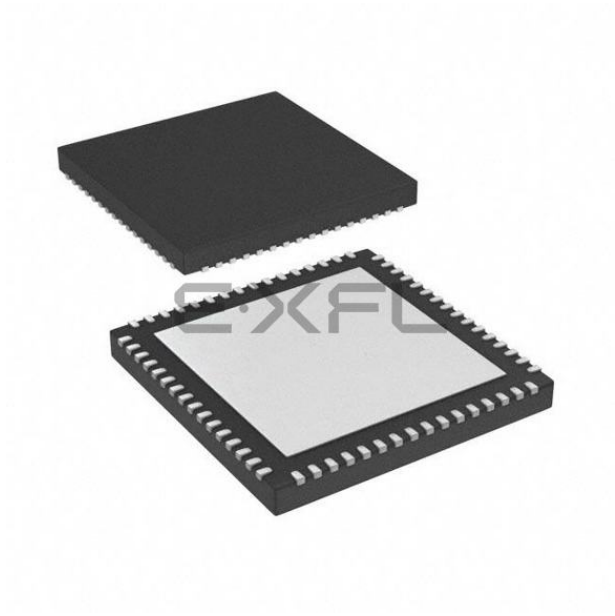
Microchip Technology - DSPIC33CH128MP506T-I/MR Datasheet



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	53
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 34x12b; D/A 4x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-VFQFN Exposed Pad
Supplier Device Package	64-QFN (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/dspic33ch128mp506t-i-mr

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.

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

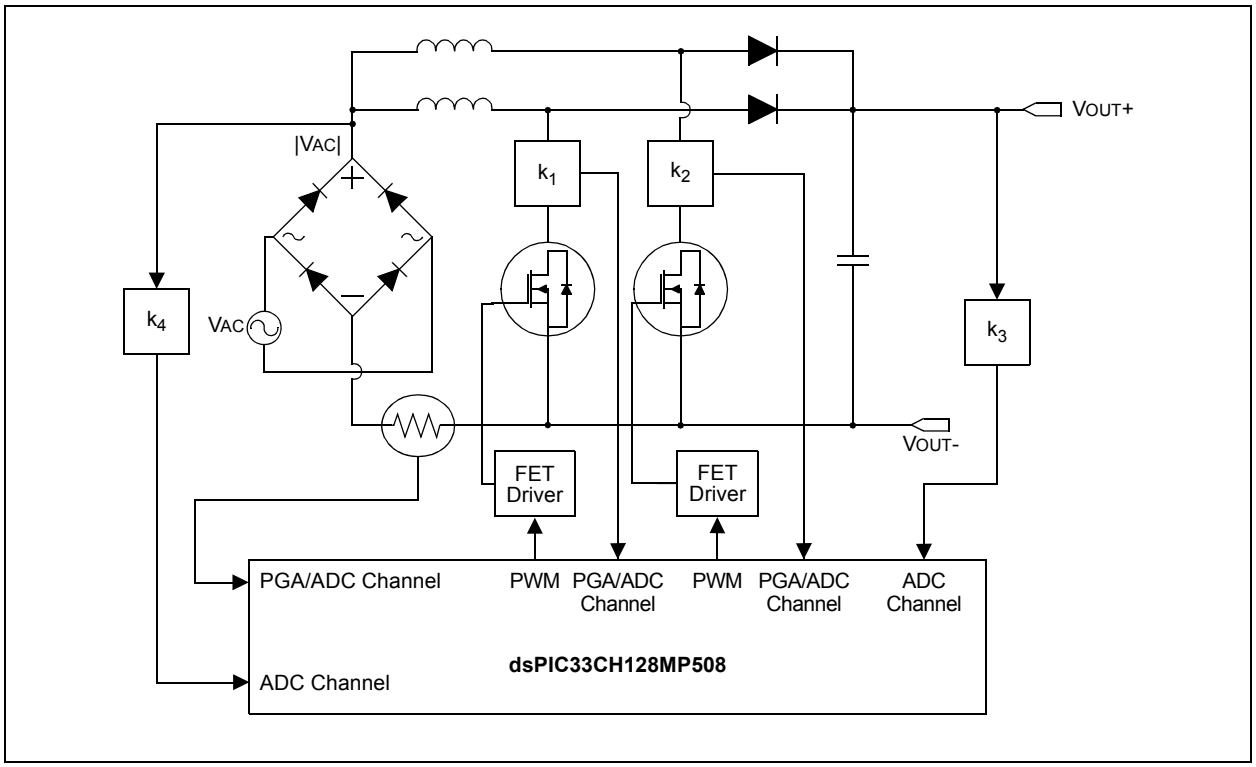


TABLE 3-18: MASTER SFR BLOCK F00h

Register	Address	All Resets	Register	Address	All Resets	Register	Address	All Resets
Reset			PMD1	FA4	----000-00000-00	PCTRAPH	FC2	--------00000000
RCON	F80	00--x-0000000011	PMD2	FA6	--------00000000	FEXL	FC4	xxxxxxxxxxxxxxxx
Oscillator			PMD3	FA8	--------0-----0-	FEXH	FC6	--------xxxxxxxx
OSCCON	F84	-000-yyy0-0-0--0	PMD4	FAA	------------0---	DPCL	FCE	xxxxxxxxxxxxxxxx
CLKDIV	F86	00110000--000001	PMD6	FAE	--000000--------	DPCH	FD0	--------xxxxxxxx
PLLFBD	F88	----000010010110	PMD7	FB0	-------x----0---	APPO	FD2	xxxxxxxxxxxxxxxx
PLLDIV	F8A	------00-011-001	PMD8	FB2	---00--0--xx000-	APPI	FD4	xxxxxxxxxxxxxxxx
OSCTUN	F8C	----------000000	**WDT**			APPS	FD6	----------xxxxx
ACLKCON1	F8E	00-----0--000001	WDTCONL	FB4	0--0000000000000	STROUTL	FD8	xxxxxxxxxxxxxxxx
APLLFBD1	F90	----000010010110	WDTCONH	FB6	0000000000000000	STROUTH	FDA	xxxxxxxxxxxxxxxx
APLLDIV1	F92	------00-011-001	REFOCONL	FB8	0-000-00----0000	STROVCNT	FDC	xxxxxxxxxxxxxxxx
CANCLKCON	F9A	----xxxx-xxxxxxx	REFOCONH	FBA	-000000000000000	JDATAH	FFA	xxxxxxxxxxxxxxxx
PMD			REFOTRIML	FBC	0000000000000000	JDATAL	FFC	xxxxxxxxxxxxxxxx
PMDCON	FA0	----0----------	PCTRAPL	FC0	0000000000000000			

Legend: x = unknown or indeterminate value; "-" =unimplemented bits; y = value set by Configuration bits. Address and Reset values are in hexadecimal and binary, respectively.

REGISTER 3-49: RPINR13: PERIPHERAL PIN SELECT INPUT REGISTER 13

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCI11R7	PCI11R6	PCI11R5	PCI11R4	PCI11R3	PCI11R2	PCI11R1	PCI11R0
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
PCI10R7	PCI10R6	PCI10R5	PCI10R4	PCI10R3	PCI10R2	PCI10R1	PCI10R0
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 **PCI11R<7:0>:** Assign PWM Input 11 (PCI11) to the Corresponding RPn Pin bits
See Table 3-30.

bit 7-0 **PCI10R<7:0>:** Assign PWM Input 10 (PCI10) to the Corresponding RPn Pin bits
See Table 3-30.

REGISTER 3-50: RPINR14: PERIPHERAL PIN SELECT INPUT REGISTER 14

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
QEIB1R7	QEIB1R6	QEIB1R5	QEIB1R4	QEIB1R3	QEIB1R2	QEIB1R1	QEIB1R0
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
QEIA1R7	QEIA1R6	QEIA1R5	QEIA1R4	QEIA1R3	QEIA1R2	QEIA1R1	QEIA1R0
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 **QEIB1R<7:0>:** Assign QEI Input B (QEIB1) to the Corresponding RPn Pin bits
See Table 3-30.

bit 7-0 **QEIA1R<7:0>:** Assign QEI Input A (QEIA1) to the Corresponding RPn Pin bits
See Table 3-30.

REGISTER 3-187: PTGHOLD: PTG HOLD REGISTER(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
PTGHOLD<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
PTGHOLD<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 **PTGHOLD<15:0>:** PTG General Purpose Hold Register bits
This register holds the user-supplied data to be copied to the PTGTxLIM, PTGCxLIM, PTGSDLIM or PTGL0 register using the PTGCOPY command.

Note 1: These bits are read-only when the module is executing Step commands.

REGISTER 3-188: PTGT0LIM: PTG TIMER0 LIMIT REGISTER(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
PTGT0LIM<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
PTGT0LIM<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 **PTGT0LIM<15:0>:** PTG Timer0 Limit Register bits
General Purpose Timer0 Limit register.

Note 1: These bits are read-only when the module is executing Step commands.

4.1.8 ARITHMETIC LOGIC UNIT (ALU)

The dsPIC33CH128MP508S1 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.

4.1.8.1 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

4.1.8.2 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 quotient for all divide instructions ends up in W0 and the remainder in W1. 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.

4.1.9 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 and NEG.

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 4-2: 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	$A = A + (x \cdot y)$	Yes
MAC	$A = A + x^2$	No
MOVSAC	No change in A	Yes
MPY	$A = x \cdot y$	No
MPY	$A = x^2$	No
MPY.N	$A = -x \cdot y$	No
MSC	$A = A - x \cdot y$	Yes

4.2 Slave Memory Organization

Note: 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/PIC24E Program Memory"** (DS70000613) in the *"dsPIC33/PIC24 Family Reference Manual"*, which is available from the Microchip web site (www.microchip.com).

The dsPIC33CH128MP508S1 family architecture features separate program and data memory spaces, and buses. This architecture also allows the direct access of program memory from the Data Space (DS) during code execution.

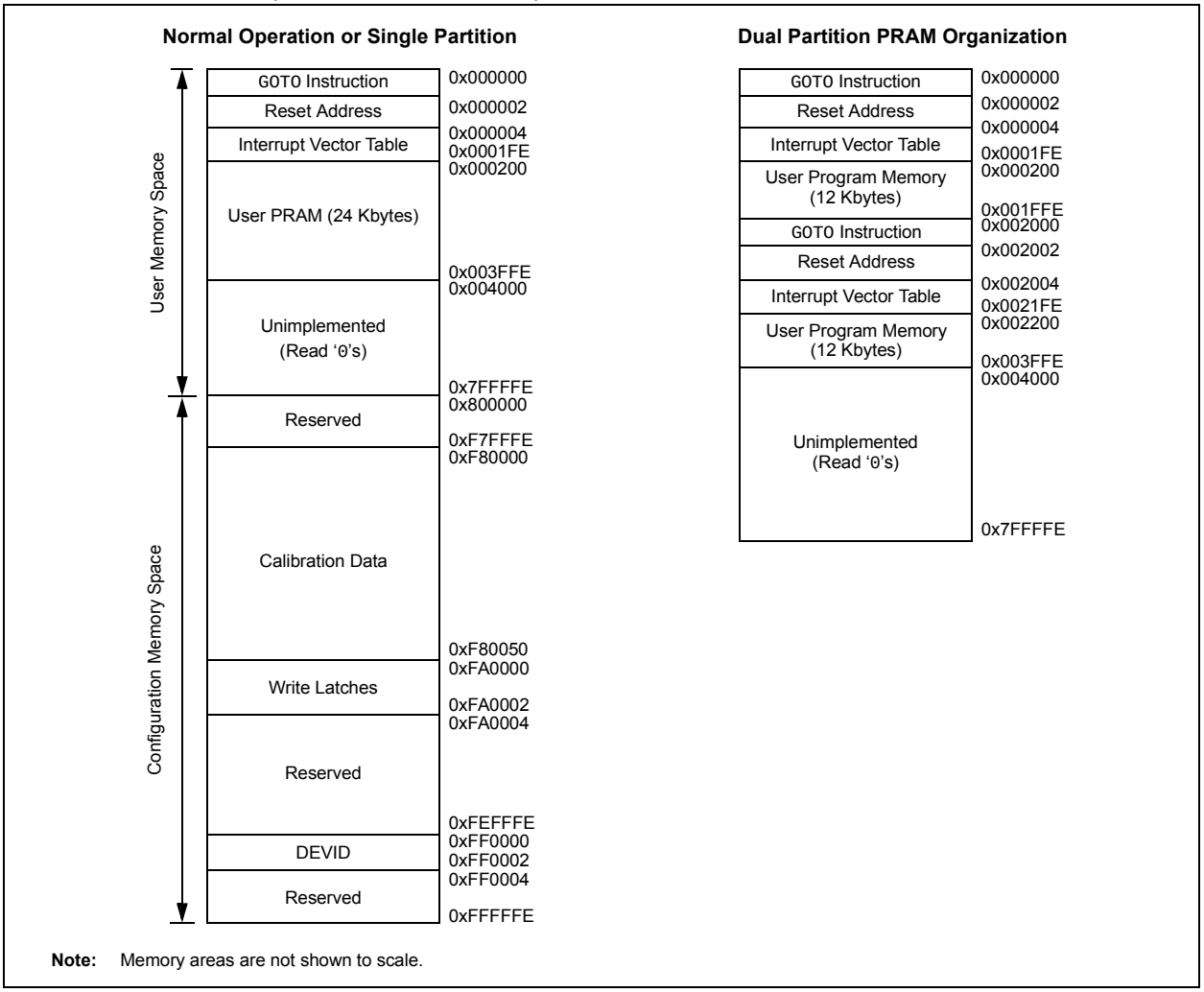
4.2.1 PROGRAM ADDRESS SPACE

The program address memory space of the dsPIC33CH128MP508S1 family devices is 4M instructions. The space is addressable by a 24-bit value derived either from the 23-bit PC during program execution, or from table operation or Data Space remapping, as described in **Section 4.2.8 "Interfacing Program and Data Memory Spaces"**.

User application access to the program memory space is restricted to the lower half of the address range (0x000000 to 0x7FFFFF). The exception is the use of TBLRD operations, which use TBLPAG<7> to permit access to calibration data and Device ID sections of the configuration memory space.

The PRAM for the Slave dsPIC33CH128MP508S1 devices implements two 12-Kbyte PRAM panels with a total of 24 Kbytes of PRAM available for the Slave device. All variants of the Slave have the same amount of PRAM available, irrespective of the size of the Flash available on the Master Flash program memory, as shown in Figure 4-3.

FIGURE 4-3: PRAM (PROGRAM MEMORY) FOR SLAVE dsPIC33CH128MP508S1 DEVICES



Note: Memory areas are not shown to scale.

4.3 Slave PRAM Program Memory

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 **"Dual Partition Flash Program Memory"** (DS70005156) in the *"dsPIC33/PIC24 Family Reference Manual"*, which is available from the Microchip web site (www.microchip.com)

2: Though the reference to the chapter is **"Dual Partition Flash Program Memory"** (DS70005156), the program memory for the Slave code is PRAM. Therefore, after each POR, the Master will have to reload the content of the Slave PRAM.

The dsPIC33CH128MP508S1 family devices contain internal PRAM program memory for storing and executing application code. The PRAM program memory array is organized into rows of 128 instructions or 64 double instruction words. Though the PRAM is volatile, it is writable during normal operation over the entire VDD range.

PRAM memory can be programmed in three ways:

- In-Circuit Serial Programming™ (ICSP™)
- Run-Time Self-Programming (RTSP)
- Master to Slave Image Loading (MSIL)

ICSP allows for a dsPIC33CH128MP508S1 family device to be serially programmed in the application circuit. Since the Slave PRAM is volatile, Slave PRAM ICSP programming is supported only as a development and debugging feature.

RTSP allows the Slave PRAM user application code to update itself during run time. This feature is capable of writing a single program memory word (two instructions) or an entire row as needed.

Master to Slave Image Loading allows the Master user code to load the Slave PRAM at run time. A Slave PRAM compatible image is stored in Master Flash memory. At run time, the Master user code is responsible for loading and verifying the contents of the Slave PRAM.

Note: In an actual application mode, the Slave PRAM is loaded by the Master, so the ICSP mode of PRAM operation is valid only for the Debug mode during the code development.

4.3.1 PRAM PROGRAMMING OPERATIONS

For ICSP and RTSP programming of the Slave PRAM, TBLWTL and TBLWTH instructions are used to write to the NVM write latches. An NVM write operation then writes the contents of both latches to the PRAM, starting at the address defined in the NVMADR and NVMADRU registers.

For Master to Slave Image Loading (MSIL) of the Slave PRAM, the Master user code is responsible for transferring the Slave image contents stored in the Master Flash to the Slave PRAM. The LDSLV instruction is used along with the DSRPAG and DSWPAG registers to transfer a single 24-bit instruction to the Slave PRAM.

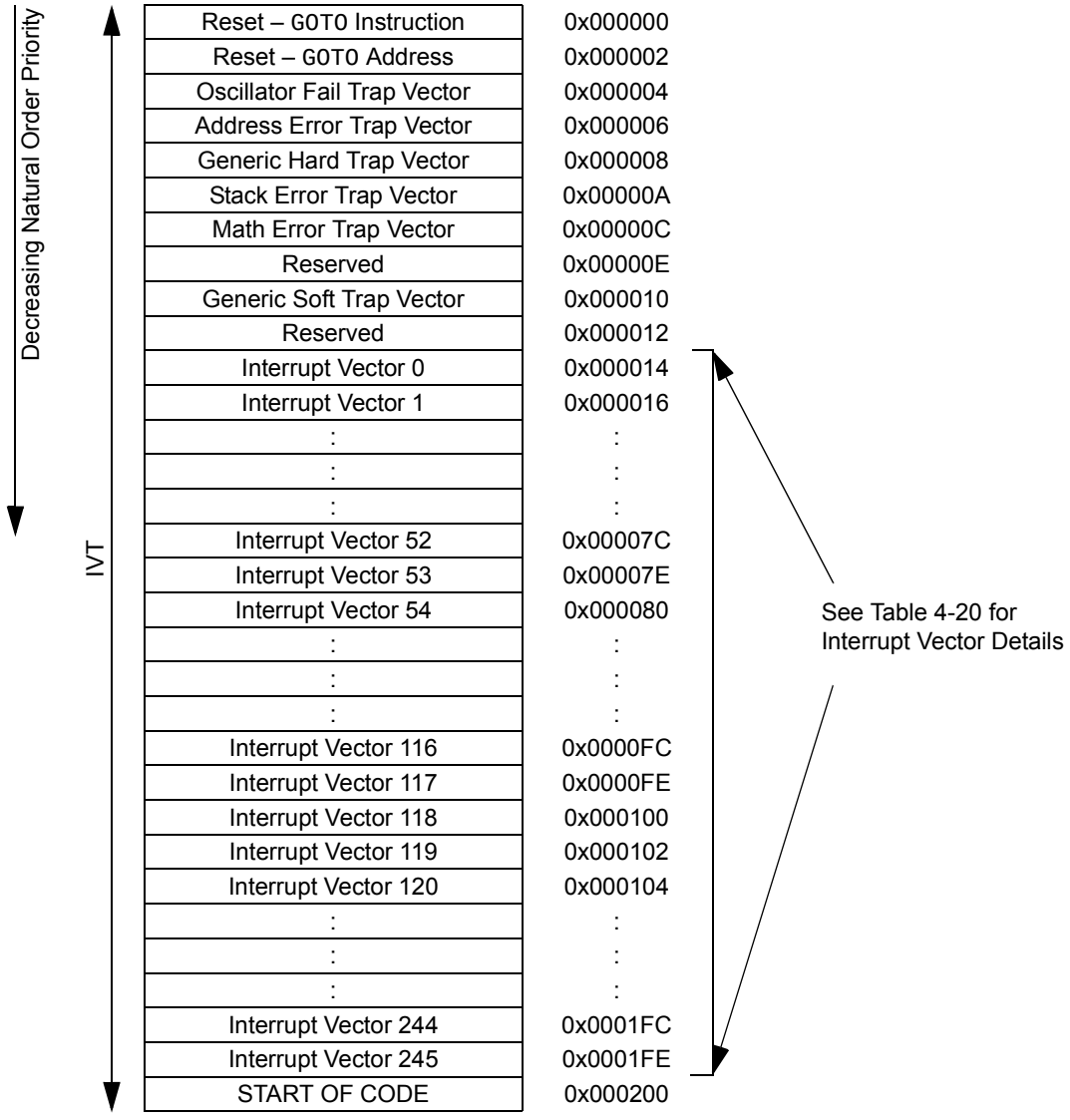
The VFSLV instruction allows the Master user code to verify that the PRAM has been loaded correctly.

Note: Master to Slave Image Loading is the only supported method for programming the Slave PRAM in a final user application.

Regardless of the method used to program the PRAM, a few basic requirements should be met:

- A full 48-bit double instruction word should always be programmed to a PRAM location. Either instruction may simply be a NOP to fulfill this requirement. This ensures a valid ECC value is generated for each pair of instructions written.
- Assuming the above step is followed, the last 24-bit location in implemented program space, or prior to any unprogrammed region in program space, should never be executed. The penultimate instruction in either case must contain a program flow change instruction, such as a RETURN or a BRA instruction.

FIGURE 4-16: dsPIC33CH128MP508S1 FAMILY INTERRUPT VECTOR TABLE



Note: In Dual Partition modes, each partition has a dedicated Interrupt Vector Table.

TABLE 4-20: SLAVE INTERRUPT VECTOR DETAILS (CONTINUED)

Interrupt Source	Vector #	IRQ #	IVT Address	Interrupt Bit Location		
				Flag	Enable	Priority
CND – Change Notice Interrupt D	83	75	0x0000AA	IFS4<11>	IEC4<11>	IPC18<14:12>
CNE – Change Notice Interrupt E	84	76	0x0000AC	IFS4<12>	IEC4<12>	IPC19<2:0>
Reserved	85	77	—	—	—	—
CMP1 – Slave Comparator 1 Interrupt	86	78	0x0000B0	IFS4<14>	IEC4<14>	IPC19<10:8>
CMP2 – Slave Comparator 2 Interrupt	87	79	0x0000B2	IFS4<15>	IEC4<15>	IPC19<14:12>
CMP3 – Slave Comparator 3 Interrupt	88	80	0x0000B4	IFS5<0>	IEC5<0>	IPC20<2:0>
Reserved	89	81	0x0000B6	—	—	—
PTG0 – PTG Int. Trigger Master 0	90	82	0x0000B8	IFS5<2>	IEC5<2>	IPC20<10:8>
PTG1 – PTG Int. Trigger Master 1	91	83	0x0000BA	IFS5<3>	IEC5<3>	IPC20<14:12>
PTG2 – PTG Int. Trigger Master 2	92	84	0x0000BC	IFS5<4>	IEC5<4>	IPC21<2:0>
PTG3 – PTG Int. Trigger Master 3	93	85	0x0000BE	IFS5<5>	IEC5<6>	IPC21<6:4>
Reserved	94-97	86-89	0x0000C0	—	—	—
ADC – ADC Global Interrupt	98	90	0x0000C8	IFS5<10>	IEC5<10>	IPC22<10:8>
ADCAN0 – ADC AN0 Interrupt	99	91	0x0000CA	IFS5<11>	IEC5<11>	IPC22<14:12>
ADCAN1 – ADC AN1 Interrupt	100	92	0x0000CC	IFS5<12>	IEC5<12>	IPC23<2:0>
ADCAN2 – ADC AN2 Interrupt	101	93	0x0000CE	IFS5<13>	IEC5<13>	IPC23<6:4>
ADCAN3 – ADC AN3 Interrupt	102	94	0x0000D0	IFS5<14>	IEC5<14>	IPC23<10:8>
ADCAN4 – ADC AN4 Interrupt	103	95	0x0000D2	IFS5<15>	IEC5<15>	IPC23<14:12>
ADCAN5 – ADC AN5 Interrupt	104	96	0x0000D4	IFS6<0>	IEC6<0>	IPC24<2:0>
ADCAN6 – ADC AN6 Interrupt	105	97	0x0000D6	IFS6<1>	IEC6<1>	IPC24<6:4>
ADCAN7 – ADC AN7 Interrupt	106	98	0x0000D8	IFS6<2>	IEC6<2>	IPC24<10:8>
ADCAN8 – ADC AN8 Interrupt	107	99	0x0000DA	IFS6<3>	IEC6<3>	IPC24<14:12>
ADCAN9 – ADC AN9 Interrupt	108	100	0x0000DC	IFS6<4>	IEC6<4>	IPC25<2:0>
ADCAN10 – ADC AN10 Interrupt	109	101	0x0000DE	IFS6<5>	IEC6<5>	IPC25<6:4>
ADCAN11 – ADC AN11 Interrupt	110	102	0x0000E0	IFS6<6>	IEC6<6>	IPC25<10:8>
ADCAN12 – ADC AN12 Interrupt	111	103	0x0000E2	IFS6<7>	IEC6<7>	IPC25<14:12>
ADCAN13 – ADC AN13 Interrupt	112	104	0x0000E4	IFS6<8>	IEC6<8>	IPC26<2:0>
ADCAN14 – ADC AN14 Interrupt	113	105	0x0000E6	IFS6<9>	IEC6<9>	IPC26<6:4>
ADCAN15 – ADC AN15 Interrupt	114	106	0x0000E8	IFS6<10>	IEC6<10>	IPC26<10:8>
ADCAN16 – ADC AN16 Interrupt	115	107	0x0000EA	IFS6<11>	IEC6<11>	IPC26<14:12>
ADCAN17 – ADC AN17 Interrupt	116	108	0x0000EC	IFS6<12>	IEC6<12>	IPC27<2:0>
ADCAN18 – ADC AN18 Interrupt	117	109	0x0000EE	IFS6<13>	IEC6<13>	IPC27<6:4>
ADCAN19 – ADC AN19 Interrupt	118	110	0x0000F0	IFS6<14>	IEC6<14>	IPC27<10:8>
ADCAN20 – ADC AN20 Interrupt	119	111	0x0000F2	IFS6<15>	IEC6<15>	IPC27<14:12>
Reserved	120-122	112-114	0x0000F4-0x0000F8	—	—	—
ADFLT – ADC Fault	123	115	0x0000FA	IFS7<3>	IEC7<3>	IPC28<14:12>
ADCMP0 – ADC Digital Comparator 0	124	116	0x0000FC	IFS7<4>	IEC7<4>	IPC29<2:0>
ADCMP1 – ADC Digital Comparator 1	125	117	0x0000FE	IFS7<5>	IEC7<5>	IPC29<6:4>
ADCMP2 – ADC Digital Comparator 2	126	118	0x000100	IFS7<6>	IEC7<6>	IPC29<10:8>
ADCMP3 – ADC Digital Comparator 3	127	119	0x000102	IFS7<7>	IEC7<7>	IPC29<14:12>
ADFLTR0 – ADC Oversample Filter 0	128	120	0x000104	IFS7<8>	IEC7<8>	IPC30<2:0>
ADFLTR1 – ADC Oversample Filter 1	129	121	0x000106	IFS7<9>	IEC7<9>	IPC30<6:4>
ADFLTR2 – ADC Oversample Filter 2	130	122	0x000108	IFS7<10>	IEC7<10>	IPC30<10:8>
ADFLTR3 – ADC Oversample Filter 3	131	123	0x00010A	IFS7<11>	IEC7<11>	IPC30<14:12>
CLC1P – CLC1 Positive Edge	132	124	0x00010C	IFS7<12>	IEC7<12>	IPC31<2:0>
CLC2P – CLC2 Positive Edge	133	125	0x00010E	IFS7<13>	IEC7<13>	IPC31<6:4>

REGISTER 4-88: ADCON3H: ADC CONTROL REGISTER 3 HIGH

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CLKSEL1	CLKSEL0	CLKDIV5	CLKDIV4	CLKDIV3	CLKDIV2	CLKDIV1	CLKDIV0
bit 15							bit 8

R/W-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
SHREN	—	—	—	—	—	C1EN	C0EN
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 **CLKSEL<1:0>:** ADC Module Clock Source Selection bits
11 = FVCO/4
10 = AFVCODIV
01 = FOSC
00 = FP (FOSC/2)

bit 13-8 **CLKDIV<5:0>:** ADC Module Clock Source Divider bits
The divider forms a TCORESRC clock used by all ADC cores (shared and dedicated) from the TSRC ADC module clock source selected by the CLKSEL<1:0> bits. Then, each ADC core individually divides the TCORESRC clock to get a core-specific TADCORE clock using the ADCS<6:0> bits in the ADCOREExH register or the SHRADCS<6:0> bits in the ADCON2L register.
111111 = 64 Source Clock Periods
...
000011 = 4 Source Clock Periods
000010 = 3 Source Clock Periods
000001 = 2 Source Clock Periods
000000 = 1 Source Clock Period

bit 7 **SHREN:** Shared ADC Core Enable bit
1 = Shared ADC core is enabled
0 = Shared ADC core is disabled

bit 6-2 **Unimplemented:** Read as '0'

bit 1 **C1EN:** Dedicated ADC Core 1 Enable bits
1 = Dedicated ADC Core 1 is enabled
0 = Dedicated ADC Core 1 is disabled

bit 0 **C0EN:** Dedicated ADC Core 0 Enable bits
1 = Dedicated ADC Core 0 is enabled
0 = Dedicated ADC Core 0 is disabled

REGISTER 4-111: ADFLxCON: ADC DIGITAL FILTER x CONTROL REGISTER (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	HSC/R-0
FLEN	MODE1	MODE0	OVRSAM2	OVRSAM1	OVRSAM0	IE	RDY
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
—	—	—	FLCHSEL4	FLCHSEL3	FLCHSEL2	FLCHSEL1	FLCHSEL0
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 **FLEN:** Filter Enable bit
1 = Filter is enabled
0 = Filter is disabled and the RDY bit is cleared

bit 14-13 **MODE<1:0>:** Filter Mode bits
11 = Averaging mode
10 = Reserved
01 = Reserved
00 = Oversampling mode

bit 12-10 **OVRSAM<2:0>:** Filter Averaging/Oversampling Ratio bits
If MODE<1:0> = 00:
111 = 128x (16-bit result in the ADFLxDAT register is in 12.4 format)
110 = 32x (15-bit result in the ADFLxDAT register is in 12.3 format)
101 = 8x (14-bit result in the ADFLxDAT register is in 12.2 format)
100 = 2x (13-bit result in the ADFLxDAT register is in 12.1 format)
011 = 256x (16-bit result in the ADFLxDAT register is in 12.4 format)
010 = 64x (15-bit result in the ADFLxDAT register is in 12.3 format)
001 = 16x (14-bit result in the ADFLxDAT register is in 12.2 format)
000 = 4x (13-bit result in the ADFLxDAT register is in 12.1 format)
If MODE<1:0> = 11 (12-bit result in the ADFLxDAT register in all instances):
111 = 256x
110 = 128x
101 = 64x
100 = 32x
011 = 16x
110 = 8x
001 = 4x
000 = 2x

bit 9 **IE:** Filter Common ADC Interrupt Enable bit
1 = Common ADC interrupt will be generated when the filter result will be ready
0 = Common ADC interrupt will not be generated for the filter

bit 8 **RDY:** Oversampling Filter Data Ready Flag bit
This bit is cleared by hardware when the result is read from the ADFLxDAT register.
1 = Data in the ADFLxDAT register is ready
0 = The ADFLxDAT register has been read and new data in the ADFLxDAT register is not ready

bit 7-5 **Unimplemented:** Read as '0'

4.8.3 PGA CONTROL REGISTERS

REGISTER 4-112: PGAxCON: PGAx CONTROL 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
PGAEN	PGAOEN	SELPI2	SELPI1	SELPI0	SELNI2	SELNI1	SELNI0
bit 15							bit 8

U-0	U-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	HIGAIN	—	GAIN2	GAIN1	GAIN0
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 **PGAEN:** PGAx Enable bit
1 = PGAx module is enabled
0 = PGAx module is disabled (reduces power consumption)

bit 14 **PGAOEN:** PGAx Output Enable bit
1 = PGAx output is connected to the DACOUT pin
0 = PGAx output is not connected to the DACOUT pin

bit 13-11 **SELPI<2:0>:** PGAx Positive Input Selection bits
111 = Reserved
110 = Reserved
101 = Reserved
100 = Reserved
011 = Ground
010 = Ground
001 = S1PGAxP2
000 = S1PGAxP1

bit 10-8 **SELNI<2:0>:** PGAx Negative Input Selection bits
111 = Reserved
110 = Reserved
101 = Reserved
100 = Reserved
011 = Ground (Single-Ended mode)
010 = Reserved
001 = S1PGAxN2
000 = Ground (Single-Ended mode)

bit 7-5 **Unimplemented:** Read as '0'

bit 4 **HIGAIN:** High-Gain Select bit
This bit, when asserted, enables a 50% increase in gain as specified by the GAIN<2:0> bits.

bit 3 **Unimplemented:** Read as '0'

bit 2-0 **GAIN<2:0>:** PGAx Gain Selection bits
111 = Reserved
110 = Reserved
101 = Gain of 32x
100 = Gain of 16x
011 = Gain of 8x
010 = Gain of 4x
001 = Reserved
000 = Reserved

REGISTER 6-4: OSCTUN: FRC OSCILLATOR TUNING REGISTER (MASTER)

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

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	TUN<5:0>					
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 15-6 **Unimplemented:** Read as '0'

bit 5-0 **TUN<5:0>:** FRC Oscillator Tuning bits

011111 = Maximum frequency deviation of 1.74% (MHz)
011110 = Center frequency + 1.693% (MHz)
...
000001 = Center frequency + 0.047% (MHz)
000000 = Center frequency (8.00 MHz nominal)
111111 = Center frequency – 0.047% (MHz)
...
100001 = Center frequency – 1.693% (MHz)
100000 = Minimum frequency deviation of -1.74% (MHz)

9.0 HIGH-RESOLUTION PWM (HSPWM) WITH FINE EDGE PLACEMENT

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 **"High-Resolution PWM with Fine Edge Placement"** (DS70005320) in the *"dsPIC33/PIC24 Family Reference Manual"*, which is available from the Microchip web site (www.microchip.com).

2: The PWM is identical for both Master core and Slave core. The x is common for both Master core and Slave core (where the x represents the number of the specific module being addressed). The number of HSPWM modules available on the Master core and Slave core is different and they are located in different SFR locations.

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 is PWM1 to PWM4 and the Slave is PWM1 to PWM8.

Table 9-1 shows an overview of the PWM module.

TABLE 9-1: PWM MODULE OVERVIEW

	Number of PWM Modules	Identical (Modules)
Master Core	4	Yes
Slave Core	8	Yes

The High-Speed PWM (HSPWM) module is a Pulse-Width Modulated (PWM) module to support both motor control and power supply applications. This flexible module provides features to support many types of Motor Control (MC) and Power Control (PC) applications, including:

- AC-to-DC Converters
- DC-to-DC Converters
- AC and DC Motors: BLDC, PMSM, ACIM, SRM, etc.
- Inverters
- Battery Chargers
- Digital Lighting
- Power Factor Correction (PFC)

9.1 Features

- Up to Eight Independent PWM Generators for Slave Core, each with Dual Outputs
- Up to Four Independent PWM Generators for Master Core, each with Dual Outputs
- Operating modes:
  - Independent Edge mode
  - Variable Phase PWM mode
  - Center-Aligned mode
  - Double Update Center-Aligned mode
  - Dual Edge Center-Aligned mode
  - Dual PWM mode
- Output modes:
  - Complementary
  - Independent
  - Push-Pull
- Dead-Time Generator
- Leading-Edge Blanking (LEB)
- Output Override for Fault Handling
- Flexible Period/Duty Cycle Updating Options
- Programmable Control Inputs (PCI)
- Advanced Triggering Options
- Six Combinatorial Logic Outputs
- Six PWM Event Outputs

REGISTER 12-11: VELxHLDL: VELOCITY x COUNTER HOLD REGISTER LOW

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
VELHLD<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
VELHLD<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 **VELHLD<15:0>:** Velocity Counter Hold Value bits

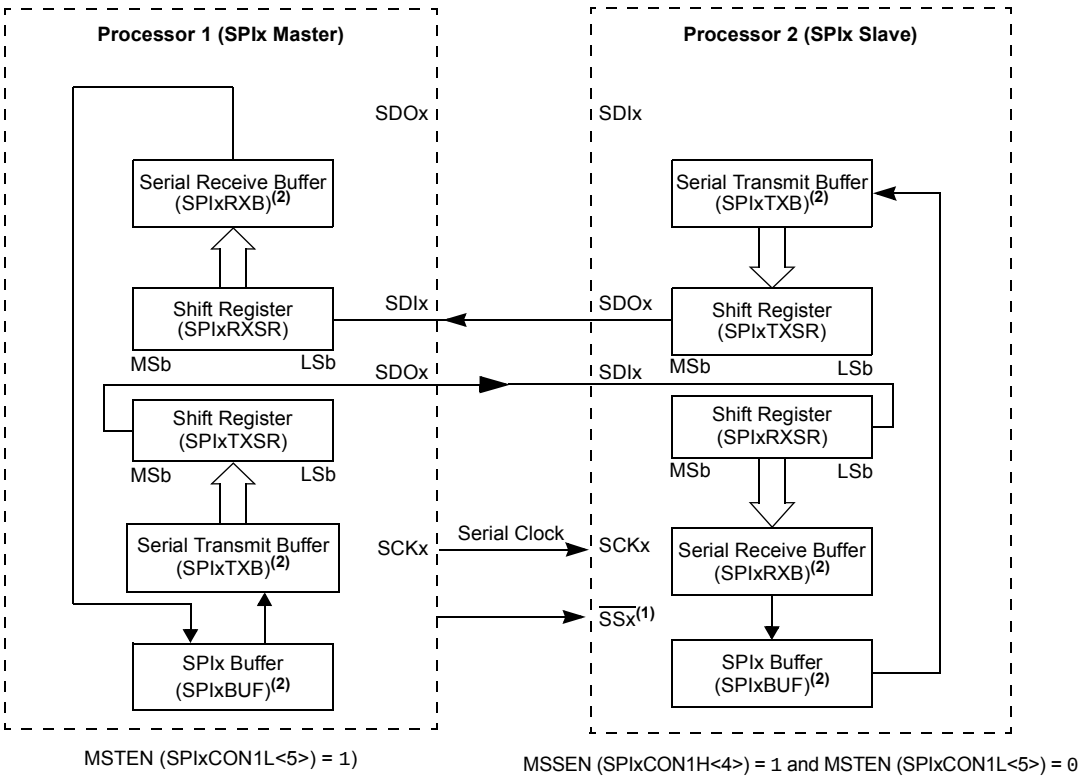
REGISTER 12-12: VELxHLDH: VELOCITY x COUNTER HOLD REGISTER HIGH

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
VELHLD<31:24>							
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
VELHLD<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 **VELHLD<31:16>:** Velocity Counter Hold Value bits

FIGURE 14-3: SPIx MASTER/SLAVE CONNECTION (STANDARD MODE)



Note 1: Using the $\overline{SSx}$ pin in Slave mode of operation is optional.

2: User must write transmit data to read the received data from SPIxBUF. The SPIxTXB and SPIxRXB registers are memory-mapped to SPIxBUF.

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

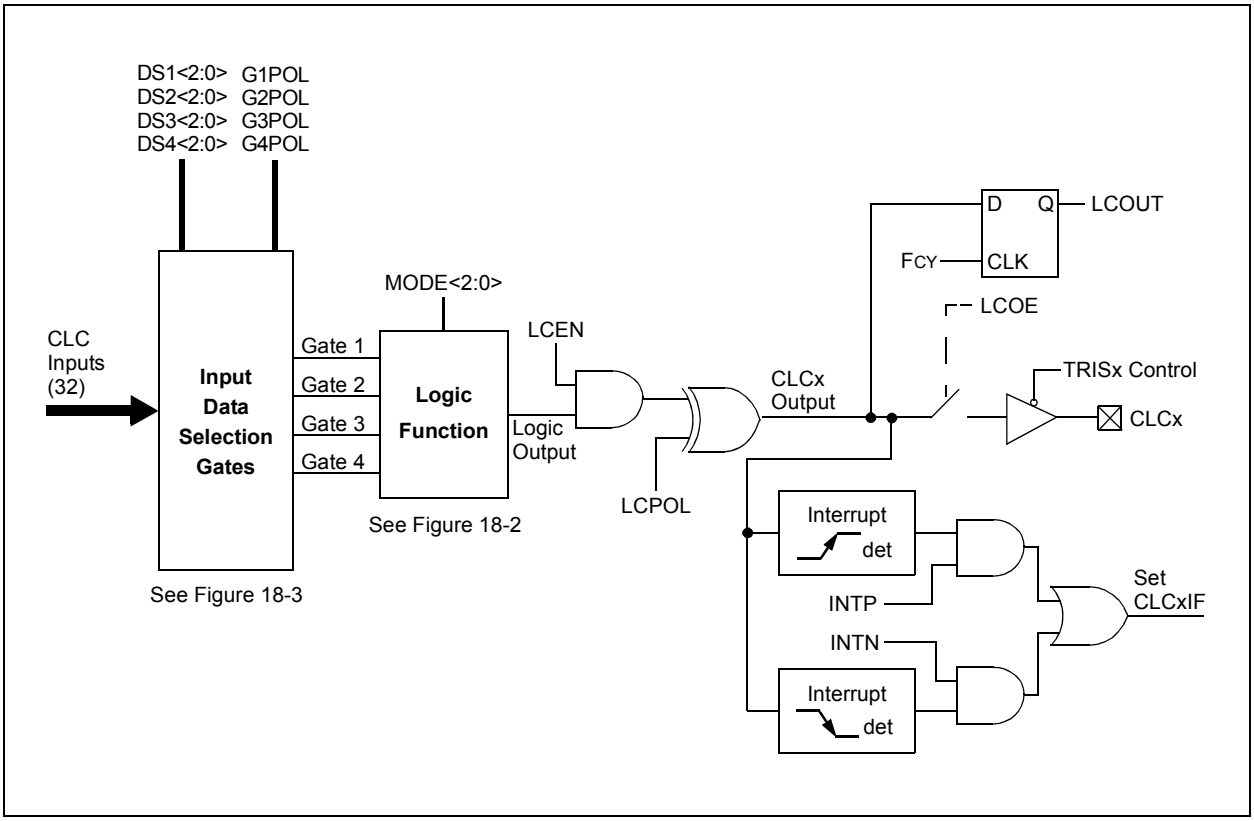
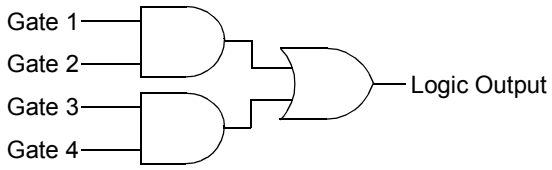


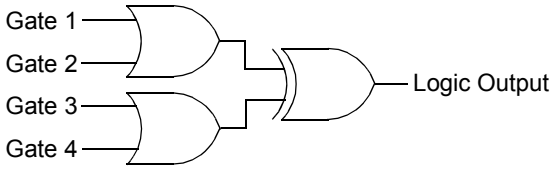
FIGURE 18-2: CLCx LOGIC FUNCTION COMBINATORIAL OPTIONS

AND – OR



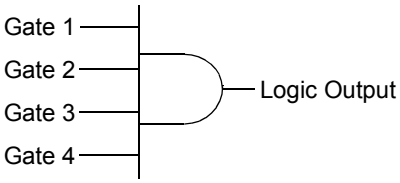
MODE<2:0> = 000

OR – XOR



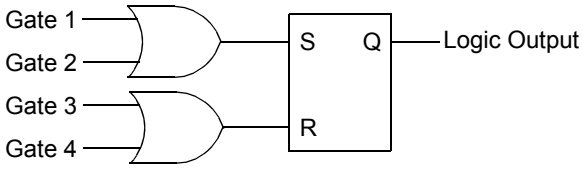
MODE<2:0> = 001

4-Input AND



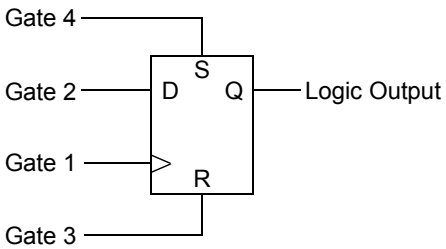
MODE<2:0> = 010

S-R Latch



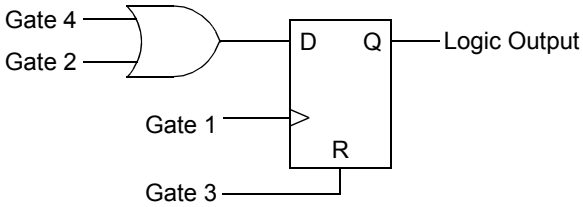
MODE<2:0> = 011

1-Input D Flip-Flop with S and R



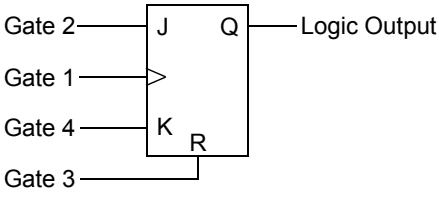
MODE<2:0> = 100

2-Input D Flip-Flop with R



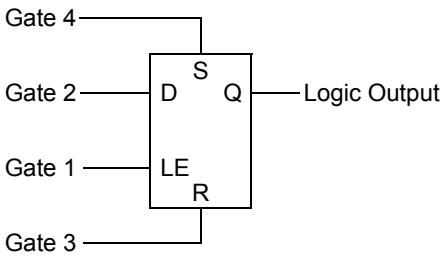
MODE<2:0> = 101

J-K Flip-Flop with R



MODE<2:0> = 110

1-Input Transparent Latch with S and R



MODE<2:0> = 111

APPENDIX A: REVISION HISTORY

Revision A (August 2017)

This is the initial version of the document.

Revision B (June 2018)

This revision incorporates the following updates:

- Registers:
  - Updates Register 3-10, Register 3-13, Register 3-14, Register 3-15, Register 3-102, Register 3-103, Register 3-116, Register 3-117, Register 3-126, Register 3-127, Register 3-129, Register 3-132, Register 3-134, Register 3-135, Register 3-137, Register 3-138, Register 3-162, Register 3-196, Register 4-10, Register 4-11, Register 4-12, Register 4-13, Register 4-14, Register 4-15, Register 4-83 Register 4-86, Register 4-88, Register 10-1, Register 10-5, Register 11-1, Register 11-5, Register 15-3, Register 12-4, Register 12-15, Register 12-16, Register 12-23, Register 12-24, Register 18-3, Register 21-5, Register 21-14, Register 21-26, Register 21-33, Register 21-34, Register 21-35 and Register 21-37.
  - Deletes ADCSSL: ADC CVD Scan Select Register Low, FOSCSEL: Oscillator Source Selection Register, FOSC: Oscillator Configuration Register, FS1OSCSEL: Slave Oscillator Source Selection Register and FS1OSC: Slave Oscillator Configuration Register.
- Tables:
  - Updates Table 2, Table 3, Table 4, Table 5, Table 6, Table 7, Table 8, Table 9, Table 1-1, Table 3-4-Table 3-18 (adds additional information to the legend), Table 3-27, Table 3-35, Table 3-36, Table 3-37, Table 3-45, Table 4-3-Table 4-15 (adds additional information to the legend), Table 4-24, Table 4-33 through Table 4-37, Table 15-1, Table 21-2, Table 21-5, Table 22-2, Table 24-3, Table 24-5, Table 24-6, Table 24-7, Table 24-8, Table 24-9, Table 24-10, Table 24-11, Table 24-12, Table 24-13, Table 24-15, Table 24-16 Table 24-14, Table 24-17, Table 24-22, Table 24-29, Table 24-34-Table 24-40, Table 24-41, Table 24-44, Table 24-45 and Table 24-48.
  - Adds Table 24-13 through Table 24-17.
- Figures:
  - Updates Figure 3-24, Figure 3-26, Figure 4-7, Figure 4-20, Figure 14-5, Figure 14-6, Figure 14-7, Figure 14-8, Figure 20-1, Figure 21-2 and Figure .
- Sections:
  - Adds **"Referenced Sources"** section to front matter.
- Miscellaneous:
  - Adds headings to all SFR and Register tables.
  - Adds Error Correcting Code (ECC) information.
  - Adds the 48-Lead UQFN package to the document.
  - Removes External Count with External Gate information.