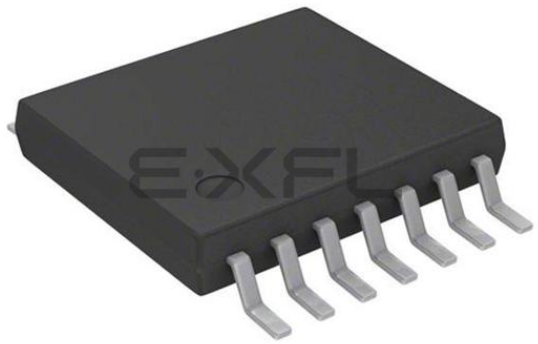


Welcome to E-XFL.COM

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	16-Bit
Speed	32MHz
Connectivity	I²C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, HLVD, POR, PWM, WDT
Number of I/O	12
Program Memory Size	4KB (1.375K x 24)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	14-TSSOP (0.173", 4.40mm Width)
Supplier Device Package	14-TSSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24f04kl100-e-st

TABLE 1-4: PIC24F16KL40X/30X FAMILY PINOUT DESCRIPTIONS (CONTINUED)

Function	Pin Number				I/O	Buffer	Description
	20-Pin PDIP/ SSOP/ SOIC	20-Pin QFN	28-Pin SPDIP/ SSOP/ SOIC	28-Pin QFN			
PGEC1	5	2	5	2	I/O	ST	ICSP™ Clock 1
PCED1	4	1	4	1	I/O	ST	ICSP Data 1
PGEC2	2	19	22	19	I/O	ST	ICSP Clock 2
PGED2	3	20	21	18	I/O	ST	ICSP Data 2
PGEC3	10	7	15	12	I/O	ST	ICSP Clock 3
PGED3	9	6	14	11	I/O	ST	ICSP Data 3
RA0	2	19	2	27	I/O	ST	PORTA Pins
RA1	3	20	3	28	I/O	ST	
RA2	7	4	9	6	I/O	ST	
RA3	8	5	10	7	I/O	ST	
RA4	10	7	12	9	I/O	ST	
RA5	1	18	1	26	I	ST	
RA6	14	11	20	17	I/O	ST	
RA7	—	—	19	16	I/O	ST	
RB0	4	1	4	1	I/O	ST	PORTB Pins
RB1	5	2	5	2	I/O	ST	
RB2	6	3	6	3	I/O	ST	
RB3	—	—	7	4	I/O	ST	
RB4	9	6	11	8	I/O	ST	
RB5	—	—	14	11	I/O	ST	
RB6	—	—	15	12	I/O	ST	
RB7	11	8	16	13	I/O	ST	
RB8	12	9	17	14	I/O	ST	
RB9	13	10	18	15	I/O	ST	
RB10	—	—	21	18	I/O	ST	
RB11	—	—	22	19	I/O	ST	
RB12	15	12	23	20	I/O	ST	
RB13	16	13	24	21	I/O	ST	
RB14	17	14	25	22	I/O	ST	
RB15	18	15	26	23	I/O	ST	
REFO	18	15	26	23	O	—	Reference Clock Output
SCK1	15	12	22	19	I/O	ST	MSSP1 SPI Serial Input/Output Clock
SCK2	18	15	14	11	I/O	ST	MSSP2 SPI Serial Input/Output Clock
SCL1	12	9	17	14	I/O	I^2C	MSSP1 I^2C Clock Input/Output
SCL2	18	15	7	4	I/O	I^2C	MSSP2 I^2C Clock Input/Output
SCLKI	10	7	12	9	I	ST	Digital Secondary Clock Input
SDA1	13	10	18	15	I/O	I^2C	MSSP1 I^2C Data Input/Output
SDA2	2	19	2	27	I/O	I^2C	MSSP2 I^2C Data Input/Output
SDI1	17	14	21	18	I	ST	MSSP1 SPI Serial Data Input
SDI2	2	19	19	16	I	ST	MSSP2 SPI Serial Data Input
SDO1	16	13	24	21	O	—	MSSP1 SPI Serial Data Output
SDO2	3	20	15	12	O	—	MSSP2 SPI Serial Data Output

Legend: TTL = TTL input buffer
ANA = Analog level input/output
ST = Schmitt Trigger input buffer
I^2C = I^2C™/SMBus input buffer

5.0 FLASH PROGRAM MEMORY

Note: This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information on Flash Programming, refer to the *"dsPIC33/PIC24 Family Reference Manual"*, **"Program Memory"** (DS39715).

The PIC24F16KL402 family of devices contains internal Flash program memory for storing and executing application code. The memory is readable, writable and erasable when operating with VDD over 1.8V.

Flash memory can be programmed in three ways:

- In-Circuit Serial Programming™ (ICSP™)
- Run-Time Self Programming (RTSP)
- Enhanced In-Circuit Serial Programming (Enhanced ICSP)

ICSP allows a PIC24F device to be serially programmed while in the end application circuit. This is simply done with two lines for the programming clock and programming data (which are named PGECx and PGEDx, respectively), and three other lines for power (VDD), ground (VSS) and Master Clear/Program mode Entry Voltage ($\overline{MCLR}$/VPP). This allows customers to manufacture boards with unprogrammed devices and then program the microcontroller just before shipping the product. This also allows the most recent firmware or custom firmware to be programmed.

Run-Time Self Programming (RTSP) is accomplished using TBLRD (Table Read) and TBLWT (Table Write) instructions. With RTSP, the user may write program memory data in blocks of 32 instructions (96 bytes) at a time, and erase program memory in blocks of 32, 64 and 128 instructions (96,192 and 384 bytes) at a time.

The NVMOP<1:0> (NVMCON<1:0>) bits decide the erase block size.

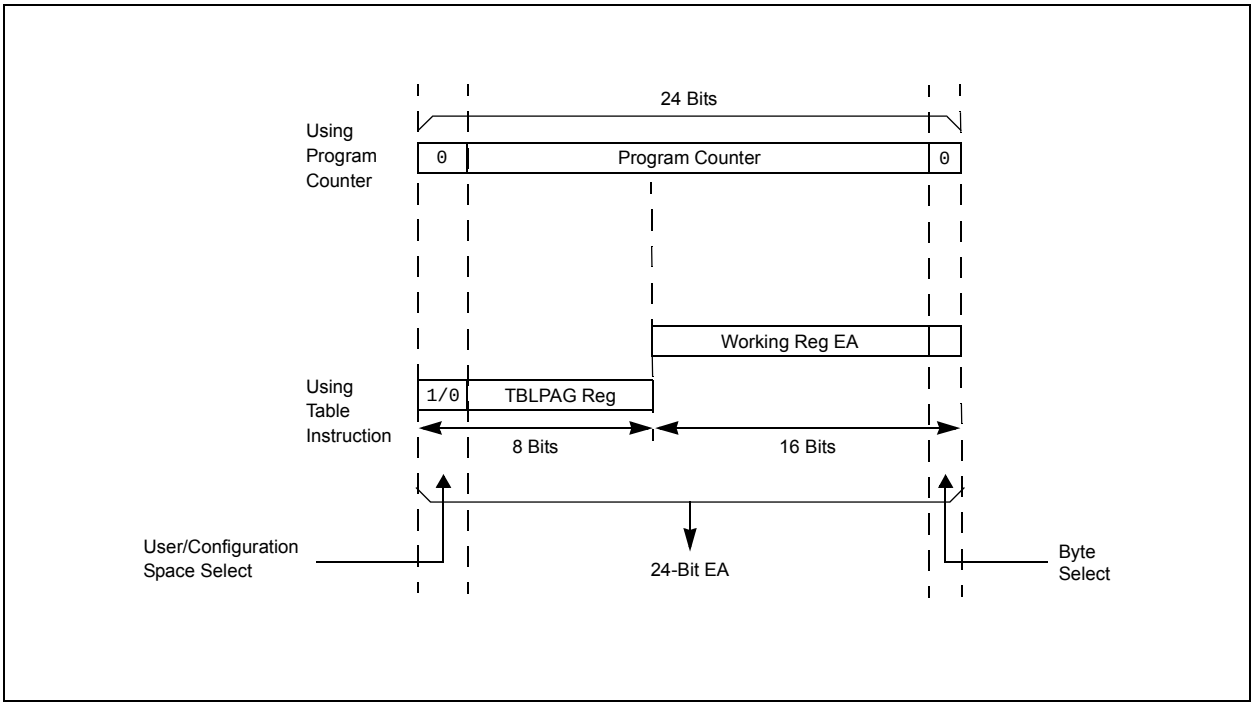
5.1 Table Instructions and Flash Programming

Regardless of the method used, Flash memory programming is done with the Table Read and Table Write instructions. These allow direct read and write access to the program memory space from the data memory while the device is in normal operating mode. The 24-bit target address in the program memory is formed using the TBLPAG<7:0> bits and the Effective Address (EA) from a W register, specified in the table instruction, as depicted in Figure 5-1.

The TBLRDL and TBLWTL instructions are used to read or write to bits<15:0> of program memory. TBLRDL and TBLWTL can access program memory in both Word and Byte modes.

The TBLRDH and TBLWTH instructions are used to read or write to bits<23:16> of program memory. TBLRDH and TBLWTH can also access program memory in Word or Byte mode.

FIGURE 5-1: ADDRESSING FOR TABLE REGISTERS



5.2 RTSP Operation

The PIC24F Flash program memory array is organized into rows of 32 instructions or 96 bytes. RTSP allows the user to erase blocks of 1 row, 2 rows and 4 rows (32, 64 and 128 instructions) at a time, and to program one row at a time.

The 1-row (96 bytes), 2-row (192 bytes) and 4-row (384 bytes) erase blocks and single row write block (96 bytes) are edge-aligned, from the beginning of program memory.

When data is written to program memory using TBLWT instructions, the data is not written directly to memory. Instead, data written using Table Writes is stored in holding latches until the programming sequence is executed.

Any number of TBLWT instructions can be executed and a write will be successfully performed. However, 32 TBLWT instructions are required to write the full row of memory.

The basic sequence for RTSP programming is to set up a Table Pointer, then do a series of TBLWT instructions to load the buffers. Programming is performed by setting the control bits in the NVMCON register.

Data can be loaded in any order and the holding registers can be written to multiple times before performing a write operation. Subsequent writes, however, will wipe out any previous writes.

Note:	Writing to a location multiple times without erasing it is not recommended.

All of the Table Write operations are single-word writes (two instruction cycles), because only the buffers are written. A programming cycle is required for programming each row.

5.3 Enhanced In-Circuit Serial Programming

Enhanced ICSP uses an on-board bootloader, known as the program executive, to manage the programming process. Using an SPI data frame format, the program executive can erase, program and verify program memory. For more information on Enhanced ICSP, see the device programming specification.

5.4 Control Registers

There are two SFRs used to read and write the program Flash memory: NVMCON and NVMKEY.

The NVMCON register (Register 5-1) controls the blocks that need to be erased, which memory type is to be programmed and when the programming cycle starts.

NVMKEY is a write-only register that is used for write protection. To start a programming or erase sequence, the user must consecutively write 55h and AAh to the NVMKEY register. For more information, refer to **Section 5.5 “Programming Operations”**.

5.5 Programming Operations

A complete programming sequence is necessary for programming or erasing the internal Flash in RTSP mode. During a programming or erase operation, the processor stalls (waits) until the operation is finished. Setting the WR bit (NVMCON<15>) starts the operation and the WR bit is automatically cleared when the operation is finished.

EXAMPLE 5-4: LOADING THE WRITE BUFFERS – 'C' LANGUAGE CODE

```
// C example using MPLAB C30

  #define NUM_INSTRUCTION_PER_ROW 64
   int __attribute__ ((space(auto_psv))) progAddr = &progAddr; // Global variable located in Pgm Memory
  unsigned int offset;
  unsigned int i;
  unsigned int progData[2*NUM_INSTRUCTION_PER_ROW];          // Buffer of data to write

  //Set up NVMCON for row programming
  NVMCON = 0x4004;                                          // Initialize NVMCON

  //Set up pointer to the first memory location to be written
  TBLPAG = __builtin_tblpage(&progAddr);                    // Initialize PM Page Boundary SFR
  offset = &progAddr & 0xFFFF;                              // Initialize lower word of address

  //Perform TBLWT instructions to write necessary number of latches
  for(i=0; i < 2*NUM_INSTRUCTION_PER_ROW; i++)
  {
      __builtin_tblwtl(offset, progData[i++]);              // Write to address low word
      __builtin_tblwth(offset, progData[i]);                // Write to upper byte
      offset = offset + 2;                                  // Increment address
   }
```

EXAMPLE 5-5: INITIATING A PROGRAMMING SEQUENCE – ASSEMBLY LANGUAGE CODE

```
        DISI    #5                      ; Block all interrupts
                                          for next 5 instructions
        MOV     #0x55, W0
        MOV     W0, NVMKEY              ; Write the 55 key
        MOV     #0xAA, W1               ;
        MOV     W1, NVMKEY              ; Write the AA key
        BSET    NVMCON, #WR             ; Start the erase sequence
        NOP                             ; 2 NOPs required after setting WR
        NOP                             ;
        BTSC    NVMCON, #15             ; Wait for the sequence to be completed
        BRA     $-2                     ;
```

EXAMPLE 5-6: INITIATING A PROGRAMMING SEQUENCE – 'C' LANGUAGE CODE

```
// C example using MPLAB C30

asm("DISI #5");              // Block all interrupts for next 5 instructions

__builtin_write_NVM();       // Perform unlock sequence and set WR
```

REGISTER 8-13: IEC2: INTERRUPT ENABLE CONTROL REGISTER 2

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
—	—	T3GIE	—	—	—	—	—
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 15-6 **Unimplemented:** Read as '0'

bit 5 **T3GIF:** Timer3 External Gate Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 4-0 **Unimplemented:** Read as '0'

REGISTER 8-14: IEC3: INTERRUPT ENABLE CONTROL REGISTER 3

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	U-0
—	—	—	—	—	BCL2IE(1)	SSP2IE(1)	—
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 15-3 **Unimplemented:** Read as '0'

bit 2 **BCL2IE:** MSSP2 I^2C™ Bus Collision Interrupt Enable bit(1)
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 1 **SSP2IF:** MSSP2 SPI/I^2C Event Interrupt Enable bit(1)
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

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

Note 1: These bits are unimplemented on PIC24FXXKL10X and PIC24FXXKL20X devices.

REGISTER 8-15: IEC4: INTERRUPT ENABLE CONTROL REGISTER 4

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	HLVDIE
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	U-0
—	—	—	—	—	U2ERIE(1)	U1ERIE	—
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 15-9 **Unimplemented:** Read as '0'

bit 8 **HLVDIE:** High/Low-Voltage Detect Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 7-3 **Unimplemented:** Read as '0'

bit 2 **U2ERIE:** UART2 Error Interrupt Enable bit(1)
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

bit 1 **U1ERIE:** UART1 Error Interrupt Enable bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

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

Note 1: This bit is unimplemented on PIC24FXXKL10X and PIC24FXXKL20X devices.

REGISTER 8-16: IEC5: INTERRUPT ENABLE CONTROL REGISTER 5

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	ULPWUIE
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 15-1 **Unimplemented:** Read as '0'

bit 0 **ULPWUIE:** Ultra Low-Power Wake-up Interrupt Enable Bit
1 = Interrupt request is enabled
0 = Interrupt request is not enabled

REGISTER 8-25: IPC9: INTERRUPT PRIORITY CONTROL REGISTER 9

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	T3GIP2	T3GIP1	T3GIP0	—	—	—	—
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 15-7 **Unimplemented:** Read as ‘0’

bit 6-4 **T3GIP<2:0>:** Timer3 External Gate Interrupt Priority bits
111 = Interrupt is Priority 7 (highest priority interrupt)
•
•
•
001 = Interrupt is Priority 1
000 = Interrupt source is disabled

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

10.3 Ultra Low-Power Wake-up

The Ultra Low-Power Wake-up (ULPWU) on pin, RB0, allows a slow falling voltage to generate an interrupt without excess current consumption. This feature provides a low-power technique for periodically waking up the device from Sleep mode.

To use this feature:

1. Charge the capacitor on RB0 by configuring the RB0 pin to an output and setting it to '1'.
2. Stop charging the capacitor by configuring RB0 as an input.
3. Discharge the capacitor by setting the ULPEN and ULPSINK bits in the ULPWCON register.
4. Configure Sleep mode.
5. Enter Sleep mode.

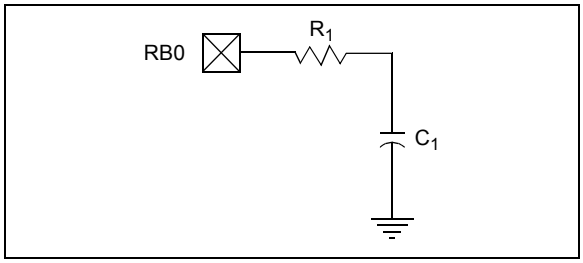
The time-out is dependent on the discharge time of the RC circuit on RB0. When the voltage on RB0 drops below V_{IL}, the device wakes up and executes the next instruction.

When the ULPWU module wakes the device from Sleep mode, the ULPWUIF bit (IFS5<0>) is set. Software can check this bit upon wake-up to determine the wake-up source.

See Example 10-2 for initializing the ULPWU module.

A series resistor, between RB0 and the external capacitor, provides overcurrent protection for the RB0/AN2/ULPWU pin and enables software calibration of the time-out (see Figure 10-1).

FIGURE 10-1: SERIES RESISTOR



A timer can be used to measure the charge time and discharge time of the capacitor. The charge time can then be adjusted to provide the desired delay in Sleep. This technique compensates for the affects of temperature, voltage and component accuracy. The peripheral can also be configured as a simple, programmable Low-Voltage Detect (LVD) or temperature sensor.

EXAMPLE 10-2: ULTRA LOW-POWER WAKE-UP INITIALIZATION

```
//*****************************************************************************
// 1. Charge the capacitor on RB0
//*****************************************************************************
    TRISBbits.TRISB0 = 0;
    LATBbits.LATB0 = 1;
    for(i = 0; i < 10000; i++) Nop();
//*****************************************************************************
//2. Stop Charging the capacitor on RB0
//*****************************************************************************
    TRISBbits.TRISB0 = 1;
//*****************************************************************************
//3. Enable ULPWU Interrupt
//*****************************************************************************
IFS5bits.ULPWUIF = 0;
IEC5bits.ULPWUIE = 1;
IPC20bits.ULPWUIP = 0x7;
//*****************************************************************************
//4. Enable the Ultra Low Power Wakeup module and allow capacitor discharge
//*****************************************************************************
    ULPWCONbits.ULPEN = 1;
    ULPWCONbits.ULPSINK = 1;
//*****************************************************************************
//5. Enter Sleep Mode
//*****************************************************************************
    Sleep();
//for Sleep, execution will resume here
```

15.0 TIMER4 MODULE

Note:	This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information on Timers, refer to the *"dsPIC33/PIC24 Family Reference Manual"*, **"Timers"** (DS39704).

The Timer4 module is implemented in PIC24FXXKL30X/40X devices only. It has the following features:

- Eight-bit Timer register (TMR4)
- Eight-bit Period register (PR4)
- Readable and writable (all registers)
- Software programmable prescaler (1:1, 1:4, 1:16)
- Software programmable postscaler (1:1 to 1:16)
- Interrupt on TMR4 match of PR4

The Timer4 module has a control register shown in Register 15-1. Timer4 can be shut off by clearing control bit, TMR4ON (T4CON<2>), to minimize power consumption. The prescaler and postscaler selection of Timer4 is controlled by this register.

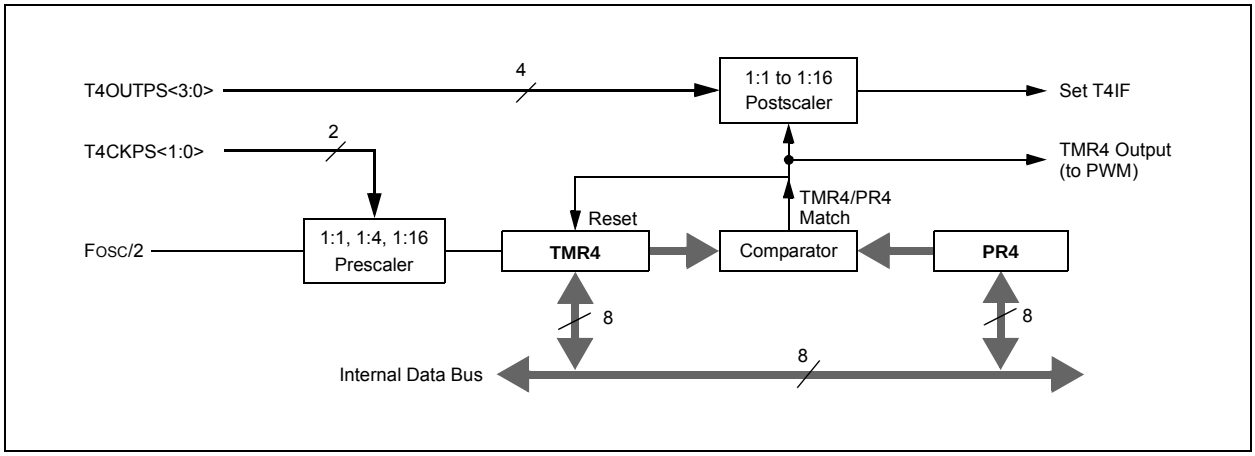
The prescaler and postscaler counters are cleared when any of the following occurs:

- A write to the TMR4 register
- A write to the T4CON register
- Any device Reset (POR, BOR, $\overline{\text{MCLR}}$ or WDT Reset)

TMR4 is not cleared when T4CON is written.

Figure 15-1 is a simplified block diagram of the Timer4 module.

FIGURE 15-1: TIMER4 BLOCK DIAGRAM



REGISTER 15-1: T4CON: TIMER4 CONTROL REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	T4OUTPS3	T4OUTPS2	T4OUTPS1	T4OUTPS0	TMR4ON	T4CKPS1	T4CKPS0
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 15-7 **Unimplemented:** Read as ‘0’

bit 6-3 **T4OUTPS<3:0>:** Timer4 Output Postscale Select bits
1111 = 1:16 Postscale
1110 = 1:15 Postscale
•
•
•
0001 = 1:2 Postscale
0000 = 1:1 Postscale

bit 2 **TMR4ON:** Timer4 On bit
1 = Timer4 is on
0 = Timer4 is off

bit 1-0 **T4CKPS<1:0>:** Timer4 Clock Prescale Select bits
10 = Prescaler is 16
01 = Prescaler is 4
00 = Prescaler is 1

17.0 MASTER SYNCHRONOUS SERIAL PORT (MSSP)

Note: This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information on MSSP, refer to the *"dsPIC33/PIC24 Family Reference Manual"*.

The Master Synchronous Serial Port (MSSP) module is an 8-bit serial interface, useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be serial EEPROMs, Shift registers, display drivers, A/D Converters, etc. The MSSP module can operate in one of two modes:

- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I^2C™)
  - Full Master mode
  - Slave mode (with general address call)

The SPI interface supports these modes in hardware:

- Master mode
- Slave mode
- Daisy-Chaining Operation in Slave mode
- Synchronized Slave operation

The I^2C interface supports the following modes in hardware:

- Master mode
- Multi-Master mode
- Slave mode with 10-Bit And 7-Bit Addressing and Address Masking
- Byte NACKing
- Selectable Address and Data Hold and Interrupt Masking

17.1 I/O Pin Configuration for SPI

In SPI Master mode, the MSSP module will assert control over any pins associated with the SDOx and SCKx outputs. This does not automatically disable other digital functions associated with the pin, and may result in the module driving the digital I/O port inputs. To prevent this, the MSSP module outputs must be disconnected from their output pins while the module is in SPI Master mode. While disabling the module temporarily may be an option, it may not be a practical solution in all applications.

The SDOx and SCKx outputs for the module can be selectively disabled by using the SDOxDIS and SCKxDIS bits in the PADCFG1 register (Register 17-10). Setting the bit disconnects the corresponding output for a particular module from its assigned pin.

REGISTER 19-3: AD1CON3: A/D CONTROL REGISTER 3

R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADRC	EXTSAM	PUMPEN	SAMC4	SAMC3	SAMC2	SAMC1	SAMC0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	ADCS5	ADCS4	ADCS3	ADCS2	ADCS1	ADCS0
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 15 **ADRC:** A/D Conversion Clock Source bit
1 = A/D internal RC clock
0 = Clock derived from system clock

bit 14 **EXTSAM:** Extended Sampling Time bit
1 = A/D is still sampling after SAMP = 0
0 = A/D is finished sampling

bit 13 **PUMPEN:** Charge Pump Enable bit
1 = Charge pump for switches is enabled
0 = Charge pump for switches is disabled

bit 12-8 **SAMC<4:0>:** Auto-Sample Time bits
11111 = 31 TAD
•
•
•
00001 = 1 TAD
00000 = 0 TAD (not recommended)

bit 7-6 **Unimplemented:** Maintain as '0'

bit 5-0 **ADCS<5:0>:** A/D Conversion Clock Select bits
11111 = 64 • TCY
11110 = 63 • TCY
•
•
•
00001 = 2 • TCY
00000 = TCY

REGISTER 23-6: FPOR: RESET CONFIGURATION REGISTER

R/P-1	R/P-1	R/P-1	R/P-1	R/P-1	U-0	R/P-1	R/P-1
MCLRE(1)	BORV1(2)	BORV0(2)	I2C1SEL(3)	PWRTEN	—	BOREN1	BOREN0
bit 7							bit 0

Legend:			
R = Readable bit	P = Programmable 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 **MCLRE:** $\overline{\text{MCLR}}$ Pin Enable bit(1)
1 = $\overline{\text{MCLR}}$ pin is enabled; RA5 input pin is disabled
0 = RA5 input pin is enabled; $\overline{\text{MCLR}}$ is disabled

bit 6-5 **BORV<1:0>:** Brown-out Reset Enable bits(2)
11 = Brown-out Reset is set to the low trip point
10 = Brown-out Reset is set to the middle trip point
01 = Brown-out Reset is set to the high trip point
00 = Downside protection on POR is enabled (Low-Power BOR is selected)

bit 4 **I2C1SEL:** Alternate MSSP1 I^2C™ Pin Mapping bit(3)
1 = Default location for SCL1/SDA1 pins (RB8 and RB9)
0 = Alternate location for SCL1/SDA1 pins (ASCL1/RB6 and ASDA1/RB5)

bit 3 **PWRTEN:** Power-up Timer Enable bit
1 = PWRT is enabled
0 = PWRT is disabled

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

bit 1-0 **BOREN<1:0>:** Brown-out Reset Enable bits
11 = BOR is enabled in hardware; SBOREN bit is disabled
10 = BOR is enabled only while device is active and disabled in Sleep; SBOREN bit is disabled
01 = BOR is controlled with the SBOREN bit setting
00 = BOR is disabled in hardware; SBOREN bit is disabled

Note 1: The MCLRE fuse can only be changed when using the VPP-Based ICSP™ mode entry. This prevents a user from accidentally locking out the device from the low-voltage test entry.
2: Refer to Table 26-5 for BOR trip point voltages.
3: Implemented in 28-pin devices only. This bit position must be programmed (= 1) in all other devices for I^2C functionality to be available.

23.3 Unique ID

A read-only Unique ID value is stored at addresses, 800802h through 800808h. This factory programmed value is unique to each microcontroller produced in the PIC24F16KL402 family. To access this region, use Table Read instructions or Program Space Visibility.

To ensure a globally Unique ID across other Microchip microcontroller families, the “Unique ID” value should be further concatenated with the family and Device ID values stored at address, FF0000h.

REGISTER 23-8: DEVID: DEVICE ID REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 23							bit 16

R	R	R	R	R	R	R	R
FAMID7	FAMID6	FAMID5	FAMID4	FAMID3	FAMID2	FAMID1	FAMID0
bit 15							bit 8

R	R	R	R	R	R	R	R
DEV7	DEV6	DEV5	DEV4	DEV3	DEV2	DEV1	DEV0
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 23-16 **Unimplemented:** Read as ‘0’

bit 15-8 **FAMID<7:0>:** Device Family Identifier bits
01001011 = PIC24F16KL402 family

bit 7-0 **DEV<7:0>:** Individual Device Identifier bits
00000001 = PIC24F04KL100
00000010 = PIC24F04KL101

00000101 = PIC24F08KL200
00000110 = PIC24F08KL201

00001010 = PIC24F08KL301
00000000 = PIC24F08KL302

00001110 = PIC24F08KL401
00000100 = PIC24F08KL402
00011110 = PIC24F16KL401
00010100 = PIC24F16KL402

24.11 Demonstration/Development Boards, Evaluation Kits and Starter Kits

A wide variety of demonstration, development and evaluation boards for various PIC MCUs and dsPIC DSCs allows quick application development on fully functional systems. Most boards include prototyping areas for adding custom circuitry and provide application firmware and source code for examination and modification.

The boards support a variety of features, including LEDs, temperature sensors, switches, speakers, RS-232 interfaces, LCD displays, potentiometers and additional EEPROM memory.

The demonstration and development boards can be used in teaching environments, for prototyping custom circuits and for learning about various microcontroller applications.

In addition to the PICDEM™ and dsPICDEM™ demonstration/development board series of circuits, Microchip has a line of evaluation kits and demonstration software for analog filter design, KEELOQ® security ICs, CAN, IrDA®, PowerSmart battery management, SEEVAL® evaluation system, Sigma-Delta ADC, flow rate sensing, plus many more.

Also available are starter kits that contain everything needed to experience the specified device. This usually includes a single application and debug capability, all on one board.

Check the Microchip web page (www.microchip.com) for the complete list of demonstration, development and evaluation kits.

24.12 Third-Party Development Tools

Microchip also offers a great collection of tools from third-party vendors. These tools are carefully selected to offer good value and unique functionality.

- Device Programmers and Gang Programmers from companies, such as SoftLog and CCS
- Software Tools from companies, such as Gimpel and Trace Systems
- Protocol Analyzers from companies, such as Saleae and Total Phase
- Demonstration Boards from companies, such as MikroElektronika, Digilent® and Olimex
- Embedded Ethernet Solutions from companies, such as EZ Web Lynx, WIZnet and IPLogika®

TABLE 25-2: INSTRUCTION SET OVERVIEW (CONTINUED)

Assembly Mnemonic	Assembly Syntax		Description	# of Words	# of Cycles	Status Flags Affected
GOTO	GOTO	Expr	Go to Address	2	2	None
	GOTO	Wn	Go to Indirect	1	2	None
INC	INC	f	f = f + 1	1	1	C, DC, N, OV, Z
	INC	f,WREG	WREG = f + 1	1	1	C, DC, N, OV, Z
	INC	Ws,Wd	Wd = Ws + 1	1	1	C, DC, N, OV, Z
INC2	INC2	f	f = f + 2	1	1	C, DC, N, OV, Z
	INC2	f,WREG	WREG = f + 2	1	1	C, DC, N, OV, Z
	INC2	Ws,Wd	Wd = Ws + 2	1	1	C, DC, N, OV, Z
IOR	IOR	f	f = f .IOR. WREG	1	1	N, Z
	IOR	f,WREG	WREG = f .IOR. WREG	1	1	N, Z
	IOR	#lit10,Wn	Wd = lit10 .IOR. Wd	1	1	N, Z
	IOR	Wb,Ws,Wd	Wd = Wb .IOR. Ws	1	1	N, Z
	IOR	Wb,#lit5,Wd	Wd = Wb .IOR. lit5	1	1	N, Z
LNK	LNK	#lit14	Link Frame Pointer	1	1	None
LSR	LSR	f	f = Logical Right Shift f	1	1	C, N, OV, Z
	LSR	f,WREG	WREG = Logical Right Shift f	1	1	C, N, OV, Z
	LSR	Ws,Wd	Wd = Logical Right Shift Ws	1	1	C, N, OV, Z
	LSR	Wb,Wns,Wnd	Wnd = Logical Right Shift Wb by Wns	1	1	N, Z
	LSR	Wb,#lit5,Wnd	Wnd = Logical Right Shift Wb by lit5	1	1	N, Z
MOV	MOV	f,Wn	Move f to Wn	1	1	None
	MOV	[Wns+Slit10],Wnd	Move [Wns+Slit10] to Wnd	1	1	None
	MOV	f	Move f to f	1	1	N, Z
	MOV	f,WREG	Move f to WREG	1	1	None
	MOV	#lit16,Wn	Move 16-bit Literal to Wn	1	1	None
	MOV.b	#lit8,Wn	Move 8-bit Literal to Wn	1	1	None
	MOV	Wn,f	Move Wn to f	1	1	None
	MOV	Wns,[Wns+Slit10]	Move Wns to [Wns+Slit10]	1	1	None
	MOV	Wso,Wdo	Move Ws to Wd	1	1	None
	MOV	WREG,f	Move WREG to f	1	1	None
	MOV.D	Wns,Wd	Move Double from W(ns):W(ns+1) to Wd	1	2	None
	MOV.D	Ws,Wnd	Move Double from Ws to W(nd+1):W(nd)	1	2	None
MUL	MUL.SS	Wb,Ws,Wnd	{Wnd+1, Wnd} = Signed(Wb) * Signed(Ws)	1	1	None
	MUL.SU	Wb,Ws,Wnd	{Wnd+1, Wnd} = Signed(Wb) * Unsigned(Ws)	1	1	None
	MUL.US	Wb,Ws,Wnd	{Wnd+1, Wnd} = Unsigned(Wb) * Signed(Ws)	1	1	None
	MUL.UU	Wb,Ws,Wnd	{Wnd+1, Wnd} = Unsigned(Wb) * Unsigned(Ws)	1	1	None
	MUL.SU	Wb,#lit5,Wnd	{Wnd+1, Wnd} = Signed(Wb) * Unsigned(lit5)	1	1	None
	MUL.UU	Wb,#lit5,Wnd	{Wnd+1, Wnd} = Unsigned(Wb) * Unsigned(lit5)	1	1	None
	MUL	f	W3:W2 = f * WREG	1	1	None
NEG	NEG	f	f = $\overline{f}$ + 1	1	1	C, DC, N, OV, Z
	NEG	f,WREG	WREG = $\overline{f}$ + 1	1	1	C, DC, N, OV, Z
	NEG	Ws,Wd	Wd = $\overline{Ws}$ + 1	1	1	C, DC, N, OV, Z
NOP	NOP		No Operation	1	1	None
	NOPR		No Operation	1	1	None
POP	POP	f	Pop f from Top-of-Stack (TOS)	1	1	None
	POP	Wdo	Pop from Top-of-Stack (TOS) to Wdo	1	1	None
	POP.D	Wnd	Pop from Top-of-Stack (TOS) to W(nd):W(nd+1)	1	2	None
	POP.S		Pop Shadow Registers	1	1	All
PUSH	PUSH	f	Push f to Top-of-Stack (TOS)	1	1	None
	PUSH	Wso	Push Wso to Top-of-Stack (TOS)	1	1	None
	PUSH.D	Wns	Push W(ns):W(ns+1) to Top-of-Stack (TOS)	1	2	None
	PUSH.S		Push Shadow Registers	1	1	None

TABLE 26-8: DC CHARACTERISTICS: POWER-DOWN CURRENT (I_{PD})

DC CHARACTERISTICS			Standard Operating Conditions: 1.8V to 3.6V Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended			
Parameter No.	**Typical(1)**	**Max**	**Units**	**Conditions**		
Power-Down Current (I_{PD})						
DC60	0.01	0.20	µA	-40°C	1.8V	Sleep Mode(2)
	0.03	0.20	µA	+25°C		
	0.06	0.87	µA	+60°C		
	0.20	1.35	µA	+85°C		
	—	8.00	µA	+125ºC		
	0.01	0.54	µA	-40°C	3.3V	
	0.03	0.54	µA	+25°C		
	0.08	1.68	µA	+60°C		
	0.25	2.45	µA	+85°C		
	—	10.00	µA	+125ºC		

Note 1: Data in the Typical column is at 3.3V, +25°C unless otherwise stated.

2: Base I_{PD} is measured with all peripherals and clocks disabled. All I/Os are configured as outputs and set low; PMDx bits are set to '1' and WDT, etc., are all disabled

26.2 AC Characteristics and Timing Parameters

The information contained in this section defines the PIC24F16KL402 Family AC characteristics and timing parameters.

TABLE 26-16: TEMPERATURE AND VOLTAGE SPECIFICATIONS – AC

AC CHARACTERISTICS	**Standard Operating Conditions: 1.8V to 3.6V** **Operating temperature -40°C ≤ TA ≤ +85°C for Industrial** Operating voltage VDD range as described in **Section 26.1 "DC Characteristics"**.

FIGURE 26-3: LOAD CONDITIONS FOR DEVICE TIMING SPECIFICATIONS

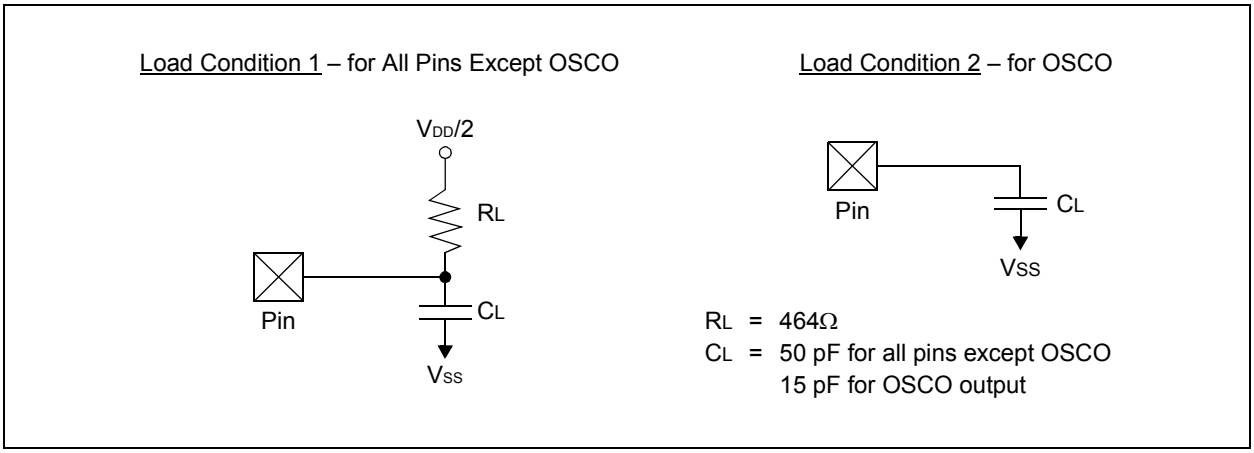


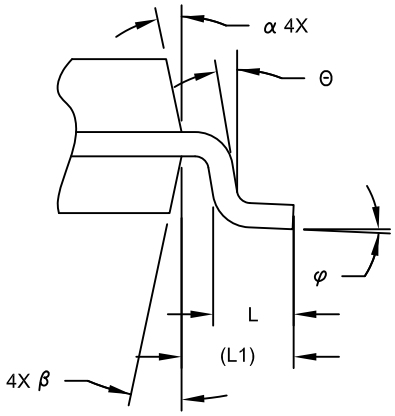
TABLE 26-17: CAPACITIVE LOADING REQUIREMENTS ON OUTPUT PINS

Param No.	Symbol	Characteristic	Min	Typ(1)	Max	Units	Conditions
DO50	COSC2	OSCO/CLKO Pin	—	—	15	pF	In XT and HS modes when external clock is used to drive OSCI
DO56	CIO	All I/O Pins and OSCO	—	—	50	pF	EC mode
DO58	CB	SCLx, SDAx	—	—	400	pF	In I^2C™ mode

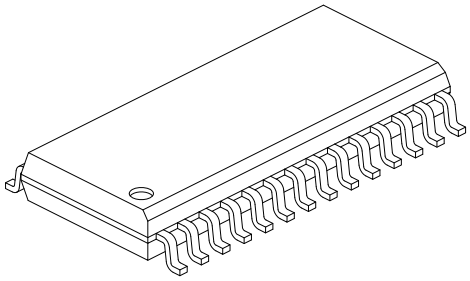
Note 1: Data in "Typ" column is at 3.3V, +25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

28-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

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



VIEW C



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		28	
Pitch	e		1.27 BSC	
Overall Height	A	-	-	2.65
Molded Package Thickness	A2	2.05	-	-
Standoff §	A1	0.10	-	0.30
Overall Width	E		10.30 BSC	
Molded Package Width	E1		7.50 BSC	
Overall Length	D		17.90 BSC	
Chamfer (Optional)	h	0.25	-	0.75
Foot Length	L	0.40	-	1.27
Footprint	L1		1.40 REF	
Lead Angle	Θ	0°	-	-
Foot Angle	φ	0°	-	8°
Lead Thickness	c	0.18	-	0.33
Lead Width	b	0.31	-	0.51
Mold Draft Angle Top	α	5°	-	15°
Mold Draft Angle Bottom	β	5°	-	15°

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic
3. Dimension D does not include mold flash, protrusions or gate burrs, which shall not exceed 0.15 mm per end. Dimension E1 does not include interlead flash or protrusion, which shall not exceed 0.25 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M

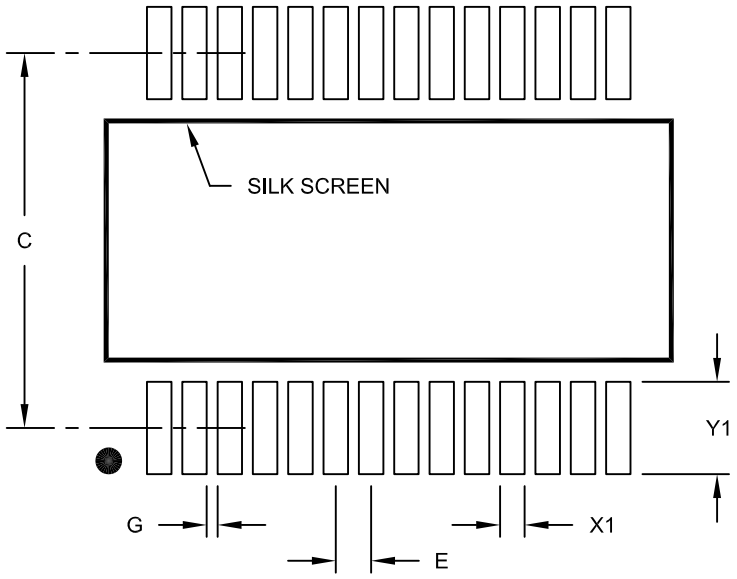
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

   REF: Reference Dimension, usually without tolerance, for information purposes only.
5. Datums A & B to be determined at Datum H.

Microchip Technology Drawing C04-052C Sheet 2 of 2

28-Lead Plastic Shrink Small Outline (SS) - 5.30 mm Body [SSOP]

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



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.65 BSC	
Contact Pad Spacing	C		7.20	
Contact Pad Width (X28)	X1			0.45
Contact Pad Length (X28)	Y1			1.75
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2073A