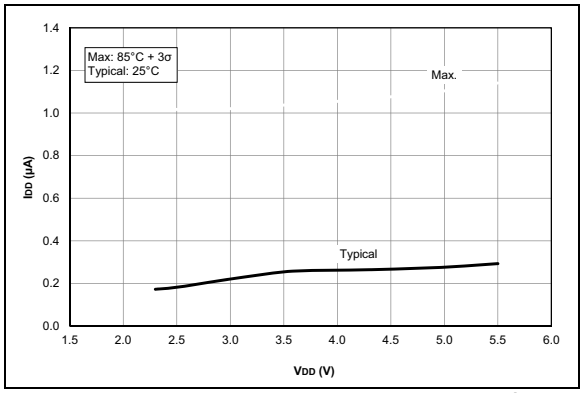


FIGURE 37-49: *IPD, ADC Non-Converting, PIC16F1777/8/9 Only.*

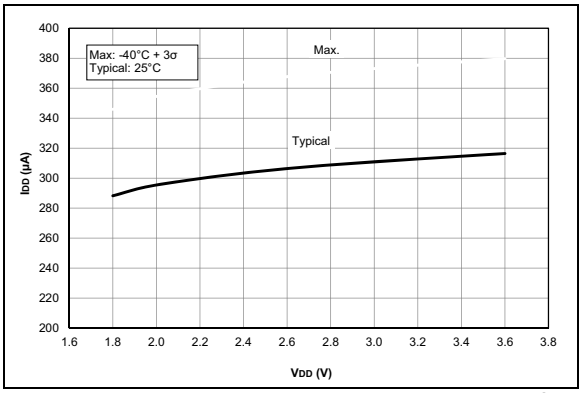


FIGURE 37-50: *IPD, Comparator, NP Mode (VREGPM = 0), PIC16LF1777/8/9 Only.*

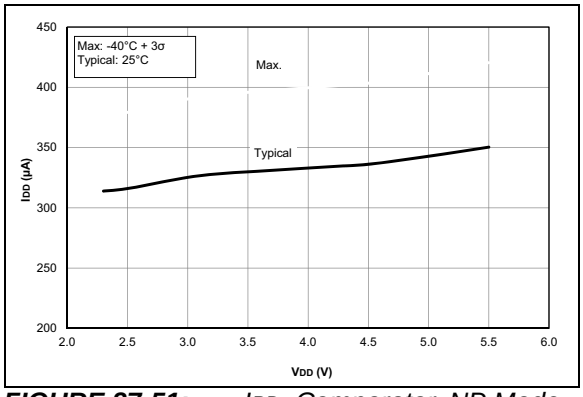


FIGURE 37-51: *IPD, Comparator, NP Mode (VREGPM = 0), PIC16F1777/8/9 Only.*

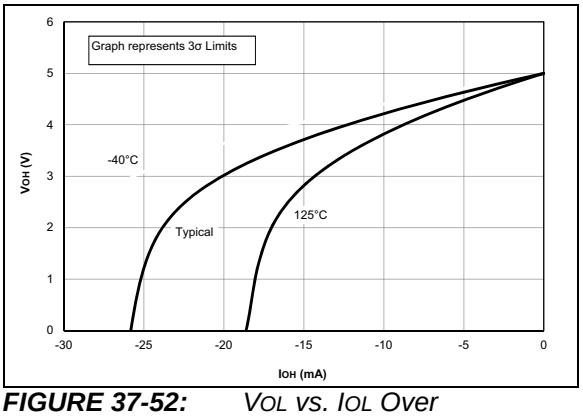


FIGURE 37-52: *VOL vs. IOL Over Temperature, VDD = 5.0V, PIC16F1777/8/9 Only.*

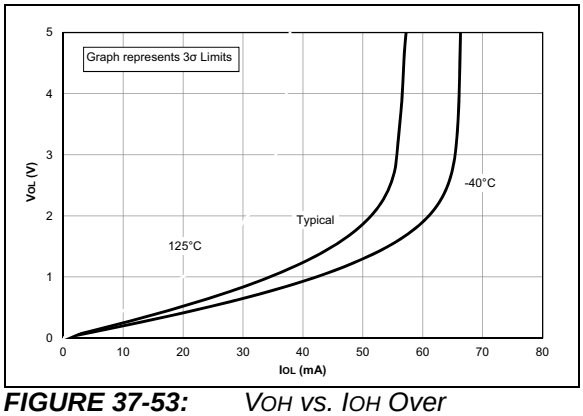


FIGURE 37-53: *VOH vs. IOH Over Temperature, VDD = 5.0V, PIC16F1777/8/9 Only.*

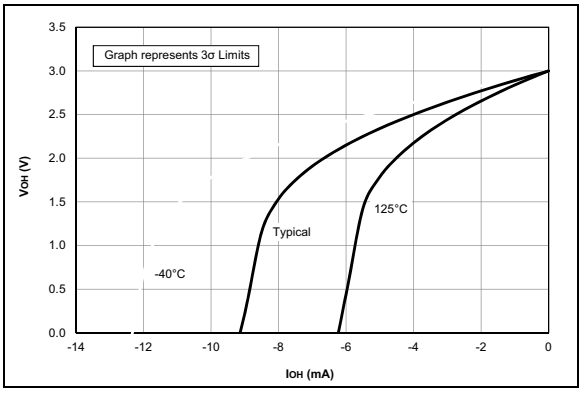


FIGURE 37-54: *VOL vs. IOL Over Temperature, VDD = 3.0V.*

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

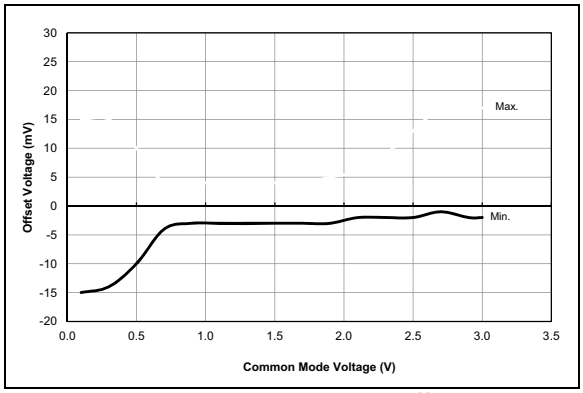


FIGURE 37-103: *Comparator Offset, NP Mode (CxSP = 1), VDD = 3.0V, Typical Measured Values at 25°C.*

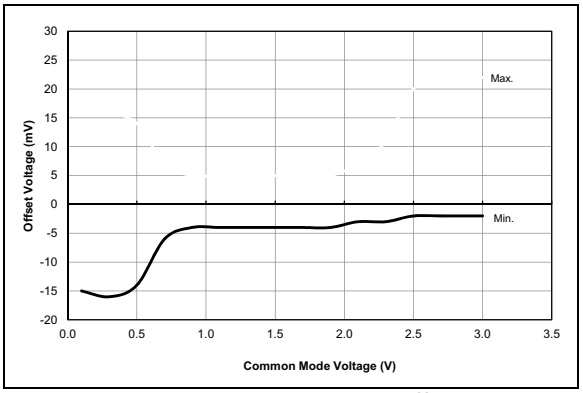


FIGURE 37-104: *Comparator Offset, NP Mode (CxSP = 1), VDD = 3.0V, Typical Measured Values from -40°C to 125°C.*

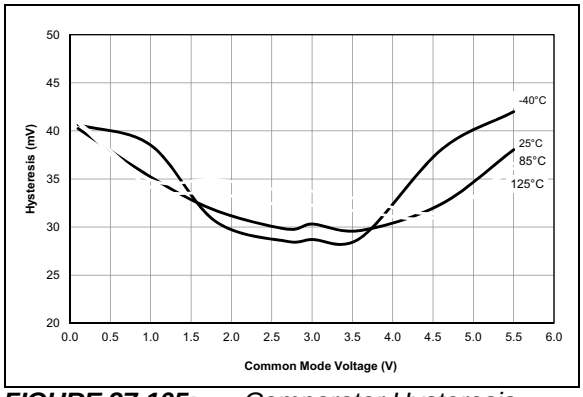


FIGURE 37-105: *Comparator Hysteresis, NP Mode (CxSP = 1), VDD = 5.5V, Typical Measured Values, PIC16F1777/8/9 only.*

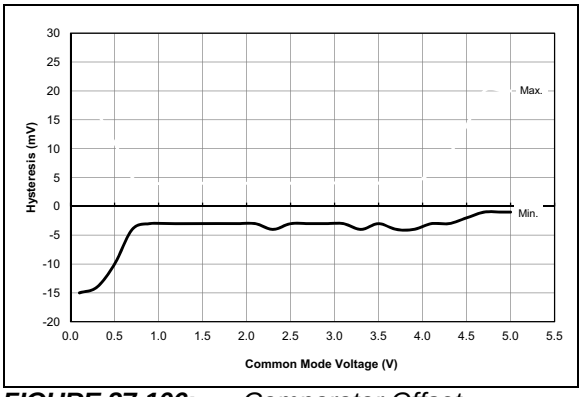


FIGURE 37-106: *Comparator Offset, NP Mode (CxSP = 1), VDD = 5.0V, Typical Measured Values at 25°C, PIC16F1777/8/9 only.*

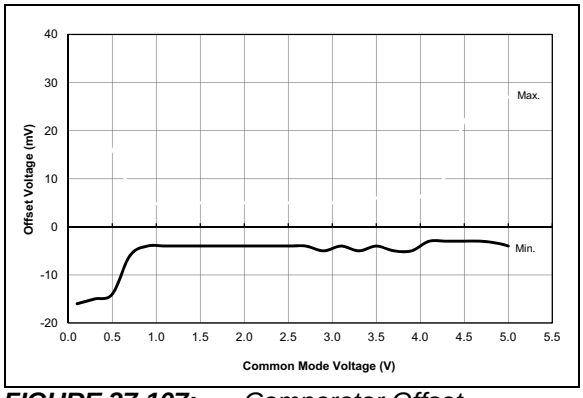


FIGURE 37-107: *Comparator Offset, NP Mode (CxSP = 1), VDD = 5.5V, Typical Measured Values from -40°C to 125°C, PIC16F1777/8/9 only.*

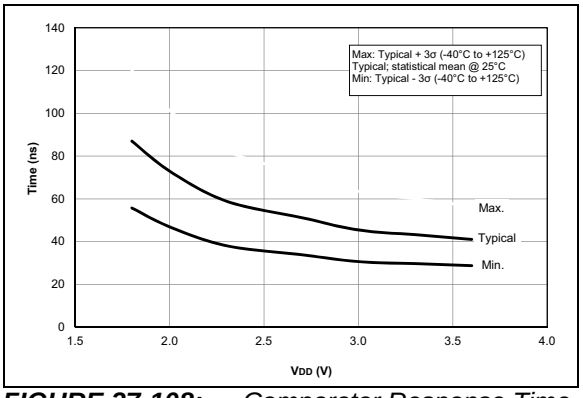


FIGURE 37-108: *Comparator Response Time Over Voltage, NP Mode (CxSP = 1), Typical Measured Values.*

38.0 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

38.1 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

38.2 MPLAB XC Compilers

The MPLAB XC Compilers are complete ANSI C compilers for all of Microchip's 8, 16, and 32-bit MCU and DSC devices. These compilers provide powerful integration capabilities, superior code optimization and ease of use. MPLAB XC Compilers run on Windows, Linux or MAC OS X.

For easy source level debugging, the compilers provide debug information that is optimized to the MPLAB X IDE.

The free MPLAB XC Compiler editions support all devices and commands, with no time or memory restrictions, and offer sufficient code optimization for most applications.

MPLAB XC Compilers include an assembler, linker and utilities. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. MPLAB XC Compiler uses the assembler to produce its object file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

38.3 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code, and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB X IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multipurpose source files
- Directives that allow complete control over the assembly process

38.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

38.5 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC DSC devices. MPLAB XC Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility