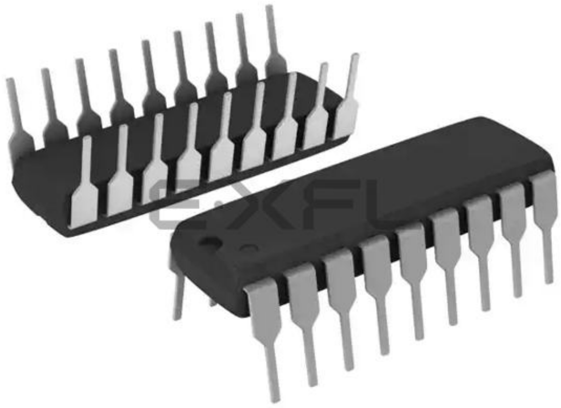


Welcome to E-XFL.COM

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	20MHz
Connectivity	UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	16
Program Memory Size	1.75KB (1K x 14)
Program Memory Type	FLASH
EEPROM Size	128 x 8
RAM Size	224 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Through Hole
Package / Case	18-DIP (0.300", 7.62mm)
Supplier Device Package	18-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f627a-e-p

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

QUALITY MANAGEMENT SYSTEM CERTIFIED BY DNV

== ISO/TS 16949:2002 ==

Microchip received ISO/TS-16949:2002 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

PIC16F627A/628A/648A

18-pin Flash-Based, 8-Bit CMOS Microcontrollers with nanoWatt Technology

High-Performance RISC CPU:

- Operating speeds from DC – 20 MHz
- Interrupt capability
- 8-level deep hardware stack
- Direct, Indirect and Relative Addressing modes
- 35 single-word instructions:
  - All instructions single cycle except branches

Special Microcontroller Features:

- Internal and external oscillator options:
  - Precision internal 4 MHz oscillator factory calibrated to ±1%
  - Low-power internal 48 kHz oscillator
  - External Oscillator support for crystals and resonators
- Power-saving Sleep mode
- Programmable weak pull-ups on PORTB
- Multiplexed Master Clear/Input-pin
- Watchdog Timer with independent oscillator for reliable operation
- Low-voltage programming
- In-Circuit Serial Programming™ (via two pins)
- Programmable code protection
- Brown-out Reset
- Power-on Reset
- Power-up Timer and Oscillator Start-up Timer
- Wide operating voltage range (2.0-5.5V)
- Industrial and extended temperature range
- High-Endurance Flash/EEPROM cell:
  - 100,000 write Flash endurance
  - 1,000,000 write EEPROM endurance
  - 40 year data retention

Low-Power Features:

- Standby Current:
  - 100 nA @ 2.0V, typical
- Operating Current:
  - 12 μA @ 32 kHz, 2.0V, typical
  - 120 μA @ 1 MHz, 2.0V, typical
- Watchdog Timer Current:
  - 1 μA @ 2.0V, typical
- Timer1 Oscillator Current:
  - 1.2 μA @ 32 kHz, 2.0V, typical
- Dual-speed Internal Oscillator:
  - Run-time selectable between 4 MHz and 48 kHz
  - 4 μs wake-up from Sleep, 3.0V, typical

Peripheral Features:

- 16 I/O pins with individual direction control
- High current sink/source for direct LED drive
- Analog comparator module with:
  - Two analog comparators
  - Programmable on-chip voltage reference (V_{REF}) module
  - Selectable internal or external reference
  - Comparator outputs are externally accessible
- Timer0: 8-bit timer/counter with 8-bit programmable prescaler
- Timer1: 16-bit timer/counter with external crystal/ clock capability
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Capture, Compare, PWM module:
  - 16-bit Capture/Compare
  - 10-bit PWM
- Addressable Universal Synchronous/Asynchronous Receiver/Transmitter USART/SCI

Device	Program Memory	Data Memory		I/O	CCP (PWM)	USART	Comparators	Timers 8/16-bit
	Flash (words)	SRAM (bytes)	EEPROM (bytes)					
PIC16F627A	1024	224	128	16	1	Y	2	2/1
PIC16F628A	2048	224	128	16	1	Y	2	2/1
PIC16F648A	4096	256	256	16	1	Y	2	2/1

FIGURE 5-4: BLOCK DIAGRAM OF RA4/T0CKI/CMP2 PIN

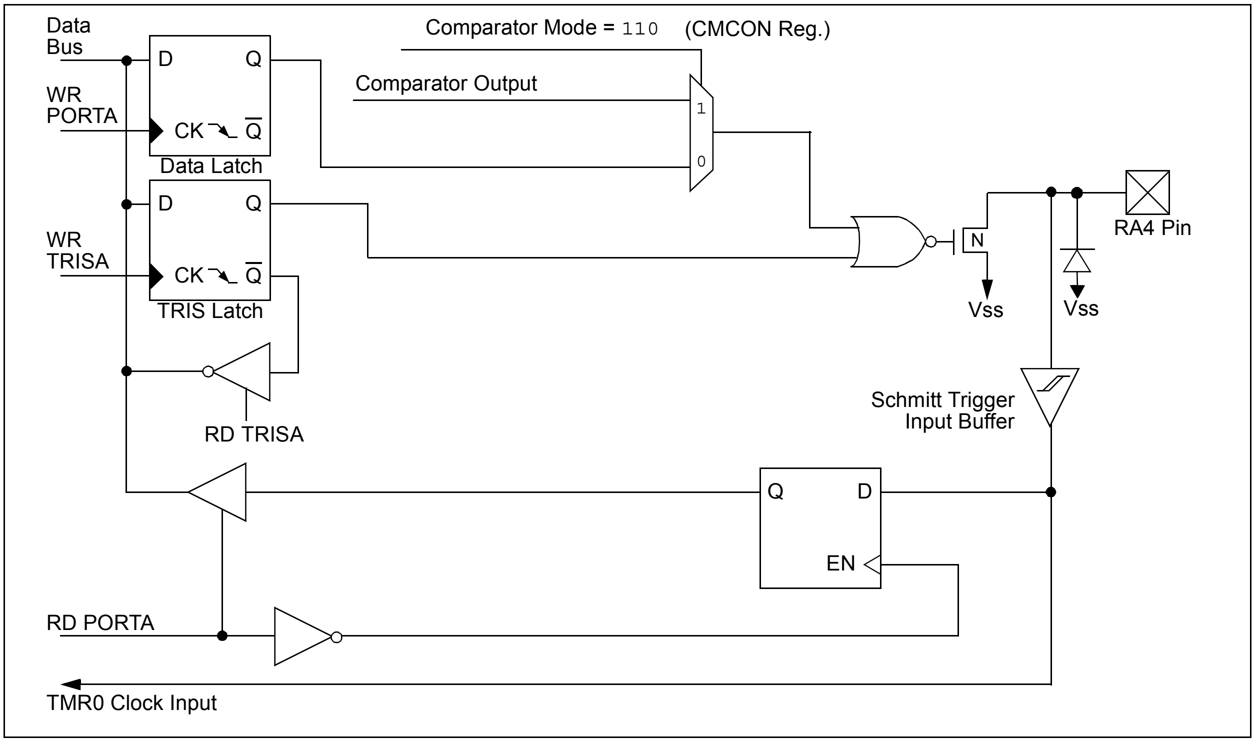


FIGURE 5-5: BLOCK DIAGRAM OF THE RA5/MCLR/VPP PIN

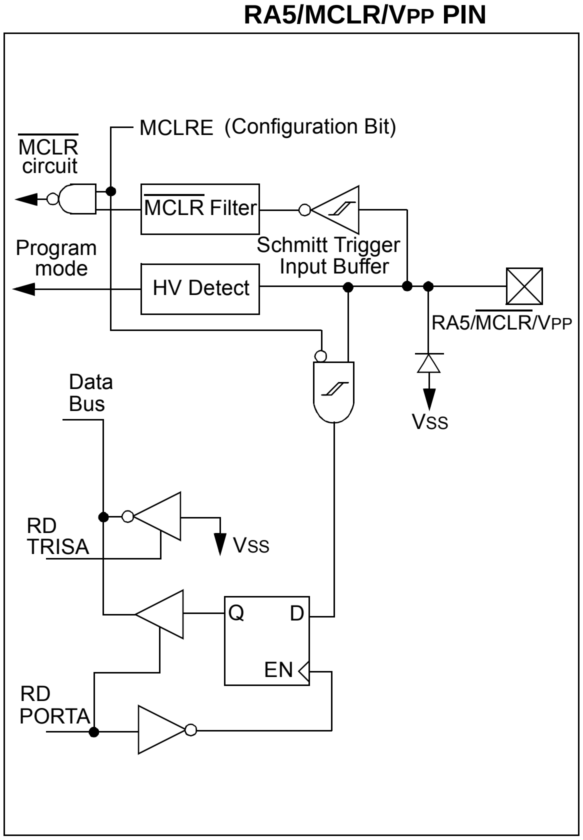
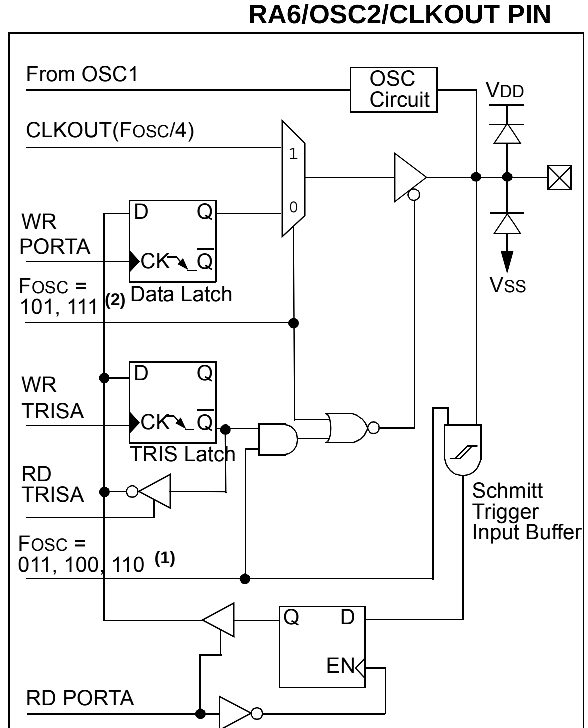


FIGURE 5-6: BLOCK DIAGRAM OF RA6/OSC2/CLKOUT PIN



Note 1: INTOSC with RA6 = I/O or RC with RA6 = I/O.

2: INTOSC with RA6 = CLKOUT or RC with RA6 = CLKOUT.

7.3 Timer1 Operation in Asynchronous Counter Mode

If control bit $\overline{\text{T1SYNC}}$ (T1CON<2>) is set, the external clock input is not synchronized. The timer continues to increment asynchronous to the internal phase clocks. 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 (**Section 7.3.2 “Reading and Writing Timer1 in Asynchronous Counter Mode”**).

Note:	In Asynchronous Counter mode, Timer1 cannot be used as a time base for capture or compare operations.

7.3.1 EXTERNAL CLOCK INPUT TIMING WITH UNSYNCHRONIZED CLOCK

If control bit $\overline{\text{T1SYNC}}$ is set, the timer will increment completely asynchronously. The input clock must meet certain minimum high and low time requirements. Refer to Table 17-8 in the Electrical Specifications Section, timing parameters 45, 46 and 47.

7.3.2 READING AND WRITING TIMER1 IN ASYNCHRONOUS COUNTER MODE

Reading the TMR1H or TMR1L register, while the timer is running from an external asynchronous clock, will produce 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 timer register.

Reading the 16-bit value requires some care. Example 7-1 is an example routine to read the 16-bit timer value. This is useful if the timer cannot be stopped.

EXAMPLE 7-1: READING A 16-BIT FREE-RUNNING TIMER

```
; All interrupts are disabled
    MOVF    TMR1H, W    ;Read high byte
    MOVWF   TMPH        ;
    MOVF    TMR1L, W    ;Read low byte
    MOVWF   TMPL        ;
    MOVF    TMR1H, W    ;Read high byte
    SUBWF   TMPH, W     ;Sub 1st read with
                        ;2nd read
    BTFSC   STATUS,Z    ;Is result = 0
    GOTO    CONTINUE    ;Good 16-bit read
;
; TMR1L may have rolled over between the
; read of the high and low bytes. Reading
; the high and low bytes now will read a good
; value.
;
    MOVF    TMR1H, W    ;Read high byte
    MOVWF   TMPH        ;
    MOVF    TMR1L, W    ;Read low byte
    MOVWF   TMPL        ;
; Re-enable the Interrupts (if required)
CONTINUE                ;Continue with your
                        ;code
```

9.3.2 PWM DUTY CYCLE

The PWM duty cycle is specified by writing to the CCPR1L register and to the CCP1CON<5:4> bits. Up to 10-bit resolution is available: the CCPR1L contains the eight MSbs and the CCP1CON<5:4> contains the two LSbs. This 10-bit value is represented by CCPR1L:CCP1CON<5:4>. The following equation is used to calculate the PWM duty cycle in time:

$$\text{PWM duty cycle} = (\text{CCPR1L:CCP1CON<5:4>}) \cdot T_{OSC} \cdot \text{TMR2 prescale value}$$

CCPR1L and CCP1CON<5:4> can be written to at any time, but the duty cycle value is not latched into CCPR1H until after a match between PR2 and TMR2 occurs (i.e., the period is complete). In PWM mode, CCPR1H is a read-only register.

The CCPR1H register and a 2-bit internal latch are used to double buffer the PWM duty cycle. This double buffering is essential for glitchless PWM operation.

When the CCPR1H and 2-bit latch match TMR2 concatenated with an internal 2-bit Q clock or 2 bits of the TMR2 prescaler, the CCP1 pin is cleared.

Maximum PWM resolution (bits) for a given PWM frequency:

$$\text{PWM Resolution} = \frac{\log\left(\frac{F_{OSC}}{F_{PWM} \times \text{TMR2 Prescaler}}\right)}{\log(2)} \text{ bits}$$

Note:	If the PWM duty cycle value is longer than the PWM period the CCP1 pin will not be cleared.

For an example PWM period and duty cycle calculation, see the *PIC® Mid-Range Reference Manual* (DS33023).

9.3.3 SET-UP FOR PWM OPERATION

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

1. Set the PWM period by writing to the PR2 register.
2. Set the PWM duty cycle by writing to the CCPR1L register and CCP1CON<5:4> bits.
3. Make the CCP1 pin an output by clearing the TRISB<3> bit.
4. Set the TMR2 prescale value and enable Timer2 by writing to T2CON.

TABLE 9-3: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 20 MHz

PWM Frequency	1.22 kHz	4.88 kHz	19.53 kHz	78.12 kHz	156.3 kHz	208.3 kHz
Timer Prescaler (1, 4, 16)	16	4	1	1	1	1
PR2 Value	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
Maximum Resolution (bits)	10	10	10	8	7	6.5

TABLE 9-4: REGISTERS ASSOCIATED WITH PWM AND TIMER2

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR	Value on all other Resets
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	0000 -000	0000 -000
8Ch	PIE1	EEIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	0000 -000	0000 -000
86h, 186h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	1111 1111
11h	TMR2	Timer2 Module's Register								0000 0000	0000 0000
92h	PR2	Timer2 Module's Period Register								1111 1111	1111 1111
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	uuuu uuuu
15h	CCPR1L	Capture/Compare/PWM Register 1 (LSB)								xxxx xxxx	uuuu uuuu
16h	CCPR1H	Capture/Compare/PWM Register 1 (MSB)								xxxx xxxx	uuuu uuuu
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not used by PWM and Timer2.

NOTES:

TABLE 12-3: BAUD RATES FOR SYNCHRONOUS MODE

BAUD RATE (K)	Fosc = 20 MHz KBAUD	ERROR	SPBRG value (decimal)	16 MHz KBAUD	ERROR	SPBRG value (decimal)	10 MHz KBAUD	ERROR	SPBRG value (decimal)
0.3	NA	—	—	NA	—	—	NA	—	—
1.2	NA	—	—	NA	—	—	NA	—	—
2.4	NA	—	—	NA	—	—	NA	—	—
9.6	NA	—	—	NA	—	—	9.766	+1.73%	255
19.2	19.53	+1.73%	255	19.23	+0.16%	207	19.23	+0.16%	129
76.8	76.92	+0.16%	64	76.92	+0.16%	51	75.76	-1.36%	32
96	96.15	+0.16%	51	95.24	-0.79%	41	96.15	+0.16%	25
300	294.1	-1.96	16	307.69	+2.56%	12	312.5	+4.17%	7
500	500	0	9	500	0	7	500	0	4
HIGH	5000	—	0	4000	—	0	2500	—	0
LOW	19.53	—	255	15.625	—	255	9.766	—	255

BAUD RATE (K)	Fosc = 7.15909 MHz KBAUD	ERROR	SPBRG value (decimal)	5.0688 MHz KBAUD	ERROR	SPBRG value (decimal)	4 MHz KBAUD	ERROR	SPBRG value (decimal)
0.3	NA	—	—	NA	—	—	NA	—	—
1.2	NA	—	—	NA	—	—	NA	—	—
2.4	NA	—	—	NA	—	—	NA	—	—
9.6	9.622	+0.23%	185	9.6	0	131	9.615	+0.16%	103
19.2	19.24	+0.23%	92	19.2	0	65	19.231	+0.16%	51
76.8	77.82	+1.32	22	79.2	+3.13%	15	75.923	+0.16%	12
96	94.20	-1.88	18	97.48	+1.54%	12	1000	+4.17%	9
300	298.3	-0.57	5	316.8	5.60%	3	NA	—	—
500	NA	—	—	NA	—	—	NA	—	—
HIGH	1789.8	—	0	1267	—	0	100	—	0
LOW	6.991	—	255	4.950	—	255	3.906	—	255

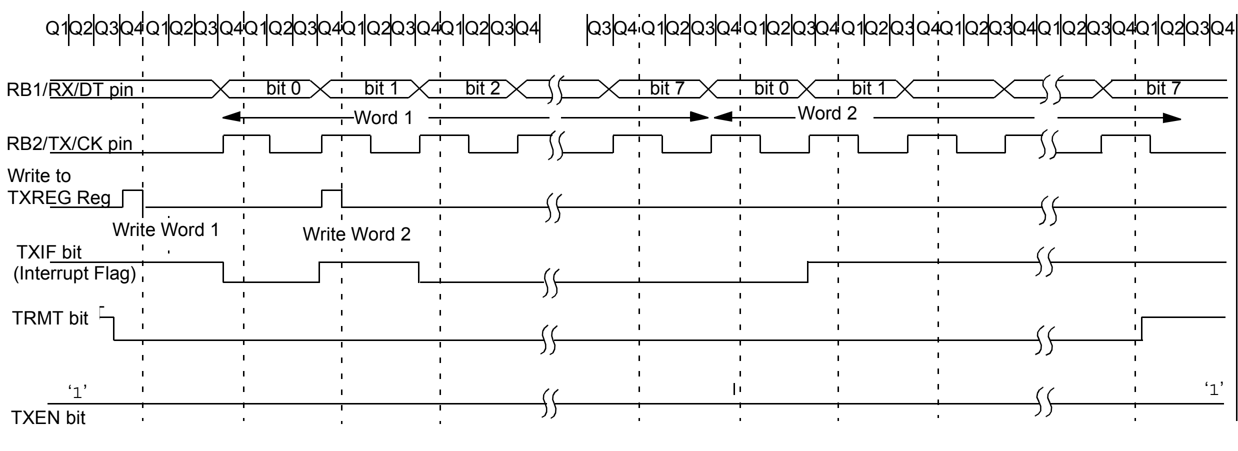
BAUD RATE (K)	Fosc = 3.579545 MHz KBAUD	ERROR	SPBRG value (decimal)	1 MHz KBAUD	ERROR	SPBRG value (decimal)	32.768 kHz KBAUD	ERROR	SPBRG value (decimal)
0.3	NA	—	—	NA	—	—	0.303	+1.14%	26
1.2	NA	—	—	1.202	+0.16%	207	1.170	-2.48%	6
2.4	NA	—	—	2.404	+0.16%	103	NA	—	—
9.6	9.622	+0.23%	92	9.615	+0.16%	25	NA	—	—
19.2	19.04	-0.83%	46	19.24	+0.16%	12	NA	—	—
76.8	74.57	-2.90%	11	83.34	+8.51%	2	NA	—	—
96	99.43	+3.57%	8	NA	—	—	NA	—	—
300	298.3	0.57%	2	NA	—	—	NA	—	—
500	NA	—	—	NA	—	—	NA	—	—
HIGH	894.9	—	0	250	—	0	8.192	—	0
LOW	3.496	—	255	0.9766	—	255	0.032	—	255

TABLE 12-9: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR	Value on all other Resets
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	0000 -000	0000 -000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADEN	FERR	OERR	RX9D	0000 000x	0000 000x
19h	TXREG	USART Transmit Data Register								0000 0000	0000 0000
8Ch	PIE1	EEIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	0000 -000	0000 -000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented, read as '0'. Shaded cells are not used for synchronous master transmission.

FIGURE 12-8: SYNCHRONOUS TRANSMISSION



Note: Sync Master Mode; SPBRG = 0. Continuous transmission of two 8-bit words.

FIGURE 12-9: SYNCHRONOUS TRANSMISSION (THROUGH TXEN)

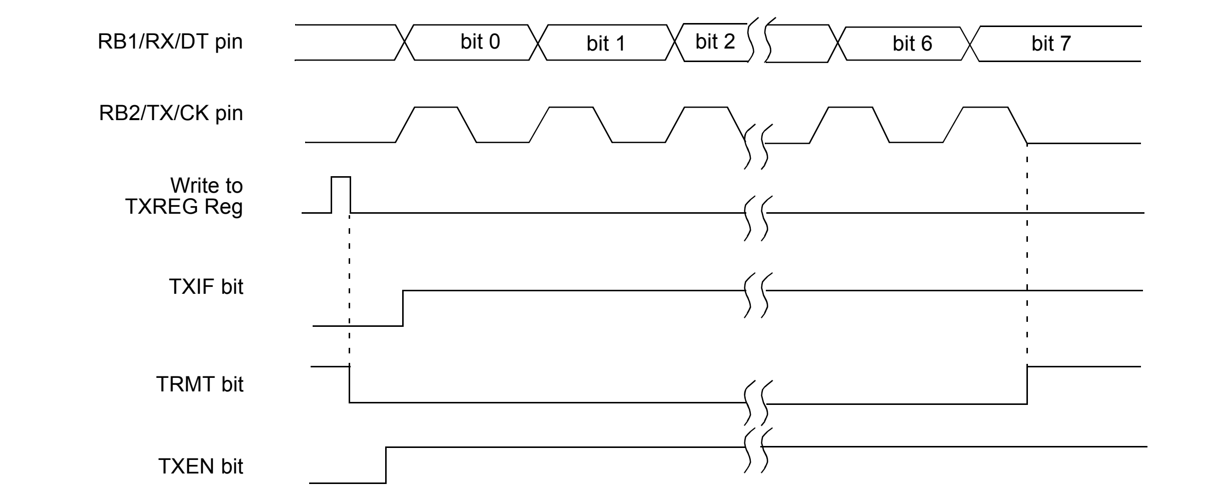
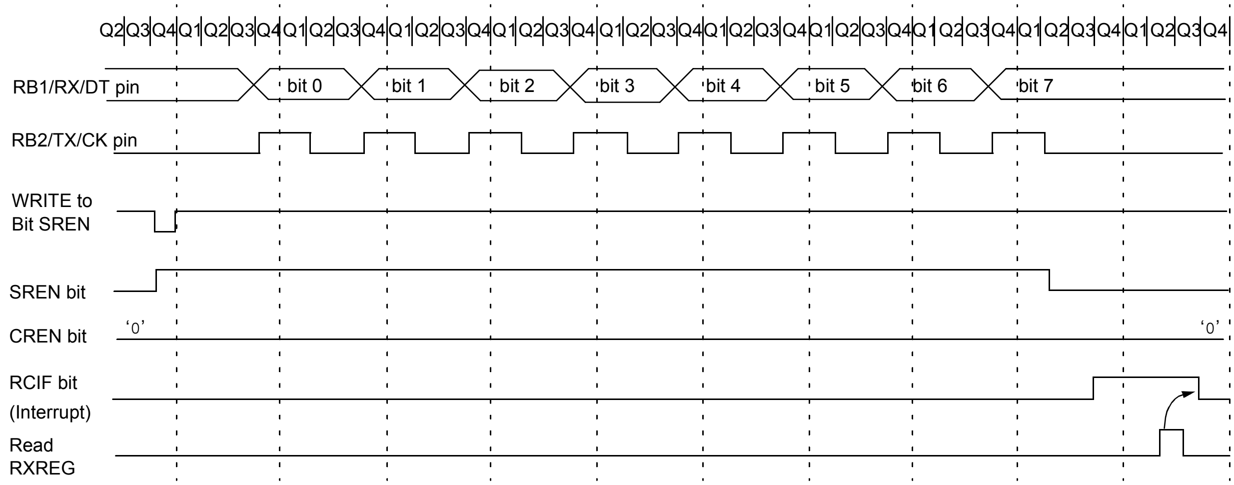


FIGURE 12-10: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)



Note: Timing diagram demonstrates Sync Master mode with bit SREN = `1` and bit BRG = `0`.

12.5 USART Synchronous Slave Mode

Synchronous Slave mode differs from the Master mode in the fact that the shift clock is supplied externally at the RB2/TX/CK pin (instead of being supplied internally in Master mode). This allows the device to transfer or receive data while in Sleep mode. Slave mode is entered by clearing bit CSRC (TXSTA<7>).

12.5.1 USART SYNCHRONOUS SLAVE TRANSMIT

The operation of the Synchronous Master and Slave modes are identical except in the case of the Sleep mode.

If two words are written to the TXREG and then the `SLEEP` instruction is executed, the following will occur:

a) The first word will immediately transfer to the TSR register and transmit.
b) The second word will remain in TXREG register.
c) Flag bit TXIF will not be set.
d) When the first word has been shifted out of TSR, the TXREG register will transfer the second word to the TSR and flag bit TXIF will now be set.
e) If enable bit TXIE is set, the interrupt will wake the chip from Sleep and if the global interrupt is enabled, the program will branch to the interrupt vector (0004h).

Follow these steps when setting up a Synchronous Slave Transmission:

1. TRISB<1> and TRISB<2> should both be set to '`1`' to configure the RB1/RX/DT and RB2/TX/CK pins as inputs. Output drive, when required, is controlled by the peripheral circuitry.
2. Enable the synchronous slave serial port by setting bits SYNC and SPEN and clearing bit CSRC.
3. Clear bits CREN and SREN.
4. If interrupts are desired, then set enable bit TXIE.
5. If 9-bit transmission is desired, then set bit TX9.
6. Enable the transmission by setting enable bit TXEN.
7. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
8. Start transmission by loading data to the TXREG register.

13.7 Using the Data EEPROM

The data EEPROM is a high endurance, byte addressable array that has been optimized for the storage of frequently changing information (e.g., program variables or other data that are updated often). When variables in one section change frequently, while variables in another section do not change, it is possible to exceed the total number of write cycles to the EEPROM (specification D124) without exceeding the total number of write cycles to a single byte (specifications D120 and D120A). If this is the case, then an array refresh must be performed. For this reason, variables that change infrequently (such as constants, IDs, calibration, etc.) should be stored in Flash program memory.

A simple data EEPROM refresh routine is shown in Example 13-4.

Note:	If data EEPROM is only used to store constants and/or data that changes rarely, an array refresh is likely not required. See specification D124.

EXAMPLE 13-4: DATA EEPROM REFRESH ROUTINE

```
    BANKSEL    0X80           ;select Bank1
    CLRF       EEADR          ;start at address 0
    BCF        INTCON, GIE    ;disable interrupts
    BTFSC      INTCON, GIE    ;see AN576
    GOTO       $ - 2
    BSF        EECON1, WREN   ;enable EE writes
Loop
    BSF        EECON1, RD     ;retrieve data into EEDATA
    MOVLW      0x55           ;first step of ...
    MOVWF      EECON2         ;... required sequence
    MOVLW      0xAA           ;second step of ...
    MOVWF      EECON2         ;... required sequence
    BSF        EECON1, WR     ;start write sequence
    BTFSC      EECON1, WR     ;wait for write complete
    GOTO       $ - 1

#IFDEF __16F648A              ;256 bytes in 16F648A

    INCFSZ     EEADR, f       ;test for end of memory
#ELSE                         ;128 bytes in 16F627A/628A
    INCF       EEADR, f       ;next address
    BTFSS      EEADR, 7       ;test for end of memory
#ENDIF                        ;end of conditional assembly

    GOTO       Loop           ;repeat for all locations

    BCF        EECON1, WREN   ;disable EE writes
    BSF        INTCON, GIE    ;enable interrupts (optional)
```

14.2.3 EXTERNAL CRYSTAL OSCILLATOR CIRCUIT

Either a prepackaged oscillator can be used or a simple oscillator circuit with TTL gates can be built. Prepackaged oscillators provide a wide operating range and better stability. A well-designed crystal oscillator will provide good performance with TTL gates. Two types of crystal oscillator circuits can be used; one with series resonance, or one with parallel resonance.

Figure 14-2 shows implementation of a parallel resonant oscillator circuit. The circuit is designed to use the fundamental frequency of the crystal. The 74AS04 inverter performs the 180° phase shift that a parallel oscillator requires. The 4.7 kΩ resistor provides the negative feedback for stability. The 10 kΩ potentiometers bias the 74AS04 in the linear region. This could be used for external oscillator designs.

FIGURE 14-2: EXTERNAL PARALLEL RESONANT CRYSTAL OSCILLATOR CIRCUIT

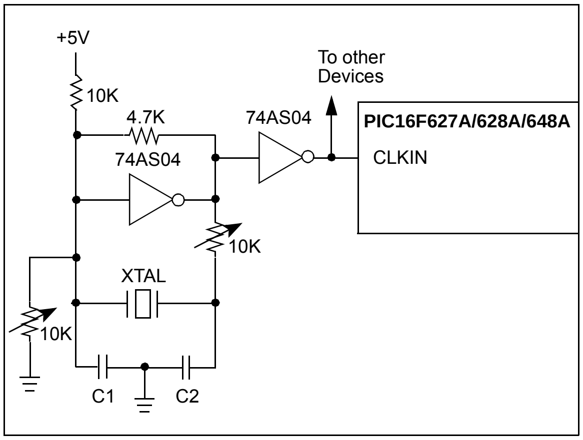
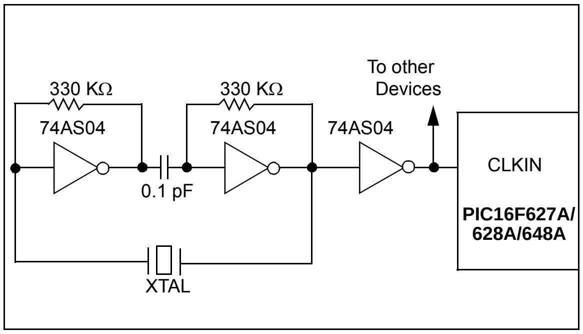


Figure 14-3 shows a series resonant oscillator circuit. This circuit is also designed to use the fundamental frequency of the crystal. The inverter performs a 180° phase shift in a series resonant oscillator circuit. The 330 kΩ resistors provide the negative feedback to bias the inverters in their linear region.

FIGURE 14-3: EXTERNAL SERIES RESONANT CRYSTAL OSCILLATOR CIRCUIT



14.2.4 PRECISION INTERNAL 4 MHZ OSCILLATOR

The internal precision oscillator provides a fixed 4 MHz (nominal) system clock at VDD = 5V and 25°C. See **Section 17.0 "Electrical Specifications"**, for information on variation over voltage and temperature.

14.2.5 EXTERNAL CLOCK IN

For applications where a clock is already available elsewhere, users may directly drive the PIC16F627A/628A/648A provided that this external clock source meets the AC/DC timing requirements listed in **Section 17.6 "Timing Diagrams and Specifications"**. Figure 14-4 below shows how an external clock circuit should be configured.

FIGURE 14-4: EXTERNAL CLOCK INPUT OPERATION (EC, HS, XT OR LP OSC CONFIGURATION)

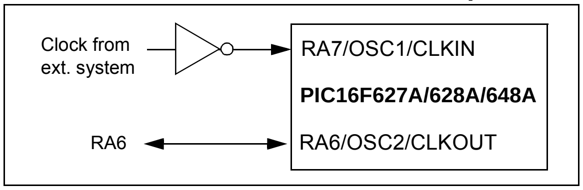
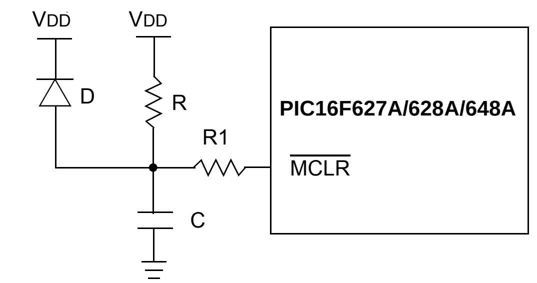


FIGURE 14-11: EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW VDD POWER-UP)

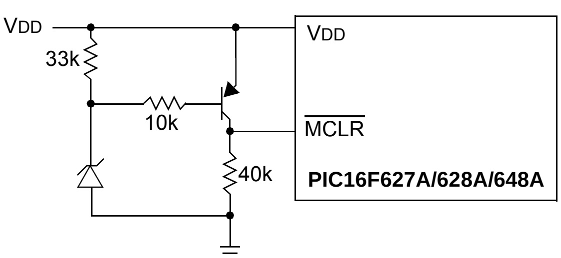


Note 1: External Power-on Reset circuit is required only if VDD power-up slope is too slow. The diode D helps discharge the capacitor quickly when VDD powers down.

2: R < 40 kΩ is recommended to make sure that voltage drop across R does not violate the device's electrical specification.

3: R1 = 100Ω to 1 kΩ will limit any current flowing into MCLR from external capacitor C in the event of MCLR/VPP pin breakdown due to Electrostatic Discharge (ESD) or Electrical Overstress (EOS).

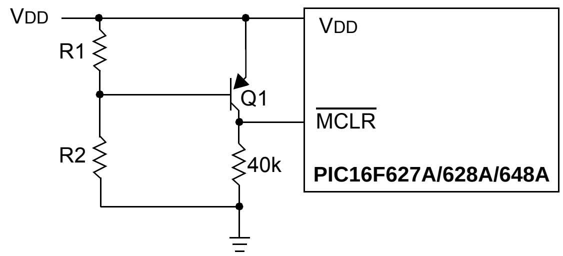
FIGURE 14-12: EXTERNAL BROWN-OUT PROTECTION CIRCUIT 1



Note 1: This circuit will activate Reset when VDD goes below (Vz + 0.7V) where Vz = Zener voltage.

2: Internal Brown-out Reset circuitry should be disabled when using this circuit.

FIGURE 14-13: EXTERNAL BROWN-OUT PROTECTION CIRCUIT 2



Note 1: This brown-out circuit is less expensive, albeit less accurate. Transistor Q1 turns off when VDD is below a certain level such that:

$$V_{DD} \times \frac{R1}{R1 + R2} = 0.7\text{ V}$$

2: Internal Brown-out Reset should be disabled when using this circuit.

3: Resistors should be adjusted for the characteristics of the transistor.

NOTES:

TABLE 15-2: PIC16F627A/628A/648A INSTRUCTION SET

Mnemonic,	Operands	Description	Cycles	14-Bit Opcode MSb			LSb	Status Affected	Notes
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C,DC,Z	**1, 2**
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z	**1, 2**
CLRF	f	Clear f	1	00	0001	lfff	ffff	Z	**2**
CLRW	—	Clear W	1	00	0001	0xxx	xxxx	Z	
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z	**1, 2**
DECF	f, d	Decrement f	1	00	0011	dfff	ffff	Z	**1, 2**
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		**1, 2, 3**
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z	**1, 2**
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		**1, 2, 3**
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	**1, 2**
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z	**1, 2**
MOVWF	f	Move W to f	1	00	0000	lfff	ffff		
NOP	—	No Operation	1	00	0000	0xx0	0000		
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	**1, 2**
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	**1, 2**
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C,DC,Z	**1, 2**
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff		**1, 2**
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	**1, 2**
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff		**1, 2**
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff		**1, 2**
BTFSC	f, b	Bit Test f, Skip if Clear	1(2)	01	10bb	bfff	ffff		**3**
BTFSS	f, b	Bit Test f, Skip if Set	1(2)	01	11bb	bfff	ffff		**3**
LITERAL AND CONTROL OPERATIONS									
ADDLW	k	Add literal and W	1	11	111x	kkkk	kkkk	C,DC,Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
CALL	k	Call subroutine	2	10	0kkk	kkkk	kkkk		
CLRWDT	—	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}$,$\overline{PD}$	
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk		
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLW	k	Move literal to W	1	11	00xx	kkkk	kkkk		
RETFIE	—	Return from interrupt	2	00	0000	0000	1001		
RETLW	k	Return with literal in W	2	11	01xx	kkkk	kkkk		
RETURN	—	Return from Subroutine	2	00	0000	0000	1000		
SLEEP	—	Go into Standby mode	1	00	0000	0110	0011	$\overline{TO}$,$\overline{PD}$	
SUBLW	k	Subtract W from literal	1	11	110x	kkkk	kkkk	C,DC,Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

Note 1: When an I/O register is modified as a function of itself (e.g., `MOVF PORTB, 1`), the value used will be that value present on the pins themselves. For example, if the data latch is '`1`' for a pin configured as input and is driven low by an external device, the data will be written back with a '`0`'.

2: If this instruction is executed on the TMR0 register (and, where applicable, d = `1`), the prescaler will be cleared if assigned to the Timer0 Module.

3: If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a `NOP`.

RETLW	Return with Literal in W
Syntax:	[*label*] RETLW k
Operands:	$0 \leq k \leq 255$
Operation:	k → (W); TOS → PC
Status Affected:	None
Encoding:	`11` | `01xx` | `kkkk` | `kkkk`
Description:	The W register is loaded with the eight-bit literal ‘k’. The program counter is loaded from the top of the stack (the return address). This is a two-cycle instruction.
Words:	1
Cycles:	2

Example

```
              CALL TABLE;W contains table
                 ;offset value
              •  ;W now has table value
              •
TABLE         •
              ADDWF PC;W = offset
              RETLW k1;Begin table
              RETLW k2;
              •
              •
              •
              RETLW kn; End of table
```

Before Instruction
W = 0x07
After Instruction
W = value of k8

RETURN	Return from Subroutine
Syntax:	[*label*] RETURN
Operands:	None
Operation:	TOS → PC
Status Affected:	None
Encoding:	`00` | `0000` | `0000` | `1000`
Description:	Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a two-cycle instruction.
Words:	1
Cycles:	2

Example

```
RETURN
```

After Interrupt
PC = TOS

RLF	Rotate Left f through Carry
Syntax:	[*label*] RLF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	See description below
Status Affected:	C
Encoding:	`00` | `1101` | `dfff` | `ffff`
Description:	The contents of register ‘f’ are rotated one bit to the left through the Carry Flag. If ‘d’ is ‘0’, the result is placed in the W register. If ‘d’ is ‘1’, the result is stored back in register ‘f’.

Words:	1
Cycles:	1

Example

```
RLF      REG1, 0
```

Before Instruction
REG1 = 1110 0110
C = 0
After Instruction
REG1 = 1110 0110
W = 1100 1100
C = 1

16.2 MPLAB C Compilers for Various Device Families

The MPLAB C Compiler code development systems are complete ANSI C compilers for Microchip's PIC18, PIC24 and PIC32 families of microcontrollers and the dsPIC30 and dsPIC33 families of digital signal controllers. These compilers provide powerful integration capabilities, superior code optimization and ease of use.

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

16.3 HI-TECH C for Various Device Families

The HI-TECH C Compiler code development systems are complete ANSI C compilers for Microchip's PIC family of microcontrollers and the dsPIC family of digital signal controllers. These compilers provide powerful integration capabilities, omniscient code generation and ease of use.

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

The compilers include a macro assembler, linker, preprocessor, and one-step driver, and can run on multiple platforms.

16.4 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 IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multi-purpose source files
- Directives that allow complete control over the assembly process

16.5 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler and the MPLAB C18 C Compiler. 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

16.6 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC devices. MPLAB C 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 IDE compatibility

17.2 DC Characteristics: PIC16F627A/628A/648A (Industrial) PIC16LF627A/628A/648A (Industrial)

DC CHARACTERISTICS		Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ Ta ≤ +85°C for industrial					
Param No.	LF and F Device Characteristics	Min†	Typ	Max	Units	Conditions: VDD	Conditions: Note
Supply Voltage (VDD)							
D001	LF	2.0	—	5.5	V	—	
	LF/F	3.0	—	5.5	V	—	
Power-down Base Current (IPD)							
D020	LF	—	0.01	0.80	μA	2.0	WDT, BOR, Comparators, VREF and T1OSC: disabled
	LF/F	—	0.01	0.85	μA	3.0	
		—	0.02	2.7	μA	5.0	
Peripheral Module Current (ΔIMOD)(1)							
D021	LF	—	1	2.0	μA	2.0	WDT Current
	LF/F	—	2	3.4	μA	3.0	
		—	9	17.0	μA	5.0	
D022	LF/F	—	29	52	μA	4.5	BOR Current
		—	30	55	μA	5.0	
D023	LF	—	15	22	μA	2.0	Comparator Current (Both comparators enabled)
	LF/F	—	22	37	μA	3.0	
		—	44	68	μA	5.0	
D024	LF	—	34	55	μA	2.0	VREF Current
	LF/F	—	50	75	μA	3.0	
		—	80	110	μA	5.0	
D025	LF	—	1.2	2.0	μA	2.0	T1OSC Current
	LF/F	—	1.3	2.2	μA	3.0	
		—	1.8	2.9	μA	5.0	
Supply Current (IDD)							
D010	LF	—	10	15	μA	2.0	FOSC = 32 kHz LP Oscillator Mode
	LF/F	—	15	25	μA	3.0	
		—	28	48	μA	5.0	
D011	LF	—	125	190	μA	2.0	FOSC = 1 MHz XT Oscillator Mode
	LF/F	—	175	340	μA	3.0	
		—	320	520	μA	5.0	
D012	LF	—	250	350	μA	2.0	FOSC = 4 MHz XT Oscillator Mode
	LF/F	—	450	600	μA	3.0	
		—	710	995	μA	5.0	
D012A	LF	—	395	465	μA	2.0	FOSC = 4 MHz INTOSC
	LF/F	—	565	785	μA	3.0	
		—	0.895	1.3	mA	5.0	
D013	LF/F	—	2.5	2.9	mA	4.5	FOSC = 20 MHz HS Oscillator Mode
		—	2.75	3.3	mA	5.0	

Note 1: The "Δ" current is the additional current consumed when this peripheral is enabled. This current should be added to the base IDD or IPD measurement. Max values should be used when calculating total current consumption.

FIGURE 17-7: BROWN-OUT RESET TIMING

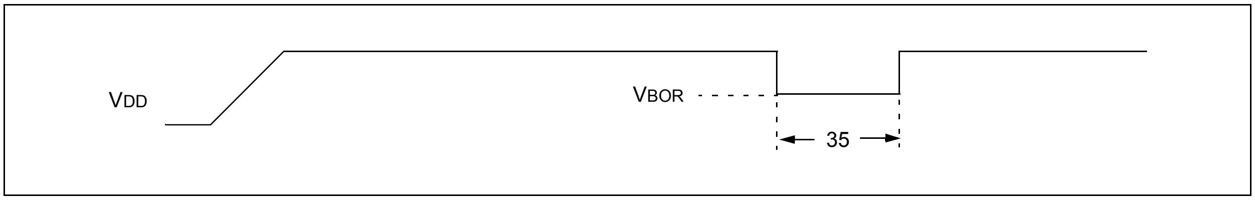


TABLE 17-7: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER REQUIREMENTS

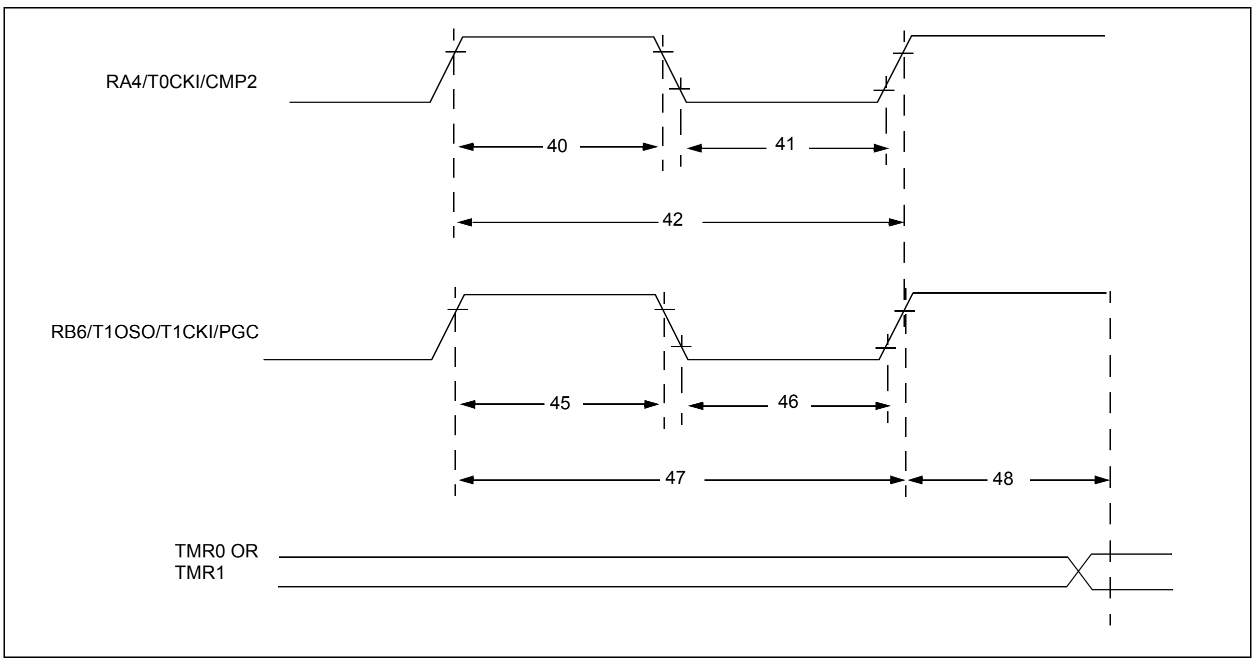
Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
30	TMCL	MCLR Pulse Width (low)	2000	—	—	ns	VDD = 5V, -40°C to +85°C
31	TWDT	Watchdog Timer Time out Period (No Prescaler)	7*	18	33*	ms	VDD = 5V, -40°C to +85°C
32	TOST	Oscillation Start-up Timer Period	—	1024 TOSC	—	—	TOSC = OSC1 period
33	TPWRT	Power-up Timer Period	28*	72	132*	ms	VDD = 5V, -40°C to +85°C
34	TIOZ	I/O High-impedance from MCLR Low or Watchdog Timer Reset	—	—	2.0*	μs	
35	TBOR	Brown-out Reset pulse width	100*	—	—	μs	VDD ≤ VBOR (D005)

Legend: TBD = To Be Determined.

* These parameters are characterized but not tested.

† Data in “Typ” column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 17-8: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS



18.0 DC AND AC CHARACTERISTICS GRAPHS AND TABLES

The graphs and tables provided in this section are for **design guidance** and are **not tested**.

In some graphs or tables, the data presented are **outside specified operating range** (i.e., outside specified VDD range). This is for **information only** and devices are ensured to operate properly only within the specified range.

The data presented in this section is a **statistical summary** of data collected on units from different lots over a period of time and matrix samples. 'Typical' represents the mean of the distribution at 25°C. 'Max' or 'Min' represents (mean + 3σ) or (mean - 3σ) respectively, where σ is standard deviation, over the whole temperature range.

FIGURE 18-1: TYPICAL BASELINE IPD vs. VDD (-40°C TO 25°C)

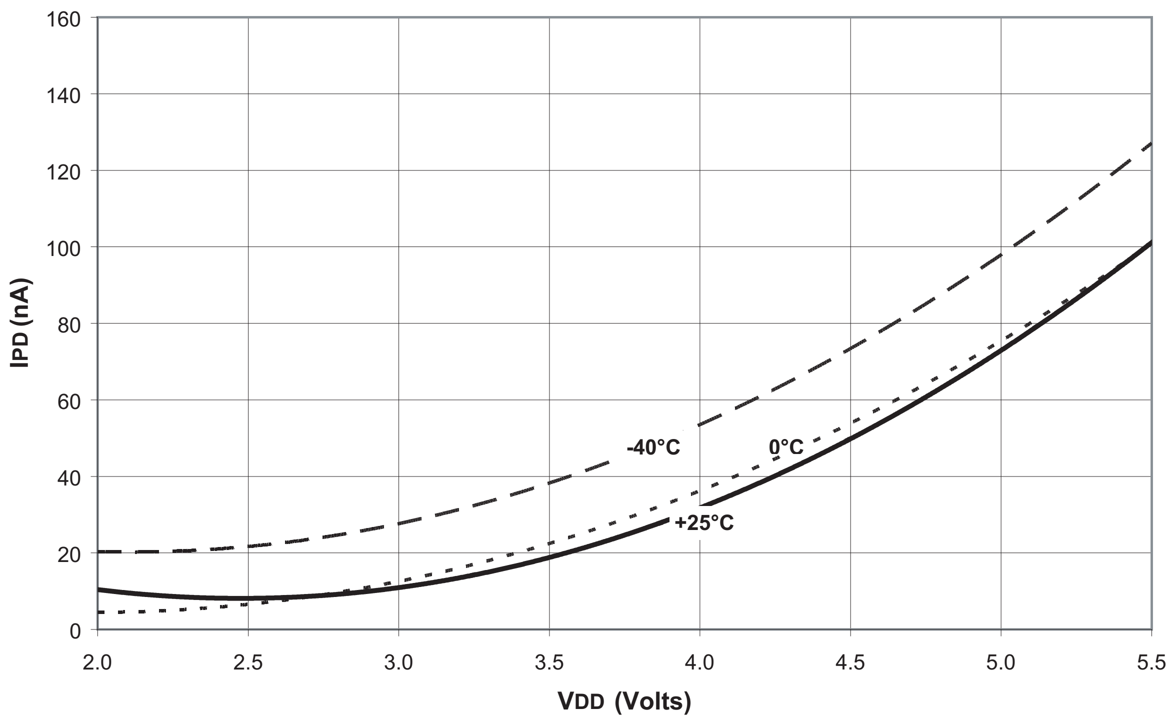


FIGURE 18-6: TYPICAL VREF IPD vs. VDD

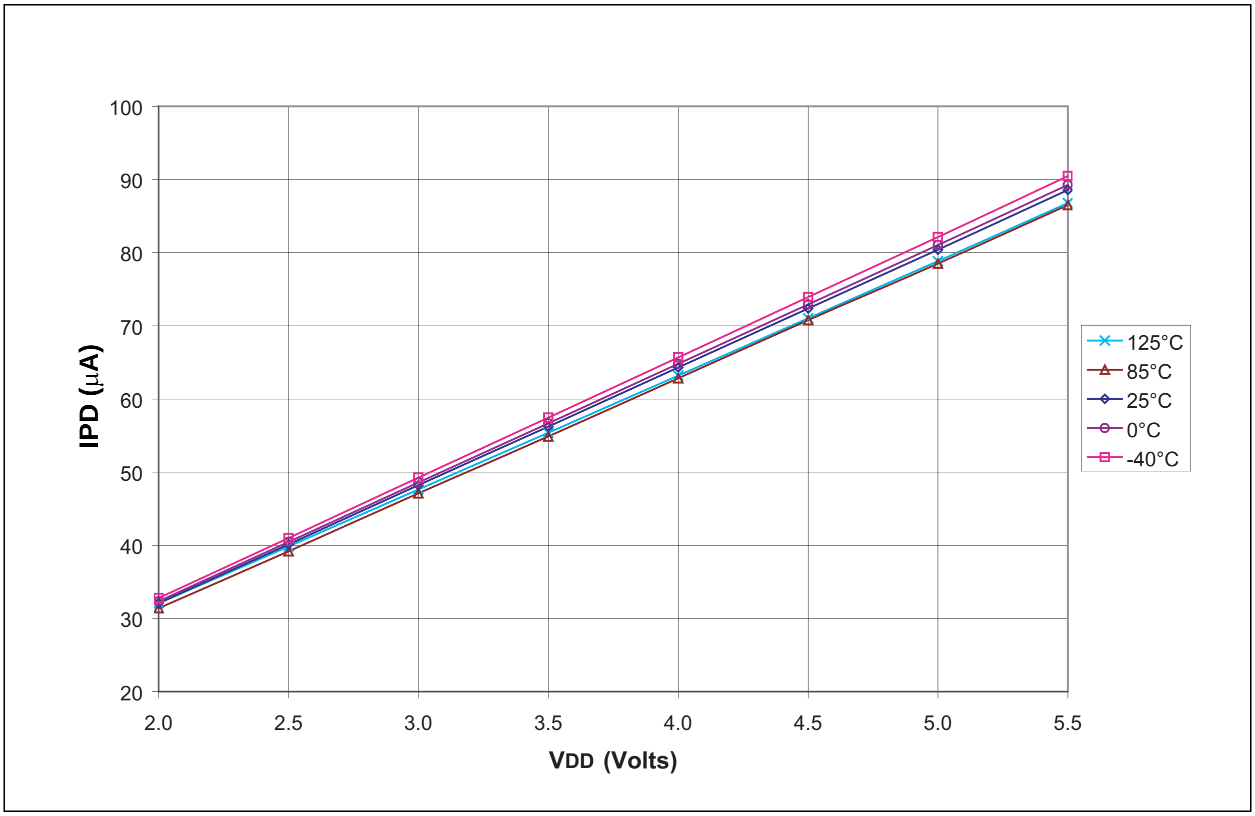


FIGURE 18-7: TYPICAL WDT IPD vs. VDD