



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	Obsolete
Core Processor	PIC
Core Size	8-Bit
Speed	40MHz
Connectivity	CANbus, I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	52
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	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f6585t-i-pt

PIC18F6585/8585/6680/8680

TABLE 1-2: PIC18F6585/8585/6680/8680 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PIC18F6X8X		PIC18F8X8X			
	TQFP	PLCC	TQFP			
RA0/AN0 RA0 AN0	24	34	30	I/O I	TTL Analog	PORTA is a bidirectional I/O port. Digital I/O. Analog input 0. Digital I/O. Analog input 1. Digital I/O. Analog input 2. A/D reference voltage (Low) input. Digital I/O. Analog input 3. A/D reference voltage (High) input. Digital I/O – Open-drain when configured as output. Timer0 external clock input. Digital I/O. Analog input 4. Low-voltage detect input. See the OSC2/CLKO/RA6 pin.
RA1/AN1 RA1 AN1	23	33	29	I/O I	TTL Analog	
RA2/AN2/VREF- RA2 AN2 VREF-	22	32	28	I/O I I	TTL Analog Analog	
RA3/AN3/VREF+ RA3 AN3 VREF+	21	31	27	I/O I I	TTL Analog Analog	
RA4/T0CKI RA4 T0CKI	28	39	34	I/O I	ST/OD ST	
RA5/AN4/LVDIN RA5 AN4 LVDIN	27	38	33	I/O I I	TTL Analog Analog	
RA6						

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 VDD)

- Note 1:** Alternate assignment for CCP2 in all operating modes except Microcontroller – applies to PIC18F8X8X only.
2: Default assignment when CCP2MX is set.
3: External memory interface functions are only available on PIC18F8X8X devices.
4: CCP2 is multiplexed with this pin by default when configured in Microcontroller mode; otherwise, it is multiplexed with either RB3 or RC1.
5: PORTH and PORTJ are only available on PIC18F8X8X (80-pin) devices.
6: PSP is available in Microcontroller mode only.
7: On PIC18F8X8X devices, these pins can be multiplexed with RH7/RH6 by changing the ECCPMX configuration bit.

PIC18F6585/8585/6680/8680

TABLE 1-2: PIC18F6585/8585/6680/8680 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PIC18F6X8X		PIC18F8X8X			
	TQFP	PLCC	TQFP			
RD0/PSP0/AD0 RD0 PSP0 ⁽⁶⁾ AD0 ⁽³⁾	58	3	72	I/O I/O I/O	ST TTL TTL	PORTD is a bidirectional I/O port. These pins have TTL input buffers when external memory is enabled. Digital I/O. Parallel Slave Port data. External memory address/data 0.
RD1/PSP1/AD1 RD1 PSP1 ⁽⁶⁾ AD1 ⁽³⁾	55	67	69	I/O I/O I/O	ST TTL TTL	Digital I/O. Parallel Slave Port data. External memory address/data 1.
RD2/PSP2/AD2 RD2 PSP2 ⁽⁶⁾ AD2 ⁽³⁾	54	66	68	I/O I/O I/O	ST TTL TTL	Digital I/O. Parallel Slave Port data. External memory address/data 2.
RD3/PSP3/AD3 RD3 PSP3 ⁽⁶⁾ AD3 ⁽³⁾	53	65	67	I/O I/O I/O	ST TTL TTL	Digital I/O. Parallel Slave Port data. External memory address/data 3.
RD4/PSP4/AD4 RD4 PSP4 ⁽⁶⁾ AD4 ⁽³⁾	52	64	66	I/O I/O I/O	ST TTL TTL	Digital I/O. Parallel Slave Port data. External memory address/data 4.
RD5/PSP5/AD5 RD5 PSP5 ⁽⁶⁾ AD5 ⁽³⁾	51	63	65	I/O I/O I/O	ST TTL TTL	Digital I/O. Parallel Slave Port data. External memory address/data 5.
RD6/PSP6/AD6 RD6 PSP6 ⁽⁶⁾ AD6 ⁽³⁾	50	62	64	I/O I/O I/O	ST TTL TTL	Digital I/O. Parallel Slave Port data. External memory address/data 6.
RD7/PSP7/AD7 RD7 PSP7 ⁽⁶⁾ AD7 ⁽³⁾	49	61	63	I/O I/O I/O	ST TTL TTL	Digital I/O. Parallel Slave Port data. External memory address/data 7.

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:** Alternate assignment for CCP2 in all operating modes except Microcontroller – applies to PIC18F8X8X only.
2: Default assignment when CCP2MX is set.
3: External memory interface functions are only available on PIC18F8X8X devices.
4: CCP2 is multiplexed with this pin by default when configured in Microcontroller mode; otherwise, it is multiplexed with either RB3 or RC1.
5: PORTH and PORTJ are only available on PIC18F8X8X (80-pin) devices.
6: PSP is available in Microcontroller mode only.
7: On PIC18F8X8X devices, these pins can be multiplexed with RH7/RH6 by changing the ECCPMX configuration bit.

PIC18F6585/8585/6680/8680

REGISTER 6-1: MEMCON REGISTER

R/W-0	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0
EBDIS ⁽¹⁾	—	WAIT1	WAIT0	—	—	WM1	WM0
bit 7						bit 0	

bit 7 **EBDIS:** External Bus Disable bit⁽¹⁾

1 = External system bus disabled, all external bus drivers are mapped as I/O ports

0 = External system bus enabled and I/O ports are disabled

Note 1: This bit is ignored when device is accessing external memory either to fetch an instruction or perform TBLRD/TBLWT.

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

bit 5-4 **WAIT<1:0>:** Table Reads and Writes Bus Cycle Wait Count bits

11 = Table reads and writes will wait 0 Tcy

10 = Table reads and writes will wait 1 Tcy

01 = Table reads and writes will wait 2 Tcy

00 = Table reads and writes will wait 3 Tcy

bit 3-2 **Unimplemented:** Read as '0'

bit 1-0 **WM<1:0>:** TBLWT Operation with 16-bit Bus bits

1x = Word Write mode: LSB and MSB word output, \overline{WRH} active when \overline{MSB} written

01 = Byte Select mode: TABLAT data copied on both MS and LS Byte, \overline{WRH} and (\overline{UB} or \overline{LB}) will activate

00 = Byte Write mode: TABLAT data copied on both MS and LS Byte, \overline{WRH} or \overline{WRL} will activate

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

Note: The MEMCON register is held in Reset in Microcontroller mode.

7.0 DATA EEPROM MEMORY

The data EEPROM is readable and writable during normal operation over the entire VDD range. The data memory is not directly mapped in the register file space. Instead, it is indirectly addressed through the Special Function Registers (SFR).

There are five SFRs used to read and write the program and data EEPROM memory. These registers are:

- EECON1
- EECON2
- EEDATA
- EEADR
- EEADRH

The EEPROM data memory allows byte read and write. When interfacing to the data memory block, EEDATA holds the 8-bit data for read/write and EEADR holds the address of the EEPROM location being accessed. These devices have 1024 bytes of data EEPROM with an address range from 0h to 3FFh.

The EEPROM data memory is rated for high erase/write cycles. A byte write automatically erases the location and writes the new data (erase-before-write). The write time is controlled by an on-chip timer. The write time will vary with voltage and temperature as well as from chip to chip. Please refer to parameter D122 (Electrical Characteristics, **Section 27.0 “Electrical Characteristics”**) for exact limits.

7.1 EEADRH:EEADR

The address register pair, EEADRH:EEADR, can address up to a maximum of 1024 bytes of data EEPROM.

7.2 EECON1 and EECON2 Registers

EECON1 is the control register for EEPROM memory accesses.

EECON2 is not a physical register. Reading EECON2 will read all '0's. The EECON2 register is used exclusively in the EEPROM write sequence.

Control bits RD and WR initiate read and write operations, respectively. These bits cannot be cleared, only set in software. They are cleared in hardware at the completion of the read or write operation. The inability to clear the WR bit in software prevents the accidental or premature termination of a write operation.

The WREN bit, when set, will allow a write operation. On power-up, the WREN bit is clear. The WRERR bit is set when a write operation is interrupted by a MCLR Reset or a WDT Time-out Reset during normal operation. In these situations, the user can check the WRERR bit and rewrite the location. It is necessary to reload the data and address registers (EEDATA and EEADR) due to the Reset condition forcing the contents of the registers to zero.

Note: Interrupt flag bit, EEIF in the PIR2 register, is set when write is complete. It must be cleared in software.
--

9.0 INTERRUPTS

The PIC18F6585/8585/6680/8680 devices have multiple interrupt sources and an interrupt priority feature that allows each interrupt source to be assigned a high or a low priority level. The high priority interrupt vector is at 000008h while the low priority interrupt vector is at 000018h. High priority interrupt events will override any low priority interrupts that may be in progress.

There are thirteen registers which are used to control interrupt operation. They are:

- RCON
- INTCON
- INTCON2
- INTCON3
- PIR1, PIR2, PIR3
- PIE1, PIE2, PIE3
- IPR1, IPR2, IPR3

It is recommended that the Microchip header files supplied with MPLAB® IDE be used for the symbolic bit names in these registers. This allows the assembler/compiler to automatically take care of the placement of these bits within the specified register.

Each interrupt source (except INT0) has three bits to control its operation. The functions of these bits are:

- Flag bit to indicate that an interrupt event occurred
- Enable bit that allows program execution to branch to the interrupt vector address when the flag bit is set
- Priority bit to select high priority or low priority

The interrupt priority feature is enabled by setting the IPEN bit (RCON<7>). When interrupt priority is enabled, there are two bits which enable interrupts globally. Setting the GIEH bit (INTCON<7>) enables all interrupts that have the priority bit set. Setting the GIEL bit (INTCON<6>) enables all interrupts that have the priority bit cleared. When the interrupt flag, enable bit and appropriate global interrupt enable bit are set, the interrupt will vector immediately to address 000008h or 000018h depending on the priority level. Individual interrupts can be disabled through their corresponding enable bits.

When the IPEN bit is cleared (default state), the interrupt priority feature is disabled and interrupts are compatible with PIC® mid-range devices. In Compatibility mode, the interrupt priority bits for each source have no effect. INTCON<6> is the PEIE bit which enables/disables all peripheral interrupt sources. INTCON<7> is the GIE bit which enables/disables all interrupt sources. All interrupts branch to address 000008h in Compatibility mode.

When an interrupt is responded to, the global interrupt enable bit is cleared to disable further interrupts. If the IPEN bit is cleared, this is the GIE bit. If interrupt priority levels are used, this will be either the GIEH or GIEL bit. High priority interrupt sources can interrupt a low priority interrupt.

The return address is pushed onto the stack and the PC is loaded with the interrupt vector address (000008h or 000018h). Once in the Interrupt Service Routine, the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bits must be cleared in software before re-enabling interrupts to avoid recursive interrupts.

The “return from interrupt” instruction, RETFIE, exits the interrupt routine and sets the GIE bit (GIEH or GIEL if priority levels are used) which re-enables interrupts.

For external interrupt events, such as the INT pins or the PORTB input change interrupt, the interrupt latency will be three to four instruction cycles. The exact latency is the same for one- or two-cycle instructions. Individual interrupt flag bits are set regardless of the status of their corresponding enable bit or the GIE bit.

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

10.0 I/O PORTS

Depending on the device selected, there are either seven or nine I/O ports available on PIC18F6X8X/8X8X devices. Some of their pins are multiplexed with one or more alternate functions from the other 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.

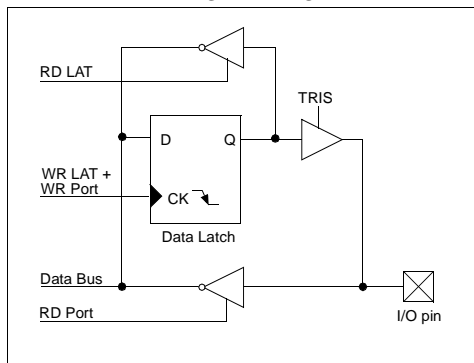
Each port has three registers for its operation. These registers are:

- TRIS register (data direction register)
- PORT register (reads the levels on the pins of the device)
- LAT register (output latch)

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

A simplified version of a generic I/O port and its operation is shown in Figure 10-1.

FIGURE 10-1: SIMPLIFIED BLOCK DIAGRAM OF PORT/LAT/TRIS OPERATION



10.1 PORTA, TRISA and LATA Registers

PORTA is a 7-bit wide, bidirectional port. The corresponding data direction register is TRISA. Setting a TRISA bit (= 1) will make the corresponding PORTA pin an input (i.e., put the corresponding output driver in a high-impedance mode). Clearing a TRISA bit (= 0) will make the corresponding PORTA pin an output (i.e., put the contents of the output latch on the selected pin).

Reading the PORTA register reads the status of the pins, whereas writing to it will write to the port latch.

The Data Latch register (LATA) is also memory mapped. Read-modify-write operations on the LATA register read and write the latched output value for PORTA.

The RA4 pin is multiplexed with the Timer0 module clock input to become the RA4/T0CKI pin. The RA4/T0CKI pin is a Schmitt Trigger input and an open-drain output. All other RA port pins have TTL input levels and full CMOS output drivers.

The RA6 pin is only enabled as a general I/O pin in ECIO and RCIO Oscillator modes.

The other PORTA pins are multiplexed with analog inputs and the analog VREF+ and VREF- inputs. The operation of each pin is selected by clearing/setting the control bits in the ADCON1 register (A/D Control Register 1).

Note: On a Power-on Reset, RA5 and RA3:RA0 are configured as analog inputs and read as '0'. RA6 and RA4 are configured as digital inputs.

The TRISA register controls the direction of the RA pins even when they are being used as analog inputs. The user must ensure the bits in the TRISA register are maintained set when using them as analog inputs.

EXAMPLE 10-1: INITIALIZING PORTA

```
CLRF   PORTA    ; Initialize PORTA by
                ; clearing output
                ; data latches
CLRF   LATA      ; Alternate method
                ; to clear output
                ; data latches
MOVLW  0Fh      ; Configure A/D
MOVWF  ADCON1   ; for digital inputs
MOVLW  0CFh     ; Value used to
                ; initialize data
                ; direction
MOVWF  TRISA    ; Set RA<3:0> as inputs
                ; RA<5:4> as outputs
```


PIC18F6585/8585/6680/8680

REGISTER 10-1: PSPCON REGISTER

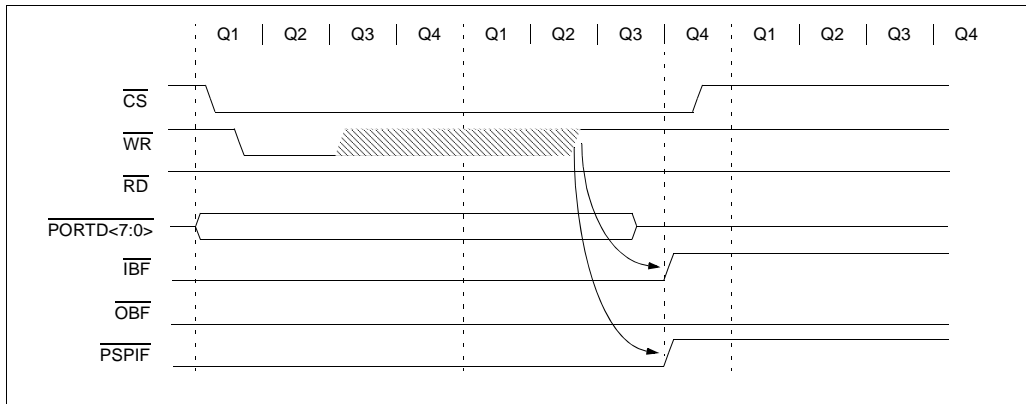
R-0	R-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
IBF	OBF	IBOV	PSPMODE	—	—	—	—
bit 7				bit 0			

- bit 7 **IBF:** Input Buffer Full Status bit
 1 = A data byte has been received and is waiting to be read by the CPU
 0 = No data byte has been received
- bit 6 **OBF:** Output Buffer Full Status bit
 1 = The output buffer still holds a previously written data byte
 0 = The output buffer has been read
- bit 5 **IBOV:** Input Buffer Overflow Detect bit
 1 = A write occurred when a previously input data byte has not been read
 (must be cleared in software)
 0 = No overflow occurred
- bit 4 **PSPMODE:** Parallel Slave Port Mode Select bit
 1 = Parallel Slave Port mode
 0 = General Purpose I/O mode
- bit 3-0 **Unimplemented:** Read as '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

FIGURE 10-29: PARALLEL SLAVE PORT WRITE WAVEFORMS



15.0 CAPTURE/COMPARE/PWM (CCP) MODULES

PIC18FXX80/XX85 devices contain a total of two CCP modules: CCP1 and CCP2. CCP1 is an enhanced version of the CCP2 module. CCP1 is fully backward compatible with the CCP2 module.

The CCP1 module differs from CCP2 in the following respect:

- CCP1 contains a special trigger event that may reset Timer1 or the Timer3 register pair
- CCP1 contains "CAN Message Time-Stamp Trigger"
- CCP1 contains enhanced PWM output with programmable dead band and auto-shutdown functionality

Additionally, the CCP2 special event trigger may be used to start an A/D conversion if the A/D module is enabled.

To avoid duplicate information, this section describes basic CCP module operation that applies to both CCP1 and CCP2. Enhanced CCP functionality of the CCP1 module is described in **Section 16.0 "Enhanced Capture/Compare/PWM (ECCP) Module"**.

The control registers for the CCP1 and CCP2 modules are shown in Register 15-1 and Register 15-2, respectively. Table 15-2 details the interactions of the CCP and ECCP modules.

REGISTER 15-1: CCP1CON REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0
bit 7							bit 0

bit 7-6 **P1M1:P1M0**: Enhanced PWM Output Configuration bits

If CCP1M<3:2> = 00, 01, 10:

xx = P1A assigned as capture/compare input; P1B, P1C, P1D assigned as port pins

If CCP1M<3:2> = 11:

00 = Single output; P1A modulated; P1B, P1C, P1D assigned as port pins

01 = Full-bridge output forward; P1D modulated; P1A active; P1B, P1C inactive

10 = Half-bridge output; P1A, P1B modulated with dead-band control; P1C, P1D assigned as port pins

11 = Full-bridge output reverse; P1B modulated; P1C active; P1A, P1D inactive

bit 5-4 **DC1B1:DC1B0**: PWM Duty Cycle bit 1 and bit 0

Capture mode:

Unused.

Compare mode:

Unused.

PWM mode:

These bits are the two LSBs of the 10-bit PWM duty cycle. The eight MSBs of the duty cycle are found in CCP1L.

bit 3-0 **CCP1M3:CCP1M0**: Enhanced CCP Mode Select bits

0000 = Capture/Compare/PWM off (resets CCP1 module)

0001 = Reserved

0010 = Compare mode, toggle output on match

0011 = Reserved

0100 = Capture mode, every falling edge

0101 = Capture mode, every rising edge

0110 = Capture mode, every 4th rising edge

0111 = Capture mode, every 16th rising edge

1000 = Compare mode, initialize CCP pin low, on compare match force CCP pin high

1001 = Compare mode, initialize CCP pin high, on compare match force CCP pin low

1010 = Compare mode, generate software interrupt only, CCP pin is unaffected

1011 = Compare mode, trigger special event, resets TMR1 or TMR3

1100 = PWM mode; P1A, P1C active-high; P1B, P1D active-high

1101 = PWM mode; P1A, P1C active-high; P1B, P1D active-low

1110 = PWM mode; P1A, P1C active-low; P1B, P1D active-high

1111 = PWM mode; P1A, P1C active-low; P1B, P1D active-low

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

15.1 CCP Module

Both CCP1 and CCP2 are comprised of two 8-bit registers: CCPRxL (low byte) and CCPRxH (high byte), $1 \leq x \leq 2$. The CCPxCON register controls the operation of CCPx. All are readable and writable.

Table 15-1 shows the timer resources of the CCP module modes.

TABLE 15-1: CCP MODE – TIMER RESOURCE

CCP Mode	Timer Resource
Capture	Timer1 or Timer3
Compare	Timer1 or Timer3
PWM	Timer2

15.2 Capture Mode

In Capture mode, CCPRxH:CCPRxL captures the 16-bit value of the TMR1 or TMR3 register when an event occurs on pin CCPn. An event is defined as:

- every falling edge
- every rising edge
- every 4th rising edge
- every 16th rising edge

An event is selected by control bits CCPxM3:CCPxM0 (CCPxCON<3:0>). When a capture is made, the interrupt request flag bit, CCPxIF (PIR registers), is set. It must be cleared in software. If another capture occurs before the value in register CCPRx is read, the old captured value will be lost.

15.2.1 CCP PIN CONFIGURATION

In Capture mode, the CCPx pin should be configured as an input by setting the appropriate TRIS bit.

Note: If the CCPx is configured as an output, a write to the port can cause a capture condition.

15.2.2 TIMER1/TIMER3 MODE SELECTION

The timer used with each CCP module is selected in the T3CCP2:T3CCP1 bits of the T3CON register. The timers used with the capture feature (either Timer1 or Timer3) must be running in Timer mode or Synchronized Counter mode. In Asynchronous Counter mode, the capture operation may not work.

TABLE 15-2: INTERACTION OF CCP MODULES

CCP1 Mode	CCP2 Mode	Interaction
Capture	Capture	TMR1 or TMR3 time base. Time base can be different for each CCP.
Capture	Compare	The compare could be configured for the special event trigger which clears either TMR1 or TMR3 depending upon which time base is used.
Compare	Compare	The compare(s) could be configured for the special event trigger which clears TMR1 or TMR3 depending upon which time base is used.
PWM	PWM	The PWMs will have the same frequency and update rate (TMR2 interrupt).
PWM	Capture	None.
PWM	Compare	None.

PIC18F6585/8585/6680/8680

TABLE 18-3: BAUD RATES FOR ASYNCHRONOUS MODES (CONTINUED)

BAUD RATE (K)	SYNC = 0, BRGH = 0, BRG16 = 1											
	Fosc = 40.000 MHz			Fosc = 20.000 MHz			Fosc = 10.000 MHz			Fosc = 8.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	0.300	0.00	8332	0.300	0.02	4165	0.300	0.02	2082	300	-0.04	1665
1.2	1.200	0.02	2082	1.200	-0.03	1041	1.200	-0.03	520	1201	-0.16	415
2.4	2.402	0.06	1040	2.399	-0.03	520	2.404	0.16	259	2403	-0.16	207
9.6	9.615	0.16	259	9.615	0.16	129	9.615	0.16	64	9615	-0.16	51
19.2	19.231	0.16	129	19.231	0.16	64	19.531	1.73	31	19230	-0.16	25
57.6	58.140	0.94	42	56.818	-1.36	21	56.818	-1.36	10	55555	3.55	8
115.2	113.636	-1.36	21	113.636	-1.36	10	125.000	8.51	4	—	—	—

BAUD RATE (K)	SYNC = 0, BRGH = 0, BRG16 = 1								
	Fosc = 4.000 MHz			Fosc = 2.000 MHz			Fosc = 1.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	0.300	0.04	832	300	-0.16	415	300	-0.16	207
1.2	1.202	0.16	207	1201	-0.16	103	1201	-0.16	51
2.4	2.404	0.16	103	2403	-0.16	51	2403	-0.16	25
9.6	9.615	0.16	25	9615	-0.16	12	—	—	—
19.2	19.231	0.16	12	—	—	—	—	—	—
57.6	62.500	8.51	3	—	—	—	—	—	—
115.2	125.000	8.51	1	—	—	—	—	—	—

BAUD RATE (K)	SYNC = 0, BRGH = 1, BRG16 = 1 or SYNC = 1, BRG16 = 1											
	Fosc = 40.000 MHz			Fosc = 20.000 MHz			Fosc = 10.000 MHz			Fosc = 8.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	0.300	0.00	33332	0.300	0.00	16665	0.300	0.00	8332	300	-0.01	6665
1.2	1.200	0.00	8332	1.200	0.02	4165	1.200	0.02	2082	1200	-0.04	1665
2.4	2.400	0.02	4165	2.400	0.02	2082	2.402	0.06	1040	2400	-0.04	832
9.6	9.606	0.06	1040	9.596	-0.03	520	9.615	0.16	259	9615	-0.16	207
19.2	19.193	-0.03	520	19.231	0.16	259	19.231	0.16	129	19230	-0.16	103
57.6	57.803	0.35	172	57.471	-0.22	86	58.140	0.94	42	57142	0.79	34
115.2	114.943	-0.22	86	116.279	0.94	42	113.636	-1.36	21	117647	-2.12	16

BAUD RATE (K)	SYNC = 0, BRGH = 1, BRG16 = 1 or SYNC = 1, BRG16 = 1								
	Fosc = 4.000 MHz			Fosc = 2.000 MHz			Fosc = 1.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	0.300	0.01	3332	300	-0.04	1665	300	-0.04	832
1.2	1.200	0.04	832	1201	-0.16	415	1201	-0.16	207
2.4	2.404	0.16	415	2403	-0.16	207	2403	-0.16	103
9.6	9.615	0.16	103	9615	-0.16	51	9615	-0.16	25
19.2	19.231	0.16	51	19230	-0.16	25	19230	-0.16	12
57.6	58.824	2.12	16	55555	3.55	8	—	—	—
115.2	111.111	-3.55	8	—	—	—	—	—	—

PIC18F6585/8585/6680/8680

22.2 Operation

Depending on the power source for the device voltage, the voltage normally decreases relatively slowly. This means that the LVD module does not need to be constantly operating. To decrease the current requirements, the LVD circuitry only needs to be enabled for short periods where the voltage is checked. After doing the check, the LVD module may be disabled.

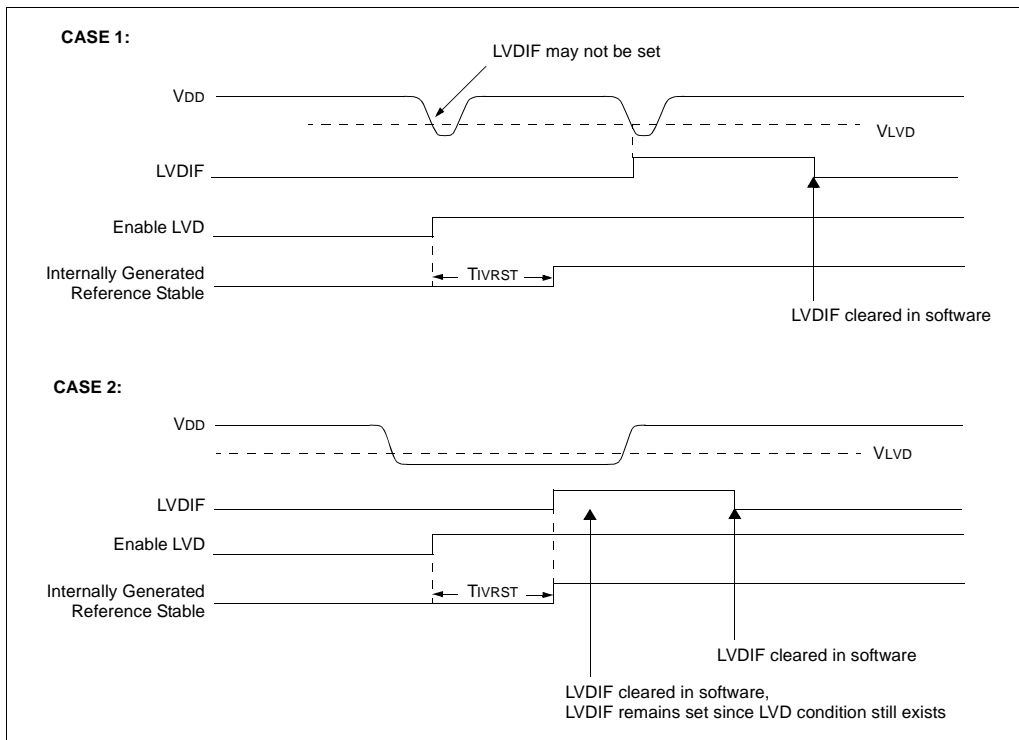
Each time that the LVD module is enabled, the circuitry requires some time to stabilize. After the circuitry has stabilized, all status flags may be cleared. The module will then indicate the proper state of the system.

The following steps are needed to set up the LVD module:

1. Write the value to the LVDL3:LVDL0 bits (LVDCON register) which selects the desired LVD trip point.
2. Ensure that LVD interrupts are disabled (the LVDIE bit is cleared or the GIE bit is cleared).
3. Enable the LVD module (set the LVDEN bit in the LVDCON register).
4. Wait for the LVD module to stabilize (the IRVST bit to become set).
5. Clear the LVD interrupt flag which may have falsely become set until the LVD module has stabilized (clear the LVDIF bit).
6. Enable the LVD interrupt (set the LVDIE and the GIE bits).

Figure 22-4 shows typical waveforms that the LVD module may be used to detect.

FIGURE 22-4: LOW-VOLTAGE DETECT WAVEFORMS



PIC18F6585/8585/6680/8680

REGISTER 23-50: MSEL2: MASK SELECT REGISTER 2⁽¹⁾

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
FIL11_1	FIL11_0	FIL10_1	FIL10_0	FIL9_1	FIL9_0	FIL8_1	FIL8_0
bit 7						bit 0	

bit 7-6 **FIL11_1:FIL11_0:** Filter 11 Select bits 1 and 0

11 = No mask
10 = Filter 15
01 = Acceptance Mask 1
00 = Acceptance Mask 0

bit 5-4 **FIL10_1:FIL10_0:** Filter 10 Select bits 1 and 0

11 = No mask
10 = Filter 15
01 = Acceptance Mask 1
00 = Acceptance Mask 0

bit 3-2 **FIL9_1:FIL9_0:** Filter 9 Select bits 1 and 0

11 = No mask
10 = Filter 15
01 = Acceptance Mask 1
00 = Acceptance Mask 0

bit 1-0 **FIL8_1:FIL8_0:** Filter 8 Select bits 1 and 0

11 = No mask
10 = Filter 15
01 = Acceptance Mask 1
00 = Acceptance Mask 0

Note 1: This register is available in Mode 1 and 2 only.

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

23.15.2 TRANSMIT INTERRUPT

When the transmit interrupt is enabled, an interrupt will be generated when the associated transmit buffer becomes empty and is ready to be loaded with a new message. In Mode 0, there are separate interrupt enable/disable and flag bits for each of the three dedicated transmit buffers. The TXBnIF bit will be set to indicate the source of the interrupt. The interrupt is cleared by the MCU resetting the TXBnIF bit to a '0'. In Mode 1 and 2, all transmit buffers share one interrupt enable/disable and flag bits. In Mode 1 and 2, TXBIE in PIE3 and TXBIF in PIR3 indicate when a transmit buffer has completed transmission of its message. TXBnIF, TXBnIE and TXBnIP in PIR3, PIE3 and IPR3, respectively, are not used in Mode 1 and 2. Individual transmit buffer interrupts can be enabled or disabled by setting or clearing TXBIE and BnIE register bits. When a shared interrupt occurs, user firmware must poll the TXREQ bit of all transmit buffers to detect the source of interrupt.

23.15.3 RECEIVE INTERRUPT

When the receive interrupt is enabled, an interrupt will be generated when a message has been successfully received and loaded into the associated receive buffer. This interrupt is activated immediately after receiving the End Of Frame (EOF) field.

In Mode 0, the RXBnIF bit is set to indicate the source of the interrupt. The interrupt is cleared by the MCU resetting the RXBnIF bit to a '0'.

In Mode 1 and 2, all receive buffers share one interrupt. Individual receive buffer interrupts can be controlled by the RXBnIE and BIE registers. In Mode 1, when a shared receive interrupt occurs, user firmware must poll the RXFUL bit of each receive buffer to detect the source of interrupt. In Mode 2, a receive interrupt indicates that the new message is loaded into FIFO. FIFO can be read by using FIFO pointer bits, FP.

In Mode 2, the FIFOWMIF bit indicates if the FIFO high watermark is reached. The FIFO high watermark is defined by the FIFOWM bit in the ECANCON register.

23.15.4 MESSAGE ERROR INTERRUPT

When an error occurs during transmission or reception of a message, the message error flag, IRXIF, will be set and if the IRXIE bit is set, an interrupt will be generated. This is intended to be used to facilitate baud rate determination when used in conjunction with Listen Only mode.

23.15.5 BUS ACTIVITY WAKE-UP INTERRUPT

When the PIC18F6585/8585/6680/8680 devices are in Sleep mode and the bus activity wake-up interrupt is enabled, an interrupt will be generated and the WAKIF bit will be set when activity is detected on the CAN bus. This interrupt causes the PIC18F6585/8585/6680/8680 devices to exit Sleep mode. The interrupt is reset by the MCU, clearing the WAKIF bit.

23.15.6 ERROR INTERRUPT

When the error interrupt is enabled, an interrupt is generated if an overflow condition occurs or if the error state of the transmitter or receiver has changed. The error flags in COMSTAT will indicate one of the following conditions.

23.15.6.1 Receiver Overflow

An overflow condition occurs when the MAB has assembled a valid received message (the message meets the criteria of the acceptance filters) and the receive buffer associated with the filter is not available for loading of a new message. The associated COMSTAT.RXnOVFL bit will be set to indicate the overflow condition. This bit must be cleared by the MCU.

23.15.6.2 Receiver Warning

The receive error counter has reached the MCU warning limit of 96.

23.15.6.3 Transmitter Warning

The transmit error counter has reached the MCU warning limit of 96.

23.15.6.4 Receiver Bus Passive

The receive error counter has exceeded the error-passive limit of 127 and the device has gone to error-passive state.

23.15.6.5 Transmitter Bus Passive

The transmit error counter has exceeded the error-passive limit of 127 and the device has gone to error-passive state.

23.15.6.6 Bus-Off

The transmit error counter has exceeded 255 and the device has gone to bus-off state.

23.15.6.7 Interrupt Acknowledge

Interrupts are directly associated with one or more status flags in the PIR register. Interrupts are pending as long as one of the flags is set. Once an interrupt flag is set by the device, the flag can not be reset by the microcontroller until the interrupt condition is removed.

PIC18F6585/8585/6680/8680

BZ Branch if Zero

Syntax: [*label*] BZ n

Operands: $-128 \leq n \leq 127$

Operation: if Zero bit is '1'
 $(PC) + 2 + 2n \rightarrow PC$

Status Affected: None

Encoding:

1110	0000	nnnn	nnnn
------	------	------	------

Description: If the Zero bit is '1', then the program will branch.
 The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be $PC+2+2n$. This instruction is then a two-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:

If Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	Write to PC
No operation	No operation	No operation	No operation

If No Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	No operation

Example: HERE BZ Jump

Before Instruction

PC = address (HERE)

After Instruction

If Zero = 1;
 PC = address (Jump)
 If Zero = 0;
 PC = address (HERE+2)

CALL Subroutine Call

Syntax: [*label*] CALL k [,s]

Operands: $0 \leq k \leq 1048575$
 $s \in [0,1]$

Operation: $(PC) + 4 \rightarrow TOS$,
 $k \rightarrow PC<20:1>$,
 if $s = 1$
 $(W) \rightarrow WS$,
 $(STATUS) \rightarrow STATUSS$,
 $(BSR) \rightarrow BSRs$

Status Affected: None

Encoding:

1110	110s	k ₇ kkk	kkkk ₀
1111	k ₁₉ kkk	kkkk	kkkk ₈

Description: Subroutine call of entire 2-Mbyte memory range. First, return address $(PC+4)$ is pushed onto the return stack. If 's' = 1, the W, Status and BSR registers are also pushed into their respective shadow registers, WS, STATUSS and BSRs. If 's' = 0, no update occurs (default). Then, the 20-bit value 'k' is loaded into $PC<20:1>$. CALL is a two-cycle instruction.

Words: 2

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'<7:0>, Push PC to stack	Push PC to stack	Read literal 'k'<19:8>, Write to PC
No operation	No operation	No operation	No operation

Example: HERE CALL THERE, 1

Before Instruction

PC = address (HERE)

After Instruction

PC = address (THERE)
 TOS = address (HERE + 4)
 WS = W
 BSRs = BSR
 STATUSS = STATUS

PIC18F6585/8585/6680/8680

GOTO Unconditional Branch

Syntax: [*label*] GOTO *k*

Operands: $0 \leq k \leq 1048575$

Operation: $k \rightarrow PC<20:1>$

Status Affected: None

Encoding:

1st word ($k<7:0>$)

1110

1111

k_7kkk

$kkkk_0$

2nd word ($k<19:8>$)

1111

$k_{19}kkk$

$kkkk$

$kkkk_8$

Description: GOTO allows an unconditional branch anywhere within entire 2-Mbyte memory range. The 20-bit value 'k' is loaded into PC<20:1>. GOTO is always a two-cycle instruction.

Words: 2

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'<7:0>.	No operation	Read literal 'k'<19:8>. Write to PC
No operation	No operation	No operation	No operation

Example: GOTO THERE

After Instruction

PC = Address (THERE)

INCF Increment f

Syntax: [*label*] INCF f [,d [,a]]

Operands: $0 \leq f \leq 255$

$d \in [0,1]$

$a \in [0,1]$

Operation: $(f) + 1 \rightarrow \text{dest}$

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

Encoding:

0010

10da

ffff

ffff

Description: The contents of register 'f' are incremented. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default). If 'a' is '0', the Access Bank will be selected, overriding the BSR value. If 'a' = 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: INCF CNT, 1, 0

Before Instruction

CNT = 0xFF
Z = 0
C = ?
DC = ?

After Instruction

CNT = 0x00
Z = 1
C = 1
DC = 1

PIC18F6585/8585/6680/8680

RRNCF Rotate Right f (no carry)

Syntax: [*label*] RRNCF f [,d [,a]]

Operands: $0 \leq f \leq 255$
 $d \in [0,1]$
 $a \in [0,1]$

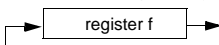
Operation: $(f \ll n) \rightarrow \text{dest} \ll n-1$,
 $(f \ll 0) \rightarrow \text{dest} \ll 7$

Status Affected: N, Z

Encoding:

0100	00da	ffff	ffff
------	------	------	------

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

Before Instruction

REG = 1101 0111

After Instruction

REG = 1110 1011

Example 2: RRNCF REG, 0, 0

Before Instruction

W = ?

REG = 1101 0111

After Instruction

W = 1110 1011

REG = 1101 0111

SETF Set f

Syntax: [*label*] SETF f [,a]

Operands: $0 \leq f \leq 255$
 $a \in [0,1]$

Operation: FFh \rightarrow f

Status Affected: None

Encoding:

0110	100a	ffff	ffff
------	------	------	------

Description: The contents of the specified register are set to FFh. 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 register 'f'

Example: SETF REG, 1

Before Instruction

REG = 0x5A

After Instruction

REG = 0xFF

PIC18F6585/8585/6680/8680

NOTES:

APPENDIX C: CONVERSION CONSIDERATIONS

This appendix discusses the considerations for converting from previous versions of a device to the ones listed in this data sheet. Typically, these changes are due to the differences in the process technology used. An example of this type of conversion is from a PIC17C756 to a PIC18F8720.

Not Applicable

APPENDIX D: MIGRATION FROM MID-RANGE TO ENHANCED DEVICES

A detailed discussion of the differences between the mid-range MCU devices (i.e., PIC16CXXX) and the enhanced devices (i.e., PIC18FXXX) is provided in AN716, "Migrating Designs from PIC16C74A/74B to PIC18C442." The changes discussed, while device specific, are generally applicable to all mid-range to enhanced device migrations.

This Application Note is available as Literature Number DS00716.

PIC18F6585/8585/6680/8680

F

Firmware Instructions.....	365
Flash Program Memory	83
Associated Registers	92
Control Registers	84
Erase Sequence	88
Erasing.....	88
Operation During Code	
Protection.....	92
Reading.....	87
Table Pointer	
Boundaries Based on Operation.....	86
Table Pointer Boundaries	86
Table Reads and Table Writes	83
Write Sequence	90
Writing to.....	89
Protection Against Spurious	
Writes.....	92
Unexpected Termination.....	92
Write Verify	92

G

General Call Address Support	212
GOTO	386

H

Hardware Multiplier	107
Introduction	107
Operation	107
Performance Comparison	
(table).....	107

I

I/O Ports.....	125
I ² C Bus Data Requirements	
(Slave Mode).....	442
I ² C Bus Start/Stop Bits Requirements	
(Slave Mode).....	441
I ² C Mode	
General Call Address Support	212
Master Mode	
Operation	214
Read/Write Bit Information	
(R/W Bit)	202, 203
Serial Clock (RC3/SCK/SCL).....	203
ID Locations	345, 362
INCF.....	386
INCFSZ	387
In-Circuit Debugger.....	362
Resources (table).....	362
In-Circuit Serial Programming	
(ICSP)	345, 362
Indirect Addressing	
INDF and FSR Registers	79
Operation	79
Indirect File Operand	59
INFSNZ	387
Instruction Flow/Pipelining	57
Instruction Format	367

Instruction Set.....	365
ADDLW.....	371
ADDWF	371
ADDWFC.....	372
ANDLW.....	372
ANDWF	373
BC.....	373
BCF	374
BN.....	374
BNC	375
BNN.....	375
BNOV	376
BNZ	376
BOV	379
BRA	377
BSF.....	377
BTFSZ.....	378
BTFSS	378
BTG	379
BZ.....	380
CALL.....	380
CLRF	381
CLRWDT	381
COMF	382
CPFSEQ.....	382
CPFSGT	383
CPFSLT	383
DAW	384
DCFSNZ	385
DECF.....	384
DECFSZ	385
GOTO	386
INCF	386
INCFSZ	387
INFSNZ	387
IORLW.....	388
IORWF.....	388
LFSR	389
MOVF	389
MOVFF	390
MOVLB	390
MOVLW	391
MOVWF	391
MULLW.....	392
MULWF	392
NEGF.....	393
NOP.....	393
POP	394
PUSH.....	394
RCALL.....	395
RESET.....	395
RETFIE	396
RETLW	396
RETURN.....	397
RLCF	397
RLNCF.....	398
RRCF	398
RRNCF	399
SETF	399