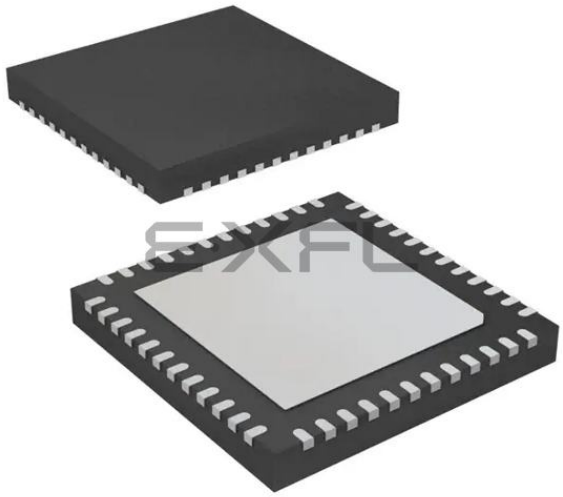


Welcome to E-XFL.COM

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	64MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	44
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 43x12b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-UFQFN Exposed Pad
Supplier Device Package	48-UQFN (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f55k42t-i-mv

4.7.2 DIRECT ADDRESSING

Direct addressing 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 4.5.2 "General Purpose Register File"**) or a location in the Access Bank (**Section 4.5.4 "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 4.5.1 "Bank Select Register (BSR)"**) are used with the address to determine the complete 14-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 MOVFFL, include the entire 14-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.

4.7.3 INDIRECT ADDRESSING

Indirect addressing 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 which are to be read or written. Since the FSRs are themselves located in RAM as Special File 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 4-6.

EXAMPLE 4-6: HOW TO CLEAR RAM (BANK 1) USING INDIRECT ADDRESSING

```
        LFSR   FSR0, 100h ;
NEXT    CLRF   POSTINC0   ; Clear INDF
                          ; register then
                          ; inc pointer
        BTFSS  FSR0H, 1   ; All done with
                          ; Bank1?
        BRA    NEXT       ; NO, clear next
CONTINUE                  ; YES, continue
```

4.7.3.1 FSR Registers and the INDF Operand

At the core of indirect addressing are three sets of registers: FSR0, FSR1 and FSR2. Each represents a pair of 8-bit registers, FSRnH and FSRnL. Each FSR pair holds a 14-bit value, therefore, the two upper bits of the FSRnH register are not used. The 14-bit FSR value can address the entire range of the data memory in a linear fashion. The FSR register pairs, then, serve as pointers to data memory locations.

Indirect addressing is accomplished with a set of Indirect File Operands, INDF0 through INDF2. These can be thought of as "virtual" registers; they are mapped in the SFR space but are not physically implemented. Reading or writing to a particular INDF register actually accesses the data addressed by its corresponding FSR register pair. A read from INDF1, for example, reads the data at the address indicated by FSR1H:FSR1L. Instructions that use the INDF registers as operands actually use the contents of their corresponding FSR as a pointer to the instruction's target. The INDF operand is just a convenient way of using the pointer.

Because indirect addressing uses a full 14-bit address, data RAM banking is not necessary. Thus, the current contents of the BSR and the Access RAM bit have no effect on determining the target address.

6.1 Power-on Reset (POR)

The POR circuit holds the device in Reset until VDD has reached an acceptable level for minimum operation. Slow rising VDD, fast operating speeds or analog performance may require greater than minimum VDD. The PWRT, BOR or MCLR features can be used to extend the start-up period until all device operation conditions have been met.

6.2 Brown-out Reset (BOR)

The BOR circuit holds the device in Reset when VDD reaches a selectable minimum level. Between the POR and BOR, complete voltage range coverage for execution protection can be implemented.

The Brown-out Reset module has four operating modes controlled by the BOREN<1:0> bits in Configuration Words. The four operating modes are:

- BOR is always on
- BOR is off when in Sleep
- BOR is controlled by software
- BOR is always off

Refer to Table 6-1 for more information.

The Brown-out Reset voltage level is selectable by configuring the BORV<1:0> bits in Configuration Words.

A VDD noise rejection filter prevents the BOR from triggering on small events. If VDD falls below VBOR for a duration greater than parameter TBORDC, the device will reset. See Table 44-13 for more information.

6.2.1 BOR IS ALWAYS ON

When the BOREN bits of Configuration Words are programmed to '11', the BOR is always on. The device start-up will be delayed until the BOR is ready and VDD is higher than the BOR threshold.

BOR protection is active during Sleep. The BOR does not delay wake-up from Sleep.

6.2.2 BOR IS OFF IN SLEEP

When the BOREN bits of Configuration Words are programmed to '10', the BOR is on, except in Sleep. The device start-up will be delayed until the BOR is ready and VDD is higher than the BOR threshold.

BOR protection is not active during Sleep. The device wake-up will be delayed until the BOR is ready.

6.2.3 BOR CONTROLLED BY SOFTWARE

When the BOREN bits of Configuration Words are programmed to '01', the BOR is controlled by the SBOREN bit of the BORCON register. The device start-up is not delayed by the BOR ready condition or the VDD level.

BOR protection begins as soon as the BOR circuit is ready. The status of the BOR circuit is reflected in the BORRDY bit of the BORCON register.

BOR protection is unchanged by Sleep.

6.2.4 BOR AND BULK ERASE

BOR is forced ON during PFM Bulk Erase operations to make sure that a safe erase voltage is maintained for a successful erase cycle.

During Bulk Erase, the BOR is enabled at 2.45V for F and LF devices, even if it is configured to some other value. If VDD falls, the erase cycle will be aborted, but the device will not be reset.

Example 12-3 shows the sequence to do a 16 x 16 unsigned multiplication. Equation 12-1 shows the algorithm that is used. The 32-bit result is stored in four registers (RES<3:0>).

EQUATION 12-1: 16 x 16 UNSIGNED MULTIPLICATION ALGORITHM

$$
\begin{aligned}
RES3:RES0 &= ARG1H:ARG1L \bullet ARG2H:ARG2L \\
&= (ARG1H \bullet ARG2H \bullet 2^{16}) + \\
&\quad (ARG1H \bullet ARG2L \bullet 2^{8}) + \\
&\quad (ARG1L \bullet ARG2H \bullet 2^{8}) + \\
&\quad (ARG1L \bullet ARG2L)
\end{aligned}
$$

EXAMPLE 12-3: 16 x 16 UNSIGNED MULTIPLY ROUTINE

```
    MOVF    ARG1L, W
    MULWF   ARG2L           ; ARG1L * ARG2L->
                            ; PRODH:PRODL
    MOVFF   PRODH, RES1     ;
    MOVFF   PRODL, RES0     ;
;
    MOVF    ARG1H, W
    MULWF   ARG2H           ; ARG1H * ARG2H->
                            ; PRODH:PRODL
    MOVFF   PRODH, RES3     ;
    MOVFF   PRODL, RES2     ;
;
    MOVF    ARG1L, W
    MULWF   ARG2H           ; ARG1L * ARG2H->
                            ; PRODH:PRODL
    MOVF    PRODL, W        ;
    ADDWF   RES1, F         ; Add cross
    MOVF    PRODH, W        ; products
    ADDWFC  RES2, F         ;
    CLRF    WREG            ;
    ADDWFC  RES3, F         ;
;
    MOVF    ARG1H, W        ;
    MULWF   ARG2L           ; ARG1H * ARG2L->
                            ; PRODH:PRODL
    MOVF    PRODL, W        ;
    ADDWF   RES1, F         ; Add cross
    MOVF    PRODH, W        ; products
    ADDWFC  RES2, F         ;
    CLRF    WREG            ;
    ADDWFC  RES3, F         ;
```

Example 12-4 shows the sequence to do a 16 x 16 signed multiply. Equation 12-2 shows the algorithm used. The 32-bit result is stored in four registers (RES<3:0>). To account for the sign bits of the arguments, the MSb for each argument pair is tested and the appropriate subtractions are done.

EQUATION 12-2: 16 x 16 SIGNED MULTIPLICATION ALGORITHM

$$
\begin{aligned}
RES3:RES0 &= ARG1H:ARG1L \bullet ARG2H:ARG2L \\
&= (ARG1H \bullet ARG2H \bullet 2^{16}) + \\
&\quad (ARG1H \bullet ARG2L \bullet 2^{8}) + \\
&\quad (ARG1L \bullet ARG2H \bullet 2^{8}) + \\
&\quad (ARG1L \bullet ARG2L) + \\
&\quad (-1 \bullet ARG2H\langle 7\rangle \bullet ARG1H:ARG1L \bullet 2^{16}) + \\
&\quad (-1 \bullet ARG1H\langle 7\rangle \bullet ARG2H:ARG2L \bullet 2^{16})
\end{aligned}
$$

EXAMPLE 12-4: 16 x 16 SIGNED MULTIPLY ROUTINE

```
    MOVF    ARG1L, W
    MULWF   ARG2L           ; ARG1L * ARG2L ->
                            ; PRODH:PRODL
    MOVFF   PRODH, RES1     ;
    MOVFF   PRODL, RES0     ;
;
    MOVF    ARG1H, W
    MULWF   ARG2H           ; ARG1H * ARG2H ->
                            ; PRODH:PRODL
    MOVFF   PRODH, RES3     ;
    MOVFF   PRODL, RES2     ;
;
    MOVF    ARG1L, W
    MULWF   ARG2H           ; ARG1L * ARG2H ->
                            ; PRODH:PRODL
    MOVF    PRODL, W        ;
    ADDWF   RES1, F         ; Add cross
    MOVF    PRODH, W        ; products
    ADDWFC  RES2, F         ;
    CLRF    WREG            ;
    ADDWFC  RES3, F         ;
;
    MOVF    ARG1H, W        ;
    MULWF   ARG2L           ; ARG1H * ARG2L ->
                            ; PRODH:PRODL
    MOVF    PRODL, W        ;
    ADDWF   RES1, F         ; Add cross
    MOVF    PRODH, W        ; products
    ADDWFC  RES2, F         ;
    CLRF    WREG            ;
    ADDWFC  RES3, F         ;
;
    BTFSS   ARG2H, 7        ; ARG2H:ARG2L neg?
    BRA     SIGN_ARG1       ; no, check ARG1
    MOVF    ARG1L, W        ;
    SUBWF   RES2            ;
    MOVF    ARG1H, W        ;
    SUBWFB  RES3
;
SIGN_ARG1
    BTFSS   ARG1H, 7        ; ARG1H:ARG1L neg?
    BRA     CONT_CODE       ; no, done
    MOVF    ARG2L, W        ;
    SUBWF   RES2            ;
    MOVF    ARG2H, W        ;
    SUBWFB  RES3
;
CONT_CODE
    :
```

EXAMPLE 13-3: ERASING A PROGRAM FLASH MEMORY BLOCK

```
; This sample row erase routine assumes the following:
; 1. A valid address within the erase row is loaded in variables TBLPTR register

            CLRF        NVMCON1             ; Setup PFM Access
            MOVLW       CODE_ADDR_UPPER     ; load TBLPTR with the base
            MOVWF       TBLPTRU             ; address of the memory block
            MOVLW       CODE_ADDR_HIGH
            MOVWF       TBLPTRH
            MOVLW       CODE_ADDR_LOW
            MOVWF       TBLPTRL
ERASE_BLOCK
            BCF         NVMCON1, REG0       ; point to Program Flash Memory
            BSF         NVMCON1, REG1       ; access Program Flash Memory
            BSF         NVMCON1, WREN       ; enable write to memory
            BSF         NVMCON1, FREE       ; enable block Erase operation
            BCF         INTCON0, GIE        ; disable interrupts
            MOVLW       55h
Required    MOVWF       NVMCON2             ; write 55h
Sequence    MOVLW       AAh
            MOVWF       NVMCON2             ; write AAh
            BSF         NVMCON1, WR         ; start erase (CPU stalls)
            BSF         INTCON0, GIE        ; re-enable interrupts
```

TABLE 14-2: SUMMARY OF REGISTERS ASSOCIATED WITH CRC

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CRCACCH	ACC<15:8>								219
CRCACCL	ACC<7:0>								220
CRCCON0	EN	GO	BUSY	ACCM	—	—	SHIFTM	FULL	218
CRCCON1	DLEN<3:0>				PLEN<3:0>				218
CRCDATH	DATA<15:8>								219
CRCDATL	DATA<7:0>								219
CRCSHIFTH	SHIFT<15:8>								220
CRCSHIFTL	SHIFT<7:0>								220
CRCXORH	X<15:8>								221
CRCXORL	X<7:1>							—	221
SCANCON0	EN	TRIGEN	SGO	—	—	MREG	BURSTMD	BUSY	222
SCANHADRU	—	—	HADR<21:16>						224
SCANHADRH	HADR<15:8>								225
SCANHADRL	HADR<7:0>								225
SCANLADRU	—	—	LADR<21:16>						223
SCANLADRH	LADR<15:8>								223
SCANLADRL	LADR<7:0>								224
SCANTRIG	—	—	—	—	TSEL<3:0>				226

Legend: — = unimplemented location, read as '0'. Shaded cells are not used for the CRC module.

REGISTER 15-18: DMAxDSZL: DMAx DESTINATION SIZE LOW REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
DSZ<7:0>							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n/n = Value at POR and BOR/Value at all other Resets	1 = bit is set	0 = bit is cleared	x = bit is unknown u = bit is unchanged

bit 7-0 **DSZ<7:0>:** Destination Message Size bits

REGISTER 15-19: DMAxDSZH: DMAx DESTINATION SIZE HIGH REGISTER

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	DSZ<11:8>			
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n/n = Value at POR and BOR/Value at all other Resets	1 = bit is set	0 = bit is cleared	x = bit is unknown u = bit is unchanged

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

bit 3-0 **DSZ<11:8>:** Destination Message Size bits

REGISTER 15-20: DMAxDCNTL: DMAx DESTINATION COUNT LOW REGISTER

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
DCNT<7:0>							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n/n = Value at POR and BOR/Value at all other Resets	1 = bit is set	0 = bit is cleared	x = bit is unknown u = bit is unchanged

bit 7-0 **DCNT<7:0>:** Current Destination Byte Count

TABLE 16-11: SUMMARY OF REGISTERS ASSOCIATED WITH I/O (CONTINUED)

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INLVLC	INLVLC7	INLVLC6	INLVLC5	INLVLC4(5)	INLVLC3(5)	INLVLC2	INLVLC1	INLVLC0	270
INLVLD(6)	INLVLD7	INLVLD6	INLVLD5	INLVLD4	INLVLD3	INLVLD2	INLVLD1(5)	INLVLD0(5)	270
INLVLF(7)	INLVLF7	INLVLF6	INLVLF5	INLVLF4	INLVLF3	INLVLF2	INLVLF1	INLVLF0	270
INLVLE	—	—	—	—	INLVLE3	—	—	—	270
RB1I2C	—	SLEW	PU<1:0>		—	—	TH<1:0>		271
RB2I2C	—	SLEW	PU<1:0>		—	—	TH<1:0>		271
RC3I2C	—	SLEW	PU<1:0>		—	—	TH<1:0>		271
RC4I2C	—	SLEW	PU<1:0>		—	—	TH<1:0>		271
RD0I2C(6)	—	SLEW	PU<1:0>		—	—	TH<1:0>		271
RD1I2C(6)	—	SLEW	PU<1:0>		—	—	TH<1:0>		271

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by I/O Ports.

Note 1: Bits RB6 and RB7 read '1' while in Debug mode.
2: Bit PORTE3 is read-only, and will read '1' when MCLRE = 1 (Master Clear enabled).
3: Bits RB6 and RB7 read '1' while in Debug mode.
4: If MCLRE = 1, the weak pull-up in RE3 is always enabled; bit WPUE3 is not affected.
5: Any peripheral using the I^2C pins read the I^2C ST inputs when enabled via RxyI2C.
6: Unimplemented in PIC18(L)F26/27K42.
7: Unimplemented in PIC18(L)F26/27/45/46/47K42 parts.

17.8 Register Definitions: PPS Input Selection

REGISTER 17-1: xxxPPS: PERIPHERAL xxx INPUT SELECTION

U-0	U-0	R/W-m/u(1,3)	R/W-m/u(1)	R/W-m/u(1)	R/W-m/u(1)	R/W-m/u(1)	R/W-m/u(1)
—	—	xxxPPS<5:0>					
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	-n/n = Value at POR and BOR/Value at all other Resets
u = Bit is unchanged	x = Bit is unknown	q = value depends on peripheral
'1' = Bit is set	U = Unimplemented bit, read as '0'	m = value depends on default location for that input
'0' = Bit is cleared		

bit 7-6 **Unimplemented:** Read as '0'

bit 5-3 **xxxPPS<5:3>:** Peripheral xxx Input PORTx Pin Selection bits

See Table 17-1 for the list of available ports and default pin locations.

101 = PORTF(2)
100 = PORTE(3)
011 = PORTD(3)
010 = PORTC
001 = PORTB
000 = PORTA

bit 2-0 **xxxPPS<2:0>:** Peripheral xxx Input PORTx Pin Selection bits

111 = Peripheral input is from PORTx Pin 7 (Rx7)
110 = Peripheral input is from PORTx Pin 6 (Rx6)
101 = Peripheral input is from PORTx Pin 5 (Rx5)
100 = Peripheral input is from PORTx Pin 4 (Rx4)
011 = Peripheral input is from PORTx Pin 3 (Rx3)
010 = Peripheral input is from PORTx Pin 2 (Rx2)
001 = Peripheral input is from PORTx Pin 1 (Rx1)
000 = Peripheral input is from PORTx Pin 0 (Rx0)

Note 1: The Reset value 'm' of this register is determined by device default locations for that input.
2: Reserved on PIC18LF26/27/45/46/57K42 parts.
3: Reserved on PIC18LF26/27K42 parts.

REGISTER 19-2: PMD1: PMD CONTROL REGISTER 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
NCO1MD	TMR6MD	TMR5MD	TMR4MD	TMR3MD	TMR2MD	TMR1MD	TMR0MD
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	q = Value depends on condition

bit 7 **NCO1MD:** Disable NCO1 Module bit
1 = NCO1 module disabled
0 = NCO1 module enabled

bit 6 **TMR6MD:** Disable Timer TMR6 bit
1 = TMR6 module disabled
0 = TMR6 module enabled

bit 5 **TMR5MD:** Disable Timer TMR5 bit
1 = TMR5 module disabled
0 = TMR5 module enabled

bit 4 **TMR4MD:** Disable Timer TMR4 bit
1 = TMR4 module disabled
0 = TMR4 module enabled

bit 3 **TMR3MD:** Disable Timer TMR3 bit
1 = TMR3 module disabled
0 = TMR3 module enabled

bit 2 **TMR2MD:** Disable Timer TMR2 bit
1 = TMR2 module disabled
0 = TMR2 module enabled

bit 1 **TMR1MD:** Disable Timer TMR1 bit
1 = TMR1 module disabled
0 = TMR1 module enabled

bit 0 **TMR0MD:** Disable Timer TMR0 bit
1 = TMR0 module disabled
0 = TMR0 module enabled

22.5.6 EDGE-TRIGGERED ONE-SHOT MODE

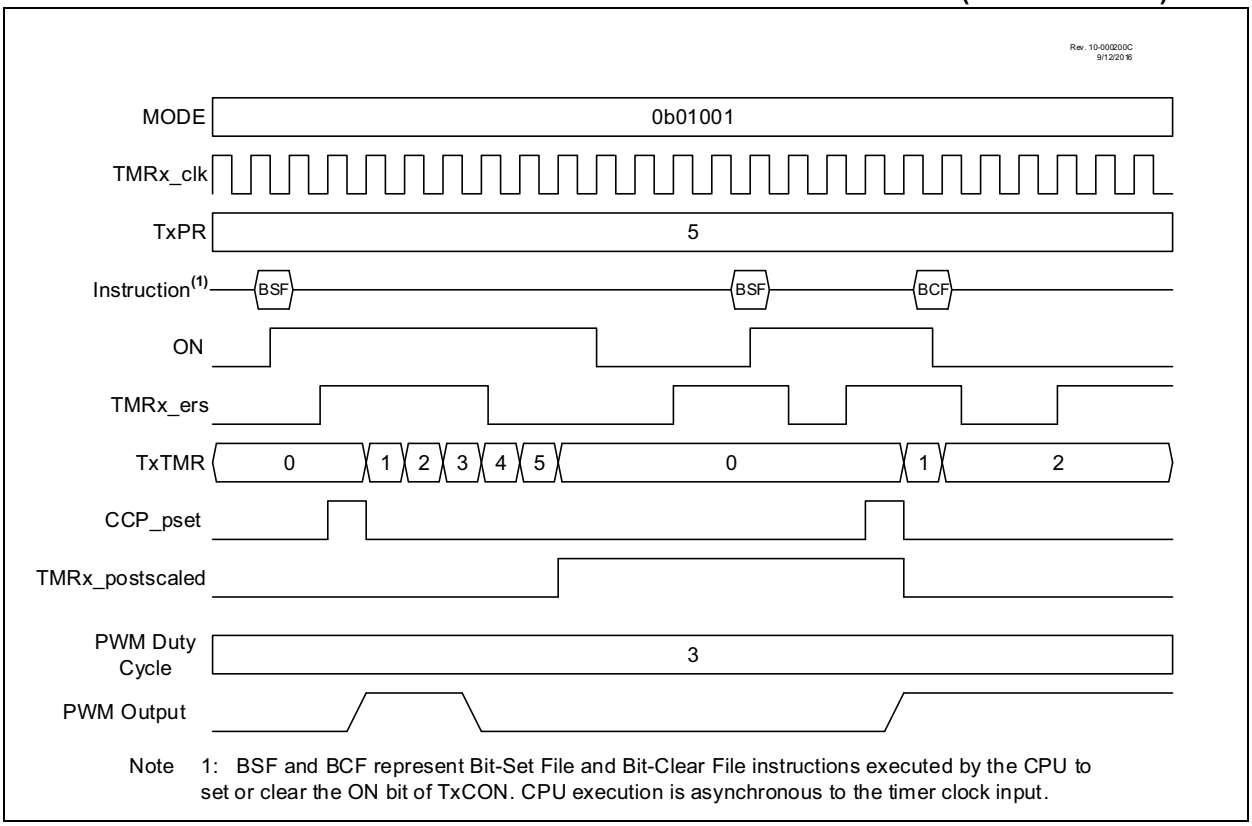
The Edge-Triggered One-Shot modes start the timer on an edge from the external signal input, after the ON bit is set, and clear the ON bit when the timer matches the T2PR period value. The following edges will start the timer:

- Rising edge (MODE<4:0> = 01001)
- Falling edge (MODE<4:0> = 01010)
- Rising or Falling edge (MODE<4:0>='01011')

If the timer is halted by clearing the ON bit then another TMRx_ers edge is required after the ON bit is set to resume counting. Figure 22-9 illustrates operation in the rising edge One-Shot mode.

When Edge-Triggered One-Shot mode is used in conjunction with the CCP then the edge-trigger will activate the PWM drive and the PWM drive will deactivate when the timer matches the CCPRx pulse width value and stay deactivated when the timer halts at the T2PR period count match.

FIGURE 22-9: EDGE TRIGGERED ONE-SHOT MODE TIMING DIAGRAM (MODE = 01001)



Note 1: BSF and BCF represent Bit-Set File and Bit-Clear File instructions executed by the CPU to set or clear the ON bit of TxCON. CPU execution is asynchronous to the timer clock input.

25.7 Interrupts

The SMT can trigger an interrupt under three different conditions:

- PW Acquisition Complete
- PR Acquisition Complete
- Counter Period Match

The interrupts are controlled by the PIR and PIE registers of the device.

25.7.1 PW AND PR ACQUISITION INTERRUPTS

The SMT can trigger interrupts whenever it updates the SMT1CPW and SMT1CPR registers, the circumstances for which are dependent on the SMT mode, and are discussed in each mode's specific section. The SMT1CPW interrupt is controlled by SMT1PWAIF and SMT1PWAIE bits in the respective PIR and PIE registers. The SMT1CPR interrupt is controlled by the SMT1PRAIF and SMT1PRAIE bits, also located in the respective PIR and PIE registers.

In synchronous SMT modes, the interrupt trigger is synchronized to the SMT1CLK. In Asynchronous modes, the interrupt trigger is asynchronous. In either mode, once triggered, the interrupt will be synchronized to the CPU clock.

25.7.2 COUNTER PERIOD MATCH INTERRUPT

As described in **Section 25.1.2 "Period Match interrupt"**, the SMT will also interrupt upon SMT1TMR, matching SMT1PR with its period match limit functionality described in **Section 25.3 "Halt Operation"**. The period match interrupt is controlled by SMT1IF and SMT1IE, located in the respective PIR and PIE registers.

REGISTER 26-2: CWGxCON1: CWG CONTROL REGISTER 1

U-0	U-0	R-x	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	IN	—	POLD	POLC	POLB	POLA
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	q = Value depends on condition

bit 7-6 **Unimplemented:** Read as '0'

bit 5 **IN:** CWG Input Value bit (read-only)

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

bit 3 **POLD:** CWGxD Output Polarity bit
1 = Signal output is inverted polarity
0 = Signal output is normal polarity

bit 2 **POLC:** CWGxC Output Polarity bit
1 = Signal output is inverted polarity
0 = Signal output is normal polarity

bit 1 **POLB:** CWGxB Output Polarity bit
1 = Signal output is inverted polarity
0 = Signal output is normal polarity

bit 0 **POLA:** CWGxA Output Polarity bit
1 = Signal output is inverted polarity
0 = Signal output is normal polarity

TABLE 26-2: SUMMARY OF REGISTERS ASSOCIATED WITH CWG

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CWGxCON0	EN	LD	—	—	—	MODE<2:0>			424
CWGxCON1	—	—	IN	—	POLD	POLC	POLB	POLA	425
CWGxCLK	—	—	—	—	—	—	—	CS	426
CWGxISM	—	—	—	ISM<4:0>					427
CWGxSTR	OVRD	OVRC	OVRB	OVRA	STRD	STRC	STRB	STRA	428
CWGxAS0	SHUTDOWN	REN	LSBD<1:0>		LSAC<1:0>		—	—	429
CWGxAS1	—	AS6E	AS5E	AS4E	AS3E	AS2E	AS1E	AS0E	430
CWGxDBR	—	—	DBR<5:0>						431
CWGxDBF	—	—	DBF<5:0>						431

Legend: – = unimplemented locations read as '0'. Shaded cells are not used by CWG.

REGISTER 30-3: MD1CARH: MODULATION HIGH CARRIER CONTROL REGISTER

U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	CH<4:0>(1)				
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-5 **Unimplemented:** Read as '0'

bit 4-0 **CH<4:0>:** Modulator Carrier High Selection bits(1)
See Table 30-1 for signal list

Note 1: Unused selections provide an input value.

REGISTER 30-4: MD1CARL: MODULATION LOW CARRIER CONTROL REGISTER

U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	CL<4:0>(1)				
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-5 **Unimplemented:** Read as '0'

bit 4-0 **CL<4:0>:** Modulator Carrier Low Input Selection bits(1)
See Table 30-1 for signal list

Note 1: Unused selections provide a zero as the input value.

34.0 FIXED VOLTAGE REFERENCE (FVR)

The Fixed Voltage Reference, or FVR, is a stable voltage reference, independent of VDD, with 1.024V, 2.048V or 4.096V selectable output levels. The output of the FVR can be configured to supply a reference voltage to the following:

- ADC input channel
- ADC positive reference
- Comparator input
- Digital-to-Analog Converter (DAC)

The FVR can be enabled by setting the EN bit of the FVRCON register.

Note:	Fixed Voltage Reference output cannot exceed VDD.

34.1 Independent Gain Amplifiers

The output of the FVR, which is connected to the ADC, Comparators, and DAC, is routed through two independent programmable gain amplifiers. Each amplifier can be programmed for a gain of 1x, 2x or 4x, to produce the three possible voltage levels.

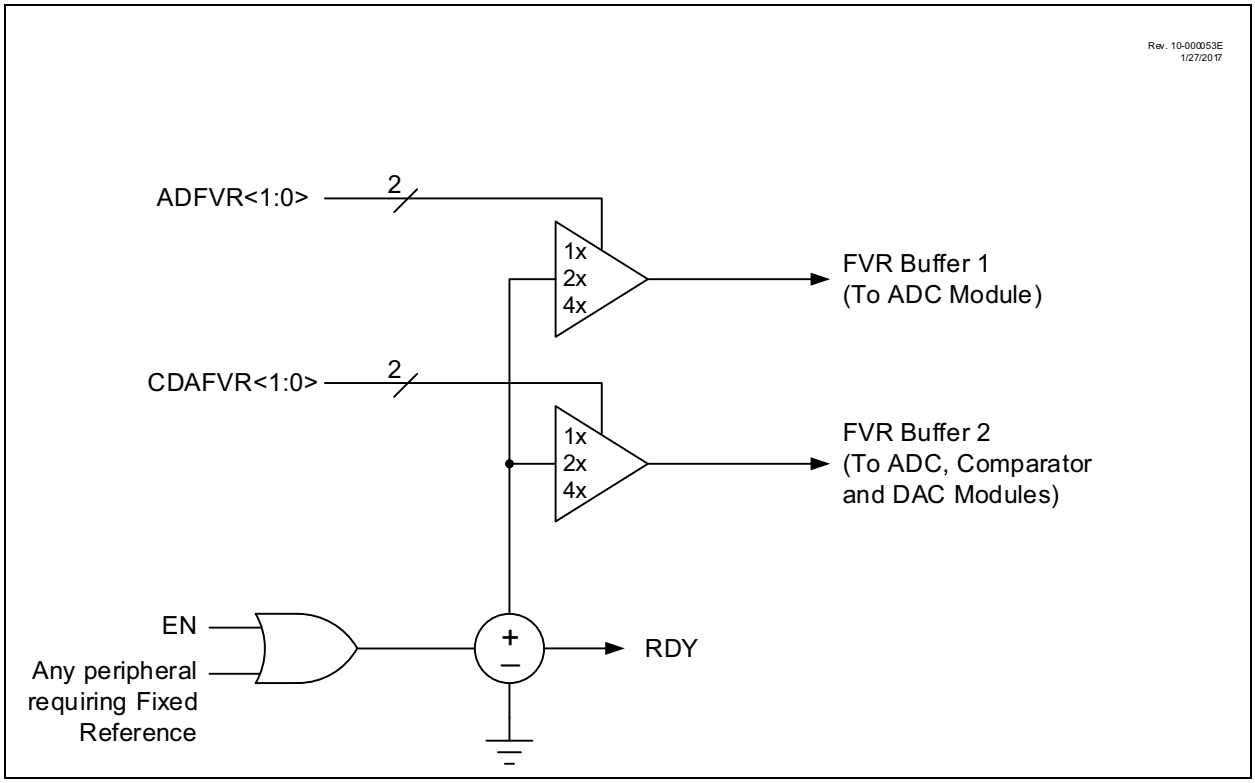
The ADFVR<1:0> bits of the FVRCON register are used to enable and configure the gain amplifier settings for the reference supplied to the ADC module. Reference **Section 36.0 "Analog-to-Digital Converter with Computation (ADC2) Module"** for additional information.

The CDAFVR<1:0> bits of the FVRCON register are used to enable and configure the gain amplifier settings for the reference supplied to the DAC and comparator module. Reference **Section 37.0 "5-Bit Digital-to-Analog Converter (DAC) Module"** and **Section 38.0 "Comparator Module"** for additional information.

34.2 FVR Stabilization Period

When the Fixed Voltage Reference module is enabled, it requires time for the reference and amplifier circuits to stabilize. Once the circuits stabilize and are ready for use, the RDY bit of the FVRCON register will be set.

FIGURE 34-1: VOLTAGE REFERENCE BLOCK DIAGRAM



36.1 ADC Configuration

When configuring and using the ADC the following functions must be considered:

- Port configuration
- Channel selection
- ADC voltage reference selection
- ADC conversion clock source
- Interrupt control
- Result formatting
- Conversion Trigger Selection
- ADC Acquisition Time
- ADC Precharge Time
- Additional Sample and Hold Capacitor
- Single/Double Sample Conversion
- Guard Ring Outputs

36.1.1 PORT CONFIGURATION

The ADC can be used to convert both analog and digital signals. When converting analog signals, the I/O pin should be configured for analog by setting the associated TRIS and ANSEL bits. Refer to **Section 16.0 "I/O Ports"** for more information.

Note:	Analog voltages on any pin that is defined as a digital input may cause the input buffer to conduct excess current.

36.1.2 CHANNEL SELECTION

There are several channel selections available:

- Eight PORTA pins (RA<7:0>)
- Eight PORTB pins (RB<7:0>)
- Eight PORTC pins (RC<7:0>)
- Eight PORTD pins (RD<7:0>, PIC18(L)F45/46/47/55/56/57K42 only)
- Three PORTE pins (RE<2:0>, PIC18(L)F45/46/47/55/56/57K42 only)
- Eight PORTF pins (RD<7:0>, PIC18(L)F55/56/57K42 only)
- Temperature Indicator
- DAC output
- Fixed Voltage Reference (FVR)
- VSS (ground)

The ADPCH register determines which channel is connected to the sample and hold circuit.

When changing channels, a delay is required before starting the next conversion.

Refer to **Section 36.2 "ADC Operation"** for more information.

36.1.3 ADC VOLTAGE REFERENCE

The PREF<1:0> bits of the ADREF register provide control of the positive voltage reference. The positive voltage reference can be:

- VREF+ pin
- VDD
- FVR outputs

The NREF bit of the ADREF register provides control of the negative voltage reference. The negative voltage reference can be:

- VREF- pin
- VSS

See **Section 34.0 "Fixed Voltage Reference (FVR)"** for more details on the Fixed Voltage Reference.

36.1.4 CONVERSION CLOCK

The source of the conversion clock is software selectable via the ADCLK register and the CS bits of the ADCON0 register. If FOSC is selected as the ADC clock, there is a prescaler available to divide the clock so that it meets the ADC clock period specification. The ADC clock source options are the following:

- FOSC/(2*n)(where n is from 1 to 128)
- FRC (dedicated RC oscillator)

The time to complete one bit conversion is defined as TAD. Refer Figure 36-2 for the complete timing details of the ADC conversion.

For correct conversion, the appropriate TAD specification must be met. Refer to Table 44-16 for more information. Table 36-1 gives examples of appropriate ADC clock selections.

Note 1:	Unless using the FRC, any changes in the system clock frequency will change the ADC clock frequency, which may adversely affect the ADC result.
2:	The internal control logic of the ADC runs off of the clock selected by the CS bit of ADCON0. What this can mean is when the CS bit of ADCON0 is set to '1' (ADC runs on FRC), there may be unexpected delays in operation when setting ADC control bits.

38.0 COMPARATOR MODULE

Note: The PIC18(L)F26/27/45/46/47/55/56/57K42 devices have two comparators. Therefore, all information in this section refers to both C1 and C2.

Comparators are used to interface analog circuits to a digital circuit by comparing two analog voltages and providing a digital indication of their relative magnitudes. Comparators are very useful mixed signal building blocks because they provide analog functionality independent of program execution.

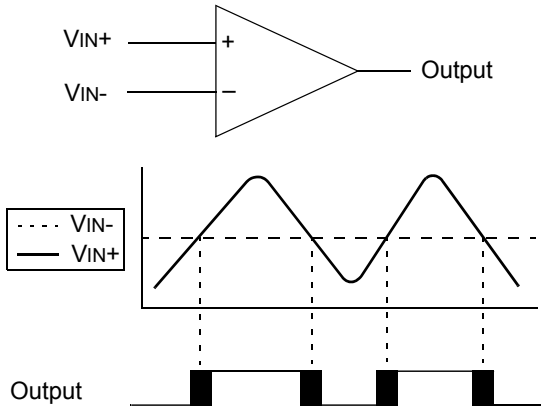
The analog comparator module includes the following features:

- Programmable input selection
- Programmable output polarity
- Rising/falling output edge interrupts

38.1 Comparator Overview

A single comparator is shown in Figure 38-1 along with the relationship between the analog input levels and the digital output. When the analog voltage at VIN+ is less than the analog voltage at VIN-, the output of the comparator is a digital low level. When the analog voltage at VIN+ is greater than the analog voltage at VIN-, the output of the comparator is a digital high level.

FIGURE 38-1: SINGLE COMPARATOR



Note: The black areas of the output of the comparator represents the uncertainty due to input offsets and response time.

REGISTER 38-4: CMxPCH: COMPARATOR x NON-INVERTING CHANNEL SELECT REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	—	PCH<2:0>		
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 7-3 **Unimplemented:** Read as '0'

bit 2-0 **PCH<2:0>:** Comparator Non-Inverting Input Channel Select bits

111 = Vss
110 = FVR_Buffer2
101 = DAC_Output
100 = PCH not connected
011 = PCH not connected
010 = PCH not connected
001 = CxIN1+
000 = CxIN0+

REGISTER 38-5: CMOUT: COMPARATOR OUTPUT REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R-0/0	R-0/0
—	—	—	—	—	—	C2OUT	C1OUT
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 7-2 **Unimplemented:** Read as '0'

bit 1 **C2OUT:** Mirror copy of C2OUT bit

bit 0 **C1OUT:** Mirror copy of C1OUT bit

TABLE 38-3: SUMMARY OF REGISTERS ASSOCIATED WITH COMPARATOR MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
CMxCON0	EN	OUT	—	POL	—	—	HYS	SYNC	648
CMxCON1	—	—	—	—	—	—	INTP	INTN	649
CMxNCH	—	—	—	—	—	NCH<2:0>			649
CMxPCH	—	—	—	—	—	PCH<2:0>			650
CMOUT	—	—	—	—	—	—	C2OUT	C1OUT	650

Legend: — = unimplemented, read as '0'. Shaded cells are unused by the comparator module.

TABLE 42-1: REGISTER FILE SUMMARY FOR PIC18(L)F26/27/45/46/47/55/56/57K42 DEVICES

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
3A5Bh	RB2I2C	—	SLEW	PU		—	—	TH		263
3A5Ah	RB1I2C	—	SLEW	PU		—	—	TH		263
3A59h	—	Reserved, maintain as '0'								
3A58h	—	Reserved, maintain as '0'								
3A57h	IOCBF	IOCBF7	IOCBF6	IOCBF5	IOCBF4	IOCBF3	IOCBF2	IOCBF1	IOCBF0	287
3A56h	IOCBN	IOCBN7	IOCBN6	IOCBN5	IOCBN4	IOCBN3	IOCBN2	IOCBN1	IOCBN0	287
3A55h	IOCBP	IOCBP7	IOCBP6	IOCBP5	IOCBP4	IOCBP3	IOCBP2	IOCBP1	IOCBP0	287
3A54h	INLVLB	INLVLB7	INLVLB6	INLVLB5	INLVLB4	INLVLB3	INLVLB2	INLVLB1	INLVLB0	270
3A53h	SLRCONB	SLRB7	SLRB6	SLRB5	SLRB4	SLRB3	SLRB2	SLRB1	SLRB0	269
3A52h	ODCONB	ODCB7	ODCB6	ODCB5	ODCB4	ODCB3	ODCB2	ODCB1	ODCB0	268
3A51h	WPUB	WPUB7	WPUB6	WPUB5	WPUB4	WPUB3	WPUB2	WPUB1	WPUB0	267
3A50h	ANSELB	ANSELB7	ANSELB6	ANSELB5	ANSELB4	ANSELB3	ANSELB2	ANSELB1	ANSELB0	266
3A4Fh - 3A4Ah	—	Unimplemented								
3A49h	—	Reserved, maintain as '0'								
3A48h	—	Reserved, maintain as '0'								
3A47h	IOCAF	IOCAF7	IOCAF6	IOCAF5	IOCAF4	IOCAF3	IOCAF2	IOCAF1	IOCAF0	287
3A46h	IOCAN	IOCAN7	IOCAN6	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1	IOCAN0	287
3A45h	IOCAP	IOCAP7	IOCAP6	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1	IOCAP0	287
3A44h	INLVLA	INLVLA7	INLVLA6	INLVLA5	INLVLA4	INLVLA3	INLVLA2	INLVLA1	INLVLA0	270
3A43h	SLRCONA	SLRA7	SLRA6	SLRA5	SLRA4	SLRA3	SLRA2	SLRA1	SLRA0	269
3A42h	ODCONA	ODCA7	ODCA6	ODCA5	ODCA4	ODCA3	ODCA2	ODCA1	ODCA0	268
3A41h	WPUA	WPUA7	WPUA6	WPUA5	WPUA4	WPUA3	WPUA2	WPUA1	WPUA0	267
3A40h	ANSELA	ANSELA7	ANSELA6	ANSELA5	ANSELA4	ANSELA3	ANSELA2	ANSELA1	ANSELA0	266
3A3Fh - 3A30h	—	Unimplemented								
3A2Fh	RF7PPS(3)	—	—	—	RF7PPS4	RF7PPS3	RF7PPS2	RF7PPS1	RF7PPS0	280
3A2Eh	RF6PPS(3)	—	—	—	RF6PPS4	RF6PPS3	RF6PPS2	RF6PPS1	RF6PPS0	280
3A2Dh	RF5PPS(3)	—	—	—	RF5PPS4	RF5PPS3	RF5PPS2	RF5PPS1	RF5PPS0	280
3A2Ch	RF4PPS(3)	—	—	—	RF4PPS4	RF4PPS3	RF4PPS2	RF4PPS1	RF4PPS0	280
3A2Bh	RF3PPS(3)	—	—	—	RF3PPS4	RF3PPS3	RF3PPS2	RF3PPS1	RF3PPS0	280
3A2Ah	RF2PPS(3)	—	—	—	RF2PPS4	RF2PPS3	RF2PPS2	RF2PPS1	RF2PPS0	280
3A29h	RF1PPS(3)	—	—	—	RF1PPS4	RF1PPS3	RF1PPS2	RF1PPS1	RF1PPS0	280
3A28h	RF0PPS(3)	—	—	—	RF0PPS4	RF0PPS3	RF0PPS2	RF0PPS1	RF0PPS0	280
3A27h- 3A23h	—	Unimplemented								
3A22h	RE2PPS(2)	—	—	—	RE2PPS4	RE2PPS3	RE2PPS2	RE2PPS1	RE2PPS0	280
3A21h	RE1PPS(2)	—	—	—	RE1PPS4	RE1PPS3	RE1PPS2	RE1PPS1	RE1PPS0	280
3A20h	RE0PPS(2)	—	—	—	RE0PPS4	RE0PPS3	RE0PPS2	RE0PPS1	RE0PPS0	280
3A1Fh	RD7PPS(2)	—	—	—	RD7PPS4	RD7PPS3	RD7PPS2	RD7PPS1	RD7PPS0	280
3A1Eh	RD6PPS(2)	—	—	—	RD6PPS4	RD6PPS3	RD6PPS2	RD6PPS1	RD6PPS0	280
3A1Dh	RD5PPS(2)	—	—	—	RD5PPS4	RD5PPS3	RD5PPS2	RD5PPS1	RD5PPS0	280
3A1Ch	RD4PPS(2)	—	—	—	RD4PPS4	RD4PPS3	RD4PPS2	RD4PPS1	RD4PPS0	280
3A1Bh	RD3PPS(2)	—	—	—	RD3PPS4	RD3PPS3	RD3PPS2	RD3PPS1	RD3PPS0	280
3A1Ah	RD2PPS(2)	—	—	—	RD2PPS4	RD2PPS3	RD2PPS2	RD2PPS1	RD2PPS0	280
3A19h	RD1PPS(2)	—	—	—	RD1PPS4	RD1PPS3	RD1PPS2	RD1PPS1	RD1PPS0	280
3A18h	RD0PPS(2)	—	—	—	RD0PPS4	RD0PPS3	RD0PPS2	RD0PPS1	RD0PPS0	280
3A17h	RC7PPS	—	—	—	RC7PPS4	RC7PPS3	RC7PPS2	RC7PPS1	RC7PPS0	280

Legend: x = unknown, u = unchanged, — = unimplemented, q = value depends on condition

Note 1: Unimplemented in LF devices.
2: Unimplemented in PIC18(L)F26/27K42.
3: Unimplemented on PIC18(L)F26/27/45/46/47K42 devices.
4: Unimplemented in PIC18(L)F45/55K42.

TABLE 42-1: REGISTER FILE SUMMARY FOR PIC18(L)F26/27/45/46/47/55/56/57K42 DEVICES

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
3989h	IPR9	—	—	—	—	CLC3IP	CWG3IP	CCP3IP	TMR6IP	165
3988h	IPR8	TMR5GIP	TMR5IP	—	—	—	—	—	—	164
3987h	IPR7	—	—	INT2IP	CLC2IP	CWG2IP	-	CCP2IP	TMR4IP	164
3986h	IPR6	TMR3GIP	TMR3IP	U2IP	U2EIP	U2TXIP	U2RXIP	I2C2EIP	I2C2IP	163
3985h	IPR5	I2C2TXIP	I2C2RXIP	DMA2AIP	DMA2ORIP	DMA2DCN-TIP	DMA2SCN-TIP	C2IP	INT1IP	162
3984h	IPR4	CLC1IP	CWG1IP	NCO1IP	—	CCP1IP	TMR2IP	TMR1GIP	TMR1IP	161
3983h	IPR3	TMR0IP	U1IP	U1EIP	U1TXIP	U1RXIP	I2C1EIP	I2C1IP	I2C1TXIP	160
3982h	IPR2	I2C1RXIP	SPI1IP	SPI1TXIP	SPI1RXIP	DMA1AIP	DMA1ORIP	DMA1DCN-TIP	DMA1SCNTIP	159
3981h	IPR1	SMT1PWAIP	SMT1PRAIP	SMT1IP	C1IP	ADTIP	ADIP	ZCDIP	INT0IP	158
3980h	IPR0	IOCIP	CRCIP	SCANIP	NVMIP	CSWIP	OSFIP	HLVDIP	SWIP	157
397Fh - 397Eh	—	Unimplemented								
397Dh	SCANTRIG	—	—	—	—	TSEL				226
397Ch	SCANCON0	EN	TRIGEN	SGO	—	—	MREG	BURSTMD	BUSY	222
397Bh	SCANHADRU	—	—	HADR						224
397Ah	SCANHADRH	HADR								225
3979h	SCANHADRL	HADR								225
3978h	SCANLADRU	—	—	LADR						223
3977h	SCANLADRH	LADR								223
3976h	SCANLADRL	LADR								224
3975h - 396Ah	—	Unimplemented								
3969h	CRCCON1	DLEN				PLEN				218
3968h	CRCCON0	EN	CRCGO	BUSY	ACCM	—	—	SHIFTM	FULL	218
3967h	CRCXORH	X15	X14	X13	X12	X11	X10	X9	X8	221
3966h	CRCXORL	X7	X6	X5	X4	X3	X2	X1	—	221
3965h	CRCSHIFTH	SHFT15	SHFT14	SHFT13	SHFT12	SHFT11	SHFT10	SHFT9	SHFT8	220
3964h	CRCSHIFTL	SHFT7	SHFT6	SHFT5	SHFT4	SHFT3	SHFT2	SHFT1	SHFT0	220
3963h	CRCACCH	ACC15	ACC14	ACC13	ACC12	ACC11	ACC10	ACC9	ACC8	219
3962h	CRCACCL	ACC7	ACC6	ACC5	ACC4	ACC3	ACC2	ACC1	ACC0	220
3961h	CRCDATH	DATA15	DATA14	DATA13	DATA12	DATA11	DATA10	DATA9	DATA8	219
3960h	CRCDATL	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0	219
395Fh	WDTTMR	WDTTMR					STATE	PSCNT		185
395Eh	WDTPSH	PSCNT								184
395Dh	WDTPSL	PSCNT								184
395Ch	WDTCON1	—	CS			—	WINDOW			183
395Bh	WDTCON0	—	—	PS					SEN	182
395Ah - 38A0h	—	Unimplemented								
389Fh	IVTADU	AD								167
389Eh	IVTADH	AD								167
389Dh	IVTADL	AD								167
389Ch - 3891h	—	Unimplemented								
3890h	PRODH_SHAD	PRODH								125
388Fh	PRODL_SHAD	PRODL								125
388Eh	FSR2H_SHAD	—	—	FSR2H						125

Legend: x = unknown, u = unchanged, — = unimplemented, q = value depends on condition

Note 1: Unimplemented in LF devices.
2: Unimplemented in PIC18(L)F26/27K42.
3: Unimplemented on PIC18(L)F26/27/45/46/47K42 devices.
4: Unimplemented in PIC18(L)F45/55K42.