Microchip Technology - DSPIC33CH64MP508T-I/PT 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	69
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 34x12b; D/A 4x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	80-TQFP
Supplier Device Package	80-TQFP (12x12)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/dspic33ch64mp508t-i-pt

FIGURE 1-1: SLAVE CORE CODE TRANSFER BLOCK DIAGRAM

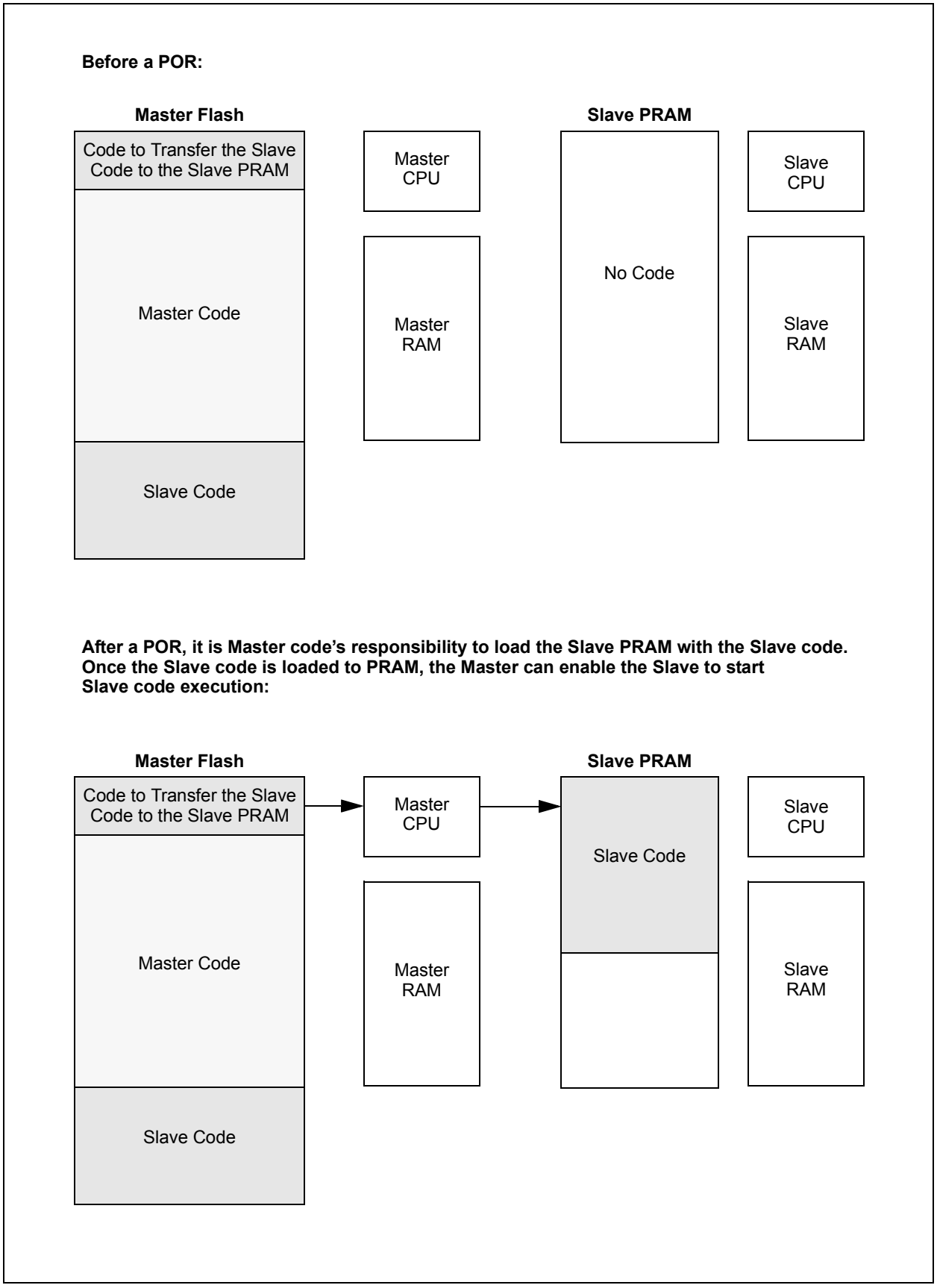


TABLE 3-15: MASTER SFR BLOCK C00h

Register	Address	All Resets	Register	Address	All Resets	Register	Address	All Resets
ADC (Continued)			ADCBUF9	C1E	0000000000000000	**DAC**		
ADCON5L	C00	0-------0-------	ADCBUF10	C20	0000000000000000	DACCTRL1L	C80	000-----0000-000
ADCON5H	C02	0---xxxx0-------	ADCBUF11	C22	0000000000000000	DACCTRL2L	C84	------0001010101
ADCAL1H	C0A	00000-00-000----	ADCBUF12	C24	0000000000000000	DACCTRL2H	C86	------0010001010
ADCBUF0	C0C	0000000000000000	ADCBUF13	C26	0000000000000000	DAC1CONL	C88	000--000x0000000
ADCBUF1	C0E	0000000000000000	ADCBUF14	C28	0000000000000000	DAC1CONH	C8A	------0000000000
ADCBUF2	C10	0000000000000000	ADCBUF15	C2A	0000000000000000	DAC1DATL	C8C	0000000000000000
ADCBUF3	C12	0000000000000000	ADCBUF16	C2C	0000000000000000	DAC1DATH	C8E	0000000000000000
ADCBUF4	C14	0000000000000000	ADCBUF17	C2E	0000000000000000	SLP1CONL	C90	0000000000000000
ADCBUF5	C16	0000000000000000	ADCBUF18	C30	0000000000000000	SLP1CONH	C92	0---000---------
ADCBUF6	C18	0000000000000000	ADCBUF19	C32	0000000000000000	SLP1DAT	C94	0000000000000000
ADCBUF7	C1A	0000000000000000	ADCBUF20	C34	0000000000000000	VREGCON	CFC	0---------000000
ADCBUF8	C1C	0000000000000000						

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

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 3-7: NVMKEY: NONVOLATILE MEMORY KEY REGISTER

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

W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
NVMKEY<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 **Unimplemented:** Read as ‘0’

bit 7-0 **NVMKEY<7:0>:** NVM Key Register bits (write-only)

REGISTER 3-8: NVMSRCADR: NVM SOURCE DATA ADDRESS 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
NVMSRCADR<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
NVMSRCADR<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 **NVMSRCADR<15:0>:** NVM Source Data Address bits

The RAM address of the data to be programmed into Flash when the NVMOP<3:0> bits are set to row programming.

TABLE 3-38: PORTE REGISTER SUMMARY

ANSLE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
TRISE	TRISE<15:0>																
PORTE	RE<15:0>																
LATE	LATE<15:0>																
ODCE	ODCE<15:0>																
CNPUE	CNPUE<15:0>																
CNPDE	CNPDE<15:0>																
CNCONE	ON	—	—	—	CNSTYLE	—	—	—	—	—	—	—	—	—	—	—	—
CNEN0E	CNEN0E<15:0>																
CNSTATE	CNSTATE<15:0>																
CNEN1E	CNEN1E<15:0>																
CNFE	CNFE<15:0>																

REGISTER 3-126: C1TXREQH: CAN TRANSMIT REQUEST REGISTER HIGH

S/HC-0	S/HC-0	S/HC-0	S/HC-0	S/HC-0	S/HC-0	S/HC-0	S/HC-0
TXREQ<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	S/HC-0
TXREQ<23:16>							
bit 7							bit 0

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

bit 15-0 **TXREQ<31:16>:** Unimplemented

REGISTER 3-127: C1TXREQL: CAN TRANSMIT REQUEST REGISTER LOW

S/HC-0	S/HC-0	S/HC-0	S/HC-0	S/HC-0	S/HC-0	S/HC-0	S/HC-0
TXREQ<15:8>							
bit 15							bit 8

S/HC-0	S/HC-0	S/HC-0	S/HC-0	S/HC-0	S/HC-0	S/HC-0	S/HC-0s
TXREQ<7:1>							TXREQ0
bit 7							bit 0

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

bit 15-8 **TXREQ<15:8>:** Unimplemented

bit 7-1 **TXREQ<7:1>:** Message Send Request bits

TXEN = 1 (object configured as a transmit object):
Setting this bit to '1' requests sending a message. The bit will automatically clear when the message(s) queued in the object is (are) successfully sent. This bit can NOT be used for aborting a transmission.

TXEN = 0 (object configured as a receive object):
This bit has no effect.

bit 0 **TXREQ0:** Transmit Queue Message Send Request bit

Setting this bit to '1' requests sending a message. The bit will automatically clear when the message(s) queued in the object is (are) successfully sent. This bit can NOT be used for aborting a transmission.

REGISTER 3-141: C1TEFUAH: CAN TRANSMIT EVENT FIFO USER ADDRESS REGISTER HIGH(1)

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

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
TEFUA<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 **TEFUA<31:16>:** Transmit Event FIFO User Address bits
A read of this register will return the address where the next event 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 3-142: C1TEFUAL: CAN TRANSMIT EVENT FIFO USER ADDRESS REGISTER LOW(1)

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

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
TEFUA<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 **TEFUA<15:0>:** Transmit Event FIFO User Address bits
A read of this register will return the address where the next event 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 3-167: ADEIEL: ADC EARLY INTERRUPT ENABLE 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
EIEN<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
EIEN<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 **EIEN<15:0>:** Early Interrupt Enable for Corresponding Analog Input bits
1 = Early interrupt is enabled for the channel
0 = Early interrupt is disabled for the channel

REGISTER 3-168: ADEIEH: ADC EARLY INTERRUPT ENABLE 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	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	EIEN<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 **EIEN<20:16>:** Early Interrupt Enable for Corresponding Analog Input bits
1 = Early interrupt is enabled for the channel
0 = Early interrupt is disabled for the channel

4.2.2.5 X and Y Data Spaces

The dsPIC33CH128MP508S1 family core has two Data Spaces, X and Y. These Data Spaces can be considered either separate (for some DSP instructions) or as one unified linear address range (for MCU instructions). The Data Spaces are accessed using two Address Generation Units (AGUs) and separate data paths. This feature allows certain instructions to concurrently fetch two words from RAM, thereby enabling efficient execution of DSP algorithms, such as Finite Impulse Response (FIR) filtering and Fast Fourier Transform (FFT).

The X Data Space is used by all instructions and supports all addressing modes. X Data Space has separate read and write data buses. The X read data bus is the read data path for all instructions that view Data Space as combined X and Y address space. It is also the X data prefetch path for the dual operand DSP instructions (`MAC` class).

The Y Data Space is used in concert with the X Data Space by the `MAC` class of instructions (`CLR`, `ED`, `EDAC`, `MAC`, `MOVSAC`, `MPY`, `MPY.N` and `MSC`) to provide two concurrent data read paths.

Both the X and Y Data Spaces support Modulo Addressing mode for all instructions, subject to addressing mode restrictions. Bit-Reversed Addressing mode is only supported for writes to X Data Space.

All data memory writes, including in DSP instructions, view Data Space as combined X and Y address space. The boundary between the X and Y Data Spaces is device-dependent and is not user-programmable.

4.2.3 MEMORY RESOURCES

Many useful resources are provided on the main product page of the Microchip web site for the devices listed in this data sheet. This product page contains the latest updates and additional information.

4.2.3.1 Key Resources

- **"dsPIC33E/PIC24E Program Memory"** (DS70000613) in the *"dsPIC33/PIC24 Family Reference Manual"*
- Code Samples
- Application Notes
- Software Libraries
- Webinars
- All Related *"dsPIC33/PIC24 Family Reference Manual"* Sections
- Development Tools

TABLE 4-7: SLAVE SFR BLOCK 400h

Register	Address	All Resets	Register	Address	All Resets	Register	Address	All Resets
High-Speed PWM (Continued)			PG6CLPCIL	44A	0000000000000000	PG7DC	492	0000000000000000
PG5CONL	402	0-00000000000000	PG6CLPCIH	44C	0000-00000000000	PG7DCA	494	--------00000000
PG5CONH	404	000-000000--0000	PG6FFPCIL	44E	0000000000000000	PG7PER	496	0000000000000000
PG5STAT	406	0000000000000000	PG6FFPCIH	450	0000-00000000000	PG7TRIGA	498	0000000000000000
PG5IOCONL	408	0000000000000000	PG6SPCIL	452	0000000000000000	PG7TRIGB	49A	0000000000000000
PG5IOCONH	40A	-000---0--000000	PG6SPCIH	454	0000-00000000000	PG7TRIGC	49C	0000000000000000
PG5EVTL	40C	00000000---00000	PG6LEBL	456	0000000000000000	PG7DTL	49E	--00000000000000
PG5EVTH	40E	0000--0000000000	PG6LEBH	458	-----000----0000	PG7DTH	4A0	--00000000000000
PG5FPCIL	410	0000000000000000	PG6PHASE	45A	0000000000000000	PG7CAP	4A2	0000000000000000
PG5FPCIH	412	0000-00000000000	PG6DC	45C	0000000000000000	PG8CONL	4A4	0-00000000000000
PG5CLPCIL	414	0000000000000000	PG6DCA	45E	--------00000000	PG8CONH	4A6	000-000000--0000
PG5CLPCIH	416	0000-00000000000	PG6PER	460	0000000000000000	PG8STAT	4A8	0000000000000000
PG5FFPCIL	418	0000000000000000	PG6TRIGA	462	0000000000000000	PG8IOCONL	4AA	0000000000000000
PG5FFPCIH	41A	0000-00000000000	PG6TRIGB	464	0000000000000000	PG8IOCONH	4AC	-000---0--000000
PG5SPCIL	41C	0000000000000000	PG6TRIGC	466	0000000000000000	PG8EVTL	4AE	00000000---00000
PG5SPCIH	41E	0000-00000000000	PG6DTL	468	--00000000000000	PG8EVTH	4B0	0000--0000000000
PG5LEBL	420	0000000000000000	PG6DTH	46A	--00000000000000	PG8FPCIL	4B2	0000000000000000
PG5LEBH	422	-----000----0000	PG6CAP	46C	0000000000000000	PG8FPCIH	4B4	0000-00000000000
PG5PHASE	424	0000000000000000	PG7CONL	46E	0-00000000000000	PG8CLPCIL	4B6	0000000000000000
PG5DC	426	0000000000000000	PG7CONH	470	000-000000--0000	PG8CLPCIH	4B8	0000-00000000000
PG5DCA	428	--------00000000	PG7STAT	472	0000000000000000	PG8FFPCIL	4BA	0000000000000000
PG5PER	42A	0000000000000000	PG7IOCONL	474	0000000000000000	PG8FFPCIH	4BC	0000-00000000000
PG5TRIGA	42C	0000000000000000	PG7IOCONH	476	-000---0--000000	PG8SPCIL	4BE	0000000000000000
PG5TRIGB	42E	0000000000000000	PG7EVTL	478	00000000---00000	PG8SPCIH	4C0	0000-00000000000
PG5TRIGC	430	0000000000000000	PG7EVTH	47A	0000--0000000000	PG8LEBL	4C2	0000000000000000
PG5DTL	432	--00000000000000	PG7FPCIL	47C	0000000000000000	PG8LEBH	4C4	-----000----0000
PG5DTH	434	--00000000000000	PG7FPCIH	47E	0000-00000000000	PG8PHASE	4C6	0000000000000000
PG5CAP	436	0000000000000000	PG7CLPCIL	480	0000000000000000	PG8DC	4C8	0000000000000000
PG6CONL	438	0-00000000000000	PG7CLPCIH	482	0000-00000000000	PG8DCA	4CA	--------00000000
PG6CONH	43A	000-000000--0000	PG7FFPCIL	484	0000000000000000	PG8PER	4CC	0000000000000000
PG6STAT	43C	0000000000000000	PG7FFPCIH	486	0000-00000000000	PG8TRIGA	4CE	0000000000000000
PG6IOCONL	43E	0000000000000000	PG7SPCIL	488	0000000000000000	PG8TRIGB	4D0	0000000000000000
PG6IOCONH	440	-000---0--000000	PG7SPCIH	48A	0000-00000000000	PG8TRIGC	4D2	0000000000000000
PG6EVTL	442	00000000---00000	PG7LEBL	48C	0000000000000000	PG8DTL	4D4	--00000000000000
PG6EVTH	444	0000--0000000000	PG7LEBH	48E	-----000----0000	PG8DTH	4D6	--00000000000000
PG6FPCIL	446	0000000000000000	PG7PHASE	490	0000000000000000	PG8CAP	4D8	0000000000000000
PG6FPCIH	448	0000-00000000000						

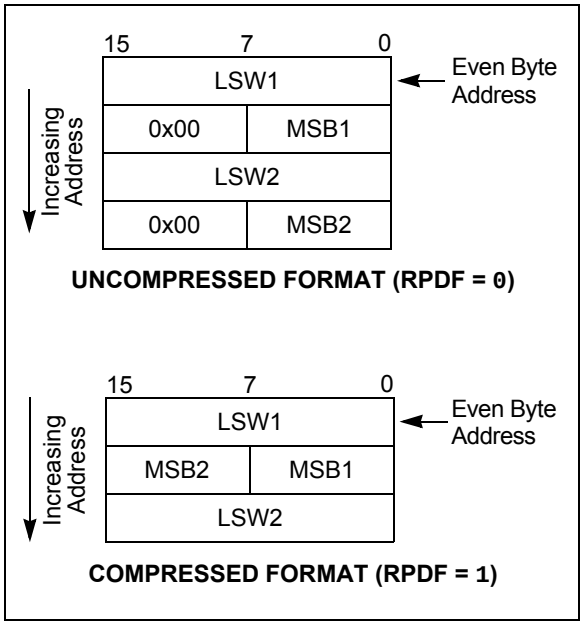
Legend: x = unknown or indeterminate value; "-" = unimplemented bits. Address and Reset values are in hexadecimal and binary, respectively.

Row programming is performed by first loading 128 instructions into data RAM and then loading the address of the first instruction in that row into the NVMSRCADRL/H register. Once the write has been initiated, the device will automatically load two instructions into the write latches, and write them to the program space destination address defined by the NVMADR/U registers.

The operation will increment the NVMSRCADRL/H and the NVMADR/U registers until all double instruction words have been programmed.

The RPDF bit (NVMCON<9>) selects the format of the stored data in RAM to be either compressed or uncompressed. See Figure 4-14 for data formatting. Compressed data helps to reduce the amount of required RAM by using the upper byte of the second word for the MSB of the second instruction.

FIGURE 4-14: UNCOMPRESSED/ COMPRESSED FORMAT



4.3.3 MASTER TO SLAVE IMAGE LOADING (MSIL)

Master to Slave Image Loading allows the Master user application code to transfer the Slave image stored in the Master Flash to the Slave PRAM. This is the only supported method for programming the Slave PRAM in a final user application.

The LDSLV instruction is executed by the Master user application to transfer a single 24-bit instruction from the Master Flash address, defined by Ws<14:0> (DSRPAG), to the Slave PRAM address, defined by Wd<14:0> (DSWPAG).

The LDSLV instruction should be executed in pairs to ensure correct ECC value generation for each double instruction word that is loaded into the Slave PRAM. The Slave image instruction found at a given even address should be loaded first. This will be the lower instruction word of a 48-bit double instruction word. The upper instruction word should then be loaded from the following odd address. After the pair of LDSLV instructions is executed by the Master user application, both 24-bit Slave image instructions and the generated 7-bit ECC value are actually loaded into the PRAM destination address locations.

The VFSLV instruction allows the Master user application to verify that the PRAM has been loaded correctly. The VFSLV instruction compares the 24-bit instruction word stored in the Master Flash address, defined by Ws<14:0> (DSRPAG), to the 24 bit instruction written to the Slave PRAM address, defined by Wd<14:0> (DSWPAG).

The VFSLV instruction should also be executed in pairs. The lower instruction word found on a given even address should be verified first. The upper instruction word found in the following odd address should then be verified. Then, the Slave image instruction pair read from the Master Flash will have a valid generated ECC value. This full double instruction word with ECC is then compared to the 55-bit value that was actually loaded into the PRAM destination locations. The entire Slave image may be loaded into the PRAM first and then subsequently verified. To make this process simpler, the Microchip `libpic30.h` library has implemented a routine which can be called once to either load or verify the entire Slave image.

The `__program_slave(core#, verify, &slave_image)` routine uses the “verify” parameter to determine if the routine will run using LDSLV instructions or VFSLV instructions. A ‘0’ will load the entire Slave image to the PRAM and a ‘1’ will verify the entire Slave image in the PRAM. An example of how this routine may be used to load and verify the contents of the Slave PRAM is shown in Example 4-2.

TABLE 4-23: SLAVE INTERRUPT PRIORITY REGISTERS

Register	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
IPC0	—	CNBIP2	CNBIP1	CNBIP0	—	CNAIP2	CNAIP1	CNAIP0	—	T1IP2	T1IP1	T1IP0	—	INT0IP2	INT0IP1	INT0IP0
IPC1	—	CCT1IP2	CCT1IP1	CCT1IP0	—	CCP1IP2	CCP1IP1	CCP1IP0	—	—	—	—	—	DMA0IP2	DMA0IP1	DMA0IP0
IPC2	—	U1RXIP2	U1RXIP1	U1RXIP0	—	SPI1TXIP2	SPI1TXIP1	SPI1TXIP0	—	SPI1RXIP2	SPI1RXIP1	SPI1RXIP0	—	DMA1IP2	DMA1IP1	DMA1IP0
IPC3	—	INT1IP2	INT1IP1	INT1IP0	—	NVMIP2	NVMIP1	NVMIP0	—	ECCSBEIP2	ECCSBEIP1	ECCSBEIP0	—	U1TXIP2	U1TXIP1	U1TXIP0
IPC4	—	CNCIP2	CNCIP1	CNCIP0	—	—	—	—	—	MI2C1IP2	MI2C1IP1	MI2C1IP0	—	SI2C1IP2	SI2C1IP1	SI2C1IP0
IPC5	—	CCP2IP2	CCP2IP1	CCP2IP0	—	—	—	—	—	—	—	—	—	INT2IP2	INT2IP1	INT2IP0
IPC6	—	—	—	—	—	INT3IP2	INT3IP1	INT3IP0	—	—	—	—	—	CCT2IP2	CCT2IP1	CCT2IP0
IPC7	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
IPC8	—	CCP3IP2	CCP3IP1	CCP3IP0	—	—	—	—	—	—	—	—	—	—	—	—
IPC9	—	—	—	—	—	—	—	—	—	—	—	—	—	CCT3IP2	CCT3IP1	CCT3IP0
IPC10	—	—	—	—	—	—	—	—	—	CCT4IP2	CCT4IP1	CCT4IP0	—	CCP4IP2	CCP4IP1	CCP4IP0
IPC11	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
IPC12	—	—	—	—	—	—	—	—	—	U1EIP2	U1EIP1	U1EIP0	—	QEI1IP2	QEI1IP1	QEI1IP0
IPC13	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
IPC14	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
IPC15	—	—	—	—	—	JTAGIP2	JTAGIP1	JTAGIP0	—	ICDIP2	ICDIP1	ICDIP0	—	—	—	—
IPC16	—	PWM1IP2	PWM1IP1	PWM1IP0	—	—	—	—	—	—	—	—	—	I2C1BCIP2	I2C1BCIP1	I2C1BCIP0
IPC17	—	PWM5IP2	PWM5IP1	PWM5IP0	—	PWM4IP2	PWM4IP1	PWM4IP0	—	PWM3IP2	PWM3IP1	PWM3IP0	—	PWM2IP2	PWM2IP1	PWM2IP0
IPC18	—	CNDIP2	CNDIP1	CNDIP0	—	PWM8IP2	PWM8IP1	PWM8IP0	—	PWM7IP2	PWM7IP1	PWM7IP0	—	PWM6IP2	PWM6IP1	PWM6IP0
IPC19	—	CMP2IP2	CMP2IP1	CMP2IP0	—	CMP1IP2	CMP1IP1	CMP1IP0	—	—	—	—	—	CNEIP2	CNEIP1	CNEIP0
IPC20	—	PTG1IP2	PTG1IP1	PTG1IP0	—	PTG0IP2	PTG0IP1	PTG0IP0	—	—	—	—	—	CMP3IP2	CMP3IP1	CMP3IP0
IPC21	—	—	—	—	—	—	—	—	—	PTG3IP2	PTG3IP1	PTG3IP0	—	PTG12P2	PTG12P1	PTG12P0
IPC22	—	ADCAN0IP2	ADCAN0IP1	ADCAN0IP0	—	ADCIP2	ADCIP1	ADCIP	—	—	—	—	—	—	—	—
IPC23	—	ADCAN4IP2	ADCAN4IP1	ADCAN4IP0	—	ADCAN3IP2	ADCAN3IP1	ADCAN3IP0	—	ADCAN2IP2	ADCAN2IP1	ADCAN2IP0	—	ADCAN1IP2	ADCAN1IP1	ADCAN1IP0
IPC24	—	ADCAN8IP2	ADCAN8IP1	ADCAN8IP0	—	ADCAN7IP2	ADCAN7IP1	ADCAN7IP0	—	ADCAN6IP2	ADCAN6IP1	ADCAN6IP0	—	ADCAN5IP2	ADCAN5IP1	ADCAN5IP0
IPC25	—	ADCAN12IP2	ADCAN12IP1	ADCAN12IP0	—	ADCAN11IP2	ADCAN11IP1	ADCAN11IP0	—	ADCAN10IP2	ADCAN10IP1	ADCAN10IP0	—	ADCAN9IP2	ADCAN9IP1	ADCAN9IP0
IPC26	—	ADCAN16IP2	ADCAN16IP1	ADCAN16IP0	—	ADCAN15IP2	ADCAN15IP1	ADCAN15IP0	—	ADCAN14IP2	ADCAN14IP1	ADCAN14IP0	—	ADCAN13IP2	ADCAN13IP1	ADCAN13IP0
IPC27	—	ADCAN20IP2	ADCAN20IP1	ADCAN20IP0	—	ADCAN19IP2	ADCAN19IP1	ADCAN19IP0	—	ADCAN18IP2	ADCAN18IP1	ADCAN18IP0	—	ADCAN17IP2	ADCAN17IP1	ADCAN17IP0
IPC28	—	ADFLTIP2	ADFLTIP1	ADFLTIP0	—	—	—	—	—	—	—	—	—	ADCAN21IP2	ADCAN21IP1	ADCAN21IP0
IPC29	—	ADCMP3IP2	ADCMP3IP1	ADCMP3IP0	—	ADCMP2IP2	ADCMP2IP1	ADCMP2IP0	—	ADCMP1IP2	ADCMP1IP1	ADCMP1IP0	—	ADCMP0IP2	ADCMP0IP1	ADCMP0IP0
IPC30	—	ADFLTR3IP2	ADFLTR3IP1	ADFLTR3IP0	—	ADFLTR2IP2	ADFLTR2IP1	ADFLTR2IP0	—	ADFLTR1IP2	ADFLTR1IP1	ADFLTR1IP0	—	ADFLTR0IP2	ADFLTR0IP1	ADFLTR0IP0
IPC31	—	—	—	—	—	SPI1IP2	SPI1IP1	SPI1IP0	—	CLC2PEIP2	CLC2PEIP1	CLC2PEIP0	—	CLC1PEIP2	CLC1PEIP1	CLC1PEIP0
IPC32	—	MSIBIP2	MSIBIP1	MSIBIP0	—	MSIAIP2	MSIAIP1	MSIAIP0	—	MSMIP2	MSMIP1	MSMIP0	—	—	—	—
IPC33	—	MSIFIP2	MSIFIP1	MSIFIP0	—	MSIEIP2	MSIEIP1	MSIEIP0	—	MSIDIP2	MSIDIP1	MSIDIP0	—	MSICIP2	MSICIP1	MSICIP0
IPC34	—	MSIWFEIP2	MSIWFEIP1	MSIWFEIP0	—	MSIDTIP2	MSIDTIP1	MSIDTIP0	—	MSIHIP2	MSIHIP1	MSIHIP0	—	MSIGIP2	MSIGIP1	MSIGIP0
IPC35	—	—	—	—	—	—	—	—	—	MSIMRSTIP2	MSIMRSTIP1	MSIMRSTIP0	—	MSIFLTIP2	MSIFLTIP1	MSIFLTIP0

REGISTER 7-10: PMD1: SLAVE PERIPHERAL MODULE DISABLE 1 CONTROL REGISTER

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	U-0
—	—	—	—	T1MD	QEIMD	PWMMD	—
bit 15							bit 8

R/W-0	U-0	R/W-0	U-0	R/W-0	U-0	U-0	R/W-0
I2C1MD	—	U1MD	—	SPI1MD	—	—	ADC1MD
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 **T1MD:** Timer1 Module Disable bit
1 = Timer1 module is disabled
0 = Timer1 module is enabled

bit 10 **QEIMD:** QEI Module Disable bit
1 = QEI module is disabled
0 = QEI module is enabled

bit 9 **PWMMD:** PWM Module Disable bit
1 = PWM module is disabled
0 = PWM module is enabled

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

bit 7 **I2C1MD:** I2C1 Module Disable bit
1 = I2C1 module is disabled
0 = I2C1 module is enabled

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

bit 5 **U1MD:** UART1 Module Disable bit
1 = UART1 module is disabled
0 = UART1 module is enabled

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

bit 3 **SPI1MD:** SPI1 Module Disable bit
1 = SPI1 module is disabled
0 = SPI1 module is enabled

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

bit 0 **ADC1MD:** ADC Module Disable bit
1 = ADC module is disabled
0 = ADC module is enabled

13.0 UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMITTER (UART)

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 **"Multiprotocol Universal Asynchronous Receiver Transmitter (UART) Module"** (DS70005288) in the *"dsPIC33/PIC24 Family Reference Manual"*, which is available from the Microchip web site (www.microchip.com).

2: The UART 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 UART 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 UART is UART1 and UART2, and the Slave UART is UART1.

Table 13-1 shows an overview of the module.

TABLE 13-1: UART MODULE OVERVIEW

	Number of UART Modules	Identical (Modules)
Master Core	2	Yes
Slave Core	1	Yes

The Universal Asynchronous Receiver Transmitter (UART) is a flexible serial communication peripheral used to interface dsPIC® microcontrollers with other equipment, including computers and peripherals. The UART is a full-duplex, asynchronous communication channel that can be used to implement protocols, such as RS-232 and RS-485. The UART also supports the following hardware extensions:

- LIN/J2602
- IrDA®
- Direct Matrix Architecture (DMX)
- Smart Card

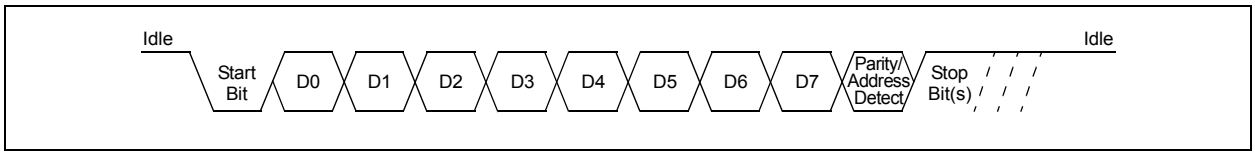
The primary features of the UART are:

- Full or Half-Duplex Operation
- Up to 8-Deep TX and RX First In, First Out (FIFO) Buffers
- 8-Bit or 9-Bit Data Width
- Configurable Stop Bit Length
- Flow Control
- Auto-Baud Calibration
- Parity, Framing and Buffer Overrun Error Detection
- Address Detect
- Break Transmission
- Transmit and Receive Polarity Control
- Manchester Encoder/Decoder
- Operation in Sleep mode
- Wake from Sleep on Sync Break Received Interrupt

13.2 Character Frame

A typical UART character frame is shown in Figure 13-2. The Idle state is high with a ‘Start’ condition indicated by a falling edge. The Start bit is followed by the number of data, parity/address detect and Stop bits defined by the MOD<3:0> (UxMODE<3:0>) bits selected.

FIGURE 13-2: UART CHARACTER FRAME



13.3 Data Buffers

Both transmit and receive functions use buffers to store data shifted to/from the pins. These buffers are FIFOs and are accessed by reading the SFRs, UxTXREG and UxRXREG, respectively. Each data buffer has multiple flags associated with its operation to allow software to read the status. Interrupts can also be configured based on the space available in the buffers. The transmit and receive buffers can be cleared and their pointers reset using the associated TX/RX Buffer Empty Status bits, UTXBE (UxSTAH<5>) and URXBE (UxSTAH<1>).

13.4 Protocol Extensions

The UART provides hardware support for LIN/J2602, IrDA®, DMX and smart card protocol extensions to reduce software overhead. A protocol extension is enabled by writing a value to the MOD<3:0> (UxMODE<3:0>) selection bits and further configured using the UARTx Timing Parameter registers, UxP1 (Register 13-9), UxP2 (Register 13-10), UxP3 (Register 13-11) and UxP3H (Register 13-12). Details regarding operation and usage are discussed in their respective chapters. Not all protocols are available on all devices.

REGISTER 13-11: UxP3: UARTx TIMING PARAMETER 3 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
P3<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
P3<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 **P3<15:0>:** Parameter 3 bits

DMX RX:
The last byte number to receive – 1, not including Start code (bits<8:0>).

LIN Slave RX:
Number of bytes to receive (bits<7:0>).

Asynchronous RX:
Used to mask the UxP2 address bits; 1 = P2 address bit is used, 0 = P2 address bit is masked off (bits<7:0>).

Smart Card Mode:
Waiting Time Counter bits (bits<15:0>).

Other Modes:
Not used.

REGISTER 13-12: UxP3H: UARTx TIMING PARAMETER 3 REGISTER HIGH

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

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
P3<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 15-8 **Unimplemented:** Read as '0'

bit 7-0 **P3<23:16>:** Parameter 3 High bits

Smart Card Mode:
Waiting Time Counter bits (bits<23:16>).

Other Modes:
Not used.

REGISTER 13-15: UxSCCON: UARTx SMART CARD CONFIGURATION REGISTER

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	U-0
—	—	TXRPT1	TXRPT0	CONV	T0PD	PRTCL	—
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 5-4 **TXRPT<1:0>:** Transmit Repeat Selection bits
11 = Retransmit the error byte four times
10 = Retransmit the error byte three times
01 = Retransmit the error byte twice
00 = Retransmit the error byte once

bit 3 **CONV:** Logic Convention Selection bit
1 = Inverse logic convention
0 = Direct logic convention

bit 2 **T0PD:** Pull-Down Duration for T = 0 Error Handling bit
1 = 2 ETU
0 = 1 ETU

bit 1 **PRTCL:** Smart Card Protocol Selection bit
1 = T = 1
0 = T = 0

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

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 22-2: INSTRUCTION SET OVERVIEW (CONTINUED)

Base Instr #	Assembly Mnemonic	Assembly Syntax	Description	# of Words	# of Cycles(1)	Status Flags Affected
60	MIN	MIN Acc	If Accumulator A Less than B Load Accumulator with B or vice versa	1	1	N,OV,Z
		MIN.V Acc, Wd	If Accumulator A Less than B Accumulator Force Minimum Data Range Limit with Limit Excess Result	1	1	N,OV,Z
		MINZ Acc	Accumulator Force Minimum Data Range Limit	1	1	N,OV,Z
		MINZ.V Acc, Wd	Accumulator Force Minimum Data Range Limit with Limit Excess Result	1	1	N,OV,Z
61	MOV	MOV f,Wn	Move f to Wn	1	1	None
		MOV f	Move f to f	1	1	None
		MOV f,WREG	Move f to WREG	1	1	None
		MOV #lit16,Wn	Move 16-bit Literal to Wn	1	1	None
		MOV.b #lit8,Wn	Move 8-bit Literal to Wn	1	1	None
		MOV Wn,f	Move Wn to f	1	1	None
		MOV Wso,Wdo	Move Ws to Wd	1	1	None
		MOV WREG,f	Move WREG to f	1	1	None
		MOV.D Wns,Wd	Move Double from W(ns):W(ns + 1) to Wd	1	2	None
		MOV.D Ws,Wnd	Move Double from Ws to W(nd + 1):W(nd)	1	2	None
62	MOVPAG	MOVPAG #lit10,DSRPAG	Move 10-bit Literal to DSRPAG	1	1	None
		MOVPAG #lit8,TBLPAG	Move 8-bit Literal to TBLPAG	1	1	None
		MOVPAG Ws, DSRPAG	Move Ws<9:0> to DSRPAG	1	1	None
		MOVPAG Ws, TBLPAG	Move Ws<7:0> to TBLPAG	1	1	None
64	MOVSAC	MOVSAC Acc,Wx,Wxd,Wy,Wyd,AWB	Prefetch and Store Accumulator	1	1	None
65	MPY	MPY Wm*Wn,Acc,Wx,Wxd,Wy,Wyd	Multiply Wm by Wn to Accumulator	1	1	OA,OB,OAB, SA,SB,SAB
		MPY Wm*Wm,Acc,Wx,Wxd,Wy,Wyd	Square Wm to Accumulator	1	1	OA,OB,OAB, SA,SB,SAB
66	MPY.N	MPY.N Wm*Wn,Acc,Wx,Wxd,Wy,Wyd	-(Multiply Wm by Wn) to Accumulator	1	1	None
67	MSC	MSC Wm*Wm,Acc,Wx,Wxd,Wy,Wyd, AWB	Multiply and Subtract from Accumulator	1	1	OA,OB,OAB, SA,SB,SAB
68	MUL	MUL.SS Wb,Ws,Wnd	{Wnd + 1, Wnd} = Signed(Wb) * Signed(Ws)	1	1	None
		MUL.SS Wb,Ws,Acc	Accumulator = Signed(Wb) * Signed(Ws)	1	1	None
		MUL.SU Wb,Ws,Wnd	{Wnd + 1, Wnd} = Signed(Wb) * Unsigned(Ws)	1	1	None
		MUL.SU Wb,Ws,Acc	Accumulator = Signed(Wb) * Unsigned(Ws)	1	1	None
		MUL.SU Wb,#lit5,Acc	Accumulator = Signed(Wb) * Unsigned(lit5)	1	1	None
		MUL.US Wb,Ws,Wnd	{Wnd + 1, Wnd} = Unsigned(Wb) * Signed(Ws)	1	1	None
		MUL.US Wb,Ws,Acc	Accumulator = Unsigned(Wb) * Signed(Ws)	1	1	None
		MUL.UU Wb,Ws,Wnd	{Wnd + 1, Wnd} = Unsigned(Wb) * Unsigned(Ws)	1	1	None
		MUL.UU Wb,#lit5,Acc	Accumulator = Unsigned(Wb) * Unsigned(lit5)	1	1	None
		MUL.UU Wb,Ws,Acc	Accumulator = Unsigned(Wb) * Unsigned(Ws)	1	1	None
		MULW.SS Wb,Ws,Wnd	Wnd = Signed(Wb) * Signed(Ws)	1	1	None
		MULW.SU Wb,Ws,Wnd	Wnd = Signed(Wb) * Unsigned(Ws)	1	1	None
		MULW.US Wb,Ws,Wnd	Wnd = Unsigned(Wb) * Signed(Ws)	1	1	None
		MULW.UU Wb,Ws,Wnd	Wnd = Unsigned(Wb) * Unsigned(Ws)	1	1	None
		MUL.SU Wb,#lit5,Wnd	{Wnd + 1, Wnd} = Signed(Wb) * Unsigned(lit5)	1	1	None
		MUL.SU Wb,#lit5,Wnd	Wnd = Signed(Wb) * Unsigned(lit5)	1	1	None
		MUL.UU Wb,#lit5,Wnd	{Wnd + 1, Wnd} = Unsigned(Wb) * Unsigned(lit5)	1	1	None
		MUL.UU Wb,#lit5,Wnd	Wnd = Unsigned(Wb) * Unsigned(lit5)	1	1	None
		MUL f	W3:W2 = f * WREG	1	1	None

Note 1: Read and Read-Modify-Write (e.g., bit operations and logical operations) on non-CPU SFRs incur an additional instruction cycle.

2: Cycle times for Slave core are different for Master core, as shown in 2.

3: For dsPIC33CH128MP508 devices, the divide instructions must be preceded with a "REPEAT #5" instruction, such that they are executed six consecutive times

24.0 ELECTRICAL CHARACTERISTICS

This section provides an overview of the dsPIC33CH128MP508 family electrical characteristics. Additional information will be provided in future revisions of this document as it becomes available.

Absolute maximum ratings for the dsPIC33CH128MP508 family are listed below. Exposure to these maximum rating conditions for extended periods may affect device reliability. Functional operation of the device at these, or any other conditions above the parameters indicated in the operation listings of this specification, is not implied.

Absolute Maximum Ratings(1)

Ambient temperature under bias ..... -40°C to +125°C
Storage temperature ..... -65°C to +150°C
Voltage on V_{DD} with respect to V_{SS} ..... -0.3V to +4.0V
Voltage on any pin that is not 5V tolerant with respect to V_{SS}(3) ..... -0.3V to (V_{DD} + 0.3V)
Voltage on any 5V tolerant pin with respect to V_{SS} when $V_{DD} \geq 3.0V$(3) ..... -0.3V to +5.5V
Voltage on any 5V tolerant pin with respect to Vss when $V_{DD} < 3.0V$(3) ..... -0.3V to +3.6V
Maximum current out of V_{SS} pin ..... 300 mA
Maximum current into V_{DD} pin(2) ..... 300 mA
Maximum current sunk/sourced by any 4x I/O pin ..... 15 mA
Maximum current sunk/sourced by any 8x I/O pin ..... 25 mA
Maximum current sunk by all ports(2) ..... 200 mA

Note 1: Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those, or any other conditions above those indicated in the operation listings of this specification, is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

2: Maximum allowable current is a function of device maximum power dissipation (see Table 24-2).

3: See the **“Pin Diagrams”** section for the 5V tolerant pins.