

Welcome to [E-XFL.COM](https://www.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 Dual-Core
Speed	180MHz, 200MHz
Connectivity	CANbus, I ² C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, Motor Control PWM, POR, PWM, QEI, WDT
Number of I/O	27
Program Memory Size	88KB (88K x 8)
Program Memory Type	FLASH, PRAM
EEPROM Size	-
RAM Size	20K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 32x12b; D/A 4x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	36-UQFN Exposed Pad
Supplier Device Package	36-UQFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/dspic33ch64mp503-e-m5

dsPIC33CH128MP508 FAMILY

2.6 Oscillator Value Conditions on Device Start-up

If the PLL of the target device is enabled and configured for the device start-up oscillator, the maximum oscillator source frequency must be limited to a certain frequency (see **Section 6.0 “Oscillator with High-Frequency PLL”**) to comply with device PLL start-up conditions. This means that if the external oscillator frequency is outside this range, the application must start up in the FRC mode first. The default PLL settings after a POR with an oscillator frequency outside this range will violate the device operating speed.

Once the device powers up, the application firmware can initialize the PLL SFRs, CLKDIV and PLLFBD, to a suitable value, and then perform a clock switch to the Oscillator + PLL clock source. Note that clock switching must be enabled in the device Configuration Word.

2.7 Unused I/Os

Unused I/O pins should be configured as outputs and driven to a logic low state.

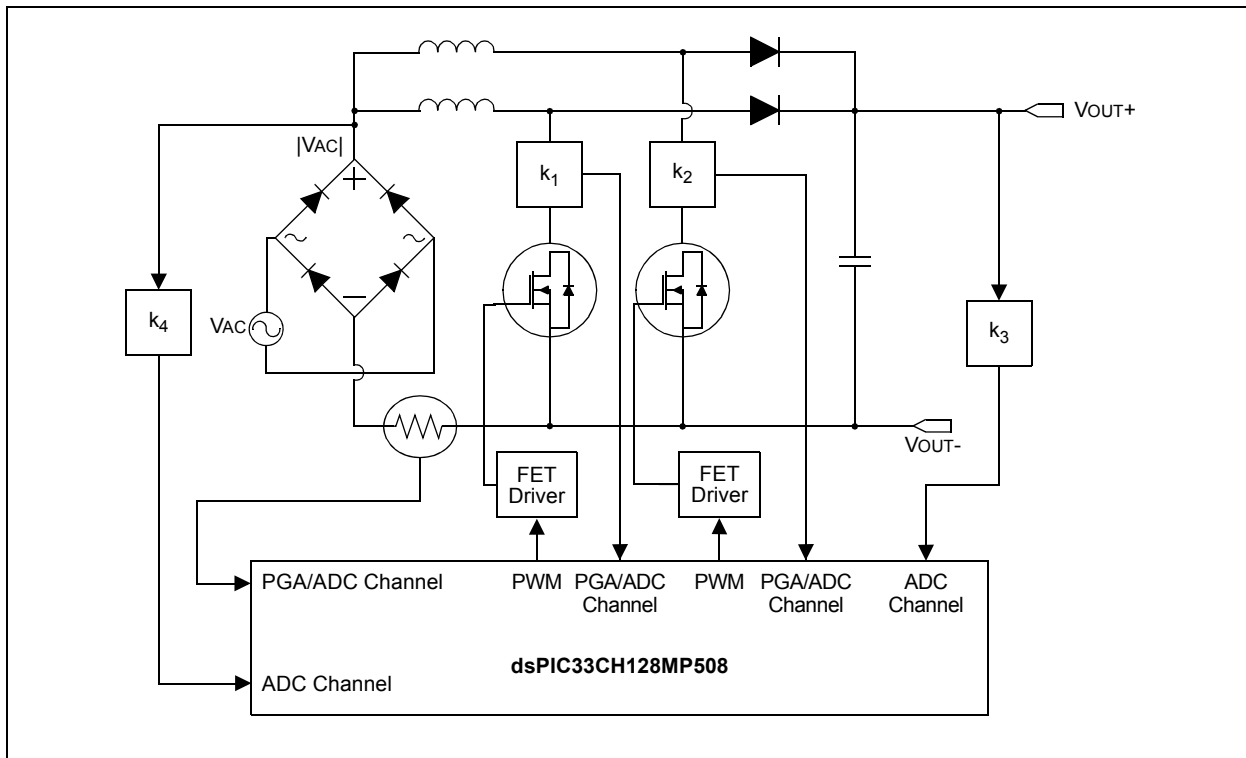
Alternatively, connect a 1k to 10k resistor between Vss and unused pins, and drive the output to logic low.

2.8 Targeted Applications

- Power Factor Correction (PFC):
 - Interleaved PFC
 - Critical Conduction PFC
 - Bridgeless PFC
- DC/DC Converters:
 - Buck, Boost, Forward, Flyback, Push-Pull
 - Half/Full-Bridge
 - Phase-Shift Full-Bridge
 - Resonant Converters
- DC/AC:
 - Half/Full-Bridge Inverter
 - Resonant Inverter
- Motor Control
 - BLDC
 - PMSM
 - SR
 - ACIM

Examples of typical application connections are shown in Figure 2-4 through Figure 2-6.

FIGURE 2-4: INTERLEAVED PFC



3.2.5.2 Extended X Data Space

The lower portion of the base address space range, between 0x0000 and 0x7FFF, is always accessible, regardless of the contents of the Data Space Read Page register. It is indirectly addressable through the register indirect instructions. It can be regarded as being located in the default EDS Page 0 (i.e., EDS address range of 0x000000 to 0x007FFF with the base address bit, EA<15> = 0, for this address range). However, Page 0 cannot be accessed through the upper 32 Kbytes, 0x8000 to 0xFFFF, of base Data Space in combination with DSRPAG = 0x00. Consequently, DSRPAG is initialized to 0x001 at Reset.

Note 1: DSRPAG should not be used to access Page 0. An EDS access with DSRPAG set to 0x000 will generate an address error trap.

2: Clearing the DSRPAG in software has no effect.

The remaining PSV pages are only accessible using the DSRPAG register in combination with the upper 32 Kbytes, 0x8000 to 0xFFFF, of the base address, where the base address bit, EA<15> = 1.

3.2.5.3 Software Stack

The W15 register serves as a dedicated Software Stack Pointer (SSP), and is automatically modified by exception processing, subroutine calls and returns; however, W15 can be referenced by any instruction in the same manner as all other W registers. This simplifies reading, writing and manipulating the Stack Pointer (for example, creating stack frames).

Note: To protect against misaligned stack accesses, W15<0> is fixed to '0' by the hardware.

W15 is initialized to 0x1000 during all Resets. This address ensures that the SSP points to valid RAM in all dsPIC33CH128MP508 devices and permits stack availability for non-maskable trap exceptions. These can occur before the SSP is initialized by the user software. You can reprogram the SSP during initialization to any location within Data Space.

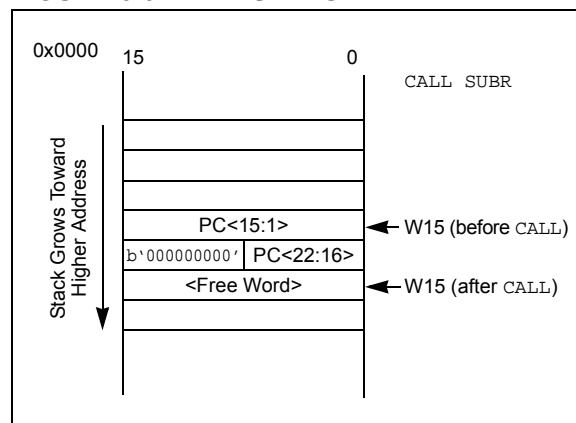
The Software Stack Pointer always points to the first available free word and fills the software stack, working from lower toward higher addresses. Figure 3-9 illustrates how it pre-decrements for a stack pop (read) and post-increments for a stack push (writes).

When the PC is pushed onto the stack, PC<15:0> are pushed onto the first available stack word, then PC<22:16> are pushed into the second available stack location. For a PC push during any CALL instruction, the MSB of the PC is zero-extended before the push, as shown in Figure 3-9. During exception processing, the MSB of the PC is concatenated with the lower eight bits of the CPU STATUS Register, SR. This allows the contents of SRL to be preserved automatically during interrupt processing.

Note 1: To maintain system Stack Pointer (W15) coherency, W15 is never subject to (EDS) paging, and is therefore, restricted to an address range of 0x0000 to 0xFFFF. The same applies to the W14 when used as a Stack Frame Pointer (SFA = 1).

2: As the stack can be placed in, and can access X and Y spaces, care must be taken regarding its use, particularly with regard to local automatic variables in a C development environment

FIGURE 3-9: CALL STACK FRAME



dsPIC33CH128MP508 FAMILY

REGISTER 3-139: C1FIFOUAHx: CAN FIFO USER ADDRESS REGISTER x (x = 1 TO 7) HIGH⁽¹⁾

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
FIFOUA<31:24>							
bit 15				bit 8			

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
FIFOUA<23:16>							
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 15-0 **FIFOUA<31:16>**: FIFO User Address bits

TXEN = 1 (FIFO configured as a transmit buffer):

A read of this register will return the address where the next message is to be written (FIFO head).

TXEN = 0 (FIFO configured as a receive buffer):

A read of this register will return the address where the next message is to be read (FIFO tail).

Note 1: This register is not ensured to read correctly in Configuration mode and should only be accessed when the module is not in Configuration mode.

REGISTER 3-140: C1FIFOUALx: CAN FIFO USER ADDRESS REGISTER x (x = 1 TO 7) LOW⁽¹⁾

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
FIFOUA<15:8>							
bit 15				bit 8			

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
FIFOUA<7:0>							
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 15-0 **FIFOUA<15:0>**: FIFO User Address bits

TXEN = 1 (FIFO configured as a transmit buffer):

A read of this register will return the address where the next message is to be written (FIFO head).

TXEN = 0 (FIFO configured as a receive buffer):

A read of this register will return the address where the next message is to be read (FIFO tail).

Note 1: This register is not ensured to read correctly in Configuration mode and should only be accessed when the module is not in Configuration mode.

dsPIC33CH128MP508 FAMILY

4.1.5 PROGRAMMER'S MODEL

The programmer's model for the dsPIC33CH128MP508S1 family is shown in Figure 4-2. All registers in the programmer's model are memory-mapped and can be manipulated directly by instructions. Table 4-1 lists a description of each register.

In addition to the registers contained in the programmer's model, the dsPIC33CH128MP508S1 devices contain control registers for Modulo Addressing, Bit-Reversed Addressing and interrupts. These registers are described in subsequent sections of this document.

All registers associated with the programmer's model are memory-mapped, as shown in Figure 4-3.

TABLE 4-1: PROGRAMMER'S MODEL REGISTER DESCRIPTIONS

Register(s) Name	Description
W0 through W15 ⁽¹⁾	Working Register Array
W0 through W14 ⁽¹⁾	Alternate 1 Working Register Array
W0 through W14 ⁽¹⁾	Alternate 2 Working Register Array
W0 through W14 ⁽¹⁾	Alternate 3 Working Register Array
W0 through W14 ⁽¹⁾	Alternate 4 Working Register Array
ACCA, ACCB	40-Bit DSP Accumulators (Additional 4 Alternate Accumulators)
PC	23-Bit Program Counter
SR	ALU and DSP Engine STATUS Register
SPLIM	Stack Pointer Limit Value Register
TBLPAG	Table Memory Page Address Register
DSRPAG	Extended Data Space (EDS) Read Page Register
RCOUNT	REPEAT Loop Counter Register
DCOUNT	DO Loop Counter Register
DOSTARTH ⁽²⁾ , DOSTARTL ⁽²⁾	DO Loop Start Address Register (High and Low)
DOENDH, DOENDL	DO Loop End Address Register (High and Low)
CORCON	Contains DSP Engine, DO Loop Control and Trap Status bits

Note 1: Memory-mapped W0 through W14 represent the value of the register in the currently active CPU context.

2: The DOSTARTH and DOSTARTL registers are read-only.

dsPIC33CH128MP508 FAMILY

REGISTER 4-1: SR: CPU STATUS REGISTER (CONTINUED)

- bit 7-5 **IPL<2:0>**: CPU Interrupt Priority Level Status bits^(1,2)
- 111 = CPU Interrupt Priority Level is 7 (15); user interrupts are disabled
 - 110 = CPU Interrupt Priority Level is 6 (14)
 - 101 = CPU Interrupt Priority Level is 5 (13)
 - 100 = CPU Interrupt Priority Level is 4 (12)
 - 011 = CPU Interrupt Priority Level is 3 (11)
 - 010 = CPU Interrupt Priority Level is 2 (10)
 - 001 = CPU Interrupt Priority Level is 1 (9)
 - 000 = CPU Interrupt Priority Level is 0 (8)
- bit 4 **RA**: REPEAT Loop Active bit
- 1 = REPEAT loop is in progress
 - 0 = REPEAT loop is not in progress
- bit 3 **N**: MCU ALU Negative bit
- 1 = Result was negative
 - 0 = Result was non-negative (zero or positive)
- bit 2 **OV**: MCU ALU Overflow bit
- This bit is used for signed arithmetic (2's complement). It indicates an overflow of the magnitude that causes the sign bit to change state.
- 1 = Overflow occurred for signed arithmetic (in this arithmetic operation)
 - 0 = No overflow occurred
- bit 1 **Z**: MCU ALU Zero bit
- 1 = An operation that affects the Z bit has set it at some time in the past
 - 0 = The most recent operation that affects the Z bit has cleared it (i.e., a non-zero result)
- bit 0 **C**: MCU ALU Carry/Borrow bit
- 1 = A carry-out from the Most Significant bit of the result occurred
 - 0 = No carry-out from the Most Significant bit of the result occurred

- Note 1:** The IPL<2:0> bits are concatenated with the IPL<3> bit (CORCON<3>) to form the CPU Interrupt Priority Level. The value in parentheses indicates the IPL, if IPL<3> = 1. User interrupts are disabled when IPL<3> = 1.
- 2:** The IPL<2:0> Status bits are read-only when the NSTDIS bit (INTCON1<15>) = 1.
- 3:** A data write to the SR register can modify the SA and SB bits by either a data write to SA and SB or by clearing the SAB bit. To avoid a possible SA or SB bit write race condition, the SA and SB bits should not be modified using bit operations.

dsPIC33CH128MP508 FAMILY

4.2.1.1 Program Memory Organization

The program memory space is organized in word-addressable blocks. Although it is treated as 24 bits wide, it is more appropriate to think of each address of the program memory as a lower and upper word, with the upper byte of the upper word being unimplemented. The lower word always has an even address, while the upper word has an odd address (Figure 4-4).

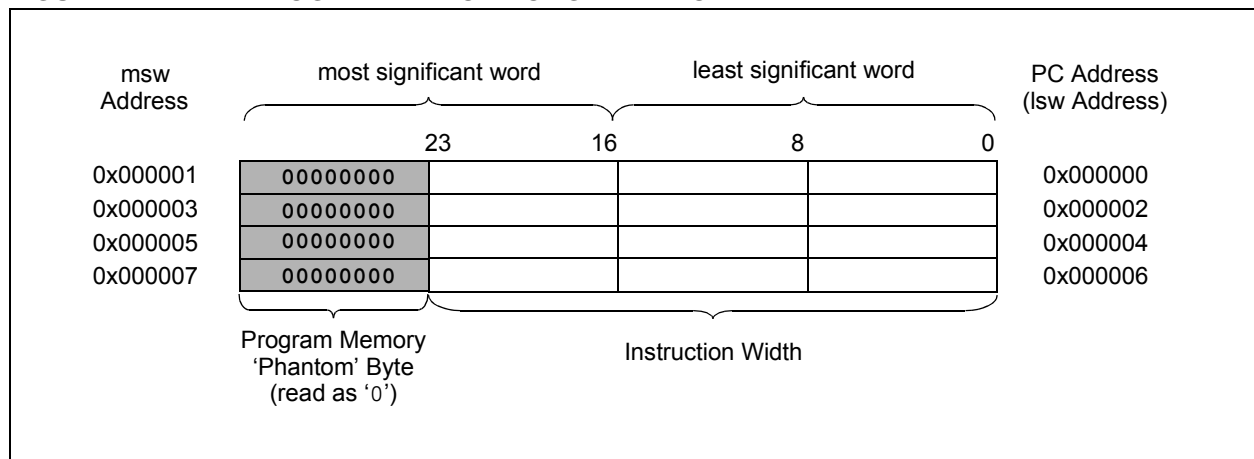
Program memory addresses are always word-aligned on the lower word, and addresses are incremented, or decremented, by two, during code execution. This arrangement provides compatibility with data memory space addressing and makes data in the program memory space accessible.

4.2.1.2 Interrupt and Trap Vectors

All dsPIC33CH128MP508S1 family devices reserve the addresses between 0x000000 and 0x000200 for hard-coded program execution vectors. A hardware Reset vector is provided to redirect code execution from the default value of the PC on device Reset to the actual start of code. A GOTO instruction is programmed by the user application at address, 0x000000, of PRAM memory, with the actual address for the start of code at address, 0x000200, of Flash memory.

A more detailed discussion of the Interrupt Vector Tables (IVTs) is provided in Table 4-20.

FIGURE 4-4: PROGRAM MEMORY ORGANIZATION



dsPIC33CH128MP508 FAMILY

4.2.2 DATA ADDRESS SPACE (SLAVE)

The dsPIC33CH128MP508S1 family CPU has a separate 16-bit wide data memory space. The Data Space is accessed using separate Address Generation Units (AGUs) for read and write operations. The data memory map is shown in Figure 4-5.

All Effective Addresses (EAs) in the data memory space are 16 bits wide and point to bytes within the Data Space. This arrangement gives a base Data Space address range of 64 Kbytes or 32K words.

The lower half of the data memory space (i.e., when $EA<15> = 0$) is used for implemented memory addresses, while the upper half ($EA<15> = 1$) is reserved for the Program Space Visibility (PSV).

The dsPIC33CH128MP508S1 family devices implement up to 4 Kbytes of data memory. If an EA points to a location outside of this area, an all-zero word or byte is returned.

4.2.2.1 Data Space Width

The data memory space is organized in byte-addressable, 16-bit wide blocks. Data is aligned in data memory and registers as 16-bit words, but all Data Space EAs resolve to bytes. The Least Significant Bytes (LSBs) of each word have even addresses, while the Most Significant Bytes (MSBs) have odd addresses.

4.2.2.2 Data Memory Organization and Alignment

To maintain backward compatibility with PIC® MCU devices and improve Data Space memory usage efficiency, the dsPIC33CH128MP508S1 family instruction set supports both word and byte operations. As a consequence of byte accessibility, all Effective Address calculations are internally scaled to step through word-aligned memory. For example, the core recognizes that Post-Modified Register Indirect Addressing mode [Ws++] results in a value of $Ws + 1$ for byte operations and $Ws + 2$ for word operations.

A data byte read, reads the complete word that 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 data path. 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 that matches the byte address.

All word accesses must be aligned to an even address. Misaligned word data fetches are not supported, so care must be taken when mixing byte and word operations, or translating from 8-bit MCU code. If a misaligned read or write is attempted, an address error trap is generated. If the error occurred on a read, the instruction underway is completed. If the error occurred on a write, the instruction is executed but the write does not occur. In either case, a trap is then executed, allowing the system and/or user application to examine the machine state prior to execution of the address Fault.

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

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

4.2.2.3 SFR Space

The first 4 Kbytes of the Near Data Space, from 0x0000 to 0x0FFF, is primarily occupied by Special Function Registers (SFRs). These are used by the dsPIC33CH128MP508S1 family core and peripheral modules for controlling the operation of the device.

SFRs are distributed among the modules that they control and are generally grouped together by module. Much of the SFR space contains unused addresses; these are read as '0'.

<p>Note: The actual set of peripheral features and interrupts varies by the device. Refer to the corresponding device tables and pinout diagrams for device-specific information.</p>
--

4.2.2.4 Near Data Space

The 8-Kbyte area, between 0x0000 and 0x1FFF, is referred to as the Near Data Space. Locations in this space are directly addressable through a 13-bit absolute address field within all memory direct instructions. Additionally, the whole Data Space is addressable using MOV instructions, which support Memory Direct Addressing mode with a 16-bit address field, or by using Indirect Addressing mode using a Working register as an Address Pointer.

dsPIC33CH128MP508 FAMILY

REGISTER 4-24: TRISx: OUTPUT ENABLE FOR PORTx REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TRISx<15:8>							
bit 15				bit 8			

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TRISx<7:0>							
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 15-0 **TRISx<15:0>**: Output Enable for PORTx bits
 1 = LATx[n] is not driven on PORTx[n] pin
 0 = LATx[n] is driven on PORTx[n] pin

REGISTER 4-25: PORTx: INPUT DATA FOR PORTx REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
PORTx<15:8>							
bit 15				bit 8			

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
PORTx<7:0>							
bit 7				bit 0			

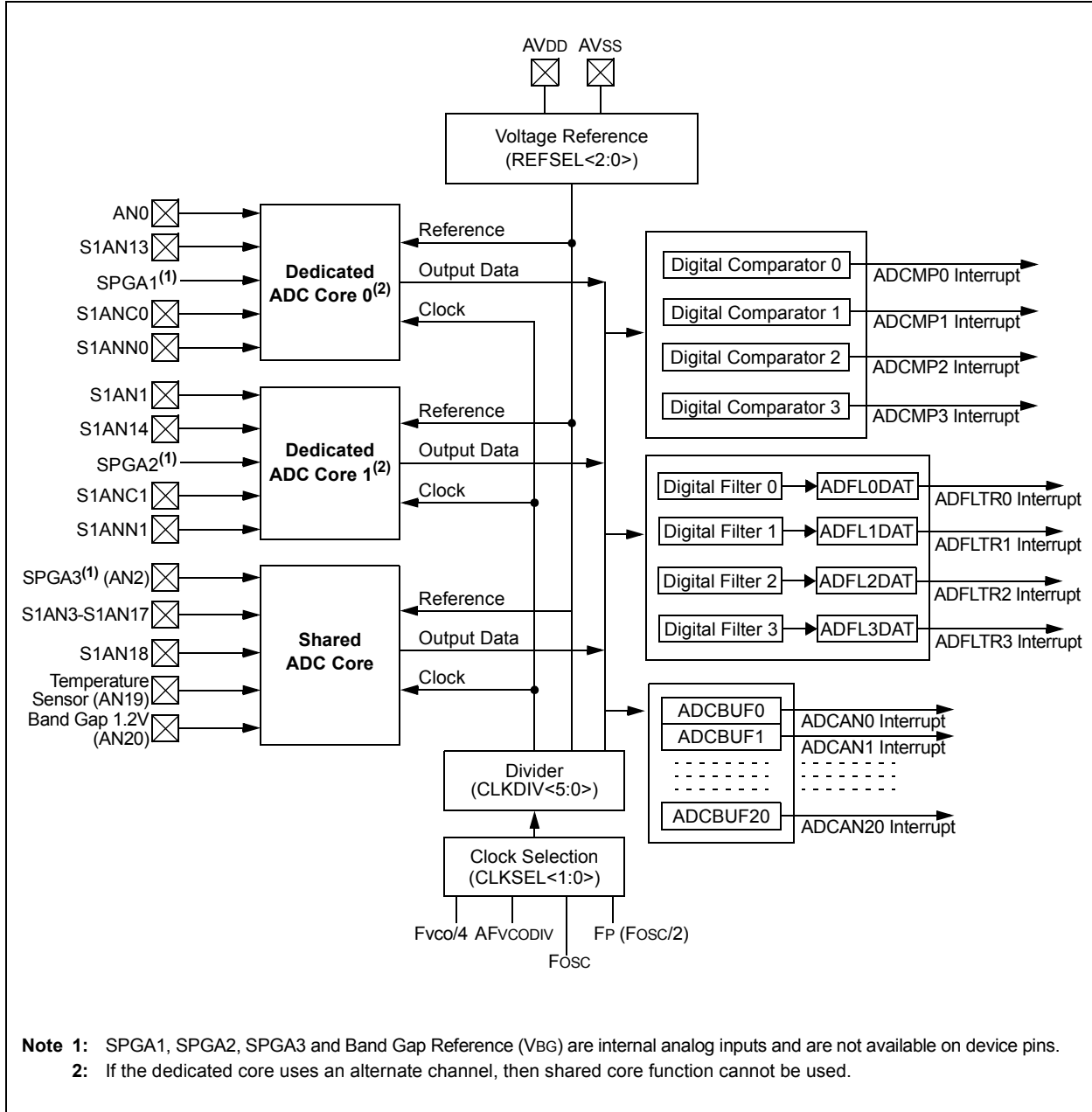
Legend:

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 15-0 **PORTx<15:0>**: PORTx Data Input Value bits

dsPIC33CH128MP508 FAMILY

FIGURE 4-20: ADC MODULE BLOCK DIAGRAM



5.0 MASTER SLAVE INTERFACE (MSI)

Note 1: This data sheet summarizes the features of the dsPIC33CH128MP508 family of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to “**Master Slave Interface (MSI) Module**” (DS70005278) in the “*dsPIC33/PIC24 Family Reference Manual*”, which is available from the Microchip web site (www.microchip.com).

The Master Slave Interface (MSI) module is a bridge between the Master and a Slave processor system, each of which operates within independent clock domains. The Master and Slave have their own registers to communicate between the MSI modules; the Master MSI registers are located in the Master SFR space and the Slave MSI registers are in the Slave SFR space. The Master Slave Interface (MSI) includes these characteristics:

- 16 Unidirectional Data Mailbox Registers:
 - Direction of each Mailbox register is fuse-selectable
 - Byte and word-addressable
- Eight Mailbox Data Flow Control Protocol Blocks:
 - Individual fuse enables
 - Write port active; read port passive (i.e., no read data request required)
 - Automatic, interrupt driven (or polled), data flow control mechanism across MSI clock boundary
 - Fuse assignable to any of the Mailbox registers, supports any length data buffers (up to the number of available Mailbox registers)
 - DMA transfer compatible
- Master to Slave and Slave to Master Interrupt Request with Acknowledge Data Flow Control
- Optional (parameterized) 2-Channel FIFO Memory Structure
- Parameterized Depth (between 16 and 32 words):
 - One read and one write channel
 - Circular operation with empty and full status, and interrupts
 - Overflow/underflow detection with interrupts to Master core and Slave core
 - Interrupt-based, software polled or DMA transfer compatible

- Master and Slave Processor Cross-Boundary Control and Status:
 - Readable operating mode status for both processors
 - Slave enable from Master (subject to satisfying a hardware write interlock sequencer)
 - Master interrupt when Slave is reset during code execution
 - Slave interrupt when Master is reset during code execution
- Optional (fuse) Decoupling of Master and Slave Resets; POR/BOR/MCLR always Resets Master and Slave; Influence of Remaining Run-Time Resets on the Slave Enable is Fuse-Programmable

5.1 Master MSI Control Registers

The following registers are associated with the Master MSI module and are located in the Master SFR space.

- Register 5-1: MSI1CON
- Register 5-2: MSI1STAT
- Register 5-3: MSI1KEY
- Register 5-4: MSI1MBXS
- Register 5-5: MSI1MBXnD
- Register 5-6: MSI1FIFOCS
- Register 5-7: MRSWFDATA
- Register 5-8: MWSRFDATA

dsPIC33CH128MP508 FAMILY

5.2 Slave MSI Control Registers

The following registers are associated with the Slave MSI module and are located in the Slave SFR space.

- Register 5-9: SI1CON
- Register 5-10: SI1STAT
- Register 5-11: SI1MBX
- Register 5-12: SI1MBXnD
- Register 5-13: SI1FIFOCS
- Register 5-14: SWMRFDATA
- Register 5-15: SRMWFDATA

REGISTER 5-9: SI1CON: MSI1 SLAVE CONTROL REGISTER

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	RFITSEL1	RFITSEL0	STMIRQ	MTSIACK
bit 15							bit 8

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
MRSTIE	—	—	—	—	—	—	—
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 15-12 **Unimplemented:** Read as '0'

bit 11-10 **RFITSEL<1:0>:** Read FIFO Interrupt Threshold Select bits

11 = Triggers data valid interrupt when FIFO is full after Slave write

10 = Triggers data valid interrupt when FIFO is 75% full after Slave write

01 = Triggers data valid interrupt when FIFO is 50% full after Slave write

00 = Triggers data valid interrupt when 1st FIFO entry is written by Slave

bit 9 **STMIRQ:** Slave to Master Interrupt Request bit

1 = Interrupts the Master

0 = Does not interrupt the Master

bit 8 **MTSIACK:** Slave to Acknowledge Master Interrupt bit

1 = If MTSIRQ = 1, Slave Acknowledges Master interrupt request, else protocol error

0 = If MTSIRQ = 0, Slave has not yet Acknowledged Master interrupt request, else no Master to Slave interrupt request is pending

bit 7 **MRSTIE:** Master Reset Event Interrupt Enable bit

1 = Slave Master Reset event interrupt occurs when Master enters Reset state

0 = Slave Master Reset event interrupt does not occur when Master enters Reset state

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

dsPIC33CH128MP508 FAMILY

15.4 I²C Control/Status Registers

REGISTER 15-1: I2CxCONL: I2Cx CONTROL REGISTER LOW

R/W-0	U-0	HC/R/W-0	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0
I2CEN	—	I2CSIDL	SCLREL ⁽¹⁾	STRICT	A10M	DISSLW	SMEN
bit 15							bit 8

R/W-0	R/W-0	R/W-0	HC/R/W-0	HC/R/W-0	HC/R/W-0	HC/R/W-0	HC/R/W-0
GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
bit 7							bit 0

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

- bit 15 **I2CEN:** I2Cx Enable bit (writable from software only)
1 = Enables the I2Cx module, and configures the SDAx and SCLx pins as serial port pins
0 = Disables the I2Cx module; all I²C pins are controlled by port functions
- bit 14 **Unimplemented:** Read as '0'
- bit 13 **I2CSIDL:** I2Cx Stop in Idle Mode bit
1 = Discontinues module operation when device enters Idle mode
0 = Continues module operation in Idle mode
- bit 12 **SCLREL:** SCLx Release Control bit (I²C Slave mode only)⁽¹⁾
1 = Releases the SCLx clock
0 = Holds the SCLx clock low (clock stretch)
If STREN = 1:⁽²⁾
User software may write '0' to initiate a clock stretch and write '1' to release the clock. Hardware clears at the beginning of every Slave data byte transmission. Hardware clears at the end of every Slave address byte reception. Hardware clears at the end of every Slave data byte reception.
If STREN = 0:
User software may only write '1' to release the clock. Hardware clears at the beginning of every Slave data byte transmission. Hardware clears at the end of every Slave address byte reception.
- bit 11 **STRICT:** I2Cx Strict Reserved Address Rule Enable bit
1 = Strict Reserved Addressing is enforced; for reserved addresses, refer to Table 15-2.
(In Slave Mode) – The device doesn't respond to reserved address space and addresses falling in that category are NACKed.
(In Master Mode) – The device is allowed to generate addresses with reserved address space.
0 = Reserved Addressing would be Acknowledged.
(In Slave Mode) – The device will respond to an address falling in the reserved address space. When there is a match with any of the reserved addresses, the device will generate an ACK.
(In Master Mode) – Reserved.
- bit 10 **A10M:** 10-Bit Slave Address Flag bit
1 = I2CxADD is a 10-bit Slave address
0 = I2CxADD is a 7-bit Slave address
- bit 9 **DISSLW:** Slew Rate Control Disable bit
1 = Slew rate control is disabled for Standard Speed mode (100 kHz, also disabled for 1 MHz mode)
0 = Slew rate control is enabled for High-Speed mode (400 kHz)

- Note 1:** Automatically cleared to '0' at the beginning of Slave transmission; automatically cleared to '0' at the end of Slave reception.
- 2:** Automatically cleared to '0' at the beginning of Slave transmission.

18.0 CONFIGURABLE LOGIC CELL (CLC)

Note 1: This data sheet summarizes the features of the dsPIC33CH128MP508 family of devices. It is not intended to be a comprehensive reference source. For more information, refer to “**Configurable Logic Cell (CLC)**” (DS70005298) in the “dsPIC33/PIC24 Family Reference Manual”, which is available from the Microchip web site (www.microchip.com). The information in this data sheet supersedes the information in the FRM.

- 2: The CLC is identical for both Master core and Slave core (where the x represents the number of the specific module being addressed in Master or Slave).
- 3: All associated register names are the same on the Master core and the Slave core. The Slave code will be developed in a separate project in MPLAB® X IDE with the device selection, dsPIC33CH128MP508**S1**, where the **S1** indicates the Slave device. The Master and Slave are CLC1 and CLC2.

The Configurable Logic Cell (CLC) module allows the user to specify combinations of signals as inputs to a logic function and to use the logic output to control other peripherals or I/O pins. This provides greater flexibility and potential in embedded designs, since the CLC module can operate outside the limitations of software execution, and supports a vast amount of output designs.

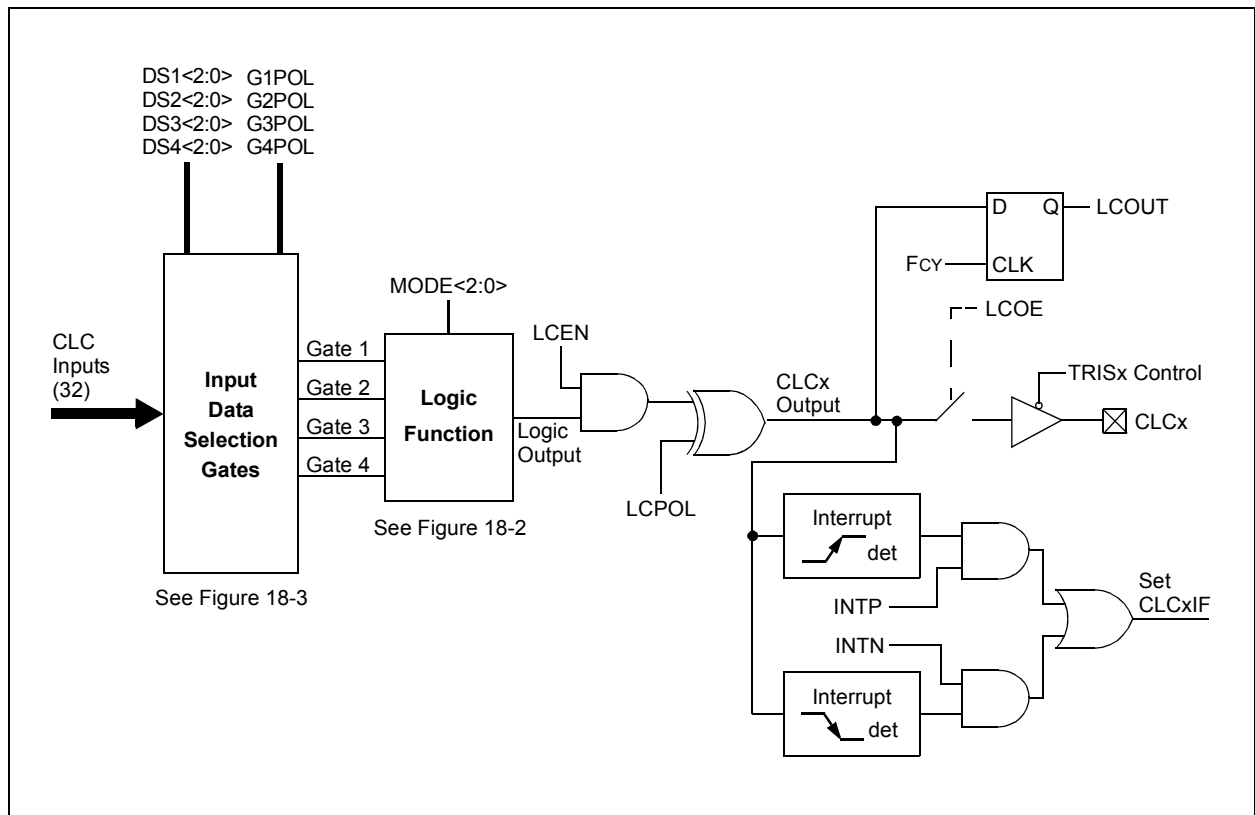
There are four input gates to the selected logic function. These four input gates select from a pool of up to 32 signals that are selected using four data source selection multiplexers. Table 18-1 shows an overview of the module.

TABLE 18-1: CLC MODULE OVERVIEW

	Number of CLC Modules	Identical (Modules)
Master	4	Yes
Slave	4	Yes

Figure 18-3 shows the details of the data source multiplexers and Figure 18-2 shows the logic input gate connections.

FIGURE 18-1: CLCx MODULE



dsPIC33CH128MP508 FAMILY

REGISTER 18-5: CLCxGLSH: CLCx GATE LOGIC INPUT SELECT HIGH REGISTER (CONTINUED)

- bit 3 **G3D2T:** Gate 3 Data Source 2 True Enable bit
1 = Data Source 2 signal is enabled for Gate 3
0 = Data Source 2 signal is disabled for Gate 3
- bit 2 **G3D2N:** Gate 3 Data Source 2 Negated Enable bit
1 = Data Source 2 inverted signal is enabled for Gate 3
0 = Data Source 2 inverted signal is disabled for Gate 3
- bit 1 **G3D1T:** Gate 3 Data Source 1 True Enable bit
1 = Data Source 1 signal is enabled for Gate 3
0 = Data Source 1 signal is disabled for Gate 3
- bit 0 **G3D1N:** Gate 3 Data Source 1 Negated Enable bit
1 = Data Source 1 inverted signal is enabled for Gate 3
0 = Data Source 1 inverted signal is disabled for Gate 3

TABLE 21-3: SLAVE CONFIGURATION REGISTERS MAP

Register Name	Bits 23-16	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
FS1OSCSEL	—	—	—	—	—	—	—	—	—	S1IESO	—	—	—	—	S1FNOSC<2:0>		
FS1OSC	—	—	—	—	—	—	—	—	r ⁽¹⁾	S1FCKSM<1:0>		—	—	—	S1OSCIOfNC	—	—
FS1WDT	—	S1FWDTEN	S1SWDTPS<4:0>				S1WDTWIN<1:0>		S1WINDIS	S1RCLKSEL<1:0>		S1RWDTPS<4:0>					
FS1POR	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FS1ICD	—	S1NOBTSWP	—	S1ISOLAT	—	—	—	—	—	r ⁽¹⁾	—	—	—	—	—	S1ICS<1:0>	
FS1DEVOPT	—	S1MSRE	S1SSRE	S1SPI1PIN	—	—	—	—	—	—	—	—	—	S1ALTI2C1	—	—	—
FS1ALTREG	—	—	S1CTXT4<2:0>			—	S1CTXT3<2:0>		—	S1CTXT2<2:0>			—	S1CTXT1<2:0>			

Legend: — = unimplemented bit, read as '1'; r = reserved bit.

Note 1: Bit is reserved, maintain as '1'.

dsPIC33CH128MP508 FAMILY

REGISTER 21-19: FMBXHCN CONFIGURATION REGISTER

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23						bit 16	

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 15						bit 8	

R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
HS<H:A>EN							
bit 7						bit 0	

Legend:	PO = Program Once 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 23-8 **Unimplemented:** Read as '1'
- bit 7-0 **HS<H:A>EN:** Mailbox Data Flow Control Protocol Block x Enable Fuses bits (x = A, B, C, D, E, F, G, H)
 1 = Mailbox data flow control handshake protocol block is disabled
 0 = Mailbox data flow control handshake protocol block is enabled

REGISTER 21-20: FCFGPR0: PORTA CONFIGURATION REGISTER

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23						bit 16	

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 15						bit 8	

U-1	U-1	U-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
—			CPRA<4:0>				
bit 7						bit 0	

Legend:	PO = Program Once 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 23-5 **Unimplemented:** Read as '1'
- bit 4-0 **CPRA<4:0>:** Configure PORTA Ownership bits
 1 = Master core owns pin
 0 = Slave core owns pin

REGISTER 21-37: RCON: RESET CONTROL REGISTER⁽¹⁾ (CONTINUED)

bit 0 **POR:** Power-on Reset Flag bit
 1 = Power-on Reset has occurred
 0 = Power-on Reset has not occurred

Note 1: All of the Reset status bits can be set or cleared in software. Setting one of these bits in software does not cause a device Reset.

23.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers (MCU) and dsPIC® digital signal controllers (DSC) are supported with a full range of software and hardware development tools:

- Integrated Development Environment
 - MPLAB® X IDE Software
- Compilers/Assemblers/Linkers
 - MPLAB XC Compiler
 - MPASM™ Assembler
 - MPLINK™ Object Linker/
MPLIB™ Object Librarian
 - MPLAB Assembler/Linker/Librarian for
Various Device Families
- Simulators
 - MPLAB X SIM Software Simulator
- Emulators
 - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debuggers/Programmers
 - MPLAB ICD 3
 - PICKit™ 3
- Device Programmers
 - MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards,
Evaluation Kits and Starter Kits
- Third-party development tools

23.1 MPLAB X Integrated Development Environment Software

The MPLAB X IDE is a single, unified graphical user interface for Microchip and third-party software, and hardware development tool that runs on Windows®, Linux and Mac OS® X. Based on the NetBeans IDE, MPLAB X IDE is an entirely new IDE with a host of free software components and plug-ins for high-performance application development and debugging. Moving between tools and upgrading from software simulators to hardware debugging and programming tools is simple with the seamless user interface.

With complete project management, visual call graphs, a configurable watch window and a feature-rich editor that includes code completion and context menus, MPLAB X IDE is flexible and friendly enough for new users. With the ability to support multiple tools on multiple projects with simultaneous debugging, MPLAB X IDE is also suitable for the needs of experienced users.

Feature-Rich Editor:

- Color syntax highlighting
- Smart code completion makes suggestions and provides hints as you type
- Automatic code formatting based on user-defined rules
- Live parsing

User-Friendly, Customizable Interface:

- Fully customizable interface: toolbars, toolbar buttons, windows, window placement, etc.
- Call graph window

Project-Based Workspaces:

- Multiple projects
- Multiple tools
- Multiple configurations
- Simultaneous debugging sessions

File History and Bug Tracking:

- Local file history feature
- Built-in support for Bugzilla issue tracker

dsPIC33CH128MP508 FAMILY

TABLE 24-46: DACx MODULE SPECIFICATIONS

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	Min.	Typ. ⁽¹⁾	Max.	Units	Comments
DA02	CVRES	Resolution	12			bits	
DA03	INL	Integral Nonlinearity Error	-38	—	0	LSB	
DA04	DNL	Differential Nonlinearity Error	-5	—	5	LSB	
DA05	E0FF	Offset Error	-3.5	—	21.5	LSB	Internal node at comparator input
DA06	EG	Gain Error	0	—	41	%	Internal node at comparator input
DA07	TSET	Settling Time	—	750	—	ns	Output with 2% of desired output voltage with a 5-95% or 95-5% step
DA08	VOUT	Voltage Output Range	0.165	—	3.135	V	VDD = 3.3V

Note 1: Parameters are for design guidance only and are not tested in manufacturing.

TABLE 24-47: DACx OUTPUT (DACOUT PIN) SPECIFICATIONS

Operating Conditions: 3.0V to 3.6V (unless otherwise stated) ⁽¹⁾							
Operating temperature -40°C ≤ TA ≤ +85°C for Industrial							
-40°C ≤ TA ≤ +125°C for Extended							
Param No.	Symbol	Characteristic	Min.	Typ.	Max.	Units	Comments
DA11	RLOAD	Resistive Output Load Impedance	10K	—	—	Ohm	
DA11a	CLOAD	Output Load Capacitance	—	—	30	pF	Including output pin capacitance
DA12	IOUT	Output Current Drive Strength	—	3	—	mA	Sink and source
DA13	INL	Integral Nonlinearity Error	-50	—	0	LSB	Includes INL of DACx module (DA03)
DA14	DNL	Differential Nonlinearity Error	-5	—	5	LSB	Includes DNL of DACx module (DA04)
DA30	E0FF	Offset Error	-150	—	0	LSB	Includes offset error of DACx module (DA05)
DA31	EG	Gain Error	-146	—	0	LSB	Includes gain error of DACx module (DA06)

Note 1: The DACx module is functional at VBORMIN < VDD < VDDMIN, but with degraded performance. Unless otherwise stated, module functionality is tested, but not characterized.

dsPIC33CH128MP508 FAMILY

Doze Mode.....	473	High-Speed, 12-Bit Analog-to-Digital Converter (Master ADC).....	221
DSP Engine	45, 271	Control/Status Registers.....	225
dsPIC33CH128MP508S1 Family Interrupt Vector Table.....	315	Features Overview	221
Dual Watchdog Timer (Dual WDT)	704	Resources	224
Control Registers	706	High-Speed, 12-Bit Analog-to-Digital Converter (Slave ADC).....	383
Overview	704	Control/Status Registers.....	387
E		Features Overview	383
ECCADDR1 (ECC Fault Inject Address Compare High).....	87	Resources	386
ECCADDR2 (ECC Fault Inject Address Compare Low)	87	HSPWM	
ECCSTATH (ECC System Status Display High)	88	Architecture	502
ECCSTATL (ECC System Status Display Low).....	88	Control Registers.....	503
Electrical Characteristics.....	727	Overview.....	501
AC	742	I	
ADC Specifications	762	i ² C	
Constant-Current Source Specifications	765	Clock Rates	625
DACx Output (DACOUT Pin) Specifications	764	Communicating as Master in Single Master Environment	623
DACx Specifications	764	Control/Status Registers.....	627
High-Speed Analog Comparator Specifications.....	763	Reserved Addresses	626
I/O Pin Input Injection Current Specifications	740	Setting Baud Rate as Bus Master	625
I/O Pin Input Specifications.....	739	Slave Address Masking	625
I/O Pin Output Specifications	740	In-Circuit Debugger.....	710
Operating Voltage Specifications.....	729	MPLAB ICD 3	725
Program Memory	741	PICkit 3 Programmer	725
Equations		In-Circuit Emulation	667
Frame Time Calculations	635	In-Circuit Serial Programming (ICSP).....	667, 710
I ² C Baud Rate Reload Calculation.....	625	Input Change Notification (ICN).....	123, 341
Master/Slave Core AFPLLO Calculation	438	Instruction Addressing Modes	70, 290
Master/Slave Core AFVCO Calculation.....	438	File Register Instructions	70, 290
Master/Slave Core FPLLO Calculation.....	435	Fundamental Modes Supported	70, 290
Master/Slave Core FVCO Calculation	435	MAC Instructions	71, 291
Relationship Between Device and SPIx Clock Speed	622	MCU Instructions	70, 290
SYNMIN and SYNMAX Calculations	636	Move and Accumulator Instructions	71, 291
Tick Period Calculation	635	Other Instructions	71, 291
Errata	19	Instruction Set Summary	713
Error Correcting Code (ECC).....	80	Overview.....	716
Control Registers	86	Symbols Used in Opcode Descriptions	714
Fault Injection.....	81	Instruction-Based Power-Saving Modes.....	471
F		Idle.....	472
Flexible Configuration	667	Sleep	472
G		Inter-Integrated Circuit. See I ² C.	
Getting Started Guidelines	29	Internet Address	802
Connection Requirements	29	Interrupts Coincident with Power Save Instructions	472
Decoupling Capacitors	29	J	
External Oscillator Pins.....	31	JTAG Boundary Scan Interface.....	667
ICSP Pins.....	31	JTAG Interface	710
Master Clear (MCLR) Pin.....	30	M	
Oscillator Value Conditions on Start-up.....	32	Master CPU	35
Targeted Applications	32	Addressing Modes.....	36
Unused I/Os	32	Control/Status Registers.....	41
H		Data Space Addressing.....	36
High-Resolution PWM (HSPWM) with Fine Edge Placement.....	501	Instruction Set.....	35
High-Speed Analog Comparator with Slope Compensation DAC	553	Registers	35
		Resources	40
		Master Flash Program Memory	77
		Control Registers.....	81
		Operations	77
		RTSP Operation	79