



Welcome to [E-XFL.COM](https://www.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.

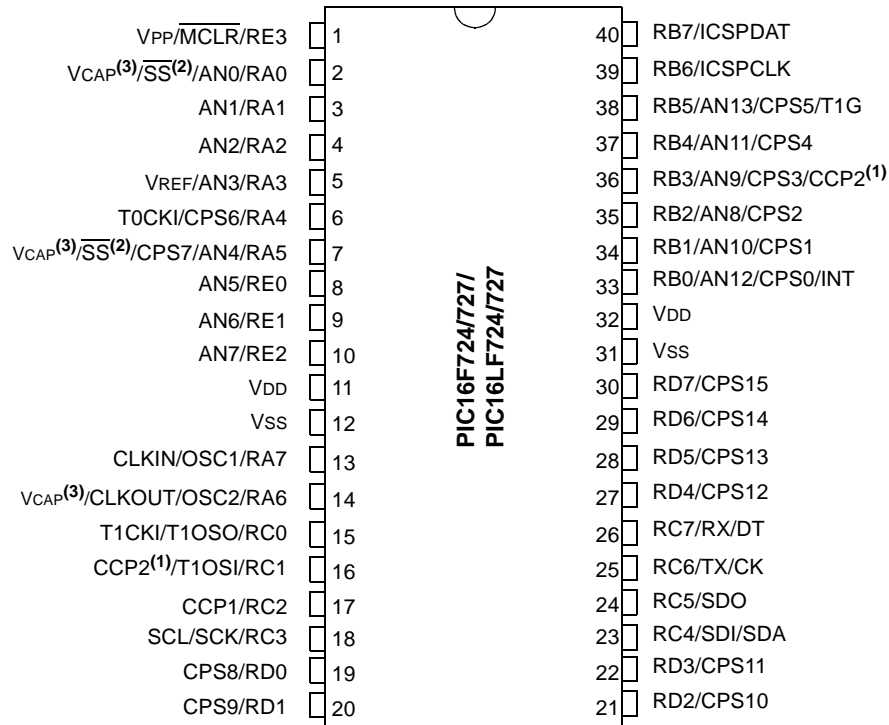
Applications of "[Embedded - Microcontrollers](#)"

Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	20MHz
Connectivity	I ² C, 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	-
RAM Size	368 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 14x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	40-UQFN Exposed Pad
Supplier Device Package	40-UQFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f727-i-mv

PIC16(L)F722/3/4/6/7

Pin Diagrams – 40-PIN PDIP (PIC16F724/727/PIC16LF724/727)



- Note 1:** CCP2 pin location may be selected as RB3 or RC1.
Note 2: SS pin location may be selected as RA5 or RA0.
Note 3: PIC16F724/727 devices only.

FIGURE 1-1: PIC16F722/723/726/PIC16LF722/723/726 BLOCK DIAGRAM



EXAMPLE 4-1: SAVING W, STATUS AND PCLATH REGISTERS IN RAM

```
MOVWF    W_TEMP           ;Copy W to W_TEMP register
SWAPF    STATUS,W         ;Swap status to be saved into W
                        ;Swaps are used because they do not affect the status bits
BANKSEL   STATUS_TEMP      ;Select regardless of current bank
MOVWF    STATUS_TEMP      ;Copy status to bank zero STATUS_TEMP register
MOVF     PCLATH,W         ;Copy PCLATH to W register
MOVWF    PCLATH_TEMP      ;Copy W register to PCLATH_TEMP
:
:(ISR)    ;Insert user code here
:
BANKSEL   STATUS_TEMP      ;Select regardless of current bank
MOVF     PCLATH_TEMP,W    ;
MOVWF    PCLATH           ;Restore PCLATH
SWAPF    STATUS_TEMP,W    ;Swap STATUS_TEMP register into W
                        ;(sets bank to original state)
MOVWF    STATUS           ;Move W into STATUS register
SWAPF    W_TEMP,F         ;Swap W_TEMP
SWAPF    W_TEMP,W         ;Swap W_TEMP into W
```

4.5.1 INTCON REGISTER

The INTCON register is a readable and writable register, which contains the various enable and flag bits for TMR0 register overflow, PORTB change and external RB0/INT/SEG0 pin interrupts.

Note: Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the global enable bit, GIE of the INTCON register. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

6.5 PORTD and TRISD Registers

PORTD is a 8-bit wide, bidirectional port. The corresponding data direction register is TRISD (Register 6-13). Setting a TRISD bit (= 1) will make the corresponding PORTD pin an input (i.e., put the corresponding output driver in a High-Impedance mode). Clearing a TRISD bit (= 0) will make the corresponding PORTD pin an output (i.e., enable the output driver and put the contents of the output latch on the selected pin). Example 6-4 shows how to initialize PORTD.

Reading the PORTD register (Register 6-12) reads the status of the pins, whereas writing to it will write to the PORT latch. All write operations are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, this value is modified and then written to the PORT data latch.

Note: PORTD is available on PIC16F724/LF724 and PIC16F727/LF727 only.

The TRISD register (Register 6-13) controls the PORTD pin output drivers, even when they are being used as analog inputs. The user should ensure the bits in the TRISD register are maintained set when using them as analog inputs. I/O pins configured as analog input always read '0'.

EXAMPLE 6-4: INITIALIZING PORTD

```
BANKSEL PORTD      ;
CLRF   PORTD        ;Init PORTD
BANKSEL ANSELDD     ;
CLRF   ANSELDD      ;Make PORTD digital
BANKSEL TRISD       ;
MOVLW  B'00001100' ;Set RD<3:2> as inputs
MOVWF  TRISD        ;and set RD<7:4,1:0>
                      ;as outputs
```

6.5.1 ANSELDD REGISTER

The ANSELDD register (Register 6-9) is used to configure the Input mode of an I/O pin to analog. Setting the appropriate ANSELDD bit high will cause all digital reads on the pin to be read as '0' and allow analog functions on the pin to operate correctly.

The state of the ANSELDD bits has no affect on digital output functions. A pin with TRIS clear and ANSEL set will still operate as a digital output, but the Input mode will be analog. This can cause unexpected behavior when executing read-modify-write instructions on the affected port.

Note: The ANSELDD register must be initialized to configure an analog channel as a digital input. Pins configured as analog inputs will read '0'.

REGISTER 6-12: PORTD: PORTD REGISTER⁽¹⁾

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
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'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 7-0 **RD<7:0>**: PORTD General Purpose I/O Pin bits

1 = Port pin is > V_{IH}

0 = Port pin is < V_{IL}

Note 1: PORTD is not implemented on PIC16F722/723/726/PIC16LF722/723/726 devices, read as '0'.

6.5.6 RD4/CPS12

Figure 6-21 shows the diagram for these pins. They are configurable to function as one of the following:

- a general purpose I/O
- a capacitive sensing input

6.5.7 RD5/CPS13

Figure 6-21 shows the diagram for these pins. They are configurable to function as one of the following:

- a general purpose I/O
- a capacitive sensing input

6.5.8 RD6/CPS14

Figure 6-21 shows the diagram for these pins. They are configurable to function as one of the following:

- a general purpose I/O
- a capacitive sensing input

6.5.9 RD7/CPS15

Figure 6-21 shows the diagram for these pins. They are configurable to function as one of the following:

- a general purpose I/O
- a capacitive sensing input

FIGURE 6-21: BLOCK DIAGRAM OF RD<7:0>

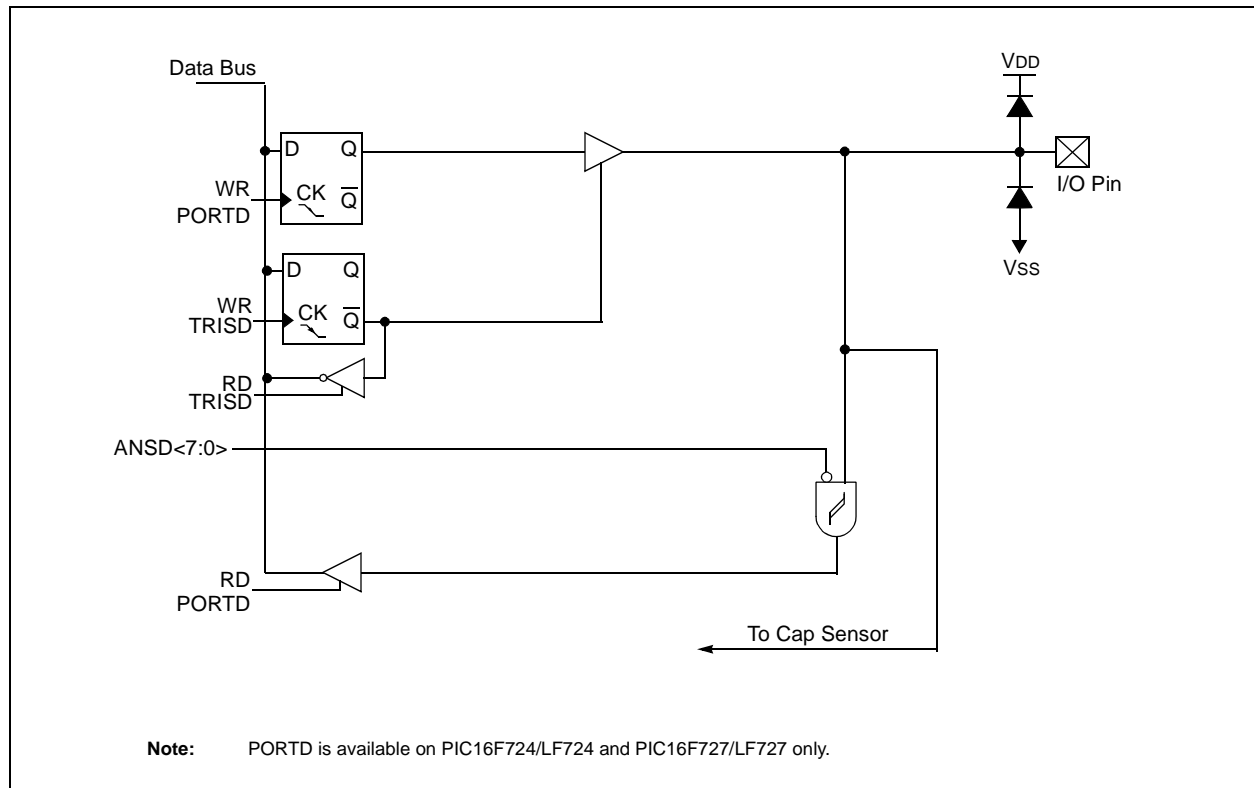


TABLE 6-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTD⁽¹⁾

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
ANSELD	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	1111 1111
CPSCON0	CPSON	—	—	—	CPSRNG1	CPSRNG0	CPSOUT	T0XCS	0--- 0000	0--- 0000
CPSCON1	—	—	—	—	CPSCH3	CPSCH2	CPSCH1	CPSCH0	---- 0000	---- 0000
PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	xxxx xxxx	xxxx xxxx
TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	1111 1111	1111 1111

Legend: x = unknown, u = unchanged, — = unimplemented locations read as '0'. Shaded cells are not used by PORTD.

Note 1: These registers are not implemented on the PIC16F722/723/726/PIC16LF722/723/726 devices, read as '0'.

PIC16(L)F722/3/4/6/7

TABLE 12-5: WDT/TIMER1 GATE INTERACTION

WDTE	TMR1GE = 1 and T1GSS = 11	WDT Oscillator Enable	WDT Reset	Wake-up	WDT Available for T1G Source
1	N	Y	Y	Y	N
1	Y	Y	Y	Y	Y
0	Y	Y	N	N	Y
0	N	N	N	N	N

12.6.3 TIMER1 GATE TOGGLE MODE

When Timer1 Gate Toggle mode is enabled, it is possible to measure the full-cycle length of a Timer1 gate signal, as opposed to the duration of a single level pulse.

The Timer1 Gate source is routed through a flip-flop that changes state on every incrementing edge of the signal. See Figure 12-4 for timing details.

Timer1 Gate Toggle mode is enabled by setting the T1GTM bit of the T1GCON register. When the T1GTM bit is cleared, the flip-flop is cleared and held clear. This is necessary in order to control which edge is measured.

Note: Enabling Toggle mode at the same time as changing the gate polarity may result in indeterminate operation.

12.6.4 TIMER1 GATE SINGLE-PULSE MODE

When Timer1 Gate Single-Pulse mode is enabled, it is possible to capture a single pulse gate event. Timer1 Gate Single-Pulse mode is first enabled by setting the T1GSPM bit in the T1GCON register. Next, the T1GGO/DONE bit in the T1GCON register must be set. The Timer1 will be fully enabled on the next incrementing edge. On the next trailing edge of the pulse, the T1GGO/DONE bit will automatically be cleared. No other gate events will be allowed to increment Timer1 until the T1GGO/DONE bit is once again set in software.

Clearing the T1GSPM bit of the T1GCON register will also clear the T1GGO/DONE bit. See Figure 12-5 for timing details.

Enabling the Toggle mode and the Single-Pulse mode simultaneously will permit both sections to work together. This allows the cycle times on the Timer1 Gate source to be measured. See Figure 12-6 for timing details.

12.6.5 TIMER1 GATE VALUE STATUS

When Timer1 Gate Value Status is utilized, it is possible to read the most current level of the gate control value. The value is stored in the T1GVAL bit in the T1GCON register. The T1GVAL bit is valid even when the Timer1 Gate is not enabled (TMR1GE bit is cleared).

12.6.6 TIMER1 GATE EVENT INTERRUPT

When Timer1 Gate Event Interrupt is enabled, it is possible to generate an interrupt upon the completion of a gate event. When the falling edge of T1GVAL occurs, the TMR1GIF flag bit in the PIR1 register will be set. If the TMR1GIE bit in the PIE1 register is set, then an interrupt will be recognized.

The TMR1GIF flag bit operates even when the Timer1 Gate is not enabled (TMR1GE bit is cleared).

12.7 Timer1 Interrupt

The Timer1 register pair (TMR1H:TMR1L) increments to FFFFh and rolls over to 0000h. When Timer1 rolls over, the Timer1 interrupt flag bit of the PIR1 register is set. To enable the interrupt on rollover, you must set these bits:

- TMR1ON bit of the T1CON register
- TMR1IE bit of the PIE1 register
- PEIE bit of the INTCON register
- GIE bit of the INTCON register

The interrupt is cleared by clearing the TMR1IF bit in the Interrupt Service Routine.

Note: The TMR1H:TMR1L register pair and the TMR1IF bit should be cleared before enabling interrupts.

12.8 Timer1 Operation During Sleep

Timer1 can only operate during Sleep when setup in Asynchronous Counter mode. In this mode, an external crystal or clock source can be used to increment the counter. To set up the timer to wake the device:

- TMR1ON bit of the T1CON register must be set
- TMR1IE bit of the PIE1 register must be set
- PEIE bit of the INTCON register must be set
- $\overline{T1SYNC}$ bit of the T1CON register must be set
- TMR1CS bits of the T1CON register must be configured
- T1OSCEN bit of the T1CON register must be configured
- TMR1GIE bit of the T1GCON register must be configured

The device will wake-up on an overflow and execute the next instructions. If the GIE bit of the INTCON register is set, the device will call the Interrupt Service Routine (0004h).

12.9 CCP Capture/Compare Time Base

The CCP module uses the TMR1H:TMR1L register pair as the time base when operating in Capture or Compare mode.

In Capture mode, the value in the TMR1H:TMR1L register pair is copied into the CCPR1H:CCPR1L register pair on a configured event.

In Compare mode, an event is triggered when the value CCPR1H:CCPR1L register pair matches the value in the TMR1H:TMR1L register pair. This event can be a Special Event Trigger.

For more information, see **Section 15.0 “Capture/Compare/PWM (CCP) Module”**.

12.10 CCP Special Event Trigger

When the CCP is configured to trigger a special event, the trigger will clear the TMR1H:TMR1L register pair. This special event does not cause a Timer1 interrupt. The CCP module may still be configured to generate a CCP interrupt.

In this mode of operation, the CCPR1H:CCPR1L register pair becomes the period register for Timer1.

Timer1 should be synchronized to the $F_{osc}/4$ to utilize the Special Event Trigger. Asynchronous operation of Timer1 can cause a Special Event Trigger to be missed.

In the event that a write to TMR1H or TMR1L coincides with a Special Event Trigger from the CCP, the write will take precedence.

For more information, see **Section 9.2.5 “Special Event Trigger”**.

FIGURE 12-2: TIMER1 INCREMENTING EDGE

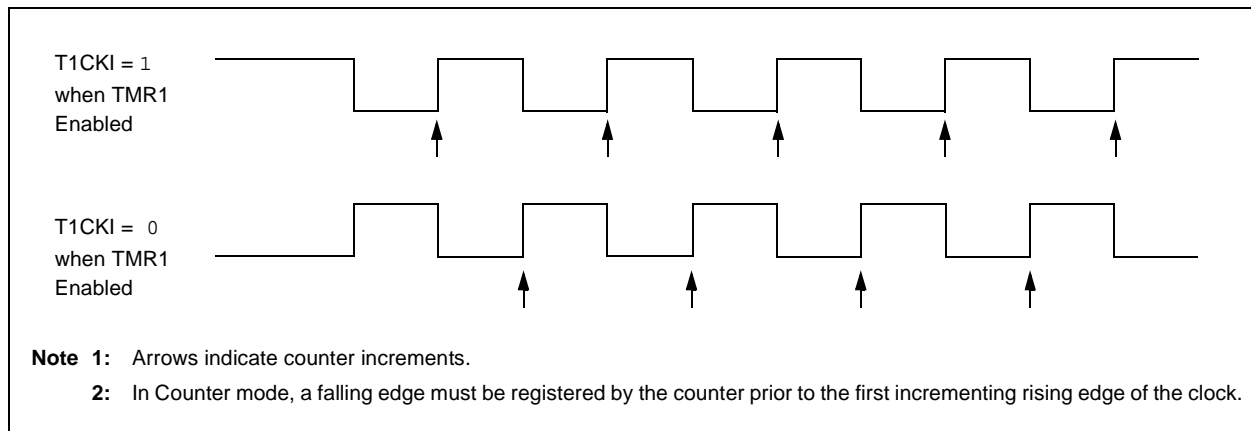
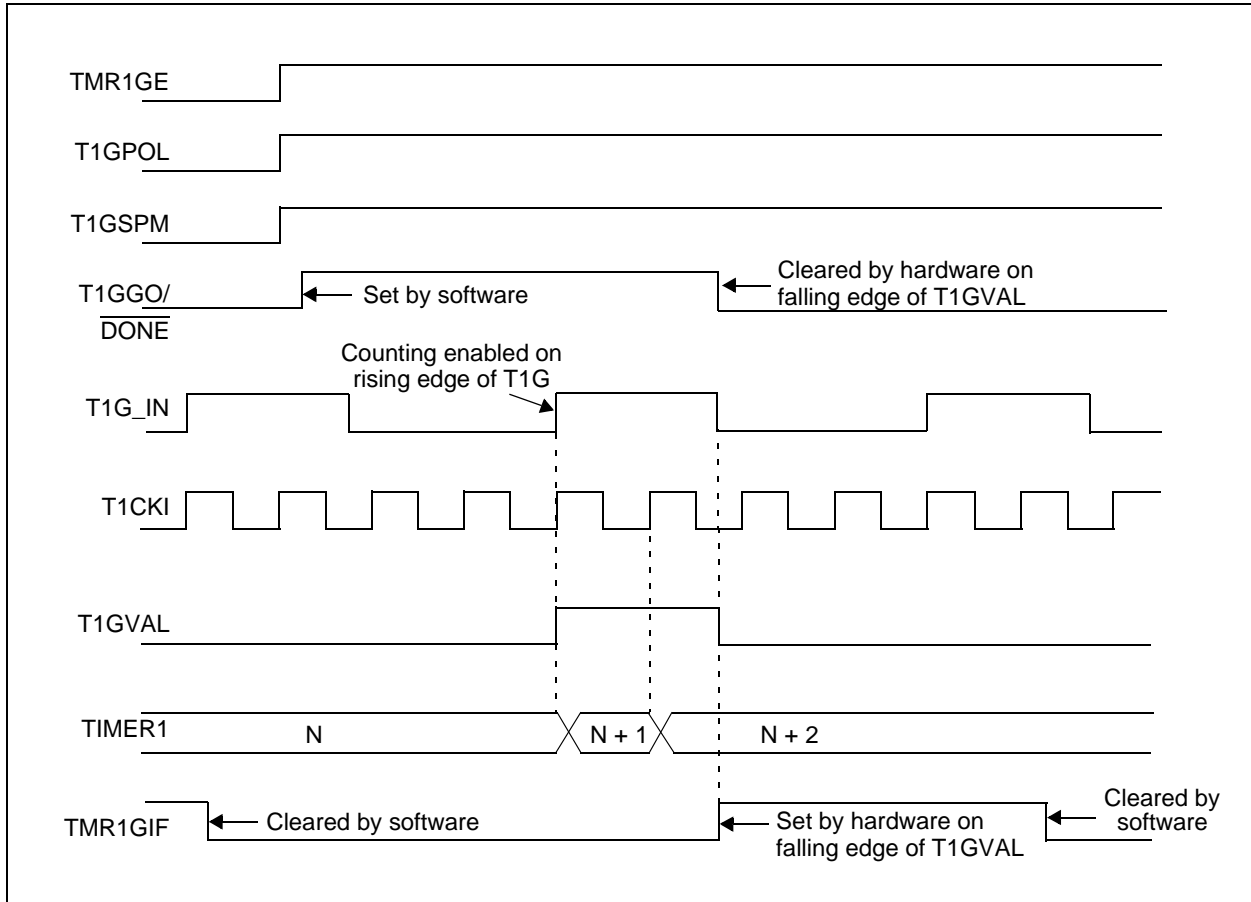


FIGURE 12-5: TIMER1 GATE SINGLE-PULSE MODE



16.1.2 AUSART ASYNCHRONOUS RECEIVER

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

16.1.2.1 Enabling the Receiver

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

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

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

Setting the CREN bit of the RCSTA register enables the receiver circuitry of the AUSART. Clearing the SYNC bit of the TXSTA register configures the AUSART for asynchronous operation. Setting the SPEN bit of the RCSTA register enables the AUSART and automatically configures the RX/DT I/O pin as an input.

Note: When the SPEN bit is set the TX/CK I/O pin is automatically configured as an output, regardless of the state of the corresponding TRIS bit and whether or not the AUSART transmitter is enabled. The PORT latch is disconnected from the output driver so it is not possible to use the TX/CK pin as a general purpose output.

16.1.2.2 Receiving Data

The receiver data recovery circuit initiates character reception on the falling edge of the first bit. The first bit, also known as the Start bit, is always a zero. The data recovery circuit counts one-half bit time to the center of the Start bit and verifies that the bit is still a zero. If it is not a zero then the data recovery circuit aborts character reception, without generating an error, and resumes looking for the falling edge of the Start bit. If the Start bit zero verification succeeds then the data recovery circuit counts a full bit time to the center of the next bit. The bit is then sampled by a majority detect circuit and the resulting '0' or '1' is shifted into the RSR. This repeats until all data bits have been sampled and shifted into the RSR. One final bit time is measured and the level sampled. This is the Stop bit, which is always a '1'. If the data recovery circuit samples a '0' in the Stop bit position then a framing error is set for this character, otherwise the framing error is cleared for this character. Refer to **Section 16.1.2.4 "Receive Framing Error"** for more information on framing errors.

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

Note: If the receive FIFO is overrun, no additional characters will be received until the overrun condition is cleared. Refer to **Section 16.1.2.5 "Receive Overrun Error"** for more information on overrun errors.

16.1.2.3 Receive Interrupts

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

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

- RCIE interrupt enable bit of the PIE1 register
- PEIE peripheral interrupt enable bit of the INTCON register
- GIE global interrupt enable bit of the INTCON register

The RCIF interrupt flag bit of the PIR1 register will be set when there is an unread character in the FIFO, regardless of the state of interrupt enable bits.

TABLE 16-5: BAUD RATES FOR ASYNCHRONOUS MODES

BAUD RATE	SYNC = 0, BRGH = 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	0.16	207
1200	—	—	—	1202	0.16	207	1200	0.00	191	1202	0.16	51
2400	2404	0.16	207	2404	0.16	103	2400	0.00	95	2404	0.16	25
9600	9615	0.16	51	9615	0.16	25	9600	0.00	23	—	—	—
10417	10417	0.00	47	10417	0.00	23	10473	0.53	21	10417	0.00	5
19.2k	19231	0.16	25	19.23k	0.16	12	19.2k	0.00	11	—	—	—
57.6k	55556	-3.55	8	—	—	—	57.60k	0.00	3	—	—	—
115.2k	—	—	—	—	—	—	115.2k	0.00	1	—	—	—

17.1.2 SLAVE MODE

For any SPI device acting as a slave, the data is transmitted and received as external clock pulses appear on SCK pin. This external clock must meet the minimum high and low times as specified in the electrical specifications.

17.1.2.1 Slave Mode Operation

The SSP consists of a transmit/receive shift register (SSPSR) and a buffer register (SSPBUF). The SSPSR shifts the data in and out of the device, MSb first. The SSPBUF holds the data that was written to the SSPSR until the received data is ready.

The slave has no control as to when data will be clocked in or out of the device. All data that is to be transmitted, to a master or another slave, must be loaded into the SSPBUF register before the first clock pulse is received.

Once eight bits of data have been received:

- Received byte is moved to the SSPBUF register
- BF bit of the SSPSTAT register is set
- SSPIF bit of the PIR1 register is set

Any write to the SSPBUF register during transmission/reception of data will be ignored and the Write Collision Detect bit, WCOL of the SSPCON register, will be set. User software must clear the WCOL bit so that it can be determined if the following write(s) to the SSPBUF register completed successfully.

The user's firmware must read SSPBUF, clearing the BF flag, or the SSPOV bit of the SSPCON register will be set with the reception of the next byte and communication will be disabled.

A SPI module transmits and receives at the same time, occasionally causing dummy data to be transmitted/received. It is up to the user to determine which data is to be used and what can be discarded.

17.1.2.2 Enabling Slave I/O

To enable the serial port, the SSPEN bit of the SSPCON register must be set. If a Slave mode of operation is selected in the SSPM bits of the SSPCON register, the SDI, SDO, SCK pins will be assigned as serial port pins.

For these pins to function as serial port pins, they must have their corresponding data direction bits set or cleared in the associated TRIS register as follows:

- SDI configured as input
- SDO configured as output
- SCK configured as input

Optionally, a fourth pin, Slave Select (\overline{SS}) may be used in Slave mode. Slave Select may be configured to operate on one of the following pins via the SSSEL bit in the APFCON register.

- RA5/AN4/ \overline{SS}
- RA0/AN0/ \overline{SS}

Upon selection of a Slave Select pin, the appropriate bits must be set in the ANSELA and TRISA registers. Slave Select must be set as an input by setting the corresponding bit in TRISA, and digital I/O must be enabled on the \overline{SS} pin by clearing the corresponding bit of the ANSELA register.

17.1.2.3 Slave Mode Setup

When initializing the SSP module to SPI Slave mode, compatibility must be ensured with the master device. This is done by programming the appropriate control bits of the SSPCON and SSPSTAT registers. These control bits allow the following to be specified:

- SCK as clock input
- Idle state of SCK (CKP bit)
- Data input sample phase (SMP bit)
- Output data on rising/falling edge of SCK (CKE bit)

Figure 17-4 and Figure 17-5 show example waveforms of Slave mode operation.

PIC16(L)F722/3/4/6/7

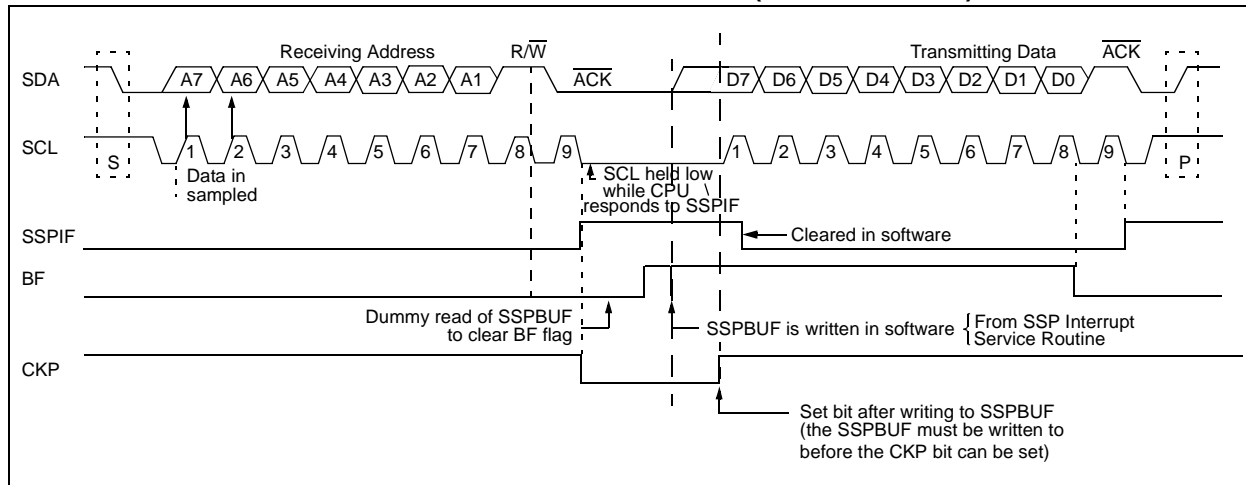
17.2.6 TRANSMISSION

When the $\overline{R/W}$ bit of the received address byte is set and an address match occurs, the $\overline{R/W}$ bit of the SSPSTAT register is set and the slave will respond to the master by reading out data. After the address match, an \overline{ACK} pulse is generated by the slave hardware and the SCL pin is held low (clock is automatically stretched) until the slave is ready to respond. See **Section 17.2.7 “Clock Stretching”**. The data the slave will transmit must be loaded into the SSPBUF register, which sets the BF bit. The SCL line is released by setting the CKP bit of the SSPCON register.

An SSP interrupt is generated for each transferred data byte. The SSPIF flag bit of the PIR1 register initiates an SSP interrupt, and must be cleared by software before the next byte is transmitted. The BF bit of the SSPSTAT register is cleared on the falling edge of the eighth received clock pulse. The SSPIF flag bit is set on the falling edge of the ninth clock pulse.

Following the eighth falling clock edge, control of the SDA line is released back to the master so that the master can acknowledge or not acknowledge the response. If the master sends a not acknowledge, the slave's transmission is complete and the slave must monitor for the next Start condition. If the master acknowledges, control of the bus is returned to the slave to transmit another byte of data. Just as with the previous byte, the clock is stretched by the slave, data must be loaded into the SSPBUF and CKP must be set to release the clock line (SCL).

FIGURE 17-12: I²C WAVEFORMS FOR TRANSMISSION (7-BIT ADDRESS)



PIC16(L)F722/3/4/6/7

17.2.7 CLOCK STRETCHING

During any SCL low phase, any device on the I²C bus may hold the SCL line low and delay, or pause, the transmission of data. This “stretching” of a transmission allows devices to slow down communication on the bus. The SCL line must be constantly sampled by the master to ensure that all devices on the bus have released SCL for more data.

Stretching usually occurs after an $\overline{\text{ACK}}$ bit of a transmission, delaying the first bit of the next byte. The SSP module hardware automatically stretches for two conditions:

- After a 10-bit address byte is received (update SSPADD register)
- Anytime the CKP bit of the SSPCON register is cleared by hardware

The module will hold SCL low until the CKP bit is set. This allows the user slave software to update SSPBUF with data that may not be readily available. In 10-bit addressing modes, the SSPADD register must be updated after receiving the first and second address bytes. The SSP module will hold the SCL line low until the SSPADD has a byte written to it. The UA bit of the SSPSTAT register will be set, along with SSPIF, indicating an address update is needed.

17.2.8 FIRMWARE MASTER MODE

Master mode of operation is supported in firmware using interrupt generation on the detection of the Start and Stop conditions. The Stop (P) and Start (S) bits of the SSPSTAT register are cleared from a Reset or when the SSP module is disabled (SSPEN cleared). The Stop (P) and Start (S) bits will toggle based on the Start and Stop conditions. Control of the I²C bus may be taken when the P bit is set or the bus is Idle and both the S and P bits are clear.

In Firmware Master mode, the SCL and SDA lines are manipulated by setting/clearing the corresponding TRIS bit(s). The output level is always low, irrespective of the value(s) in the corresponding PORT register bit(s). When transmitting a ‘1’, the TRIS bit must be set (input) and a ‘0’, the TRIS bit must be clear (output).

The following events will cause the SSP Interrupt Flag bit, SSPIF, to be set (SSP Interrupt will occur if enabled):

- Start condition
- Stop condition
- Data transfer byte transmitted/received

Firmware Master Mode of operation can be done with either the Slave mode Idle (SSPM<3:0> = 1011), or with either of the Slave modes in which interrupts are enabled. When both master and slave functionality is enabled, the software needs to differentiate the source(s) of the interrupt.

Refer to Application Note AN554, *Software Implementation of I²C™ Bus Master* (DS00554) for more information.

17.2.9 MULTI-MASTER MODE

In Multi-Master mode, the interrupt generation on the detection of the Start and Stop conditions allow the determination of when the bus is free. The Stop (P) and Start (S) bits are cleared from a Reset or when the SSP module is disabled. The Stop (P) and Start (S) bits will toggle based on the Start and Stop conditions. Control of the I²C bus may be taken when the P bit of the SSPSTAT register is set or when the bus is Idle, and both the S and P bits are clear. When the bus is busy, enabling the SSP Interrupt will generate the interrupt when the Stop condition occurs.

In Multi-Master operation, the SDA line must be monitored to see if the signal level is the expected output level. This check only needs to be done when a high level is output. If a high level is expected and a low level is present, the device needs to release the SDA and SCL lines (set TRIS bits). There are two stages where this arbitration of the bus can be lost. They are the Address Transfer and Data Transfer stages.

When the slave logic is enabled, the slave continues to receive. If arbitration was lost during the address transfer stage, communication to the device may be in progress. If addressed, an ACK pulse will be generated. If arbitration was lost during the data transfer stage, the device will need to re-transfer the data at a later time.

Refer to Application Note AN578, *Use of the SSP Module in the I²C™ Multi-Master Environment* (DS00578) for more information.

19.0 POWER-DOWN MODE (SLEEP)

The Power-down mode is entered by executing a `SLEEP` instruction.

If the Watchdog Timer is enabled:

- `WDT` will be cleared but keeps running.
- `PD` bit of the `STATUS` register is cleared.
- `TO` bit of the `STATUS` register is set.
- Oscillator driver is turned off.
- Timer1 oscillator is unaffected
- I/O ports maintain the status they had before `SLEEP` was executed (driving high, low or high-impedance).

For lowest current consumption in this mode, all I/O pins should be either at `VDD` or `VSS`, with no external circuitry drawing current from the I/O pin. I/O pins that are high-impedance inputs should be pulled high or low externally to avoid switching currents caused by floating inputs. The `T0CKI` input should also be at `VDD` or `VSS` for lowest current consumption. The contribution from on-chip pull-ups on `PORTB` should be considered.

The `MCLR` pin must be at a logic high level when external `MCLR` is enabled.

Note: A Reset generated by a `WDT` time out does not drive `MCLR` pin low.

19.1 Wake-up from Sleep

The device can wake up from Sleep through one of the following events:

1. External Reset input on `MCLR` pin.
2. Watchdog Timer wake-up (if `WDT` was enabled).
3. Interrupt from `RB0/INT` pin, `PORTB` change or a peripheral interrupt.

The first event will cause a device Reset. The two latter events are considered a continuation of program execution. The `TO` and `PD` bits in the `STATUS` register can be used to determine the cause of device Reset. The `PD` bit, which is set on power-up, is cleared when Sleep is invoked. `TO` bit is cleared if `WDT` wake-up occurred.

The following peripheral interrupts can wake the device from Sleep:

1. `TMR1` Interrupt. Timer1 must be operating as an asynchronous counter.
2. `USART` Receive Interrupt (Synchronous Slave mode only)
3. A/D conversion (when A/D clock source is `RC`)
4. Interrupt-on-change
5. External Interrupt from `INT` pin
6. Capture event on `CCP1` or `CCP2`
7. `SSP` Interrupt in `SPI` or `I2C` Slave mode

Other peripherals cannot generate interrupts since during Sleep, no on-chip clocks are present.

When the `SLEEP` instruction is being executed, the next instruction (`PC + 1`) is prefetched. For the device to wake-up through an interrupt event, the corresponding interrupt enable bit must be set (enabled). Wake-up is regardless of the state of the `GIE` bit. If the `GIE` bit is clear (disabled), the device continues execution at the instruction after the `SLEEP` instruction. If the `GIE` bit is set (enabled), the device executes the instruction after the `SLEEP` instruction, then branches to the interrupt address (0004h). In cases where the execution of the instruction following `SLEEP` is not desirable, the user should have a `NOP` after the `SLEEP` instruction.

Note: If the global interrupts are disabled (`GIE` is cleared), but any interrupt source has both its interrupt enable bit and the corresponding interrupt flag bits set, the device will immediately wake-up from Sleep. The `SLEEP` instruction is completely executed.

The `WDT` is cleared when the device wakes up from Sleep, regardless of the source of wake-up.

PIC16(L)F722/3/4/6/7

23.3 DC Characteristics: PIC16(L)F722/3/4/6/7-I/E (Power-Down) (Continued)

PIC16LF722/3/4/6/7			Standard Operating Conditions (unless otherwise stated)					
			Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended					
PIC16F722/3/4/6/7			Standard Operating Conditions (unless otherwise stated)					
			Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended					
Param No.	Device Characteristics	Min.	Typ†	Max. +85°C	Max. +125°C	Units	Conditions	
							VDD	Note
D027		Power-down Base Current (IPD) ⁽²⁾						
		—	0.06	0.7	5.0	μA	1.8	A/D Current (Note 1, Note 4), no conversion in progress
		—	0.08	1.0	5.5	μA	3.0	
D027		—	6	10.7	18	μA	1.8	A/D Current (Note 1, Note 4), no conversion in progress
		—	7	10.6	20	μA	3.0	
		—	7.2	11.9	22	μA	5.0	
D027A		—	250	400	—	μA	1.8	A/D Current (Note 1, Note 4), conversion in progress
		—	250	400	—	μA	3.0	
D027A		—	280	430	—	μA	1.8	A/D Current (Note 1, Note 4, Note 5), conversion in progress
		—	280	430	—	μA	3.0	
		—	280	430	—	μA	5.0	
D028		—	2.2	3.2	14.4	μA	1.8	Cap Sense Low Power Oscillator mode
		—	3.3	4.4	15.6	μA	3.0	
D028		—	6.5	13	21	μA	1.8	Cap Sense Low Power Oscillator mode
		—	8	14	23	μA	3.0	
		—	8	14	25	μA	5.0	
D028A		—	4.2	6	17	μA	1.8	Cap Sense Medium Power Oscillator mode
		—	6	7	18	μA	3.0	
D028A		—	8.5	15.5	23	μA	1.8	Cap Sense Medium Power Oscillator mode
		—	11	17	24	μA	3.0	
		—	11	18	27	μA	5.0	
D028B		—	12	14	25	μA	1.8	Cap Sense High Power Oscillator mode
		—	32	35	44	μA	3.0	
D028B		—	16	20	31	μA	1.8	Cap Sense High Power Oscillator mode
		—	36	41	50	μA	3.0	
		—	42	49	58	μA	5.0	

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

- Note 1:** The peripheral current is the sum of the base IDD or IPD and the additional current consumed when this peripheral is enabled. The peripheral Δ current can be determined by subtracting the base IDD or IPD current from this limit. Max values should be used when calculating total current consumption.
- 2:** The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD.
- 3:** Fixed Voltage Reference is automatically enabled whenever the BOR is enabled
- 4:** A/D oscillator source is FRC
- 5:** 0.1 μF capacitor on VCAP (RA0).

PIC16(L)F722/3/4/6/7

TABLE 23-2: OSCILLATOR PARAMETERS

Standard Operating Conditions (unless otherwise stated)								
Operating Temperature $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$								
Param No.	Sym.	Characteristic	Freq. Tolerance	Min.	Typ†	Max.	Units	Conditions
OS08	HFosc	Internal Calibrated HFINTOSC Frequency ⁽²⁾	$\pm 2\%$	—	16.0	—	MHz	$0^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$, $V_{DD} \geq 2.5\text{V}$
			$\pm 5\%$	—	16.0	—	MHz	$-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$
OS08A	MFosc	Internal Calibrated MFINTOSC Frequency ⁽²⁾	$\pm 2\%$	—	500	—	kHz	$0^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$, $V_{DD} \geq 2.5\text{V}$
			$\pm 5\%$	—	500	10	kHz	$-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$
OS10*	Tiosc st	HFINTOSC Wake-up from Sleep Start-up Time	—	—	5	8	μs	
		MFINTOSC Wake-up from Sleep Start-up Time	—	—	20	30	μs	

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

- Note 1:** Instruction cycle period (Tcy) equals four times the input oscillator time base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min" values with an external clock applied to the OSC1 pin. When an external clock input is used, the "max" cycle time limit is "DC" (no clock) for all devices.
- 2:** To ensure these oscillator frequency tolerances, VDD and VSS must be capacitively decoupled as close to the device as possible. 0.1 μF and 0.01 μF values in parallel are recommended.
- 3:** By design.

FIGURE 23-7: CLKOUT AND I/O TIMING

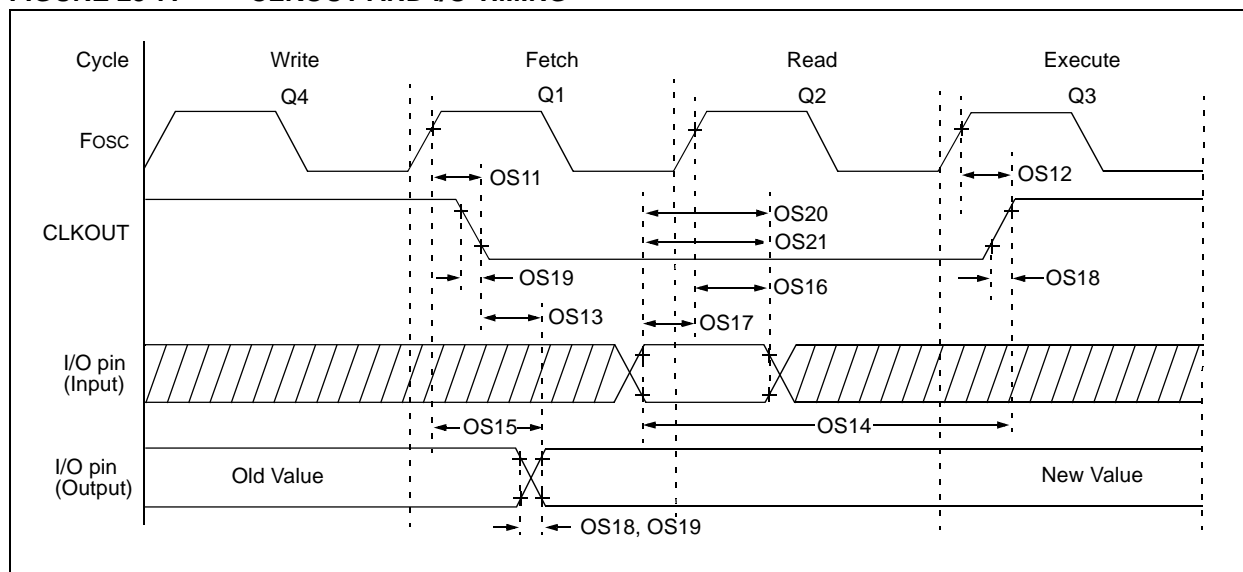


FIGURE 24-7: PIC16F722/3/4/6/7 TYPICAL I_{DD} vs. V_{DD} OVER F_{osc} , EXTRC MODE, $V_{CAP} = 0.1 \mu F$

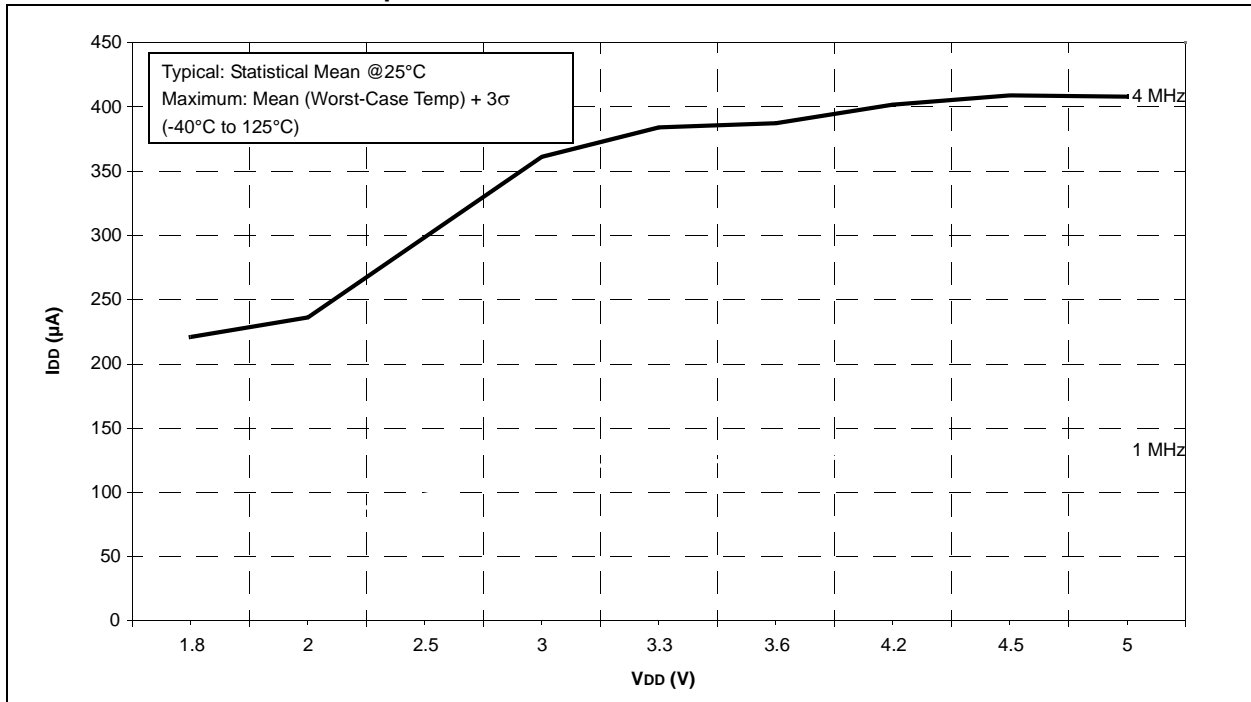


FIGURE 24-8: PIC16LF722/3/4/6/7 TYPICAL I_{DD} vs. V_{DD} OVER F_{osc} , EXTRC MODE

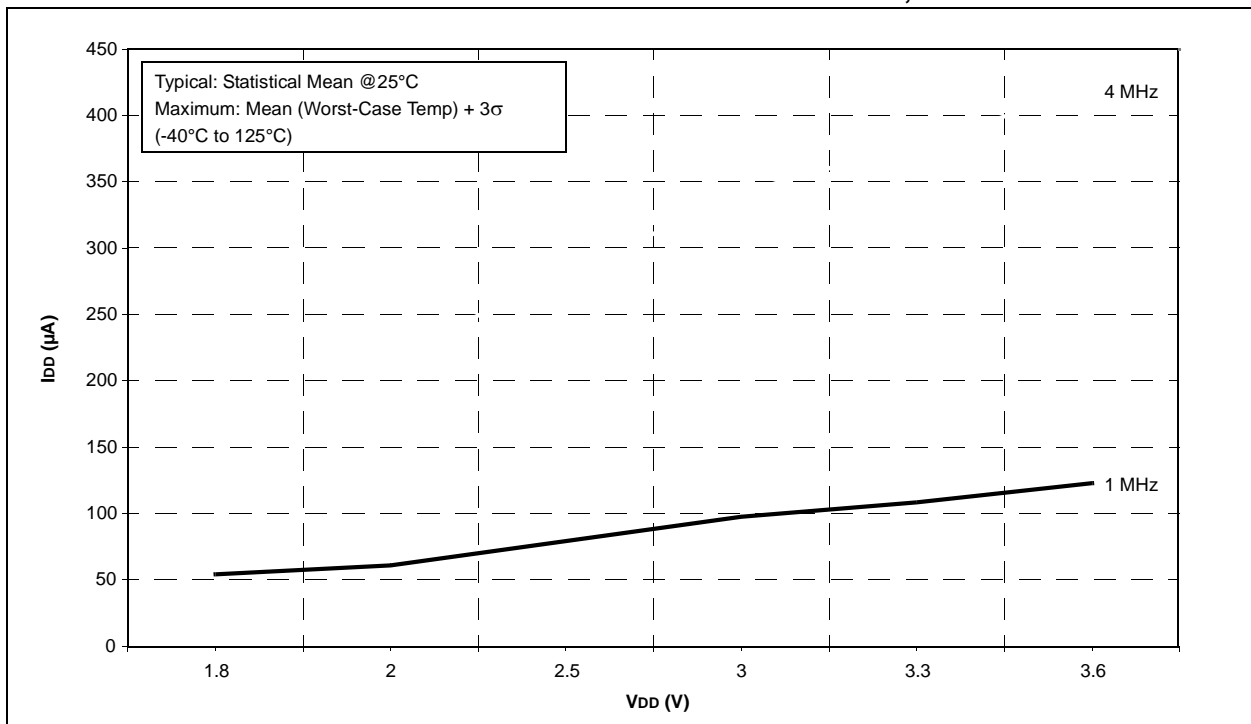


FIGURE 24-11: PIC16F722/3/4/6/7 TYPICAL I_{DD} vs. F_{OSC} OVER V_{DD} , HS MODE, $V_{CAP} = 0.1 \mu F$

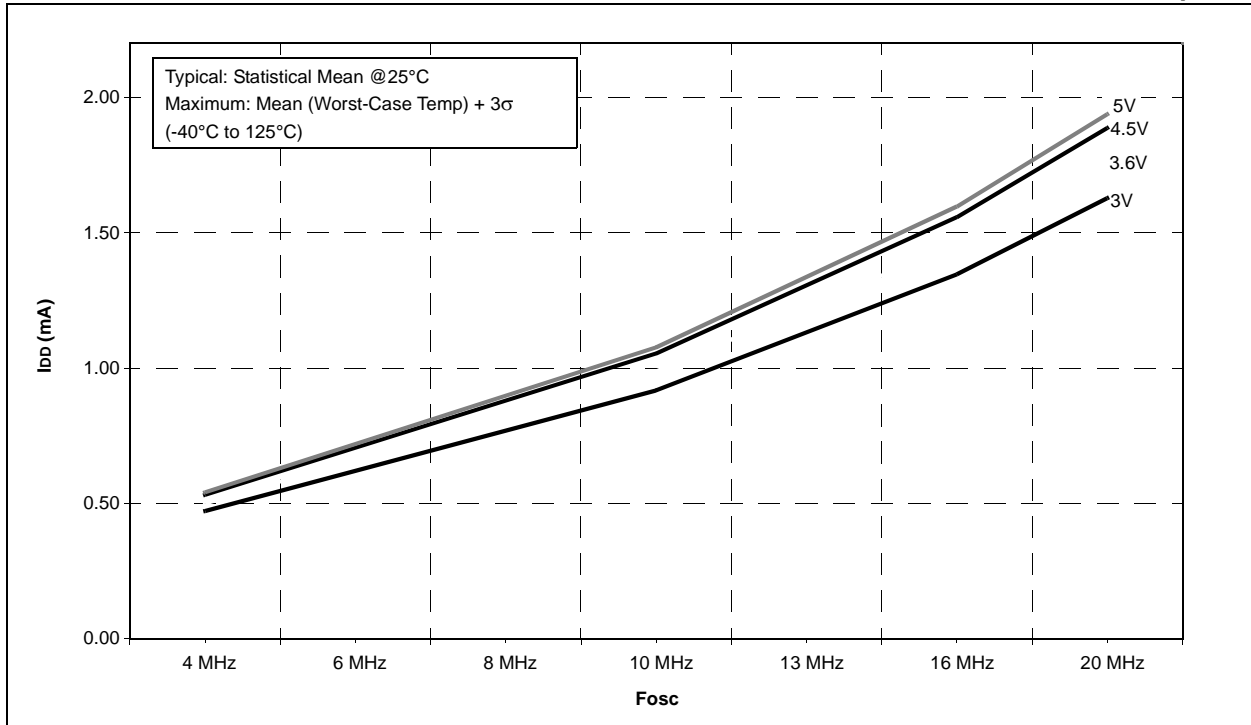
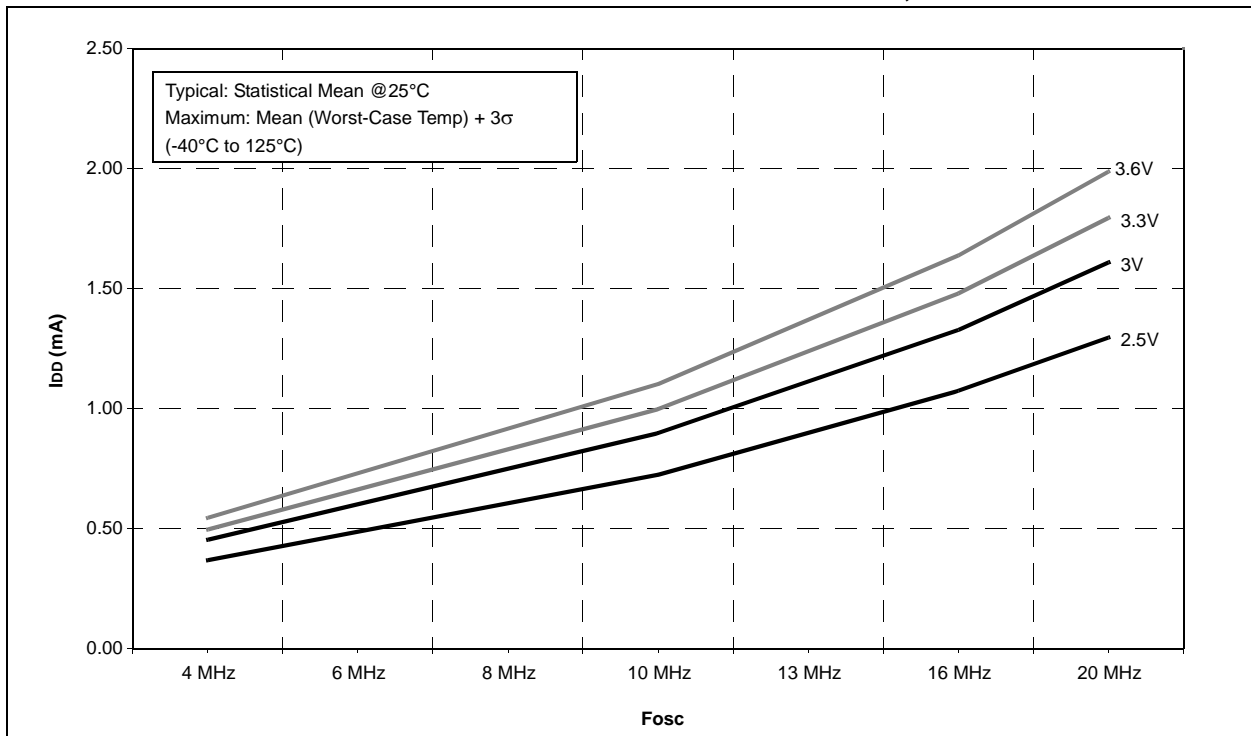


FIGURE 24-12: PIC16LF722/3/4/6/7 TYPICAL I_{DD} vs. F_{OSC} OVER V_{DD} , HS MODE



PIC16(L)F722/3/4/6/7

FIGURE 24-61: PIC16F722/3/4/6/7 A/D INTERNAL RC OSCILLATOR PERIOD

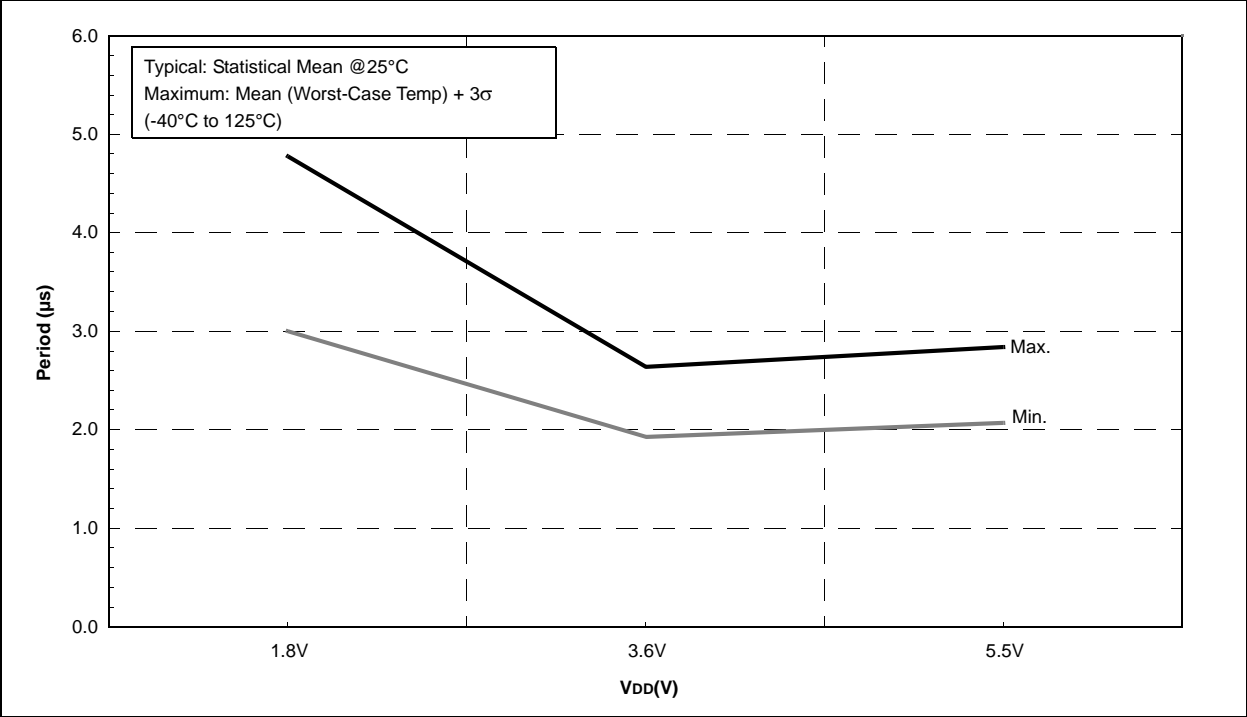


FIGURE 24-62: PIC16F722/3/4/6/7 CAP SENSE OUTPUT CURRENT, POWER MODE = HIGH

