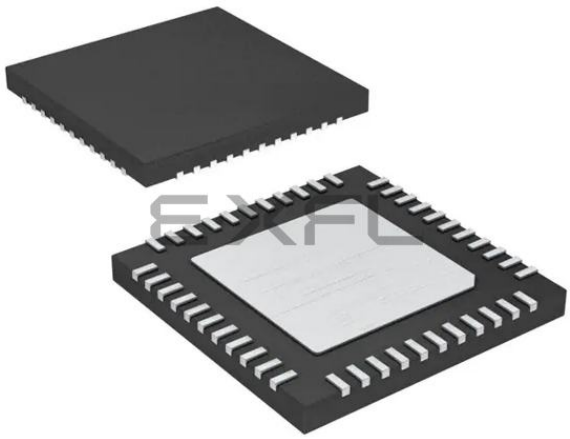


Welcome to E-XFL.COM

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	dsPIC
Core Size	16-Bit
Speed	30 MIPs
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, Motor Control PWM, QEI, POR, PWM, WDT
Number of I/O	20
Program Memory Size	24KB (8K x 24)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.5V ~ 5.5V
Data Converters	A/D 6x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-VQFN Exposed Pad
Supplier Device Package	44-QFN (8x8)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/dspic30f3010t-30i-ml

dsPIC30F3010/3011

High Performance, 16-Bit Digital Signal Controllers

Note: This data sheet summarizes features of this group of dsPIC30F devices and is not intended to be a complete reference source. For more information on the CPU, peripherals, register descriptions and general device functionality, refer to the "*dsPIC30F Family Reference Manual*" (DS70046). For more information on the device instruction set and programming, refer to the *"16-bit MCU and DSC Programmer's Reference Manual"* (DS70157).

High-Performance Modified RISC CPU:

- Modified Harvard Architecture
- C Compiler Optimized Instruction Set Architecture with Flexible Addressing modes
- 83 Base Instructions
- 24-bit Wide Instructions, 16-bit Wide Data Path
- 24 Kbytes On-Chip Flash Program Space (8K instruction words)
- 1 Kbyte of On-Chip Data RAM
- 1 Kbyte of Nonvolatile Data EEPROM
- 16 x 16-bit Working Register Array
- Up to 30 MIPs Operation:
  - DC to 40 MHz external clock input
  - 4 MHz-10 MHz oscillator input with PLL active (4x, 8x, 16x)
- 29 Interrupt Sources
  - 3 external interrupt sources
  - 8 user-selectable priority levels for each interrupt source
  - 4 processor trap sources

DSP Engine Features:

- Dual Data Fetch
- Accumulator Write Back for DSP Operations
- Modulo and Bit-Reversed Addressing modes
- Two, 40-bit Wide Accumulators with Optional saturation Logic
- 17-bit x 17-bit Single-Cycle Hardware Fractional/ Integer Multiplier
- All DSP Instructions Single Cycle
- ±16-bit Single-Cycle Shift

Peripheral Features:

- High-Current Sink/Source I/O Pins: 25 mA/25 mA
- Timer module with Programmable Prescaler:
  - Five 16-bit timers/counters; optionally pair 16-bit timers into 32-bit timer modules
- 16-bit Capture Input Functions
- 16-bit Compare/PWM Output Functions
- 3-Wire SPI modules (supports 4 Frame modes)
- I^2C™ module Supports Multi-Master/Slave mode and 7-bit/10-bit Addressing
- 2 UART modules with FIFO Buffers

Motor Control PWM Module Features:

- 6 PWM Output Channels
  - Complementary or Independent Output modes
  - Edge and Center-Aligned modes
- 3 Duty Cycle Generators
- Dedicated Time Base
- Programmable Output Polarity
- Dead-Time Control for Complementary mode
- Manual Output Control
- Trigger for A/D Conversions

Quadrature Encoder Interface Module Features:

- Phase A, Phase B and Index Pulse Input
- 16-bit Up/Down Position Counter
- Count Direction Status
- Position Measurement (x2 and x4) mode
- Programmable Digital Noise Filters on Inputs
- Alternate 16-bit Timer/Counter mode
- Interrupt on Position Counter Rollover/Underflow

Analog Features:

- 10-bit Analog-to-Digital Converter (ADC) with 4 Sample and Hold (S&H) Inputs:
  - 1 Msps conversion rate
  - 9 input channels
  - Conversion available during Sleep and Idle
- Programmable Brown-out Reset

2.0 CPU ARCHITECTURE OVERVIEW

Note:	This data sheet summarizes features of this group of dsPIC30F devices and is not intended to be a complete reference source. For more information on the CPU, peripherals, register descriptions and general device functionality, refer to the "*dsPIC30F Family Reference Manual*" (DS70046). For more information on the device instruction set and programming, refer to the "*16-bit MCU and DSC Programmer's Reference Manual*" (DS70157).

2.1 Core Overview

The core has a 24-bit instruction word. The Program Counter (PC) is 23 bits wide with the Least Significant bit (LSb) always clear (see **Section 3.1 "Program Address Space"**), and the Most Significant bit (MSb) is ignored during normal program execution, except for certain specialized instructions. Thus, the PC can address up to 4M instruction words of user program space. An instruction prefetch mechanism is used to help maintain throughput. Program loop constructs, free from loop count management overhead, are supported using the DO and REPEAT instructions, both of which are interruptible at any point.

The working register array consists of 16x16-bit registers, each of which can act as data, address or offset registers. One working register (W15) operates as a Software Stack Pointer (SP) for interrupts and calls.

The data space is 64 Kbytes (32K words) and is split into two blocks, referred to as X and Y data memory. Each block has its own independent Address Generation Unit (AGU). Most instructions operate solely through the X memory AGU, which provides the appearance of a single unified data space. The Multiply-Accumulate (MAC) class of dual source DSP instructions operate through both the X and Y AGUs, splitting the data address space into two parts (see **Section 3.2 "Data Address Space"**). The X and Y data space boundary is device specific and cannot be altered by the user. Each data word consists of 2 bytes, and most instructions can address data either as words or bytes.

There are two methods of accessing data stored in program memory:

- The upper 32 Kbytes of data space memory can be mapped into the lower half (user space) of program space at any 16K program word boundary, defined by the 8-bit Program Space Visibility Page (PSVPAG) register. This lets any instruction access program space as if it were data space, with a limitation that the access requires an additional cycle. Moreover, only the lower 16 bits of each instruction word can be accessed using this method.
- Linear indirect access of 32K word pages within program space is also possible using any working register, via table read and write instructions. Table read and write instructions can be used to access all 24 bits of an instruction word.

Overhead-free circular buffers (Modulo Addressing) are supported in both X and Y address spaces. This is primarily intended to remove the loop overhead for DSP algorithms.

The X AGU also supports Bit-Reversed Addressing on destination effective addresses, to greatly simplify input or output data reordering for radix-2 FFT algorithms. Refer to **Section 4.0 "Address Generator Units"** for details on Modulo and Bit-Reversed addressing.

The core supports Inherent (no operand), Relative, Literal, Memory Direct, Register Direct, Register Indirect, Register Offset and Literal Offset Addressing modes. Instructions are associated with predefined addressing modes, depending upon their functional requirements.

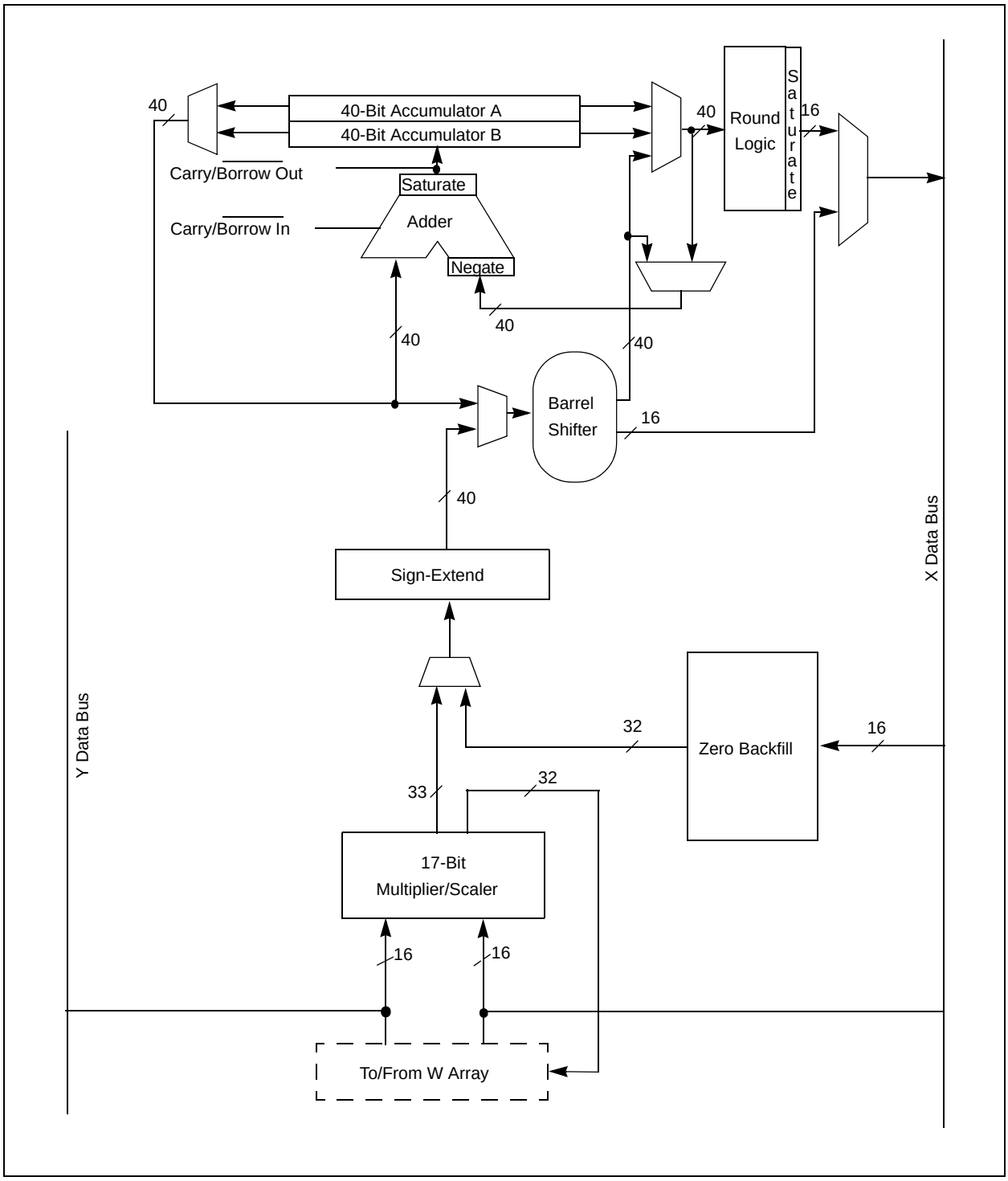
For most instructions, the core is capable of executing a data (or program data) memory read, a working register (data) read, a data memory write and a program (instruction) memory read per instruction cycle. As a result, 3 operand instructions are supported, allowing C = A + B operations to be executed in a single cycle.

A DSP engine has been included to significantly enhance the core arithmetic capability and throughput. It features a high-speed 17-bit by 17-bit multiplier, a 40-bit ALU, two 40-bit saturating accumulators and a 40-bit bidirectional barrel shifter. Data in the accumulator or any working register can be shifted up to 16 bits right or 16 bits left in a single cycle. The DSP instructions operate seamlessly with all other instructions and have been designed for optimal real-time performance. The MAC class of instructions can concurrently fetch two data operands from memory, while multiplying two W registers. To enable this concurrent fetching of data operands, the data space has been split for these instructions and linear for all others. This has been achieved in a transparent and flexible manner, by dedicating certain working registers to each address space for the MAC class of instructions.

The core does not support a multi-stage instruction pipeline. However, a single stage instruction prefetch mechanism is used, which accesses and partially decodes instructions a cycle ahead of execution, in order to maximize available execution time. Most instructions execute in a single cycle, with certain exceptions.

The core features a vectored exception processing structure for traps and interrupts, with 62 independent vectors. The exceptions consist of up to 8 traps (of which 4 are reserved) and 54 interrupts. Each interrupt is prioritized based on a user assigned priority between 1 and 7 (1 being the lowest priority and 7 being the highest) in conjunction with a predetermined 'natural order'. Traps have fixed priorities, ranging from 8 to 15.

FIGURE 2-2: DSP ENGINE BLOCK DIAGRAM



7.3 Writing to the Data EEPROM

To write an EEPROM data location, the following sequence must be followed:

1. Erase data EEPROM word.
   a) Select the word, data EEPROM, erase and set WREN bit in the NVMCON register.
   b) Write the address of word to be erased into the NVMADRU/NVMADR.
   c) Enable the NVM interrupt (optional).
   d) Write 0x55 to NVMKEY.
   e) Write 0xAA to NVMKEY.
   f) Set the WR bit. This will begin the erase cycle.
   g) Either poll the NVMIF bit or wait for the NVMIF interrupt.
   h) The WR bit is cleared when the erase cycle ends.
2. Write the data word into the data EEPROM write latches.
3. Program 1 data word into the data EEPROM.
   a) Select the word, data EEPROM, program and set the WREN bit in the NVMCON register.
   b) Enable the NVM write done interrupt (optional).
   c) Write 0x55 to NVMKEY.
   d) Write 0xAA to NVMKEY.
   e) Set the WR bit. This will begin the program cycle.
   f) Either poll the NVMIF bit or wait for the NVM interrupt.
   g) The WR bit is cleared when the write cycle ends.

The write will not initiate if the above sequence is not exactly followed (write 0x55 to NVMKEY, write 0xAA to NVMCON, then set WR bit) for each word. It is strongly recommended that interrupts be disabled during this code segment.

Additionally, the WREN bit in NVMCON must be set to enable writes. This mechanism prevents accidental writes to data EEPROM due to unexpected code execution. The WREN bit should be kept clear at all times, except when updating the EEPROM. The WREN bit is not cleared by hardware.

After a write sequence has been initiated, clearing the WREN bit will not affect the current write cycle. The WR bit will be inhibited from being set unless the WREN bit is set. The WREN bit must be set on a previous instruction. Both WR and WREN cannot be set with the same instruction.

At the completion of the write cycle, the WR bit is cleared in hardware and the Nonvolatile Memory Write Complete Interrupt Flag bit (NVMIF) is set. The user may either enable this interrupt, or poll this bit. NVMIF must be cleared by software.

7.3.1 WRITING A WORD OF DATA EEPROM

Once the user has erased the word to be programmed, then a table write instruction is used to write one write latch, as shown in Example 7-4.

EXAMPLE 7-4: DATA EEPROM WORD WRITE

```
; Point to data memory
    MOV             #LOW_ADDR_WORD,W0                ; Init pointer
    MOV             #HIGH_ADDR_WORD,W1
    MOV             W1,TBLPAG
    MOV             #LOW(WORD),W2                    ; Get data
    TBLWTL          W2,[ W0]                         ; Write data
; The NVMADR captures last table access address
; Select data EEPROM for 1 word op
    MOV             #0x4004,W0
    MOV             W0,NVMCON

; Operate key to allow write operation
    DISI    #5                                       ; Block all interrupts with priority <7
                                                     ; for next 5 instructions
    MOV             #0x55,W0
    MOV             W0,NVMKEY                        ; Write the 0x55 key
    MOV             #0xAA,W1
    MOV             W1,NVMKEY                        ; Write the 0xAA key
    BSET            NVMCON,#WR                       ; Initiate program sequence
    NOP
    NOP
; Write cycle will complete in 2mS. CPU is not stalled for the Data Write Cycle
; User can poll WR bit, use NVMIF or Timer IRQ to determine write complete
```

TABLE 8-1: dsPIC30F3011 PORT REGISTER MAP(1)

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset State
TRISB	02C6	—	—	—	—	—	—	—	TRISB8	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	0000 0001 1111 1111
PORTB	02C8	—	—	—	—	—	—	—	RB8	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	0000 0000 0000 0000
LATB	02CA	—	—	—	—	—	—	—	LATB8	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	0000 0000 0000 0000
TRISC	02CC	TRISC15	TRISC14	TRISC13	—	—	—	—	—	—	—	—	—	—	—	—	—	1110 0000 0000 0000
PORTC	02CE	RC15	RC14	RC13	—	—	—	—	—	—	—	—	—	—	—	—	—	0000 0000 0000 0000
LATC	02D0	LATC15	LATC14	LATC13	—	—	—	—	—	—	—	—	—	—	—	—	—	0000 0000 0000 0000
TRISD	02D2	—	—	—	—	—	—	—	—	—	—	—	—	TRISD3	TRISD2	TRISD1	TRISD0	0000 0000 0000 1111
PORTD	02D4	—	—	—	—	—	—	—	—	—	—	—	—	RD3	RD2	RD1	RD0	0000 0000 0000 0000
LATD	02D6	—	—	—	—	—	—	—	—	—	—	—	—	LATD3	LATD2	LATD1	LATD0	0000 0000 0000 0000
TRISE	02D8	—	—	—	—	—	—	—	TRISE8	—	—	TRISE5	TRISE4	TRISE3	TRISE2	TRISE1	TRISE0	0000 0001 0011 1111
PORTE	02DA	—	—	—	—	—	—	—	RE8	—	—	RE5	RE4	RE3	RE2	RE1	RE0	0000 0000 0000 0000
LATE	02DC	—	—	—	—	—	—	—	LATE8	—	—	LATE5	LATE4	LATE3	LATE2	LATE1	LATE0	0000 0000 0000 0000
TRISF	02DE	—	—	—	—	—	—	—	—	—	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	TRISF0	0000 0000 0111 1111
PORTF	02E0	—	—	—	—	—	—	—	—	—	RF6	RF5	RF4	RF3	RF2	RF1	RF0	0000 0000 0000 0000
LATF	02E2	—	—	—	—	—	—	—	—	—	LATF6	LATF5	LATF4	LATF3	LATF2	LATF1	LATF0	0000 0000 0000 0000

Legend: — = unimplemented bit, read as '0'

Note 1: Refer to the *"dsPIC30F Family Reference Manual"* (DS70046) for descriptions of register bit fields. Not all peripherals, and therefore their bit positions, are available on this device.

TABLE 11-1: TIMER4/5 REGISTER MAP(1)

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset State
TMR4	0114	Timer4 Register																uuuu uuuu uuuu uuuu
TMR5HLD	0116	Timer5 Holding Register (For 32-bit operations only)																uuuu uuuu uuuu uuuu
TMR5	0118	Timer5 Register																uuuu uuuu uuuu uuuu
PR4	011A	Period Register 4																1111 1111 1111 1111
PR5	011C	Period Register 5																1111 1111 1111 1111
T4CON	011E	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	T45	—	TCS	—	0000 0000 0000 0000
T5CON	0120	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	—	TCS	—	0000 0000 0000 0000

Legend: u = uninitialized bit; — = unimplemented bit, read as '0'

Note 1: Refer to *"dsPIC30F Family Reference Manual"* (DS70046) for descriptions of register bit fields.

NOTES:

14.1 Quadrature Encoder Interface Logic

A typical incremental (a.k.a. optical) encoder has three outputs: Phase A, Phase B and an index pulse. These signals are useful and often required in position and speed control of ACIM and SR motors.

The two channels, Phase A (QEA) and Phase B (QEB), have a unique relationship. If Phase A leads Phase B, then the direction (of the motor) is deemed positive or forward. If Phase A lags Phase B, then the direction (of the motor) is deemed negative or reverse.

A third channel, termed index pulse, occurs once per revolution and is used as a reference to establish an absolute position. The index pulse coincides with Phase A and Phase B, both low.

14.2 16-Bit Up/Down Position Counter Mode

The 16-bit up/down counter counts up or down on every count pulse, which is generated by the difference of the Phase A and Phase B input signals. The counter acts as an integrator, whose count value is proportional to position. The direction of the count is determined by the UPDN signal, which is generated by the Quadrature Encoder Interface logic.

14.2.1 POSITION COUNTER ERROR CHECKING

Position count error checking in the QEI is provided for and indicated by the CNTERR bit (QEICON<15>). The error checking only applies when the position counter is configured for Reset on the Index Pulse modes (QEIM<2:0> = 110 or 100). In these modes, the contents of the POSCNT register are compared with the values (0xFFFF or MAXCNT + 1, depending on direction). If these values are detected, an error condition is generated by setting the CNTERR bit and a QEI count error interrupt is generated. The QEI count error interrupt can be disabled by setting the CEID bit (DFLTCON<8>). The position counter continues to count encoder edges after an error has been detected. The POSCNT register continues to count up/down until a natural rollover/underflow. No interrupt is generated for the natural rollover/underflow event. The CNTERR bit is a read/write bit and reset in software by the user.

14.2.2 POSITION COUNTER RESET

The Position Counter Reset Enable bit, POSRES (QEI<2>), controls whether the position counter is reset when the index pulse is detected. This bit is only applicable when QEIM<2:0> = 100 or 110.

If the POSRES bit is set to '1', then the position counter is reset when the index pulse is detected. If the POSRES bit is set to '0', then the position counter is not reset when the index pulse is detected. The position counter will continue counting up or down, and will be reset on the rollover or underflow condition.

When selecting the INDX signal to reset the Position Counter (POSCNT), the user has to specify the states on QEA and QEB input pins. These states have to be matched in order for a Reset to occur. These states are selected by the IMV<1:0> bits in the DFLTCON register.

The IMV<1:0> (Index Match Value) bits allow the user to specify the state of the QEA and QEB input pins during an index pulse when the POSCNT register is to be reset.

In x4 Quadrature Count mode:
IMV1 = Required state of Phase B input signal for match on index pulse
IMV0 = Required state of Phase A input signal for match on index pulse

In x2 Quadrature Count mode:
IMV1 = Selects phase input signal for index state match (0 = Phase A, 1 = Phase B)
IMV0 = Required state of the selected phase input signal for match on index pulse

The interrupt is still generated on the detection of the index pulse and not on the position counter overflow/underflow.

14.2.3 COUNT DIRECTION STATUS

As mentioned in the previous section, the QEI logic generates an UPDN signal based upon the relationship between Phase A and Phase B. In addition to the output pin, the state of this internal UPDN signal is supplied to a SFR bit, UPDN (QEICON<11>), as a read-only bit.

Note:	QEI pins are multiplexed with analog inputs. The user must insure that all QEI associated pins are set as digital inputs in the ADPCFG register.

TABLE 14-1: QEI REGISTER MAP(1)

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset State
QEICON	0122	CNTERR	—	QEISIDL	INDX	UPDN	QEIM2	QEIM1	QEIM0	SWPAB	—	TQGATE	TQCKPS1	TQCKPS0	POSRES	TQCS	UPDN_SRC	0000 0000 0000 0000
DFLTCON	0124	—	—	—	—	—	IMV1	IMV0	CEID	QEOUT	QECK2	QECK1	QECK0	—	—	—	—	0000 0000 0000 0000
POSCNT	0126	Position Counter<15:0>																0000 0000 0000 0000
MAXCNT	0128	Maximun Count<15:0>																1111 1111 1111 1111
ADPCFG	02A8	—	—	—	—	—	—	—	PCFG8	PCFG7	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	0000 0000 0000 0000

Legend: — = unimplemented bit, read as '0'

Note 1: Refer to the *"dsPIC30F Family Reference Manual"* (DS70046) for descriptions of register bit fields.

NOTES:

TABLE 21-2: INSTRUCTION SET OVERVIEW (CONTINUED)

Base Instr #	Assembly Mnemonic	Assembly Syntax		Description	# of words	# of cycles	Status Flags Affected
63	RLC	RLC	f	f = Rotate Left through Carry f	1	1	C,N,Z
		RLC	f,WREG	WREG = Rotate Left through Carry f	1	1	C,N,Z
		RLC	Ws,Wd	Wd = Rotate Left through Carry Ws	1	1	C,N,Z
64	RLNC	RLNC	f	f = Rotate Left (No Carry) f	1	1	N,Z
		RLNC	f,WREG	WREG = Rotate Left (No Carry) f	1	1	N,Z
		RLNC	Ws,Wd	Wd = Rotate Left (No Carry) Ws	1	1	N,Z
65	RRC	RRC	f	f = Rotate Right through Carry f	1	1	C,N,Z
		RRC	f,WREG	WREG = Rotate Right through Carry f	1	1	C,N,Z
		RRC	Ws,Wd	Wd = Rotate Right through Carry Ws	1	1	C,N,Z
66	RRNC	RRNC	f	f = Rotate Right (No Carry) f	1	1	N,Z
		RRNC	f,WREG	WREG = Rotate Right (No Carry) f	1	1	N,Z
		RRNC	Ws,Wd	Wd = Rotate Right (No Carry) Ws	1	1	N,Z
67	SAC	SAC	Acc,#Slit4,Wdo	Store Accumulator	1	1	None
		SAC.R	Acc,#Slit4,Wdo	Store Rounded Accumulator	1	1	None
68	SE	SE	Ws,Wnd	Wnd = Sign-Extended Ws	1	1	C,N,Z
69	SETM	SETM	f	f = 0xFFFF	1	1	None
		SETM	WREG	WREG = 0xFFFF	1	1	None
		SETM	Ws	Ws = 0xFFFF	1	1	None
70	SFTAC	SFTAC	Acc,Wn	Arithmetic Shift Accumulator by (Wn)	1	1	OA,OB,OAB, SA,SB,SAB
		SFTAC	Acc,#Slit6	Arithmetic Shift Accumulator by Slit6	1	1	OA,OB,OAB, SA,SB,SAB
71	SL	SL	f	f = Left Shift f	1	1	C,N,OV,Z
		SL	f,WREG	WREG = Left Shift f	1	1	C,N,OV,Z
		SL	Ws,Wd	Wd = Left Shift Ws	1	1	C,N,OV,Z
		SL	Wb,Wns,Wnd	Wnd = Left Shift Wb by Wns	1	1	N,Z
		SL	Wb,#lit5,Wnd	Wnd = Left Shift Wb by lit5	1	1	N,Z
72	SUB	SUB	Acc	Subtract Accumulators	1	1	OA,OB,OAB, SA,SB,SAB
		SUB	f	f = f – WREG	1	1	C,DC,N,OV,Z
		SUB	f,WREG	WREG = f – WREG	1	1	C,DC,N,OV,Z
		SUB	#lit10,Wn	Wn = Wn – lit10	1	1	C,DC,N,OV,Z
		SUB	Wb,Ws,Wd	Wd = Wb – Ws	1	1	C,DC,N,OV,Z
		SUB	Wb,#lit5,Wd	Wd = Wb – lit5	1	1	C,DC,N,OV,Z
73	SUBB	SUBB	f	f = f – WREG – ($\overline{C}$)	1	1	C,DC,N,OV,Z
		SUBB	f,WREG	WREG = f – WREG – ($\overline{C}$)	1	1	C,DC,N,OV,Z
		SUBB	#lit10,Wn	Wn = Wn – lit10 – ($\overline{C}$)	1	1	C,DC,N,OV,Z
		SUBB	Wb,Ws,Wd	Wd = Wb – Ws – ($\overline{C}$)	1	1	C,DC,N,OV,Z
		SUBB	Wb,#lit5,Wd	Wd = Wb – lit5 – ($\overline{C}$)	1	1	C,DC,N,OV,Z
74	SUBR	SUBR	f	f = WREG – f	1	1	C,DC,N,OV,Z
		SUBR	f,WREG	WREG = WREG – f	1	1	C,DC,N,OV,Z
		SUBR	Wb,Ws,Wd	Wd = Ws – Wb	1	1	C,DC,N,OV,Z
		SUBR	Wb,#lit5,Wd	Wd = lit5 – Wb	1	1	C,DC,N,OV,Z
75	SUBBR	SUBBR	f	f = WREG – f – ($\overline{C}$)	1	1	C,DC,N,OV,Z
		SUBBR	f,WREG	WREG = WREG – f – ($\overline{C}$)	1	1	C,DC,N,OV,Z
		SUBBR	Wb,Ws,Wd	Wd = Ws – Wb – ($\overline{C}$)	1	1	C,DC,N,OV,Z
		SUBBR	Wb,#lit5,Wd	Wd = lit5 – Wb – ($\overline{C}$)	1	1	C,DC,N,OV,Z
76	SWAP	SWAP.b	Wn	Wn = Nibble Swap Wn	1	1	None
		SWAP	Wn	Wn = Byte Swap Wn	1	1	None
77	TBLRDH	TBLRDH	Ws,Wd	Read Prog<23:16> to Wd<7:0>	1	2	None
78	TBLRDL	TBLRDL	Ws,Wd	Read Prog<15:0> to Wd	1	2	None
79	TBLWTH	TBLWTH	Ws,Wd	Write Ws<7:0> to Prog<23:16>	1	2	None

TABLE 21-2: INSTRUCTION SET OVERVIEW (CONTINUED)

Base Instr #	Assembly Mnemonic	Assembly Syntax		Description	# of words	# of cycles	Status Flags Affected
80	TBLWTL	TBLWTL	Ws,Wd	Write Ws to Prog<15:0>	1	2	None
81	ULNK	ULNK		Unlink Frame Pointer	1	1	None
82	XOR	XOR	f	f = f .XOR. WREG	1	1	N,Z
		XOR	f,WREG	WREG = f .XOR. WREG	1	1	N,Z
		XOR	#lit10,Wn	Wd = lit10 .XOR. Wd	1	1	N,Z
		XOR	Wb,Ws,Wd	Wd = Wb .XOR. Ws	1	1	N,Z
		XOR	Wb,#lit5,Wd	Wd = Wb .XOR. lit5	1	1	N,Z
83	ZE	ZE	Ws,Wnd	Wnd = Zero-Extend Ws	1	1	C,Z,N

22.11 PICkit 2 Development Programmer/Debugger and PICkit 2 Debug Express

The PICkit™ 2 Development Programmer/Debugger is a low-cost development tool with an easy to use interface for programming and debugging Microchip's Flash families of microcontrollers. The full featured Windows® programming interface supports baseline (PIC10F, PIC12F5xx, PIC16F5xx), midrange (PIC12F6xx, PIC16F), PIC18F, PIC24, dsPIC30, dsPIC33, and PIC32 families of 8-bit, 16-bit, and 32-bit microcontrollers, and many Microchip Serial EEPROM products. With Microchip's powerful MPLAB Integrated Development Environment (IDE) the PICkit™ 2 enables in-circuit debugging on most PIC® microcontrollers. In-Circuit-Debugging runs, halts and single steps the program while the PIC microcontroller is embedded in the application. When halted at a breakpoint, the file registers can be examined and modified.

The PICkit 2 Debug Express include the PICkit 2, demo board and microcontroller, hookup cables and CDROM with user's guide, lessons, tutorial, compiler and MPLAB IDE software.

22.12 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at VDDMIN and VDDMAX for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages and a modular, detachable socket assembly to support various package types. The ICSP™ cable assembly is included as a standard item. In Stand-Alone mode, the MPLAB PM3 Device Programmer can read, verify and program PIC devices without a PC connection. It can also set code protection in this mode. The MPLAB PM3 connects to the host PC via an RS-232 or USB cable. The MPLAB PM3 has high-speed communications and optimized algorithms for quick programming of large memory devices and incorporates an MMC card for file storage and data applications.

22.13 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.

23.1 DC Characteristics

TABLE 23-1: OPERATING MIPS VS. VOLTAGE

VDD Range	Temp Range	Max MIPS	
		dsPIC30F301X-30I	dsPIC30F301X-20E
4.5-5.5V	-40°C to 85°C	30	—
4.5-5.5V	-40°C to 125°C	—	20
3.0-3.6V	-40°C to 85°C	20	—
3.0-3.6V	-40°C to 125°C	—	15
2.5-3.0V	-40°C to 85°C	10	—

TABLE 23-2: THERMAL OPERATING CONDITIONS

Rating	Symbol	Min	Typ	Max	Unit
dsPIC30F301X-30I					
Operating Junction Temperature Range	TJ	-40	—	+125	°C
Operating Ambient Temperature Range	TA	-40	—	+85	°C
dsPIC30F301X-20E					
Operating Junction Temperature Range	TJ	-40	—	+150	°C
Operating Ambient Temperature Range	TA	-40	—	+125	°C
Power Dissipation: Internal chip power dissipation: $P_{INT} = V_{DD} \times (I_{DD} - \sum I_{OH})$ I/O Pin power dissipation: $P_{I/O} = \sum (\{V_{DD} - V_{OH}\} \times I_{OH}) + \sum (V_{OL} \times I_{OL})$	PD	PINT + PI/O			W
Maximum Allowed Power Dissipation	PDMAX	(TJ – TA)/θJA			W

TABLE 23-3: THERMAL PACKAGING CHARACTERISTICS

Characteristic	Symbol	Typ	Max	Unit	Notes
Package Thermal Resistance, 28-pin SPDIP (SP)	θJA	42	—	°C/W	**1**
Package Thermal Resistance, 28-pin SOIC (SO)	θJA	49	—	°C/W	**1**
Package Thermal Resistance, 40-pin PDIP (P)	θJA	37	—	°C/W	**1**
Package Thermal Resistance, 44-pin TQFP (PT, 10x10x1 mm)	θJA	45	—	°C/W	**1**
Package Thermal Resistance, 44-pin QFN (ML)	θJA	28	—	°C/W	**1**

Note 1: Junction to ambient thermal resistance, Theta-ja (θJA) numbers are achieved by package simulations.

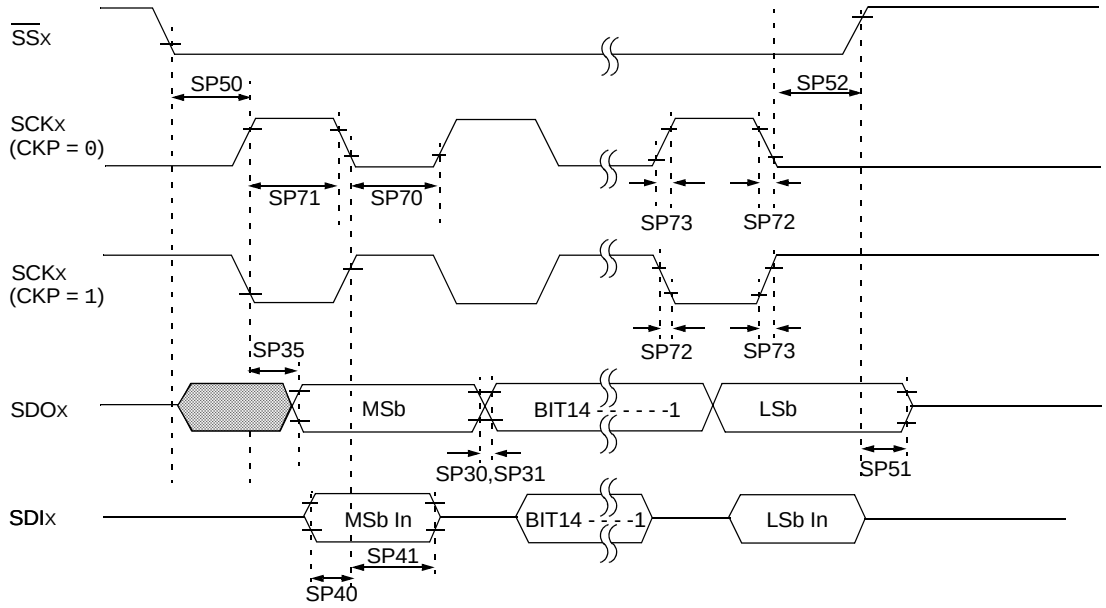
TABLE 23-23: TIMER2 AND TIMER4 EXTERNAL CLOCK TIMING REQUIREMENTS

AC CHARACTERISTICS				Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	**Symbol**	**Characteristic**		**Min**	**Typ**	**Max**	**Units**	**Conditions**
TB10	TtxH	TxCK High Time	Synchronous, no prescaler	0.5 TCY + 20	—	—	ns	Must also meet parameter TB15
			Synchronous, with prescaler	10	—	—	ns	
TB11	TtxL	TxCK Low Time	Synchronous, no prescaler	0.5 TCY + 20	—	—	ns	Must also meet parameter TB15
			Synchronous, with prescaler	10	—	—	ns	
TB15	TtxP	TxCK Input Period	Synchronous, no prescaler	TCY + 10	—	—	ns	N = prescale value (1, 8, 64, 256)
			Synchronous, with prescaler	Greater of: 20 ns or (TCY + 40)/N				
TB20	TCKEXTMRL	Delay from External TxCK Clock Edge to Timer Increment		0.5 TCY	—	1.5 TCY	—	

TABLE 23-24: TIMER3 AND TIMER5 EXTERNAL CLOCK TIMING REQUIREMENTS

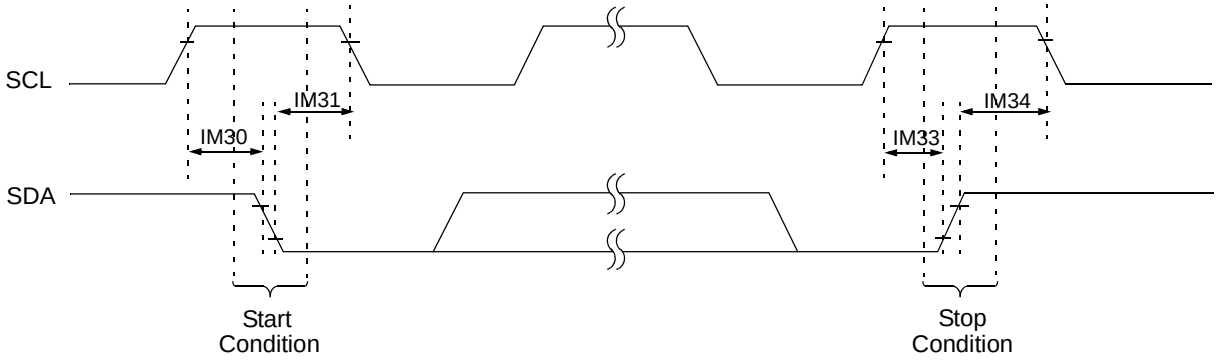
AC CHARACTERISTICS				Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	**Symbol**	**Characteristic**		**Min**	**Typ**	**Max**	**Units**	**Conditions**
TC10	TtxH	TxCK High Time	Synchronous	0.5 TCY + 20	—	—	ns	Must also meet parameter TC15
TC11	TtxL	TxCK Low Time	Synchronous	0.5 TCY + 20	—	—	ns	Must also meet parameter TC15
TC15	TtxP	TxCK Input Period	Synchronous, no prescaler	TCY + 10	—	—	ns	N = prescale value (1, 8, 64, 256)
			Synchronous, with prescaler	Greater of: 20 ns or (TCY + 40)/N				
TC20	TCKEXTMRL	Delay from External TxCK Clock Edge to Timer Increment		0.5 TCY	—	1.5 TCY	—	

FIGURE 23-18: SPI MODULE SLAVE MODE (CKE = 0) TIMING CHARACTERISTICS



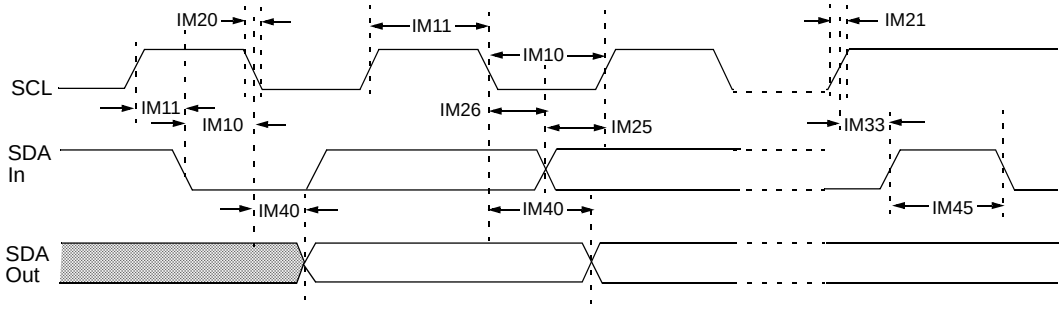
Note: Refer to Figure 23-2 for load conditions.

FIGURE 23-20: I²C™ BUS START/STOP BITS TIMING CHARACTERISTICS (MASTER MODE)



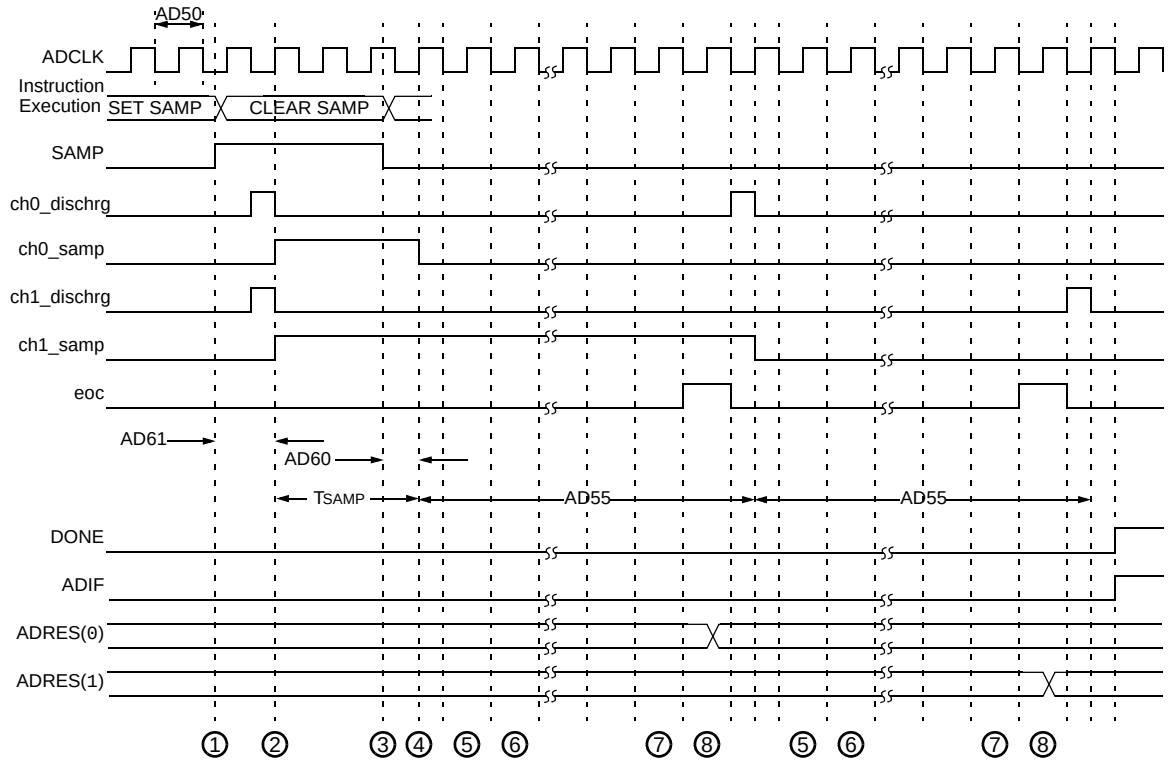
Note: Refer to Figure 23-2 for load conditions.

FIGURE 23-21: I²C™ BUS DATA TIMING CHARACTERISTICS (MASTER MODE)



Note: Refer to Figure 23-2 for load conditions.

FIGURE 23-24: 10-BIT HIGH-SPEED ADC TIMING CHARACTERISTICS (CHPS = 01, SIMSAM = 0, ASAM = 0, SSRC = 000)



1 — Software sets ADCON. SAMP to start sampling.

2 — Sampling starts after discharge period.
TSAMP is described in **Section 17, "10-Bit A/D Converter"** of the *"dsPIC30F Family Reference Manual"*, (DS70046).

3 — Software clears ADCON. SAMP to start conversion.

4 — Sampling ends, conversion sequence starts.

5 — Convert bit 9.

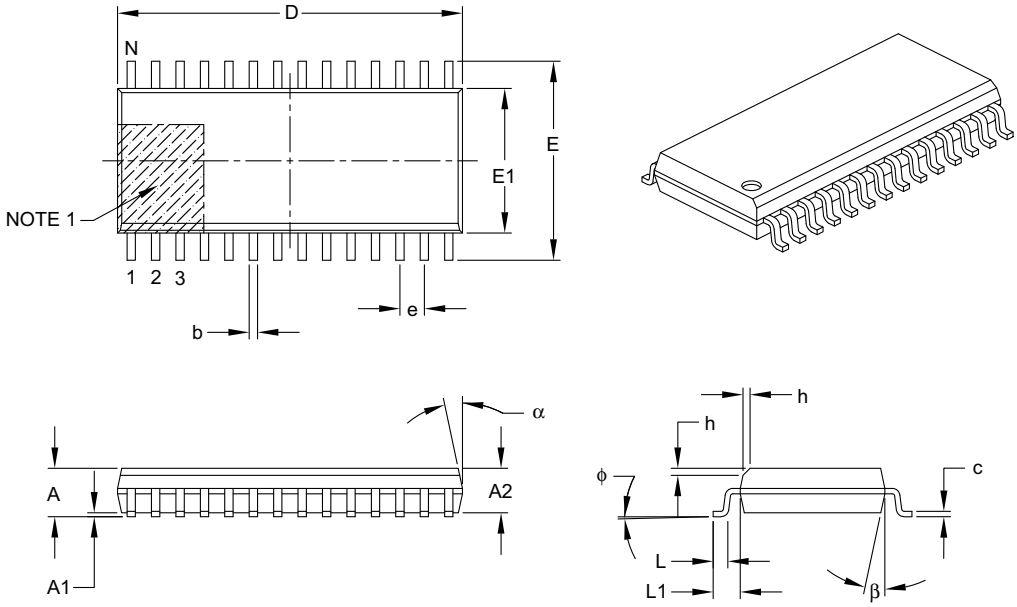
6 — Convert bit 8.

8 — Convert bit 0.

9 — One TAD for end of conversion.

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



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	
Foot Angle Top	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. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15 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.

Microchip Technology Drawing C04-052B

NOTES: