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	CANbus, I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, Motor Control PWM, QEI, POR, PWM, WDT
Number of I/O	68
Program Memory Size	144KB (48K x 24)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	8K 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 (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/dspic30f6010a-30i-pf

1.0 DEVICE 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).

This document contains device-specific information for the dsPIC30F6010A and dsPIC30F6015 devices. The dsPIC30F devices contain extensive Digital Signal Processor (DSP) functionality within a high-performance 16-bit microcontroller (MCU) architecture. Figure 1-1 shows a device block diagram for the dsPIC30F6010A device. Figure 1-2 shows a device block diagram for the dsPIC30F6015 device.

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 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 chapter summarizes the CPU and peripheral functions of the dsPIC30F6010A/6015.

2.1 Core Overview

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

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

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

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

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

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

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

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

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

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

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

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

All byte loads into any W register are loaded into the LSB. The MSB is not modified.

A sign-extend (SE) instruction is provided to allow users to translate 8-bit signed data to 16-bit signed values. Alternatively, for 16-bit unsigned data, users can clear the MSB of any W register by executing a zero-extend (ZE) instruction on the appropriate address.

Although most instructions are capable of operating on word or byte data sizes, it should be noted that some instructions, including the DSP instructions, operate only on words.

3.2.5 NEAR DATA SPACE

An 8 Kbyte 'near' data space is reserved in X address memory space between 0x0000 and 0x1FFF, which is directly addressable via a 13-bit absolute address field within all memory direct instructions. The remaining X address space and all of the Y address space is addressable indirectly. Additionally, the whole of X data space is addressable using MOV instructions, which support memory direct addressing with a 16-bit address field.

3.2.6 SOFTWARE STACK

The dsPIC DSC device contains a software stack. W15 is used as the Stack Pointer.

The Stack Pointer always points to the first available free word and grows from lower addresses towards higher addresses. It pre-decrements for stack pops and post-increments for stack pushes, as shown in Figure 3-9. Note that for a PC push during any CALL instruction, the MSB of the PC is zero-extended before the push, ensuring that the MSB is always clear.

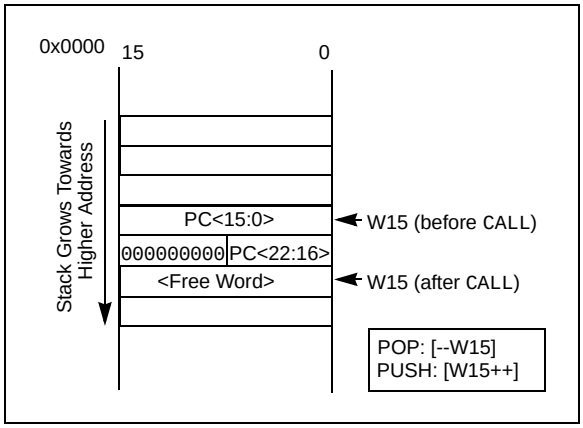
Note:	A PC push during exception processing will concatenate the SRL register to the MSB of the PC prior to the push.

There is a Stack Pointer Limit register (SPLIM) associated with the Stack Pointer. SPLIM is uninitialized at Reset. As is the case for the Stack Pointer, SPLIM<0> is forced to '0', because all stack operations must be word-aligned. Whenever an Effective Address (EA) is generated using W15 as a source or destination pointer, the address thus generated is compared with the value in SPLIM. If the contents of the Stack Pointer (W15) and the SPLIM register are equal and a push operation is performed, a stack error trap will not occur. The stack error trap will occur on a subsequent push operation. Thus, for example, if it is desirable to cause a stack error trap when the stack grows beyond address 0x2000 in RAM, initialize the SPLIM with the value, 0x1FFE.

Similarly, a Stack Pointer underflow (stack error) trap is generated when the Stack Pointer address is found to be less than 0x0800, thus preventing the stack from interfering with the Special Function Register (SFR) space.

A write to the SPLIM register should not be immediately followed by an indirect read operation using W15.

FIGURE 3-9: CALL STACK FRAME



3.2.7 DATA RAM PROTECTION FEATURE

The dsPIC30F6010A/6015 devices support Data RAM protection features which enable segments of RAM to be protected when used in conjunction with Boot and Secure Code Segment Security. BSRAM (Secure RAM segment for BS) is accessible only from the Boot Segment Flash code when enabled. SSRAM (Secure RAM segment for RAM) is accessible only from the Secure Segment Flash code when enabled.

See Table 3-3 for an overview of the BSRAM and SSRAM SFRs.

4.2.3 MODULO ADDRESSING APPLICABILITY

Modulo Addressing can be applied to the Effective Address (EA) 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 remains 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.

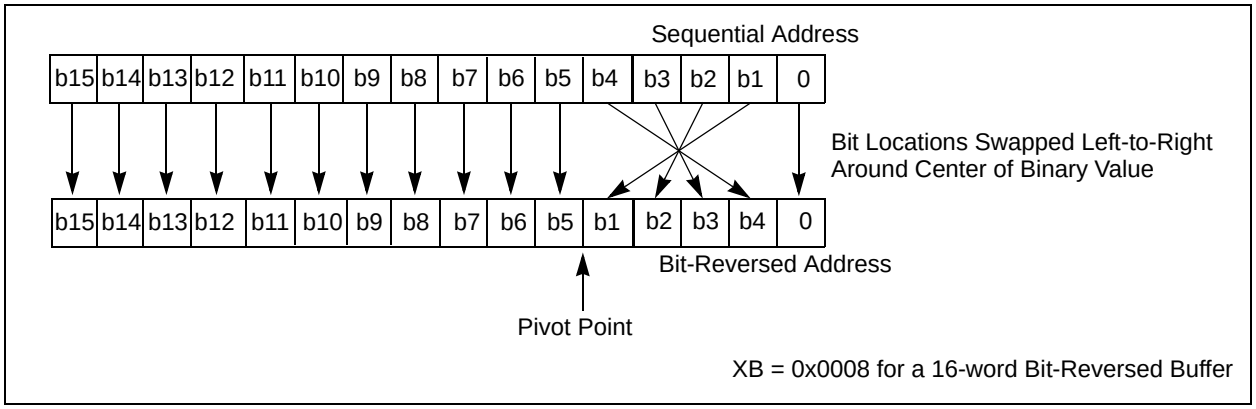
Note:	All Bit-Reversed EA calculations assume word-sized data (LSb of every EA is always clear). The XB value is scaled accordingly to generate compatible (byte) addresses.

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.

FIGURE 4-2: BIT-REVERSED ADDRESS EXAMPLE



5.2 Reset Sequence

A Reset is not a true exception because the interrupt controller is not involved in the Reset process. The processor initializes its registers in response to a Reset which forces the PC to zero. The processor then begins program execution at location 0x000000. A GOTO instruction is stored in the first program memory location, immediately followed by the address target for the GOTO instruction. The processor executes the GOTO to the specified address and then begins operation at the specified target (start) address.

5.2.1 RESET SOURCES

There are 6 sources of error which will cause a device Reset.

- Watchdog Time-out:
  The watchdog has timed out, indicating that the processor is no longer executing the correct flow of code.
- Uninitialized W Register Trap:
  An attempt to use an uninitialized W register as an Address Pointer will cause a Reset.
- Illegal Instruction Trap:
  Attempted execution of any unused opcodes will result in an illegal instruction trap. Note that a fetch of an illegal instruction does not result in an illegal instruction trap if that instruction is flushed prior to execution due to a flow change.
- Brown-out Reset (BOR):
  A momentary dip in the power supply to the device has been detected which may result in malfunction.
- Trap Lockout:
  Occurrence of multiple trap conditions simultaneously will cause a Reset.

5.3 Traps

Traps can be considered as non-maskable interrupts indicating a software or hardware error, which adhere to a predefined priority, as shown in Figure 5-1. They are intended to provide the user a means to correct erroneous operation during debug and when operating within the application.

Note:	If the user does not intend to take corrective action in the event of a trap error condition, these vectors must be loaded with the address of a default handler that simply contains the RESET instruction. If, on the other hand, one of the vectors containing an invalid address is called, an address error trap is generated.

Note that many of these trap conditions can only be detected when they occur. Consequently, the questionable instruction is allowed to complete prior to trap exception processing. If the user chooses to recover from the error, the result of the erroneous action that caused the trap may have to be corrected.

There are 8 fixed priority levels for traps: Level 8 through Level 15, which means that IPL3 is always set during processing of a trap.

If the user is not currently executing a trap, and he sets the IPL<3:0> bits to a value of '0111' (Level 7), then all interrupts are disabled, but traps can still be processed.

5.3.1 TRAP SOURCES

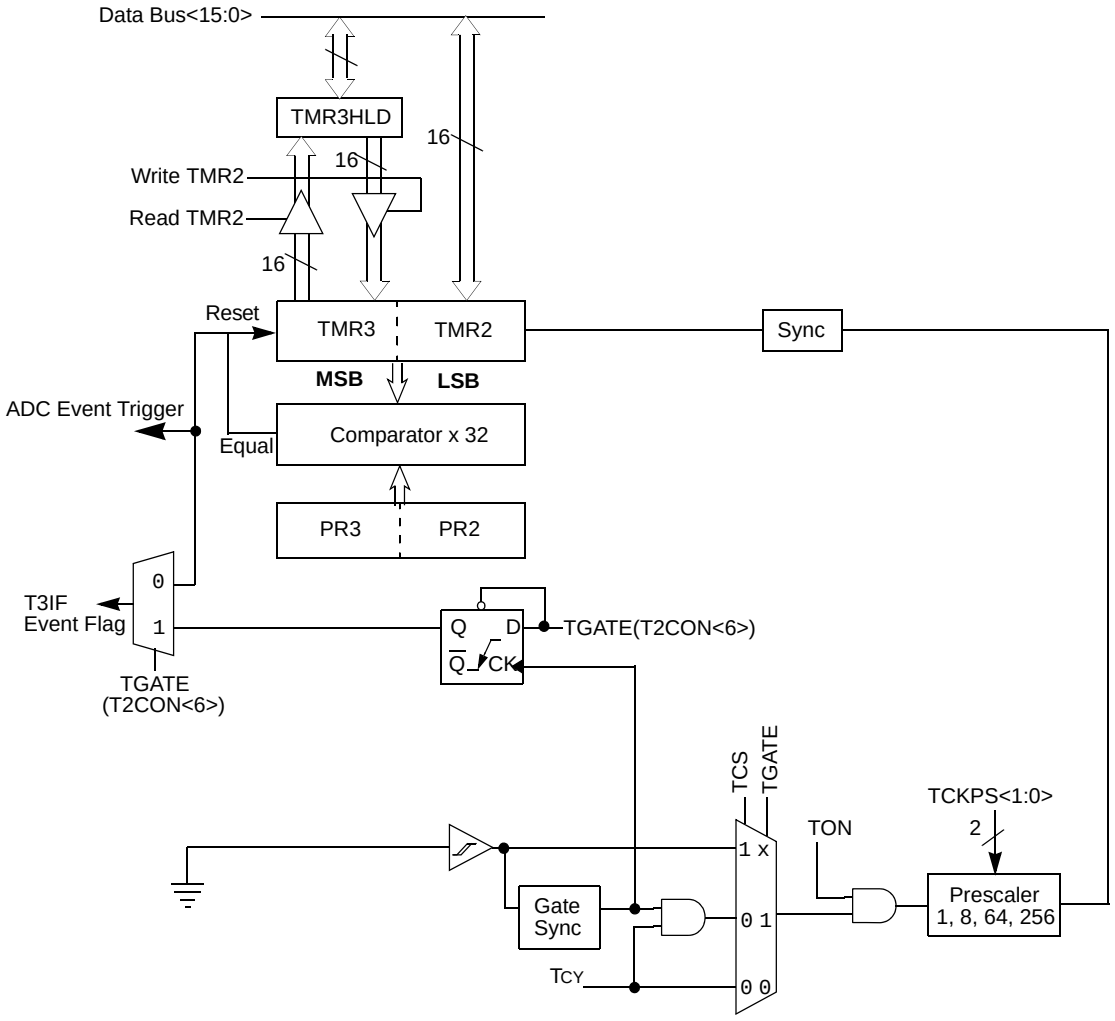
The following traps are provided with increasing priority. However, since all traps can be nested, priority has little effect.

Math Error Trap:

The math error trap executes under the following four circumstances:

1. Should an attempt be made to divide by zero, the divide operation will be aborted on a cycle boundary and the trap taken.
2. If enabled, a math error trap will be taken when an arithmetic operation on either Accumulator A or B causes an overflow from bit 31 and the Accumulator Guard bits are not utilized.
3. If enabled, a math error trap will be taken when an arithmetic operation on either Accumulator A or B causes a catastrophic overflow from bit 39 and all saturation is disabled.
4. If the shift amount specified in a shift instruction is greater than the maximum allowed shift amount, a trap will occur.

FIGURE 10-2: 32-BIT TIMER2/3 BLOCK DIAGRAM FOR dsPIC30F6015



Note: Timer Configuration bit T32, T2CON(<3>) must be set to '1' for a 32-bit timer/counter operation. All control bits are respective to the T2CON register.

TABLE 12-1: INPUT CAPTURE 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
IC1BUF	0140	Input 1 Capture Register																uuuu uuuu uuuu uuuu
IC1CON	0142	—	—	ICSIDL	—	—	—	—	—	ICTMR	ICI<1:0>		ICOV	ICBNE	ICM<2:0>			0000 0000 0000 0000
IC2BUF	0144	Input 2 Capture Register																uuuu uuuu uuuu uuuu
IC2CON	0146	—	—	ICSIDL	—	—	—	—	—	ICTMR	ICI<1:0>		ICOV	ICBNE	ICM<2:0>			0000 0000 0000 0000
IC3BUF	0148	Input 3 Capture Register																uuuu uuuu uuuu uuuu
IC3CON	014A	—	—	ICSIDL	—	—	—	—	—	ICTMR	ICI<1:0>		ICOV	ICBNE	ICM<2:0>			0000 0000 0000 0000
IC4BUF	014C	Input 4 Capture Register																uuuu uuuu uuuu uuuu
IC4CON	014E	—	—	ICSIDL	—	—	—	—	—	ICTMR	ICI<1:0>		ICOV	ICBNE	ICM<2:0>			0000 0000 0000 0000
IC5BUF	0150	Input 5 Capture Register																uuuu uuuu uuuu uuuu
IC5CON	0152	—	—	ICSIDL	—	—	—	—	—	ICTMR	ICI<1:0>		ICOV	ICBNE	ICM<2:0>			0000 0000 0000 0000
IC6BUF	0154	Input 6 Capture Register																uuuu uuuu uuuu uuuu
IC6CON	0156	—	—	ICSIDL	—	—	—	—	—	ICTMR	ICI<1:0>		ICOV	ICBNE	ICM<2:0>			0000 0000 0000 0000
IC7BUF	0158	Input 7 Capture Register																uuuu uuuu uuuu uuuu
IC7CON	015A	—	—	ICSIDL	—	—	—	—	—	ICTMR	ICI<1:0>		ICOV	ICBNE	ICM<2:0>			0000 0000 0000 0000
IC8BUF	015C	Input 8 Capture Register																uuuu uuuu uuuu uuuu
IC8CON	015E	—	—	ICSIDL	—	—	—	—	—	ICTMR	ICI<1:0>		ICOV	ICBNE	ICM<2:0>			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.

NOTES:

19.0 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).

19.1 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 dsPIC30F6010A has two CAN modules. The dsPIC30F6015 has only one.

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, CAN2.0B 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 timer module 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.

19.2 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 a 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.

19.6.2 PRESCALER SETTING

There is a programmable prescaler, with integral values ranging from 1 to 64, in addition to a fixed divide-by-2 for clock generation. The Time Quantum (TQ) is a fixed unit of time derived from the oscillator period, and is given by Equation 19-1, where FCAN is FCY (if the CANCKS bit is set or 4 FCY (if CANCKS is cleared).

Note:	FCAN must not exceed 30 MHz. If CANCKS = 0, then FCY must not exceed 7.5 MHz.

EQUATION 19-1: TIME QUANTUM FOR CLOCK GENERATION

$$T_Q = 2 (BRP<5:0> + 1)/F_{CAN}$$

19.6.3 PROPAGATION SEGMENT

This part of the bit time is used to compensate physical delay times within the network. These delay times consist of the signal propagation time on the bus line and the internal delay time of the nodes. The Propagation Segment can be programmed from 1 TQ to 8 TQ by setting the PRSEG<2:0> bits (CiCFG2<2:0>).

19.6.4 PHASE SEGMENTS

The phase segments are used to optimally locate the sampling of the received bit within the transmitted bit time. The sampling point is between Phase1 Seg and Phase2 Seg. These segments are lengthened or shortened by re-synchronization. The end of the Phase1 Seg determines the sampling point within a bit period. The segment is programmable from 1 TQ to 8 TQ. Phase2 Seg provides delay to the next transmitted data transition. The segment is programmable from 1 TQ to 8 TQ, or it may be defined to be equal to the greater of Phase1 Seg or the Information Processing Time (2 TQ). The Phase1 Seg is initialized by setting bits SEG1PH<2:0> (CiCFG2<5:3>), and Phase2 Seg is initialized by setting SEG2PH<2:0> (CiCFG2<10:8>).

The following requirement must be fulfilled while setting the lengths of the Phase Segments:

- Propagation Segment + Phase1 Seg > = Phase2 Seg

19.6.5 SAMPLE POINT

The sample point is the point of time at which the bus level is read and interpreted as the value of that respective bit. The location is at the end of Phase1 Seg. If the bit timing is slow and contains many TQ, it is possible to specify multiple sampling of the bus line at the sample point. The level determined by the CAN bus then corresponds to the result from the majority decision of three values. The majority samples are taken at the sample point and twice before with a distance of TQ/2. The CAN module allows the user to chose between sampling three times at the same point or once at the same point, by setting or clearing the SAM bit (CiCFG2<6>).

Typically, the sampling of the bit should take place at about 60-70% through the bit time, depending on the system parameters.

19.6.6 SYNCHRONIZATION

To compensate for phase shifts between the oscillator frequencies of the different bus stations, each CAN controller must be able to synchronize to the relevant signal edge of the incoming signal. When an edge in the transmitted data is detected, the logic will compare the location of the edge to the expected time (Synchronous Segment). The circuit will then adjust the values of Phase1 Seg and Phase2 Seg. There are 2 mechanisms used to synchronize.

19.6.6.1 Hard Synchronization

Hard synchronization is only done whenever there is a recessive to dominant edge during bus Idle, indicating the start of a message. After hard synchronization, the bit time counters are restarted with the Synchronous Segment. Hard synchronization forces the edge which has caused the hard synchronization to lie within the synchronization segment of the restarted bit time. If a hard synchronization is done, there will not be a resynchronization within that bit time.

19.6.6.2 Re-synchronization

As a result of re-synchronization, Phase1 Seg may be lengthened or Phase2 Seg may be shortened. The amount of lengthening or shortening of the phase buffer segment has an upper bound known as the synchronization jump width, and is specified by the SJW<1:0> bits (CiCFG1<7:6>). The value of the synchronization jump width will be added to Phase1 Seg or subtracted from Phase2 Seg. The re-synchronization jump width is programmable between 1 TQ and 4 TQ.

The following requirement must be fulfilled while setting the SJW<1:0> bits:

- Phase2 Seg > Synchronization Jump Width

TABLE 19-2: CAN2 REGISTER MAP FOR dsPIC30F6010A(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
C2RXF0SID	03C0	—	—	—	Receive Acceptance Filter 0 Standard Identifier<10:0>											—	EXIDE	000u uuuu uuuu uu0u
C2RXF0EIDH	03C2	—	—	—	—	Receive Acceptance Filter 0 Extended Identifier<17:6>												0000 uuuu uuuu uuuu
C2RXF0EIDL	03C4	Receive Acceptance Filter 0 Extended Identifier<5:0>						—	—	—	—	—	—	—	—	—	—	uuuu uu00 0000 0000
C2RXF1SID	03C8	—	—	—	Receive Acceptance Filter 1 Standard Identifier<10:0>											—	EXIDE	000u uuuu uuuu uu0u
C2RXF1EIDH	03CA	—	—	—	—	Receive Acceptance Filter 1 Extended Identifier<17:6>												0000 uuuu uuuu uuuu
C2RXF1EIDL	03CC	Receive Acceptance Filter 1 Extended Identifier<5:0>						—	—	—	—	—	—	—	—	—	—	uuuu uu00 0000 0000
C2RXF2SID	03D0	—	—	—	Receive Acceptance Filter 2 Standard Identifier<10:0>											—	EXIDE	000u uuuu uuuu uu0u
C2RXF2EIDH	03D2	—	—	—	—	Receive Acceptance Filter 2 Extended Identifier<17:6>												0000 uuuu uuuu uuuu
C2RXF2EIDL	03D4	Receive Acceptance Filter 2 Extended Identifier<5:0>						—	—	—	—	—	—	—	—	—	—	uuuu uu00 0000 0000
C2RXF3SID	03D8	—	—	—	Receive Acceptance Filter 3 Standard Identifier<10:0>											—	EXIDE	000u uuuu uuuu uu0u
C2RXF3EIDH	03DA	—	—	—	—	Receive Acceptance Filter 3 Extended Identifier<17:6>												0000 uuuu uuuu uuuu
C2RXF3EIDL	03DC	Receive Acceptance Filter 3 Extended Identifier<5:0>						—	—	—	—	—	—	—	—	—	—	uuuu uu00 0000 0000
C2RXF4SID	03E0	—	—	—	Receive Acceptance Filter 4 Standard Identifier<10:0>											—	EXIDE	000u uuuu uuuu uu0u
C2RXF4EIDH	03E2	—	—	—	—	Receive Acceptance Filter 4 Extended Identifier<17:6>												0000 uuuu uuuu uuuu
C2RXF4EIDL	03E4	Receive Acceptance Filter 4 Extended Identifier<5:0>						—	—	—	—	—	—	—	—	—	—	uuuu uu00 0000 0000
C2RXF5SID	03E8	—	—	—	Receive Acceptance Filter 5 Standard Identifier <10:0>											—	EXIDE	000u uuuu uuuu uu0u
C2RXF5EIDH	03EA	—	—	—	—	Receive Acceptance Filter 5 Extended Identifier<17:6>												0000 uuuu uuuu uuuu
C2RXF5EIDL	03EC	Receive Acceptance Filter 5 Extended Identifier<5:0>						—	—	—	—	—	—	—	—	—	—	uuuu uu00 0000 0000
C2RXM0SID	03F0	—	—	—	Receive Acceptance Mask 0 Standard Identifier<10:0>											—	MIDE	000u uuuu uuuu uu0u
C2RXM0EIDH	03F2	—	—	—	—	Receive Acceptance Mask 0 Extended Identifier<17:6>												0000 uuuu uuuu uuuu
C2RXM0EIDL	03F4	Receive Acceptance Mask 0 Extended Identifier<5:0>						—	—	—	—	—	—	—	—	—	—	uuuu uu00 0000 0000
C2RXM1SID	03F8	—	—	—	Receive Acceptance Mask 1 Standard Identifier<10:0>											—	MIDE	000u uuuu uuuu uu0u
C2RXM1EIDH	03FA	—	—	—	—	Receive Acceptance Mask 1 Extended Identifier<17:6>												0000 uuuu uuuu uuuu
C2RXM1EIDL	03FC	Receive Acceptance Mask 1 Extended Identifier<5:0>						—	—	—	—	—	—	—	—	—	—	uuuu uu00 0000 0000
C2TX2SID	0400	Transmit Buffer 2 Standard Identifier<10:6>					—	—	—	Transmit Buffer 2 Standard Identifier<5:0>						SRR	TXIDE	uuuu u000 uuuu uuuu
C2TX2EID	0402	Transmit Buffer 2 Extended Identifier<17:14>				—	—	—	—	Transmit Buffer 2 Extended Identifier<13:6>								uuuu 0000 uuuu uuuu
C2TX2DLC	0404	Transmit Buffer 2 Extended Identifier<5:0>						TXRTR	TXRB1	TXRB0	DLC<3:0>				—	—	—	uuuu uuuu uuuu u000
C2TX2B1	0406	Transmit Buffer 2 Byte 1								Transmit Buffer 2 Byte 0								uuuu uuuu uuuu uuuu
C2TX2B2	0408	Transmit Buffer 2 Byte 3								Transmit Buffer 2 Byte 2								uuuu uuuu uuuu uuuu
C2TX2B3	040A	Transmit Buffer 2 Byte 5								Transmit Buffer 2 Byte 4								uuuu uuuu uuuu uuuu
C2TX2B4	040C	Transmit Buffer 2 Byte 7								Transmit Buffer 2 Byte 6								uuuu uuuu uuuu uuuu
C2TX2CON	040E	—	—	—	—	—	—	—	—	—	TXABT	TXLARB	TXERR	TXREQ	—	TXPRI<1:0>		0000 0000 0000 0000
C2TX1SID	0410	Transmit Buffer 1 Standard Identifier<10:6>					—	—	—	Transmit Buffer 1 Standard Identifier<5:0>						SRR	TXIDE	uuuu u000 uuuu uuuu
C2TX1EID	0412	Transmit Buffer 1 Extended Identifier<17:14>				—	—	—	—	Transmit Buffer 1 Extended Identifier<13:6>								uuuu 0000 uuuu uuuu
C2TX1DLC	0414	Transmit Buffer 1 Extended Identifier<5:0>						TXRTR	TXRB1	TXRB0	DLC<3:0>				—	—	—	uuuu uuuu uuuu u000

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.

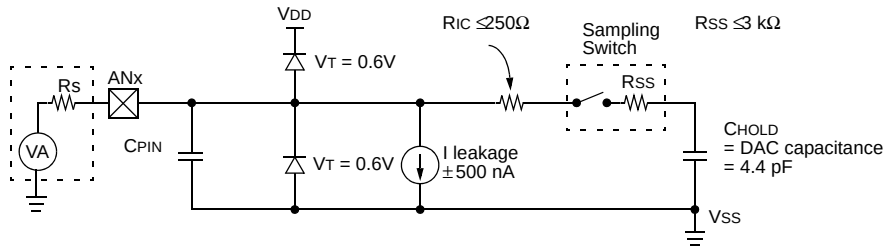
20.8 A/D Acquisition Requirements

The analog input model of the 10-bit A/D converter is shown in Figure 20-3. The total sampling time for the A/D is a function of the internal amplifier settling time, device VDD and the holding capacitor charge time.

For the A/D converter to meet its specified accuracy, the charge holding capacitor (CHOLD) must be allowed to fully charge to the voltage level on the analog input pin. The analog output source impedance (RS), the interconnect impedance (RIC), and the internal sampling switch (RSS) impedance combine to directly affect the time required to charge the capacitor CHOLD. The combined impedance must therefore be small enough to fully charge the holding capacitor within the chosen sample time. To minimize the effects of pin leakage currents on the accuracy of the A/D converter, the maximum recommended source impedance, RS, is 5 kΩ for conversion rates up to 500 ksps and a maximum of 500Ω for conversion rates up to 1 Msps. After the analog input channel is selected (changed), this sampling function must be completed prior to starting the conversion. The internal holding capacitor will be in a discharged state prior to each sample operation.

The user must allow at least 1 TAD period of sampling time, TSAMP, between conversions to allow each sample to be acquired. This sample time may be controlled manually in software by setting/clearing the SAMP bit, or it may be automatically controlled by the A/D converter. In an automatic configuration, the user must allow enough time between conversion triggers so that the minimum sample time can be satisfied. Refer to **Section 24.0 "Electrical Characteristics"** for TAD and sample time requirements.

FIGURE 20-3: A/D CONVERTER ANALOG INPUT MODEL



Legend:	CPIN	= input capacitance
	VT	= threshold voltage
	I leakage	= leakage current at the pin due to various junctions
	RIC	= interconnect resistance
	RSS	= sampling switch resistance
	CHOLD	= sample/hold capacitance (from DAC)

Note: CPIN value depends on device package and is not tested. Effect of CPIN negligible if Rs ≤5 kΩ

20.9 Module Power-Down Modes

The module has 3 internal power modes. When the ADON bit is '1', the module is in Active mode; it is fully powered and functional. When ADON is '0', the module is in Off mode. The digital and analog portions of the circuit are disabled for maximum current savings. In order to return to the Active mode from Off mode, the user must wait for the ADC circuitry to stabilize.

20.10 A/D Operation During CPU Sleep and Idle Modes

20.10.1 A/D OPERATION DURING CPU SLEEP MODE

When the device enters Sleep mode, all clock sources to the module are shutdown and stay at logic '0'.

If Sleep occurs in the middle of a conversion, the conversion is aborted. The converter will not continue with a partially completed conversion on exit from Sleep mode.

Register contents are not affected by the device entering or leaving Sleep mode.

The A/D module can operate during Sleep mode if the A/D clock source is set to RC (ADRC = 1). When the RC clock source is selected, the A/D module waits one instruction cycle before starting the conversion. This allows the SLEEP instruction to be executed, which eliminates all digital switching noise from the conversion. When the conversion is complete, the DONE bit will be set and the result loaded into the ADCBUF register.

If the A/D interrupt is enabled, the device will wake-up from Sleep. If the A/D interrupt is not enabled, the A/D module will then be turned off, although the ADON bit will remain set.

20.10.2 A/D OPERATION DURING CPU IDLE MODE

The ADSIDL bit selects if the module will stop on Idle or continue on Idle. If ADSIDL = 0, the module will continue operation on assertion of Idle mode. If ADSIDL = 1, the module will stop on Idle.

20.11 Effects of a Reset

A device Reset forces all registers to their Reset state. This forces the A/D module to be turned off, and any conversion and acquisition sequence is aborted. The values that are in the ADCBUF registers are not modified. The A/D Result register will contain unknown data after a Power-on Reset.

20.12 Output Formats

The A/D result is 10 bits wide. The data buffer RAM is also 10 bits wide. The 10-bit data can be read in one of four different formats. The FORM<1:0> bits select the format. Each of the output formats translates to a 16-bit result on the data bus.

Write data will always be in right justified (integer) format.

FIGURE 20-4: A/D OUTPUT DATA FORMATS

RAM Contents:							d09	d08	d07	d06	d05	d04	d03	d02	d01	d00
Read to Bus:																
Signed Fractional (1.15)	$\overline{d09}$	d08	d07	d06	d05	d04	d03	d02	d01	d00	0	0	0	0	0	0
Fractional (1.15)	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00	0	0	0	0	0	0
Signed Integer	$\overline{d09}$	$\overline{d09}$	$\overline{d09}$	$\overline{d09}$	$\overline{d09}$	$\overline{d09}$	$\overline{d09}$	d08	d07	d06	d05	d04	d03	d02	d01	d00
Integer	0	0	0	0	0	0	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00

21.4 Watchdog Timer (WDT)

21.4.1 WATCHDOG TIMER OPERATION

The primary function of the Watchdog Timer (WDT) is to reset the processor in the event of a software malfunction. The WDT is a free running timer, which runs off an on-chip RC oscillator, requiring no external component. Therefore, the WDT timer will continue to operate even if the main processor clock (e.g., the crystal oscillator) fails.

21.4.2 ENABLING AND DISABLING THE WDT

The Watchdog Timer can be “Enabled” or “Disabled” only through a Configuration bit (FWDTEN) in the Configuration register FWDT.

Setting FWDTEN = 1 enables the Watchdog Timer. The enabling is done when programming the device. By default, after chip-erase, FWDTEN bit = 1. Any device programmer capable of programming dsPIC30F devices allows programming of this and other Configuration bits.

If enabled, the WDT will increment until it overflows or “times out”. A WDT time-out will force a device Reset (except during Sleep). To prevent a WDT time-out, the user must clear the Watchdog Timer using a CLRWDT instruction.

If a WDT times out during Sleep, the device will wake-up. The WDTO bit in the RCON register will be cleared to indicate a wake-up resulting from a WDT time-out.

Setting FWDTEN = 0 allows user software to enable/disable the Watchdog Timer via the SWDTEN (RCON<5>) control bit.

21.5 Power-Saving Modes

There are two power-saving states that can be entered through the execution of a special instruction, PWRSAV.

These are: Sleep and Idle.

The format of the PWRSAV instruction is as follows:

PWRSAV <parameter>, where ‘parameter’ defines Idle or Sleep mode.

21.5.1 SLEEP MODE

In Sleep mode, the clock to the CPU and peripherals is shut down. If an on-chip oscillator is being used, it is shut down.

The Fail-Safe Clock Monitor is not functional during Sleep, since there is no clock to monitor. However, LPRC clock remains active if WDT is operational during Sleep.

The Brown-out protection circuit, if enabled, will remain functional during Sleep.

The processor wakes up from Sleep if at least one of the following conditions has occurred:

- any interrupt that is individually enabled and meets the required priority level
- any Reset (POR, BOR and $\overline{\text{MCLR}}$)
- WDT time-out

On waking up from Sleep mode, the processor will restart the same clock that was active prior to entry into Sleep mode. When clock switching is enabled, bits COSC<2:0> will determine the oscillator source that will be used on wake-up. If clock switch is disabled, then there is only one system clock.

Note:	If a POR or BOR occurred, the selection of the oscillator is based on the FOS<2:0> and FPR<4:0> Configuration bits.

If the clock source is an oscillator, the clock to the device is held off until OST times out (indicating a stable oscillator). If PLL is used, the system clock is held off until LOCK = 1 (indicating that the PLL is stable). Either way, TPOR, TLOCK and TPWRT delays are applied.

If EC, FRC, LPRC or ERC oscillators are used, then a delay of TPOR (~ 10 µs) is applied. This is the smallest delay possible on wake-up from Sleep.

Moreover, if LP oscillator was active during Sleep, and LP is the oscillator used on wake-up, then the start-up delay will be equal to TPOR. PWRT delay and OST timer delay are not applied. In order to have the smallest possible start-up delay when waking up from Sleep, one of these faster wake-up options should be selected before entering Sleep.

Any interrupt that is individually enabled (using the corresponding IE bit) and meets the prevailing priority level will be able to wake-up the processor. The processor will process the interrupt and branch to the ISR. The Sleep Status bit in RCON register is set upon wake-up.

Note:	In spite of various delays applied (TPOR, TLOCK and TPWRT), the crystal oscillator (and PLL) may not be active at the end of the time-out (e.g., for low-frequency crystals). In such cases, if FSCM is enabled, then the device will detect this as a clock failure and process the clock failure trap, the FRC oscillator will be enabled, and the user will have to re-enable the crystal oscillator. If FSCM is not enabled, then the device will simply suspend execution of code until the clock is stable, and will remain in Sleep until the oscillator clock has started.

All Resets will wake-up the processor from Sleep mode. Any Reset, other than POR, will set the Sleep Status bit. In a POR, the Sleep bit is cleared.

If Watchdog Timer is enabled, then the processor will wake-up from Sleep mode upon WDT time-out. The Sleep and WDTO Status bits are both set.

TABLE 21-7: SYSTEM INTEGRATION REGISTER MAP FOR dsPIC30F6010A DEVICES(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
RCON	0740	TRAPR	IOPUWR	BGST	—	—	—	—	—	EXTR	SWR	SWDTEN	WDTO	SLEEP	IDLE	BOR	POR	Depends on type of Reset.
OSCCON	0742	—	COSC<2:0>			—	NOSC<2:0>			POST<1:0>		LOCK	—	CF	—	LPOSCEN	OSWEN	Depends on Configuration bits.
OSCTUN	0744	—	—	—	—	—	—	—	—	—	—	TUN<5:0>						0000 0000 0000 0000
PMD1	0770	T5MD	T4MD	T3MD	T2MD	T1MD	QEIMD	PWMMD	—	I2CMD	U2MD	U1MD	SPI2MD	SPI1MD	C2MD	C1MD	ADCMD	0000 0000 0000 0000
PMD2	0772	IC8MD	IC7MD	IC6MD	IC5MD	IC4MD	IC3MD	IC2MD	IC1MD	OC8MD	OC7MD	OC6MD	OC5MD	OC4MD	OC3MD	OC2MD	OC1MD	0000 0000 0000 0000

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

TABLE 21-8: SYSTEM INTEGRATION REGISTER MAP FOR dsPIC30F6015 DEVICES(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
RCON	0740	TRAPR	IOPUWR	BGST	—	—	—	—	—	EXTR	SWR	SWDTEN	WDTO	SLEEP	IDLE	BOR	POR	Depends on type of Reset.
OSCCON	0742	—	COSC<2:0>			—	NOSC<2:0>			POST<1:0>		LOCK	—	CF	—	LPOSCEN	OSWEN	Depends on Configuration bits.
OSCTUN	0744	—	—	—	—	—	—	—	—	—	—	TUN<5:0>						0000 0000 0000 0000
PMD1	0770	T5MD	T4MD	T3MD	T2MD	T1MD	QEIMD	PWMMD	—	I2CMD	U2MD	U1MD	SPI2MD	SPI1MD	—	C1MD	ADCMD	0000 0000 0000 0000
PMD2	0772	IC8MD	IC7MD	IC6MD	IC5MD	IC4MD	IC3MD	IC2MD	IC1MD	OC8MD	OC7MD	OC6MD	OC5MD	OC4MD	OC3MD	OC2MD	OC1MD	0000 0000 0000 0000

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

TABLE 21-9: DEVICE CONFIGURATION REGISTER MAP(1)

Address	Name	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
0xF80000	FOSC	FCKSM<1:0>		—	—	—	FOS<2:0>			—	—	—	FPR<4:0>				
0xF80002	FWDT	FWDTEN	—	—	—	—	—	—	—	—	—	FWPSA<1:0>		FWPSB<3:0>			
0xF80004	FBORPOR	MCLREN	—	—	—	—	PWMPIN	HPOL	LPOL	BOREN	—	BORV<1:0>		—	—	FPWRT<1:0>	
0xF80006	FBS	—	—	RBS<1:0>		—	—	—	EBS	—	—	—	—	BSS<2:0>			BWRP
0xF80008	FSS	—	—	RSS<1:0>		—	—	ESS<1:0>		—	—	—	—	SSS<2:0>			SWRP
0xF8000A	FGS	—	—	—	—	—	—	—	—	—	—	—	—	—	GSS<1:0>		GWRP
0xF8000C	FICD	BKBUG	COE	—	—	—	—	—	—	—	—	—	—	—	—	ICS<1:0>	

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

TABLE 22-2: INSTRUCTION SET OVERVIEW (CONTINUED)

Base Instr #	Assembly Mnemonic	Assembly Syntax		Description	# of words	# of cycles	Status Flags Affected
72	SUB	SUB	Acc	Subtract Accumulators	1	1	OA,OB,OAB, SA,SB,SAB
		SUB	f	f = f – WREG	1	1	C,DC,N,OV,Z
		SUB	f,WREG	WREG = f – WREG	1	1	C,DC,N,OV,Z
		SUB	#lit10,Wn	Wn = Wn – lit10	1	1	C,DC,N,OV,Z
		SUB	Wb,Ws,Wd	Wd = Wb – Ws	1	1	C,DC,N,OV,Z
		SUB	Wb,#lit5,Wd	Wd = Wb – lit5	1	1	C,DC,N,OV,Z
73	SUBB	SUBB	f	f = f – WREG – ($\overline{C}$)	1	1	C,DC,N,OV,Z
		SUBB	f,WREG	WREG = f – WREG – ($\overline{C}$)	1	1	C,DC,N,OV,Z
		SUBB	#lit10,Wn	Wn = Wn – lit10 - ($\overline{C}$)	1	1	C,DC,N,OV,Z
		SUBB	Wb,Ws,Wd	Wd = Wb – Ws – ($\overline{C}$)	1	1	C,DC,N,OV,Z
		SUBB	Wb,#lit5,Wd	Wd = Wb – lit5 – ($\overline{C}$)	1	1	C,DC,N,OV,Z
74	SUBR	SUBR	f	f = WREG – f	1	1	C,DC,N,OV,Z
		SUBR	f,WREG	WREG = WREG – f	1	1	C,DC,N,OV,Z
		SUBR	Wb,Ws,Wd	Wd = Ws – Wb	1	1	C,DC,N,OV,Z
		SUBR	Wb,#lit5,Wd	Wd = lit5 - Wb	1	1	C,DC,N,OV,Z
75	SUBBR	SUBBR	f	f = WREG – f - ($\overline{C}$)	1	1	C,DC,N,OV,Z
		SUBBR	f,WREG	WREG = WREG – f – ($\overline{C}$)	1	1	C,DC,N,OV,Z
		SUBBR	Wb,Ws,Wd	Wd = Ws – Wb – ($\overline{C}$)	1	1	C,DC,N,OV,Z
		SUBBR	Wb,#lit5,Wd	Wd = lit5 – Wb – ($\overline{C}$)	1	1	C,DC,N,OV,Z
76	SWAP	SWAP.b	Wn	Wn = nibble swap Wn	1	1	None
		SWAP	Wn	Wn = byte swap Wn	1	1	None
77	TBLRDH	TBLRDH	Ws,Wd	Read Prog<23:16> to Wd<7:0>	1	2	None
78	TBLRDL	TBLRDL	Ws,Wd	Read Prog<15:0> to Wd	1	2	None
79	TBLWTH	TBLWTH	Ws,Wd	Write Ws<7:0> to Prog<23:16>	1	2	None
80	TBLWTL	TBLWTL	Ws,Wd	Write Ws to Prog<15:0>	1	2	None
81	ULNK	ULNK		Unlink Frame Pointer	1	1	None
82	XOR	XOR	f	f = f .XOR. WREG	1	1	N,Z
		XOR	f,WREG	WREG = f .XOR. WREG	1	1	N,Z
		XOR	#lit10,Wn	Wd = lit10 .XOR. Wd	1	1	N,Z
		XOR	Wb,Ws,Wd	Wd = Wb .XOR. Ws	1	1	N,Z
		XOR	Wb,#lit5,Wd	Wd = Wb .XOR. lit5	1	1	N,Z
83	ZE	ZE	Ws,Wnd	Wnd = Zero-Extend Ws	1	1	C,Z,N

TABLE 24-11: 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	BOR Hysteresis		—	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 24-12: 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	0.8	2	2.6	ms	RTSP
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	0.8	2	2.6	ms	RTSP
D135	TRETD	Characteristic Retention	40	100	—	Year	Provided no other specifications are violated
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.

TABLE 24-15: PLL CLOCK TIMING SPECIFICATIONS (V_{DD} = 2.5 TO 5.5 V)

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**
OS50	FPLLI	PLL Input Frequency Range(2)	4	—	10	MHz	EC with 4x PLL
			4	—	10	MHz	EC with 8x PLL
			4	—	7.5(4)	MHz	EC with 16x PLL
			4	—	10	MHz	XT with 4x PLL
			4	—	10	MHz	XT with 8x PLL
			4	—	7.5(4)	MHz	XT with 16x PLL
			5(3)	—	10	MHz	HS/2 with 4x PLL
			5(3)	—	10	MHz	HS/2 with 8x PLL
			5(3)	—	7.5(4)	MHz	HS/2 with 16x PLL
			4	—	8.33(3)	MHz	HS/3 with 4x PLL
			4	—	8.33(3)	MHz	HS/3 with 8x PLL
			4	—	7.5(4)	MHz	HS/3 with 16x PLL
OS51	FSYS	On-Chip PLL Output(2)	16	—	120	MHz	EC, XT, HS/2, HS/3 modes with PLL
OS52	TLOC	PLL Start-up Time (Lock Time)	—	20	50	µs	—

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: Limited by oscillator frequency range.

4: Limited by device operating frequency range.

TABLE 24-16: PLL JITTER

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.	**Characteristic**	**Min**	**Typ(1)**	**Max**	**Units**	**Conditions**	
OS61	x4 PLL	—	0.251	0.413	%	-40°C ≤TA ≤+85°C	VDD = 3.0 to 3.6V
		—	0.251	0.413	%	-40°C ≤TA ≤+125°C	VDD = 3.0 to 3.6V
		—	0.256	0.47	%	-40°C ≤TA ≤+85°C	VDD = 4.5 to 5.5V
		—	0.256	0.47	%	-40°C ≤TA ≤+125°C	VDD = 4.5 to 5.5V
	x8 PLL	—	0.355	0.584	%	-40°C ≤TA ≤+85°C	VDD = 3.0 to 3.6V
		—	0.355	0.584	%	-40°C ≤TA ≤+125°C	VDD = 3.0 to 3.6V
		—	0.362	0.664	%	-40°C ≤TA ≤+85°C	VDD = 4.5 to 5.5V
		—	0.362	0.664	%	-40°C ≤TA ≤+125°C	VDD = 4.5 to 5.5V
	x16 PLL	—	0.67	0.92	%	-40°C ≤TA ≤+85°C	VDD = 3.0 to 3.6V
		—	0.632	0.956	%	-40°C ≤TA ≤+85°C	VDD = 4.5 to 5.5V
		—	0.632	0.956	%	-40°C ≤TA ≤+125°C	VDD = 4.5 to 5.5V

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

TABLE 24-40: 10-BIT HIGH-SPEED A/D MODULE SPECIFICATIONS(1)

AC CHARACTERISTICS			Standard Operating Conditions: 2.7V 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**
Device Supply							
AD01	AVDD	Module VDD Supply	Greater of VDD – 0.3 or 2.7	—	Lesser of VDD + 0.3 or 5.5	V	—
AD02	AVSS	Module VSS Supply	Vss - 0.3	—	VSS + 0.3	V	—
Reference Inputs							
AD05	VREFH	Reference Voltage High	AVss + 2.7	—	AVDD	V	—
AD06	VREFL	Reference Voltage Low	AVss	—	AVDD – 2.7	V	—
AD07	VREF	Absolute Reference Voltage	AVss – 0.3	—	AVDD + 0.3	V	—
AD08	IREF	Current Drain	—	200 .001	300 3	μA μA	A/D operating A/D off
Analog Input							
AD10	VINH-VINL	Full-Scale Input Span	VREFL		VREFH	V	—
AD11	Vin	Absolute Input Voltage	AVSS - 0.3	—	AVDD + 0.3	V	—
AD12	—	Leakage Current	—	±0.001	±0.244	μA	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 5V Source Impedance = 5 kΩ
AD13	—	Leakage Current	—	±0.001	±0.244	μA	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V Source Impedance = 5 kΩ
AD17	RIN	Recommended Impedance of Analog Voltage Source	—	—	—	Ω	See Table 20-2
DC Accuracy							
AD20	Nr	Resolution	10 data bits			bits	—
AD21	INL	Integral Nonlinearity(2)	—	±1	±1	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 5V
AD21A	INL	Integral Nonlinearity(2)	—	±1	±1	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD22	DNL	Differential Nonlinearity(2)	—	±1	±1	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 5V
AD22A	DNL	Differential Nonlinearity(2)	—	±1	±1	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD23	GERR	Gain Error(2)	±1	±5	±6	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 5V
AD23A	GERR	Gain Error(2)	±1	±5	±6	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V

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

2: Measurements taken with external VREF+ and VREF- used as the ADC voltage references.

3: The A/D conversion result never decreases with an increase in the input voltage, and has no missing codes.