Microchip Technology - DSPIC33CH128MP502-I/2N 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 Dual-Core
Speed	180MHz, 200MHz
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	21
Program Memory Size	152KB (152K 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 23x12b; D/A 4x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-UQFN Exposed Pad
Supplier Device Package	28-UQFN (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/dspic33ch128mp502-i-2n

TABLE 1: MASTER AND SLAVE CORE FEATURES

Feature	Master Core	Slave Core	Shared
Core Frequency	90 MIPS @ 180 MHz	100 MIPS @ 200 MHz	—
Program Memory	64K-128 Kbytes	24 Kbytes (PRAM)(2)	—
Internal Data RAM	16 Kbytes	4 Kbytes	—
16-Bit Timer	1	1	—
DMA	6	2	—
SCCP (Capture/Compare/Timer)	8	4	—
UART	2	1	—
SPI/I²S	2	1	—
I²C	2	1	—
CAN FD	1	—	—
SENT	2	—	—
CRC	1	—	—
QEI	1	1	—
PTG	1	—	—
CLC	4	4	—
16-Bit High-Speed PWM	4	8	—
ADC 12-Bit	1	3	—
Digital Comparator	4	4	—
12-Bit DAC/Analog CMP Module	1	3	—
Watchdog Timer	1	1	—
Deadman Timer	1	—	—
Input/Output	69	69	69
Simple Breakpoints	5	2	—
PGAs(1)	—	3	3
DAC Output Buffer	—	—	1
Oscillator	1	1	1

Note 1: Slave owns the peripheral/feature, but it is shared with the Master.

2: Dual Partition feature is available on Slave PRAM.

TABLE 7: 48-PIN QFN/TQFP/UQFN

Pin #	Master Core	Slave Core
1	**RP46**/PWM1H/RB14	**S1RP46**/S1PWM6L/S1RB14
2	**RP47**/PWM1L/RB15	**S1RP47**/S1PWM6H/S1RB15
3	**RP60**/RC12	**S1RP60**/S1PWM3H/S1RC12
4	**RP61**/RC13	**S1RP61**/S1PWM3L/S1RC13
5	$\overline{\text{MCLR}}$	—
6	RD13	S1ANN0/S1PGA1N2/S1RD13
7	AN12/IBIAS3/**RP48**/RC0	S1AN10/**S1RP48**/S1RC0
8	AN0/CMP1A/RA0	S1RA0
9	AN1/RA1	S1AN15/S1RA1
10	AN2/RA2	S1AN16/S1RA2
11	AN3/IBIAS0/RA3	S1AN0/S1CMP1A/S1PGA1P1/S1RA3
12	AN4/IBIAS1/RA4	$\overline{\text{S1MCLR3}}$/S1AN1/S1CMP2A/S1PGA2P1/S1PGA3P2/S1RA4
13	AVDD	AVDD
14	AVSS	AVSS
15	AN13/ISRC0/**RP49**/RC1	S1ANA1/**S1RP49**/S1RC1
16	AN14/ISRC1/**RP50**/RC2	S1ANA0/**S1RP50**/S1RC2
17	**RP54**/RC6	S1AN11/S1CMP1B/**S1RP54**/S1RC6
18	VDD	VDD
19	VSS	VSS
20	CMP1B/**RP51**/RC3	S1AN8/S1CMP3B/**S1RP51**/S1RC3
21	OSCI/CLKI/AN5/**RP32**/RB0	S1AN5/**S1RP32**/S1RB0
22	OSCO/CLKO/AN6/IBIAS2/**RP33**/RB1	S1AN4/**S1RP33**/S1RB1
23	ISRC3/RD10	S1AN13/S1CMP2B/S1RD10
24	AN15/ISRC2/**RP55**/RC7	S1AN12/**S1RP55**/S1RC7
25	DACOUT/AN7/CMP1D/**RP34**/INT0/RB2	$\overline{\text{S1MCLR2}}$/S1AN3/S1ANC0/S1ANC1/S1CMP1D/S1CMP2D/S1CMP3D/**S1RP34**/S1INT0/S1RB2
26	PGD2/AN8/**RP35**/RB3	S1PGD2/S1AN18/S1CMP3A/S1PGA3P1/**S1RP35**/S1RB3
27	PGC2/**RP36**/RB4	S1PGC2/S1AN9/**S1RP36**/S1PWM5L/S1RB4
28	**RP56**/ASDA1/SCK2/RC8	**S1RP56**/S1ASDA1/S1SCK1/S1RC8
29	**RP57**/ASCL1/SDI2/RC9	**S1RP57**/S1ASCL1/S1SDI1/S1RC9
30	SDO2/PCI19/RD8	S1SDO1/S1PCI19/S1RD8
31	VSS	VSS
32	VDD	VDD
33	PGD3/**RP37**/SDA2/RB5	S1PGD3/**S1RP37**/S1RB5
34	PGC3/**RP38**/SCL2/RB6	S1PGC3/**S1RP38**/S1RB6
35	TDO/AN9/**RP39**/RB7	$\overline{\text{S1MCLR1}}$/S1AN6/**S1RP39**/S1PWM5H/S1RB7
36	PGD1/AN10/**RP40**/SCL1/RB8	S1PGD1/S1AN7/**S1RP40**/S1SCL1/S1RB8
37	PGC1/AN11/**RP41**/SDA1/RB9	S1PGC1/**S1RP41**/S1SDA1/S1RB9
38	**RP52**/RC4	**S1RP52**/S1PWM2H/S1RC4
39	**RP53**/RC5	**S1RP53**/S1PWM2L/S1RC5
40	**RP58**/RC10	**S1RP58**/S1PWM1H/S1RC10
41	**RP59**/RC11	**S1RP59**/S1PWM1L/S1RC11
42	VSS	VSS
43	VDD	VDD
44	**RP65**/RD1	**S1RP65**/S1PWM4H/S1RD1
45	TMS/**RP42**/PWM3H/RB10	**S1RP42**/S1PWM8L/S1RB10
46	TCK/**RP43**/PWM3L/RB11	**S1RP43**/S1PWM8H/S1RB11
47	TDI/**RP44**/PWM2H/RB12	**S1RP44**/S1PWM7L/S1RB12
48	**RP45**/PWM2L/RB13	**S1RP45**/S1PWM7H/S1RB13

Legend: **RPn** and **S1RPn** represent remappable pins for Peripheral Pin Select functions.

3.0 MASTER MODULES

3.1 Master CPU

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 **"dsPIC33E Enhanced CPU"** (DS70005158) in the *"dsPIC33/PIC24 Family Reference Manual"*, which is available from the Microchip web site (www.microchip.com).

There are two independent CPU cores in the dsPIC33CH128MP508 family. The Master and Slave cores are similar, except for the fact that the Slave core can run at a higher speed than the Master core.

The Slave core fetches instructions from the PRAM and the Master core fetches the code from the Flash. The Master and Slave cores can run independently asynchronously, at the same speed or at a different speed. This section discusses the Master core.

Note: All of the associated register names are the same on the Master, as well as on the Slave. 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 dsPIC33CH128MP508 family CPU has a 16-bit (data) modified Harvard architecture with an enhanced instruction set, including significant support for Digital Signal Processing (DSP). The CPU has a 24-bit instruction word with a variable length opcode field. The Program Counter (PC) is 23 bits wide and addresses up to 4M x 24 bits of user program memory space.

An instruction prefetch mechanism helps maintain throughput and provides predictable execution. Most instructions execute in a single-cycle effective execution rate, with the exception of instructions that change the program flow, the double-word move (`MOV.D`) instruction, PSV accesses and the table instructions. Overhead-free program loop constructs are supported using the `DO` and `REPEAT` instructions, both of which are interruptible at any point.

3.1.1 REGISTERS

The dsPIC33CH128MP508 devices have sixteen, 16-bit Working registers in the programmer's model. Each of the Working registers can act as a Data, Address or Address Offset register. The 16th Working register (W15) operates as a Software Stack Pointer for interrupts and calls.

In addition, the dsPIC33CH128MP508 devices include four Alternate Working register sets, which consist of W0 through W14. The Alternate Working registers can be made persistent to help reduce the saving and restoring of register content during Interrupt Service Routines (ISRs). The Alternate Working registers can be assigned to a specific Interrupt Priority Level (IPL1 through IPL7) by configuring the CTXTx<2:0> bits in the FALTREG Configuration register. The Alternate Working registers can also be accessed manually by using the `CTXTSWP` instruction. The CCTXI<2:0> and MCTXI<2:0> bits in the CTXTSTAT register can be used to identify the current, and most recent, manually selected Working register sets.

3.1.2 INSTRUCTION SET

The instruction set for dsPIC33CH128MP508 devices has two classes of instructions: the MCU class of instructions and the DSP class of instructions. These two instruction classes are seamlessly integrated into the architecture and execute from a single execution unit. The instruction set includes many addressing modes and was designed for optimum C compiler efficiency.

3.1.8 ARITHMETIC LOGIC UNIT (ALU)

The dsPIC33CH128MP508 family ALU is 16 bits wide and is capable of addition, subtraction, bit shifts and logic operations. Unless otherwise mentioned, arithmetic operations are two's complement in nature. Depending on the operation, the ALU can affect the values of the Carry (C), Zero (Z), Negative (N), Overflow (OV) and Digit Carry (DC) Status bits in the SR register. The C and DC Status bits operate as Borrow and Digit Borrow bits, respectively, for subtraction operations.

The ALU can perform 8-bit or 16-bit operations, depending on the mode of the instruction that is used. Data for the ALU operation can come from the W register array or data memory, depending on the addressing mode of the instruction. Likewise, output data from the ALU can be written to the W register array or a data memory location.

Refer to the *"16-Bit MCU and DSC Programmer's Reference Manual"* (DS70000157) for information on the SR bits affected by each instruction.

The core CPU incorporates hardware support for both multiplication and division. This includes a dedicated hardware multiplier and support hardware for 16-bit divisor division.

3.1.8.1 Multiplier

Using the high-speed, 17-bit x 17-bit multiplier, the ALU supports unsigned, signed or mixed-sign operation in several MCU multiplication modes:

- 16-bit x 16-bit signed
- 16-bit x 16-bit unsigned
- 16-bit signed x 5-bit (literal) unsigned
- 16-bit signed x 16-bit unsigned
- 16-bit unsigned x 5-bit (literal) unsigned
- 16-bit unsigned x 16-bit signed
- 8-bit unsigned x 8-bit unsigned

3.1.8.2 Divider

The divide block supports 32-bit/16-bit and 16-bit/16-bit signed and unsigned integer divide operations with the following data sizes:

- 32-bit signed/16-bit signed divide
- 32-bit unsigned/16-bit unsigned divide
- 16-bit signed/16-bit signed divide
- 16-bit unsigned/16-bit unsigned divide

The 16-bit signed and unsigned `DIV` instructions can specify any W register for both the 16-bit divisor (Wn) and any W register (aligned) pair (W(m + 1):Wm) for the 32-bit dividend. The divide algorithm takes one cycle per bit of divisor, so both 32-bit/16-bit and 16-bit/16-bit instructions take the same number of cycles to execute. There are additional instructions: `DIV2` and `DIVF2`. Divide instructions will complete in six cycles.

3.1.9 DSP ENGINE

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

The DSP engine can also perform inherent accumulator-to-accumulator operations that require no additional data. These instructions are, `ADD`, `SUB`, `NEG`, `MIN` and `MAX`.

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

- Fractional or integer DSP multiply (IF)
- Signed, unsigned or mixed-sign DSP multiply (USx)
- Conventional or convergent rounding (RND)
- Automatic saturation on/off for ACCA (SATA)
- Automatic saturation on/off for ACCB (SATB)
- Automatic saturation on/off for writes to data memory (SATDW)
- Accumulator Saturation mode selection (ACCSAT)

TABLE 3-2: DSP INSTRUCTIONS SUMMARY

Instruction	Algebraic Operation	ACC Write-Back
`CLR`	$A = 0$	Yes
`ED`	$A = (x - y)^2$	No
`EDAC`	$A = A + (x - y)^2$	No
`MAC`	$A = A + (x \bullet y)$	Yes
`MAC`	$A = A + x^2$	No
`MOVSAC`	No change in A	Yes
`MPY`	$A = x \bullet y$	No
`MPY`	$A = x^2$	No
`MPY.N`	$A = -x \bullet y$	No
`MSC`	$A = A - x \bullet y$	Yes

REGISTER 3-104: C1NBTCFGH: CAN NOMINAL BIT TIME CONFIGURATION REGISTER HIGH(1)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BRP<7:0>							
bit 15							bit 8

R/W-0	R/W-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-0	R/W-0
TSEG1<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 15-8 **BRP<7:0>:** Baud Rate Prescaler bits
1111 1111 = TQ = 256/FSYS
...
0000 0000 = TQ = 1/FSYS

bit 7-0 **TSEG1<7:0>:** Time Segment 1 bits (Propagation Segment + Phase Segment 1)
1111 1111 = Length is 256 x TQ
...
0000 0000 = Length is 1 x TQ

Note 1: These bits can only be modified in Configuration mode (OPMOD<2:0> = 100).

REGISTER 3-105: C1NBTCFGL: CAN NOMINAL BIT TIME CONFIGURATION REGISTER LOW(1)

U-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1	R/W-1	R/W-1
—	TSEG2<6:0>						
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1	R/W-1	R/W-1
—	SJW<6:0>						
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 **Unimplemented:** Read as '0'

bit 14-8 **TSEG2<6:0>:** Time Segment 2 bits (Phase Segment 2)
111 1111 = Length is 128 x TQ
...
000 0000 = Length is 1 x TQ

bit 7 **Unimplemented:** Read as '0'

bit 6-0 **SJW<6:0>:** Synchronization Jump Width bits
111 1111 = Length is 128 x TQ
...
000 0000 = Length is 1 x TQ

Note 1: These bits can only be modified in Configuration mode (OPMOD<2:0> = 100).

REGISTER 3-120: C1RXOVIFH: CAN RECEIVE OVERFLOW INTERRUPT STATUS REGISTER HIGH(1)

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
RFOVIF<31:24>							
bit 15							bit 8

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
RFOVIF<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 **RFOVIF<31:16>:** Unimplemented

Note 1: C1RXOVIFH: FIFO: RFOVIFx (flag needs to be cleared in the FIFO register).

REGISTER 3-121: C1RXOVIFL: CAN RECEIVE OVERFLOW INTERRUPT STATUS REGISTER LOW(1)

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
RFOVIF<15:8>							
bit 15							bit 8

R-0	R-0	R-0	R-0	R-0	R-0	R-0	U-0
RFOVIF<7:1>							—
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 15-8 **RFOVIF<15:8>:** Unimplemented

bit 7-1 **RFOVIF<7:1>:** Receive FIFO Overflow Interrupt Pending bits
1 = Interrupt is pending
0 = Interrupt is not pending

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

Note 1: C1RXOVIFL: FIFO: RFOVIFx (flag needs to be cleared in the FIFO register).

REGISTER 3-163: ADCON5L: ADC CONTROL REGISTER 5 LOW

HSC/R-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
SHRRDY	—	—	—	—	—	—	—
bit 15							bit 8

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

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

bit 15 **SHRRDY:** Shared ADC Core Ready Flag bit
1 = ADC core is powered and ready for operation
0 = ADC core is not ready for operation

bit 14-8 **Unimplemented:** Read as '0'

bit 7 **SHRPWR:** Shared ADC Core Power Enable bit
1 = ADC core is powered
0 = ADC core is off

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

REGISTER 3-176: ADSTATL: ADC DATA READY STATUS REGISTER LOW

HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0
AN<15:8>RDY							
bit 15							bit 8

HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0
AN<7:0>RDY							
bit 7							bit 0

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

bit 15-0 **AN<15:0>RDY:** Common Interrupt Enable for Corresponding Analog Input bits
1 = Channel conversion result is ready in the corresponding ADCBUFx register
0 = Channel conversion result is not ready

REGISTER 3-177: ADSTATH: ADC DATA READY STATUS REGISTER HIGH

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0
—	—	—	AN<20:16>RDY				
bit 7							bit 0

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

bit 15-5 **Unimplemented:** Read as '0'

bit 4-0 **AN<20:16>RDY:** Common Interrupt Enable for Corresponding Analog Input bits
1 = Channel conversion result is ready in the corresponding ADCBUFx register
0 = Channel conversion result is not ready

REGISTER 3-180: ADCMPxENL: ADC DIGITAL COMPARATOR x CHANNEL ENABLE REGISTER LOW (x = 0, 1, 2, 3)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CMPEN<15:8>							
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
CMPEN<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 **CMPEN<15:0>:** Comparator Enable for Corresponding Input Channel bits
1 = Conversion result for corresponding channel is used by the comparator
0 = Conversion result for corresponding channel is not used by the comparator

REGISTER 3-181: ADCMPxENH: ADC DIGITAL COMPARATOR x CHANNEL ENABLE REGISTER HIGH (x = 0, 1, 2, 3)

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

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	CMPEN<20:16>				
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 15-5 **Unimplemented:** Read as '0'

bit 4-0 **CMPEN<20:16>:** Comparator Enable for Corresponding Input Channel bits
1 = Conversion result for corresponding channel is used by the comparator
0 = Conversion result for corresponding channel is not used by the comparator

REGISTER 4-43: RPINR11: PERIPHERAL PIN SELECT INPUT REGISTER 11

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
OCFBR7	OCFBR6	OCFBR5	OCFBR4	OCFBR3	OCFBR2	OCFBR1	OCFBR0
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
OCFAR7	OCFAR6	OCFAR5	OCFAR4	OCFAR3	OCFAR2	OCFAR1	OCFAR0
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 15-8 **OCFBR<7:0>:** Assign Output Compare Fault B (S1OCFB) to the Corresponding S1RPn Pin bits
See Table 4-27

bit 7-0 **OCFBA<7:0>:** Assign Output Compare Fault A (S1OCFA) to the Corresponding S1RPn Pin bits
See Table 4-27

REGISTER 4-44: RPINR12: PERIPHERAL PIN SELECT INPUT REGISTER 12

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCI9R7	PCI9R6	PCI9R5	PCI9R4	PCI9R3	PCI9R2	PCI9R1	PCI9R0
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
PCI8R7	PCI8R6	PCI8R5	PCI8R4	PCI8R3	PCI8R2	PCI8R1	PCI8R0
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 15-8 **PCI9R<7:0>:** Assign PWM Input 9 (S1PCI9) to the Corresponding S1RPn Pin bits
See Table 4-27.

bit 7-0 **PCI8R<7:0>:** Assign PWM Input 8 (S1PCI8) to the Corresponding S1RPn Pin bits
See Table 4-27.

REGISTER 4-47: RPINR15: PERIPHERAL PIN SELECT INPUT REGISTER 15

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
QEIHOM1R7	QEIHOM1R6	QEIHOM1R5	QEIHOM1R4	QEIHOM1R3	QEIHOM1R2	QEIHOM1R1	QEIHOM1R0
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
QEINDX1R7	QEINDX1R6	QEINDX1R5	QEINDX1R4	QEINDX1R3	QEINDX1R2	QEINDX1R1	QEINDX1R0
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 15-8 **QEIHOM1R<7:0>:** Assign QEI Home 1 Input (S1QEIHOM1) to the Corresponding S1RPn Pin bits See Table 4-27.

bit 7-0 **QEINDX1R<7:0>:** Assign QEI Index 1 Input (S1QEINDX1) to the Corresponding S1RPn Pin bits See Table 4-27.

REGISTER 4-48: RPINR18: PERIPHERAL PIN SELECT INPUT REGISTER 18

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
U1DSRR7	U1DSRR6	U1DSRR5	U1DSRR4	U1DSRR3	U1DSRR2	U1DSRR1	U1DSRR0
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
U1RXR7	U1RXR6	U1RXR5	U1RXR4	U1RXR3	U1RXR2	U1RXR1	U1RXR0
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 15-8 **U1DSRR<7:0>:** Assign UART1 Data-Set-Ready ($\overline{\text{S1U1DSR}}$) to the Corresponding S1RPn Pin bits See Table 4-27.

bit 7-0 **U1RXR<7:0>:** Assign UART1 Receive (S1U1RX) to the Corresponding S1RPn Pin bits See Table 4-27.

REGISTER 4-59: RPINR47: PERIPHERAL PIN SELECT INPUT REGISTER 47

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADCTRGR7	ADCTRGR6	ADCTRGR5	ADCTRGR4	ADCTRGR3	ADCTRGR2	ADCTRGR1	ADCTRGR0
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
CLCINDR7	CLCINDR6	CLCINDR5	CLCINDR4	CLCINDR3	CLCINDR2	CLCINDR1	CLCINDR0
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 15-8 **ADCTRGR<7:0>:** Assign ADC External Trigger Input (S1ADCTRG) to the Corresponding S1RPn Pin bits
See Table 4-27.

bit 7-0 **CLCINDR<7:0>:** Assign CLC Input D (S1CLCIND) to the Corresponding S1RPn Pin bits
See Table 4-27.

5.3 Slave Processor Control

The MSI contains three control bits related to Slave processor control within the MSI1CON register.

5.3.1 SLAVE ENABLE (SLVEN) CONTROL

The SLVEN (MSI1CON<15>) control bit provides a means for the Master processor to enable or disable the Slave processor.

The Slave is disabled when SLVEN (MSI1CON<15>) = 0. In this state:

- The Slave is held in the Reset state
- The Master has access to the Slave PRAM (to load it out of a device Reset)
- The Slave Reset status bit, SLVRST (MSI1STAT<15>) = 1

The Slave is enabled when SLVEN (MSI1CON<15>) = 1. In this state:

- The Slave Reset is released and it will start to execute code in whatever mode it is configured to operate in
- The Master processor will no longer have access to the Slave PRAM
- The Slave Reset status bit, SLVRST (MSI1STAT<15>) = 0

Note: The SLVRST (MSI1STAT<15>) status bit indicates when the Slave is in Reset. The associated interrupt only occurs when the Slave enters the Reset state after having previously not been in Reset. That is, no interrupt can be generated until the Slave is first enabled.

The SLVEN bit may only be modified after satisfying the hardware write interlock. The SLVEN bit is protected from unexpected writes through a software unlocking sequence that is based on the MSI1KEY register. Given the critical nature of the MSI control interface, the MSI macro unlock mechanism is independent from that of the Flash controller for added robustness.

Completing a predefined data write sequence to the MSI1KEY register will open a window. The SLVEN bit should be written on the first instruction that follows the unlock sequence. No other bits within the MSI1CON register are affected by the interlock. The MSI1KEY register is not a physical register. A read of the MSI1KEY register will read all '0's.

When the SLVEN bit lock is enabled (i.e., the bits are locked and cannot be modified), the instruction sequence shown in Example 5-1 must be executed to open the lock. The unlock sequence is a prerequisite to both setting and clearing the target control bit.

Note: It is recommended to enable SRSTIE (MSI1CON<7>) = 1 prior to enabling the SLVEN bit. This will make the design robust and will update the Master with the Reset state of the Slave.

EXAMPLE 5-1: MSI ENABLE OPERATION

```
//Unlock Key to allow MSI Enable control
MOV.b   #0x55, W0
MOV.b   WREG, MSI1KEY
MOV.b   #0xAA, W0
MOV.b   WREG, MSI1KEY
// Enable MSI
BSET    MSI1CON, SLVEN
```

EXAMPLE 5-2: MSI ENABLE OPERATION IN C CODE

```
#include <libpic30.h>
_start_slave();
```

5.4 Slave Reset Coupling Control

In all operating modes, the user may couple or decouple the Master Run-Time Resets to the Slave Reset by using the Master Slave Reset Enable (S1MSRE) fuse. The Resets are effectively coupled by directing the selected Reset source to the SLVEN bit Reset.

In all operating modes, the user may also choose whether the SLVEN bit is reset or not in the event of a Slave Run-Time Reset by using the Slave Reset Enable (S1SSRE) fuse.

A user may choose to reset SLVEN in the event of a Slave Reset because that event could be an indicator of a problem with Slave execution. The Slave would be placed in Reset and the Master alerted (via the Slave Reset event interrupt, need to make SRSTIE (MSI1CON<7> = 1) to attempt to rectify the problem. The Master must re-enable the Slave by setting the SLVEN bit again.

Alternatively, the user may choose to not halt the Slave in the event of a Slave Reset, and just allow it to restart execution after a Reset and continue operation as soon as possible. The Slave Reset event interrupt would still occur, but could be ignored by the Master.

TABLE 7-2: MASTER PMD REGISTERS

Register	Bit 15	Bit14	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
PMDCONL	—	—	—	—	PMDLOCK	—	—	—	—	—	—	—	—	—	—	—
PMD1	—	—	—	—	T1MD	QEIMD	PWMMD	—	I2C1MD	U2MD	U1MD	SPI2MD	SPI1MD	—	C1MD	ADC1MD
PMD2	—	—	—	—	—	—	—	—	CCP8MD	CCP7MD	CCP6MD	CCP5MD	CCP4MD	CCP3MD	CCP2MD	CCP1MD
PMD3	—	—	—	—	—	—	—	—	CRCMD	—	—	—	—	—	I2C2MD	—
PMD4	—	—	—	—	—	—	—	—	—	—	—	—	REFOMD	—	—	—
PMD6	—	—	DMA5MD	DMA4MD	DMA3MD	DMA2MD	DMA1MD	DMA0MD	—	—	—	—	—	—	—	—
PMD7	—	—	—	—	—	—	—	CMP1MD	—	—	—	—	PTGMD	—	—	—
PMD8	—	—	—	SENT2MD	SENT1MD	—	—	—	—	—	CLC4MD	CLC3MD	CLC2MD	CLC1MD	BIASMD	—

TABLE 7-3: SLAVE PMD REGISTERS

Register	Bit 15	Bit14	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
PMDCON	—	—	—	—	PMDLOCK	—	—	—	—	—	—	—	—	—	—	—
PMD1	—	—	—	—	T1MD	QEIMD	PWMMD	—	I2C1MD	—	U1MD	—	SPI1MD	—	—	ADC1MD
PMD2	—	—	—	—	—	—	—	—	—	—	—	—	CCP4MD	CCP3MD	CCP2MD	CCP1MD
PMD4	—	—	—	—	—	—	—	—	—	—	—	—	REFOMD	—	—	—
PMD6	—	—	—	—	—	—	DMA1MD	DMA0MD	—	—	—	—	—	—	—	—
PMD7	—	—	—	—	—	CMP3MD	CMP2MD	CMP1MD	—	—	—	—	—	—	PGA1MD	—
PMD8	—	PGA3MD	—	—	—	PGA2MD	—	—	—	—	CLC4MD	CLC3MD	CLC2MD	CLC1MD	—	—

REGISTER 21-28: FS1POR CONFIGURATION REGISTER (SLAVE)

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	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
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 23-0 **Unimplemented:** Read as ‘1’

21.9 JTAG Interface

The dsPIC33CH128MP508 family devices implement a JTAG interface, which supports boundary scan device testing. Detailed information on this interface will be provided in future revisions of this document.

Note: Refer to **"Programming and Diagnostics"** (DS70608) in the *"dsPIC33/PIC24 Family Reference Manual"* for further information on usage, configuration and operation of the JTAG interface.

21.10 In-Circuit Serial Programming™ (ICSP™)

The dsPIC33CH128MP508 family devices can be serially programmed while in the end application circuit. This is done with two lines for clock and data, and three other lines for power, ground and the programming sequence. Serial programming allows customers to manufacture boards with unprogrammed devices and then program the device just before shipping the product. Serial programming also allows the most recent firmware or a custom firmware to be programmed. Refer to the *"dsPIC33CH128MP508 Family Flash Programming Specification"* (DS70005285) for details about In-Circuit Serial Programming (ICSP).

Any of the three pairs of programming clock/data pins can be used:

- PGC1 and PGD1
- PGC2 and PGD2
- PGC3 and PGD3

Note: Both Master core and Slave core can be used with MPLAB® ICD to debug at the same time. There are PGCx and PGDx pins dedicated for the Master core and Slave core (S1PGCx and S1PGDx) to make this possible. $\overline{\text{MCLR}}$ is the same for programming the Master core and the Slave core. S1MCLRx is used only when the Master and Slave are debugged simultaneously.

21.11 In-Circuit Debugger

When MPLAB® ICD 3 or the REAL ICE™ emulator is selected as a debugger, the in-circuit debugging functionality is enabled. This function allows simple debugging functions when used with MPLAB IDE. Debugging functionality is controlled through the PGCx (Emulation/Debug Clock) and PGDx (Emulation/Debug Data) pin functions.

Any of the three pairs of debugging clock/data pins can be used:

- PGC1 and PGD1 Master Debug or Slave Debug
- PGC2 and PGD2 Master Debug or Slave Debug
- PGC3 and PGD3 Master Debug or Slave Debug for debugging Master and Slave simultaneously, two MPLAB ICD debuggers or the REAL ICE™ emulator are required. This mode of debugging, where the Master and Slave are simultaneously debugged, is called the Dual Debug mode. S1MCLRx and S1PGCx/S1PGDx are used only in Dual Debug mode.

The Dual Debug mode of operation needs the following PGCx/PGDx pins:

- $\overline{\text{MCLR}}$, PGC1 and PGD1 for Master Debug, and S1MCLR1, S1PGC1 and S1PGD1 for Slave Debug
- $\overline{\text{MCLR}}$, PGC2 and PGD2 for Master Debug, and S1MCLR2, S1PGC2 and S1PGD2 for Slave Debug
- $\overline{\text{MCLR}}$, PGC3 and PGD3 for Master Debug, and S1MCLR3, S1PGC3 and S1PGD3 for Slave Debug

To use the in-circuit debugger function of the device, the design must implement ICSP connections to $\overline{\text{MCLR}}$, V_{DD}, V_{SS} and the PGCx/PGDx pin pair. In addition, when the feature is enabled, some of the resources are not available for general use. These resources include the first 80 bytes of data RAM and two or five (in Dual Debug) I/O pins (PGCx and PGDx).

There are three modes of debugging the dual core family of dsPIC33CH128MP508:

1. Master Only Debug
2. Slave Only Debug
3. Dual Debug

21.11.1 MASTER ONLY DEBUG

In Master Only Debug, only the Master project will be debugged. There is no project for Slave or no Slave code. The main project will be for dsPIC33CHXXXMP50X/20X and the user has to use $\overline{\text{MCLR}}$ and PGCx/PGDx for debugging. This is similar to debugging any single core existing device.

TABLE 24-4: OPERATING VOLTAGE 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**	**Conditions**
Operating Voltage							
DC10	VDD	**Supply Voltage**	3.0	—	3.6	V	
DC12	VDR	**RAM Retention Voltage(2)**	1.8	—	—	V	
DC16	VPOR	**VDD Start Voltage** to Ensure Internal Power-on Reset Signal	—	—	VSS	V	
DC17	SVDD	**VDD Rise Rate** to Ensure Internal Power-on Reset Signal	1.0	—	—	V/ms	0V-3V in 3 ms
BO10	VBOR	**BOR Event on VDD Transition** High-to-Low(3)	2.68	2.84	2.99	V	

Note 1: Device is functional at VBORMIN < VDD < VDDMIN. Analog modules (ADC and comparators) may have degraded performance.

2: This is the limit to which VDD may be lowered and the RAM contents will always be retained.

3: Parameters are characterized but not tested.

TABLE 24-9: DC CHARACTERISTICS: IDLE CURRENT (IIDLE) (MASTER IDLE/SLAVE SLEEP)

DC CHARACTERISTICS	Master (Idle) + Slave (Sleep)		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.	**Typ.**	**Max.**	**Units**	**Conditions**		
Idle Current (IIDLE)(1)						
DC40a	6.6	8.4	mA	-40°C	3.3V	10 MIPS (N = 1, N2 = 5, N3 = 2, M = 50, FVCO = 400 MHz, FPLLO = 40 MHz)
	6.7	11.9	mA	+25°C		
	6.9	17.9	mA	+85°C		
	10.9	24.9	mA	+125°C		
DC41a	7.3	9.2	mA	-40°C	3.3V	20 MIPS (N = 1, N2 = 5, N3 = 1, M = 50, FVCO = 400 MHz, FPLLO = 80 MHz)
	7.5	12.7	mA	+25°C		
	7.7	18.7	mA	+85°C		
	11.7	25.7	mA	+125°C		
DC42a	9.2	11.1	mA	-40°C	3.3V	40 MIPS (N = 1, N2 = 3, N3 = 1, M = 60, FVCO = 480 MHz, FPLLO = 160 MHz)
	9.4	14.8	mA	+25°C		
	9.5	20.7	mA	+85°C		
	13.5	27.5	mA	+125°C		
DC43a	11.8	13.9	mA	-40°C	3.3V	70 MIPS (N = 1, N2 = 2, N3 = 1, M = 70, FVCO = 560 MHz, FPLLO = 280 MHz)
	12.0	17.6	mA	+25°C		
	12.1	23.5	mA	+85°C		
	16.1	30.1	mA	+125°C		
DC44a	14.1	16.3	mA	-40°C	3.3V	90 MIPS (N = 1, N2 = 2, N3 = 1, M = 90, FVCO = 720 MHz, FPLLO = 360 MHz)
	14.2	20	mA	+25°C		
	14.3	25.9	mA	+85°C		
	18.2	32.3	mA	+125°C		

Note 1: Base Idle current (IIDLE) is measured as follows:

- FIN = 8 MHz, FPFD = 8 MHz
- CLKO is configured as an I/O input pin in the Configuration Word
- All I/O pins are configured as output low
- $\overline{\text{MCLR}}$ = VDD, WDT and FSCM are disabled
- No peripheral modules are operating or being clocked (all defined PMDx bits are set)
- The NVMSIDL bit (NVMCON<12>) = 1 (i.e., Flash regulator is set to standby while the device is in Idle mode)
- JTAG is disabled

TABLE 24-39: SPIx SLAVE MODE (FULL-DUPLEX, CKE = 1, CKP = x, SMP = 0) 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.(2)**	**Max.**	**Units**	**Conditions**
SP10	FscP	Maximum SCKx Input Frequency	—	—	15	MHz	Using PPS pins
			—	—	40	MHz	SPI2 dedicated pins
SP72	TscF	SCKx Input Fall Time	—	—	—	ns	See Parameter DO32 **(Note 3)**
SP73	TscR	SCKx Input Rise Time	—	—	—	ns	See Parameter DO31 **(Note 3)**
SP30	TdoF	SDOx Data Output Fall Time	—	—	—	ns	See Parameter DO32 **(Note 3)**
SP31	TdoR	SDOx Data Output Rise Time	—	—	—	ns	See Parameter DO31 **(Note 3)**
SP35	TscH2doV, TscL2doV	SDOx Data Output Valid After SCKx Edge	—	6	20	ns	
SP36	TdoV2scH, TdoV2scL	SDOx Data Output Setup to First SCKx Edge	30	—	—	ns	Using PPS pins
			20	—	—	ns	SPI2 dedicated pins
SP40	TdiV2scH, TdiV2scL	Setup Time of SDIx Data Input to SCKx Edge	30	—	—	ns	Using PPS pins
			10	—	—	ns	SPI2 dedicated pins
SP41	TscH2diL, TscL2diL	Hold Time of SDIx Data Input to SCKx Edge	30	—	—	ns	Using PPS pins
			15	—	—	ns	SPI2 dedicated pins
SP50	TssL2scH, TssL2scL	$\overline{SSx}$ ↓ to SCKx ↑ or SCKx ↓ Input	120	—	—	ns	
SP51	TssH2doZ	$\overline{SSx}$ ↑ to SDOx Output High-Impedance	8	—	50	ns	**(Note 3)**
SP52	TscH2ssH, TscL2ssH	$\overline{SSx}$ ↑ After SCKx Edge	1.5 TCY + 40	—	—	ns	**(Note 3)**
SP60	TssL2doV	SDOx Data Output Valid After $\overline{SSx}$ Edge	—	—	50	ns	

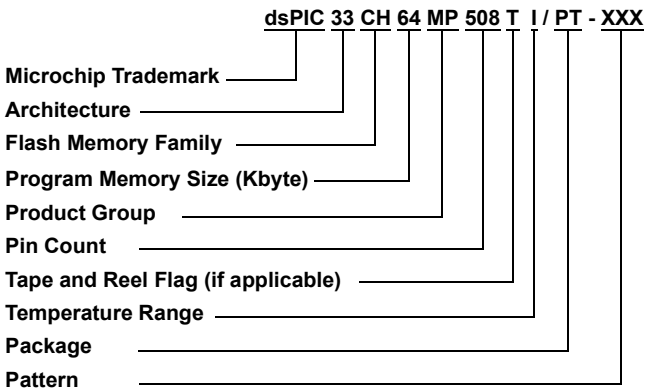
Note 1: These parameters are characterized but not tested in manufacturing.

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

3: Assumes 50 pF load on all SPIx pins.

PRODUCT IDENTIFICATION SYSTEM

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



Architecture:	33	=	16-Bit Digital Signal Controller
Flash Memory Family:	CH	=	Dual Core
Product Group:	MP	=	Motor Control/Power Supply
Pin Count:	02	=	28-pin
	03	=	36-pin
	04	=	48-pin
	06	=	64-pin
	08	=	80-pin
Temperature Range:	I	=	-40°C to +85°C (Industrial)
	E	=	-40°C to +125°C (Extended)
Package:	SS	=	Plastic Shrink Small Outline – (28-pin) 5.30 mm body (SSOP)
	2N	=	Ultra Thin Plastic Quad Flat, No Lead – (28-pin) 6x6 mm body (UQFN)
	M5	=	Ultra Thin Plastic Quad Flat, No Lead – (36-pin) 5x5 mm body (UQFN)
	PT	=	Thin Quad Flatpack – (48-pin) 7x7 mm body (TQFP)
	M4	=	Ultra Thin Plastic Quad Flat, No Lead – (48-pin) 6x6 mm body (UQFN)
	PT	=	Plastic Thin Quad Flatpack – (64-pin) 10x10 mm body (TQFP)
	MR	=	Plastic Quad Flat, No Lead – (64-pin) 9x9 mm body (QFN)
	PT	=	Plastic Thin Quad Flatpack – (80-pin) 12x12 mm body (TQFP)

Examples:

dsPIC33CH128MP506-I/PT:
dsPIC33, Enhanced Performance, 128-Kbyte Program Memory, SMPS, 64-Pin, Industrial Temperature, TQFP Package.