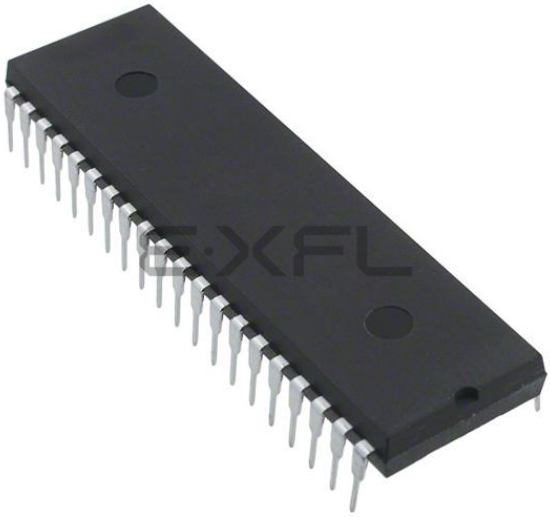


Welcome to E-XFL.COM

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	32MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	36
Program Memory Size	14KB (8K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 34x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	40-DIP (0.600", 15.24mm)
Supplier Device Package	40-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf1567-i-p

TABLE 3: 40/44-PIN ALLOCATION TABLE (PIC16LF1567)

I/O	40-Pin PDIP	40-Pin UQFN	44-Pin TQFP	Analog Channel	ADC and CVD	Timers	PWM	EUSART	MSSP	Interrupt	Pull-Up	Basic
RA0	2	17	19	AN20	—	—	PWM10	—	SS1(1)	—	—	—
RA1	3	18	20	AN10	—	—	PWM11	—	SS2	—	—	—
RA2	4	19	21	AN0	VREF-	—	PWM12	—	—	—	—	—
RA3	5	20	22	AN1	VREF+		PWM13	—	—	—	—	—
RA4	6	21	23	AN2	—	T0CKI	—	—	—	—	—	—
RA5	7	22	24	AN21	—	—	—	—	SS1(1)	—	—	—
RA6	14	29	31	AN22	ADTRIG	—	—	—	—	—	—	CLKOUT
RA7	13	28	30	AN11	—	—	—	—	—	—	—	CLKIN
RB0	33	8	8	AN16	—	—	PWM20	—	—	INT IOC	Y	—
RB1	34	9	9	AN27	—	—	PWM21	—	—	IOC	Y	—
RB2	35	10	10	AN17	—	—	PWM22	—	—	IOC	Y	—
RB3	36	11	11	AN28	—	—	PWM23	—	—	IOC	Y	—
RB4	37	12	14	AN18	AD1GRDA(1) AD2GRDA(1)	—	—	—	—	IOC	Y	—
RB5	38	13	15	AN29	AD1GRDA(1) AD2GRDA(1)	T1G	—	—	—	IOC	Y	—
RB6	39	14	16	AN19	AD1GRDB(1) AD2GRDB(1)	—	—	—	—	IOC	Y	ICSPCLK ICDCLK
RB7	40	15	17	AN40	AD1GRDB(1) AD2GRDB(1)	—	—	—	—	IOC	Y	ICSPDAT ICDDAT
RC0	15	30	32	AN12	—	T1CKI	—	—	SDO2	—	—	—
RC1	16	31	35	AN23	—	—	PWM2	—	SCL2 SCK2	—	—	—
RC2	17	32	36	AN13	—	—	PWM1	—	SDA2 SDI2	—	—	—
RC3	18	33	37	AN24	—	—	—	—	SCL1 SCK1	—	—	—
RC4	23	38	42	AN14	—	—	—	—	SDA1 SDI1	—	—	—
RC5	24	39	43	AN25	—	—	—	—	SDO1 I2CLVL	—	—	—
RC6	25	40	44	AN15	—	—	—	TX CK	—	—	—	—
RC7	26	1	1	AN26	—	—	—	RX DT	—	—	—	—
RD0	19	34	38	AN42	—	—	—	—	—	—	—	—

EXAMPLE 3-2: ACCESSING PROGRAM MEMORY VIA FSR

```
constants
     DW          DATA0    ; First constant
     DW          DATA1    ; Second constant
     DW          DATA2
     DW          DATA3
my_function
     ;… LOTS OF CODE…
     MOVLW    DATA_INDEX
     ADDLW    LOW constants
     MOVWF    FSR1L
     MOVLW    HIGH constants; MSb is set
automatically
     MOVWF    FSR1H
     BTFSC    STATUS,C     ; carry from ADDLW?
     INCF     FSR1H,f      ; yes
     MOVIW    0[FSR1]
;THE PROGRAM MEMORY IS IN W
```

3.2 Data Memory Organization

The data memory is partitioned in 32 memory banks with 128 bytes in a bank. Each bank consists of (Figure 3-2):

- 12 core registers
- 20 Special Function Registers (SFR)
- Up to 80 bytes of General Purpose RAM (GPR)
- 16 bytes of common RAM

The active bank is selected by writing the bank number into the Bank Select Register (BSR). Unimplemented memory will read as '0'. All data memory can be accessed either directly (via instructions that use the file registers) or indirectly via the two File Select Registers (FSR). See **Section 3.5 "Indirect Addressing"** for more information.

Data memory uses a 12-bit address. The upper five bits of the address define the Bank address and the lower seven bits select the registers/RAM in that bank.

3.2.1 CORE REGISTERS

The core registers contain the registers that directly affect the basic operation. The core registers occupy the first 12 addresses of every data memory bank (addresses x00h/x08h through x0Bh/x8Bh). These registers are listed below in Table 3-2. For detailed information, see Table 3-10.

TABLE 3-2: CORE REGISTERS

Addresses	BANKx
x00h or x80h	INDF0
x01h or x81h	INDF1
x02h or x82h	PCL
x03h or x83h	STATUS
x04h or x84h	FSR0L
x05h or x85h	FSR0H
x06h or x86h	FSR1L
x07h or x87h	FSR1H
x08h or x88h	BSR
x09h or x89h	WREG
x0Ah or x8Ah	PCLATH
x0Bh or x8Bh	INTCON

3.2.1.1 STATUS Register

The STATUS register, shown in Register 3-1, contains:

- the arithmetic status of the ALU
- the Reset status

The STATUS register can be the destination for any instruction, like any other register. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, the $\overline{TO}$ and $\overline{PD}$ bits are not writable. Therefore, the result of an instruction with the STATUS register as destination may be different than intended.

For example, `CLRF STATUS` will clear the upper three bits and set the Z bit. This leaves the STATUS register as '`000u u1uu`' (where u = unchanged).

It is recommended, therefore, that only `BCF`, `BSF`, `SWAPF` and `MOVWF` instructions are used to alter the STATUS register, because these instructions do not affect any Status bits. For other instructions not affecting any Status bits (refer to **Section 24.0 "Instruction Set Summary"**).

Note 1: The C and DC bits operate as $\overline{Borrow}$ and Digit $\overline{Borrow}$ out bits, respectively, in subtraction.

TABLE 3-4: PIC16LF1567 MEMORY MAP, BANKS 0-7

	BANK 0		BANK 1		BANK 2		BANK 3		BANK 4		BANK 5		BANK 6		BANK 7
000h	CPU Core Register, see Table 3-2 for specifics														
001h															
002h															
003h															
004h															
005h															
006h															
007h															
008h															
009h															
00Ah															
00Bh															
00Ch	PORTA	08Ch	TRISA	10Ch	LATA	18Ch	ANSELA	20Ch		28Ch	—	30Ch	—	38Ch	—
00Dh	PORTB	08Dh	TRISB	10Dh	LATB	18Dh	ANSELB	20Dh	WPUB	28Dh	—	30Dh	—	38Dh	—
00Eh	PORTC	08Eh	TRISC	10Eh	LATC	18Eh	ANSELC	20Eh	—	28Eh	—	30Eh	—	38Eh	—
00Fh	PORTD	08Fh	TRISD	10Fh	LATD	18Fh	ANSELD	20Fh	—	28Fh	—	30Fh	—	38Fh	—
010h	PORTE	090h	TRISE	110h	LATE	190h	ANSELE	210h	WPUE	290h	—	310h	—	390h	—
011h	PIR1	091h	PIE1	111h	—	191h	PMADRL	211h	SSP1BUF	291h	—	311h	—	391h	—
012h	PIR2	092h	PIE2	112h	—	192h	PMADRH	212h	SSP1ADD	292h	—	312h	—	392h	—
013h	—	093h	—	113h	—	193h	PMDATL	213h	SSP1MSK	293h	—	313h	—	393h	—
014h	—	094h	—	114h	—	194h	PMDATH	214h	SSP1STAT	294h	—	314h	—	394h	IOCBP
015h	TMR0	095h	OPTION_REG	115h	—	195h	PMCON1	215h	SSP1CON1	295h	—	315h	—	395h	IOCBN
016h	TMR1L	096h	PCON	116h	BORCON	196h	PMCON2	216h	SSP1CON2	296h	—	316h	—	396h	IOCBF
017h	TMR1H	097h	WDTCON	117h	FVRCON	197h	—	217h	SSP1CON3	297h	—	317h	—	397h	—
018h	T1CON	098h	—	118h	—	198h	—	218h	SSPLVL	298h	—	318h	—	398h	—
019h	T1GCON	099h	OSCCON	119h	—	199h	RCREG	219h	SSP2BUF	299h	—	319h	—	399h	—
01Ah	TMR2	09Ah	OSCSTAT	11Ah	—	19Ah	TXREG	21Ah	SSP2ADD	29Ah	—	31Ah	—	39Ah	—
01Bh	PR2	09Bh	ADRESL	11Bh	—	19Bh	SPBRGL	21Bh	SSP2MSK	29Bh	—	31Bh	—	39Bh	—
01Ch	T2CON	09Ch	ADRESH	11Ch	—	19Ch	SPBRGH	21Ch	SSP2STAT	29Ch	—	31Ch	—	39Ch	—
01Dh	—	09Dh	ADCON0	11Dh	APFCON	19Dh	RCSTA	21Dh	SSP2CON1	29Dh	—	31Dh	—	39Dh	—
01Eh	—	09Eh	ADCON1	11Eh	—	19Eh	TXSTA	21Eh	SSP2CON2	29Eh	—	31Eh	—	39Eh	—
01Fh	—	09Fh	ADCON2	11Fh	—	19Fh	BAUDCON	21Fh	SSP2CON3	29Fh	—	31Fh	—	39Fh	—
020h	General Purpose Register 96 Bytes	0A0h	General Purpose Register 80 Bytes	120h	General Purpose Register 80 Bytes	1A0h	General Purpose Register 80 Bytes	220h	General Purpose Register 80 Bytes	2A0h	General Purpose Register 80 Bytes	320h	General Purpose Register 80 Bytes	3A0h	General Purpose Register 80 Bytes
06Fh		0EFh		16Fh		1EFh		26Fh		2EFh		36Fh		3EFh	
070h		0F0h	Accesses 70h – 7Fh	170h	Accesses 70h – 7Fh	1F0h	Accesses 70h – 7Fh	270h	Accesses 70h – 7Fh	2F0h	Accesses 70h – 7Fh	370h	Accesses 70h – 7Fh	3F0h	Accesses 70h – 7Fh
07Fh		0FFh		17Fh		1FFh		27Fh		2FFh		37Fh		3FFh	

Legend: = Unimplemented data memory locations, read as '0'.

Note 1: These ADC registers are the same as the registers in Bank 14.

TABLE 3-11: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Addr.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
Bank 15											
780h	INDF0(1)	Addressing this location uses contents of FSR0H/FSR0L to address data memory (not a physical register)								xxxx xxxx	uuuu uuuu
781h	INDF1(1)	Addressing this location uses contents of FSR1H/FSR1L to address data memory (not a physical register)								xxxx xxxx	uuuu uuuu
782h	PCL(1)	Program Counter (PC) Least Significant Byte								0000 0000	0000 0000
783h	STATUS(1)	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	---1 1000	---q quuu
784h	FSR0L(1)	Indirect Data Memory Address 0 Low Pointer								0000 0000	uuuu uuuu
785h	FSR0H(1)	Indirect Data Memory Address 0 High Pointer								0000 0000	0000 0000
786h	FSR1L(1)	Indirect Data Memory Address 1 Low Pointer								0000 0000	uuuu uuuu
787h	FSR1H(1)	Indirect Data Memory Address 1 High Pointer								0000 0000	0000 0000
788h	BSR(1)	—	—	—	BSR<4:0>					---0 0000	---0 0000
789h	WREG(1)	Working Register								0000 0000	uuuu uuuu
78Ah	PCLATH(1)	—	Write Buffer for the upper 7 bits of the Program Counter							-000 0000	-000 0000
78Bh	INTCON(1)	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	0000 0000	0000 000u
78Ch	—	Unimplemented								—	—
78Dh	—	Unimplemented								—	—
78Eh	—	Unimplemented								—	—
78Fh	—	Unimplemented								—	—
790h	—	Unimplemented								—	—
791h	AD2CON0	CHS<5:0>						GO/$\overline{DONE}$2	AD2CON	0000 0000	0000 0000
792h	—	Unimplemented								—	—
793h	AD2CON2	—	TRIGSEL<2:0>			—	—	—	—	-000 ----	-000 ----
794h	AD2CON3	AD2EPPOL	AD2IPPOL	—	—	—	—	AD2IPEN	AD2DSEN	00-- --00	00-- --00
795h	—	Unimplemented								—	—
796h	AD2PRECON	—	ADPRE<6:0>							-000 0000	-000 0000
797h	AD2ACQCON	—	ADACQ<6:0>							-000 0000	-000 0000
798h	AD2GRD	GRD2BOE	GRD2AOE	GRD2POL	—	—	—	—	TX2POL	000- ---x	000- ---u
799h	AD2CAPCON	—	—	—	—	ADD2CAP<3:0>				---- 0000	---- 0000
79Ah	AAD2RES0L	ADRESL								xxxx xxxx	uuuu uuuu
79Bh	AAD2RES0H	ADRESH								xxxx xxxx	uuuu uuuu
79Ch	AAD2RES1L	ADRESL								xxxx xxxx	uuuu uuuu
79Dh	AAD2RES1H	ADRESH								xxxx xxxx	uuuu uuuu
79Eh	AD2CH0	CH27	CH26	CH25	CH24	CH23	CH22	CH21	CH20	0000 0000	0000 0000
79Fh	AD2CH1	—	—	—	—	—	CH40	CH29	CH28	-----000	-----000
	AD2CH1(2)	CH45	CH44	CH43	CH42	CH41	CH40	CH29	CH28	00000000	00000000

Legend: x = unknown, u = unchanged, q = depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations unimplemented, read as '0'.

Note 1: These registers can be accessed from any bank.
2: PIC16LF1567.
3: These registers/bits are available at two address locations, in Bank 1 and Bank 14.
4: PIC16LF1566 only.
5: Unimplemented, read as '1'.

6.1 Power-on Reset (POR)

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

6.1.1 POWER-UP TIMER (PWRT)

The Power-up Timer provides a nominal 64 ms time-out on POR or Brown-out Reset.

The device is held in Reset as long as PWRT is active. The PWRT delay allows additional time for the VDD to rise to an acceptable level. The Power-up Timer is enabled by clearing the PWRTE bit in Configuration Words.

The Power-up Timer starts after the release of the POR and BOR.

For additional information, refer to Application Note AN607, *"Power-up Trouble Shooting"* (DS00000607).

6.2 Brown-out Reset (BOR)

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

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

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

Refer to Table 6-1 for more information.

The Brown-out Reset voltage level is selectable by configuring the BORV bit in Configuration Words.

A VDD noise rejection filter prevents the BOR from triggering on small events. If VDD falls below VBOR for a duration greater than parameter TBORDC, the device will reset. See Figure 6-2 for more information.

6.2.1 BOR IS ALWAYS ON

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

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

6.2.2 BOR IS OFF IN SLEEP

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

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

6.2.3 BOR CONTROLLED BY SOFTWARE

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

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

BOR protection is unchanged by Sleep.

TABLE 6-1: BOR OPERATING MODES

BOREN<1:0>	SBOREN	Device Mode	BOR Mode	Instruction Execution upon: Release of POR or Wake-up from Sleep
11	X	X	Active	Waits for BOR ready(1) (BORRDY = 1)
10	X	Awake	Active	Waits for BOR ready (BORRDY = 1)
		Sleep	Disabled	
01	1	X	Active	Waits for BOR ready(1) (BORRDY = 1)
	0	X	Disabled	Begins immediately (BORRDY = x)
00	X	X	Disabled	

Note 1: In these specific cases, "release of POR" and "wake-up from Sleep," there is no delay in start-up. The BOR ready flag, (BORRDY = 1), will be set before the CPU is ready to execute instructions because the BOR circuit is forced on by the BOREN<1:0> bits.

REGISTER 7-2: PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-1/1	R/W-0/0	R/W-0/0
TMR1GIE	AD1IE	RCIE	TXIE	SSP1IE	SSP2IE	TMR2IE	TMR1IE
bit 7							bit 0

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

bit 7 **TMR1GIE:** Timer1 Gate Interrupt Enable bit
1 = Enables the Timer1 gate acquisition interrupt
0 = Disables the Timer1 gate acquisition interrupt

bit 6 **AD1IE:** Analog-to-Digital Converter (ADC1) Interrupt Enable bit
1 = Enables the ADC interrupt
0 = Disables the ADC interrupt

bit 5 **RCIE:** USART Receive Interrupt Enable bit
1 = Enables the USART receive interrupt
0 = Disables the USART receive interrupt

bit 4 **TXIE:** USART Transmit Interrupt Enable bit
1 = Enables the USART transmit interrupt
0 = Disables the USART transmit interrupt

bit 3 **SSP1IE:** Synchronous Serial Port (MSSP1) Interrupt Enable bit
1 = Enables the MSSP1 interrupt
0 = Disables the MSSP1 interrupt

bit 2 **SSP2IE:** Synchronous Serial Port (MSSP1) Interrupt Enable bit
1 = Enables the MSSP2 interrupt
0 = Disables the MSSP2 interrupt

bit 1 **TMR2IE:** TMR2 to PR2 Match Interrupt Enable bit
1 = Enables the Timer2 to PR2 match interrupt
0 = Disables the Timer2 to PR2 match interrupt

bit 0 **TMR1IE:** Timer1 Overflow Interrupt Enable bit
1 = Enables the Timer1 overflow interrupt
0 = Disables the Timer1 overflow interrupt

Note: Bit PEIE of the INTCON register must be set to enable any peripheral interrupt.

10.2.1 READING THE FLASH PROGRAM MEMORY

To read a program memory location, the user must:

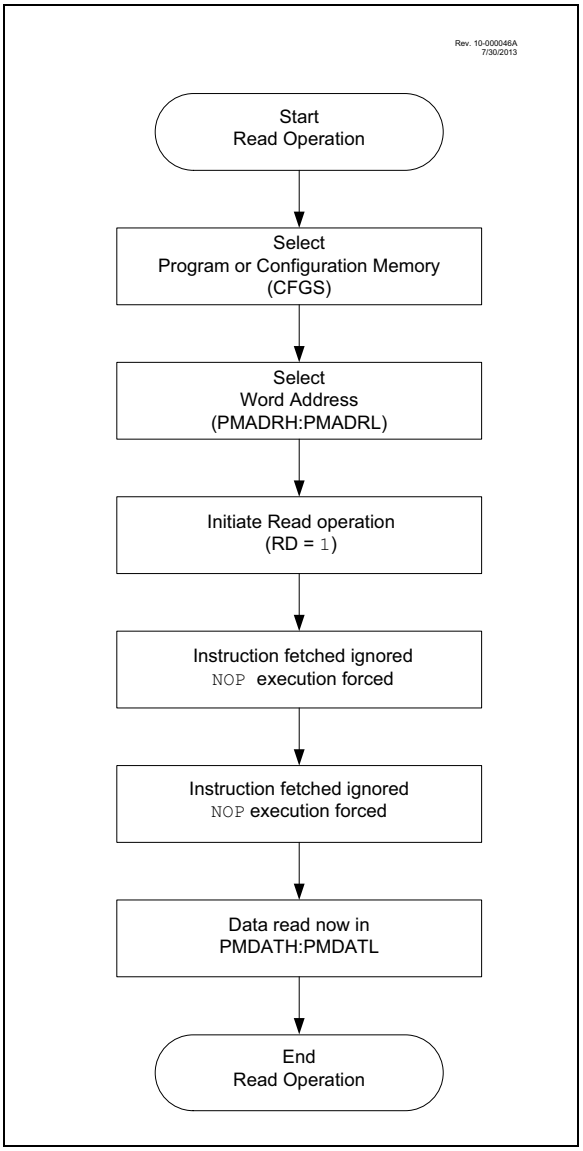
1. Write the desired address to the PMADRH:PMADRL register pair.
2. Clear the CFGS bit of the PMCON1 register.
3. Then, set control bit RD of the PMCON1 register.

Once the read control bit is set, the program memory Flash controller will use the second instruction cycle to read the data. This causes the second instruction immediately following the "BSF PMCON1,RD" instruction to be ignored. The data is available in the very next cycle, in the PMDATH:PMDATL register pair; therefore, it can be read as two bytes in the following instructions.

PMDATH:PMDATL register pair will hold this value until another read or until it is written to by the user.

Note:	The two instructions following a program memory read are required to be NOPs. This prevents the user from executing a 2-cycle instruction on the next instruction after the RD bit is set.

FIGURE 10-1: FLASH PROGRAM MEMORY READ FLOWCHART



11.4 Register Definitions: PORTA

REGISTER 11-2: PORTA: PORTA REGISTER

R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R-x/x	R/W-x/x	R/W-x/x	R/W-x/x
RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0
bit 7							bit 0

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

bit 7-0 **RA<7:0>**: RA7:RA0 PORTA I/O Value bits(1)
1 = Port pin is ≥ VIH
0 = Port pin is ≤ VIL

Note 1: Writes to PORTA are actually written to corresponding LATA register. Reads from PORTA register is return of actual I/O pin values.

REGISTER 11-3: TRISA: PORTA TRI-STATE REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0
bit 7							bit 0

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

bit 7-0 **TRISA<7:0>:** PORTA Tri-State Control bit
1 = PORTA pin configured as an input (tri-stated)
0 = PORTA pin configured as an output

15.1.5 INTERRUPTS

The ADC module allows for the ability to generate an interrupt upon completion of an Analog-to-Digital conversion. The ADCx interrupt flag is the ADxIF bit in the PIRx register. The ADCx interrupt enable is the ADxIE bit in the PIEx register. The ADxIF bit must be cleared in software.

Note 1:	The ADxIF bit is set at the completion of every conversion, regardless of whether or not the ADCx interrupt is enabled.
2:	The ADC operates during Sleep only when the FRC oscillator is selected.

This interrupt can be generated while the device is operating or while in Sleep. If the device is in Sleep, the interrupt will wake-up the device. Upon waking from Sleep, the next instruction following the SLEEP instruction is always executed. If the user is attempting to wake-up from Sleep and resume in-line code execution, the GIE and PEIE bits of the INTCON register must be disabled. If the GIE and PEIE bits of the INTCON register are enabled, execution will switch to the Interrupt Service Routine.

15.1.6 RESULT FORMATTING

The 10-bit ADC conversion result can be supplied in two formats, left justified or right justified. The ADFM bit of the ADCON1/ADCOMCON register controls the output format.

Figure 15-3 shows the two output formats.

FIGURE 15-3: 10-BIT ADC CONVERSION RESULT FORMAT

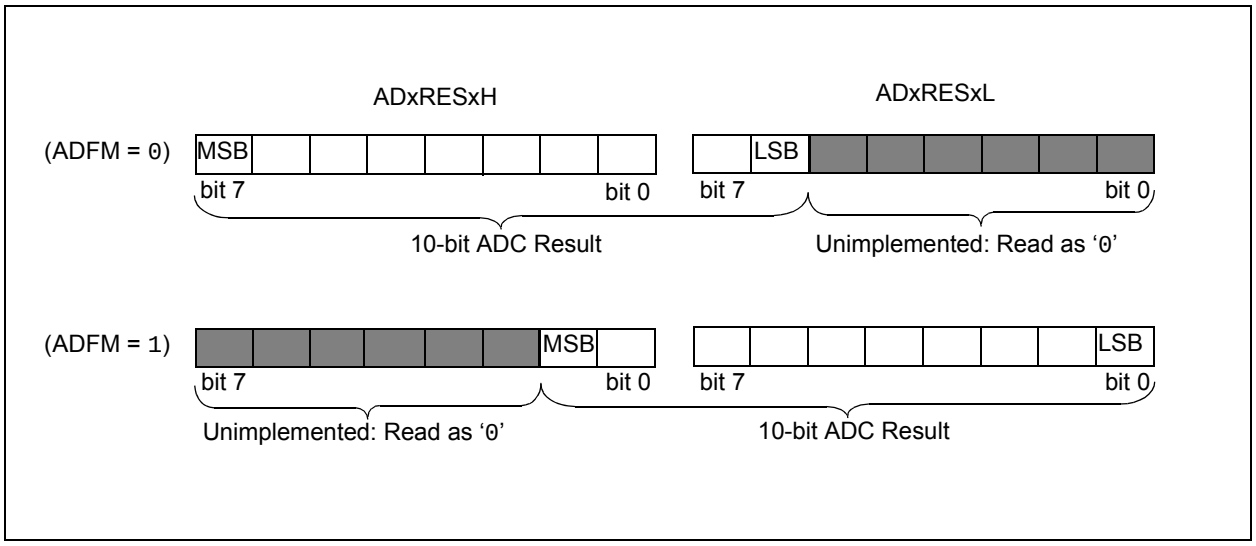
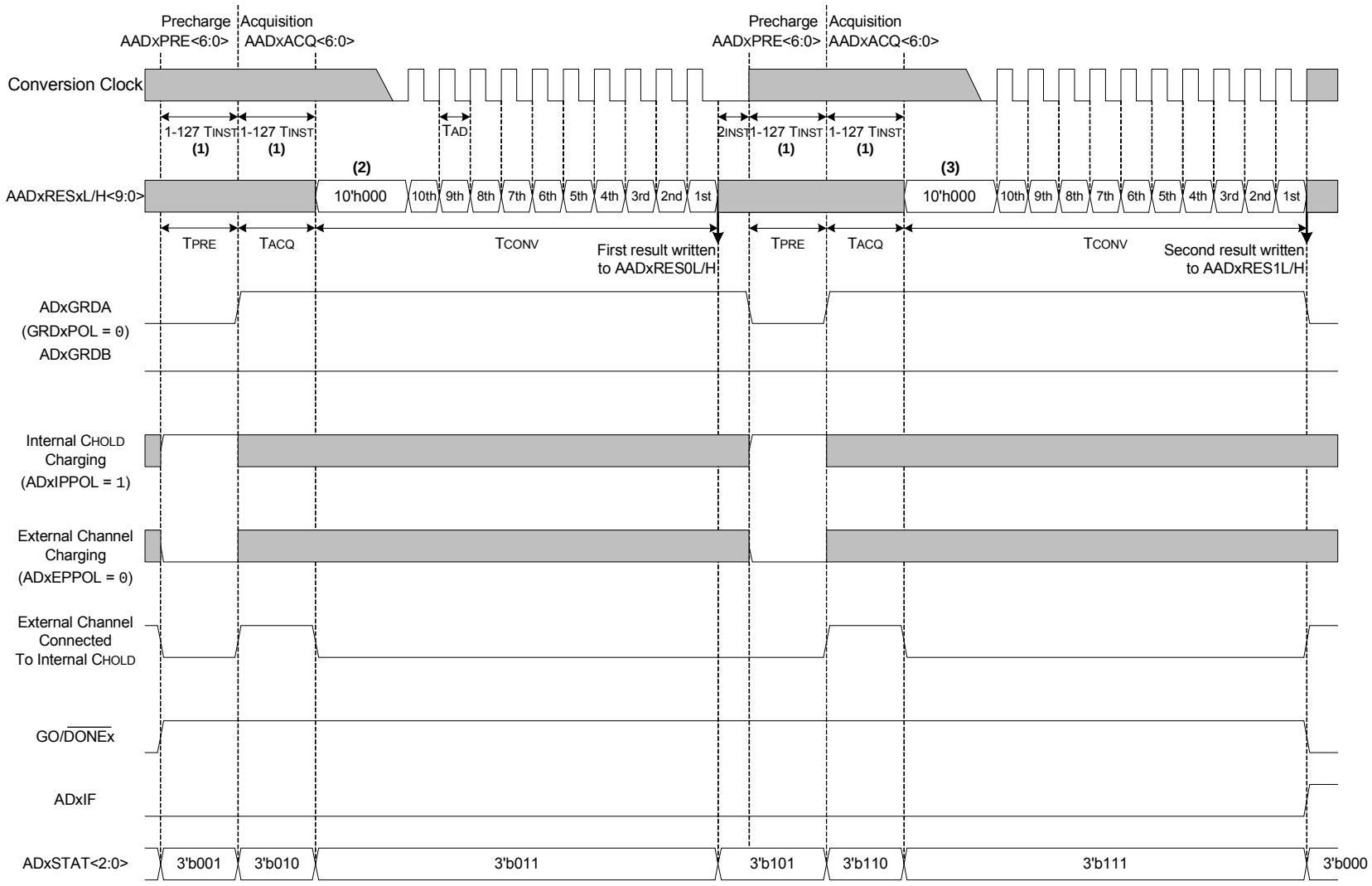


FIGURE 16-6: DOUBLE SAMPLE CONVERSION SEQUENCE (ADDSEN = 1 AND ADIPEN = 0)



Note 1: When the conversion clock is ADCRC, the precharge and acquisition timers are clocked by ADCRC.
2: The AADxRES0L/H registers are set to zero during this period.
3: The AADxRES1L/H registers are set to zero during this period.

REGISTER 16-3: AD1CH1: HARDWARE CVD 1 SECONDARY CHANNEL SELECT REGISTER(1,2,3,4)

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
CH35(5)	CH34(5)	CH33(5)	CH32(5)	CH31(5)	CH30(5)	CH19	CH18
bit 7							bit 0

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

bit 7-0 **CHx:** Channel x to A/S 1 Connection(1, 2, 3, 4)

1 = ANx is connected to A/D 1

0 = ANx is not connected to A/D 1

Note 1: This register selects secondary channels which are connected in parallel to the primary channel selected in AD1CON0. Precharge bias is applied to both the primary and secondary channels.

2: If the same channel is selected as both primary and secondary then the selection as primary takes precedence.

3: Enabling these bits automatically overrides the corresponding TRISx bit to tri-state the selected pin.

4: In the same way that the CHS bits in AD1CON0 only close the switch when the ADC is enabled, these connections and the TRISx overrides are only active if the ADC is enabled by setting ADxON.

5: PIC16LF1567 only. Unimplemented/ Read as '0' on PIC16LF1566.

18.8 Register Definitions: Timer1 Control

REGISTER 18-1: T1CON: TIMER1 CONTROL REGISTER

R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	U-0	R/W-0/u	U-0	R/W-0/u
TMR1CS<1:0>		T1CKPS<1:0>		—	$\overline{\text{T1SYNC}}$	—	TMR1ON
bit 7							bit 0

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

bit 7-6 **TMR1CS<1:0>:** Timer1 Clock Source Select bits
- 11 = Timer1 clock source is LFINTOSC
- 10 = Timer1 clock source is external clock from T1CKI pin (on the rising edge)
- 01 = Timer1 clock source is system clock (FOSC)
- 00 = Timer1 clock source is instruction clock (FOSC/4)

bit 5-4 **T1CKPS<1:0>:** Timer1 Input Clock Prescale Select bits
- 11 = 1:8 Prescale value
- 10 = 1:4 Prescale value
- 01 = 1:2 Prescale value
- 00 = 1:1 Prescale value

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

bit 2 **$\overline{\text{T1SYNC}}$:** Timer1 Synchronization Control bit
- 1 = Do not synchronize asynchronous clock input
- 0 = Synchronize asynchronous clock input with system clock (FOSC)

bit 1 **Unimplemented:** Read as '0'

bit 0 **TMR1ON:** Timer1 On bit
- 1 = Enables Timer1
- 0 = Stops Timer1 and clears Timer1 gate flip-flop

TABLE 19-1: SUMMARY OF REGISTERS ASSOCIATED WITH TIMER2/4

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	84
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	SSP2IE	TMR2IE	TMR1IE	85
PIE2	—	AD2IE	—	—	BCL1IE	BCL2IE	TMR4IE	—	86
PIR1	TMR1GIF	AD1IF	RCIF	TXIF	SSP1IF	SSP2IF	TMR2IF	TMR1IF	87
PIR2	—	AD2IF	—	—	BCL1IF	BCL2IF	TMR4IF	—	88
PR2	Timer2 Module Period Register								194*
PR4	Timer4 Module Period Register								194*
T2CON	—	T2OUTPS<3:0>				TMR2ON	T2CKPS1	T2CKPS0	195
T4CON	—	T4OUTPS<3:0>				TMR4ON	T4CKPS1	T4CKPS0	195
TMR2	Holding Register for the 8-bit TMR2 Register								193*
TMR4	Holding Register for the 8-bit TMR4 Register(1)								193*

Legend: — = unimplemented location, read as '0'. Shaded cells are not used for Timer2/4 module.

* Page provides register information.

FIGURE 20-14: I²C SLAVE, 7-BIT ADDRESS, RECEPTION (SEN = 0, AHEN = 0, DHEN = 0)

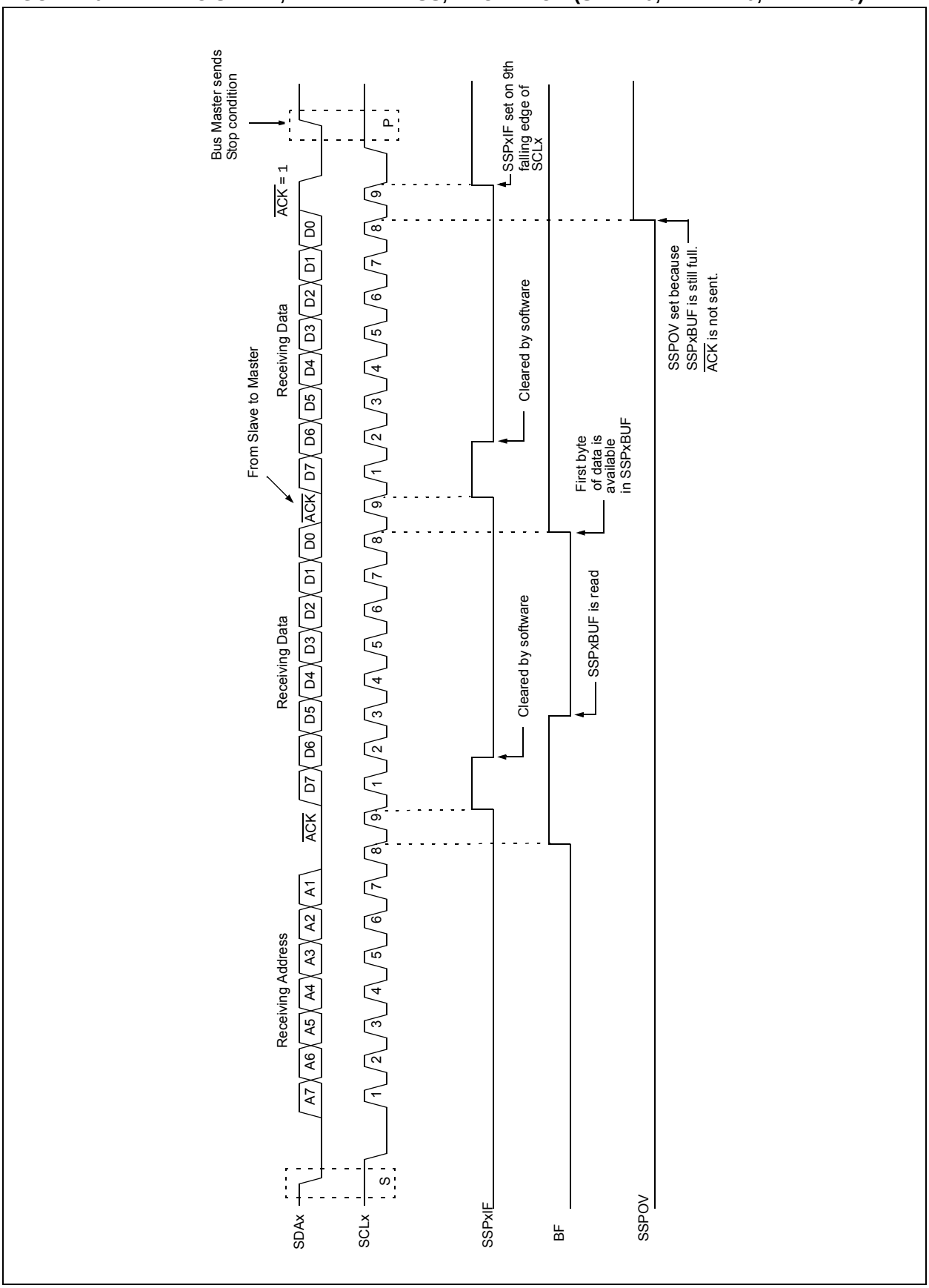


FIGURE 20-18: I²C SLAVE, 7-BIT ADDRESS, TRANSMISSION (AHEN = 0)

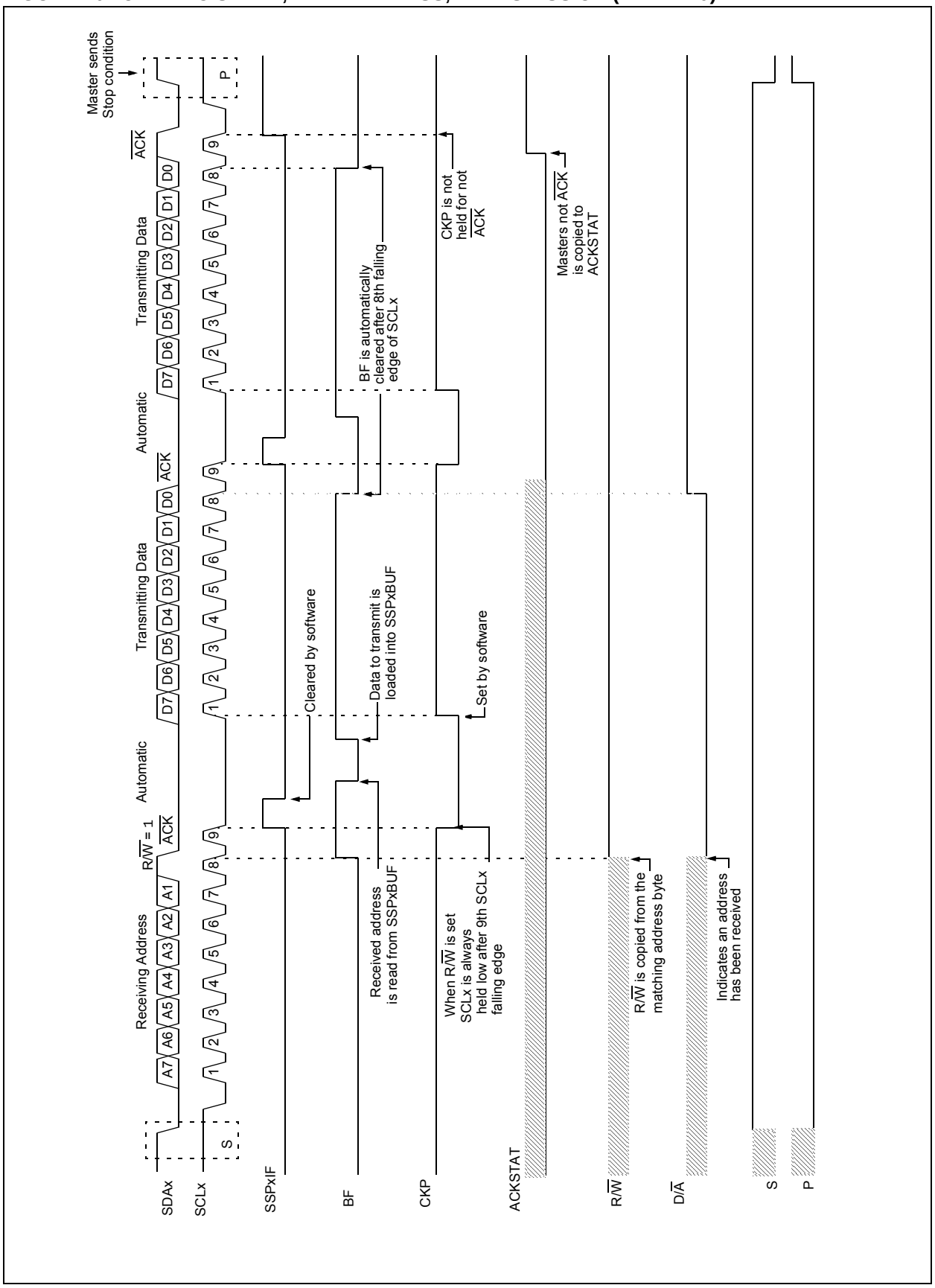


FIGURE 20-34: BUS COLLISION DURING START CONDITION (SCLX = 0)

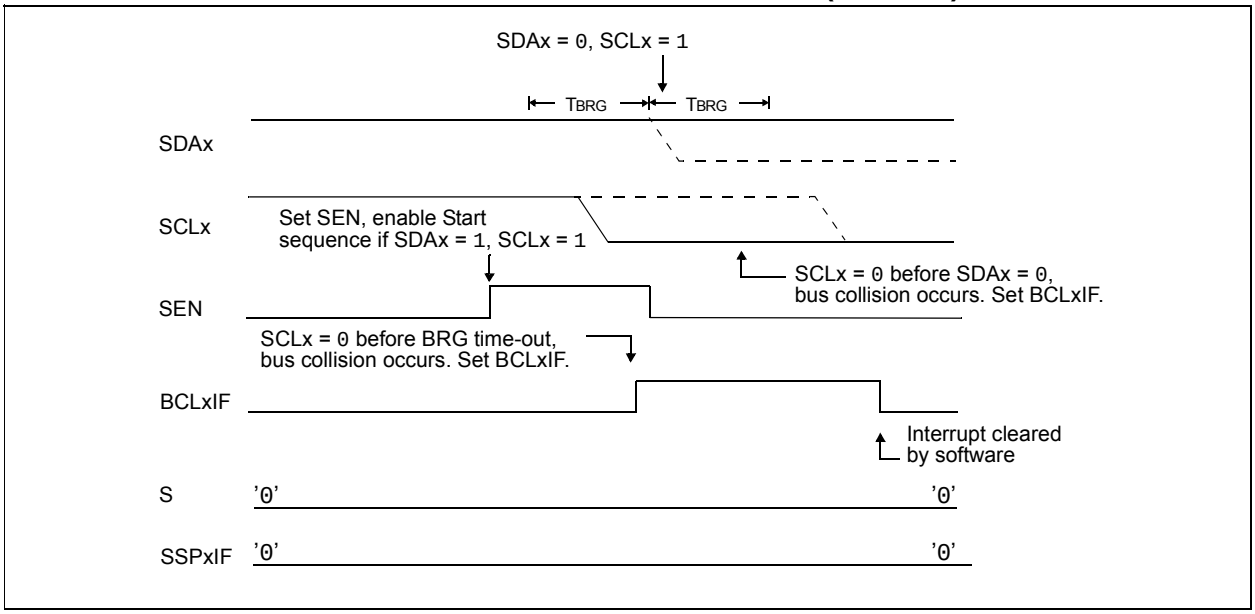


FIGURE 20-35: BRG RESET DUE TO SDA ARBITRATION DURING START CONDITION

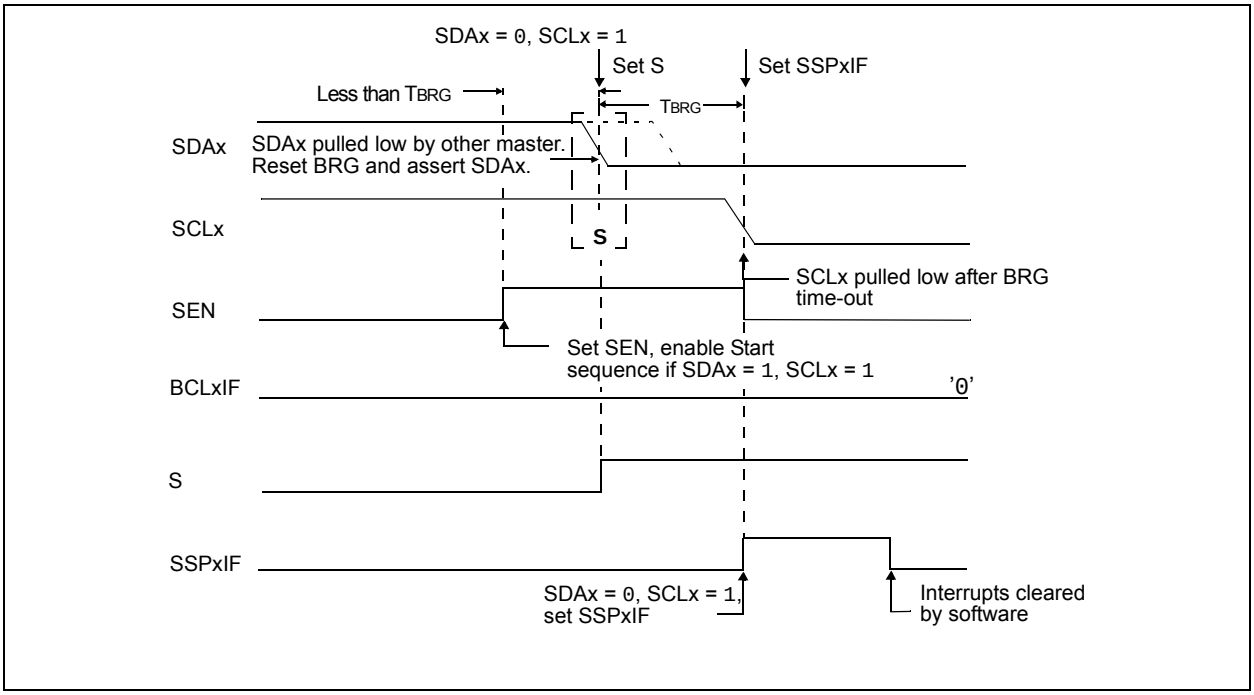


TABLE 22-4: SUMMARY OF REGISTERS ASSOCIATED WITH PWM

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
PR2	Timer2 module Period Register								193*
PWM1CON	PWM1EN	PWM1OE	PWM1OUT	PWM1POL	—	—	—	—	286
PWM1DCH	PWM1DCH<7:0>								287
PWM1DCL	PWM1DCL<7:6>		—	—	—	—	—	—	287
PWM2CON	PWM2EN	PWM2OE	PWM2OUT	PWM2POL	—	—	—	—	286
PWM2DCH	PWM2DCH<7:0>								287
PWM2DCL	PWM2DCL<7:6>		—	—	—	—	—	—	287
T2CON	—	T2OUTPS<3:0>				TMR2ON	T2CKPS<1:0>		195
TMR2	Timer2 module Register								193*
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	115
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	123

Legend: - = Unimplemented locations, read as '0', u = unchanged, x = unknown. Shaded cells are not used by the PWM.

* Page provides register information.

24.2 Instruction Descriptions

ADDFSR — Add Literal to FSRn

Syntax:	[*label*] ADDFSR FSRn, k
Operands:	-32 ≤ k ≤ 31 n ∈ [0, 1]
Operation:	FSR(n) + k → FSR(n)
Status Affected:	None
Description:	The signed 6-bit literal ‘k’ is added to the contents of the FSRnH:FSRnL register pair. FSRn is limited to the range 0000h - FFFFh. Moving beyond these bounds will cause the FSR to wrap-around.

ADDLW — Add literal and W

Syntax:	[*label*] ADDLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) + k → (W)
Status Affected:	C, DC, Z
Description:	The contents of the W register are added to the 8-bit literal ‘k’ and the result is placed in the W register.

ADDWF — Add W and f

Syntax:	[*label*] ADDWF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(W) + (f) → (destination)
Status Affected:	C, DC, Z
Description:	Add the contents of the W register with register ‘f’. If ‘d’ is ‘0’, the result is stored in the W register. If ‘d’ is ‘1’, the result is stored back in register ‘f’.

ADDWFC — ADD W and CARRY bit to f

Syntax:	[*label*] ADDWFC f {,d}
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(W) + (f) + (C) → dest
Status Affected:	C, DC, Z
Description:	Add W, the Carry flag and data memory location ‘f’. If ‘d’ is ‘0’, the result is placed in W. If ‘d’ is ‘1’, the result is placed in data memory location ‘f’.

ANDLW — AND literal with W

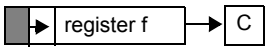
Syntax:	[*label*] ANDLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) .AND. (k) → (W)
Status Affected:	Z
Description:	The contents of W register are AND’ed with the 8-bit literal ‘k’. The result is placed in the W register.

ANDWF — AND W with f

Syntax:	[*label*] ANDWF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(W) .AND. (f) → (destination)
Status Affected:	Z
Description:	AND the W register with register ‘f’. If ‘d’ is ‘0’, the result is stored in the W register. If ‘d’ is ‘1’, the result is stored back in register ‘f’.

ASRF — Arithmetic Right Shift

Syntax:	[*label*] ASRF f {,d}
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f<7>)→ dest<7> (f<7:1>) → dest<6:0>, (f<0>) → C,
Status Affected:	C, Z
Description:	The contents of register ‘f’ are shifted one bit to the right through the Carry flag. The MSb remains unchanged. If ‘d’ is ‘0’, the result is placed in W. If ‘d’ is ‘1’, the result is stored back in register ‘f’.



DECFSZ **Decrement f, Skip if 0**

Syntax:	[*label*] DECFSZ f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f) - 1 → (destination); skip if result = 0
Status Affected:	None
Description:	The contents of register 'f' are decremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'. If the result is '1', the next instruction is executed. If the result is '0', then a NOP is executed instead, making it a 2-cycle instruction.

GOTO **Unconditional Branch**

Syntax:	[*label*] GOTO k
Operands:	$0 \le k \le 2047$
Operation:	k → PC<10:0> PCLATH<6:3> → PC<14:11>
Status Affected:	None
Description:	GOTO is an unconditional branch. The 11-bit immediate value is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH<4:3>. GOTO is a 2-cycle instruction.

INCF **Increment f**

Syntax:	[*label*] INCF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f) + 1 → (destination)
Status Affected:	Z
Description:	The contents of register 'f' are incremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.

INCFSZ **Increment f, Skip if 0**

Syntax:	[*label*] INCFSZ f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f) + 1 → (destination), skip if result = 0
Status Affected:	None
Description:	The contents of register 'f' are incremented. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'. If the result is '1', the next instruction is executed. If the result is '0', a NOP is executed instead, making it a 2-cycle instruction.

IORLW **Inclusive OR literal with W**

Syntax:	[*label*] IORLW k
Operands:	$0 \le k \le 255$
Operation:	(W) .OR. k → (W)
Status Affected:	Z
Description:	The contents of the W register are OR'ed with the 8-bit literal 'k'. The result is placed in the W register.

IORWF **Inclusive OR W with f**

Syntax:	[*label*] IORWF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(W) .OR. (f) → (destination)
Status Affected:	Z
Description:	Inclusive OR the W register with register 'f'. If 'd' is '0', the result is placed in the W register. If 'd' is '1', the result is placed back in register 'f'.

FIGURE 25-13: SPI MASTER MODE TIMING (CKE = 0, SMP = 0)

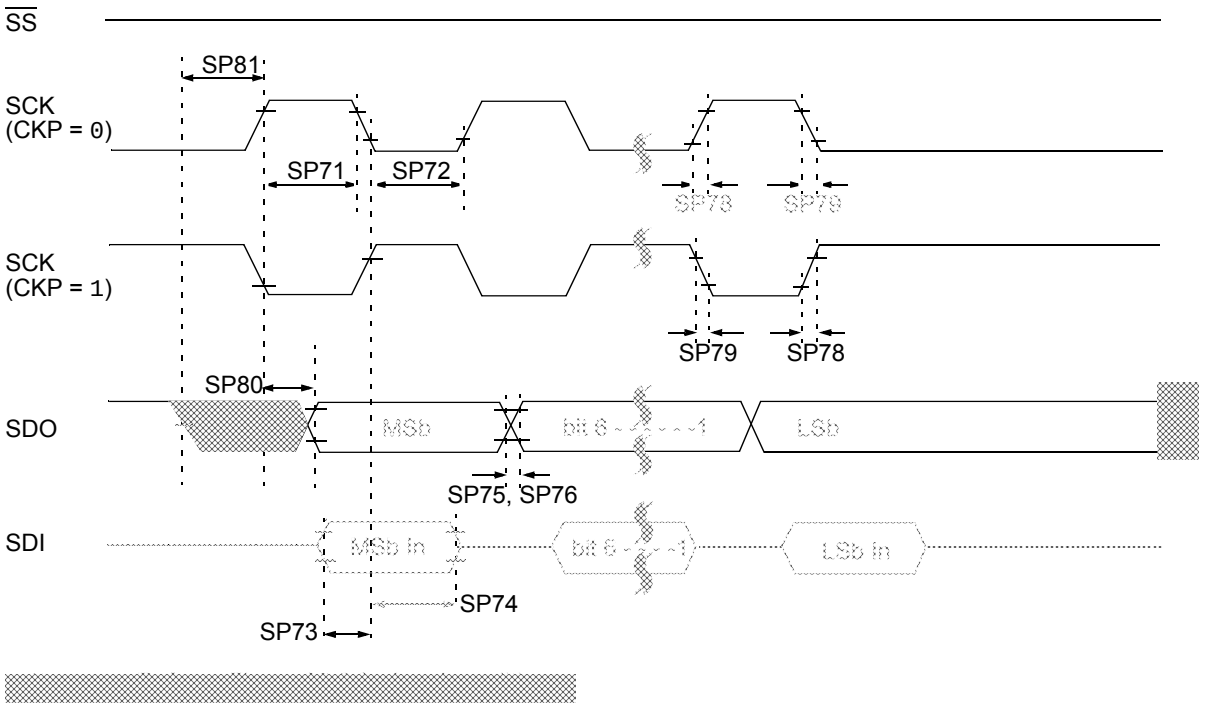
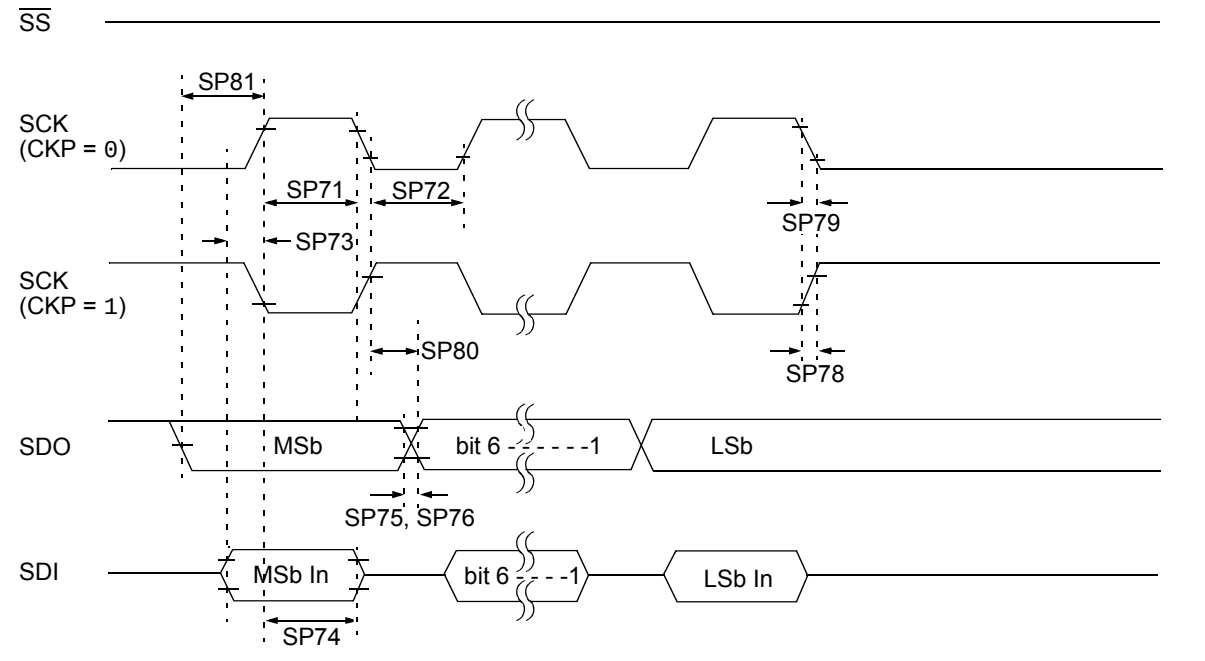


FIGURE 25-14: SPI MASTER MODE TIMING (CKE = 1, SMP = 1)



Note: Refer to Figure 25-3 for Load conditions.