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	41.667MHz
Connectivity	Ethernet, I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	55
Program Memory Size	96KB (48K x 16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	3808 x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 3.6V
Data Converters	A/D 15x10b
Oscillator Type	Internal
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/pic18f86j65-i-pt

NOTES:

TABLE 1-4: PIC18F66J60/66J65/67J60 PINOUT I/O DESCRIPTIONS

Pin Name	Pin Number	Pin Type	Buffer Type	Description
	TQFP			
$\overline{MCLR}$	7	I	ST	Master Clear (Reset) input. This pin is an active-low Reset to the device.
OSC1/CLKI	39			Oscillator crystal or external clock input.
OSC1		I	ST	Oscillator crystal input or external clock source input. ST buffer when configured in internal RC mode; CMOS otherwise.
CLKI		I	CMOS	External clock source input. Always associated with pin function, OSC1. (See related OSC2/CLKO pin.)
OSC2/CLKO	40			Oscillator crystal or clock output.
OSC2		O	—	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode.
CLKO		O	—	In Internal RC mode, OSC2 pin outputs CLKO which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
				PORTA is a bidirectional I/O port.
RA0/LEDA/AN0	24			
RA0		I/O	TTL	Digital I/O.
LEDA		O	—	Ethernet LEDA indicator output.
AN0		I	Analog	Analog Input 0.
RA1/LEDB/AN1	23			
RA1		I/O	TTL	Digital I/O.
LEDB		O	—	Ethernet LEDB indicator output.
AN1		I	Analog	Analog Input 1.
RA2/AN2/V_{REF-}	22			
RA2		I/O	TTL	Digital I/O.
AN2		I	Analog	Analog Input 2.
V_{REF-}		I	Analog	A/D reference voltage (low) input.
RA3/AN3/V_{REF+}	21			
RA3		I/O	TTL	Digital I/O.
AN3		I	Analog	Analog Input 3.
V_{REF+}		I	Analog	A/D reference voltage (high) input.
RA4/T0CKI	28			
RA4		I/O	ST	Digital I/O.
T0CKI		I	ST	Timer0 external clock input.
RA5/AN4	27			
RA5		I/O	TTL	Digital I/O.
AN4		I	Analog	Analog Input 4.

Legend: TTL = TTL compatible input
ST = Schmitt Trigger input with CMOS levels
I = Input
P = Power
CMOS = CMOS compatible input or output
Analog = Analog input
O = Output
OD = Open-Drain (no P diode to V_{DD})

TABLE 1-4: PIC18F66J60/66J65/67J60 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number	Pin Type	Buffer Type	Description
	TQFP			
				PORTD is a bidirectional I/O port.
RD0/P1B	60			
RD0		I/O	ST	Digital I/O.
P1B		O	—	ECCP1 PWM Output B.
RD1/ECCP3/P3A	59			
RD1		I/O	ST	Digital I/O.
ECCP3		I/O	ST	Capture 3 input/Compare 3 output/PWM3 output.
P3A		O	—	ECCP3 PWM Output A.
RD2/CCP4/P3D	58			
RD2		I/O	ST	Digital I/O.
CCP4		I/O	ST	Capture 4 input/Compare 4 output/PWM4 output.
P3D		O	—	CCP4 PWM Output D.

Legend: TTL = TTL compatible input; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; Analog = Analog input; I = Input; O = Output; P = Power; OD = Open-Drain (no P diode to V_{DD})

TABLE 1-5: PIC18F86J60/86J65/87J60 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number	Pin Type	Buffer Type	Description
	TQFP			
				PORTC is a bidirectional I/O port.
RC0/T1OSO/T13CKI	36			
RC0		I/O	ST	Digital I/O.
T1OSO		O	—	Timer1 oscillator output.
T13CKI		I	ST	Timer1/Timer3 external clock input.
RC1/T1OSI/ECCP2/P2A	35			
RC1		I/O	ST	Digital I/O.
T1OSI		I	CMOS	Timer1 oscillator input.
ECCP2(1)		I/O	ST	Capture 2 input/Compare 2 output/PWM2 output.
P2A(1)		O	—	ECCP2 PWM Output A.
RC2/ECCP1/P1A	43			
RC2		I/O	ST	Digital I/O.
ECCP1		I/O	ST	Capture 1 input/Compare 1 output/PWM1 output.
P1A		O	—	ECCP1 PWM Output A.
RC3/SCK1/SCL1	44			
RC3		I/O	ST	Digital I/O.
SCK1		I/O	ST	Synchronous serial clock input/output for SPI mode.
SCL1		I/O	ST	Synchronous serial clock input/output for I^2C™ mode.
RC4/SDI1/SDA1	45			
RC4		I/O	ST	Digital I/O.
SDI1		I	ST	SPI data in.
SDA1		I/O	ST	I^2C data I/O.
RC5/SDO1	46			
RC5		I/O	ST	Digital I/O.
SDO1		O	—	SPI data out.
RC6/TX1/CK1	37			
RC6		I/O	ST	Digital I/O.
TX1		O	—	EUSART1 asynchronous transmit.
CK1		I/O	ST	EUSART1 synchronous clock (see related RX1/DT1 pin).
RC7/RX1/DT1	38			
RC7		I/O	ST	Digital I/O.
RX1		I	ST	EUSART1 asynchronous receive.
DT1		I/O	ST	EUSART1 synchronous data (see related TX1/CK1 pin).

Legend: TTL = TTL compatible input; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; Analog = Analog input; I = Input; O = Output; P = Power; OD = Open-Drain (no P diode to V_{DD})

Note 1: Default assignment for ECCP2/P2A when CCP2MX Configuration bit is set.

2: Default assignments for P1B/P1C/P3B/P3C (ECCPMX Configuration bit is set).

3: Alternate assignment for ECCP2/P2A when CCP2MX Configuration bit is cleared.

4: Alternate assignments for P1B/P1C/P3B/P3C (ECCPMX Configuration bit is cleared).

2.0 GUIDELINES FOR GETTING STARTED WITH PIC18FJ MICROCONTROLLERS

2.1 Basic Connection Requirements

Getting started with the PIC18F97J60 family family of 8-bit microcontrollers requires attention to a minimal set of device pin connections before proceeding with development.

The following pins must always be connected:

- All VDD and VSS pins
  (see **Section 2.2 "Power Supply Pins"**)
- All AVDD and AVSS pins, regardless of whether or not the analog device features are used
  (see **Section 2.2 "Power Supply Pins"**)
- $\overline{\text{MCLR}}$ pin
  (see **Section 2.3 "Master Clear (MCLR) Pin"**)
- ENVREG (if implemented) and VCAP/VDDCORE pins
  (see **Section 2.4 "Voltage Regulator Pins (ENVREG and VCAP/VDDCORE)"**)

These pins must also be connected if they are being used in the end application:

- PGC/PGD pins used for In-Circuit Serial Programming™ (ICSP™) and debugging purposes
  (see **Section 2.5 "ICSP Pins"**)
- OSCI and OSCO pins when an external oscillator source is used
  (see **Section 2.6 "External Oscillator Pins"**)

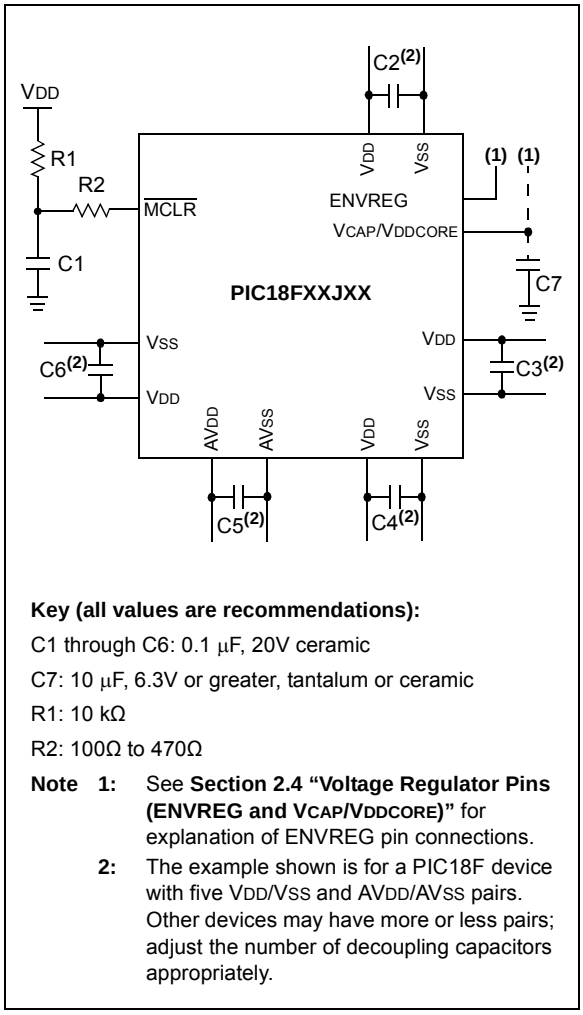
Additionally, the following pins may be required:

- VREF+/VREF- pins are used when external voltage reference for analog modules is implemented

Note:	The AVDD and AVSS pins must always be connected, regardless of whether any of the analog modules are being used.

The minimum mandatory connections are shown in Figure 2-1.

FIGURE 2-1: RECOMMENDED MINIMUM CONNECTIONS



Key (all values are recommendations):

C1 through C6: 0.1 μF, 20V ceramic

C7: 10 μF, 6.3V or greater, tantalum or ceramic

R1: 10 kΩ

R2: 100Ω to 470Ω

Note 1: See **Section 2.4 "Voltage Regulator Pins (ENVREG and VCAP/VDDCORE)"** for explanation of ENVREG pin connections.

2: The example shown is for a PIC18F device with five VDD/VSS and AVDD/AVSS pairs. Other devices may have more or less pairs; adjust the number of decoupling capacitors appropriately.

4.2.3 RC_RUN MODE

In RC_RUN mode, the CPU and peripherals are clocked from the internal oscillator; the primary clock is shut down. This mode provides the best power conservation of all the Run modes while still executing code. It works well for user applications which are not highly timing-sensitive or do not require high-speed clocks at all times.

This mode is entered by setting SCS<1:0> to '11'. When the clock source is switched to the INTRC (see Figure 4-3), the primary oscillator is shut down and the OSTS bit is cleared.

On transitions from RC_RUN mode to PRI_RUN mode, the device continues to be clocked from the INTRC while the primary clock is started. When the primary clock becomes ready, a clock switch to the primary clock occurs (see Figure 4-4). When the clock switch is complete, the OSTS bit is set and the primary clock is providing the device clock. The IDLEN and SCS bits are not affected by the switch. The INTRC source will continue to run if either the WDT or Fail-Safe Clock Monitor is enabled.

FIGURE 4-3: TRANSITION TIMING TO RC_RUN MODE

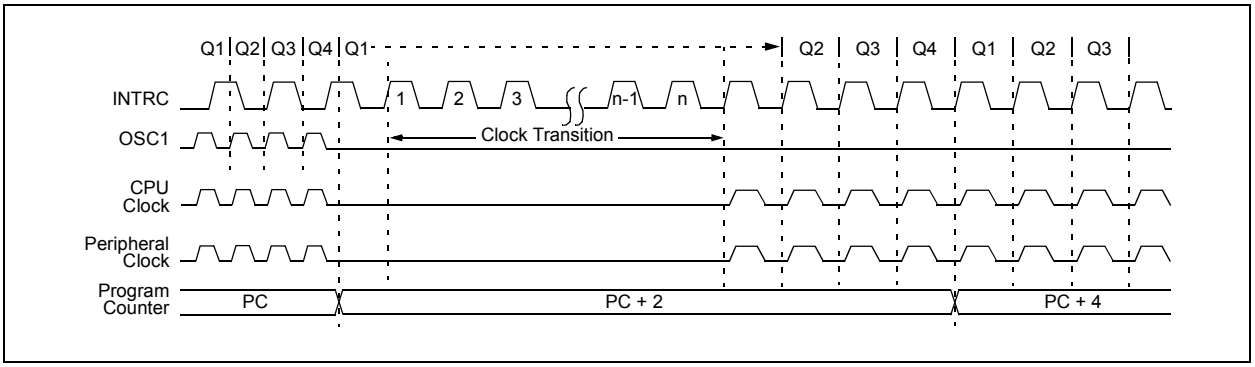
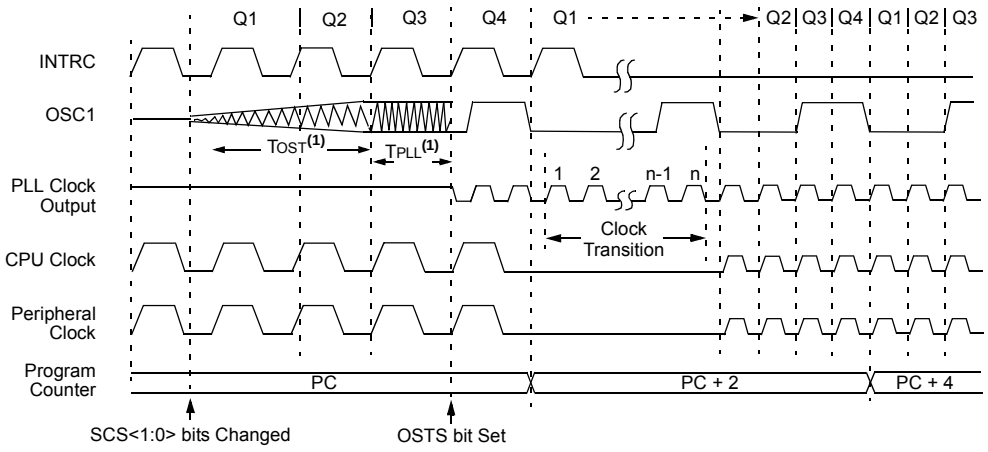


FIGURE 4-4: TRANSITION TIMING FROM RC_RUN MODE TO PRI_RUN MODE



Note 1: TOST = 1024 TOSC; TPLL = 2 ms (approx). These intervals are not shown to scale.

4.4.1 PRI_IDLE MODE

This mode is unique among the three low-power Idle modes in that it does not disable the primary device clock. For timing-sensitive applications, this allows for the fastest resumption of device operation with its more accurate primary clock source, since the clock source does not have to “warm up” or transition from another oscillator.

PRI_IDLE mode is entered from PRI_RUN mode by setting the IDLEN bit and executing a SLEEP instruction. If the device is in another Run mode, set IDLEN first, then set the SCS<1:0> bits to ‘10’ and execute SLEEP. Although the CPU is disabled, the peripherals continue to be clocked from the primary clock source specified by the FOSC<1:0> Configuration bits. The OSTS bit remains set (see Figure 4-7).

When a wake event occurs, the CPU is clocked from the primary clock source. A delay of interval, TCSD, is required between the wake event and when code execution starts. This is required to allow the CPU to become ready to execute instructions. After the wake-up, the OSTS bit remains set. The IDLEN and SCS bits are not affected by the wake-up (see Figure 4-8).

4.4.2 SEC_IDLE MODE

In SEC_IDLE mode, the CPU is disabled but the peripherals continue to be clocked from the Timer1 oscillator. This mode is entered from SEC_RUN by setting the IDLEN bit and executing a SLEEP instruction. If the device is in another Run mode, set IDLEN first, then set SCS<1:0> to ‘01’ and execute SLEEP. When the clock source is switched to the Timer1 oscillator, the primary oscillator is shut down, the OSTS bit is cleared and the T1RUN bit is set.

When a wake event occurs, the peripherals continue to be clocked from the Timer1 oscillator. After an interval of TCSD, following the wake event, the CPU begins executing code being clocked by the Timer1 oscillator. The IDLEN and SCS bits are not affected by the wake-up; the Timer1 oscillator continues to run (see Figure 4-8).

Note:	The Timer1 oscillator should already be running prior to entering SEC_IDLE mode. If the T1OSCEN bit is not set when the SLEEP instruction is executed, the SLEEP instruction will be ignored and entry to SEC_IDLE mode will not occur. If the Timer1 oscillator is enabled, but not yet running, peripheral clocks will be delayed until the oscillator has started. In such situations, initial oscillator operation is far from stable and unpredictable operation may result.

FIGURE 4-7: TRANSITION TIMING FOR ENTRY TO IDLE MODE

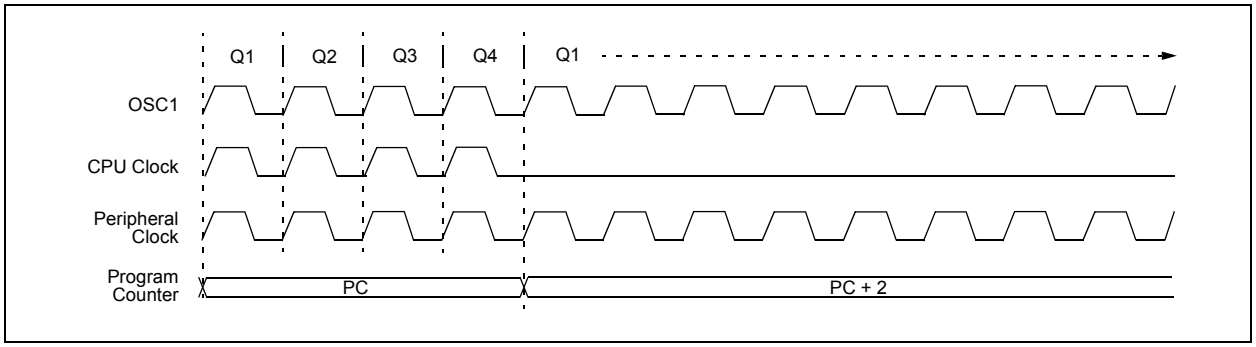
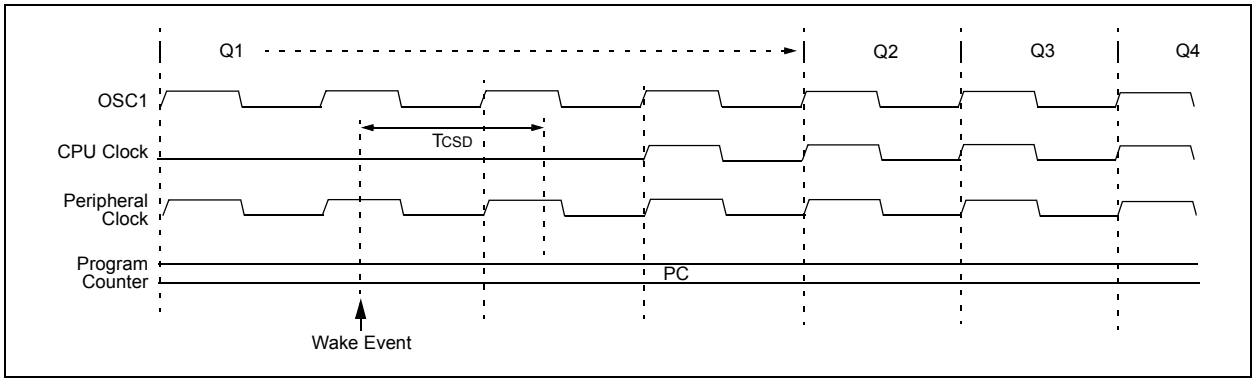


FIGURE 4-8: TRANSITION TIMING FOR WAKE FROM IDLE TO RUN MODE



REGISTER 5-1: RCON: RESET CONTROL REGISTER

R/W-0	U-0	R/W-1	R/W-1	R-1	R-1	R/W-0	R/W-0
IPEN	—	$\overline{CM}$	$\overline{RI}$	$\overline{TO}$	$\overline{PD}$	$\overline{POR}$	$\overline{BOR}$
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 **IPEN:** Interrupt Priority Enable bit
1 = Enable priority levels on interrupts
0 = Disable priority levels on interrupts (PIC16CXXX Compatibility mode)

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

bit 5 **$\overline{CM}$:** Configuration Mismatch Flag bit
1 = A Configuration Mismatch Reset has not occurred
0 = A Configuration Mismatch Reset has occurred (must be set in software after a Configuration Mismatch Reset occurs)

bit 4 **$\overline{RI}$:** RESET Instruction Flag bit
1 = The RESET instruction was not executed (set by firmware only)
0 = The RESET instruction was executed causing a device Reset (must be set in software after a Brown-out Reset occurs)

bit 3 **$\overline{TO}$:** Watchdog Timer Time-out Flag bit
1 = Set by power-up, CLRWDT instruction or SLEEP instruction
0 = A WDT time-out occurred

bit 2 **$\overline{PD}$:** Power-Down Detection Flag bit
1 = Set by power-up or by the CLRWDT instruction
0 = Set by execution of the SLEEP instruction

bit 1 **$\overline{POR}$:** Power-on Reset Status bit
1 = A Power-on Reset has not occurred (set by firmware only)
0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

bit 0 **$\overline{BOR}$:** Brown-out Reset Status bit
1 = A Brown-out Reset has not occurred (set by firmware only)
0 = A Brown-out Reset occurred (must be set in software after a Brown-out Reset occurs)

Note 1: It is recommended that the $\overline{POR}$ bit be set after a Power-on Reset has been detected so that subsequent Power-on Resets may be detected.

2: If the on-chip voltage regulator is disabled, $\overline{BOR}$ remains '0' at all times. See **Section 5.4.1 "Detecting BOR"** for more information.

3: Brown-out Reset is said to have occurred when $\overline{BOR}$ is '0' and $\overline{POR}$ is '1' (assuming that $\overline{POR}$ was set to '1' by software immediately after a Power-on Reset).

A CM Reset behaves similarly to a Master Clear Reset, RESET instruction, WDT time-out or Stack Event Reset. As with all hard and power Reset events, the device Configuration Words are reloaded from the Flash Configuration Words in program memory as the device restarts.

5.6 Power-up Timer (PWRT)

PIC18F97J60 family of devices incorporates an on-chip Power-up Timer (PWRT) to help regulate the Power-on Reset process. The PWRT is always enabled. The main function is to ensure that the device voltage is stable before code is executed.

The Power-up Timer (PWRT) of the PIC18F97J60 family devices is an 11-bit counter which uses the INTRC source as the clock input. This yields an approximate time interval of 2048 x 32 μs = 66 ms. While the PWRT is counting, the device is held in Reset.

The power-up time delay depends on the INTRC clock and will vary from chip-to-chip due to temperature and process variation. See DC Parameter 33 for details.

5.6.1 TIME-OUT SEQUENCE

The PWRT time-out is invoked after the POR pulse has cleared. The total time-out will vary based on the status of the PWRT. Figure 5-3, Figure 5-4, Figure 5-5 and Figure 5-6 all depict time-out sequences on power-up.

Since the time-outs occur from the POR pulse, if $\overline{MCLR}$ is kept low long enough, the PWRT will expire. Bringing $\overline{MCLR}$ high will begin execution immediately (Figure 5-5). This is useful for testing purposes or to synchronize more than one PIC18FXXJ6X device operating in parallel.

FIGURE 5-3: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ TIED TO VDD, VDD RISE < TPWRT)

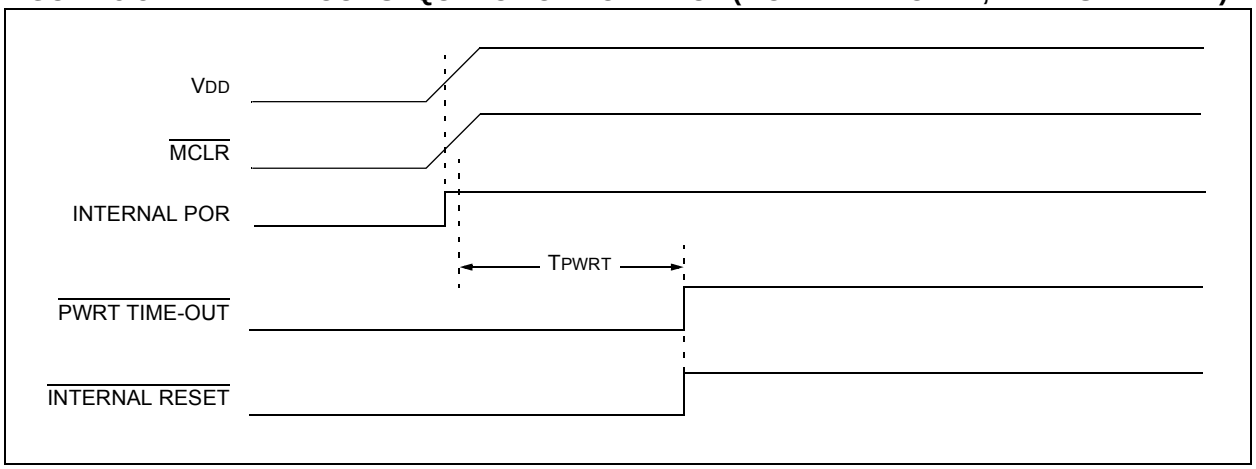


FIGURE 5-4: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ NOT TIED TO VDD): CASE 1

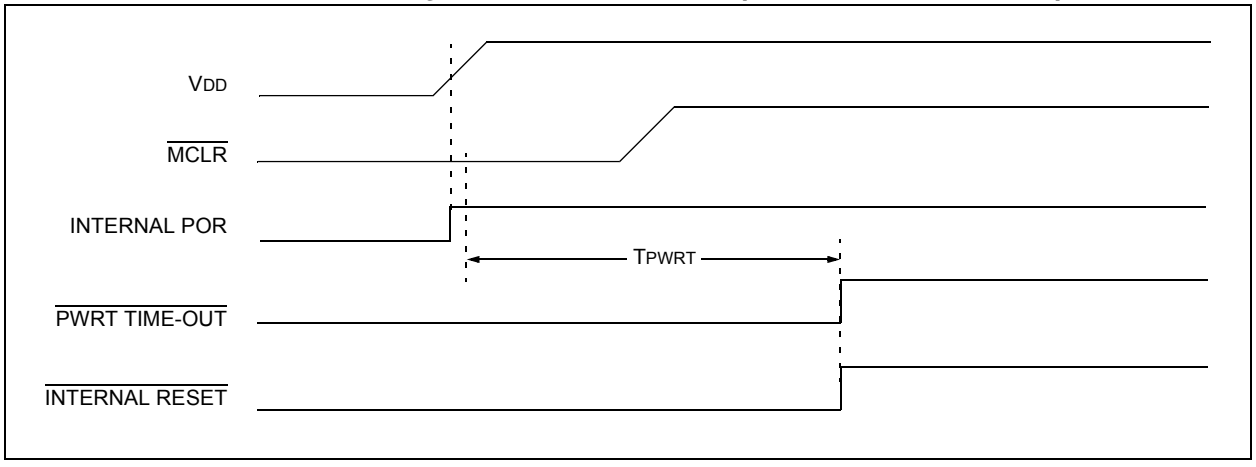


TABLE 5-2: INITIALIZATION CONDITIONS FOR ALL REGISTERS

Register	Applicable Devices			Power-on Reset, Brown-out Reset	MCLR Reset, WDT Reset, RESET Instruction, Stack Resets, CM Reset	Wake-up via WDT or Interrupt
TOSU	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---0 0000	---0 0000	---0 uuuu(1)
TOSH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu(1)
TOSL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu(1)
STKPTR	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	00-0 0000	uu-0 0000	uu-u uuuu(1)
PCLATU	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---0 0000	---0 0000	---u uuuu
PCLATH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
PCL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	PC + 2(2)
TBLPTRU	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	--00 0000	--00 0000	--uu uuuu
TBLPTRH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
TBLPTRL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
TABLAT	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 0000	0000 0000	uuuu uuuu
PRODH	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
PRODL	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
INTCON	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	0000 000x	0000 000u	uuuu uuuu(3)
INTCON2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	1111 1111	1111 1111	uuuu uuuu(3)
INTCON3	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	1100 0000	1100 0000	uuuu uuuu(3)
INDF0	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
POSTINC0	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
POSTDEC0	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
PREINC0	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
PLUSW0	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
FSR0H	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---- xxxx	---- uuuu	---- uuuu
FSR0L	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
WREG	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF1	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
POSTINC1	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
POSTDEC1	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
PREINC1	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
PLUSW1	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
FSR1H	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---- xxxx	---- uuuu	---- uuuu
FSR1L	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu uuuu
BSR	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---- 0000	---- 0000	---- uuuu
INDF2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
POSTINC2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
POSTDEC2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
PREINC2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
PLUSW2	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	N/A	N/A	N/A
FSR2H	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	---- xxxx	---- uuuu	---- uuuu
FSR2L	PIC18F6XJ6X	PIC18F8XJ6X	PIC18F9XJ6X	xxxx xxxx	uuuu uuuu	uuuu 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: 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.

2: 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).

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

4: See Table 5-1 for Reset value for specific condition.

6.4 Data Addressing Modes

Note: The execution of some instructions in the core PIC18 instruction set are changed when the PIC18 extended instruction set is enabled. See **Section 6.6 "Data Memory and the Extended Instruction Set"** for more information.

While the program memory can be addressed in only one way, through the program counter, information in the data memory space can be addressed in several ways. For most instructions, the addressing mode is fixed. Other instructions may use up to three modes, depending on which operands are used and whether or not the extended instruction set is enabled.

The addressing modes are:

- Inherent
- Literal
- Direct
- Indirect

An additional addressing mode, Indexed Literal Offset, is available when the extended instruction set is enabled (XINST Configuration bit = 1). Its operation is discussed in greater detail in **Section 6.6.1 "Indexed Addressing with Literal Offset"**.

6.4.1 INHERENT AND LITERAL ADDRESSING

Many PIC18 control instructions do not need any argument at all. They either perform an operation that globally affects the device, or they operate implicitly on one register. This addressing mode is known as Inherent Addressing. Examples include SLEEP, RESET and DAW.

Other instructions work in a similar way, but require an additional explicit argument in the opcode. This is known as Literal Addressing mode because they require some literal value as an argument. Examples include ADDLW and MOVLW, which respectively, add or move a literal value to the W register. Other examples include CALL and GOTO, which include a 20-bit program memory address.

6.4.2 DIRECT ADDRESSING

Direct Addressing mode specifies all or part of the source and/or destination address of the operation within the opcode itself. The options are specified by the arguments accompanying the instruction.

In the core PIC18 instruction set, bit-oriented and byte-oriented instructions use some version of Direct Addressing by default. All of these instructions include some 8-bit literal address as their Least Significant Byte. This address specifies either a register address in one of the banks of data RAM (**Section 6.3.3 "General Purpose Register File"**) or a location in the Access Bank (**Section 6.3.2 "Access Bank"**) as the data source for the instruction.

The Access RAM bit, 'a', determines how the address is interpreted. When 'a' is '1', the contents of the BSR (**Section 6.3.1 "Bank Select Register"**) are used with the address to determine the complete 12-bit address of the register. When 'a' is '0', the address is interpreted as being a register in the Access Bank. Addressing that uses the Access RAM is sometimes also known as Direct Forced Addressing mode.

A few instructions, such as MOVFF, include the entire 12-bit address (either source or destination) in their opcodes. In these cases, the BSR is ignored entirely.

The destination of the operation's results is determined by the destination bit, 'd'. When 'd' is '1', the results are stored back in the source register, overwriting its original contents. When 'd' is '0', the results are stored in the W register. Instructions without the 'd' argument have a destination that is implicit in the instruction. Their destination is either the target register being operated on or the W register.

6.4.3 INDIRECT ADDRESSING

Indirect Addressing mode allows the user to access a location in data memory without giving a fixed address in the instruction. This is done by using File Select Registers (FSRs) as pointers to the locations to be read or written to. Since the FSRs are themselves located in RAM as Special Function Registers, they can also be directly manipulated under program control. This makes FSRs very useful in implementing data structures, such as tables and arrays in data memory.

The registers for Indirect Addressing are also implemented with Indirect File Operands (INDFs) that permit automatic manipulation of the pointer value with auto-incrementing, auto-decrementing or offsetting with another value. This allows for efficient code using loops, such as the example of clearing an entire RAM bank in Example 6-5. It also enables users to perform Indexed Addressing and other Stack Pointer operations for program memory in data memory.

EXAMPLE 6-5: HOW TO CLEAR RAM (BANK 1) USING INDIRECT ADDRESSING

```
        LFSRFSR0, 100h;
NEXT    CLRFPOSTINC0; Clear INDF
                    ; register then
                    ; inc pointer
        BTFSSFSR0H, 1; All done with
                    ; Bank1?
        BRA NEXT    ; NO, clear next
CONTINUE            ; YES, continue
```

NOTES:

11.0 I/O PORTS

Depending on the device selected and features enabled, there are up to nine ports available. Some pins of the I/O ports are multiplexed with an alternate function from the peripheral features on the device. In general, when a peripheral is enabled, that pin may not be used as a general purpose I/O pin.

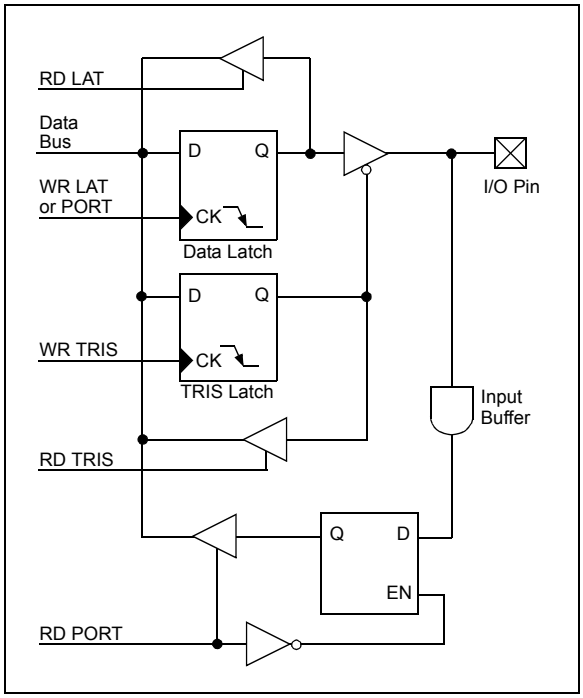
Each port has three registers for its operation. These registers are:

- TRIS register (Data Direction register)
- PORT register (reads the levels on the pins of the device)
- LAT register (Output Latch register)

The Output Latch (LAT register) is useful for read-modify-write operations on the value that the I/O pins are driving.

A simplified model of a generic I/O port, without the interfaces to other peripherals, is shown in Figure 11-1.

FIGURE 11-1: GENERIC I/O PORT OPERATION



11.1 I/O Port Pin Capabilities

When developing an application, the capabilities of the port pins must be considered. Outputs on some pins have higher output drive strength than others. Similarly, some pins can tolerate higher than V_{DD} input levels.

11.1.1 PIN OUTPUT DRIVE

The output pin drive strengths vary for groups of pins intended to meet the needs for a variety of applications. PORTB and PORTC are designed to drive higher loads, such as LEDs. The external memory interface ports (PORTD, PORTE and PORTJ) are designed to drive medium loads. All other ports are designed for small loads, typically indication only. Table 11-1 summarizes the output capabilities. Refer to **Section 28.0 "Electrical Characteristics"** for more details.

TABLE 11-1: OUTPUT DRIVE LEVELS

Port	Drive	Description
PORTA(1)	Minimum	Intended for indication.
PORTF(2)		
PORTG(2)		
PORTH(3)		
PORTD(2)	Medium	Sufficient drive levels for external memory interfacing, as well as indication.
PORTE		
PORTJ(3)		
PORTB	High	Suitable for direct LED drive levels.
PORTC		

Note 1: The exceptions are RA<1:0>, which are capable of directly driving LEDs.

2: Partially implemented on 64-pin and 80-pin devices; fully implemented on 100-pin devices.

3: Unimplemented on 64-pin devices.

TABLE 11-13: PORTF FUNCTIONS

Pin Name	Function	TRIS Setting	I/O	I/O Type	Description
RF0/AN5(1)	RF0(1)	0	O	DIG	LATF<0> data output; not affected by analog input.
		1	I	ST	PORTF<0> data input; disabled when analog input is enabled.
	AN5(1)	1	I	ANA	A/D Input Channel 5. Default configuration on POR.
RF1/AN6/ C2OUT	RF1	0	O	DIG	LATF<1> data output; not affected by analog input.
		1	I	ST	PORTF<1> data input; disabled when analog input is enabled.
	AN6	1	I	ANA	A/D Input Channel 6. Default configuration on POR.
	C2OUT	0	O	DIG	Comparator 2 output; takes priority over port data.
RF2/AN7/ C1OUT	RF2	0	O	DIG	LATF<2> data output; not affected by analog input.
		1	I	ST	PORTF<2> data input; disabled when analog input is enabled.
	AN7	1	I	ANA	A/D Input Channel 7. Default configuration on POR.
	C1OUT	0	O	TTL	Comparator 1 output; takes priority over port data.
RF3/AN8	RF3	0	O	DIG	LATF<3> data output; not affected by analog input.
		1	I	ST	PORTF<3> data input; disabled when analog input is enabled.
	AN8	1	I	ANA	A/D Input Channel 8 and Comparator C2+ input. Default input configuration on POR; not affected by analog output.
RF4/AN9	RF4	0	O	DIG	LATF<4> data output; not affected by analog input.
		1	I	ST	PORTF<4> data input; disabled when analog input is enabled.
	AN9	1	I	ANA	A/D Input Channel 9 and Comparator C2- input. Default input configuration on POR; does not affect digital output.
RF5/AN10/ CVREF	RF5	0	O	DIG	LATF<5> data output; not affected by analog input. Disabled when CVREF output is enabled.
		1	I	ST	PORTF<5> data input; disabled when analog input is enabled. Disabled when CVREF output is enabled.
	AN10	1	I	ANA	A/D Input Channel 10 and Comparator C1+ input. Default input configuration on POR.
	CVREF	x	O	ANA	Comparator voltage reference output. Enabling this feature disables digital I/O.
RF6/AN11	RF6	0	O	DIG	LATF<6> data output; not affected by analog input.
		1	I	ST	PORTF<6> data input; disabled when analog input is enabled.
	AN11	1	I	ANA	A/D Input Channel 11 and Comparator C1- input. Default input configuration on POR; does not affect digital output.
RF7/$\overline{SS1}$	RF7	0	O	DIG	LATF<7> data output.
		1	I	ST	PORTF<7> data input.
	$\overline{SS1}$	1	I	TTL	Slave select input for MSSP1 module.

Legend: O = Output, I = Input, ANA = Analog Signal, DIG = Digital Output, ST = Schmitt Buffer Input, TTL = TTL Buffer Input, x = Don't care (TRIS bit does not affect port direction or is overridden for this option).

Note 1: Implemented on 100-pin devices only.

TABLE 11-14: SUMMARY OF REGISTERS ASSOCIATED WITH PORTF

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
PORTF	RF7	RF6	RF5	RF4	RF3	RF2	RF1	RF0(1)	72
LATF	LATF7	LATF6	LATF5	LATF4	LATF3	LATF2	LATF1	LATF0(1)	72
TRISF	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	TRISF0(1)	71
ADCON1	—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0	70
CMCON	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0	70
CVRCON	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0	70

Legend: — = unimplemented, read as '0'. Shaded cells are not used by PORTF.

Note 1: Implemented on 100-pin devices only.

TABLE 11-19: PORTJ FUNCTIONS

Pin Name	Function	TRIS Setting	I/O	I/O Type	Description
RJ0/ALE(1)	RJ0(1)	0	O	DIG	LATJ<0> data output.
		1	I	ST	PORTJ<0> data input; weak pull-up when RJPU bit is set.
	ALE(1)	x	O	DIG	External memory interface address latch enable control output; takes priority over digital I/O.
RJ1/$\overline{OE}$(1)	RJ1(1)	0	O	DIG	LATJ<1> data output.
		1	I	ST	PORTJ<1> data input; weak pull-up when RJPU bit is set.
	$\overline{OE}$(1)	x	O	DIG	External memory interface output enable control output; takes priority over digital I/O.
RJ2/$\overline{WRL}$(1)	RJ2(1)	0	O	DIG	LATJ<2> data output.
		1	I	ST	PORTJ<2> data input; weak pull-up when RJPU bit is set.
	$\overline{WRL}$(1)	x	O	DIG	External memory bus write low byte control; takes priority over digital I/O.
RJ3/$\overline{WRH}$(1)	RJ3(1)	0	O	DIG	LATJ<3> data output.
		1	I	ST	PORTJ<3> data input; weak pull-up when RJPU bit is set.
	$\overline{WRH}$(1)	x	O	DIG	External memory interface write high byte control output; takes priority over digital I/O.
RJ4/BA0	RJ4	0	O	DIG	LATJ<4> data output.
		1	I	ST	PORTJ<4> data input; weak pull-up when RJPU bit is set.
	BA0(2)	x	O	DIG	External Memory Interface Byte Address 0 control output; takes priority over digital I/O.
RJ5/$\overline{CE}$	RJ5	0	O	DIG	LATJ<5> data output.
		1	I	ST	PORTJ<5> data input; weak pull-up when RJPU bit is set.
	$\overline{CE}$(2)	x	O	DIG	External memory interface chip enable control output; takes priority over digital I/O.
RJ6/$\overline{LB}$(1)	RJ6(1)	0	O	DIG	LATJ<6> data output.
		1	I	ST	PORTJ<6> data input; weak pull-up when RJPU bit is set.
	$\overline{LB}$(1)	x	O	DIG	External memory interface lower byte enable control output; takes priority over digital I/O.
RJ7/$\overline{UB}$(1)	RJ7(1)	0	O	DIG	LATJ<7> data output.
		1	I	ST	PORTJ<7> data input; weak pull-up when RJPU bit is set.
	$\overline{UB}$(1)	x	O	DIG	External memory interface upper byte enable control output; takes priority over digital I/O.

Legend: O = Output, I = Input, DIG = Digital Output, ST = Schmitt Buffer Input,
x = Don't care (TRIS bit does not affect port direction or is overridden for this option).

Note 1: Implemented on 100-pin devices only.

2: EMB functions are implemented on 100-pin devices only.

TABLE 11-20: SUMMARY OF REGISTERS ASSOCIATED WITH PORTJ

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
PORTJ	RJ7(1)	RJ6(1)	RJ5	RJ4	RJ3(1)	RJ2(1)	RJ1(1)	RJ0(1)	72
LATJ	LATJ7(1)	LATJ6(1)	LATJ5	LATJ4	LATJ3(1)	LATJ2(1)	LATJ1(1)	LATJ0(1)	71
TRISJ	TRISJ7(1)	TRISJ6(1)	TRISJ5	TRISJ4	TRISJ3(1)	TRISJ2(1)	TRISJ1(1)	TRISJ0(1)	71
PORTA	RJPU	—	RA5	RA4	RA3	RA2	RA1	RA0	72

Legend: — = unimplemented, read as '0'. Shaded cells are not used by PORTJ.

Note 1: Implemented on 100-pin devices only.

16.0 TIMER4 MODULE

The Timer4 module has the following features:

- 8-Bit Timer register (TMR4)
- 8-Bit Period register (PR4)
- Readable and writable (both registers)
- Software programmable prescaler (1:1, 1:4, 1:16)
- Software programmable postscaler (1:1 to 1:16)
- Interrupt on TMR4 match of PR4

Timer4 has a control register, shown in Register 16-1. Timer4 can be shut off by clearing control bit, TMR4ON (T4CON<2>), to minimize power consumption. The prescaler and postscaler selection of Timer4 is also controlled by this register. Figure 16-1 is a simplified block diagram of the Timer4 module.

16.1 Timer4 Operation

Timer4 can be used as the PWM time base for the PWM mode of the CCP module. The TMR4 register is readable and writable, and is cleared on any device Reset. The input clock (FOSC/4) has a prescale option of 1:1, 1:4 or 1:16, selected by control bits, T4CKPS<1:0> (T4CON<1:0>). The match output of TMR4 goes through a 4-bit postscaler (which gives a 1:1 to 1:16 scaling inclusive) to generate a TMR4 interrupt, latched in flag bit, TMR4IF (PIR3<3>).

The prescaler and postscaler counters are cleared when any of the following occurs:

- A write to the TMR4 register
- A write to the T4CON register
- Any device Reset (Power-on Reset, $\overline{\text{MCLR}}$ Reset, Watchdog Timer Reset or Brown-out Reset)

TMR4 is not cleared when T4CON is written.

REGISTER 16-1: T4CON: TIMER4 CONTROL REGISTER

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	T4OUTPS3	T4OUTPS2	T4OUTPS1	T4OUTPS0	TMR4ON	T4CKPS1	T4CKPS0
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 **Unimplemented:** Read as ‘0’

bit 6-3 **T4OUTPS<3:0>**: Timer4 Output Postscale Select bits
0000 = 1:1 Postscale
0001 = 1:2 Postscale
•
•
•
1111 = 1:16 Postscale

bit 2 **TMR4ON**: Timer4 On bit
1 = Timer4 is on
0 = Timer4 is off

bit 1-0 **T4CKPS<1:0>**: Timer4 Clock Prescale Select bits
00 = Prescaler is 1
01 = Prescaler is 4
1x = Prescaler is 16

18.1 ECCPx Outputs and Configuration

Each of the Enhanced CCPx modules may have up to four PWM outputs, depending on the selected operating mode. These outputs, designated PxA through PxD, are multiplexed with various I/O pins. Some ECCPx pin assignments are constant, while others change based on device configuration. For those pins that do change, the controlling bits are:

- CCP2MX Configuration bit (80-pin and 100-pin devices only)
- ECCPMX Configuration bit (80-pin and 100-pin devices only)
- Program memory operating mode set by the EMB Configuration bits (100-pin devices only)

The pin assignments for the Enhanced CCPx modules are summarized in Table 18-1, Table 18-2 and Table 18-3. To configure the I/O pins as PWM outputs, the proper PWM mode must be selected by setting the PxMx and CCPxMx bits (CCPxCON<7:6> and <3:0>, respectively). The appropriate TRIS direction bits for the corresponding port pins must also be set as outputs.

18.1.1 ECCP1/ECCP3 OUTPUTS AND PROGRAM MEMORY MODE

In 100-pin devices, the use of Extended Microcontroller mode has an indirect effect on the ECCP1 and ECCP3 pins in Enhanced PWM modes. By default, PWM outputs, P1B/P1C and P3B/P3C, are multiplexed to PORTE pins, along with the high-order byte of the external memory bus. When the bus is active in Extended Microcontroller mode, it overrides the Enhanced CCPx outputs and makes them unavailable. Because of this, ECCP1 and ECCP3 can only be used in compatible (single output) PWM modes when the device is in Extended Microcontroller mode with default pin configuration.

An exception to this configuration is when a 12-bit address width is selected for the external bus (EMB<1:0> Configuration bits = 10). In this case, the upper pins of PORTE continue to operate as digital I/O, even when the external bus is active. P1B/P1C and P3B/P3C remain available for use as Enhanced PWM outputs.

If an application requires the use of additional PWM outputs during Extended Microcontroller mode, the P1B/P1C and P3B/P3C outputs can be reassigned to the upper bits of PORTH. This is done by clearing the ECCPMX Configuration bit.

18.1.2 ECCP2 OUTPUTS AND PROGRAM MEMORY MODES

For 100-pin devices, the Program Memory mode of the device (**Section 6.1.3 “PIC18F9XJ60/9XJ65 Program Memory Modes”**) also impacts pin multiplexing for the module.

The ECCP2 input/output (ECCP2/P2A) can be multiplexed to one of three pins. The default assignment (CCP2MX Configuration bit is set) for all devices is RC1. Clearing CCP2MX reassigns ECCP2/P2A to RE7 in 80-pin and 100-pin devices.

An additional option exists for 100-pin devices. When these devices are operating in Microcontroller mode, the multiplexing options described above still apply. In Extended Microcontroller mode, clearing CCP2MX reassigns ECCP2/P2A to RB3.

18.1.3 USE OF CCP4 AND CCP5 WITH ECCP1 AND ECCP3

Only the ECCP2 module has four dedicated, output pins that are available for use. Assuming that the I/O ports or other multiplexed functions on those pins are not needed, they may be used without interfering with any other CCPx module.

ECCP1 and ECCP3, on the other hand, only have three dedicated output pins: ECCPx/PxA, PxB and PxC. Whenever these modules are configured for Quad PWM mode, the pin normally used for CCP4 or CCP5 becomes the PxD output pin for ECCP3 and ECCP1, respectively. The CCP4 and CCP5 modules remain functional but their outputs are overridden.

18.1.4 ECCPx MODULES AND TIMER RESOURCES

Like the standard CCPx modules, the ECCPx modules can utilize Timers 1, 2, 3 or 4, depending on the mode selected. Timer1 and Timer3 are available for modules in Capture or Compare modes, while Timer2 and Timer4 are available for modules in PWM mode. Additional details on timer resources are provided in **Section 17.1.1 “CCPx/ECCPx Modules and Timer Resources”**.

20.4.3.3 Reception

When the R/$\overline{W}$ bit of the address byte is clear and an address match occurs, the R/$\overline{W}$ bit of the SSPxSTAT register is cleared. The received address is loaded into the SSPxBUF register and the SDAx line is held low ($\overline{ACK}$).

When the address byte overflow condition exists, then the no Acknowledge ($\overline{ACK}$) pulse is given. An overflow condition is defined as either bit, BF (SSPxSTAT<0>), is set, or bit, SSPOV (SSPxCON1<6>), is set.

An MSSP interrupt is generated for each data transfer byte. The interrupt flag bit, SSPxIF, must be cleared in software. The SSPxSTAT register is used to determine the status of the byte.

If SEN is enabled (SSPxCON2<0> = 1), SCKx/SCLx (RC3 or RD6) will be held low (clock stretch) following each data transfer. The clock must be released by setting bit, CKP (SSPxCON1<4>). See **Section 20.4.4 "Clock Stretching"** for more details.

20.4.3.4 Transmission

When the R/$\overline{W}$ bit of the incoming address byte is set and an address match occurs, the R/$\overline{W}$ bit of the SSPxSTAT register is set. The received address is loaded into the SSPxBUF register. The $\overline{ACK}$ pulse will be sent on the ninth bit and pin RC3 or RD6 is held low, regardless of SEN (see **Section 20.4.4 "Clock Stretching"** for more details). By stretching the clock, the master will be unable to assert another clock pulse until the slave is done preparing the transmit data. The transmit data must be loaded into the SSPxBUF register which also loads the SSPxSR register. Then, pin, RC3 or RD6, should be enabled by setting bit, CKP (SSPxCON1<4>). The eight data bits are shifted out on the falling edge of the SCLx input. This ensures that the SDAx signal is valid during the SCLx high time (Figure 20-10).

The $\overline{ACK}$ pulse from the master-receiver is latched on the rising edge of the ninth SCLx input pulse. If the SDAx line is high (not $\overline{ACK}$), then the data transfer is complete. In this case, when the $\overline{ACK}$ is latched by the slave, the slave logic is reset (resets SSPxSTAT register) and the slave monitors for another occurrence of the Start bit. If the SDAx line was low ($\overline{ACK}$), the next transmit data must be loaded into the SSPxBUF register. Again, pin, RC3 or RD6, must be enabled by setting bit, CKP.

An MSSP interrupt is generated for each data transfer byte. The SSPxIF bit must be cleared in software and the SSPxSTAT register is used to determine the status of the byte. The SSPxIF bit is set on the falling edge of the ninth clock pulse.

26.0 INSTRUCTION SET SUMMARY

The PIC18F97J60 family of devices incorporates the standard set of 75 PIC18 core instructions, as well as an extended set of 8 new instructions for the optimization of code that is recursive or that utilizes a software stack. The extended set is discussed later in this section.

26.1 Standard Instruction Set

The standard PIC18 instruction set adds many enhancements to the previous PIC® MCU instruction sets, while maintaining an easy migration from these PIC MCU instruction sets. Most instructions are a single program memory word (16 bits), but there are four instructions that require two program memory locations.

Each single-word instruction is a 16-bit word divided into an opcode, which specifies the instruction type and one or more operands, which further specify the operation of the instruction.

The instruction set is highly orthogonal and is grouped into four basic categories:

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal** operations
- **Control** operations

The PIC18 instruction set summary in Table 26-2 lists **byte-oriented**, **bit-oriented**, **literal** and **control** operations. Table 26-1 shows the opcode field descriptions.

Most **byte-oriented** instructions have three operands:

1. The file register (specified by ‘f’)
2. The destination of the result (specified by ‘d’)
3. The accessed memory (specified by ‘a’)

The file register designator ‘f’ specifies which file register is to be used by the instruction. The destination designator ‘d’ specifies where the result of the operation is to be placed. If ‘d’ is zero, the result is placed in the WREG register. If ‘d’ is one, the result is placed in the file register specified in the instruction.

All **bit-oriented** instructions have three operands:

1. The file register (specified by ‘f’)
2. The bit in the file register (specified by ‘b’)
3. The accessed memory (specified by ‘a’)

The bit field designator ‘b’ selects the number of the bit affected by the operation, while the file register designator ‘f’ represents the number of the file in which the bit is located.

The **literal** instructions may use some of the following operands:

- A literal value to be loaded into a file register (specified by ‘k’)
- The desired FSR register to load the literal value into (specified by ‘f’)
- No operand required (specified by ‘—’)

The **control** instructions may use some of the following operands:

- A program memory address (specified by ‘n’)
- The mode of the CALL or RETURN instructions (specified by ‘s’)
- The mode of the table read and table write instructions (specified by ‘m’)
- No operand required (specified by ‘—’)

All instructions are a single word, except for four double-word instructions. These instructions were made double-word to contain the required information in 32 bits. In the second word, the 4 MSbs are ‘1’s. If this second word is executed as an instruction (by itself), it will execute as a NOP.

All single-word instructions are executed in a single instruction cycle, unless a conditional test is true or the program counter is changed as a result of the instruction. In these cases, the execution takes two instruction cycles with the additional instruction cycle(s) executed as a NOP.

The double-word instructions execute in two instruction cycles.

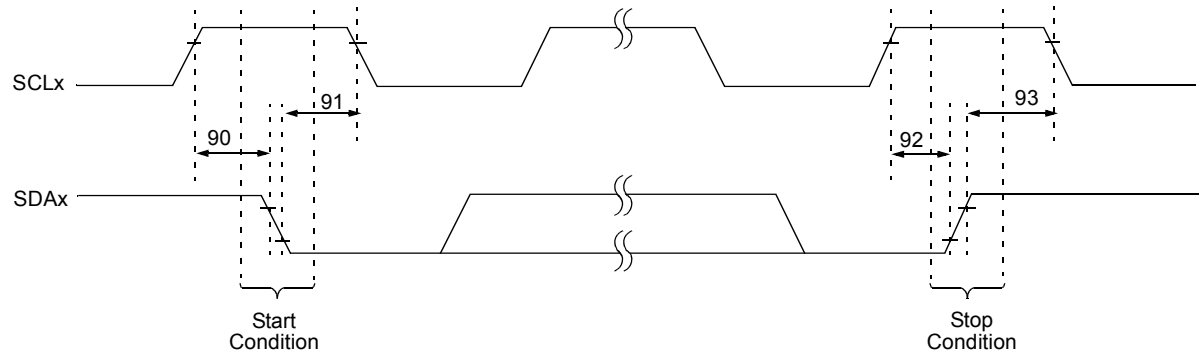
One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1 μs. If a conditional test is true, or the program counter is changed as a result of an instruction, the instruction execution time is 2 μs. Two-word branch instructions (if true) would take 3 μs.

Figure 26-1 shows the general formats that the instructions can have. All examples use the convention ‘nnh’ to represent a hexadecimal number.

The Instruction Set Summary, shown in Table 26-2, lists the standard instructions recognized by the Microchip MPASM™ Assembler.

Section 26.1.1 “Standard Instruction Set” provides a description of each instruction.

FIGURE 28-17: MASTER SSP I²C™ BUS START/STOP BITS TIMING WAVEFORMS



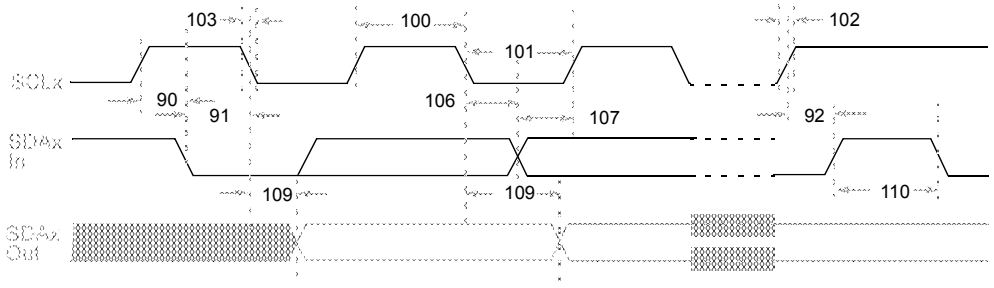
Note: Refer to Figure 28-3 for load conditions.

TABLE 28-22: MASTER SSP I²C™ BUS START/STOP BITS REQUIREMENTS

Param. No.	Symbol	Characteristic		Min	Max	Units	Conditions
90	TSU:STA	Start Condition Setup Time	100 kHz mode	2(TOSC)(BRG + 1)	—	ns	Only relevant for Repeated Start condition
			400 kHz mode	2(TOSC)(BRG + 1)	—		
			1 MHz mode(1)	2(TOSC)(BRG + 1)	—		
91	THD:STA	Start Condition Hold Time	100 kHz mode	2(TOSC)(BRG + 1)	—	ns	After this period, the first clock pulse is generated
			400 kHz mode	2(TOSC)(BRG + 1)	—		
			1 MHz mode(1)	2(TOSC)(BRG + 1)	—		
92	TSU:STO	Stop Condition Setup Time	100 kHz mode	2(TOSC)(BRG + 1)	—	ns	
			400 kHz mode	2(TOSC)(BRG + 1)	—		
			1 MHz mode(1)	2(TOSC)(BRG + 1)	—		
93	THD:STO	Stop Condition Hold Time	100 kHz mode	2(TOSC)(BRG + 1)	—	ns	
			400 kHz mode	2(TOSC)(BRG + 1)	—		
			1 MHz mode(1)	2(TOSC)(BRG + 1)	—		

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

FIGURE 28-18: MASTER SSP I²C™ BUS DATA TIMING



Note: Refer to Figure 28-3 for load conditions.