



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	40MHz
Connectivity	CANbus, I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, HLVD, POR, PWM, WDT
Number of I/O	25
Program Memory Size	64KB (32K 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 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	28-DIP (0.300", 7.62mm)
Supplier Device Package	28-SPDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f2680-i-sp

PIC18F2585/2680/4585/4680

3.5.4 EXIT WITHOUT AN OSCILLATOR START-UP DELAY

Certain exits from power managed modes do not invoke the OST at all. There are two cases:

- PRI_IDLE mode where the primary clock source is not stopped; and
- the primary clock source is not any of the LP, XT, HS or HSPLL modes.

In these instances, the primary clock source either does not require an oscillator start-up delay, since it is already running (PRI_IDLE), or normally does not require an oscillator start-up delay (RC, EC and INTIO Oscillator modes). However, a fixed delay of interval T_{CSD} following the wake event is still required when leaving Sleep and Idle modes to allow the CPU to prepare for execution. Instruction execution resumes on the first clock cycle following this delay.

TABLE 3-2: EXIT DELAY ON WAKE-UP BY RESET FROM SLEEP MODE OR ANY IDLE MODE (BY CLOCK SOURCES)

Clock Source Before Wake-up	Clock Source After Wake-up	Exit Delay	Clock Ready Status Bit (OSCCON)
Primary Device Clock (PRI_IDLE mode)	LP, XT, HS	$T_{CSD}^{(2)}$	OSTS
	HSPLL		—
	EC, RC		IOFS
	INTRC ⁽¹⁾		—
	INTOSC ⁽³⁾		IOFS
T1OSC or INTRC ⁽¹⁾	LP, XT, HS	$T_{OST}^{(4)}$	OSTS
	HSPLL	$T_{OST} + t_{rc}^{(4)}$	
	EC, RC	$T_{CSD}^{(2)}$	—
	INTRC ⁽¹⁾		IOFS
	INTOSC ⁽²⁾	$T_{IOBST}^{(5)}$	IOFS
INTOSC ⁽³⁾	LP, XT, HS	$T_{OST}^{(4)}$	OSTS
	HSPLL	$T_{OST} + t_{rc}^{(4)}$	
	EC, RC	$T_{CSD}^{(2)}$	—
	INTRC ⁽¹⁾		IOFS
	INTOSC ⁽²⁾	None	IOFS
None (Sleep mode)	LP, XT, HS	$T_{OST}^{(4)}$	OSTS
	HSPLL	$T_{OST} + t_{rc}^{(4)}$	
	EC, RC	$T_{CSD}^{(2)}$	—
	INTRC ⁽¹⁾		IOFS
	INTOSC ⁽²⁾	$T_{IOBST}^{(5)}$	IOFS

Note 1: In this instance, refers specifically to the 31 kHz INTRC clock source.

2: T_{CSD} (parameter 38) is a required delay when waking from Sleep and all Idle modes and runs concurrently with any other required delays (see **Section 3.4 “Idle Modes”**).

3: Includes both the INTOSC 8 MHz source and postscaler derived frequencies.

4: T_{OST} is the Oscillator Start-up Timer (parameter 32). t_{rc} is the PLL Lock-out Timer (parameter F12); it is also designated as T_{PLL} .

5: Execution continues during T_{IOBST} (parameter 39), the INTOSC stabilization period.

5.3 Data Memory Organization

Note: The operation of some aspects of data memory are changed when the PIC18 extended instruction set is enabled. See **Section 5.6 “Data Memory and the Extended Instruction Set”** for more information.

The data memory in PIC18 devices is implemented as static RAM. Each register in the data memory has a 12-bit address, allowing up to 4096 bytes of data memory. The memory space is divided into as many as 16 banks that contain 256 bytes each; PIC18F2585/2680/4585/4680 devices implement all 16 banks. Figure 5-5 shows the data memory organization for the PIC18F2585/2680/4585/4680 devices.

The data memory contains Special Function Registers (SFRs) and General Purpose Registers (GPRs). The SFRs are used for control and status of the controller and peripheral functions, while GPRs are used for data storage and scratchpad operations in the user's application. Any read of an unimplemented location will read as '0's.

The instruction set and architecture allow operations across all banks. The entire data memory may be accessed by Direct, Indirect or Indexed Addressing modes. Addressing modes are discussed later in this subsection.

To ensure that commonly used registers (SFRs and select GPRs) can be accessed in a single cycle, PIC18 devices implement an Access Bank. This is a 256-byte memory space that provides fast access to SFRs and the lower portion of GPR Bank 0 without using the BSR. **Section 5.3.2 “Access Bank”** provides a detailed description of the Access RAM.

5.3.1 BANK SELECT REGISTER (BSR)

Large areas of data memory require an efficient addressing scheme to make rapid access to any address possible. Ideally, this means that an entire address does not need to be provided for each read or write operation. For PIC18 devices, this is accomplished with a RAM banking scheme. This divides the memory space into 16 contiguous banks of 256 bytes. Depending on the instruction, each location can be addressed directly by its full 12-bit address, or an 8-bit low-order address and a 4-bit bank pointer.

Most instructions in the PIC18 instruction set make use of the bank pointer, known as the Bank Select Register (BSR). This SFR holds the 4 Most Significant bits of a location's address; the instruction itself includes the 8 Least Significant bits. Only the four lower bits of the BSR are implemented (BSR3:BSR0). The upper four bits are unused; they will always read '0' and cannot be written to. The BSR can be loaded directly by using the `MOVLB` instruction.

The value of the BSR indicates the bank in data memory; the 8 bits in the instruction show the location in the bank and can be thought of as an offset from the bank's lower boundary. The relationship between the BSR's value and the bank division in data memory is shown in Figure 5-6.

Since up to 16 registers may share the same low-order address, the user must always be careful to ensure that the proper bank is selected before performing a data read or write. For example, writing what should be program data to an 8-bit address of F9h while the BSR is 0Fh, will end up resetting the program counter.

While any bank can be selected, only those banks that are actually implemented can be read or written to. Writes to unimplemented banks are ignored, while reads from unimplemented banks will return '0's. Even so, the STATUS register will still be affected as if the operation was successful. The data memory map in Figure 5-5 indicates which banks are implemented.

In the core PIC18 instruction set, only the `MOVFF` instruction fully specifies the 12-bit address of the source and target registers. This instruction ignores the BSR completely when it executes. All other instructions include only the low-order address as an operand and must use either the BSR or the Access Bank to locate their target registers.

PIC18F2585/2680/4585/4680

TABLE 5-2: REGISTER FILE SUMMARY (PIC18F2585/2680/4585/4680) (CONTINUED)

File Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Details on page:
B2DLC ⁽⁸⁾ Receive mode	—	RXRTR	RB1	RB0	DLC3	DLC2	DLC1	DLC0	-xxx xxxx	56, 300
B2DLC ⁽⁸⁾ Transmit mode	—	TXRTR	—	—	DLC3	DLC2	DLC1	DLC0	-x-- xxxx	56, 301
B2EIDL ⁽⁸⁾	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	57, 299
B2EIDH ⁽⁸⁾	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	57, 298
B2SIDL ⁽⁸⁾ Receive mode	SID2	SID1	SID0	SRR	EXID	—	EID17	EID16	xxxx x-xx	56, 297
B2SIDL ⁽⁸⁾ Transmit mode	SID2	SID1	SID0	—	EXIDE	—	EID17	EID16	xxx- x-xx	56, 297
B2SIDH ⁽⁸⁾	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	57, 296
B2CON ⁽⁸⁾ Receive mode	RXFUL	RXM1	RXRTRRO	FILHIT4	FILHIT3	FILHIT2	FILHIT1	FILHIT0	0000 0000	58, 294
B2CON ⁽⁸⁾ Transmit mode	TXBIF	RXM1	TXLARB	TXERR	TXREQ	RTREN	TXPRI1	TXPRI0	0000 0000	58, 295
B1D7 ⁽⁸⁾	B1D77	B1D76	B1D75	B1D74	B1D73	B1D72	B1D71	B1D70	xxxx xxxx	58, 299
B1D6 ⁽⁸⁾	B1D67	B1D66	B1D65	B1D64	B1D63	B1D62	B1D61	B1D60	xxxx xxxx	58, 299
B1D5 ⁽⁸⁾	B1D57	B1D56	B1D55	B1D54	B1D53	B1D52	B1D51	B1D50	xxxx xxxx	58, 299
B1D4 ⁽⁸⁾	B1D47	B1D46	B1D45	B1D44	B1D43	B1D42	B1D41	B1D40	xxxx xxxx	58, 299
B1D3 ⁽⁸⁾	B1D37	B1D36	B1D35	B1D34	B1D33	B1D32	B1D31	B1D30	xxxx xxxx	58, 299
B1D2 ⁽⁸⁾	B1D27	B1D26	B1D25	B1D24	B1D23	B1D22	B1D21	B1D20	xxxx xxxx	58, 299
B1D1 ⁽⁸⁾	B1D17	B1D16	B1D15	B1D14	B1D13	B1D12	B1D11	B1D10	xxxx xxxx	58, 299
B1D0 ⁽⁸⁾	B1D07	B1D06	B1D05	B1D04	B1D03	B1D02	B1D01	B1D00	xxxx xxxx	58, 299
B1DLC ⁽⁸⁾ Receive mode	—	RXRTR	RB1	RB0	DLC3	DLC2	DLC1	DLC0	-xxx xxxx	56, 300
B1DLC ⁽⁸⁾ Transmit mode	—	TXRTR	—	—	DLC3	DLC2	DLC1	DLC0	-x-- xxxx	56, 301
B1EIDL ⁽⁸⁾	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	58, 299
B1EIDH ⁽⁸⁾	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	58, 298
B1SIDL ⁽⁸⁾ Receive mode	SID2	SID1	SID0	SRR	EXID	—	EID17	EID16	xxxx x-xx	56, 297
B1SIDL ⁽⁸⁾ Transmit mode	SID2	SID1	SID0	—	EXIDE	—	EID17	EID16	xxx- x-xx	56, 297
B1SIDH ⁽⁸⁾	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	58, 296
B1CON ⁽⁸⁾ Receive mode	RXFUL	RXM1	RXRTRRO	FILHIT4	FILHIT3	FILHIT2	FILHIT1	FILHIT0	0000 0000	58, 295
B1CON ⁽⁸⁾ Transmit mode	TXBIF	TXABT	TXLARB	TXERR	TXREQ	RTREN	TXPRI1	TXPRI0	0000 0000	58, 295
B0D7 ⁽⁸⁾	B0D77	B0D76	B0D75	B0D74	B0D73	B0D72	B0D71	B0D70	xxxx xxxx	58, 299
B0D6 ⁽⁸⁾	B0D67	B0D66	B0D65	B0D64	B0D63	B0D62	B0D61	B0D60	xxxx xxxx	58, 299
B0D5 ⁽⁸⁾	B0D57	B0D56	B0D55	B0D54	B0D53	B0D52	B0D51	B0D50	xxxx xxxx	58, 299
B0D4 ⁽⁸⁾	B0D47	B0D46	B0D45	B0D44	B0D43	B0D42	B0D41	B0D40	xxxx xxxx	58, 299
B0D3 ⁽⁸⁾	B0D37	B0D36	B0D35	B0D34	B0D33	B0D32	B0D31	B0D30	xxxx xxxx	58, 299
B0D2 ⁽⁸⁾	B0D27	B0D26	B0D25	B0D24	B0D23	B0D22	B0D21	B0D20	xxxx xxxx	58, 299
B0D1 ⁽⁸⁾	B0D17	B0D16	B0D15	B0D14	B0D13	B0D12	B0D11	B0D10	xxxx xxxx	58, 299
B0D0 ⁽⁸⁾	B0D07	B0D06	B0D05	B0D04	B0D03	B0D02	B0D01	B0D00	xxxx xxxx	58, 299

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

Note 1: Bit 21 of the PC is only available in Test mode and Serial Programming modes.

2: The SBOREN bit is only available when CONFIG2L<1:0> = 01; otherwise, it is disabled and reads as '0'. See **Section 4.4 “Brown-out Reset (BOR)”**.

3: These registers and/or bits are not implemented on PIC18F2X8X devices and are read as '0'. Reset values are shown for PIC18F4X8X devices; individual unimplemented bits should be interpreted as '—'.

4: The PLEN bit is only available in specific oscillator configuration; otherwise, it is disabled and reads as '0'. See **Section 2.6.4 “PLL in INTOSC Modes”**.

5: The RE3 bit is only available when Master Clear Reset is disabled (CONFIG3H<7> = 0); otherwise, RE3 reads as '0'. This bit is read-only.

6: RA6/RA7 and their associated latch and direction bits are individually configured as port pins based on various primary oscillator modes. When disabled, these bits read as '0'.

7: CAN bits have multiple functions depending on the selected mode of the CAN module.

8: This register reads all '0's until the ECAN™ technology is set up in Mode 1 or Mode 2.

9: These registers are available on PIC18F4X8X devices only.

PIC18F2585/2680/4585/4680

5.4 Data Addressing Modes

Note: The execution of some instructions in the core PIC18 instruction set are changed when the PIC18 extended instruction set is enabled. See **Section 5.6 “Data Memory and the Extended Instruction Set”** for more information.

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

The addressing modes are:

- Inherent
- Literal
- Direct
- Indirect

An additional addressing mode, Indexed Literal Offset, is available when the extended instruction set is enabled (XINST Configuration bit = 1). Its operation is discussed in greater detail in **Section 5.6.1 “Indexed Addressing with Literal Offset”**.

5.4.1 INHERENT AND LITERAL ADDRESSING

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

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

5.4.2 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 5.3.3 “General**

Purpose Register File”) or a location in the Access Bank (**Section 5.3.2 “Access Bank”**) as the data source for the instruction.

The Access RAM bit ‘a’ determines how the address is interpreted. When ‘a’ is ‘1’, the contents of the BSR (**Section 5.3.1 “Bank Select Register (BSR)”**) are used with the address to determine the complete 12-bit address of the register. When ‘a’ is ‘0’, the address is interpreted as being a register in the Access Bank. Addressing that uses the Access RAM is sometimes also known as Direct Forced Addressing mode.

A few instructions, such as `MOVFF`, include the entire 12-bit address (either source or destination) in their opcodes. In those 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.

5.4.3 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 to be read or written to. 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 5-5.

EXAMPLE 5-5: 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

PIC18F2585/2680/4585/4680

17.3.2 OPERATION

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bits (SSPCON1<5:0> and SSPSTAT<7:6>). These control bits allow the following to be specified:

- Master mode (SCK is the clock output)
- Slave mode (SCK is the clock input)
- Clock Polarity (Idle state of SCK)
- Data Input Sample Phase (middle or end of data output time)
- Clock Edge (output data on rising/falling edge of SCK)
- Clock Rate (Master mode only)
- Slave Select mode (Slave mode only)

The MSSP consists of a transmit/receive shift register (SSPSR) and a buffer register (SSPBUF). The SSPSR shifts the data in and out of the device, MSb first. The SSPBUF holds the data that was written to the SSPSR until the received data is ready. Once the 8 bits of data have been received, that byte is moved to the SSPBUF register. Then, the Buffer Full detect bit, BF (SSPSTAT<0>) and the interrupt flag bit, SSPIF, are set. This double-buffering of the received data (SSPBUF) allows the next byte to start reception before

reading the data that was just received. Any write to the SSPBUF register during transmission/reception of data will be ignored and the write collision detect bit, WCOL (SSPCON1<7>), will be set. User software must clear the WCOL bit so that it can be determined if the following write(s) to the SSPBUF register completed successfully.

When the application software is expecting to receive valid data, the SSPBUF should be read before the next byte of data to transfer is written to the SSPBUF. The Buffer Full bit, BF (SSPSTAT<0>), indicates when SSPBUF has been loaded with the received data (transmission is complete). When the SSPBUF is read, the BF bit is cleared. This data may be irrelevant if the SPI is only a transmitter. Generally, the MSSP interrupt is used to determine when the transmission/reception has completed. The SSPBUF must be read and/or written. If the interrupt method is not going to be used, then software polling can be done to ensure that a write collision does not occur. Example 17-1 shows the loading of the SSPBUF (SSPSR) for data transmission.

The SSPSR is not directly readable or writable and can only be accessed by addressing the SSPBUF register. Additionally, the MSSP status register (SSPSTAT) indicates the various status conditions.

EXAMPLE 17-1: LOADING THE SSPBUF (SSPSR) REGISTER

LOOP	BTFSS	SSPSTAT, BF	;Has data been received (transmit complete)?
	BRA	LOOP	;No
	MOVF	SSPBUF, W	;WREG reg = contents of SSPBUF
	MOVWF	RXDATA	;Save in user RAM, if data is meaningful
	MOVF	TXDATA, W	;W reg = contents of TXDATA
	MOVWF	SSPBUF	;New data to xmit

PIC18F2585/2680/4585/4680

17.4.2 OPERATION

The MSSP module functions are enabled by setting MSSP Enable bit, SSPEN (SSPCON<5>).

The SSPCON1 register allows control of the I²C operation. Four mode selection bits (SSPCON<3:0>) allow one of the following I²C modes to be selected:

- I²C Master mode, clock = (Fosc/4) x (SSPADD + 1)
- I²C Slave mode (7-bit address)
- I²C Slave mode (10-bit address)
- I²C Slave mode (7-bit address) with Start and Stop bit interrupts enabled
- I²C Slave mode (10-bit address) with Start and Stop bit interrupts enabled
- I²C Firmware Controlled Master mode, slave is Idle

Selection of any I²C mode with the SSPEN bit set, forces the SCL and SDA pins to be open-drain, provided these pins are programmed to inputs by setting the appropriate TRISC bits. To ensure proper operation of the module, pull-up resistors must be provided externally to the SCL and SDA pins.

17.4.3 SLAVE MODE

In Slave mode, the SCL and SDA pins must be configured as inputs (TRISC<4:3> set). The MSSP module will override the input state with the output data when required (slave-transmitter).

The I²C Slave mode hardware will always generate an interrupt on an address match. Through the mode select bits, the user can also choose to interrupt on Start and Stop bits

When an address is matched, or the data transfer after an address match is received, the hardware automatically will generate the Acknowledge (ACK) pulse and load the SSPBUF register with the received value currently in the SSPSR register.

Any combination of the following conditions will cause the MSSP module not to give this ACK pulse:

- The Buffer Full bit, BF (SSPSTAT<0>), was set before the transfer was received.
- The overflow bit, SSPOV (SSPCON<6>), was set before the transfer was received.

In this case, the SSPSR register value is not loaded into the SSPBUF, but bit SSPIF (PIR1<3>) is set. The BF bit is cleared by reading the SSPBUF register, while bit SSPOV is cleared through software.

The SCL clock input must have a minimum high and low for proper operation. The high and low times of the I²C specification, as well as the requirement of the MSSP module, are shown in timing parameter 100 and parameter 101.

17.4.3.1 Addressing

Once the MSSP module has been enabled, it waits for a Start condition to occur. Following the Start condition, the 8-bits are shifted into the SSPSR register. All incoming bits are sampled with the rising edge of the clock (SCL) line. The value of register SSPSR<7:1> is compared to the value of the SSPADD register. The address is compared on the falling edge of the eighth clock (SCL) pulse. If the addresses match and the BF and SSPOV bits are clear, the following events occur:

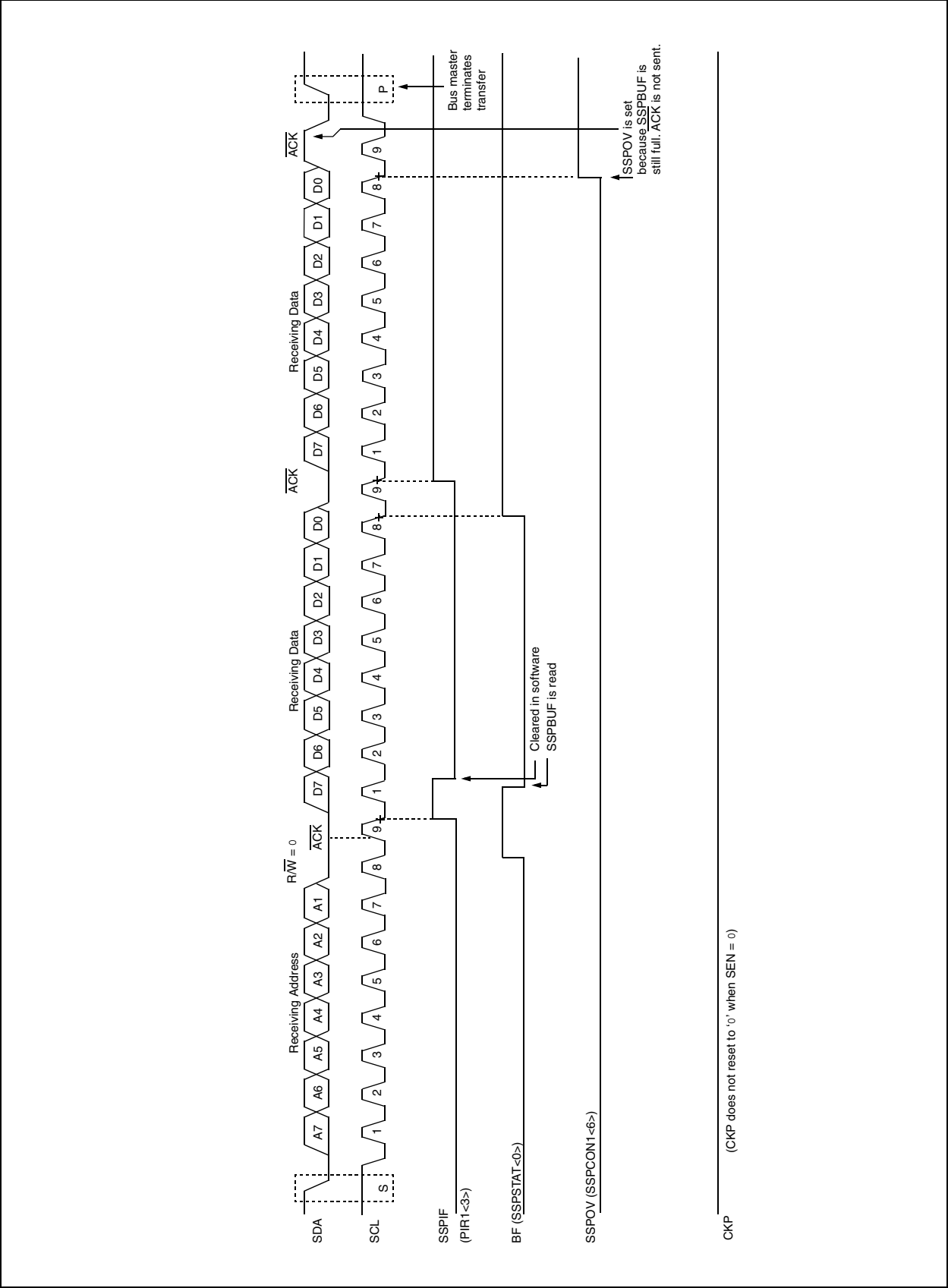
1. The SSPSR register value is loaded into the SSPBUF register.
2. The Buffer Full bit, BF, is set.
3. An ACK pulse is generated.
4. MSSP Interrupt Flag bit, SSPIF (PIR1<3>), is set (interrupt is generated, if enabled) on the falling edge of the ninth SCL pulse.

In 10-bit Address mode, two address bytes need to be received by the slave. The five Most Significant bits (MSBs) of the first address byte specify if this is a 10-bit address. Bit R/W (SSPSTAT<2>) must specify a write so the slave device will receive the second address byte. For a 10-bit address, the first byte would equal '11110 A9 A8 0', where 'A9' and 'A8' are the two MSBs of the address. The sequence of events for 10-bit address is as follows, with steps 7 through 9 for the slave-transmitter:

1. Receive first (high) byte of address (bits SSPIF, BF and UA (SSPSTAT<1>) are set).
2. Update the SSPADD register with second (low) byte of address (clears bit UA and releases the SCL line).
3. Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.
4. Receive second (low) byte of address (bits SSPIF, BF and UA are set).
5. Update the SSPADD register with the first (high) byte of address. If match releases SCL line, this will clear bit UA.
6. Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.
7. Receive Repeated Start condition.
8. Receive first (high) byte of address (bits SSPIF and BF are set).
9. Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.

PIC18F2585/2680/4585/4680

FIGURE 17-8: I²C™ SLAVE MODE TIMING WITH SEN = 0 (RECEPTION, 7-BIT ADDRESS)



19.2 Selecting and Configuring Automatic Acquisition Time

The ADCON2 register allows the user to select an acquisition time that occurs each time the GO/DONE bit is set.

When the GO/DONE bit is set, sampling is stopped and a conversion begins. The user is responsible for ensuring the required acquisition time has passed between selecting the desired input channel and setting the GO/DONE bit. This occurs when the ACQT2:ACQT0 bits (ADCON2<5:3>) remain in their Reset state ('000') and is compatible with devices that do not offer programmable acquisition times.

If desired, the ACQT bits can be set to select a programmable acquisition time for the A/D module. When the GO/DONE bit is set, the A/D module continues to sample the input for the selected acquisition time, then automatically begins a conversion. Since the acquisition time is programmed, there may be no need to wait for an acquisition time between selecting a channel and setting the GO/DONE bit.

In either case, when the conversion is completed, the GO/DONE bit is cleared, the ADIF flag is set and the A/D begins sampling the currently selected channel again. If an acquisition time is programmed, there is nothing to indicate if the acquisition time has ended or if the conversion has begun.

19.3 Selecting the A/D Conversion Clock

The A/D conversion time per bit is defined as TAD. The A/D conversion requires 11 TAD per 10-bit conversion. The source of the A/D conversion clock is software selectable. There are seven possible options for TAD:

- 2 TOSC
- 4 TOSC
- 8 TOSC
- 16 TOSC
- 32 TOSC
- 64 TOSC
- Internal RC Oscillator

For correct A/D conversions, the A/D conversion clock (TAD) must be as short as possible, but greater than the minimum TAD (approximately 2 μ s, see parameter 130 for more information).

Table 19-1 shows the resultant TAD times derived from the device operating frequencies and the A/D clock source selected.

TABLE 19-1: TAD vs. DEVICE OPERATING FREQUENCIES

AD Clock Source (TAD)		Maximum Device Frequency	
Operation	ADCS2:ADCS0	PIC18FX585/X680	PIC18LFX585/X680 ⁽⁴⁾
2 TOSC	000	2.86 MHz	1.43 kHz
4 TOSC	100	5.71 MHz	2.86 MHz
8 TOSC	001	11.43 MHz	5.72 MHz
16 TOSC	101	22.86 MHz	11.43 MHz
32 TOSC	010	40.0 MHz	22.86 MHz
64 TOSC	110	40.0 MHz	22.86 MHz
RC ⁽³⁾	x11	1.00 MHz ⁽¹⁾	1.00 MHz ⁽²⁾

Note 1: The RC source has a typical TAD time of 4 ms.

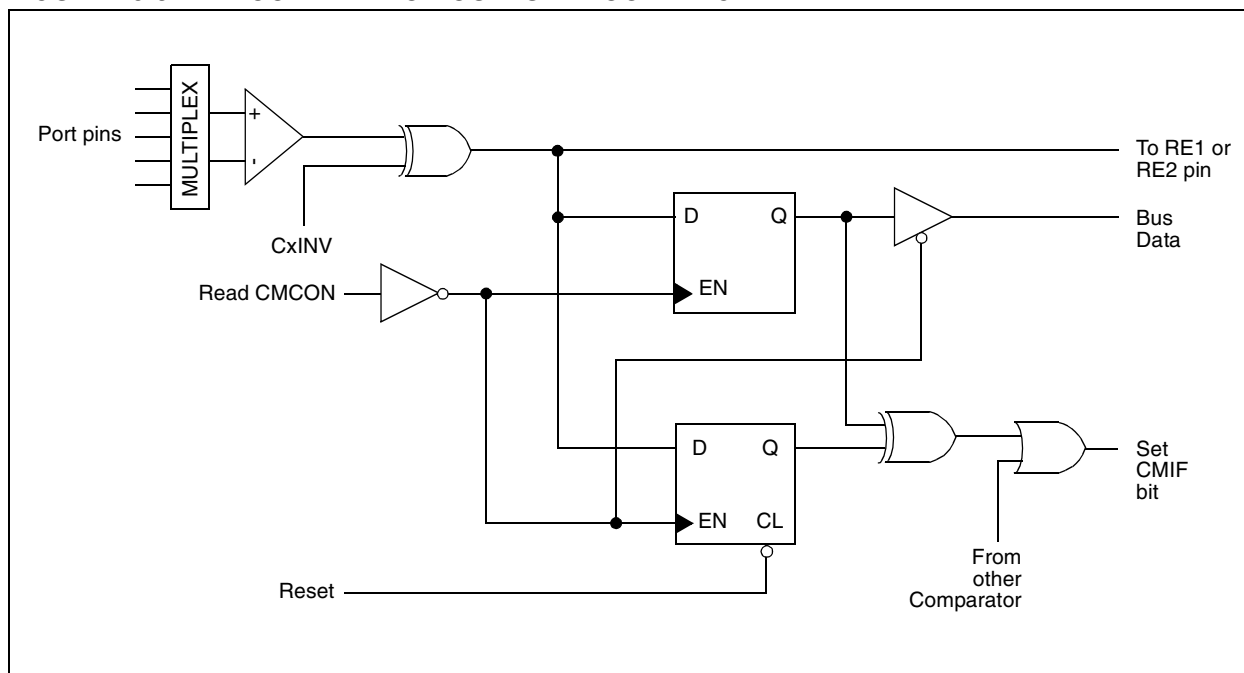
2: The RC source has a typical TAD time of 6 ms.

3: For device frequencies above 1 MHz, the device must be in Sleep for the entire conversion or the A/D accuracy may be out of specification.

4: Low-power (PIC18LFXxxx) devices only.

PIC18F2585/2680/4585/4680

FIGURE 20-3: COMPARATOR OUTPUT BLOCK DIAGRAM



20.6 Comparator Interrupts

The comparator interrupt flag is set whenever there is a change in the output value of either comparator. Software will need to maintain information about the status of the output bits, as read from CMCON<7:6>, to determine the actual change that occurred. The CMIF bit (PIR2<6>) is the Comparator Interrupt Flag. The CMIF bit must be reset by clearing it. Since it is also possible to write a '1' to this register, a simulated interrupt may be initiated.

Both the CMIE bit (PIE2<6>) and the PEIE bit (INTCON<6>) must be set to enable the interrupt. In addition, the GIE bit (INTCON<7>) must also be set. If any of these bits are clear, the interrupt is not enabled, though the CMIF bit will still be set if an interrupt condition occurs.

Note: If a change in the CMCON register (C1OUT or C2OUT) should occur when a read operation is being executed (start of the Q2 cycle), then the CMIF (PIR registers) interrupt flag may not get set.

The user, in the Interrupt Service Routine, can clear the interrupt in the following manner:

- Any read or write of CMCON will end the mismatch condition.
- Clear flag bit CMIF.

A mismatch condition will continue to set flag bit CMIF. Reading CMCON will end the mismatch condition and allow flag bit CMIF to be cleared.

20.7 Comparator Operation During Sleep

When a comparator is active and the device is placed in Sleep mode, the comparator remains active and the interrupt is functional if enabled. This interrupt will wake-up the device from Sleep mode, when enabled. While the comparator is powered up, higher Sleep currents than shown in the power-down current specification will occur. Each operational comparator will consume additional current, as shown in the comparator specifications. To minimize power consumption while in Sleep mode, turn off the comparators (CM2:CM0 = 111) before entering Sleep. If the device wakes up from Sleep, the contents of the CMCON register are not affected.

20.8 Effects of a Reset

A device Reset forces the CMCON register to its Reset state, causing the comparator module to be in the Comparator Reset mode (CM2:CM0 = 000). This ensures that all potential inputs are analog inputs. Device current is minimized when analog inputs are present at Reset time. The comparators are powered down during the Reset interval.

22.6 Operation During Sleep

When enabled, the HLVD circuitry continues to operate during Sleep. If the device voltage crosses the trip point, the HLVDIF bit will be set and the device will wake-up from Sleep. Device execution will continue from the interrupt vector address if interrupts have been globally enabled.

22.7 Effects of a Reset

A device Reset forces all registers to their Reset state. This forces the HLVD module to be turned off.

TABLE 22-1: REGISTERS ASSOCIATED WITH HIGH/LOW-VOLTAGE DETECT MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page
HLVDCON	VDIRMAG	—	IRVST	HLVDEN	HLVDL3	HLVDL2	HLVDL1	HLVDL0	50
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	49
PIR2	OSCFIF	CMIF	—	EEIF	BCLIF	HLVDIF	TMR3IF	ECCP1IF	51
PIE2	OSCFIE	CMIE	—	EEIE	BCLIE	HLVDIE	TMR3IE	ECCP1IE	52
IPR2	OSCFIP	CMIP	—	EEIP	BCLIP	HLVDIP	TMR3IP	ECCP1IP	51

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

PIC18F2585/2680/4585/4680

REGISTER 23-39: RXFnEIDH: RECEIVE ACCEPTANCE FILTER n EXTENDED IDENTIFIER REGISTERS, HIGH BYTE $[0 \leq n \leq 15]^{(1)}$

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8

bit 7

bit 0

bit 7-0

EID15:EID8: Extended Identifier Filter bits

Note 1: Registers RXF6EIDH:RXF15EIDH are 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

REGISTER 23-40: RXFnEIDL: RECEIVE ACCEPTANCE FILTER n EXTENDED IDENTIFIER REGISTERS, LOW BYTE $[0 \leq n \leq 15]^{(1)}$

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0

bit 7

bit 0

bit 7-0

EID7:EID0: Extended Identifier Filter bits

Note 1: Registers RXF6EIDL:RXF15EIDL are 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

REGISTER 23-41: RXMnSIDH: RECEIVE ACCEPTANCE MASK n STANDARD IDENTIFIER MASK REGISTERS, HIGH BYTE $[0 \leq n \leq 1]$

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3

bit 7

bit 0

bit 7-0

SID10:SID3: Standard Identifier Mask bits or Extended Identifier Mask bits EID28:EID21

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

PIC18F2585/2680/4585/4680

REGISTER 23-42: RXMnSIDL: RECEIVE ACCEPTANCE MASK n STANDARD IDENTIFIER MASK REGISTERS, LOW BYTE [$0 \leq n \leq 1$]

R/W-x	R/W-x	R/W-x	U-0	R/W-0	U-0	R/W-x	R/W-x
SID2	SID1	SID0	—	EXIDEN ⁽¹⁾	—	EID17	EID16
bit 7				bit 0			

bit 7-5 **SID2:SID0:** Standard Identifier Mask bits or Extended Identifier Mask bits EID20:EID18

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

bit 3 Mode 0:

Unimplemented: Read as '0'

Mode 1, 2:

EXIDEN: Extended Identifier Filter Enable Mask bit⁽¹⁾

1 = Messages selected by the EXIDEN bit in RXFnSIDL will be accepted

0 = Both standard and extended identifier messages will be accepted

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

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

bit 1-0 **EID17:EID16:** Extended Identifier Mask bits

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

REGISTER 23-43: RXMnEIDH: RECEIVE ACCEPTANCE MASK n EXTENDED IDENTIFIER MASK REGISTERS, HIGH BYTE [$0 \leq n \leq 1$]

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8
bit 7				bit 0			

bit 7-0 **EID15:EID8:** Extended Identifier Mask bits

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

REGISTER 23-44: RXMnEIDL: RECEIVE ACCEPTANCE MASK n EXTENDED IDENTIFIER MASK REGISTERS, LOW BYTE [$0 \leq n \leq 1$]

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0
bit 7				bit 0			

bit 7-0 **EID7:EID0:** Extended Identifier Mask bits

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

PIC18F2585/2680/4585/4680

23.2.6 CAN INTERRUPT REGISTERS

The registers in this section are the same as described in **Section 9.0 “Interrupts”**. They are duplicated here for convenience.

REGISTER 23-56: PIR3: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 3

Mode 0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	IRXIF	WAKIF	ERRIF	TXB2IF	TXB1IF ⁽¹⁾	TXB0IF ⁽¹⁾	RXB1IF	RXB0IF
Mode 1, 2	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	IRXIF	WAKIF	ERRIF	TXBnIF	TXB1IF ⁽¹⁾	TXB0IF ⁽¹⁾	RXBnIF	FIFOWMIF
bit 7								bit 0

- bit 7 **IRXIF:** CAN Invalid Received Message Interrupt Flag bit
1 = An invalid message has occurred on the CAN bus
0 = No invalid message on CAN bus
- bit 6 **WAKIF:** CAN bus Activity Wake-up Interrupt Flag bit
1 = Activity on CAN bus has occurred
0 = No activity on CAN bus
- bit 5 **ERRIF:** CAN bus Error Interrupt Flag bit
1 = An error has occurred in the CAN module (multiple sources)
0 = No CAN module errors
- bit 4 When CAN is in Mode 0:
TXB2IF: CAN Transmit Buffer 2 Interrupt Flag bit
1 = Transmit Buffer 2 has completed transmission of a message and may be reloaded
0 = Transmit Buffer 2 has not completed transmission of a message
When CAN is in Mode 1 or 2:
TXBnIF: Any Transmit Buffer Interrupt Flag bit
1 = One or more transmit buffers have completed transmission of a message and may be reloaded
0 = No transmit buffer is ready for reload
- bit 3 **TXB1IF:** CAN Transmit Buffer 1 Interrupt Flag bit⁽¹⁾
1 = Transmit Buffer 1 has completed transmission of a message and may be reloaded
0 = Transmit Buffer 1 has not completed transmission of a message
- bit 2 **TXB0IF:** CAN Transmit Buffer 0 Interrupt Flag bit⁽¹⁾
1 = Transmit Buffer 0 has completed transmission of a message and may be reloaded
0 = Transmit Buffer 0 has not completed transmission of a message
- bit 1 When CAN is in Mode 0:
RXB1IF: CAN Receive Buffer 1 Interrupt Flag bit
1 = Receive Buffer 1 has received a new message
0 = Receive Buffer 1 has not received a new message
When CAN is in Mode 1 or 2:
RXBnIF: Any Receive Buffer Interrupt Flag bit
1 = One or more receive buffers has received a new message
0 = No receive buffer has received a new message
- bit 0 When CAN is in Mode 0:
RXB0IF: CAN Receive Buffer 0 Interrupt Flag bit
1 = Receive Buffer 0 has received a new message
0 = Receive Buffer 0 has not received a new message
When CAN is in Mode 1:
Unimplemented: Read as ‘0’
When CAN is in Mode 2:
FIFOWMIF: FIFO Watermark Interrupt Flag bit
1 = FIFO high watermark is reached
0 = FIFO high watermark is not reached

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

23.9.2 TIME QUANTA

As already mentioned, the Time Quanta is a fixed unit derived from the oscillator period and baud rate prescaler. Its relationship to T_{BIT} and the Nominal Bit Rate is shown in Example 23-6.

EXAMPLE 23-6: CALCULATING T_Q, NOMINAL BIT RATE AND NOMINAL BIT TIME

$$T_Q (\mu s) = (2 * (BRP + 1)) / FOSC (MHz)$$

$$T_{BIT} (\mu s) = T_Q (\mu s) * \text{number of } T_Q \text{ per bit interval}$$

$$\text{Nominal Bit Rate (bits/s)} = 1 / T_{BIT}$$

This frequency (FOSC) refers to the effective frequency used. If, for example, a 10 MHz external signal is used along with a PLL, then the effective frequency will be 4 x 10 MHz which equals 40 MHz.

CASE 1:

For FOSC = 16 MHz, BRP<5:0> = 00h and Nominal Bit Time = 8 T_Q:

$$T_Q = (2 * 1) / 16 = 0.125 \mu s (125 \text{ ns})$$

$$T_{BIT} = 8 * 0.125 = 1 \mu s (10^{-6} s)$$

$$\text{Nominal Bit Rate} = 1 / 10^{-6} = 10^6 \text{ bits/s (1 Mb/s)}$$

CASE 2:

For FOSC = 20 MHz, BRP<5:0> = 01h and Nominal Bit Time = 8 T_Q:

$$T_Q = (2 * 2) / 20 = 0.2 \mu s (200 \text{ ns})$$

$$T_{BIT} = 8 * 0.2 = 1.6 \mu s (1.6 * 10^{-6} s)$$

$$\text{Nominal Bit Rate} = 1 / 1.6 * 10^{-6} s = 625,000 \text{ bits/s (625 Kb/s)}$$

CASE 3:

For FOSC = 25 MHz, BRP<5:0> = 3Fh and Nominal Bit Time = 25 T_Q:

$$T_Q = (2 * 64) / 25 = 5.12 \mu s$$

$$T_{BIT} = 25 * 5.12 = 128 \mu s (1.28 * 10^{-4} s)$$

$$\text{Nominal Bit Rate} = 1 / 1.28 * 10^{-4} = 7813 \text{ bits/s (7.8 Kb/s)}$$

The frequencies of the oscillators in the different nodes must be coordinated in order to provide a system wide specified nominal bit time. This means that all oscillators must have a T_{OSC} that is an integral divisor of T_Q. It should also be noted that although the number of T_Q is programmable from 4 to 25, the usable minimum is 8 T_Q. There is no assurance that a bit time of less than 8 T_Q in length will operate correctly.

23.9.3 SYNCHRONIZATION SEGMENT

This part of the bit time is used to synchronize the various CAN nodes on the bus. The edge of the input signal is expected to occur during the sync segment. The duration is 1 T_Q.

23.9.4 PROPAGATION SEGMENT

This part of the bit time is used to compensate for physical delay times within the network. These delay times consist of the signal propagation time on the bus line and the internal delay time of the nodes. The length of the Propagation Segment can be programmed from 1 T_Q to 8 T_Q by setting the PRSEG2:PRSEG0 bits.

23.9.5 PHASE BUFFER SEGMENTS

The phase buffer segments are used to optimally locate the sampling point of the received bit within the nominal bit time. The sampling point occurs between Phase Segment 1 and Phase Segment 2. These segments can be lengthened or shortened by the resynchronization process. The end of Phase Segment 1 determines the sampling point within a bit time. Phase Segment 1 is programmable from 1 T_Q to 8 T_Q in duration. Phase Segment 2 provides delay before the next transmitted data transition and is also programmable from 1 T_Q to 8 T_Q in duration. However, due to IPT requirements, the actual minimum length of Phase Segment 2 is 2 T_Q, or it may be defined to be equal to the greater of Phase Segment 1 or the Information Processing Time (IPT). The sampling point should be as late as possible or approximately 80% of the bit time.

23.9.6 SAMPLE POINT

The sample point is the point of time at which the bus level is read and the value of the received bit is determined. The sampling point occurs at the end of Phase Segment 1. If the bit timing is slow and contains many T_Q, it is possible to specify multiple sampling of the bus line at the sample point. The value of the received bit is determined to be the value of the majority decision of three values. The three samples are taken at the sample point and twice before, with a time of T_Q/2 between each sample.

23.9.7 INFORMATION PROCESSING TIME

The Information Processing Time (IPT) is the time segment starting at the sample point that is reserved for calculation of the subsequent bit level. The CAN specification defines this time to be less than or equal to 2 T_Q. The PIC18F2585/2680/4585/4680 devices define this time to be 2 T_Q. Thus, Phase Segment 2 must be at least 2 T_Q long.

23.13 Bit Timing Configuration Registers

The Baud Rate Control registers (BRGCON1, BRGCON2, BRGCON3) control the bit timing for the CAN bus interface. These registers can only be modified when the PIC18F2585/2680/4585/4680 devices are in Configuration mode.

23.13.1 BRGCON1

The BRP bits control the baud rate prescaler. The SJW<1:0> bits select the synchronization jump width in terms of multiples of T_Q.

23.13.2 BRGCON2

The PRSEG bits set the length of the propagation segment in terms of T_Q. The SEG1PH bits set the length of Phase Segment 1 in T_Q. The SAM bit controls how many times the RXCAN pin is sampled. Setting this bit to a '1' causes the bus to be sampled three times: twice at T_Q/2 before the sample point and once at the normal sample point (which is at the end of Phase Segment 1). The value of the bus is determined to be the value read during at least two of the samples. If the SAM bit is set to a '0', then the RXCAN pin is sampled only once at the sample point. The SEG2PHTS bit controls how the length of Phase Segment 2 is determined. If this bit is set to a '1', then the length of Phase Segment 2 is determined by the SEG2PH bits of BRGCON3. If the SEG2PHTS bit is set to a '0', then the length of Phase Segment 2 is the greater of Phase Segment 1 and the Information Processing Time (which is fixed at 2 T_Q for the PIC18F2585/2680/4585/4680).

23.13.3 BRGCON3

The PHSEG2<2:0> bits set the length (in T_Q) of Phase Segment 2 if the SEG2PHTS bit is set to a '1'. If the SEG2PHTS bit is set to a '0', then the PHSEG2<2:0> bits have no effect.

23.14 Error Detection

The CAN protocol provides sophisticated error detection mechanisms. The following errors can be detected.

23.14.1 CRC ERROR

With the Cyclic Redundancy Check (CRC), the transmitter calculates special check bits for the bit sequence, from the start of a frame until the end of the data field. This CRC sequence is transmitted in the CRC field. The receiving node also calculates the CRC sequence using the same formula and performs a comparison to the received sequence. If a mismatch is detected, a CRC error has occurred and an error frame is generated. The message is repeated.

23.14.2 ACKNOWLEDGE ERROR

In the Acknowledge field of a message, the transmitter checks if the Acknowledge slot (which was sent out as a recessive bit) contains a dominant bit. If not, no other node has received the frame correctly. An Acknowledge error has occurred, an error frame is generated and the message will have to be repeated.

23.14.3 FORM ERROR

If a node detects a dominant bit in one of the four segments, including End-Of-Frame, interframe space, Acknowledge delimiter or CRC delimiter, then a form error has occurred and an error frame is generated. The message is repeated.

23.14.4 BIT ERROR

A bit error occurs if a transmitter sends a dominant bit and detects a recessive bit, or if it sends a recessive bit and detects a dominant bit, when monitoring the actual bus level and comparing it to the just transmitted bit. In the case where the transmitter sends a recessive bit and a dominant bit is detected during the arbitration field and the Acknowledge slot, no bit error is generated because normal arbitration is occurring.

23.14.5 STUFF BIT ERROR

If, between the Start-Of-Frame and the CRC delimiter, six consecutive bits with the same polarity are detected, the bit stuffing rule has been violated. A stuff bit error occurs and an error frame is generated. The message is repeated.

23.14.6 ERROR STATES

Detected errors are made public to all other nodes via error frames. The transmission of the erroneous message is aborted and the frame is repeated as soon as possible. Furthermore, each CAN node is in one of the three error states: "error-active", "error-passive" or "bus-off", according to the value of the internal error counters. The error-active state is the usual state where the bus node can transmit messages and activate error frames (made of dominant bits) without any restrictions. In the error-passive state, messages and passive error frames (made of recessive bits) may be transmitted. The bus-off state makes it temporarily impossible for the station to participate in the bus communication. During this state, messages can neither be received nor transmitted.

23.14.7 ERROR MODES AND ERROR COUNTERS

The PIC18F2585/2680/4585/4680 devices contain two error counters: the Receive Error Counter (RXERRCNT) and the Transmit Error Counter (TXERRCNT). The values of both counters can be read by the MCU. These counters are incremented or decremented in accordance with the CAN bus specification.

PIC18F2585/2680/4585/4680

24.5 Program Verification and Code Protection

The overall structure of the code protection on the PIC18 Flash devices differs significantly from other PIC® devices.

The user program memory is divided into five blocks. One of these is a boot block of 2 Kbytes. The remainder of the memory is divided into four blocks on binary boundaries.

Each of the five blocks has three code protection bits associated with them. They are:

- Code-Protect bit (CPn)
- Write-Protect bit (WRTn)
- External Block Table Read bit (EBTRn)

Figure 24-5 shows the program memory organization for 48 and 64-Kbyte devices and the specific code protection bit associated with each block. The actual locations of the bits are summarized in Table 24-3.

FIGURE 24-5: CODE-PROTECTED PROGRAM MEMORY FOR PIC18F2585/2680/4585/4680

MEMORY SIZE/DEVICE		Address Range	Block Code Protection Controlled By:
48 Kbytes (PIC18F2585/4585)	64 Kbytes (PIC18F2680/4680)		
Boot Block	Boot Block	000000h 0007FFh	CPB, WRTB, EBTRB
Block 0	Block 0	000800h 003FFFh	CP0, WRT0, EBTR0
Block 1	Block 1	004000h 007FFFh	CP1, WRT1, EBTR1
Block 2	Block 2	008000h 00B7FFh	CP2, WRT2, EBTR2
Unimplemented Read '0's	Block 3	00C000h 00FFFFh	CP3, WRT3, EBTR3
Unimplemented Read '0's	Unimplemented Read '0's	010000h 1FFFFFFh	(Unimplemented Memory Space)

TABLE 24-3: SUMMARY OF CODE PROTECTION REGISTERS

File Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
300008h	CONFIG5L	—	—	—	—	CP3*	CP2	CP1	CP0
300009h	CONFIG5H	CPD	CPB	—	—	—	—	—	—
30000Ah	CONFIG6L	—	—	—	—	WRT3*	WRT2	WRT1	WRT0
30000Bh	CONFIG6H	WRTD	WRTB	WRTC	—	—	—	—	—
30000Ch	CONFIG7L	—	—	—	—	EBTR3*	EBTR2	EBTR1	EBTR0
30000Dh	CONFIG7H	—	EBTRB	—	—	—	—	—	—

Legend: Shaded cells are unimplemented.

* Unimplemented in PIC18FX585 devices; maintain this bit set.

PIC18F2585/2680/4585/4680

BZ Branch if Zero

Syntax:	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:	<p>If the Zero bit is '1', then the program will branch.</p> <p>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.</p>			
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:	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:	<table><tr><td>1110</td><td>110s</td><td>k₇kkk</td><td>kkkk₀</td></tr><tr><td>1111</td><td>k₁₉kkk</td><td>kkkk</td><td>kkkk₈</td></tr></table>				1110	110s	k ₇ kkk	kkkk ₀	1111	k ₁₉ kkk	kkkk	kkkk ₈
1110	110s	k ₇ kkk	kkkk ₀									
1111	k ₁₉ kkk	kkkk	kkkk ₈									
1st word (k<7:0>)												
2nd word (k<19:8>)												

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	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

PIC18F2585/2680/4585/4680

TBLWT

Table Write

Syntax: TBLWT (*; *+; *-; +*)

Operands: None

Operation:

- if TBLWT*,
(TABLAT) → Holding Register;
TBLPTR – No Change;
- if TBLWT*+,
(TABLAT) → Holding Register;
(TBLPTR) + 1 → TBLPTR;
- if TBLWT*-,
(TABLAT) → Holding Register;
(TBLPTR) – 1 → TBLPTR;
- if TBLWT*+*,
(TBLPTR) + 1 → TBLPTR;
(TABLAT) → Holding Register;

Status Affected: None

Encoding:

0000	0000	0000	11nn nn=0 * =1 *+ =2 *- =3 +*
------	------	------	-------------------------------------------

Description: This instruction uses the 3 LSBs of the TBLPTR to determine which of the 8 holding registers the TABLAT is written to. The holding registers are used to program the contents of Program Memory (P.M.). (Refer to **Section 6.0 “Flash Program Memory”** for additional details on programming Flash memory.)

The TBLPTR (a 21-bit pointer) points to each byte in the program memory. TBLPTR has a 2-MByte address range. The LSb of the TBLPTR selects which byte of the program memory location to access.

TBLPTR[0] = 0: Least Significant Byte of Program Memory Word

TBLPTR[0] = 1: Most Significant Byte of Program Memory Word

The TBLWT instruction can modify the value of TBLPTR as follows:

- no change
- post-increment
- post-decrement
- pre-increment

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	No operation	No operation
No operation	No operation (Read TABLAT)	No operation	No operation (Write to Holding Register)

TBLWT

Table Write (Continued)

Example 1: TBLWT *+;

Before Instruction

TABLAT	=	55h
TBLPTR	=	00A356h
HOLDING REGISTER (00A356h)	=	FFh

After Instructions (table write completion)

TABLAT	=	55h
TBLPTR	=	00A357h
HOLDING REGISTER (00A356h)	=	55h

Example 2: TBLWT +*;

Before Instruction

TABLAT	=	34h
TBLPTR	=	01389Ah
HOLDING REGISTER (01389Ah)	=	FFh
HOLDING REGISTER (01389Bh)	=	FFh

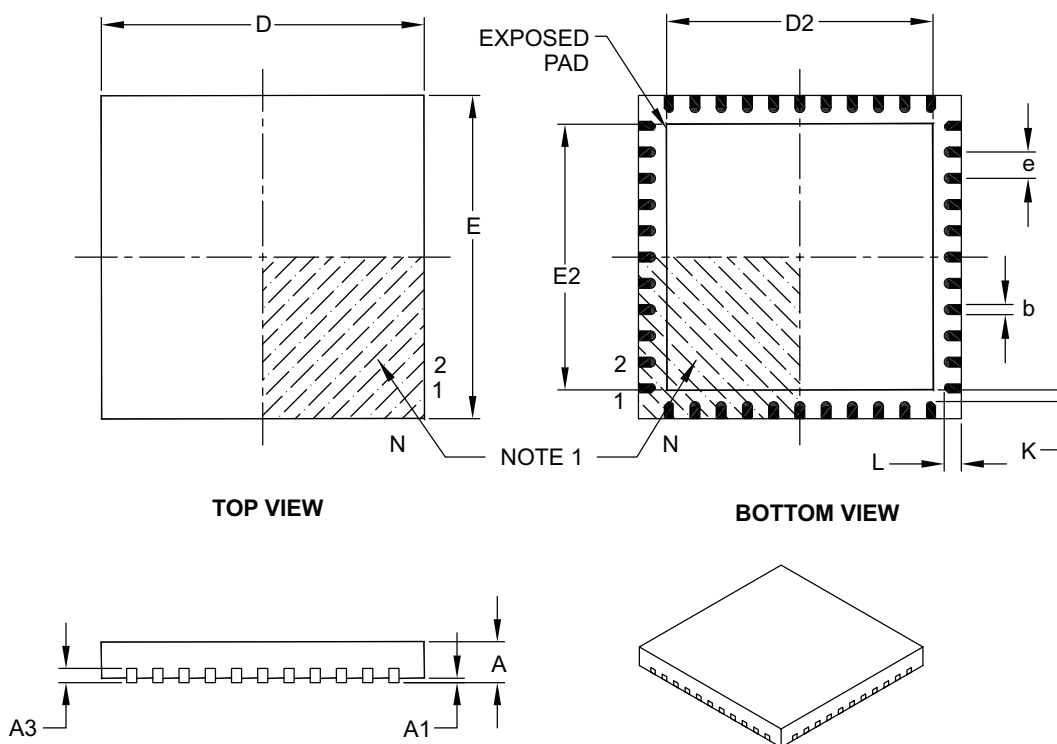
After Instruction (table write completion)

TABLAT	=	34h
TBLPTR	=	01389Bh
HOLDING REGISTER (01389Ah)	=	FFh
HOLDING REGISTER (01389Bh)	=	34h

PIC18F2585/2680/4585/4680

44-Lead Plastic Quad Flat, No Lead Package (ML) – 8x8 mm Body [QFN]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N	44		
Pitch	e	0.65 BSC		
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3	0.20 REF		
Overall Width	E	8.00 BSC		
Exposed Pad Width	E2	6.30	6.45	6.80
Overall Length	D	8.00 BSC		
Exposed Pad Length	D2	6.30	6.45	6.80
Contact Width	b	0.25	0.30	0.38
Contact Length	L	0.30	0.40	0.50
Contact-to-Exposed Pad	K	0.20	—	—

Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Package is saw singulated.
- Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-103B

PIC18F2585/2680/4585/4680

PIC18F2585/2680/4585/4680 PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

<u>PART NO.</u>	<u>X</u>	<u>/XX</u>	<u>XXX</u>
Device	Temperature Range	Package	Pattern
Device	PIC18F2585/2680 ⁽¹⁾ , PIC18F4585/4680 ⁽¹⁾ , PIC18F2585/2680T ⁽²⁾ , PIC18F4585/4680T ⁽²⁾ ; VDD range 4.2V to 5.5V PIC18LF2585/2680 ⁽¹⁾ , PIC18LF4585/4680 ⁽¹⁾ , PIC18LF2585/2680T ⁽²⁾ , PIC18LF4585/4680T ⁽²⁾ ; VDD range 2.0V to 5.5V		
Temperature Range	I = -40°C to +85°C (Industrial) E = -40°C to +125°C (Extended)		
Package	PT = TQFP (Thin Quad Flatpack) SO = SOIC SP = Skinny Plastic DIP P = PDIP ML = QFN		
Pattern	QTP, SQTP, Code or Special Requirements (blank otherwise)		

Examples:

- a) PIC18LF4680-I/P 301 = Industrial temp., PDIP package, Extended VDD limits, QTP pattern #301.
- b) PIC18LF2585-I/SO = Industrial temp., SOIC package, Extended VDD limits.
- c) PIC18F4585-I/P = Industrial temp., PDIP package, normal VDD limits.

Note 1: F = Standard Voltage Range
LF = Wide Voltage Range
2: T = in tape and reel PLCC and TQFP packages only.