Details	
Product Status	Active
Core Processor	dsPIC
Core Size	16-Bit
Speed	30 MIPs
Connectivity	CANbus, I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	68
Program Memory Size	132KB (44K x 24)
Program Memory Type	FLASH
EEPROM Size	2K x 8
RAM Size	6K x 8
Voltage - Supply (Vcc/Vdd)	2.5V ~ 5.5V
Data Converters	A/D 16x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	80-TQFP
Supplier Device Package	80-TQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/dspic30f6013-30i-pf

Table of Contents

1.0 Device Overview 9
2.0 CPU Architecture Overview..... 15
3.0 Memory Organization..... 25
4.0 Address Generator Units..... 39
5.0 Interrupts 45
6.0 Flash Program Memory..... 51
7.0 Data EEPROM Memory 57
8.0 I/O Ports 63
9.0 Timer1 Module 69
10.0 Timer2/3 Module 73
11.0 Timer4/5 Module 79
12.0 Input Capture Module..... 83
13.0 Output Compare Module 87
14.0 SPI Module..... 91
15.0 I2C Module..... 95
16.0 Universal Asynchronous Receiver Transmitter (UART) Module 103
17.0 CAN Module..... 111
18.0 Data Converter Interface (DCI) Module..... 125
19.0 12-bit Analog-to-Digital Converter (A/D) Module..... 135
20.0 System Integration 145
21.0 Instruction Set Summary..... 161
22.0 Development Support..... 169
23.0 Electrical Characteristics..... 173
24.0 Packaging Information..... 211
Appendix A: Revision History..... 215
Index 217
The Microchip Web Site..... 223
Customer Change Notification Service 223
Customer Support..... 223
Reader Response 224
Product Identification System 225

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@microchip.com** or fax the **Reader Response Form** in the back of this data sheet to (480) 792-4150. We welcome your feedback.

Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000A is version A of document DS30000).

Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; http://www.microchip.com
- Your local Microchip sales office (see last page)

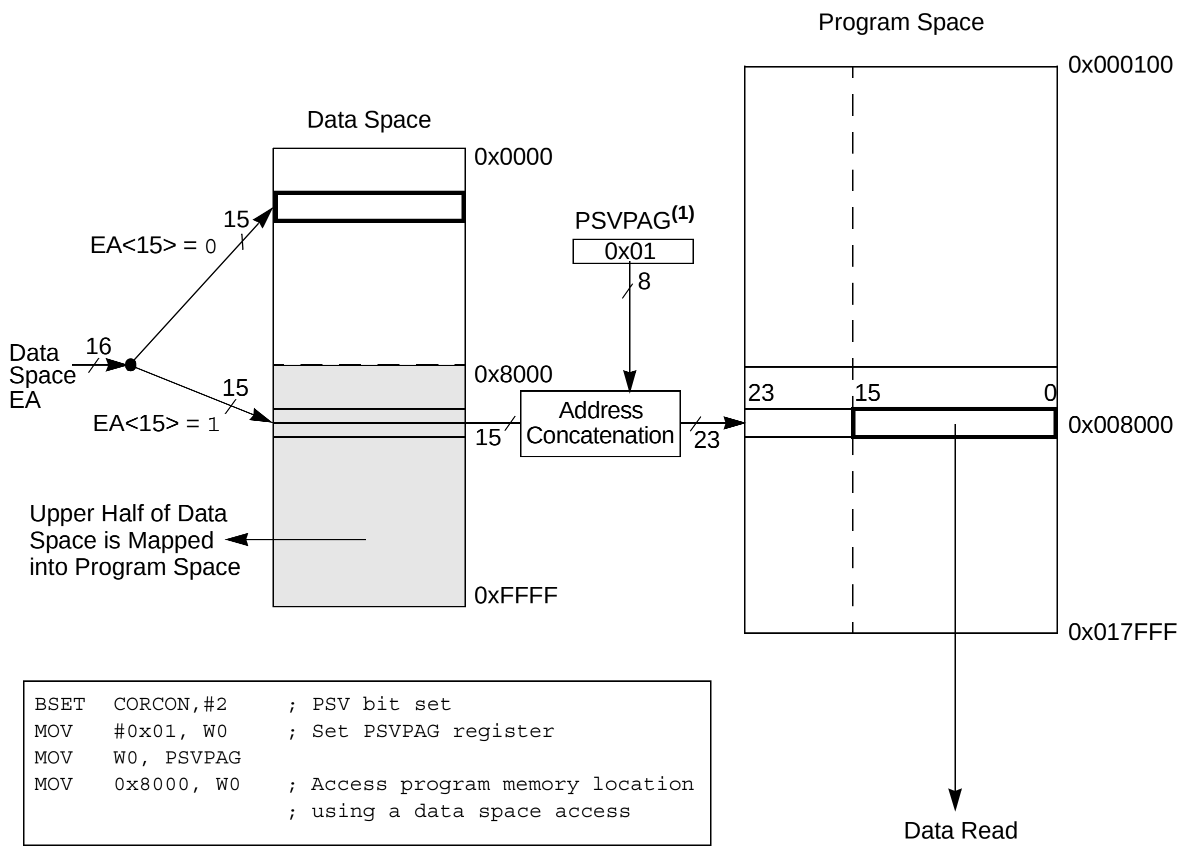
When contacting a sales office, please specify which device, revision of silicon and data sheet (include literature number) you are using.

Customer Notification System

Register on our web site at **www.microchip.com** to receive the most current information on all of our products.

NOTES:

FIGURE 3-6: DATA SPACE WINDOW INTO PROGRAM SPACE OPERATION




```
BSET   CORCON,#2     ; PSV bit set
MOV    #0x01, W0     ; Set PSVPAG register
MOV    W0, PSVPAG
MOV    0x8000, W0    ; Access program memory location
                     ; using a data space access
```

Note: PSVPAG is an 8-bit register, containing bits <22:15> of the program space address (i.e., it defines the page in program space to which the upper half of data space is being mapped).

FIGURE 3-9: DATA SPACE FOR MCU AND DSP (MAC CLASS) INSTRUCTIONS EXAMPLE

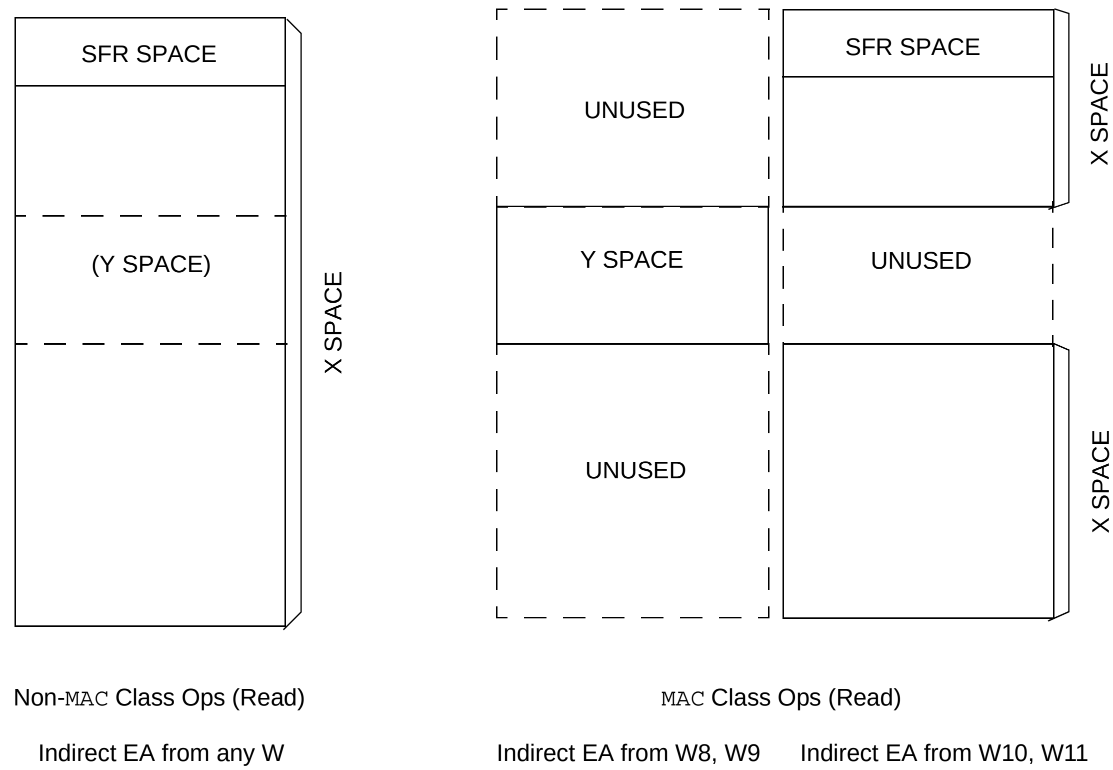


TABLE 3-2: EFFECT OF INVALID MEMORY ACCESSES

Attempted Operation	Data Returned
EA = an unimplemented address	0x0000
W8 or W9 used to access Y data space in a MAC instruction	0x0000
W10 or W11 used to access X data space in a MAC instruction	0x0000

All effective addresses are 16 bits wide and point to bytes within the data space. Therefore, the data space address range is 64 Kbytes or 32K words.

3.2.3 DATA SPACE WIDTH

The core data width is 16 bits. All internal registers are organized as 16-bit wide words. Data space memory is organized in byte addressable, 16-bit wide blocks.

3.2.4 DATA ALIGNMENT

To help maintain backward compatibility with PIC® MCU devices and improve data space memory usage efficiency, the dsPIC30F instruction set supports both word and byte operations. Data is aligned in data memory and registers as words, but all data space EAs resolve to bytes. Data byte reads will read the complete word which contains the byte, using the LSb of any EA to determine which byte to select. The selected byte is placed onto the LSB of the X data path (no byte accesses are possible from the Y data path as the MAC class of instruction can only fetch words). That is, data memory and registers are organized as two parallel byte wide entities with shared (word) address decode but separate write lines. Data byte writes only write to the corresponding side of the array or register which matches the byte address.

As a consequence of this byte accessibility, all effective address calculations (including those generated by the DSP operations which are restricted to word sized data) are internally scaled to step through word aligned memory. For example, the core would recognize that Post-Modified Register Indirect Addressing mode [Ws++] will result in a value of Ws + 1 for byte operations and Ws + 2 for word operations.

4.2.3 MODULO ADDRESSING APPLICABILITY

Modulo Addressing can be applied to the effective address calculation associated with any W register. It is important to realize that the address boundaries check for addresses less than, or greater than the upper (for incrementing buffers), and lower (for decrementing buffers) boundary addresses (not just equal to). Address changes may, therefore, jump beyond boundaries and still be adjusted correctly.

Note:	The modulo corrected effective address is written back to the register only when Pre-Modify or Post-Modify Addressing mode is used to compute the effective address. When an address offset (e.g., [W7 + W2]) is used, modulo address correction is performed but the contents of the register remain unchanged.

4.3 Bit-Reversed Addressing

Bit-Reversed Addressing is intended to simplify data re-ordering for radix-2 FFT algorithms. It is supported by the X AGU for data writes only.

The modifier, which may be a constant value or register contents, is regarded as having its bit order reversed. The address source and destination are kept in normal order. Thus, the only operand requiring reversal is the modifier.

4.3.1 BIT-REVERSED ADDRESSING IMPLEMENTATION

Bit-Reversed Addressing is enabled when:

1. BWM (W register selection) in the MODCON register is any value other than '`15`' (the stack cannot be accessed using Bit-Reversed Addressing) **and**
2. the BREN bit is set in the XBREV register **and**
3. the Addressing mode used is Register Indirect with Pre-Increment or Post-Increment.

If the length of a bit-reversed buffer is $M = 2^N$ bytes, then the last 'N' bits of the data buffer start address must be zeros.

XB<14:0> is the bit-reversed address modifier or 'pivot point' which is typically a constant. In the case of an FFT computation, its value is equal to half of the FFT data buffer size.

Note:	All bit-reversed EA calculations assume word sized data (LSb of every EA is always clear). The XB value is scaled accordingly to generate compatible (byte) addresses.

When enabled, Bit-Reversed Addressing will only be executed for register indirect with pre-increment or post-increment addressing and word sized data writes. It will not function for any other addressing mode or for byte sized data, and normal addresses will be generated instead. When Bit-Reversed Addressing is active, the W address pointer will always be added to the address modifier (XB) and the offset associated with the Register Indirect Addressing mode will be ignored. In addition, as word sized data is a requirement, the LSb of the EA is ignored (and always clear).

Note:	Modulo Addressing and Bit-Reversed Addressing should not be enabled together. In the event that the user attempts to do this, Bit-Reversed Addressing will assume priority when active for the X WAGU, and X WAGU Modulo Addressing will be disabled. However, Modulo Addressing will continue to function in the X RAGU.

If Bit-Reversed Addressing has already been enabled by setting the BREN (XBREV<15>) bit, then a write to the XBREV register should not be immediately followed by an indirect read operation using the W register that has been designated as the bit-reversed pointer.

6.6.3 LOADING WRITE LATCHES

Example 6-2 shows a sequence of instructions that can be used to load the 96 bytes of write latches. 32 TBLWTL and 32 TBLWTH instructions are needed to load the write latches selected by the table pointer.

EXAMPLE 6-2: LOADING WRITE LATCHES

```
; Set up a pointer to the first program memory location to be written
; program memory selected, and writes enabled
       MOV     #0x0000,W0                  ;
       MOV     W0,TBLPAG                   ; Initialize PM Page Boundary SFR
       MOV     #0x6000,W0                  ; An example program memory address
; Perform the TBLWT instructions to write the latches
; 0th_program_word
       MOV     #LOW_WORD_0,W2              ;
       MOV     #HIGH_BYTE_0,W3             ;
       TBLWTL  W2,[W0]                     ; Write PM low word into program latch
       TBLWTH  W3,[W0++]                   ; Write PM high byte into program latch
; 1st_program_word
       MOV     #LOW_WORD_1,W2              ;
       MOV     #HIGH_BYTE_1,W3             ;
       TBLWTL  W2,[W0]                     ; Write PM low word into program latch
       TBLWTH  W3,[W0++]                   ; Write PM high byte into program latch
;  2nd_program_word
       MOV     #LOW_WORD_2,W2              ;
       MOV     #HIGH_BYTE_2,W3             ;
       TBLWTL  W2, [W0]                    ; Write PM low word into program latch
       TBLWTH  W3, [W0++]                  ; Write PM high byte into program latch
       •
       •
       •
; 31st_program_word
       MOV     #LOW_WORD_31,W2             ;
       MOV     #HIGH_BYTE_31,W3            ;
       TBLWTL  W2, [W0]                    ; Write PM low word into program latch
       TBLWTH  W3, [W0++]                  ; Write PM high byte into program latch
```

Note: In Example 6-2, the contents of the upper byte of W3 has no effect.

6.6.4 INITIATING THE PROGRAMMING SEQUENCE

For protection, the write initiate sequence for NVMKEY must be used to allow any erase or program operation to proceed. After the programming command has been executed, the user must wait for the programming time until programming is complete. The two instructions following the start of the programming sequence should be NOPs.

EXAMPLE 6-3: INITIATING A PROGRAMMING SEQUENCE

```
       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
```

9.4 Timer Interrupt

The 16-bit timer has the ability to generate an interrupt on period match. When the timer count matches the Period register, the T1IF bit is asserted and an interrupt will be generated if enabled. The T1IF bit must be cleared in software. The timer interrupt flag, T1IF, is located in the IFS0 Control register in the interrupt controller.

When the Gated Time Accumulation mode is enabled, an interrupt will also be generated on the falling edge of the gate signal (at the end of the accumulation cycle).

Enabling an interrupt is accomplished via the respective timer interrupt enable bit, T1IE. The timer interrupt enable bit is located in the IEC0 Control register in the interrupt controller.

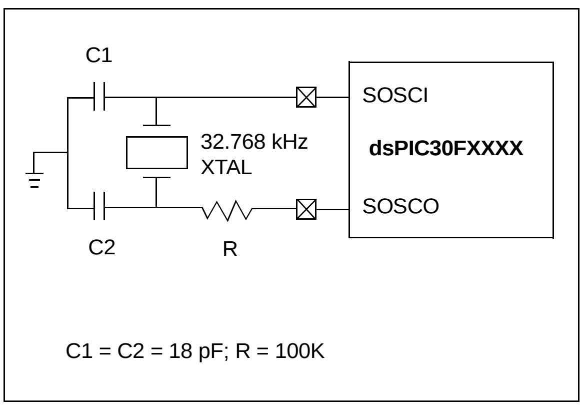
9.5 Real-Time Clock

Timer1, when operating in Real-Time Clock (RTC) mode, provides time of day and event time-stamping capabilities. Key operational features of the RTC are:

- Operation from 32 kHz LP oscillator
- 8-bit prescaler
- Low power
- Real-Time Clock interrupts

These Operating modes are determined by setting the appropriate bit(s) in the T1CON Control register.

FIGURE 9-2: RECOMMENDED COMPONENTS FOR TIMER1 LP OSCILLATOR RTC



9.5.1 RTC OSCILLATOR OPERATION

When the TON = 1, TCS = 1 and TGATE = 0, the timer increments on the rising edge of the 32 kHz LP oscillator output signal, up to the value specified in the Period register and is then Reset to '0'.

The TSYNC bit must be asserted to a logic '0' (Asynchronous mode) for correct operation.

Enabling LPOSCEN (OSCCON<1>) will disable the normal Timer and Counter modes and enable a timer carry-out wake-up event.

When the CPU enters Sleep mode, the RTC will continue to operate provided the 32 kHz external crystal oscillator is active and the control bits have not been changed. The TSIDL bit should be cleared to '0' in order for RTC to continue operation in Idle mode.

9.5.2 RTC INTERRUPTS

When an interrupt event occurs, the respective interrupt flag, T1IF, is asserted and an interrupt will be generated if enabled. The T1IF bit must be cleared in software. The respective Timer interrupt flag, T1IF, is located in the IFS0 Status register in the interrupt controller.

Enabling an interrupt is accomplished via the respective timer interrupt enable bit, T1IE. The timer interrupt enable bit is located in the IEC0 Control register in the interrupt controller.

TABLE 11-1: TIMER4/5 REGISTER MAP

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

Legend: u = uninitialized

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

13.1 Timer2 and Timer3 Selection Mode

Each output compare channel can select between one of two 16-bit timers, Timer2 or Timer3.

The selection of the timers is controlled by the OCTSEL bit (OCxCON<3>). Timer2 is the default timer resource for the output compare module.

13.2 Simple Output Compare Match Mode

When control bits OCM<2:0> (OCxCON<2:0>) = `001`, `010` or `011`, the selected output compare channel is configured for one of three simple Output Compare Match modes:

- Compare forces I/O pin low
- Compare forces I/O pin high
- Compare toggles I/O pin

The OCxR register is used in these modes. The OCxR register is loaded with a value and is compared to the selected incrementing timer count. When a compare occurs, one of these Compare Match modes occurs. If the counter resets to zero before reaching the value in OCxR, the state of the OCx pin remains unchanged.

13.3 Dual Output Compare Match Mode

When control bits OCM<2:0> (OCxCON<2:0>) = `100` or `101`, the selected output compare channel is configured for one of two Dual Output Compare modes, which are:

- Single Output Pulse mode
- Continuous Output Pulse mode

13.3.1 SINGLE PULSE MODE

For the user to configure the module for the generation of a single output pulse, the following steps are required (assuming timer is off):

- Determine instruction cycle time T_{CY}.
- Calculate desired pulse width value based on T_{CY}.
- Calculate time to start pulse from timer start value of 0x0000.
- Write pulse width start and stop times into OCxR and OCxRS Compare registers (x denotes channel 1, 2, ...,N).
- Set Timer Period register to value equal to, or greater than value in OCxRS Compare register.
- Set OCM<2:0> = `100`.
- Enable timer, TON (TxCON<15>) = `1`.

To initiate another single pulse, issue another write to set OCM<2:0> = `100`.

13.3.2 CONTINUOUS PULSE MODE

For the user to configure the module for the generation of a continuous stream of output pulses, the following steps are required:

- Determine instruction cycle time T_{CY}.
- Calculate desired pulse value based on T_{CY}.
- Calculate timer to start pulse width from timer start value of 0x0000.
- Write pulse width start and stop times into OCxR and OCxRS (x denotes channel 1, 2, ...,N) Compare registers, respectively.
- Set Timer Period register to value equal to, or greater than value in OCxRS Compare register.
- Set OCM<2:0> = `101`.
- Enable timer, TON (TxCON<15>) = `1`.

13.4 Simple PWM Mode

When control bits OCM<2:0> (OCxCON<2:0>) = `110` or `111`, the selected output compare channel is configured for the PWM mode of operation. When configured for the PWM mode of operation, OCxR is the main latch (read only) and OCxRS is the secondary latch. This enables glitchless PWM transitions.

The user must perform the following steps in order to configure the output compare module for PWM operation:

1. Set the PWM period by writing to the appropriate period register.
2. Set the PWM duty cycle by writing to the OCxRS register.
3. Configure the output compare module for PWM operation.
4. Set the TMRx prescale value and enable the Timer, TON (TxCON<15>) = `1`.

13.4.1 INPUT PIN FAULT PROTECTION FOR PWM

When control bits OCM<2:0> (OCxCON<2:0>) = `111`, the selected output compare channel is again configured for the PWM mode of operation with the additional feature of input Fault protection. While in this mode, if a logic '`0`' is detected on the OCFA/B pin, the respective PWM output pin is placed in the high impedance input state. The OCFLT bit (OCxCON<4>) indicates whether a Fault condition has occurred. This state will be maintained until both of the following events have occurred:

- The external Fault condition has been removed.
- The PWM mode has been re-enabled by writing to the appropriate control bits.

17.0 CAN 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).

17.1 Overview

The Controller Area Network (CAN) module is a serial interface, useful for communicating with other CAN modules or microcontroller devices. This interface/protocol was designed to allow communications within noisy environments.

The CAN module is a communication controller implementing the CAN 2.0 A/B protocol, as defined in the BOSCH specification. The module will support CAN 1.2, CAN 2.0A, CAN 2.0B Passive, and CAN 2.0B Active versions of the protocol. The module implementation is a full CAN system. The CAN specification is not covered within this data sheet. The reader may refer to the BOSCH CAN specification for further details.

The module features are as follows:

- Implementation of the CAN protocol CAN 1.2, CAN 2.0A and CAN 2.0B
- Standard and extended data frames
- 0-8 bytes data length
- Programmable bit rate up to 1 Mbit/sec
- Support for remote frames
- Double-buffered receiver with two prioritized received message storage buffers (each buffer may contain up to 8 bytes of data)
- 6 full (standard/extended identifier) acceptance filters, 2 associated with the high priority receive buffer and 4 associated with the low priority receive buffer
- 2 full acceptance filter masks, one each associated with the high and low priority receive buffers
- Three transmit buffers with application specified prioritization and abort capability (each buffer may contain up to 8 bytes of data)
- Programmable wake-up functionality with integrated low-pass filter
- Programmable Loopback mode supports self-test operation
- Signaling via interrupt capabilities for all CAN receiver and transmitter error states
- Programmable clock source
- Programmable link to Input Capture module (IC2, for both CAN1 and CAN2) for time-stamping and network synchronization
- Low-power Sleep and Idle mode

The CAN bus module consists of a protocol engine and message buffering/control. The CAN protocol engine handles all functions for receiving and transmitting messages on the CAN bus. Messages are transmitted by first loading the appropriate data registers. Status and errors can be checked by reading the appropriate registers. Any message detected on the CAN bus is checked for errors and then matched against filters to see if it should be received and stored in one of the receive registers.

17.2 Frame Types

The CAN module transmits various types of frames which include data messages or remote transmission requests initiated by the user, as other frames that are automatically generated for control purposes. The following frame types are supported:

- Standard Data Frame:

  A standard data frame is generated by a node when the node wishes to transmit data. It includes an 11-bit standard identifier (SID) but not an 18-bit extended identifier (EID).

- Extended Data Frame:

  An extended data frame is similar to a standard data frame but includes an extended identifier as well.

- Remote Frame:

  It is possible for a destination node to request the data from the source. For this purpose, the destination node sends a remote frame with an identifier that matches the identifier of the required data frame. The appropriate data source node will then send a data frame as a response to this remote request.

- Error Frame:

  An error frame is generated by any node that detects a bus error. An error frame consists of 2 fields: an error flag field and an error delimiter field.

- Overload Frame:

  An overload frame can be generated by a node as a result of 2 conditions. First, the node detects a dominant bit during interframe space which is an illegal condition. Second, due to internal conditions, the node is not yet able to start reception of the next message. A node may generate a maximum of 2 sequential overload frames to delay the start of the next message.

- Interframe Space:

  Interframe space separates a proceeding frame (of whatever type) from a following data or remote frame.

17.4 Message Reception

17.4.1 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.

17.4.2 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 (MAB), 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.

17.4.3 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.

17.4.4 RECEIVE OVERRUN

An overrun condition occurs when the Message Assembly Buffer (MAB) 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, indicating that both RXB0 and RXB1 are full, the message will be lost and an overrun will be indicated for RXB1.

17.4.5 RECEIVE ERRORS

The CAN module will detect the following receive errors:

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

These receive errors do not generate an interrupt. However, 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.

17.4.6 RECEIVE INTERRUPTS

Receive interrupts can be divided into 3 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.

19.1 ADC Result Buffer

The module contains a 16-word dual port read only buffer, called ADCBUF0...ADCBUFF, to buffer the ADC results. The RAM is 12 bits wide but the data obtained is represented in one of four different 16-bit data formats. The contents of the sixteen A/D Conversion Result Buffer registers, ADCBUF0 through ADCBUFF, cannot be written by user software.

19.2 Conversion Operation

After the ADC module has been configured, the sample acquisition is started by setting the SAMP bit. Various sources, such as a programmable bit, timer time-outs and external events, will terminate acquisition and start a conversion. When the A/D conversion is complete, the result is loaded into ADCBUF0...ADCBUFF, and the DONE bit and the A/D interrupt flag ADIF are set after the number of samples specified by the SMPI bit. The ADC module can be configured for different interrupt rates as described in **Section 19.3 "Selecting the Conversion Sequence"**.

The following steps should be followed for doing an A/D conversion:

1. Configure the ADC module:
   - Configure analog pins, voltage reference and digital I/O
   - Select ADC input channels
   - Select ADC conversion clock
   - Select ADC conversion trigger
   - Turn on ADC module
2. Configure ADC interrupt (if required):
   - Clear ADIF bit
   - Select ADC interrupt priority
3. Start sampling.
4. Wait the required acquisition time.
5. Trigger acquisition end, start conversion:
6. Wait for A/D conversion to complete, by either:
   - Waiting for the ADC interrupt, or
   - Waiting for the DONE bit to get set.
7. Read ADC result buffer, clear ADIF if required.

19.3 Selecting the Conversion Sequence

Several groups of control bits select the sequence in which the A/D connects inputs to the sample/hold channel, converts a channel, writes the buffer memory and generates interrupts.

The sequence is controlled by the sampling clocks.

The SMPI bits select the number of acquisition/ conversion sequences that would be performed before an interrupt occurs. This can vary from 1 sample per interrupt to 16 samples per interrupt.

The BUFM bit will split the 16-word results buffer into two 8-word groups. Writing to the 8-word buffers will be alternated on each interrupt event.

Use of the BUFM bit will depend on how much time is available for the moving of the buffers after the interrupt.

If the processor can quickly unload a full buffer within the time it takes to acquire and convert one channel, the BUFM bit can be '`0`' and up to 16 conversions (corresponding to the 16 input channels) may be done per interrupt. The processor will have one acquisition and conversion time to move the sixteen conversions.

If the processor cannot unload the buffer within the acquisition and conversion time, the BUFM bit should be '`1`'. For example, if SMPI<3:0> (ADCON2<5:2>) = `0111`, then eight conversions will be loaded into 1/2 of the buffer, following which an interrupt occurs. The next eight conversions will be loaded into the other 1/2 of the buffer. The processor will have the entire time between interrupts to move the eight conversions.

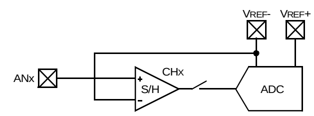
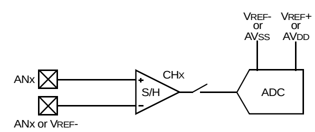
The ALTS bit can be used to alternate the inputs selected during the sampling sequence. The input multiplexer has two sets of sample inputs: MUX A and MUX B. If the ALTS bit is '`0`', only the MUX A inputs are selected for sampling. If the ALTS bit is '`1`' and SMPI<3:0> = `0000` on the first sample/convert sequence, the MUX A inputs are selected and on the next acquire/convert sequence, the MUX B inputs are selected.

The CSCNA bit (ADCON2<10>) will allow the multiplexer input to be alternately scanned across a selected number of analog inputs for the MUX A group. The inputs are selected by the ADCSSL register. If a particular bit in the ADCSSL register is '`1`', the corresponding input is selected. The inputs are always scanned from lower to higher numbered inputs, starting after each interrupt. If the number of inputs selected is greater than the number of samples taken per interrupt, the higher numbered inputs are unused.

19.7 ADC Speeds

The dsPIC30F 12-bit ADC specifications permit a maximum of 200 ksps sampling rate. The table below summarizes the conversion speeds for the dsPIC30F 12-bit ADC and the required operating conditions.

TABLE 19-1: 12-BIT ADC EXTENDED CONVERSION RATES

dsPIC30F 12-bit ADC Conversion Rates						
Speed	**TAD Minimum**	**Sampling Time Min**	**R_S Max**	**VDD**	**Temperature**	**Channels Configuration**
Up to 200 ksps(1)	334 ns	1 TAD	2.5 kΩ	4.5V to 5.5V	-40°C to +85°C	
Up to 100 ksps	668 ns	1 TAD	2.5 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 19-2 for recommended circuit.

TABLE 19-2: A/D CONVERTER REGISTER MAP

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 uuuu uuuu uuuu
ADCBUF1	0282	—	—	—	—	ADC Data Buffer 1												0000 uuuu uuuu uuuu
ADCBUF2	0284	—	—	—	—	ADC Data Buffer 2												0000 uuuu uuuu uuuu
ADCBUF3	0286	—	—	—	—	ADC Data Buffer 3												0000 uuuu uuuu uuuu
ADCBUF4	0288	—	—	—	—	ADC Data Buffer 4												0000 uuuu uuuu uuuu
ADCBUF5	028A	—	—	—	—	ADC Data Buffer 5												0000 uuuu uuuu uuuu
ADCBUF6	028C	—	—	—	—	ADC Data Buffer 6												0000 uuuu uuuu uuuu
ADCBUF7	028E	—	—	—	—	ADC Data Buffer 7												0000 uuuu uuuu uuuu
ADCBUF8	0290	—	—	—	—	ADC Data Buffer 8												0000 uuuu uuuu uuuu
ADCBUF9	0292	—	—	—	—	ADC Data Buffer 9												0000 uuuu uuuu uuuu
ADCBUFA	0294	—	—	—	—	ADC Data Buffer 10												0000 uuuu uuuu uuuu
ADCBUFB	0296	—	—	—	—	ADC Data Buffer 11												0000 uuuu uuuu uuuu
ADCBUFC	0298	—	—	—	—	ADC Data Buffer 12												0000 uuuu uuuu uuuu
ADCBUFD	029A	—	—	—	—	ADC Data Buffer 13												0000 uuuu uuuu uuuu
ADCBUFE	029C	—	—	—	—	ADC Data Buffer 14												0000 uuuu uuuu uuuu
ADCBUFF	029E	—	—	—	—	ADC Data Buffer 15												0000 uuuu uuuu uuuu
ADCON1	02A0	ADON	—	ADSIDL	—	—	—	FORM<1:0>		SSRC<2:0>			—	—	ASAM	SAMP	DONE	0000 0000 0000 0000
ADCON2	02A2	VCFG<2:0>			—	—	CSCNA	—	—	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	—	—	—	CH0NB	CH0SB<3:0>				—	—	—	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

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

TABLE 20-1: OSCILLATOR OPERATING MODES

Oscillator Mode	Description
XTL	200 kHz-4 MHz crystal on OSC1:OSC2.
XT	4 MHz-10 MHz crystal on OSC1:OSC2.
XT w/PLL 4x	4 MHz-10 MHz crystal on OSC1:OSC2, 4x PLL enabled.
XT w/PLL 8x	4 MHz-10 MHz crystal on OSC1:OSC2, 8x PLL enabled.
XT w/PLL 16x	4 MHz-10 MHz crystal on OSC1:OSC2, 16x PLL enabled(1).
LP	32 kHz crystal on SOSCO:SOSCI(2).
HS	10 MHz-25 MHz crystal.
EC	External clock input (0-40 MHz).
ECIO	External clock input (0-40 MHz), OSC2 pin is I/O.
EC w/PLL 4x	External clock input (4-10 MHz), OSC2 pin is I/O, 4x PLL enabled(1).
EC w/PLL 8x	External clock input (4-10 MHz), OSC2 pin is I/O, 8x PLL enabled(1).
EC w/PLL 16x	External clock input (4-7.5 MHz), OSC2 pin is I/O, 16x PLL enabled(1).
ERC	External RC oscillator, OSC2 pin is FOSC/4 output(3).
ERCIO	External RC oscillator, OSC2 pin is I/O(3).
FRC	7.37 MHz internal RC oscillator.
LPRC	512 kHz internal RC oscillator.

Note 1: dsPIC30F maximum operating frequency of 120 MHz must be met.

2: LP oscillator can be conveniently shared as system clock, as well as real-time clock for Timer1.

3: Requires external R and C. Frequency operation up to 4 MHz.

20.3 Reset

The dsPIC30F differentiates between various kinds of Reset:

a) Power-on Reset (POR)
b) $\overline{\text{MCLR}}$ Reset during normal operation
c) $\overline{\text{MCLR}}$ Reset during Sleep
d) Watchdog Timer (WDT) Reset (during normal operation)
e) Programmable Brown-out Reset (BOR)
f) `RESET` Instruction
g) Reset caused by trap lockup (TRAPR)
h) Reset caused by illegal opcode or by using an uninitialized W register as an address pointer (IOPUWR)

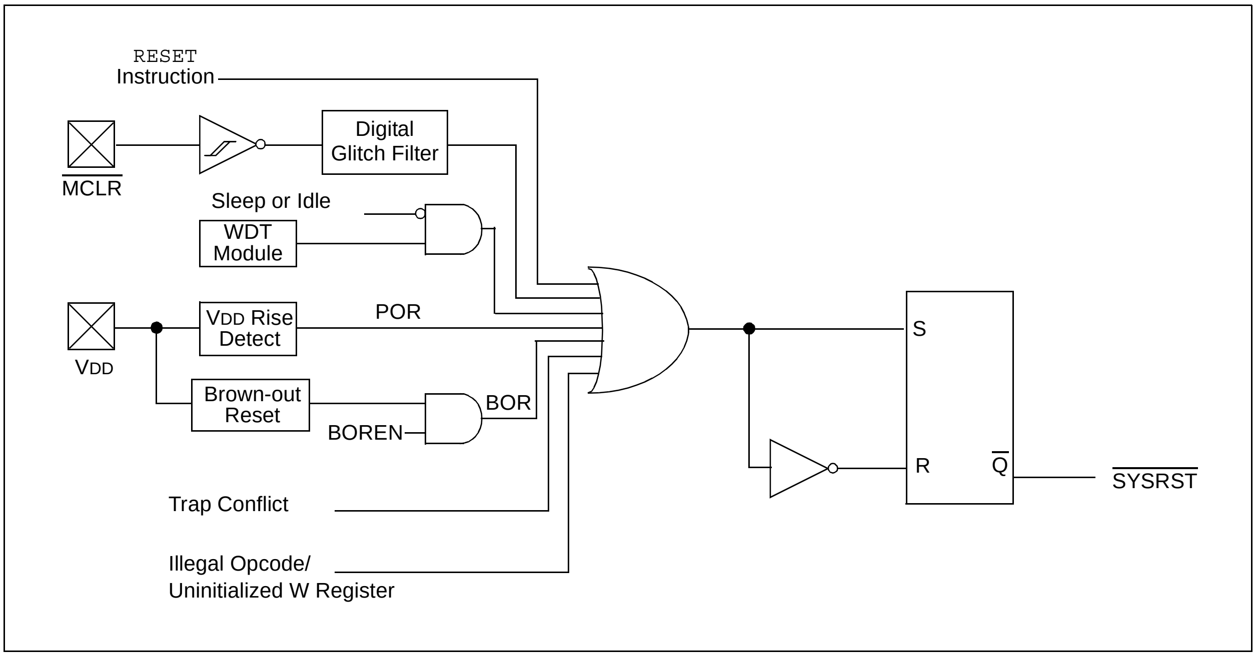
Different registers are affected in different ways by various Reset conditions. Most registers are not affected by a WDT wake-up since this is viewed as the resumption of normal operation. Status bits from the RCON register are set or cleared differently in different Reset situations, as indicated in Table 20-5. These bits are used in software to determine the nature of the Reset.

A block diagram of the On-Chip Reset Circuit is shown in Figure 20-2.

A $\overline{\text{MCLR}}$ noise filter is provided in the $\overline{\text{MCLR}}$ Reset path. The filter detects and ignores small pulses.

Internally generated Resets do not drive $\overline{\text{MCLR}}$ pin low.

FIGURE 20-2: RESET SYSTEM BLOCK DIAGRAM



20.3.1 POR: POWER-ON RESET

A power-on event will generate an internal POR pulse when a V_{DD} rise is detected. The Reset pulse will occur at the POR circuit threshold voltage (V_{POR}) which is nominally 1.85V. The device supply voltage characteristics must meet specified starting voltage and rise rate requirements. The POR pulse will reset a POR timer and place the device in the Reset state. The POR also selects the device clock source identified by the oscillator configuration fuses.

The POR circuit inserts a small delay, T_{POR}, which is nominally 10 µs and ensures that the device bias circuits are stable. Furthermore, a user selected power-up time-out (T_{PWRT}) is applied. The T_{PWRT} parameter is based on device Configuration bits and can be 0 ms (no delay), 4 ms, 16 ms, or 64 ms. The total delay is at device power-up, $T_{POR} + T_{PWRT}$. When these delays have expired, $\overline{\text{SYSRST}}$ will be negated on the next leading edge of the Q1 clock and the PC will jump to the Reset vector.

The timing for the $\overline{\text{SYSRST}}$ signal is shown in Figure 20-3 through Figure 20-5.

TABLE 21-2: INSTRUCTION SET OVERVIEW (CONTINUED)

Base Instr #	Assembly Mnemonic	Assembly Syntax	Description	# of Words	# of Cycles	Status Flags Affected
9	BTG	BTG f,#bit4	Bit Toggle f	1	1	None
		BTG Ws,#bit4	Bit Toggle Ws	1	1	None
10	BTSC	BTSC f,#bit4	Bit Test f, Skip if Clear	1	1 (2 or 3)	None
		BTSC Ws,#bit4	Bit Test Ws, Skip if Clear	1	1 (2 or 3)	None
11	BTSS	BTSS f,#bit4	Bit Test f, Skip if Set	1	1 (2 or 3)	None
		BTSS Ws,#bit4	Bit Test Ws, Skip if Set	1	1 (2 or 3)	None
12	BTST	BTST f,#bit4	Bit Test f	1	1	Z
		BTST.C Ws,#bit4	Bit Test Ws to C	1	1	C
		BTST.Z Ws,#bit4	Bit Test Ws to Z	1	1	Z
		BTST.C Ws,Wb	Bit Test Ws<Wb> to C	1	1	C
		BTST.Z Ws,Wb	Bit Test Ws<Wb> to Z	1	1	Z
13	BTSTS	BTSTS f,#bit4	Bit Test then Set f	1	1	Z
		BTSTS.C Ws,#bit4	Bit Test Ws to C, then Set	1	1	C
		BTSTS.Z Ws,#bit4	Bit Test Ws to Z, then Set	1	1	Z
14	CALL	CALL lit23	Call subroutine	2	2	None
		CALL Wn	Call indirect subroutine	1	2	None
15	CLR	CLR f	f = 0x0000	1	1	None
		CLR WREG	WREG = 0x0000	1	1	None
		CLR Ws	Ws = 0x0000	1	1	None
		CLR Acc,Wx,Wxd,Wy,Wyd,AWB	Clear Accumulator	1	1	OA,OB,SA,SB
16	CLRWDT	CLRWDT	Clear Watchdog Timer	1	1	WDTO,Sleep
17	COM	COM f	f = $\overline{f}$	1	1	N,Z
		COM f,WREG	WREG = $\overline{f}$	1	1	N,Z
		COM Ws,Wd	Wd = $\overline{Ws}$	1	1	N,Z
18	CP	CP f	Compare f with WREG	1	1	C,DC,N,OV,Z
		CP Wb,#lit5	Compare Wb with lit5	1	1	C,DC,N,OV,Z
		CP Wb,Ws	Compare Wb with Ws (Wb - Ws)	1	1	C,DC,N,OV,Z
19	CP0	CP0 f	Compare f with 0x0000	1	1	C,DC,N,OV,Z
		CP0 Ws	Compare Ws with 0x0000	1	1	C,DC,N,OV,Z
20	CPB	CPB f	Compare f with WREG, with Borrow	1	1	C,DC,N,OV,Z
		CPB Wb,#lit5	Compare Wb with lit5, with Borrow	1	1	C,DC,N,OV,Z
		CPB Wb,Ws	Compare Wb with Ws, with Borrow (Wb - Ws - $\overline{C}$)	1	1	C,DC,N,OV,Z
21	CPSEQ	CPSEQ Wb, Wn	Compare Wb with Wn, skip if =	1	1 (2 or 3)	None
22	CPSGT	CPSGT Wb, Wn	Compare Wb with Wn, skip if >	1	1 (2 or 3)	None
23	CPSLT	CPSLT Wb, Wn	Compare Wb with Wn, skip if <	1	1 (2 or 3)	None
24	CPSNE	CPSNE Wb, Wn	Compare Wb with Wn, skip if ≠	1	1 (2 or 3)	None
25	DAW	DAW Wn	Wn = decimal adjust Wn	1	1	C
26	DEC	DEC f	f = f -1	1	1	C,DC,N,OV,Z
		DEC f,WREG	WREG = f -1	1	1	C,DC,N,OV,Z
		DEC Ws,Wd	Wd = Ws - 1	1	1	C,DC,N,OV,Z
27	DEC2	DEC2 f	f = f -2	1	1	C,DC,N,OV,Z
		DEC2 f,WREG	WREG = f -2	1	1	C,DC,N,OV,Z
		DEC2 Ws,Wd	Wd = Ws - 2	1	1	C,DC,N,OV,Z
28	DISI	DISI #lit14	Disable Interrupts for k instruction cycles	1	1	None

TABLE 23-11: ELECTRICAL CHARACTERISTICS: BOR

DC 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(1)**	**Max**	**Units**	**Conditions**
BO10	VBOR	BOR Voltage(2) on VDD transition high to low	BORV = 11(3)	—	—	—	V	Not in operating range
			BORV = 10	2.6	—	2.71	V	
			BORV = 01	4.1	—	4.4	V	
			BORV = 00	4.58	—	4.73	V	
BO15	VBHYS			—	5	—	mV	

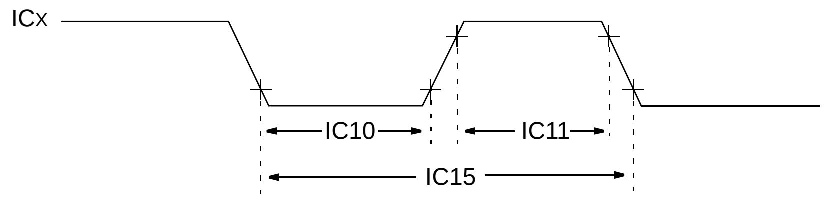
Note 1: Data in "Typ" column is at 5V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.
2: These parameters are characterized but not tested in manufacturing.
3: '11' values not in usable operating range.

TABLE 23-12: DC CHARACTERISTICS: PROGRAM AND EEPROM

DC 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(1)**	**Max**	**Units**	**Conditions**
		Data EEPROM Memory(2)					
D120	ED	Byte Endurance	100K	1M	—	E/W	-40°C ≤ TA ≤ +85°C
D121	VDRW	VDD for Read/Write	VMIN	—	5.5	V	Using EECON to read/write VMIN = Minimum operating voltage
D122	TDEW	Erase/Write Cycle Time	—	2	—	ms	
D123	TRETD	Characteristic Retention	40	100	—	Year	Provided no other specifications are violated
D124	IDEW	IDD During Programming	—	10	30	mA	Row Erase
		Program FLASH Memory(2)					
D130	EP	Cell Endurance	10K	100K	—	E/W	-40°C ≤ TA ≤ +85°C
D131	VPR	VDD for Read	VMIN	—	5.5	V	VMIN = Minimum operating voltage
D132	VEB	VDD for Bulk Erase	4.5	—	5.5	V	
D133	VPEW	VDD for Erase/Write	3.0	—	5.5	V	
D134	TPEW	Erase/Write Cycle Time	—	2	—	ms	
D135	TRETD	Characteristic Retention	40	100	—	Year	Provided no other specifications are violated
D136	TEB	ICSP Block Erase Time	—	4	—	ms	
D137	IPEW	IDD During Programming	—	10	30	mA	Row Erase
D138	IEB	IDD During Programming	—	10	30	mA	Bulk Erase

Note 1: Data in "Typ" column is at 5V, 25°C unless otherwise stated.
2: These parameters are characterized but not tested in manufacturing.

FIGURE 23-9: INPUT CAPTURE (CAPx) TIMING CHARACTERISTICS



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

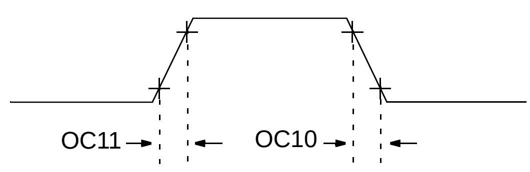
TABLE 23-26: INPUT CAPTURE TIMING REQUIREMENTS

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**	**Max**	**Units**	**Conditions**
IC10	TccL	ICx Input Low Time	No Prescaler	0.5 TCY + 20	—	ns	
			With Prescaler	10	—	ns	
IC11	TccH	ICx Input High Time	No Prescaler	0.5 TCY + 20	—	ns	
			With Prescaler	10	—	ns	
IC15	TccP	ICx Input Period		(2 TCY + 40)/N	—	ns	N = prescale value (1, 4, 16)

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

FIGURE 23-10: OUTPUT COMPARE MODULE (OCx) TIMING CHARACTERISTICS

OCx
(Output Compare
or PWM Mode)



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

TABLE 23-27: OUTPUT COMPARE MODULE 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**
OC10	TccF	OCx Output Fall Time	—	—	—	ns	See Parameter D032
OC11	TccR	OCx Output Rise Time	—	—	—	ns	See Parameter D031

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.

TABLE 23-29: DCI MODULE (MULTICHANNEL, I²S MODES) 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
CS10	TCSCKL	CSCK Input Low Time (CSCK pin is an input)	TCY/2 + 20	—	—	ns	—
		CSCK Output Low Time(3) (CSCK pin is an output)	30	—	—	ns	—
CS11	TCSCKH	CSCK Input High Time (CSCK pin is an input)	TCY/2 + 20	—	—	ns	—
		CSCK Output High Time(3) (CSCK pin is an output)	30	—	—	ns	—
CS20	TCSCKF	CSCK Output Fall Time(4) (CSCK pin is an output)	—	10	25	ns	—
CS21	TCSCKR	CSCK Output Rise Time(4) (CSCK pin is an output)	—	10	25	ns	—
CS30	TCSDOF	CSDO Data Output Fall Time(4)	—	10	25	ns	—
CS31	TCSDOR	CSDO Data Output Rise Time(4)	—	10	25	ns	—
CS35	TDV	Clock edge to CSDO data valid	—	—	10	ns	—
CS36	TDIV	Clock edge to CSDO tri-stated	10	—	20	ns	—
CS40	TCSDI	Setup time of CSDI data input to CSCK edge (CSCK pin is input or output)	20	—	—	ns	—
CS41	THCSDI	Hold time of CSDI data input to CSCK edge (CSCK pin is input or output)	20	—	—	ns	—
CS50	TCOFSF(1)	COFS Fall Time (COFS pin is output)	—	10	25	ns	—
CS51	TCOFSR(1)	COFS Rise Time (COFS pin is output)	—	10	25	ns	—
CS55	TSCOFS	Setup time of COFS data input to CSCK edge (COFS pin is input)	20	—	—	ns	—
CS56	THCOFS	Hold time of COFS data input to CSCK edge (COFS pin is input)	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 CSCK is 100 ns. Therefore, the clock generated in Master mode must not violate this specification.
4: Assumes 50 pF load on all DCI pins.