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	40MHz
Connectivity	CANbus, I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	68
Program Memory Size	32KB (16K x 16)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	1.5K x 8
Voltage - Supply (Vcc/Vdd)	4.2V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	84-LCC (J-Lead)
Supplier Device Package	84-PLCC (29.31x29.31)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18c858-i-l

1.0 DEVICE OVERVIEW

This document contains device specific information for the following three devices:

1. PIC18C658
2. PIC18C858

The PIC18C658 is available in 64-pin TQFP and 68-pin PLCC packages. The PIC18C858 is available in 80-pin TQFP and 84-pin PLCC packages.

An overview of features is shown in Table 1-1.

The following two figures are device block diagrams sorted by pin count; 64/68-pin for Figure 1-1 and 80/84-pin for Figure 1-2. The 64/68-pin and 80/84-pin pinouts are listed in Table 1-2.

TABLE 1-1: DEVICE FEATURES

Features			PIC18C658	PIC18C858
Operating Frequency			DC - 40 MHz	DC - 40 MHz
Program Memory	Internal	Bytes	32 K	32 K
		# of Single word Instructions	16384	16384
Data Memory (Bytes)			1536	1536
Interrupt sources			21	21
I/O Ports			Ports A – G	Ports A – H, J, K
Timers			4	4
Capture/Compare/PWM modules			2	2
Serial Communications			MSSP, CAN Addressable USART	MSSP, CAN Addressable USART
Parallel Communications			PSP	PSP
10-bit Analog-to-Digital Module			12 input channels	16 input channels
Analog Comparators			2	2
RESETS (and Delays)			POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST)
Programmable Low Voltage Detect			Yes	Yes
Programmable Brown-out Reset			Yes	Yes
CAN Module			Yes	Yes
In-Circuit Serial Programming (ICSP™)			Yes	Yes
Instruction Set			75 Instructions	75 Instructions
Packages			64-pin TQFP 68-pin CERQUAD (Windowed) 68-pin PLCC	80-pin TQFP 84-pin CERQUAD (Windowed) 84-pin PLCC

TABLE 1-2: PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number				Pin Type	Buffer Type	Description
	PIC18C658		PIC18C858				
	TQFP	PLCC	TQFP	PLCC			
							PORTA is a bi-directional I/O port
RA0/AN0	24	34	30	42			
RA0					I/O	TTL	Digital I/O
AN0					I	Analog	Analog input 0
RA1/AN1	23	33	29	41			
RA1					I/O	TTL	Digital I/O
AN1					I	Analog	Analog input 1
RA2/AN2/VREF-	22	32	28	40			
RA2					I/O	TTL	Digital I/O
AN2					I	Analog	Analog input 2
VREF-					I	Analog	A/D reference voltage (Low) input
RA3/AN3/VREF+	21	31	27	39			
RA3					I/O	TTL	Digital I/O
AN3					I	Analog	Analog input 3
VREF+					I	Analog	A/D reference voltage (High) input
RA4/T0CKI	28	39	34	47			
RA4					I/O	ST/OD	Digital I/O – Open drain when configured as output
T0CKI					I	ST	Timer0 external clock input
RA5/AN4/$\overline{SS}$/LVDIN	27	38	33	46			
RA5					I/O	TTL	Digital I/O
AN4					I	Analog	Analog input 4
$\overline{SS}$					I	ST	SPI slave select input
LVDIN					I	Analog	Low voltage detect input
RA6							See the OSC2/CLKO/RA6 pin

Legend: TTL = TTL compatible input
ST = Schmitt Trigger input with CMOS levels
I = Input
P = Power
CMOS = CMOS compatible input or output
Analog = Analog input
O = Output
OD = Open Drain (no P diode to VDD)

TABLE 1-2: PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number				Pin Type	Buffer Type	Description
	PIC18C658		PIC18C858				
	TQFP	PLCC	TQFP	PLCC			
							PORTC is a bi-directional I/O port
RC0/T1OSO/T13CKI	30	41	36	49			
RC0					I/O	ST	Digital I/O
T1OSO					O	—	Timer1 oscillator output
T13CKI					I	ST	Timer1/Timer3 external clock input
RC1/T1OSI	29	40	35	48			
RC1					I/O	ST	Digital I/O
T1OSI					I	CMOS	Timer1 oscillator input
RC2/CCP1	33	44	43	56			
RC2					I/O	ST	Digital I/O
CCP1					I/O	ST	Capture1 input/Compare1 output/PWM1 output
RC3/SCK/SCL	34	45	44	57			
RC3					I/O	ST	Digital I/O
SCK					I/O	ST	Synchronous serial clock input/output for SPI mode
SCL					I/O	ST	Synchronous serial clock input/output for I^2C mode
RC4/SDI/SDA	35	46	45	58			
RC4					I/O	ST	Digital I/O
SDI					I	ST	SPI data in
SDA					I/O	ST	I^2C data I/O
RC5/SDO	36	47	46	59			
RC5					I/O	ST	Digital I/O
SDO					O	—	SPI data out
RC6/TX/CK	31	42	37	50			
RC6					I/O	ST	Digital I/O
TX					O	—	USART asynchronous transmit
CK					I/O	ST	USART synchronous clock (See RX/DT)
RC7/RX/DT	32	43	38	51			
RC7					I/O	ST	Digital I/O
RX					I	ST	USART asynchronous receive
DT					I/O	ST	USART synchronous data (See TX/CK)

Legend: TTL = TTL compatible input
ST = Schmitt Trigger input with CMOS levels
I = Input
P = Power
CMOS = CMOS compatible input or output
Analog = Analog input
O = Output
OD = Open Drain (no P diode to V_{DD})

2.6 Oscillator Switching Feature

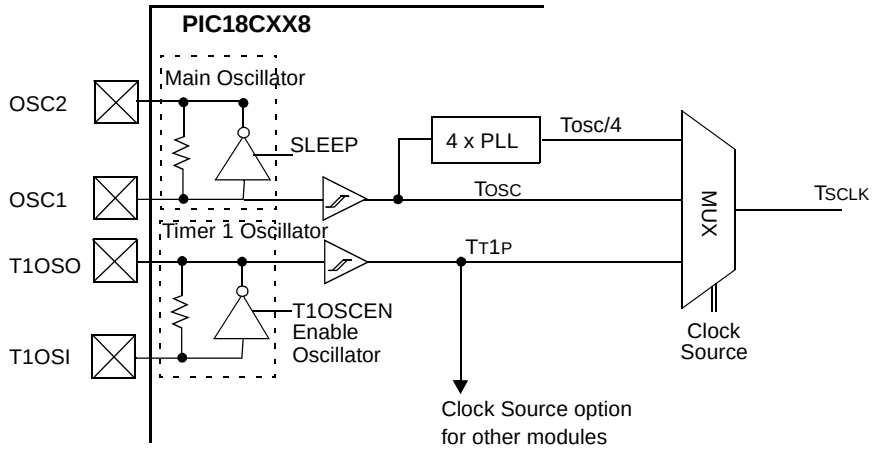
The PIC18CXX8 devices include a feature that allows the system clock source to be switched from the main oscillator to an alternate low frequency clock source. For the PIC18CXX8 devices, this alternate clock source is the Timer1 oscillator. If a low frequency crystal (32 kHz, for example) has been attached to the Timer1 oscillator pins and the Timer1 oscillator has been enabled, the device can switch to a low power execution mode. Figure 2-6 shows a block diagram of the system clock sources. The clock switching feature is enabled by programming the Oscillator Switching Enable (OSCSEN) bit in Configuration register CONFIG1H to a '0'. Clock switching is disabled in an erased device. See Section 9 for further details of the Timer1 oscillator. See Section 22.0 for Configuration Register details.

2.6.1 SYSTEM CLOCK SWITCH BIT

The system clock source switching is performed under software control. The system clock switch bit, SCS (OSCCON register), controls the clock switching. When the SCS bit is '0', the system clock source comes from the main oscillator selected by the FOSC2:FOSC0 configuration bits. When the SCS bit is set, the system clock source will come from the Timer1 oscillator. The SCS bit is cleared on all forms of RESET.

Note:	The Timer1 oscillator must be enabled to switch the system clock source. The Timer1 oscillator is enabled by setting the T1OSCEN bit in the Timer1 control register (T1CON). If the Timer1 oscillator is not enabled, any write to the SCS bit will be ignored (SCS bit forced cleared) and the main oscillator will continue to be the system clock source.

FIGURE 2-6: DEVICE CLOCK SOURCES



Note: I/O pins have diode protection to VDD and VSS.

REGISTER 2-1: OSCCON REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-1
—	—	—	—	—	—	—	SCS
bit 7							bit 0

bit 7-1 **Unimplemented:** Read as '0'

bit 0 **SCS:** System Clock Switch bit
when OSCSEN configuration bit = '0' and T1OSCEN bit is set:
1 = Switch to Timer1 Oscillator/Clock pin
0 = Use primary Oscillator/Clock input pin
when OSCSEN is clear or T1OSCEN is clear:
bit is forced clear

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

TABLE 3-3: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices		Power-on Reset, Brown-out Reset	$\overline{\text{MCLR}}$ Reset WDT Reset RESET Instruction Stack Resets	Wake-up via WDT or Interrupt
CANCON	658	858	xxxx xxx-	uuuu uuu-	uuuu uuu-
CANSTAT	658	858	xxx- xxx-	uuu- uuu-	uuu- uuu-
RXB0D7	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0D6	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0D5	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0D4	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0D3	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0D2	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0D1	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0D0	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0DLC	658	858	0xxx xxxx	0uuu uuuu	uuuu uuuu
RXB0EIDL	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0EIDH	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0SIDL	658	858	xxxx x-xx	uuuu u-uu	uuuu u-uu
RXB0SIDH	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB0CON	658	858	000- 0000	000- 0000	uuu- uuuu
RXB1D7	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1D6	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1D5	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1D4	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1D3	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1D2	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1D1	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1D0	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1DLC	658	858	0xxx xxxx	0uuu uuuu	uuuu uuuu
RXB1EIDL	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1EIDH	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1SIDL	658	858	xxxx x0xx	uuuu u0uu	uuuu uuuu
RXB1SIDH	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXB1CON	658	858	0000 0000	0000 0000	uuuu uuuu
TXB0D7	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB0D6	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB0D5	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB0D4	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB0D3	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB0D2	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB0D1	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition

Note 1: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

2: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

3: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.

4: See Table 3-2 for RESET value for specific condition.

5: Bit 6 of PORTA, LATA, and TRISA are enabled in ECIO and RCIO oscillator modes only. In all other oscillator modes, they are disabled and read '0'.

6: The long write enable is only reset on a POR or $\overline{\text{MCLR}}$.

7: Available on PIC18C858 only.

TABLE 3-3: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices		Power-on Reset, Brown-out Reset	$\overline{\text{MCLR}}$ Reset WDT Reset RESET Instruction Stack Resets	Wake-up via WDT or Interrupt
TXB0D0	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB0DLC	658	858	0x00 xxxx	0u00 uuuu	uuuu uuuu
TXB0EIDL	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB0EIDH	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB0SIDL	658	858	xxx0 x0xx	uuu0 u0uu	uuuu uuuu
TXB0SIDH	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB0CON	658	858	0000 0000	0000 0000	uuuu uuuu
TXB1D7	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB1D6	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB1D5	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB1D4	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB1D3	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB1D2	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB1D1	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB1D0	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB1DLC	658	858	0x00 xxxx	0u00 uuuu	uuuu uuuu
TXB1EIDL	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB1EIDH	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB1SIDL	658	858	xxx0 x0xx	uuu0 u0uu	uuuu uuuu
TXB1SIDH	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB1CON	658	858	0000 0000	0000 0000	uuuu uuuu
TXB2D7	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB2D6	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB2D5	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB2D4	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB2D3	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB2D2	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB2D1	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB2D0	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB2DLC	658	858	0x00 xxxx	0u00 uuuu	uuuu uuuu
TXB2EIDL	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB2EIDH	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB2SIDL	658	858	xxx0 x0xx	uuu0 u0uu	uuuu uuuu
TXB2SIDH	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
TXB2CON	658	858	0000 0000	0000 0000	uuuu uuuu
RXM1EIDL	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu
RXM1EIDH	658	858	xxxx xxxx	uuuu uuuu	uuuu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition

Note 1: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

2: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

3: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.

4: See Table 3-2 for RESET value for specific condition.

5: Bit 6 of PORTA, LATA, and TRISA are enabled in ECIO and RCIO oscillator modes only. In all other oscillator modes, they are disabled and read '0'.

6: The long write enable is only reset on a POR or $\overline{\text{MCLR}}$.

7: Available on PIC18C858 only.

4.6 Instruction Flow/Pipelining

An “Instruction Cycle” consists of four Q cycles (Q1, Q2, Q3 and Q4). The instruction fetch and execute are pipelined such that fetch takes one instruction cycle while decode and execute takes another instruction cycle. However, due to the pipelining, each instruction effectively executes in one cycle. If an instruction causes the program counter to change (e.g., GOTO), two cycles are required to complete the instruction (Example 4-2).

A fetch cycle begins with the program counter (PC) incrementing in Q1.

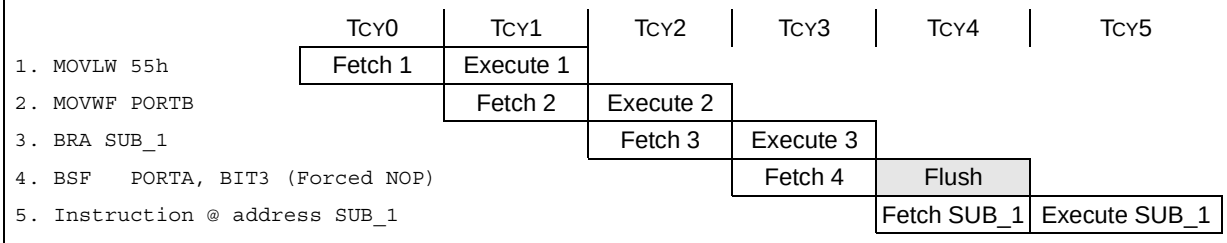
In the execution cycle, the fetched instruction is latched into the “Instruction Register” (IR) in cycle Q1. This instruction is then decoded and executed during the Q2, Q3, and Q4 cycles. Data memory is read during Q2 (operand read) and written during Q4 (destination write).

4.7 Instructions in Program Memory

The program memory is addressed in bytes. Instructions are stored as two bytes or four bytes in program memory. The Least Significant Byte of an instruction word is always stored in a program memory location with an even address (LSB = ’0’). Figure 4-1 shows an example of how instruction words are stored in the program memory. To maintain alignment with instruction boundaries, the PC increments in steps of 2 and the LSB will always read ’0’ (See Section 4.4).

The CALL and GOTO instructions have an absolute program memory address embedded into the instruction. Since instructions are always stored on word boundaries, the data contained in the instruction is a word address. The word address is written to PC<20:1>, which accesses the desired byte address in program memory. Instruction #2 in Figure 4-1 shows how the instruction “GOTO 000006h” is encoded in the program memory. Program branch instructions that encode a relative address offset operate in the same manner. The offset value stored in a branch instruction represents the number of single word instructions by which the PC will be offset. Section 23.0 provides further details of the instruction set.

EXAMPLE 4-2: INSTRUCTION PIPELINE FLOW



All instructions are single cycle, except for any program branches. These take two cycles since the fetch instruction is “flushed” from the pipeline while the new instruction is being fetched and then executed.

TABLE 4-1: INSTRUCTIONS IN PROGRAM MEMORY

Instruction	Opcode	Memory	Address
—			000007h
MOVLW 055h	0E55h	55h	000008h
		0Eh	000009h
GOTO 000006h	EF03h, F000h	03h	00000Ah
		EFh	00000Bh
		00h	00000Ch
		F0h	00000Dh
MOVFF 123h, 456h	C123h, F456h	23h	00000Eh
		C1h	00000Fh
		56h	000010h
		F4h	000011h
—			000012h

5.2.3 LONG WRITE INTERRUPTS

The long write must be terminated by a RESET or any interrupt.

The interrupt source must have its interrupt enable bit set. When the source sets its interrupt flag, programming will terminate. This will occur regardless of the settings of interrupt priority bits, the GIE/GIEH bit or the PIE/GIEL bit.

Depending on the states of interrupt priority bits, the GIE/GIEH bit or the PIE/GIEL bit, program execution can either be vectored to the high or low priority Interrupt Service Routine (ISR), or continue execution from where programming commenced.

In either case, the interrupt flag will not be cleared when programming is terminated and will need to be cleared by the software.

5.3 Unexpected Termination of Write Operations

If a write is terminated by an unplanned event such as loss of power, an unexpected RESET, or an interrupt that was not disabled, the memory location just programmed should be verified and reprogrammed if needed.

TABLE 5-2: SLEEP MODE, INTERRUPT ENABLE BITS AND INTERRUPT RESULTS

GIE/ GIEH	PIE/ GIEL	Priority	Interrupt Enable	Interrupt Flag	Action
X	X	X	0 (default)	X	Long write continues even if interrupt flag becomes set during SLEEP.
X	X	X	1	0	Long write continues, will wake when the interrupt flag is set.
0 (default)	0 (default)	X	1	1	Terminates long write, executes next instruction. Interrupt flag not cleared.
0 (default)	1	1 high priority (default)	1	1	Terminates long write, executes next instruction. Interrupt flag not cleared.
1	0 (default)	0 low	1	1	Terminates long write, executes next instruction. Interrupt flag not cleared.
0 (default)	1	0 low	1	1	Terminates long write, branches to low priority interrupt vector. Interrupt flag can be cleared by ISR.
1	0 (default)	1 high priority (default)	1	1	Terminates long write, branches to high priority interrupt vector. Interrupt flag can be cleared by ISR.

11.0 TIMER1 MODULE

The Timer1 module timer/counter has the following features:

- 16-bit timer/counter
  (Two 8-bit registers: TMR1H and TMR1L)
- Readable and writable (both registers)
- Internal or external clock select
- Interrupt on overflow from FFFFh to 0000h
- RESET from CCP module special event trigger

Register 11-1 shows the Timer1 control register. This register controls the operating mode of the Timer1 module as well as contains the Timer1 oscillator enable bit (T1OSCEN). Timer1 can be enabled/disabled by setting/clearing control bit TMR1ON (T1CON register).

Figure 11-1 is a simplified block diagram of the Timer1 module.

Note:	Timer1 is disabled on POR.

REGISTER 11-1: T1CON REGISTER

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RD16	—	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON
bit 7							bit 0

bit 7 **RD16:** 16-bit Read/Write Mode Enable bit

1 = Enables register Read/Write of TImer1 in one 16-bit operation
0 = Enables register Read/Write of Timer1 in two 8-bit operations

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

bit 5-4 **T1CKPS1:T1CKPS0**: Timer1 Input Clock Prescale Select bits

11 = 1:8 Prescale value
10 = 1:4 Prescale value
01 = 1:2 Prescale value
00 = 1:1 Prescale value

bit 3 **T1OSCEN:** Timer1 Oscillator Enable bit

1 = Timer1 Oscillator is enabled
0 = Timer1 Oscillator is shut off
The oscillator inverter and feedback resistor are turned off to eliminate power drain.

bit 2 **$\overline{\text{T1SYNC}}$:** Timer1 External Clock Input Synchronization Select bit

When TMR1CS = 1:
1 = Do not synchronize external clock input
0 = Synchronize external clock input

When TMR1CS = 0:
This bit is ignored. Timer1 uses the internal clock when TMR1CS = 0.

bit 1 **TMR1CS:** Timer1 Clock Source Select bit

1 = External clock from pin RC0/T1OSO/T13CKI (on the rising edge)
0 = Internal clock (FOSC/4)

bit 0 **TMR1ON:** Timer1 On bit

1 = Enables Timer1
0 = Stops Timer1

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

TABLE 11-2: REGISTERS ASSOCIATED WITH TIMER1 AS A TIMER/COUNTER

Name	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
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
PIR1	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
PIE1	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
IPR1	PSPIP	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	0000 0000	0000 0000
TMR1L	Holding register for the Least Significant Byte of the 16-bit TMR1 register								xxxx xxxx	uuuu uuuu
TMR1H	Holding register for the Most Significant Byte of the 16-bit TMR1 register								xxxx xxxx	uuuu uuuu
T1CON	RD16	—	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	0-00 0000	u-uu uuuu

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by the Timer1 module.

REGISTER 15-2: SSPCON1 REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
bit 7							bit 0

bit 7 **WCOL:** Write Collision Detect bit

Master mode:
1 = A write to the SSPBUF register was attempted while the I^2C conditions were not valid for a transmission to be started
0 = No collision

Slave mode:
1 = The SSPBUF register is written while it is still transmitting the previous word (must be cleared in software)
0 = No collision

bit 6 **SSPOV:** Receive Overflow Indicator bit

In SPI mode:

1 = A new byte is received while the SSPBUF register is still holding the previous data. In case of overflow, the data in SSPSR is lost. Overflow can only occur in Slave mode. In Slave mode, the user must read the SSPBUF, even if only transmitting data, to avoid setting overflow. In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPBUF register. (Must be cleared in software.)
0 = No overflow

In I^2C mode:

1 = A byte is received while the SSPBUF register is still holding the previous byte. SSPOV is a "don't care" in Transmit mode. (Must be cleared in software.)
0 = No overflow

bit 5 **SSPEN:** Synchronous Serial Port Enable bit
In both modes, when enabled, these pins must be properly configured as input or output.

In SPI mode:
1 = Enables serial port and configures SCK, SDO, SDI, and $\overline{SS}$ as the source of the serial port pins
0 = Disables serial port and configures these pins as I/O port pins

In I^2C mode:
1 = Enables the serial port and configures the SDA and SCL pins as the source of the serial port pins
0 = Disables serial port and configures these pins as I/O port pins

bit 4 **CKP:** Clock Polarity Select bit

In SPI mode:
1 = Idle state for clock is a high level
0 = Idle state for clock is a low level

In I^2C Slave mode:
SCK release control
1 = Enable clock
0 = Holds clock low (clock stretch). (Used to ensure data setup time.)

In I^2C Master mode
Unused in this mode

15.4.2 GENERAL CALL ADDRESS SUPPORT

The addressing procedure for the I^2C bus is such that the first byte after the START condition usually determines which device will be the slave addressed by the master. The exception is the general call address, which can address all devices. When this address is used, all devices should, in theory, respond with an acknowledge.

The general call address is one of eight addresses reserved for specific purposes by the I^2C protocol. It consists of all 0's with R/$\overline{W}$ = 0.

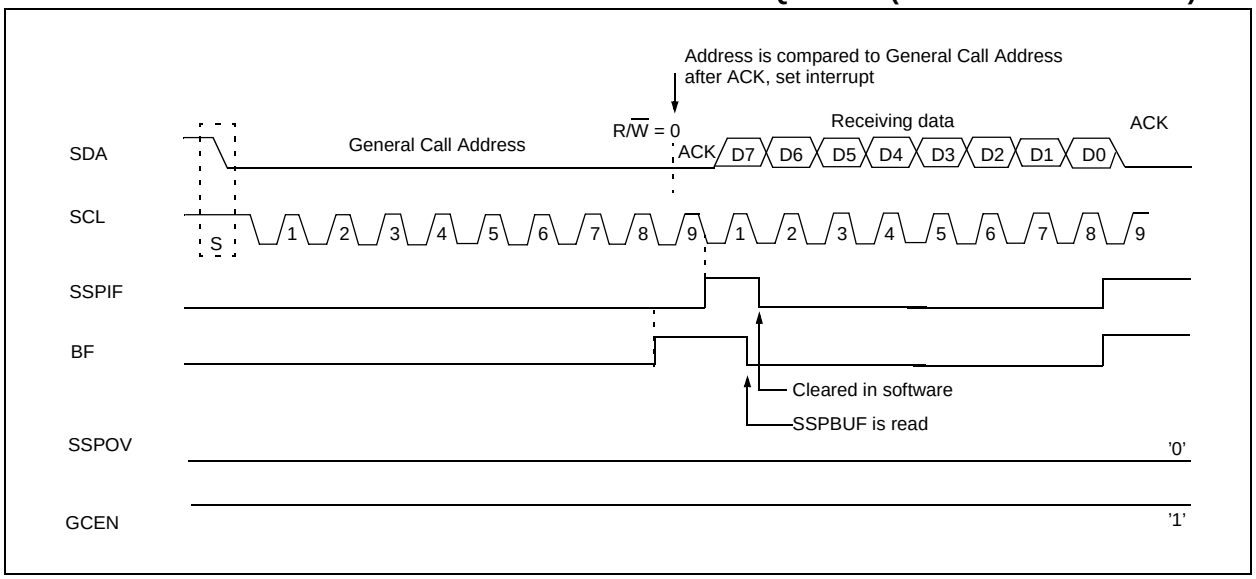
The general call address is recognized (enabled) when the General Call Enable (GCEN) bit is set (SSPCON2 register). Following a START bit detect, eight bits are shifted into the SSPSR and the address is compared against the SSPADD. It is also compared to the general call address and fixed in hardware.

If the general call address matches, the SSPSR is transferred to the SSPBUF, the BF bit is set (eighth bit), and on the falling edge of the ninth bit ($\overline{ACK}$ bit), the SSPIF interrupt flag bit is set.

When the interrupt is serviced, the source for the interrupt can be checked by reading the contents of the SSPBUF. The value can be used to determine if the address was device specific or a general call address.

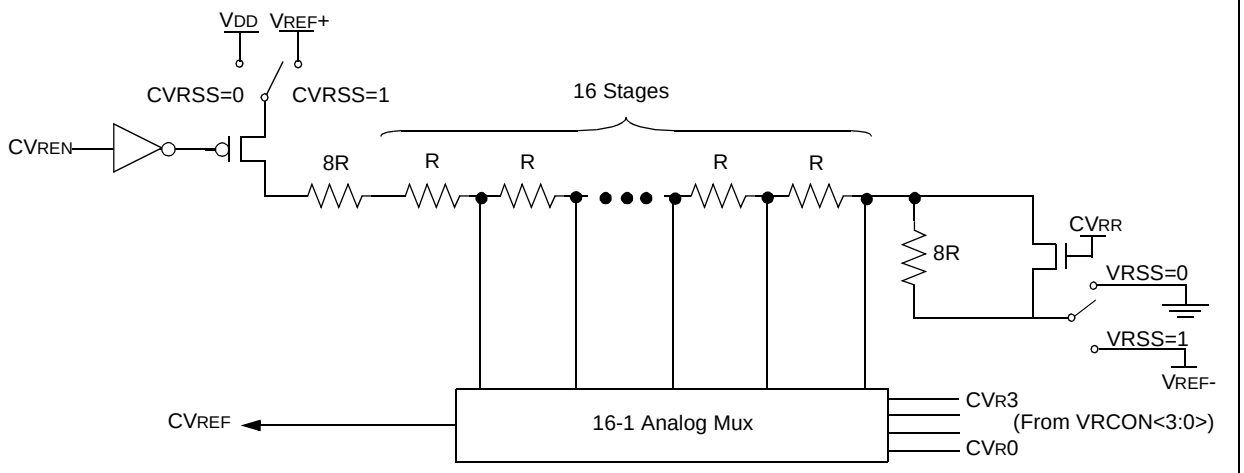
In 10-bit mode, the SSPADD is required to be updated for the second half of the address to match, and the UA bit is set (SSPSTAT register). If the general call address is sampled when the GCEN bit is set, and while the slave is configured in 10-bit address mode; then, the second half of the address is not necessary. The UA bit will not be set, and the slave will begin receiving data after the Acknowledge (Figure 15-9).

FIGURE 15-9: SLAVE MODE GENERAL CALL ADDRESS SEQUENCE (7 OR 10-BIT ADDRESS)



NOTES:

FIGURE 20-1: VOLTAGE REFERENCE BLOCK DIAGRAM



Note: R is defined in Section 25.0.

TABLE 23-2: PIC18CXX8 INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	16-Bit Instruction Word MSb			LSb	Status Affected	Notes
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	f [,d] [,a]	Add WREG and f	1	0010	01da	ffff	ffff	C, DC, Z, OV, N	1, 2, 6
ADDWFC	f [,d] [,a]	Add WREG and Carry bit to f	1	0010	00da	ffff	ffff	C, DC, Z, OV, N	1, 2, 6
ANDWF	f [,d] [,a]	AND WREG with f	1	0001	01da	ffff	ffff	Z, N	1,2, 6
CLRF	f [,a]	Clear f	1	0110	101a	ffff	ffff	Z	2, 6
COMF	f [,d] [,a]	Complement f	1	0001	11da	ffff	ffff	Z, N	1, 2, 6
CPFSEQ	f [,a]	Compare f with WREG, skip =	1 (2 or 3)	0110	001a	ffff	ffff	None	4, 6
CPFSGT	f [,a]	Compare f with WREG, skip >	1 (2 or 3)	0110	010a	ffff	ffff	None	4, 6
CPFSLT	f [,a]	Compare f with WREG, skip <	1 (2 or 3)	0110	000a	ffff	ffff	None	1, 2, 6
DECF	f [,d] [,a]	Decrement f	1	0000	01da	ffff	ffff	C, DC, Z, OV, N	1, 2, 3, 4, 6
DECFSZ	f [,d] [,a]	Decrement f, Skip if 0	1 (2 or 3)	0010	11da	ffff	ffff	None	1, 2, 3, 4, 6
DCFSNZ	f [,d] [,a]	Decrement f, Skip if Not 0	1 (2 or 3)	0100	11da	ffff	ffff	None	1, 2, 6
INCF	f [,d] [,a]	Increment f	1	0010	10da	ffff	ffff	C, DC, Z, OV, N	1, 2, 3, 4, 6
INCFSZ	f [,d] [,a]	Increment f, Skip if 0	1 (2 or 3)	0011	11da	ffff	ffff	None	4, 6
INFSNZ	f [,d] [,a]	Increment f, Skip if Not 0	1 (2 or 3)	0100	10da	ffff	ffff	None	1, 2, 6
IORWF	f [,d] [,a]	Inclusive OR WREG with f	1	0001	00da	ffff	ffff	Z, N	1, 2, 6
MOVF	f [,d] [,a]	Move f	1	0101	00da	ffff	ffff	Z, N	1, 6
MOVFF	f_s, f_d	Move f_s (source) to 1st word f_d (destination)2nd word	2	1100 1111	ffff ffff	ffff ffff	ffff ffff	None	
MOVWF	f [,a]	Move WREG to f	1	0110	111a	ffff	ffff	None	6
MULWF	f [,a]	Multiply WREG with f	1	0000	001a	ffff	ffff	None	6
NEGF	f [,a]	Negate f	1	0110	110a	ffff	ffff	C, DC, Z, OV, N	1, 2, 6
RLCF	f [,d] [,a]	Rotate Left f through Carry	1	0011	01da	ffff	ffff	C, Z, N	6
RLNCF	f [,d] [,a]	Rotate Left f (No Carry)	1	0100	01da	ffff	ffff	Z, N	1, 2, 6
RRCF	f [,d] [,a]	Rotate Right f through Carry	1	0011	00da	ffff	ffff	C, Z, N	6
RRNCF	f [,d] [,a]	Rotate Right f (No Carry)	1	0100	00da	ffff	ffff	Z, N	6
SETF	f [,a]	Set f	1	0110	100a	ffff	ffff	None	6
SUBFWB	f [,d] [,a]	Subtract f from WREG with borrow	1	0101	01da	ffff	ffff	C, DC, Z, OV, N	1, 2, 6
SUBWF	f [,d] [,a]	Subtract WREG from f	1	0101	11da	ffff	ffff	C, DC, Z, OV, N	6
SUBWFB	f [,d] [,a]	Subtract WREG from f with borrow	1	0101	10da	ffff	ffff	C, DC, Z, OV, N	1, 2, 6
SWAPF	f [,d] [,a]	Swap nibbles in f	1	0011	10da	ffff	ffff	None	4, 6
TSTFSZ	f [,a]	Test f, skip if 0	1 (2 or 3)	0110	011a	ffff	ffff	None	1, 2, 6
XORWF	f [,d] [,a]	Exclusive OR WREG with f	1	0001	10da	ffff	ffff	Z, N	6
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	f, b [,a]	Bit Clear f	1	1001	bbba	ffff	ffff	None	1, 2, 6
BSF	f, b [,a]	Bit Set f	1	1000	bbba	ffff	ffff	None	1, 2, 6
BTFSC	f, b [,a]	Bit Test f, Skip if Clear	1 (2 or 3)	1011	bbba	ffff	ffff	None	3, 4, 6
BTFSS	f, b [,a]	Bit Test f, Skip if Set	1 (2 or 3)	1010	bbba	ffff	ffff	None	3, 4, 6
BTG	f [,d] [,a]	Bit Toggle f	1	0111	bbba	ffff	ffff	None	1, 2, 6

Note 1: When a PORT register is modified as a function of itself (e.g., `MOVF PORTB, 1, 0`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

2: If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned.

3: If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a `NOP`.

4: Some instructions are 2 word instructions. The second word of these instructions will be executed as a `NOP`, unless the first word of the instruction retrieves the information embedded in these 16-bits. This ensures that all program memory locations have a valid instruction.

5: If the table write starts the write cycle to internal memory, the write will continue until terminated.

6: Microchip Assembler MASM automatically defaults destination bit 'd' to '1', while access bit 'a' defaults to '1' or '0' according to address of register being used.

INCFSZ **Increment f, skip if 0**

Syntax: [*label*] INCFSZ f [,d [,a]]

Operands: $0 \le f \le 255$
$d \in [0,1]$
$a \in [0,1]$

Operation: (f) + 1 → dest,
skip if result = 0

Status Affected: None

Encoding:

0011	11da	ffff	ffff

Description: The contents of register 'f' are incremented. If 'd' is 0, the result is placed in WREG. If 'd' is 1, the result is placed back in register 'f' (default).

If the result is 0, the next instruction, which is already fetched, is discarded, and a NOP is executed instead making it a two-cycle instruction. If 'a' is 0, the Access Bank will be selected, overriding the BSR value. If 'a' is 1, the Bank will be selected as per the BSR value.

Words: 1

Cycles: 1(2)
Note: 3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

If skip and followed by 2-word instruction:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation

Example:
```
HERE    INCFSZ   CNT
NZERO   :
ZERO    :
```

Before Instruction
PC = Address (HERE)

After Instruction
CNT = CNT + 1
If CNT = 0;
PC = Address (ZERO)
If CNT ≠ 0;
PC = Address (NZERO)

INFSNZ **Increment f, skip if not 0**

Syntax: [*label*] INFSNZ f [, d [,a]]

Operands: $0 \le f \le 255$
$d \in [0,1]$
$a \in [0,1]$

Operation: (f) + 1 → dest,
skip if result ≠ 0

Status Affected: None

Encoding:

0100	10da	ffff	ffff

Description: The contents of register 'f' are incremented. If 'd' is 0, the result is placed in WREG. If 'd' is 1, the result is placed back in register 'f' (default).

If the result is not 0, the next instruction, which is already fetched, is discarded, and a NOP is executed instead making it a two-cycle instruction. If 'a' is 0, the Access Bank will be selected, overriding the BSR value. If 'a' is 1, the Bank will be selected as per the BSR value.

Words: 1

Cycles: 1(2)
Note: 3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write to destination

If skip:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation

If skip and followed by 2-word instruction:

Q1	Q2	Q3	Q4
No operation	No operation	No operation	No operation
No operation	No operation	No operation	No operation

Example:
```
HERE    INFSNZ   REG
ZERO
NZERO
```

Before Instruction
PC = Address (HERE)

After Instruction
REG = REG + 1
If REG ≠ 0;
PC = Address (NZERO)
If REG = 0;
PC = Address (ZERO)

NEGF **Negate f**

Syntax: [*label*] NEGF f [,a]

Operands: $0 \le f \le 255$
$a \in [0,1]$

Operation: $(\overline{f}) + 1 \rightarrow f$

Status Affected: N,OV, C, DC, Z

Encoding:

0110	110a	ffff	ffff

Description: Location 'f' is negated using two's complement. The result is placed in the data memory location 'f'. If 'a' is 0, the Access Bank will be selected, overriding the BSR value. If 'a' is 1, the Bank will be selected as per the BSR value.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write register 'f'

Example: NEGF REG

Before Instruction
REG = 0011 1010 [0x3A]
N = ?
OV = ?
C = ?
DC = ?
Z = ?

After Instruction
REG = 1100 0110 [0xC6]
N = 1
OV = 0
C = 0
DC = 0
Z = 0

NOP **No Operation**

Syntax: [*label*] NOP

Operands: None

Operation: No operation

Status Affected: None

Encoding:

0000 1111	0000 xxxx	0000 xxxx	0000 xxxx

Description: No operation.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	No operation	No operation

Example:

None.

FIGURE 25-3: LOW VOLTAGE DETECT CHARACTERISTICS

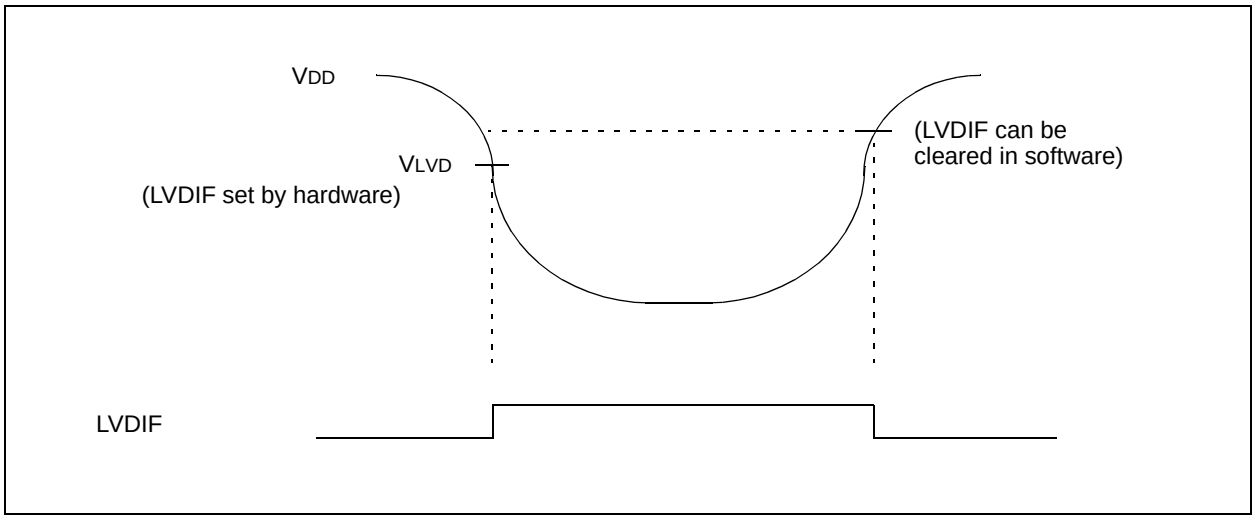


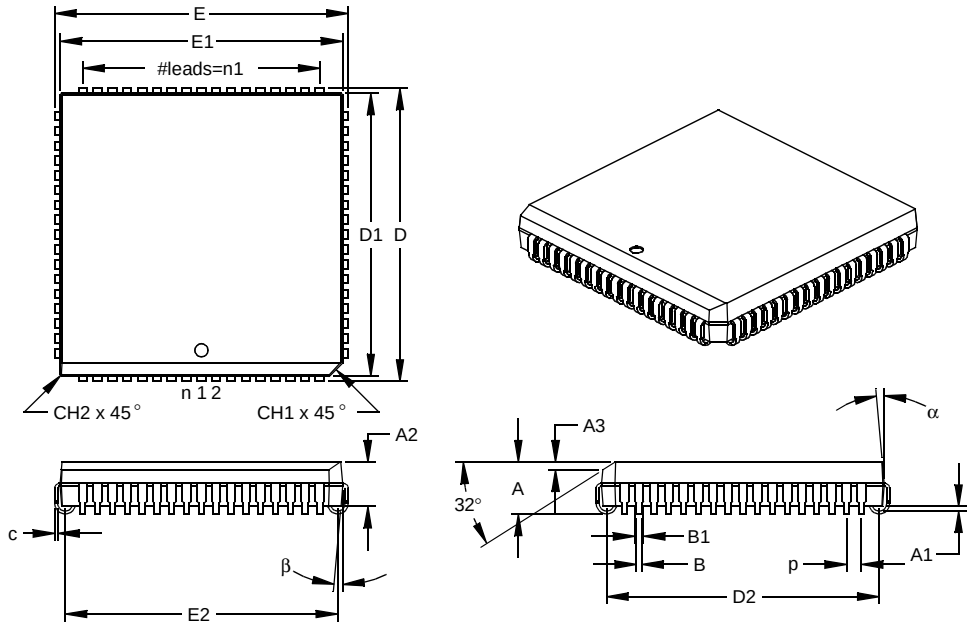
TABLE 25-1: LOW VOLTAGE DETECT CHARACTERISTICS

			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended				
Param No.	**Symbol**	**Characteristic/**		**Min**	**Max**	**Units**	**Conditions**
D420	VLVD	LVD Voltage	LVDL<3:0> = 0100	2.5	2.66	V	
			LVDL<3:0> = 0101	2.7	2.86	V	
			LVDL<3:0> = 0110	2.8	2.98	V	
			LVDL<3:0> = 0111	3.0	3.2	V	
			LVDL<3:0> = 1000	3.3	3.52	V	
			LVDL<3:0> = 1001	3.5	3.72	V	
			LVDL<3:0> = 1010	3.6	3.84	V	
			LVDL<3:0> = 1011	3.8	4.04	V	
			LVDL<3:0> = 1100	4.0	4.26	V	
			LVDL<3:0> = 1101	4.2	4.46	V	
			LVDL<3:0> = 1110	4.5	4.78	V	

TABLE 25-5: PLL CLOCK TIMING SPECIFICATION (V_{DD} = 4.2V - 5.5V)

Param No.	Symbol	Characteristic	Min	Max	Units	Conditions
7	TPLL	PLL Start-up Time (Lock Time)	—	2	ms	
	ΔCLK	CLKOUT Stability (Jitter) using PLL	-2	+2	%	

68-Lead Plastic Leaded Chip Carrier (L) – Square (PLCC)



Units		INCHES*			MILLIMETERS		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		68			68	
Pitch	p		.050			1.27	
Pins per Side	n1		17			17	
Overall Height	A	.165	.173	.180	4.19	4.39	4.57
Molded Package Thickness	A2	.145	.153	.160	3.68	3.87	4.06
Standoff §	A1	.020	.028	.035	0.51	0.71	0.89
Side 1 Chamfer Height	A3	.024	.029	.034	0.61	0.74	0.86
Corner Chamfer 1	CH1	.040	.045	.050	1.02	1.14	1.27
Corner Chamfer (others)	CH2	.000	.005	.010	0.00	0.13	0.25
Overall Width	E	.985	.990	.995	25.02	25.15	25.27
Overall Length	D	.985	.990	.995	25.02	25.15	25.27
Molded Package Width	E1	.950	.954	.958	24.13	24.23	24.33
Molded Package Length	D1	.950	.954	.958	24.13	24.23	24.33
Footprint Width	E2	.890	.920	.930	22.61	23.37	23.62
Footprint Length	D2	.890	.920	.930	22.61	23.37	23.62
Lead Thickness	c	.008	.011	.013	0.20	0.27	0.33
Upper Lead Width	B1	.026	.029	.032	0.66	0.74	0.81
Lower Lead Width	B	.013	.020	.021	0.33	0.51	0.53
Mold Draft Angle Top	α	0	5	10	0	5	10
Mold Draft Angle Bottom	β	0	5	10	0	5	10

* Controlling Parameter
§ Significant Characteristic
Notes:
Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010” (0.254mm) per side.
JEDEC Equivalent: MO-047
Drawing No. C04-049

