

Welcome to [E-XFL.COM](https://www.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	I ² C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	17
Program Memory Size	14KB (8K x 14)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 17x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Through Hole
Package / Case	20-DIP (0.300", 7.62mm)
Supplier Device Package	20-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf1559-e-p

Table of Contents

1.0	Device Overview	9
2.0	Enhanced Mid-Range CPU	16
3.0	Memory Organization	18
4.0	Device Configuration	52
5.0	Oscillator Module.....	56
6.0	Resets	64
7.0	Interrupts	72
8.0	Power-Down Mode (Sleep)	82
9.0	Watchdog Timer (WDT)	84
10.0	Flash Program Memory Control	88
11.0	I/O Ports	105
12.0	Interrupt-on-Change	118
13.0	Fixed Voltage Reference (FVR)	123
14.0	Temperature Indicator Module	125
15.0	Analog-to-Digital Converter (ADC) Module	127
16.0	Hardware Capacitive Voltage Divider (CVD) Module.....	141
17.0	Timer0 Module	164
18.0	Timer1 Module with Gate Control.....	167
19.0	Timer2 Module	178
20.0	Master Synchronous Serial Port (MSSP) Module	181
21.0	Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART).....	234
22.0	Pulse-Width Modulation (PWM) Module	261
23.0	In-Circuit Serial Programming™ (ICSP™)	267
24.0	Instruction Set Summary	269
25.0	Electrical Specifications.....	283
26.0	DC and AC Characteristics Graphs and Charts	305
27.0	Development Support.....	310
28.0	Packaging Information.....	314
Appendix A: Data Sheet Revision History.....		336
The Microchip Website		337
Customer Change Notification Service		337
Customer Support.....		337
Product Identification System		338

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at docerrors@microchip.com. We welcome your feedback.

Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Website at:

<http://www.microchip.com>

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000000A is version A of document DS30000000).

Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Website; <http://www.microchip.com>
- Your local Microchip sales office (see last page)

When contacting a sales office, please specify which device, revision of silicon and data sheet (include literature number) you are using.

Customer Notification System

Register on our website at www.microchip.com to receive the most current information on all of our products.

3.2.6 CORE FUNCTION REGISTERS SUMMARY

The Core Function registers listed in Table 3-8 can be addressed from any Bank.

TABLE 3-8: CORE FUNCTION REGISTERS SUMMARY

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 0-31											
x00h or x80h	INDF0	Addressing this location uses contents of FSR0H/FSR0L to address data memory (not a physical register)								xxxx xxxx	uuuu uuuu
x01h or x81h	INDF1	Addressing this location uses contents of FSR1H/FSR1L to address data memory (not a physical register)								xxxx xxxx	uuuu uuuu
x02h or x82h	PCL	Program Counter (PC) Least Significant Byte								0000 0000	0000 0000
x03h or x83h	STATUS	—	—	—	\overline{TO}	\overline{PD}	Z	DC	C	---1 1000	---q quuu
x04h or x84h	FSR0L	Indirect Data Memory Address 0 Low Pointer								0000 0000	uuuu uuuu
x05h or x85h	FSR0H	Indirect Data Memory Address 0 High Pointer								0000 0000	0000 0000
x06h or x86h	FSR1L	Indirect Data Memory Address 1 Low Pointer								0000 0000	uuuu uuuu
x07h or x87h	FSR1H	Indirect Data Memory Address 1 High Pointer								0000 0000	0000 0000
x08h or x88h	BSR	—	—	—	BSR<4:0>					---0 0000	---0 0000
x09h or x89h	WREG	Working Register								0000 0000	uuuu uuuu
x0Ah or x8Ah	PCLATH	—	Write Buffer for the upper 7 bits of the Program Counter							-000 0000	-000 0000
x0Bh or x8Bh	INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	0000 0000	0000 0000

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

PIC16LF1554/1559

TABLE 3-9: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Address	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 3											
180h	INDF0 ⁽¹⁾	Addressing this location uses contents of FSR0H/FSR0L to address data memory (not a physical register)								xxxx xxxx	uuuu uuuu
181h	INDF1 ⁽¹⁾	Addressing this location uses contents of FSR1H/FSR1L to address data memory (not a physical register)								xxxx xxxx	uuuu uuuu
182h	PCL ⁽¹⁾	Program Counter (PC) Least Significant Byte								0000 0000	0000 0000
183h	STATUS ⁽¹⁾	—	—	—	TO	PD	Z	DC	C	---1 1000	---q quuu
184h	FSR0L ⁽¹⁾	Indirect Data Memory Address 0 Low Pointer								0000 0000	uuuu uuuu
185h	FSR0H ⁽¹⁾	Indirect Data Memory Address 0 High Pointer								0000 0000	0000 0000
186h	FSR1L ⁽¹⁾	Indirect Data Memory Address 1 Low Pointer								0000 0000	uuuu uuuu
187h	FSR1H ⁽¹⁾	Indirect Data Memory Address 1 High Pointer								0000 0000	0000 0000
188h	BSR ⁽¹⁾	—	—	—	BSR<4:0>					---0 0000	---0 0000
189h	WREG ⁽¹⁾	Working Register								0000 0000	uuuu uuuu
18Ah	PCLATH ⁽¹⁾	—	Write Buffer for the upper 7 bits of the Program Counter							-000 0000	-000 0000
18Bh	INTCON ⁽¹⁾	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	0000 0000	0000 0000
18Ch	ANSELA	—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	1111 -111	1111 -111
18Dh	ANSELB ⁽²⁾	Unimplemented								—	—
	ANSELB ⁽³⁾	ANSB7	ANSB6	ANSB5	ANSB4	—	—	—	—	1111 ----	1111 ----
18Eh	ANSELC ⁽²⁾	—	—	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	--11 1111	--11 1111
	ANSELC ⁽³⁾	ANSC7	ANSC6	ANSC5	ANSC4	ANSC3	ANSC2	ANSC1	ANSC0	1111 1111	1111 1111
18Fh	—	Unimplemented								—	—
190h	—	Unimplemented								—	—
191h	PMADRL	Program Memory Address Register Low Byte								0000 0000	0000 0000
192h	PMADRH	—	Program Memory Address Register High Byte							1000 0000	1000 0000
193h	PMDATL	Program Memory Read Data Register Low Byte								xxxx xxxx	uuuu uuuu
194h	PMDATH	—	—	Program Memory Read Data Register High Byte					--xx xxxx	--uu uuuu	
195h	PMCON1	—	CFG5	LWLO	FREE	WRERR	WREN	WR	RD	-000 x000	-000 q000
196h	PMCON2	Program Memory Control Register 2								0000 0000	0000 0000
197h	—	Unimplemented								—	—
198h	—	Unimplemented								—	—
199h	RCREG	USART Receive Data Register								0000 0000	0000 0000
19Ah	TXREG	USART Transmit Data Register								0000 0000	0000 0000
19Bh	SPBRGL	SPBRG Low								0000 0000	0000 0000
19Ch	SPBRGH	SPBRG High								0000 0000	0000 0000
19Dh	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
19Eh	TXSTA	CSRC	TX9	TXEN	SYNC	SEnDB	BRGH	TRMT	TX9D	0000 0010	0000 0010
19Fh	BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	01-0 0-00

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: PIC16LF1554.
 - 3: PIC16LF1559.
 - 4: These registers/bits are available at two address locations, in Bank 1 and Bank 14.

PIC16LF1554/1559

TABLE 3-9: SPECIAL FUNCTION REGISTER SUMMARY (CONTINUED)

Address	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 31											
F80h	INDF0 ⁽¹⁾	Addressing this location uses contents of FSR0H/FSR0L to address data memory (not a physical register)								xxxx xxxx	uuuu uuuu
F81h	INDF1 ⁽¹⁾	Addressing this location uses contents of FSR1H/FSR1L to address data memory (not a physical register)								xxxx xxxx	uuuu uuuu
F82h	PCL ⁽¹⁾	Program Counter (PC) Least Significant Byte								0000 0000	0000 0000
F83h	STATUS ⁽¹⁾	—	—	—	\overline{TO}	\overline{PD}	Z	DC	C	---1 1000	---q quuu
F84h	FSR0L ⁽¹⁾	Indirect Data Memory Address 0 Low Pointer								0000 0000	uuuu uuuu
F85h	FSR0H ⁽¹⁾	Indirect Data Memory Address 0 High Pointer								0000 0000	0000 0000
F86h	FSR1L ⁽¹⁾	Indirect Data Memory Address 1 Low Pointer								0000 0000	uuuu uuuu
F87h	FSR1H ⁽¹⁾	Indirect Data Memory Address 1 High Pointer								0000 0000	0000 0000
F88h	BSR ⁽¹⁾	—	—	—	BSR<4:0>					---0 0000	---0 0000
F89h	WREG ⁽¹⁾	Working Register								0000 0000	uuuu uuuu
F8Ah	PCLATH ⁽¹⁾	—	Write Buffer for the upper 7 bits of the Program Counter							-000 0000	-000 0000
F8Bh	INTCON ⁽¹⁾	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	0000 0000	0000 0000
F8Ch — FE2h	—	Unimplemented								—	—
FE3h	—	Unimplemented								—	—
FE4h	STATUS_ SHAD	—	—	—	—	—	Z	DC	C	---- -xxx	---- -uuu
FE5h	WREG_SHAD	Working Register Normal (Non-ICD) Shadow								xxxx xxxx	uuuu uuuu
FE6h	BSR_SHAD	—	—	—	Bank Select Register Normal (Non-ICD) Shadow					---x xxxx	---u uuuu
FE7h	PCLATH_ SHAD	—	Program Counter Latch High Register Normal (Non-ICD) Shadow							-xxx xxxx	uuuu uuuu
FE8h	FSR0L_SHAD	Indirect Data Memory Address 0 Low Pointer Normal (Non-ICD) Shadow								xxxx xxxx	uuuu uuuu
FE9h	FSR0H_SHAD	Indirect Data Memory Address 0 High Pointer Normal (Non-ICD) Shadow								xxxx xxxx	uuuu uuuu
FEAh	FSR1L_SHAD	Indirect Data Memory Address 1 Low Pointer Normal (Non-ICD) Shadow								xxxx xxxx	uuuu uuuu
FEBh	FSR1H_SHAD	Indirect Data Memory Address 1 High Pointer Normal (Non-ICD) Shadow								xxxx xxxx	uuuu uuuu
FECh	—	Unimplemented								—	—
FEDh	STKPTR	—	—	—	Current Stack pointer					---1 1111	---1 1111
FEEh	TOSL	Top of Stack Low byte								xxxx xxxx	uuuu uuuu
FEFh	TOSH	—	Top of Stack High byte							-xxx xxxx	-uuu uuuu

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: PIC16LF1554.

3: PIC16LF1559.

4: These registers/bits are available at two address locations, in Bank 1 and Bank 14.

FIGURE 15-4: ANALOG INPUT MODEL

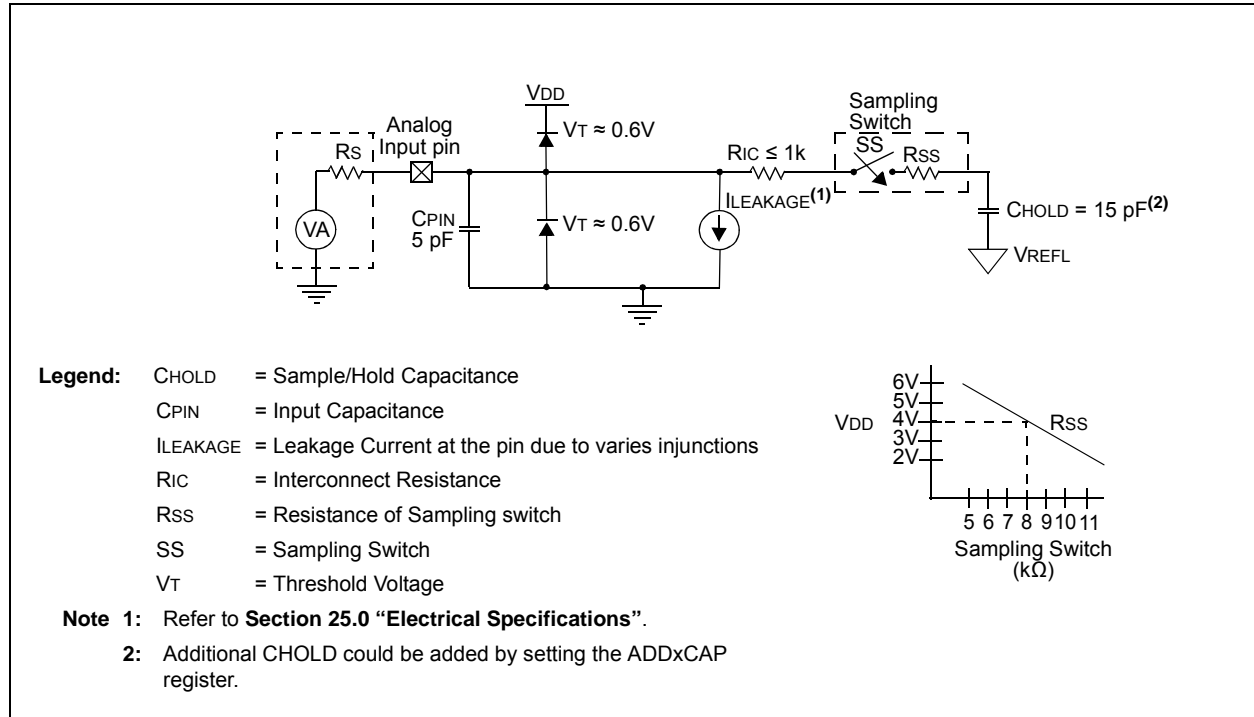
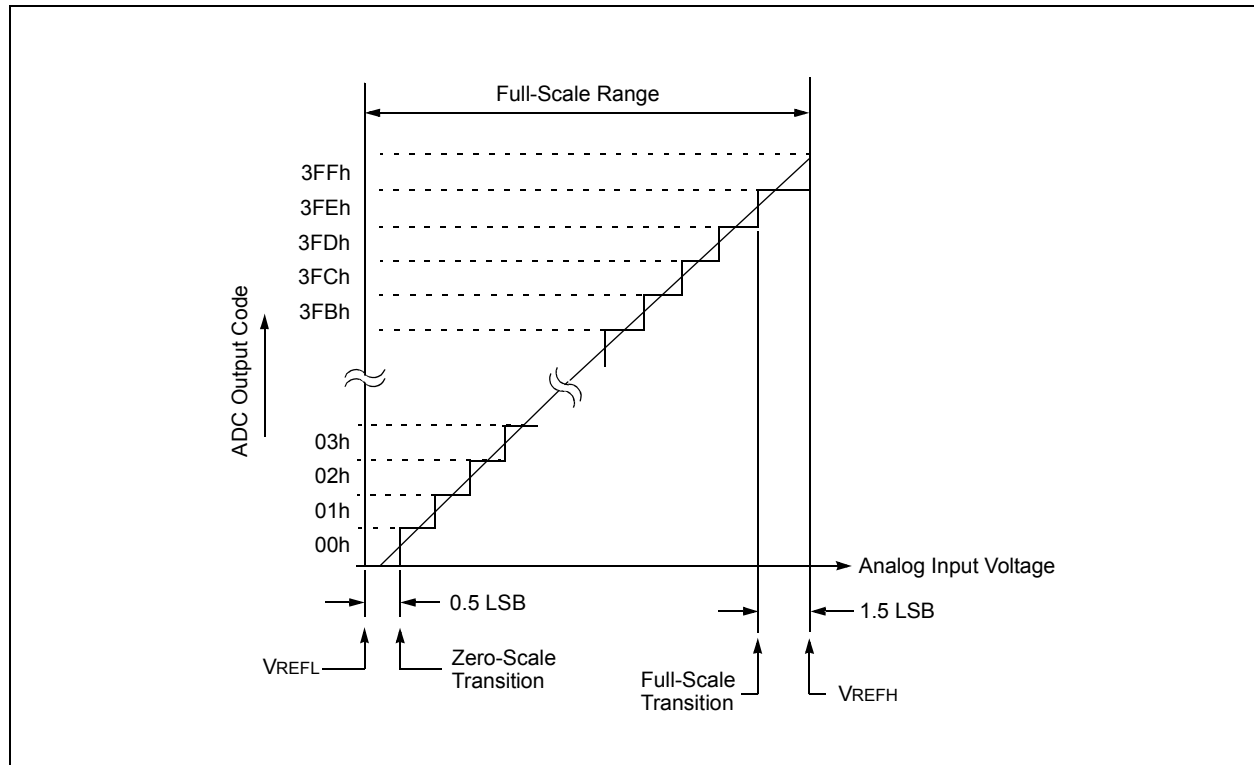


FIGURE 15-5: ADC TRANSFER FUNCTION



PIC16LF1554/1559

17.0 TIMER0 MODULE

The Timer0 module is an 8-bit timer/counter with the following features:

- 8-bit timer/counter register (TMR0)
- 3-bit prescaler (independent of Watchdog Timer)
- Programmable internal or external clock source
- Programmable external clock edge selection
- Interrupt on overflow
- TMR0 can be used to gate Timer1

Figure 17-1 is a block diagram of the Timer0 module.

17.1 Timer0 Operation

The Timer0 module can be used as either an 8-bit timer or an 8-bit counter.

17.1.1 8-BIT TIMER MODE

The Timer0 module will increment every instruction cycle, if used without a prescaler. 8-bit Timer mode is selected by clearing the TMR0CS bit of the OPTION_REG register.

When TMR0 is written, the increment is inhibited for two instruction cycles immediately following the write.

Note: The value written to the TMR0 register can be adjusted, in order to account for the two instruction cycle delay when TMR0 is written.

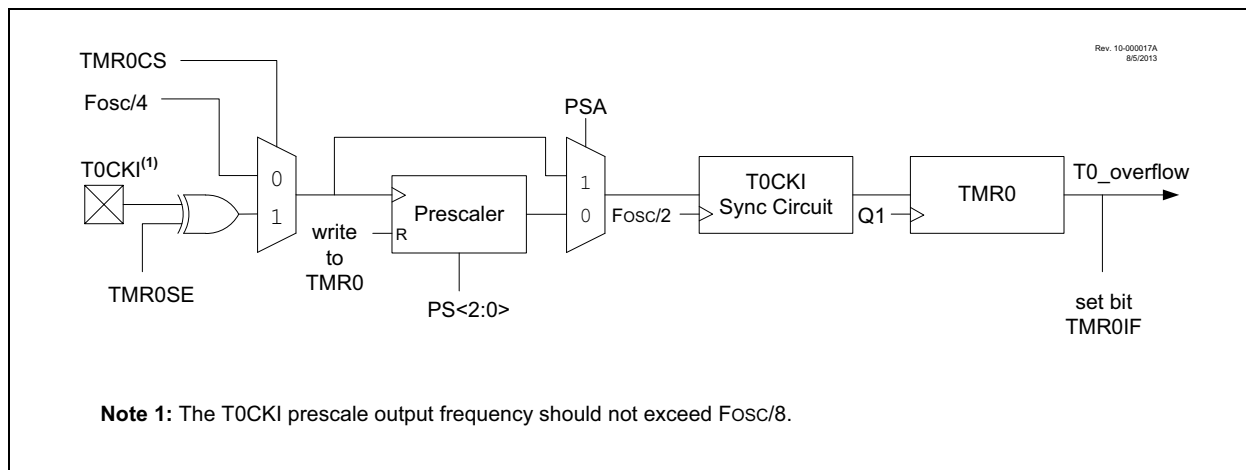
17.1.2 8-BIT COUNTER MODE

In 8-Bit Counter mode, the Timer0 module will increment on every rising or falling edge of the T0CKI pin.

8-Bit Counter mode using the T0CKI pin is selected by setting the TMR0CS bit in the OPTION_REG register to '1'.

The rising or falling transition of the incrementing edge for either input source is determined by the TMR0SE bit in the OPTION_REG register.

FIGURE 17-1: TIMER0 BLOCK DIAGRAM



18.6 Timer1 Interrupt

The Timer1 register pair (TMR1H:TMR1L) increments to FFFFh and rolls over to 0000h. When Timer1 rolls over, the Timer1 interrupt flag bit of the PIR1 register is set. To enable the interrupt on rollover, you must set these bits:

- TMR1ON bit of the T1CON register
- TMR1IE bit of the PIE1 register
- PEIE bit of the INTCON register
- GIE bit of the INTCON register

The interrupt is cleared by clearing the TMR1IF bit in the Interrupt Service Routine.

Note: The TMR1H:TMR1L register pair and the TMR1IF bit should be cleared before enabling interrupts.

18.7 Timer1 Operation During Sleep

Timer1 can only operate during Sleep when setup in Asynchronous Counter mode. In this mode, an external crystal or clock source can be used to increment the counter. To set up the timer to wake the device:

- TMR1ON bit of the T1CON register must be set
- TMR1IE bit of the PIE1 register must be set
- PEIE bit of the INTCON register must be set
- T1SYNC bit of the T1CON register must be set
- TMR1CS bits of the T1CON register must be configured

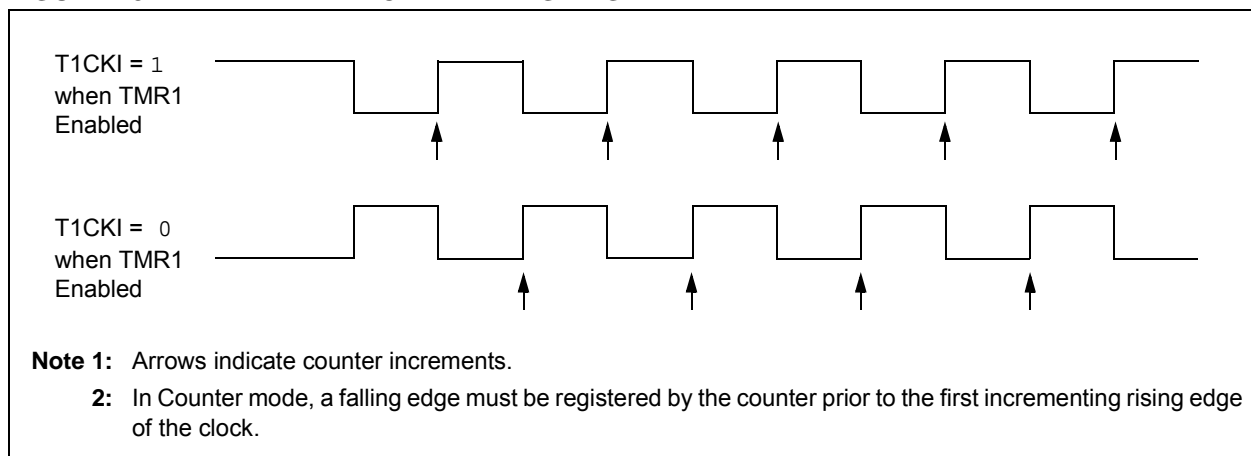
The device will wake-up on an overflow and execute the next instructions. If the GIE bit of the INTCON register is set, the device will call the Interrupt Service Routine.

Timer1 oscillator will continue to operate in Sleep regardless of the T1SYNC bit setting.

18.7.1 ALTERNATE PIN LOCATIONS

This module incorporates I/O pins that can be moved to other locations with the use of the alternate pin function register, APFCON. To determine which pins can be moved and what their default locations are upon a Reset, see **Section 11.1 “Alternate Pin Function”** for more information.

FIGURE 18-2: TIMER1 INCREMENTING EDGE



PIC16LF1554/1559

19.0 TIMER2 MODULE

The Timer2 module incorporates the following features:

- 8-bit Timer and Period registers (TMR2 and PR2, respectively)
- Readable and writable (both registers)
- Software programmable prescaler (1:1, 1:4, 1:16, and 1:64)
- Software programmable postscaler (1:1 to 1:16)
- Interrupt on TMR2 match with PR2

See Figure 19-1 for a block diagram of Timer2.

FIGURE 19-1: TIMER2 BLOCK DIAGRAM

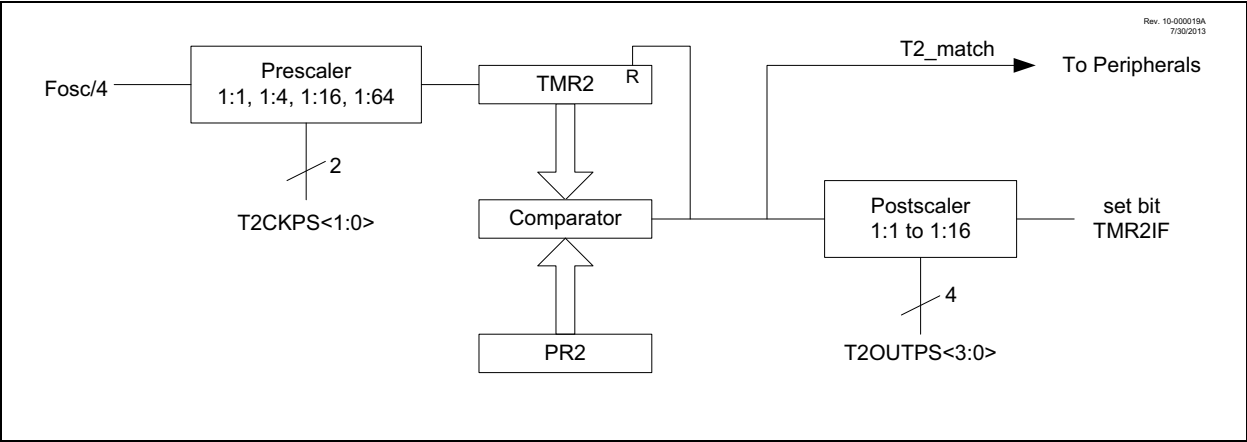
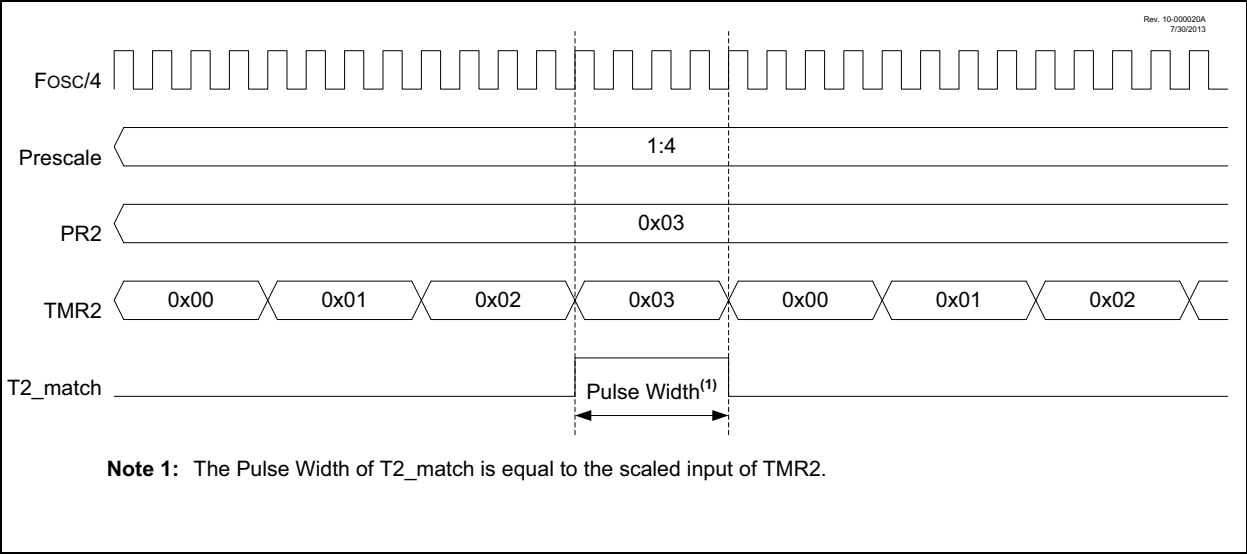


FIGURE 19-2: TIMER2 TIMING DIAGRAM



19.1 Timer2 Operation

The clock input to the Timer2 module is the system instruction clock ($F_{osc}/4$).

TMR2 increments from 00h on each clock edge.

A 4-bit counter/prescaler on the clock input allows direct input, divide-by-4 and divide-by-16 prescale options. These options are selected by the prescaler control bits, $T2CKPS<1:0>$ of the T2CON register. The value of TMR2 is compared to that of the Period register, PR2, on each clock cycle. When the two values match, the comparator generates a match signal as the timer output. This signal also resets the value of TMR2 to 00h on the next cycle and drives the output counter/postscaler (see **Section 19.2 “Timer2 Interrupt”**).

The TMR2 and PR2 registers are both directly readable and writable. The TMR2 register is cleared on any device Reset, whereas the PR2 register initializes to FFh. Both the prescaler and postscaler counters are cleared on the following events:

- a write to the TMR2 register
- a write to the T2CON register
- Power-On Reset (POR)
- Brown-Out Reset (BOR)
- \overline{MCLR} Reset
- Watchdog Timer (WDT) Reset
- Stack Overflow Reset
- Stack Underflow Reset
- RESET Instruction

Note: TMR2 is not cleared when T2CON is written.

19.2 Timer2 Interrupt

Timer2 can also generate an optional device interrupt. The Timer2 output signal (T2_match) provides the input for the 4-bit counter/postscaler. This counter generates the TMR2 match interrupt flag which is latched in TMR2IF of the PIR1 register. The interrupt is enabled by setting the TMR2 Match Interrupt Enable bit, TMR2IE of the PIE1 register.

A range of 16 postscale options (from 1:1 through 1:16 inclusive) can be selected with the postscaler control bits, $T2OUTPS<3:0>$, of the T2CON register.

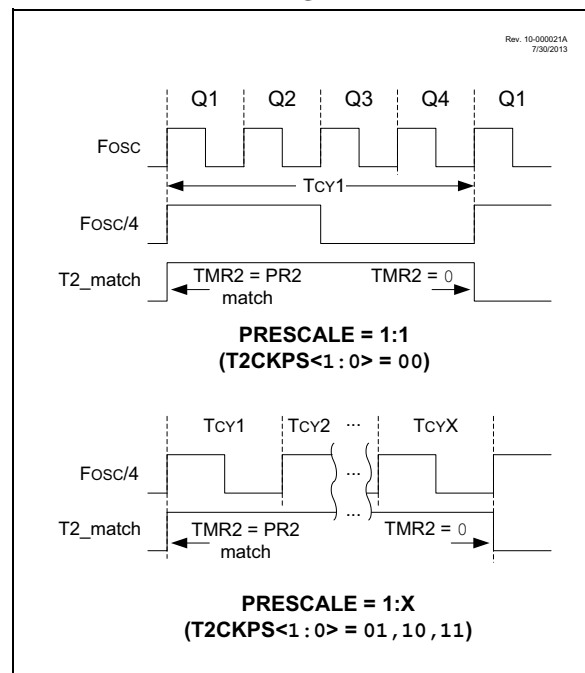
19.3 Timer2 Output

The output of TMR2 is T2_match. T2_match is available to the following peripherals:

- Configurable Logic Cell (CLC)
- Master Synchronous Serial Port (MSSP)
- Numerically Controlled Oscillator (NCO)
- Pulse Width Modulator (PWM)

The T2_match signal is synchronous with the system clock. Figure 19-3 shows two examples of the timing of the T2_match signal relative to F_{osc} and prescale value, $T2CKPS<1:0>$. The upper diagram illustrates 1:1 prescale timing and the lower diagram, 1:X prescale timing.

FIGURE 19-3: T2_MATCH TIMING DIAGRAM



19.4 Timer2 Operation During Sleep

Timer2 cannot be operated while the processor is in Sleep mode. The contents of the TMR2 and PR2 registers will remain unchanged while the processor is in Sleep mode.

PIC16LF1554/1559

20.2.3 SPI MASTER MODE

The master can initiate the data transfer at any time because it controls the SCK line. The master determines when the slave (Processor 2, Figure 20-5) is to broadcast data by the software protocol.

In Master mode, the data is transmitted/received as soon as the SSPBUF register is written to. If the SPI is only going to receive, the SDO output could be disabled (programmed as an input). The SSPSR register will continue to shift in the signal present on the SDI pin at the programmed clock rate. As each byte is received, it will be loaded into the SSPBUF register as if a normal received byte (interrupts and Status bits appropriately set).

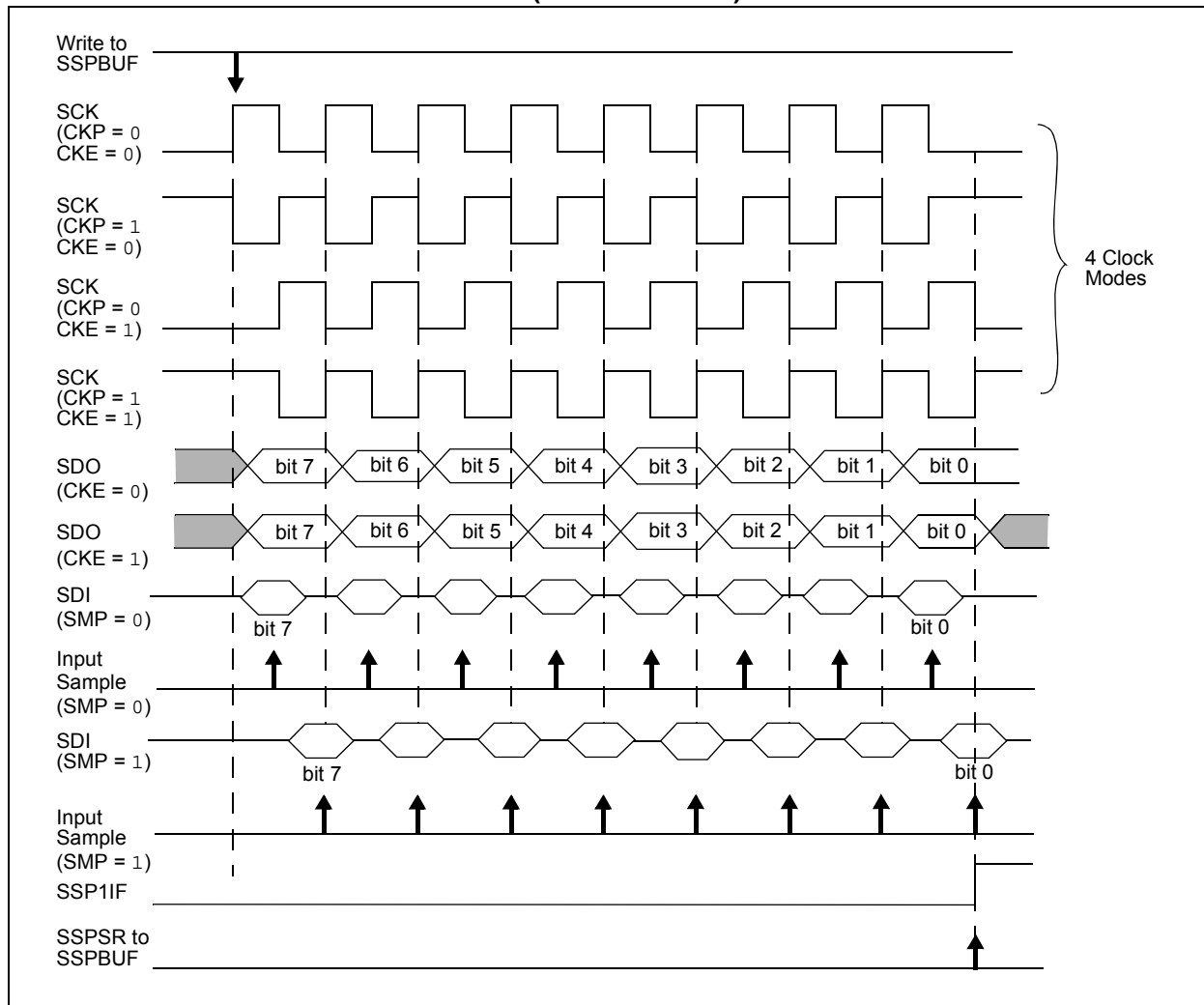
The clock polarity is selected by appropriately programming the CKP bit of the SSPCON1 register and the CKE bit of the SSPSTAT register. This then, would give waveforms for SPI communication as shown in Figure 20-6, Figure 20-8, Figure 20-9 and Figure 20-10, where the MSb is transmitted first. In Master mode, the SPI clock rate (bit rate) is user programmable to be one of the following:

- $F_{osc}/4$ (or T_{CY})
- $F_{osc}/16$ (or $4 * T_{CY}$)
- $F_{osc}/64$ (or $16 * T_{CY}$)
- $F_{osc}/(4 * (SSPADD + 1))$

Figure 20-6 shows the waveforms for Master mode.

When the CKE bit is set, the SDO data is valid before there is a clock edge on SCK. The change of the input sample is shown based on the state of the SMP bit. The time when the SSPBUF is loaded with the received data is shown.

FIGURE 20-6: SPI MODE WAVEFORM (MASTER MODE)



PIC16LF1554/1559

20.2.6 SPI OPERATION IN SLEEP MODE

In SPI Master mode, module clocks may be operating at a different speed than when in Full-Power mode; in the case of the Sleep mode, all clocks are halted.

Special care must be taken by the user when the MSSP clock is much faster than the system clock.

In Slave mode, when MSSP interrupts are enabled, after the master completes sending data, an MSSP interrupt will wake the controller from Sleep.

If an exit from Sleep mode is not desired, MSSP interrupts should be disabled.

In SPI Master mode, when the Sleep mode is selected, all module clocks are halted and the transmission/reception will remain in that state until the device wakes. After the device returns to Run mode, the module will resume transmitting and receiving data.

In SPI Slave mode, the SPI Transmit/Receive Shift register operates asynchronously to the device. This allows the device to be placed in Sleep mode and data to be shifted into the SPI Transmit/Receive Shift register. When all eight bits have been received, the MSSP interrupt flag bit will be set and if enabled, will wake the device.

TABLE 20-1: SUMMARY OF REGISTERS ASSOCIATED WITH SPI OPERATION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
ANSELA	—	—	ANSA5	ANSA4	—	ANSA2	ANSA1	ANSA0	109
APFCON	RXDTSEL	SDOSEL	SSSEL	SDSEL	—	TXCKSEL	GRDBSEL	GRDASEL	106
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	77
PIE1	TMR1GIE	AD1IE	RCIE	TXIE	SSP1IE	—	TMR2IE	TMR1IE	78
PIR1	TMR1GIF	AD1IF	RCIF	TXIF	SSP1IF	—	TMR2IF	TMR1IF	80
SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								185*
SSPCON1	WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>				230
SSPCON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	232
SSPSTAT	SMP	CKE	D/A	P	S	R/W	UA	BF	229
TRISA	—	—	TRISA5	TRISA4	— ⁽¹⁾	TRISA2	TRISA1	TRISA0	108

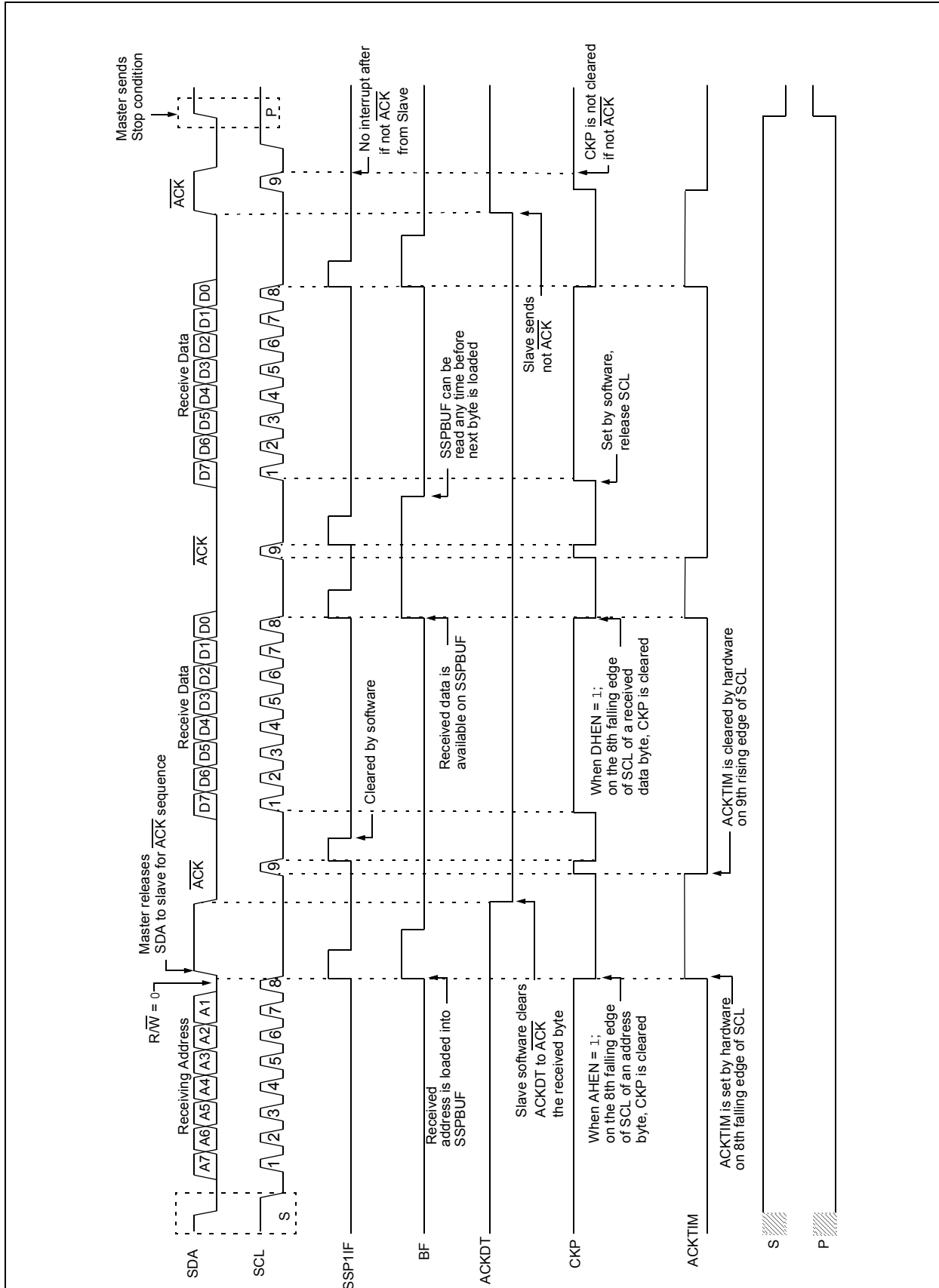
Legend: — = Unimplemented location, read as '0'. Shaded cells are not used by the MSSP in SPI mode.

* Page provides register information.

Note 1: Unimplemented, read as '1'.

PIC16LF1554/1559

FIGURE 20-17: I²C SLAVE, 7-BIT ADDRESS, RECEPTION (SEN = 1, AHEN = 1, DHEN = 1)



20.5.3.3 7-Bit Transmission with Address Hold Enabled

Setting the AHEN bit of the SSPCON3 register enables additional clock stretching and interrupt generation after the 8th falling edge of a received matching address. Once a matching address has been clocked in, CKP is cleared and the SSP1IF interrupt is set.

Figure 20-19 displays a standard waveform of a 7-bit Address Slave Transmission with AHEN enabled.

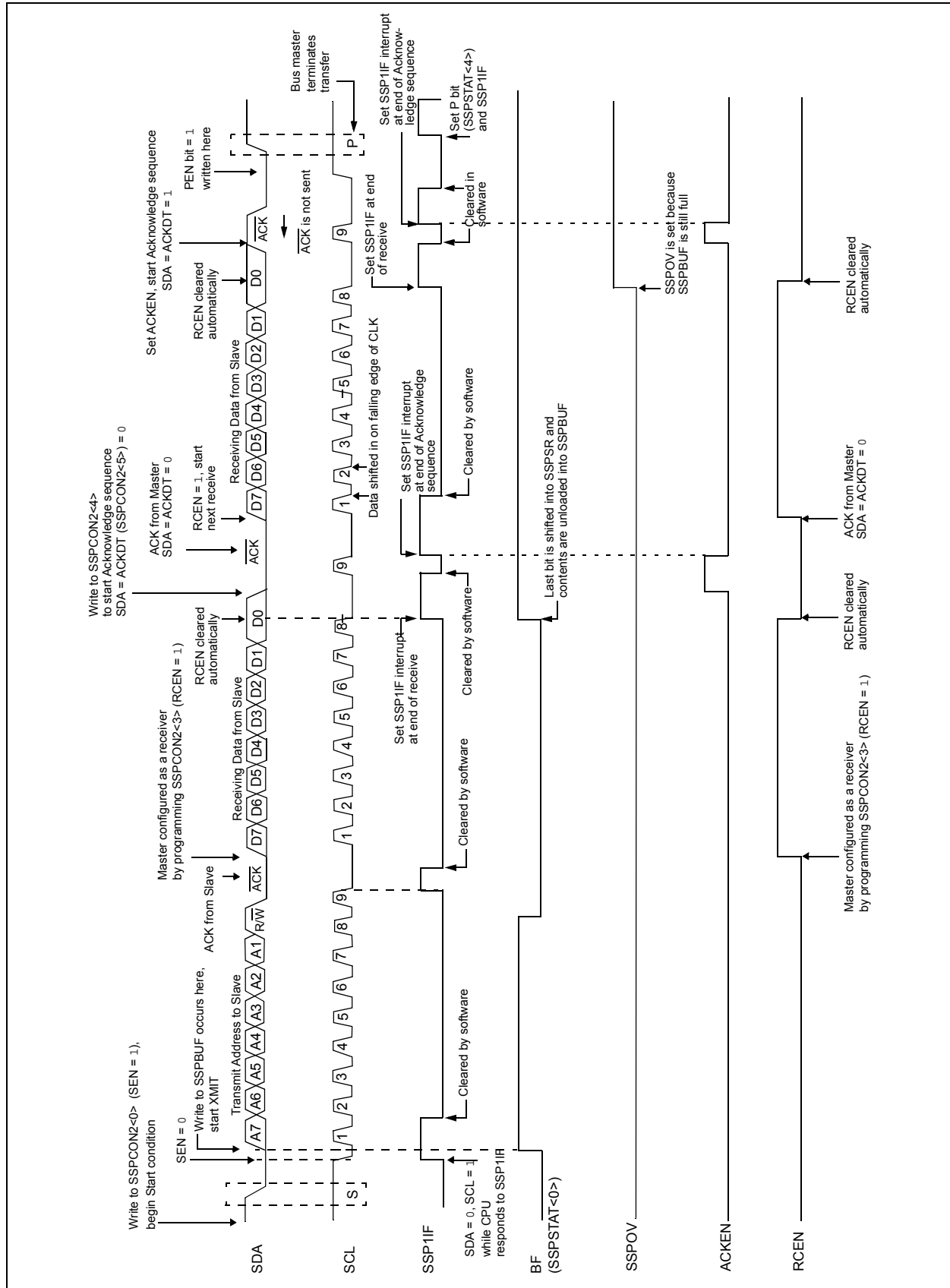
1. Bus starts Idle.
2. Master sends Start condition; the S bit of SSPSTAT is set; SSP1IF is set if interrupt on Start detect is enabled.
3. Master sends matching address with $\overline{R/\overline{W}}$ bit set. After the 8th falling edge of the SCL line the CKP bit is cleared and SSP1IF interrupt is generated.
4. Slave software clears SSP1IF.
5. Slave software reads ACKTIM bit of SSPCON3 register, and $\overline{R/\overline{W}}$ and $\overline{D/\overline{A}}$ of the SSPSTAT register to determine the source of the interrupt.
6. Slave reads the address value from the SSPBUF register clearing the BF bit.
7. Slave software decides from this information if it wishes to ACK or not ACK and sets the ACKDT bit of the SSPCON2 register accordingly.
8. Slave sets the CKP bit releasing SCL.
9. Master clocks in the \overline{ACK} value from the slave.
10. Slave hardware automatically clears the CKP bit and sets SSP1IF after the \overline{ACK} if the $\overline{R/\overline{W}}$ bit is set.
11. Slave software clears SSP1IF.
12. Slave loads value to transmit to the master into SSPBUF setting the BF bit.

Note: SSPBUF cannot be loaded until after the \overline{ACK} .

13. Slave sets CKP bit releasing the clock.
14. Master clocks out the data from the slave and sends an \overline{ACK} value on the 9th SCL pulse.
15. Slave hardware copies the \overline{ACK} value into the ACKSTAT bit of the SSPCON2 register.
16. Steps 10-15 are repeated for each byte transmitted to the master from the slave.
17. If the master sends a not \overline{ACK} the slave releases the bus allowing the master to send a Stop and end the communication.

Note: Master must send a not \overline{ACK} on the last byte to ensure that the slave releases the SCL line to receive a Stop.

FIGURE 20-29: I²C MASTER MODE WAVEFORM (RECEPTION, 7-BIT ADDRESS)



20.6.13.2 Bus Collision During a Repeated Start Condition

During a Repeated Start condition, a bus collision occurs if:

- A low level is sampled on SDA when SCL goes from low level to high level (Case 1).
- SCL goes low before SDA is asserted low, indicating that another master is attempting to transmit a data '1' (Case 2).

When the user releases SDA and the pin is allowed to float high, the BRG is loaded with SSPADD and counts down to zero. The SCL pin is then deasserted and when sampled high, the SDA pin is sampled.

If SDA is low, a bus collision has occurred (i.e., another master is attempting to transmit a data '0', Figure 20-36). If SDA is sampled high, the BRG is reloaded and begins counting. If SDA goes from high-to-low before the BRG times out, no bus collision occurs because no two masters can assert SDA at exactly the same time.

If SCL goes from high-to-low before the BRG times out and SDA has not already been asserted, a bus collision occurs. In this case, another master is attempting to transmit a data '1' during the Repeated Start condition, see Figure 20-37.

If, at the end of the BRG time-out, both SCL and SDA are still high, the SDA pin is driven low and the BRG is reloaded and begins counting. At the end of the count, regardless of the status of the SCL pin, the SCL pin is driven low and the Repeated Start condition is complete.

FIGURE 20-36: BUS COLLISION DURING A REPEATED START CONDITION (CASE 1)

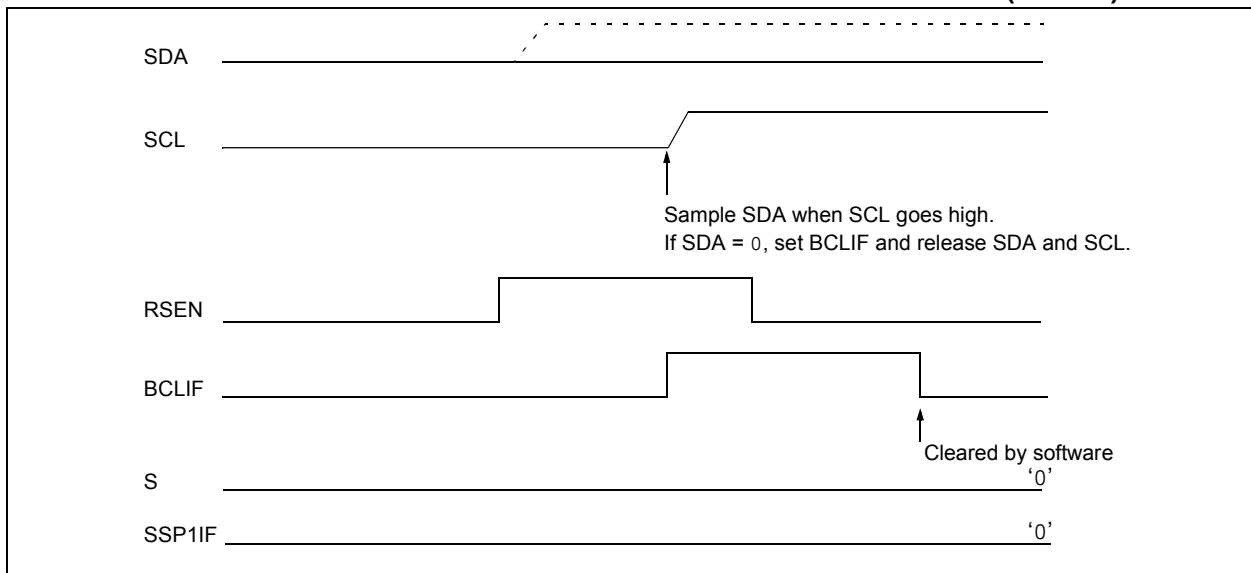
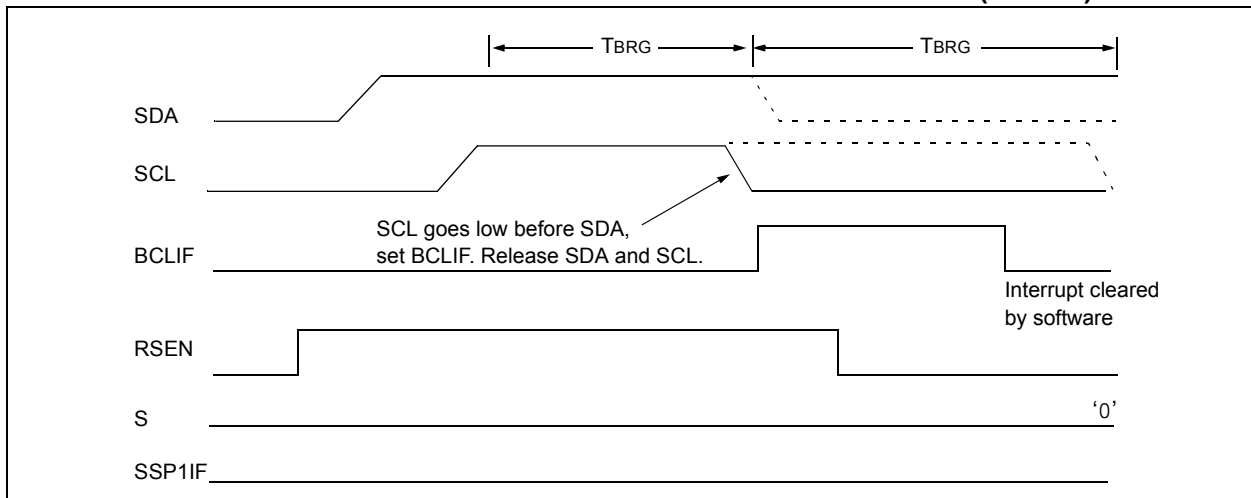


FIGURE 20-37: BUS COLLISION DURING REPEATED START CONDITION (CASE 2)



REGISTER 20-3: SSPCON2: SSP CONTROL REGISTER 2

R/W-0/0	R-0/0	R/W-0/0	R/S/HS-0/0	R/S/HS-0/0	R/S/HS-0/0	R/S/HS-0/0	R/W/HS-0/0
GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
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	HC = Cleared by hardware S = User set

- bit 7 **GCEN:** General Call Enable bit (in I²C Slave mode only)
1 = Enable interrupt when a general call address (0x00 or 00h) is received in the SSPSR
0 = General call address disabled
- bit 6 **ACKSTAT:** Acknowledge Status bit (in I²C mode only)
1 = Acknowledge was not received
0 = Acknowledge was received
- bit 5 **ACKDT:** Acknowledge Data bit (in I²C mode only)
In Receive mode:
Value transmitted when the user initiates an Acknowledge sequence at the end of a receive
1 = Not Acknowledge
0 = Acknowledge
- bit 4 **ACKEN:** Acknowledge Sequence Enable bit (in I²C Master mode only)⁽¹⁾
In Master Receive mode:
1 = Initiate Acknowledge sequence on SDA and SCL pins, and transmit ACKDT data bit.
Automatically cleared by hardware.
0 = Acknowledge sequence idle
- bit 3 **RCEN:** Receive Enable bit (in I²C Master mode only)⁽¹⁾
1 = Enables Receive mode for I²C
0 = Receive idle
- bit 2 **PEN:** Stop Condition Enable bit (in I²C Master mode only)⁽¹⁾
SCK Release Control:
1 = Initiate Stop condition on SDA and SCL pins. Automatically cleared by hardware.
0 = Stop condition Idle
- bit 1 **RSEN:** Repeated Start Condition Enable bit (in I²C Master mode only)⁽¹⁾
1 = Initiate Repeated Start condition on SDA and SCL pins. Automatically cleared by hardware.
0 = Repeated Start condition Idle
- bit 0 **SEN:** Start Condition Enable/Stretch Enable bit⁽¹⁾
In Master mode:
1 = Initiate Start condition on SDA and SCL pins. Automatically cleared by hardware.
0 = Start condition Idle
In Slave mode:
1 = Clock stretching is enabled for both slave transmit and slave receive (stretch enabled)
0 = Clock stretching is disabled

Note 1: If the I²C module is not in the Idle mode, this bit may not be set (no spooling) and the SSPBUF may not be written (or writes to the SSPBUF are disabled).

21.1.2.8 Asynchronous Reception Set-up:

1. Initialize the SPBRGH, SPBRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section 21.4 “EUSART Baud Rate Generator (BRG)”**).
2. Clear the ANSELx bit for the RX pin (if applicable).
3. Enable the serial port by setting the SPEN bit. The SYNC bit must be clear for asynchronous operation.
4. If interrupts are desired, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
5. If 9-bit reception is desired, set the RX9 bit.
6. Enable reception by setting the CREN bit.
7. The RCIF interrupt flag bit will be set when a character is transferred from the RSR to the receive buffer. An interrupt will be generated if the RCIE interrupt enable bit was also set.
8. Read the RCSTA register to get the error flags and, if 9-bit data reception is enabled, the ninth data bit.
9. Get the received eight Least Significant data bits from the receive buffer by reading the RCREG register.
10. If an overrun occurred, clear the OERR flag by clearing the CREN receiver enable bit.

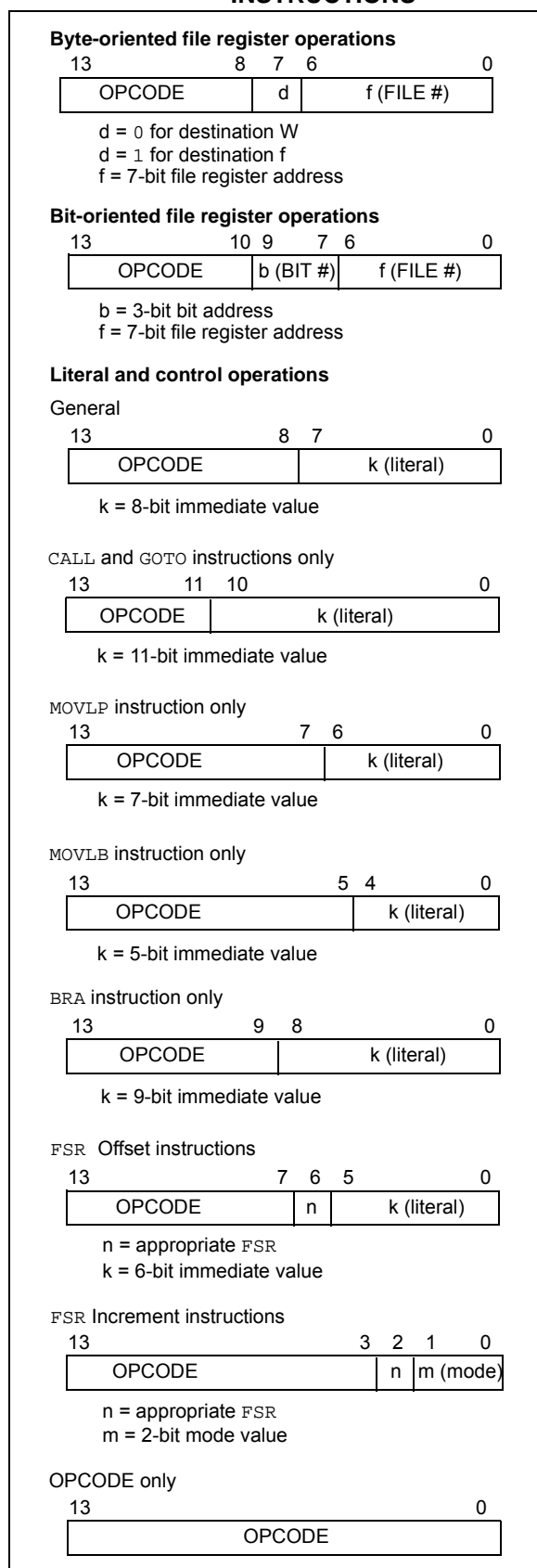
21.1.2.9 9-Bit Address Detection Mode Set-up

This mode would typically be used in RS-485 systems. To set up an asynchronous reception with address detect enable:

1. Initialize the SPBRGH, SPBRGL register pair and the BRGH and BRG16 bits to achieve the desired baud rate (see **Section 21.4 “EUSART Baud Rate Generator (BRG)”**).
2. Clear the ANSELx bit for the RX pin (if applicable).
3. Enable the serial port by setting the SPEN bit. The SYNC bit must be clear for asynchronous operation.
4. If interrupts are desired, set the RCIE bit of the PIE1 register and the GIE and PEIE bits of the INTCON register.
5. Enable 9-bit reception by setting the RX9 bit.
6. Enable address detection by setting the ADDEN bit.
7. Enable reception by setting the CREN bit.
8. The RCIF interrupt flag bit will be set when a character with the ninth bit set is transferred from the RSR to the receive buffer. An interrupt will be generated if the RCIE interrupt enable bit was also set.
9. Read the RCSTA register to get the error flags. The ninth data bit will always be set.
10. Get the received eight Least Significant data bits from the receive buffer by reading the RCREG register. Software determines if this is the device's address.
11. If an overrun occurred, clear the OERR flag by clearing the CREN receiver enable bit.
12. If the device has been addressed, clear the ADDEN bit to allow all received data into the receive buffer and generate interrupts.

PIC16LF1554/1559

FIGURE 24-1: GENERAL FORMAT FOR INSTRUCTIONS



PIC16LF1554/1559

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

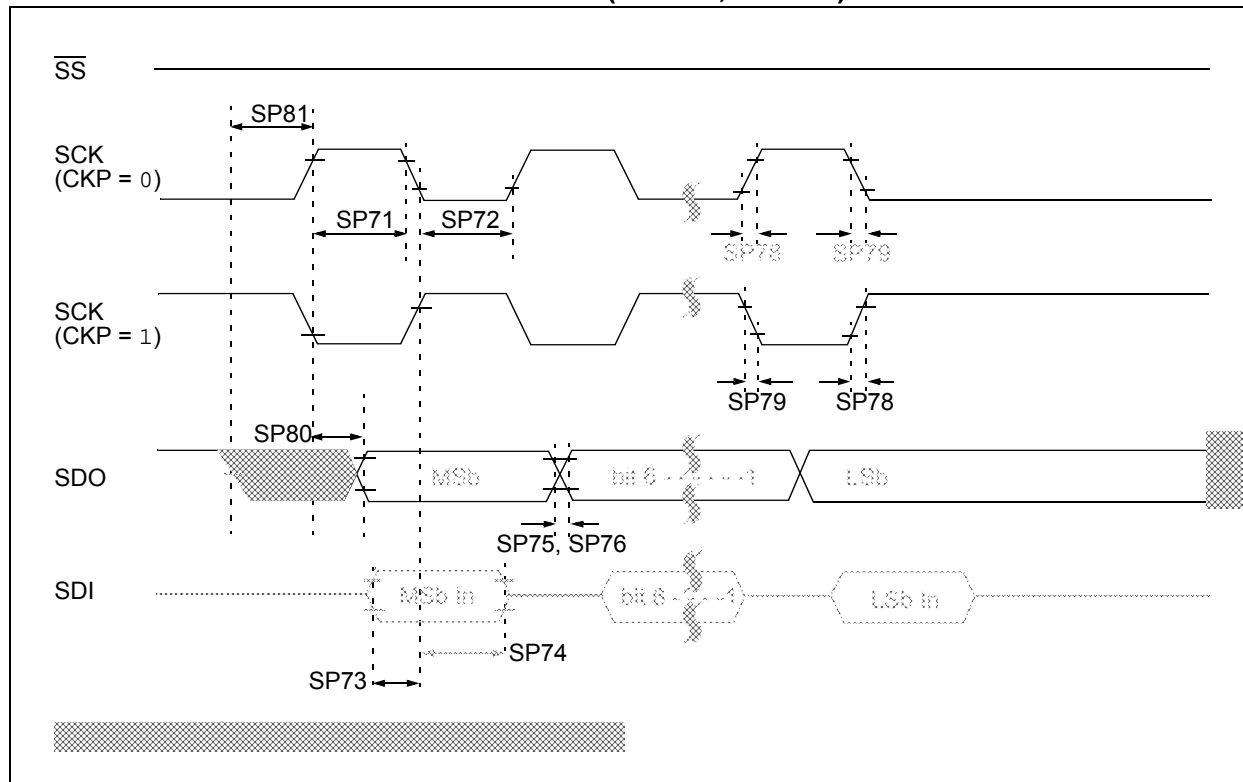


FIGURE 25-14: SPI MASTER MODE TIMING (CKE = 1, SMP = 1)

