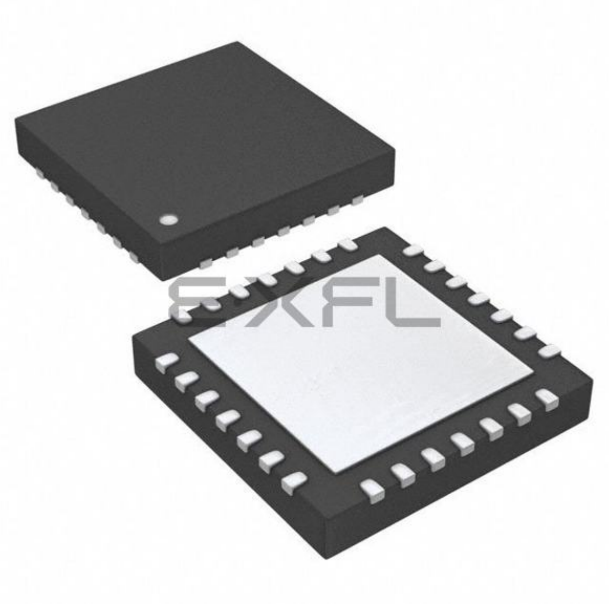


Welcome to E-XFL.COM

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, LCD, POR, PWM, WDT
Number of I/O	24
Program Memory Size	14KB (8K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 5.5V
Data Converters	A/D 20x12b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	28-UFQFN Exposed Pad
Supplier Device Package	28-UQFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f19155-e-mv

1.1 Register and Bit Naming Conventions

1.1.1 REGISTER NAMES

When there are multiple instances of the same peripheral in a device, the peripheral control registers will be depicted as the concatenation of a peripheral identifier, peripheral instance, and control identifier. The control registers section will show just one instance of all the register names with an 'x' in the place of the peripheral instance number. This naming convention may also be applied to peripherals when there is only one instance of that peripheral in the device to maintain compatibility with other devices in the family that contain more than one.

1.1.2 BIT NAMES

There are two variants for bit names:

- Short name: Bit function abbreviation
- Long name: Peripheral abbreviation + short name

1.1.2.1 Short Bit Names

Short bit names are an abbreviation for the bit function. For example, some peripherals are enabled with the EN bit. The bit names shown in the registers are the short name variant.

Short bit names are useful when accessing bits in C programs. The general format for accessing bits by the short name is *RegisterName*bits.*ShortName*. For example, the enable bit, EN, in the COG1CON0 register can be set in C programs with the instruction `COG1CON0bits.EN = 1`.

Short names are generally not useful in assembly programs because the same name may be used by different peripherals in different bit positions. When this occurs, during the include file generation, all instances of that short bit name are appended with an underscore plus the name of the register in which the bit resides to avoid naming contentions.

1.1.2.2 Long Bit Names

Long bit names are constructed by adding a peripheral abbreviation prefix to the short name. The prefix is unique to the peripheral, thereby making every long bit name unique. The long bit name for the COG1 enable bit is the COG1 prefix, G1, appended with the enable bit short name, EN, resulting in the unique bit name G1EN.

Long bit names are useful in both C and assembly programs. For example, in C the COG1CON0 enable bit can be set with the `G1EN = 1` instruction. In assembly, this bit can be set with the `BSF COG1CON0,G1EN` instruction.

1.1.2.3 Bit Fields

Bit fields are two or more adjacent bits in the same register. Bit fields adhere only to the short bit naming convention. For example, the three Least Significant bits of the COG1CON0 register contain the mode control bits. The short name for this field is MD. There is no long bit name variant. Bit field access is only possible in C programs. The following example demonstrates a C program instruction for setting the COG1 to the Push-Pull mode:

```
COG1CON0bits.MD = 0x5;
```

Individual bits in a bit field can also be accessed with long and short bit names. Each bit is the field name appended with the number of the bit position within the field. For example, the Most Significant mode bit has the short bit name MD2 and the long bit name is G1MD2. The following two examples demonstrate assembly program sequences for setting the COG1 to Push-Pull mode.

EXAMPLE 1-1: ASSEMBLY SEQUENCE 1 FOR SETTING COG1 TO PUSH-PULL MODE

```
MOVLW  ~(1<<G1MD1)
ANDWF  COG1CON0,F
MOVLW  1<<G1MD2 | 1<<G1MD0
IORWF  COG1CON0,F
```

EXAMPLE 1-2: ASSEMBLY SEQUENCE 2 FOR SETTING COG1 TO PUSH-PULL MODE

```
BSF    COG1CON0,G1MD2
BCF    COG1CON0,G1MD1
BSF    COG1CON0,G1MD0
```

1.1.3 REGISTER AND BIT NAMING EXCEPTIONS

1.1.3.1 Status, Interrupt, and Mirror Bits

Status, interrupt enables, interrupt flags, and mirror bits are contained in registers that span more than one peripheral. In these cases, the bit name shown is unique so there is no prefix or short name variant.

1.1.3.2 Legacy Peripherals

There are some peripherals that do not strictly adhere to these naming conventions. Peripherals that have existed for many years and are present in almost every device are the exceptions. These exceptions were necessary to limit the adverse impact of the new conventions on legacy code. Peripherals that do adhere to the new convention will include a table in the registers section indicating the long name prefix for each peripheral instance. Peripherals that fall into the exception category will not have this table. These peripherals include, but are not limited to, the following:

- EUSART
- MSSP

TABLE 1-2: PIC16(L)F19155/56 PINOUT DESCRIPTION (CONTINUED)

Name	Function	Input Type	Output Type	Description
RC0/T1CKI(1)/SMTWIN1(1)/IOCC0/SOSCO	RC0	TTL/ST	CMOS/OD	General purpose I/O.
	T1CKI(1)	—	—	Timer1 clock input.
	SMTWIN1(1)	—	—	SMT window input.
	IOCC0	TTL/ST	—	Interrupt-on-change input.
	SOSCO	#VALUE!	AN	32.768 kHz secondary oscillator crystal driver output.
RC1/T4IN(1)/SMTSIG1(1)/CCP2(1)/IOCC1/SOSCI	RC1	TTL/ST	CMOS/OD	General purpose I/O.
	T4IN(1)	—	—	Timer4 external input.
	SMTSIG1(1)	—	—	SMT signal input.
	CCP2(1)	—	—	CCP Capture Input.
	IOCC1	TTL/ST	—	Interrupt-on-change input.
	SOSCI	—	—	32.768 kHz secondary oscillator crystal driver input.
RC2/CCP1(1)/IOCC2/ANC2/SEG18/COM2	RC2	TTL/ST	CMOS/OD	General purpose I/O.
	CCP1(1)	—	—	CCP Capture Input.
	IOCC2	TTL/ST	—	Interrupt-on-change input.
	ANC2	AN	—	ADC Channel input.
	SEG18	—	AN	LCD Analog output.
	COM2	—	AN	LCD Driver Common Outputs.
RC3/T2IN/SCL(3,4)/SCK(1)/SEG19	RC3	TTL/ST	CMOS/OD	General purpose I/O.
	T2IN(1)	—	—	Timer2 external input.
	SCL(3,4)	I^2C	OD	MSSP I^2Cclock input/output.
	SCK(1)	TTL/ST	—	MSSP SPI clock input/output
	IOCC3	TTL/ST	—	Interrupt-on-change input.
	ANC3	AN	—	ADC Channel input.
	SEG19	—	AN	LCD Analog output.
RC4/SDA(3,4)/SDI(1)/IOCC4/ANC4/SEG20	RC4	TTL/ST	CMOS/OD	General purpose I/O.
	SDA(3,4)	TTL/ST	—	MSSP I^2C data input/output.
	SDI(1)	I^2C	OD	MSSP SPI serial data in.
	IOCC4	TTL/ST	—	Interrupt-on-change input.
	ANC4	AN	—	ADC Channel input.
	SEG20	—	AN	LCD Analog output.
RC6/CK1(3)/TX1(1)/IOCC6/ANC6/SEG22/COM5/VLCD2	RC6	TTL/ST	CMOS/OD	General purpose I/O.
	CK1(3)	—	—	EUSART synchronous clock out
	TX1(1)	—	—	EUSART asynchronous TX data out
	IOCC6	TTL/ST	—	Interrupt-on-change input.
	ANC6	AN	—	ADC Channel input.
	SEG22	—	AN	LCD Analog output.
	COM5	—	AN	LCD Driver Common Outputs.
	VLCD2	AN	—	LCD analog input

Legend: AN = Analog input or output; CMOS = CMOS compatible input or output; OD = Open-Drain; TTL = TTL compatible input‘; ST = Schmitt Trigger input with CMOS levels; I^2C = Schmitt Trigger input with I^2C; HV = High Voltage; XTAL = Crystal levels

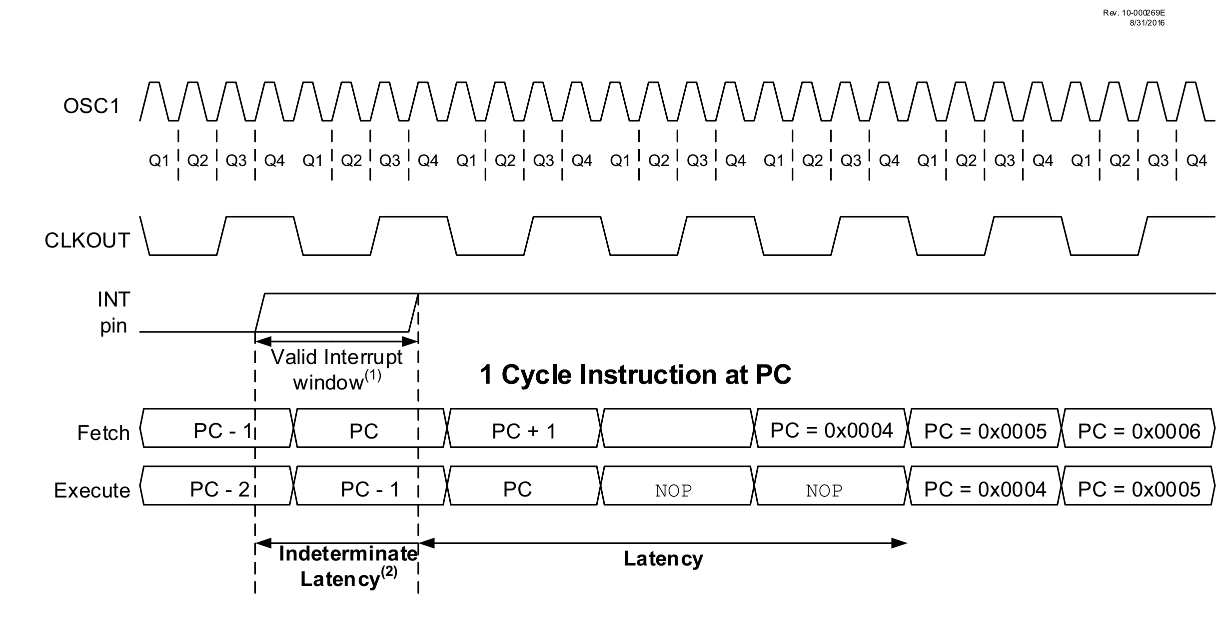
Note 1: This is a PPS remappable input signal. The input function may be moved from the default location shown to one of several other PORTx pins. Refer to Table 14-2 for details on which PORT pins may be used for this signal.

2: All output signals shown in this row are PPS remappable. These signals may be mapped to output onto one of several PORTx pin options as described in Table 14-3.

3: This is a bidirectional signal. For normal module operation, the firmware should map this signal to the same pin in both the PPS input and PPS output registers.

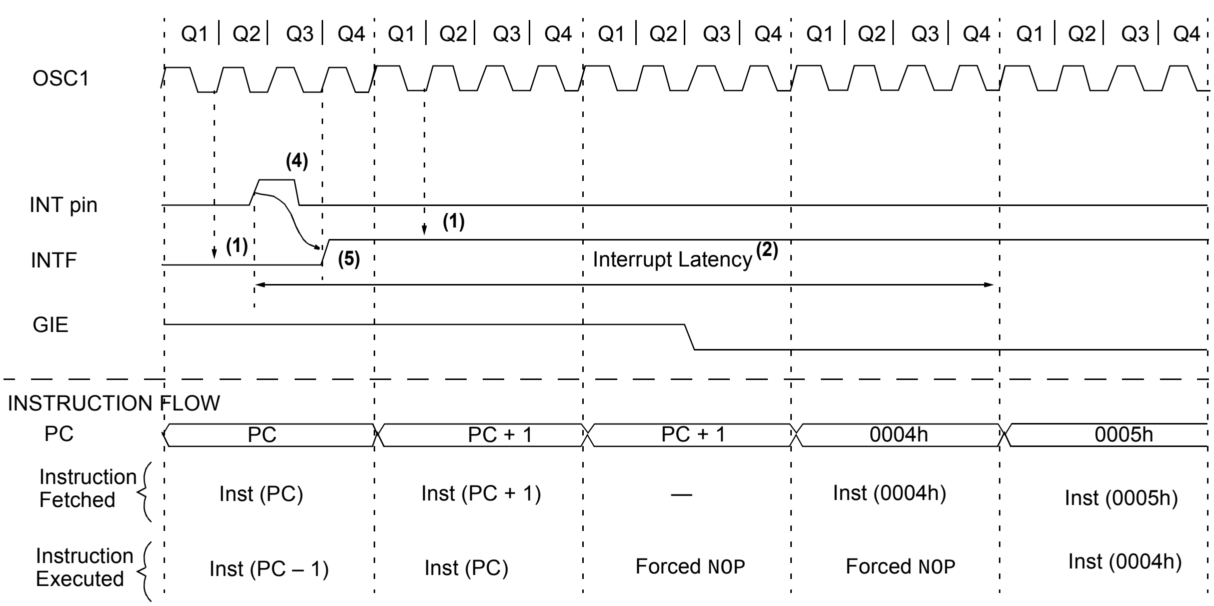
4: These pins are configured for I^2C logic levels. The SCLx/SDAx signals may be assigned to any of the RB1/RB2/RC3/RC4 pins. PPS assignments to the other pins (e.g., RA5) will operate, but input logic levels will be standard TTL/ST, as selected by the INLVL register, instead of the I^2C specific or SMBus input buffer thresholds.

FIGURE 10-2: INTERRUPT LATENCY



Note 1: An interrupt may occur at any time during the interrupt window.
2: Since an interrupt may occur any time during the interrupt window, the actual latency can vary.

FIGURE 10-3: INT PIN INTERRUPT TIMING



Note 1: INTF flag is sampled here (every Q1).

2: Asynchronous interrupt latency = 3-5 TCY. Synchronous latency = 3-4 TCY, where TCY = instruction cycle time. Latency is the same whether Inst (PC) is a single cycle or a 2-cycle instruction.

3: For minimum width of INT pulse, refer to AC specifications in **Section 39.0 “Electrical Specifications”**.

4: INTF is enabled to be set any time during the Q4-Q1 cycles.

REGISTER 10-8: PIE6: PERIPHERAL INTERRUPT ENABLE REGISTER 6

R/W-0/0	U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0
CRIE	—	—	—	—	—	CCP2IE	CCP1IE
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	HS = Hardware set

bit 7 **CRIE:** Clock Recovery Interrupt Enable bit
1 = Clock Recovery interrupt is enabled
0 = Clock Recovery interrupt is disabled

bit 6-2 **Unimplemented:** Read as ‘0’

bit 1 **CCP2IE:** CCP2 Interrupt Enable bit
1 = CCP2 interrupt is enabled
0 = CCP2 interrupt is disabled

bit 0 **CCP1IE:** CCP1 Interrupt Enable bit
1 = CCP1 interrupt is enabled
0 = CCP1 interrupt is disabled

Note: Bit PEIE of the INTCON register must be set to enable any peripheral interrupt controlled by registers PIE1-PIE8.

TABLE 10-1: SUMMARY OF REGISTERS ASSOCIATED WITH INTERRUPTS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	164
PIE0	—	—	TMR0IE	IOCIE	—	—	—	INTE	165
PIE1	OSFIE	CSWIE	—	—	—	—	ADTIE	ADIE	166
PIE2	—	ZCDIE	—	—	—	—	C2IE	C1IE	167
PIE3	RC2IE	TX2IE	RC1IE	TX1IE	—	—	BCL1IE	SSP1IE	168
PIE4	—	—	—	—	TMR4IE	—	TMR2IE	TMR1IE	169
PIE5	CLC4IE	CLC3IE	CLC2IE	CLC1IE	—	—	—	TMR1GIE	170
PIE6	CRIE	—	—	—	—	—	CCP2IE	CCP1IE	171
PIE7	—	—	NVMIE	—	—	—	—	CWG1IE	172
PIE8	LCDIE	RTCCIE	—	—	—	SMT1PWAIE	SMT1PRAIE	SMT1IE	173
PIR0	—	—	TMR0IF	IOCIF	—	—	—	INTF	174
PIR1	OSFIF	CSWIF	—	—	—	—	ADTIF	ADIF	175
PIR2	—	ZCDIF	—	—	—	—	C2IF	C1IF	176
PIR3	RC2IF	TX2IF	RC1IF	TX1IF	—	—	BCL1IF	SSP1IF	177
PIR4	—	—	—	—	TMR4IF	—	TMR2IF	TMR1IF	178
PIR5	CLC4IF	CLC3IF	CLC2IF	CLC1IF	—	—	—	TMR1GIF	179
PIR6	CRIF	—	—	—	—	—	CCP2IF	CCP1IF	180
PIR7	—	—	NVMIF	—	—	—	—	CWG1IF	181
PIR8	LCDIF	RTCCIF	—	—	—	SMT1PWAIF	SMT1PRAIF	SMT1IF	182

Legend: — = unimplemented location, read as ‘0’. Shaded cells are not used by interrupts.

11.2.3.2 Peripheral Usage in Sleep

Some peripherals that can operate in Sleep mode will not operate properly with the Low-Power Sleep mode selected. The Low-Power Sleep mode is intended for use with these peripherals:

- Brown-out Reset (BOR)
- Windowed Watchdog Timer (WWDT)
- External interrupt pin/interrupt-on-change pins
- Timer1 (with external clock source)

It is the responsibility of the end user to determine what is acceptable for their application when setting the VREGPM settings in order to ensure operation in Sleep.

Note:	The PIC16LF19155/56/75/76/85/86 does not have a configurable Low-Power Sleep mode. PIC16LF19155/56/75/76/85/86 is an unregulated device and is always in the lowest power state when in Sleep, with no wake-up time penalty. This device has a lower maximum V_{DD} and I/O voltage than the PIC16F19155/56/75/76/85/86. See **Section 39.0 "Electrical Specifications"** for more information.

11.3 IDLE Mode

When the Idle Enable (IDLEN) bit is clear (IDLEN = 0), the SLEEP instruction will put the device into full Sleep mode (see **Section 11.2 "Sleep Mode"**). When IDLEN is set (IDLEN = 1), the SLEEP instruction will put the device into IDLE mode. In IDLE mode, the CPU and memory operations are halted, but the peripheral clocks continue to run. This mode is similar to DOZE mode, except that in IDLE both the CPU and PFM are shut off.

Note:	Peripherals using F_{OSC} will continue running while in Idle (but not in Sleep). Peripherals using HFINTOSC, LFINTOSC, or SOSC will continue running in both Idle and Sleep.

Note:	If CLKOUT is enabled (CLKOUT = 0, Configuration Word 1), the output will continue operating while in Idle.

11.3.0.1 Idle and Interrupts

IDLE mode ends when an interrupt occurs (even if GIE = 0), but IDLEN is not changed. The device can re-enter IDLE by executing the SLEEP instruction.

If Recover-on-Interrupt is enabled (ROI = 1), the interrupt that brings the device out of Idle also restores full-speed CPU execution when doze is also enabled.

11.3.0.2 Idle and WWDT

When in IDLE, the WWDT Reset is blocked and will instead wake the device. The WWDT wake-up is not an interrupt, therefore ROI does not apply.

Note:	The WWDT can bring the device out of IDLE, in the same way it brings the device out of Sleep. The DOZEN bit is not affected.

REGISTER 14-35: LATE: PORTE DATA LATCH REGISTER

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	U-0	R/W-x/u	R/W-x/u
LATE7	LATE6	LATE5	LATE4	LATE3	—	LATE1	LATE0
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 7-3 **LATE<7:3>**: RE<7:3> Output Latch Value bits(1)

bit 2 **Unimplemented:** Read as ‘0’

bit 1-0 **LATE<1:0>**: RE<1:0> Output Latch Value bits(1)

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

REGISTER 14-36: ANSELE: PORTE ANALOG SELECT REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	U-0	R/W-1/1	R/W-1/1
ANSE7	ANSE6	ANSE5	ANSE4	ANSE3	—	ANSE1	ANSE0
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 7-3 **ANSE<7:3>**: Analog Select between Analog or Digital Function on pins RE<7:3>, respectively
1 = Analog input. Pin is assigned as analog input(1). Digital input buffer disabled.
0 = Digital I/O. Pin is assigned to port or digital special function.

bit 2 **Unimplemented:** Read as ‘0’

bit 1-0 **ANSE<1:0>**: Analog Select between Analog or Digital Function on pins RE<1:0>, respectively
1 = Analog input. Pin is assigned as analog input(1). Digital input buffer disabled.
0 = Digital I/O. Pin is assigned to port or digital special function.

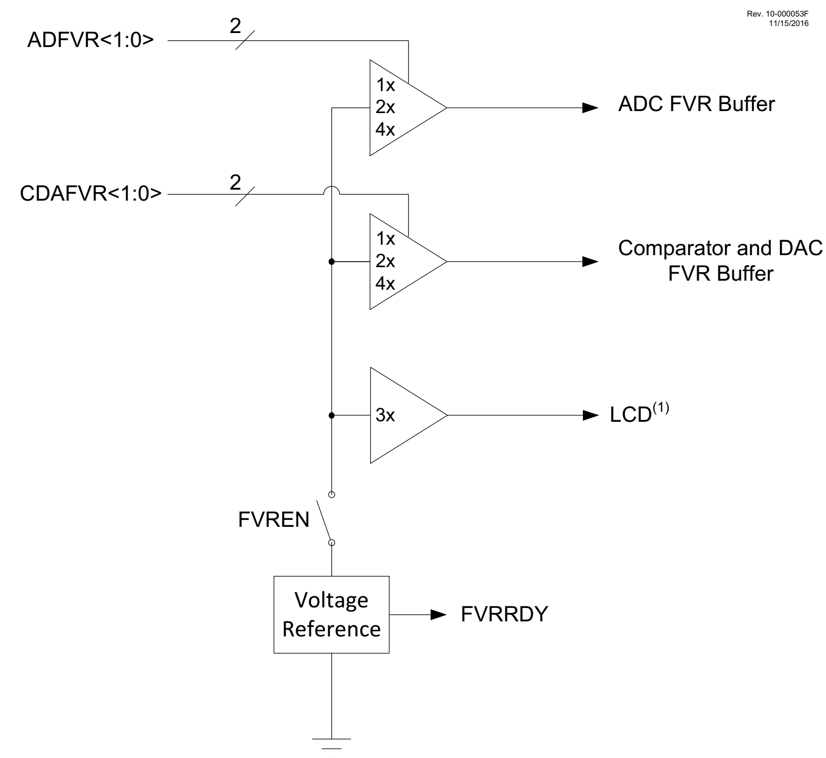
Note 1: When setting a pin to an analog input, the corresponding TRIS bit must be set to Input mode in order to allow external control of the voltage on the pin.

TABLE 15-4: SUMMARY OF REGISTERS ASSOCIATED WITH THE PPS MODULE (CONTINUED)

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
RD1PPS	—	—	—	RD1PPS<4:0>					265
RD2PPS	—	—	—	RD2PPS<4:0>					265
RD3PPS	—	—	—	RD3PPS<4:0>					265
RD4PPS	—	—	—	RD4PPS<4:0>					265
RD5PPS	—	—	—	RD5PPS<4:0>					265
RD6PPS	—	—	—	RD6PPS<4:0>					265
RD7PPS	—	—	—	RD7PPS<4:0>					265
RE0PPS	—	—	—	RE0PPS<4:0>					265
RE1PPS	—	—	—	RE1PPS<4:0>					265
RE2PPS	—	—	—	RE2PPS<4:0>					265
RF0PPS	—	—	—	RF0PPS<4:0>					265
RF1PPS	—	—	—	RF1PPS<4:0>					265
RF2PPS	—	—	—	RF2PPS<4:0>					265
RF3PPS	—	—	—	RF3PPS<4:0>					265
RF4PPS	—	—	—	RF4PPS<4:0>					265
RF5PPS	—	—	—	RF5PPS<4:0>					265
RF6PPS	—	—	—	RF6PPS<4:0>					265
RF7PPS	—	—	—	RF7PPS<4:0>					265

Legend: — = unimplemented, read as ‘0’. Shaded cells are unused by the PPS module.

FIGURE 18-1: VOLTAGE REFERENCE BLOCK DIAGRAM



Note 1: To enable the 3x Buffer to the LCD module, the LCDVSRC<3:0> of the LCDVCON2 register need to be set to `0b0011`.

21.4 Operation During Sleep

The DAC continues to function during Sleep. When the device wakes up from Sleep through an interrupt or a Watchdog Timer time-out, the contents of the DAC1CON0 register are not affected.

21.5 Effects of a Reset

A device Reset affects the following:

- DAC is disabled.
- DAC output voltage is removed from the DAC1OUT1/2 pins.
- The DAC1R<4:0> range select bits are cleared.

REGISTER 22-2: CMxCON1: COMPARATOR Cx CONTROL REGISTER 1

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0
—	—	—	—	—	—	INTP	INTN
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 7-2 **Unimplemented:** Read as ‘0’

bit 1 **INTP:** Comparator Interrupt on Positive-Going Edge Enable bits
1 = The CxIF interrupt flag will be set upon a positive-going edge of the CxOUT bit
0 = No interrupt flag will be set on a positive-going edge of the CxOUT bit

bit 0 **INTN:** Comparator Interrupt on Negative-Going Edge Enable bits
1 = The CxIF interrupt flag will be set upon a negative-going edge of the CxOUT bit
0 = No interrupt flag will be set on a negative-going edge of the CxOUT bit

27.5.9 EDGE-TRIGGERED MONOSTABLE MODES

The Edge-Triggered Monostable modes start the timer on an edge from the external Reset signal input, after the ON bit is set, and stop incrementing the timer when the timer matches the PRx period value. The following edges will start the timer:

- Rising edge (MODE<4:0> = `10001`)
- Falling edge (MODE<4:0> = `10010`)
- Rising or Falling edge (MODE<4:0> = `10011`)

When an Edge-Triggered Monostable mode is used in conjunction with the CCP PWM operation the PWM drive goes active with the external Reset signal edge that starts the timer, but will not go active when the timer matches the PRx value. While the timer is incrementing, additional edges on the external Reset signal will not affect the CCP PWM.

FIGURE 28-8: HIGH AND LOW-MEASURE MODE REPEAT ACQUISITION TIMING DIAGRAM

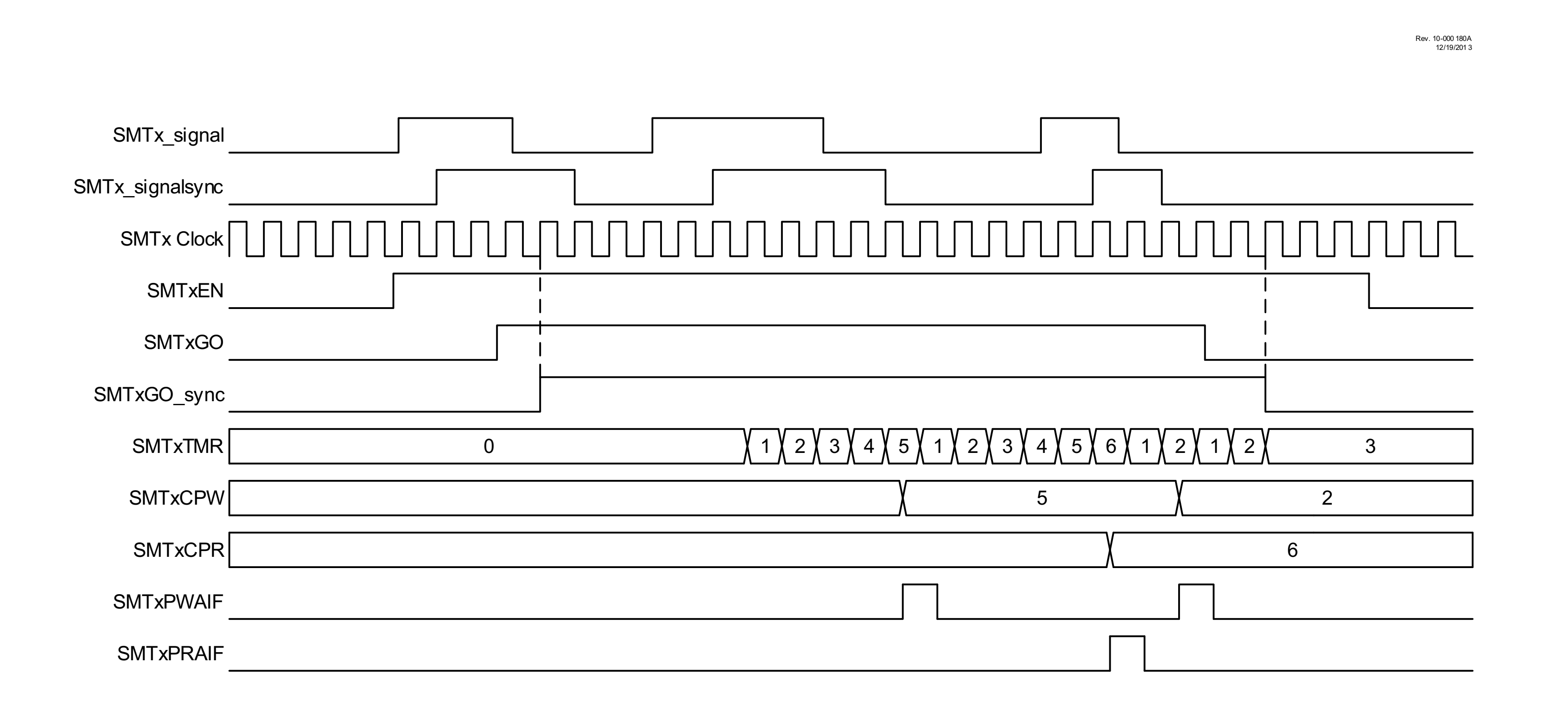
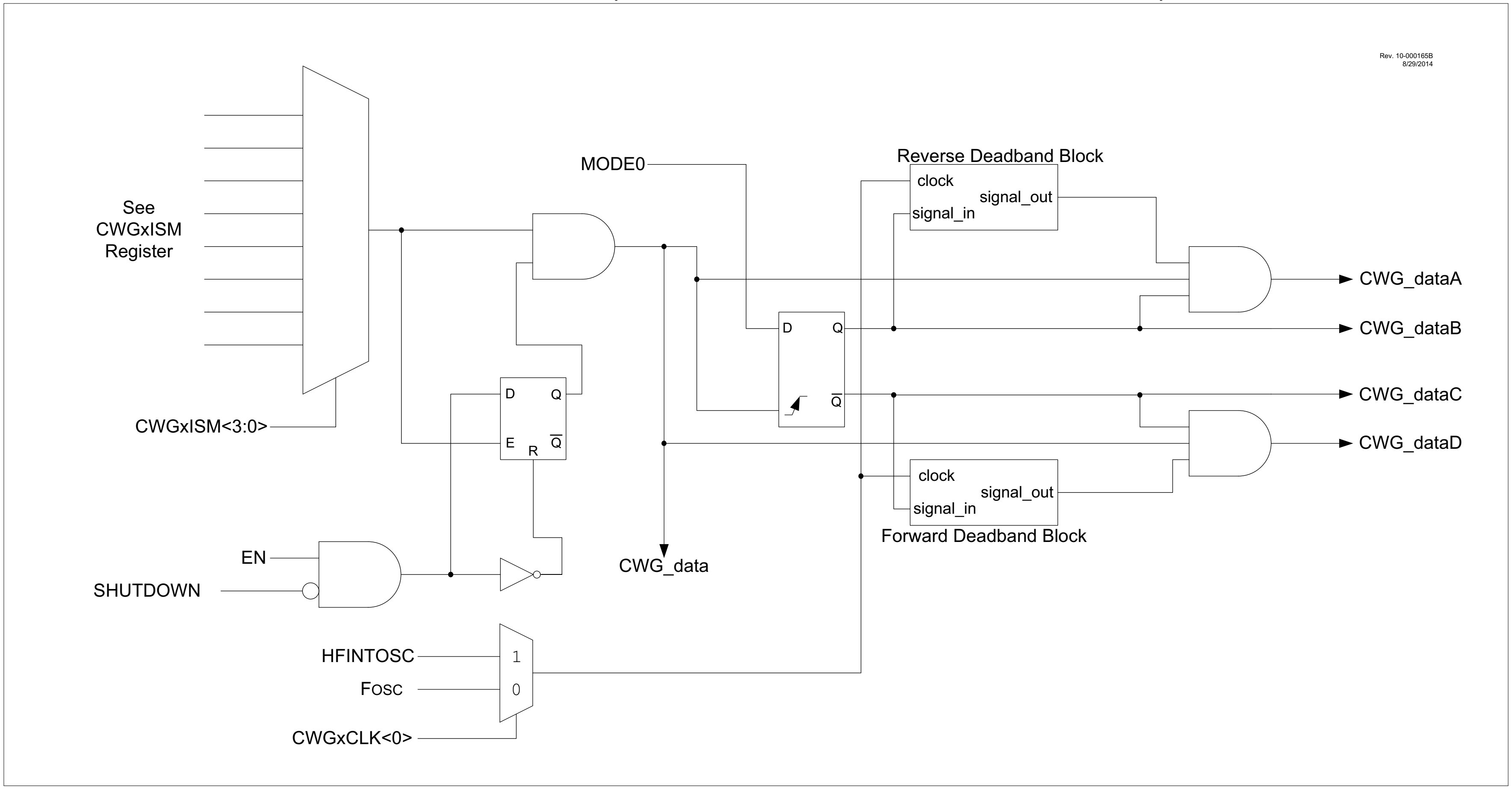


FIGURE 31-3: SIMPLIFIED CWG BLOCK DIAGRAM (FORWARD AND REVERSE FULL-BRIDGE MODES)

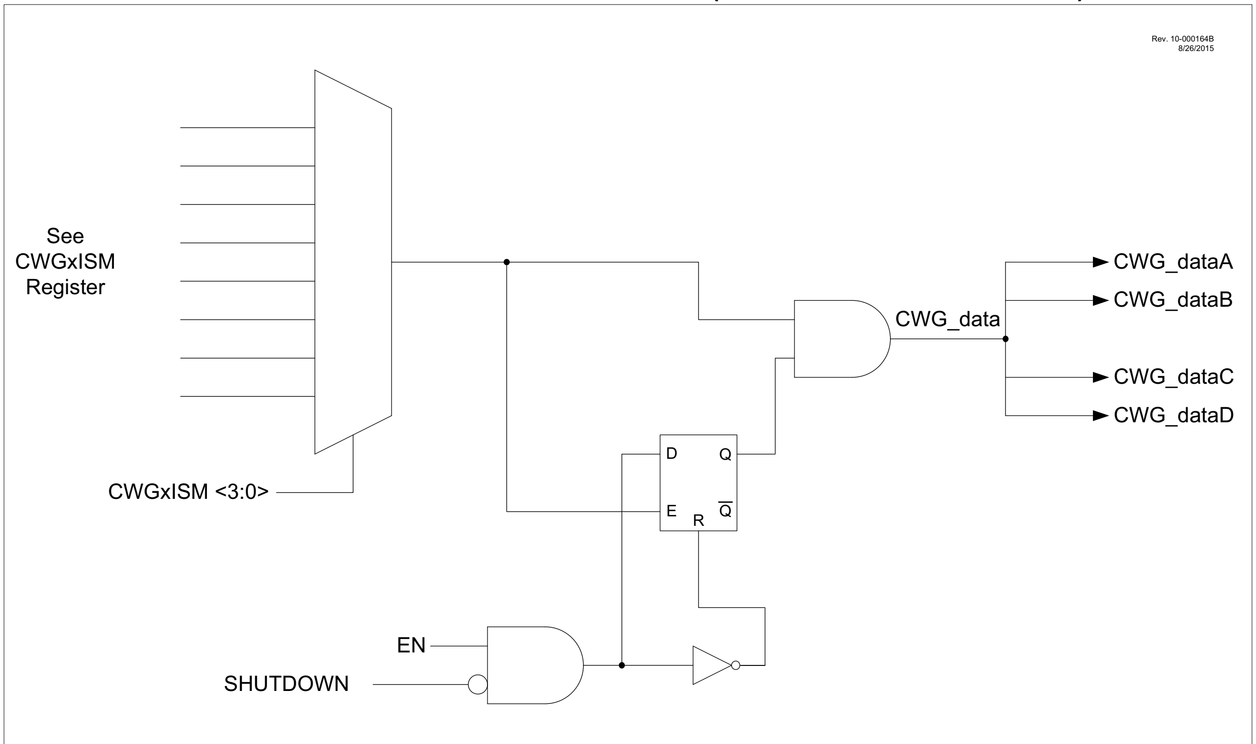


31.1.4 STEERING MODES

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 31.9 "CWG Steering Mode"**.

FIGURE 31-4: SIMPLIFIED CWG BLOCK DIAGRAM (OUTPUT STEERING MODES)



31.2 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.

TABLE 32-4: SUMMARY OF REGISTERS ASSOCIATED WITH CLCx

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	164
PIR5	CLC4IF	CLC3IF	CLC2IF	CLC1IF	—	—	—	TMR1GIF	179
PIE5	CLC4IE	CLC4IE	CLC2IE	CLC1IE	—	—	—	TMR1GIE	170
CLC1CON	LC1EN	—	LC1OUT	LC1INTP	LC1INTN	LC1MODE<2:0>			501
CLC1POL	LC1POL	—	—	—	LC1G4POL	LC1G3POL	LC1G2POL	LC1G1POL	502
CLC1SEL0	—	—	LC1D1S<5:0>						503
CLC1SEL1	—	—	LC1D2S<5:0>						503
CLC1SEL2	—	—	LC1D3S<5:0>						503
CLC1SEL3	—	—	LC1D4S<5:0>						504
CLC1GLS0	LC1G1D4T	LC1G1D4N	LC1G1D3T	LC1G1D3N	LC1G1D2T	LC1G1D2N	LC1G1D1T	LC1G1D1N	505
CLC1GLS1	LC1G2D4T	LC1G2D4N	LC1G2D3T	LC1G2D3N	LC1G2D2T	LC1G2D2N	LC1G2D1T	LC1G2D1N	506
CLC1GLS2	LC1G3D4T	LC1G3D4N	LC1G3D3T	LC1G3D3N	LC1G3D2T	LC1G3D2N	LC1G3D1T	LC1G3D1N	507
CLC1GLS3	LC1G4D4T	LC1G4D4N	LC1G4D3T	LC1G4D3N	LC1G4D2T	LC1G4D2N	LC1G4D1T	LC1G4D1N	508
CLC2CON	LC2EN	—	LC2OUT	LC2INTP	LC2INTN	LC2MODE<2:0>			501
CLC2POL	LC2POL	—	—	—	LC2G4POL	LC2G3POL	LC2G2POL	LC2G1POL	502
CLC2SEL0	—	—	LC2D1S<5:0>						503
CLC2SEL1	—	—	LC2D2S<5:0>						503
CLC2SEL2	—	—	LC2D3S<5:0>						503
CLC2SEL3	—	—	LC2D4S<5:0>						504
CLC2GLS0	LC2G1D4T	LC2G1D4N	LC2G1D3T	LC2G1D3N	LC2G1D2T	LC2G1D2N	LC2G1D1T	LC2G1D1N	505
CLC2GLS1	LC2G2D4T	LC2G2D4N	LC2G2D3T	LC2G2D3N	LC2G2D2T	LC2G2D2N	LC2G2D1T	LC2G2D1N	506
CLC2GLS2	LC2G3D4T	LC2G3D4N	LC2G3D3T	LC2G3D3N	LC2G3D2T	LC2G3D2N	LC2G3D1T	LC2G3D1N	507
CLC2GLS3	LC2G4D4T	LC2G4D4N	LC2G4D3T	LC2G4D3N	LC2G4D2T	LC2G4D2N	LC2G4D1T	LC2G4D1N	508
CLC3CON	LC3EN	—	LC3OUT	LC3INTP	LC3INTN	LC3MODE<2:0>			501
CLC3POL	LC3POL	—	—	—	LC3G4POL	LC3G3POL	LC3G2POL	LC3G1POL	502
CLC3SEL0	—	—	LC3D1S<5:0>						503
CLC3SEL1	—	—	LC3D2S<5:0>						503
CLC3SEL2	—	—	LC3D3S<5:0>						503
CLC3SEL3	—	—	LC3D4S<5:0>						504
CLC3GLS0	LC3G1D4T	LC3G1D4N	LC3G1D3T	LC3G1D3N	LC3G1D2T	LC3G1D2N	LC3G1D1T	LC3G1D1N	505
CLC3GLS1	LC3G2D4T	LC3G2D4N	LC3G2D3T	LC3G2D3N	LC3G2D2T	LC3G2D2N	LC3G2D1T	LC3G2D1N	506
CLC3GLS2	LC3G3D4T	LC3G3D4N	LC3G3D3T	LC3G3D3N	LC3G3D2T	LC3G3D2N	LC3G3D1T	LC3G3D1N	507
CLC3GLS3	LC3G4D4T	LC3G4D4N	LC3G4D3T	LC3G4D3N	LC3G4D2T	LC3G4D2N	LC3G4D1T	LC3G4D1N	508
CLC4CON	LC4EN	—	LC4OUT	LC4INTP	LC4INTN	LC4MODE<2:0>			501
CLC4POL	LC4POL	—	—	—	LC4G4POL	LC4G3POL	LC4G2POL	LC4G1POL	502
CLC4SEL0	—	—	LC4D1S<5:0>						503
CLC4SEL1	—	—	LC4D2S<5:0>						503
CLC4SEL2	—	—	LC4D3S<5:0>						503
CLC4SEL3	—	—	LC4D4S<5:0>						504
CLC4GLS0	LC4G1D4T	LC4G1D4N	LC4G1D3T	LC4G1D3N	LC4G1D2T	LC4G1D2N	LC4G1D1T	LC4G1D1N	505

Legend: — = unimplemented, read as ‘0’. Shaded cells are unused by the CLCx modules.

FIGURE 33-34: BUS COLLISION DURING START CONDITION (SCL = 0)

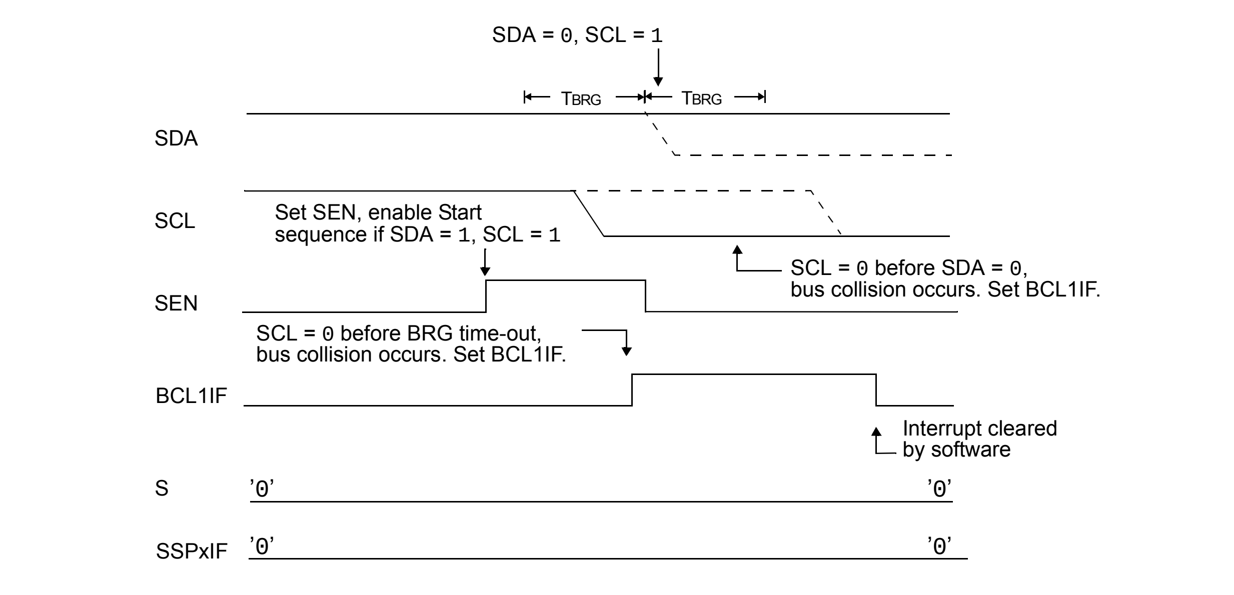
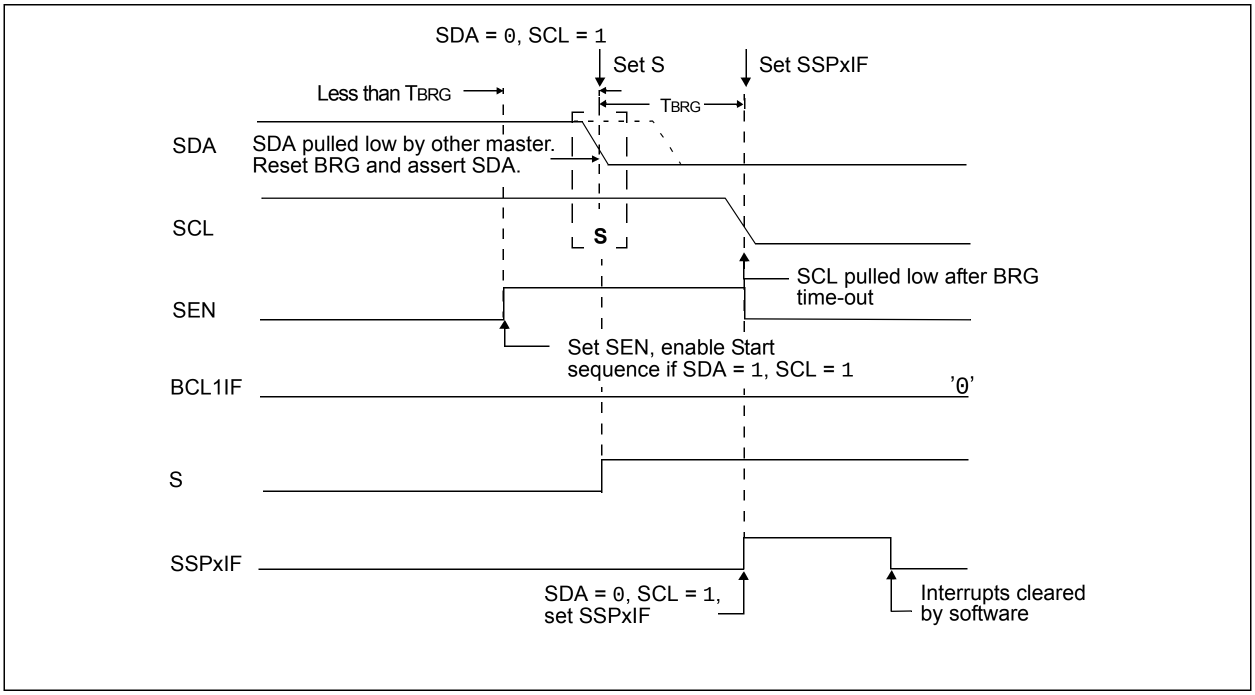


FIGURE 33-35: BRG RESET DUE TO SDA ARBITRATION DURING START CONDITION



35.13 LCD Interrupts

The LCD timing generation provides an interrupt that defines the LCD frame timing. This interrupt can be used to coordinate the writing of the pixel data with the start of a new frame, which produces a visually crisp transition of the image.

This interrupt can also be used to synchronize external events to the LCD. For example, the interface to an external segment driver can be synchronized for segment data updates to the LCD frame.

A new frame is defined as beginning at the leading edge of the COM0 common signal. The interrupt will be set immediately after the LCD controller completes accessing all pixel data required for a frame. This will occur at a fixed interval before the frame boundary (TFINT), as shown in Figure 35-20.

The LCD controller will begin to access the next frame between the interrupt and when the controller accesses the data (TFWR). New data must be written within TFWR, as this is when the LCD controller will begin to access the data for the next frame.

When the LCD driver is running with Type-B waveforms, and the LMUX<3:0> bits are not equal to '0001', the following issues may arise.

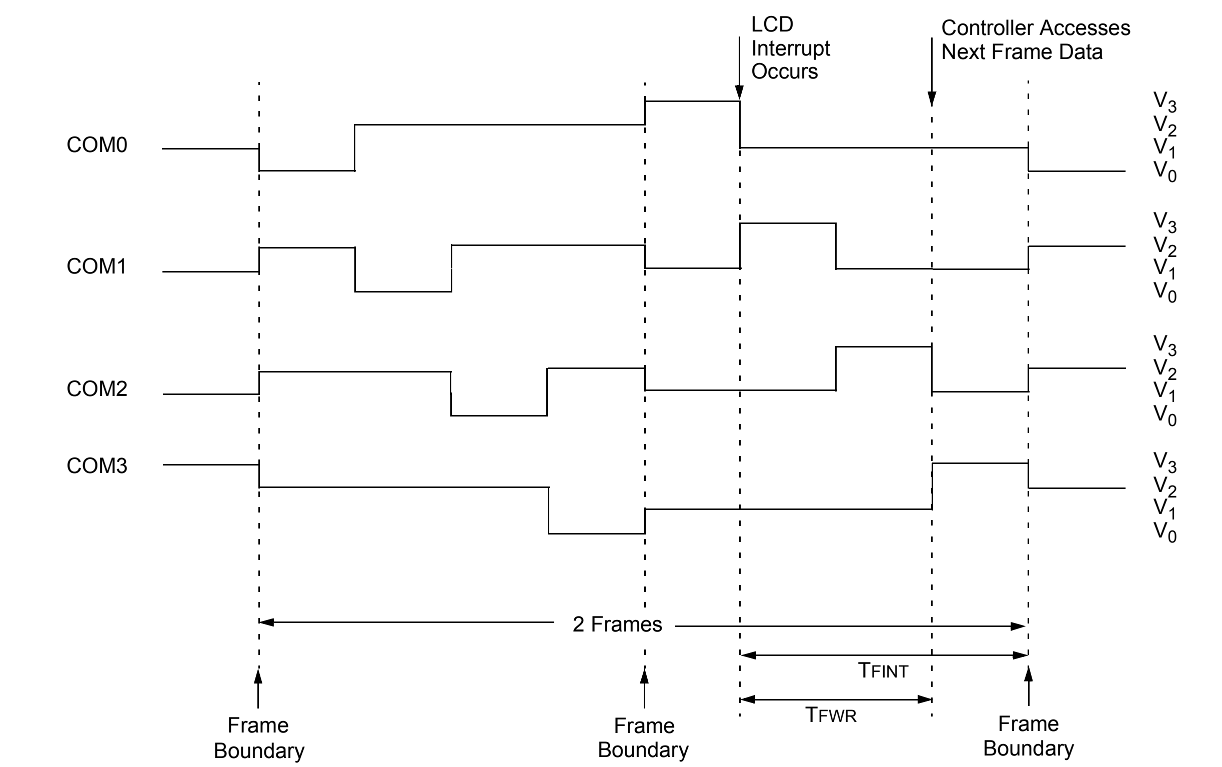
Since the DC voltage on the pixel takes two frames to maintain 0V, the pixel data must not change between subsequent frames. If the pixel data were allowed to change, the waveform for the odd frames would not necessarily be the complement of the waveform generated in the even frames and a DC component would be introduced into the panel.

Because of this, using Type-B waveforms requires synchronizing the LCD pixel updates to occur within a subframe after the frame interrupt.

To correctly sequence writing in Type-B, the interrupt only occurs on complete phase intervals. If the user attempts to write when the write is disabled, the WERR bit (LCDCON<5>) is set.

Note:	The interrupt is not generated when the Type-A waveform is selected and when the Type-B with no multiplex (static) is selected.

FIGURE 35-20: EXAMPLE WAVEFORMS AND INTERRUPT TIMING IN QUARTER DUTY CYCLE DRIVE



TFWR = TFRAME/2 * (LMUX<3:0> + 1) + TCY/2

TFINT = (TFWR/2 – (2 TCY + 40 ns)) → Minimum = 1.5(TFRAME/4) – (2 TCY + 40 ns)
(TFWR/2 – (1 TCY + 40 ns)) → Maximum = 1.5(TFRAME/4) – (1 TCY + 40 ns)

LSLF — Logical Left Shift

Syntax: [*label*] LSLF f {,d}

Operands: $0 \leq f \leq 127$
$d \in [0,1]$

Operation: (f<7>) → C
(f<6:0>) → dest<7:1>
0 → dest<0>

Status Affected: C, Z

Description: The contents of register 'f' are shifted one bit to the left through the Carry flag. A '0' is shifted into the LSb. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f'.

LSRF — Logical Right Shift

Syntax: [*label*] LSRF f {,d}

Operands: $0 \leq f \leq 127$
$d \in [0,1]$

Operation: 0 → 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. A '0' is shifted into the MSb. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f'.

MOVF — Move f

Syntax: [*label*] MOVF f,d

Operands: $0 \leq f \leq 127$
$d \in [0,1]$

Operation: (f) → (dest)

Status Affected: Z

Description: The contents of register f is moved to a destination dependent upon the status of d. If d = 0, destination is W register. If d = 1, the destination is file register f itself. d = 1 is useful to test a file register since status flag Z is affected.

Words: 1

Cycles: 1

Example:

```
MOVF    FSR, 0
```

After Instruction
W = value in FSR register
Z = 1

TABLE 38-1: REGISTER FILE SUMMARY FOR PIC16(L)F19155/56/75/76/85/86 DEVICES

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
68Dh	—	Unimplemented								
68Eh	—	Unimplemented								
68Fh	—	Unimplemented								
690h	—	Unimplemented								
691h	—	Unimplemented								
692h	—	Unimplemented								
693h	—	Unimplemented								
694h	—	Unimplemented								
695h	—	Unimplemented								
696h	—	Unimplemented								
697h	—	Unimplemented								
698h	—	Unimplemented								
699h	—	Unimplemented								
69Ah	—	Unimplemented								
69Bh	—	Unimplemented								
69Ch	—	Unimplemented								
69Dh	—	Unimplemented								
69Eh	—	Unimplemented								
69Fh	—	Unimplemented								
70Ch	PIR0	—	—	TMR0IF	IOCIF	—	—	—	INTF	174
70Dh	PIR1	OSFIF	CSWIF	—	—	—	—	ADTIF	ADIF	175
70Eh	PIR2	—	ZCDIF	—	—	—	—	C2IF	C1IF	176
70Fh	PIR3	RC2IF	TX2IF	RC1IF	TX1IF	—	—	BCL1IF	SSP1IF	177
710h	PIR4	—	—	—	—	TMR4IF	—	TMR2IF	TMR1IF	178
711h	PIR5	CLC4IF	CLC3IF	CLC2IF	CLC1IF	—	—	—	TMR1GIF	179
712h	PIR6	CRIF	—	—	—	—	—	CCP2IF	CCP1IF	180
713h	PIR7	—	—	NVMIF	—	—	—	—	CWG1IF	181
714h	PIR8	LCDIF	RTCCIF	—	—	—	SMT1PWAIF	SMT1PRAIF	SMT1IF	182
715h	—	Unimplemented								
716h	PIE0	—	—	TMR0IE	IOCIE	—	—	—	INTE	165
717h	PIE1	OSFIE	CSWIE	—	—	—	—	ADTIE	ADIE	166
718h	PIE2	—	ZCDIE	—	—	—	—	C2IE	C1IE	167
719h	PIE3	RC2IE	TX2IE	RC1IE	TX1IE	—	—	BCL1IE	SSP1IE	168
71Ah	PIE4	—	—	—	—	TMR4IF	—	TMR2IE	TMR1IE	169
71Bh	PIE5	CLC4IE	CLC3IE	CLC2IE	CLC1IE	—	—	—	TMR1GIE	170
71Ch	PIE6	CRIE	—	—	—	—	—	CCP2IE	CCP1IE	171
71Dh	PIE7	—	—	NVMIE	—	—	—	—	CWG1IE	172
71Eh	PIE8	LCDIE	RTCCIE	—	—	—	SMT1PWAIE	SMT1PRAIE	SMT1IE	173
71Fh	—	Unimplemented								

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