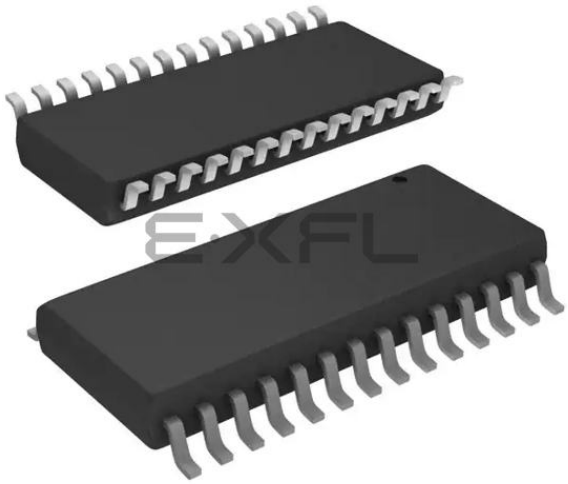


Welcome to E-XFL.COM

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	14KB (8K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	1K 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-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf18855-e-so

TABLE 3: 40/44-PIN ALLOCATION TABLE (PIC16F18875) (CONTINUED)

I/O	40-Pin PDIP	40-Pin UQFN	44-Pin QFN	44-Pin TQFP	ADC	Voltage Reference	DAC	Comparators	Zero-Cross Detect	MSSP (SPI/I2C)	EUSART	DSM	Timers/SMT	CCP and PWM	CWG	CLC	NCO	Clock Reference (CLKR)	Interrupt-on-Change	Basic
RC0	15	30	34	32	ANC0	—	—	—	—	—	—	—	T1CKI(1) T3CKI(1) T3G(1) SMTWIN1(1)	—	—	—	—	—	IOCC0	SOSCO
RC1	16	31	35	35	ANC1	—	—	—	—	—	—	—	SMTSIG1(1)	CCP2(1)	—	—	—	—	IOCC1	SOSCI
RC2	17	32	36	36	ANC2	—	—	—	—	—	—	—	T5CKI(1)	CCP1(1)	—	—	—	—	IOCC2	—
RC3	18	33	37	37	ANC3	—	—	—	—	SCL1(3,4) SCK1(1)	—	—	T2IN(1)	—	—	—	—	—	IOCC3	—
RC4	23	38	42	42	ANC4	—	—	—	—	SDA1(3,4) SDI1(1)	—	—	—	—	—	—	—	—	IOCC4	—
RC5	24	39	43	43	ANC5	—	—	—	—	—	—	—	T4IN(1)	—	—	—	—	—	IOCC5	—
RC6	25	40	44	44	ANC6	—	—	—	—	—	CK(3)	—	—	—	—	—	—	—	IOCC6	—
RC7	26	1	1	1	ANC7	—	—	—	—	—	RX(1) DT(3)	—	—	—	—	—	—	—	IOCC7	—
RD0	19	34	38	38	AND0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RD1	20	35	39	39	AND1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RD2	21	36	40	40	AND2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RD3	22	37	41	41	AND3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RD4	27	2	2	2	AND4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RD5	28	3	3	3	AND5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RD6	29	4	4	4	AND6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RD7	30	5	5	5	AND7	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RE0	8	23	25	25	ANE0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RE1	9	24	26	26	ANE1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RE2	10	25	27	27	ANE2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

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 13-1 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 13-3.

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.

2.1 Automatic Interrupt Context Saving

During interrupts, certain registers are automatically saved in shadow registers and restored when returning from the interrupt. This saves stack space and user code. See **Section 7.5 "Automatic Context Saving"** for more information.

2.2 16-Level Stack with Overflow and Underflow

These devices have a hardware stack memory 15 bits wide and 16 words deep. A Stack Overflow or Underflow will set the appropriate bit (STKOVF or STKUNF) in the PCON register, and if enabled, will cause a software Reset. See **Section 3.4 "Stack"** for more details.

2.3 File Select Registers

There are two 16-bit File Select Registers (FSR). FSRs can access all file registers and program memory, which allows one Data Pointer for all memory. When an FSR points to program memory, there is one additional instruction cycle in instructions using INDF to allow the data to be fetched. General purpose memory can now also be addressed linearly, providing the ability to access contiguous data larger than 80 bytes. There are also new instructions to support the FSRs. See **Section 3.5 "Indirect Addressing"** for more details.

2.4 Instruction Set

There are 49 instructions for the enhanced mid-range CPU to support the features of the CPU. See **Section 36.0 "Instruction Set Summary"** for more details.

TABLE 3-13: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-31 (CONTINUED)

Address	Name	PIC16(L)F18855	PIC16(L)F18875	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
Bank 31													
CPU CORE REGISTERS; see Table 3-2 for specifics													
F8Ch — FE3h		—		Unimplemented								—	—
FE4h	STATUS_SHAD			—	—	—	—	—	Z_SHAD	DC_SHAD	C_SHAD	---- -xxx	---- -uuu
FE5h	WREG_SHAD			WREG_SHAD								xxxx xxxx	uuuu uuuu
FE6h	BSR_SHAD			—	—	—	BSR_SHAD					---x xxxx	---u uuuu
FE7h	PCLATH_SHAD			—	PCLATH_SHAD							-xxx xxxx	-uuu uuuu
FE8h	FSR0L_SHAD			FSR0L_SHAD								xxxx xxxx	uuuu uuuu
FE9h	FSR0H_SHAD			FSR0H_SHAD								xxxx xxxx	uuuu uuuu
FEAh	FSR1L_SHAD			FSR1L_SHAD								xxxx xxxx	uuuu uuuu
FEBh	FSR1H_SHAD			FSR1H_SHAD								xxxx xxxx	uuuu uuuu
FECh	__	—		Unimplemented								—	
FEDh	STKPTR			—	—	—	STKPTR<4;0>					---1 1111	---1 1111
FEEh	TOSL			TOSL<7:0>								xxxx xxxx	xxxx xxxx
FEFh	TOSH			—	TOSH<6:0>							-xxx xxxx	-xxx xxxx

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

Note 1: Register present on PIC16F18855/75 devices only.

2: Unimplemented, read as '1'.

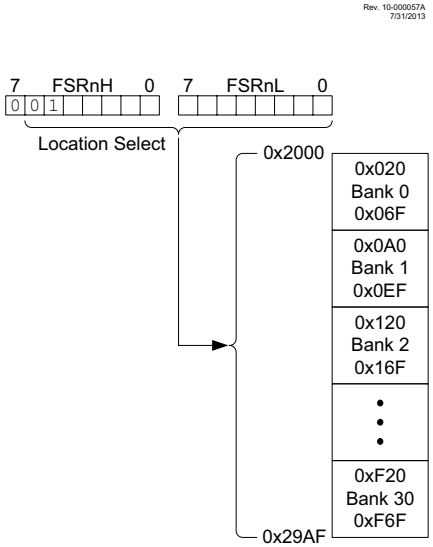
3.5.2 LINEAR DATA MEMORY

The linear data memory is the region from FSR address 0x2000 to FSR address 0x29AF. This region is a virtual region that points back to the 80-byte blocks of GPR memory in all the banks.

Unimplemented memory reads as 0x00. Use of the linear data memory region allows buffers to be larger than 80 bytes because incrementing the FSR beyond one bank will go directly to the GPR memory of the next bank.

The 16 bytes of common memory are not included in the linear data memory region.

FIGURE 3-10: TRADITIONAL DATA MEMORY MAP



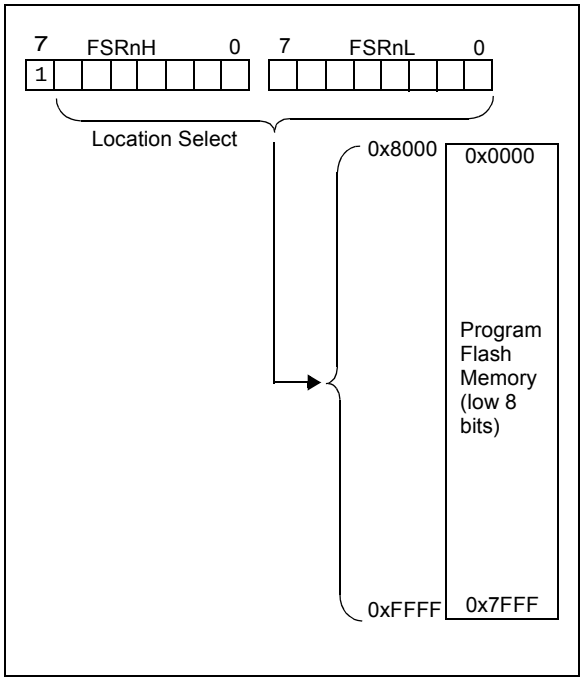
3.5.3 DATA EEPROM MEMORY

The EEPROM memory can be read or written through the NVMCONx/NVMADRx/NVMDATx register interface (see section **Section 10.2 "Data EEPROM Memory"**). However, to make access to the EEPROM memory easier, read-only access to the EEPROM contents are also available through indirect addressing by an FSR. When the MSB of the FSR (ex: FSRxH) is set to 0x70, the lower 8-bit address value (in FSRxL) determines the EEPROM location that may be read from (through the INDF register). In other words, the EEPROM address range 0x00-0xFF is mapped into the FSR address space between 0x7000-0x70FF. Writing to the EEPROM cannot be accomplished via the FSR/INDF interface. Reads from the EEPROM through the FSR/INDF interface will require one additional instruction cycle to complete.

3.5.4 PROGRAM FLASH MEMORY

To make constant data access easier, the entire Program Flash Memory is mapped to the upper half of the FSR address space. When the MSB of FSRnH is set, the lower 15 bits are the address in program memory which will be accessed through INDF. Only the lower eight bits of each memory location is accessible via INDF. Writing to the Program Flash Memory cannot be accomplished via the FSR/INDF interface. All instructions that access Program Flash Memory via the FSR/INDF interface will require one additional instruction cycle to complete.

FIGURE 3-11: PROGRAM FLASH MEMORY MAP



EXAMPLE 10-3: ERASING ONE ROW OF PROGRAM FLASH MEMORY (PFM)

```
; This sample row erase routine assumes the following:
; 1.A valid address within the erase row is loaded in variables ADDRH:ADDRL
; 2.ADDRH and ADDRL are located in common RAM (locations 0x70 - 0x7F)

BANKSEL         NVMADRL
MOVF            ADDRL,W
MOVWF           NVMADRL             ; Load lower 8 bits of erase address boundary
MOVF            ADDRH,W
MOVWF           NVMADRH             ; Load upper 6 bits of erase address boundary
BCF             NVMCON1,NVMREGS     ; Choose PFM memory area
BSF             NVMCON1,FREE        ; Specify an erase operation
BSF             NVMCON1,WREN        ; Enable writes
BCF             INTCON,GIE          ; Disable interrupts during unlock sequence

; ------------------------------REQUIRED UNLOCK SEQUENCE:-----------------------------

MOVLW           55h                 ; Load 55h to get ready for unlock sequence
MOVWF           NVMCON2             ; First step is to load 55h into NVMCON2
MOVLW           AAh                 ; Second step is to load AAh into W
MOVWF           NVMCON2             ; Third step is to load AAh into NVMCON2
BSF             NVMCON1,WR          ; Final step is to set WR bit

; ------------------------------------------------------------------------------------

BSF             INTCON,GIE          ; Re-enable interrupts, erase is complete
BCF             NVMCON1,WREN        ; Disable writes
```

TABLE 10-2: NVM ORGANIZATION AND ACCESS INFORMATION

Master Values			NVMREG Access			FSR Access	
Memory Function	**ICSP™ Address**	**Memory Type**	**NVMREGS bit (NVMCON1)**	**NVMADR <15:0>**	**Allowed Operations**	**FSR Address**	**FSR Programming Address**
Reset Vector	0000h	PFM	0	8000h	Read Write	8000h	Read-Only
User Memory	0001h		0	8001h		8001h	
	0003h			8003h		8003h	
INT Vector	0004h		0	8004h		8004h	
User Memory	0005h		0	8005h		8005h	
	07FFh			87FFh		87FFh	
User ID	8000h	PFM	1	8000h	Read Write	No Access	
	8003h			8003h			
Reserved	8004h	—	–	8004h	—		
Rev ID	8005h	PFM	1	8005h	Read Write		
Device ID	8006h		1	8006h			
CONFIG1	8007h		1	8007h	Read-Only		
CONFIG2	8008h		1	8008h			
CONFIG3	8009h		1	8009h			
CONFIG4	800Ah		1	800Ah			
CONFIG5	800Bh		1	800Bh			
User Memory	F000h	EEPROM	1	F000h	Read Write	7000h	Read-Only
	F0FFh			F0FFh		70FFh	

11.0 CYCLIC REDUNDANCY CHECK (CRC) MODULE

The Cyclic Redundancy Check (CRC) module provides a software-configurable hardware-implemented CRC checksum generator. This module includes the following features:

- Any standard CRC up to 16 bits can be used
- Configurable Polynomial
- Any seed value up to 16 bits can be used
- Standard and reversed bit order available
- Augmented zeros can be added automatically or by the user
- Memory scanner for fast CRC calculations on program memory user data
- Software loadable data registers for calculating CRC values not from the memory scanner

11.1 CRC Module Overview

The CRC module provides a means for calculating a check value of program memory. The CRC module is coupled with a memory scanner for faster CRC calculations. The memory scanner can automatically provide data to the CRC module. The CRC module can also be operated by directly writing data to SFRs, without using the scanner.

11.2 CRC Functional Overview

The CRC module can be used to detect bit errors in the Flash memory using the built-in memory scanner or through user input RAM memory. The CRC module can accept up to a 16-bit polynomial with up to a 16-bit seed value. A CRC calculated check value (or checksum) will then be generated into the CRCACC<15:0> registers for user storage. The CRC module uses an XOR shift register implementation to perform the polynomial division required for the CRC calculation.

EXAMPLE 11-1: BASIC CRC OPERATION EXAMPLE

CRC-16-ANSI

$x^{16} + x^{15} + x^2 + 1$ (17 bits)

Standard 16-bit representation = 0x8005

```
CRCXORH = 0b10000000
CRCXORL = 0b0000010- (1)
```

Data Sequence:
0x55, 0x66, 0x77, 0x88

```
DLEN = 0b0111
PLEN = 0b1111
```

Data entered into the CRC:

SHIFTM = 0:

```
01010101 01100110 01110111 10001000
```

SHIFTM = 1:

```
10101010 01100110 11101110 00010001
```

Check Value (ACCM = 1):

SHIFTM = 0: 0x32D6

```
CRCACCH = 0b00110010
CRCACCL = 0b11010110
```

SHIFTM = 1: 0x6BA2

```
CRCACCH = 0b01101011
CRCACCL = 0b10100010
```

Note 1: Bit 0 is unimplemented. The LSb of any CRC polynomial is always '1' and will always be treated as a '1' by the CRC for calculating the CRC check value. This bit will be read in software as a '0'.

11.3 CRC Polynomial Implementation

Any standard polynomial up to 17 bits can be used. The PLEN<3:0> bits are used to specify how long the polynomial used will be. For an x^n polynomial, PLEN = n-2. In an n-bit polynomial the x^n bit and the LSb will be used as a '1' in the CRC calculation because the MSb and LSb must always be a '1' for a CRC polynomial. For example, if using CRC-16-ANSI, the polynomial will look like 0x8005. This will be implemented into the CRCXOR<15:1> registers, as shown in Example 11-1.

REGISTER 11-11: SCANCON0: SCANNER ACCESS CONTROL REGISTER 0

R/W-0/0	R/W/HC-0/0	R-0	R-0	R/W-0/0	U-0	R/W-0/0	R/W-0/0
EN(1)	SCANGO(2, 3)	BUSY(4)	INVALID	INTM	—	MODE<1:0>(5)	
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 **EN:** Scanner Enable bit(1)
- 1 = Scanner is enabled
- 0 = Scanner is disabled, internal states are reset

bit 6 **SCANGO:** Scanner GO bit(2, 3)
- 1 = When the CRC sends a ready signal, NVM will be accessed according to MDx and data passed to the client peripheral.
- 0 = Scanner operations will not occur

bit 5 **BUSY:** Scanner Busy Indicator bit(4)
- 1 = Scanner cycle is in process
- 0 = Scanner cycle is complete (or never started)

bit 4 **INVALID:** Scanner Abort signal bit
- 1 = SCANLADRL/H has incremented or contains an invalid address(6)
- 0 = SCANLADRL/H points to a valid address

bit 3 **INTM:** NVM Scanner Interrupt Management Mode Select bit

If MODE = 10:

This bit is ignored

If MODE = 01 (CPU is stalled until all data is transferred):
- 1 = SCANGO is overridden (to zero) during interrupt operation; scanner resumes after returning from interrupt
- 0 = SCANGO is not affected by interrupts, the interrupt response will be affected

If MODE = 00 or 11:
- 1 = SCANGO is overridden (to zero) during interrupt operation; scan operations resume after returning from interrupt
- 0 = Interrupts do not prevent NVM access

bit 2 **Unimplemented:** Read as '0'

bit 1-0 **MODE<1:0>:** Memory Access Mode bits(5)
- 11 = Triggered mode
- 10 = Peek mode
- 01 = Burst mode
- 00 = Concurrent mode

Note 1: Setting EN = 0 (SCANCON0 register) does not affect any other register content.

2: This bit is cleared when LADR > HADR (and a data cycle is not occurring).

3: If INTM = 1, this bit is overridden (to zero, but not cleared) during an interrupt response.

4: BUSY = 1 when the NVM is being accessed, or when the CRC sends a ready signal.

5: See Table 11-1 for more detailed information.

6: An invalid address happens when the entire range of the PFM is scanned and completed, i.e., device memory is 0x4000 and SCANHADR = 0x3FFF, after the last scan SCANLADR increments to 0x4000, the address is invalid.

REGISTER 12-4: LATA: PORTA DATA LATCH 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
LATA7	LATA6	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0
bit 7							bit 0

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

bit 7-0 **LATA<7:0>**: RA<7:0> Output Latch Value bits(1)

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

REGISTER 12-5: ANSELA: PORTA ANALOG SELECT REGISTER

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
ANSA7	ANSA6	ANSA5	ANSA4	ANSA3	ANSA2	ANSA1	ANSA0
bit 7							bit 0

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

bit 7-0 **ANSA<7:0>**: Analog Select between Analog or Digital Function on pins RA<7:0>, 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.

REGISTER 12-30: CCDPC: CURRENT CONTROLLED DRIVE POSITIVE PORTC REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
CCDPC7	CCDPC6	CCDPC5	CCDPC4	CCDPC3	CCDPC2	CCDPC1	CCDPC0
bit 7							bit 0

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

bit 7-0 **CCDPC<7:0>:** RC<7:0> Current Controlled Drive Positive Control bits(1)
- 1 = Current-controlled source enabled
- 0 = Current-controlled source disabled

Note 1: If CCDPCy is set, when CCDEN = 0 (Register 12-1), operation of the pin is undefined.

REGISTER 12-31: CCDNC: CURRENT CONTROLLED DRIVE NEGATIVE PORTC REGISTER

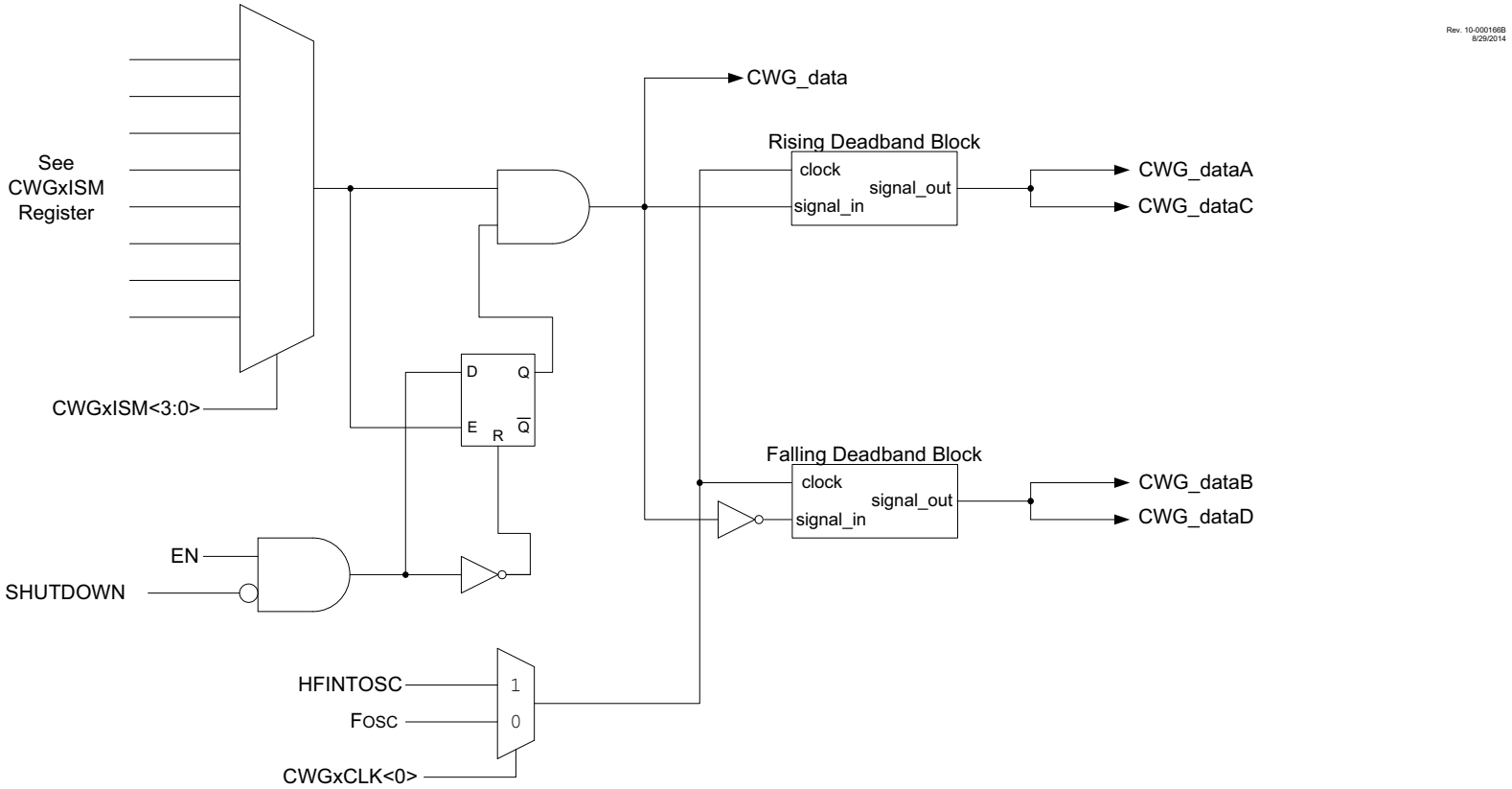
R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
CCDNC7	CCDNC6	CCDNC5	CCDNC4	CCDNC3	CCDNC2	CCDNC1	CCDNC0
bit 7							bit 0

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

bit 7-0 **CCDNC<7:0>:** RC<7:0> Current Controlled Drive Negative Control bits(1)
- 1 = Current-controlled source enabled
- 0 = Current-controlled source disabled

Note 1: If CCDNCy is set, when CCDEN = 0 (Register 12-1), operation of the pin is undefined.

FIGURE 20-1: SIMPLIFIED CWG BLOCK DIAGRAM (HALF-BRIDGE MODE)



REGISTER 22-9: 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

23.2 ADC Operation

23.2.1 STARTING A CONVERSION

To enable the ADC module, the ADON bit of the ADCON0 register must be set to a '1'. A conversion may be started by any of the following:

- Software setting the ADGO bit of ADCON0 to '1'
- An external trigger (selected by Register 23-3)
- A continuous-mode retrigger (see section **Section 23.5.8 "Continuous Sampling Mode"**)

.

Note:	The ADGO bit should not be set in the same instruction that turns on the ADC. Refer to **Section 23.2.7 "ADC Conversion Procedure (Basic Mode)"**.

23.2.2 COMPLETION OF A CONVERSION

When any individual conversion is complete, the value already in ADRES is written into ADPREV (if ADPSIS=1) and the new conversion results appear in ADRES. When the conversion completes, the ADC module will:

- Clear the ADGO bit (Unless the ADCONT bit of ADCON0 is set)
- Set the ADIF Interrupt Flag bit
- Set the ADMATH bit
- Update ADACC

When ADDSEN=0 then after every conversion, or when ADDSEN=1 then after every other conversion, the following events occur:

- ADERR is calculated
- ADTIF is set if ADERR calculation meets threshold requirements

In addition, on the completion of every conversion if ADDSEN=0, or every other conversion if ADDSEN=1:

- ADSTPE is calculated
- Depending on ADSTPE, the threshold comparison may set ADTIF

Importantly, filter and threshold computations occur after the conversion itself is complete. As such, interrupt handlers responding to ADIF should check ADTIF before reading filter and threshold results.

23.2.3 TERMINATING A CONVERSION

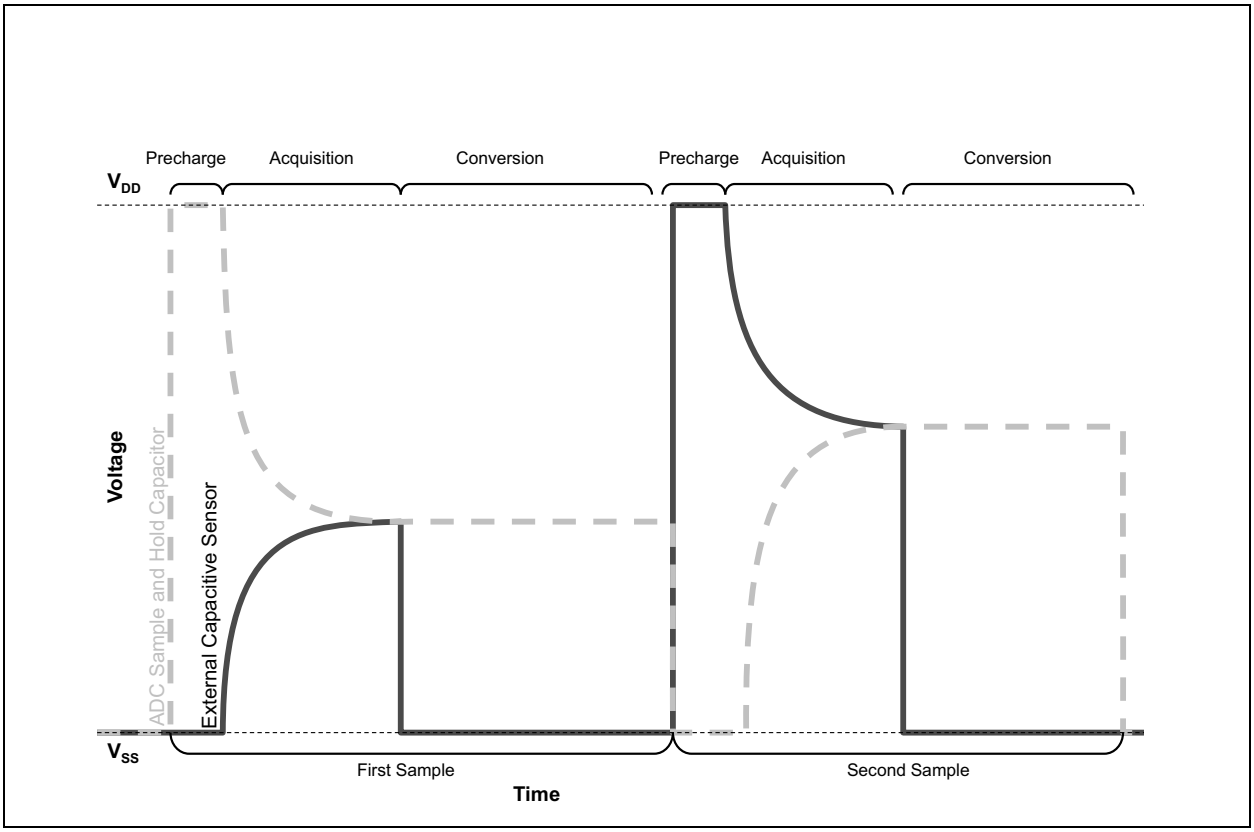
If a conversion must be terminated before completion, the ADGO bit can be cleared in software. The ADRESH and ADRESL registers will be updated with the partially complete Analog-to-Digital conversion sample. Incomplete bits will match the last bit converted. In this case, filter and/or threshold occur.

Note:	A device Reset forces all registers to their Reset state. Thus, the ADC module is turned off and any pending conversion is terminated.

23.4.1 CVD OPERATION

A CVD operation begins with the ADC's internal sample and hold capacitor (C_{HOLD}) being disconnected from the path which connects it to the external capacitive sensor node. While disconnected, C_{HOLD} is precharged to V_{DD} or V_{SS}, while the path to the sensor node is also discharged to V_{DD} or V_{SS}. Typically, this node is discharged to the level opposite that of C_{HOLD}. When the precharge phase is complete, the V_{DD}/V_{SS} bias paths for the two nodes are shut off and C_{HOLD} and the path to the external sensor node are reconnected, at which time the acquisition phase of the CVD operation begins. During acquisition, a capacitive voltage divider is formed between the precharged C_{HOLD} and sensor nodes, which results in a final voltage level setting on C_{HOLD}, which is determined by the capacitances and precharge levels of the two nodes. After acquisition, the ADC converts the voltage level on C_{HOLD}. This process is then repeated with the selected precharge levels for both the C_{HOLD} and the inverted sensor nodes. Figure 23-7 shows the waveform for two inverted CVD measurements, which is known as differential CVD measurement.

FIGURE 23-7: DIFFERENTIAL CVD MEASUREMENT WAVEFORM



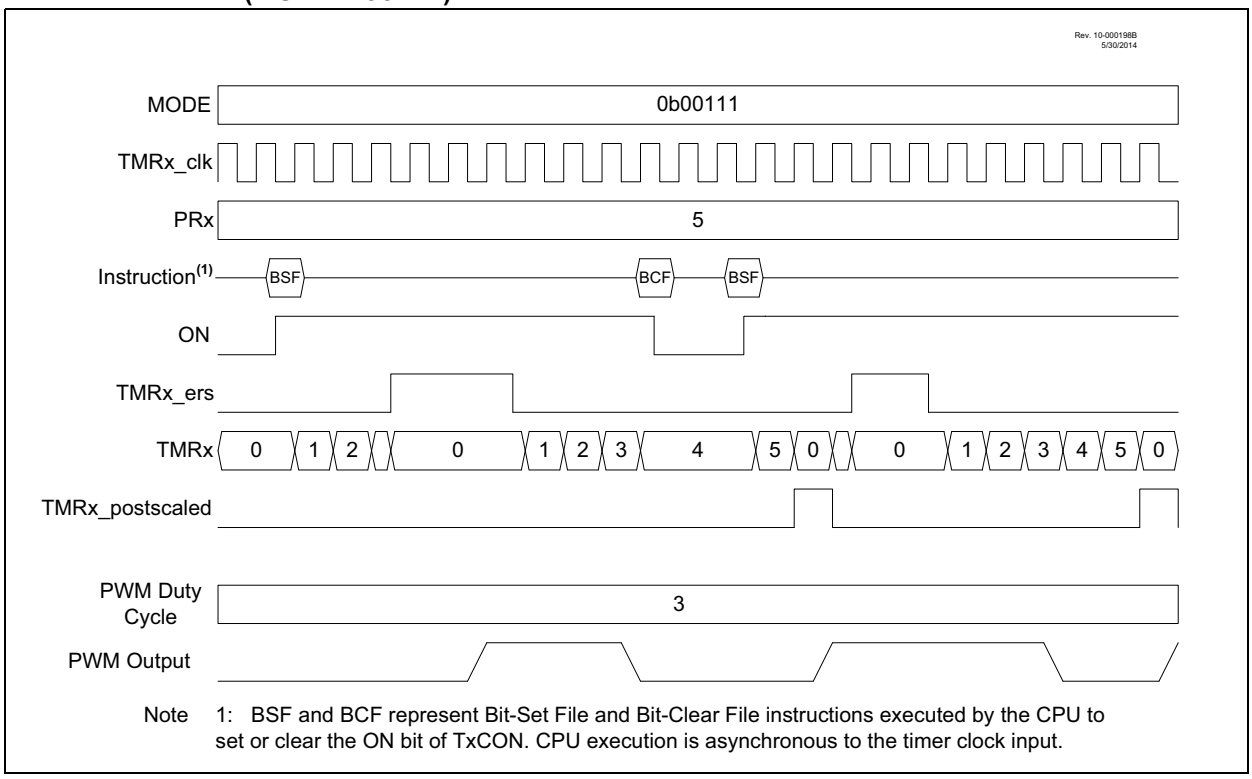
29.5.4 LEVEL-TRIGGERED HARDWARE LIMIT MODE

In the Level-Triggered Hardware Limit Timer modes the counter is reset by high or low levels of the external signal TMRx_ers, as shown in Figure 29-7. Selecting MODE<4:0> = `00110` will cause the timer to reset on a low level external signal. Selecting MODE<4:0> = `00111` will cause the timer to reset on a high level external signal. In the example, the counter is reset while TMRx_ers = `1`. ON is controlled by BSF and BCF instructions. When ON = `0` the external signal is ignored.

When the CCP uses the timer as the PWM time base then the PWM output will be set high when the timer starts counting and then set low only when the timer count matches the CCPRx value. The timer is reset when either the timer count matches the PRx value or two clock periods after the external Reset signal goes true and stays true.

The timer starts counting, and the PWM output is set high, on either the clock following the PRx match or two clocks after the external Reset signal relinquishes the Reset. The PWM output will remain high until the timer counts up to match the CCPRx pulse width value. If the external Reset signal goes true while the PWM output is high then the PWM output will remain high until the Reset signal is released allowing the timer to count up to match the CCPRx value.

FIGURE 29-7: LEVEL-TRIGGERED HARDWARE LIMIT MODE TIMING DIAGRAM (MODE = `00111`)



Note 1: BSF and BCF represent Bit-Set File and Bit-Clear File instructions executed by the CPU to set or clear the ON bit of TxCON. CPU execution is asynchronous to the timer clock input.

31.0 MASTER SYNCHRONOUS SERIAL PORT (MSSP) MODULES

31.1 MSSP Module Overview

The Master Synchronous Serial Port (MSSP) module is a serial interface useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be serial EEPROMs, shift registers, display drivers, A/D converters, etc. The MSSP module can operate in one of two modes:

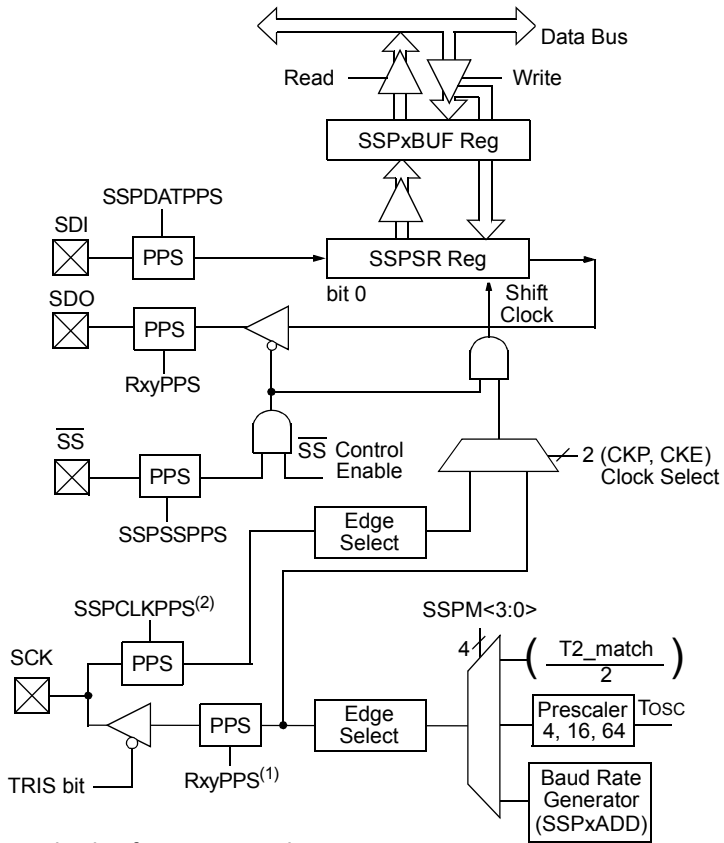
- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I^2C)

The SPI interface supports the following modes and features:

- Master mode
- Slave mode
- Clock Parity
- Slave Select Synchronization (Slave mode only)
- Daisy-chain connection of slave devices

Figure 31-1 is a block diagram of the SPI interface module.

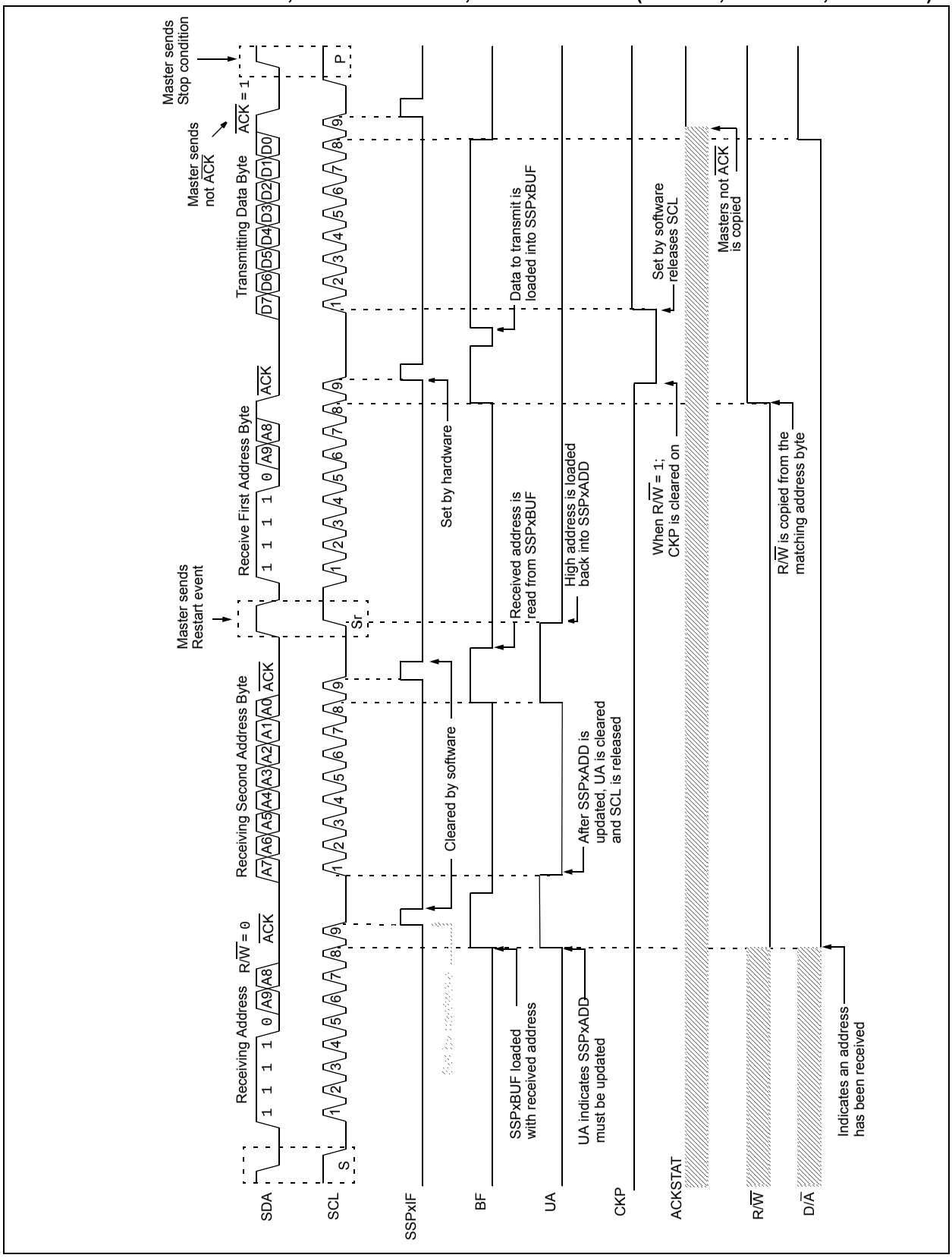
FIGURE 31-1: MSSP BLOCK DIAGRAM (SPI MODE)



Note 1: Output selection for master mode

2: Input selection for slave mode

FIGURE 31-22: I²C SLAVE, 10-BIT ADDRESS, TRANSMISSION (SEN = 0, AHEN = 0, DHEN = 0)



31.6.4 I²C MASTER MODE START CONDITION TIMING

To initiate a Start condition (Figure 31-26), the user sets the Start Enable bit, SEN bit of the SSPxCON2 register. If the SDA and SCL pins are sampled high, the Baud Rate Generator is reloaded with the contents of SSPxADD<7:0> and starts its count. If SCL and SDA are both sampled high when the Baud Rate Generator times out (TBRG), the SDA pin is driven low. The action of the SDA being driven low while SCL is high is the Start condition and causes the S bit of the SSPxSTAT1 register to be set. Following this, the Baud Rate Generator is reloaded with the contents of SSPxADD<7:0> and resumes its count. When the Baud Rate Generator times out (TBRG), the SEN bit of the SSPxCON2 register will be automatically cleared by hardware; the Baud Rate Generator is suspended, leaving the SDA line held low and the Start condition is complete.

Note 1: If at the beginning of the Start condition, the SDA and SCL pins are already sampled low, or if during the Start condition, the SCL line is sampled low before the SDA line is driven low, a bus collision occurs, the Bus Collision Interrupt Flag, BCLIF, is set, the Start condition is aborted and the I²C module is reset into its Idle state.

2: The Philips I²C specification states that a bus collision cannot occur on a Start.

FIGURE 31-26: FIRST START BIT TIMING

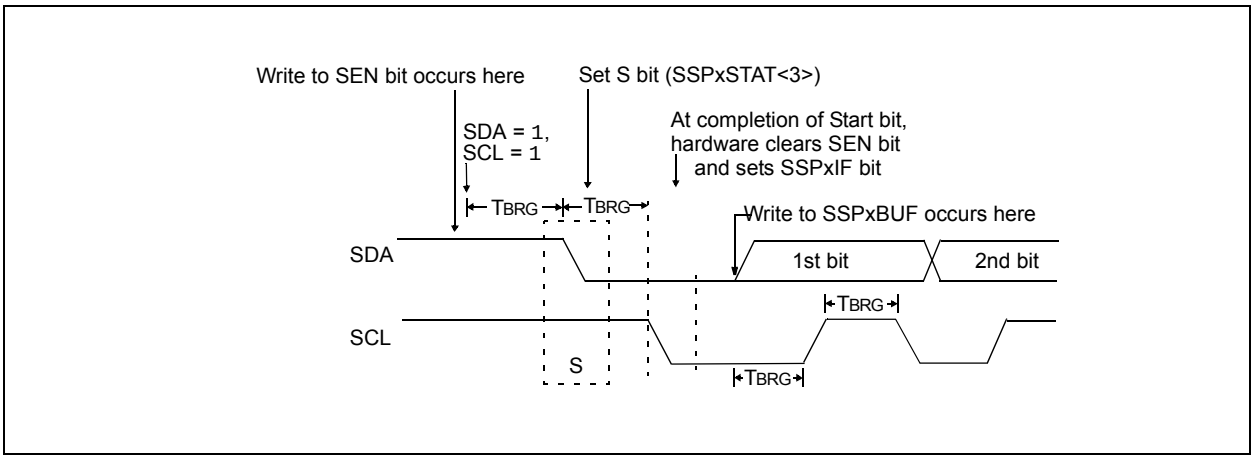


FIGURE 32-15: TIME OF FLIGHT MODE SINGLE ACQUISITION TIMING DIAGRAM

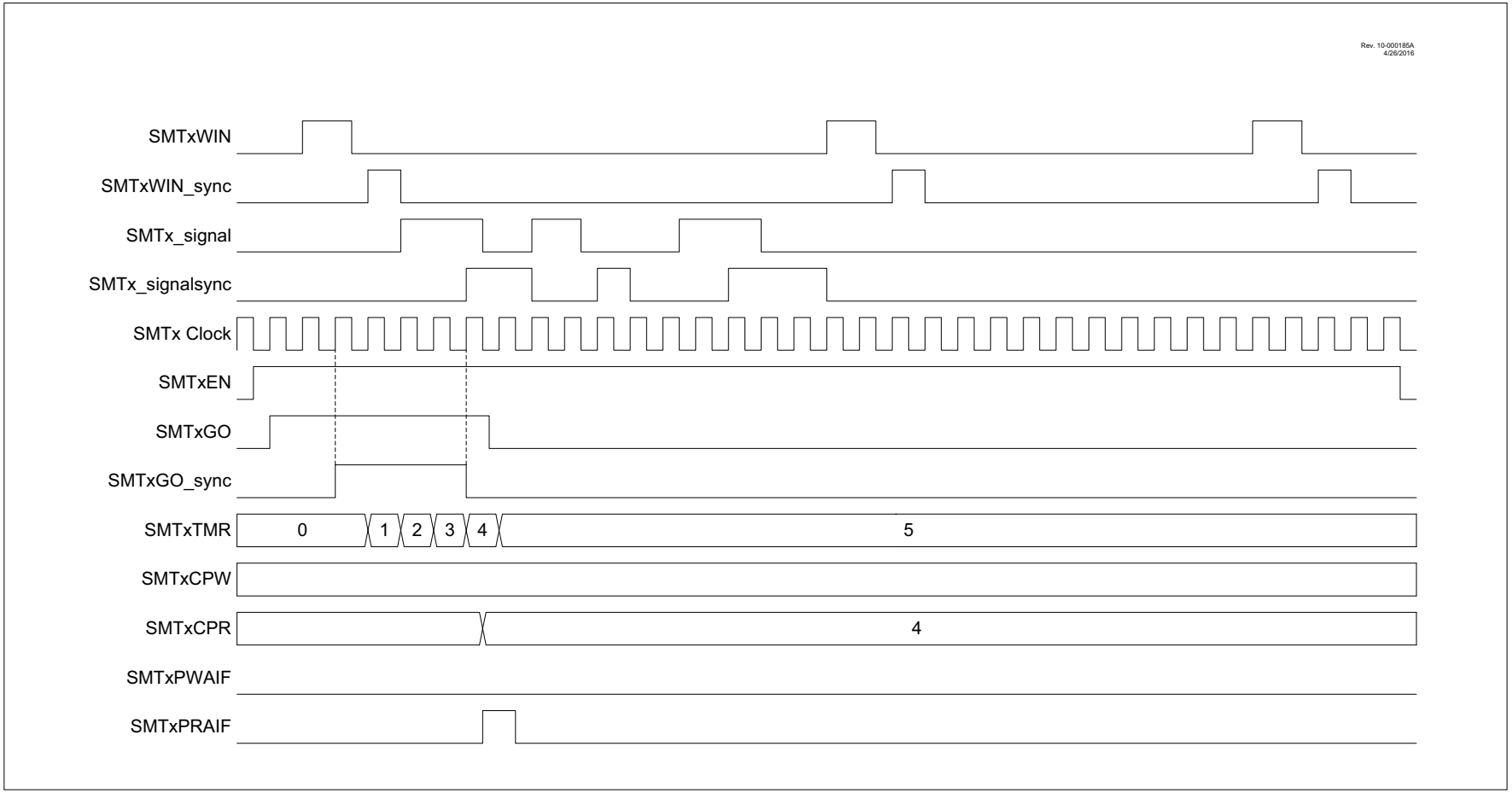


TABLE 37-4: I/O PORTS

Standard Operating Conditions (unless otherwise stated)							
Param. No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
	V_{IL}	**Input Low Voltage**					
		I/O PORT:					
D300		with TTL buffer	—	—	0.8	V	$4.5V \le V_{DD} \le 5.5V$
D301			—	—	0.15 V_{DD}	V	$1.8V \le V_{DD} \le 4.5V$
D302		with Schmitt Trigger buffer	—	—	0.2 V_{DD}	V	$2.0V \le V_{DD} \le 5.5V$
D303		with I^2C levels	—	—	0.3 V_{DD}	V	
D304		with SMBus levels	—	—	0.8	V	$2.7V \le V_{DD} \le 5.5V$
D305		$\overline{MCLR}$	—	—	0.2 V_{DD}	V	
	V_{IH}	**Input High Voltage**					
		I/O PORT:					
D320		with TTL buffer	2.0	—	—	V	$4.5V \le V_{DD} \le 5.5V$
D321			0.25 V_{DD} + 0.8	—	—	V	$1.8V \le V_{DD} \le 4.5V$
D322		with Schmitt Trigger buffer	0.8 V_{DD}	—	—	V	$2.0V \le V_{DD} \le 5.5V$
D323		with I^2C levels	0.7 V_{DD}	—	—	V	
D324		with SMBus levels	2.1	—	—	V	$2.7V \le V_{DD} \le 5.5V$
D325		$\overline{MCLR}$	0.7 V_{DD}	—	—	V	
	I_{IL}	**Input Leakage Current(1)**					
D340		I/O Ports	—	± 5	± 125	nA	$V_{SS} \le V_{PIN} \le V_{DD}$, Pin at high-impedance, 85°C
D341			—	± 5	± 1000	nA	$V_{SS} \le V_{PIN} \le V_{DD}$, Pin at high-impedance, 125°C
D342		$\overline{MCLR}$(2)	—	± 50	± 200	nA	$V_{SS} \le V_{PIN} \le V_{DD}$, Pin at high-impedance, 85°C
	I_{PUR}	**Weak Pull-up Current**					
D350			25	120	200	µA	V_{DD} = 3.0V, V_{PIN} = V_{SS}
	V_{OL}	**Output Low Voltage**					
D360		I/O ports	—	—	0.6	V	I_{OL} = 10.0mA, V_{DD} = 3.0V
	V_{OH}	**Output High Voltage**					
D370		I/O ports	V_{DD} - 0.7	—	—	V	I_{OH} = 6.0 mA, V_{DD} = 3.0V
D380	C_{IO}	**All I/O pins**	—	5	50	pF	

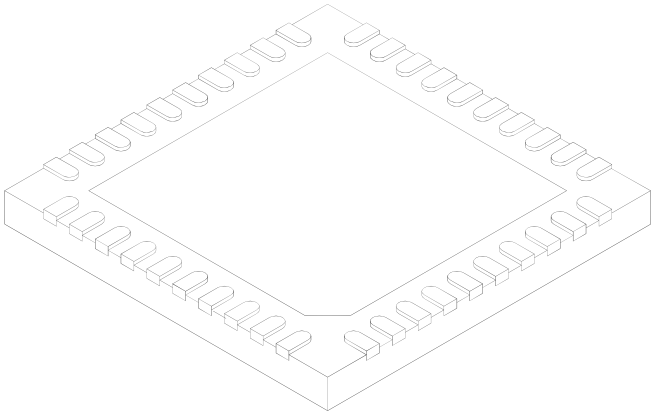
† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: Negative current is defined as current sourced by the pin.

2: The leakage current on the $\overline{MCLR}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

40-Lead Ultra Thin Plastic Quad Flat, No Lead Package (MV) – 5x5x0.5 mm Body [UQFN]

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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		40	
Pitch	e		0.40 BSC	
Overall Height	A	0.45	0.50	0.55
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3		0.127 REF	
Overall Width	E		5.00 BSC	
Exposed Pad Width	E2	3.60	3.70	3.80
Overall Length	D		5.00 BSC	
Exposed Pad Length	D2	3.60	3.70	3.80
Contact Width	b	0.15	0.20	0.25
Contact Length	L	0.30	0.40	0.50
Contact-to-Exposed Pad	K	0.20	-	-

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated.
3. Dimensioning and tolerancing per ASME Y14.5M.

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-156A Sheet 2 of 2