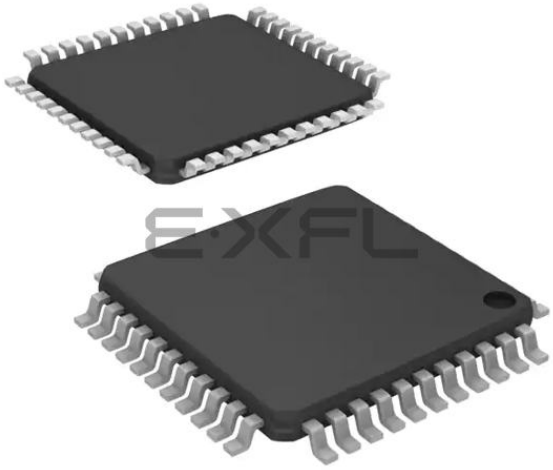


Welcome to E-XFL.COM

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, POR, PWM, WDT
Number of I/O	30
Program Memory Size	24KB (8K 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 13x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-TQFP
Supplier Device Package	44-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/dspic30f3014t-30i-pt

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

This section contains a brief overview of the CPU architecture of the dsPIC30F.

The core has a 24-bit instruction word. The Program Counter (PC) is 23 bits wide with the Least Significant bit (LSb) always clear (refer to **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 16-bit x 16-bit registers, each of which can act as data, address or offset registers. One working register (W15) operates as a Software Stack Pointer 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 x 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 15 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 is 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.

2.4.1 MULTIPLIER

The 17-bit x 17-bit multiplier is capable of signed or unsigned operation and can multiplex its output using a scaler to support either 1.31 fractional (Q31) or 32-bit integer results. Unsigned operands are zero-extended into the 17th bit of the multiplier input value. Signed operands are sign-extended into the 17th bit of the multiplier input value. The output of the 17-bit x 17-bit multiplier/scaler is a 33-bit value, which is sign-extended to 40 bits. Integer data is inherently represented as a signed two's complement value, where the MSB is defined as a sign bit. Generally speaking, the range of an N-bit two's complement integer is -2^{N-1} to $2^{N-1} - 1$. For a 16-bit integer, the data range is -32768 (0x8000) to 32767 (0x7FFF) including '0'. For a 32-bit integer, the data range is -2,147,483,648 (0x8000 0000) to 2,147,483,645 (0x7FFF FFFF).

When the multiplier is configured for fractional multiplication, the data is represented as a two's complement fraction, where the MSB is defined as a sign bit and the radix point is implied to lie just after the sign bit (QX format). The range of an N-bit two's complement fraction with this implied radix point is -1.0 to $(1 - 2^{1-N})$. For a 16-bit fraction, the Q15 data range is -1.0 (0x8000) to 0.999969482 (0x7FFF) including '0' and has a precision of 3.01518×10^{-5}. In Fractional mode, the 16x16 multiply operation generates a 1.31 product, which has a precision of 4.65661×10^{-10}.

The same multiplier is used to support the MCU multiply instructions, which includes integer 16-bit signed, unsigned and mixed sign multiplies.

The MUL instruction can be directed to use byte or word-sized operands. Byte operands direct a 16-bit result, and word operands direct a 32-bit result to the specified register(s) in the W array.

2.4.2 DATA ACCUMULATORS AND ADDER/SUBTRACTER

The data accumulator consists of a 40-bit adder/subtracter with automatic sign extension logic. It can select one of two accumulators (A or B) as its pre-accumulation source and post-accumulation destination. For the ADD and LAC instructions, the data to be accumulated or loaded can be optionally scaled via the barrel shifter prior to accumulation.

2.4.2.1 Adder/Subtracter, Overflow and Saturation

The adder/subtracter is a 40-bit adder with an optional zero input into one side and either true or complement data into the other input. In the case of addition, the carry/borrow input is active-high and the other input is true data (not complemented), whereas in the case of subtraction, the carry/borrow input is active-low and the other input is complemented. The adder/subtracter generates overflow Status bits, SA/SB and OA/OB, which are latched and reflected in the STATUS register:

- Overflow from bit 39: this is a catastrophic overflow in which the sign of the accumulator is destroyed.
- Overflow into guard bits 32 through 39: this is a recoverable overflow. This bit is set whenever all the guard bits are not identical to each other.

The adder has an additional saturation block which controls accumulator data saturation if selected. It uses the result of the adder, the overflow Status bits described above, and the SATA/B (CORCON<7:6>) and ACCSAT (CORCON<4>) mode control bits to determine when and to what value to saturate.

Six STATUS register bits have been provided to support saturation and overflow. They are:

1. OA:
   AccA overflowed into guard bits
2. OB:
   AccB overflowed into guard bits
3. SA:
   AccA saturated (bit 31 overflow and saturation)
   or
   AccA overflowed into guard bits and saturated (bit 39 overflow and saturation)
4. SB:
   AccB saturated (bit 31 overflow and saturation)
   or
   AccB overflowed into guard bits and saturated (bit 39 overflow and saturation)
5. OAB:
   Logical OR of OA and OB
6. SAB:
   Logical OR of SA and SB

The OA and OB bits are modified each time data passes through the adder/subtracter. When set, they indicate that the most recent operation has overflowed into the accumulator guard bits (bits 32 through 39). The OA and OB bits can also optionally generate an arithmetic warning trap when set and the corresponding overflow trap flag enable bit (OVATE, OVBTE) in the INTCON1 register (refer to **Section 8.0 "Interrupts"**) is set. This allows the user to take immediate action, for example, to correct system gain.

NOTES:

5.4 RTSP Operation

The dsPIC30F Flash program memory is organized into rows and panels. Each row consists of 32 instructions or 96 bytes. Each panel consists of 128 rows or 4K x 24 instructions. RTSP allows the user to erase one row (32 instructions) at a time and to program four instructions at one time. RTSP may be used to program multiple program memory panels, but the Table Pointer must be changed at each panel boundary.

Each panel of program memory contains write latches that hold 32 instructions of programming data. Prior to the actual programming operation, the write data must be loaded into the panel write latches. The data to be programmed into the panel is loaded in sequential order into the write latches; instruction 0, instruction 1, etc. The instruction words loaded must always be from a 32 address boundary.

The basic sequence for RTSP programming is to set up a Table Pointer, then do a series of TBLWT instructions to load the write latches. Programming is performed by setting the special bits in the NVMCON register. 32 TBLWTL and four TBLWTH instructions are required to load the 32 instructions. If multiple panel programming is required, the Table Pointer needs to be changed and the next set of multiple write latches written.

All of the table write operations are single-word writes (2 instruction cycles), because only the table latches are written. A programming cycle is required for programming each row.

The Flash program memory is readable, writable and erasable during normal operation over the entire VDD range.

5.5 Control Registers

The four SFRs used to read and write the program Flash memory are:

- NVMCON
- NVMADR
- NVMADRU
- NVMKEY

5.5.1 NVMCON REGISTER

The NVMCON register controls which blocks are to be erased, which memory type is to be programmed and the start of the programming cycle.

5.5.2 NVMADR REGISTER

The NVMADR register is used to hold the lower two bytes of the Effective Address. The NVMADR register captures the EA<15:0> of the last table instruction that has been executed and selects the row to write.

5.5.3 NVMADRU REGISTER

The NVMADRU register is used to hold the upper byte of the Effective Address. The NVMADRU register captures the EA<23:16> of the last table instruction that has been executed.

5.5.4 NVMKEY REGISTER

NVMKEY is a write-only register that is used for write protection. To start a programming or erase sequence, the user must consecutively write 0x55 and 0xAA to the NVMKEY register. Refer to **Section 5.6 “Programming Operations”** for further details.

Note:	The user can also directly write to the NVMADR and NVMADRU registers to specify a program memory address for erasing or programming.

5.6 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.

5.6.1 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/NVMDR.
   c) Write 0x55 to NVMKEY.
   d) Write 0xAA to NVMKEY.
   e) Set the WR bit. This begins erase cycle.
   f) CPU stalls 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 begins program cycle.
   e) CPU stalls 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.

5.6.2 ERASING A ROW OF PROGRAM MEMORY

Example 5-1 shows a code sequence that can be used to erase a row (32 instructions) of program memory.

EXAMPLE 5-1: 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    ; Intialize 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
```

10.0 TIMER2/3 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).

This section describes the 32-bit general purpose timer module (Timer2/3) and associated operational modes. Figure 10-1 depicts the simplified block diagram of the 32-bit Timer2/3 module. Figure 10-2 and Figure 10-3 show Timer2/3 configured as two independent 16-bit timers, Timer2 and Timer3, respectively.

The Timer2/3 module is a 32-bit timer (which can be configured as two 16-bit timers) with selectable operating modes. These timers are utilized by other peripheral modules, such as:

- Input Capture
- Output Compare/Simple PWM

The following sections provide a detailed description, including setup and control registers, along with associated block diagrams for the operational modes of the timers.

The 32-bit timer has the following modes:

- Two independent 16-bit timers (Timer2 and Timer3) with all 16-bit operating modes (except Asynchronous Counter mode)
- Single 32-bit timer operation
- Single 32-bit synchronous counter

Further, the following operational characteristics are supported:

- ADC event trigger
- Timer gate operation
- Selectable prescaler settings
- Timer operation during Idle and Sleep modes
- Interrupt on a 32-bit Period register match

These operating modes are determined by setting the appropriate bit(s) in the 16-bit T2CON and T3CON SFRs.

For 32-bit timer/counter operation, Timer2 is the lsw and Timer3 is the msw of the 32-bit timer.

Note:	For 32-bit timer operation, T3CON control bits are ignored. Only T2CON control bits are used for setup and control. Timer2 clock and gate inputs are utilized for the 32-bit timer module, but an interrupt is generated with the Timer3 Interrupt Flag (T3IF) and the interrupt is enabled with the Timer3 interrupt enable bit (T3IE).

16-Bit Timer Mode: In the 16-bit mode, Timer2 and Timer3 can be configured as two independent 16-bit timers. Each timer can be set up in either 16-bit Timer mode or 16-bit Synchronous Counter mode. See **Section 9.0 "Timer1 Module"** for details on these two operating modes.

The only functional difference between Timer2 and Timer3 is that Timer2 provides synchronization of the clock prescaler output. This is useful for high-frequency external clock inputs.

32-Bit Timer Mode: In the 32-Bit Timer mode, the timer increments on every instruction cycle, up to a match value preloaded into the combined 32-bit Period register, PR3/PR2, then resets to '0' and continues to count.

For synchronous 32-bit reads of the Timer2/Timer3 pair, reading the lsw (TMR2 register) causes the msw to be read and latched into a 16-bit holding register, termed TMR3HLD.

For synchronous 32-bit writes, the holding register (TMR3HLD) must first be written to. When followed by a write to the TMR2 register, the contents of TMR3HLD is transferred and latched into the MSB of the 32-bit timer (TMR3).

32-Bit Synchronous Counter Mode: In the 32-Bit Synchronous Counter mode, the timer increments on the rising edge of the applied external clock signal which is synchronized with the internal phase clocks. The timer counts up to a match value preloaded in the combined 32-bit Period register, PR3/PR2, then resets to '0' and continues.

When the timer is configured for the Synchronous Counter mode of operation and the CPU goes into the Idle mode, the timer stops incrementing unless the TSIDL (T2CON<13>) bit = 0. If TSIDL = 1, the timer module logic resumes the incrementing sequence upon termination of the CPU Idle mode.

TABLE 10-1: dsPIC30F3014/4013 TIMER2/3 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
TMR2	0106	Timer2 Register																uuuu uuuu uuuu uuuu
TMR3HLD	0108	Timer3 Holding Register (for 32-bit timer operations only)																uuuu uuuu uuuu uuuu
TMR3	010A	Timer3 Register																uuuu uuuu uuuu uuuu
PR2	010C	Period Register 2																1111 1111 1111 1111
PR3	010E	Period Register 3																1111 1111 1111 1111
T2CON	0110	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	T32	—	TCS	—	0000 0000 0000 0000
T3CON	0112	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	—	TCS	—	0000 0000 0000 0000

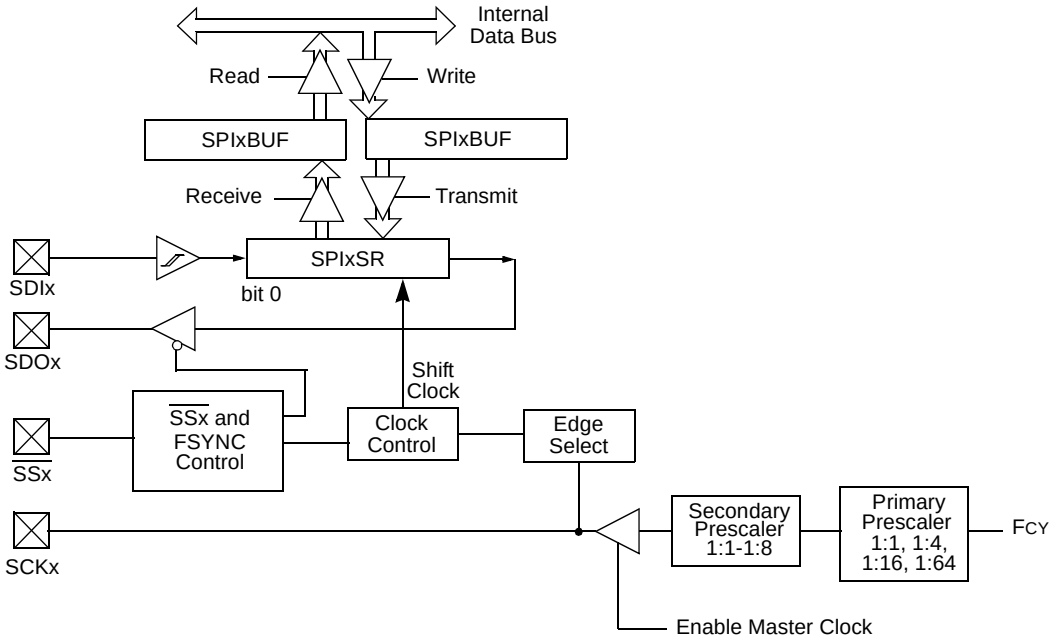
Legend: u = uninitialized bit; — = unimplemented bit, read as '0'

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

15.2 Framed SPI Support

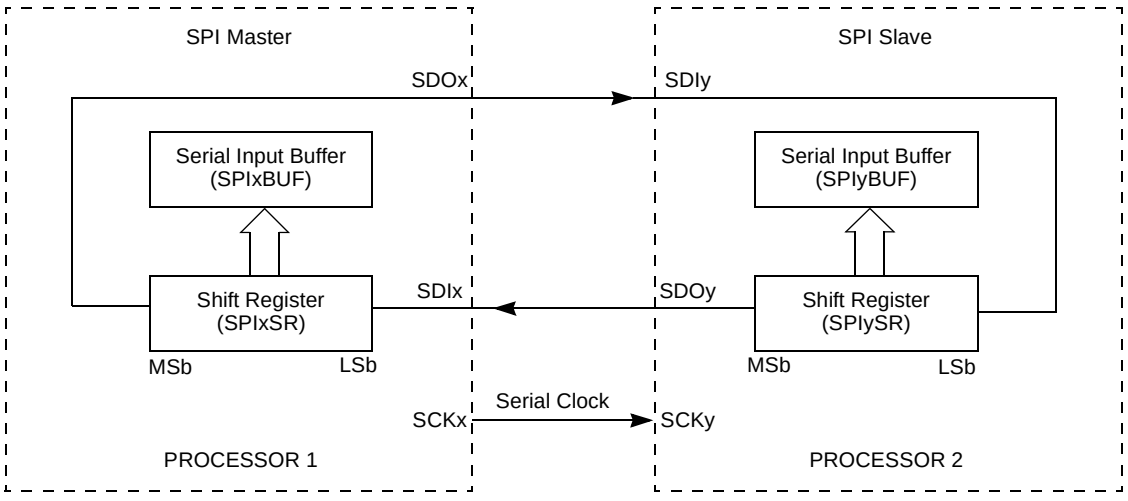
The module supports a basic framed SPI protocol in Master or Slave mode. The control bit, FRMEN, enables framed SPI support and causes the $\overline{SSx}$ pin to perform the Frame Synchronization pulse (FSYNC) function. The control bit, SPIFSD, determines whether the $\overline{SSx}$ pin is an input or an output (i.e., whether the module receives or generates the Frame Synchronization pulse). The frame pulse is an active-high pulse for a single SPI clock cycle. When Frame Synchronization is enabled, the data transmission starts only on the subsequent transmit edge of the SPI clock.

FIGURE 15-1: SPI BLOCK DIAGRAM



Note: x = 1 or 2.

FIGURE 15-2: SPI MASTER/SLAVE CONNECTION



Note: x = 1 or 2, y = 1 or 2.

- Receive Error Interrupts:

  A receive error interrupt is indicated by the ERRIF bit. This bit shows that an error condition occurred. The source of the error can be determined by checking the bits in the CAN Interrupt register, CiINTF.

  - Invalid Message Received:

    If any type of error occurred during reception of the last message, an error is indicated by the IVRIF bit.

  - Receiver Overrun:

    The RXnOVR bit indicates that an overrun condition occurred.

  - Receiver Warning:

    The RXWAR bit indicates that the Receive Error Counter (RERRCNT<7:0>) has reached the warning limit of 96.

  - Receiver Error Passive:

    The RXEP bit indicates that the Receive Error Counter has exceeded the error passive limit of 127 and the module has gone into error passive state.

17.5 Message Transmission

17.5.1 TRANSMIT BUFFERS

The CAN module has three transmit buffers. Each of the three buffers occupies 14 bytes of data. Eight of the bytes are the maximum 8 bytes of the transmitted message. Five bytes hold the standard and extended identifiers and other message arbitration information.

17.5.2 TRANSMIT MESSAGE PRIORITY

Transmit priority is a prioritization within each node of the pending transmittable messages. There are 4 levels of transmit priority. If TXPRI<1:0> (CiTXnCON<1:0>, where n = 0, 1 or 2, represents a particular transmit buffer) for a particular message buffer is set to '11', that buffer has the highest priority. If TXPRI<1:0> for a particular message buffer is set to '10' or '01', that buffer has an intermediate priority. If TXPRI<1:0> for a particular message buffer is '00', that buffer has the lowest priority.

17.5.3 TRANSMISSION SEQUENCE

To initiate transmission of the message, the TXREQ bit (CiTXnCON<3>) must be set. The CAN bus module resolves any timing conflicts between setting of the TXREQ bit and the Start-of-Frame (SOF), ensuring that if the priority was changed, it is resolved correctly before the SOF occurs. When TXREQ is set, the TXABT (CiTXnCON<6>), TXLARB (CiTXnCON<5>) and TXERR (CiTXnCON<4>) flag bits are automatically cleared.

Setting TXREQ bit simply flags a message buffer as enqueued for transmission. When the module detects an available bus, it begins transmitting the message which has been determined to have the highest priority.

If the transmission completes successfully on the first attempt, the TXREQ bit is cleared automatically and an interrupt is generated if TX1IE was set.

If the message transmission fails, one of the error condition flags is set, and the TXREQ bit remains set, indicating that the message is still pending for transmission. If the message encountered an error condition during the transmission attempt, the TXERR bit is set, and the error condition may cause an interrupt. If the message loses arbitration during the transmission attempt, the TXLARB bit is set. No interrupt is generated to signal the loss of arbitration.

17.5.4 ABORTING MESSAGE TRANSMISSION

The system can also abort a message by clearing the TXREQ bit associated with each message buffer. Setting the ABAT bit (CiCTRL<12>) requests an abort of all pending messages. If the message has not yet started transmission, or if the message started but is interrupted by loss of arbitration or an error, the abort is processed. The abort is indicated when the module sets the TXABT bit and the TXnIF flag is not automatically set.

17.5.5 TRANSMISSION ERRORS

The CAN module detects the following transmission errors:

- Acknowledge error
- Form error
- Bit error

These transmission errors do not necessarily generate an interrupt but are indicated by the transmission error counter. However, each of these errors causes the transmission error counter to be incremented by one. Once the value of the error counter exceeds the value of 96, the ERRIF (CiINTF<5>) and the TXWAR bit (CiINTF<10>) are set. Once the value of the error counter exceeds the value of 96, an interrupt is generated and the TXWAR bit in the Error Flag register is set.

17.5.6 TRANSMIT INTERRUPTS

Transmit interrupts can be divided into 2 major groups, each including various conditions that generate interrupts:

- Transmit Interrupt:

  At least one of the three transmit buffers is empty (not scheduled) and can be loaded to schedule a message for transmission. The TXnIF flags are read to determine which transmit buffer is available and caused the interrupt.

- Transmit Error Interrupts:

  A transmission error interrupt is indicated by the ERRIF flag. This flag shows that an error condition occurred. The source of the error can be determined by checking the error flags in the CAN Interrupt register, CiINTF. The flags in this register are related to receive and transmit errors.

  - Transmitter Warning Interrupt:

    The TXWAR bit indicates that the Transmit Error Counter has reached the CPU warning limit of 96.

  - Transmitter Error Passive:

    The TXEP bit (CiINTF<12>) indicates that the Transmit Error Counter has exceeded the error passive limit of 127 and the module has gone to error passive state.

  - Bus Off:

    The TXBO bit (CiINTF<13>) indicates that the Transmit Error Counter (TERRCNT<7:0>) has exceeded 255 and the module has gone to the bus off state.

17.6 Baud Rate Setting

All nodes on any particular CAN bus must have the same nominal bit rate. In order to set the baud rate, the following parameters have to be initialized:

- Synchronization Jump Width
- Baud Rate Prescaler
- Phase Segments
- Length determination of Phase Segment 2
- Sample Point
- Propagation Segment bits

17.6.1 BIT TIMING

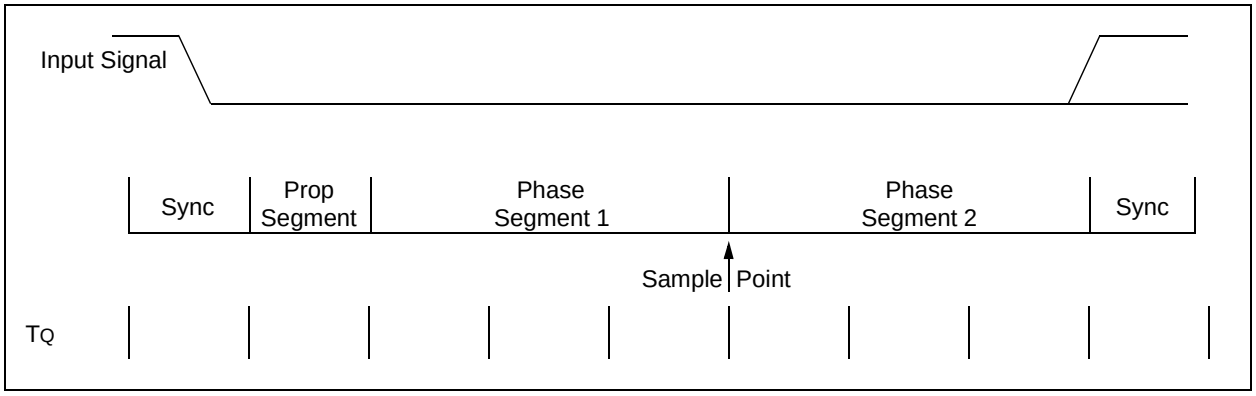
All controllers on the CAN bus must have the same baud rate and bit length. However, different controllers are not required to have the same master oscillator clock. At different clock frequencies of the individual controllers, the baud rate has to be adjusted by adjusting the number of time quanta in each segment.

The nominal bit time can be thought of as being divided into separate non-overlapping time segments. These segments are shown in Figure 17-2.

- Synchronization Segment (Sync Seg)
- Propagation Time Segment (Prop Seg)
- Phase Segment 1 (Phase1 Seg)
- Phase Segment 2 (Phase2 Seg)

The time segments and also the nominal bit time are made up of integer units of time called time quanta or TQ. By definition, the nominal bit time has a minimum of 8 TQ and a maximum of 25 TQ. Also, by definition, the minimum nominal bit time is 1 μsec corresponding to a maximum bit rate of 1 MHz.

FIGURE 17-2: CAN BIT TIMING



size and Frame Sync generator control bits. A new I^2S data transfer boundary is marked by a high-to-low or a low-to-high transition edge on the COFS pin.

18.3.6 SLAVE FRAME SYNC OPERATION

When the DCI module is operating as a Frame Sync slave (COFSD = 1), data transfers are controlled by the Codec device attached to the DCI module. The COFSM control bits control how the DCI module responds to incoming COFS signals.

In the Multichannel mode, a new data frame transfer begins one CSCK cycle after the COFS pin is sampled high (see Figure 18-2). The pulse on the COFS pin resets the Frame Sync generator logic.

In the I^2S mode, a new data word is transferred one CSCK cycle after a low-to-high or a high-to-low transition is sampled on the COFS pin. A rising or falling edge on the COFS pin resets the Frame Sync generator logic.

In the AC-Link mode, the tag slot and subsequent data slots for the next frame is transferred one CSCK cycle after the COFS pin is sampled high.

The COFSG and WS bits must be configured to provide the proper frame length when the module is operating in the Slave mode. Once a valid Frame Sync pulse has been sampled by the module on the COFS pin, an entire data frame transfer takes place. The module will not respond to further Frame Sync pulses until the data frame transfer has completed.

FIGURE 18-2: FRAME SYNC TIMING, MULTICHANNEL MODE

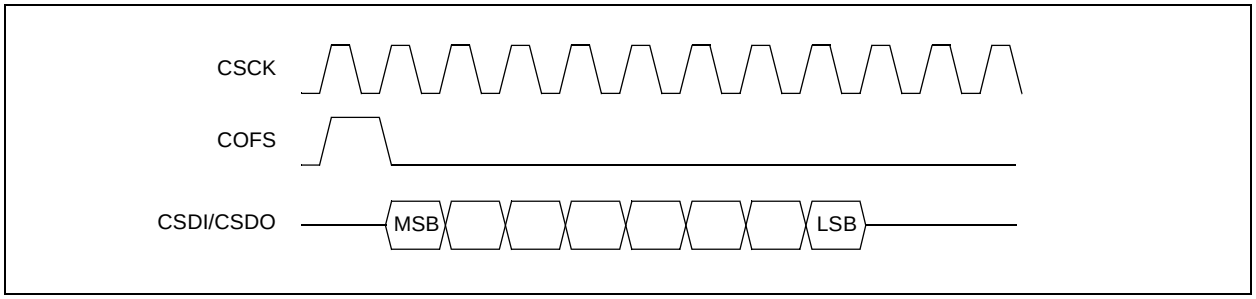


FIGURE 18-3: FRAME SYNC TIMING, AC-LINK START-OF-FRAME

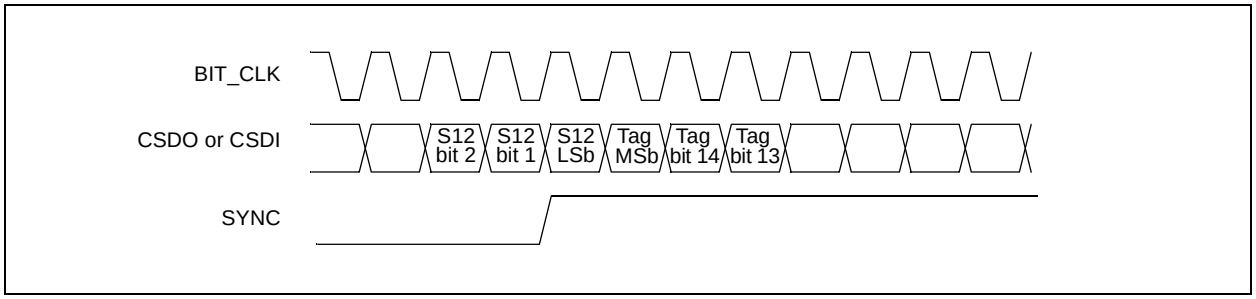
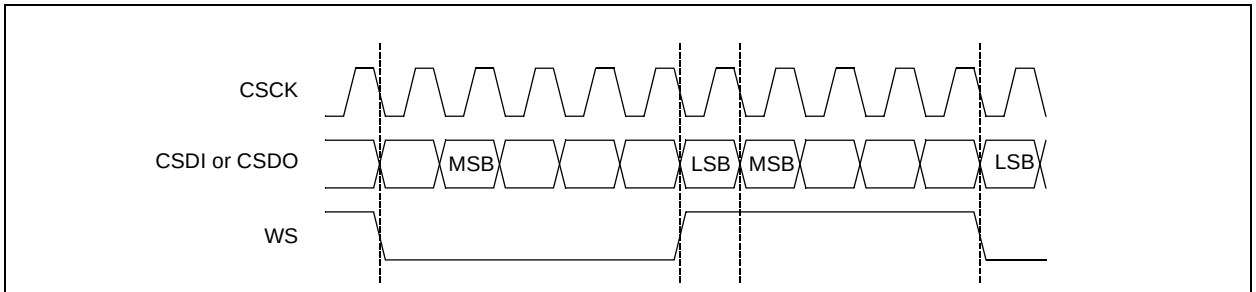


FIGURE 18-4: I^2S INTERFACE FRAME SYNC TIMING



Note: A 5-bit transfer is shown here for illustration purposes. The I^2S protocol does not specify word length – this will be system dependent.

18.6.2 20-BIT AC-LINK MODE

The 20-bit AC-Link mode allows all bits in the data time slots to be transmitted and received but does not maintain data alignment in the TXBUF and RXBUF registers.

The 20-bit AC-Link mode functions similar to the Multichannel mode of the DCI module, except for the duty cycle of the Frame Synchronization signal. The AC-Link Frame Synchronization signal should remain high for 16 CSCK cycles and should be low for the following 240 cycles.

The 20-bit mode treats each 256-bit AC-Link frame as sixteen, 16-bit time slots. In the 20-bit AC-Link mode, the module operates as if COFSG<3:0> = 1111 and WS<3:0> = 1111. The data alignment for 20-bit data slots is ignored. For example, an entire AC-Link data frame can be transmitted and received in a packed fashion by setting all bits in the TSCON and RSCON SFRs. Since the total available buffer length is 64 bits, it would take 4 consecutive interrupts to transfer the AC-Link frame. The application software must keep track of the current AC-Link frame segment.

18.7 I^2S Mode Operation

The DCI module is configured for I^2S mode by writing a value of '01' to the COFSM<1:0> control bits in the DCICON1 SFR. When operating in the I^2S mode, the DCI module generates Frame Synchronization signals with a 50% duty cycle. Each edge of the Frame Synchronization signal marks the boundary of a new data word transfer.

The user must also select the frame length and data word size using the COFSG and WS control bits in the DCICON2 SFR.

18.7.1 I^2S FRAME AND DATA WORD LENGTH SELECTION

The WS and COFSG control bits are set to produce the period for one half of an I^2S data frame. That is, the frame length is the total number of CSCK cycles required for a left or a right data word transfer.

The BLEN bits must be set for the desired buffer length. Setting BLEN<1:0> = 01 produces a CPU interrupt, once per I^2S frame.

18.7.2 I^2S DATA JUSTIFICATION

As per the I^2S specification, a data word transfer, by default, begins one CSCK cycle after a transition of the WS signal. A 'MSb left justified' option can be selected using the DJST control bit in the DCICON1 SFR.

If DJST = 1, the I^2S data transfers are MSb left justified. The MSb of the data word is presented on the CSDO pin during the same CSCK cycle as the rising or falling edge of the COFS signal. The CSDO pin is tri-stated after the data word has been sent.

21.0 INSTRUCTION SET SUMMARY

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 dsPIC30F instruction set adds many enhancements to the previous PIC® MCU instruction sets, while maintaining an easy migration from PIC MCU instruction sets.

Most instructions are a single program memory word (24 bits). Only three instructions require two program memory locations.

Each single-word instruction is a 24-bit word divided into an 8-bit opcode which specifies the instruction type, and one or more operands which further specify the operation of the instruction.

The instruction set is highly orthogonal and is grouped into five basic categories:

- Word or byte-oriented operations
- Bit-oriented operations
- Literal operations
- DSP operations
- Control operations

Table 21-1 shows the general symbols used in describing the instructions.

The dsPIC30F instruction set summary in Table 21-2 lists all the instructions, along with the status flags affected by each instruction.

Most word or byte-oriented W register instructions (including barrel shift instructions) have three operands:

- The first source operand which is typically a register 'Wb' without any address modifier
- The second source operand which is typically a register 'Ws' with or without an address modifier
- The destination of the result which is typically a register 'Wd' with or without an address modifier

However, word or byte-oriented file register instructions have two operands:

- The file register specified by the value 'f'
- The destination, which could either be the file register 'f' or the W0 register, which is denoted as 'WREG'

Most bit-oriented instructions (including simple rotate/shift instructions) have two operands:

- The W register (with or without an address modifier) or file register (specified by the value of 'Ws' or 'f')
- The bit in the W register or file register (specified by a literal value or indirectly by the contents of register 'Wb')

The literal instructions that involve data movement may use some of the following operands:

- A literal value to be loaded into a W register or file register (specified by the value of 'k')
- The W register or file register where the literal value is to be loaded (specified by 'Wb' or 'f')

However, literal instructions that involve arithmetic or logical operations use some of the following operands:

- The first source operand which is a register 'Wb' without any address modifier
- The second source operand which is a literal value
- The destination of the result (only if not the same as the first source operand) which is typically a register 'Wd' with or without an address modifier

The MAC class of DSP instructions may use some of the following operands:

- The accumulator (A or B) to be used (required operand)
- The W registers to be used as the two operands
- The X and Y address space prefetch operations
- The X and Y address space prefetch destinations
- The accumulator write-back destination

The other DSP instructions do not involve any multiplication, and may include:

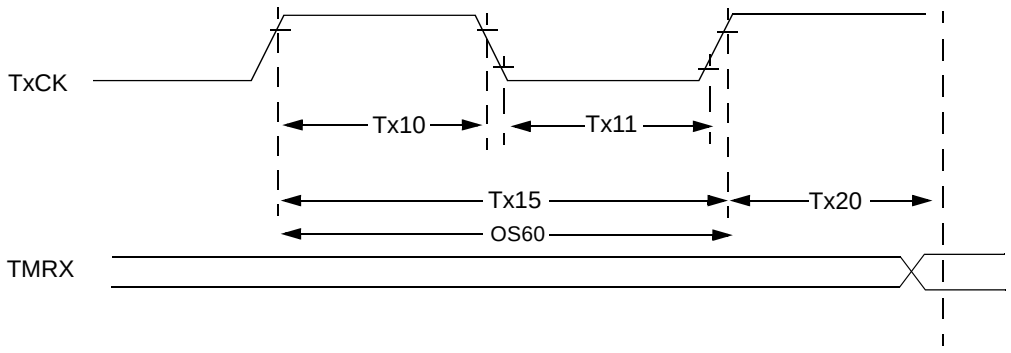
- The accumulator to be used (required)
- The source or destination operand (designated as Wso or Wdo, respectively) with or without an address modifier
- The amount of shift specified by a W register 'Wn' or a literal value

The control instructions may use some of the following operands:

- A program memory address
- The mode of the table read and table write instructions

All instructions are a single word, except for certain double word instructions, which were made double word instructions so that all the required information is available in these 48 bits. In the second word, the 8 MSbs are '0's. If this second word is executed as an instruction (by itself), it executes as a NOP.

FIGURE 23-8: TYPE A, B AND C TIMER EXTERNAL CLOCK TIMING CHARACTERISTICS



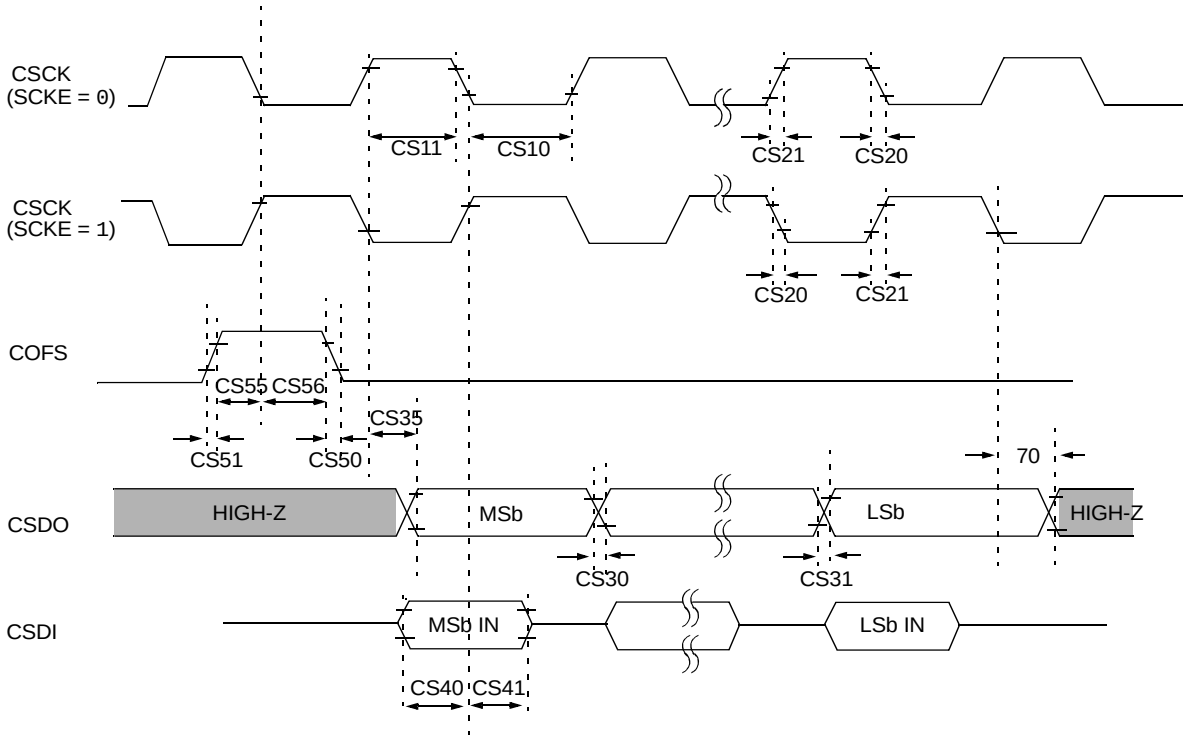
Note: Refer to Figure 23-3 for load conditions.

TABLE 23-22: TYPE A TIMER (TIMER1) 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**
TA10	TTXH	TxCK High Time	Synchronous, no prescaler	0.5 TCY + 20	—	—	ns	Must also meet parameter TA15
			Synchronous, with prescaler	10	—	—	ns	
			Asynchronous	10	—	—	ns	
TA11	TTXL	TxCK Low Time	Synchronous, no prescaler	0.5 TCY + 20	—	—	ns	Must also meet parameter TA15
			Synchronous, with prescaler	10	—	—	ns	
			Asynchronous	10	—	—	ns	
TA15	TTXP	TxCK Input Period	Synchronous, no prescaler	TCY + 10	—	—	ns	
			Synchronous, with prescaler	Greater of: 20 ns or (TCY + 40)/N	—	—	—	N = prescale value (1, 8, 64, 256)
			Asynchronous	20	—	—	ns	
OS60	Ft1	SOSC1/T1CK Oscillator Input Frequency Range (oscillator enabled by setting bit, TCS (T1CON<1>)		DC	—	50	kHz	
TA20	TCKEXTMRL	Delay from External TxCK Clock Edge to Timer Increment		0.5 TCY	—	1.5 TCY	—	

Note 1: Timer1 is a Type A.

FIGURE 23-12: DCI MODULE (MULTICHANNEL, I²S MODES) TIMING CHARACTERISTICS



Note: Refer to Figure 23-3 for load conditions.

TABLE 23-31: SPI MODULE MASTER MODE (CKE = 1) 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(1)**	**Min**	**Typ(2)**	**Max**	**Units**	**Conditions**
SP10	TscL	SCKx Output Low Time(3)	TCY/2	—	—	ns	
SP11	TscH	SCKx Output High Time(3)	TCY/2	—	—	ns	
SP20	TscF	SCKx Output Fall Time(4)	—	—	—	ns	See parameter DO32
SP21	TscR	SCKx Output Rise Time(4)	—	—	—	ns	See parameter DO31
SP30	TdoF	SDOx Data Output Fall Time(4)	—	—	—	ns	See parameter DO32
SP31	TdoR	SDOx Data Output Rise Time(4)	—	—	—	ns	See parameter DO31
SP35	TscH2do, TscL2doV	SDOx Data Output Valid after SCKx Edge	—	—	30	ns	
SP36	TdoV2sc, TdoV2scL	SDOx Data Output Setup to First SCKx Edge	30	—	—	ns	
SP40	TdiV2scH, TdiV2scL	Setup Time of SDIx Data Input to SCKx Edge	20	—	—	ns	
SP41	TscH2diL, TscL2diL	Hold Time of SDIx Data Input to SCKx Edge	20	—	—	ns	

Note 1: These parameters are characterized but not tested in manufacturing.

2: Data in "Typ" column is at 5V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

3: The minimum clock period for SCKx is 100 ns. Therefore, the clock generated in Master mode must not violate this specification.

4: Assumes 50 pF load on all SPI pins.

FIGURE 23-16: SPI MODULE SLAVE MODE (CKE = 0) TIMING CHARACTERISTICS

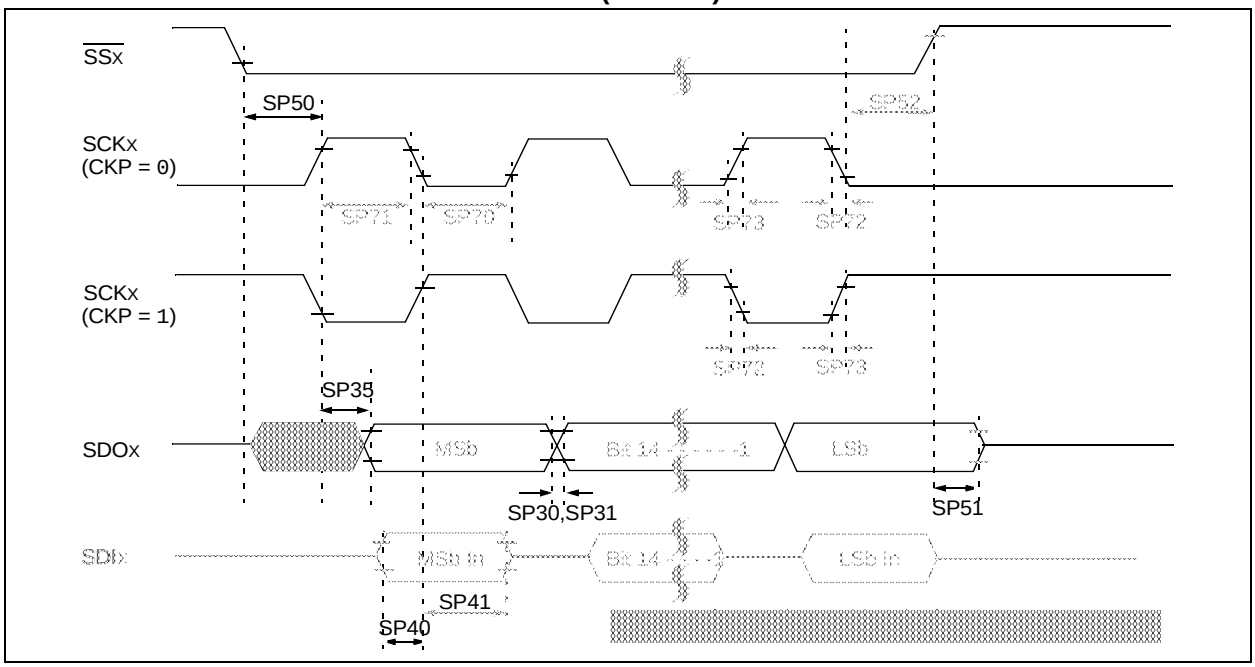
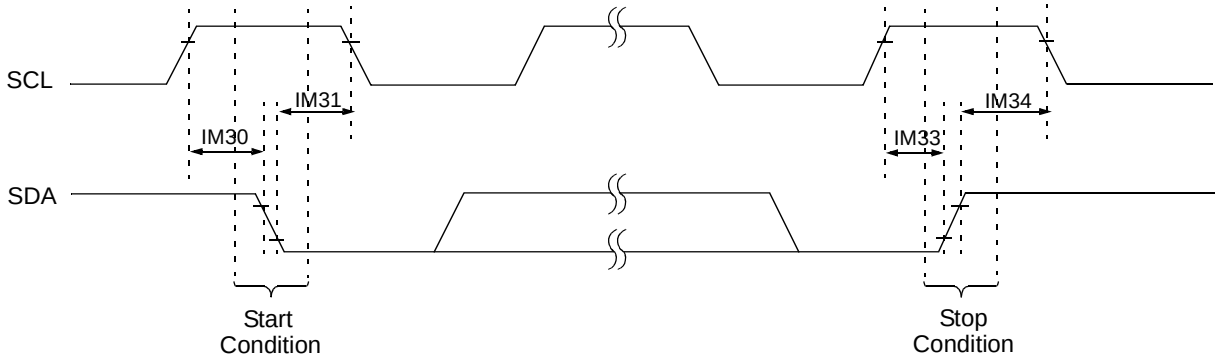


FIGURE 23-18: I²C™ BUS START/STOP BITS TIMING CHARACTERISTICS (MASTER MODE)



Note: Refer to Figure 23-3 for load conditions.

FIGURE 23-19: I²C™ BUS DATA TIMING CHARACTERISTICS (MASTER MODE)

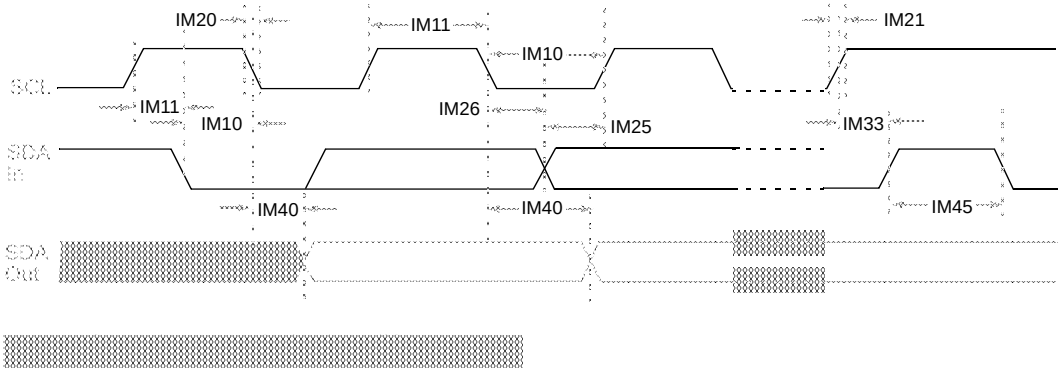


TABLE 23-34: I²C™ BUS DATA TIMING REQUIREMENTS (MASTER MODE)

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(1)**	**Max**	**Units**	**Conditions**
IM10	TLO:SCL	Clock Low Time	100 kHz mode	TCY/2 (BRG + 1)	—	µs	
			400 kHz mode	TCY/2 (BRG + 1)	—	µs	
			1 MHz mode(2)	TCY/2 (BRG + 1)	—	µs	
IM11	THI:SCL	Clock High Time	100 kHz mode	TCY/2 (BRG + 1)	—	µs	
			400 kHz mode	TCY/2 (BRG + 1)	—	µs	
			1 MHz mode(2)	TCY/2 (BRG + 1)	—	µs	
IM20	TF:SCL	SDA and SCL Fall Time	100 kHz mode	—	300	ns	CB is specified to be from 10 to 400 pF
			400 kHz mode	20 + 0.1 CB	300	ns	
			1 MHz mode(2)	—	100	ns	

Note 1: BRG is the value of the I²C Baud Rate Generator. Refer to **Section 21. "Inter-Integrated Circuit™ (I²C)"** in the "*dsPIC30F Family Reference Manual*" (DS70046).

2: Maximum pin capacitance = 10 pF for all I²C pins (for 1 MHz mode only).

FIGURE 23-22: CAN MODULE I/O TIMING CHARACTERISTICS

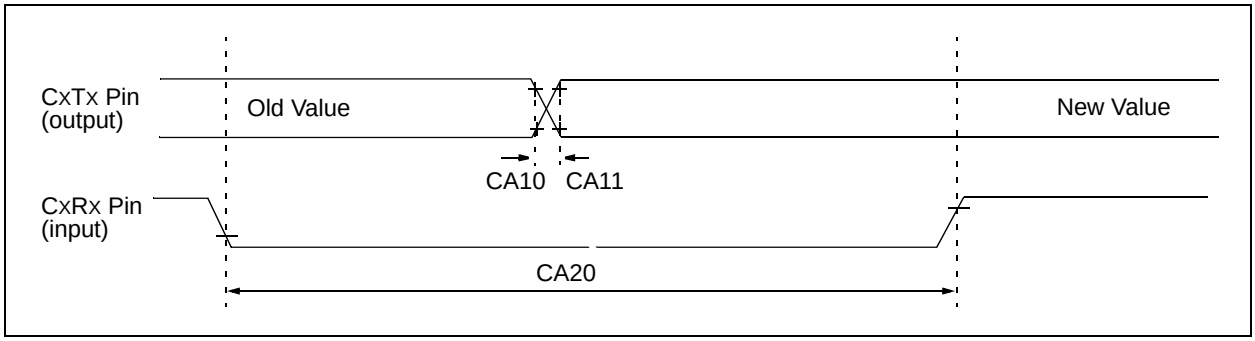


TABLE 23-36: CAN MODULE I/O TIMING REQUIREMENTS

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(1)**	**Min**	**Typ(2)**	**Max**	**Units**	**Conditions**
CA10	TioF	Port Output Fall Time	—	10	25	ns	
CA11	TioR	Port Output Rise Time	—	10	25	ns	
CA20	Tcwf	Pulse Width to Trigger CAN Wake-up Filter	500	—	—	ns	

Note 1: These parameters are characterized but not tested in manufacturing.

2: Data in "Typ" column is at 5V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

NOTES: