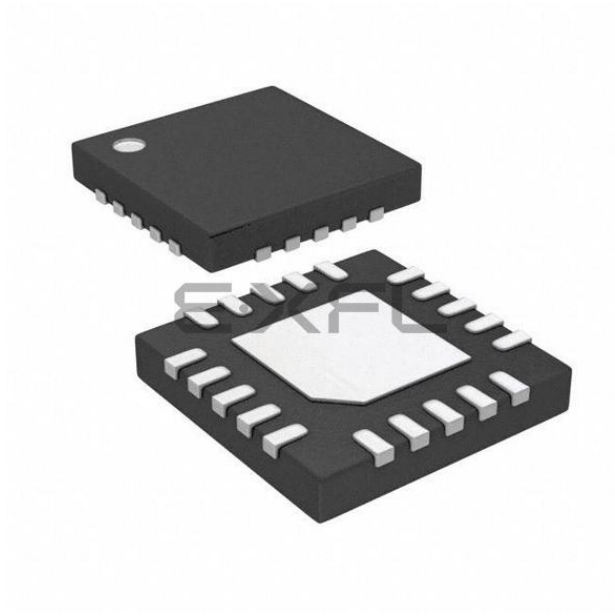


Welcome to E-XFL.COM

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	18
Program Memory Size	28KB (16K x 14)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 17x12b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	20-UFQFN Exposed Pad
Supplier Device Package	20-UQFN (4x4)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lf18446t-i-gz

Pin Diagrams

1 14/16-Pin Diagrams

Figure 1. 14-Pin PDIP, SOIC, TSSOP

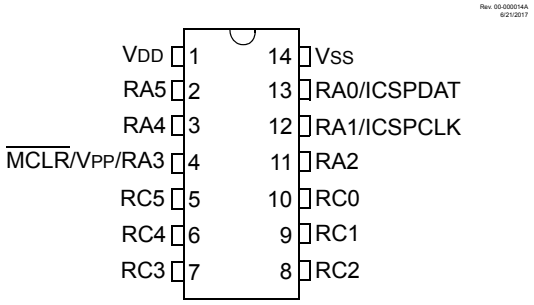
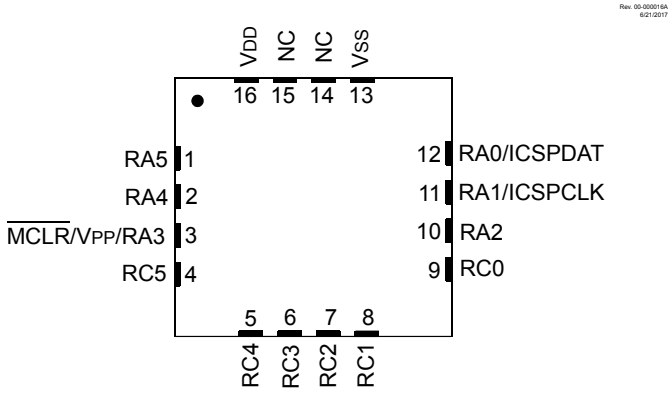


Figure 2. 16-Pin UQFN (4x4)



Note: It is recommended that the exposed bottom pad be connected to V_{SS}.

Related Links

14/16-Pin Allocation Table

2 20-Pin Diagrams

Figure 3. 20-Pin PDIP, SOIC, SSOP

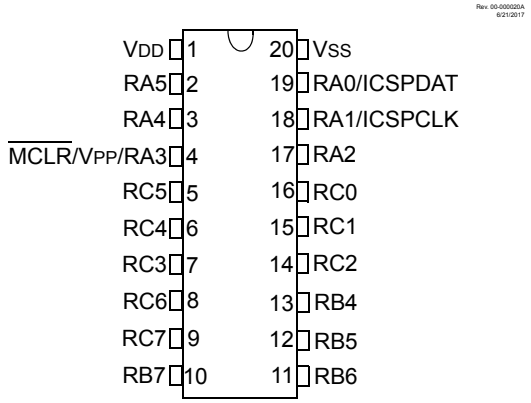
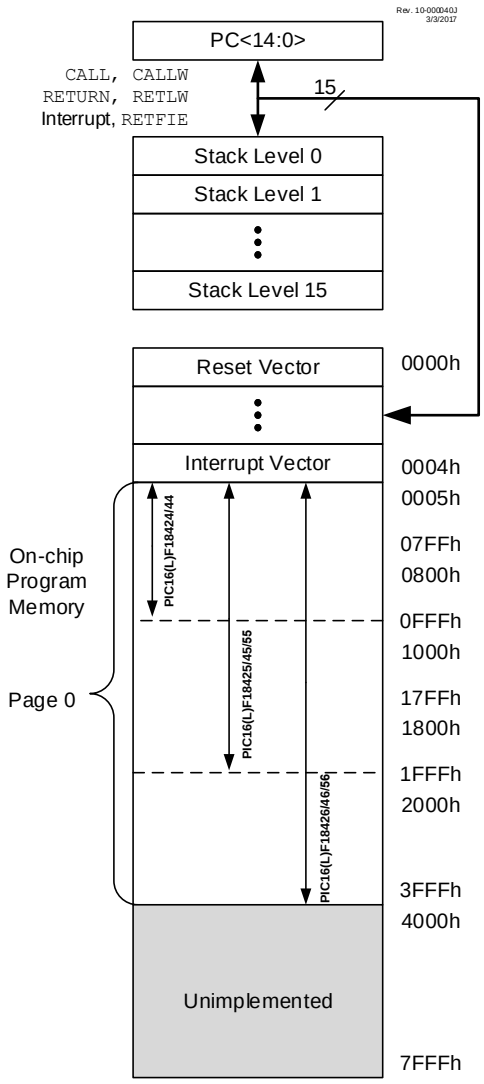


Figure 7-1. Program Memory and Stack



Related Links

CONFIG5

Memory Violation

7.1.1 Reading Program Memory as Data

There are three methods of accessing constants in program memory. The first method is to use tables of `RETLW` instructions. The second method is to set an FSR to point to the program memory. The third method is to use the NVMREG interface to access the program memory.

Related Links

NVMREG Access

Condition	Program Counter	STATUS Register	PCON0 Register	PCON1 Register
Interrupt Wake-up from Sleep	PC + 1(1)	---1 0uuu	uuuu uuuu	---- --u-
RESET Instruction Executed	0	---u uuuu	uuuu u0uu	---- --u-
Stack Overflow Reset (STVREN = 1)	0	---u uuuu	1uuu uuuu	---- --u-
Stack Underflow Reset (STVREN = 1)	0	---u uuuu	u1uu uuuu	---- --u-
Memory Violation Reset ($\overline{MEMV}$ = 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.

Related Links

STATUS

8.13 Power Control (PCONx) Register

The Power Control (PCONx) registers contain flag bits to differentiate between a:

- Brown-out Reset ($\overline{BOR}$)
- Power-on Reset ($\overline{POR}$)
- Reset Instruction Reset ($\overline{RI}$)
- $\overline{MCLR}$ Reset ($\overline{RMCLR}$)
- Watchdog Timer Reset ($\overline{RWDT}$)
- Watchdog Window Violation ($\overline{WDTWV}$)
- Stack Underflow Reset (STKUNF)
- Stack Overflow Reset (STKOVF)
- Memory Violation Reset ($\overline{MEMV}$)

Hardware will change the corresponding register bit during the Reset process; if the Reset was not caused by the condition, the bit remains unchanged.

Software should reset the bit to the inactive state after restart (hardware will not reset the bit).

Software may also set any PCONx bit to the active state, so that user code may be tested, but no Reset action will be generated.

Related Links

Determining the Cause of a Reset

PCON0

PCON1

Value	Condition	Description
1	PWM mode	Output trailing edge occurred (must be cleared in software)
0	PWM mode	Output trailing edge did not occur

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 appropriate interrupt flag bits are clear prior to enabling an interrupt.

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

Related Links

Interrupts During Doze

```
    BSF     NVMCON1, WR       ; Step 4: Set WR bit to begin write/erase
    BSF     INTCON, GIE       ; Re-enable interrupts
```

Note:

1. Sequence begins when NVMCON2 is written; steps 1-4 must occur in the cycle-accurate order shown.
2. Opcodes shown are illustrative; any instruction that has the indicated effect may be used.

13.4.3 NVMREG Write to EEPROM

Writing to the EEPROM is accomplished by the following steps:

1. Set the NVMREGS and WREN bits of the NVMCON1 register.
2. Write the desired address (address +7000h) into the NVMADRH:NVMADRL register pair.
3. Perform the unlock sequence as described in the *"NVM Unlock Sequence"* section.

A single EEPROM byte is written with NVMDATA. The operation includes an implicit erase cycle for that byte (it is not necessary to set the FREE bit), and requires many instruction cycles to finish. CPU execution continues in parallel and, when complete, WR is cleared by hardware, NVMIF is set, and an interrupt will occur if NVMIE is also set. Software must poll the WR bit to determine when writing is complete, or wait for the interrupt to occur. WREN will remain unchanged. Once the EEPROM write operation begins, clearing the WR bit will have no effect; the operation will run to completion.

Related Links

NVM Unlock Sequence

NVMREG Erase of Program Memory

13.4.4 NVMREG Erase of Program Memory

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

1. Clear the NVMREGS bit of the NVMCON1 register to erase program memory locations, or set the NMVREGS bit to erase User ID locations.
2. Write the desired address into the NVMADRH:NVMADRL register pair.
3. Set the FREE and WREN bits of the NVMCON1 register.
4. Perform the unlock sequence as described in the *"NVM Unlock Sequence"* section.

If the program memory address is write-protected, the WR bit will be cleared and the erase operation will not take place.

While erasing program memory, CPU operation is suspended, and resumes when the operation is complete. Upon completion, the NVMIF is set, and an interrupt will occur if the NVMIE bit is also set.

Write latch data is not affected by erase operations, and WREN will remain unchanged.

13.6.4 NVMCON2

Name: NVMCON2
Address: 0x81F

Nonvolatile Memory Control 2 Register

Bit	7	6	5	4	3	2	1	0
	NVMCON2[7:0]							
Access	WO	WO	WO	WO	WO	WO	WO	WO
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – NVMCON2[7:0] Flash Memory Unlock Pattern bits

Note: To unlock writes, a 55h must be written first followed by an AAh before setting the WR bit of the NVMCON1 register. The value written to this register is used to unlock the writes.

14.6.7 LATA

Name: LATA
Address: 0x018

Output Latch Register

Bit	7	6	5	4	3	2	1	0
			LATA5	LATA4		LATA2	LATA1	LATA0
Access			R/W	R/W	RO	R/W	R/W	R/W
Reset			x	x	1	x	x	x

Bits 4, 5 – LATAn Output Latch A Value bits
Reset States: POR/BOR = xx
All Other Resets = uu

Bits 0, 1, 2 – LATAn Output Latch A Value bits
Reset States: POR/BOR = xxx
All Other Resets = uuu

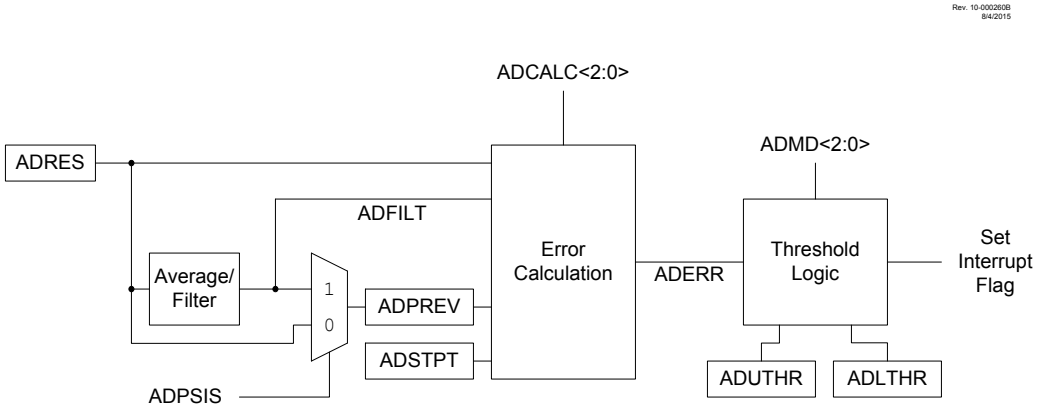
Note: Writes to LATA are equivalent with writes to the corresponding PORTA register. Reads from LATA register return register values, not I/O pin values.

Output Signal Name	RxyPPS Register Value
C2OUT	0x12
CCP1OUT	0x09
CCP2OUT	0x0A
CCP3OUT	0x0B
CCP4OUT	0x0C
CLC1OUT	0x01
CLC2OUT	0x02
CLC3OUT	0x03
CLC4OUT	0x04
CK(1)/TX1	0x0F
CLKR	0x19
CWG1A	0x05
CWG1B	0x06
CWG1C	0x07
CWG1D	0x08
CWG2A	0x1B
CWG2B	0x1C
CWG2C	0x1D
CWG2D	0x1E
DSM1OUT	0x1A
DT(1)	0x10
NCO1OUT	0x18
PWM6OUT	0x0D
PWM7OUT	0x0E
SCK1	0x13
SCL1	0x13
SDA1	0x14
SDO1	0x14
SCK2	0x15
SCL2	0x15
SDA2	0x16
SDO2	0x16

20.6 Computation Operation

The ADC module hardware is equipped with post conversion computation features. These features provide data post-processing functions that can be applied to the ADC conversion result, including digital filtering/averaging and threshold comparison functions.

Figure 20-10. Computational Features Simplified Block Diagram



The operation of the ADC computational features is controlled by MD bits.

The module can be operated in one of five modes:

- **Basic**: This is a legacy mode. In this mode, ADC conversion occurs on single (DSEN = 0) or double (DSEN = 1) samples. ADIF is set after all the conversions are complete.
- **Accumulate**: With each trigger, the ADC conversion result is added to accumulator and ADCNT increments. ADIF is set after each conversion. ADTIF is set according to the calculation mode.
- **Average**: With each trigger, the ADC conversion result is added to the accumulator. When the ADRPT number of samples have been accumulated, a threshold test is performed. Upon the next trigger, the accumulator is cleared. For the subsequent tests, additional ADRPT samples are required to be accumulated.
- **Burst Average**: At the trigger, the accumulator is cleared. The ADC conversion results are then collected repetitively until ADRPT samples are accumulated and finally the threshold is tested.
- **Low-Pass Filter (LPF)**: With each trigger, the ADC conversion result is sent through a filter. When ADRPT samples have occurred, a threshold test is performed. Every trigger after that, the ADC conversion result is sent through the filter and another threshold test is performed.

The five modes are summarized in following table.

26.14.4 TMRxGATE

Name: TMRxGATE
Address: 0x210,0x216,0x21C

Timer Gate Source Selection Register

Bit	7	6	5	4	3	2	1	0
				GSS[4:0]				
Access				R/W	R/W	R/W	R/W	R/W
Reset				0	0	0	0	0

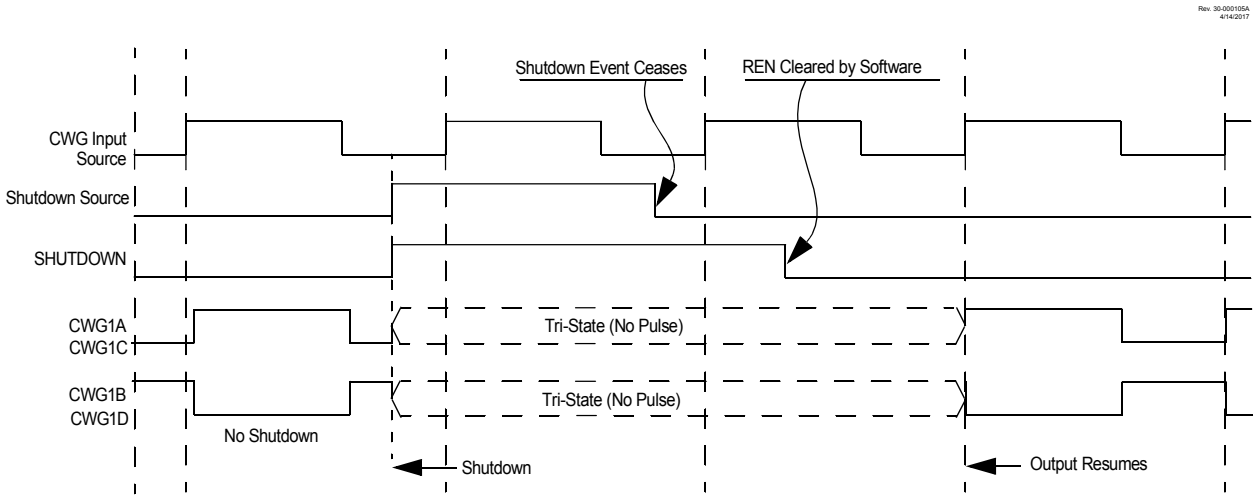
Bits 4:0 – GSS[4:0] Timer Gate Source Selection bits
Refer to the gate source selection table.

Reset States: POR/BOR = 00000
All Other Resets = uuuuu

Once all auto-shutdown sources are removed, the software must clear SHUTDOWN. Once SHUTDOWN is cleared, the CWG module will resume operation upon the first rising edge of the CWG data input.

Important: The SHUTDOWN bit cannot be cleared in software if the auto-shutdown condition is still present.

Figure 31-17. SHUTDOWN FUNCTIONALITY, AUTO-RESTART DISABLED (REN = 0, LSAC = 01, LSBD = 01)



31.11.2.2 Auto-Restart

When the REN bit is set (REN = 1), the CWG module will restart from the shutdown state automatically.

Once all auto-shutdown conditions are removed, the hardware will automatically clear SHUTDOWN. Once SHUTDOWN is cleared, the CWG module will resume operation upon the first rising edge of the CWG data input.

Important: The SHUTDOWN bit cannot be cleared in software if the auto-shutdown condition is still present.

31.15.7 CWGxAS1

Name: CWGxAS1
Address: 0x613,0x61D

CWG Auto-Shutdown Control Register 1

Bit	7	6	5	4	3	2	1	0
			AS5E	AS4E	AS3E	AS2E	AS1E	AS0E
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5 – ASyE CWG Auto-shutdown Source ASyE Enable bit(1)

Table 31-5. Shutdown Sources

ASyE	Source
AS6E	CLC2_out/CLC3_out (low causes shutdown)
AS5E	CMP2_out (low causes shutdown)
AS4E	CMP1_out (low causes shutdown)
AS3E	TMR6_postscaled (high causes shutdown)
AS2E	TMR4_postscaled (high causes shutdown)
AS1E	TMR2_postscaled (high causes shutdown)
AS0E	Pin selected by CWGxPPS (low causes shutdown)

Value	Description
1	Auto-shutdown for source ASyE is enabled
0	Auto-shutdown for source ASyE is disabled

Note: This bit may be written while EN = 0 (CWGxCON0), to place the outputs into the shutdown configuration.
The outputs will remain in auto-shutdown state until the next rising edge of the CWG data input after this bit is cleared.

35.5.6 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 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.

35.5.6.1 Normal Clock Stretching

Following an $\overline{ACK}$ if the R/W bit 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 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.

Important:

1. The BF bit has no effect on if the clock will be stretched or not. This is different than previous versions of the module that would not stretch the clock, clear CKP, if SSPxBUF was read before the ninth falling edge of SCL.
2. Previous versions of the module did not stretch the clock for a transmission if SSPxBUF was loaded before the ninth falling edge of SCL. It is now always cleared for read requests.

35.5.6.2 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.

Important: Previous versions of the module did not stretch the clock if the second address byte did not match.

35.5.6.3 Byte NACKing

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

35.5.7 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 the following figure).

36.5 Register Summary - EUSART

Offset	Name	Bit Pos.								
0x0119	RC1REG	7:0	RCREG[7:0]							
0x011A	TX1REG	7:0	TXREG[7:0]							
0x011B	SP1BRG	7:0	SPBRGL[7:0]							
		15:8	SPBRGH[7:0]							
0x011D	RC1STA	7:0	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
0x011E	TX1STA	7:0	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D
0x011F	BAUD1CON	7:0	ABDOVF	RCIDL		SCKP	BRG16		WUE	ABDEN

36.6 Register Definitions: EUSART Control

37.3.3 SMTxSTAT

Name: SMTxSTAT
Address: 0x049A

SMT Status Register

Bit	7	6	5	4	3	2	1	0
	CPRUP	CPWUP		RST		TS	WS	AS
Access	R/W/HC	R/W/HC		R/W		RO	RO	RO
Reset	0	0		0		0	0	0

Bit 7 – CPRUP SMT Manual Period Buffer Update bit

Value	Description
1	Request update to SMTxCPR registers
0	SMTxCPR registers update is complete

Bit 6 – CPWUP SMT Manual Pulse Width Buffer Update bit

Value	Description
1	Request update to SMTxCPW registers
0	SMTxCPW registers update is complete

Bit 4 – RST SMT Manual Timer Reset bit

Value	Description
1	Request Reset to SMTxTMR registers
0	SMTxTMR registers update is complete

Bit 2 – TS SMT GO Value Status bit

Value	Description
1	SMTxTMR is incrementing
0	SMTxTMR is not incrementing

Bit 1 – WS SMT Window Status bit

Value	Description
1	SMT window is open
0	SMT window is closed

Bit 0 – AS SMT Signal Value Status bit

Value	Description
1	SMT acquisition is in progress
0	SMT acquisition is not in progress

Offset	Name	Bit Pos.								
0x1EB8	MDCARLPPS	7:0				PORT[1:0]		PIN[2:0]		
0x1EB9	MDCARHPPS	7:0				PORT[1:0]		PIN[2:0]		
0x1EBA	MDSRCPPS	7:0				PORT[1:0]		PIN[2:0]		
0x1EBB	CLCIN1PPS	7:0				PORT[1:0]		PIN[2:0]		
0x1EBC	CLCIN2PPS	7:0				PORT[1:0]		PIN[2:0]		
0x1EBD	CLCIN3PPS	7:0				PORT[1:0]		PIN[2:0]		
0x1EBE	CLCIN4PPS	7:0				PORT[1:0]		PIN[2:0]		
0x1EBF ... 0x1EC2	Reserved									
0x1EC3	ADACTPPS	7:0				PORT[1:0]		PIN[2:0]		
0x1EC4	Reserved									
0x1EC5	SSP1CLKPPS	7:0				PORT[1:0]		PIN[2:0]		
0x1EC6	SSP1DATPPS	7:0				PORT[1:0]		PIN[2:0]		
0x1EC7	SSP1SSPPS	7:0				PORT[1:0]		PIN[2:0]		
0x1EC8	SSP2CLKPPS	7:0				PORT[1:0]		PIN[2:0]		
0x1EC9	SSP2DATPPS	7:0				PORT[1:0]		PIN[2:0]		
0x1ECA	SSP2SSPPS	7:0				PORT[1:0]		PIN[2:0]		
0x1ECB	RX1PPS	7:0				PORT[1:0]		PIN[2:0]		
0x1ECC	CK1PPS	7:0				PORT[1:0]		PIN[2:0]		
0x1ECD ... 0x1EFF	Reserved									
0x1F00	INDF0	7:0	INDF0[7:0]							
0x1F01	INDF1	7:0	INDF1[7:0]							
0x1F02	PCL	7:0	PCL[7:0]							
0x1F03	STATUS	7:0				$\overline{TO}$	$\overline{PD}$	Z	DC	C
0x1F04	FSR0	7:0	FSRL[7:0]							
		15:8	FSRH[7:0]							
0x1F06	FSR1	7:0	FSRL[7:0]							
		15:8	FSRH[7:0]							
0x1F08	BSR	7:0			BSR[5:0]					
0x1F09	WREG	7:0	WREG[7:0]							
0x1F0A	PCLATH	7:0		PCLATH[6:0]						
0x1F0B	INTCON	7:0	GIE	PEIE						INTEDG
0x1F0C ... 0x1F0F	Reserved									
0x1F10	RA0PPS	7:0			PPS[5:0]					
0x1F11	RA1PPS	7:0			PPS[5:0]					
0x1F12	RA2PPS	7:0			PPS[5:0]					
0x1F13	Reserved									
0x1F14	RA4PPS	7:0			PPS[5:0]					
0x1F15	RA5PPS	7:0			PPS[5:0]					
0x1F16 ...	Reserved									

MOVLW	Move literal to W			
Description:	The 8-bit literal ‘k’ is loaded into W register. The “don’t cares” will assemble as ‘0’s.			
Words:	1			
Cycles:	1			

Example:	MOVLW	5Ah
After Instruction W = 5Ah		

MOVWF	Move W to f			
Syntax:	[*label*] MOVWF f			
Operands:	0 ≤ f ≤ 127			
Operation:	(W) → f			
Status Affected:	None			
Description:	Move data from W to register ‘f’.			
Words:	1			
Cycles:	1			

Example:	MOVWF	LATA
Before Instruction LATA = FFh W = 4Fh After Instruction LATA = 4Fh W = 4Fh		

MOVWI	Move W to INDFn
Syntax:	[*label*] MOVWI ++FSRn [*label*] MOVWI --FSRn [*label*] MOVWI FSRn++ [*label*] MOVWI FSRn-- [*label*] MOVWI k[FSRn]
Operands:	n ∈ [0,1] mm ∈ [00,01,10,11]

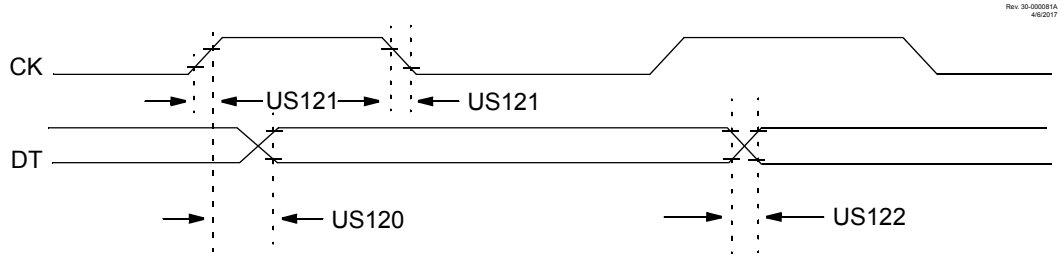
I/O and CLKOUT Timing Specifications

42.4.16 EUSART Synchronous Transmission Requirements

Table 42-22.

Standard Operating Conditions (unless otherwise stated)						
Param No.	**Sym.**	**Characteristic**	**Min.**	**Max.**	**Units**	**Conditions**
US120	$T_{CK}H2_{DT}V$	SYNC XMIT (Master and Slave)	—	80	ns	$3.0V \le V_{DD} \le 5.5V$
		Clock high to data-out valid	—	100	ns	$1.8V \le V_{DD} \le 5.5V$
US121	T_{CKRF}	Clock out rise time and fall time	—	45	ns	$3.0V \le V_{DD} \le 5.5V$
		(Master mode)	—	50	ns	$1.8V \le V_{DD} \le 5.5V$
US122	T_{DTRF}	Data-out rise time and fall time	—	45	ns	$3.0V \le V_{DD} \le 5.5V$
			—	50	ns	$1.8V \le V_{DD} \le 5.5V$

Figure 42-15. EUSART Synchronous Transmission (Master/Slave) Timing



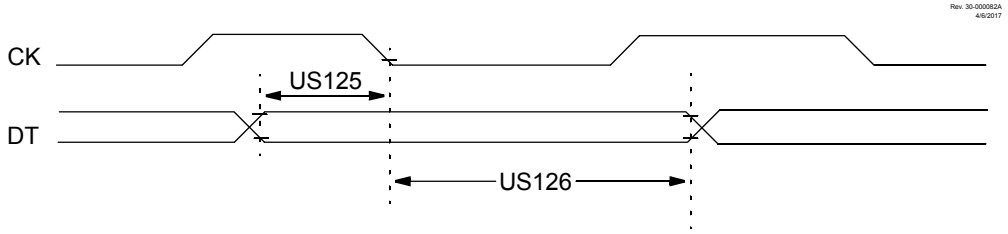
Note: Refer to Figure 42-4 for load conditions.

42.4.17 EUSART Synchronous Receive Requirements

Table 42-23.

Standard Operating Conditions (unless otherwise stated)						
Param No.	**Sym.**	**Characteristic**	**Min.**	**Max.**	**Units**	**Conditions**
US125	$T_{DT}V2_{CKL}$	SYNC RCV (Master and Slave)	10	—	ns	
		Data-setup before CK ↓ (DT hold time)				
US126	$T_{CK}L2_{DTL}$	Data-hold after CK ↓ (DT hold time)	15	—	ns	

Figure 42-16. EUSART Synchronous Receive (Master/Slave) Timing



Note: Refer to Figure 42-4 for load conditions.