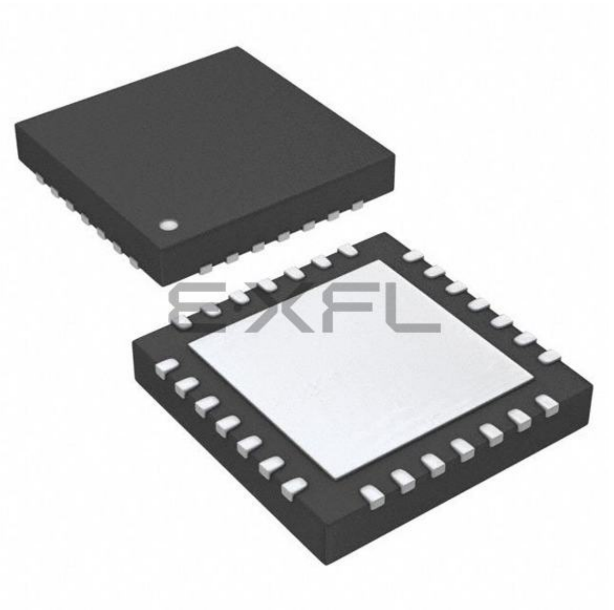


Welcome to E-XFL.COM

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	25
Program Memory Size	28KB (16K x 14)
Program Memory Type	FLASH
EEPROM Size	224 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 24x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	28-VQFN Exposed Pad
Supplier Device Package	28-QFN (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf15356-e-ml

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:

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

Example 2:

```
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 1-3: PIC16(L)F15375/76 PINOUT DESCRIPTION (CONTINUED)

Name	Function	Input Type	Output Type	Description
RD1/AND1/SDA1(1)/SDI1(1)	RD1	TTL/ST	CMOS/OD	General purpose I/O.
	AND1	AN	—	ADC Channel D0 input.
	SDA1(1)	I^2C	OD	MSSP1 I^2C serial data input/output.
	SDI1(1)	TTL/ST	—	MSSP1 SPI serial data input.
RD2/AND2	RD2	TTL/ST	CMOS/OD	General purpose I/O.
	AND2	AN	—	ADC Channel D0 input.
RD3/AND3	RD3	TTL/ST	CMOS/OD	General purpose I/O.
	AND3	AN	—	ADC Channel D0 input.
RD4/AND4	RD4	TTL/ST	CMOS/OD	General purpose I/O.
	AND4	AN	—	ADC Channel D0 input.
RD5/AND5	RD5	TTL/ST	CMOS/OD	General purpose I/O.
	AND5	AN	—	ADC Channel D0 input.
RD6/AND6	RD6	TTL/ST	CMOS/OD	General purpose I/O.
	AND6	AN	—	ADC Channel D0 input.
RD7/AND7	RD7	TTL/ST	CMOS/OD	General purpose I/O.
	AND7	AN	—	ADC Channel D0 input.
RE0/ANE0	RE0	TTL/ST	CMOS/OD	General purpose I/O.
	ANE0	AN	—	ADC Channel D0 input.
RE1/ANE1	RE1	TTL/ST	CMOS/OD	General purpose I/O.
	ANE1	AN	—	ADC Channel D0 input.
RE2/ANE2	RE2	TTL/ST	CMOS/OD	General purpose I/O.
	ANE2	AN	—	ADC Channel D0 input.
RE3/$\overline{\text{MCLR}}$/IOCE3	RE3	TTL/ST	—	General purpose input only (when $\overline{\text{MCLR}}$ is disabled by the Configuration bit).
	$\overline{\text{MCLR}}$	ST	—	Master clear input with internal weak pull-up resistor.
	IOCE3	TTL/ST	—	Interrupt-on-change input.
VDD	VDD	Power	—	Positive supply voltage input.
VSS	VSS	Power	—	Ground reference.

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 15-4 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 15-5, Table 15-6 and Table 15-6.

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.

TABLE 1-4: PIC16(L)F15385/86 PINOUT DESCRIPTION (CONTINUED)

Name	Function	Input Type	Output Type	Description
RB4/ANB4/ADACT(1)/IOCB4	RB4	TTL/ST	CMOS/OD	General purpose I/O.
	ANB4	AN	—	ADC Channel B4 input.
	ADACT(1)	TTL/ST	—	ADC Auto-Conversion Trigger input.
	IOCB4	TTL/ST	—	Interrupt-on-change input.
RB5/ANB5/IOCB5	RB5	TTL/ST	CMOS/OD	General purpose I/O.
	ANB5	AN	—	ADC Channel B5 input.
	IOCB5	TTL/ST	—	Interrupt-on-change input.
RB6/ANB6/CLCIN2(1)/TX2/CK2(1)/ IOCB6/ICSPCLK	RB6	TTL/ST	CMOS/OD	General purpose I/O.
	ANB6	AN	—	ADC Channel B6 input.
	CLCIN2(1)	TTL/ST	—	Configurable Logic Cell source input.
	TX2	TTL/ST	—	EUSART2 Asynchronous mode receiver data input.
	CK2(1)	TTL/ST	CMOS/OD	EUSART2 Synchronous mode clock input/output.
	IOCB6	TTL/ST	—	Interrupt-on-change input.
	ICSPCLK	ST	—	In-Circuit Serial Programming™ and debugging clock input.
RB7/ANB7/DAC1OUT2/CLCIN3(1)/ RX2/DT2(1)/IOCB7/ICSPDAT	RB7	TTL/ST	CMOS/OD	General purpose I/O.
	ANB7	AN	—	ADC Channel B7 input.
	DAC1OUT2	—	AN	Digital-to-Analog Converter output.
	CLCIN3(1)	TTL/ST	—	Configurable Logic Cell source input.
	RX2	TTL/ST	—	EUSART2 Asynchronous mode receiver data input.
	DT2	TTL/ST	CMOS/OD	EUSART2 Synchronous mode data input/output.
	IOCB7	TTL/ST	—	Interrupt-on-change input.
	ICSPDAT	ST	CMOS	In-Circuit Serial Programming™ and debugging data input/out-put.
RC0/ANC0/T1CKI(1)/IOCC0/SOSCO	RC0	TTL/ST	CMOS/OD	General purpose I/O.
	ANC0	AN	—	ADC Channel C0 input.
	T1CKI(1)	TTL/ST	—	Timer1 external digital clock input.
	IOCC0	TTL/ST	—	Interrupt-on-change input.
	SOSCO	—	AN	32.768 kHz secondary oscillator crystal driver output.
RC1/ANC1/CCP2(1)/IOCC1/SOSCI	RC1	TTL/ST	CMOS/OD	General purpose I/O.
	ANC1	AN	—	ADC Channel C1 input.
	CCP2(1)	TTL/ST	CMOS/OD	Capture/compare/PWM2 (default input location for capture function).
	IOCC1	TTL/ST	—	Interrupt-on-change input.
	SOSCI	AN	—	32.768 kHz secondary oscillator crystal driver 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 15-4 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 15-5, Table 15-6 and Table 15-7.

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.

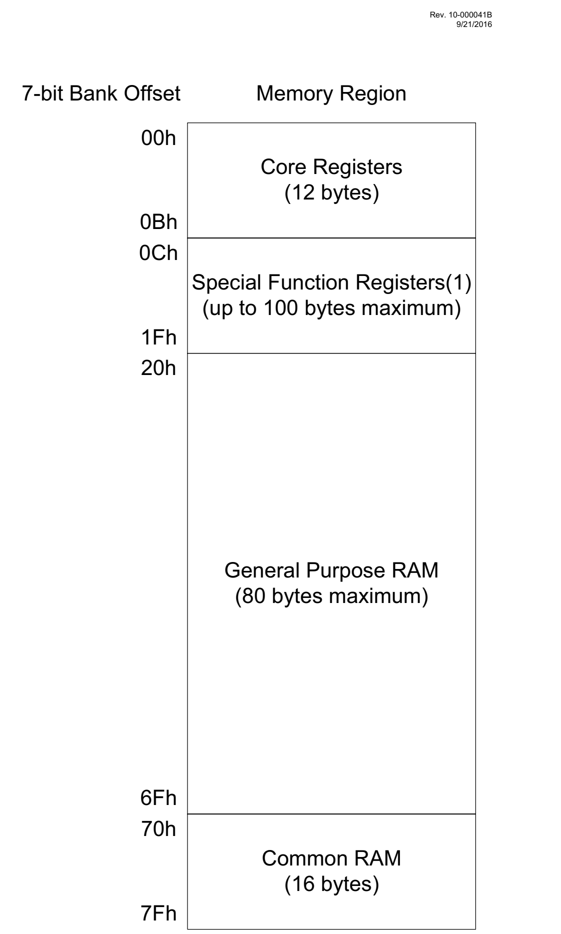
4: These pins are configured for I^2C logic levels. The SCLx/SDAx signals may be assigned to any of the RB1/RB2/RC3/RC4 pins. PPS assignments to the other pins (e.g., RA5) will operate, but input logic levels will be standard TTL/ST, as selected by the INLVL register, instead of the I^2C specific or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

4.3 Data Memory Organization

The data memory is partitioned into 64 memory banks with 128 bytes in each bank. Each bank consists of:

- 12 core registers
- Up to 100 Special Function Registers (SFR)
- Up to 80 bytes of General Purpose RAM (GPR)
- 16 bytes of common RAM

FIGURE 4-3: BANKED MEMORY PARTITIONING



Note 1: This table shows the address for an example bank with 20 Bytes of SFRs only.

4.3.1 BANK SELECTION

The active bank is selected by writing the bank number into the Bank Select Register (BSR). All data memory can be accessed either directly (via instructions that use the file registers) or indirectly via the two File Select Registers (FSR). See **Section 4.6 "Indirect Addressing"** for more information.

Data memory uses a 13-bit address. The upper six bits of the address define the Bank address and the lower seven bits select the registers/RAM in that bank.

4.3.2 CORE REGISTERS

The core registers contain the registers that directly affect the basic operation. The core registers occupy the first 12 addresses of every data memory bank (addresses x00h/x08h through x0Bh/x8Bh). These registers are listed below in Table 4-3.

TABLE 4-3: CORE REGISTERS

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

4.3.2.1 STATUS Register

The STATUS register, shown in Register 4-1, contains:

- the arithmetic status of the ALU
- the Reset status

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

For example, `CLRF STATUS` will clear bits <4:3> and <1:0>, and set the Z bit. This leaves the STATUS register as '`000u u1uu`' (where u = unchanged).

It is recommended, therefore, that only `BCF, BSF, SWAPF` and `MOVWF` instructions are used to alter the STATUS register, because these instructions do not affect any Status bits. For other instructions not affecting any Status bits, refer to **Section 36.0 "Instruction Set Summary"**.

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

REGISTER 4-1: STATUS: STATUS REGISTER

U-0	U-0	U-0	R-1/q	R-1/q	R/W-0/u	R/W-0/u	R/W-0/u
—	—	—	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC(1)	C(1)
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 7-5 **Unimplemented:** Read as '`0`'

bit 4 **$\overline{\text{TO}}$:** Time-Out bit
`1` = After power-up, `CLRWDT` instruction or `SLEEP` instruction
`0` = A WDT time-out occurred

bit 3 **$\overline{\text{PD}}$:** Power-Down bit
`1` = After power-up or by the `CLRWDT` instruction
`0` = By execution of the `SLEEP` instruction

bit 2 **Z:** Zero bit
`1` = The result of an arithmetic or logic operation is zero
`0` = The result of an arithmetic or logic operation is not zero

bit 1 **DC:** Digit Carry/$\overline{\text{Digit Borrow}}$ bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions)(1)
`1` = A carry-out from the 4th low-order bit of the result occurred
`0` = No carry-out from the 4th low-order bit of the result

bit 0 **C:** Carry/$\overline{\text{Borrow}}$ bit(1) (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions)(1)
`1` = A carry-out from the Most Significant bit of the result occurred
`0` = No carry-out from the Most Significant bit of the result occurred

Note 1: For $\overline{\text{Borrow}}$, the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For rotate (`RRF`, `RLF`) instructions, this bit is loaded with either the high-order or low-order bit of the source register.

REGISTER 5-3: CONFIGURATION WORD 3: WINDOWED WATCHDOG

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
WDTCCS2	WDTCCS1	WDTCCS0	WDTCWS2	WDTCWS1	WDTCWS0
bit 13					bit 8

U-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	R/P-1
—	WDTE1	WDTE0	WDTCPS4	WDTCPS3	WDTCPS2	WDTCPS1	WDTCPS0
bit 7							bit 0

Legend:			
R = Readable bit	P = Programmable bit	x = Bit is unknown	U = Unimplemented bit, read as ‘1’
‘0’ = Bit is cleared	‘1’ = Bit is set	W = Writable bit	n = Value when blank or after Bulk Erase

bit 13-11 **WDTCCS<2:0>**: WDT Input Clock Selector bits
111 = Software Control
110 = Reserved
.
.
.
.
010 = SOSC 32 kHz
001 = WDT reference clock is the 31.0 kHz LFINTOSC
000 = WDT reference clock is the 31.25 kHz HFINTOSC (MFINTOSC) output

bit 10-8 **WDTCWS<2:0>**: WDT Window Select bits

WDTCWS	**WDTWS at POR**			**Software control of WDTWS?**	**Keyed access required?**
	Value	**Window delay Percent of time**	**Window opening Percent of time**		
111	111	n/a	100	Yes	No
110	111	n/a	100	No	Yes
101	101	25	75		
100	100	37.5	62.5		
011	011	50	50		
010	010	62.5	37.5		
001	001	75	25		
000	000	87.5	12.5		

bit 7 **Unimplemented**: Read as ‘1’

bit 6-5 **WDTE<1:0>**: WDT Operating mode:
11 =WDT enabled regardless of Sleep; SWDTEN is ignored
10 =WDT enabled while Sleep = 0, suspended when Sleep = 1; SWDTEN ignored
01 =WDT enabled/disabled by SWDTEN bit in WDTCON0
00 =WDT disabled, SWDTEN is ignored

8.13 Determining the Cause of a Reset

Upon any Reset, multiple bits in the STATUS and PCON registers are updated to indicate the cause of the Reset. Table 8-3 and Table 8-4 show the Reset conditions of these registers.

TABLE 8-3: RESET STATUS BITS AND THEIR SIGNIFICANCE

STOVF	STKUNF	$\overline{\text{RWDT}}$	$\overline{\text{RMCLR}}$	$\overline{\text{RI}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	$\overline{\text{TO}}$	$\overline{\text{PD}}$	$\overline{\text{MEMV}}$	Condition
0	0	1	1	1	0	x	1	1	1	Power-on Reset
0	0	1	1	1	0	x	0	x	u	Illegal, $\overline{\text{TO}}$ is set on $\overline{\text{POR}}$
0	0	1	1	1	0	x	x	0	u	Illegal, $\overline{\text{PD}}$ is set on $\overline{\text{POR}}$
0	0	u	1	1	u	0	1	1	u	Brown-out Reset
u	u	0	u	u	u	u	0	u	u	WWDT Reset
u	u	u	u	u	u	u	0	0	u	WWDT Wake-up from Sleep
u	u	u	u	u	u	u	1	0	u	Interrupt Wake-up from Sleep
u	u	u	0	u	u	u	u	u	1	$\overline{\text{MCLR}}$ Reset during normal operation
u	u	u	0	u	u	u	1	0	u	$\overline{\text{MCLR}}$ Reset during Sleep
u	u	u	u	0	u	u	u	u	u	RESET Instruction Executed
1	u	u	u	u	u	u	u	u	u	Stack Overflow Reset (STVREN = 1)
u	1	u	u	u	u	u	u	u	u	Stack Underflow Reset (STVREN = 1)
u	u	u	u	u	u	u	u	u	0	Memory violation Reset

TABLE 8-4: RESET CONDITION FOR SPECIAL REGISTERS

Condition	Program Counter	STATUS Register	PCON0 Register	PCON1 Register
Power-on Reset	0000h	---1 1000	0011 110x	---- --1-
$\overline{\text{MCLR}}$ Reset during normal operation	0000h	---u uuuu	uuuu 0uuu	---- --1-
$\overline{\text{MCLR}}$ Reset during Sleep	0000h	---1 0uuu	uuuu 0uuu	---- --u-
WWDT Timeout Reset	0000h	---0 uuuu	uuu0 uuuu	---- --u-
WWDT Wake-up from Sleep	PC + 1	---0 0uuu	uuuu uuuu	---- --u-
WWDT Window Violation	0000h	---u uuuu	uu0u uuuu	---- --u-
Brown-out Reset	0000h	---1 1000	0011 11u0	---- --u-
Interrupt Wake-up from Sleep	PC + 1(1)	---1 0uuu	uuuu uuuu	---- --u-
RESET Instruction Executed	0000h	---u uuuu	uuuu u0uu	---- --u-
Stack Overflow Reset (STVREN = 1)	0000h	---u uuuu	1uuu uuuu	---- --u-
Stack Underflow Reset (STVREN = 1)	0000h	---u uuuu	u1uu uuuu	---- --u-
Memory Violation Reset ($\overline{\text{MEMV}}$ = 0)	0	-uuu uuuu	uuuu uuuu	---- --0-

Legend: u = unchanged, x = unknown, - = unimplemented bit, reads as '0'.

Note 1: When the wake-up is due to an interrupt and Global Enable bit (GIE) is set, the return address is pushed on the stack and PC is loaded with the interrupt vector (0004h) after execution of PC + 1.

REGISTER 10-13: PIR3: PERIPHERAL INTERRUPT REQUEST REGISTER 3

R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0
RC2IF	TX2IF	RC1IF	TX1IF	BCL2IF	SSP2IF	BCL1IF	SSP1IF
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 clearable

bit 7 **RC2IF:** EUSART2 Receive Interrupt Flag (Read-Only) bit(1)
1 = The EUSART2 receive buffer is not empty (contains at least one byte)
0 = The EUSART2 receive buffer is empty

bit 6 **TX2IF:** EUSART2 Transmit Interrupt Flag (Read-Only) bit(2)
1 = The EUSART2 transmit buffer contains at least one unoccupied space
0 = The EUSART2 transmit buffer is currently full. The application firmware should not write to TXxREG

bit 5 **RC1IF:** EUSART1 Receive Interrupt Flag (read-only) bit (1)
1 = The EUSART1 receive buffer is not empty (contains at least one byte)
0 = The EUSART1 receive buffer is empty

bit 4 **TX1IF:** EUSART1 Transmit Interrupt Flag (read-only) bit(2)
1 = The EUSART1 transmit buffer contains at least one unoccupied space
0 = The EUSART1 transmit buffer is currently full. The application firmware should not write to TXxREG again, until more room becomes available in the transmit buffer.

bit 3 **BCL2IF:** MSSP2 Bus Collision Interrupt Flag bit
1 = A bus collision was detected (must be cleared in software)
0 = No bus collision was detected

bit 2 **SSP2IF:** Synchronous Serial Port (MSSP2) Interrupt Flag bit
1 = The Transmission/Reception/Bus Condition is complete (must be cleared in software)
0 = Waiting for the Transmission/Reception/Bus Condition in progress

bit 1 **BCL1IF:** MSSP1 Bus Collision Interrupt Flag bit
1 = A bus collision was detected (must be cleared in software)
0 = No bus collision was detected

bit 0 **SSP1IF:** Synchronous Serial Port (MSSP1) Interrupt Flag bit
1 = The Transmission/Reception/Bus Condition is complete (must be cleared in software)
0 = Waiting for the Transmission/Reception/Bus Condition in progress

Note 1: The RCxIF flag is a read-only bit. To clear the RCxIF flag, the firmware must read from RCxREG enough times to remove all bytes from the receive buffer.

2: The TXxIF flag is a read-only bit, indicating if there is room in the transmit buffer. To clear the TX1IF flag, the firmware must write enough data to TXxREG to completely fill all available bytes in the buffer. The TXxIF flag does not indicate transmit completion (use TRMT for this purpose instead).

> **Note:** Interrupt flag bits are set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the Global Enable bit, GIE, of the INTCON register. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

17.5 Register Definitions: Interrupt-on-Change Control

REGISTER 17-1: IOCAP: INTERRUPT-ON-CHANGE PORTA POSITIVE EDGE REGISTER

U-0	U-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
—	—	IOCAP5	IOCAP4	IOCAP3	IOCAP2	IOCAP1(1)	IOCAP0(1)
bit 7							bit 0

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

bit 7-6 **Unimplemented:** read as ‘0’

bit 5-0 **IOCAP<5:0>:** Interrupt-on-Change PORTA Positive Edge Enable bits
1 = Interrupt-on-Change enabled on the pin for a positive-going edge. IOCAFx bit and IOCIF flag will be set upon detecting an edge.
0 = Interrupt-on-Change disabled for the associated pin.

REGISTER 17-2: IOCAN: INTERRUPT-ON-CHANGE PORTA NEGATIVE EDGE REGISTER

U-0	U-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
—	—	IOCAN5	IOCAN4	IOCAN3	IOCAN2	IOCAN1(1)	IOCAN0(1)
bit 7							bit 0

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

bit 7-6 **Unimplemented:** read as ‘0’

bit 5-0 **IOCAN<5:0>:** Interrupt-on-Change PORTA Negative Edge Enable bits
1 = Interrupt-on-Change enabled on the pin for a negative-going edge. IOCAFx bit and IOCIF flag will be set upon detecting an edge.
0 = Interrupt-on-Change disabled for the associated pin.

20.1 ADC Configuration

When configuring and using the ADC the following functions must be considered:

- Port configuration
- Channel selection
- ADC voltage reference selection
- ADC conversion clock source
- Interrupt control
- Result formatting

20.1.1 PORT CONFIGURATION

The ADC can be used to convert both analog and digital signals. When converting analog signals, the I/O pin will be configured for analog by setting the associated TRIS and ANSEL bits. Refer to **Section 14.0 "I/O Ports"** for more information.

Note:	Analog voltages on any pin that is defined as a digital input may cause the input buffer to conduct excess current.

20.1.2 CHANNEL SELECTION

There are several channel selections available:

- Seven Port A channels
- Seven Port B channels
- Seven Port C channels
- Seven Port D channels[(1)]
- Seven Port E channels[(1)]
- Seven Port F channels[(2)]
- Temperature Indicator
- DAC output
- Fixed Voltage Reference (FVR)
- AVSS (Ground)

Note 1:	Present on PIC16(L)F15375/76/85/86 only.
2:	Present on PIC16(L)F15385/86 only.

The CHS<5:0> bits of the ADCON0 register (Register 20-1) determine which channel is connected to the sample and hold circuit.

When changing channels, a delay is required before starting the next conversion. Refer to **Section 20.2 "ADC Operation"** for more information.

20.1.3 ADC VOLTAGE REFERENCE

The ADPREF<1:0> bits of the ADCON1 register provides control of the positive voltage reference. The positive voltage reference can be:

- VREF+ pin
- VDD
- FVR 2.048V
- FVR 4.096V (Not available on LF devices)

The ADPREF bit of the ADCON1 register provides control of the negative voltage reference. The negative voltage reference can be:

- VREF- pin
- VSS

See **Section 18.0 "Fixed Voltage Reference (FVR)"** for more details on the Fixed Voltage Reference.

20.1.4 CONVERSION CLOCK

The source of the conversion clock is software selectable via the ADCS<2:0> bits of the ADCON1 register. There are seven possible clock options:

- FOSC/2
- FOSC/4
- FOSC/8
- FOSC/16
- FOSC/32
- FOSC/64
- ADCRC (dedicated RC oscillator)

The time to complete one bit conversion is defined as TAD. One full 10-bit conversion requires 11.5 TAD periods as shown in Figure 20-2.

For correct conversion, the appropriate TAD specification must be met. Refer to Table 37-13 for more information. Table 20-1 gives examples of appropriate ADC clock selections.

Note:	Unless using the ADCRC, any changes in the system clock frequency will change the ADC clock frequency, which may adversely affect the ADC result.

REGISTER 25-1: T0CON0: TIMER0 CONTROL REGISTER 0

R/W-0/0	U-0	R-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
T0EN	—	T0OUT	T016BIT	T0OUTPS<3:0>			
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 **T0EN:** Timer0 Enable bit
1 = The module is enabled and operating
0 = The module is disabled and in the lowest power mode

bit 6 **Unimplemented:** Read as ‘0’

bit 5 **T0OUT:** Timer0 Output bit (read-only)
Timer0 output bit

bit 4 **T016BIT:** Timer0 Operating as 16-bit Timer Select bit
1 = Timer0 is a 16-bit timer
0 = Timer0 is an 8-bit timer

bit 3-0 **T0OUTPS<3:0>:** Timer0 output postscaler (divider) select bits
1111 = 1:16 Postscaler
1110 = 1:15 Postscaler
1101 = 1:14 Postscaler
1100 = 1:13 Postscaler
1011 = 1:12 Postscaler
1010 = 1:11 Postscaler
1001 = 1:10 Postscaler
1000 = 1:9 Postscaler
0111 = 1:8 Postscaler
0110 = 1:7 Postscaler
0101 = 1:6 Postscaler
0100 = 1:5 Postscaler
0011 = 1:4 Postscaler
0010 = 1:3 Postscaler
0001 = 1:2 Postscaler
0000 = 1:1 Postscaler

REGISTER 26-2: T1GCON: TIMER1 GATE CONTROL REGISTER

R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W/HC-0/u	R-x/x	U-0	U-0
GE	GPOL	GTM	GSPM	GGO/$\overline{\text{DONE}}$	GVAL	—	—
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 = Bit is cleared by hardware

bit 7 **GE:** Timer1 Gate Enable bit
<u>If ON = `0`</u>:
This bit is ignored
<u>If ON = `1`</u>:
`1` = Timer1 counting is controlled by the Timer1 gate function
`0` = Timer1 is always counting

bit 6 **GPOL:** Timer1 Gate Polarity bit
`1` = Timer1 gate is active-high (Timer1 counts when gate is high)
`0` = Timer1 gate is active-low (Timer1 counts when gate is low)

bit 5 **GTM:** Timer1 Gate Toggle Mode bit
`1` = Timer1 Gate Toggle mode is enabled
`0` = Timer1 Gate Toggle mode is disabled and toggle flip-flop is cleared
Timer1 gate flip-flop toggles on every rising edge.

bit 4 **GSPM:** Timer1 Gate Single-Pulse Mode bit
`1` = Timer1 Gate Single-Pulse mode is enabled
`0` = Timer1 Gate Single-Pulse mode is disabled

bit 3 **GGO/$\overline{\text{DONE}}$:** Timer1 Gate Single-Pulse Acquisition Status bit
`1` = Timer1 gate single-pulse acquisition is ready, waiting for an edge
`0` = Timer1 gate single-pulse acquisition has completed or has not been started
This bit is automatically cleared when GSPM is cleared

bit 2 **GVAL:** Timer1 Gate Value Status bit
Indicates the current state of the Timer1 gate that could be provided to TMR1H:TMR1L
Unaffected by Timer1 Gate Enable (GE)

bit 1-0 **Unimplemented:** Read as ‘`0`’

28.1 Capture Mode

Capture mode makes use of the 16-bit Timer1 resource. When an event occurs on the capture source, the 16-bit CCPRxH:CCPRxL register pair captures and stores the 16-bit value of the TMR1H:TMR1L register pair, respectively. An event is defined as one of the following and is configured by the CCPxMODE<3:0> bits of the CCPxCON register:

- Every falling edge
- Every rising edge
- Every 4th rising edge
- Every 16th rising edge

When a capture is made, the Interrupt Request Flag bit CCPxIF of the PIR6 register is set. The interrupt flag must be cleared in software. If another capture occurs before the value in the CCPRxH, CCPRxL register pair is read, the old captured value is overwritten by the new captured value.

Figure 28-1 shows a simplified diagram of the capture operation.

28.1.1 CAPTURE SOURCES

In Capture mode, the CCPx pin should be configured as an input by setting the associated TRIS control bit.

Note:	If the CCPx pin is configured as an output, a write to the port can cause a capture condition.

The capture source is selected by configuring the CCPxCTS<2:0> bits of the CCPxCAP register. The following sources can be selected:

- CCPxPPS input
- C1OUT_sync
- C2OUT_sync
- IOC_interrupt
- LC1_out
- LC2_out
- LC3_out
- LC4_out

FIGURE 28-1: CAPTURE MODE OPERATION BLOCK DIAGRAM

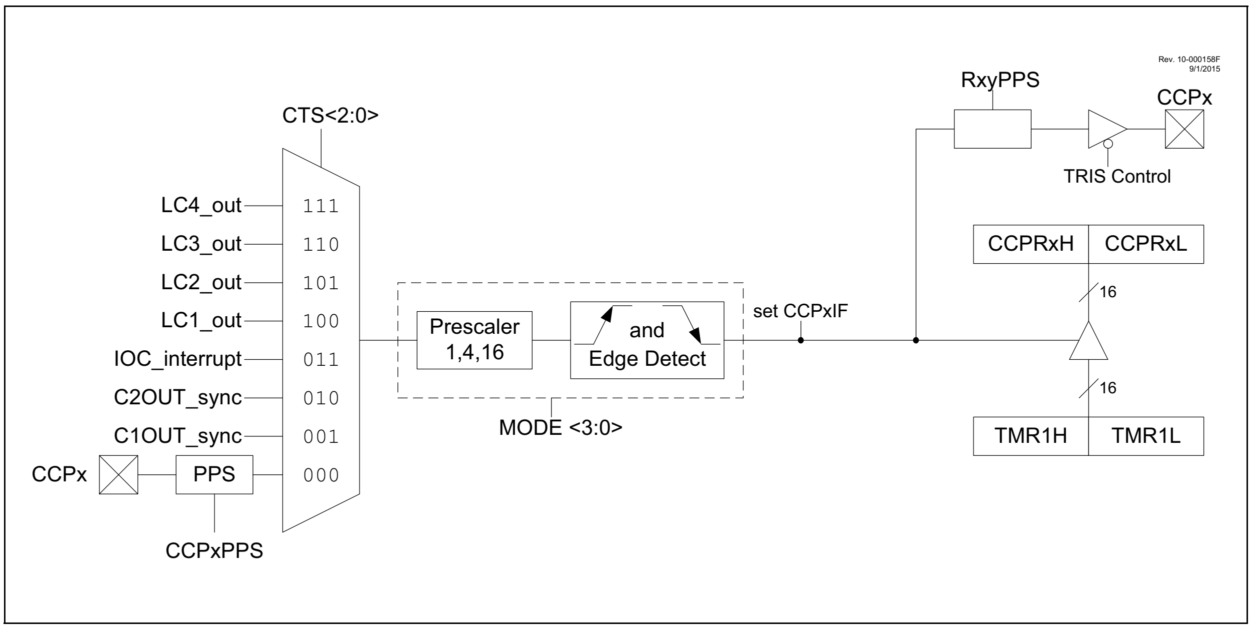


FIGURE 30-3: SIMPLIFIED CWG BLOCK DIAGRAM (FORWARD AND REVERSE FULL-BRIDGE MODES)

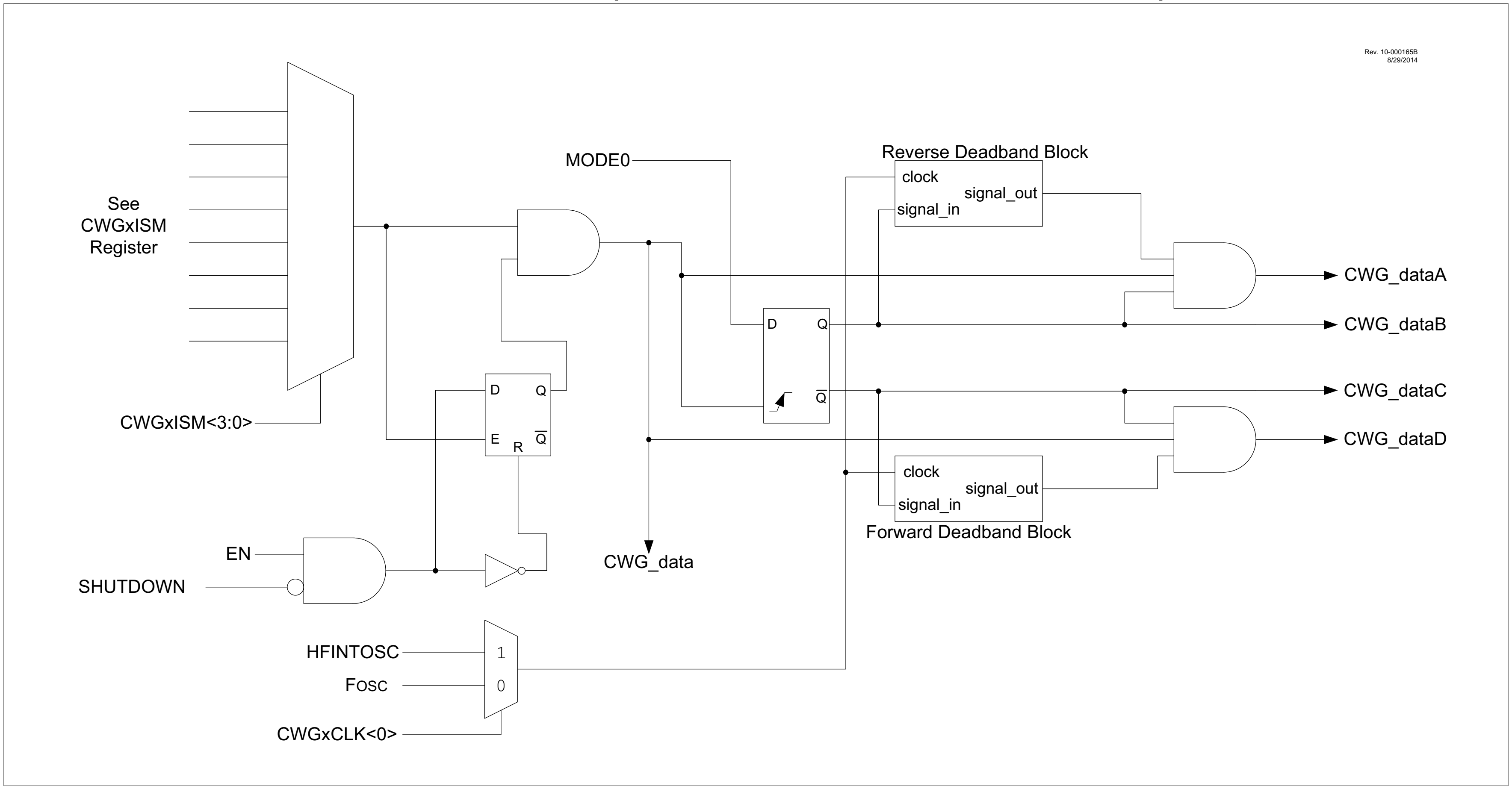
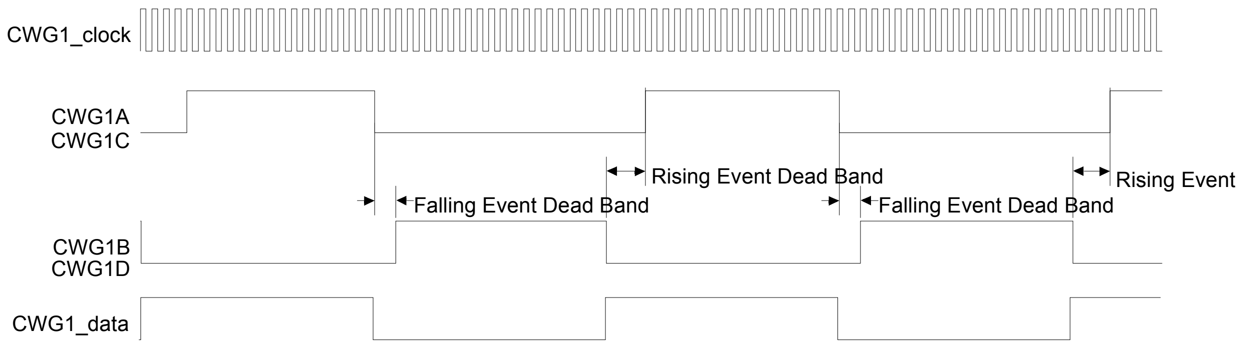


FIGURE 30-9: CWG HALF-BRIDGE MODE OPERATION



Note: CWG1_rising_src = CCP1_out, CWG1_falling_src = ~CCP1_out

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

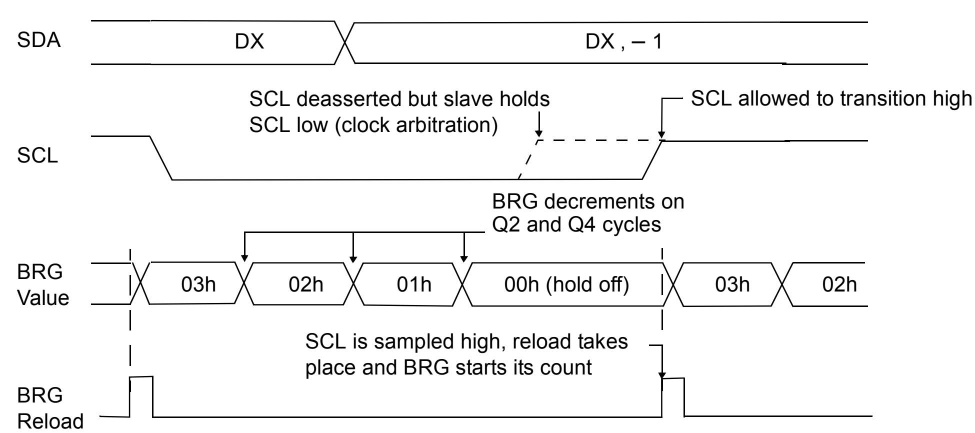
bit 1 **POLB:** CWG1B Output Polarity bit
1 = Signal output is inverted polarity
0 = Signal output is normal polarity

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

32.6.2 CLOCK ARBITRATION

Clock arbitration occurs when the master, during any receive, transmit or Repeated Start/Stop condition, releases the SCL pin (SCL allowed to float high). When the SCL pin is allowed to float high, the Baud Rate Generator (BRG) is suspended from counting until the SCL pin is actually sampled high. When the SCL pin is sampled high, the Baud Rate Generator is reloaded with the contents of SSPxADD<7:0> and begins counting. This ensures that the SCL high time will always be at least one BRG rollover count in the event that the clock is held low by an external device (Figure 32-25).

FIGURE 32-25: BAUD RATE GENERATOR TIMING WITH CLOCK ARBITRATION



32.6.3 WCOL STATUS FLAG

If the user writes the SSPxBUF when a Start, Restart, Stop, Receive or Transmit sequence is in progress, the WCOL is set and the contents of the buffer are unchanged (the write does not occur). Any time the WCOL bit is set it indicates that an action on SSPxBUF was attempted while the module was not idle.

Note:	Because queuing of events is not allowed, writing to the lower five bits of SSPxCON2 is disabled until the Start condition is complete.

33.1.2.3 Receive Interrupts

The RXxIF interrupt flag bit of the PIR3 register is set whenever the EUSART receiver is enabled and there is an unread character in the receive FIFO. The RXxIF interrupt flag bit is read-only, it cannot be set or cleared by software.

RXxIF interrupts are enabled by setting all of the following bits:

- RXxIE, Interrupt Enable bit of the PIE3 register
- PEIE, Peripheral Interrupt Enable bit of the INTCON register
- GIE, Global Interrupt Enable bit of the INTCON register

The RXxIF interrupt flag bit will be set when there is an unread character in the FIFO, regardless of the state of interrupt enable bits.

33.1.2.4 Receive Framing Error

Each character in the receive FIFO buffer has a corresponding framing error Status bit. A framing error indicates that a Stop bit was not seen at the expected time. The framing error status is accessed via the FERR bit of the RCxSTA register. The FERR bit represents the status of the top unread character in the receive FIFO. Therefore, the FERR bit must be read before reading the RCxREG.

The FERR bit is read-only and only applies to the top unread character in the receive FIFO. A framing error (FERR = `1`) does not preclude reception of additional characters. It is not necessary to clear the FERR bit. Reading the next character from the FIFO buffer will advance the FIFO to the next character and the next corresponding framing error.

The FERR bit can be forced clear by clearing the SPEN bit of the RCxSTA register which resets the EUSART. Clearing the CREN bit of the RCxSTA register does not affect the FERR bit. A framing error by itself does not generate an interrupt.

Note:	If all receive characters in the receive FIFO have framing errors, repeated reads of the RCxREG will not clear the FERR bit.

33.1.2.5 Receive Overrun Error

The receive FIFO buffer can hold two characters. An overrun error will be generated if a third character, in its entirety, is received before the FIFO is accessed. When this happens the OERR bit of the RCxSTA register is set. The characters already in the FIFO buffer can be read but no additional characters will be received until the error is cleared. The error must be cleared by either clearing the CREN bit of the RCxSTA register or by resetting the EUSART by clearing the SPEN bit of the RCxSTA register.

33.1.2.6 Receiving 9-Bit Characters

The EUSART supports 9-bit character reception. When the RX9 bit of the RCxSTA register is set the EUSART will shift nine bits into the RSR for each character received. The RX9D bit of the RCxSTA register is the ninth and Most Significant data bit of the top unread character in the receive FIFO. When reading 9-bit data from the receive FIFO buffer, the RX9D data bit must be read before reading the eight Least Significant bits from the RCxREG.

33.1.2.7 Address Detection

A special Address Detection mode is available for use when multiple receivers share the same transmission line, such as in RS-485 systems. Address detection is enabled by setting the ADDEN bit of the RCxSTA register.

Address detection requires 9-bit character reception. When address detection is enabled, only characters with the ninth data bit set will be transferred to the receive FIFO buffer, thereby setting the RXxIF interrupt bit. All other characters will be ignored.

Upon receiving an address character, user software determines if the address matches its own. Upon address match, user software must disable address detection by clearing the ADDEN bit before the next Stop bit occurs. When user software detects the end of the message, determined by the message protocol used, software places the receiver back into the Address Detection mode by setting the ADDEN bit.

FIGURE 37-17: SPI MASTER MODE TIMING (CKE = 0, SMP = 0)

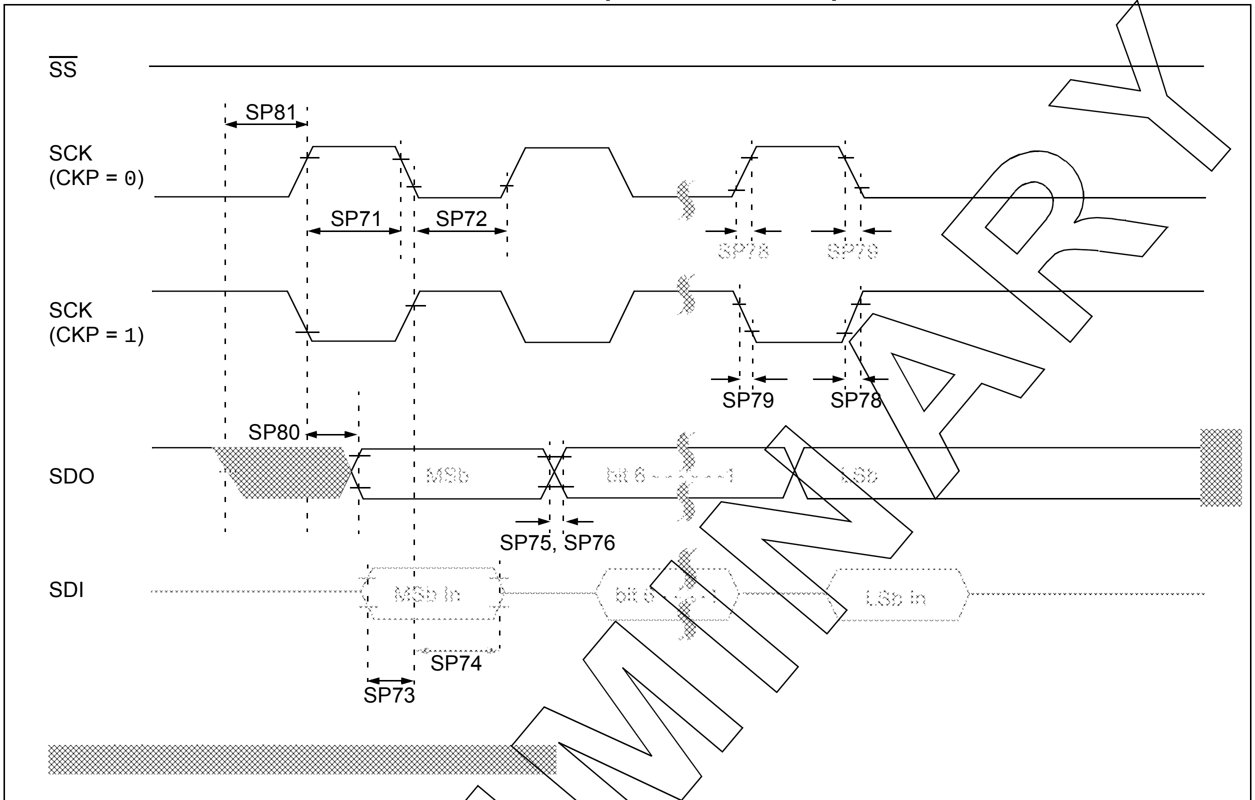
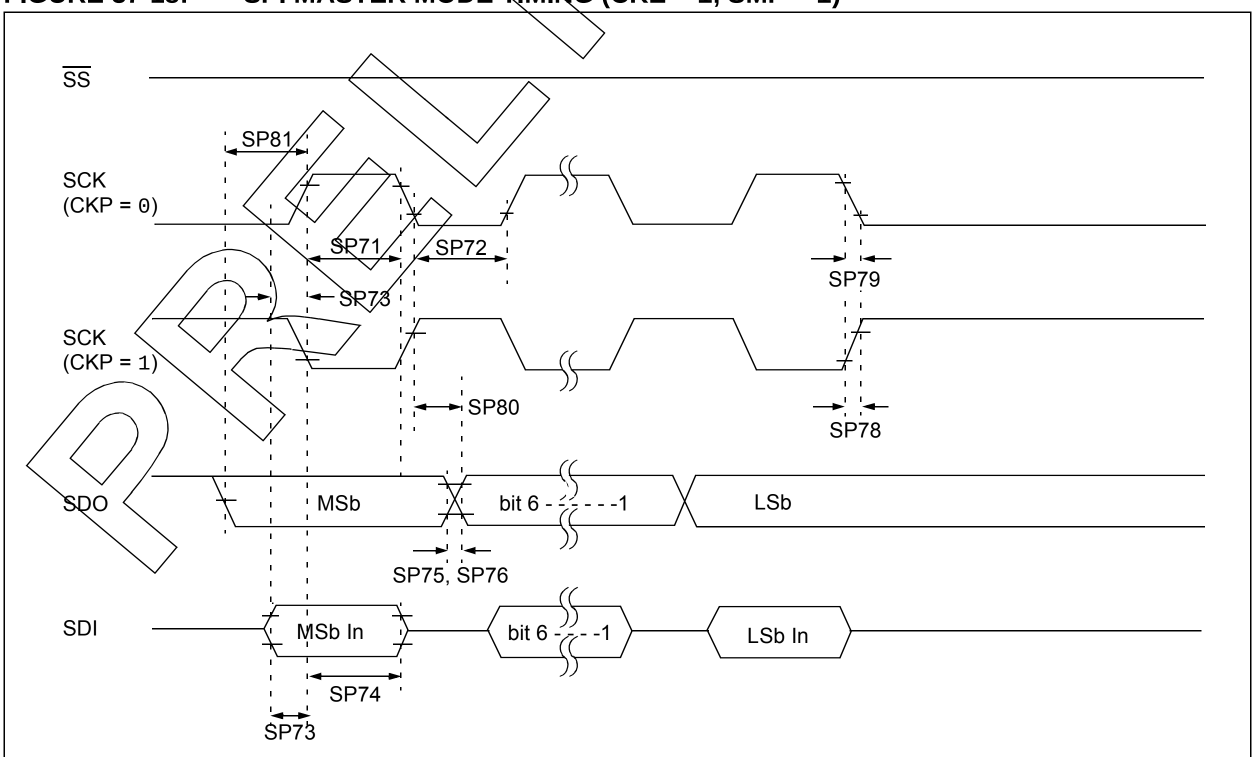


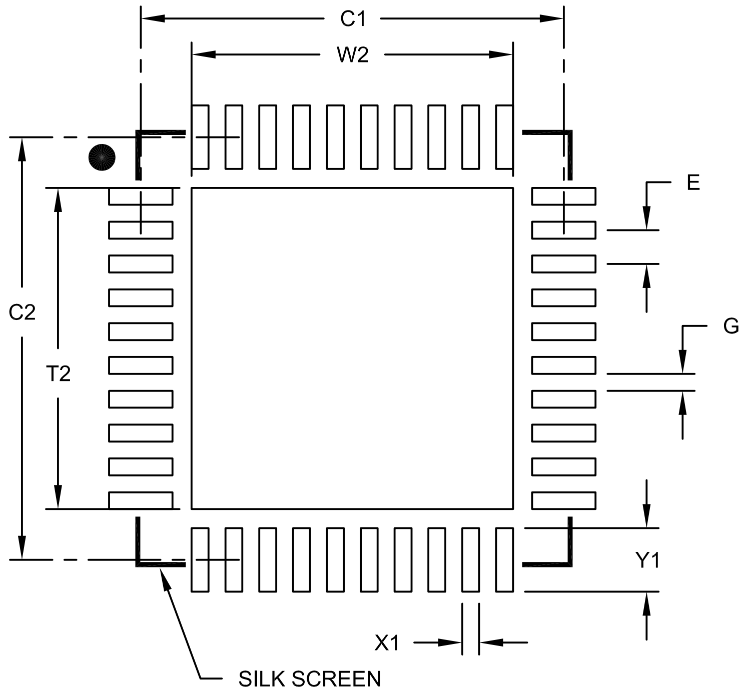
FIGURE 37-18: SPI MASTER MODE TIMING (CKE = 1, SMP = 1)



Note: Refer to Figure 37-4 for load conditions.

40-Lead Plastic Ultra Thin Quad Flat, No Lead Package (MV) - 5x5 mm Body [UQFN]

Note:	For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.40 BSC	
Optional Center Pad Width	W2			3.80
Optional Center Pad Length	T2			3.80
Contact Pad Spacing	C1		5.00	
Contact Pad Spacing	C2		5.00	
Contact Pad Width (X40)	X1			0.20
Contact Pad Length (X40)	Y1			0.75
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2156B