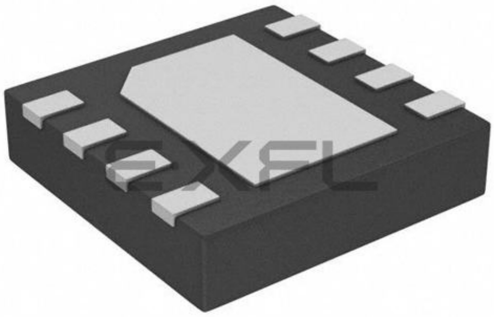


Welcome to E-XFL.COM

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	6
Program Memory Size	3.5KB (2K x 14)
Program Memory Type	FLASH
EEPROM Size	224 x 8
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 5x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	8-VDFN Exposed Pad
Supplier Device Package	8-DFN (3x3)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf15313-i-rf

2.0 GUIDELINES FOR GETTING STARTED WITH PIC16(L)F15313/23 MICROCONTROLLERS

2.1 Basic Connection Requirements

Getting started with the PIC16(L)F15313/23 family of 8-bit microcontrollers requires attention to a minimal set of device pin connections before proceeding with development.

The following pins must always be connected:

- All VDD and VSS pins (see **Section 2.2 "Power Supply Pins"**)
- $\overline{\text{MCLR}}$ pin (see **Section 2.3 "Master Clear (MCLR) Pin"**)

These pins must also be connected if they are being used in the end application:

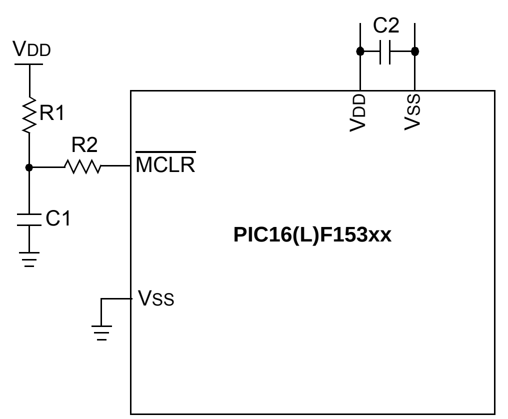
- ICSPCLK/ICSPDAT pins used for In-Circuit Serial Programming™ (ICSP™) and debugging purposes (see **Section 2.4 "ICSP™ Pins"**)
- OSCI and OSCO pins when an external oscillator source is used (see **Section 2.5 "External Oscillator Pins"**)

Additionally, the following pins may be required:

- VREF+/VREF- pins are used when external voltage reference for analog modules is implemented

The minimum mandatory connections are shown in Figure 2-1.

FIGURE 2-1: RECOMMENDED MINIMUM CONNECTIONS



Key (all values are recommendations):

C1: 10nF, 16V ceramic

C2: 0.1uF, 16V ceramic

R1: 10 kΩ

R2: 100Ω to 470Ω

2.2 Power Supply Pins

2.2.1 DECOUPLING CAPACITORS

The use of decoupling capacitors on every pair of power supply pins (VDD and VSS) is required.

Consider the following criteria when using decoupling capacitors:

- **Value and type of capacitor:** A 0.1 μF (100 nF), 10-25V capacitor is recommended. The capacitor should be a low-ESR device, with a resonance frequency in the range of 200 MHz and higher. Ceramic capacitors are recommended.
- **Placement on the printed circuit board:** The decoupling capacitors should be placed as close to the pins as possible. It is recommended to place the capacitors on the same side of the board as the device. If space is constricted, the capacitor can be placed on another layer on the PCB using a via; however, ensure that the trace length from the pin to the capacitor is no greater than 0.25 inch (6 mm).
- **Handling high-frequency noise:** If the board is experiencing high-frequency noise (upward of tens of MHz), add a second ceramic type capacitor in parallel to the above described decoupling capacitor. The value of the second capacitor can be in the range of 0.01 μF to 0.001 μF. Place this second capacitor next to each primary decoupling capacitor. In high-speed circuit designs, consider implementing a decade pair of capacitances as close to the power and ground pins as possible (e.g., 0.1 μF in parallel with 0.001 μF).
- **Maximizing performance:** On the board layout from the power supply circuit, run the power and return traces to the decoupling capacitors first, and then to the device pins. This ensures that the decoupling capacitors are first in the power chain. Equally important is to keep the trace length between the capacitor and the power pins to a minimum, thereby reducing PCB trace inductance.

2.2.2 TANK CAPACITORS

On boards with power traces running longer than six inches in length, it is suggested to use a tank capacitor for integrated circuits, including microcontrollers, to supply a local power source. The value of the tank capacitor should be determined based on the trace resistance that connects the power supply source to the device, and the maximum current drawn by the device in the application. In other words, select the tank capacitor so that it meets the acceptable voltage sag at the device. Typical values range from 4.7 μF to 47 μF.

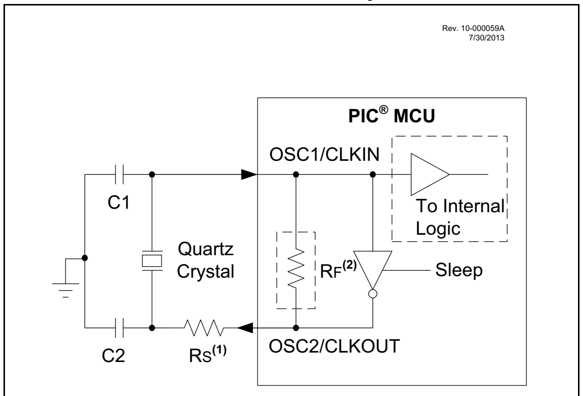
TABLE 4-10: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-63 (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on: MCLR
Bank 60 (Continued)											
1E2Bh	CLC3GLS1	LC3G2D4T	LC3G4D3N	LC3G2D3T	LC3G2D3N	LC3G2D2T	LC3G2D2N	LC3G2D1T	LC3G2D1N	xxxx xxxx	uuuu uuuu
1E2Ch	CLC3GLS2	LC3G3D4T	LC3G4D3N	LC3G3D3T	LC3G3D3N	LC3G3D2T	LC3G3D2N	LC3G3D1T	LC3G3D1N	xxxx xxxx	uuuu uuuu
1E2Dh	CLC3GLS3	LC3G4D4T	LC3G4D3N	LC3G4D3T	LC3G4D3N	LC3G4D2T	LC3G4D2N	LC3G4D1T	LC3G4D1N	xxxx xxxx	uuuu uuuu
1E2Eh	CLC4CON	LC4EN	—	LC4OUT	LC4INTP	LC4INTN	LC4MODE<2:0>			0-00 0000	0-00 0000
1E2Fh	CLC4POL	LC4POL	—	—	—	LC4G4POL	LC4G3POL	LC4G2POL	LC4G1POL	0--- xxxx	0--- uuuu
1E30h	CLC4SEL0	—	—	LC4D1S<5:0>						--xx xxxx	--uu uuuu
1E31h	CLC4SEL1	—	—	LC4D2S<5:0>						--xx xxxx	--uu uuuu
1E32h	CLC4SEL2	—	—	LC4D3S<5:0>						--xx xxxx	--uu uuuu
1E33h	CLC4SEL3	—	—	LC4D4S<5:0>						--xx xxxx	--uu uuuu
1E34h	CLC4GLS0	LC4G1D4T	LC4G4D3N	LC4G1D3T	LC4G1D3N	LC4G1D2T	LC4G1D2N	LC4G1D1T	LC4G1D1N	xxxx xxxx	uuuu uuuu
1E35h	CLC4GLS1	LC4G2D4T	LC4G4D3N	LC4G2D3T	LC4G2D3N	LC4G2D2T	LC4G2D2N	LC4G2D1T	LC4G2D1N	xxxx xxxx	uuuu uuuu
1E36h	CLC4GLS2	LC4G3D4T	LC4G4D3N	LC4G3D3T	LC4G3D3N	LC4G3D2T	LC4G3D2N	LC4G3D1T	LC4G3D1N	xxxx xxxx	uuuu uuuu
1E37h	CLC4GLS3	LC4G4D4T	LC4G4D3N	LC4G4D3T	LC4G4D3N	LC4G4D2T	LC4G4D2N	LC4G4D1T	LC4G4D1N	xxxx xxxx	uuuu uuuu
1E38h — 1E6Fh	—	Unimplemented								—	—

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

Preliminary

FIGURE 9-3: QUARTZ CRYSTAL OPERATION (LP, XT OR HS MODE)



Note 1: A series resistor (Rs) may be required for quartz crystals with low drive level.

2: The value of RF varies with the Oscillator mode selected (typically between 2 MΩ and 10 MΩ).

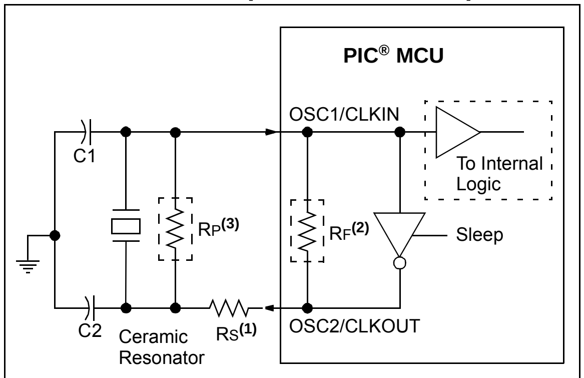
Note 1: Quartz crystal characteristics vary according to type, package and manufacturer. The user should consult the manufacturer data sheets for specifications and recommended application.

2: Always verify oscillator performance over the VDD and temperature range that is expected for the application.

3: For oscillator design assistance, reference the following Microchip Application Notes:

- AN826, "*Crystal Oscillator Basics and Crystal Selection for rfPIC® and PIC® Devices*" (DS00826)
- AN849, "*Basic PIC® Oscillator Design*" (DS00849)
- AN943, "*Practical PIC® Oscillator Analysis and Design*" (DS00943)
- AN949, "*Making Your Oscillator Work*" (DS00949)

FIGURE 9-4: CERAMIC RESONATOR OPERATION (XT OR HS MODE)



Note 1: A series resistor (RS) may be required for ceramic resonators with low drive level.

2: The value of RF varies with the Oscillator mode selected (typically between 2 MΩ to 10 MΩ).

3: An additional parallel feedback resistor (RP) may be required for proper ceramic resonator operation.

9.2.1.3 Oscillator Start-up Timer (OST)

If the oscillator module is configured for LP, XT or HS modes, the Oscillator Start-up Timer (OST) counts 1024 oscillations from OSC1. This occurs following a Power-on Reset (POR), Brown-out Reset (BOR) or a wake-up from Sleep. The OST ensures that the oscillator circuit, using a quartz crystal resonator or ceramic resonator, has started and is providing a stable system clock to the oscillator module.

13.3.4 NVMREG WRITE TO PROGRAM MEMORY

Program memory is programmed using the following steps:

1. Load the address of the row to be programmed into NVMADRH:NVMADRL.
2. Load each write latch with data.
3. Initiate a programming operation.
4. Repeat steps 1 through 3 until all data is written.

Before writing to program memory, the word(s) to be written must be erased or previously unwritten. Program memory can only be erased one row at a time. No automatic erase occurs upon the initiation of the write.

Program memory can be written one or more words at a time. The maximum number of words written at one time is equal to the number of write latches. See Figure 13-4 (row writes to program memory with 32 write latches) for more details.

The write latches are aligned to the Flash row address boundary defined by the upper ten bits of NVMADRH:NVMADRL, (NVMADRH<6:0>:NVMADRL<7:5>) with the lower five bits of NVMADRL, (NVMADRL<4:0>) determining the write latch being loaded. Write operations do not cross these boundaries. At the completion of a program memory write operation, the data in the write latches is reset to contain 0x3FFF.

The following steps should be completed to load the write latches and program a row of program memory. These steps are divided into two parts. First, each write latch is loaded with data from the NVMDATH:NVMDATL using the unlock sequence with LWLO = 1. When the last word to be loaded into the write latch is ready, the LWLO bit is cleared and the unlock sequence executed. This initiates the programming operation, writing all the latches into Flash program memory.

Note:	The special unlock sequence is required to load a write latch with data or initiate a Flash programming operation. If the unlock sequence is interrupted, writing to the latches or program memory will not be initiated.

1. Set the WREN bit of the NVMCON1 register.
2. Clear the NVMREGS bit of the NVMCON1 register.
3. Set the LWLO bit of the NVMCON1 register. When the LWLO bit of the NVMCON1 register is '1', the write sequence will only load the write latches and will not initiate the write to Flash program memory.
4. Load the NVMADRH:NVMADRL register pair with the address of the location to be written.
5. Load the NVMDATH:NVMDATL register pair with the program memory data to be written.
6. Execute the unlock sequence (**Section 13.3.2 "NVM Unlock Sequence"**). The write latch is now loaded.
7. Increment the NVMADRH:NVMADRL register pair to point to the next location.
8. Repeat steps 5 through 7 until all but the last write latch has been loaded.
9. Clear the LWLO bit of the NVMCON1 register. When the LWLO bit of the NVMCON1 register is '0', the write sequence will initiate the write to Flash program memory.
10. Load the NVMDATH:NVMDATL register pair with the program memory data to be written.
11. Execute the unlock sequence (**Section 13.3.2 "NVM Unlock Sequence"**). The entire program memory latch content is now written to Flash program memory.

Note:	The program memory write latches are reset to the blank state (0x3FFF) at the completion of every write or erase operation. As a result, it is not necessary to load all the program memory write latches. Unloaded latches will remain in the blank state.

An example of the complete write sequence is shown in Example 13-4. The initial address is loaded into the NVMADRH:NVMADRL register pair; the data is loaded using indirect addressing.

EXAMPLE 13-4: WRITING TO PROGRAM FLASH MEMORY

```
; This write routine assumes the following:
; 1. 64 bytes of data are loaded, starting at the address in DATA_ADDR
; 2. Each word of data to be written is made up of two adjacent bytes in DATA_ADDR,
;    stored in little endian format
; 3. A valid starting address (the least significant bits = 00000) is loaded in ADDRH:ADDRL
; 4. ADDRH and ADDRL are located in common RAM (locations 0x70 - 0x7F)
; 5. NVM interrupts are not taken into account

        BANKSEL       NVMADRH
        MOVF          ADDRH,W
        MOVWF         NVMADRH                ; Load initial address
        MOVF          ADDRL,W
        MOVWF         NVMADRL
        MOVLW         LOW DATA_ADDR          ; Load initial data address
        MOVWF         FSR0L
        MOVLW         HIGH DATA_ADDR
        MOVWF         FSR0H
        BCF           NVMCON1,NVMREGS        ; Set Program Flash Memory as write location
        BSF           NVMCON1,WREN           ; Enable writes
        BSF           NVMCON1,LWLO           ; Load only write latches

LOOP
        MOVIW         FSR0++
        MOVWF         NVMDATL                ; Load first data byte
        MOVIW         FSR0++
        MOVWF         NVMDATH                ; Load second data byte

        MOVF          NVMADRL,W
        XORLW         0x1F                   ; Check if lower bits of address are 00000
        ANDLW         0x1F                   ; and if on last of 32 addresses
        BTFSC         STATUS,Z               ; Last of 32 words?
        GOTO          START_WRITE            ; If so, go write latches into memory

        CALL          UNLOCK_SEQ             ; If not, go load latch
        INCF          NVMADRL,F              ; Increment address
        GOTO          LOOP

START_WRITE
        BCF           NVMCON1,LWLO           ; Latch writes complete, now write memory
        CALL          UNLOCK_SEQ             ; Perform required unlock sequence
        BCF           NVMCON1,WREN           ; Disable writes

UNLOCK_SEQ
        MOVLW         55h
        BCF           INTCON,GIE             ; Disable interrupts
        MOVWF         NVMCON2                ; Begin unlock sequence
        MOVLW         AAh
        MOVWF         NVMCON2
        BSF           NVMCON1,WR
        BSF           INTCON,GIE             ; Unlock sequence complete, re-enable interrupts
        return
```

26.6 Timer1 Interrupts

The timer register pair (TMR1H:TMR1L) increments to FFFFh and rolls over to 0000h. When the timer rolls over, the respective timer interrupt flag bit of the PIR5 register is set. To enable the interrupt on rollover, you must set these bits:

- ON bit of the T1CON register
- TMR1IE bit of the PIE4 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:	To avoid immediate interrupt vectoring, the TMR1H:TMR1L register pair should be preloaded with a value that is not imminently about to rollover, and the TMR1IF flag should be cleared prior to enabling the timer interrupts.

26.7 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:

- ON bit of the T1CON register must be set
- TMR1IE bit of the PIE4 register must be set
- PEIE bit of the INTCON register must be set
- $\overline{\text{SYNC}}$ bit of the T1CON register must be set
- CS bits of the T1CLK register must be configured
- The timer clock source must be enabled and continue operation during sleep.

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.

26.8 CCP Capture/Compare Time Base

The CCP modules use the TMR1H:TMR1L register pair as the time base when operating in Capture or Compare mode.

In Capture mode, the value in the TMR1H:TMR1L register pair is copied into the CCPRxH:CCPRxL register pair on a configured event.

In Compare mode, an event is triggered when the value CCPRxH:CCPRxL register pair matches the value in the TMR1H:TMR1L register pair. This event can be an Auto-conversion Trigger.

For more information, see **Section 28.0 "Capture/Compare/PWM Modules"**.

26.9 CCP Auto-Conversion Trigger

When any of the CCP's are configured to trigger an auto-conversion, the trigger will clear the TMR1H:TMR1L register pair. This auto-conversion does not cause a timer interrupt. The CCP module may still be configured to generate a CCP interrupt.

In this mode of operation, the CCPRxH:CCPRxL register pair becomes the period register for Timer1.

The timer should be synchronized and FOSC/4 should be selected as the clock source in order to utilize the Auto-conversion Trigger. Asynchronous operation of the timer can cause an Auto-conversion Trigger to be missed.

In the event that a write to TMR1H or TMR1L coincides with an Auto-conversion Trigger from the CCP, the write will take precedence.

For more information, see **Section 28.2.4 "Compare During Sleep"**.

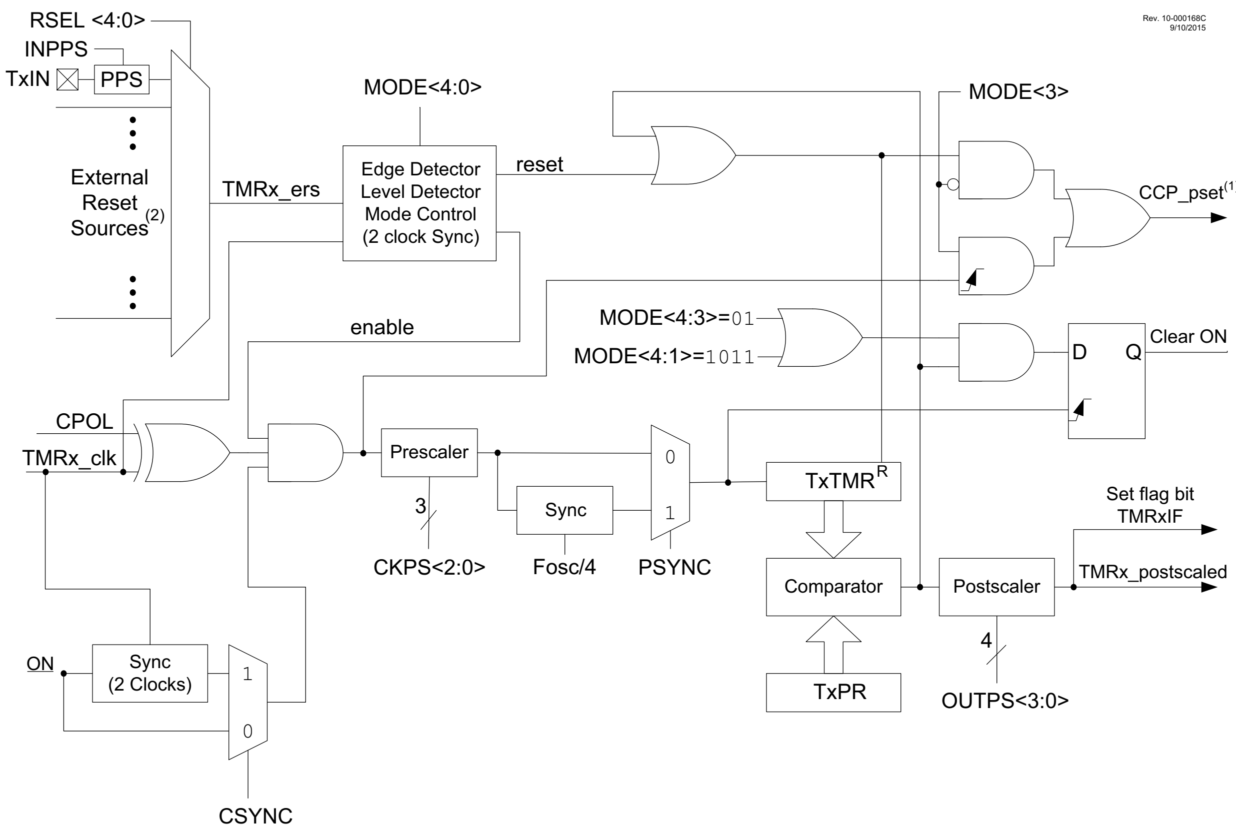
27.0 TIMER2 MODULE WITH HARDWARE LIMIT TIMER (HLT)

The Timer2 module is an 8-bit timer that can operate as free-running period counters or in conjunction with external signals that control start, run, freeze, and reset operation in One-Shot and Monostable modes of operation. Sophisticated waveform control such as pulse density modulation are possible by combining the operation of this timer with other internal peripherals such as the comparators and CCP modules. Features of the timer include:

- 8-bit timer register
- 8-bit period register
- Selectable external hardware timer Resets
- Programmable prescaler (1:1 to 1:128)
- Programmable postscaler (1:1 to 1:16)
- Selectable synchronous/asynchronous operation
- Alternate clock sources
- Interrupt-on-period
- Three modes of operation:
  - Free Running Period
  - One-shot
  - Monostable

See Figure 27-1 for a block diagram of Timer2. See Figure 27-2 for the clock source block diagram.

FIGURE 27-1: TIMER2 BLOCK DIAGRAM



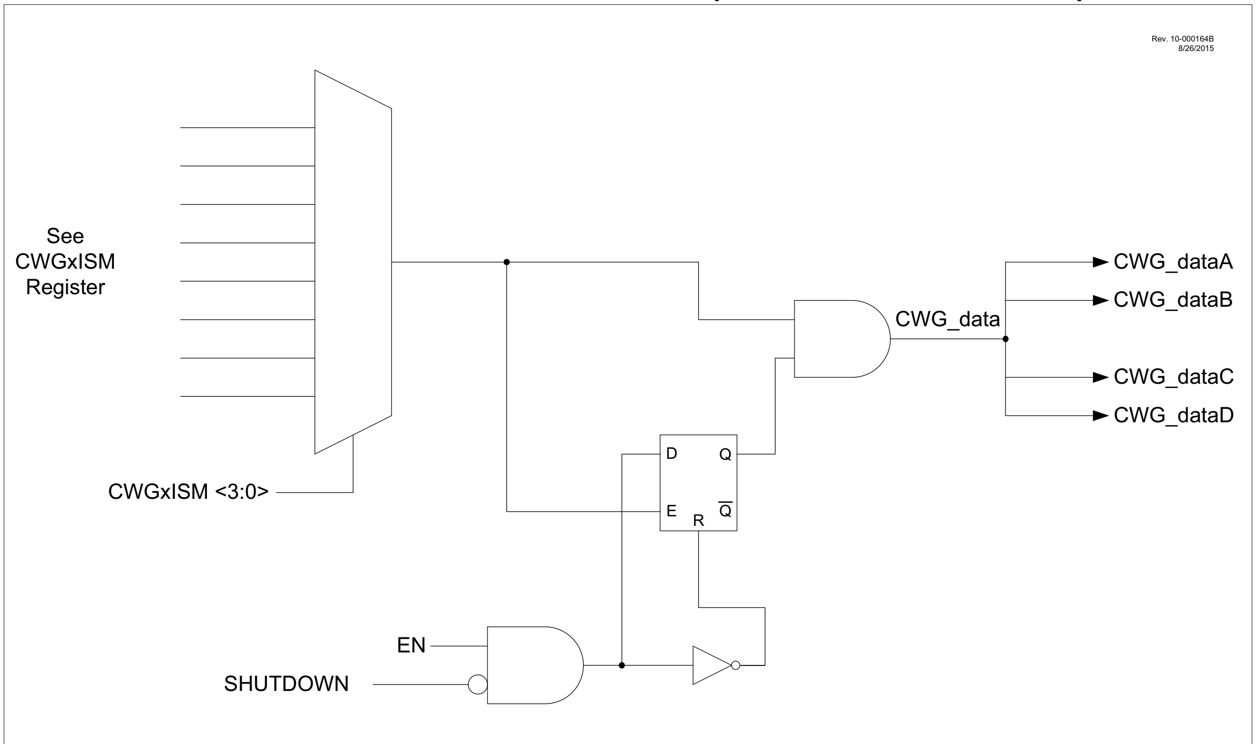
Note 1: Signal to the CCP to trigger the PWM pulse.
2: See Register 27-4 for external Reset sources.

30.1.4 STEERING MODES

In Steering modes, the data input can be steered to any or all of the four CWG output pins. In Synchronous Steering mode, changes to steering selection registers take effect on the next rising input.

In Non-Synchronous mode, steering takes effect on the next instruction cycle. Additional details are provided in **Section 30.9 "CWG Steering Mode"**.

FIGURE 30-4: SIMPLIFIED CWG BLOCK DIAGRAM (OUTPUT STEERING MODES)



30.2 Clock Source

The CWG module allows the following clock sources to be selected:

- Fosc (system clock)
- HFINTOSC (16 MHz only)

The clock sources are selected using the CS bit of the CWG1CLKCON register.

REGISTER 30-2: CWG1CON1: CWG1 CONTROL REGISTER 1

U-0	U-0	R-x	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	IN	—	POLD	POLC	POLB	POLA
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	q = Value depends on condition

bit 7-6 **Unimplemented:** Read as '0'

bit 5 **IN:** CWG Input Value bit

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

bit 3 **POLD:** CWG1D Output Polarity bit
1 = Signal output is inverted polarity
0 = Signal output is normal polarity

bit 2 **POLC:** CWG1C Output Polarity bit
1 = Signal output is inverted polarity
0 = Signal output is normal polarity

bit 1 **POLB:** CWG1B Output Polarity bit
1 = Signal output is inverted polarity
0 = Signal output is normal polarity

bit 0 **POLA:** CWG1A Output Polarity bit
1 = Signal output is inverted polarity
0 = Signal output is normal polarity

TABLE 30-3: SUMMARY OF REGISTERS ASSOCIATED WITH CWG

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CWG1CLKCON	—	—	—	—	—	—	—	CS	341
CWG1ISM	—	—	—	—	IS<3:0>				341
CWG1DBR	—	—	DBR<5:0>						337
CWG1DBF	—	—	DBF<5:0>						337
CWG1CON0	EN	LD	—	—	—	MODE<2:0>			340
CWG1CON1	—	—	IN	—	POLD	POLC	POLB	POLA	336
CWG1AS0	SHUTDOWN	REN	LSBD<1:0>		LSAC<1:0>		—	—	338
CWG1AS1	—	—	—	AS4E	AS3E	AS2E	AS1E	AS0E	339
CWG1STR	OVRD	OVRC	OVRB	OVRA	STRD	STRC	STRB	STRA	340

Legend: – = unimplemented locations read as '0'. Shaded cells are not used by CWG.

FIGURE 31-2: INPUT DATA SELECTION AND GATING

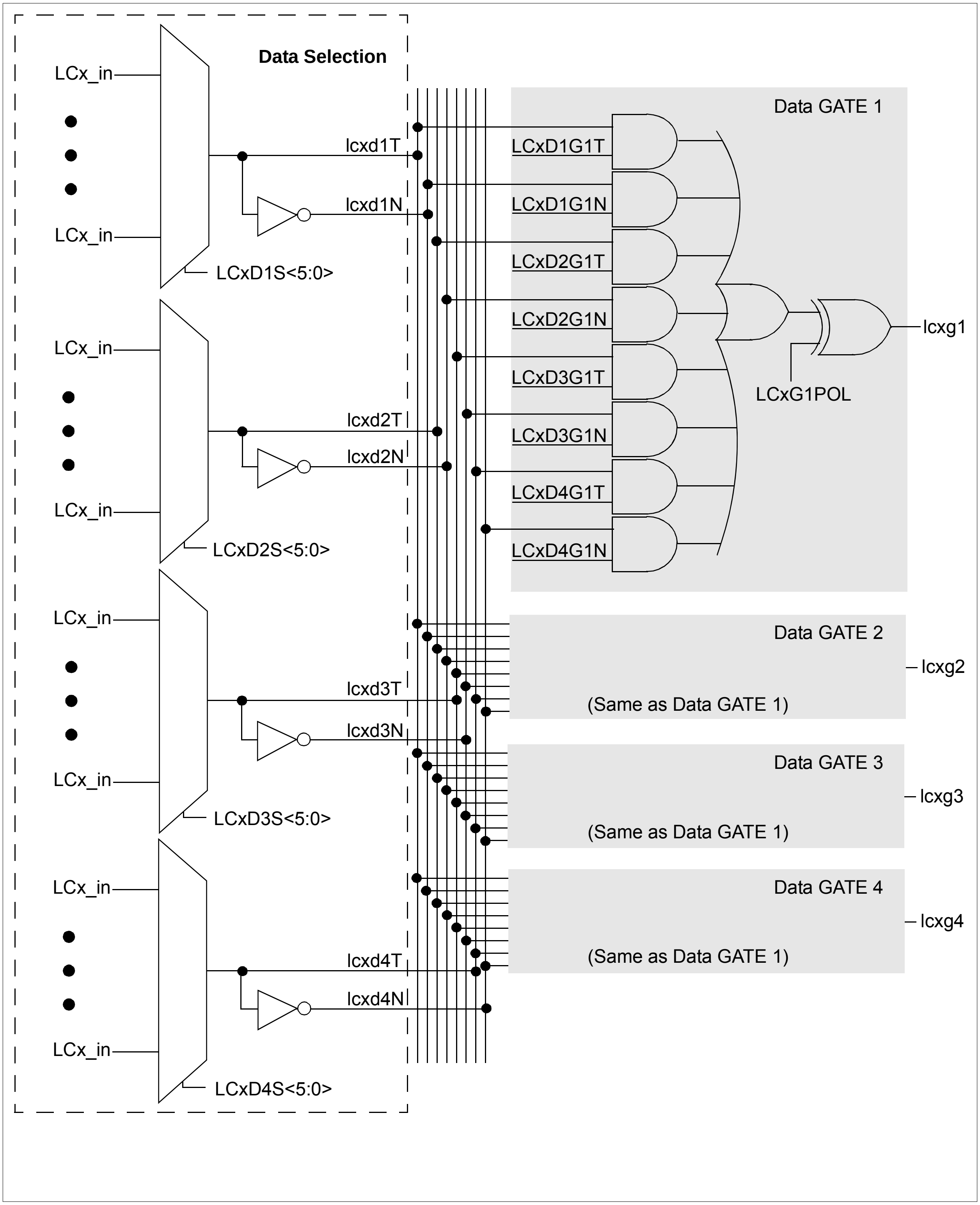
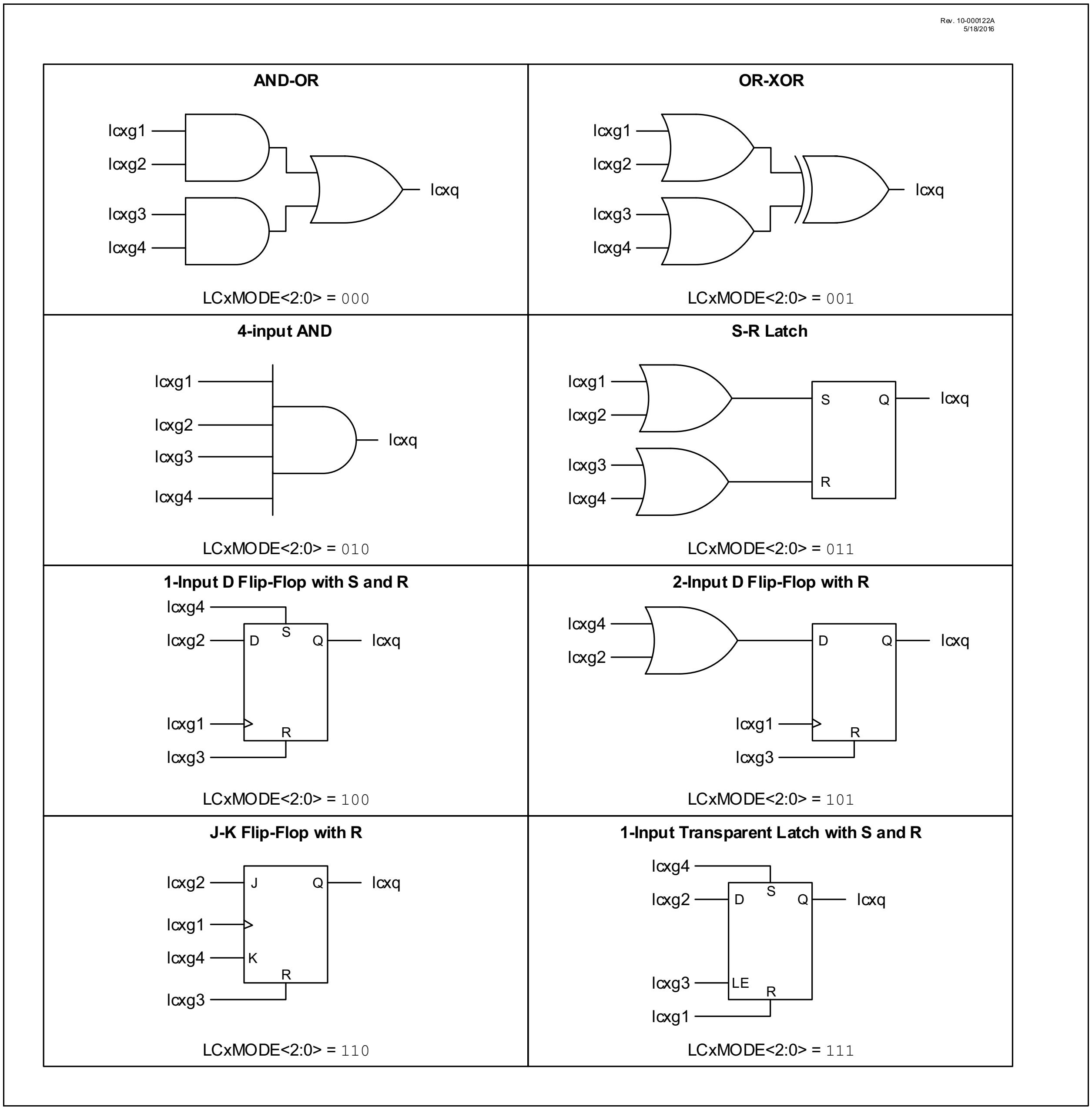


FIGURE 31-3: PROGRAMMABLE LOGIC FUNCTIONS



REGISTER 31-9: CLCxGLS2: GATE 2 LOGIC SELECT REGISTER

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
LCxG3D4T	LCxG3D4N	LCxG3D3T	LCxG3D3N	LCxG3D2T	LCxG3D2N	LCxG3D1T	LCxG3D1N
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 **LCxG3D4T:** Gate 2 Data 4 True (non-inverted) bit
1 = CLCIN3 (true) is gated into CLCx Gate 2
0 = CLCIN3 (true) is not gated into CLCx Gate 2

bit 6 **LCxG3D4N:** Gate 2 Data 4 Negated (inverted) bit
1 = CLCIN3 (inverted) is gated into CLCx Gate 2
0 = CLCIN3 (inverted) is not gated into CLCx Gate 2

bit 5 **LCxG3D3T:** Gate 2 Data 3 True (non-inverted) bit
1 = CLCIN2 (true) is gated into CLCx Gate 2
0 = CLCIN2 (true) is not gated into CLCx Gate 2

bit 4 **LCxG3D3N:** Gate 2 Data 3 Negated (inverted) bit
1 = CLCIN2 (inverted) is gated into CLCx Gate 2
0 = CLCIN2 (inverted) is not gated into CLCx Gate 2

bit 3 **LCxG3D2T:** Gate 2 Data 2 True (non-inverted) bit
1 = CLCIN1 (true) is gated into CLCx Gate 2
0 = CLCIN1 (true) is not gated into CLCx Gate 2

bit 2 **LCxG3D2N:** Gate 2 Data 2 Negated (inverted) bit
1 = CLCIN1 (inverted) is gated into CLCx Gate 2
0 = CLCIN1 (inverted) is not gated into CLCx Gate 2

bit 1 **LCxG3D1T:** Gate 2 Data 1 True (non-inverted) bit
1 = CLCIN0 (true) is gated into CLCx Gate 2
0 = CLCIN0 (true) is not gated into CLCx Gate 2

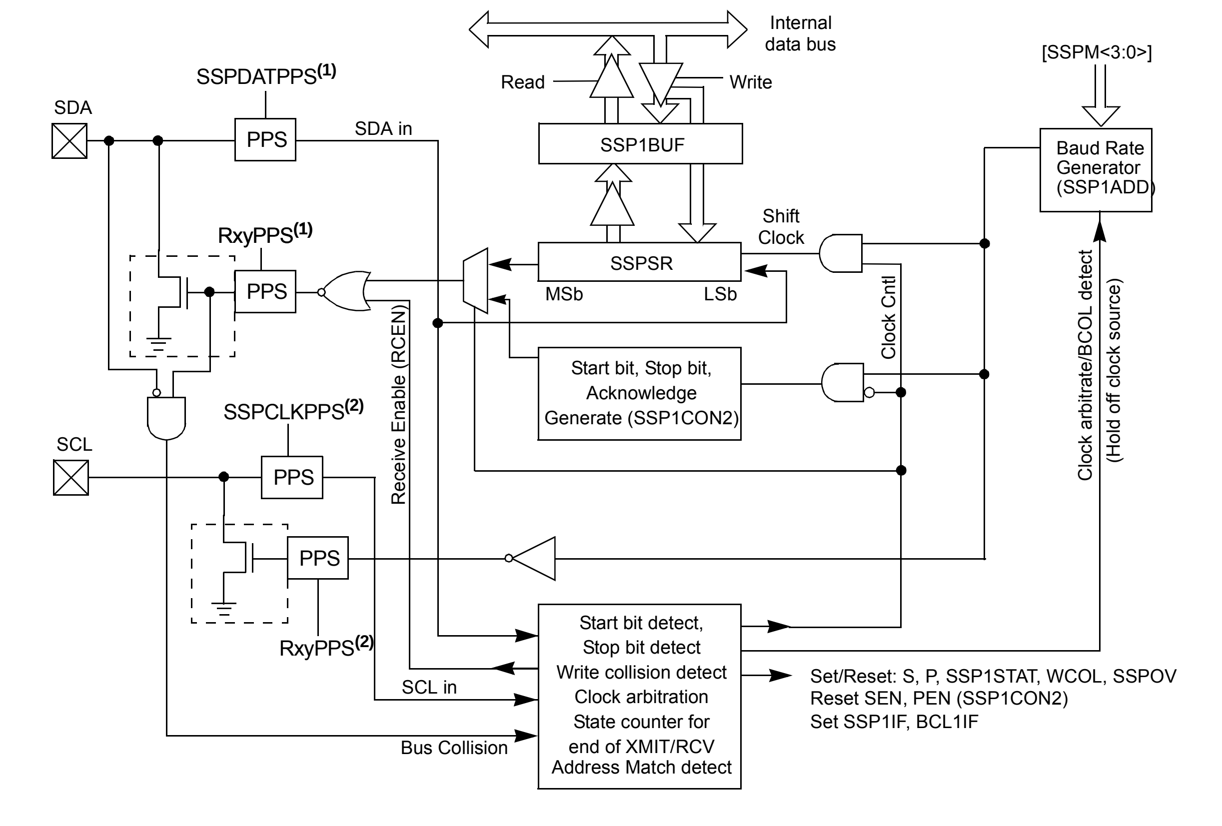
bit 0 **LCxG3D1N:** Gate 2 Data 1 Negated (inverted) bit
1 = CLCIN0 (inverted) is gated into CLCx Gate 2
0 = CLCIN0 (inverted) is not gated into CLCx Gate 2

The I²C interface supports the following modes and features:

- Master mode
- Slave mode
- Byte NACKing (Slave mode)
- Limited multi-master support
- 7-bit and 10-bit addressing
- Start and Stop interrupts
- Interrupt masking
- Clock stretching
- Bus collision detection
- General call address matching
- Address masking
- Selectable SDA hold times

Figure 32-2 is a block diagram of the I²C interface module in Master mode. Figure 32-3 is a diagram of the I²C interface module in Slave mode.

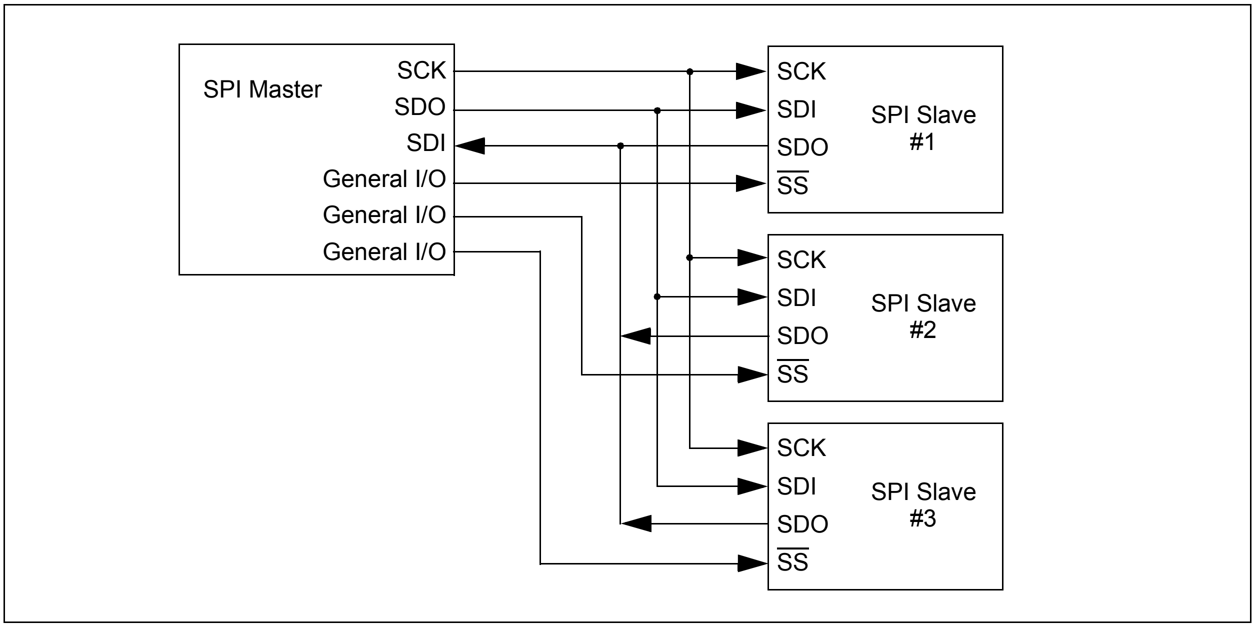
FIGURE 32-2: MSSP BLOCK DIAGRAM (I²C MASTER MODE)



Note 1: SDA pin selections must be the same for input and output.

2: SCL pin selections must be the same for input and output.

FIGURE 32-4: SPI MASTER AND MULTIPLE SLAVE CONNECTION



32.2.1 SPI MODE REGISTERS

The MSSP module has five registers for SPI mode operation. These are:

- MSSP STATUS register (SSP1STAT)
- MSSP Control register 1 (SSP1CON1)
- MSSP Control register 3 (SSP1CON3)
- MSSP Data Buffer register (SSP1BUF)
- MSSP Address register (SSP1ADD)
- MSSP Shift register (SSP1SR)
  (Not directly accessible)

SSP1CON1 and SSP1STAT are the control and status registers in SPI mode operation. The SSP1CON1 register is readable and writable. The lower six bits of the SSP1STAT are read-only. The upper two bits of the SSP1STAT are read/write.

In one SPI master mode, SSP1ADD can be loaded with a value used in the Baud Rate Generator. More information on the Baud Rate Generator is available in **Section 32.7 "Baud Rate Generator"**.

SSP1SR is the shift register used for shifting data in and out. SSP1BUF provides indirect access to the SSP1SR register. SSP1BUF is the buffer register to which data bytes are written, and from which data bytes are read.

In receive operations, SSP1SR and SSP1BUF together create a buffered receiver. When SSP1SR receives a complete byte, it is transferred to SSP1BUF and the SSP1IF interrupt is set.

During transmission, the SSP1BUF is not buffered. A write to SSP1BUF will write to both SSP1BUF and SSP1SR.

32.5.8 GENERAL CALL ADDRESS SUPPORT

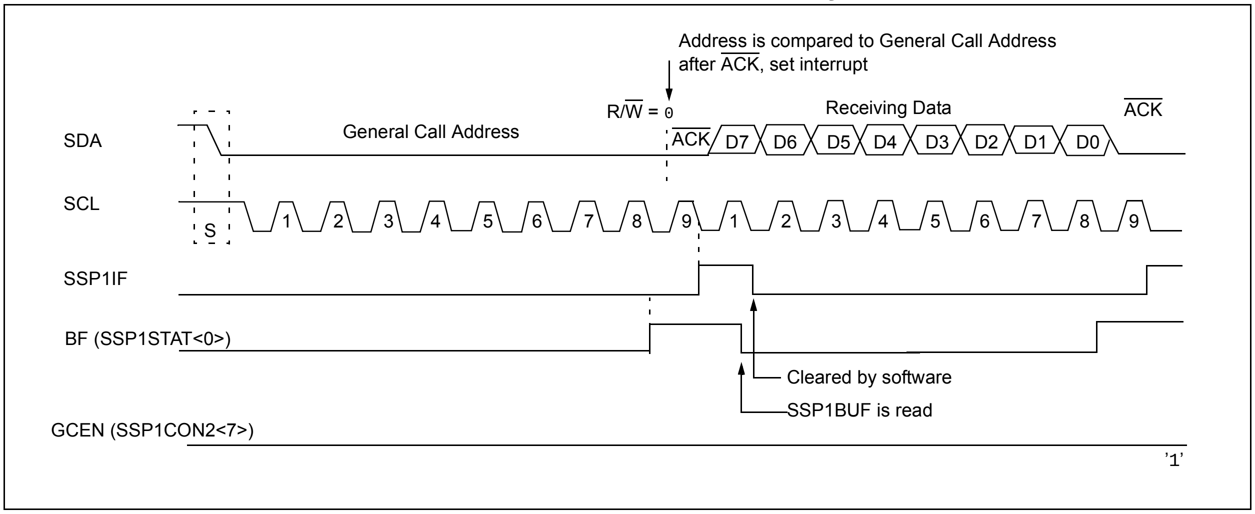
The addressing procedure for the I^{2}C bus is such that the first byte after the Start condition usually determines which device will be the slave addressed by the master device. The exception is the general call address which can address all devices. When this address is used, all devices should, in theory, respond with an acknowledge.

The general call address is a reserved address in the I^{2}C protocol, defined as address `0x00`. When the GCEN bit of the SSP1CON2 register is set, the slave module will automatically $\overline{\text{ACK}}$ the reception of this address regardless of the value stored in SSP1ADD. After the slave clocks in an address of all zeros with the R/$\overline{\text{W}}$ bit clear, an interrupt is generated and slave software can read SSP1BUF and respond. Figure 32-24 shows a general call reception sequence.

In 10-bit Address mode, the UA bit will not be set on the reception of the general call address. The slave will prepare to receive the second byte as data, just as it would in 7-bit mode.

If the AHEN bit of the SSP1CON3 register is set, just as with any other address reception, the slave hardware will stretch the clock after the eighth falling edge of SCL. The slave must then set its ACKDT value and release the clock with communication progressing as it would normally.

FIGURE 32-24: SLAVE MODE GENERAL CALL ADDRESS SEQUENCE



32.5.9 SSP MASK REGISTER

An SSP Mask (SSP1MSK) register (Register 32-5) is available in I^{2}C Slave mode as a mask for the value held in the SSP1SR register during an address comparison operation. A zero ('`0`') bit in the SSP1MSK register has the effect of making the corresponding bit of the received address a "don't care".

This register is reset to all '`1`'s upon any Reset condition and, therefore, has no effect on standard SSP operation until written with a mask value.

The SSP Mask register is active during:

- 7-bit Address mode: address compare of A<7:1>.
- 10-bit Address mode: address compare of A<7:0> only. The SSP mask has no effect during the reception of the first (high) byte of the address.

32.6.13.2 Bus Collision During a Repeated Start Condition

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

a) A low level is sampled on SDA when SCL goes from low level to high level (Case 1).
b) 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 SSP1ADD 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 32-36). 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 32-37.

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 32-36: BUS COLLISION DURING A REPEATED START CONDITION (CASE 1)

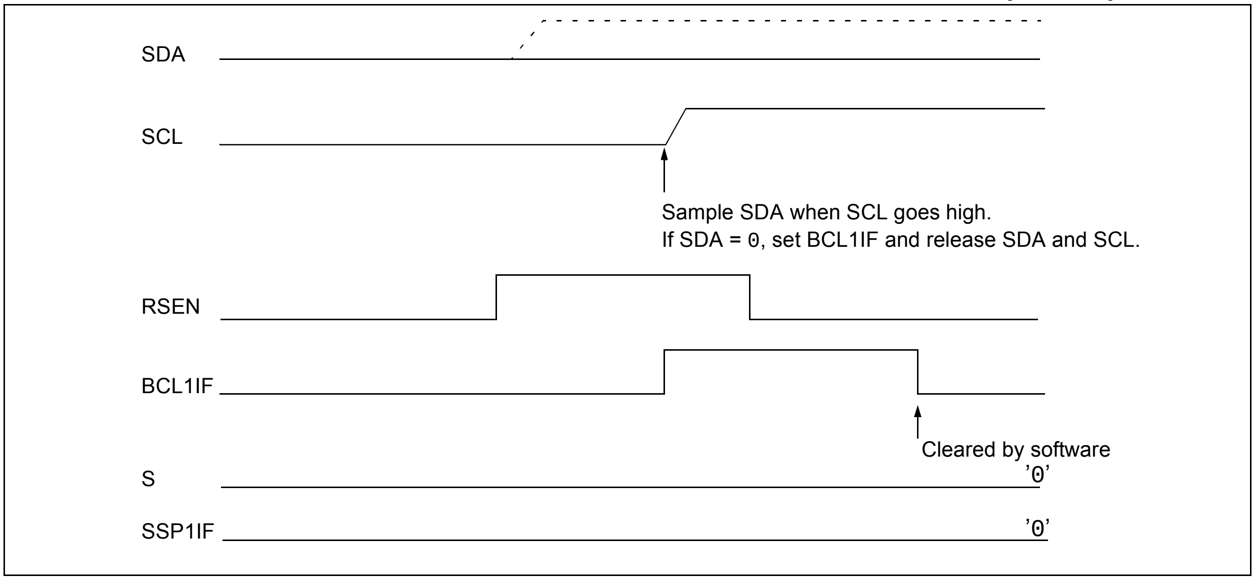
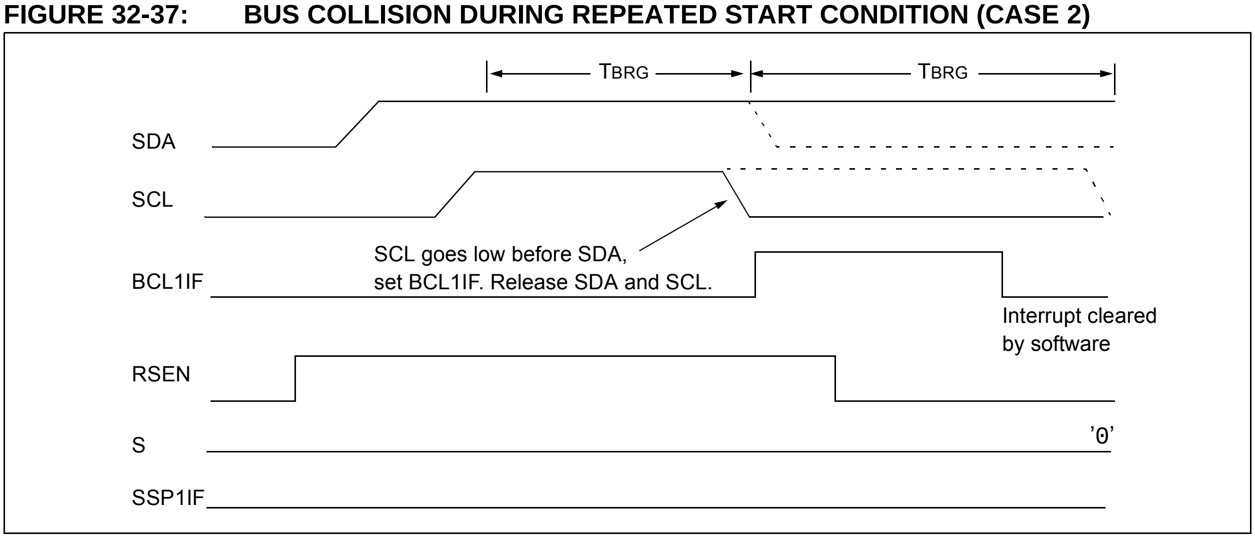


FIGURE 32-37: BUS COLLISION DURING REPEATED START CONDITION (CASE 2)



REGISTER 32-4: SSP1CON3: SSP1 CONTROL REGISTER 3

R-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
ACKTIM(3)	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN
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 **ACKTIM:** Acknowledge Time Status bit (I^{2}C mode only)(3)
1 = Indicates the I^{2}C bus is in an Acknowledge sequence, set on 8th falling edge of SCL clock
0 = Not an Acknowledge sequence, cleared on 9TH rising edge of SCL clock

bit 6 **PCIE**: Stop Condition Interrupt Enable bit (I^{2}C mode only)
1 = Enable interrupt on detection of Stop condition
0 = Stop detection interrupts are disabled(2)

bit 5 **SCIE**: Start Condition Interrupt Enable bit (I^{2}C mode only)
1 = Enable interrupt on detection of Start or Restart conditions
0 = Start detection interrupts are disabled(2)

bit 4 **BOEN:** Buffer Overwrite Enable bit
In SPI Slave mode:(1)
1 = SSPBUF updates every time that a new data byte is shifted in ignoring the BF bit
0 = If new byte is received with BF bit of the SSPSTAT register already set, SSPOV bit of the SSPCON1 register is set, and the buffer is not updated
In I^{2}C Master mode and SPI Master mode:
This bit is ignored.
In I^{2}C Slave mode:
1 = SSPBUF is updated and $\overline{ACK}$ is generated for a received address/data byte, ignoring the state of the SSPOV bit only if the BF bit = 0.
0 = SSPBUF is only updated when SSPOV is clear

bit 3 **SDAHT:** SDA Hold Time Selection bit (I^{2}C mode only)
1 = Minimum of 300 ns hold time on SDA after the falling edge of SCL
0 = Minimum of 100 ns hold time on SDA after the falling edge of SCL

bit 2 **SBCDE:** Slave Mode Bus Collision Detect Enable bit (I^{2}C Slave mode only)
If, on the rising edge of SCL, SDA is sampled low when the module is outputting a high state, the BCL1IF bit of the PIR3 register is set, and bus goes idle
1 = Enable slave bus collision interrupts
0 = Slave bus collision interrupts are disabled

bit 1 **AHEN:** Address Hold Enable bit (I^{2}C Slave mode only)
1 = Following the eighth falling edge of SCL for a matching received address byte; CKP bit of the SSPCON1 register will be cleared and the SCL will be held low.
0 = Address holding is disabled

bit 0 **DHEN:** Data Hold Enable bit (I^{2}C Slave mode only)
1 = Following the eighth falling edge of SCL for a received data byte; slave hardware clears the CKP bit of the SSP-CON1 register and SCL is held low.
0 = Data holding is disabled

Note 1: For daisy-chained SPI operation; allows the user to ignore all but the last received byte. SSPOV is still set when a new byte is received and BF = 1, but hardware continues to write the most recent byte to SSPBUF.
2: This bit has no effect in Slave modes that Start and Stop condition detection is explicitly listed as enabled.
3: The ACKTIM Status bit is only active when the AHEN bit or DHEN bit is set.

REGISTER 34-1: CLKRCON: REFERENCE CLOCK CONTROL REGISTER

R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
CLKREN	—	—	CLKRDC<1:0>		CLKRDIV<2:0>		
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 **CLKREN:** Reference Clock Module Enable bit
1 = Reference clock module enabled
0 = Reference clock module is disabled

bit 6-5 **Unimplemented:** Read as ‘0’

bit 4-3 **CLKRDC<1:0>:** Reference Clock Duty Cycle bits (1)
11 = Clock outputs duty cycle of 75%
10 = Clock outputs duty cycle of 50%
01 = Clock outputs duty cycle of 25%
00 = Clock outputs duty cycle of 0%

bit 2-0 **CLKRDIV<2:0>:** Reference Clock Divider bits
111 = Base clock value divided by 128
110 = Base clock value divided by 64
101 = Base clock value divided by 32
100 = Base clock value divided by 16
011 = Base clock value divided by 8
010 = Base clock value divided by 4
001 = Base clock value divided by 2
000 = Base clock value

Note 1: Bits are valid for reference clock divider values of two or larger, the base clock cannot be further divided.

TABLE 37-17: ZERO CROSS DETECT (ZCD) SPECIFICATIONS

Standard Operating Conditions (unless otherwise stated) VDD = 3.0V, TA = 25°C							
Param. No.	**Sym.**	**Characteristics**	**Min.**	**Typ†**	**Max.**	**Units**	**Comments**
ZC01	VPINZC	Voltage on Zero Cross Pin	—	0.75	—	V	
ZC02	IZCD_MAX	Maximum source or sink current	—	—	600	μA	
ZC03	TRESPH	Response Time, Rising Edge	—	1	—	μs	
	TRESPL	Response Time, Falling Edge	—	1	—	μs	

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 37-12: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS