



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	40MHz
Connectivity	CANbus, EBI/EMI, I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	68
Program Memory Size	48KB (24K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	3.25K x 8
Voltage - Supply (Vcc/Vdd)	4.2V ~ 5.5V
Data Converters	A/D 12x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	80-TQFP
Supplier Device Package	80-TQFP (12x12)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f8585-i-pt

PIC18F6585/8585/6680/8680

2.0 OSCILLATOR CONFIGURATIONS

2.1 Oscillator Types

The PIC18F6585/8585/6680/8680 devices can be operated in eleven different oscillator modes. The user can program four configuration bits (FOSC3, FOSC2, FOSC1 and FOSC0) to select one of these eleven modes:

1. LP Low-Power Crystal
2. XT Crystal/Resonator
3. HS High-Speed Crystal/Resonator
4. RC External Resistor/Capacitor
5. EC External Clock
6. ECIO External Clock with I/O pin enabled
7. HS+PLL High-Speed Crystal/Resonator with PLL enabled
8. RCIO External Resistor/Capacitor with I/O pin enabled
9. ECIO+SPLL External Clock with software controlled PLL
10. ECIO+PLL External Clock with PLL and I/O pin enabled
11. HS+SPLL High-Speed Crystal/Resonator with software control

2.2 Crystal Oscillator/Ceramic Resonators

In XT, LP, HS, HS+PLL or HS+SPLL Oscillator modes, a crystal or ceramic resonator is connected to the OSC1 and OSC2 pins to establish oscillation. Figure 2-1 shows the pin connections.

The PIC18F6585/8585/6680/8680 oscillator design requires the use of a parallel cut crystal.

Note: Use of a series cut crystal may give a frequency out of the crystal manufacturers specifications.

FIGURE 2-1: CRYSTAL/CERAMIC RESONATOR OPERATION (HS, XT OR LP CONFIGURATION)

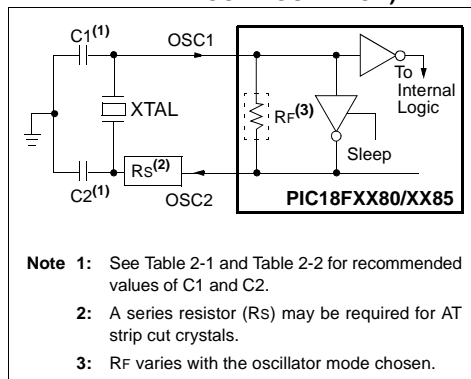


TABLE 2-1: CAPACITOR SELECTION FOR CERAMIC RESONATORS

Ranges Tested:			
Mode	Freq	C1	C2
XT	455 kHz	68-100 pF	68-100 pF
	2.0 MHz	15-68 pF	15-68 pF
	4.0 MHz	15-68 pF	15-68 pF
HS	8.0 MHz	10-68 pF	10-68 pF
	16.0 MHz	10-22 pF	10-22 pF
These values are for design guidance only. See notes following this table.			
Resonators Used:			
2.0 MHz	Murata Erie CSA2.00MG	± 0.5%	
4.0 MHz	Murata Erie CSA4.00MG	± 0.5%	
8.0 MHz	Murata Erie CSA8.00MT	± 0.5%	
16.0 MHz	Murata Erie CSA16.00MX	± 0.5%	
All resonators used did not have built-in capacitors.			

Note 1: Higher capacitance increases the stability of the oscillator, but also increases the start-up time.

2: When operating below 3V VDD, or when using certain ceramic resonators at any voltage, it may be necessary to use high gain HS mode, try a lower frequency resonator, or switch to a crystal oscillator.

3: Since each resonator/crystal has its own characteristics, the user should consult the resonator/crystal manufacturer for appropriate values of external components, or verify oscillator performance.

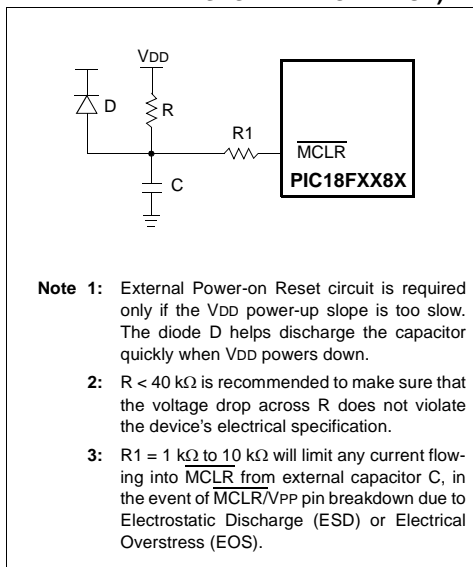
PIC18F6585/8585/6680/8680

3.1 Power-on Reset (POR)

A Power-on Reset pulse is generated on-chip when VDD rise is detected. To take advantage of the POR circuitry, tie the MCLR pin through a 1 k Ω to 10 k Ω resistor to VDD. This will eliminate external RC components usually needed to create a Power-on Reset delay. A minimum rise rate for VDD is specified (parameter D004). For a slow rise time, see Figure 3-2.

When the device starts normal operation (i.e., exits the Reset condition), device operating parameters (voltage, frequency, temperature, etc.) must be met to ensure operation. If these conditions are not met, the device must be held in Reset until the operating conditions are met.

FIGURE 3-2: EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW VDD POWER-UP)



3.2 Power-up Timer (PWRT)

The Power-up Timer provides a fixed nominal time-out (parameter #33) only on power-up from the POR. The Power-up Timer operates on an internal RC oscillator. The chip is kept in Reset as long as the PWRT is active. The PWRT's time delay allows VDD to rise to an acceptable level. A configuration bit is provided to enable/disable the PWRT.

The power-up time delay will vary from chip-to-chip due to VDD, temperature and process variation. See DC parameter #33 for details.

3.3 Oscillator Start-up Timer (OST)

The Oscillator Start-up Timer (OST) provides 1024 oscillator cycles (from OSC1 input) delay after the PWRT delay is over (parameter #32). This ensures that the crystal oscillator or resonator has started and stabilized.

The OST time-out is invoked only for XT, LP and HS modes and only on Power-on Reset, or wake-up from Sleep.

3.4 PLL Lock Time-out

With the PLL enabled, the time-out sequence following a Power-on Reset is different from other oscillator modes. A portion of the Power-up Timer is used to provide a fixed time-out that is sufficient for the PLL to lock to the main oscillator frequency. This PLL lock time-out (TPLL) is typically 2 ms and follows the oscillator start-up time-out (OST).

3.5 Brown-out Reset (BOR)

A configuration bit, BOREN, can disable (if clear/programmed), or enable (if set) the Brown-out Reset circuitry. If VDD falls below parameter D005 for greater than parameter #35, the brown-out situation will reset the chip. A Reset may not occur if VDD falls below parameter D005 for less than parameter #35. The chip will remain in Brown-out Reset until VDD rises above BVDD. If the Power-up Timer is enabled, it will be invoked after VDD rises above BVDD; it then will keep the chip in Reset for an additional time delay (parameter #33). If VDD drops below BVDD while the Power-up Timer is running, the chip will go back into a Brown-out Reset and the Power-up Timer will be initialized. Once VDD rises above BVDD, the Power-up Timer will execute the additional time delay.

3.6 Time-out Sequence

On power-up, the time-out sequence is as follows: First, PWRT time-out is invoked after the POR time delay has expired. Then, OST is activated. The total time-out will vary based on oscillator configuration and the status of the PWRT. For example, in RC mode with the PWRT disabled, there will be no time-out at all. Figure 3-3, Figure 3-4, Figure 3-5, Figure 3-6 and Figure 3-7 depict time-out sequences on power-up.

Since the time-outs occur from the POR pulse, the time-outs will expire if MCLR is kept low long enough. Bringing MCLR high will begin execution immediately (Figure 3-5). This is useful for testing purposes or to synchronize more than one PIC18FXX8X device operating in parallel.

Table 3-2 shows the Reset conditions for some Special Function Registers while Table 3-3 shows the Reset conditions for all of the registers.

PIC18F6585/8585/6680/8680

TABLE 3-3: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices		Power-on Reset, Brown-out Reset	MCLR Resets WDT Reset RESET Instruction Stack Resets	Wake-up via WDT or Interrupt
RXFCON0 ⁽⁷⁾	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
RXFBCON7 ⁽⁷⁾	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
RXFBCON6 ⁽⁷⁾	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
RXFBCON5 ⁽⁷⁾	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
RXFBCON4 ⁽⁷⁾	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
RXFBCON3 ⁽⁷⁾	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
RXFBCON2 ⁽⁷⁾	PIC18F6X8X	PIC18F8X8X	0001 0001	0001 0001	uuuu uuuu
RXFBCON1 ⁽⁷⁾	PIC18F6X8X	PIC18F8X8X	0001 0001	0001 0001	uuuu uuuu
RXFBCON0 ⁽⁷⁾	PIC18F6X8X	PIC18F8X8X	0000 0000	0000 0000	uuuu uuuu
RXF15EIDL ⁽⁷⁾	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF15EIDH ⁽⁷⁾	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF15SIDL ⁽⁷⁾	PIC18F6X8X	PIC18F8X8X	xxx- x-xx	uuu- u-uu	uuu- u-uu
RXF15SIDH ⁽⁷⁾	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF14EIDL ⁽⁷⁾	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF14EIDH ⁽⁷⁾	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF14SIDL ⁽⁷⁾	PIC18F6X8X	PIC18F8X8X	xxx- x-xx	uuu- u-uu	uuu- u-uu
RXF14SIDH ⁽⁷⁾	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF13EIDL ⁽⁷⁾	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF13EIDH ⁽⁷⁾	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF13SIDL ⁽⁷⁾	PIC18F6X8X	PIC18F8X8X	xxx- x-xx	uuu- u-uu	uuu- u-uu
RXF13SIDH ⁽⁷⁾	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF12EIDL ⁽⁷⁾	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF12EIDH ⁽⁷⁾	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF12SIDL ⁽⁷⁾	PIC18F6X8X	PIC18F8X8X	xxx- x-xx	uuu- u-uu	uuu- u-uu
RXF12SIDH ⁽⁷⁾	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF11EIDL ⁽⁷⁾	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF11EIDH ⁽⁷⁾	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF11SIDL ⁽⁷⁾	PIC18F6X8X	PIC18F8X8X	xxx- x-xx	uuu- u-uu	uuu- u-uu
RXF11SIDH ⁽⁷⁾	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXF10EIDL ⁽⁷⁾	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	-uuu uuuu
RXF10EIDH ⁽⁷⁾	PIC18F6X8X	PIC18F8X8X	xxxx xxxx	uuuu uuuu	-uuu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition.
Shaded cells indicate conditions do not apply for the designated device.

Note 1: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

- When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).
- When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.
- See Table 3-2 for Reset value for specific condition.
- Bit 6 of PORTA, LATA, and TRISA are enabled in ECIO and RCIO Oscillator modes only. In all other oscillator modes, they are disabled and read '0'.
- Bit 6 of PORTA, LATA and TRISA are not available on all devices. When unimplemented, they read '0'.
- This register reads all '0's until ECAN is set up in Mode 1 or Mode 2.

4.0 MEMORY ORGANIZATION

There are three memory blocks in PIC18F6585/8585/6680/8680 devices. They are:

- Program Memory
- Data RAM
- Data EEPROM

Data and program memory use separate busses which allows for concurrent access of these blocks. Additional detailed information for Flash program memory and data EEPROM is provided in **Section 5.0 “Flash Program Memory”** and **Section 7.0 “Data EEPROM Memory”**, respectively.

In addition to on-chip Flash, the PIC18F8X8X devices are also capable of accessing external program memory through an external memory bus. Depending on the selected operating mode (discussed in **Section 4.1.1 “PIC18F8X8X Program Memory Modes”**), the controllers may access either internal or external program memory exclusively, or both internal and external memory in selected blocks. Additional information on the external memory interface is provided in **Section 6.0 “External Memory Interface”**.

4.1 Program Memory Organization

A 21-bit program counter is capable of addressing the 2-Mbyte program memory space. Accessing a location between the physically implemented memory and the 2-Mbyte address will cause a read of all '0's (a NOP instruction).

The PIC18F6585 and PIC18F8585 each have 48 Kbytes of on-chip Flash memory, while the PIC18F6680 and PIC18F8680 have 64 Kbytes of Flash. This means that PIC18FX585 devices can store internally up to 24,576 single-word instructions and PIC18FX680 devices can store up to 32,768 single-word instructions.

The Reset vector address is at 0000h and the interrupt vector addresses are at 0008h and 0018h.

Figure 4-1 shows the program memory map for PIC18F6585/8585 devices while Figure 4-2 shows the program memory map for PIC18F6680/8680 devices.

4.1.1 PIC18F8X8X PROGRAM MEMORY MODES

PIC18F8X8X devices differ significantly from their PIC18 predecessors in their utilization of program memory. In addition to available on-chip Flash program memory, these controllers can also address up to 2 Mbytes of external program memory through the external memory interface. There are four distinct operating modes available to the controllers:

- Microprocessor (MP)
- Microprocessor with Boot Block (MPBB)
- Extended Microcontroller (EMC)
- Microcontroller (MC)

The Program Memory mode is determined by setting the two Least Significant bits of the CONFIG3L configuration byte, as shown in Register 4-1. (See also **Section 24.1 “Configuration Bits”** for additional details on the device configuration bits.)

The Program Memory modes operate as follows:

- The **Microprocessor Mode** permits access only to external program memory; the contents of the on-chip Flash memory are ignored. The 21-bit program counter permits access to a 2-MByte linear program memory space.
- The **Microprocessor with Boot Block Mode** accesses on-chip Flash memory from addresses 000000h to 0007FFh. Above this, external program memory is accessed all the way up to the 2-MByte limit. Program execution automatically switches between the two memories as required.
- The **Microcontroller Mode** accesses only on-chip Flash memory. Attempts to read above the physical limit of the on-chip Flash (0BFFFh for the PIC18F8585, 0FFFFh for the PIC18F8680) causes a read of all '0's (a NOP instruction). The Microcontroller mode is the only operating mode available to PIC18F6X8X devices.
- The **Extended Microcontroller Mode** allows access to both internal and external program memories as a single block. The device can access its entire on-chip Flash memory; above this, the device accesses external program memory up to the 2-MByte program space limit. As with Boot Block mode, execution automatically switches between the two memories as required.

In all modes, the microcontroller has complete access to data RAM and EEPROM.

Figure 4-3 compares the memory maps of the different Program Memory modes. The differences between on-chip and external memory access limitations are more fully explained in Table 4-1.

PIC18F6585/8585/6680/8680

TABLE 4-2: SPECIAL FUNCTION REGISTER MAP (CONTINUED)

Address	Name	Address	Name	Address	Name	Address	Name
F7Fh	SPBRGH	F5Fh	CANCON_RO0	F3Fh	CANCON_RO2	F1Fh	RXM1EIDL
F7Eh	BAUDCON	F5Eh	CANSTAT_RO0	F3Eh	CANSTAT_RO2	F1Eh	RXM1EIDH
F7Dh	— ⁽¹⁾	F5Dh	RXB1D7	F3Dh	TXB1D7	F1Dh	RXM1SIDL
F7Ch	— ⁽¹⁾	F5Ch	RXB1D6	F3Ch	TXB1D6	F1Ch	RXM1SIDH
F7Bh	— ⁽¹⁾	F5Bh	RXB1D5	F3Bh	TXB1D5	F1Bh	RXM0EIDL
F7Ah	— ⁽¹⁾	F5Ah	RXB1D4	F3Ah	TXB1D4	F1Ah	RXM0EIDH
F79h	ECCP1DEL	F59h	RXB1D3	F39h	TXB1D3	F19h	RXM0SIDL
F78h	— ⁽¹⁾	F58h	RXB1D2	F38h	TXB1D2	F18h	RXM0SIDH
F77h	ECANCON	F57h	RXB1D1	F37h	TXB1D1	F17h	RXF5EIDL
F76h	TXERRCNT	F56h	RXB1D0	F36h	TXB1D0	F16h	RXF5EIDH
F75h	RXERRCNT	F55h	RXB1DLC	F35h	TXB1DLC	F15h	RXF5SIDL
F74h	COMSTAT	F54h	RXB1EIDL	F34h	TXB1EIDL	F14h	RXF5SIDH
F73h	CIOCON	F53h	RXB1EIDH	F33h	TXB1EIDH	F13h	RXF4EIDL
F72h	BRGCON3	F52h	RXB1SIDL	F32h	TXB1SIDL	F12h	RXF4EIDH
F71h	BRGCON2	F51h	RXB1SIDH	F31h	TXB1SIDH	F11h	RXF4SIDL
F70h	BRGCON1	F50h	RXB1CON	F30h	TXB1CON	F10h	RXF4SIDH
F6Fh	CANCON	F4Fh	CANCON_RO1	F2Fh	CANCON_RO3	F0Fh	RXF3EIDL
F6Eh	CANSTAT	F4Eh	CANSTAT_RO1	F2Eh	CANSTAT_RO3	F0Eh	RXF3EIDH
F6Dh	RXB0D7	F4Dh	TXB0D7	F2Dh	TXB2D7	F0Dh	RXF3SIDL
F6Ch	RXB0D6	F4Ch	TXB0D6	F2Ch	TXB2D6	F0Ch	RXF3SIDH
F6Bh	RXB0D5	F4Bh	TXB0D5	F2Bh	TXB2D5	F0Bh	RXF2EIDL
F6Ah	RXB0D4	F4Ah	TXB0D4	F2Ah	TXB2D4	F0Ah	RXF2EIDH
F69h	RXB0D3	F49h	TXB0D3	F29h	TXB2D3	F09h	RXF2SIDL
F68h	RXB0D2	F48h	TXB0D2	F28h	TXB2D2	F08h	RXF2SIDH
F67h	RXB0D1	F47h	TXB0D1	F27h	TXB2D1	F07h	RXF1EIDL
F66h	RXB0D0	F46h	TXB0D0	F26h	TXB2D0	F06h	RXF1EIDH
F65h	RXB0DLC	F45h	TXB0DLC	F25h	TXB2DLC	F05h	RXF1SIDL
F64h	RXB0EIDL	F44h	TXB0EIDL	F24h	TXB2EIDL	F04h	RXF1SIDH
F63h	RXB0EIDH	F43h	TXB0EIDH	F23h	TXB2EIDH	F03h	RXF0EIDL
F62h	RXB0SIDL	F42h	TXB0SIDL	F22h	TXB2SIDL	F02h	RXF0EIDH
F61h	RXB0SIDH	F41h	TXB0SIDH	F21h	TXB2SIDH	F01h	RXF0SIDL
F60h	RXB0CON	F40h	TXB0CON	F20h	TXB2CON	F00h	RXF0SIDH

- Note 1:** Unimplemented registers are read as '0'.
- 2:** This register is not available on PIC18F6X8X devices.
- 3:** This is not a physical register.

6.0 EXTERNAL MEMORY INTERFACE

Note: The external memory interface is not implemented on PIC18F6X8X (64/68-pin) devices.

The external memory interface is a feature of the PIC18F8X8X devices that allows the controller to access external memory devices (such as Flash, EPROM, SRAM, etc.) as program memory.

The physical implementation of the interface uses 27 pins. These pins are reserved for external address/data bus functions; they are multiplexed with I/O port pins on four ports. Three I/O ports are multiplexed with the address/data bus, while the fourth port is multiplexed with the bus control signals. The I/O port functions are enabled when the EBDIS bit in the MEMCON register is set (see Register 6-1). A list of the multiplexed pins and their functions is provided in Table 6-1.

As implemented in the PIC18F8X8X devices, the interface operates in a similar manner to the external memory interface introduced on PIC18C601/801 microcontrollers. The most notable difference is that the interface on PIC18F8X8X devices only operates in 16-bit modes. The 8-bit mode is not supported.

For a more complete discussion of the operating modes that use the external memory interface, refer to **Section 4.1.1 “PIC18F8X8X Program Memory Modes”**.

6.1 Program Memory Modes and the External Memory Interface

As previously noted, PIC18F8X8X controllers are capable of operating in any one of four program memory modes using combinations of on-chip and external program memory. The functions of the multiplexed port pins depend on the program memory mode selected as well as the setting of the EBDIS bit.

In **Microprocessor Mode**, the external bus is always active and the port pins have only the external bus function.

In **Microcontroller Mode**, the bus is not active and the pins have their port functions only. Writes to the MEMCOM register are not permitted.

In **Microprocessor with Boot Block** or **Extended Microcontroller Mode**, the external program memory bus shares I/O port functions on the pins. When the device is fetching or doing table read/table write operations on the external program memory space, the pins will have the external bus function. If the device is fetching and accessing internal program memory locations only, the EBDIS control bit will change the pins from external memory to I/O port functions. When EBDIS = 0, the pins function as the external bus. When EBDIS = 1, the pins function as I/O ports.

PIC18F6585/8585/6680/8680

If the device fetches or accesses external memory while $EBDIS = 1$, the pins will switch to external bus. If the $EBDIS$ bit is set by a program executing from external memory, the action of setting the bit will be delayed until the program branches into the internal memory. At that time, the pins will change from external bus to I/O ports.

When the device is executing out of internal memory (with $EBDIS = 0$) in Microprocessor with Boot Block mode or Extended Microcontroller mode, the control signals will be inactive. They will go to a state where the $AD<15:0>$, $A<19:16>$ are tri-state; the \overline{OE} , \overline{WRH} , \overline{WRL} , \overline{UB} and \overline{LB} signals are '1'; and ALE and $BA0$ are '0'.

TABLE 6-1: PIC18F8X8X EXTERNAL BUS – I/O PORT FUNCTIONS

Name	Port	Bit	Function
RD0/AD0	PORTD	bit 0	Input/Output or System Bus Address bit 0 or Data bit 0
RD1/AD1	PORTD	bit 1	Input/Output or System Bus Address bit 1 or Data bit 1
RD2/AD2	PORTD	bit 2	Input/Output or System Bus Address bit 2 or Data bit 2
RD3/AD3	PORTD	bit 3	Input/Output or System Bus Address bit 3 or Data bit 3
RD4/AD4	PORTD	bit 4	Input/Output or System Bus Address bit 4 or Data bit 4
RD5/AD5	PORTD	bit 5	Input/Output or System Bus Address bit 5 or Data bit 5
RD6/AD6	PORTD	bit 6	Input/Output or System Bus Address bit 6 or Data bit 6
RD7/AD7	PORTD	bit 7	Input/Output or System Bus Address bit 7 or Data bit 7
RE0/AD8	PORTE	bit 0	Input/Output or System Bus Address bit 8 or Data bit 8
RE1/AD9	PORTE	bit 1	Input/Output or System Bus Address bit 9 or Data bit 9
RE2/AD10	PORTE	bit 2	Input/Output or System Bus Address bit 10 or Data bit 10
RE3/AD11	PORTE	bit 3	Input/Output or System Bus Address bit 11 or Data bit 11
RE4/AD12	PORTE	bit 4	Input/Output or System Bus Address bit 12 or Data bit 12
RE5/AD13	PORTE	bit 5	Input/Output or System Bus Address bit 13 or Data bit 13
RE6/AD14	PORTE	bit 6	Input/Output or System Bus Address bit 14 or Data bit 14
RE7/AD15	PORTE	bit 7	Input/Output or System Bus Address bit 15 or Data bit 15
RH0/A16	PORTH	bit 0	Input/Output or System Bus Address bit 16
RH1/A17	PORTH	bit 1	Input/Output or System Bus Address bit 17
RH2/A18	PORTH	bit 2	Input/Output or System Bus Address bit 18
RH3/A19	PORTH	bit 3	Input/Output or System Bus Address bit 19
RJ0/ALE	PORTJ	bit 0	Input/Output or System Bus Address Latch Enable (ALE) Control pin
RJ1/ \overline{OE}	PORTJ	bit 1	Input/Output or System Bus Output Enable (\overline{OE}) Control pin
RJ2/ \overline{WRL}	PORTJ	bit 2	Input/Output or System Bus Write Low (\overline{WRL}) Control pin
RJ3/ \overline{WRH}	PORTJ	bit 3	Input/Output or System Bus Write High (\overline{WRH}) Control pin
RJ4/BA0	PORTJ	bit 4	Input/Output or System Bus Byte Address bit 0
RJ5/ \overline{CE}	PORTJ	bit 5	Input/Output or Chip Enable
RJ6/ \overline{LB}	PORTJ	bit 6	Input/Output or System Bus Lower Byte Enable (\overline{LB}) Control pin
RJ7/ \overline{UB}	PORTJ	bit 7	Input/Output or System Bus Upper Byte Enable (\overline{UB}) Control pin

PIC18F6585/8585/6680/8680

NOTES:

PIC18F6585/8585/6680/8680

REGISTER 9-9: PIE3: PERIPHERAL INTERRUPT ENABLE REGISTER 3

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
IRXIE	WAKIE	ERRIE	TXB2IE/ TXBnIE	TXB1IE ⁽¹⁾	TXB0IE ⁽¹⁾	RXB1IE/ RXBnIE	RXB0IE/ FIFOWMIE

bit 7

bit 0

- bit 7 **IRXIE:** CAN Invalid Received Message Interrupt Enable bit
1 = Enable invalid message received interrupt
0 = Disable invalid message received interrupt
- bit 6 **WAKIE:** CAN bus Activity Wake-up Interrupt Enable bit
1 = Enable bus activity wake-up interrupt
0 = Disable bus activity wake-up interrupt
- bit 5 **ERRIE:** CAN bus Error Interrupt Enable bit
1 = Enable CAN bus error interrupt
0 = Disable CAN bus error interrupt
- bit 4 When CAN is in Mode 0:
TXB2IE: CAN Transmit Buffer 2 Interrupt Enable bit
1 = Enable Transmit Buffer 2 interrupt
0 = Disable Transmit Buffer 2 interrupt
When CAN is in Mode 1 or 2:
TXBnIE: CAN Transmit Buffer Interrupts Enable bit
1 = Enable transmit buffer interrupt; individual interrupt is enabled by TXBIE and BIE0
0 = Disable all transmit buffer interrupts
- bit 3 **TXB1IE:** CAN Transmit Buffer 1 Interrupt Enable bit⁽¹⁾
1 = Enable Transmit Buffer 1 interrupt
0 = Disable Transmit Buffer 1 interrupt
- bit 2 **TXB0IE:** CAN Transmit Buffer 0 Interrupt Enable bit⁽¹⁾
1 = Enable Transmit Buffer 0 interrupt
0 = Disable Transmit Buffer 0 interrupt
- bit 1 When CAN is in Mode 0:
RXB1IE: CAN Receive Buffer 1 Interrupt Enable bit
1 = Enable Receive Buffer 1 interrupt
0 = Disable Receive Buffer 1 interrupt
When CAN is in Mode 1 or 2:
RXBnIE: CAN Receive Buffer Interrupts Enable bit
1 = Enable receive buffer interrupt; individual interrupt is enabled by BIE0
0 = Disable all receive buffer interrupts
- bit 0 When CAN is in Mode 0:
RXB0IE: CAN Receive Buffer 0 Interrupt Enable bit
1 = Enable Receive Buffer 0 interrupt
0 = Disable Receive Buffer 0 interrupt
When CAN is in Mode 1:
Unimplemented: Read as '0'
When CAN is in Mode 2:
FIFOWMIE: FIFO Watermark Interrupt Enable bit
1 = Enable FIFO watermark interrupt
0 = Disable FIFO watermark interrupt

Note 1: In CAN Mode 1 and 2, this bit is forced to '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

PIC18F6585/8585/6680/8680

REGISTER 9-11: IPR2: PERIPHERAL INTERRUPT PRIORITY REGISTER 2

U-0	R/W-1	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	CMIP	—	EEIP	BCLIP	LVDIP	TMR3IP	CCP2IP

bit 7

bit 0

bit 7 **Unimplemented:** Read as '0'

bit 6 **CMIP:** Comparator Interrupt Priority bit

1 = High priority

0 = Low priority

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

bit 4 **EEIP:** Data EEPROM/Flash Write Operation Interrupt Priority bit

1 = High priority

0 = Low priority

bit 3 **BCLIP:** Bus Collision Interrupt Priority bit

1 = High priority

0 = Low priority

bit 2 **LVDIP:** Low-Voltage Detect Interrupt Priority bit

1 = High priority

0 = Low priority

bit 1 **TMR3IP:** TMR3 Overflow Interrupt Priority bit

1 = High priority

0 = Low priority

bit 0 **CCP2IP:** CCP2 Interrupt Priority bit

1 = High priority

0 = Low priority

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

14.1 Timer3 Operation

Timer3 can operate in one of these modes:

- As a timer
- As a synchronous counter
- As an asynchronous counter

The operating mode is determined by the clock select bit, TMR3CS (T3CON<1>).

When TMR3CS = 0, Timer3 increments every instruction cycle. When TMR3CS = 1, Timer3 increments on every rising edge of the Timer1 external clock input or the Timer1 oscillator if enabled.

When the Timer1 oscillator is enabled (T1OSCEN is set), the RC1/T1OSI and RC0/T1OSO/T13CKI pins become inputs. That is, the TRISC<1:0> value is ignored and the pins are read as '0'.

Timer3 also has an internal "Reset input". This Reset can be generated by the CCP module (**Section 14.0 "Timer3 Module"**).

FIGURE 14-1: TIMER3 BLOCK DIAGRAM

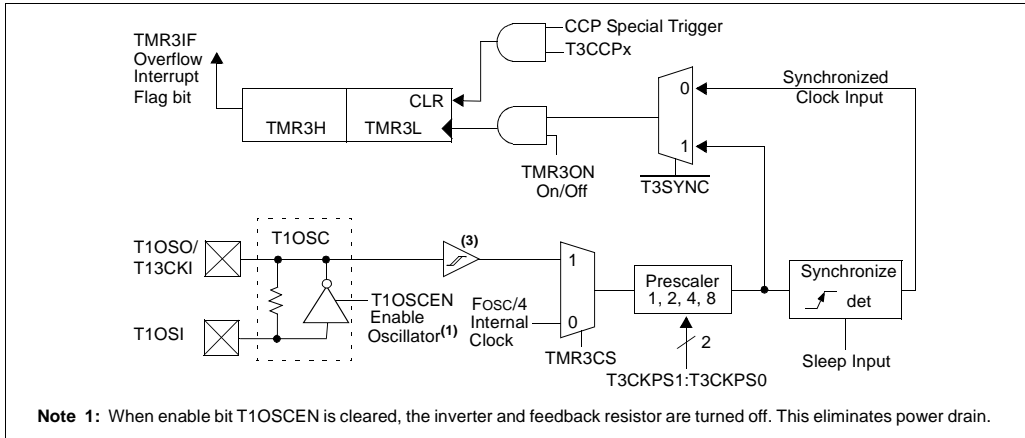
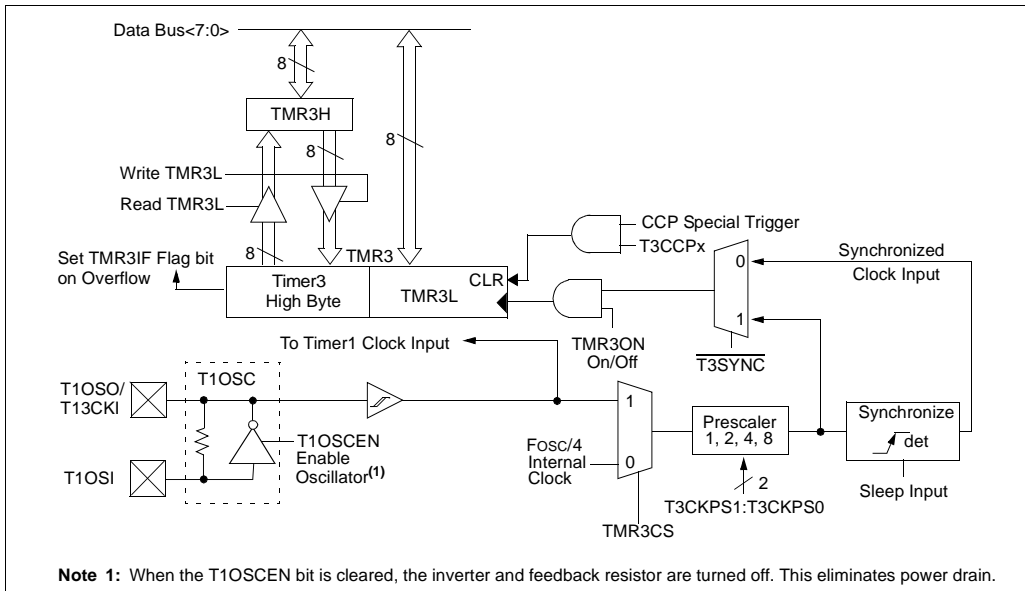


FIGURE 14-2: TIMER3 BLOCK DIAGRAM CONFIGURED IN 16-BIT READ/WRITE MODE



PIC18F6585/8585/6680/8680

FIGURE 17-5: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 0)

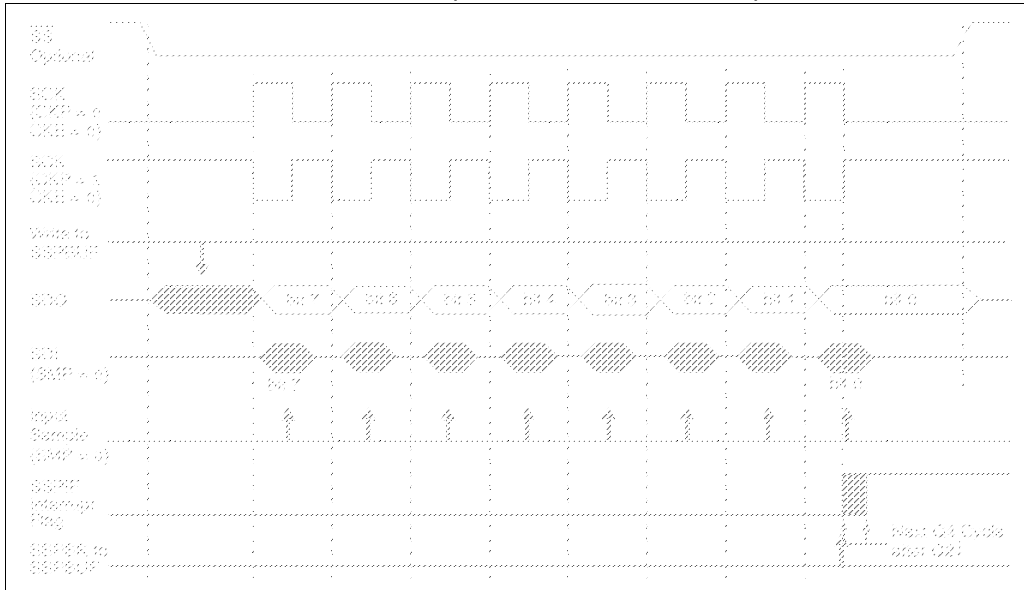
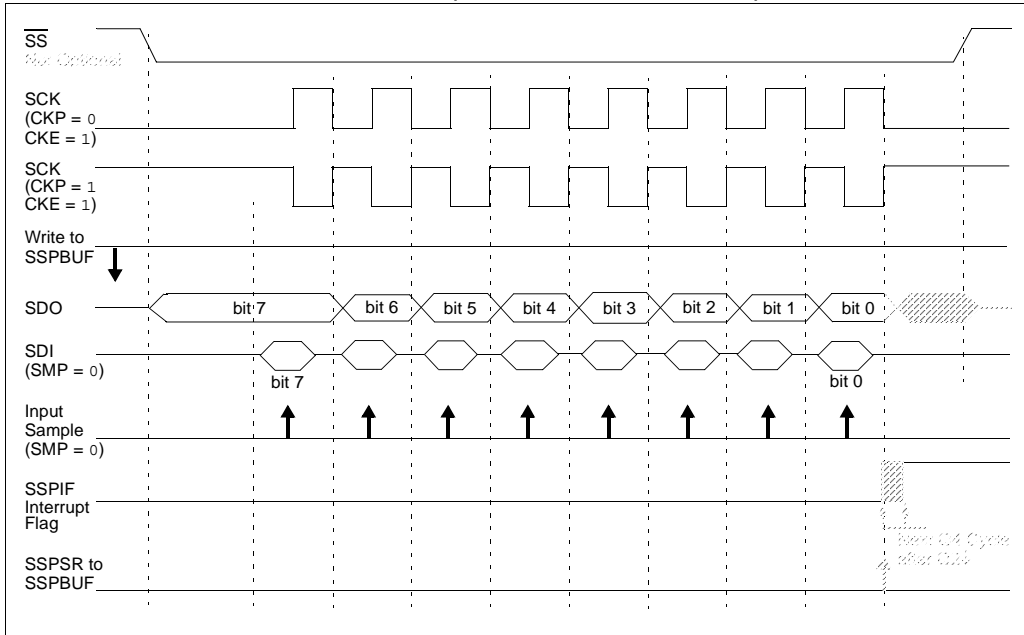


FIGURE 17-6: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 1)



PIC18F6585/8585/6680/8680

EXAMPLE 23-2: WIN AND ICODE BITS USAGE IN INTERRUPT SERVICE ROUTINE TO ACCESS TX/RX BUFFERS (CONTINUED)

```
ErrorInterrupt
    BCF    PIR3, ERRIF          ; Clear the interrupt flag
    ...                          ; Handle error.
    RETFIE

TXB2Interrupt
    BCF    PIR3, TXB2IF        ; Clear the interrupt flag
    GOTO   AccessBuffer

TXB1Interrupt
    BCF    PIR3, TXB1IF        ; Clear the interrupt flag
    GOTO   AccessBuffer

TXB0Interrupt
    BCF    PIR3, TXB0IF        ; Clear the interrupt flag
    GOTO   AccessBuffer

RXB1Interrupt
    BCF    PIR3, RXB1IF        ; Clear the interrupt flag
    GOTO   AccessBuffer

RXB0Interrupt
    BCF    PIR3, RXB0IF        ; Clear the interrupt flag
    GOTO   AccessBuffer

AccessBuffer
    ; This is either TX or RX interrupt
    ; Copy CANSTAT.ICODE bits to CANCON.WIN bits
    MOVF   TempCANCON, W       ; Clear CANCON.WIN bits before copying
                                ; new ones.
    ANDLW  B'11110001'         ; Use previously saved CANCON value to
                                ; make sure same value.
    MOVWF  TempCANCON          ; Copy masked value back to TempCANCON
    MOVF   TempCANSTAT, W      ; Retrieve ICODE bits
    ANDLW  B'00001110'         ; Use previously saved CANSTAT value
                                ; to make sure same value.
    IORWF  TempCANCON          ; Copy ICODE bits to WIN bits.
    MOVFF  TempCANCON, CANCON  ; Copy the result to actual CANCON
    ; Access current buffer...
    ; User code
    ; Restore CANCON.WIN bits
    MOVF   CANCON, W           ; Preserve current non WIN bits
    ANDLW  B'11110001'         ; Restore original WIN bits
    IORWF  TempCANCON          ; Do not need to restore CANSTAT - it is read-only register.
    ; Return from interrupt or check for another module interrupt source
```

EXAMPLE 23-3: TRANSMITTING A CAN MESSAGE USING BANKED METHOD

```
; Need to transmit Standard Identifier message 123h using TXB0 buffer.
; To successfully transmit, CAN module must be either in Normal or Loopback mode.
; TXB0 buffer is not in access bank. And since we want banked method, we need to make sure
; that correct bank is selected.
BANKSEL TXB0CON                ; One BANKSEL in beginning will make sure that we are
                                ; in correct bank for rest of the buffer access.

; Now load transmit data into TXB0 buffer.
MOVLW MY_DATA_BYTE1            ; Load first data byte into buffer
MOVWF TXB0D0                   ; Compiler will automatically set "BANKED" bit
; Load rest of data bytes - up to 8 bytes into TXB0 buffer.
...
; Load message identifier
MOVLW 60H                      ; Load SID2:SID0, EXIDE = 0
MOVWF TXB0SIDL
MOVLW 24H                      ; Load SID10:SID3
MOVWF TXB0SIDH
; No need to load TXB0EIDL:TXB0EIDH, as we are transmitting Standard Identifier Message only.

; Now that all data bytes are loaded, mark it for transmission.
MOVLW B'00001000'             ; Normal priority; Request transmission
MOVWF TXB0CON

; If required, wait for message to get transmitted
BTFSC TXB0CON, TXREQ           ; Is it transmitted?
BRA $-2                       ; No. Continue to wait...

; Message is transmitted.
```

PIC18F6585/8585/6680/8680

REGISTER 24-7: CONFIG5L: CONFIGURATION REGISTER 5 LOW (BYTE ADDRESS 300008h)

U-0	U-0	U-0	U-0	R/C-1	R/C-1	R/C-1	R/C-1
—	—	—	—	CP3 ⁽¹⁾	CP2	CP1	CP0
bit 7				bit 0			

bit 7-4 **Unimplemented:** Read as '0'

bit 3 **CP3:** Code Protection bit⁽¹⁾

1 = Block 3 (00C000-00FFFFh) not code-protected

0 = Block 3 (00C000-00FFFFh) code-protected

Note 1: Unimplemented in PIC18FX585 devices; maintain this bit set.

bit 2 **CP2:** Code Protection bit

1 = Block 2 (008000-00BFFFh) not code-protected

0 = Block 2 (008000-00BFFFh) code-protected

bit 1 **CP1:** Code Protection bit

1 = Block 1 (004000-007FFFh) not code-protected

0 = Block 1 (004000-007FFFh) code-protected

bit 0 **CP0:** Code Protection bit

1 = Block 0 (000800-003FFFh) not code-protected

0 = Block 0 (000800-003FFFh) code-protected

Legend:

R = Readable bit

C = Clearable bit

U = Unimplemented bit, read as '0'

- n = Value when device is unprogrammed

u = Unchanged from programmed state

REGISTER 24-8: CONFIG5H: CONFIGURATION REGISTER 5 HIGH (BYTE ADDRESS 300009h)

R/C-1	R/C-1	U-0	U-0	U-0	U-0	U-0	U-0
CPD	CPB	—	—	—	—	—	—
bit 7				bit 0			

bit 7 **CPD:** Data EEPROM Code Protection bit

1 = Data EEPROM not code-protected

0 = Data EEPROM code-protected

bit 6 **CPB:** Boot Block Code Protection bit

1 = Boot block (000000-0007FFh) not code-protected

0 = Boot block (000000-0007FFh) code-protected

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

Legend:

R = Readable bit

C = Clearable bit

U = Unimplemented bit, read as '0'

- n = Value when device is unprogrammed

u = Unchanged from programmed state

PIC18F6585/8585/6680/8680

TABLE 25-2: PIC18FXXX INSTRUCTION SET (CONTINUED)

Mnemonic, Operands	Description	Cycles	16-Bit Instruction Word				Status Affected	Notes
			MSb		LSb			
LITERAL OPERATIONS								
ADDLW k	Add literal and WREG	1	0000	1111	kkkk	kkkk	C, DC, Z, OV, N	
ANDLW k	AND literal with WREG	1	0000	1011	kkkk	kkkk	Z, N	
IORLW k	Inclusive OR literal with WREG	1	0000	1001	kkkk	kkkk	Z, N	
LFSR f, k	Move literal (12-bit) 2nd word to FSRx 1st word	2	1110	1110	00ff	kkkk	None	
			1111	0000	kkkk	kkkk		
MOVLB k	Move literal to BSR<3:0>	1	0000	0001	0000	kkkk	None	
MOVLW k	Move literal to WREG	1	0000	1110	kkkk	kkkk	None	
MULLW k	Multiply literal with WREG	1	0000	1101	kkkk	kkkk	None	
RETLW k	Return with literal in WREG	2	0000	1100	kkkk	kkkk	None	
SUBLW k	Subtract WREG from literal	1	0000	1000	kkkk	kkkk	C, DC, Z, OV, N	
XORLW k	Exclusive OR literal with WREG	1	0000	1010	kkkk	kkkk	Z, N	
DATA MEMORY ↔ PROGRAM MEMORY OPERATIONS								
TBLRD*	Table Read	2	0000	0000	0000	1000	None	
TBLRD*+	Table Read with post-increment		0000	0000	0000	1001	None	
TBLRD*-	Table Read with post-decrement		0000	0000	0000	1010	None	
TBLRD+*	Table Read with pre-increment		0000	0000	0000	1011	None	
TBLWT*	Table Write	2 (5)	0000	0000	0000	1100	None	
TBLWT*+	Table Write with post-increment		0000	0000	0000	1101	None	
TBLWT*-	Table Write with post-decrement		0000	0000	0000	1110	None	
TBLWT+*	Table Write with pre-increment		0000	0000	0000	1111	None	

Note 1: When a Port register is modified as a function of itself (e.g., `MOVF PORTB, 1, 0`), 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.
- 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`.
- 4: Some instructions are two-word instructions. The second word of these instructions will be executed as a `NOP` unless the first word of the instruction retrieves the information embedded in these 16 bits. This ensures that all program memory locations have a valid instruction.
- 5: If the table write starts the write cycle to internal memory, the write will continue until terminated.

PIC18F6585/8585/6680/8680

SUBLW Subtract W from literal

Syntax: `[label] SUBLW k`

Operands: $0 \leq k \leq 255$

Operation: $k - (W) \rightarrow W$

Status Affected: N, OV, C, DC, Z

Encoding:

0000	1000	kkkk	kkkk
------	------	------	------

Description: W is subtracted from the eight-bit literal 'k'. The result is placed in W.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process Data	Write to W

Example 1: `SUBLW 0x02`

Before Instruction

W = 1
C = ?

After Instruction

W = 1
C = 1 ; result is positive
Z = 0
N = 0

Example 2: `SUBLW 0x02`

Before Instruction

W = 2
C = ?

After Instruction

W = 0
C = 1 ; result is zero
Z = 1
N = 0

Example 3: `SUBLW 0x02`

Before Instruction

W = 3
C = ?

After Instruction

W = FF ; (2's complement)
C = 0 ; result is negative
Z = 0
N = 1

SUBWF Subtract W from f

Syntax: `[label] SUBWF f[,d[,a]]`

Operands: $0 \leq f \leq 255$

$d \in [0,1]$

$a \in [0,1]$

Operation: $(f) - (W) \rightarrow \text{dest}$

Status Affected: N, OV, C, DC, Z

Encoding:

0101	11da	ffff	ffff
------	------	------	------

Description: Subtract W 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' (default). If 'a' is '0', the Access Bank will be selected, overriding the BSR value. If 'a' is '1', then the bank will be selected as per the BSR value (default).

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination

Example 1: `SUBWF REG, 1, 0`

Before Instruction

REG = 3
W = 2
C = ?

After Instruction

REG = 1
W = 2
C = 1 ; result is positive
Z = 0
N = 0

Example 2: `SUBWF REG, 0, 0`

Before Instruction

REG = 2
W = 2
C = ?

After Instruction

REG = 2
W = 0
C = 1 ; result is zero
Z = 1
N = 0

Example 3: `SUBWF REG, 1, 0`

Before Instruction

REG = 1
W = 2
C = ?

After Instruction

REG = FFh ; (2's complement)
W = 2
C = 0 ; result is negative
Z = 0
N = 1

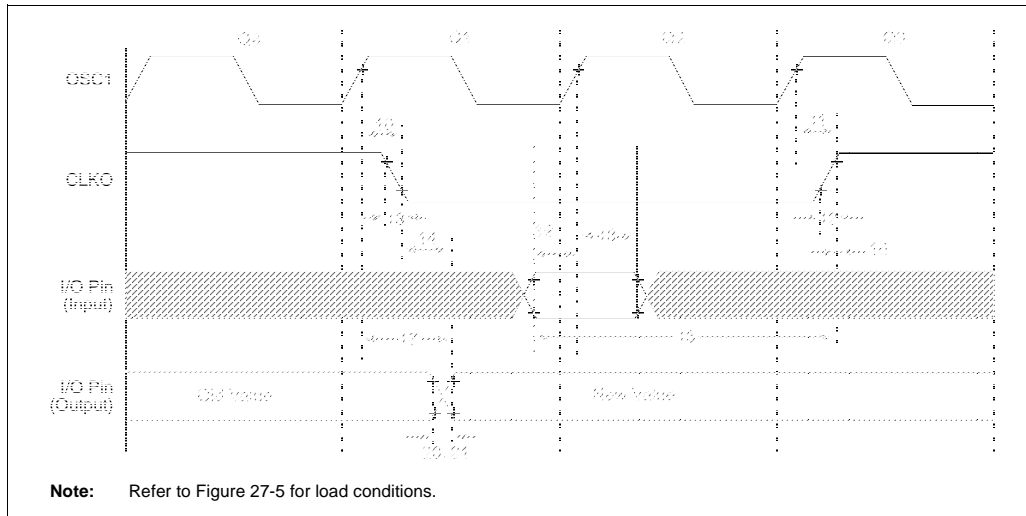
PIC18F6585/8585/6680/8680

TABLE 27-7: PLL CLOCK TIMING SPECIFICATIONS (V_{DD} = 4.2 TO 5.5V)

Param. No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
—	FOSC	Oscillator Frequency Range	4	—	10	MHz	HS mode
—	FSYS	On-Chip VCO System Frequency	16	—	40	MHz	HS mode
—	t _{rc}	PLL Start-up Time (Lock Time)	—	—	2	ms	
—	ΔCLK	CLKO Stability (Jitter)	-2	—	+2	%	

† Data in "Typ" column is at 5V, 25°C, unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 27-7: CLKO AND I/O TIMING



PIC18F6585/8585/6680/8680

TABLE 27-22: MASTER SSP I²C BUS DATA REQUIREMENTS

Param. No.	Symbol	Characteristic		Min	Max	Units	Conditions
100	THIGH	Clock High Time	100 kHz mode	2(Tosc)(BRG + 1)	—	ms	
			400 kHz mode	2(Tosc)(BRG + 1)	—	ms	
			1 MHz mode ⁽¹⁾	2(Tosc)(BRG + 1)	—	ms	
101	TLOW	Clock Low Time	100 kHz mode	2(Tosc)(BRG + 1)	—	ms	
			400 kHz mode	2(Tosc)(BRG + 1)	—	ms	
			1 MHz mode ⁽¹⁾	2(Tosc)(BRG + 1)	—	ms	
102	Tr	SDA and SCL Rise Time	100 kHz mode	—	1000	ns	Cb is specified to be from 10 to 400 pF
			400 kHz mode	20 + 0.1 Cb	300	ns	
			1 MHz mode ⁽¹⁾	—	300	ns	
103	Tf	SDA and SCL Fall Time	100 kHz mode	—	300	ns	Cb is specified to be from 10 to 400 pF
			400 kHz mode	20 + 0.1 Cb	300	ns	
			1 MHz mode ⁽¹⁾	—	100	ns	
90	TSU:STA	Start Condition Setup Time	100 kHz mode	2(Tosc)(BRG + 1)	—	ms	Only relevant for Repeated Start condition
			400 kHz mode	2(Tosc)(BRG + 1)	—	ms	
			1 MHz mode ⁽¹⁾	2(Tosc)(BRG + 1)	—	ms	
91	THD:STA	Start Condition Hold Time	100 kHz mode	2(Tosc)(BRG + 1)	—	ms	After this period, the first clock pulse is generated
			400 kHz mode	2(Tosc)(BRG + 1)	—	ms	
			1 MHz mode ⁽¹⁾	2(Tosc)(BRG + 1)	—	ms	
106	THD:DAT	Data Input Hold Time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	ms	
			1 MHz mode ⁽¹⁾	TBD	—	ns	
107	TSU:DAT	Data Input Setup Time	100 kHz mode	250	—	ns	(Note 2)
			400 kHz mode	100	—	ns	
			1 MHz mode ⁽¹⁾	TBD	—	ns	
92	TSU:STO	Stop Condition Setup Time	100 kHz mode	2(Tosc)(BRG + 1)	—	ms	
			400 kHz mode	2(Tosc)(BRG + 1)	—	ms	
			1 MHz mode ⁽¹⁾	2(Tosc)(BRG + 1)	—	ms	
109	TAA	Output Valid from Clock	100 kHz mode	—	3500	ns	
			400 kHz mode	—	1000	ns	
			1 MHz mode ⁽¹⁾	—	—	ns	
110	TBUF	Bus Free Time	100 kHz mode	4.7	—	ms	Time the bus must be free before a new transmission can start
			400 kHz mode	1.3	—	ms	
			1 MHz mode ⁽¹⁾	TBD	—	ms	
D102	Cb	Bus Capacitive Loading		—	400	pF	

Note 1: Maximum pin capacitance = 10 pF for all I²C pins.

- 2:** A Fast mode I²C bus device can be used in a Standard mode I²C bus system, but parameter #107 ≥ 250 ns, must then be met. This will automatically be the case if the device does not stretch the low period of the SCL signal. If such a device does stretch the low period of the SCL signal, it must output the next data bit to the SDA line, parameter #102 + parameter #107 = 1000 + 250 = 1250 ns (for 100 kHz mode), before the SCL line is released.

PIC18F6585/8585/6680/8680

FIGURE 28-11: TYPICAL AND MAXIMUM I_{T1OSC} vs. V_{DD} (TIMER1 AS SYSTEM CLOCK)

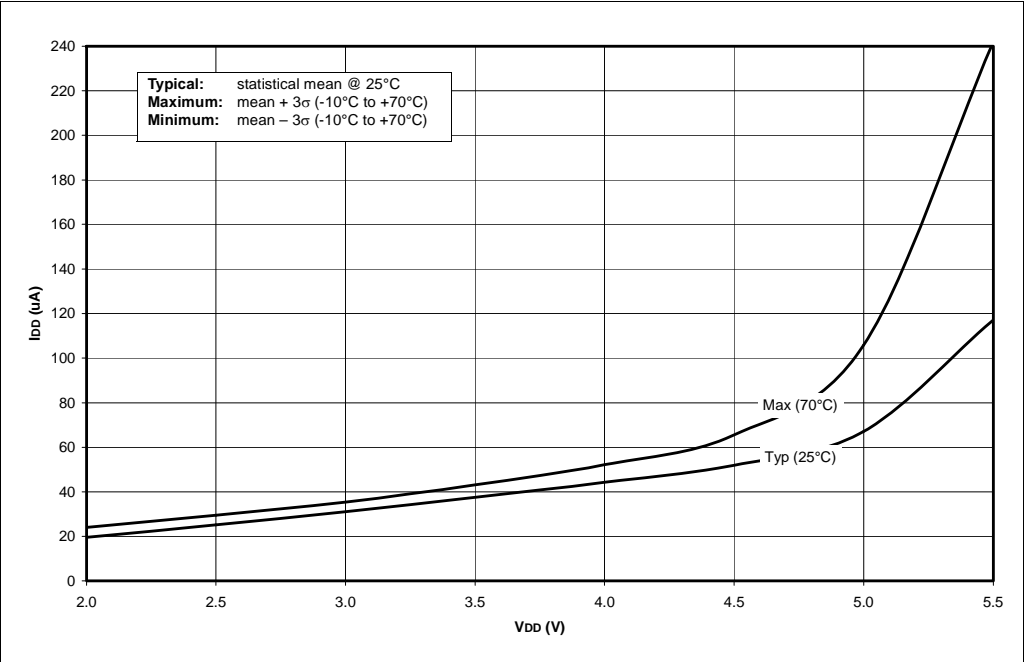


FIGURE 28-12: AVERAGE F_{OSC} vs. V_{DD} FOR VARIOUS R's (RC MODE, C = 20 pF, TEMP = 25°C)

