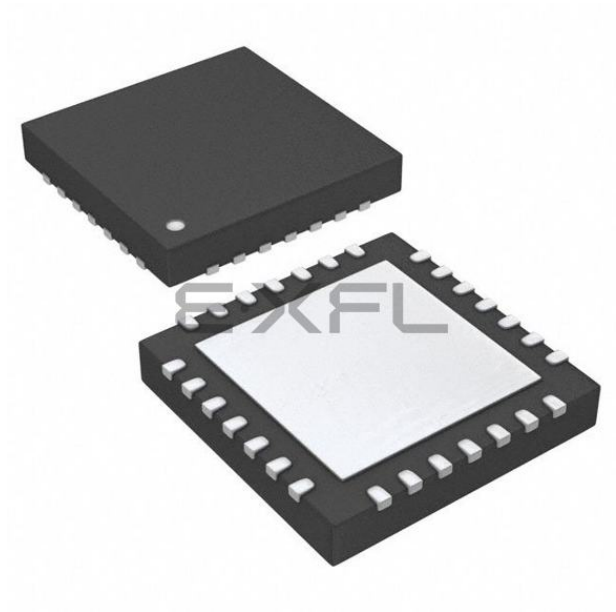


Welcome to E-XFL.COM

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)	1.8V ~ 3.6V
Data Converters	A/D 20x12b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°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/pic16lf19155-i-mv

TABLE 5: 48-PIN ALLOCATION TABLE (PIC16(L)F19185/86) (CONTINUED)

I/O(2)	48-Pin TQFP/QFN	ADC	Reference	Comparator	Zero-Cross Detect	DAC	Timers/SMT	CCP	PWM	CWG	MSSP	EUSART	CLC	RTCC	LCD	Interrupt-on-Change	High Current	Pull-up	Basic
RC0	34	—	—	—	—	—	T1CKI(1) SMTWIN1(1)	—	—	—	—	—	—	—	—	IOCC0	—	Y	SOSCO
RC1	35	—	—	—	—	—	SMTSIG1(1) T4IN(1)	CCP2(1)	—	—	—	—	—	—	—	IOCC1	—	Y	SOSCI
RC2	40	ANC2	—	—	—	—	—	CCP1(1)	—	—	—	—	—	—	COM2 SEG18	IOCC2	—	Y	—
RC3	41	ANC3	—	—	—	—	T2IN(1)	—	—	—	SCK(1) SCL(1,3,4)	—	—	—	SEG19	IOCC3	—	Y	—
RC4	46	ANC4	—	—	—	—	—	—	—	—	SDI(1) SDA(1,3,4)	—	—	—	SEG20	IOCC4	—	Y	—
RC6	48	ANC6	—	—	—	—	—	—	—	—	—	TX1(1) CK1(1)	—	—	SEG22 VLCD2	IOCC6	—	Y	—
RC7	1	ANC7	—	—	—	—	—	—	—	—	—	RX1(1) DT1(1)	—	—	SEG23 VLCD1	IOCC7	—	Y	—
RD0	42	AND0	—	—	—	—	—	—	—	—	—	—	—	—	SEG24	—	—	Y	—
RD1	43	AND1	—	—	—	—	—	—	—	—	—	—	—	—	SEG25	—	—	Y	—
RD2	44	AND2	—	—	—	—	—	—	—	—	—	—	—	—	COM5 SEG26	—	—	Y	—
RD3	45	AND3	—	—	—	—	—	—	—	—	—	—	—	—	COM4 SEG27	—	—	Y	—
RD4	2	AND4	—	—	—	—	—	—	—	—	—	—	—	—	SEG28	—	—	Y	—
RD5	3	AND5	—	—	—	—	—	—	—	—	—	—	—	—	SEG29	—	—	Y	—
RD6	4	AND6	—	—	—	—	—	—	—	—	—	—	—	—	SEG30	—	—	Y	—
RD7	5	AND7	—	—	—	—	—	—	—	—	—	—	—	—	SEG31	—	—	Y	—
RE0	27	ANE0	—	—	—	—	—	—	—	—	—	—	—	—	SEG32	—	—	Y	—
RE1	28	ANE1	—	—	—	—	—	—	—	—	—	—	—	—	COM6 SEG33	—	—	Y	—
RE2	29	ANE2	—	—	—	—	—	—	—	—	—	—	—	—	COM7 SEG34	—	—	Y	—
RE3	20	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IOCE3	—	Y	$\overline{MCLR}$

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.
2: All digital output signals shown in this row are PPS remappable. These signals may be mapped to output onto one or more PORTx pin options.
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. PPS assignments to the other pins will operate, but input logic levels will be standard TTL/ST as selected by INLCVL register, instead of the I^2C specific or SMBUS input buffer thresholds.
5: These are alternative I^2C logic levels pins.
6: In I^2C logic levels configuration, these pins can operate as either SCL and SDA pins.

2.0 GUIDELINES FOR GETTING STARTED WITH PIC16(L)F19155/56/75/76/85/86 MICROCONTROLLERS

2.1 Basic Connection Requirements

Getting started with the PIC16(L)F19155/56/75/76/85/86 family of 8-bit microcontrollers requires attention to a minimal set of device pin connections before proceeding with development.

The following pins must always be connected:

- All VDD and VSS pins (see **Section 2.2 "Power Supply Pins"**)
- MCLR pin (when configured for external operation) (see **Section 2.3 "Master Clear (MCLR) Pin"**)

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

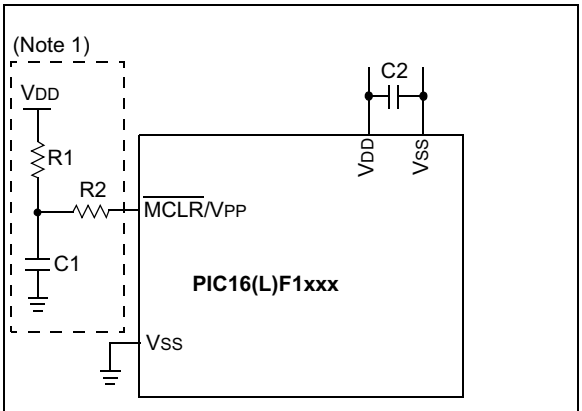
- ICSPCLK/ICSPDAT pins used for In-Circuit Serial Programming™ (ICSP™) and debugging purposes (see **Section 2.4 "ICSP™ Pins"**)
- OSC1, OSC2, SOSCO and SOSCI pins when an external oscillator source is used (see **Section 2.5 "External Oscillator Pins"**)

Additionally, the following pins may be required:

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

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

FIGURE 2-1: RECOMMENDED MINIMUM CONNECTIONS



Key (all values are recommendations):
C1: 10 nF, 16V ceramic
C2: 0.1 µF, 16V ceramic
R1: 10 kΩ
R2: 100Ω to 470Ω

Note1: Only when MCLRE configuration bit is 1 and the MCLR pin does not have a weak pull-up.

2.2 Power Supply Pins

2.2.1 DECOUPLING CAPACITORS

The use of decoupling capacitors on every pair of power supply pins (VDD and VSS) is required. All VDD and VSS pins must be connected. None can be left floating.

Consider the following criteria when using decoupling capacitors:

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

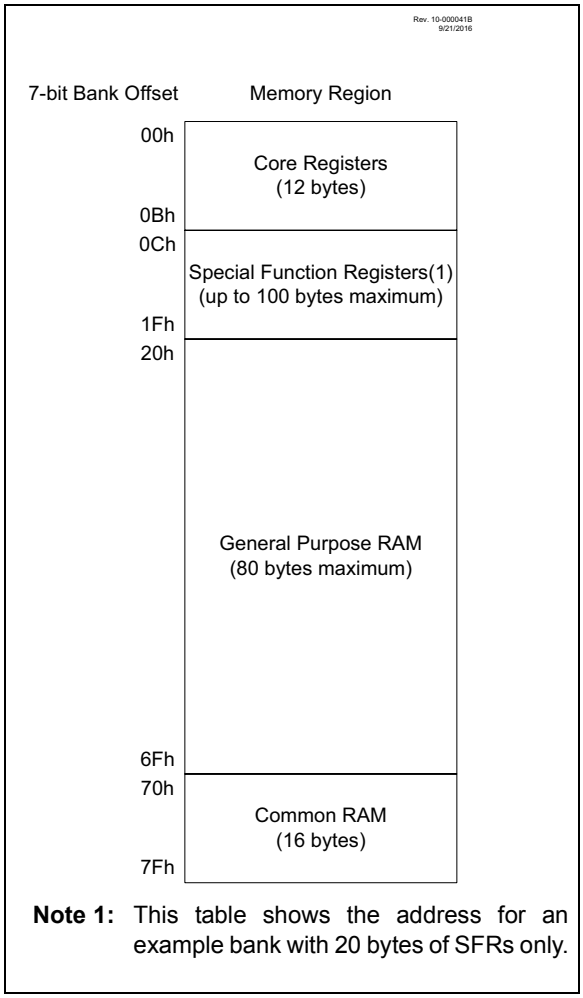
2.2.2 TANK CAPACITORS

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

4.3 Data Memory Organization

The data memory is partitioned into 64 memory banks with 128 bytes in each bank. Each bank consists of (**Section 4.3.6 "Device Memory Maps"**):

FIGURE 4-2: BANKED MEMORY PARTITIONING



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

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

4.3.1 BANK SELECTION

The active bank is selected by writing the bank number into the Bank Select Register (BSR). Unimplemented memory will read as '0'. 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 12-bit address. The upper five 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. For detailed information, see 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 $\overline{TO}$ and $\overline{PD}$ 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 the upper three bits 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 37.0 "Instruction Set Summary"**).

Note 1: The C and DC bits operate as $\overline{Borrow}$ and $\overline{Digit\ Borrow}$ 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
—	—	—	$\overline{TO}$	$\overline{PD}$	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 **$\overline{TO}$:** Time-Out bit
- 1 = After power-up, `CLRWDT` instruction or `SLEEP` instruction
- 0 = A WDT time-out occurred

bit 3 **$\overline{PD}$:** 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/$\overline{Digit\ Borrow}$ 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/$\overline{Borrow}$ 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 $\overline{Borrow}$, 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.

4.3.3 SPECIAL FUNCTION REGISTER

The Special Function Registers are registers used by the application to control the desired operation of peripheral functions in the device. The Special Function Registers occupy the 20 bytes of the data banks 0-59 and 100 bytes of the data banks 60-63, after the core registers.

The SFRs associated with the operation of the peripherals are described in the appropriate peripheral chapter of this data sheet.

4.3.4 GENERAL PURPOSE RAM

There are up to 80 bytes of GPR in each data memory bank.

TABLE 4-4: GENERAL PURPOSE RAM SIZE AND BANK LOCATION

Bank	PIC16(L)F19155, PIC16(L)F19175, PIC16(L)F19185		PIC16(L)F19156, PIC16(L)F19176, PIC16(L)F19186	
	Address	Size (Bytes)	Address	Size (Bytes)
0	020h-07Fh	96	020h-07Fh	96
1	0A0h-0EFh	80	0A0h-0EFh	80
2	120h-16Fh	80	120h-16Fh	80
3	1A0h-1EFh	80	1A0h-1EFh	80
4	220h-26Fh	80	220h-26Fh	80
5	2A0h-2EFh	80	2A0h-2EFh	80
6	320h-36Fh	80	320h-36Fh	80
7	3A0h-3EFh	80	3A0h-EFh	80
8	420h-46Fh	80	420h-46Fh	80
9	4A0h-4EFh	80	4A0h-4EFh	80
10	520h-560h	80	520h-560h	80
11	5A0h-5EFh	80	5A0h-5EFh	80
12	620h-64Fh	48	620h-64Fh	80
13			6A0h-6EFh	80
14			720h-76Fh	80
15			7A0h-7EFh	80
16			820h-96Fh	80
17			8A0h-8EFh	80
18			920h-96Fh	80
19			9A0h-9EFh	80
20			A20h-A6Fh	80
21			AA0h-AEFh	80
22			B20h-B6Fh	80
23			BA0h-BEFh	80
24			C20h-C6Fh	80
25			CA0h-CBFh	32

Legend: = Unimplemented GPR locations

TABLE 4-12: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-63 PIC16(L)F19155/56/75/76/85/86 (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on: MCLR
Bank 15											
CPU CORE REGISTERS; see Table 4-3 for specifics											
78Ch	—	Unimplemented								---- ----	---- ----
78Dh	—	Unimplemented								---- ----	---- ----
78Eh	—	Unimplemented								---- ----	---- ----
78Fh	—	Unimplemented								---- ----	---- ----
790h	—	Unimplemented								---- ----	---- ----
791h	—	Unimplemented								---- ----	---- ----
792h	—	Unimplemented								---- ----	---- ----
793h	—	Unimplemented								---- ----	---- ----
794h	—	Unimplemented								---- ----	---- ----
795h	—	Unimplemented								---- ----	---- ----
796h	PMD0	SYSCMD	FVRMD	ACTMD	—	—	NVMMD	—	IOCMD	000- -0-0	000- -0-0
797h	PMD1	—	—	—	TMR4MD	—	TMR2MD	TMR1MD	TMR0MD	---0 -000	0000 0000
798h	PMD2	RTCCMD	DACMD	ADCMD	—	—	CMP2MD	CMP1MD	ZCDMD	000- -000	000- -000
799h	PMD3	—	—	—	—	PWM4MD	PWM3MD	CCP2MD	CCP1MD	0000 0000	0000 0000
79Ah	PMD4	UART2MD	UART1MD	—	MSSP1MD	—	—	—	CWG1MD	00-0 ---0	00-0 ---0
79Bh	PMD5	—	SMT1MD	LCDMD	CLC4MD	CLC3MD	CLC2MD	CLC1MD	—	-000 000-	-000 000-
79Ch	—	Unimplemented								---- ----	---- ----
79Dh	—	Unimplemented								---- ----	---- ----
79Eh	—	Unimplemented								---- ----	---- ----
79Fh	—	Unimplemented								---- ----	---- ----

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

Note 1: Unimplemented data memory locations, read as '0'.

REGISTER 5-4: CONFIGURATION WORD 4: MEMORY

R/W-1	U-1	R/W-1	R/W-1	R/W-1	R/W-1
LVP	—	$\overline{\text{WRTSAF}}$(1)	$\overline{\text{WRTD}}$(1)	$\overline{\text{WRTC}}$(1)	$\overline{\text{WRTB}}$(1)
bit 13	12	11	10	9	bit 8

R/W-1	U-1	U-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{WRTAPP}}$(1)	—	—	$\overline{\text{SAFEN}}$(1)	$\overline{\text{BBEN}}$(1)	BBSIZE2	BBSIZE1	BBSIZE0
bit 7	6	5	4	3	2	1	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 **LVP:** Low Voltage Programming Enable bit
1 = Low voltage programming enabled. $\overline{\text{MCLR}}$/VPP pin function is $\overline{\text{MCLR}}$. MCLRE Configuration bit is ignored.
0 = HV on $\overline{\text{MCLR}}$/VPP must be used for programming.
The LVP bit cannot be written (to zero) while operating from the LVP programming interface. The purpose of this rule is to prevent the user from dropping out of LVP mode while programming from LVP mode, or accidentally eliminating LVP mode from the configuration state.
The preconditioned (erased) state for this bit is critical.

bit 12 **Unimplemented:** Read as '1'

bit 11 **$\overline{\text{WRTSAF}}$:** Storage Area Flash Write Protection bit
1 = SAF NOT write-protected
0 = SAF write-protected
Unimplemented, if SAF is not supported in the device family and only applicable if $\overline{\text{SAFEN}}$ = 0.

bit 10 **$\overline{\text{WRTD}}$:** Data EEPROM Write Protection bit
1 = Data EEPROM NOT write-protected
0 = Data EEPROM write-protected
Unimplemented if data EEPROM is not present.

bit 9 **$\overline{\text{WRTC}}$:** Configuration Register Write Protection bit
1 = Configuration Register NOT write-protected
0 = Configuration Register write-protected

bit 8 **$\overline{\text{WRTB}}$:** Boot Block Write Protection bit
1 = Boot Block NOT write-protected
0 = Boot Block write-protected
Only applicable if $\overline{\text{BBEN}}$ = 0.

bit 7 **$\overline{\text{WRTAPP}}$:** Application Block Write Protection bit
1 = Application Block NOT write-protected
0 = Application Block write-protected

bit 6-5 **Unimplemented:** Read as '1'.

bit 4 **$\overline{\text{SAFEN}}$:** SAF Enable bit
1 = SAF disabled
0 = SAF enabled

bit 3 **$\overline{\text{BBEN}}$:** Boot Block Enable bit
1 = Boot Block disabled
0 = Boot Block enabled

bit 2-0 **BBSIZE<2:0>:** Boot Block Size Selection bits
BBSIZE is used only when $\overline{\text{BBEN}}$ = 0
BBSIZ bits can only be written while $\overline{\text{BBEN}}$ = 1; after $\overline{\text{BBEN}}$ = 0, BBSIZ is write-protected.

Note 1: Bits are implemented as sticky bits. Once protection is enabled, it can only be reset through a Bulk Erase.

TABLE 14-5: SUMMARY OF REGISTERS ASSOCIATED WITH PORTD

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	241
TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	241
LATD	LATD7	LATD6	LATD5	LATD4	LATD3	LATD2	LATD1	LATD0	242
ANSELD	ANSD7	ANSD6	ANSD5	ANSD4	ANSD3	ANSD2	ANSD1	ANSD0	242
WPUD	WPUD7	WPUD6	WPUD5	WPUD4	WPUD3	WPUD2	WPUD1	WPUD0	243
ODCOND	ODCD7	ODCD6	ODCD5	ODCD4	ODCD3	ODCD2	ODCD1	ODCD0	243
SLRCOND	SLRD7	SLRD6	SLRD5	SLRD4	SLRD3	SLRD2	SLRD1	SLRD0	244
INLVLD	INLVLD7	INLVLD6	INLVLD5	INLVLD4	INLVLD3	INLVLD2	INLVLD1	INLVLD0	244

Legend: x = unknown, u = unchanged, – = unimplemented locations read as '0'. Shaded cells are not used by PORTD.

Note 1: Unimplemented, read as '1'.

REGISTER 14-46: ODCONF: PORTF OPEN-DRAIN CONTROL 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
ODCF7	ODCF6	ODCF5	ODCF4	ODCF3	ODCF2	ODCF1	ODCF0
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 **ODCF<7:0>:** PORTF Open-Drain Enable bits
For RF<7:0> pins, respectively
1 = Port pin operates as open-drain drive (sink current only)
0 = Port pin operates as standard push-pull drive (source and sink current)

REGISTER 14-47: SLRCONF: PORTF SLEW RATE CONTROL 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
SLRF7	SLRF6	SLRF5	SLRF4	SLRF3	SLRF2	SLRF1	SLRF0
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 **SLRF<7:0>:** PORTF Slew Rate Enable bits
For RF<7:0> pins, respectively
1 = Port pin slew rate is limited
0 = Port pin slews at maximum rate

REGISTER 14-48: INLVLF: PORTF INPUT LEVEL CONTROL 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
INLVLF7	INLVLF6	INLVLF5	INLVLF4	INLVLF3	INLVLF2	INLVLF1	INLVLF0
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 **INLVLF<7:0>:** PORTF Input Level Select bits
For RF<7:0> pins, respectively
1 = ST input used for PORT reads and interrupt-on-change
0 = TTL input used for PORT reads and interrupt-on-change

23.9 Register Definitions: ZCD Control

REGISTER 23-1: ZCDCON: ZERO-CROSS DETECTION CONTROL REGISTER

R/W-q/q	U-0	R-x/x	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
SEN	—	OUT	POL	—	—	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	q = value depends on Configuration bits

bit 7 **SEN**: Zero-Cross Detection Enable bit
1 = Zero-cross detect is enabled. ZCD pin is forced to output to source and sink current.
0 = Zero-cross detect is disabled. ZCD pin operates according to PPS and TRIS controls.

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

bit 5 **OUT:** Zero-Cross Detection Logic Level bit
POL bit = 1:
1 = ZCD pin is sourcing current
0 = ZCD pin is sinking current
POL bit = 0:
1 = ZCD pin is sinking current
0 = ZCD pin is sourcing current

bit 4 **POL:** Zero-Cross Detection Logic Output Polarity bit
1 = ZCD logic output is inverted
0 = ZCD logic output is not inverted

bit 3-2 **Unimplemented:** Read as '0'

bit 1 **INTP:** Zero-Cross Positive Edge Interrupt Enable bit
1 = ZCDIF bit is set on low-to-high ZCDx_output transition
0 = ZCDIF bit is unaffected by low-to-high ZCDx_output transition

bit 0 **INTN:** Zero-Cross Negative Edge Interrupt Enable bit
1 = ZCDIF bit is set on high-to-low ZCDx_output transition
0 = ZCDIF bit is unaffected by high-to-low ZCDx_output transition

TABLE 23-1: SUMMARY OF REGISTERS ASSOCIATED WITH THE ZCD MODULE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
PIE3	RC2IE	TX2IE	RC1IE	TX1IE	—	—	BCL1IE	SSP1IE	168
PIR3	RC2IF	TX2IF	RC1IF	TX1IF	—	—	BCL1IF	SSP1IF	177
ZCDxCON	SEN	—	OUT	POL	—	—	INTP	INTN	349

Legend: — = unimplemented, read as '0'. Shaded cells are unused by the ZCD module.

TABLE 23-2: SUMMARY OF CONFIGURATION WORD WITH THE ZCD MODULE

Name	Bits	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	Register on Page
CONFIG2	13:8	—	—	$\overline{\text{DEBUG}}$	STVREN	PPS1WAY	ZCDDIS	BORV	—	121
	7:0	BOREN <1:0>		$\overline{\text{LPBOREN}}$	—	—	—	$\overline{\text{PWRTE}}$	MCLRE	

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by the ZCD module.

REGISTER 24-12: ALRMMTH: ALARM MONTH CONTROL REGISTER

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	ALRMHMONTH	ALRMLMONTH <3:0>			
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7-5 **Unimplemented:** Read as ‘0’

bit 4 **ALRMHMONTH:** Binary Coded Decimal value of months ‘10’ digit; valid value from 0 to 1

bit 3-0 **ALRMLMONTH<3:0>:** Binary Coded Decimal value of months ‘1’ digit; valid value from 0 to 9

REGISTER 24-13: ALRMWD: ALARM WEEKDAY CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-x	R/W-x	R/W-x
—	—	—	—	—	ALRMLWDAY<2:0>		
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 7-3 **Unimplemented:** Read as ‘0’

bit 2-0 **ALRMLWDAY<2:0>:** Binary Coded Decimal value of weekdays ‘1’ digit; valid values from 0 to 6.

REGISTER 24-14: ALRMDAY: ALARM DAY CONTROL REGISTER

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	ALRMHDAY<1:0>		ALRMLDAY<3:0>			
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

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

bit 5-4 **ALRMHDAY<1:0>:** Binary Coded Decimal value of days ‘10’ digit; valid value from 0 to 3

bit 3-0 **ALRMLDAY<3:0>:** Binary Coded Decimal value of days ‘1’ digit; valid value from 0 to 9

25.0 TIMER0 MODULE

The Timer0 module is an 8/16-bit timer/counter with the following features:

- 16-bit timer/counter
- 8-bit timer/counter with programmable period
- Synchronous or asynchronous operation
- Selectable clock sources
- Programmable prescaler (independent of Watchdog Timer)
- Programmable postscaler
- Operation during Sleep mode
- Interrupt on match or overflow
- Output on I/O pin (via PPS) or to other peripherals

25.1 Timer0 Operation

Timer0 can operate as either an 8-bit timer/counter or a 16-bit timer/counter. The mode is selected with the T016BIT bit of the T0CON register.

25.1.1 16-BIT MODE

In normal operation, TMR0 increments on the rising edge of the clock source. A 15-bit prescaler on the clock input gives several prescale options (see prescaler control bits, T0CKPS<3:0> in the T0CON1 register).

25.1.1.1 Timer0 Reads and Writes in 16-Bit Mode

TMR0H is not the actual high byte of Timer0 in 16-bit mode. It is actually a buffered version of the real high byte of Timer0, which is neither directly readable nor writable (see Figure 25-1). TMR0H is updated with the contents of the high byte of Timer0 during a read of TMR0L. This provides the ability to read all 16 bits of Timer0 without having to verify that the read of the high and low byte was valid, due to a rollover between successive reads of the high and low byte.

Similarly, a write to the high byte of Timer0 must also take place through the TMR0H Buffer register. The high byte is updated with the contents of TMR0H when a write occurs to TMR0L. This allows all 16 bits of Timer0 to be updated at once.

25.1.2 8-BIT MODE

In normal operation, TMR0 increments on the rising edge of the clock source. A 15-bit prescaler on the clock input gives several prescale options (see prescaler control bits, T0CKPS<3:0> in the T0CON1 register).

The value of TMR0L is compared to that of the Period buffer, a copy of TMR0H, on each clock cycle. When the two values match, the following events happen:

- TMR0_out goes high for one prescaled clock period
- TMR0L is reset
- The contents of TMR0H are copied to the period buffer

In 8-bit mode, the TMR0L and TMR0H registers are both directly readable and writable. The TMR0L register is cleared on any device Reset, while the TMR0H register initializes at FFh.

Both the prescaler and postscaler counters are cleared on the following events:

- A write to the TMR0L register
- A write to either the T0CON0 or T0CON1 registers
- Any device Reset – Power-on Reset (POR), MCLR Reset, Watchdog Timer Reset (WDTR) or
- Brown-out Reset (BOR)

25.1.3 COUNTER MODE

In Counter mode, the prescaler is normally disabled by setting the T0CKPS bits of the T0CON1 register to '0000'. Each rising edge of the clock input (or the output of the prescaler if the prescaler is used) increments the counter by '1'.

25.1.4 TIMER MODE

In Timer mode, the Timer0 module will increment every instruction cycle as long as there is a valid clock signal and the T0CKPS bits of the T0CON1 register (Register 25-2) are set to '0000'. When a prescaler is added, the timer will increment at the rate based on the prescaler value.

25.1.5 ASYNCHRONOUS MODE

When the T0ASYNC bit of the T0CON1 register is set (T0ASYNC = '1'), the counter increments with each rising edge of the input source (or output of the prescaler, if used). Asynchronous mode allows the counter to continue operation during Sleep mode provided that the clock also continues to operate during Sleep.

25.1.6 SYNCHRONOUS MODE

When the T0ASYNC bit of the T0CON1 register is clear (T0ASYNC = 0), the counter clock is synchronized to the system oscillator (FOSC/4). When operating in Synchronous mode, the counter clock frequency cannot exceed FOSC/4.

28.7.9 COUNTER MODE

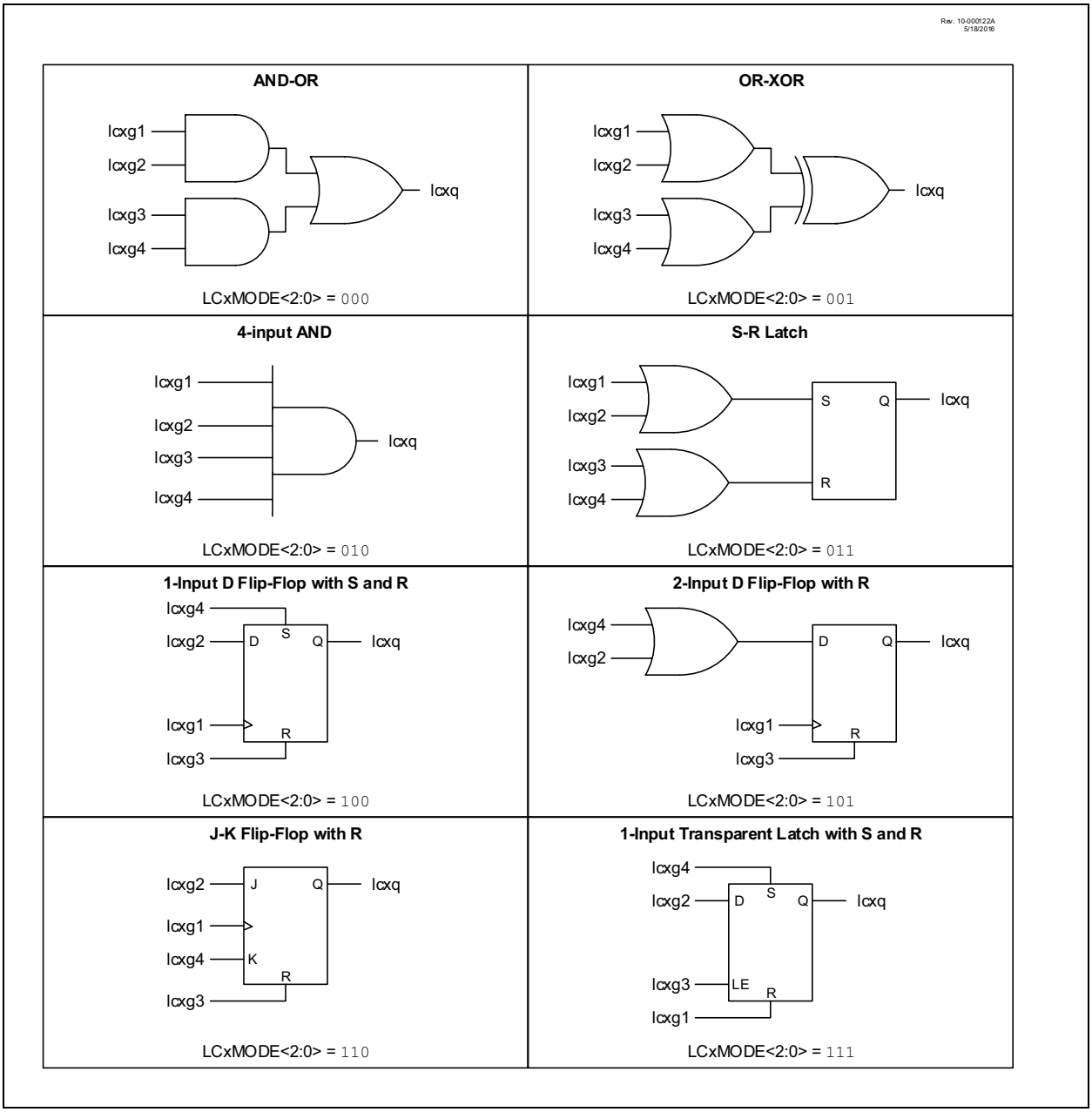
Counter mode increments the timer on each pulse of the SMTx_signal input. This mode is asynchronous to the SMT clock and uses the SMTx_signal as a time source. The SMTxCPW register will be updated with the current SMTxTMR value on the rising edge of the SMTxWIN input. See Figure 28-18.

TABLE 31-2: SUMMARY OF REGISTERS ASSOCIATED WITH CWG

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CWG1CLKCON	—	—	—	—	—	—	—	CS	492
CWG1ISM	—	—	—	—	IS<3:0>				492
CWG1DBR	—	—	DBR<5:0>						488
CWG1DBF	—	—	DBF<5:0>						488
CWG1CON0	EN	LD	—	—	—	MODE<2:0>			486
CWG1CON1	—	—	IN	—	POLD	POLC	POLB	POLA	487
CWG1AS0	SHUTDOWN	REN	LSBD<1:0>		LSAC<1:0>		—	—	489
CWG1AS1	—	—	—	AS4E	AS3E	AS2E	AS1E	AS0E	490
CWG1STR	OVRD	OVRC	OVRB	OVRA	STRD	STRC	STRB	STRA	491
INTCON	GIE	PEIE	—	—	—	—	—	INTEDG	164
PIE7	—	—	NVMIE	—	—	—	—	CWG1IE	172
PIR7	—	—	NVMIF	—	—	—	—	CWG1IF	181

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

FIGURE 32-3: PROGRAMMABLE LOGIC FUNCTIONS



33.5.6 CLOCK STRETCHING

Clock stretching occurs when a device on the bus holds the SCL line low, effectively pausing communication. The slave may stretch the clock to allow more time to handle data or prepare a response for the master device. A master device is not concerned with stretching as anytime it is active on the bus and not transferring data it is stretching. Any stretching done by a slave is invisible to the master software and handled by the hardware that generates SCL.

The CKP bit of the SSPxCON1 register is used to control stretching in software. Any time the CKP bit is cleared, the module will wait for the SCL line to go low and then hold it. Setting CKP will release SCL and allow more communication.

33.5.6.1 Normal Clock Stretching

Following an $\overline{ACK}$ if the R/$\overline{W}$ bit of SSPxSTAT is set, a read request, the slave hardware will clear CKP. This allows the slave time to update SSPxBUF with data to transfer to the master. If the SEN bit of SSPxCON2 is set, the slave hardware will always stretch the clock after the $\overline{ACK}$ sequence. Once the slave is ready; CKP is set by software and communication resumes.

33.5.6.2 10-bit Addressing Mode

In 10-bit Addressing mode, when the UA bit is set the clock is always stretched. This is the only time the SCL is stretched without CKP being cleared. SCL is released immediately after a write to SSPxADD.

33.5.6.3 Byte NACKing

When AHEN bit of SSPxCON3 is set; CKP is cleared by hardware after the eighth falling edge of SCL for a received matching address byte. When DHEN bit of SSPxCON3 is set; CKP is cleared after the eighth falling edge of SCL for received data.

Stretching after the eighth falling edge of SCL allows the slave to look at the received address or data and decide if it wants to ACK the received data.

33.5.7 CLOCK SYNCHRONIZATION AND THE CKP BIT

Any time the CKP bit is cleared, the module will wait for the SCL line to go low and then hold it. However, clearing the CKP bit will not assert the SCL output low until the SCL output is already sampled low. Therefore, the CKP bit will not assert the SCL line until an external I^2C master device has already asserted the SCL line. The SCL output will remain low until the CKP bit is set and all other devices on the I^2C bus have released SCL. This ensures that a write to the CKP bit will not violate the minimum high time requirement for SCL (see Figure 33-23).

FIGURE 33-23: CLOCK SYNCHRONIZATION TIMING

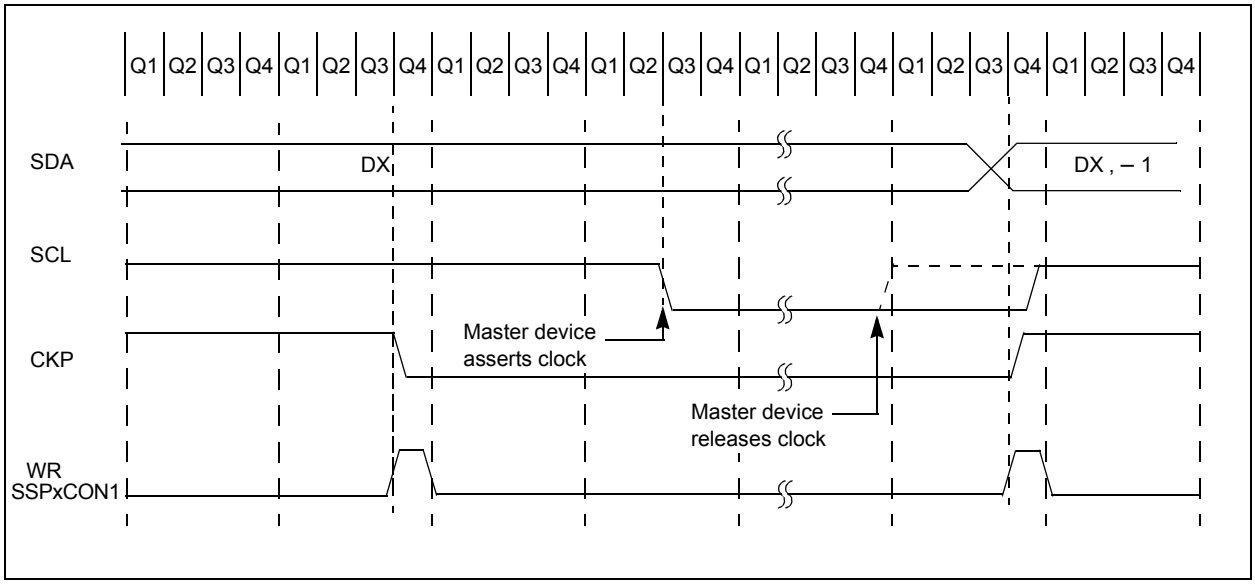
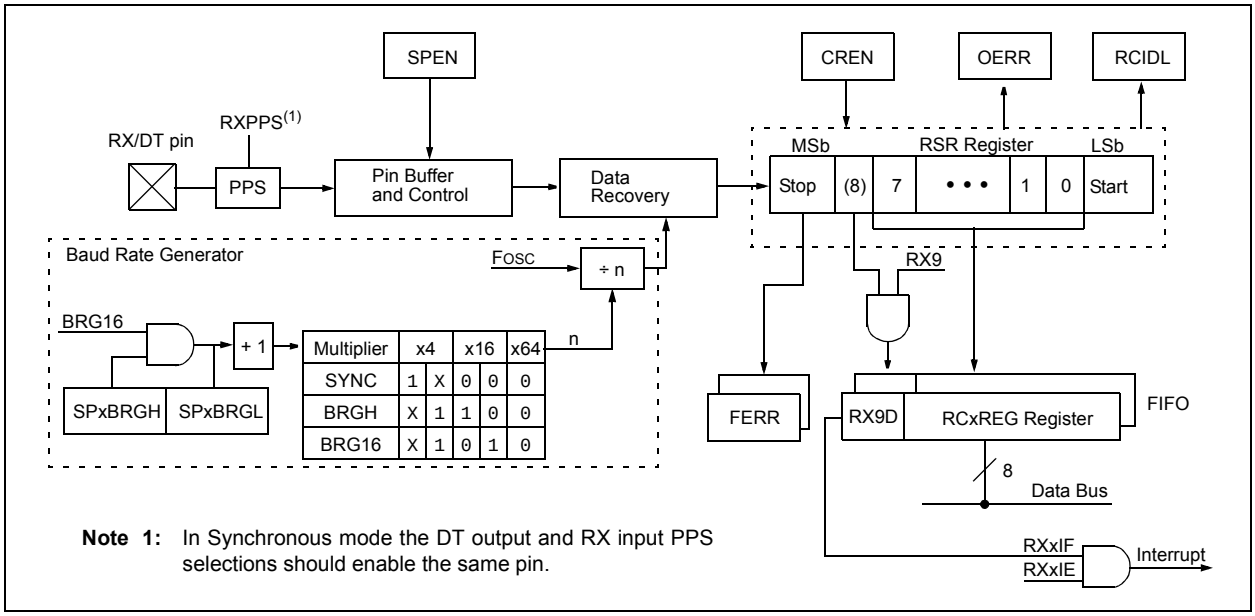


FIGURE 34-2: EUSART RECEIVE BLOCK DIAGRAM



Note 1: In Synchronous mode the DT output and RX input PPS selections should enable the same pin.

The operation of the EUSART module is controlled through three registers:

- Transmit Status and Control (TXxSTA)
- Receive Status and Control (RCxSTA)
- Baud Rate Control (BAUDxCON)

These registers are detailed in Register 34-1, Register 34-2 and Register 34-3, respectively.

The RX input pin is selected with the RXPPS. The CK input is selected with the TXPPS register. TX, CK, and DT output pins are selected with each pin's RxyPPS register. Since the RX input is coupled with the DT output in Synchronous mode, it is the user's responsibility to select the same pin for both of these functions when operating in Synchronous mode. The EUSART control logic will control the data direction drivers automatically.

REGISTER 35-1: LCDCON: LCD CONTROL REGISTER

R/W-0	R/W-0	HS/C-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
LCDEN	SLPEN	WERR	CS	LMUX3	LMUX2	LMUX1	LMUX0
bit 7							bit 0

Legend:	C = Clearable bit	HS = Bit is set by hardware	
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **LCDEN**: LCD Enable bit
1 = LCD module is enabled
0 = LCD module is disabled

bit 6 **SLPEN**: LCD Display Sleep-Enabled bit
1 = Module will stop driving in Sleep
0 = Module will continue driving in Sleep

bit 5 **WERR**: LCD Write Failed Error bit(1)
1 = Write failure to LCDDATA register occurred (must be reset in software)
0 = No LCD write error

bit 4 **CS**: Clock Source Select bit
1 = SOSC Selected
0 = LFINTOSC Selected

bit 3-0 **LMUX<3:0>**: Common Selection bits. Specifies the number of commons(2)

LMUX<3:0>	Multiplex	Bias
0000	All COMs off	—
0001	Static (COM0)	Static
0010	1/2 MUX (COM<1:0>)	1/2
0011	1/3 MUX (COM<2:0>)	1/3
0100	1/4 MUX (COM<3:0>)	1/3
0101	1/5 MUX (COM<4:0>)	1/3
0110	1/6 MUX (COM<5:0>)	1/3
0111	1/7 MUX (COM<6:0>)	1/3
1000	1/8 MUX (COM<7:0>)	1/3

Note 1: Bit can only be set by hardware and only cleared in software by writing to zero.
2: Cannot be changed when LCDEN = 1.

37.3 Instruction Descriptions

ADDFSR — Add Literal to FSRn

Syntax:	[*label*] ADDFSR FSRn, k
Operands:	-32 ≤ k ≤ 31 n ∈ [0, 1]
Operation:	FSR(n) + k → FSR(n)
Status Affected:	None
Description:	The signed 6-bit literal 'k' is added to the contents of the FSRnH:FSRnL register pair. FSRn is limited to the range 0000h-FFFFh. Moving beyond these bounds will cause the FSR to wrap-around.

ADDLW — Add literal and W

Syntax:	[*label*] ADDLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) + k → (W)
Status Affected:	C, DC, Z
Description:	The contents of the W register are added to the 8-bit literal 'k' and the result is placed in the W register.

ADDWF — Add W and f

Syntax:	[*label*] ADDWF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(W) + (f) → (destination)
Status Affected:	C, DC, Z
Description:	Add the contents of the W register with register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

ADDWFC — ADD W and CARRY bit to f

Syntax:	[*label*] ADDWFC f {,d}
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(W) + (f) + (C) → dest
Status Affected:	C, DC, Z
Description:	Add W, the Carry flag and data memory location 'f'. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed in data memory location 'f'.

ANDLW — AND literal with W

Syntax:	[*label*] ANDLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) .AND. (k) → (W)
Status Affected:	Z
Description:	The contents of W register are AND'ed with the 8-bit literal 'k'. The result is placed in the W register.

ANDWF — AND W with f

Syntax:	[*label*] ANDWF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(W) .AND. (f) → (destination)
Status Affected:	Z
Description:	AND the W register with register 'f'. If 'd' is '0', the result is stored in the W register. If 'd' is '1', the result is stored back in register 'f'.

ASRF — Arithmetic Right Shift

Syntax:	[*label*] ASRF f {,d}
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f<7>)→ 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. The MSb remains unchanged. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f'.

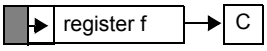


TABLE 39-13: ANALOG-TO-DIGITAL CONVERTER (ADC) CONVERSION TIMING SPECIFICATIONS

Standard Operating Conditions (unless otherwise stated)							
Param. No.	**Sym.**	**Characteristic**	**Min.**	**Typ†**	**Max.**	**Units**	**Conditions**
AD20	TAD	ADC Clock Period	1	—	9	μs	Using FOSC as the ADC clock source ADCS = 1
AD21			—	2	—	μs	Using FRC as the ADC clock source ADSC = 0
AD22	TCNV	Conversion Time	—	14 TAD + 2 TCY	—	—	Using FOSC as the ADC clock source ADCS = 1
			—	16 TAD + 2 TCY	—	—	Using FRC as the ADC clock source ADSC = 0
AD24	THCD	Sample and Hold Capacitor Disconnect Time	—	2 TAD + 1 TCY	—	—	Using FOSC as the ADC clock source ADCS = 1
			—	3 TAD + 2 TCY	—	—	Using FRC as the ADC clock source ADSC = 0

* These parameters are characterized but not tested.

† 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: A TAD of 8 μs is recommended measuring the 1/3 tap point on the VBAT.

FIGURE 39-10: ADC CONVERSION TIMING (ADC CLOCK Fosc-BASED)