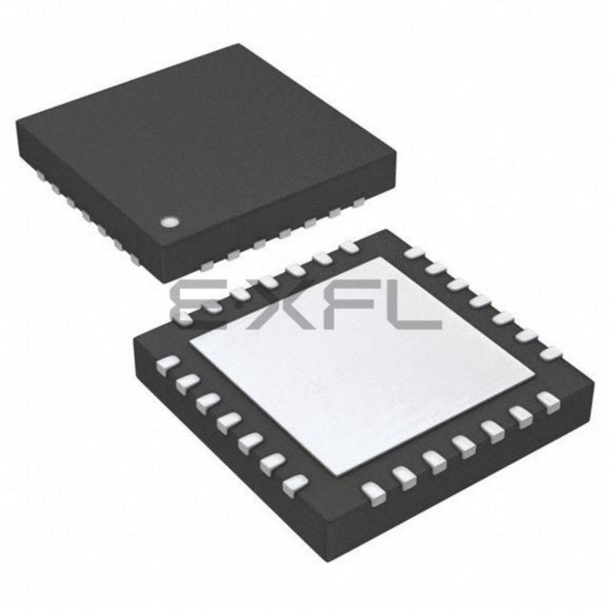


Welcome to E-XFL.COM

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	48MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	22
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	3.8K x 8
Voltage - Supply (Vcc/Vdd)	2.15V ~ 3.6V
Data Converters	A/D 10x10b/12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-VQFN Exposed Pad
Supplier Device Package	28-QFN (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f26j13-i-ml

TABLE 5-2: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices		Power-on Reset, Brown-out Reset, Wake From Deep Sleep	$\overline{\text{MCLR}}$ Resets WDT Reset RESET Instruction Stack Resets CM Resets	Wake-up via WDT or Interrupt
SPBRGH2	PIC18F2XJ13	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
BAUDCON2	PIC18F2XJ13	PIC18F4XJ13	0100 0-00	0100 0-00	uuuu u-uu
TMR3H	PIC18F2XJ13	PIC18F4XJ13	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR3L	PIC18F2XJ13	PIC18F4XJ13	xxxx xxxx	uuuu uuuu	uuuu uuuu
T3CON	PIC18F2XJ13	PIC18F4XJ13	0000 0000	uuuu uuuu	uuuu uuuu
TMR4	PIC18F2XJ13	PIC18F4XJ13	0000 0000	uuuu uuuu	uuuu uuuu
PR4	PIC18F2XJ13	PIC18F4XJ13	1111 1111	1111 1111	uuuu uuuu
T4CON	PIC18F2XJ13	PIC18F4XJ13	-000 0000	-000 0000	-uuu uuuu
SSP2BUF	PIC18F2XJ13	PIC18F4XJ13	xxxx xxxx	uuuu uuuu	uuuu uuuu
SSP2ADD	PIC18F2XJ13	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
SSP2MSK	PIC18F2XJ13	PIC18F4XJ13	1111 1111	1111 1111	uuuu uuuu
SSP2STAT	PIC18F2XJ13	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
SSP2CON1	PIC18F2XJ13	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
SSP2CON2	PIC18F2XJ13	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
CMSTAT	PIC18F2XJ13	PIC18F4XJ13	---- -111	---- -111	---- -uuu
PMADDRH(5)	—	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
PMDOUT1H(5)	—	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
PMADDRL	—	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
PMDOUT1L	—	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
PMDIN1H	—	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
PMDIN1L	—	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
TXADDRL	PIC18F2XJ13	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
TXADDRH	PIC18F2XJ13	PIC18F4XJ13	---- 0000	---- 0000	---- uuuu
RXADDRL	PIC18F2XJ13	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
RXADDRH	PIC18F2XJ13	PIC18F4XJ13	---- 0000	---- 0000	---- uuuu
DMABCL	PIC18F2XJ13	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
DMABCH	PIC18F2XJ13	PIC18F4XJ13	---- --00	---- --00	---- --uu
PMCONH	PIC18F2XJ13	PIC18F4XJ13	0--0 0000	0--0 0000	u--u uuuu
PMCONL	—	PIC18F4XJ13	000- 0000	000- 0000	uuu- uuuu
PMMODEH	—	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
PMMODEL	—	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu
PMDOUT2H	—	PIC18F4XJ13	0000 0000	0000 0000	uuuu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition.

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 the Reset value for a specific condition.

5: Not implemented on PIC18F2XJ13 devices.

6: Not implemented on "LF" devices.

6.1.4.4 Stack Full and Underflow Resets

Device Resets on stack overflow and stack underflow conditions are enabled by setting the STVREN bit in Configuration Register 1L. When STVREN is set, a full or underflow condition sets the appropriate STKFUL or STKUNF bit and then causes a device Reset. When STVREN is cleared, a full or underflow condition sets the appropriate STKFUL or STKUNF bit, but does not cause a device Reset. The STKFUL or STKUNF bits are cleared by the user software or a POR.

6.1.5 FAST REGISTER STACK (FRS)

A Fast Register Stack (FRS) is provided for the STATUS, WREG and BSR registers to provide a “fast return” option for interrupts. This stack is only one level deep and is neither readable nor writable. It is loaded with the current value of the corresponding register when the processor vectors for an interrupt. All interrupt sources push values into the Stack registers. The values in the registers are then loaded back into the working registers if the `RETFIE, FAST` instruction is used to return from the interrupt.

If both low-priority and high-priority interrupts are enabled, the Stack registers cannot be used reliably to return from low-priority interrupts. If a high-priority interrupt occurs while servicing a low-priority interrupt, the Stack register values stored by the low-priority interrupt will be overwritten. In these cases, users must save the key registers in software during a low-priority interrupt.

If interrupt priority is not used, all interrupts may use the FRS for returns from interrupt. If no interrupts are used, the FRS can be used to restore the STATUS, WREG and BSR registers at the end of a subroutine call. To use the Fast Register Stack for a subroutine call, a `CALL label, FAST` instruction must be executed to save the STATUS, WREG and BSR registers to the Fast Register Stack. A `RETURN, FAST` instruction is then executed to restore these registers from the FRS.

Example 6-1 provides a source code example that uses the FRS during a subroutine call and return.

EXAMPLE 6-1: FAST REGISTER STACK CODE EXAMPLE

```
CALL SUB1, FAST    ;STATUS, WREG, BSR
                   ;SAVED IN FAST REGISTER
                   ;STACK
          .
          .

SUB1      .
          .
    RETURN FAST    ;RESTORE VALUES SAVED
                   ;IN FAST REGISTER STACK
```

6.1.6 LOOK-UP TABLES IN PROGRAM MEMORY

There may be programming situations that require the creation of data structures or look-up tables in program memory. For PIC18 devices, look-up tables can be implemented in two ways:

- Computed `GOTO`
- Table Reads

6.1.6.1 Computed `GOTO`

A computed `GOTO` is accomplished by adding an offset to the PC. An example is shown in Example 6-2.

A look-up table can be formed with an `ADDWF PCL` instruction and a group of `RETLW nn` instructions. The W register is loaded with an offset into the table before executing a call to that table. The first instruction of the called routine is the `ADDWF PCL` instruction. The next executed instruction will be one of the `RETLW nn` instructions that returns the value, ‘`nn`’, to the calling function.

The offset value (in WREG) specifies the number of bytes that the PC should advance and should be multiples of 2 (LSb = `0`).

In this method, only one byte may be stored in each instruction location, but room on the return address stack is required.

EXAMPLE 6-2: COMPUTED `GOTO` USING AN OFFSET VALUE

```
        MOVF    OFFSET, W
        CALL    TABLE
ORG     nn00h
TABLE   ADDWF   PCL
        RETLW   nnh
        RETLW   nnh
        RETLW   nnh
        .
        .
        .
```

6.1.6.2 Table Reads

A better method of storing data in program memory allows two bytes to be stored in each instruction location.

Look-up table data may be stored two bytes per program word while programming. The Table Pointer (TBLPTR) specifies the byte address and the Table Latch (TABLAT) contains the data that is read from the program memory. Data is transferred from program memory, one byte at a time.

Table read operation is discussed further in **Section 7.1 “Table Reads and Table Writes”**.

EXAMPLE 7-3: WRITING TO FLASH PROGRAM MEMORY

```
                   MOVLW  CODE_ADDR_UPPER        ; Load TBLPTR with the base address
                   MOVWF  TBLPTRU                ; of the memory block, minus 1
                   MOVLW  CODE_ADDR_HIGH
                   MOVWF  TBLPTRH
                   MOVLW  CODE_ADDR_LOW
                   MOVWF  TBLPTRL
ERASE_BLOCK
                   BSF    EECON1, WREN           ; enable write to memory
                   BSF    EECON1, FREE           ; enable Erase operation
                   BCF    INTCON, GIE            ; disable interrupts
                   MOVLW  0x55
                   MOVWF  EECON2                 ; write 55h
                   MOVLW  0xAA
                   MOVWF  EECON2                 ; write 0AAh
                   BSF    EECON1, WR             ; start erase (CPU stall)
                   BSF    INTCON, GIE            ; re-enable interrupts
                   MOVLW  D'16'
                   MOVWF  WRITE_COUNTER          ; Need to write 16 blocks of 64 to write
                                                 ; one erase block of 1024
RESTART_BUFFER
                   MOVLW  D'64'
                   MOVWF  COUNTER
                   MOVLW  BUFFER_ADDR_HIGH       ; point to buffer
                   MOVWF  FSR0H
                   MOVLW  BUFFER_ADDR_LOW
                   MOVWF  FSR0L
FILL_BUFFER
                   ...                           ; read the new data from I2C, SPI,
                                                 ; PSP, USART, etc.
WRITE_BUFFER
                   MOVLW  D'64'                  ; number of bytes in holding register
                   MOVWF  COUNTER
WRITE_BYTE_TO_HREGS
                   MOVFF  POSTINC0, WREG         ; get low byte of buffer data
                   MOVWF  TABLAT                 ; present data to table latch
                   TBLWT+*                       ; write data, perform a short write
                                                 ; to internal TBLWT holding register.
                   DECFSZ COUNTER                ; loop until buffers are full
                   BRA    WRITE_BYTE_TO_HREGS
PROGRAM_MEMORY
                   BSF    EECON1, WREN           ; enable write to memory
                   BCF    INTCON, GIE            ; disable interrupts
                   MOVLW  0x55
    Required       MOVWF  EECON2                 ; write 55h
    Sequence       MOVLW  0xAA
                   MOVWF  EECON2                 ; write 0AAh
                   BSF    EECON1, WR             ; start program (CPU stall)
                   BSF    INTCON, GIE            ; re-enable interrupts
                   BCF    EECON1, WREN           ; disable write to memory

                   DECFSZ WRITE_COUNTER          ; done with one write cycle
                   BRA    RESTART_BUFFER         ; if not done replacing the erase block
```

TABLE 10-5: PORTB I/O SUMMARY (CONTINUED)

Pin	Function	TRIS Setting	I/O	I/O Type	Description
RB4/CCP4/ PMA1/KBI0/ SCL2(4)/RP7	RB4	0	O	DIG	LATB<4> data output; not affected by an analog input.
		1	I	TTL	PORTB<4> data input; weak pull-up when the $\overline{\text{RBPU}}$ bit is cleared. Disabled when an analog input is enabled.(1)
	CCP4(3)	1	I	ST	Capture input.
		0	O	DIG	Compare/PWM output.
	PMA1	x	I/O	ST/TTL/ DIG	Parallel Master Port address.
	KBI0	1	I	TTL	Interrupt-on-change pin.
	SCL2(4)	1	I	I^{2}C/ SMBus	I^{2}C clock input (MSSP2 module).
	RP7	1	I	ST	Remappable Peripheral Pin 7 input.
		0	O	DIG	Remappable Peripheral Pin 7 output.
RB5/CCP5/ PMA0/KBI1/ SDA2(4)/RP8	RB5	0	O	DIG	LATB<5> data output.
		1	I	TTL	PORTB<5> data input; weak pull-up when the $\overline{\text{RBPU}}$ bit is cleared.
	CCP5(3)	1	I	ST	Capture input.
		0	O	DIG	Compare/PWM output.
	PMA0(3)	x	I/O	ST/TTL/ DIG	Parallel Master Port address.
	KBI1	1	I	TTL	Interrupt-on-change pin.
	SDA2(4)	1	I	I^{2}C/ SMBus	I^{2}C data input (MSSP2 module).
	RP8	1	I	ST	Remappable Peripheral Pin 8 input.
		0	O	DIG	Remappable Peripheral Pin 8 output.
RB6/CCP6/ KBI2/PGC/RP9	RB6	0	O	DIG	LATB<6> data output.
		1	I	TTL	PORTB<6> data input; weak pull-up when the $\overline{\text{RBPU}}$ bit is cleared.
	CCP6(3)	1	I	ST	Capture input.
		0	O	DIG	Compare/PWM output.
	KBI2	1	I	TTL	Interrupt-on-change pin.
	PGC	x	I	ST	Serial execution (ICSP™) clock input for ICSP and ICD operation.(2)
	RP9	1	I	ST	Remappable Peripheral Pin 9 input.
		0	O	DIG	Remappable Peripheral Pin 9 output.

Legend: DIG = Digital level output; TTL = TTL input buffer; ST = Schmitt Trigger input buffer; ANA = Analog level input/output; x = Don’t care (TRISx bit does not affect port direction or is overridden for this option)

Note 1: Pins are configured as analog inputs by default on POR. Using these pins for digital inputs requires setting the appropriate bits in the ANCON1 register.

2: All other pin functions are disabled when ICSP™ or ICD is enabled.

3: Only on 44-pin devices (PIC18F46J13, PIC18F47J13, PIC18LF46J13 and PIC18LF47J13).

4: Only on 28-pin devices (PIC18F26J13, PIC18F27J13, PIC18LF26J13 and PIC18LF27J13).

REGISTER 10-42: RPOR18: PERIPHERAL PIN SELECT OUTPUT REGISTER 18 (BANKED ED2h)

U-0	U-0	U-0	R/$\overline{\text{W}}$-0	R/$\overline{\text{W}}$-0	R/$\overline{\text{W}}$-0	R/$\overline{\text{W}}$-0	R/$\overline{\text{W}}$-0
—	—	—	RP18R4	RP18R3	RP18R2	RP18R1	RP18R0
bit 7							bit 0

Legend:	R/$\overline{\text{W}}$ = Readable bit, Writable bit if IOLOCK = 0		
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 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **RP18R<4:0>:** Peripheral Output Function is Assigned to RP18 Output Pin bits
(see Table 10-14 for peripheral function numbers)

REGISTER 10-43: RPOR19: PERIPHERAL PIN SELECT OUTPUT REGISTER 19 (BANKED ED3h)[1]

U-0	U-0	U-0	R/$\overline{\text{W}}$-0	R/$\overline{\text{W}}$-0	R/$\overline{\text{W}}$-0	R/$\overline{\text{W}}$-0	R/$\overline{\text{W}}$-0
—	—	—	RP19R4	RP19R3	RP19R2	RP19R1	RP19R0
bit 7							bit 0

Legend:	R/$\overline{\text{W}}$ = Readable bit, Writable bit if IOLOCK = 0		
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 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **RP19R<4:0>:** Peripheral Output Function is Assigned to RP19 Output Pin bits
(see Table 10-14 for peripheral function numbers)

Note 1: RP19 pins are not available on 28-pin devices.

REGISTER 10-44: RPOR20: PERIPHERAL PIN SELECT OUTPUT REGISTER 20 (BANKED ED4h)[1]

U-0	U-0	U-0	R/$\overline{\text{W}}$-0	R/$\overline{\text{W}}$-0	R/$\overline{\text{W}}$-0	R/$\overline{\text{W}}$-0	R/$\overline{\text{W}}$-0
—	—	—	RP20R4	RP20R3	RP20R2	RP20R1	RP20R0
bit 7							bit 0

Legend:	R/$\overline{\text{W}}$ = Readable bit, Writable bit if IOLOCK = 0		
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 7-5 **Unimplemented:** Read as ‘0’

bit 4-0 **RP20R<4:0>:** Peripheral Output Function is Assigned to RP20 Output Pin bits
(see Table 10-14 for peripheral function numbers)

Note 1: RP20 pins are not available on 28-pin devices.

15.1 Timer3/5 Gate Control Register

The Timer3/5 Gate Control register (TxGCON), provided in Register 14-2, is used to control the Timerx gate.

REGISTER 15-2: TxGCON: TIMER3/5 GATE CONTROL REGISTER[1] (ACCESS F97h, BANKED F21h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-x	R/W-0	R/W-0
TMRxGE	TxGPOL	TxGTM	TxGSPM	TxGGO/$\overline{\text{TxDONE}}$	TxGVAL	TxGSS1	TxGSS0
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 **TMRxGE:** Timer Gate Enable bit

If TMRxON = 0:
This bit is ignored.

If TMRxON = 1:
1 = Timer counting is controlled by the Timerx gate function
0 = Timer counts regardless of the Timerx gate function

bit 6 **TxGPOL:** Gate Polarity bit

1 = Timer gate is active-high (Timerx counts when the gate is high)
0 = Timer gate is active-low (Timerx counts when the gate is low)

bit 5 **TxGTM:** Gate Toggle Mode bit

1 = Timer Gate Toggle mode is enabled.
0 = Timer Gate Toggle mode is disabled and toggle flip-flop is cleared
Timerx gate flip-flop toggles on every rising edge.

bit 4 **TxGSPM:** Timer Gate Single Pulse Mode bit

1 = Timer Gate Single Pulse mode is enabled and is controlling Timerx gate
0 = Timer Gate Single Pulse mode is disabled

bit 3 **TxGGO/$\overline{\text{TxDONE}}$:** Timer Gate Single Pulse Acquisition Status bit

1 = Timer gate single pulse acquisition is ready, waiting for an edge
0 = Timer gate single pulse acquisition has completed or has not been started
This bit is automatically cleared when TxGSPM is cleared.

bit 2 **TxGVAL:** Timer Gate Current State bit

Indicates the current state of the Timer gate that could be provided to TMRxH:TMRxL. Unaffected by the Timer Gate Enable bit (TMRxGE).

bit 1-0 **TxGSS<1:0>:** Timer Gate Source Select bits

11 = Comparator 2 output
10 = Comparator 1 output
01 = TMR4/6 to match PR4/6 output
00 = T3G/T5G gate input pin

Note 1: Programming the TxGCON prior to TxCON is recommended.

15.5.4 TIMER3/5 GATE SINGLE PULSE MODE

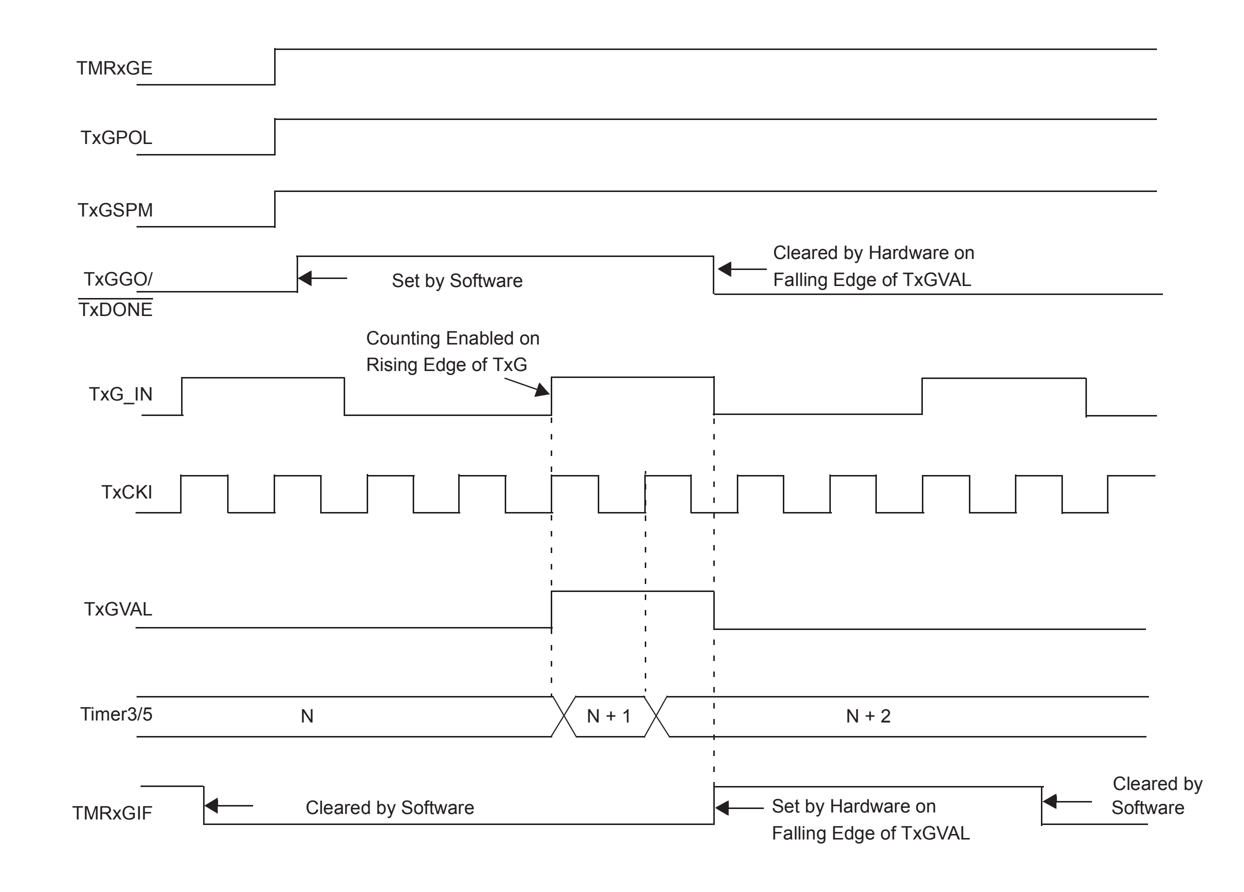
When Timer3/5 Gate Single Pulse mode is enabled, it is possible to capture a single pulse gate event. Timer3/5 Gate Single Pulse mode is first enabled by setting the TxGSPM bit (TxGCON<4>). Next, the TxGGO/$\overline{\text{TxDONE}}$ bit (TxGCON<3>) must be set.

The Timer3/5 will be fully enabled on the next incrementing edge. On the next trailing edge of the pulse, the TxGGO/$\overline{\text{TxDONE}}$ bit will automatically be cleared. No other gate events will be allowed to increment Timer3/5 until the TxGGO/$\overline{\text{TxDONE}}$ bit is once again set in software.

Clearing the TxGSPM bit will also clear the TxGGO/$\overline{\text{TxDONE}}$ bit. (For timing details, see Figure 15-4.)

Simultaneously, enabling the Toggle mode and the Single Pulse mode will permit both sections to work together. This allows the cycle times on the Timer3/5 gate source to be measured. (For timing details, see Figure 15-5.)

FIGURE 15-4: TIMER3/5 GATE SINGLE PULSE MODE



EXAMPLE 20-2: 512-BYTE SPI MASTER MODE Init AND TRANSFER

```
                                 ;For this example, let's use RP5(RB2) for SCK2,
                                 ;RP4(RB1) for SDO2, and RP3(RB0) for SDI2

                                 ;Let’s use SPI master mode, CKE = 0, CKP = 0,
                                 ;without using slave select signalling.

InitSPIPins:
   movlb      0x0F               ;Select bank 15, for access to ODCON3 register
   bcf        ODCON3, SPI2OD     ;Let’s not use open drain outputs in this example

   bcf        LATB, RB2          ;Initialize our (to be) SCK2 pin low (idle).
   bcf        LATB, RB1          ;Initialize our (to be) SDO2 pin to an idle state
   bcf        TRISB, RB1         ;Make SDO2 output, and drive low
   bcf        TRISB, RB2         ;Make SCK2 output, and drive low (idle state)
   bsf        TRISB, RB0         ;SDI2 is an input, make sure it is tri-stated

                                 ;Now we should unlock the PPS registers, so we can
                                 ;assign the MSSP2 functions to our desired I/O pins.

   movlb      0x0E               ;Select bank 14 for access to PPS registers
   bcf        INTCON, GIE        ;I/O Pin unlock sequence will not work if CPU
                                 ;services an interrupt during the sequence
   movlw      0x55               ;Unlock sequence consists of writing 0x55
   movwf      EECON2             ;and 0xAA to the EECON2 register.
   movlw      0xAA
   movwf      EECON2
   bcf        PPSCON, IOLOCK     ;We may now write to RPINRx and RPORx registers
   bsf        INTCON, GIE        ;May now turn back on interrupts if desired

   movlw      0x03               ;RP3 will be SDI2
   movwf      RPINR21            ;Assign the SDI2 function to pin RP3

   movlw      0x0A               ;Let’s assign SCK2 output to pin RP4
   movwf      RPOR4              ;RPOR4 maps output signals to RP4 pin
   movlw      0x04               ;SCK2 also needs to be configured as an input on the
                                 same pin
   movwf      RPINR22            ;SCK2 input function taken from RP4 pin
   movlw      0x09               ;0x09 is SDO2 output
   movwf      RPOR5              ;Assign SDO2 output signal to the RP5 (RB2) pin
   movlb      0x0F               ;Done with PPS registers, bank 15 has other SFRs

InitMSSP2:
   clrf       SSP2STAT           ;CKE = 0, SMP = 0 (sampled at middle of bit)
   movlw      b'00000000'        ;CKP = 0, SPI Master mode, Fosc/4
   movwf      SSP2CON1           ;MSSP2 initialized
   bsf        SSP2CON1, SSPEN    ;Enable the MSSP2 module

InitSPIDMA:
   movlw      b'00111010'        ;Full duplex, RX/TXINC enabled, no SSCON
   movwf      DMACON1            ;DLYINTEN is set, so DLYCYC3:DLYCYC0 = 1111
   movlw      b'11110000'        ;Minimum delay between bytes, interrupt
   movwf      DMACON2            ;only once when the transaction is complete
```

FIGURE 20-8: I²C SLAVE MODE TIMING WITH SEN = 0 (RECEPTION, 7-BIT ADDRESS)

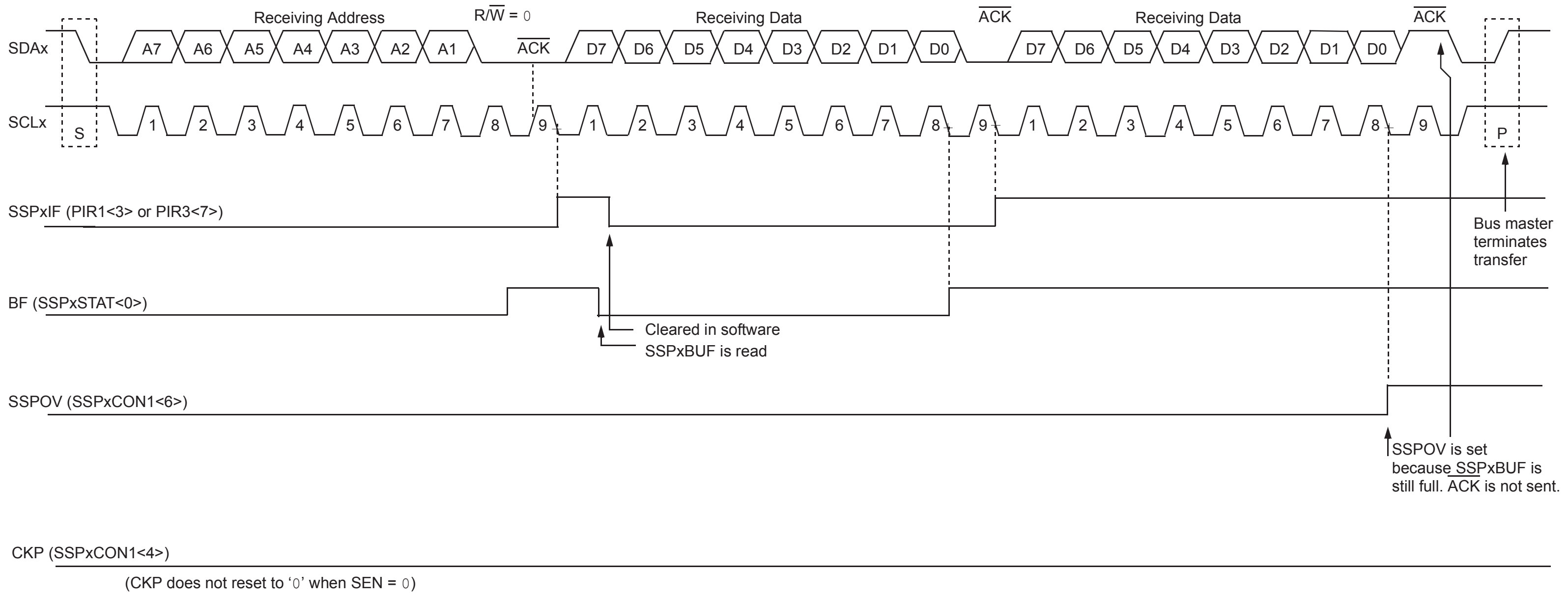


FIGURE 20-29: BUS COLLISION DURING START CONDITION (SCLx = 0)

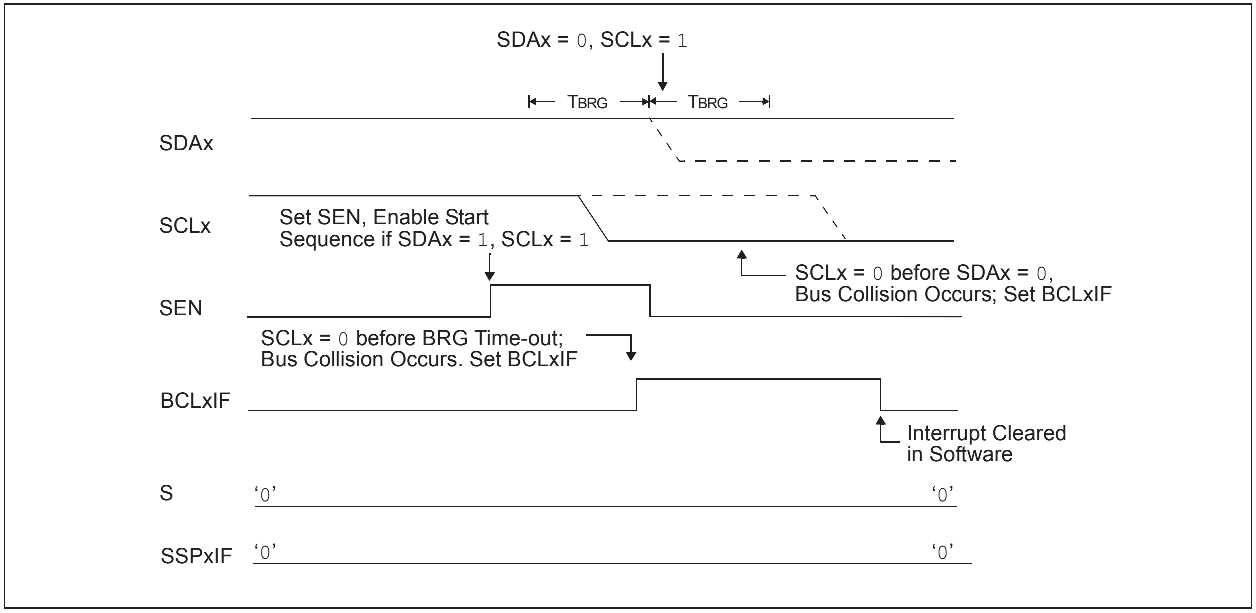
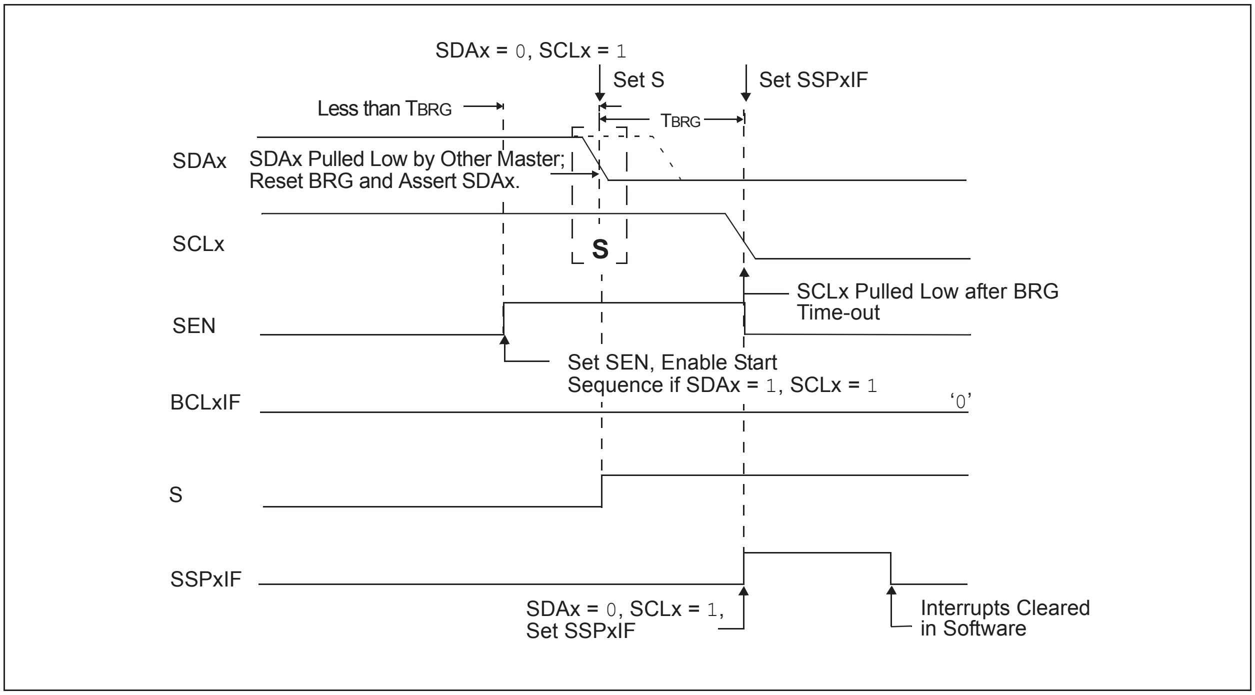

SDAx = 0, SCLx = 1
TBRG
TBRG
SDAx
SCLx
Set SEN, Enable Start Sequence if SDAx = 1, SCLx = 1
SCLx = 0 before SDAx = 0, Bus Collision Occurs; Set BCLxIF
SEN
SCLx = 0 before BRG Time-out; Bus Collision Occurs. Set BCLxIF
BCLxIF
Interrupt Cleared in Software
S
'0'
'0'
SSPxIF
'0'
'0'

FIGURE 20-30: BRG RESET DUE TO SDAx ARBITRATION DURING START CONDITION


SDAx = 0, SCLx = 1
Set S
Set SSPxIF
Less than TBRG
TBRG
SDAx
SDAx Pulled Low by Other Master; Reset BRG and Assert SDAx.
SCLx
S
SCLx Pulled Low after BRG Time-out
SEN
Set SEN, Enable Start Sequence if SDAx = 1, SCLx = 1
BCLxIF
'0'
S
SSPxIF
SDAx = 0, SCLx = 1, Set SSPxIF
Interrupts Cleared in Software

22.0 10/12-BIT ANALOG-TO-DIGITAL CONVERTER (A/D) MODULE

The Analog-to-Digital (A/D) Converter module in the PIC18F47J13 family of devices has 10 inputs for the 28-pin devices and 13 inputs for the 44-pin devices. This module allows conversion of an analog input signal to a corresponding 10- or 12-bit digital number.

The module has these registers:

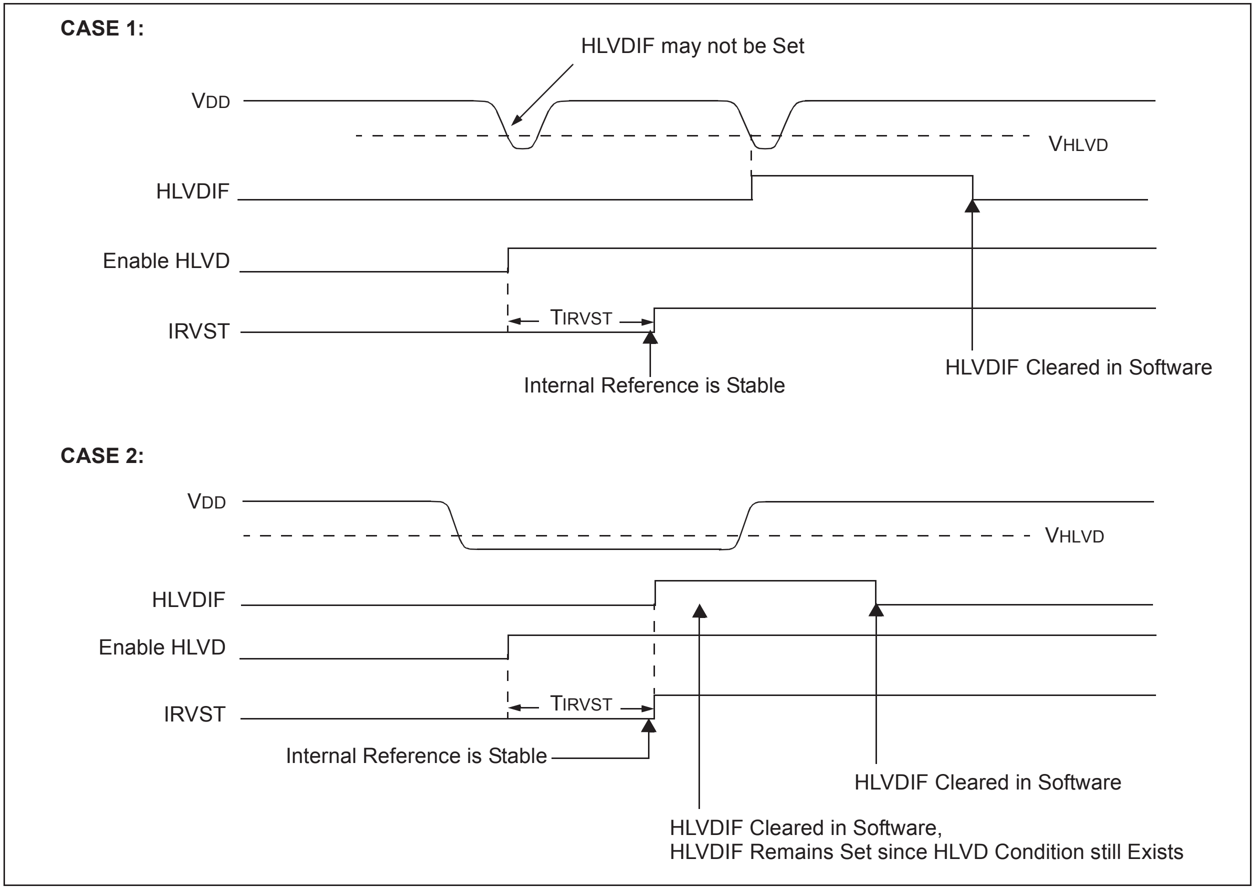
- A/D Control Register 0 (ADCON0)
- A/D Control Register 1 (ADCON1)
- A/D Port Configuration Register 0 (ANCON0)
- A/D Port Configuration Register 1 (ANCON1)
- A/D Result Registers (ADRESH and ADRESL)
- A/D Trigger Register (ADCTRIG)
- Configuration Register 3 High (ADCSEL, CONFIG3H<1>)

The ADCON0 register, shown in Register 22-1, controls the operation of the A/D module.

The ADCON1 register, shown in Register 22-2, configures the A/D clock source, programmed acquisition time and justification. The ANCON0 and ANCON1 registers, in Register 22-1 and Register 22-2, configure the functions of the port pins.

The ADCSEL Configuration bit (CONFIG3H<1>) sets the module for 10- or 12-bit conversions. The 10-Bit Conversion mode is useful for applications that favor conversion speed over conversion resolution.

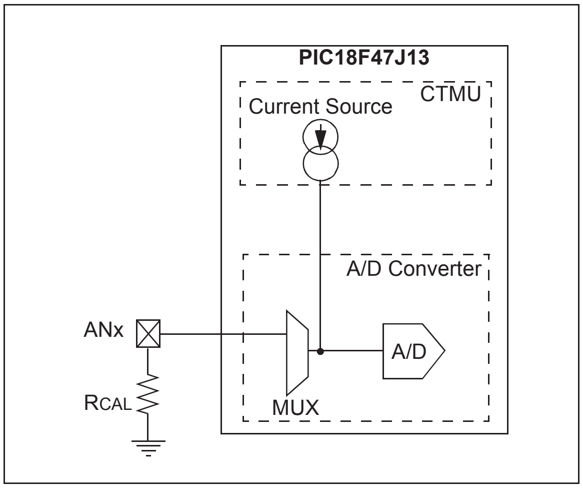
FIGURE 25-2: LOW-VOLTAGE DETECT OPERATION (VDIRMAG = 0)


CASE 1:
HLVDIF may not be Set
VDD
VHLVD
HLVDIF
Enable HLVD
TIRVST
IRVST
Internal Reference is Stable
HLVDIF Cleared in Software
CASE 2:
VDD
VHLVD
HLVDIF
Enable HLVD
TIRVST
IRVST
Internal Reference is Stable
HLVDIF Cleared in Software
HLVDIF Cleared in Software,
HLVDIF Remains Set since HLVD Condition still Exists

The CTMU current source may be trimmed with the trim bits in CTMUICON using an iterative process to get an exact desired current. Alternatively, the nominal value without adjustment may be used; it may be stored by the software for use in all subsequent capacitive or time measurements.

To calculate the optimal value for R_{CAL}, the nominal current must be chosen. For example, if the A/D Converter reference voltage is 3.3V, use 70% of full scale or 2.31V as the desired approximate voltage to be read by the A/D Converter. If the range of the CTMU current source is selected to be 0.55 μA, the resistor value needed is calculated as $R_{CAL} = 2.31\text{V}/0.55\ \mu\text{A}$, for a value of 4.2 MΩ. Similarly, if the current source is chosen to be 5.5 μA, R_{CAL} would be 420,000Ω, and 42,000Ω if the current source is set to 55 μA.

FIGURE 26-2: CTMU CURRENT SOURCE CALIBRATION CIRCUIT



A value of 70% of full-scale voltage is chosen to make sure that the A/D Converter is in a range that is well above the noise floor. Keep in mind that if an exact current is chosen that is to incorporate the trimming bits from CTMUICON, the resistor value of R_{CAL} may need to be adjusted accordingly. R_{CAL} may also be adjusted to allow for available resistor values. R_{CAL} should be of the highest precision available, keeping in mind the amount of precision needed for the circuit that the CTMU will be used to measure. A recommended minimum would be 0.1% tolerance.

The following examples show one typical method for performing a CTMU current calibration. Example 26-1 demonstrates how to initialize the A/D Converter and the CTMU. This routine is typical for applications using both modules. Example 26-2 demonstrates one method for the actual calibration routine.

TABLE 27-1: MAPPING OF THE FLASH CONFIGURATION WORDS TO THE CONFIGURATION REGISTERS

Configuration Register (Volatile)	Configuration Register Address	Flash Configuration Byte Address
CONFIG1L	300000h	XXXF8h
CONFIG1H	300001h	XXXF9h
CONFIG2L	300002h	XXXFAh
CONFIG2H	300003h	XXXFBh
CONFIG3L	300004h	XXXFCh
CONFIG3H	300005h	XXXFDh
CONFIG4L	300006h	XXXFEh
CONFIG4H	300007h	XXXFFh

TABLE 27-2: CONFIGURATION BITS AND DEVICE IDs

File Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default/ Unprog. Value[1]
300000h	CONFIG1L	$\overline{\text{DEBUG}}$	XINST	STVREN	CFGPLLEN	PLLDIV2	PLLDIV1	PLLDIV0	WDTEN	1111 1111
300001h	CONFIG1H	—[2]	—[2]	—[2]	—[2]	—	CP0	—	—	1111 -1--
300002h	CONFIG2L	IESO	FCMEN	CLKOEC	SOSCSEL1	SOSCSEL0	FOSC2	FOSC1	FOSC0	1111 1111
300003h	CONFIG2H	—[2]	—[2]	—[2]	—[2]	WDTPS3	WDTPS2	WDTPS1	WDTPS0	1111 1111
300004h	CONFIG3L	DSWDTPS3	DSWDTPS2	DSWDTPS1	DSWDTPS0	DSWDTEN	DSBOREN	RTCOSC	DSWDTOSC	1111 1111
300005h	CONFIG3H	—[2]	—[2]	—[2]	—[2]	MSSPMSK	PLLSEL	ADCSEL	IOL1WAY	1111 1111
300006h	CONFIG4L	WPCFG	WPFP6	WPFP5	WPFP4	WPFP3	WPFP2	WPFP1	WPFP0	1111 1111
300007h	CONFIG4H	—[2]	—[2]	—[2]	—[2]	—	—	WPEND	WPDIS	1111 --11
3FFFFEh	DEVID1	DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0	xx1x xxxx[3]
3FFFFFh	DEVID2	DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3	0101 10x1[3]

Legend: x = unknown, u = unchanged, - = unimplemented. Shaded cells are unimplemented, read as '0'.

Note 1: Values reflect the unprogrammed state as received from the factory and following Power-on Resets. In all other Reset states, the configuration bytes maintain their previously programmed states.

2: The value of these bits in program memory should always be programmed to '1'. This ensures that the location is executed as a NOP if it is accidentally executed.

3: See Register 27-9 and Register 27-10 for DEVID values. These registers are read-only and cannot be programmed by the user.

27.3 On-Chip Voltage Regulator

Note 1: The on-chip voltage regulator is only available in parts designated with an “F”, such as PIC18**F**26J13. The on-chip regulator is disabled on devices with “LF” in their part number.

2: The VDDCORE/VCAP pin must never be left floating. On “F” devices, it must be connected to a capacitor of size, CEFC, to ground. On “LF” devices, VDDCORE/VCAP must be connected to a power supply source between 2.0V and 2.7V.

The digital core logic of the PIC18F47J13 Family devices is designed on an advanced manufacturing process, which requires 2.0V to 2.7V. The digital core logic obtains power from the VDDCORE/VCAP power supply pin.

However, in many applications it may be inconvenient to run the I/O pins at the same core logic voltage, as it would restrict the ability of the device to interface with other higher voltage devices, such as those run at a nominal 3.3V. Therefore, all PIC18F47J13 Family devices implement a dual power supply rail topology. The core logic obtains power from the VDDCORE/VCAP pin, while the general purpose I/O pins obtain power from the VDD pin of the microcontroller, which may be supplied with a voltage between 2.15V to 3.6V (“F” device) or 2.0V to 3.6V (“LF” device).

This dual supply topology allows the microcontroller to interface with standard 3.3V logic devices, while running the core logic at a lower voltage of nominally 2.5V.

In order to make the microcontroller more convenient to use, an integrated 2.5V low dropout, low quiescent current linear regulator has been integrated on the die inside PIC18F47J13 Family devices. This regulator is designed specifically to supply the core logic of the device. It allows PIC18F47J13 Family devices to effectively run from a single power supply rail, without the need for external regulators.

The on-chip voltage regulator is always enabled on “F” devices. The VDDCORE/VCAP pin simultaneously serves as the regulator output pin and the core logic supply power input pin. A capacitor should be connected to the VDDCORE/VCAP pin to ground and is necessary for regulator stability. For example connections for PIC18F and PIC18LF devices, see Figure 27-2.

On “LF” devices, the on-chip regulator is always disabled. This allows the device to save a small amount of quiescent current consumption, which may be advantageous in some types of applications, such as those which will entirely be running at a nominal 2.5V. On “LF” devices, the VDDCORE/VCAP pin still serves as the core logic power supply input pin, and therefore, must be connected to a 2.0V to 2.7V supply rail at the application circuit board level. On these devices, the I/O pins may still optionally be supplied with a voltage between 2.0V to 3.6V, provided that VDD is always greater than, or equal to, VDDCORE/VCAP. For example connections for PIC18F and PIC18LF devices, see Figure 27-2.

Note: In parts designated with an “LF”, such as PIC18LF47J13, VDDCORE must never exceed VDD.

The specifications for core voltage and capacitance are listed in **Section 30.3 “DC Characteristics: PIC18F47J13 Family (Industrial)”**.

27.3.1 VOLTAGE REGULATOR TRACKING MODE AND LOW-VOLTAGE DETECTION

On “F” devices, the on-chip regulator provides a constant voltage of 2.5V nominal to the digital core logic. The regulator can provide this level from a VDD of about 2.5V, all the way up to the device’s VDDMAX. It does not have the capability to boost VDD levels below 2.5V. When the VDD supply input voltage drops too low to regulate 2.5V, the regulator enters Tracking mode. In Tracking mode, the regulator output follows VDD, with a typical voltage drop of 100 mV or less.

The on-chip regulator includes a simple Low-Voltage Detect (LVD) circuit. This circuit is separate and independent of the High/Low-Voltage Detect (HLVD) module described in **Section 25.0 “High/Low Voltage Detect (HLVD)”**. The on-chip regulator LVD circuit continuously monitors the VDDCORE voltage level and updates the LVDSTAT bit in the WDTCON register. The LVD detect threshold is set slightly below the normal regulation set point of the on-chip regulator.

Application firmware may optionally poll the LVDSTAT bit to determine when it is safe to run at the maximum rated frequency, so as not to inadvertently violate the voltage versus frequency requirements provided by Figure 30-1.

The VDDCORE monitoring LVD circuit is only active when the on-chip regulator is enabled. On “LF” devices, the Analog-to-Digital Converter and the HLVD module can still be used to provide firmware with VDD and VDDCORE voltage level information.

TABLE 28-1: OPCODE FIELD DESCRIPTIONS

Field	Description
a	RAM Access bit: a = 0: RAM location in Access RAM (BSR register is ignored) a = 1: RAM bank is specified by BSR register
bbb	Bit address within an 8-bit file register (0 to 7).
BSR	Bank Select Register. Used to select the current RAM bank.
C, DC, Z, OV, N	ALU Status bits: **C**arry, **D**igit **C**arry, **Z**ero, **Ov**erflow, **N**egative.
d	Destination Select bit: d = 0: store result in WREG d = 1: store result in file register f
dest	Destination: either the WREG register or the specified register file location.
f	8-bit Register file address (00h to FFh), or 2-bit FSR designator (0h to 3h).
f_s	12-bit Register file address (000h to FFFh). This is the source address.
f_d	12-bit Register file address (000h to FFFh). This is the destination address.
GIE	Global Interrupt Enable bit.
k	Literal field, constant data or label (may be either an 8-bit, 12-bit or a 20-bit value).
label	Label name.
mm * *+ *- +*	The mode of the TBLPTR register for the table read and table write instructions. Only used with table read and table write instructions: No change to register (such as TBLPTR with table reads and writes) Post-Increment register (such as TBLPTR with table reads and writes) Post-Decrement register (such as TBLPTR with table reads and writes) Pre-Increment register (such as TBLPTR with table reads and writes)
n	The relative address (2's complement number) for relative branch instructions or the direct address for Call/Branch and Return instructions.
PC	Program Counter.
PCL	Program Counter Low Byte.
PCH	Program Counter High Byte.
PCLATH	Program Counter High Byte Latch.
PCLATU	Program Counter Upper Byte Latch.
$\overline{\text{PD}}$	Power-Down bit.
PRODH	Product of Multiply High Byte.
PRODL	Product of Multiply Low Byte.
s	Fast Call/Return Mode Select bit: s = 0: do not update into/from shadow registers s = 1: certain registers loaded into/from shadow registers (Fast mode)
TBLPTR	21-bit Table Pointer (points to a program memory location).
TABLAT	8-bit Table Latch.
$\overline{\text{TO}}$	Time-out bit.
TOS	Top-of-Stack.
u	Unused or Unchanged.
WDT	Watchdog Timer.
WREG	Working register (accumulator).
x	Don't care ('0' or '1'). The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
z_s	7-bit offset value for Indirect Addressing of register files (source).
z_d	7-bit offset value for Indirect Addressing of register files (destination).
{ }	Optional argument.
[text]	Indicates an Indexed Address.
(text)	The contents of text.
[expr]<n>	Specifies bit n of the register indicated by the pointer expr.
→	Assigned to.
< >	Register bit field.
∈	In the set of.
italics	User-defined term (font is Courier New).

BTFSC — Bit Test File, Skip if Clear

Syntax: BTFSC f, b {,a}

Operands: 0 ≤ f ≤ 255
0 ≤ b ≤ 7
a ∈ [0,1]

Operation: skip if (f<b>) = 0

Status Affected: None

Encoding:

1011	bbba	ffff	ffff

Description: If bit 'b' in register 'f' is '0', then the next instruction is skipped. If bit 'b' is '0', then the next instruction fetched during the current instruction execution is discarded and a NOP is executed instead, making this a 2-cycle instruction.

If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See **Section 28.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1(2)
Note: 3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	No operation

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

If skip and followed by 2-word instruction:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation

Example:

```
HERE    BTFSC   FLAG, 1, 0
FALSE   :
TRUE    :
```

Before Instruction
PC = address (HERE)
After Instruction
If FLAG<1> = 0;
PC = address (TRUE)
If FLAG<1> = 1;
PC = address (FALSE)

BTFSS — Bit Test File, Skip if Set

Syntax: BTFSS f, b {,a}

Operands: 0 ≤ f ≤ 255
0 ≤ b < 7
a ∈ [0,1]

Operation: skip if (f<b>) = 1

Status Affected: None

Encoding:

1010	bbba	ffff	ffff

Description: If bit 'b' in register 'f' is '1', then the next instruction is skipped. If bit 'b' is '1', then the next instruction fetched during the current instruction execution is discarded and a NOP is executed instead, making this a 2-cycle instruction.

If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See **Section 28.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1(2)
Note: 3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	No operation

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

If skip and followed by 2-word instruction:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation

Example:

```
HERE    BTFSS   FLAG, 1, 0
FALSE   :
TRUE    :
```

Before Instruction
PC = address (HERE)
After Instruction
If FLAG<1> = 0;
PC = address (FALSE)
If FLAG<1> = 1;
PC = address (TRUE)

MOVSS — Move Indexed to Indexed

Syntax: MOVSS [z_s], [z_d]

Operands: $0 \le z_s \le 127$
$0 \le z_d \le 127$

Operation: $((FSR2) + z_s) \rightarrow ((FSR2) + z_d)$

Status Affected: None

Encoding:

1st word (source)	1110	1011	1zzz	$zzzz_s$
2nd word (dest.)	1111	xxxx	xzzz	$zzzz_d$

Description: The contents of the source register are moved to the destination register. The addresses of the source and destination registers are determined by adding the 7-bit literal offsets, 'z_s' or 'z_d', respectively, to the value of FSR2. Both registers can be located anywhere in the 4096-byte data memory space (000h to FFFh).

The `MOVSS` instruction cannot use the PCL, TOSU, TOSH or TOSL as the destination register.

If the resultant source address points to an Indirect Addressing register, the value returned will be 00h. If the resultant destination address points to an Indirect Addressing register, the instruction will execute as a `NOP`.

Words: 2

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Determine source addr	Determine source addr	Read source reg
Decode	Determine dest addr	Determine dest addr	Write to dest reg

Example: `MOVSS [0x05], [0x06]`

Before Instruction

FSR2	=	80h
Contents of 85h	=	33h
Contents of 86h	=	11h

After Instruction

FSR2	=	80h
Contents of 85h	=	33h
Contents of 86h	=	33h

PUSHL — Store Literal at FSR2, Decrement FSR2

Syntax: PUSHL k

Operands: $0 \le k \le 255$

Operation: $k \rightarrow (FSR2)$,
$FSR2 - 1 \rightarrow FSR2$

Status Affected: None

Encoding:

1110	1010	kkkk	kkkk

Description: The 8-bit literal 'k' is written to the data memory address specified by FSR2. FSR2 is decremented by 1 after the operation.

This instruction allows users to push values onto a software stack.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read 'k'	Process data	Write to destination

Example: `PUSHL 0x08`

Before Instruction

FSR2H:FSR2L	=	01ECh
Memory (01ECh)	=	00h

After Instruction

FSR2H:FSR2L	=	01EBh
Memory (01ECh)	=	08h

30.2 DC Characteristics: Power-Down and Supply Current PIC18F47J13 Family (Industrial)

PIC18LF47J13 Family		**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial					
PIC18F47J13 Family		**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial					
Param No.	**Device**	**Typ**	**Max**	**Units**	**Conditions**		
	Power-Down Current (IPD)(1) – Sleep mode						
	PIC18LFXXJ13	0.1	1.6	μA	-40°C	VDD = 2.0V, VDDCORE = 2.0V	**Sleep mode**, REGSLP = 1
		0.2	1.6	μA	+25°C		
		0.8	7.0	μA	+60°C		
		2.1	11.5	μA	+85°C		
	PIC18LFXXJ13	0.2	2.0	μA	-40°C	VDD = 2.5V, VDDCORE = 2.5V	
		0.5	2.0	μA	+25°C		
		1.4	9.0	μA	+60°C		
		3.2	15.0	μA	+85°C		
	PIC18FXXJ13	3.0	6.0	μA	-40°C	VDD = 2.15V VDDCORE = 10 μF Capacitor	
		3.8	6.0	μA	+25°C		
		4.7	9.0	μA	+60°C		
		6.4	18.5	μA	+85°C		
	PIC18FXXJ13	3.3	9.0	μA	-40°C	VDD = 3.3V VDDCORE = 10 μF Capacitor	
		4.2	9.0	μA	+25°C		
		5.5	12.0	μA	+60°C		
		7.8	22.0	μA	+85°C		
	Power-Down Current (IPD)(1) – Deep Sleep mode						
	PIC18FXXJ13	2	25	nA	-40°C	VDD = 2.15V, VDDCORE = 10 μF Capacitor	**Deep Sleep** mode
		9	100	nA	+25°C		
		72	250	nA	+60°C		
		262	1000	nA	+85°C		
	PIC18FXXJ13	17	50	nA	-40°C	VDD = 3.3V, VDDCORE = 10 μF Capacitor	
		53	150	nA	+25°C		
		186	400	nA	+60°C		
		503	2000	nA	+85°C		

Note 1: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in a high-impedance state and tied to VDD or VSS, and all features that add delta current are disabled (such as WDT, Timer1 oscillator, BOR, etc.).

2: The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption. All features that add delta current are disabled (such as WDT). The test conditions for all IDD measurements in active operation mode are:

$\overline{\text{OSC1}}$ = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD/VSS;
$\overline{\text{MCLR}}$ = VDD; WDT disabled unless otherwise specified.

3: Low-power Timer1 with standard, low-cost 32 kHz crystals have an operating temperature range of -10°C to +70°C. Extended temperature crystals are available at a much higher cost.

30.3 DC Characteristics: PIC18F47J13 Family (Industrial)

DC CHARACTERISTICS			**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial			
Param No.	**Symbol**	**Characteristic**	**Min**	**Max**	**Units**	**Conditions**
	VIL	**Input Low Voltage**				
		All I/O Ports:				
D030		with TTL Buffer	VSS	0.15 VDD	V	VDD < 3.3V
D030A		with TTL Buffer	VSS	0.8	V	3.3V ≤ VDD ≤ 3.6V
D031		with Schmitt Trigger Buffer	VSS	0.2 VDD	V	
D031A		SCLx/SDAx	—	0.3 VDD	V	I^2C enabled
D031B		SCLx/SDAx	—	0.8	V	SMBus enabled
D032		$\overline{\text{MCLR}}$	VSS	0.2 VDD	V	
D033		OSC1	VSS	0.3 VDD	V	HS, HSPLL modes
D033A		OSC1	VSS	0.2 VDD	V	EC, ECPLL modes
D034		T1OSI	VSS	0.3	V	T1OSCEN = 1
	VIH	**Input High Voltage**				
		I/O Ports without 5.5V Tolerance:				
D040		with TTL Buffer	0.25 VDD + 0.8V	VDD	V	VDD < 3.3V
D040A		with TTL Buffer	2.0	VDD	V	3.3V ≤ VDD ≤ 3.6V
D041		with Schmitt Trigger Buffer	0.8 VDD	VDD	V	
		I/O Ports with 5.5V Tolerance:[4]				
Dxxx		with TTL Buffer	0.25 VDD + 0.8V	5.5	V	VDD < 3.3V
DxxxA			2.0	5.5	V	3.3V ≤ VDD ≤ 3.6V
Dxxx		with Schmitt Trigger Buffer	0.8 VDD	5.5	V	
D041A		SCLx/SDAx	0.7 VDD	—	V	I^2C enabled
D041B		SCLx/SDAx	2.1	—	V	SMBus enabled; VDD ≥ 3V
D042		$\overline{\text{MCLR}}$	0.8 VDD	5.5	V	
D043		OSC1	0.7 VDD	VDD	V	HS, HSPLL modes
D043A		OSC1	0.8 VDD	VDD	V	EC, ECPLL modes
D044		T1OSI	1.6	VDD	V	T1OSCEN = 1
	IPU	**Weak Pull-up Current**				
D070	IPURB	PORTB, PORTD[3] and PORTE[3] Weak Pull-up Current	80	400	μA	VDD = 3.3V, VPIN = VSS

Note 1: The leakage current on the $\overline{\text{MCLR}}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

2: Negative current is defined as current sourced by the pin.

3: Only available in 44-pin devices.

4: Refer to Table 10-2 for pin tolerance levels.