



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.

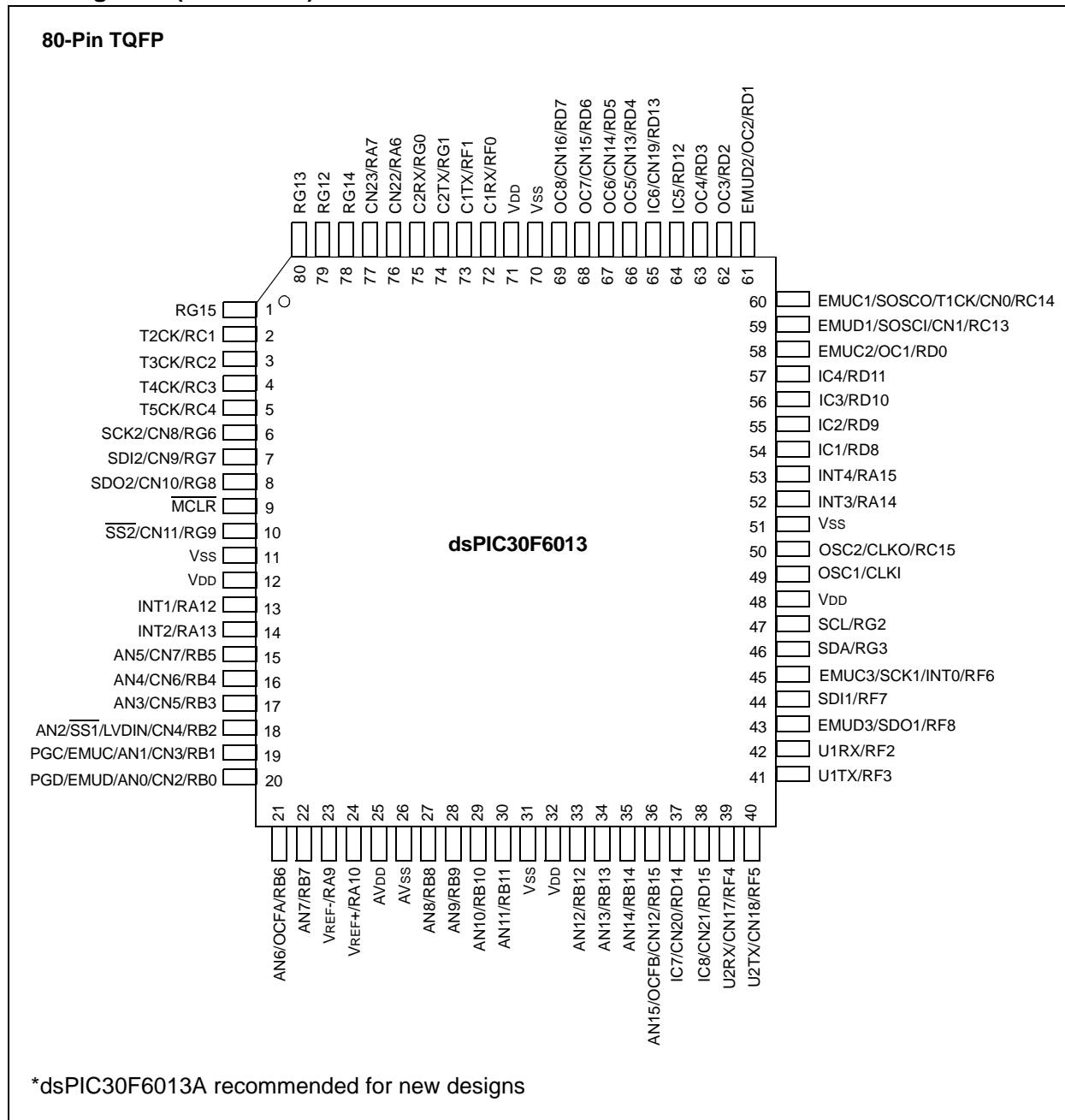
### Applications of "[Embedded - Microcontrollers](#)"

#### Details

Product Status	Obsolete
Core Processor	dsPIC
Core Size	16-Bit
Speed	20 MIPS
Connectivity	CANbus, I <sup>2</sup> C, SPI, UART/USART
Peripherals	AC'97, Brown-out Detect/Reset, I <sup>2</sup> S, LVD, POR, PWM, WDT
Number of I/O	52
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 16x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (14x14)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/dspic30f6012-20i-pf">https://www.e-xfl.com/product-detail/microchip-technology/dspic30f6012-20i-pf</a>

# dsPIC30F6011/6012/6013/6014

## Pin Diagrams (Continued)



**Note:** For descriptions of individual pins, see **Section 1.0 “Device Overview”**.

## 2.4 DSP Engine

The DSP engine consists of a high-speed 17-bit x 17-bit multiplier, a barrel shifter and a 40-bit adder/subtractor (with two target accumulators, round and saturation logic).

The dsPIC30F is a single-cycle instruction flow architecture, therefore, concurrent operation of the DSP engine with MCU instruction flow is not possible. However, some MCU ALU and DSP engine resources may be used concurrently by the same instruction (e.g., ED, EDAC).

The DSP engine also has the capability to perform inherent accumulator-to-accumulator operations, which require no additional data. These instructions are ADD, SUB and NEG.

The DSP engine has various options selected through various bits in the CPU Core Configuration register (CORCON), as listed below:

1. Fractional or integer DSP multiply (IF).
2. Signed or unsigned DSP multiply (US).
3. Conventional or convergent rounding (RND).
4. Automatic saturation on/off for ACCA (SATA).
5. Automatic saturation on/off for ACCB (SATB).
6. Automatic saturation on/off for writes to data memory (SATDW).
7. Accumulator Saturation mode selection (ACCSAT).

**Note:** For CORCON layout, see Table 3-3.

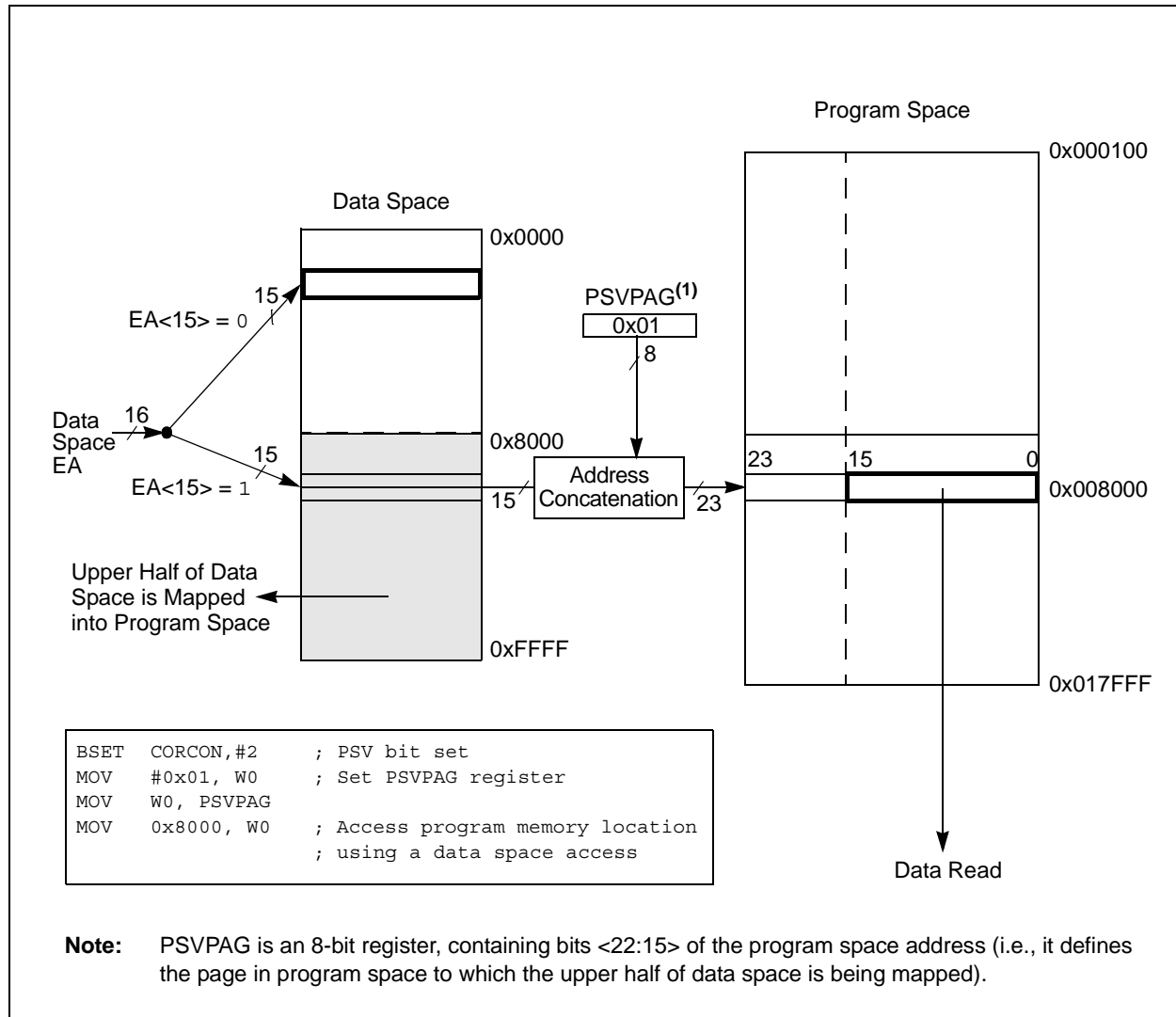
A block diagram of the DSP engine is shown in Figure 2-2.

**TABLE 2-2: DSP INSTRUCTIONS SUMMARY**

Instruction	Algebraic Operation	ACC WB?
CLR	$A = 0$	Yes
ED	$A = (x - y)^2$	No
EDAC	$A = A + (x - y)^2$	No
MAC	$A = A + (x * y)$	Yes
MAC	$A = A + x^2$	No
MOVSAC	No change in A	Yes
MPY	$A = x * y$	No
MPY.N	$A = -x * y$	No
MSC	$A = A - x * y$	Yes

# dsPIC30F6011/6012/6013/6014

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



## 3.2 Data Address Space

The core has two data spaces. The data spaces can be considered either separate (for some DSP instructions), or as one unified linear address range (for MCU instructions). The data spaces are accessed using two Address Generation Units (AGUs) and separate data paths.

### 3.2.1 DATA SPACE MEMORY MAP

The data space memory is split into two blocks, X and Y data space. A key element of this architecture is that Y space is a subset of X space, and is fully contained within X space. In order to provide an apparent linear addressing space, X and Y spaces have contiguous addresses.

When executing any instruction other than one of the MAC class of instructions, the X block consists of the 64-Kbyte data address space (including all Y addresses). When executing one of the MAC class of instructions, the X block consists of the 64-Kbyte data address space excluding the Y address block (for data reads only). In other words, all other instructions regard the entire data memory as one composite address space. The MAC class instructions extract the Y address space from data space and address it using EAs sourced from W10 and W11. The remaining X data space is addressed using W8 and W9. Both address spaces are concurrently accessed only with the MAC class instructions.

The data space memory maps are shown in Figure 3-8 and Figure 3-9.

### 3.2.2 DATA SPACES

The X data space is used by all instructions and supports all addressing modes. There are separate read and write data buses. The X read data bus is the return data path for all instructions that view data space as combined X and Y address space. It is also the X address space data path for the dual operand read instructions (MAC class). The X write data bus is the only write path to data space for all instructions.

The X data space also supports Modulo Addressing for all instructions, subject to addressing mode restrictions. Bit-Reversed Addressing is only supported for writes to X data space.

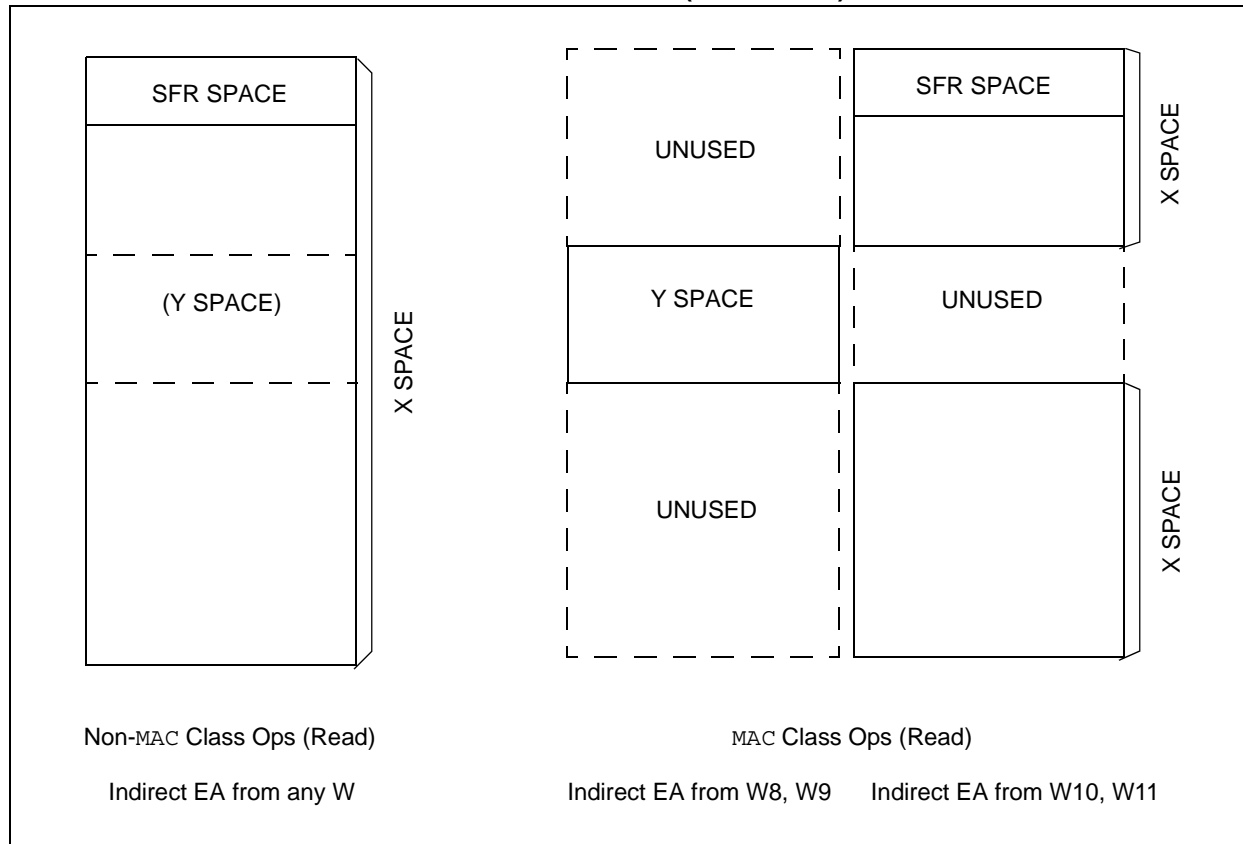
The Y data space is used in concert with the X data space by the MAC class of instructions (CLR, ED, EDAC, MAC, MOVSA, MPY, MPY.N and MSC) to provide two concurrent data read paths. No writes occur across the Y bus. This class of instructions dedicates two W register pointers, W10 and W11, to always address Y data space, independent of X data space, whereas W8 and W9 always address X data space. Note that during accumulator write back, the data address space is considered a combination of X and Y data spaces, so the write occurs across the X bus. Consequently, the write can be to any address in the entire data space.

The Y data space can only be used for the data prefetch operation associated with the MAC class of instructions. It also supports Modulo Addressing for automated circular buffers. Of course, all other instructions can access the Y data address space through the X data path as part of the composite linear space.

The boundary between the X and Y data spaces is defined as shown in Figure 3-8 and Figure 3-8 and is not user programmable. Should an EA point to data outside its own assigned address space, or to a location outside physical memory, an all zero word/byte will be returned. For example, although Y address space is visible by all non-MAC instructions using any addressing mode, an attempt by a MAC instruction to fetch data from that space using W8 or W9 (X space pointers) will return 0x0000.

# dsPIC30F6011/6012/6013/6014

**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.

**TABLE 6-1: NVM REGISTER MAP**

File 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	All RESETS
NVMCON	0760	WR	WREN	WRERR	—	—	—	—	TWRI	—	PROGOP<6:0>							0000 0000 0000 0000
NVMADR	0762	NVMADR<15:0>																uuuu uuuu uuuu uuuu
NVMADRU	0764	—	—	—	—	—	—	—	—	NVMADR<23:16>							0000 0000 uuuu uuuu	
NVMKEY	0766	—	—	—	—	—	—	—	—	KEY<7:0>							0000 0000 0000 0000	

**Legend:** u = uninitialized bit

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

# dsPIC30F6011/6012/6013/6014

## 8.3 Input Change Notification Module

The input change notification module provides the dsPIC30F devices the ability to generate interrupt requests to the processor, in response to a change of state on selected input pins. This module is capable of detecting input change of states even in Sleep mode, when the clocks are disabled. There are up to 24 external signals (CN0 through CN23) that may be selected (enabled) for generating an interrupt request on a change of state.

**TABLE 8-10: INPUT CHANGE NOTIFICATION REGISTER MAP FOR dsPIC30F6011/6012 (BITS 15-8)**

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Reset State
CNEN1	00C0	CN15IE	CN14IE	CN13IE	CN12IE	CN11IE	CN10IE	CN9IE	CN8IE	0000 0000 0000 0000
CNEN2	00C2	—	—	—	—	—	—	—	—	0000 0000 0000 0000
CNPU1	00C4	CN15PUE	CN14PUE	CN13PUE	CN12PUE	CN11PUE	CN10PUE	CN9PUE	CN8PUE	0000 0000 0000 0000
CNPU2	00C6	—	—	—	—	—	—	—	—	0000 0000 0000 0000

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

**TABLE 8-11: INPUT CHANGE NOTIFICATION REGISTER MAP FOR dsPIC30F6011/6012 (BITS 7-0)**

SFR Name	Addr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset State
CNEN1	00C0	CN7IE	CN6IE	CN5IE	CN4IE	CN3IE	CN2IE	CN1IE	CN0IE	0000 0000 0000 0000
CNEN2	00C2	—	—	—	—	—	CN18IE	CN17IE	CN16IE	0000 0000 0000 0000
CNPU1	00C4	CN7PUE	CN6PUE	CN5PUE	CN4PUE	CN3PUE	CN2PUE	CN1PUE	CN0PUE	0000 0000 0000 0000
CNPU2	00C6	—	—	—	—	—	CN18PUE	CN17PUE	CN16PUE	0000 0000 0000 0000

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

**TABLE 8-12: INPUT CHANGE NOTIFICATION REGISTER MAP FOR dsPIC30F6013/6014 (BITS 15-8)**

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Reset State
CNEN1	00C0	CN15IE	CN14IE	CN13IE	CN12IE	CN11IE	CN10IE	CN9IE	CN8IE	0000 0000 0000 0000
CNEN2	00C2	—	—	—	—	—	—	—	—	0000 0000 0000 0000
CNPU1	00C4	CN15PUE	CN14PUE	CN13PUE	CN12PUE	CN11PUE	CN10PUE	CN9PUE	CN8PUE	0000 0000 0000 0000
CNPU2	00C6	—	—	—	—	—	—	—	—	0000 0000 0000 0000

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

**TABLE 8-13: INPUT CHANGE NOTIFICATION REGISTER MAP FOR dsPIC30F6013/6014 (BITS 7-0)**

SFR Name	Addr.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset State
CNEN1	00C0	CN7IE	CN6IE	CN5IE	CN4IE	CN3IE	CN2IE	CN1IE	CN0IE	0000 0000 0000 0000
CNEN2	00C2	CN23IE	CN22IE	CN21IE	CN20IE	CN19IE	CN18IE	CN17IE	CN16IE	0000 0000 0000 0000
CNPU1	00C4	CN7PUE	CN6PUE	CN5PUE	CN4PUE	CN3PUE	CN2PUE	CN1PUE	CN0PUE	0000 0000 0000 0000
CNPU2	00C6	CN23PUE	CN22PUE	CN21PUE	CN20PUE	CN19PUE	CN18PUE	CN17PUE	CN16PUE	0000 0000 0000 0000

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



## 9.0 TIMER1 MODULE

**Note:** This data sheet summarizes features of this group of dsPIC30F devices and is not intended to be a complete reference source. For more information on the CPU, peripherals, register descriptions and general device functionality, refer to the “dsPIC30F Family Reference Manual” (DS70046).

This section describes the 16-bit General Purpose Timer1 module and associated operational modes. Figure 9-1 depicts the simplified block diagram of the 16-bit Timer1 module.

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

The Timer1 module is a 16-bit timer which can serve as the time counter for the real-time clock, or operate as a free-running interval timer/counter. The 16-bit timer has the following modes:

- 16-bit Timer
- 16-bit Synchronous Counter
- 16-bit Asynchronous Counter

Further, the following operational characteristics are supported:

- Timer gate operation
- Selectable prescaler settings
- Timer operation during CPU Idle and Sleep modes
- Interrupt on 16-bit Period register match or falling edge of external gate signal

These Operating modes are determined by setting the appropriate bit(s) in the 16-bit SFR, T1CON. Figure 9-1 presents a block diagram of the 16-bit timer module.

**16-bit Timer Mode:** In the 16-bit Timer mode, the timer increments on every instruction cycle up to a match value preloaded into the Period register PR1, then resets to '0' and continues to count.

When the CPU goes into the Idle mode, the timer will stop incrementing unless the TSIDL (T1CON<13>) bit = 0. If TSIDL = 1, the timer module logic will resume the incrementing sequence upon termination of the CPU Idle mode.

**16-bit Synchronous Counter Mode:** In the 16-bit Synchronous Counter mode, the timer increments on the rising edge of the applied external clock signal which is synchronized with the internal phase clocks. The timer counts up to a match value preloaded in PR1, then resets to '0' and continues.

When the CPU goes into the Idle mode, the timer will stop incrementing unless the respective TSIDL bit = 0. If TSIDL = 1, the timer module logic will resume the incrementing sequence upon termination of the CPU Idle mode.

**16-bit Asynchronous Counter Mode:** In the 16-bit Asynchronous Counter mode, the timer increments on every rising edge of the applied external clock signal. The timer counts up to a match value preloaded in PR1, then resets to '0' and continues.

When the timer is configured for the Asynchronous mode of operation and the CPU goes into the Idle mode, the timer will stop incrementing if TSIDL = 1.

## 10.0 TIMER2/3 MODULE

**Note:** This data sheet summarizes features of this group of dsPIC30F devices and is not intended to be a complete reference source. For more information on the CPU, peripherals, register descriptions and general device functionality, refer to the “dsPIC30F Family Reference Manual” (DS70046).

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

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

- Input Capture
- Output Compare/Simple PWM

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

The 32-bit timer has the following modes:

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

Further, the following operational characteristics are supported:

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

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

For 32-bit timer/counter operation, Timer2 is the least significant word and Timer3 is the most significant word of the 32-bit timer.

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

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

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

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

For synchronous 32-bit reads of the Timer2/Timer3 pair, reading the least significant word (TMR2 register) will cause the most significant word (msw) to be read and latched into a 16-bit holding register, termed TMR3HLD.

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

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

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

## 11.0 TIMER4/5 MODULE

**Note:** This data sheet summarizes features of this group of dsPIC30F devices and is not intended to be a complete reference source. For more information on the CPU, peripherals, register descriptions and general device functionality, refer to the "dsPIC30F Family Reference Manual" (DS70046).

This section describes the second 32-bit General Purpose Timer module (Timer4/5) and associated operational modes. Figure 11-1 depicts the simplified block diagram of the 32-bit Timer4/5 module. Figure 11-2 and Figure 11-3 show Timer4/5 configured as two independent 16-bit timers, Timer4 and Timer5, respectively.

The Timer4/5 module is similar in operation to the Timer2/3 module. However, there are some differences which are as follows:

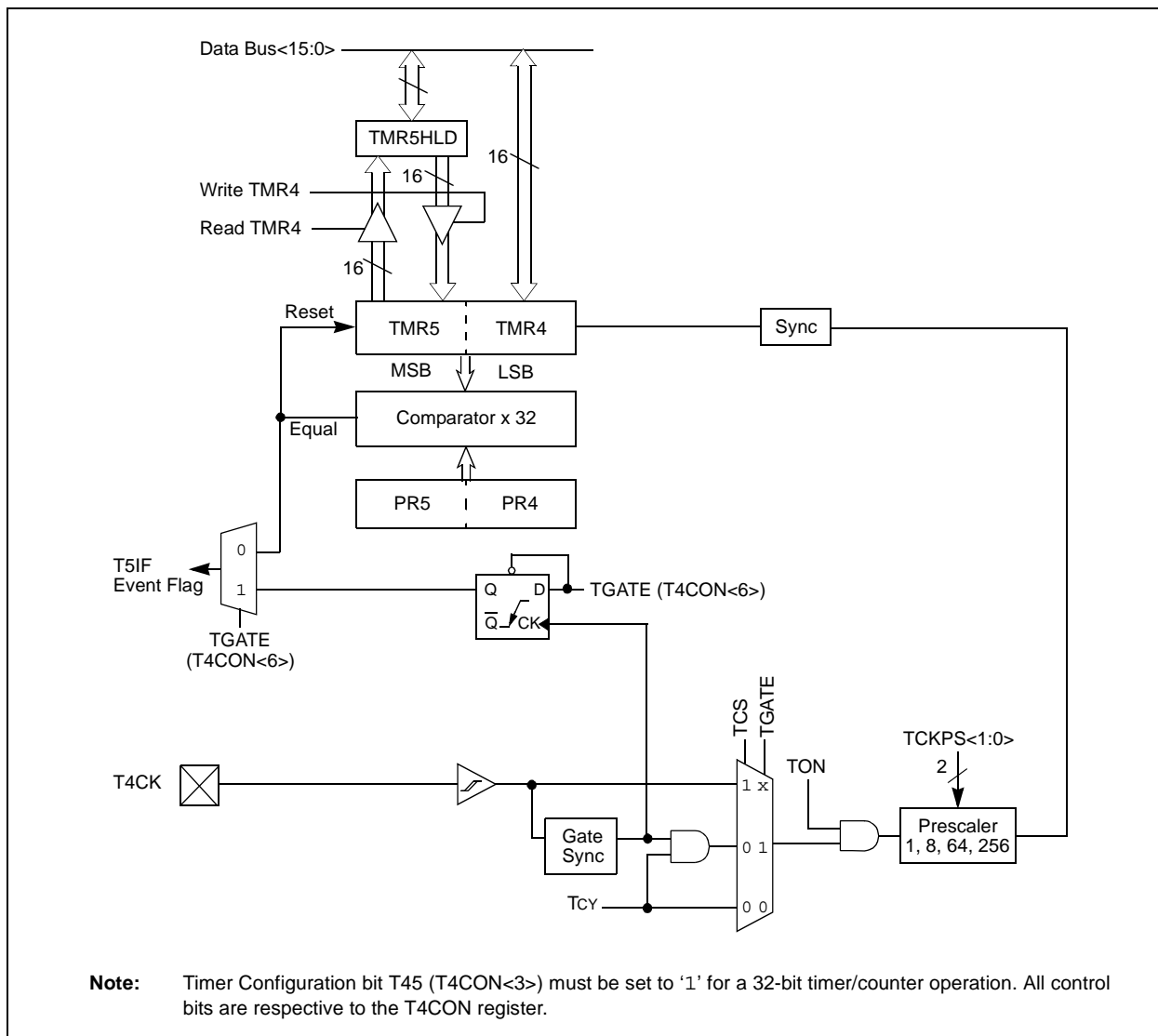
- The Timer4/5 module does not support the ADC event trigger feature
- Timer4/5 can not be utilized by other peripheral modules, such as input capture and output compare

The operating modes of the Timer4/5 module are determined by setting the appropriate bit(s) in the 16-bit T4CON and T5CON SFRs.

For 32-bit timer/counter operation, Timer4 is the lsw and Timer5 is the msw of the 32-bit timer.

**Note:** For 32-bit timer operation, T5CON control bits are ignored. Only T4CON control bits are used for setup and control. Timer4 clock and gate inputs are utilized for the 32-bit timer module but an interrupt is generated with the Timer5 interrupt flag (T5IF) and the interrupt is enabled with the Timer5 interrupt enable bit (T5IE).

**FIGURE 11-1: 32-BIT TIMER4/5 BLOCK DIAGRAM**



# dsPIC30F6011/6012/6013/6014

## 12.1.2 CAPTURE BUFFER OPERATION

Each capture channel has an associated FIFO buffer which is four 16-bit words deep. There are two status flags which provide status on the FIFO buffer:

- ICBFNE – Input Capture Buffer Not Empty
- ICOV – Input Capture Overflow

The ICBFNE will be set on the first input capture event and remain set until all capture events have been read from the FIFO. As each word is read from the FIFO, the remaining words are advanced by one position within the buffer.

In the event that the FIFO is full with four capture events and a fifth capture event occurs prior to a read of the FIFO, an overflow condition will occur and the ICOV bit will be set to a logic '1'. The fifth capture event is lost and is not stored in the FIFO. No additional events will be captured until all four events have been read from the buffer.

If a FIFO read is performed after the last read and no new capture event has been received, the read will yield indeterminate results.

## 12.1.3 TIMER2 AND TIMER3 SELECTION MODE

The input capture module consists of up to 8 input capture channels. Each channel can select between one of two timers for the time base, Timer2 or Timer3.

Selection of the timer resource is accomplished through SFR bit, ICTMR (ICxCON<7>). Timer3 is the default timer resource available for the input capture module.

## 12.1.4 HALL SENSOR MODE

When the input capture module is set for capture on every edge, rising and falling, ICM<2:0> = 001, the following operations are performed by the input capture logic:

- The input capture interrupt flag is set on every edge, rising and falling.
- The interrupt on Capture mode setting bits, ICI<1:0>, is ignored since every capture generates an interrupt.
- A capture overflow condition is not generated in this mode.

## 12.2 Input Capture Operation During Sleep and Idle Modes

An input capture event will generate a device wake-up or interrupt, if enabled, if the device is in CPU Idle or Sleep mode.

Independent of the timer being enabled, the input capture module will wake-up from the CPU Sleep or Idle mode when a capture event occurs if ICM<2:0> = 111 and the interrupt enable bit is asserted. The same wake-up can generate an interrupt if the conditions for processing the interrupt have been satisfied. The wake-up feature is useful as a method of adding extra external pin interrupts.

### 12.2.1 INPUT CAPTURE IN CPU SLEEP MODE

CPU Sleep mode allows input capture module operation with reduced functionality. In the CPU Sleep mode, the ICI<1:0> bits are not applicable and the input capture module can only function as an external interrupt source.

The capture module must be configured for interrupt only on rising edge (ICM<2:0> = 111) in order for the input capture module to be used while the device is in Sleep mode. The prescale settings of 4:1 or 16:1 are not applicable in this mode.

### 12.2.2 INPUT CAPTURE IN CPU IDLE MODE

CPU Idle mode allows input capture module operation with full functionality. In the CPU Idle mode, the Interrupt mode selected by the ICI<1:0> bits is applicable, as well as the 4:1 and 16:1 capture prescale settings which are defined by control bits ICM<2:0>. This mode requires the selected timer to be enabled. Moreover, the ICSIDL bit must be asserted to a logic '0'.

If the input capture module is defined as ICM<2:0> = 111 in CPU Idle mode, the input capture pin will serve only as an external interrupt pin.

## 12.3 Input Capture Interrupts

The input capture channels have the ability to generate an interrupt based upon the selected number of capture events. The selection number is set by control bits ICI<1:0> (ICxCON<6:5>).

Each channel provides an interrupt flag (ICxIF) bit. The respective capture channel interrupt flag is located in the corresponding IFSx Status register.

Enabling an interrupt is accomplished via the respective capture channel interrupt enable (ICxIE) bit. The capture interrupt enable bit is located in the corresponding IEC Control register.

# dsPIC30F6011/6012/6013/6014

FIGURE 14-1: SPI BLOCK DIAGRAM

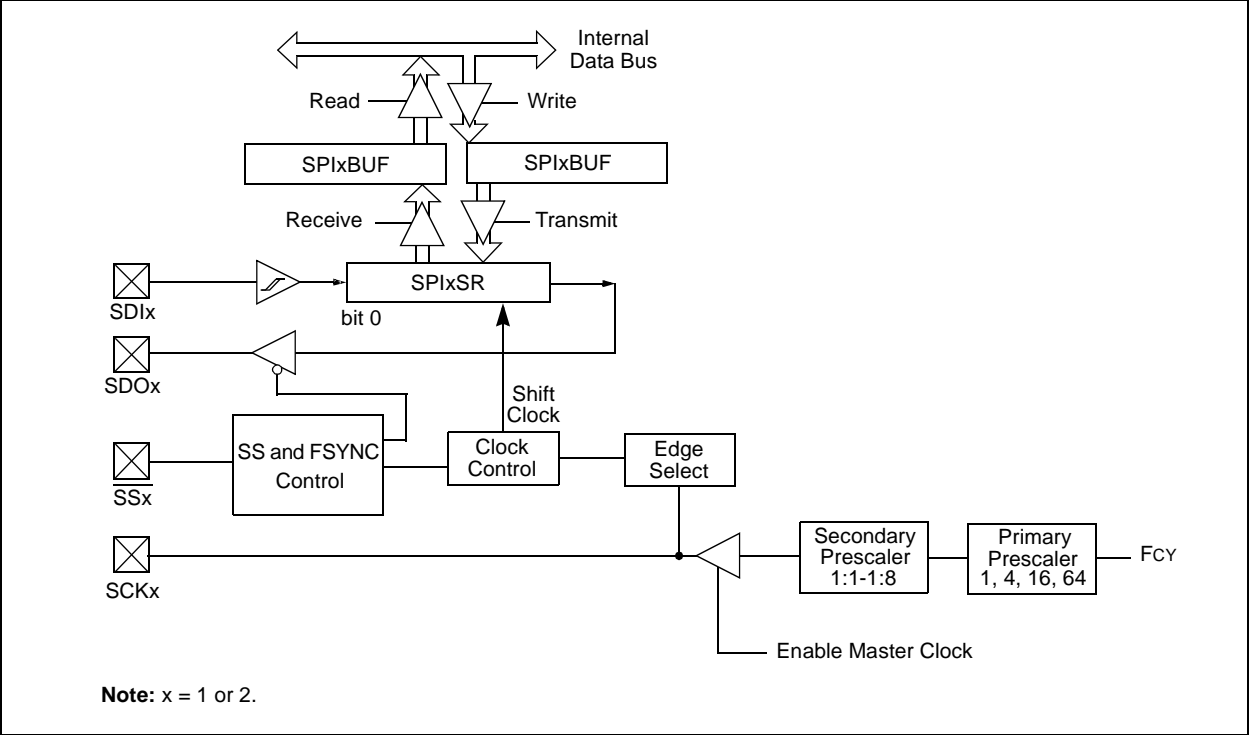
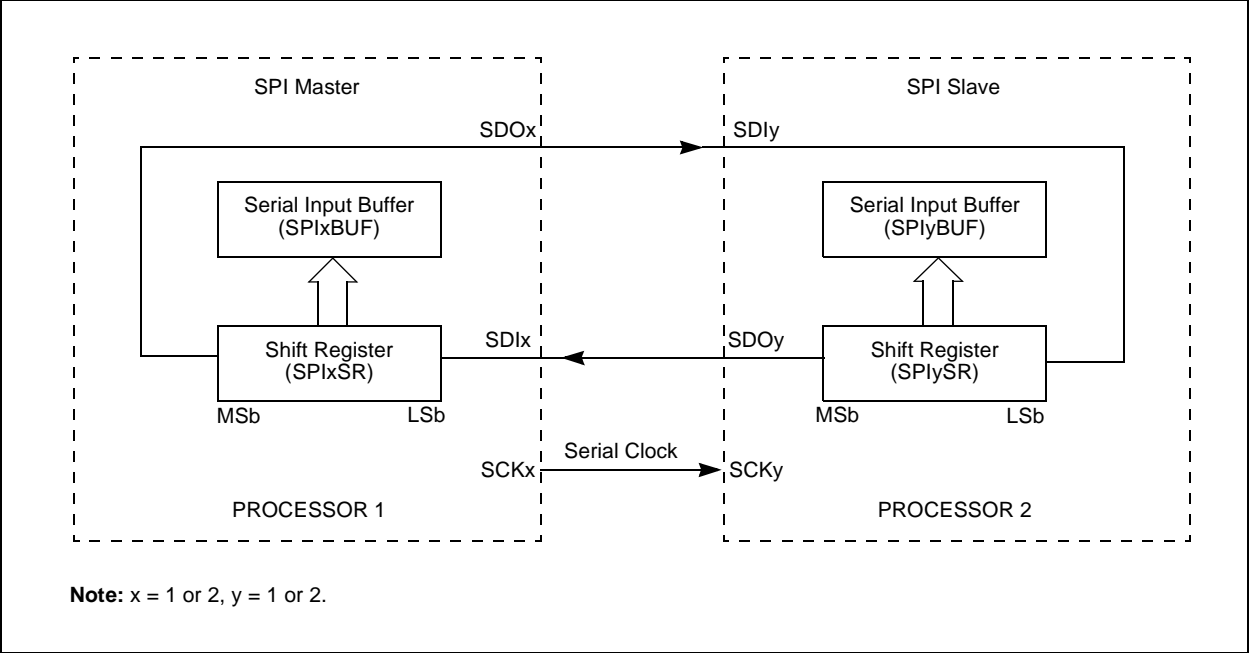


FIGURE 14-2: SPI MASTER/S�AVE CONNECTION



# dsPIC30F6011/6012/6013/6014

---

NOTES:

## 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.

TABLE 17-1: CAN1 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		
C1RXF0SID	0300	—	—	—	Receive Acceptance Filter 0 Standard Identifier <10:0>											—	EXIDE	000u uuuu uuuu uu0u		
C1RXF0EIDH	0302	—	—	—	—	Receive Acceptance Filter 0 Extended Identifier <17:6>													0000 uuuu uuuu uuuu	
C1RXF0EIDL	0304	Receive Acceptance Filter 0 Extended Identifier <5:0>					—	—	—	—	—	—	—	—	—	—	—	uuuu uu00 0000 0000		
C1RXF1SID	0308	—	—	—	Receive Acceptance Filter 1 Standard Identifier <10:0>											—	EXIDE	000u uuuu uuuu uu0u		
C1RXF1EIDH	030A	—	—	—	—	Receive Acceptance Filter 1 Extended Identifier <17:6>													0000 uuuu uuuu uuuu	
C1RXF1EIDL	030C	Receive Acceptance Filter 1 Extended Identifier <5:0>					—	—	—	—	—	—	—	—	—	—	—	uuuu uu00 0000 0000		
C1RXF2SID	0310	—	—	—	Receive Acceptance Filter 2 Standard Identifier <10:0>											—	EXIDE	000u uuuu uuuu uu0u		
C1RXF2EIDH	0312	—	—	—	—	Receive Acceptance Filter 2 Extended Identifier <17:6>													0000 uuuu uuuu uuuu	
C1RXF2EIDL	0314	Receive Acceptance Filter 2 Extended Identifier <5:0>					—	—	—	—	—	—	—	—	—	—	—	uuuu uu00 0000 0000		
C1RXF3SID	0318	—	—	—	Receive Acceptance Filter 3 Standard Identifier <10:0>											—	EXIDE	000u uuuu uuuu uu0u		
C1RXF3EIDH	031A	—	—	—	—	Receive Acceptance Filter 3 Extended Identifier <17:6>													0000 uuuu uuuu uuuu	
C1RXF3EIDL	031C	Receive Acceptance Filter 3 Extended Identifier <5:0>					—	—	—	—	—	—	—	—	—	—	—	uuuu uu00 0000 0000		
C1RXF4SID	0320	—	—	—	Receive Acceptance Filter 4 Standard Identifier <10:0>											—	EXIDE	000u uuuu uuuu uu0u		
C1RXF4EIDH	0322	—	—	—	—	Receive Acceptance Filter 4 Extended Identifier <17:6>													0000 uuuu uuuu uuuu	
C1RXF4EIDL	0324	Receive Acceptance Filter 4 Extended Identifier <5:0>					—	—	—	—	—	—	—	—	—	—	—	uuuu uu00 0000 0000		
C1RXF5SID	0328	—	—	—	Receive Acceptance Filter 5 Standard Identifier <10:0>											—	EXIDE	000u uuuu uuuu uu0u		
C1RXF5EIDH	032A	—	—	—	—	Receive Acceptance Filter 5 Extended Identifier <17:6>													0000 uuuu uuuu uuuu	
C1RXF5EIDL	032C	Receive Acceptance Filter 5 Extended Identifier <5:0>					—	—	—	—	—	—	—	—	—	—	—	uuuu uu00 0000 0000		
C1RXM0SID	0330	—	—	—	Receive Acceptance Mask 0 Standard Identifier <10:0>											—	MIDE	000u uuuu uuuu uu0u		
C1RXM0EIDH	0332	—	—	—	—	Receive Acceptance Mask 0 Extended Identifier <17:6>													0000 uuuu uuuu uuuu	
C1RXM0EIDL	0334	Receive Acceptance Mask 0 Extended Identifier <5:0>					—	—	—	—	—	—	—	—	—	—	—	uuuu uu00 0000 0000		
C1RXM1SID	0338	—	—	—	Receive Acceptance Mask 1 Standard Identifier <10:0>											—	MIDE	000u uuuu uuuu uu0u		
C1RXM1EIDH	033A	—	—	—	—	Receive Acceptance Mask 1 Extended Identifier <17:6>													0000 uuuu uuuu uuuu	
C1RXM1EIDL	033C	Receive Acceptance Mask 1 Extended Identifier <5:0>					—	—	—	—	—	—	—	—	—	—	—	uuuu uu00 0000 0000		
C1TX2SID	0340	Transmit Buffer 2 Standard Identifier <10:6>					—	—	—	Transmit Buffer 2 Standard Identifier <5:0>					SRR		TXIDE	uuuu u000 uuuu uuuu		
C1TX2EID	0342	Transmit Buffer 2 Extended Identifier <17:14>				—	—	—	Transmit Buffer 2 Extended Identifier <13:6>											uuuu 0000 uuuu uuuu
C1TX2DLC	0344	Transmit Buffer 2 Extended Identifier <5:0>					TXRTR		TXRB1	TXRB0	DLC<3:0>				—	—	—	uuuu uuuu uuuu u000		
C1TX2B1	0346	Transmit Buffer 2 Byte 1								Transmit Buffer 2 Byte 0								uuuu uuuu uuuu uuuu		
C1TX2B2	0348	Transmit Buffer 2 Byte 3								Transmit Buffer 2 Byte 2								uuuu uuuu uuuu uuuu		
C1TX2B3	034A	Transmit Buffer 2 Byte 5								Transmit Buffer 2 Byte 4								uuuu uuuu uuuu uuuu		
C1TX2B4	034C	Transmit Buffer 2 Byte 7								Transmit Buffer 2 Byte 6								uuuu uuuu uuuu uuuu		
C1TX2CON	034E	—	—	—	—	—	—	—	—	—	TXABT	TXLABB	TXERR	TXREQ	—	TXPRI<1:0>		0000 0000 0000 0000		
C1TX1SID	0350	Transmit Buffer 1 Standard Identifier <10:6>					—	—	—	Transmit Buffer 1 Standard Identifier <5:0>					SRR		TXIDE	uuuu u000 uuuu uuuu		

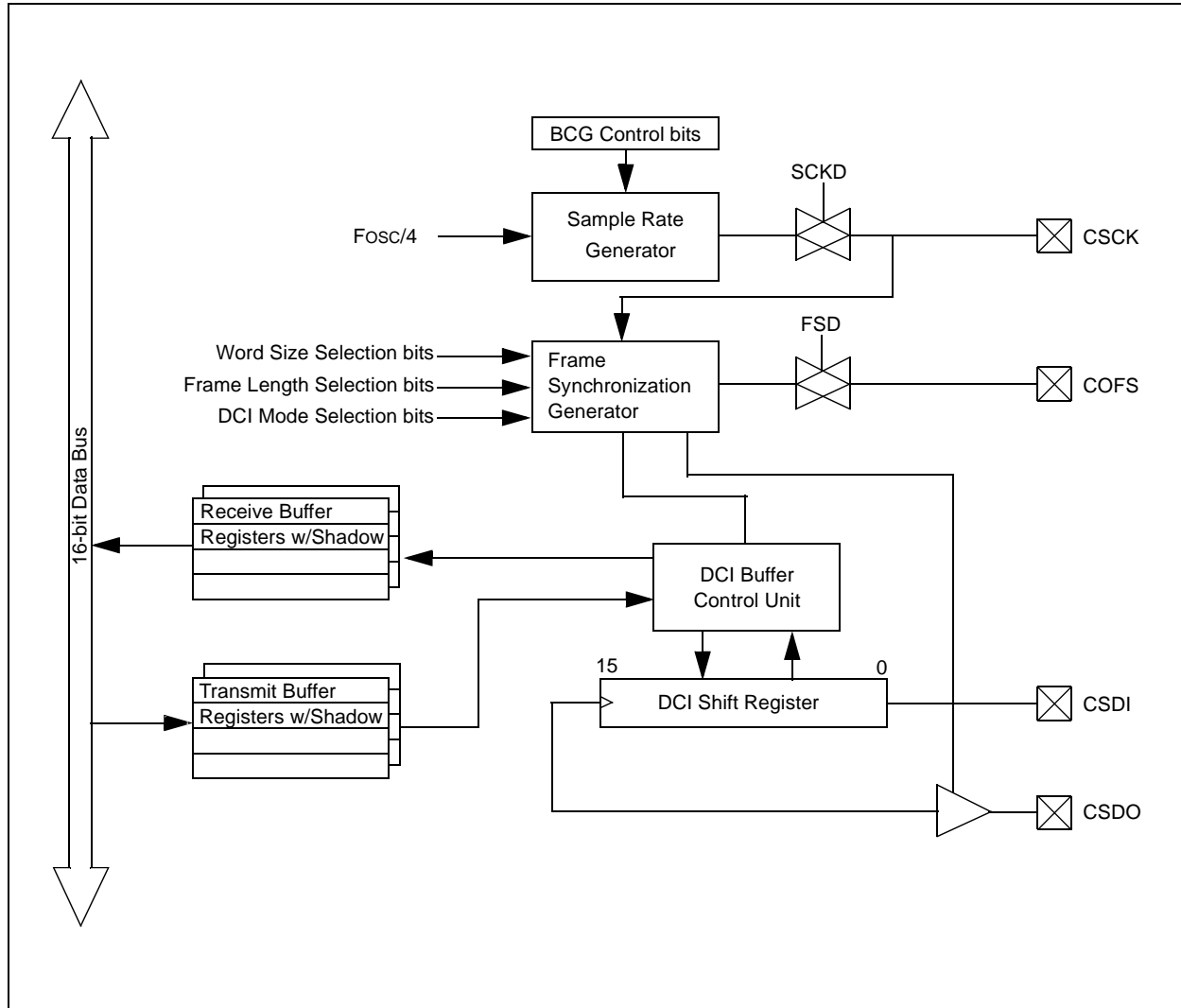
**Legend:** u = uninitialized bit

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



# dsPIC30F6011/6012/6013/6014

FIGURE 18-1: DCI MODULE BLOCK DIAGRAM



## 18.3 DCI Module Operation

### 18.3.1 MODULE ENABLE

The DCI module is enabled or disabled by setting/clearing the DCIEN control bit in the DCICON1 SFR. Clearing the DCIEN control bit has the effect of resetting the module. In particular, all counters associated with CSM generation, frame sync, and the DCI buffer control unit are reset.

The DCI clocks are shutdown when the DCIEN bit is cleared.

When enabled, the DCI controls the data direction for the four I/O pins associated with the module. The Port, LAT and TRIS register values for these I/O pins are overridden by the DCI module when the DCIEN bit is set.

It is also possible to override the CSM pin separately when the bit clock generator is enabled. This permits the bit clock generator to operate without enabling the rest of the DCI module.

### 18.3.2 WORD SIZE SELECTION BITS

The WS<3:0> word size selection bits in the DCICON2 SFR determine the number of bits in each DCI data word. Essentially, the WS<3:0> bits determine the counting period for a 4-bit counter clocked from the CSM signal.

Any data length, up to 16 bits, may be selected. The value loaded into the WS<3:0> bits is one less the desired word length. For example, a 16-bit data word size is selected when WS<3:0> = 1111.

**Note:** These WS<3:0> control bits are used only in the Multi-Channel and I<sup>2</sup>S modes. These bits have no effect in AC-Link mode since the data slot sizes are fixed by the protocol.

### 18.3.3 FRAME SYNC GENERATOR

The frame sync generator (COFSG) is a 4-bit counter that sets the frame length in data words. The frame sync generator is incremented each time the word size counter is reset (refer to **Section 18.3.2 “Word Size Selection Bits”**). The period for the frame synchronization generator is set by writing the COFSG<3:0> control bits in the DCICON2 SFR. The COFSG period in clock cycles is determined by the following formula:

#### EQUATION 18-1: COFSG PERIOD

$$\text{Frame Length} = \text{Word Length} \cdot (\text{FSG Value} + 1)$$

Frame lengths, up to 16 data words, may be selected. The frame length in CSM periods can vary up to a maximum of 256 depending on the word size that is selected.

**Note:** The COFSG control bits will have no effect in AC-Link mode since the frame length is set to 256 CSM periods by the protocol.

### 18.3.4 FRAME SYNC MODE CONTROL BITS

The type of frame sync signal is selected using the Frame Synchronization mode control bits (COFSM<1:0>) in the DCICON1 SFR. The following operating modes can be selected:

- Multi-Channel mode
- I<sup>2</sup>S mode
- AC-Link mode (16-bit)
- AC-Link mode (20-bit)

The operation of the COFSM control bits depends on whether the DCI module generates the frame sync signal as a master device, or receives the frame sync signal as a slave device.

The master device in a DSP/Codec pair is the device that generates the frame sync signal. The frame sync signal initiates data transfers on the CSDI and CSDO pins and usually has the same frequency as the data sample rate (COFS).

The DCI module is a frame sync master if the COFSD control bit is cleared and is a frame sync slave if the COFSD control bit is set.

### 18.3.5 MASTER FRAME SYNC OPERATION

When the DCI module is operating as a frame sync master device (COFSD = 0), the COFSM mode bits determine the type of frame sync pulse that is generated by the frame sync generator logic.

A new COFS signal is generated when the frame sync generator resets to '0'.

In the Multi-Channel mode, the frame sync pulse is driven high for the CSM period to initiate a data transfer. The number of CSM cycles between successive frame sync pulses will depend on the word size and frame sync generator control bits. A timing diagram for the frame sync signal in Multi-Channel mode is shown in Figure 18-2.

In the AC-Link mode of operation, the frame sync signal has a fixed period and duty cycle. The AC-Link frame sync signal is high for 16 CSM cycles and is low for 240 CSM cycles. A timing diagram with the timing details at the start of an AC-Link frame is shown in Figure 18-3.

In the I<sup>2</sup>S mode, a frame sync signal having a 50% duty cycle is generated. The period of the I<sup>2</sup>S frame sync signal in CSM cycles is determined by the word size and frame sync generator control bits. A new I<sup>2</sup>S data transfer boundary is marked by a high-to-low or a low-to-high transition edge on the COFS pin.

# dsPIC30F6011/6012/6013/6014

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

Most single-word instructions are executed in a single instruction cycle, unless a conditional test is true or the program counter is changed as a result of the instruction. In these cases, the execution takes two instruction cycles with the additional instruction cycle(s) executed as a NOP. Notable exceptions are the BRA (unconditional/computed branch), indirect CALL/GOTO, all table reads and writes, and RETURN/RETFIE instructions,

which are single-word instructions but take two or three cycles. Certain instructions that involve skipping over the subsequent instruction require either two or three cycles if the skip is performed, depending on whether the instruction being skipped is a single word or two-word instruction. Moreover, double-word moves require two cycles. The double-word instructions execute in two instruction cycles.

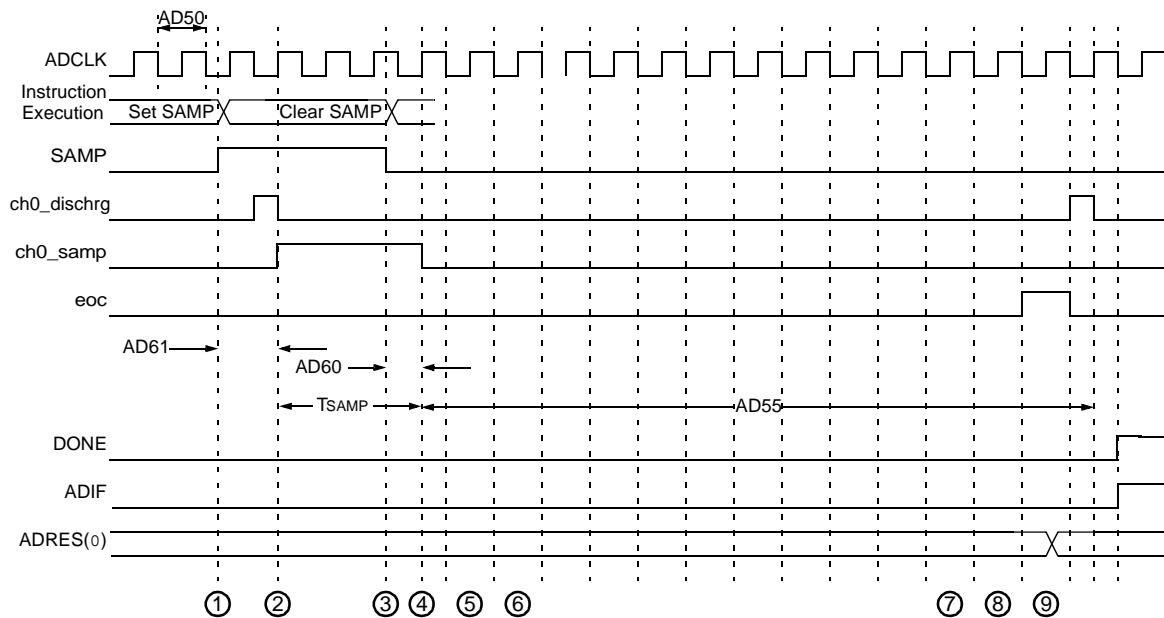
**Note:** For more details on the instruction set, refer to the “dsPIC30F/33F Programmer’s Reference Manual” (DS70157)

**TABLE 21-1: SYMBOLS USED IN OPCODE DESCRIPTIONS**

Field	Description
#text	Means literal defined by “text”
(text)	Means “content of text”
[text]	Means “the location addressed by text”
{ }	Optional field or operation
<n:m>	Register bit field
.b	Byte mode selection
.d	Double-Word mode selection
.S	Shadow register select
.w	Word mode selection (default)
Acc	One of two accumulators {A, B}
AWB	Accumulator write back destination address register $\in \{W13, [W13]+2\}$
bit4	4-bit bit selection field (used in word addressed instructions) $\in \{0...15\}$
C, DC, N, OV, Z	MCU status bits: Carry, Digit Carry, Negative, Overflow, Sticky Zero
Expr	Absolute address, label or expression (resolved by the linker)
f	File register address $\in \{0x0000...0x1FFF\}$
lit1	1-bit unsigned literal $\in \{0,1\}$
lit4	4-bit unsigned literal $\in \{0...15\}$
lit5	5-bit unsigned literal $\in \{0...31\}$
lit8	8-bit unsigned literal $\in \{0...255\}$
lit10	10-bit unsigned literal $\in \{0...255\}$ for Byte mode, $\{0:1023\}$ for Word mode
lit14	14-bit unsigned literal $\in \{0...16384\}$
lit16	16-bit unsigned literal $\in \{0...65535\}$
lit23	23-bit unsigned literal $\in \{0...8388608\}$ ; LSB must be 0
None	Field does not require an entry, may be blank
OA, OB, SA, SB	DSP status bits: ACCA Overflow, ACCB Overflow, ACCA Saturate, ACCB Saturate
PC	Program Counter
Slit10	10-bit signed literal $\in \{-512...511\}$
Slit16	16-bit signed literal $\in \{-32768...32767\}$
Slit6	6-bit signed literal $\in \{-16...16\}$

# dsPIC30F6011/6012/6013/6014

**FIGURE 23-23: 12-BIT A/D CONVERSION TIMING CHARACTERISTICS (ASAM = 0, SSRC = 000)**



- ① – Software sets ADCON. SAMP to start sampling.
- ② – Sampling starts after discharge period.  
TSAMP is described in the “*dsPIC30F Family Reference Manual*” (DS70046), Section 18.
- ③ – Software clears ADCON. SAMP to start conversion.
- ④ – Sampling ends, conversion sequence starts.
- ⑤ – Convert bit 11.
- ⑥ – Convert bit 10.
- ⑦ – Convert bit 1.
- ⑧ – Convert bit 0.
- ⑨ – One TAD for end of conversion.

# dsPIC30F6011/6012/6013/6014

---

## READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (480) 792-4150.

Please list the following information, and use this outline to provide us with your comments about this document.

To: Technical Publications Manager  
RE: Reader Response  
Total Pages Sent \_\_\_\_\_  
From: Name \_\_\_\_\_  
Company \_\_\_\_\_  
Address \_\_\_\_\_  
City / State / ZIP / Country \_\_\_\_\_  
Telephone: (\_\_\_\_\_) \_\_\_\_\_ - \_\_\_\_\_ FAX: (\_\_\_\_\_) \_\_\_\_\_ - \_\_\_\_\_

Application (optional):

Would you like a reply? \_\_\_Y \_\_\_N

Device: dsPIC30F6011/6012/6013/6014 Literature Number: DS70117F

Questions:

1. What are the best features of this document?

---

---

2. How does this document meet your hardware and software development needs?

---

---

3. Do you find the organization of this document easy to follow? If not, why?

---

---

4. What additions to the document do you think would enhance the structure and subject?

---

---

5. What deletions from the document could be made without affecting the overall usefulness?

---

---

6. Is there any incorrect or misleading information (what and where)?

---

---

7. How would you improve this document?

---

---