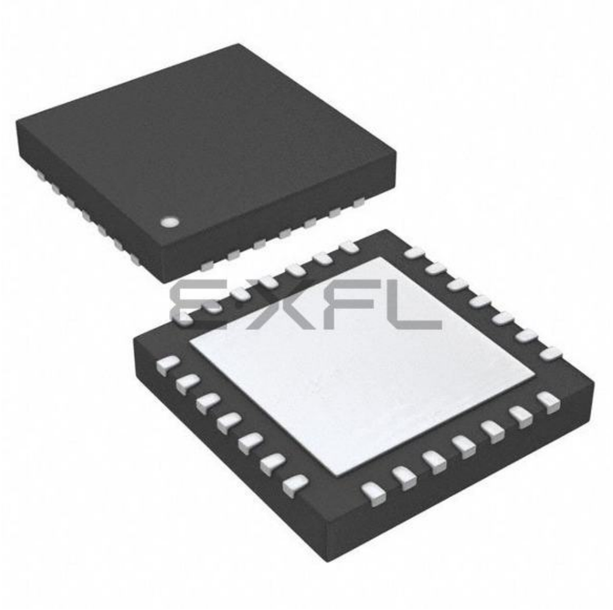
Microchip Technology - PIC18LF2580-I/ML Datasheet



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, I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, HLVD, POR, PWM, WDT
Number of I/O	25
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	1.5K x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5.5V
Data Converters	A/D 8x10b
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/pic18lf2580-i-ml

Table of Contents

NOTES:

TABLE 5-4: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices				Power-on Reset, Brown-out Reset	$\overline{\text{MCLR}}$ Resets, WDT Reset, RESET Instruction, Stack Resets	Wake-up via WDT or Interrupt
BRGCON3	2480	2580	4480	4580	00-- -000	00-- -000	uu-- -uuu
BRGCON2	2480	2580	4480	4580	0000 0000	0000 0000	uuuu uuuu
BRGCON1	2480	2580	4480	4580	0000 0000	0000 0000	uuuu uuuu
CANCON	2480	2580	4480	4580	1000 000-	1000 000-	uuuu uuu-
CANSTAT	2480	2580	4480	4580	100- 000-	100- 000-	uuu- uuu-
RXB0D7	2480	2580	4480	4580	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0D6	2480	2580	4480	4580	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0D5	2480	2580	4480	4580	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0D4	2480	2580	4480	4580	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0D3	2480	2580	4480	4580	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0D2	2480	2580	4480	4580	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0D1	2480	2580	4480	4580	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0D0	2480	2580	4480	4580	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0DLC	2480	2580	4480	4580	-xxx xxxx	-uuu uuuu	-uuu uuuu
RXB0EIDL	2480	2580	4480	4580	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0EIDH	2480	2580	4480	4580	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0SIDL	2480	2580	4480	4580	xxxx x-xx	uuuu u-uu	uuuu u-uu
RXB0SIDH	2480	2580	4480	4580	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0CON	2480	2580	4480	4580	000- 0000	000- 0000	uuu- uuuu
RXB1D7	2480	2580	4480	4580	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1D6	2480	2580	4480	4580	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1D5	2480	2580	4480	4580	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1D4	2480	2580	4480	4580	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1D3	2480	2580	4480	4580	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1D2	2480	2580	4480	4580	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1D1	2480	2580	4480	4580	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1D0	2480	2580	4480	4580	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1DLC	2480	2580	4480	4580	-xxx xxxx	-uuu uuuu	-uuu uuuu
RXB1EIDL	2480	2580	4480	4580	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1EIDH	2480	2580	4480	4580	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1SIDL	2480	2580	4480	4580	xxxx x-xx	uuuu u-uu	uuuu u-uu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition. Shaded cells indicate conditions do not apply for the designated device.

Note 1: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

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: 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.

4: See Table 5-3 for Reset value for specific condition.

5: Bits 6 and 7 of PORTA, LATA and TRISA are enabled, depending on the oscillator mode selected. When not enabled as PORTA pins, they are disabled and read '0'.

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

6.0 MEMORY ORGANIZATION

There are three types of memory in PIC18 Enhanced microcontroller devices:

- Program Memory
- Data RAM
- Data EEPROM

As Harvard architecture devices, the data and program memories use separate busses; this allows for concurrent access of the two memory spaces. The data EEPROM, for practical purposes, can be regarded as a peripheral device, since it is addressed and accessed through a set of control registers.

Additional detailed information on the operation of the Flash program memory is provided in **Section 7.0 "Flash Program Memory"**. Data EEPROM is discussed separately in **Section 8.0 "Data EEPROM Memory"**.

6.1 Program Memory Organization

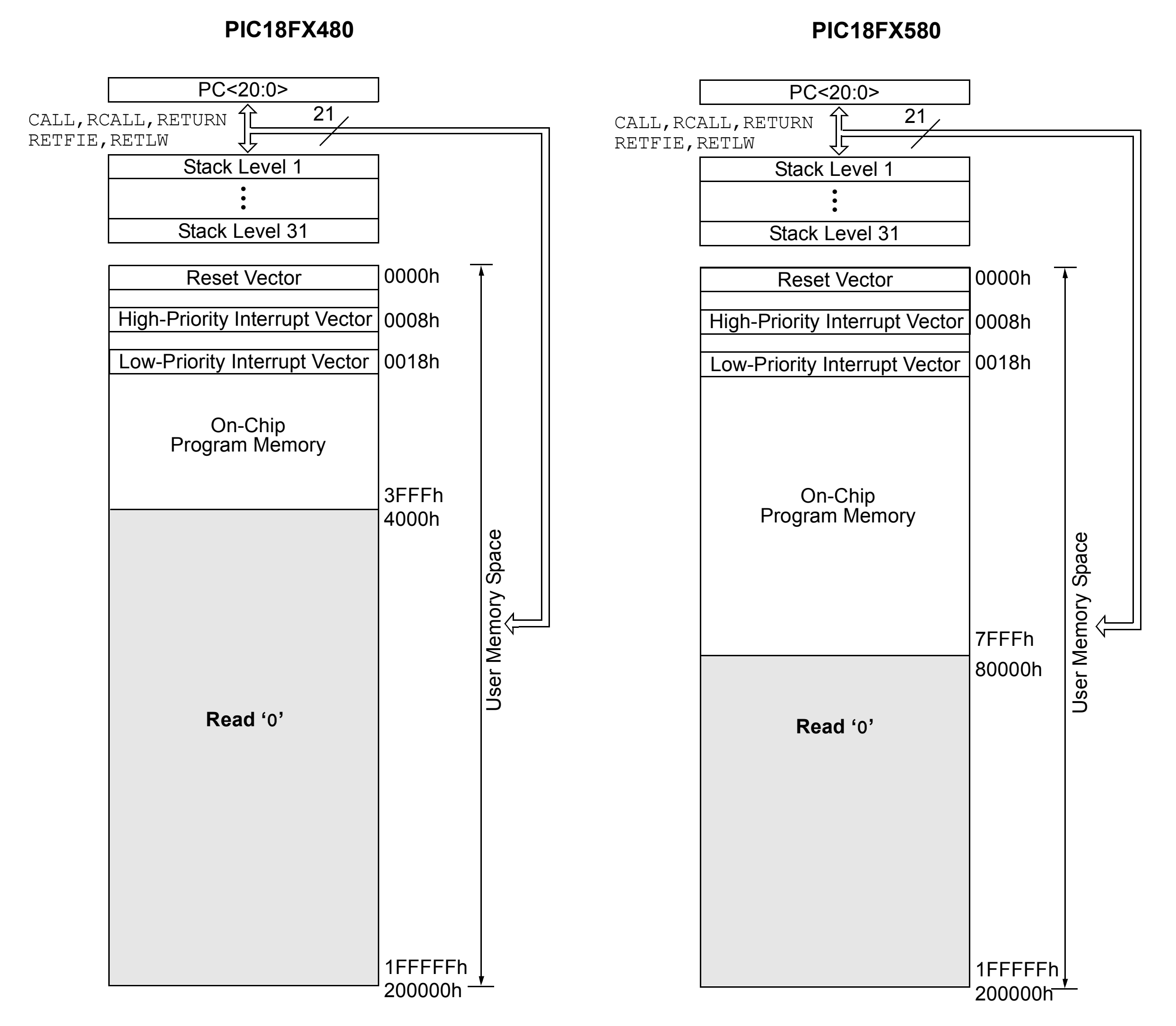
PIC18 microcontrollers implement a 21-bit program counter, which is capable of addressing a 2-Mbyte program memory space. Accessing a location between upper boundary of the physically implemented memory and the 2-Mbyte address will return all '0's (a NOP instruction).

The PIC18F2480 and PIC18F4480 each have 16 Kbytes of Flash memory and can store up to 8,192 single-word instructions. The PIC18F2580 and PIC18F4580 each have 32 Kbytes of Flash memory and can store up to 16,384 single-word instructions.

PIC18 devices have two interrupt vectors. The Reset vector address is at 0000h and the interrupt vector addresses are at 0008h and 0018h.

The program memory maps for PIC18FX480 and PIC18FX580 devices are shown in Figure 6-1.

FIGURE 6-1: PROGRAM MEMORY MAP AND STACK FOR PIC18F2480/2580/4480/4580 DEVICES



6.3 Data Memory Organization

Note:	The operation of some aspects of data memory are changed when the PIC18 extended instruction set is enabled. See **Section 6.6 “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; PIC18F2480/2580/4480/4580 devices implement all 16 banks. Figure 6-6 shows the data memory organization for the PIC18F2480/2580/4480/4580 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 6.3.2 “Access Bank”** provides a detailed description of the Access RAM.

6.3.1 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 (BSR<3:0>). 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 6-7.

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 6-6 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.

TABLE 6-2: REGISTER FILE SUMMARY (PIC18F2480/2580/4480/4580) (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:
B4D2(8)	B4D27	B4D26	B4D25	B4D24	B4D23	B4D22	B4D21	B4D20	xxxx xxxx	63, 305
B4D1(8)	B4D17	B4D16	B4D15	B4D14	B4D13	B4D12	B4D11	B4D10	xxxx xxxx	63, 305
B4D0(8)	B4D07	B4D06	B4D05	B4D04	B4D03	B4D02	B4D01	B4D00	xxxx xxxx	62, 305
B4DLC(8) Receive mode	—	RXRTR	RB1	RB0	DLC3	DLC2	DLC1	DLC0	-xxx xxxx	63, 307
B4DLC(8) Transmit mode	—	TXRTR	—	—	DLC3	DLC2	DLC1	DLC0	-x-- xxxx	63, 307
B4EIDL(8)	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	63, 305
B4EIDH(8)	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	63, 304
B4SIDL(8) Receive mode	SID2	SID1	SID0	SRR	EXID	—	EID17	EID16	xxxx x-xx	63, 303
B4SIDL(8) Transmit mode	SID2	SID1	SID0	—	EXIDE	—	EID17	EID16	xxx- x-xx	63, 303
B4SIDH(8)	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	63, 302
B4CON(8) Receive mode	RXFUL	RXM1	RXRTRRO	FILHIT4	FILHIT3	FILHIT2	FILHIT1	FILHIT0	0000 0000	63, 301
B4CON(8) Transmit mode	TXBIF	TXABT	TXLARB	TXERR	TXREQ	RTREN	TXPRI1	TXPRI0	0000 0000	63, 301
B3D7(8)	B3D77	B3D76	B3D75	B3D74	B3D73	B3D72	B3D71	B3D70	xxxx xxxx	63, 305
B3D6(8)	B3D67	B3D66	B3D65	B3D64	B3D63	B3D62	B3D61	B3D60	xxxx xxxx	63, 305
B3D5(8)	B3D57	B3D56	B3D55	B3D54	B3D53	B3D52	B3D51	B3D50	xxxx xxxx	63, 305
B3D4(8)	B3D47	B3D46	B3D45	B3D44	B3D43	B3D42	B3D41	B3D40	xxxx xxxx	63, 305
B3D3(8)	B3D37	B3D36	B3D35	B3D34	B3D33	B3D32	B3D31	B3D30	xxxx xxxx	63, 305
B3D2(8)	B3D27	B3D26	B3D25	B3D24	B3D23	B3D22	B3D21	B3D20	xxxx xxxx	63, 305
B3D1(8)	B3D17	B3D16	B3D15	B3D14	B3D13	B3D12	B3D11	B3D10	xxxx xxxx	63, 305
B3D0(8)	B3D07	B3D06	B3D05	B3D04	B3D03	B3D02	B3D01	B3D00	xxxx xxxx	63, 305
B3DLC(8) Receive mode	—	RXRTR	RB1	RB0	DLC3	DLC2	DLC1	DLC0	-xxx xxxx	63, 307
B3DLC(8) Transmit mode	—	TXRTR	—	—	DLC3	DLC2	DLC1	DLC0	-x-- xxxx	63, 307
B3EIDL(8)	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	63, 305
B3EIDH(8)	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	63, 304
B3SIDL(8) Receive mode	SID2	SID1	SID0	SRR	EXID	—	EID17	EID16	xxxx x-xx	63, 303
B3SIDL(8) Transmit mode	SID2	SID1	SID0	—	EXIDE	—	EID17	EID16	xxx- x-xx	63, 303
B3SIDH(8)	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	63, 302
B3CON(8) Receive mode	RXFUL	RXM1	RXRTRRO	FILHIT4	FILHIT3	FILHIT2	FILHIT1	FILHIT0	0000 0000	63, 301
B3CON(8) Transmit mode	TXBIF	TXABT	TXLARB	TXERR	TXREQ	RTREN	TXPRI1	TXPRI0	0000 0000	63, 301
B2D7(8)	B2D77	B2D76	B2D75	B2D74	B2D73	B2D72	B2D71	B2D70	xxxx xxxx	63, 305
B2D6(8)	B2D67	B2D66	B2D65	B2D64	B2D63	B2D62	B2D61	B2D60	xxxx xxxx	63, 305
B2D5(8)	B2D57	B2D56	B2D55	B2D54	B2D53	B2D52	B2D51	B2D50	xxxx xxxx	63, 305

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 5.4 "Brown-out Reset (BOR)"**.

3: These registers and/or bits are not implemented on PIC18F2X80 devices and are read as '0'. Reset values are shown for PIC18F4X80 devices; individual unimplemented bits should be interpreted as '—'.

4: The PLLEN bit is only available in specific oscillator configuration; otherwise, it is disabled and reads as '0'. See **Section 3.6.4 "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 PIC18F4X80 devices only.

TABLE 6-2: REGISTER FILE SUMMARY (PIC18F2480/2580/4480/4580) (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:
RXF15EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	65, 309
RXF15SIDL	SID2	SID1	SID0	—	EXIDEN	—	EID17	EID16	xxx- x-xx	65, 308
RXF15SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	65, 309
RXF14EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	65, 309
RXF14EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	65, 309
RXF14SIDL	SID2	SID1	SID0	—	EXIDEN	—	EID17	EID16	xxx- x-xx	65, 308
RXF14SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	65, 309
RXF13EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	66, 309
RXF13EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	66, 309
RXF13SIDL	SID2	SID1	SID0	—	EXIDEN	—	EID17	EID16	xxx- x-xx	66, 308
RXF13SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	66, 309
RXF12EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	66, 309
RXF12EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	66, 309
RXF12SIDL	SID2	SID1	SID0	—	EXIDEN	—	EID17	EID16	xxx- x-xx	66, 308
RXF12SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	66, 309
RXF11EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	66, 309
RXF11EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	66, 309
RXF11SIDL	SID2	SID1	SID0	—	EXIDEN	—	EID17	EID16	xxx- x-xx	66, 308
RXF11SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	66, 309
RXF10EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	66, 309
RXF10EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	66, 309
RXF10SIDL	SID2	SID1	SID0	—	EXIDEN	—	EID17	EID16	xxx- x-xx	66, 308
RXF10SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	66, 309
RXF9EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	66, 309
RXF9EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	66, 309
RXF9SIDL	SID2	SID1	SID0	—	EXIDEN	—	EID17	EID16	xxx- x-xx	66, 308
RXF9SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	66, 309
RXF8EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	66, 309
RXF8EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	66, 309
RXF8SIDL	SID2	SID1	SID0	—	EXIDEN	—	EID17	EID16	xxx- x-xx	66, 308
RXF8SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	66, 309
RXF7EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	66, 309
RXF7EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	66, 309
RXF7SIDL	SID2	SID1	SID0	—	EXIDEN	—	EID17	EID16	xxx- x-xx	66, 308
RXF7SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	66, 309
RXF6EIDL	EID7	EID6	EID5	EID4	EID3	EID2	EID1	EID0	xxxx xxxx	66, 309
RXF6EIDH	EID15	EID14	EID13	EID12	EID11	EID10	EID9	EID8	xxxx xxxx	66, 309
RXF6SIDL	SID2	SID1	SID0	—	EXIDEN	—	EID17	EID16	xxx- x-xx	66, 308
RXF6SIDH	SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3	xxxx xxxx	66, 309

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 5.4 "Brown-out Reset (BOR)"**.

3: These registers and/or bits are not implemented on PIC18F2X80 devices and are read as '0'. Reset values are shown for PIC18F4X80 devices; individual unimplemented bits should be interpreted as '—'.

4: The PLLEN bit is only available in specific oscillator configuration; otherwise, it is disabled and reads as '0'. See **Section 3.6.4 "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 PIC18F4X80 devices only.

REGISTER 10-5: PIR2: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 2

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
OSCFIF	CMIF[(1)]	—	EEIF	BCLIF	HLVDIF	TMR3IF	ECCP1IF[(1)]
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 **OSCFIF:** Oscillator Fail Interrupt Flag bit
1 = System oscillator failed, clock input has changed to INTOSC (must be cleared in software)
0 = System clock operating

bit 6 **CMIF:** Comparator Interrupt Flag bit[(1)]
1 = Comparator input has changed (must be cleared in software)
0 = Comparator input has not changed

bit 5 **Unimplemented:** Read as ‘0’

bit 4 **EEIF:** Data EEPROM/Flash Write Operation Interrupt Flag bit
1 = The write operation is complete (must be cleared in software)
0 = The write operation is not complete, or has not been started

bit 3 **BCLIF:** Bus Collision Interrupt Flag bit
1 = A bus collision occurred (must be cleared in software)
0 = No bus collision occurred

bit 2 **HLVDIF:** High/Low-Voltage Detect Interrupt Flag bit
1 = A low-voltage condition occurred (must be cleared in software)
0 = The device voltage is above the High/Low-Voltage Detect trip point

bit 1 **TMR3IF:** TMR3 Overflow Interrupt Flag bit
1 = TMR3 register overflowed (must be cleared in software)
0 = TMR3 register did not overflow

bit 0 **ECCP1IF:** CCPx Interrupt Flag bit[(1)]
Capture mode:
1 = A TMR1 register capture occurred (must be cleared in software)
0 = No TMR1 register capture occurred
Compare mode:
1 = A TMR1 register compare match occurred (must be cleared in software)
0 = No TMR1 register compare match occurred
PWM mode:
Unused in this mode.

Note 1: These bits are available in PIC18F4X80 and reserved in PIC18F2X80 devices.

18.3.6 SLAVE MODE

In Slave mode, the data is transmitted and received as the external clock pulses appear on SCK. When the last bit is latched, the SSPIF interrupt flag bit is set.

Before enabling the module in SPI Slave mode, the clock line must match the proper Idle state. The clock line can be observed by reading the SCK pin. The Idle state is determined by the CKP bit (SSPCON1<4>).

While in Slave mode, the external clock is supplied by the external clock source on the SCK pin. This external clock must meet the minimum high and low times as specified in the electrical specifications.

While in Sleep mode, the slave can transmit/receive data. When a byte is received, the device will wake-up from Sleep.

18.3.7 SLAVE SELECT SYNCHRONIZATION

The $\overline{SS}$ pin allows a Synchronous Slave mode. The SPI must be in Slave mode with $\overline{SS}$ pin control enabled (SSPCON1<3:0> = 04h). The pin must not be driven low for the $\overline{SS}$ pin to function as an input. The data latch must be high. When the $\overline{SS}$ pin is low, transmission and reception are enabled and the SDO pin is driven. When the $\overline{SS}$ pin goes high, the SDO pin is no longer driven even if in the middle of a transmitted byte and becomes a floating output. External pull-up/pull-down resistors may be desirable depending on the application.

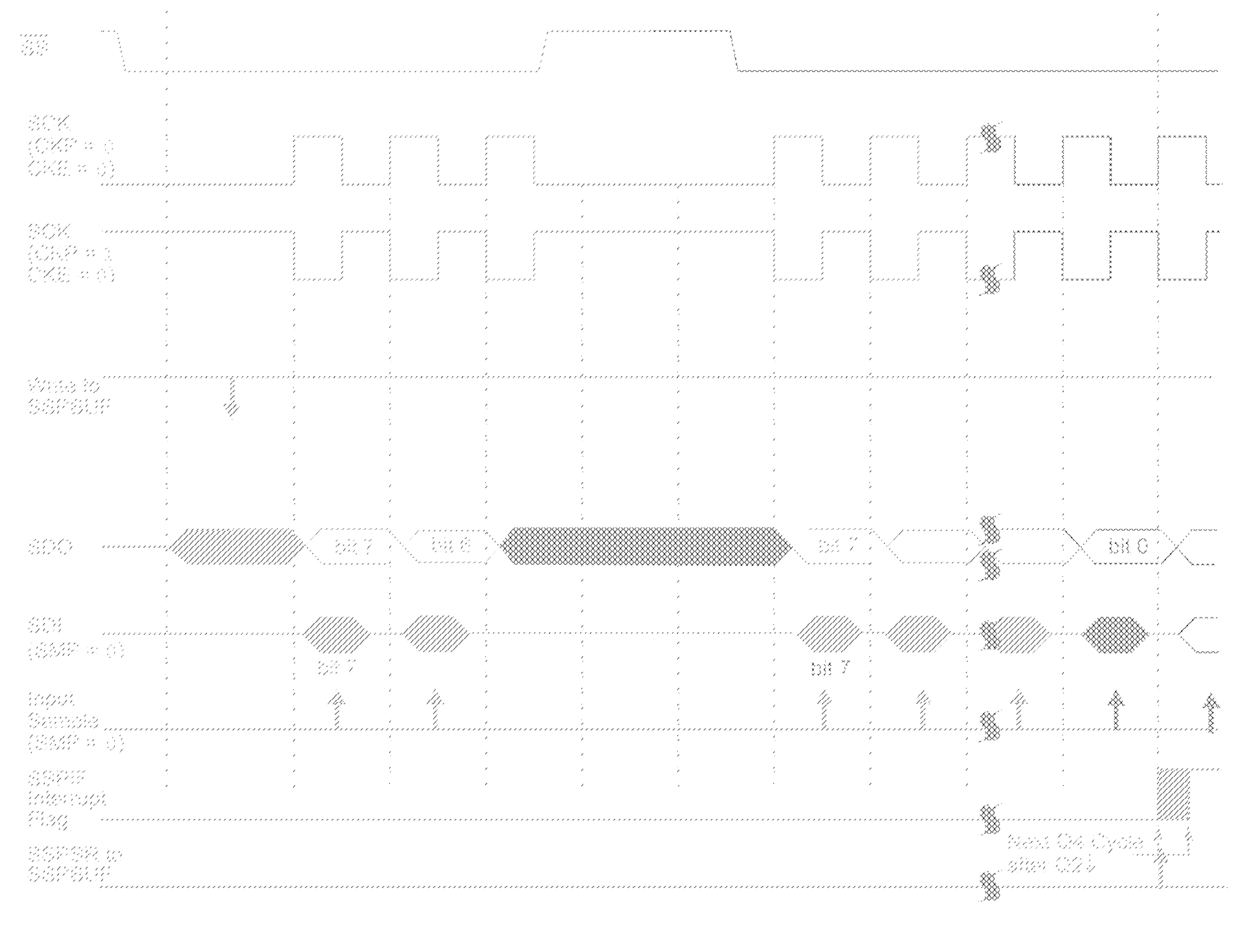
Note 1: When the SPI is in Slave mode with $\overline{SS}$ pin control enabled (SSPCON<3:0> = 0100), the SPI module will reset if the $\overline{SS}$ pin is set to V_{DD}.

2: If the SPI is used in Slave mode with CKE set, then the $\overline{SS}$ pin control must be enabled.

When the SPI module resets, the bit counter is forced to '0'. This can be done by either forcing the $\overline{SS}$ pin to a high level or clearing the SSPEN bit.

To emulate two-wire communication, the SDO pin can be connected to the SDI pin. When the SPI needs to operate as a receiver, the SDO pin can be configured as an input. This disables transmissions from the SDO. The SDI can always be left as an input (SDI function) since it cannot create a bus conflict.

FIGURE 18-4: SLAVE SYNCHRONIZATION WAVEFORM



REGISTER 18-3: SSPSTAT: MSSP STATUS REGISTER (I^2C™ MODE)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	$D/\overline{A}$	P[1]	S[1]	$R/\overline{W}$[2,3]	UA	BF
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 **SMP:** Slew Rate Control bit

<u>In Master or Slave mode:</u>
1 = Slew rate control disabled for Standard Speed mode (100 kHz and 1 MHz)
0 = Slew rate control enabled for High-Speed mode (400 kHz)

bit 6 **CKE:** SMBus Select bit

<u>In Master or Slave mode:</u>
1 = Enable SMBus specific inputs
0 = Disable SMBus specific inputs

bit 5 **$D/\overline{A}$:** Data/$\overline{\text{Address}}$ bit

<u>In Master mode:</u>
Reserved.

<u>In Slave mode:</u>
1 = Indicates that the last byte received or transmitted was data
0 = Indicates that the last byte received or transmitted was address

bit 4 **P:** Stop bit[1]

1 = Indicates that a Stop bit has been detected last
0 = Stop bit was not detected last

bit 3 **S:** Start bit[1]

1 = Indicates that a Start bit has been detected last
0 = Start bit was not detected last

bit 2 **$R/\overline{W}$:** Read/$\overline{\text{Write}}$ Information bit (I^2C mode only)[2,3]

<u>In Slave mode:</u>
1 = Read
0 = Write

<u>In Master mode:</u>
1 = Transmit is in progress
0 = Transmit is not in progress

bit 1 **UA:** Update Address bit (10-Bit Slave mode only)

1 = Indicates that the user needs to update the address in the SSPADD register
0 = Address does not need to be updated

bit 0 **BF:** Buffer Full Status bit

<u>In Receive mode:</u>
1 = Receive complete, SSPBUF is full
0 = Receive is not complete, SSPBUF is empty

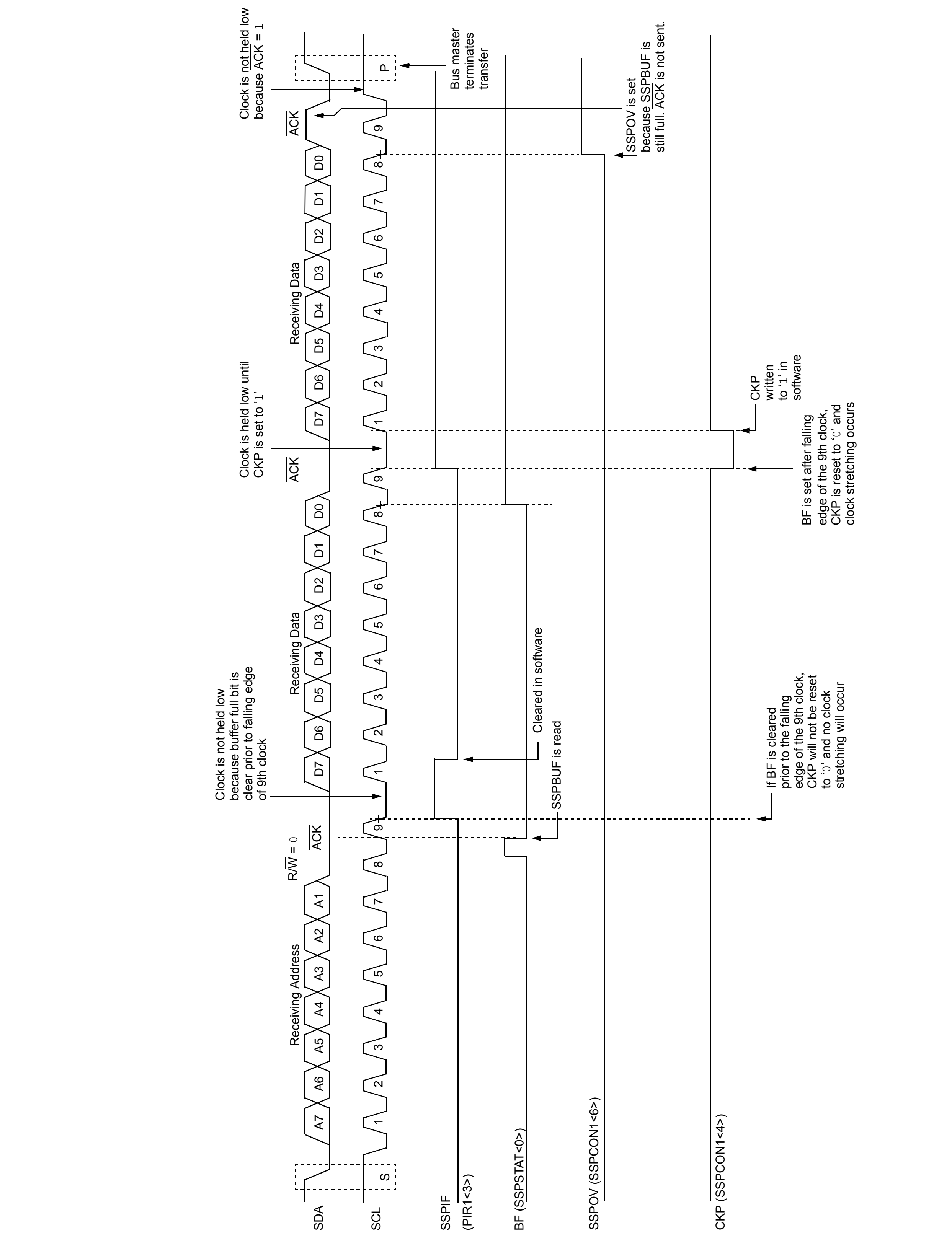
<u>In Transmit mode:</u>
1 = Data transmit in progress (does not include the $\overline{ACK}$ and Stop bits), SSPBUF is full
0 = Data transmit complete (does not include the $\overline{ACK}$ and Stop bits), SSPBUF is empty

Note 1: This bit is cleared on Reset and when SSPEN is cleared.

2: This bit holds the $R/\overline{W}$ bit information following the last address match. This bit is only valid from the address match to the next Start bit, Stop bit or not $\overline{ACK}$ bit.

3: ORing this bit with SEN, RSEN, PEN, RCEN or ACKEN will indicate if the MSSP is in Idle mode.

FIGURE 18-13: I^{2}C™ SLAVE MODE TIMING WITH SEN = 1 (RECEPTION, 7-BIT ADDRESS)


SDA
SCL
SSPIF
(PIR1<3>)
BF (SSPSTAT<0>)
SSPOV (SSPCON1<6>)
CKP (SSPCON1<4>)
S
Receiving Address
A7
A6
A5
A4
A3
A2
A1
R/$\overline{W}$ = 0
$\overline{ACK}$
1
2
3
4
5
6
7
8
9
Clock is not held low because buffer full bit is clear prior to falling edge of 9th clock
SSPBUF is read
Cleared in software
If BF is cleared prior to the falling edge of the 9th clock, CKP will not be reset to '0' and no clock stretching will occur
Receiving Data
D7
D6
D5
D4
D3
D2
D1
D0
1
2
3
4
5
6
7
8
9
$\overline{ACK}$
Clock is held low until CKP is set to '1'
BF is set after falling edge of the 9th clock, CKP is reset to '0' and clock stretching occurs
CKP written to '1' in software
Receiving Data
D7
D6
D5
D4
D3
D2
D1
D0
1
2
3
4
5
6
7
8
9
$\overline{ACK}$
Clock is not held low because $\overline{ACK}$ = 1
P
Bus master terminates transfer
SSPOV is set because SSPBUF is still full. $\overline{ACK}$ is not sent.

19.4.2 EUSART SYNCHRONOUS SLAVE RECEPTION

The operation of the Synchronous Master and Slave modes is identical, except in the case of Sleep or any Idle mode and bit, SREN, which is a “don’t care” in Slave mode.

If receive is enabled by setting the CREN bit prior to entering Sleep or any Idle mode, then a word may be received while in this low-power mode. Once the word is received, the RSR register will transfer the data to the RCREG register. If the RCIE enable bit is set, the interrupt generated will wake the chip from the low-power mode. If the global interrupt is enabled, the program will branch to the interrupt vector.

To set up a Synchronous Slave Reception:

1. Enable the synchronous master serial port by setting bits, SYNC and SPEN, and clearing bit, CSRC.
2. If interrupts are desired, set enable bit, RCIE.
3. If 9-bit reception is desired, set bit, RX9.
4. To enable reception, set enable bit, CREN.
5. Flag bit, RCIF, will be set when reception is complete. An interrupt will be generated if enable bit, RCIE, was set.
6. Read the RCSTA register to get the 9th bit (if enabled) and determine if any error occurred during reception.
7. Read the 8-bit received data by reading the RCREG register.
8. If any error occurred, clear the error by clearing bit, CREN.
9. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

TABLE 19-10: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE RECEPTION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	55
PIR1	PSPIF[1]	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	58
PIE1	PSPIE[1]	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	58
IPR1	PSPIP[1]	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	58
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	57
RCREG	EUSART Receive Register								57
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	57
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	57
SPBRGH	EUSART Baud Rate Generator Register High Byte								57
SPBRG	EUSART Baud Rate Generator Register Low Byte								57

Legend: — = unimplemented, read as ‘0’. Shaded cells are not used for synchronous slave reception.

Note 1: Reserved in PIC18F2X80 devices; always maintain these bits clear.

20.6 A/D Conversions

Figure 20-3 shows the operation of the A/D Converter after the GO/$\overline{\text{DONE}}$ bit has been set and the ACQT<2:0> bits are cleared. A conversion is started after the following instruction to allow entry into Sleep mode before the conversion begins.

Figure 20-4 shows the operation of the A/D Converter after the GO/$\overline{\text{DONE}}$ bit has been set and the ACQT<2:0> bits are set to '010' and selecting a 4 TAD acquisition time before the conversion starts.

Clearing the GO/$\overline{\text{DONE}}$ bit during a conversion will abort the current conversion. The A/D Result register pair will NOT be updated with the partially completed A/D conversion sample. This means the ADRESH:ADRESL registers will continue to contain the value of the last completed conversion (or the last value written to the ADRESH:ADRESL registers).

After the A/D conversion is completed or aborted, a 2 TAD wait is required before the next acquisition can be started. After this wait, acquisition on the selected channel is automatically started.

Note:	The GO/$\overline{\text{DONE}}$ bit should **NOT** be set in the same instruction that turns on the A/D.

FIGURE 20-3: A/D CONVERSION TAD CYCLES (ACQT<2:0> = 000, TACQ = 0)

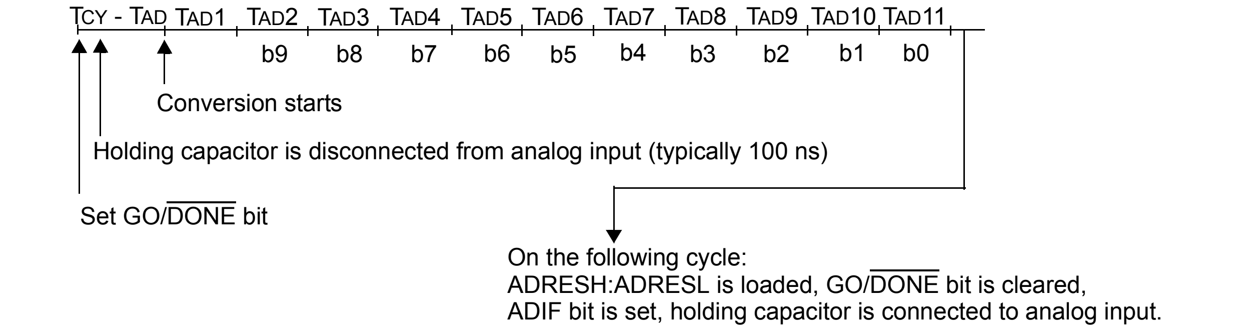
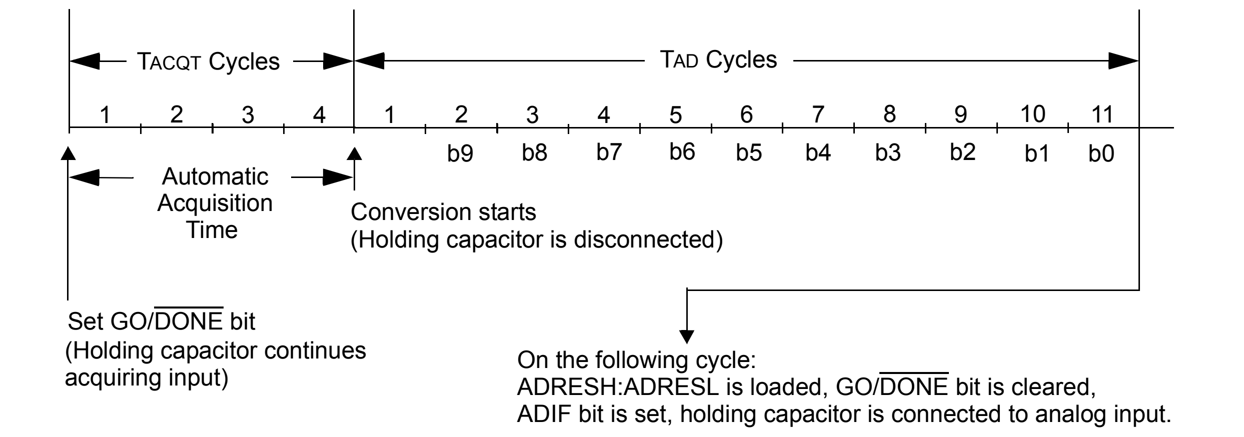

TCY - TAD
TAD1
TAD2
TAD3
TAD4
TAD5
TAD6
TAD7
TAD8
TAD9
TAD10
TAD11
b9
b8
b7
b6
b5
b4
b3
b2
b1
b0
Conversion starts
Holding capacitor is disconnected from analog input (typically 100 ns)
Set GO/DONE bit
On the following cycle:
ADRESH:ADRESL is loaded, GO/DONE bit is cleared,
ADIF bit is set, holding capacitor is connected to analog input.

FIGURE 20-4: A/D CONVERSION TAD CYCLES (ACQT<2:0> = 010, TACQ = 4 TAD)


TACQT Cycles
TAD Cycles
1
2
3
4
1
2
3
4
5
6
7
8
9
10
11
b9
b8
b7
b6
b5
b4
b3
b2
b1
b0
Automatic Acquisition Time
Conversion starts
(Holding capacitor is disconnected)
Set GO/DONE bit
(Holding capacitor continues acquiring input)
On the following cycle:
ADRESH:ADRESL is loaded, GO/DONE bit is cleared,
ADIF bit is set, holding capacitor is connected to analog input.

REGISTER 24-21: RXERRCNT: RECEIVE ERROR COUNT REGISTER

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0
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 7-0 **REC<7:0>:** Receive Error Counter bits

This register contains the receive error value as defined by the CAN specifications. When RXERRCNT > 127, the module will go into an error-passive state. RXERRCNT does not have the ability to put the module in “bus-off” state.

EXAMPLE 24-5: READING A CAN MESSAGE

```
; Need to read a pending message from RXB0 buffer.
; To receive any message, filter, mask and RXM1:RXM0 bits in RXB0CON registers must be
; programmed correctly.
;
; Make sure that there is a message pending in RXB0.
BTFSS   RXB0CON, RXFUL                ; Does RXB0 contain a message?
BRA     NoMessage                     ; No. Handle this situation...
; We have verified that a message is pending in RXB0 buffer.
; If this buffer can receive both Standard or Extended Identifier messages,
; identify type of message received.
BTFSS   RXB0SIDL, EXID                ; Is this Extended Identifier?
BRA     StandardMessage               ; No. This is Standard Identifier message.
                                      ; Yes. This is Extended Identifier message.
; Read all 29-bits of Extended Identifier message.
...
; Now read all data bytes
MOVFF   RXB0DO, MY_DATA_BYTE1
...
; Once entire message is read, mark the RXB0 that it is read and no longer FULL.
BCF     RXB0CON, RXFUL                ; This will allow CAN Module to load new messages
                                      ; into this buffer.
...
```

FIGURE 24-6: LENGTHENING A BIT PERIOD (ADDING SJW TO PHASE SEGMENT 1)

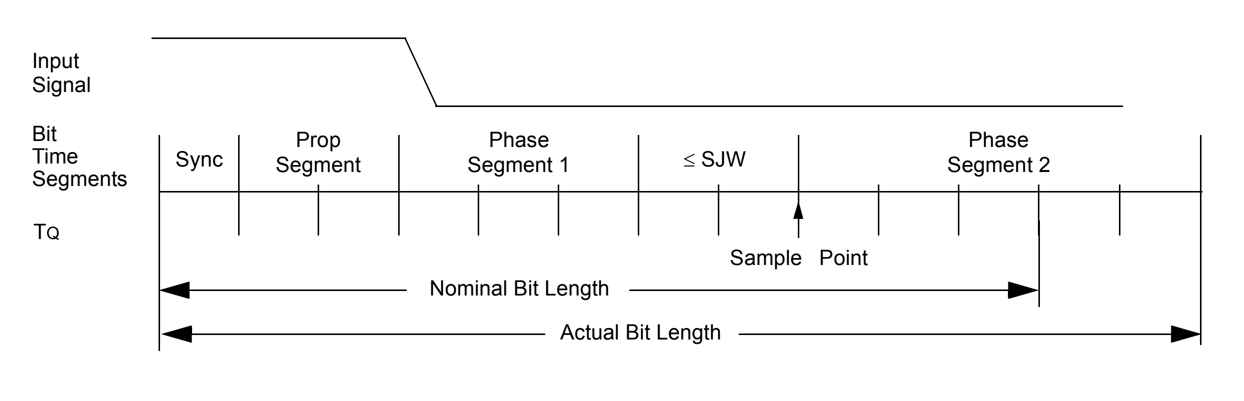
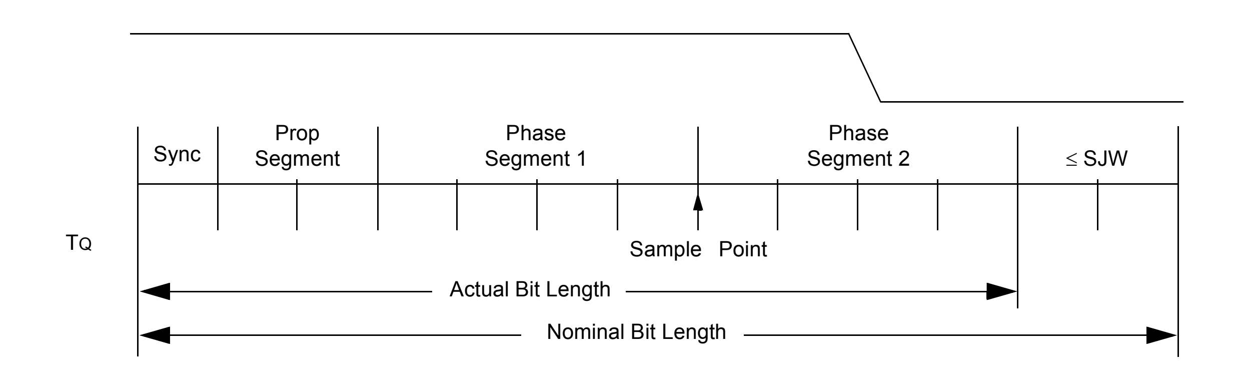


FIGURE 24-7: SHORTENING A BIT PERIOD (SUBTRACTING SJW FROM PHASE SEGMENT 2)



24.11 Programming Time Segments

Some requirements for programming of the time segments:

- Prop_Seg + Phase_Seg 1 ≥ Phase_Seg 2
- Phase_Seg 2 ≥ Sync Jump Width.

For example, assume that a 125 kHz CAN baud rate is desired, using 20 MHz for FOSC. With a TOSC of 50 ns, a baud rate prescaler value of 04h gives a TQ of 500 ns. To obtain a Nominal Bit Rate of 125 kHz, the Nominal Bit Time must be 8 μs or 16 TQ.

Using 1 TQ for the Sync_Seg, 2 TQ for the Prop_Seg and 7 TQ for Phase Segment 1 would place the sample point at 10 TQ after the transition. This leaves 6 TQ for Phase Segment 2.

By the rules above, the Sync Jump Width could be the maximum of 4 TQ. However, normally a large SJW is only necessary when the clock generation of the different nodes is inaccurate or unstable, such as using ceramic resonators. Typically, an SJW of 1 is enough.

24.12 Oscillator Tolerance

As a rule of thumb, the bit timing requirements allow ceramic resonators to be used in applications with transmission rates of up to 125 Kbit/sec. For the full bus speed range of the CAN protocol, a quartz oscillator is required. Refer to ISO11898-1 for oscillator tolerance requirements.

26.1.1 STANDARD INSTRUCTION SET

ADDLW **ADD Literal to W**

Syntax: ADDLW k

Operands: 0 ≤ k ≤ 255

Operation: (W) + k → W

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

Encoding:

0000	1111	kkkk	kkkk

Description: The contents of W are added to the 8-bit literal ‘k’ and the result is placed in W.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal ‘k’	Process Data	Write to W

Example: `ADDLW 15h`

Before Instruction
W = 10h
After Instruction
W = 25h

ADDWF **ADD W to f**

Syntax: ADDWF f {,d {,a}}

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

Operation: (W) + (f) → dest

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

Encoding:

0010	01da	ffff	ffff

Description: Add W to register ‘f’. If ‘d’ is ‘0’, the result is stored in W. If ‘d’ is ‘1’, the result is stored back in register ‘f’.

If ‘a’ is ‘0’, the Access Bank is selected. If ‘a’ is ‘1’, the BSR is used to select the GPR bank.

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 26.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode”** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register ‘f’	Process Data	Write to destination

Example: `ADDWF REG, 0, 0`

Before Instruction
W = 17h
REG = 0C2h
After Instruction
W = 0D9h
REG = 0C2h

Note:	All PIC18 instructions may take an optional label argument preceding the instruction mnemonic for use in symbolic addressing. If a label is used, the instruction format then becomes: {label} instruction argument(s).

POP Pop Top of Return Stack

Syntax:	POP
Operands:	None
Operation:	(TOS) → bit bucket
Status Affected:	None
Encoding:	`0000 0000 0000 0110`
Description:	The TOS value is pulled off the return stack and is discarded. The TOS value then becomes the previous value that was pushed onto the return stack. This instruction is provided to enable the user to properly manage the return stack to incorporate a software stack.
Words:	1
Cycles:	1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	POP TOS value	No operation

Example:
```
POP
GOTO    NEW
```

Before Instruction

TOS	=	0031A2h
Stack (1 level down)	=	014332h

After Instruction

TOS	=	014332h
PC	=	NEW

PUSH Push Top of Return Stack

Syntax:	PUSH
Operands:	None
Operation:	(PC + 2) → TOS
Status Affected:	None
Encoding:	`0000 0000 0000 0101`
Description:	The PC + 2 is pushed onto the top of the return stack. The previous TOS value is pushed down on the stack. This instruction allows implementing a software stack by modifying TOS and then pushing it onto the return stack.
Words:	1
Cycles:	1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	PUSH PC + 2 onto return stack	No operation	No operation

Example:
```
PUSH
```

Before Instruction

TOS	=	345Ah
PC	=	0124h

After Instruction

PC	=	0126h
TOS	=	0126h
Stack (1 level down)	=	345Ah

SLEEP — Enter Sleep mode

Syntax: SLEEP

Operands: None

Operation: 00h → WDT,
0 → WDT postscaler,
1 → $\overline{TO}$,
0 → $\overline{PD}$

Status Affected: $\overline{TO}$, $\overline{PD}$

Encoding:

0000	0000	0000	0011

Description: The Power-Down Status bit ($\overline{PD}$) is cleared. The Time-out Status bit ($\overline{TO}$) is set. Watchdog Timer and its postscaler are cleared.

The processor is put into Sleep mode with the oscillator stopped.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	Process Data	Go to Sleep

Example: `SLEEP`

Before Instruction
$\overline{TO}$ = ?
$\overline{PD}$ = ?

After Instruction
$\overline{TO}$ = 1 †
$\overline{PD}$ = 0

† If WDT causes wake-up, this bit is cleared.

SUBFWB — Subtract f from W with Borrow

Syntax: SUBFWB f {,d {,a}}

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

Operation: (W) – (f) – ($\overline{C}$) → dest

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

Encoding:

0101	01da	ffff	ffff

Description: Subtract register 'f' and Carry flag (borrow) from W (2's complement method). If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored in register 'f'.

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

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 26.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination

Example 1: `SUBFWB REG, 1, 0`

```
Before Instruction
     REG    =   3
     W      =   2
     C      =   1
After Instruction
     REG    =   FF
     W      =   2
     C      =   0
     Z      =   0
     N      =   1    ; result is negative
```

Example 2: `SUBFWB REG, 0, 0`

```
Before Instruction
     REG    =   2
     W      =   5
     C      =   1
After Instruction
     REG    =   2
     W      =   3
     C      =   1
     Z      =   0
     N      =   0    ; result is positive
```

Example 3: `SUBFWB REG, 1, 0`

```
Before Instruction
     REG    =   1
     W      =   2
     C      =   0
After Instruction
     REG    =   0
     W      =   2
     C      =   1
     Z      =   1    ; result is zero
     N      =   0
```

28.0 ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings (†)

Ambient temperature under bias	-40°C to +125°C
Storage temperature	-65°C to +150°C
Voltage on any pin with respect to VSS (except VDD and $\overline{\text{MCLR}}$)	-0.3V to (VDD + 0.3V)
Voltage on VDD with respect to VSS	-0.3V to +7.5V
Voltage on $\overline{\text{MCLR}}$ with respect to VSS **(Note 2)**	0V to +13.25V
Total power dissipation **(Note 1)**	1.0W
Maximum current out of VSS pin	300 mA
Maximum current into VDD pin	250 mA
Input clamp current, IIK (VI < 0 or VI > VDD)	±20 mA
Output clamp current, IOK (VO < 0 or VO > VDD)	±20 mA
Maximum output current sunk by any I/O pin	25 mA
Maximum output current sourced by any I/O pin	25 mA
Maximum current sunk by all ports	200 mA
Maximum current sourced by all ports	200 mA

Note 1: Power dissipation is calculated as follows:
Pdis = VDD x {IDD – ∑ IOH} + ∑ {(VDD – VOH) x IOH} + ∑(VOL x IOL)

2: Voltage spikes below VSS at the $\overline{\text{MCLR}}$/VPP pin, inducing currents greater than 80 mA, may cause latch-up. Thus, a series resistor of 50-100Ω should be used when applying a “low” level to the $\overline{\text{MCLR}}$/VPP/RE3 pin, rather than pulling this pin directly to VSS.

† **NOTICE:** Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

NOTES: