



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	64MHz
Connectivity	I ² C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	60
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 45x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-VFQFN Exposed Pad
Supplier Device Package	64-VQFN (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf65k40-i-mr

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

8.14 Power Control (PCON0) Register

The Power Control (PCON0) register contains flag bits to differentiate between a:

- Brown-out Reset (BOR)
- Power-on Reset (POR)
- Reset Instruction Reset (RI)
- MCLR Reset (RMCLR)
- Watchdog Timer Reset (RWDT)
- Watchdog Window Violation (WDTWV)
- Stack Underflow Reset (STKUNF)
- Stack Overflow Reset (STKOVF)

The PCON0 register bits are shown in Register 8-2.

Hardware will change the corresponding register bit during the Reset process; if the Reset was not caused by the condition, the bit remains unchanged (Table 8-3).

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

Software may also set any PCON0 bit to the active state, so that user code may be tested, but no Reset action will be generated.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BORCON	SBOREN							BORRDY	73
PCON0	STKOVF	STKUNF	WDTWV	RWDT	RMCLR	RI	POR	BOR	74
STATUS	_	TO	PD	Ν	OV	Z	DC	С	121
WDTCON0	_	_		١	NDTPS<4:0	>		SEN	83
WDTCON1	_	V	/DTCS<2:0>	>		W	/INDOW<2:()>	84

TABLE 8-4: SUMMARY OF REGISTERS ASSOCIATED WITH RESETS

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

11.1.1 TABLE READS AND TABLE WRITES

In order to read and write program memory, there are two operations that allow the processor to move bytes between the program memory space and the data RAM:

- Table Read (TBLRD)
- Table Write (TBLWT)

The program memory space is 16 bits wide, while the data RAM space is eight bits wide. Table reads and table writes move data between these two memory spaces through an 8-bit register (TABLAT).

The table read operation retrieves one byte of data directly from program memory and places it into the TABLAT register. Figure 11-1 shows the operation of a table read.

The table write operation stores one byte of data from the TABLAT register into a write block holding register. The procedure to write the contents of the holding registers into program memory is detailed in **Section 11.1.6 "Writing to Program Flash Memory"**. Figure 11-2 shows the operation of a table write with program memory and data RAM.

Table operations work with byte entities. Tables containing data, rather than program instructions, are not required to be word aligned. Therefore, a table can start and end at any byte address. If a table write is being used to write executable code into program memory, program instructions will need to be word aligned.



FIGURE 11-2: TABLE WRITE OPERATION



U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	CCP5IE	CCP4IE	CCP3IE	CCP2IE	CCP1IE
bit 7						-	bit 0
Legend:							
R = Readable	bit	W = Writable	bit	U = Unimpler	mented bit, reac	l as '0'	
-n = Value at I	POR	'1' = Bit is set		'0' = Bit is cle	ared	x = Bit is unkr	nown
bit 7-5	Unimplemen	ted: Read as ')'				
bit 4	CCP5IE: ECC 1 = Enabled 0 = Disabled	CP5 Interrupt E	nable bit				
bit 3	CCP4IE: ECC 1 = Enabled 0 = Disabled	CP4 Interrupt E	nable bit				
bit 2	CCP3IE: ECC 1 = Enabled 0 = Disabled	CP3 Interrupt E	nable bit				
bit 1	CCP2IE: ECC 1 = Enabled 0 = Disabled	CP2 Interrupt E	nable bit				
bit 0	CCP1IE: ECC 1 = Enabled 0 = Disabled	CP1 Interrupt E	nable bit				

REGISTER 14-19: PIE7: PERIPHERAL INTERRUPT ENABLE REGISTER 7

REGISTER 1/	′-2: КХУРР	'5: PIN RXY C	JUIPUI SC	JURCE SELE	CTION REGIS	SIER	
U-0	U-0	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u
—	_			RxyPF	PS<5:0>		
bit 7							bit 0
Legend:							
R = Readable b	oit	W = Writable	bit	U = Unimplen	nented bit, read	as '0'	
u = Bit is uncha	anged	x = Bit is unkn	iown	-n/n = Value a	at POR and BO	R/Value at all c	other Resets
'1' = Bit is set		'0' = Bit is clea	ared				

CICTI

bit 7-6 Unimplemented: Read as '0'

20.5.3 EDGE-TRIGGERED HARDWARE LIMIT MODE

In Hardware Limit mode the timer can be reset by the TMRx_ers external signal before the timer reaches the period count. Three types of Resets are possible:

- Reset on rising or falling edge (MODE<4:0>= 00011)
- Reset on rising edge (MODE<4:0> = 00100)
- Reset on falling edge (MODE<4:0> = 00101)

FIGURE 20-6: EDGE-TRIGGERED HARDWARE LIMIT MODE TIMING DIAGRAM (MODE = 00100)



When the timer is used in conjunction with the CCP in PWM mode then an early Reset shortens the period and restarts the PWM pulse after a two clock delay. Refer to Figure 20-6.

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
			DCH	<7:0>			
bit 7							bit 0
Legend:							
R = Readable	bit	W = Writable b	oit	U = Unimplen	nented bit, read	l as '0'	
u = Bit is unch	nanged	x = Bit is unkn	own	-n/n = Value a	t POR and BO	R/Value at all o	other Resets
'1' = Bit is set		'0' = Bit is clea	ared				

REGISTER 22-3: PWMxDCH: PWM DUTY CYCLE HIGH BITS

bit 7-0 **DCh<7:0>:** PWM Duty Cycle Most Significant bits These bits are the MSbs of the PWM duty cycle. The two LSbs are found in PWMxDCL Register.

REGISTER 22-4: PWMxDCL: PWM DUTY CYCLE LOW BITS

R/W-x/u	R/W-x/u	U-0	U-0	U-0	U-0	U-0	U-0
DCL	<7:6>	—	—	—	—	—	—
bit 7							bit 0

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

bit 7-6 DC<8:9>: PWM Duty Cycle Least Significant bits These bits are the LSbs of the PWM duty cycle. The MSbs are found in PWMxDCH Register.

bit 5-0 Unimplemented: Read as '0'

24.12 Operation During Sleep

The CWG module operates independently from the system clock and will continue to run during Sleep, provided that the clock and input sources selected remain active.

The HFINTOSC remains active during Sleep when all the following conditions are met:

- CWG module is enabled
- · Input source is active
- HFINTOSC is selected as the clock source, regardless of the system clock source selected.

In other words, if the HFINTOSC is simultaneously selected as the system clock and the CWG clock source, when the CWG is enabled and the input source is active, then the CPU will go idle during Sleep, but the HFINTOSC will remain active and the CWG will continue to operate. This will have a direct effect on the Sleep mode current.

24.13 Configuring the CWG

- Ensure that the TRIS control bits corresponding to CWG outputs are set so that all are configured as inputs, ensuring that the outputs are inactive during setup. External hardware should ensure that pin levels are held to safe levels.
- 2. Clear the EN bit, if not already cleared.
- Configure the MODE<2:0> bits of the CWG1CON0 register to set the output operating mode.
- 4. Configure the POLy bits of the CWG1CON1 register to set the output polarities.
- 5. Configure the ISM<3:0> bits of the CWG1ISM register to select the data input source.
- 6. If a steering mode is selected, configure the STRx bits to select the desired output on the CWG outputs.
- Configure the LSBD<1:0> and LSAC<1:0> bits of the CWG1ASD0 register to select the autoshutdown output override states (this is necessary even if not using auto-shutdown because start-up will be from a shutdown state).
- 8. If auto-restart is desired, set the REN bit of CWG1AS0.
- 9. If auto-shutdown is desired, configure the ASxE bits of the CWG1AS1 register to select the shutdown source.
- 10. Set the desired rising and falling dead-band times with the CWG1DBR and CWG1DBF registers.
- 11. Select the clock source in the CWG1CLKCON register.
- 12. Set the EN bit to enable the module.
- 13. Clear the TRIS bits that correspond to the CWG outputs to set them as outputs.

If auto-restart is to be used, set the REN bit and the SHUTDOWN bit will be cleared automatically. Otherwise, clear the SHUTDOWN bit in software to start the CWG.

27.10.13.1 Bus Collision During a Start Condition

During a Start condition, a bus collision occurs if:

- a) SDA or SCL are sampled low at the beginning of the Start condition (Figure 27-33).
- b) SCL is sampled low before SDA is asserted low (Figure 27-34).

During a Start condition, both the SDA and the SCL pins are monitored.

If the SDA pin is already low, or the SCL pin is already low, then all of the following occur:

- · the Start condition is aborted,
- the BCLxIF flag is set and
- the MSSP module is reset to its Idle state (Figure 27-33).

The Start condition begins with the SDA and SCL pins deasserted. When the SDA pin is sampled high, the Baud Rate Generator is loaded and counts down. If the SCL pin is sampled low while SDA is high, a bus collision occurs because it is assumed that another master is attempting to drive a data '1' during the Start condition.

If the SDA pin is sampled low during this count, the BRG is reset and the SDA line is asserted early (Figure 27-35). If, however, a '1' is sampled on the SDA pin, the SDA pin is asserted low at the end of the BRG count. The Baud Rate Generator is then reloaded and counts down to zero; if the SCL pin is sampled as '0' during this time, a bus collision does not occur. At the end of the BRG count, the SCL pin is asserted low.

Note: The reason that bus collision is not a factor during a Start condition is that no two bus masters can assert a Start condition at the exact same time. Therefore, one master will always assert SDA before the other. This condition does not cause a bus collision because the two masters must be allowed to arbitrate the first address following the Start condition. If the address is the same, arbitration must be allowed to continue into the data portion, Repeated Start or Stop conditions.





28.4.4 BREAK CHARACTER SEQUENCE

The EUSART module has the capability of sending the special Break character sequences that are required by the LIN bus standard. A Break character consists of a Start bit, followed by 12 '0' bits and a Stop bit.

To send a Break character, set the SENDB and TXEN bits of the TXxSTA register. The Break character transmission is then initiated by a write to the TXxREG. The value of data written to TXxREG will be ignored and all '0's will be transmitted.

The SENDB bit is automatically reset by hardware after the corresponding Stop bit is sent. This allows the user to preload the transmit FIFO with the next transmit byte following the Break character (typically, the Sync character in the LIN specification).

The TRMT bit of the TXxSTA register indicates when the transmit operation is active or idle, just as it does during normal transmission. See Figure 28-9 for the timing of the Break character sequence.

28.4.4.1 Break and Sync Transmit Sequence

The following sequence will start a message frame header made up of a Break, followed by an auto-baud Sync byte. This sequence is typical of a LIN bus master.

- 1. Configure the EUSART for the desired mode.
- 2. Set the TXEN and SENDB bits to enable the Break sequence.
- 3. Load the TXxREG with a dummy character to initiate transmission (the value is ignored).
- 4. Write '55h' to TXxREG to load the Sync character into the transmit FIFO buffer.
- 5. After the Break has been sent, the SENDB bit is reset by hardware and the Sync character is then transmitted.

When the TXxREG becomes empty, as indicated by the TXxIF, the next data byte can be written to TXxREG.

28.4.5 RECEIVING A BREAK CHARACTER

The Enhanced EUSART module can receive a Break character in two ways.

The first method to detect a Break character uses the FERR bit of the RCxSTA register and the received data as indicated by RCxREG. The Baud Rate Generator is assumed to have been initialized to the expected baud rate.

A Break character has been received when;

- RCxIF bit is set
- FERR bit is set
- RCxREG = 00h

The second method uses the Auto-Wake-up feature described in **Section 28.4.3 "Auto-Wake-up on Break"**. By enabling this feature, the EUSART will sample the next two transitions on RX/DT, cause an RCxIF interrupt, and receive the next data byte followed by another interrupt.

Note that following a Break character, the user will typically want to enable the Auto-Baud Detect feature. For both methods, the user can set the ABDEN bit of the BAUDxCON register before placing the EUSART in Sleep mode.



FIGURE 28-9: SEND BREAK CHARACTER SEQUENCE

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
BAUDxCON	ABDOVF	RCIDL		SCKP	BRG16	—	WUE	ABDEN	451
INTCON	GIE/GIEH	PEIE/GIEL	IPEN	-	INT3EDG	INT2EDG	INT1EDG	INT0EDG	173
PIE3	RC2IE	TX2IE	RC1IE	TX1IE	BCL2IE	SSP2IE	BCL1IE	SSP1IE	188
PIR3	RC2IF	TX2IF	RC1IF	TX1IF	BCL2IF	SSP2IF	BCL1IF	SSP1IF	177
IPR3	RC2IP	TX2IP	RC1IP	TX1IP	BCL2IP	SSP2IP	BCL1IP	SSP1IP	198
PIE4	_	_	RC5IE	TX5IE	RC4IE	TX4IE	RC3IE	TX3IE	189
PIR4	_	—	RC5IF	TX5IF	RC4IF	TX4IF	RC3IF	TX3IF	178
IPR4	_	_	RC5IP	TX5IP	RC4IP	TX4IP	RC3IP	TX3IP	199
RCxSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	450
RxyPPS	_	—			RxyPP	S<5:0>			228
TXxPPS	_	—			TXPP	S<5:0>			225
TXxREG			EUSA	RTx Transm	it Data Regis	ter			452*
TXxSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	449

SUMMARY OF REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE **TABLE 28-9**: TRANSMISSION

Legend: — = unimplemented location, read as '0'. Shaded cells are not used for synchronous slave transmission.

Page provides register information.

FIGURE 32-9: DIFFERENTIAL CVD WITH GUARD RING OUTPUT WAVEFORM





Precharge Time 1-255 TINST (TPRE)	Acquisition/ Sharing Time 1-255 TINST (TACQ)	 		TAD2	(Tradii	Co tional	nversi Fiming	on Tim of AD			on)	Tap10	TAD11
	1			b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
External and Internal Channels are charged/discharged	External and Internal Channels share charge	Holdin	I Convers ig capac	sion sta citor Cr	arts IOLD is	discon	nected	I from a	analog	input (†	typicall	ly 100 r	ıs)
If ADPRE ≠ 0	If ADACQ ≠ 0		If ADPRE = 0				On th	e follov	wing cy	ycle:			
et GO/DONE bit		If ADAC (Traditic	;Q = 0 mal Ope	eration	Start)		aadf Adi <u>f</u> Go/D	RES0H <u>bit is</u> s OONE b	:AADF et, oit is cl	RES0L eared	is load	ed,	

32.4.5 ADDITIONAL SAMPLE AND HOLD CAPACITANCE

Additional capacitance can be added in parallel with the internal sample and hold capacitor (CHOLD) by using the ADCAP register. This register selects a digitally programmable capacitance which is added to the ADC conversion bus, increasing the effective internal capacitance of the sample and hold capacitor in the ADC module. This is used to improve the match between internal and external capacitance for a better sensing performance. The additional capacitance does not affect analog performance of the ADC because it is not connected during conversion. See Figure 32-11.

32.5.5 BURST AVERAGE MODE

The Burst Average mode (ADMD = 011) acts the same as the Average mode in most respects. The one way it differs is that it continuously retriggers ADC sampling until the ADCNT value is greater than or equal to ADRPT, even if Continuous Sampling mode (see **Section 32.5.8 "Continuous Sampling mode"**) is not enabled. This allows for a threshold comparison on the average of a short burst of ADC samples.

32.5.6 LOW-PASS FILTER MODE

The Low-pass Filter mode (ADMD = 100) acts similarly to the Average mode in how it handles samples (accumulates samples until ADCNT value greater than or equal to ADRPT, then triggers threshold comparison), but instead of a simple average, it performs a low-pass filter operation on all of the samples, reducing the effect of high-frequency noise on the average, then performs a threshold comparison on the results. (see Table 32-3 for a more detailed description of the mathematical operation). In this mode, the ADCRS bits determine the cut-off frequency of the low-pass filter (as demonstrated by Table 32-4).

32.5.7 THRESHOLD COMPARISON

At the end of each computation:

- The conversion results are latched and held stable at the end-of-conversion.
- The error is calculated based on a difference calculation which is selected by the ADCALC<2:0> bits in the ADCON3 register. The value can be one of the following calculations (see Register 32-4 for more details):
 - The first derivative of single measurements
 - The CVD result in CVD mode
 - The current result vs. a setpoint
 - The current result vs. the filtered/average result
 - The first derivative of the filtered/average value
 - Filtered/average value vs. a setpoint
- The result of the calculation (ADERR) is compared to the upper and lower thresholds, ADUTH<ADUTHH:ADUTHL> and ADLTH<ADLTHH:ADLTHL> registers, to set the ADUTHR and ADLTHR flag bits. The threshold logic is selected by ADTMD<2:0> bits in the ADCON3 register. The threshold trigger option can be one of the following:
 - Never interrupt
 - Error is less than lower threshold
 - Error is greater than or equal to lower threshold
 - Error is between thresholds (inclusive)
 - Error is outside of thresholds
 - Error is less than or equal to upper threshold
 - Error is greater than upper threshold
 - Always interrupt regardless of threshold test results
 - If the threshold condition is met, the threshold interrupt flag ADTIF is set.

Note 1: The threshold tests are signed operations.2: If ADAOV is set, a threshold interrupt is

 If ADAOV is set, a threshold interrupt is signaled.



34.5 Applications

In many applications, it is desirable to detect a drop below, or rise above, a particular voltage threshold. For example, the HLVD module could be periodically enabled to detect Universal Serial Bus (USB) attach or detach. This assumes the device is powered by a lower voltage source than the USB when detached. An attach would indicate a High-Voltage Detect from, for example, 3.3V to 5V (the voltage on USB) and vice versa for a detach. This feature could save a design a few extra components and an attach signal (input pin).

For general battery applications, Figure 34-4 shows a possible voltage curve. Over time, the device voltage decreases. When the device voltage reaches voltage, VA, the HLVD logic generates an interrupt at time, TA. The interrupt could cause the execution of an Interrupt Service Routine (ISR), which would allow the application to perform "housekeeping tasks" and a controlled shutdown before the device voltage exits the valid operating range at TB. This would give the application a time window, represented by the difference between TA and TB, to safely exit.



© 2016-2017 Microchip Technology Inc.

Field	Description
a	RAM access bit a = 0: RAM location in Access RAM (BSR register is ignored) a = 1: RAM bank is specified by BSR register
bbb	Bit address within an 8-bit file register (0 to 7).
BSR	Bank Select Register. Used to select the current RAM bank.
C, DC, Z, OV, N	ALU Status bits: Carry, Digit Carry, Zero, Overflow, Negative.
d	Destination select bit d = 0: store result in WREG d = 1: store result in file register f
dest	Destination: either the WREG register or the specified register file location.
f	8-bit Register file address (00h to FFh) or 2-bit FSR designator (0h to 3h).
fs	12-bit Register file address (000h to FFFh). This is the source address.
fd	12-bit Register file address (000h to FFFh). This is the destination address.
GIE	Global Interrupt Enable bit.
k	Literal field, constant data or label (may be either an 8-bit, 12-bit or a 20-bit value).
label	Label name.
mm	The mode of the TBLPTR register for the table read and table write instructions. Only used with table read and table write instructions:
*	No change to register (such as TBLPTR with table reads and writes)
*+	Post-Increment register (such as TBLPTR with table reads and writes)
*_	Post-Decrement register (such as TBLPTR with table reads and writes)
+*	Pre-Increment register (such as TBLPTR with table reads and writes)
n	The relative address (2's complement number) for relative branch instructions or the direct address for CALL/BRANCH and RETURN instructions.
PC	Program Counter.
PCL	Program Counter Low Byte.
PCH	Program Counter High Byte.
PCLATH	Program Counter High Byte Latch.
PCLATU	Program Counter Upper Byte Latch.
PD	Power-down bit.
PRODH	Product of Multiply High Byte.
PRODL	Product of Multiply Low Byte.
s	Fast Call/Return mode select bit s = 0: do not update into/from shadow registers s = 1: certain registers loaded into/from shadow registers (Fast mode)
TBLPTR	21-bit Table Pointer (points to a Program Memory location).
TABLAT	8-bit Table Latch.
TO	Time-out bit.
TOS	Top-of-Stack.
u	Unused or unchanged.
WDT	Watchdog Timer.
WREG	Working register (accumulator).
x	Don't care ('0' or '1'). The assembler will generate code with $x = 0$. It is the recommended form of use for compatibility with all Microchip software tools.
zs	7-bit offset value for indirect addressing of register files (source).
zd	7-bit offset value for indirect addressing of register files (destination).
{ }	Optional argument.
[text]	Indicates an indexed address.
(text)	The contents of text.
[expr] <n></n>	Specifies bit n of the register indicated by the pointer expr.
\rightarrow	Assigned to.
< >	Register bit field.
e	In the set of.
italics	User defined term (font is Courier).

TABLE 36-1: OPCODE FIELD DESCRIPTIONS

PIC18(L)F65/66K40

ΒZ		Branch if	if Zero						
Synta	ax:	BZ n							
Oper	ands:	-128 ≤ n ≤ ′	127						
Oper	ation:	if ZERO bit (PC) + 2 + 2	if ZERO bit is '1' (PC) + 2 + 2n \rightarrow PC						
Statu	s Affected:	None							
Enco	ding:	1110	0000 nr	nn nnnn					
Desc	ription:	If the ZERC will branch. The 2's cor added to th have increr instruction, PC + 2 + 2 2-cycle inst	If the ZERO bit is '1', then the program will branch. The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 2 + 2n. This instruction is then a 2-cycle instruction.						
Word	ls:	1							
Cycle	es:	1(2)							
Q C If Ju	ycle Activity: mp:								
	Q1	Q2	Q3	Q4					
	Decode	Read literal 'n'	Process Data	Write to PC					
	No operation	No operation	No operation	No operation					
lf No	o Jump:								
	Q1	Q2	Q3	Q4					
	Decode	Read literal 'n'	Process Data	No operation					
<u>Exan</u>	<u>nple</u> :	HERE	BZ Jump	þ					
	Before Instruc PC After Instructio If ZERO PC If ZERO PC	tion = ad on = 1; = ad = 0; = ad	dress (HERE dress (Jump dress (HERE	;))) ; + 2)					

Syntax:											
		CALL K {,	CALL K {,S}								
Operan	ds:	0 ≤ k ≤ 104 s ∈ [0,1]	8575								
Operatio	on:	$\begin{array}{l} (PC) + 4 \rightarrow \\ k \rightarrow PC < 2i \\ \text{if } s = 1 \\ (W) \rightarrow WS \\ (Status) \rightarrow \\ (BSR) \rightarrow E \end{array}$	$\begin{array}{l} (\text{PC}) + 4 \rightarrow \text{TOS}, \\ k \rightarrow \text{PC}{<}20{:}1{>}, \\ \text{if s} = 1 \\ (\text{W}) \rightarrow \text{WS}, \\ (\text{Status}) \rightarrow \text{STATUSS}, \\ (\text{BSR}) \rightarrow \text{BSRS} \end{array}$								
Status A	ffected:	None									
Encodin 1st word 2nd wor	ig: d (k<7:0>) d(k<19:8>)	1110 1111	110s k ₁₉ kkk	k ₇ kk kkk	k k k k	kkk kkk					
		(PC + 4) is stack. If 's' registers a respective STATUSS update occ 20-bit value CALL is a	pushed of = 1, the ¹ re also pu shadow r and BSR curs (defa e 'k' is loa 2-cycle ir	onto th W, Stat ushed i registe S. If 's out). Th ded in nstructi	e retur tus and into the rs, WS ' = 0, n nen, the to PC< ion.	n I BSI eir o e 20:1					
Words:		2									
Cycles:		2									
Q Cycl	e Activity:										
	Q1	Q2	Q3	3	Q	4					
	Decode	Read literal	PUSH PUSH P	PC to k	Read 'k'<1	litera					
		'k′<7:0>,			Write	9:8>, to P(
	No	[•] k'<7:0>, No	No)	Write N	9:8>, to P(o					
(No	[·] k'<7:0>, No operation	No opera	tion	Write N opera	9:8>, to P(o ation					

fter Insi	truction			
PC	=	address	(THERE)	
TOS	S =	address	(HERE +	4)
WS	=	W		
BSF	RS =	BSR		
STA	TUSS =	Status		

37.2 MPLAB XC Compilers

The MPLAB XC Compilers are complete ANSI C compilers for all of Microchip's 8, 16, and 32-bit MCU and DSC devices. These compilers provide powerful integration capabilities, superior code optimization and ease of use. MPLAB XC Compilers run on Windows, Linux or MAC OS X.

For easy source level debugging, the compilers provide debug information that is optimized to the MPLAB X IDE.

The free MPLAB XC Compiler editions support all devices and commands, with no time or memory restrictions, and offer sufficient code optimization for most applications.

MPLAB XC Compilers include an assembler, linker and utilities. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. MPLAB XC Compiler uses the assembler to produce its object file. Notable features of the assembler include:

- · Support for the entire device instruction set
- · Support for fixed-point and floating-point data
- Command-line interface
- · Rich directive set
- · Flexible macro language
- MPLAB X IDE compatibility

37.3 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel[®] standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code, and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB X IDE projects
- User-defined macros to streamline
 assembly code
- Conditional assembly for multipurpose source files
- Directives that allow complete control over the assembly process

37.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

37.5 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC DSC devices. MPLAB XC Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- · Support for the entire device instruction set
- · Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

TABLE 38-6: THERMAL CHARACTERISTICS

Standard Operating Conditions (unless otherwise stated)								
Param No.	Sym.	Characteristic	Тур.	Units	Conditions			
TH01	θJA	Thermal Resistance Junction to Ambient	59	°C/W	64-pin TQFP package			
			28	°C/W	64-pin QFN package			
TH02	θJA	Thermal Resistance Junction to Case	18	°C/W	64-pin TQFP package			
			6	°C/W	64-pin QFN package			
TH03	Тјмах	Maximum Junction Temperature	150	°C				
TH04	PD	Power Dissipation	_	W	PD = PINTERNAL + PI/O ⁽³⁾			
TH05	PINTERNAL	Internal Power Dissipation	_	W	PINTERNAL = IDD x VDD ⁽¹⁾			
TH06	Pi/o	I/O Power Dissipation	_	W	$PI/O = \Sigma (IOL * VOL) + \Sigma (IOH * (VDD - VOH))$			
TH07	PDER	Derated Power	_	W	Pder = PDmax (Τj - Τa)/θja ⁽²⁾			

Note 1: IDD is current to run the chip alone without driving any load on the output pins.

2: TA = Ambient Temperature, TJ = Junction Temperature

3: See absolute maximum ratings for total power dissipation.



FIGURE 38-17: SPI MASTER MODE TIMING (CKE = 1, SMP = 1)



Standard Operating Conditions (unless otherwise stated)							
Param. No.	Symbol	Characteristic		Min.	Max.	Units	Conditions
SP100*	Тнідн	Clock high time	100 kHz mode	4.0	—	μS	Device must operate at a minimum of 1.5 MHz
			400 kHz mode	0.6	—	μS	Device must operate at a minimum of 10 MHz
			SSP module	1.5Tcy	_		
SP101*	TLOW	Clock low time	100 kHz mode	4.7	—	μs	Device must operate at a minimum of 1.5 MHz
			400 kHz mode	1.3	—	μS	Device must operate at a minimum of 10 MHz
			SSP module	1.5Tcy	_		
SP102*	TR	SDA and SCL rise time	100 kHz mode	—	1000	ns	
			400 kHz mode	20 + 0.1CB	300	ns	CB is specified to be from 10-400 pF
SP103*	TF	SDA and SCL fall time	100 kHz mode	—	250	ns	
			400 kHz mode	20 + 0.1CB	250	ns	CB is specified to be from 10-400 pF
SP106*	THD:DAT	Data input hold time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	μs	
SP107*	TSU:DAT	Data input setup time	100 kHz mode	250	_	ns	(Note 2)
			400 kHz mode	100	_	ns	
SP109*	ΤΑΑ	Output valid from clock	100 kHz mode	—	3500	ns	(Note 1)
			400 kHz mode	—		ns	
SP110*	TBUF	Bus free time	100 kHz mode	4.7		μS	Time the bus must be free
			400 kHz mode	1.3	—	μS	before a new transmission can start
SP111	Св	Bus capacitive loading		—	400	pF	

TABLE 38-25: I²C BUS DATA REQUIREMENTS

* These parameters are characterized but not tested.

Note 1: As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCL to avoid unintended generation of Start or Stop conditions.

2: A Fast mode (400 kHz) I²C bus device can be used in a Standard mode (100 kHz) I²C bus system, but the requirement Tsu:DAT ≥ 250 ns must then be met. This will automatically be the case if the device does not stretch the low period of the SCL signal. If such a device does stretch the low period of the SCL signal, it must output the next data bit to the SDA line TR max. + Tsu:DAT = 1000 + 250 = 1250 ns (according to the Standard mode I²C bus specification), before the SCL line is released.

APPENDIX A: REVISION HISTORY

Revision A (6/2016)

Initial release of this document.

Revision B (9/2016)

Updated Peripheral Module, Memory and Core features descriptions on cover page. Updated the PIC18(L)F2x/4xK40 Family Types Table. Updated Examples 11-1, 11-3, 11-5 and 11-6; Registers 4-2, 4-5 and 13-18; Sections 1.2, 4.4.1, 4.5, 4.5.4, 17.3, 17.5, 18.1, 18.1.1, and 18.1.1.1; Tables 4-2, 38-5 and 38-14.

Revision C (4/2017)

Updated Cover page. Updated Example 13-1; Figures 6-1 and 11-11; Registers 3-3, 3-6, 19-1, and 27-9; Sections 1.1.2, 4.3, 13.8, 23.5, 27.5.1, 27.10, 32.1.2, and 32.1.6; Tables 4-1, 10-5, 38-11 and 38-15.

New Timer 2 chapter.

Removed Section 4.4.2 and 31.2.3.

Added Section 23.5.1