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/dspic30f6010-30i-pf

2.0 CPU ARCHITECTURE OVERVIEW

Note: This data sheet summarizes features of this group of dsPIC30F devices and is not intended to be a complete reference source. For more information on the CPU, peripherals, register descriptions and general device functionality, refer to the "*dsPIC30F Family Reference Manual*" (DS70046). For more information on the device instruction set and programming, refer to the *"dsPIC30F/33F Programmer's Reference Manual"* (DS70157).

This document provides a summary of the dsPIC30F6010 CPU and peripheral function. For a complete description of this functionality, please refer to the "*dsPIC30F Family Reference Manual*" (DS70046).

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.

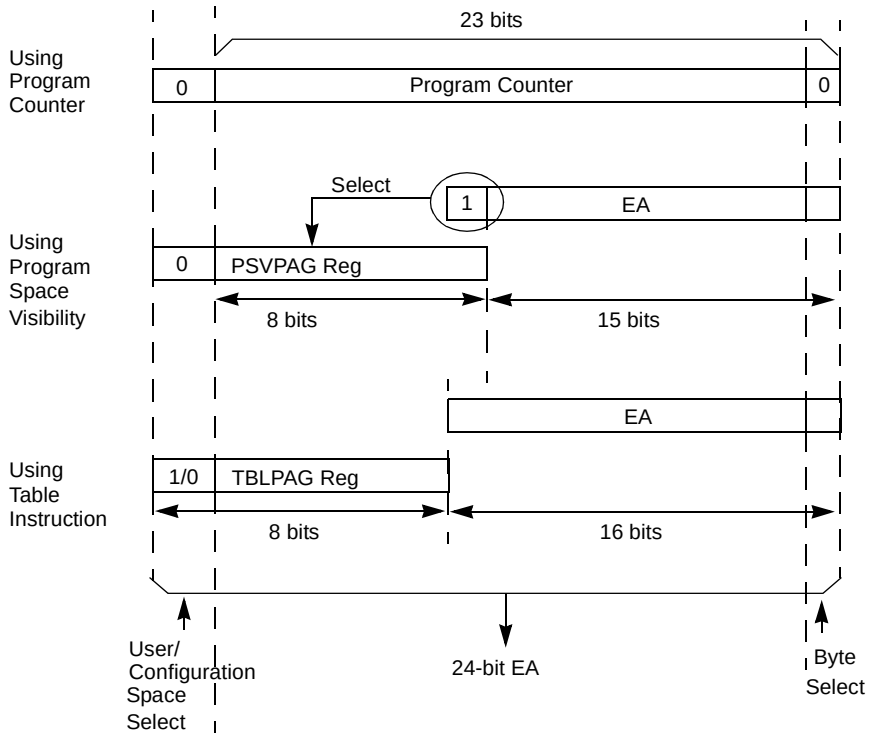
The core does not support a multi-stage instruction pipeline. However, a single stage instruction prefetch mechanism is used, which accesses and partially decodes instructions a cycle ahead of execution, in order to maximize available execution time. Most instructions execute in a single cycle, with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

TABLE 3-1: PROGRAM SPACE ADDRESS CONSTRUCTION

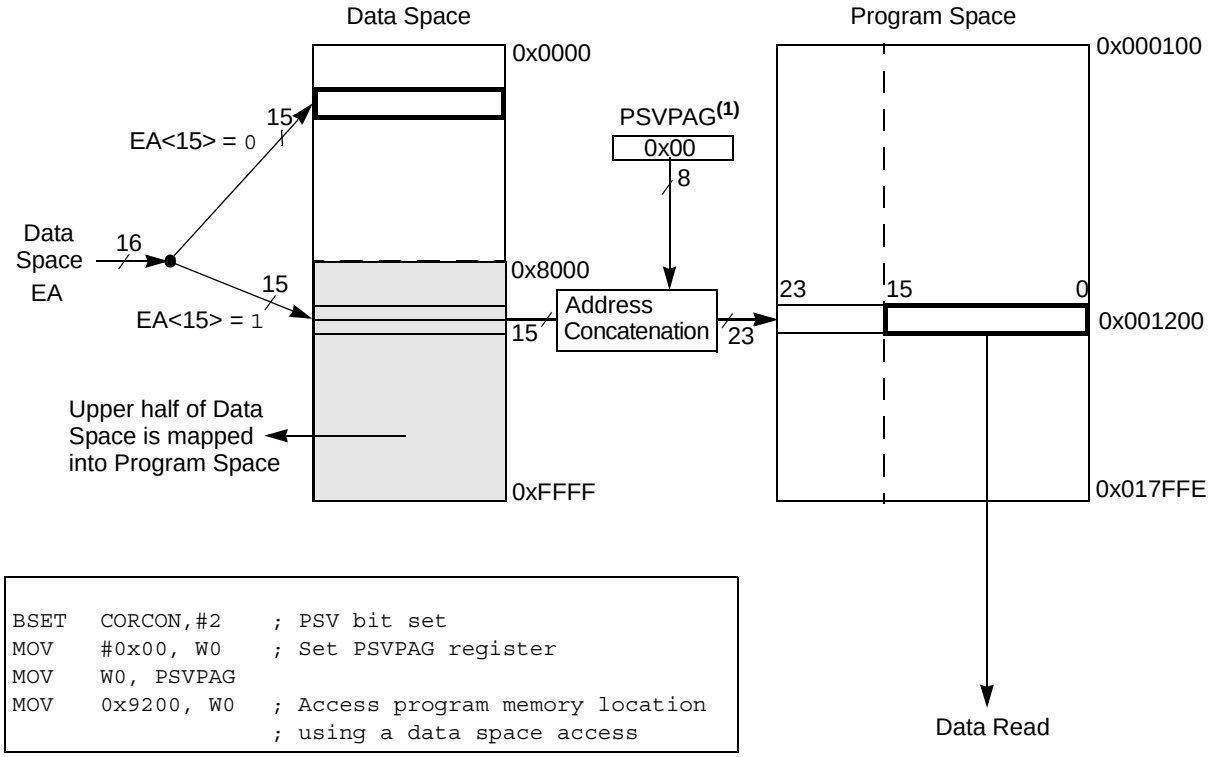
Access Type	Access Space	Program Space Address				
		<23>	<22:16>	<15>	<14:1>	<0>
Instruction Access	User	0	PC<22:1>			0
TBLRD/TBLWT	User (TBLPAG<7> = 0)	TBLPAG<7:0>		Data EA <15:0>		
TBLRD/TBLWT	Configuration (TBLPAG<7> = 1)	TBLPAG<7:0>		Data EA <15:0>		
Program Space Visibility	User	0	PSVPAG<7:0>		Data EA <14:0>	

FIGURE 3-2: DATA ACCESS FROM PROGRAM SPACE ADDRESS GENERATION



Note: Program Space Visibility cannot be used to access bits <23:16> of a word in program memory.

FIGURE 3-5: DATA SPACE WINDOW INTO PROGRAM SPACE OPERATION



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

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.

A data space memory map is shown in Figure 3-6.

Figure 3-7 shows a graphical summary of how X and Y data spaces are accessed for MCU and DSP instructions.

FIGURE 3-6: dsPIC30F6010 DATA SPACE MEMORY MAP

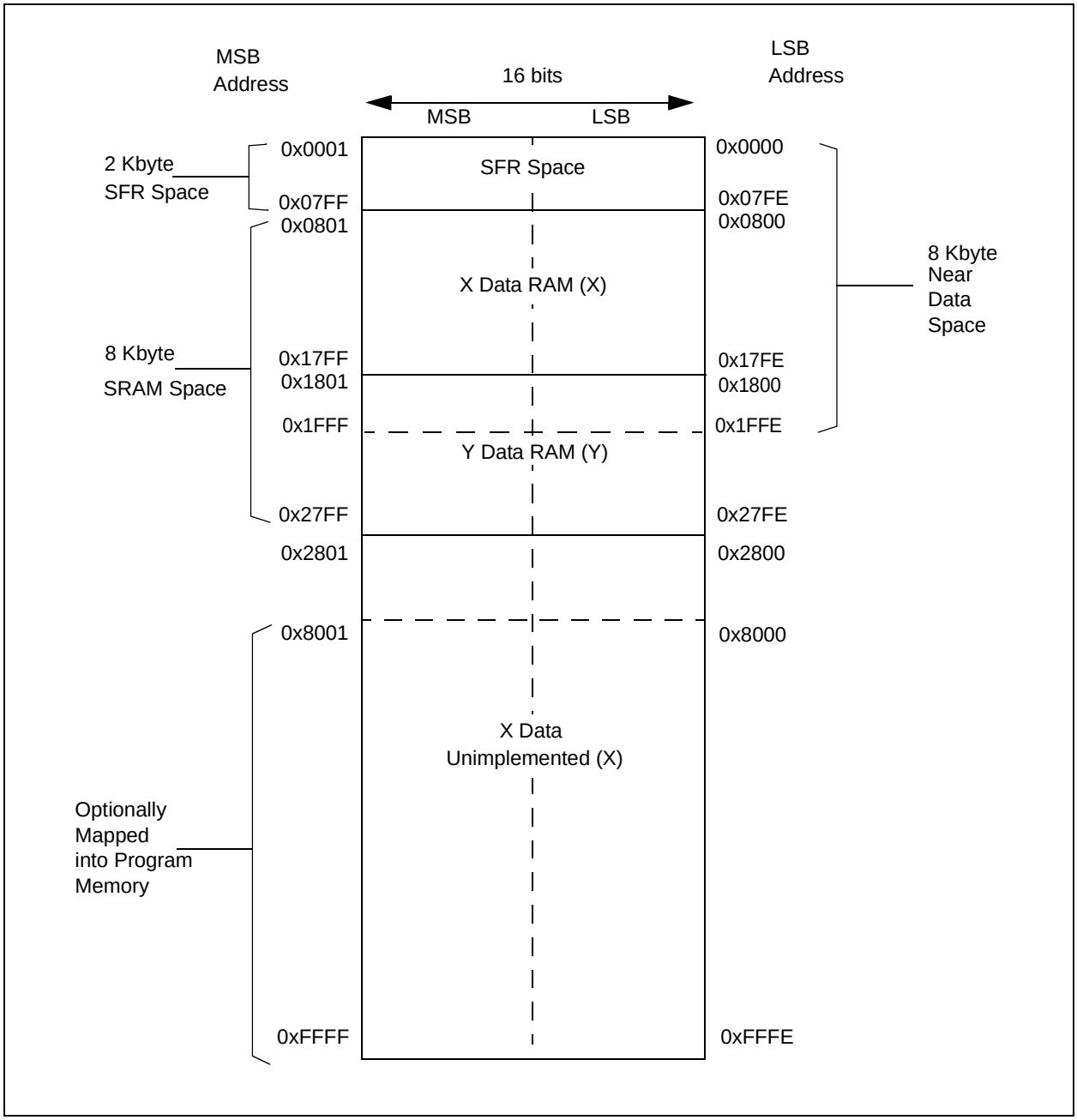
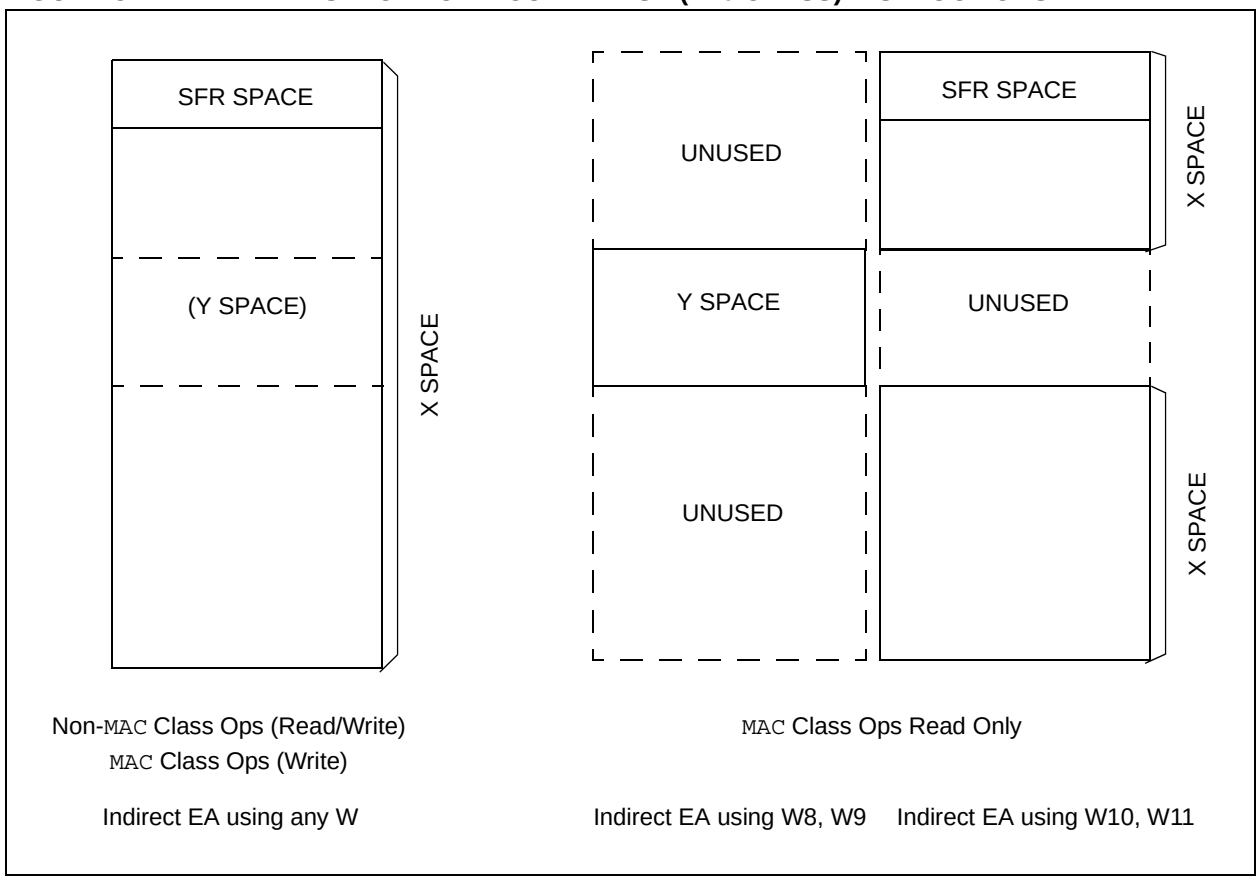


FIGURE 3-7: DATA SPACE FOR MCU AND DSP (MAC CLASS) INSTRUCTIONS EXAMPLE



Address Error Trap:

This trap is initiated when any of the following circumstances occurs:

1. A misaligned data word access is attempted.
2. A data fetch from our unimplemented data memory location is attempted.
3. A data access of an unimplemented program memory location is attempted.
4. An instruction fetch from vector space is attempted.

Note:	In the MAC class of instructions, wherein the data space is split into X and Y data space, unimplemented X space includes all of Y space, and unimplemented Y space includes all of X space.

5. Execution of a "BRA #literal" instruction or a "GOTO #literal" instruction, where literal is an unimplemented program memory address.
6. Executing instructions after modifying the PC to point to unimplemented program memory addresses. The PC may be modified by loading a value into the stack and executing a RETURN instruction.

Stack Error Trap:

This trap is initiated under the following conditions:

1. The Stack Pointer is loaded with a value which is greater than the (user programmable) limit value written into the SPLIM register (stack overflow).
2. The Stack Pointer is loaded with a value which is less than 0x0800 (simple stack underflow).

Oscillator Fail Trap:

This trap is initiated if the external oscillator fails and operation becomes reliant on an internal RC backup.

5.3.2 HARD AND SOFT TRAPS

It is possible that multiple traps can become active within the same cycle (e.g., a misaligned word stack write to an overflowed address). In such a case, the fixed priority shown in Figure 5-2 is implemented, which may require the user to check if other traps are pending, in order to completely correct the fault.

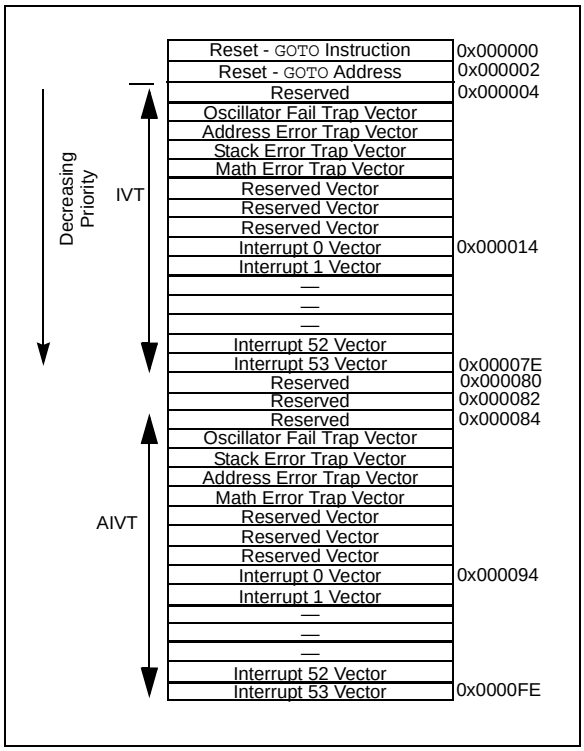
'Soft' traps include exceptions of priority level 8 through level 11, inclusive. The arithmetic error trap (level 11) falls into this category of traps.

'Hard' traps include exceptions of priority level 12 through level 15, inclusive. The address error (level 12), stack error (level 13) and oscillator error (level 14) traps fall into this category.

Each hard trap that occurs must be acknowledged before code execution of any type may continue. If a lower priority hard trap occurs while a higher priority trap is pending, acknowledged, or is being processed, a hard trap conflict will occur.

The device is automatically Reset in a hard trap conflict condition. The TRAPR status bit (RCON<15>) is set when the Reset occurs, so that the condition may be detected in software.

FIGURE 5-1: TRAP VECTORS



8.0 I/O PORTS

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).

All of the device pins (except VDD, VSS, $\overline{\text{MCLR}}$ and OSC1/CLKI) are shared between the peripherals and the parallel I/O ports.

All I/O input ports feature Schmitt Trigger inputs for improved noise immunity.

8.1 Parallel I/O (PIO) Ports

When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled. The I/O pin may be read, but the output driver for the Parallel Port bit will be disabled. If a peripheral is enabled, but the peripheral is not actively driving a pin, that pin may be driven by a port.

All port pins have three registers directly associated with the operation of the port pin. The data direction register (TRISx) determines whether the pin is an input or an output. If the Data Direction bit is a '1', then the pin is an input. All port pins are defined as inputs after a Reset. Reads from the latch (LATx), read the latch. Writes to the latch, write the latch (LATx). Reads from the port (PORTx), read the port pins, and writes to the port pins, write the latch (LATx).

Any bit and its associated data and control registers that are not valid for a particular device will be disabled. That means the corresponding LATx and TRISx registers and the port pin will read as zeros.

When a pin is shared with another peripheral or function that is defined as an input only, it is nevertheless regarded as a dedicated port because there is no other competing source of outputs. An example is the INT4 pin.

The format of the registers for PORTA are shown in Table 8-1.

The TRISA (Data Direction Control) register controls the direction of the RA<7:0> pins, as well as the INTx pins and the VREF pins. The LATA register supplies data to the outputs, and is readable/writable. Reading the PORTA register yields the state of the input pins, while writing the PORTA register modifies the contents of the LATA register.

A parallel I/O (PIO) port that shares a pin with a peripheral is, in general, subservient to the peripheral. The peripheral's output buffer data and control signals are provided to a pair of multiplexers. The multiplexers select whether the peripheral or the associated port has ownership of the output data and control signals of the I/O pad cell. Figure 8-2 shows how ports are shared with other peripherals, and the associated I/O cell (pad) to which they are connected. Table 8-1 shows the formats of the registers for the shared ports, PORTB through PORTG.

FIGURE 8-1: BLOCK DIAGRAM OF A DEDICATED PORT STRUCTURE

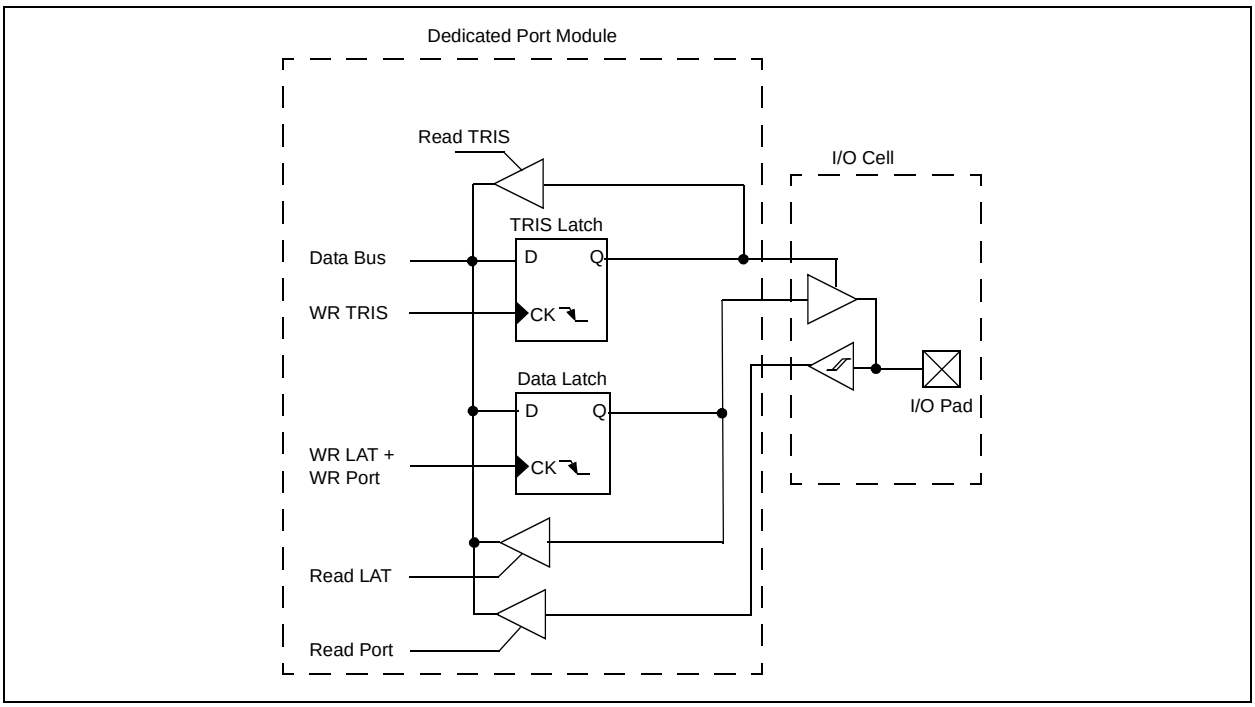
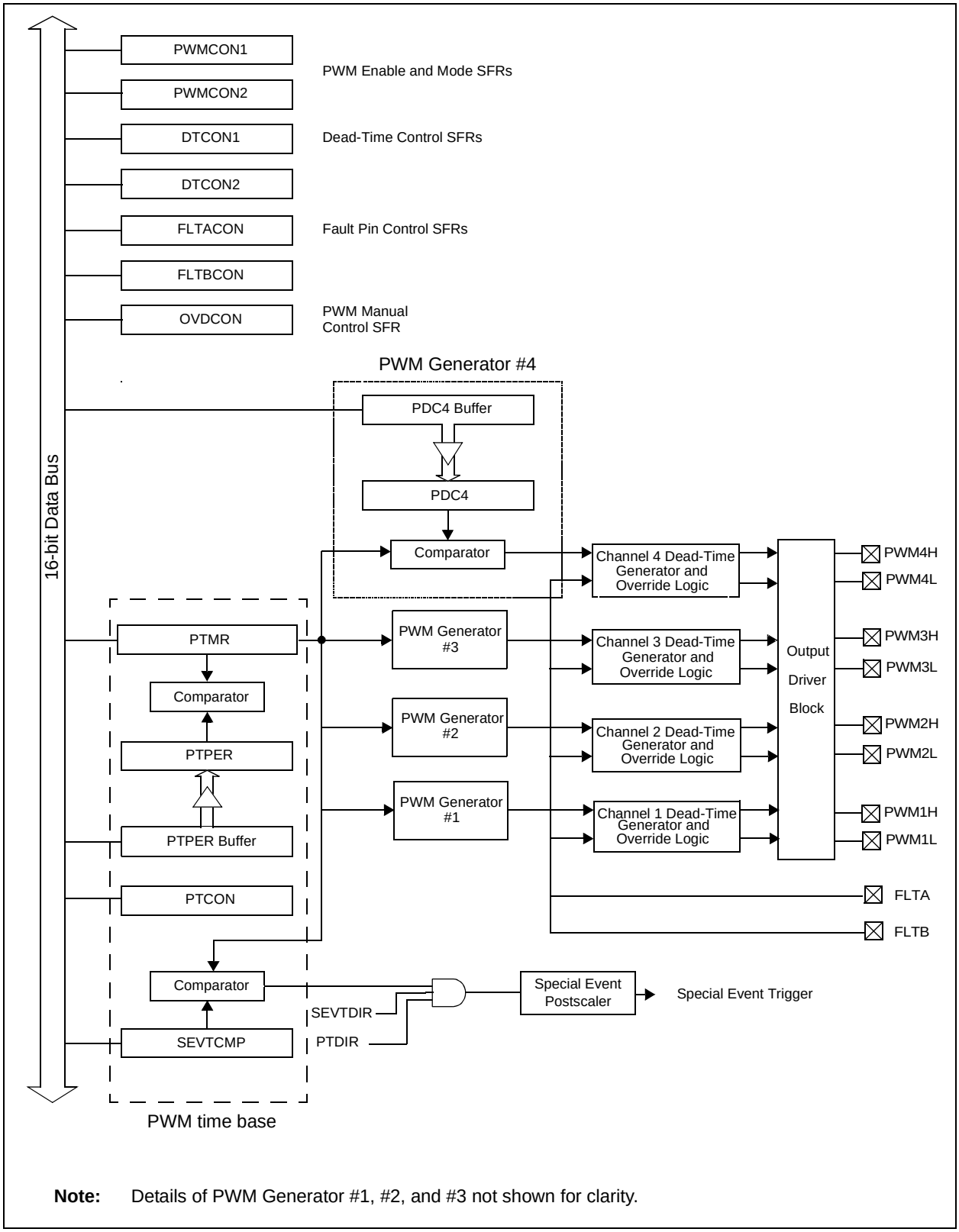


FIGURE 15-1: PWM MODULE BLOCK DIAGRAM



Note: Details of PWM Generator #1, #2, and #3 not shown for clarity.

15.11 PWM Output and Polarity Control

There are three device Configuration bits associated with the PWM module that provide PWM output pin control:

- HPOL Configuration bit
- LPOL Configuration bit
- PWMPIN Configuration bit

These three bits in the FPORBOR configuration register (see Section 21) work in conjunction with the four PWM Enable bits (PWMEN<4:1>) located in the PWMCON1 SFR. The Configuration bits and PWM Enable bits ensure that the PWM pins are in the correct states after a device Reset occurs. The PWMPIN configuration fuse allows the PWM module outputs to be optionally enabled on a device Reset. If PWMPIN = 0, the PWM outputs will be driven to their inactive states at Reset. If PWMPIN = 1 (default), the PWM outputs will be tri-stated. The HPOL bit specifies the polarity for the PWMxH outputs, whereas the LPOL bit specifies the polarity for the PWMxL outputs.

15.11.1 OUTPUT PIN CONTROL

The PEN<4:1>H and PEN<4:1>L control bits in the PWMCON1 SFR enable each high PWM output pin and each low PWM output pin, respectively. If a particular PWM output pin not enabled, it is treated as a general purpose I/O pin.

15.12 PWM Fault Pins

There are two Fault pins (FLTA and FLTB) associated with the PWM module. When asserted, these pins can optionally drive each of the PWM I/O pins to a defined state.

15.12.1 FAULT PIN ENABLE BITS

The FLTACON and FLTBCON SFRs each have 4 control bits that determine whether a particular pair of PWM I/O pins is to be controlled by the Fault input pin. To enable a specific PWM I/O pin pair for Fault overrides, the corresponding bit should be set in the FLTACON or FLTBCON register.

If all enable bits are cleared in the FLTACON or FLTBCON registers, then the corresponding Fault input pin has no effect on the PWM module and the pin may be used as a general purpose interrupt or I/O pin.

Note:	The Fault pin logic can operate independent of the PWM logic. If all the enable bits in the FLTACON/FLTBCON register are cleared, then the Fault pin(s) could be used as general purpose interrupt pin(s). Each Fault pin has an interrupt vector, Interrupt Flag bit and Interrupt Priority bits associated with it.

15.12.2 FAULT STATES

The FLTACON and FLTBCON special function registers have 8 bits each that determine the state of each PWM I/O pin when it is overridden by a Fault input. When these bits are cleared, the PWM I/O pin is driven to the inactive state. If the bit is set, the PWM I/O pin will be driven to the active state. The active and inactive states are referenced to the polarity defined for each PWM I/O pin (HPOL and LPOL polarity control bits).

A special case exists when a PWM module I/O pair is in the Complementary mode and both pins are programmed to be active on a Fault condition. The PWMxH pin always has priority in the Complementary mode, so that both I/O pins cannot be driven active simultaneously.

15.12.3 FAULT PIN PRIORITY

If both Fault input pins have been assigned to control a particular PWM I/O pin, the Fault state programmed for the Fault A input pin will take priority over the Fault B input pin.

15.12.4 FAULT INPUT MODES

Each of the Fault input pins has two modes of operation:

- **Latched Mode:** When the Fault pin is driven low, the PWM outputs will go to the states defined in the FLTACON/FLTBCON register. The PWM outputs will remain in this state until the Fault pin is driven high and the corresponding interrupt flag has been cleared in software. When both of these actions have occurred, the PWM outputs will return to normal operation at the beginning of the next PWM cycle or half-cycle boundary. If the interrupt flag is cleared before the Fault condition ends, the PWM module will wait until the Fault pin is no longer asserted, to restore the outputs.
- **Cycle-by-Cycle Mode:** When the Fault input pin is driven low, the PWM outputs remain in the defined Fault states for as long as the Fault pin is held low. After the Fault pin is driven high, the PWM outputs return to normal operation at the beginning of the following PWM cycle or half-cycle boundary.

The operating mode for each Fault input pin is selected using the FLTAM and FLTBM control bits in the FLTACON and FLTBCON Special Function Registers.

Each of the Fault pins can be controlled manually in software.

15.13 PWM Update Lockout

For a complex PWM application, the user may need to write up to four duty cycle registers and the time base period register, PTPER, at a given time. In some applications, it is important that all buffer registers be written before the new duty cycle and period values are loaded for use by the module.

The PWM update lockout feature is enabled by setting the UDIS control bit in the PWMCON2 SFR. The UDIS bit affects all duty cycle buffer registers and the PWM time base period buffer, PTPER. No duty cycle changes or period value changes will have effect while UDIS = 1.

15.14 PWM Special Event Trigger

The PWM module has a special event trigger that allows A/D conversions to be synchronized to the PWM time base. The A/D sampling and conversion time may be programmed to occur at any point within the PWM period. The special event trigger allows the user to minimize the delay between the time when A/D conversion results are acquired and the time when the duty cycle value is updated.

The PWM special event trigger has an SFR named SEVTCMP, and five control bits to control its operation. The PTMR value for which a special event trigger should occur is loaded into the SEVTCMP register. When the PWM time base is in an Up/Down Counting mode, an additional control bit is required to specify the counting phase for the special event trigger. The count phase is selected using the SEVTDIR control bit in the SEVTCMP SFR. If the SEVTDIR bit is cleared, the special event trigger will occur on the upward counting cycle of the PWM time base. If the SEVTDIR bit is set, the special event trigger will occur on the downward count cycle of the PWM time base. The SEVTDIR control bit has no effect unless the PWM time base is configured for an Up/Down Counting mode.

15.14.1 SPECIAL EVENT TRIGGER POSTSCALER

The PWM special event trigger has a postscaler that allows a 1:1 to 1:16 postscale ratio. The postscaler is configured by writing the SEVOPS<3:0> control bits in the PWMCON2 SFR.

The special event output postscaler is cleared on the following events:

- Any write to the SEVTCMP register
- Any device Reset

15.15 PWM Operation During CPU Sleep Mode

The Fault A and Fault B input pins have the ability to wake the CPU from Sleep mode. The PWM module generates an interrupt if either of the Fault pins is driven low while in Sleep.

15.16 PWM Operation During CPU Idle Mode

The PTCON SFR contains a PTSIDL control bit. This bit determines if the PWM module will continue to operate or stop when the device enters Idle mode. If PTSIDL = 0, the module will continue to operate. If PTSIDL = 1, the module will stop operation as long as the CPU remains in Idle mode.

NOTES:

TABLE 19-1: CAN1 REGISTER MAP (CONTINUED)

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
C1TX1B1	0356	Transmit Buffer 1 Byte 1								Transmit Buffer 1 Byte 0								uuuu uuuu uuuu uuuu
C1TX1B2	0358	Transmit Buffer 1 Byte 3								Transmit Buffer 1 Byte 2								uuuu uuuu uuuu uuuu
C1TX1B3	035A	Transmit Buffer 1 Byte 5								Transmit Buffer 1 Byte 4								uuuu uuuu uuuu uuuu
C1TX1B4	035C	Transmit Buffer 1 Byte 7								Transmit Buffer 1 Byte 6								uuuu uuuu uuuu uuuu
C1TX1CON	035E	—	—	—	—	—	—	—	—	—	TXABT	TXLARB	TXERR	TXREQ	—	TXPRI<1:0>		0000 0000 0000 0000
C1TX0SID	0360	Transmit Buffer 0 Standard Identifier <10:6>					—	—	—	Transmit Buffer 0 Standard Identifier <5:0>						SRR	TXIDE	uuuu u000 uuuu uuuu
C1TX0EID	0362	Transmit Buffer 0 Extended Identifier <17:14>				—	—	—	—	Transmit Buffer 0 Extended Identifier <13:6>								uuuu 0000 uuuu uuuu
C1TX0DLC	0364	Transmit Buffer 0 Extended Identifier <5:0>						TXRTR	TXRB1	TXRB0	DLC<3:0>				—	—	—	uuuu uuuu uuuu u000
C1TX0B1	0366	Transmit Buffer 0 Byte 1								Transmit Buffer 0 Byte 0								uuuu uuuu uuuu uuuu
C1TX0B2	0368	Transmit Buffer 0 Byte 3								Transmit Buffer 0 Byte 2								uuuu uuuu uuuu uuuu
C1TX0B3	036A	Transmit Buffer 0 Byte 5								Transmit Buffer 0 Byte 4								uuuu uuuu uuuu uuuu
C1TX0B4	036C	Transmit Buffer 0 Byte 7								Transmit Buffer 0 Byte 6								uuuu uuuu uuuu uuuu
C1TX0CON	036E	—	—	—	—	—	—	—	—	—	TXABT	TXLARB	TXERR	TXREQ	—	TXPRI<1:0>		0000 0000 0000 0000
C1RX1SID	0370	—	—	—	Receive Buffer 1 Standard Identifier <10:0>											SRR	RXIDE	000u uuuu uuuu uuuu
C1RX1EID	0372	—	—	—	—	Receive Buffer 1 Extended Identifier <17:6>												0000 uuuu uuuu uuuu
C1RX1DLC	0374	Receive Buffer 1 Extended Identifier <5:0>						RXRTR	RXRB1	—	—	—	RXRB0	DLC<3:0>				uuuu uuuu 000u uuuu
C1RX1B1	0376	Receive Buffer 1 Byte 1								Receive Buffer 1 Byte 0								uuuu uuuu uuuu uuuu
C1RX1B2	0378	Receive Buffer 1 Byte 3								Receive Buffer 1 Byte 2								uuuu uuuu uuuu uuuu
C1RX1B3	037A	Receive Buffer 1 Byte 5								Receive Buffer 1 Byte 4								uuuu uuuu uuuu uuuu
C1RX1B4	037C	Receive Buffer 1 Byte 7								Receive Buffer 1 Byte 6								uuuu uuuu uuuu uuuu
C1RX1CON	037E	—	—	—	—	—	—	—	—	RXFUL	—	—	—	RXRTRRO	FILHIT<2:0>			0000 0000 0000 0000
C1RX0SID	0380	—	—	—	Receive Buffer 0 Standard Identifier <10:0>											SRR	RXIDE	000u uuuu uuuu uuuu
C1RX0EID	0382	—	—	—	—	Receive Buffer 0 Extended Identifier <17:6>												0000 uuuu uuuu uuuu
C1RX0DLC	0384	Receive Buffer 0 Extended Identifier <5:0>						RXRTR	RXRB1	—	—	—	RXRB0	DLC<3:0>				uuuu uuuu 000u uuuu
C1RX0B1	0386	Receive Buffer 0 Byte 1								Receive Buffer 0 Byte 0								uuuu uuuu uuuu uuuu
C1RX0B2	0388	Receive Buffer 0 Byte 3								Receive Buffer 0 Byte 2								uuuu uuuu uuuu uuuu
C1RX0B3	038A	Receive Buffer 0 Byte 5								Receive Buffer 0 Byte 4								uuuu uuuu uuuu uuuu
C1RX0B4	038C	Receive Buffer 0 Byte 7								Receive Buffer 0 Byte 6								uuuu uuuu uuuu uuuu
C1RX0CON	038E	—	—	—	—	—	—	—	—	RXFUL	—	—	—	RXRTRRO	DBEN	JTOFF	FILHIT0	0000 0000 0000 0000
C1CTRL	0390	CANCAP	—	CSIDLE	ABAT	CANCKS	REQOP<2:0>			OPMODE<2:0>			—	ICODE<2:0>			—	0000 0100 1000 0000
C1CFG1	0392	—	—	—	—	—	—	—	—	SJW<1:0>		BRP<5:0>						0000 0000 0000 0000
C1CFG2	0394	—	WAKFIL	—	—	—	SEG2PH<2:0>			SEG2PHTS	SAM	SEG1PH<2:0>			PRSEG<2:0>			0u00 0uuu uuuu uuuu
C1INTF	0396	RX0OVR	RX1OVR	TXBO	TXEP	RXEP	TXWAR	RXWAR	EWARN	IVRIF	WAKIF	ERRIF	TX2IF	TX1IF	TX0IF	RX1IF	RX0IF	0000 0000 0000 0000
C1INTE	0398	—	—	—	—	—	—	—	—	IVRIE	WAKIE	ERRIE	TX2IE	TX1IE	TX0IE	RX1E	RX0IE	0000 0000 0000 0000
C1EC	039A	Transmit Error Count Register								Receive Error Count Register								0000 0000 0000 0000

Legend: u = uninitialized bit

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

20.4 Programming the Start of Conversion Trigger

The conversion trigger will terminate acquisition and start the requested conversions.

The SSRC<2:0> bits select the source of the conversion trigger.

The SSRC bits provide for up to 5 alternate sources of conversion trigger.

When SSRC<2:0> = 000, the conversion trigger is under software control. Clearing the SAMP bit will cause the conversion trigger.

When SSRC<2:0> = 111 (Auto-Start mode), the conversion trigger is under ADC clock control. The SAMC bits select the number of ADC clocks between the start of acquisition and the start of conversion. This provides the fastest conversion rates on multiple channels. SAMC must always be at least 1 clock cycle.

Other trigger sources can come from timer modules, Motor Control PWM module, or external interrupts.

Note: To operate the ADC at the maximum specified conversion speed, the Auto Convert Trigger option should be selected (SSRC = 111) and the Auto Sample Time bits should be set to 1 TAD (SAMC = 00001). This configuration will give a total conversion period (sample + convert) of 13 TAD.

The use of any other conversion trigger will result in additional TAD cycles to synchronize the external event to the ADC.

20.5 Aborting a Conversion

Clearing the ADON bit during a conversion will abort the current conversion and stop the sampling sequencing. The ADCBUF will not be updated with the partially completed A/D conversion sample. That is, the ADCBUF will continue to contain the value of the last completed conversion (or the last value written to the ADCBUF register).

If the clearing of the ADON bit coincides with an auto start, the clearing has a higher priority.

After the A/D conversion is aborted, a 2 TAD wait is required before the next sampling may be started by setting the SAMP bit.

If sequential sampling is specified, the A/D will continue at the next sample pulse which corresponds with the next channel converted. If simultaneous sampling is specified, the ADC will continue with the next multi-channel group conversion sequence.

20.6 Selecting the A/D Conversion Clock

The A/D conversion requires 12 TAD. The source of the A/D conversion clock is software selected using a six bit counter. There are 64 possible options for TAD.

EQUATION 20-1: A/D CONVERSION CLOCK

$$T_{AD} = T_{CY} * (0.5 * (ADCS<5:0> + 1))$$

$$ADCS<5:0> = 2 \frac{T_{AD}}{T_{CY}} - 1$$

The internal RC oscillator is selected by setting the ADRC bit.

For correct A/D conversions, the A/D conversion clock (TAD) must be selected to ensure a minimum TAD time of 83.33 nsec (for VDD = 5V). Refer to the **Section 24.0 "Electrical Characteristics"** for minimum TAD under other operating conditions.

Example 20-1 shows a sample calculation for the ADCS<5:0> bits, assuming a device operating speed of 30 MIPS.

EXAMPLE 20-1: A/D CONVERSION CLOCK CALCULATION

$T_{AD} = 154$ nsec
$T_{CY} = 33$ nsec (30 MIPS)

$$ADCS<5:0> = 2 \frac{T_{AD}}{T_{CY}} - 1$$
$$= 2 \bullet \frac{154 \text{ nsec}}{33 \text{ nsec}} - 1$$
$$= 8.33$$

Therefore,
Set ADCS<5:0> = 9

$$\text{Actual } T_{AD} = \frac{T_{CY}}{2} (ADCS<5:0> + 1)$$
$$= \frac{33 \text{ nsec}}{2} (9 + 1)$$
$$= 165 \text{ nsec}$$

20.13 Configuring Analog Port Pins

The use of the ADPCFG and TRIS registers control the operation of the A/D port pins. The port pins that are desired as analog inputs must have their corresponding TRIS bit set (input). If the TRIS bit is cleared (output), the digital output level (V_{OH} or V_{OL}) will be converted.

The A/D operation is independent of the state of the CH0SA<3:0>/CH0SB<3:0> bits and the TRIS bits.

When reading the PORT register, all pins configured as analog input channels will read as cleared.

Pins configured as digital inputs will not convert an analog input. Analog levels on any pin that is defined as a digital input (including the ANx pins), may cause the input buffer to consume current that exceeds the device specifications.

20.14 Connection Considerations

The analog inputs have diodes to V_{DD} and V_{SS} as ESD protection. This requires that the analog input be between V_{DD} and V_{SS}. If the input voltage exceeds this range by greater than 0.3V (either direction), one of the diodes becomes forward biased and it may damage the device if the input current specification is exceeded.

An external RC filter is sometimes added for anti-aliasing of the input signal. The R component should be selected to ensure that the sampling time requirements are satisfied. Any external components connected (via high impedance) to an analog input pin (capacitor, zener diode, etc.) should have very little leakage current at the pin.

TABLE 20-2: ADC REGISTER MAP

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

Legend: u = uninitialized bit

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

21.8 In-Circuit Debugger

When MPLAB ICD2 is selected as a Debugger, the In-Circuit Debugging functionality is enabled. This function allows simple debugging functions when used with MPLAB IDE. When the device has this feature enabled, some of the resources are not available for general use. These resources include the first 80 bytes of Data RAM and two I/O pins.

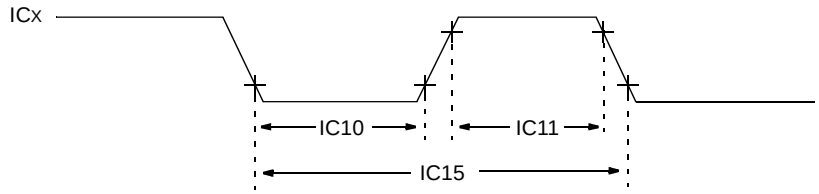
One of four pairs of Debug I/O pins may be selected by the user using configuration options in MPLAB IDE. These pin pairs are named EMUD/EMUC, EMUD1/EMUC1, EMUD2/EMUC2 and EMUD3/EMUC3.

In each case, the selected EMUD pin is the Emulation/Debug Data line, and the EMUC pin is the Emulation/Debug Clock line. These pins will interface to the MPLAB ICD 2 module available from Microchip. The selected pair of Debug I/O pins is used by MPLAB ICD 2 to send commands and receive responses, as well as to send and receive data. To use the In-Circuit Debugger function of the device, the design must implement ICSP connections to $\overline{MCLR}$, V_{DD}, V_{SS}, PGC, PGD and the selected EMUDx/EMUCx pin pair.

This gives rise to two possibilities:

1. If EMUD/EMUC is selected as the Debug I/O pin pair, then only a 5-pin interface is required, as the EMUD and EMUC pin functions are multiplexed with the PGD and PGC pin functions in all dsPIC30F devices.
2. If EMUD1/EMUC1, EMUD2/EMUC2 or EMUD3/EMUC3 is selected as the Debug I/O pin pair, then a 7-pin interface is required, as the EMUDx/EMUCx pin functions (x = 1, 2 or 3) are not multiplexed with the PGD and PGC pin functions.

FIGURE 24-10: INPUT CAPTURE (CAPx) TIMING CHARACTERISTICS



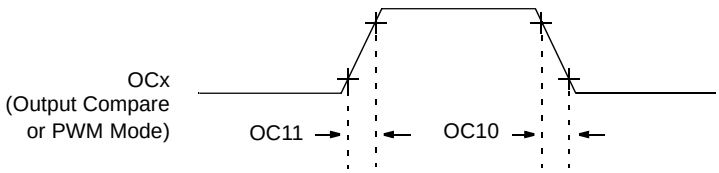
Note: Refer to Figure 24-3 for load conditions.

TABLE 24-27: INPUT CAPTURE TIMING REQUIREMENTS

AC CHARACTERISTICS				Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended			
Param No.	**Symbol**	**Characteristic(1)**		**Min**	**Max**	**Units**	**Conditions**
IC10	TccL	ICx Input Low Time	No Prescaler	0.5 TCY + 20	—	ns	
			With Prescaler	10	—	ns	
IC11	TccH	ICx Input High Time	No Prescaler	0.5 TCY + 20	—	ns	
			With Prescaler	10	—	ns	
IC15	TccP	ICx Input Period		(2 TCY + 40)/N	—	ns	N = prescale value (1, 4, 16)

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

FIGURE 24-11: OUTPUT COMPARE MODULE (OCx) TIMING CHARACTERISTICS



Note: Refer to Figure 24-3 for load conditions.

TABLE 24-28: OUTPUT COMPARE MODULE TIMING REQUIREMENTS

AC CHARACTERISTICS			Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	**Symbol**	**Characteristic(1)**	**Min**	**Typ(2)**	**Max**	**Units**	**Conditions**
OC10	TccF	OCx Output Fall Time	—	—	—	ns	See parameter D032
OC11	TccR	OCx Output Rise Time	—	—	—	ns	See parameter D031

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

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

TABLE 24-40: 10-BIT HIGH-SPEED ADC MODULE SPECIFICATIONS (CONTINUED)

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**
DC Accuracy (Continued)							
AD24	EOFF	Offset Error	±1	±2	±3	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 5V
AD24A	EOFF	Offset Error	±1	±2	±3	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD25	—	Monotonicity(2)	—	—	—	—	Guaranteed
Dynamic Performance							
AD30	THD	Total Harmonic Distortion	—	-64	-67	dB	—
AD31	SINAD	Signal to Noise and Distortion	—	57	58	dB	—
AD32	SFDR	Spurious Free Dynamic Range	—	67	71	dB	—
AD33	FNYQ	Input Signal Bandwidth	—	—	500	kHz	—
AD34	ENOB	Effective Number of Bits	9.29	9.41	—	bits	—

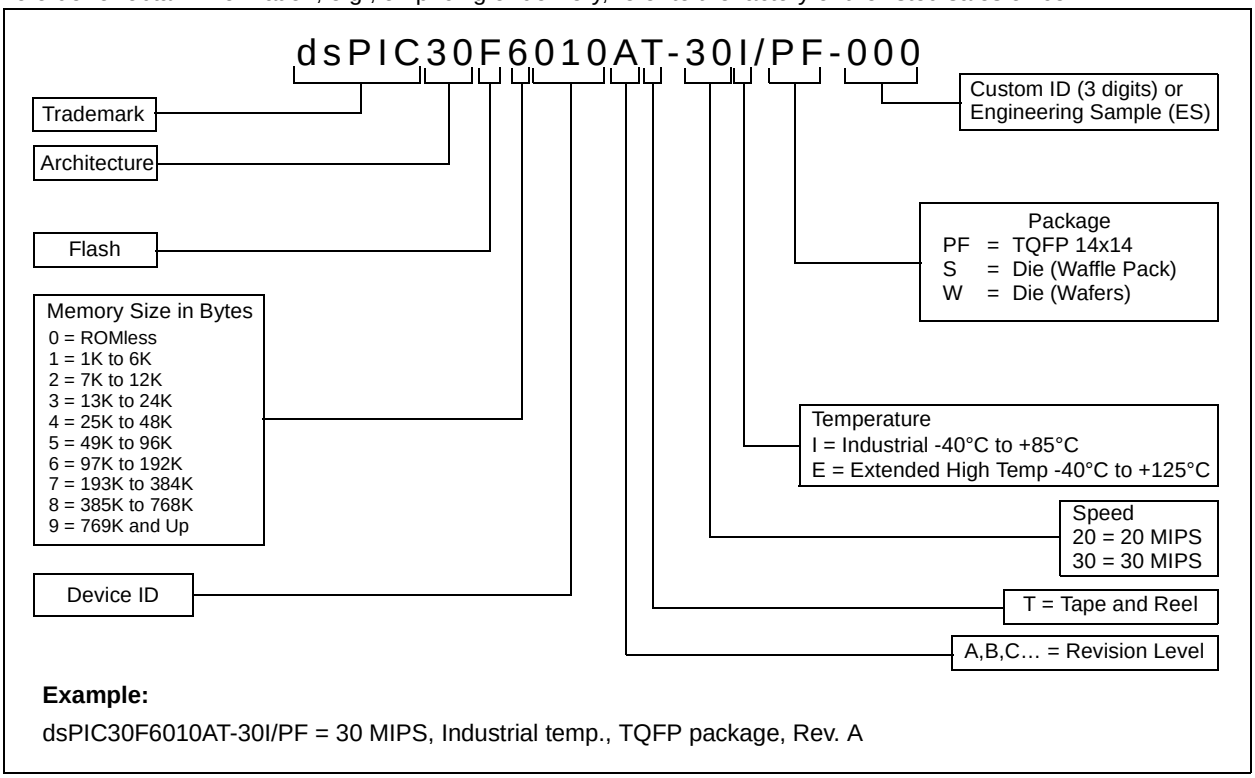
Note 1: These parameters are characterized but not tested in manufacturing..

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

3: Measurements taken with external VREF+ and VREF- used as the ADC voltage reference.

PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.



Example:

dsPIC30F6010AT-30I/PF = 30 MIPS, Industrial temp., TQFP package, Rev. A

WORLDWIDE SALES AND SERVICE

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://support.microchip.com
Web Address:
www.microchip.com

Atlanta
Alpharetta, GA
Tel: 770-640-0034
Fax: 770-640-0307

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Kokomo
Kokomo, IN
Tel: 765-864-8360
Fax: 765-864-8387

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

Santa Clara
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

Toronto
Mississauga, Ontario,
Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Habour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8528-2100
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Fuzhou
Tel: 86-591-8750-3506
Fax: 86-591-8750-3521

China - Hong Kong SAR
Tel: 852-2401-1200
Fax: 852-2401-3431

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

China - Shunde
Tel: 86-757-2839-5507
Fax: 86-757-2839-5571

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7250
Fax: 86-29-8833-7256

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-4182-8400
Fax: 91-80-4182-8422

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

Japan - Yokohama
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

Korea - Gumi
Tel: 82-54-473-4301
Fax: 82-54-473-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Penang
Tel: 60-4-646-8870
Fax: 60-4-646-5086

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-572-9526
Fax: 886-3-572-6459

Taiwan - Kaohsiung
Tel: 886-7-536-4818
Fax: 886-7-536-4803

Taiwan - Taipei
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

10/19/06