



Welcome to [E-XFL.COM](https://www.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               | CANbus, I <sup>2</sup> C, SPI, UART/USART   |
| Peripherals                | Brown-out Detect/Reset, Motor Control PWM, QEI, POR, PWM, WDT   |
| Number of I/O              | 68  |
| Program Memory Size        | 66KB (22K x 24)   |
| Program Memory Type        | FLASH   |
| EEPROM Size                | 1K x 8  |
| RAM Size                   | 2K x 8  |
| Voltage - Supply (Vcc/Vdd) | 2.5V ~ 5.5V   |
| Data Converters            | A/D 16x10b  |
| Oscillator Type            | Internal  |
| Operating Temperature      | -40°C ~ 85°C (TA)   |
| Mounting Type              | Surface Mount   |
| Package / Case             | 80-TQFP   |
| Supplier Device Package    | 80-TQFP (12x12)   |
| Purchase URL               | <a href="https://www.e-xfl.com/product-detail/microchip-technology/dspic30f5016t-30i-pt">https://www.e-xfl.com/product-detail/microchip-technology/dspic30f5016t-30i-pt</a> |

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.

## 2.2 Programmer's Model

The programmer's model is shown in Figure 2-1 and consists of 16x16-bit working registers (W0 through W15), 2x40-bit accumulators (ACCA and ACCB), STATUS register (SR), Data Table Page register (TBLPAG), Program Space Visibility Page register (PSVPAG), DO and REPEAT registers (DOSTART, DOEND, DCOUNT and RCOUNT) and Program Counter (PC). The working registers can act as data, address or offset registers. All registers are memory mapped. W0 acts as the W register for file register addressing.

Some of these registers have a shadow register associated with each of them, as shown in Figure 2-1. The shadow register is used as a temporary holding register and can transfer its contents to or from its host register upon the occurrence of an event. None of the shadow registers are accessible directly. The following rules apply for transfer of registers into and out of shadows.

- `PUSH.S` and `POP.S`  
W0, W1, W2, W3, SR (DC, N, OV, Z and C bits only) are transferred.
- `DO` instruction  
DOSTART, DOEND, DCOUNT shadows are pushed on loop start, and popped on loop end.

When a byte operation is performed on a working register, only the Least Significant Byte (LSB) of the target register is affected. However, a benefit of memory mapped working registers is that both the Least and Most Significant Bytes (MSBs) can be manipulated through byte-wide data memory space accesses.

### 2.2.1 SOFTWARE STACK POINTER/ FRAME POINTER

The dsPIC DSC devices contain a software stack. W15 is the dedicated software Stack Pointer (SP), and will be automatically modified by exception processing and subroutine calls and returns. However, W15 can be referenced by any instruction in the same manner as all other W registers. This simplifies the reading, writing and manipulation of the Stack Pointer (e.g., creating stack frames).

**Note:** In order to protect against misaligned stack accesses, W15<0> is always clear.

W15 is initialized to 0x0800 during a Reset. The user may reprogram the SP during initialization to any location within data space.

W14 has been dedicated as a Stack Frame Pointer as defined by the `LNK` and `ULNK` instructions. However, W14 can be referenced by any instruction in the same manner as all other W registers.

### 2.2.2 STATUS REGISTER

The dsPIC DSC core has a 16-bit STATUS register (SR), the LSB of which is referred to as the SR Low Byte (SRL) and the MSB as the SR High Byte (SRH). See Figure 2-1 for SR layout.

SRL contains all the MCU ALU operation Status flags (including the Z bit), as well as the CPU Interrupt Priority Level Status bits, IPL<2:0>, and the Repeat Active Status bit, RA. During exception processing, SRL is concatenated with the MSB of the PC to form a complete word value which is then stacked.

The upper byte of the SR register contains the DSP Adder/Subtractor Status bits, the DO Loop Active bit (DA) and the Digit Carry (DC) Status bit.

### 2.2.3 PROGRAM COUNTER

The Program Counter is 23 bits wide. Bit 0 is always clear. Therefore, the PC can address up to 4M instruction words.

The SA and SB bits are modified each time data passes through the adder/subtractor, but can only be cleared by the user. When set, they indicate that the accumulator has overflowed its maximum range (bit 31 for 32-bit saturation, or bit 39 for 40-bit saturation) and will be saturated (if saturation is enabled). When saturation is not enabled, SA and SB default to bit 39 overflow and thus indicate that a catastrophic overflow has occurred. If the COVTE bit in the INTCON1 register is set, SA and SB bits will generate an arithmetic warning trap when saturation is disabled.

The Overflow and Saturation Status bits can optionally be viewed in the STATUS register (SR) as the logical OR of OA and OB (in bit OAB) and the logical OR of SA and SB (in bit SAB). This allows programmers to check one bit in the STATUS register to determine if either accumulator has overflowed, or one bit to determine if either accumulator has saturated. This would be useful for complex number arithmetic which typically uses both the accumulators.

The device supports three Saturation and Overflow modes.

1. **Bit 39 Overflow and Saturation:**  
When bit 39 overflow and saturation occurs, the saturation logic loads the maximally positive 9.31 (0x7FFFFFFF) or maximally negative 9.31 value (0x80000000) into the target accumulator. The SA or SB bit is set and remains set until cleared by the user. This is referred to as 'super saturation' and provides protection against erroneous data or unexpected algorithm problems (e.g., gain calculations).
2. **Bit 31 Overflow and Saturation:**  
When bit 31 overflow and saturation occurs, the saturation logic then loads the maximally positive 1.31 value (0x007FFFFFFF) or maximally negative 1.31 value (0x00800000) into the target accumulator. The SA or SB bit is set and remains set until cleared by the user. When this Saturation mode is in effect, the guard bits are not used (so the OA, OB or OAB bits are never set).
3. **Bit 39 Catastrophic Overflow**  
The bit 39 Overflow Status bit from the adder is used to set the SA or SB bit, which remain set until cleared by the user. No saturation operation is performed and the accumulator is allowed to overflow (destroying its sign). If the COVTE bit in the INTCON1 register is set, a catastrophic overflow can initiate a trap exception.

## 2.4.2.2 Accumulator 'Write Back'

The MAC class of instructions (with the exception of MPY, MPY.N, ED and EDAC) can optionally write a rounded version of the high word (bits 31 through 16) of the accumulator that is not targeted by the instruction into data space memory. The write is performed across the X bus into combined X and Y address space. The following addressing modes are supported:

- **W13, Register Direct:**  
The rounded contents of the non-target accumulator are written into W13 as a 1.15 fraction.
- **[W13]+ = 2, Register Indirect with Post-Increment:**  
The rounded contents of the non-target accumulator are written into the address pointed to by W13 as a 1.15 fraction. W13 is then incremented by 2 (for a word write).

## 2.4.2.3 Round Logic

The round logic is a combinational block, which performs a conventional (biased) or convergent (unbiased) round function during an accumulator write (store). The Round mode is determined by the state of the RND bit in the CORCON register. It generates a 16-bit, 1.15 data value which is passed to the data space write saturation logic. If rounding is not indicated by the instruction, a truncated 1.15 data value is stored and the least significant word (lsw) is simply discarded.

Conventional rounding takes bit 15 of the accumulator, zero-extends it and adds it to the ACCxH word (bits 16 through 31 of the accumulator). If the ACCxL word (bits 0 through 15 of the accumulator) is between 0x8000 and 0xFFFF (0x8000 included), ACCxH is incremented. If ACCxL is between 0x0000 and 0x7FFF, ACCxH is left unchanged. A consequence of this algorithm is that over a succession of random rounding operations, the value will tend to be biased slightly positive.

Convergent (or unbiased) rounding operates in the same manner as conventional rounding, except when ACCxL equals 0x8000. If this is the case, the LSB (bit 16 of the accumulator) of ACCxH is examined. If it is '1', ACCxH is incremented. If it is '0', ACCxH is not modified. Assuming that bit 16 is effectively random in nature, this scheme will remove any rounding bias that may accumulate.

The SAC and SAC.R instructions store either a truncated (SAC) or rounded (SAC.R) version of the contents of the target accumulator to data memory, via the X bus (subject to data saturation, see **Section 2.4.2.4 "Data Space Write Saturation"**). Note that for the MAC class of instructions, the accumulator write back operation will function in the same manner, addressing combined MCU (X and Y) data space though the X bus. For this class of instructions, the data is always subject to rounding.

TABLE 3-3: CORE REGISTER MAP<sup>(1)</sup> (CONTINUED)

| SFR Name | Address (Home) | 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         |                     |
|----------|----------------|----------|----------|---------------|--------|----------|--------|-------|-------|----------|-------|-------|-------|----------|-------|-------|-------|---------------------|---------------------|
| MODCON   | 0046           | XMODEN   | YMODEN   | —             | —      | BWM<3:0> |        |       |       | YWM<3:0> |       |       |       | XWM<3:0> |       |       |       | 0000 0000 0000 0000 |                     |
| XMODSRT  | 0048           | XS<15:1> |          |               |        |          |        |       |       |          |       |       |       |          |       |       |       | 0                   | uuuu uuuu uuuu uuu0 |
| XMODEND  | 004A           | XE<15:1> |          |               |        |          |        |       |       |          |       |       |       |          |       |       |       | 1                   | uuuu uuuu uuuu uuu1 |
| YMODSRT  | 004C           | YS<15:1> |          |               |        |          |        |       |       |          |       |       |       |          |       |       |       | 0                   | uuuu uuuu uuuu uuu0 |
| YMODEND  | 004E           | YE<15:1> |          |               |        |          |        |       |       |          |       |       |       |          |       |       |       | 1                   | uuuu uuuu uuuu uuu1 |
| XBREV    | 0050           | BREN     | XB<14:0> |               |        |          |        |       |       |          |       |       |       |          |       |       |       | uuuu uuuu uuuu uuuu |                     |
| DISICNT  | 0052           | —        | —        | DISICNT<13:0> |        |          |        |       |       |          |       |       |       |          |       |       |       | 0000 0000 0000 0000 |                     |

**Legend:** u = uninitialized bit; — = unimplemented bit, read as '0'  
**Note 1:** Refer to the “dsPIC30F Family Reference Manual” (DS70046) for descriptions of register bit fields.

## 4.2.1 START AND END ADDRESS

The Modulo Addressing scheme requires that a starting and an ending address be specified and loaded into the 16-bit Modulo Buffer Address registers: XMODSRT, XMODEND, YMODSRT and YMODEND (see Table 3-3).

**Note:** Y-space Modulo Addressing EA calculations assume word-sized data (LSb of every EA is always clear).

The length of a circular buffer is not directly specified. It is determined by the difference between the corresponding start and end addresses. The maximum possible length of the circular buffer is 32K words (64 Kbytes).

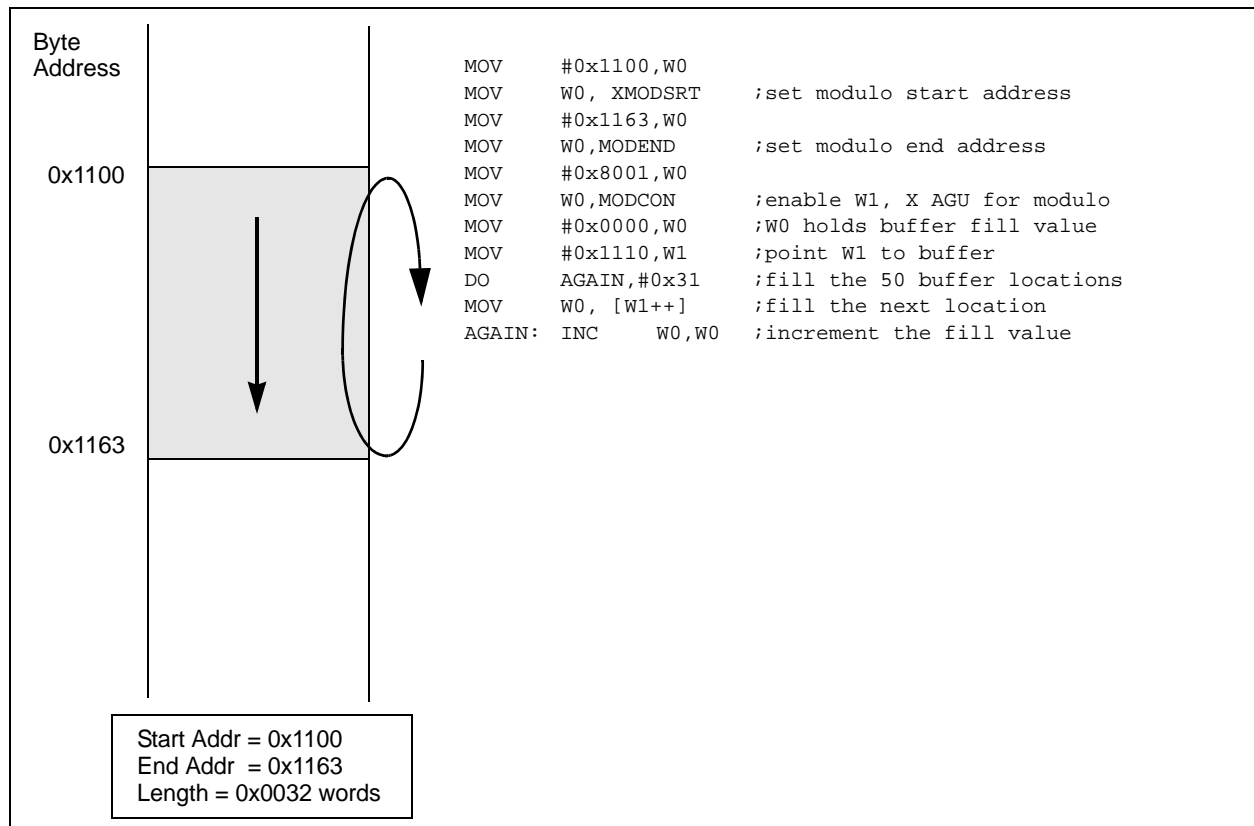
## 4.2.2 W ADDRESS REGISTER SELECTION

The Modulo and Bit-Reversed Addressing Control register, MODCON<15:0>, contains enable flags, as well as a W register field to specify the W Address registers. The XWM and YWM fields select which registers will operate with Modulo Addressing. If XWM = 15, X RAGU and X WAGU Modulo Addressing are disabled. Similarly, if YWM = 15, Y AGU Modulo Addressing is disabled.

The X Address Space Pointer W register (XWM) to which Modulo Addressing is to be applied, is stored in MODCON<3:0> (see Table 3-3). Modulo Addressing is enabled for X data space when XWM is set to any value other than 15 and the XMODEN bit is set at MODCON<15>.

The Y Address Space Pointer W register (YWM) to which Modulo Addressing is to be applied, is stored in MODCON<7:4>. Modulo Addressing is enabled for Y data space when YWM is set to any value other than 15 and the YMODEN bit is set at MODCON<14>.

**FIGURE 4-1: MODULO ADDRESSING OPERATION EXAMPLE**



## 5.0 INTERRUPTS

**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).

The dsPIC30F5015/5016 has 36 interrupt sources and 4 processor exceptions (traps), which must be arbitrated based on a priority scheme.

The CPU is responsible for reading the Interrupt Vector Table (IVT) and transferring the address contained in the interrupt vector to the program counter. The interrupt vector is transferred from the program data bus into the program counter, via a 24-bit wide multiplexer on the input of the program counter.

The Interrupt Vector Table (IVT) and Alternate Interrupt Vector Table (AIVT) are placed near the beginning of program memory (0x000004). The IVT and AIVT are shown in Figure 5-1.

The interrupt controller is responsible for pre-processing the interrupts and processor exceptions, prior to their being presented to the processor core. The peripheral interrupts and traps are enabled, prioritized and controlled using centralized Special Function Registers:

- IFS0<15:0>, IFS1<15:0>, IFS2<15:0>  
All Interrupt Request Flags are maintained in these three registers. The flags are set by their respective peripherals or external signals, and they are cleared via software.
- IEC0<15:0>, IEC1<15:0>, IEC2<15:0>  
All Interrupt Enable Control bits are maintained in these three registers. These control bits are used to individually enable interrupts from the peripherals or external signals.
- IPC0<15:0>... IPC11<7:0>  
The user assignable priority level associated with each of these 44 interrupts is held centrally in these twelve registers.
- IPL<3:0>  
The current CPU priority level is explicitly stored in the IPL bits. IPL<3> is present in the CORCON register, whereas IPL<2:0> are present in the STATUS register (SR) in the processor core.

- INTCON1<15:0>, INTCON2<15:0>  
Global interrupt control functions are derived from these two registers. INTCON1 contains the control and status flags for the processor exceptions. The INTCON2 register controls the external interrupt request signal behavior and the use of the alternate vector table.

**Note:** Interrupt flag bits get set when an interrupt condition occurs, regardless of the state of its corresponding enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

All interrupt sources can be user assigned to one of seven priority levels, 1 through 7, via the IPCx registers. Each interrupt source is associated with an interrupt vector, as shown in Table 5-1. Levels 7 and 1 represent the highest and lowest maskable priorities, respectively.

**Note:** Assigning a priority level of 0 to an interrupt source is equivalent to disabling that interrupt.

If the NSTDIS bit (INTCON1<15>) is set, nesting of interrupts is prevented. Thus, if an interrupt is currently being serviced, processing of a new interrupt is prevented, even if the new interrupt is of higher priority than the one currently being serviced.

**Note:** The IPL bits become read-only whenever the NSTDIS bit has been set to ‘1’.

Certain interrupts have specialized control bits for features like edge or level triggered interrupts, interrupt-on-change, etc. Control of these features remains within the peripheral module which generates the interrupt.

The DISI instruction can be used to disable the processing of interrupts of priorities 6 and lower for a certain number of instructions, during which the DISI bit (INTCON2<14>) remains set.

When an interrupt is serviced, the PC is loaded with the address stored in the vector location in program memory that corresponds to the interrupt. There are 63 different vectors within the IVT (refer to Figure 5-2). These vectors are contained in locations 0x000004 through 0x0000FE of program memory (refer to Figure 5-2). These locations contain 24-bit addresses, and in order to preserve robustness, an address error trap will take place should the PC attempt to fetch any of these words during normal execution. This prevents execution of random data as a result of accidentally decrementing a PC into vector space, accidentally mapping a data space address into vector space or the PC rolling over to 0x000000 after reaching the end of implemented program memory space. Execution of a GOTO instruction to this vector space will also generate an address error trap.

## 6.6 Programming Operations

A complete programming sequence is necessary for programming or erasing the internal Flash in RTSP mode. A programming operation is nominally 2 msec in duration and 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.

### 6.6.1 PROGRAMMING ALGORITHM FOR PROGRAM FLASH

The user can erase or program one row of program Flash memory at a time. The general process is:

1. Read one row of program Flash (32 instruction words) and store into data RAM as a data "image".
2. Update the data image with the desired new data.
3. Erase program Flash row.
  - a) Set up NVMCON register for multi-word, program Flash, erase, and set WREN bit.
  - b) Write address of row to be erased into NVMADRU/NVMADR.
  - c) Write 0x55 to NVMKEY.
  - d) Write 0xAA to NVMKEY.
  - e) Set the WR bit. This will begin erase cycle.
  - f) CPU will stall for the duration of the erase cycle.
  - g) The WR bit is cleared when erase cycle ends.
4. Write 32 instruction words of data from data RAM "image" into the program Flash write latches.
5. Program 32 instruction words into program Flash.
  - a) Set up NVMCON register for multi-word, program Flash, program, and set WREN bit.
  - b) Write 0x55 to NVMKEY.
  - c) Write 0xAA to NVMKEY.
  - d) Set the WR bit. This will begin program cycle.
  - e) CPU will stall for duration of the program cycle.
  - f) The WR bit is cleared by the hardware when program cycle ends.
6. Repeat steps 1 through 5 as needed to program desired amount of program Flash memory.

### 6.6.2 ERASING A ROW OF PROGRAM MEMORY

Example 6-1 shows a code sequence that can be used to erase a row (32 instructions) of program memory.

#### EXAMPLE 6-1: ERASING A ROW OF PROGRAM MEMORY

```
; Setup NVMCON for erase operation, multi word write
; program memory selected, and writes enabled
MOV    #0x4041,W0                ;
MOV    W0,NVMCON                 ; Init NVMCON SFR
; Init pointer to row to be ERASED
MOV    #tblpage(PROG_ADDR),W0    ;
MOV    W0,NVMADRU                ; Initialize PM Page Boundary SFR
MOV    #tbloffset(PROG_ADDR),W0  ; Initialize in-page EA[15:0] pointer
MOV    W0, NVMADR                ; Initialize NVMADR SFR
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                ; Start the erase sequence
NOP    ; Insert two NOPs after the erase
NOP    ; command is asserted
```





# dsPIC30F5015/5016

---

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 (QEICON<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.

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. To place the state of this signal on an I/O pin, the SFR bit, PCDOOUT (QEICON<6>), must be '1'.

## 14.3 Position Measurement Mode

There are two measurement modes which are supported and are termed x2 and x4. These modes are selected by the QEIM<2:0> mode select bits located in SFR QEICON<10:8>.

When control bits QEIM<2:0> = 100 or 101, the x2 Measurement mode is selected and the QEI logic only looks at the Phase A input for the position counter increment rate. Every rising and falling edge of the Phase A signal causes the position counter to be incremented or decremented. The Phase B signal is still utilized for the determination of the counter direction, just as in the x4 mode.

Within the x2 Measurement mode, there are two variations of how the position counter is reset:

1. Position counter reset by detection of index pulse, QEIM<2:0> = 100.
2. Position counter reset by match with MAXCNT, QEIM<2:0> = 101.

When control bits QEIM<2:0> = 110 or 111, the x4 Measurement mode is selected and the QEI logic looks at both edges of the Phase A and Phase B input signals. Every edge of both signals causes the position counter to increment or decrement.

Within the x4 Measurement mode, there are two variations of how the position counter is reset:

1. Position counter reset by detection of index pulse, QEIM<2:0> = 110.
2. Position counter reset by match with MAXCNT, QEIM<2:0> = 111.

The x4 Measurement mode provides for finer resolution data (more position counts) for determining motor position.

## 15.0 MOTOR CONTROL PWM MODULE

**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).

This module simplifies the task of generating multiple, synchronized Pulse-Width Modulated (PWM) outputs. In particular, the following power and motion control applications are supported by the PWM module:

- Three Phase AC Induction Motor
- Switched Reluctance (SR) Motor
- Brushless DC (BLDC) Motor
- Uninterruptible Power Supply (UPS)

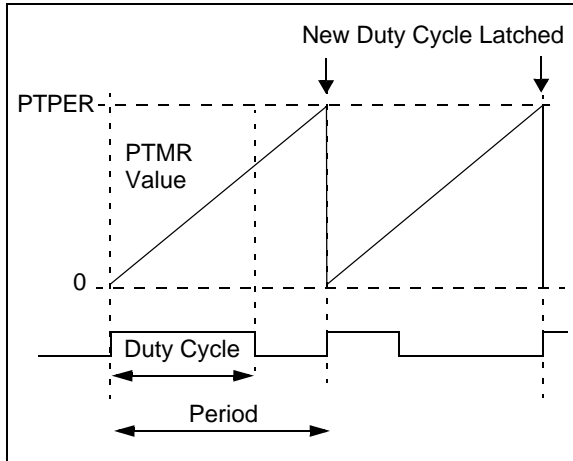
The PWM module has the following features:

- Eight PWM I/O pins with four duty cycle generators
- Up to 16-bit resolution
- 'On-the-Fly' PWM frequency changes
- Edge and Center-Aligned Output modes
- Single-Pulse Generation mode
- Interrupt support for asymmetrical updates in Center-Aligned mode
- Output override control for Electrically Commutative Motor (ECM) operation
- 'Special Event' comparator for scheduling other peripheral events
- Fault pins to optionally drive each of the PWM output pins to a defined state
- Duty cycle updates are configurable to be immediate or synchronized to the PWM time base

This module contains four duty cycle generators, numbered 1 through 4. The module has eight PWM output pins, numbered PWM1H/PWM1L through PWM4H/PWM4L. The eight I/O pins are grouped into high/low numbered pairs, denoted by the suffix H or L, respectively. For complementary loads, the low PWM pins are always the complement of the corresponding high I/O pin.

The PWM module allows several modes of operation which are beneficial for specific power control applications.

**FIGURE 15-2: EDGE-ALIGNED PWM**



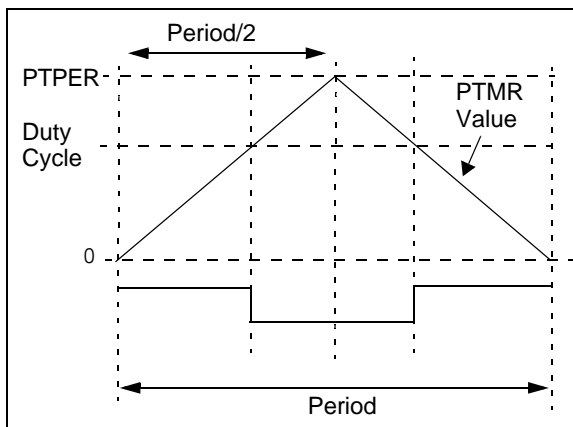
## 15.4 Center-Aligned PWM

Center-aligned PWM signals are produced by the module when the PWM time base is configured in an Up/Down Counting mode (see Figure 15-3).

The PWM compare output is driven to the active state when the value of the Duty Cycle register matches the value of PTMR and the PWM time base is counting downwards (PTDIR = 1). The PWM compare output is driven to the inactive state when the PWM time base is counting upwards (PTDIR = 0) and the value in the PTMR register matches the duty cycle value.

If the value in a particular Duty Cycle register is zero, then the output on the corresponding PWM pin will be inactive for the entire PWM period. In addition, the output on the PWM pin will be active for the entire PWM period if the value in the Duty Cycle register is equal to the value held in the PTPER register.

**FIGURE 15-3: CENTER-ALIGNED PWM**



## 15.5 PWM Duty Cycle Comparison Units

There are four 16-bit Special Function Registers (PDC1, PDC2, PDC3 and PDC4) used to specify duty cycle values for the PWM module.

The value in each Duty Cycle register determines the amount of time that the PWM output is in the active state. The Duty Cycle registers are 16 bits wide. The LSb of a Duty Cycle register determines whether the PWM edge occurs in the beginning. Thus, the PWM resolution is effectively doubled.

TABLE 18-1: UART1 REGISTER MAP<sup>(1)</sup>

| 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         |
|----------|-------|-------------------------------|--------|--------|--------|--------|--------|-------|-------|-------------------|----------|-------|-------|-------|--------|--------|-------|---------------------|
| U1MODE   | 020C  | UARTEN                        | —      | USIDL  | —      | —      | —      | —     | —     | WAKE              | LPBACK   | ABAUD | —     | —     | PDSEL1 | PDSEL0 | STSEL | 0000 0000 0000 0000 |
| U1STA    | 020E  | UTXISEL                       | —      | —      | —      | UTXBRK | UTXEN  | UTXBF | TRMT  | URXISEL1          | URXISEL0 | ADDEN | RIDLE | PERR  | FERR   | OERR   | URXDA | 0000 0001 0001 0000 |
| U1TXREG  | 0210  | —                             | —      | —      | —      | —      | —      | —     | UTX8  | Transmit Register |          |       |       |       |        |        |       | 0000 000u uuuu uuuu |
| U1RXREG  | 0212  | —                             | —      | —      | —      | —      | —      | —     | URX8  | Receive Register  |          |       |       |       |        |        |       | 0000 0000 0000 0000 |
| U1BRG    | 0214  | Baud Rate Generator Prescaler |        |        |        |        |        |       |       |                   |          |       |       |       |        |        |       | 0000 0000 0000 0000 |

**Legend:** u = uninitialized bit; — = unimplemented bit, read as '0'  
**Note 1:** Refer to the “dsPIC30F Family Reference Manual” (DS70046) for descriptions of register bit fields.

---



## 19.4 Message Reception

### 19.4.1 RECEIVE BUFFERS

The CAN bus module has 3 receive buffers. However, one of the receive buffers is always committed to monitoring the bus for incoming messages. This buffer is called the Message Assembly Buffer (MAB). So there are 2 receive buffers visible, RXB0 and RXB1, that can essentially instantaneously receive a complete message from the protocol engine.

All messages are assembled by the MAB, and are transferred to the RXBn buffers only if the acceptance filter criterion are met. When a message is received, the RXnIF flag (CiINTF<0> or CiINRF<1>) will be set. This bit can only be set by the module when a message is received. The bit is cleared by the CPU when it has completed processing the message in the buffer. If the RXnIE bit (CiINTE<0> or CiINTE<1>) is set, an interrupt will be generated when a message is received.

RXF0 and RXF1 filters with RXM0 mask are associated with RXB0. The filters RXF2, RXF3, RXF4, and RXF5 and the mask RXM1 are associated with RXB1.

### 19.4.2 MESSAGE ACCEPTANCE FILTERS

The message acceptance filters and masks are used to determine if a message in the message assembly buffer should be loaded into either of the receive buffers. Once a valid message has been received into the message assembly buffer, the identifier fields of the message are compared to the filter values. If there is a match, that message will be loaded into the appropriate receive buffer.

The acceptance filter looks at incoming messages for the RXIDE bit (CiRXnSID<0>) to determine how to compare the identifiers. If the RXIDE bit is clear, the message is a standard frame, and only filters with the EXIDE bit (CiRXFnSID<0>) clear are compared. If the RXIDE bit is set, the message is an extended frame, and only filters with the EXIDE bit set are compared. Configuring the RXM<1:0> bits to '01' or '10' can override the EXIDE bit.

### 19.4.3 MESSAGE ACCEPTANCE FILTER MASKS

The mask bits essentially determine which bits to apply the filter to. If any mask bit is set to a zero, then that bit will automatically be accepted regardless of the filter bit. There are 2 programmable acceptance filter masks associated with the receive buffers, one for each buffer.

### 19.4.4 RECEIVE OVERRUN

An overrun condition occurs when the message assembly buffer has assembled a valid received message, the message is accepted through the acceptance filters, and when the receive buffer associated with the filter has not been designated as clear of the previous message.

The overrun error flag, RXnOVR (CiINTF<15> or CiINTF<14>) and the ERRIF bit (CiINTF<5>) will be set and the message in the MAB will be discarded.

If the DBEN bit is clear, RXB1 and RXB0 operate independently. When this is the case, a message intended for RXB0 will not be diverted into RXB1 if RXB0 contains an unread message and the RX0OVR bit will be set.

If the DBEN bit is set, the overrun for RXB0 is handled differently. If a valid message is received for RXB0 and RXFUL = 1 indicates that RXB0 is full and RXFUL = 0 indicates that RXB1 is empty, the message for RXB0 will be loaded into RXB1. An overrun error will not be generated for RXB0. If a valid message is received for RXB0 and RXFUL = 1 and RXFUL = 1 indicates that both RXB0 and RXB1 are full, the message will be lost and an overrun will be indicated for RXB1.

### 19.4.5 RECEIVE ERRORS

The CAN module will detect the following receive errors:

- Cyclic Redundancy Check (CRC) Error
- Bit Stuffing Error
- Invalid Message Receive Error

The receive error counter is incremented by one in case one of these errors occur. The RXWAR bit (CiINTF<9>) indicates that the Receive Error Counter has reached the CPU warning limit of 96 and an interrupt is generated.

### 19.4.6 RECEIVE INTERRUPTS

Receive interrupts can be divided into three major groups, each including various conditions that generate interrupts:

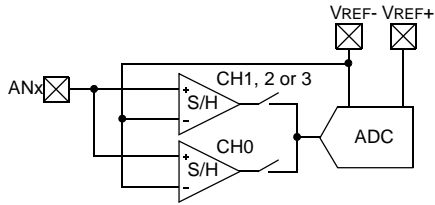
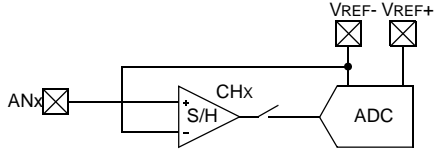
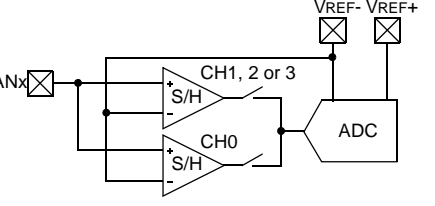
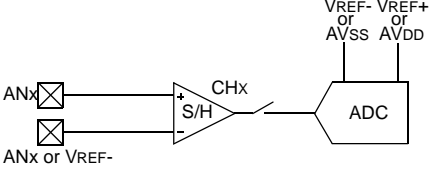
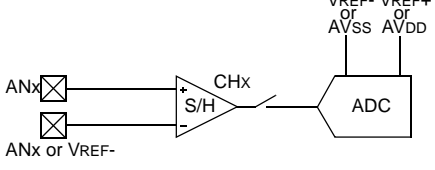
- Receive Interrupt

A message has been successfully received and loaded into one of the receive buffers. This interrupt is activated immediately after receiving the End-of-Frame (EOF) field. Reading the RXnIF flag will indicate which receive buffer caused the interrupt.

- Wake-up interrupt

The CAN module has woken up from Disable mode or the device has woken up from Sleep mode.

**TABLE 20-1: 10-BIT CONVERSION RATE PARAMETERS**

| dsPIC30F 10-bit A/D Converter Conversion Rates |             |                   |        |              |                 |   |
|--|-------------|-------------------|--------|--------------|-----------------|---|
| A/D Speed                                      | TAD Minimum | Sampling Time Min | Rs Max | VDD          | Temperature     | A/D Channels Configuration  |
| Up to 1 Msps <sup>(1)</sup>                    | 83.33 ns    | 12 TAD            | 500Ω   | 4.5V to 5.5V | -40°C to +85°C  |    |
| Up to 750 ksps <sup>(1)</sup>                  | 95.24 ns    | 2 TAD             | 500Ω   | 4.5V to 5.5V | -40°C to +85°C  |    |
| Up to 600 ksps <sup>(1)</sup>                  | 138.89 ns   | 12 TAD            | 500Ω   | 3.0V to 5.5V | -40°C to +125°C |   |
| Up to 500 ksps                                 | 153.85 ns   | 1 TAD             | 5.0 kΩ | 4.5V to 5.5V | -40°C to +125°C |  |
| Up to 300 ksps                                 | 256.41 ns   | 1 TAD             | 5.0 kΩ | 3.0V to 5.5V | -40°C to +125°C |  |

**Note 1:** External VREF- and VREF+ pins must be used for correct operation. See Figure 20-2 for recommended circuit.



**TABLE 20-2: ADC REGISTER MAP<sup>(1)</sup>**

| 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         |
|----------|-------|--------------|--------|---------|-----------|------------|--------|--------------------|-------|--------------|-------|-----------|-------|------------|-------|-------|-------|---------------------|
| ADCBUF0  | 0280  | —            | —      | —       | —         | —          | —      | ADC Data Buffer 0  |       |              |       |           |       |            |       |       |       | 0000 00uu uuuu uuuu |
| ADCBUF1  | 0282  | —            | —      | —       | —         | —          | —      | ADC Data Buffer 1  |       |              |       |           |       |            |       |       |       | 0000 00uu uuuu uuuu |
| ADCBUF2  | 0284  | —            | —      | —       | —         | —          | —      | ADC Data Buffer 2  |       |              |       |           |       |            |       |       |       | 0000 00uu uuuu uuuu |
| ADCBUF3  | 0286  | —            | —      | —       | —         | —          | —      | ADC Data Buffer 3  |       |              |       |           |       |            |       |       |       | 0000 00uu uuuu uuuu |
| ADCBUF4  | 0288  | —            | —      | —       | —         | —          | —      | ADC Data Buffer 4  |       |              |       |           |       |            |       |       |       | 0000 00uu uuuu uuuu |
| ADCBUF5  | 028A  | —            | —      | —       | —         | —          | —      | ADC Data Buffer 5  |       |              |       |           |       |            |       |       |       | 0000 00uu uuuu uuuu |
| ADCBUF6  | 028C  | —            | —      | —       | —         | —          | —      | ADC Data Buffer 6  |       |              |       |           |       |            |       |       |       | 0000 00uu uuuu uuuu |
| ADCBUF7  | 028E  | —            | —      | —       | —         | —          | —      | ADC Data Buffer 7  |       |              |       |           |       |            |       |       |       | 0000 00uu uuuu uuuu |
| ADCBUF8  | 0290  | —            | —      | —       | —         | —          | —      | ADC Data Buffer 8  |       |              |       |           |       |            |       |       |       | 0000 00uu uuuu uuuu |
| ADCBUF9  | 0292  | —            | —      | —       | —         | —          | —      | ADC Data Buffer 9  |       |              |       |           |       |            |       |       |       | 0000 00uu uuuu uuuu |
| ADCBUFA  | 0294  | —            | —      | —       | —         | —          | —      | ADC Data Buffer 10 |       |              |       |           |       |            |       |       |       | 0000 00uu uuuu uuuu |
| ADCBUFB  | 0296  | —            | —      | —       | —         | —          | —      | ADC Data Buffer 11 |       |              |       |           |       |            |       |       |       | 0000 00uu uuuu uuuu |
| ADCBUFC  | 0298  | —            | —      | —       | —         | —          | —      | ADC Data Buffer 12 |       |              |       |           |       |            |       |       |       | 0000 00uu uuuu uuuu |
| ADCBUFD  | 029A  | —            | —      | —       | —         | —          | —      | ADC Data Buffer 13 |       |              |       |           |       |            |       |       |       | 0000 00uu uuuu uuuu |
| ADCBUFE  | 029C  | —            | —      | —       | —         | —          | —      | ADC Data Buffer 14 |       |              |       |           |       |            |       |       |       | 0000 00uu uuuu uuuu |
| ADCBUFF  | 029E  | —            | —      | —       | —         | —          | —      | ADC Data Buffer 15 |       |              |       |           |       |            |       |       |       | 0000 00uu uuuu uuuu |
| ADCON1   | 02A0  | ADON         | —      | ADSIDL  | —         | —          | —      | FORM<1:0>          |       | SSRC<2:0>    |       |           | —     | SIMSAM     | ASAM  | SAMP  | DONE  | 0000 0000 0000 0000 |
| ADCON2   | 02A2  | VCFG<2:0>    |        |         | —         | —          | CSCNA  | CHPS<1:0>          |       | BUFS         | —     | SMPI<3:0> |       |            |       | BUFM  | ALTS  | 0000 0000 0000 0000 |
| ADCON3   | 02A4  | —            | —      | —       | SAMC<4:0> |            |        |                    |       | ADRC         | —     | ADCS<5:0> |       |            |       |       |       | 0000 0000 0000 0000 |
| ADCHS    | 02A6  | CH123NB<1:0> |        | CH123SB | CH0NB     | CH0SB<3:0> |        |                    |       | CH123NA<1:0> |       | CH123SA   | CH0NA | CH0SA<3:0> |       |       |       | 0000 0000 0000 0000 |
| ADPCFG   | 02A8  | PCFG15       | PCFG14 | PCFG13  | PCFG12    | PCFG11     | PCFG10 | PCFG9              | PCFG8 | PCFG7        | PCFG6 | PCFG5     | PCFG4 | PCFG3      | PCFG2 | PCFG1 | PCFG0 | 0000 0000 0000 0000 |
| ADCSSL   | 02AA  | CSSL15       | CSSL14 | CSSL13  | CSSL12    | CSSL11     | CSSL10 | CSSL9              | CSSL8 | CSSL7        | CSSL6 | CSSL5     | CSSL4 | CSSL3      | CSSL2 | CSSL1 | CSSL0 | 0000 0000 0000 0000 |

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

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

NOTES:

## 24.0 ELECTRICAL CHARACTERISTICS

This section provides an overview of dsPIC30F electrical characteristics. Additional information will be provided in future revisions of this document as it becomes available.

For detailed information about the dsPIC30F architecture and core, refer to the “dsPIC30F Family Reference Manual” (DS70046).

Absolute maximum ratings for the dsPIC30F family are listed below. Exposure to these maximum rating conditions for extended periods may affect device reliability. Functional operation of the device at these or any other conditions above the parameters indicated in the operation listings of this specification is not implied.

### Absolute Maximum Ratings<sup>(†)</sup>

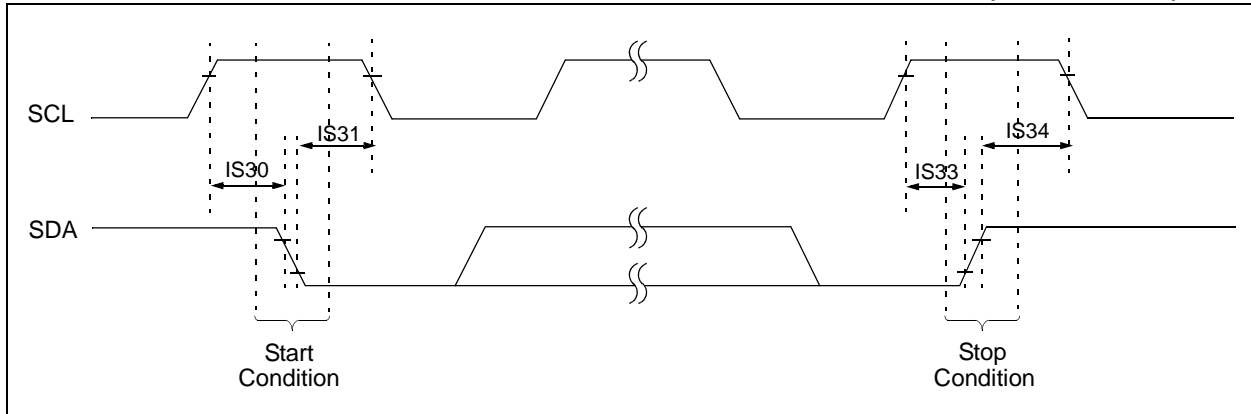
|  |                       |
|--|-----------------------|
| Ambient temperature under bias .....   | -40°C to +125°C       |
| Storage temperature .....  | -65°C to +150°C       |
| Voltage on any pin with respect to VSS (except VDD and MCLR) <sup>(1)</sup> .....        | -0.3V to (VDD + 0.3V) |
| Voltage on VDD with respect to VSS .....   | -0.3V to +5.5V        |
| Voltage on MCLR with respect to VSS.....   | 0V to +13.25V         |
| Maximum current out of VSS pin .....   | 300 mA                |
| Maximum current into VDD pin <sup>(2)</sup> .....  | 250 mA                |
| Input clamp current, I <sub>IK</sub> (V <sub>I</sub> < 0 or V <sub>I</sub> > VDD) .....  | ±20 mA                |
| Output clamp current, I <sub>OK</sub> (V <sub>O</sub> < 0 or V <sub>O</sub> > VDD) ..... | ±20 mA                |
| Maximum output current sunk by any I/O pin.....  | 25 mA                 |
| Maximum output current sourced by any I/O pin .....                                      | 25 mA                 |
| Maximum current sunk by all ports .....  | 200 mA                |
| Maximum current sourced by all ports <sup>(2)</sup> .....                                | 200 mA                |

**Note 1:** Voltage spikes below VSS at the MCLR/VPP pin, inducing currents greater than 80 mA, may cause latch-up. Thus, a series resistor of 50-100Ω should be used when applying a “low” level to the MCLR/VPP pin, rather than pulling this pin directly to VSS.

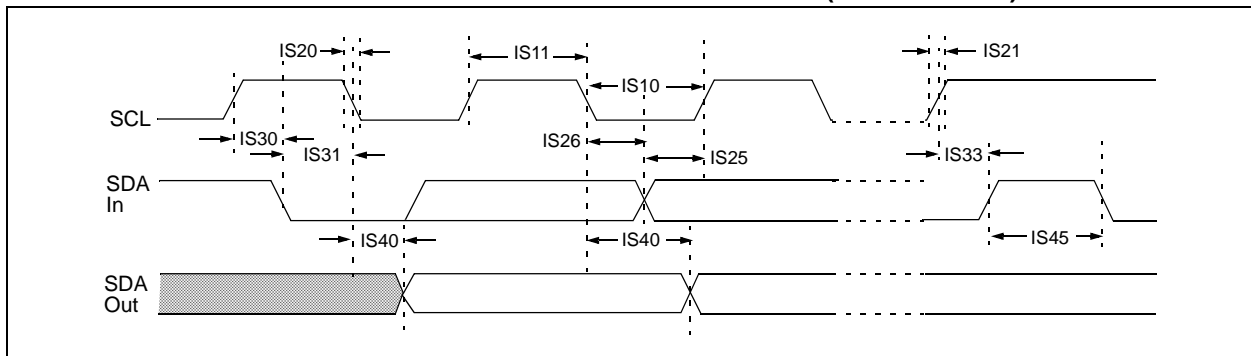
**2:** Maximum allowable current is a function of device maximum power dissipation. See Table 24-6.

**†NOTICE:** Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

**FIGURE 24-22: I<sup>2</sup>C™ BUS START/STOP BITS TIMING CHARACTERISTICS (SLAVE MODE)**

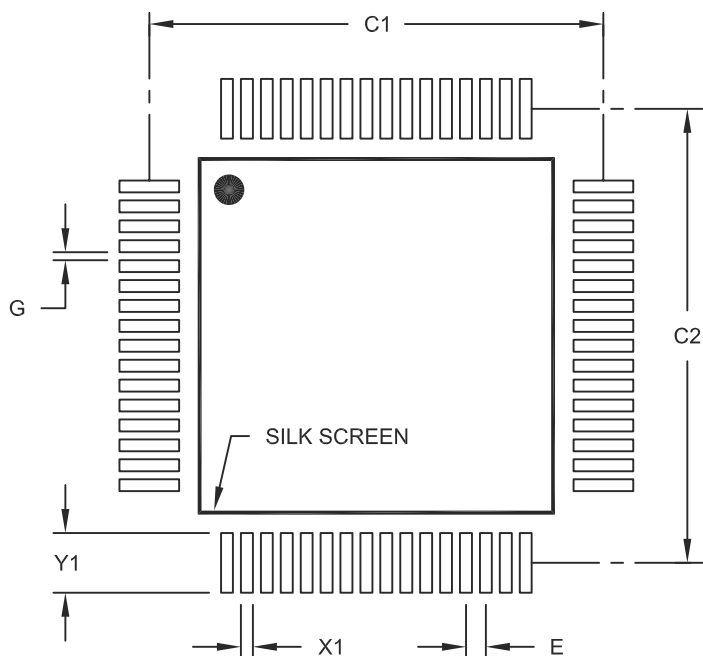


**FIGURE 24-23: I<sup>2</sup>C™ BUS DATA TIMING CHARACTERISTICS (SLAVE MODE)**



## 64-Lead Plastic Thin Quad Flatpack (PT) – 10x10x1 mm Body, 2.00 mm [TQFP]

**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  | 0.50 BSC    |       |      |
| Contact Pad Spacing      | C1 |             | 11.40 |      |
| Contact Pad Spacing      | C2 |             | 11.40 |      |
| Contact Pad Width (X64)  | X1 |             |       | 0.30 |
| Contact Pad Length (X64) | Y1 |             |       | 1.50 |
| 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. C04-2085A