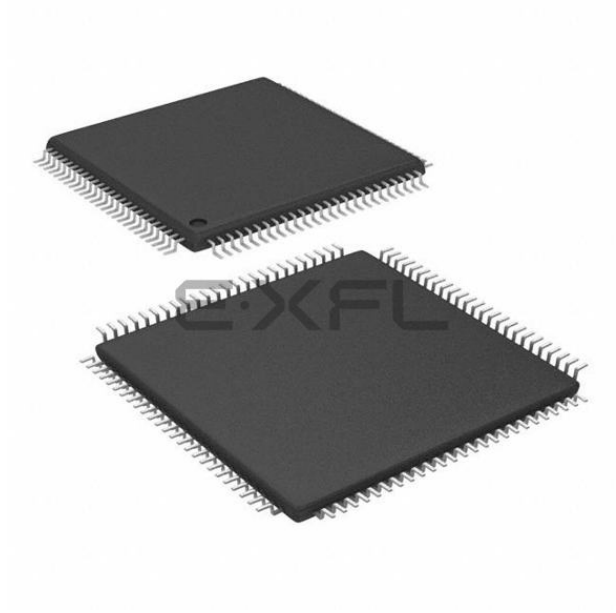
Microchip Technology - DSPIC33FJ256MC510AT-I/PF Datasheet



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	40 MIPs
Connectivity	CANbus, I²C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, Motor Control PWM, POR, PWM, QEI, WDT
Number of I/O	85
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	16K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 24x10/12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-TQFP
Supplier Device Package	100-TQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/dspic33fj256mc510at-i-pf

3.6.2.4 Data Space Write Saturation

In addition to adder/subtracter saturation, writes to data space can also be saturated – but without affecting the contents of the source accumulator. The data space write saturation logic block accepts a 16-bit, 1.15 fractional value from the round logic block as its input, together with overflow status from the original source (accumulator) and the 16-bit round adder. These inputs are combined and used to select the appropriate 1.15 fractional value as output to write to data space memory.

If the SATDW bit in the CORCON register is set, data (after rounding or truncation) is tested for overflow and adjusted accordingly. For input data greater than 0x007FFF, data written to memory is forced to the maximum positive 1.15 value, 0x7FFF. For input data less than 0xFF8000, data written to memory is forced to the maximum negative 1.15 value, 0x8000. The Most Significant bit of the source (bit 39) is used to determine the sign of the operand being tested.

If the SATDW bit in the CORCON register is not set, the input data is always passed through unmodified under all conditions.

3.6.3 BARREL SHIFTER

The barrel shifter is capable of performing up to 16-bit arithmetic or logic right shifts, or up to 16-bit left shifts in a single cycle. The source can be either of the two DSP accumulators or the X bus (to support multi-bit shifts of register or memory data).

The shifter requires a signed binary value to determine both the magnitude (number of bits) and direction of the shift operation. A positive value shifts the operand right. A negative value shifts the operand left. A value of '0' does not modify the operand.

The barrel shifter is 40 bits wide, thereby obtaining a 40-bit result for DSP shift operations and a 16-bit result for MCU shift operations. Data from the X bus is presented to the barrel shifter between bit positions 16 to 31 for right shifts and between bit positions 0 to 16 for left shifts.

TABLE 4-7: INPUT CAPTURE REGISTER MAP

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

Legend: x = unknown value on Reset, — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

TABLE 4-36: FUNDAMENTAL ADDRESSING MODES SUPPORTED

Addressing Mode	Description
File Register Direct	The address of the file register is specified explicitly.
Register Direct	The contents of a register are accessed directly.
Register Indirect	The contents of Wn forms the EA.
Register Indirect Post-Modified	The contents of Wn forms the EA. Wn is post-modified (incremented or decremented) by a constant value.
Register Indirect Pre-Modified	Wn is pre-modified (incremented or decremented) by a signed constant value to form the EA.
Register Indirect with Register Offset	The sum of Wn and Wb forms the EA.
Register Indirect with Literal Offset	The sum of Wn and a literal forms the EA.

4.3.3 MOVE AND ACCUMULATOR INSTRUCTIONS

Move instructions and the DSP accumulator class of instructions provide a greater degree of addressing flexibility than other instructions. In addition to the Addressing modes supported by most MCU instructions, move and accumulator instructions also support Register Indirect with Register Offset Addressing mode, also referred to as Register Indexed mode.

Note: For the MOV instructions, the addressing mode specified in the instruction can differ for the source and destination EA. However, the 4-bit Wb (register offset) field is shared between both source and destination (but typically only used by one).

In summary, the following addressing modes are supported by move and accumulator instructions:

- Register Direct
- Register Indirect
- Register Indirect Post-modified
- Register Indirect Pre-modified
- Register Indirect with Register Offset (Indexed)
- Register Indirect with Literal Offset
- 8-Bit Literal
- 16-Bit Literal

Note: Not all instructions support all the addressing modes given above. Individual instructions may support different subsets of these addressing modes.

4.3.4 MAC INSTRUCTIONS

The dual source operand DSP instructions (CLR, ED, EDAC, MAC, MPY, MPY.N, MOVSAC and MSC), also referred to as MAC instructions, utilize a simplified set of addressing modes to allow the user to effectively manipulate the Data Pointers through register indirect tables.

The 2-source operand prefetch registers must be members of the set {W8, W9, W10, W11}. For data reads, W8 and W9 are always directed to the X RAGU, and W10 and W11 will always be directed to the Y AGU. The Effective Addresses generated (before and after modification) must, therefore, be valid addresses within X data space for W8 and W9, and Y data space for W10 and W11.

Note: Register Indirect with Register Offset Addressing mode is only available for W9 (in X space) and W11 (in Y space).

In summary, the following addressing modes are supported by the MAC class of instructions:

- Register Indirect
- Register Indirect Post-Modified by 2
- Register Indirect Post-Modified by 4
- Register Indirect Post-Modified by 6
- Register Indirect with Register Offset (Indexed)

4.3.5 OTHER INSTRUCTIONS

Besides the various addressing modes outlined above, some instructions use literal constants of various sizes. For example, BRA (branch) instructions use 16-bit signed literals to specify the branch destination directly, whereas the DISI instruction uses a 14-bit unsigned literal field. In some instructions, such as ADD Acc, the source of an operand or result is implied by the opcode itself. Certain operations, such as NOP, do not have any operands.

4.4 Modulo Addressing

Modulo Addressing mode is a method of providing an automated means to support circular data buffers using hardware. The objective is to remove the need for software to perform data address boundary checks when executing tightly looped code, as is typical in many DSP algorithms.

5.0 FLASH PROGRAM MEMORY

Note 1: This data sheet summarizes the features of the dsPIC33FJXXXMCX06A/X08A/X10A family of devices. However, it is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **Section 5. "Flash Programming"** (DS70191) in the *"dsPIC33F/PIC24H Family Reference Manual"*, which is available from the Microchip web site (www.microchip.com).

2: Some registers and associated bits described in this section may not be available on all devices. Refer to **Section 4.0 "Memory Organization"** in this data sheet for device-specific register and bit information.

The dsPIC33FJXXXMCX06A/X08A/X10A devices contain internal Flash program memory for storing and executing application code. The memory is readable, writable and erasable during normal operation over the entire VDD range.

Flash memory can be programmed in two ways:

1. In-Circuit Serial Programming™ (ICSP™) programming capability
2. Run-Time Self-Programming (RTSP)

ICSP allows a dsPIC33FJXXXMCX06A/X08A/X10A device to be serially programmed while in the end application circuit. This is simply done with two lines for programming clock and programming data (one of the alternate programming pin pairs: PGECx/PGEDx), and three other lines for power (VDD), ground (VSS) and Master Clear (MCLR). This allows customers to manufacture boards with unprogrammed devices and then program the digital signal controller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

RTSP is accomplished using TBLRD (table read) and TBLWT (table write) instructions. With RTSP, the user can write program memory data by blocks (or 'rows') of 64 instructions (192 bytes) at a time or by single program memory word; the user can erase program memory in blocks or 'pages' of 512 instructions (1536 bytes) at a time.

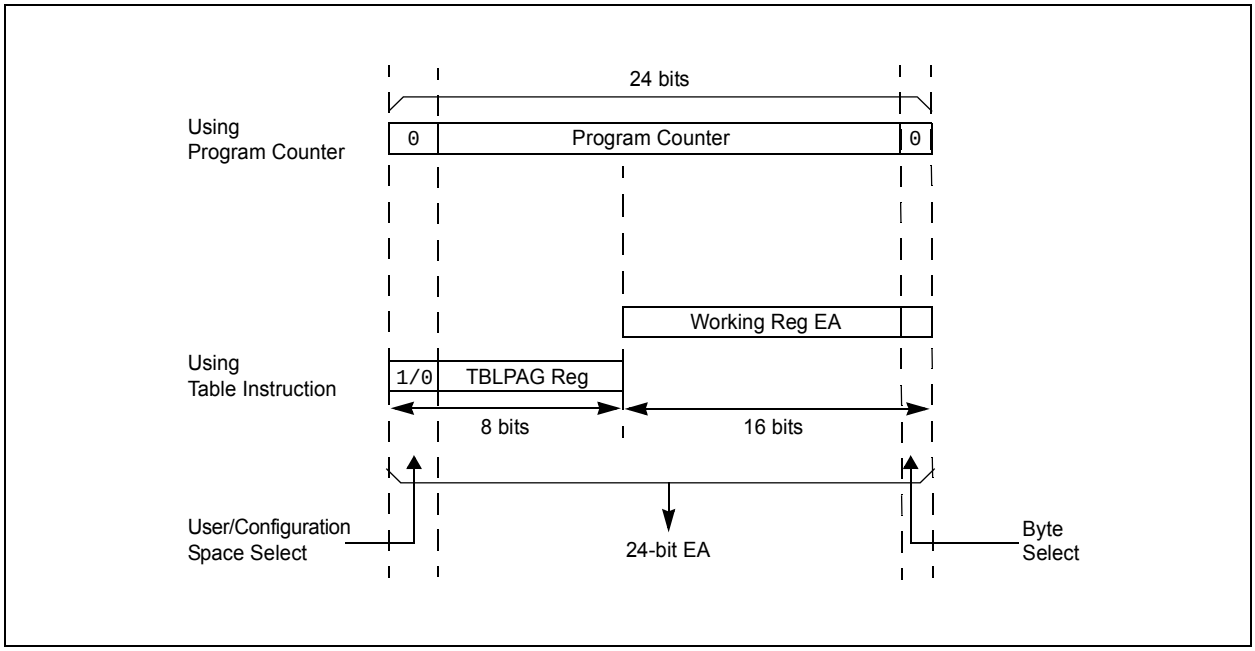
5.1 Table Instructions and Flash Programming

Regardless of the method used, all programming of Flash memory is done with the table read and table write instructions. These allow direct read and write access to the program memory space from the data memory while the device is in normal operating mode. The 24-bit target address in the program memory is formed using bits<7:0> of the TBLPAG register and the Effective Address (EA) from a W register specified in the table instruction, as shown in Figure 5-1.

The TBLRDL and TBLWTL instructions are used to read or write to bits<15:0> of program memory. TBLRDL and TBLWTL can access program memory in both Word and Byte modes.

The TBLRDH and TBLWTH instructions are used to read or write to bits<23:16> of program memory. TBLRDH and TBLWTH can also access program memory in Word or Byte mode.

FIGURE 5-1: ADDRESSING FOR TABLE REGISTERS



8.0 DIRECT MEMORY ACCESS (DMA)

Note 1: This data sheet summarizes the features of the dsPIC33FJXXXMCX06A/X08A/X10A family of devices. However, it is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **Section 22. "Direct Memory Access (DMA)"** (DS70182) in the *"dsPIC33F/PIC24H Family Reference Manual"*, which is available from the Microchip web site (www.microchip.com).

2: Some registers and associated bits described in this section may not be available on all devices. Refer to **Section 4.0 "Memory Organization"** in this data sheet for device-specific register and bit information.

Direct Memory Access (DMA) is a very efficient mechanism of copying data between peripheral SFRs (e.g., the UART Receive register and Input Capture 1 buffer) and buffers or variables stored in RAM, with minimal CPU intervention. The DMA controller can automatically copy entire blocks of data without requiring the user software to read or write the peripheral Special Function Registers (SFRs) every time a peripheral interrupt occurs. The DMA controller uses a dedicated bus for data transfers, and therefore, does not steal cycles from the code execution flow of the CPU. To exploit the DMA capability, the corresponding user buffers or variables must be located in DMA RAM.

The dsPIC33FJXXXMCX06A/X08A/X10A peripherals that can utilize DMA are listed in Table 8-1 along with their associated Interrupt Request (IRQ) numbers.

TABLE 8-1: PERIPHERALS WITH DMA SUPPORT

Peripheral	IRQ Number
INT0	0
Input Capture 1	1
Input Capture 2	5
Output Compare 1	2
Output Compare 2	6
Timer2	7
Timer3	8
SPI1	10
SPI2	33
UART1 Reception	11
UART1 Transmission	12
UART2 Reception	30
UART2 Transmission	31
ADC1	13
ADC2	21
ECAN1 Reception	34
ECAN1 Transmission	70
ECAN2 Reception	55
ECAN2 Transmission	71

The DMA controller features eight identical data transfer channels. Each channel has its own set of control and status registers. Each DMA channel can be configured to copy data, either from buffers stored in dual port DMA RAM to peripheral SFRs, or from peripheral SFRs to buffers in DMA RAM.

The DMA controller supports the following features:

- Word or byte-sized data transfers.
- Transfers from peripheral to DMA RAM or DMA RAM to peripheral.
- Indirect Addressing of DMA RAM locations with or without automatic post-increment.
- Peripheral Indirect Addressing – In some peripherals, the DMA RAM read/write addresses may be partially derived from the peripheral.
- One-Shot Block Transfers – Terminating DMA transfer after one block transfer.
- Continuous Block Transfers – Reloading DMA RAM buffer start address after every block transfer is complete.
- Ping-Pong Mode – Switching between two DMA RAM start addresses between successive block transfers, thereby filling two buffers alternately.
- Automatic or manual initiation of block transfers.
- Each channel can select from 20 possible sources of data sources or destinations.

For each DMA channel, a DMA interrupt request is generated when a block transfer is complete. Alternatively, an interrupt can be generated when half of the block has been filled.

REGISTER 17-1: QEIxCON: QEIx CONTROL REGISTER (CONTINUED)

bit 4-3 **TQCKPS<1:0>:** Timer Input Clock Prescale Select bits
11 = 1:256 prescale value
10 = 1:64 prescale value
01 = 1:8 prescale value
00 = 1:1 prescale value
(Prescaler utilized for 16-Bit Timer mode only.)

bit 2 **POSRES:** Position Counter Reset Enable bit
1 = Index pulse resets position counter
0 = Index pulse does not reset position counter
(Bit only applies when QEIM<2:0> = 100 or 110.)

bit 1 **TQCS:** Timer Clock Source Select bit
1 = External clock from QEA pin (on the rising edge)
0 = Internal clock (TCY)

bit 0 **UPDN_SRC:** Position Counter Direction Selection Control bit(1)
1 = QEB pin state defines position counter direction
0 = Control/status bit, UPDN (QEICON<11>), defines Position Counter (POSxCNT) direction

Note 1: When configured for QEI mode, the control bit is a ‘don’t care’.

REGISTER 18-2: SPIxCON1: SPIx CONTROL REGISTER 1

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	DISSCK	DISSDO	MODE16	SMP	CKE(1)
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SSEN(3)	CKP	MSTEN	SPRE<2:0>(2)			PPRE<1:0>(2)	
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-13 **Unimplemented:** Read as '0'

bit 12 **DISSCK:** Disable SCKx Pin bit (SPI Master modes only)
1 = Internal SPI clock is disabled; pin functions as I/O
0 = Internal SPI clock is enabled

bit 11 **DISSDO:** Disable SDOx Pin bit
1 = SDOx pin is not used by module; pin functions as I/O
0 = SDOx pin is controlled by the module

bit 10 **MODE16:** Word/Byte Communication Select bit
1 = Communication is word-wide (16 bits)
0 = Communication is byte-wide (8 bits)

bit 9 **SMP:** SPIx Data Input Sample Phase bit
Master mode:
1 = Input data sampled at end of data output time
0 = Input data sampled at middle of data output time
Slave mode:
SMP must be cleared when SPIx is used in Slave mode.

bit 8 **CKE:** SPIx Clock Edge Select bit(1)
1 = Serial output data changes on transition from active clock state to Idle clock state (see bit 6)
0 = Serial output data changes on transition from Idle clock state to active clock state (see bit 6)

bit 7 **SSEN:** Slave Select Enable bit (Slave mode)(3)
1 = $\overline{SSx}$ pin used for Slave mode
0 = $\overline{SSx}$ pin not used by module. Pin controlled by port function.

bit 6 **CKP:** Clock Polarity Select bit
1 = Idle state for clock is a high level; active state is a low level
0 = Idle state for clock is a low level; active state is a high level

bit 5 **MSTEN:** Master Mode Enable bit
1 = Master mode
0 = Slave mode

Note 1: The CKE bit is not used in the Framed SPI modes. The user should program this bit to '0' for the Framed SPI modes (FRMEN = 1).
2: Do not set both the primary and secondary prescalers to a value of 1:1.
3: This bit must be cleared when FRMEN = 1.

REGISTER 18-3: SPIxCON2: SPIx CONTROL REGISTER 2

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0
FRMEN	SPIFSD	FRMPOL	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	U-0
—	—	—	—	—	—	FRMDLY	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **FRMEN:** Framed SPIx Support bit
1 = Framed SPIx support enabled ($\overline{SSx}$ pin used as frame Sync pulse input/output)
0 = Framed SPIx support disabled

bit 14 **SPIFSD:** Frame Sync Pulse Direction Control bit
1 = Frame Sync pulse input (slave)
0 = Frame Sync pulse output (master)

bit 13 **FRMPOL:** Frame Sync Pulse Polarity bit
1 = Frame Sync pulse is active-high
0 = Frame Sync pulse is active-low

bit 12-2 **Unimplemented:** Read as '0'

bit 1 **FRMDLY:** Frame Sync Pulse Edge Select bit
1 = Frame Sync pulse coincides with first bit clock
0 = Frame Sync pulse precedes first bit clock

bit 0 **Unimplemented:** This bit must not be set to '1' by the user application.

REGISTER 19-3: I2CxMSK: I2Cx SLAVE MODE ADDRESS MASK REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	AMSK9	AMSK8
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
AMSK7	AMSK6	AMSK5	AMSK4	AMSK3	AMSK2	AMSK1	AMSK0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-10 **Unimplemented:** Read as ‘0’

bit 9-0 **AMSKx:** Mask for Address bit x Select bits

1 = Enable masking for bit x of incoming message address; bit match not required in this position
0 = Disable masking for bit x; bit match required in this position

REGISTER 21-1: CiCTRL1: ECAN™ CONTROL REGISTER 1

U-0	U-0	R/W-0	R/W-0	r-0	R/W-1	R/W-0	R/W-0
—	—	CSIDL	ABAT	—	REQOP<2:0>		
bit 15							bit 8

R-1	R-0	R-0	U-0	R/W-0	U-0	U-0	R/W-0
OPMODE<2:0>			—	CANCAP	—	—	WIN
bit 7							bit 0

Legend:	r = Reserved bit		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as '0'

bit 13 **CSIDL:** Stop in Idle Mode bit
1 = Discontinue module operation when device enters Idle mode
0 = Continue module operation in Idle mode

bit 12 **ABAT:** Abort All Pending Transmissions bit
1 = Signal all transmit buffers to abort transmission
0 = Module will clear this bit when all transmissions are aborted

bit 11 **Reserved:** Do no use

bit 10-8 **REQOP<2:0>:** Request Operation Mode bits
111 = Set Listen All Messages mode
110 = Reserved – do not use
101 = Reserved – do not use
100 = Set Configuration mode
011 = Set Listen Only Mode
010 = Set Loopback mode
001 = Set Disable mode
000 = Set Normal Operation mode

bit 7-5 **OPMODE<2:0>:** Operation Mode bits
111 = Module is in Listen All Messages mode
110 = Reserved
101 = Reserved
100 = Module is in Configuration mode
011 = Module is in Listen Only mode
010 = Module is in Loopback mode
001 = Module is in Disable mode
000 = Module is in Normal Operation mode

bit 4 **Unimplemented:** Read as '0'

bit 3 **CANCAP:** CAN Message Receive Timer Capture Event Enable bit
1 = Enable input capture based on CAN message receive
0 = Disable CAN capture

bit 2-1 **Unimplemented:** Read as '0'

bit 0 **WIN:** SFR Map Window Select bit
1 = Use filter window
0 = Use buffer window

REGISTER 21-19: CiFMSKSEL2: ECAN™ FILTER 15-8 MASK SELECTION REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
F15MSK<1:0>		F14MSK<1:0>		F13MSK<1:0>		F12MSK<1:0>	
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
F11MSK<1:0>		F10MSK<1:0>		F9MSK<1:0>		F8MSK<1:0>	
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-14 **F15MSK<1:0>:** Mask Source for Filter 15 bit
11 = Reserved; do not use
10 = Acceptance Mask 2 registers contain mask
01 = Acceptance Mask 1 registers contain mask
00 = Acceptance Mask 0 registers contain mask

bit 13-12 **F14MSK<1:0>:** Mask Source for Filter 14 bit
11 = Reserved; do not use
10 = Acceptance Mask 2 registers contain mask
01 = Acceptance Mask 1 registers contain mask
00 = Acceptance Mask 0 registers contain mask

bit 11-10 **F13MSK<1:0>:** Mask Source for Filter 13 bit
11 = Reserved; do not use
10 = Acceptance Mask 2 registers contain mask
01 = Acceptance Mask 1 registers contain mask
00 = Acceptance Mask 0 registers contain mask

bit 9-8 **F12MSK<1:0>:** Mask Source for Filter 12 bit
11 = Reserved; do not use
10 = Acceptance Mask 2 registers contain mask
01 = Acceptance Mask 1 registers contain mask
00 = Acceptance Mask 0 registers contain mask

bit 7-6 **F11MSK<1:0>:** Mask Source for Filter 11 bit
11 = Reserved; do not use
10 = Acceptance Mask 2 registers contain mask
01 = Acceptance Mask 1 registers contain mask
00 = Acceptance Mask 0 registers contain mask

bit 5-4 **F10MSK<1:0>:** Mask Source for Filter 10 bit
11 = Reserved; do not use
10 = Acceptance Mask 2 registers contain mask
01 = Acceptance Mask 1 registers contain mask
00 = Acceptance Mask 0 registers contain mask

bit 3-2 **F9MSK<1:0>:** Mask Source for Filter 9 bit
11 = Reserved; do not use
10 = Acceptance Mask 2 registers contain mask
01 = Acceptance Mask 1 registers contain mask
00 = Acceptance Mask 0 registers contain mask

bit 1-0 **F8MSK<1:0>:** Mask Source for Filter 8 bit
11 = Reserved; do not use
10 = Acceptance Mask 2 registers contain mask
01 = Acceptance Mask 1 registers contain mask
00 = Acceptance Mask 0 registers contain mask

22.4 ADC Helpful Tips

1. The SMPI<3:0> (AD1CON2<5:2>) control bits:
   a) Determine when the ADC interrupt flag is set and an interrupt is generated if enabled.
   b) When the CSCNA bit (AD1CON2<10>) is set to '1', determines when the ADC analog scan channel list defined in the AD1CSSL/ AD1CSSH registers starts over from the beginning.
   c) On devices without a DMA peripheral, determines when ADC result buffer pointer to ADC1BUF0-ADC1BUFF, gets reset back to the beginning at ADC1BUF0.
2. On devices without a DMA module, the ADC has 16 result buffers. ADC conversion results are stored sequentially in ADC1BUF0-ADC1BUFF regardless of which analog inputs are being used subject to the SMPI<3:0> bits (AD1CON2<5:2>) and the condition described in 1c above. There is no relationship between the ANx input being measured and which ADC buffer (ADC1BUF0-ADC1BUFF) that the conversion results will be placed in.
3. On devices with a DMA module, the ADC module has only 1 ADC result buffer, (i.e., ADC1BUF0), per ADC peripheral and the ADC conversion result must be read either by the CPU or DMA controller before the next ADC conversion is complete to avoid overwriting the previous value.
4. The DONE bit (AD1CON1<0>) is only cleared at the start of each conversion and is set at the completion of the conversion, but remains set indefinitely even through the next sample phase until the next conversion begins. If application code is monitoring the DONE bit in any kind of software loop, the user must consider this behavior because the CPU code execution is faster than the ADC. As a result, in manual sample mode, particularly where the users code is setting the SAMP bit (AD1CON1<1>), the DONE bit should also be cleared by the user application just before setting the SAMP bit.
5. On devices with two ADC modules, the ADCxPCFG registers for both ADC modules must be set to a logic '1' to configure a target I/O pin as a digital I/O pin. Failure to do so means that any alternate digital input function will always see only a logic '0' as the digital input buffer is held in Disable mode.

22.5 ADC Resources

Many useful resources related to ADC are provided on the main product page of the Microchip web site for the devices listed in this data sheet. This product page, which can be accessed using this link, contains the latest updates and additional information.

Note:	In the event you are not able to access the product page using the link above, enter this URL in your browser: http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en546066

22.5.1 KEY RESOURCES

- **Section 16. "Analog-to-Digital Converter (ADC)"** (DS70183)
- Code Samples
- Application Notes
- Software Libraries
- Webinars
- All related dsPIC33F/PIC24H Family Reference Manuals Sections
- Development Tools

REGISTER 22-6: ADxCHS0: ADCx INPUT CHANNEL 0 SELECT REGISTER

R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CH0NB	—	—	CH0SB<4:0>				
bit 15							bit 8

R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CH0NA	—	—	CH0SA<4:0>(1)				
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **CH0NB:** Channel 0 Negative Input Select for Sample B bit
Same definition as bit 7.

bit 14-13 **Unimplemented:** Read as '0'

bit 12-8 **CH0SB<4:0>:** Channel 0 Positive Input Select for Sample B bits
Same definition as bit<4:0>.

bit 7 **CH0NA:** Channel 0 Negative Input Select for Sample A bit
1 = Channel 0 negative input is AN1
0 = Channel 0 negative input is VREF-

bit 6-5 **Unimplemented:** Read as '0'

bit 4-0 **CH0SA<4:0>:** Channel 0 Positive Input Select for Sample A bits(1)
11111 = Channel 0 positive input is AN31
11110 = Channel 0 positive input is AN30
•
•
•
00010 = Channel 0 positive input is AN2
00001 = Channel 0 positive input is AN1
00000 = Channel 0 positive input is AN0

Note 1: ADC2 can only select AN0-AN15 as positive inputs.

TABLE 26-7: DC CHARACTERISTICS: POWER-DOWN CURRENT (I_{PD})

DC CHARACTERISTICS			Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended			
Parameter No.(3)	**Typical(2)**	**Max**	**Units**	**Conditions**		
Power-Down Current (I_{PD})(1)						
DC60d	50	200	µA	-40°C	3.3V	Base Power-Down Current(3)
DC60a	50	200	µA	+25°C		
DC60b	200	500	µA	+85°C		
DC60c	600	1000	µA	+125°C		
DC61d	8	13	µA	-40°C	3.3V	Watchdog Timer Current: ΔI_{WDT}(3)
DC61a	10	15	µA	+25°C		
DC61b	12	20	µA	+85°C		
DC61c	13	25	µA	+125°C		

Note 1: I_{PD} (Sleep) current is measured as follows:

- CPU core is off, oscillator is configured in EC mode and external clock active, OSC1 is driven with external square wave from rail-to-rail (EC clock overshoot/undershoot < 250 mV required)
- CLKO is configured as an I/O input pin in the Configuration word
- All I/O pins are configured as inputs and pulled to V_{SS}
- $\overline{MCLR}$ = V_{DD}, WDT and FSCM are disabled, all peripheral modules except the ADC are disabled (PMDx bits are all '1's). The following ADC settings are enabled for each ADC module (ADCx) prior to executing the PWRSAV instruction: ADON = 1, VCFG = 1, AD12B = 1 and ADxMD = 0.
- VREGS bit (RCON<8>) = 0 (i.e., core regulator is set to stand-by while the device is in Sleep mode)
- RTCC is disabled.
- JTAG is disabled

2: Data in the "Typ" column is at 3.3V, +25ºC unless otherwise stated.

3: The Watchdog Timer Current is the additional current consumed when the WDT module is enabled. This current should be added to the base I_{PD} current.

4: These currents are measured on the device containing the most memory in this family.

5: These parameters are characterized, but are not tested in manufacturing.

FIGURE 26-13: TIMERQ (QEI MODULE) EXTERNAL CLOCK TIMING CHARACTERISTICS

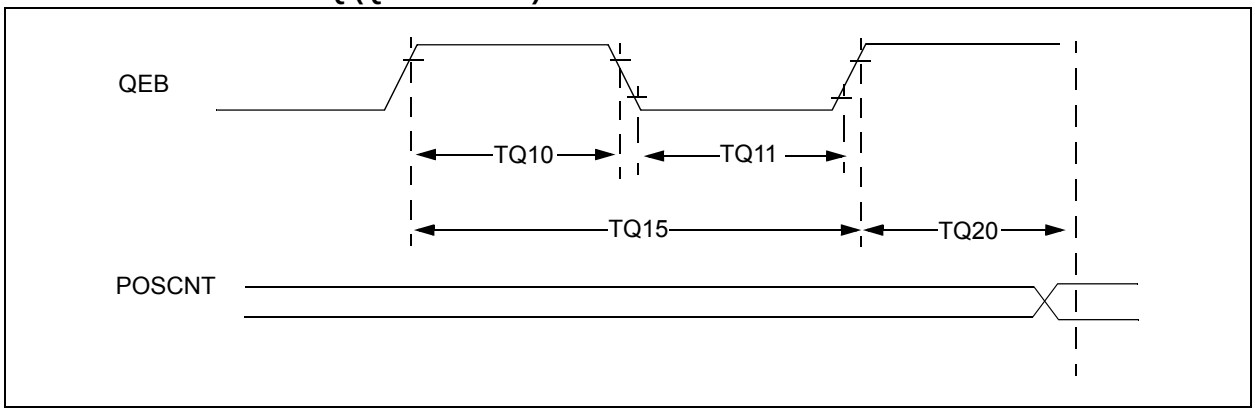


TABLE 26-31: QEI MODULE EXTERNAL CLOCK TIMING REQUIREMENTS

AC CHARACTERISTICS				Standard Operating Conditions: 3.0V to 3.6V (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**	**Max**	**Units**	**Conditions**
TQ10	TtQH	TQCK High Time	Synchronous, with prescaler	TCY + 20	—	—	ns	Must also meet parameter TQ15
TQ11	TtQL	TQCK Low Time	Synchronous, with prescaler	TCY + 20	—	—	ns	Must also meet parameter TQ15
TQ15	TtQP	TQCP Input Period	Synchronous, with prescaler	2 * TCY + 40	—	—	ns	—
TQ20	TCKEXTMRL	Delay from External TxCK Clock Edge to Timer Increment		0.5 TCY	—	1.5 TCY	—	—

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

FIGURE 26-26: CAN MODULE I/O TIMING CHARACTERISTICS

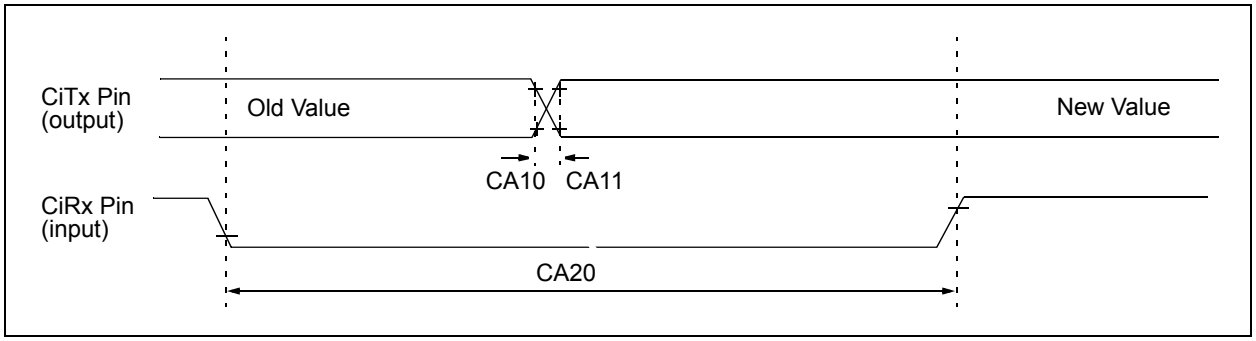
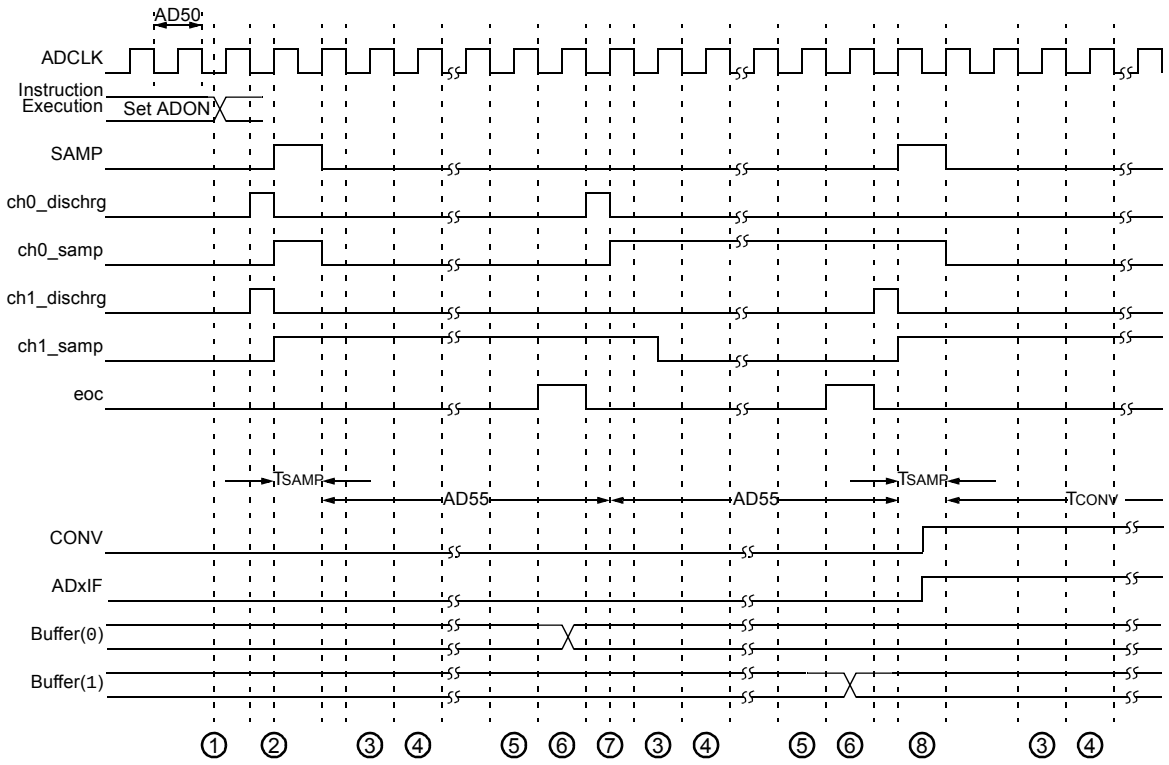


TABLE 26-42: ECAN™ TECHNOLOGY MODULE I/O TIMING REQUIREMENTS

AC CHARACTERISTICS			Standard Operating Conditions: 3.0V to 3.6V (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**	**Max**	**Units**	**Conditions**
CA10	TioF	Port Output Fall Time	—	—	—	ns	See parameter D032
CA11	TioR	Port Output Rise Time	—	—	—	ns	See parameter D031
CA20	Tcwf	Pulse Width to Trigger CAN Wake-up Filter	120	—	—	ns	—

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

FIGURE 26-29: ADC CONVERSION (10-BIT MODE) TIMING CHARACTERISTICS (CHPS<1:0> = 01, SIMSAM = 0, ASAM = 1, SSRC<2:0> = 111, SAMC<4:0> = 00001)



(1) – Software sets ADxCON. ADON to start AD operation.

(2) – Sampling starts after discharge period. TSAMP is described in **Section 16. "10/12-bit ADC with DMA"** in the *"dsPIC33F Family Reference Manual"*.

(3) – Convert bit 9.

(4) – Convert bit 8.

(5) – Convert bit 0.

(6) – One TAD for end of conversion.

(7) – Begin conversion of next channel.

(8) – Sample for time specified by SAMC<4:0>.

APPENDIX B: REVISION HISTORY

Revision A (May 2009)

This is the initial released version of the document.

Revision B (October 2009)

The revision includes the following global update:

- Added Note 2 to the shaded table that appears at the beginning of each chapter. This new note provides information regarding the availability of registers and their associated bits.

This revision also includes minor typographical and formatting changes throughout the data sheet text.

All other major changes are referenced by their respective section in the following table.

TABLE B-1: MAJOR SECTION UPDATES

Section Name	Update Description
"16-bit Digital Signal Controllers (up to 256 KB Flash and 30 KB SRAM) with Motor Control and Advanced Analog"	Added information on high temperature operation (see **"Operating Range:"**).
Section 11.0 "I/O Ports"	Changed the reference to digital-only pins to 5V tolerant pins in the second paragraph of **Section 11.2 "Open-Drain Configuration"**.
Section 20.0 "Universal Asynchronous Receiver Transmitter (UART)"	Updated the two baud rate range features to: 10 Mbps to 38 bps at 40 MIPS.
Section 22.0 "10-bit/12-bit Analog-to-Digital Converter (ADC)"	Updated the ADCx block diagram (see Figure 22-1).
Section 23.0 "Special Features"	Updated the second paragraph and removed the fourth paragraph in **Section 23.1 "Configuration Bits"**. Updated the Device Configuration Register Map (see Table 23-1).
Section 26.0 "Electrical Characteristics"	Updated the Absolute Maximum Ratings for high temperature and added Note 4. Updated Power-Down Current parameters DC60d, DC60a, DC60b, and DC60d (see Table 26-7). Added I2Cx Bus Data Timing Requirements (Master Mode) parameter IM51 (see Table 26-40). Updated the SPIx Module Slave Mode (CKE = 1) Timing Characteristics (see Figure 26-17). Updated the Internal LPRC Accuracy parameters (see Table 26-19). Updated the ADC Module Specifications (12-bit Mode) parameters AD23a, AD24a, AD23b, and AD24b (see Table 26-46). Updated the ADC Module Specifications (10-bit Mode) parameters AD23c, AD24c, AD23d, and AD24d (see Table 26-46).
Section 27.0 "High Temperature Electrical Characteristics"	Added new chapter with high temperature specifications.
"Product Identification System"	Added the "H" definition for high temperature.

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Trademarks

ISBN: 978-1-62076-343-8

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
═ ISO/TS 16949 ═

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.