Microchip Technology - DSPIC33CH64MP505-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	39
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 31x12b; D/A 4x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-TQFP
Supplier Device Package	48-TQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/dspic33ch64mp505-i-pt

3.3 Master Flash 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 **"Flash Programming"** (DS70609) in the *"dsPIC33/PIC24 Family Reference Manual"*, which is available from the Microchip web site (www.microchip.com).

2: This section refers to the **"Dual Partition Flash Program Memory"** (DS70005156) in the *"dsPIC33/PIC24 Family Reference Manual"*, but the Dual Partition is not implemented in the Master Flash.

The dsPIC33CH128MP508 family devices contain internal Flash program memory for storing and executing application code. The memory is readable, writable and erasable during normal operation over the entire VDD range.

Flash memory can be programmed in three ways:

- In-Circuit Serial Programming™ (ICSP™) programming capability
- Enhanced In-Circuit Serial Programming (Enhanced ICSP)
- Run-Time Self-Programming (RTSP)

ICSP allows for a dsPIC33CH128MP508 family device to be serially programmed while in the end application circuit. This is done with a Programming Clock and Programming Data (PGCx/PGDx) line, and three other lines for power (VDD), ground (VSS) and Master Clear (MCLR). This allows customers to manufacture boards with unprogrammed devices and then program the device just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

Enhanced In-Circuit Serial Programming uses an on-board bootloader, known as the Program Executive, to manage the programming process. Using an SPI data frame format, the Program Executive can erase, program and verify program memory. For more information on Enhanced ICSP, see the device programming specification.

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

3.3.1 FLASH PROGRAMMING OPERATIONS

For ICSP and RTSP programming of the Master Flash, `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 Flash, starting at the address defined by the contents of TBLPAG, and the NVMADR and NVMADRU registers.

Programmers can program two adjacent words (24 bits x 2) of Program Flash Memory at a time on every other word address boundary (0x000002, 0x000006, 0x00000A, etc.). To do this, it is necessary to erase the page that contains the desired address of the location the user wants to change. For protection against accidental operations, the write initiate sequence for NVMKEY must be used to allow any erase or program operation to proceed. After the programming command has been executed, the user application must wait for the programming time until programming is complete.

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

- A full 48-bit double instruction word should always be programmed to a Flash 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 should never be executed. The penultimate instruction must contain a program flow change instruction, such as a `RETURN` or `BRA` instruction.

REGISTER 3-51: 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 (QEIHOM1) to the Corresponding RPn Pin bits
See Table 3-30.

bit 7-0 **QEINDX1R<7:0>:** Assign QEI Index 1 Input (QEINDX1) to the Corresponding RPn Pin bits
See Table 3-30.

REGISTER 3-52: 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 ($\overline{\text{U1DSR}}$) to the Corresponding RPn Pin bits
See Table 3-30.

bit 7-0 **U1RXR<7:0>:** Assign UART1 Receive (U1RX) to the Corresponding RPn Pin bits
See Table 3-30.

REGISTER 3-103: C1CONL: CAN CONTROL REGISTER LOW

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1	R/W-1
CON	—	SIDL	BRSDIS	BUSY	WFT1	WFT0	WAKFIL(1)
bit 15							bit 8

R/W-0	R/W-1	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CLKSEL(1)	PXEDIS(1)	ISOCRCEN(1)	DNCNT4	DNCNT3	DNCNT2	DNCNT1	DNCNT0
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 **CON:** CAN Enable bit
1 = CAN module is enabled
0 = CAN module is disabled

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

bit 13 **SIDL:** CAN Stop in Idle Control bit
1 = Stops module operation in Idle mode
0 = Does not stop module operation in Idle mode

bit 12 **BRSDIS:** Bit Rate Switching (BRS) Disable bit
1 = Bit Rate Switching is disabled, regardless of BRS in the transmit message object
0 = Bit Rate Switching depends on BRS in the transmit message object

bit 11 **BUSY:** CAN Module is Busy bit
1 = The CAN module is active
0 = The CAN module is inactive

bit 10-9 **WFT<1:0>:** Selectable Wake-up Filter Time bits
11 = T11FILTER
10 = T10FILTER
01 = T01FILTER
00 = T00FILTER

bit 8 **WAKFIL:** Enable CAN Bus Line Wake-up Filter bit(1)
1 = Uses CAN bus line filter for wake-up
0 = CAN bus line filter is not used for wake-up

bit 7 **CLKSEL:** Module Clock Source Select bit(1)
1 = AFPLLO is selected as the source
0 = FCAN is selected as the source

bit 6 **PXEDIS:** Protocol Exception Event Detection Disabled bit(1)
A recessive "reserved bit" following a recessive FDF bit is called a Protocol Exception.
1 = Protocol Exception is treated as a form error
0 = If a Protocol Exception is detected, CAN will enter the bus integrating state

bit 5 **ISOCRCEN:** Enable ISO CRC in CAN FD Frames bit(1)
1 = Includes stuff bit count in CRC field and uses non-zero CRC initialization vector
0 = Does not include stuff bit count in CRC field and uses CRC initialization vector with all zeros

bit 4-0 **DNCNT<4:0>:** DeviceNet™ Filter Bit Number bits
10011-11111 = Invalid selection (compares up to 18 bits of data with EID)
10010 = Compares up to Data Byte 2, bit 6 with EID17
...
00001 = Compares up to Data Byte 0, bit 7 with EID0
00000 = Does not compare data bytes

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

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

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

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
FIFOUA<23:16>							
bit 7							bit 0

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

bit 15-0 **FIFOUA<31:16>:** FIFO User Address bits

TXEN = 1 (FIFO configured as a transmit buffer):
A read of this register will return the address where the next message is to be written (FIFO head).

TXEN = 0 (FIFO configured as a receive buffer):
A read of this register will return the address where the next message is to be read (FIFO tail).

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

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

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

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
FIFOUA<7:0>							
bit 7							bit 0

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

bit 15-0 **FIFOUA<15:0>:** FIFO User Address bits

TXEN = 1 (FIFO configured as a transmit buffer):
A read of this register will return the address where the next message is to be written (FIFO head).

TXEN = 0 (FIFO configured as a receive buffer):
A read of this register will return the address where the next message is to be read (FIFO tail).

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

REGISTER 3-150: C1BDIAG1L: CAN BUS DIAGNOSTICS REGISTER 1 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
EFMSGCNT<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
EFMSGCNT<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 **EFMSGCNT<15:0>:** Error-Free Message Counter bits

REGISTER 3-161: ADCON3L: ADC CONTROL REGISTER 3 LOW

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	HSC/R-0	R/W-0	HSC/R-0
REFSEL2	REFSEL1	REFSEL0	SUSPEND	SUSPCIE	SUSPRDY	SHRSAMP	CNVRTCH
bit 15							bit 8

R/W-0	HSC/R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SWLCTRG	SWCTRG	CNVCHSEL5	CNVCHSEL4	CNVCHSEL3	CNVCHSEL2	CNVCHSEL1	CNVCHSEL0
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 15-13 **REFSEL<2:0>:** ADC Reference Voltage Selection bits

Value	VREFH	VREFL
000	AVDD	AVSS

001-111 = **Unimplemented:** Do not use

bit 12 **SUSPEND:** All ADC Core Triggers Disable bit

1 = All new trigger events for all ADC cores are disabled
0 = All ADC cores can be triggered

bit 11 **SUSPCIE:** Suspend All ADC Cores Common Interrupt Enable bit

1 = Common interrupt will be generated when ADC core triggers are suspended (SUSPEND bit = 1) and all previous conversions are finished (SUSPRDY bit becomes set)
0 = Common interrupt is not generated for suspend ADC cores event

bit 10 **SUSPRDY:** All ADC Cores Suspended Flag bit

1 = ADC core is suspended (SUSPEND bit = 1) and has no conversions in progress
0 = ADC cores have previous conversions in progress

bit 9 **SHRSAMP:** Shared ADC Core Sampling Direct Control bit

This bit should be used with the individual channel conversion trigger controlled by the CNVRTCH bit. It connects an analog input, specified by the CNVCHSEL<5:0> bits, to the shared ADC core and allows extending the sampling time. This bit is not controlled by hardware and must be cleared before the conversion starts (setting CNVRTCH to '1').

1 = Shared ADC core samples an analog input specified by the CNVCHSEL<5:0> bits
0 = Sampling is controlled by the shared ADC core hardware

bit 8 **CNVRTCH:** Software Individual Channel Conversion Trigger bit

1 = Single trigger is generated for an analog input specified by the CNVCHSEL<5:0> bits; when the bit is set, it is automatically cleared by hardware on the next instruction cycle
0 = Next individual channel conversion trigger can be generated

bit 7 **SWLCTRG:** Software Level-Sensitive Common Trigger bit

1 = Triggers are continuously generated for all channels with the software; level-sensitive common trigger selected as a source in the ADTRIGnL and ADTRIGnH registers
0 = No software, level-sensitive common triggers are generated

bit 6 **SWCTRG:** Software Common Trigger bit

1 = Single trigger is generated for all channels with the software; common trigger selected as a source in the ADTRIGnL and ADTRIGnH registers; when the bit is set, it is automatically cleared by hardware on the next instruction cycle
0 = Ready to generate the next software common trigger

bit 5-0 **CNVCHSEL <5:0>:** Channel Number Selection for Software Individual Channel Conversion Trigger bits

These bits define a channel to be converted when the CNVRTCH bit is set.

REGISTER 3-164: ADCON5H: ADC CONTROL REGISTER 5 HIGH

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	WARMTIME3	WARMTIME2	WARMTIME1	WARMTIME0
bit 15							bit 8

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
SHRCIE	—	—	—	—	—	—	—
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 11-8 **WARMTIME<3:0>:** ADC Dedicated Core Power-up Delay bits

These bits determine the power-up delay in the number of the Core Source Clock Periods (TCORESRC) for all ADC cores.

1111 = 32768 Source Clock Periods
1110 = 16384 Source Clock Periods
1101 = 8192 Source Clock Periods
1100 = 4096 Source Clock Periods
1011 = 2048 Source Clock Periods
1010 = 1024 Source Clock Periods
1001 = 512 Source Clock Periods
1000 = 256 Source Clock Periods
0111 = 128 Source Clock Periods
0110 = 64 Source Clock Periods
0101 = 32 Source Clock Periods
0100 = 16 Source Clock Periods
00xx = 16 Source Clock Periods

bit 7 **SHRCIE:** Shared ADC Core Ready Common Interrupt Enable bit

1 = Common interrupt will be generated when ADC core is powered and ready for operation
0 = Common interrupt is disabled for an ADC core ready event

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

REGISTER 4-89: ADCON4L: ADC CONTROL REGISTER 4 LOW

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

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	SAMC1EN	SAMC0EN
bit 7							bit 0

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

bit 15-10 **Unimplemented:** Read as '0'

bit 9-8 **Reserved:** Must be written as '0'

bit 7-2 **Unimplemented:** Read as '0'

bit 1 **SAMC1EN:** Dedicated ADC Core 1 Conversion Delay Enable bit

1 = After trigger, the conversion will be delayed and the ADC core will continue sampling during the time specified by the SAMC<9:0> bits in the ADCORE1L register

0 = After trigger, the sampling will be stopped immediately and the conversion will be started on the next core clock cycle

bit 0 **SAMC0EN:** Dedicated ADC Core 0 Conversion Delay Enable bit

1 = After trigger, the conversion will be delayed and the ADC core will continue sampling during the time specified by the SAMC<9:0> bits in the ADCORE0L register

0 = After trigger, the sampling will be stopped immediately and the conversion will be started on the next core clock cycle

TABLE 5-1: APPLICATION MODE SLVEN RESET CONTROL TRUTH TABLE

S1MSRE	S1SSRE	SLVEN Bit Reset Source	Application Effect
0	0	Master Resets(1)	• Slave is reset and disabled in the event of a POR, BOR or $\overline{MCLR}$ Reset. Master must re-enable Slave. • Slave Run-Time Resets will not disable Slave. Slave will reset and continue execution (and may optionally interrupt Master).
1	0	Master Resets(1)	• Slave is reset and disabled in the event of a POR, BOR or $\overline{MCLR}$ Reset. Master must re-enable Slave. • Slave Run-Time Resets will not disable Slave. Slave will reset and continue execution (and may optionally interrupt Master).
0	1	Master Resets(1) and Slave Resets(2)	• Slave is reset and disabled in the event of any Slave Run-Time Reset (and may optionally interrupt Master). Master must re-enable Slave to execute the Slave code. • Master Run-Time Resets will not affect Slave operation.
1	1	POR/BOR/$\overline{MCLR}$(1) Slave Resets(2)	• Slave is reset and disabled in the event of any Slave Run-Time Reset or Master Reset. Master must re-enable Slave. This represents the default state (S1MSRE and S1SSRE are unprogrammed).

Note 1: Master Resets include any Master Reset, such as POR/BOR/$\overline{MCLR}$ Resets.

2: Slave Resets include any Slave Reset, plus POR/BOR/$\overline{MCLR}$ Resets (in Application mode).

5.4.1 INTER-PROCESSOR INTERRUPT REQUEST AND ACKNOWLEDGE

The Master and Slave processors may interrupt each other directly. The Master may issue an interrupt request to the Slave by asserting the MTSIRQ (MSI1CON<9>) control bit. Similarly, the Slave may issue an interrupt request to the Master by asserting the STMIRQ (MSI1STAT<9>) control bit.

The interrupts are Acknowledged through the use of the Interrupt Acknowledge bits, MTSIACK (MSI1STAT<8>) for the Master to Slave interrupt request and STMIACK (MSI1CON<8>) for the Slave to Master interrupt request.

5.4.2 READ ADDRESS POINTERS FOR FIFOs

The MSI macro may also include a set of two FIFOs, one for data reads from the Slave and the other for data writes to the Slave. The Read Address Pointers for the Read and Write FIFOs are held in the RDPTR<6:0> bits (MSI1CON<6:0>) and WRPTR<6:0 bits (MSI1STAT<6:0>), respectively. These bits are accessible only from within Debug mode.

REGISTER 6-14: PLLFBD: PLL FEEDBACK DIVIDER REGISTER (SLAVE)

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

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

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

bit 15-12 **Unimplemented:** Read as '0'

bit 11-8 **Reserved:** Maintain as '0'

bit 7-0 **PLLFBDIV<7:0>:** PLL Feedback Divider bits (also denoted as 'M', PLL multiplier)

11111111 = Reserved
...
11001000 = 200 maximum(1)
...
10010110 = 150 (default)
...
00010000 = 16 minimum(1)
...
00000010 = Reserved
00000001 = Reserved
00000000 = Reserved

Note 1: The allowed range is 16-200 (decimal). The rest of the values are reserved and should be avoided. The power on the default feedback divider is 150 (decimal) with an 8 MHz FRC input clock. The VCO frequency is 1.2 GHz.

REGISTER 7-4: PMD3: MASTER PERIPHERAL MODULE DISABLE 3 CONTROL REGISTER LOW(1)

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

R/W-0	U-0	U-0	U-0	U-0	U-0	R/W-0	U-0
CRCMD	—	—	—	—	—	I2C2MD	—
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 **CRCMD:** CRC Module Disable bit
1 = CRC module is disabled
0 = CRC module is enabled

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

bit 1 **I2C2MD:** I2C2 Module Disable bit
1 = I2C2 module is disabled
0 = I2C2 module is enabled

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

Note 1: This register is only available in the Master core.

8.1 Summary of DMA Operations

The DMA Controller is capable of moving data between addresses according to a number of different parameters. Each of these parameters can be independently configured for any transaction. In addition, any or all of the DMA channels can independently perform a different transaction at the same time. Transactions are classified by these parameters:

- Source and destination (SFRs and data RAM)
- Data size (byte or word)
- Trigger source
- Transfer mode (One-Shot, Repeated or Continuous)
- Addressing modes (Fixed Address or Address Blocks with or without Address Increment/Decrement)

In addition, the DMA Controller provides channel priority arbitration for all channels.

8.1.1 SOURCE AND DESTINATION

Using the DMA Controller, data may be moved between any two addresses in the Data Space. The SFR space (0000h to 0FFFh) or the data RAM space (Master is 1000h to 4FFFh and Slave is 1000 to 1FFFh) can serve as either the source or the destination. Data can be moved between these areas in either direction or between addresses in either area. The four different combinations are shown in Figure 8-2.

If it is necessary to protect areas of data RAM, the DMA Controller allows the user to set upper and lower address boundaries for operations in the Data Space above the SFR space. The boundaries are set by the DMAH and DMAL Limit registers. If a DMA channel attempts an operation outside of the address boundaries, the transaction is terminated and an interrupt is generated.

8.1.2 DATA SIZE

The DMA Controller can handle both 8-bit and 16-bit transactions. Size is user-selectable using the SIZE bit (DMACHn<1>). By default, each channel is configured for word-size transactions. When byte-size transactions are chosen, the LSB of the source and/or destination address determines if the data represents the upper or lower byte of the data RAM location.

8.1.3 TRIGGER SOURCE

The DMA Controller can use 82 of the device's interrupt sources to initiate a transaction. The DMA trigger sources occur in reverse order from their natural interrupt priority and are shown in Table 8-2.

Since the source and destination addresses for any transaction can be programmed independently of the trigger source, the DMA Controller can use any trigger to perform an operation on any peripheral. This also allows DMA channels to be cascaded to perform more complex transfer operations.

8.1.4 TRANSFER MODE

The DMA Controller supports four types of data transfers, based on the volume of data to be moved for each trigger.

- One-Shot: A single transaction occurs for each trigger.
- Continuous: A series of back-to-back transactions occur for each trigger; the number of transactions is determined by the DMACNTn transaction counter.
- Repeated One-Shot: A single transaction is performed repeatedly, once per trigger, until the DMA channel is disabled.
- Repeated Continuous: A series of transactions are performed repeatedly, one cycle per trigger, until the DMA channel is disabled.

All transfer modes allow the option to have the source and destination addresses, and counter value, automatically reloaded after the completion of a transaction.

8.1.5 ADDRESSING MODES

The DMA Controller also supports transfers between single addresses or address ranges. The four basic options are:

- Fixed-to-Fixed: Between two constant addresses
- Fixed-to-Block: From a constant source address to a range of destination addresses
- Block-to-Fixed: From a range of source addresses to a single, constant destination address
- Block-to-Block: From a range of source addresses to a range of destination addresses

The option to select auto-increment or auto-decrement of source and/or destination addresses is available for Block Addressing modes.

In addition to the four basic modes, the DMA Controller also supports Peripheral Indirect Addressing (PIA) mode, where the source or destination address is generated jointly by the DMA Controller and a PIA-capable peripheral. When enabled, the DMA channel provides a base source and/or destination address, while the peripheral provides a fixed range offset address.

REGISTER 11-9: SLPxCONL: DACx SLOPE CONTROL LOW REGISTER (CONTINUED)

bit 11-8 **SLPSTOPA<3:0>:** Slope Stop A Signal Select bits

The selected Slope Stop A signal is logically OR'd with the selected Slope Stop B signal to terminate the slope function.

Slope Stop A Signal Selection	Master	Slave
1111	1	1
1110	Slave PWM2 Trigger 2	Master PWM8 Trigger 2
1101	Slave PWM1 Trigger 2	Master PWM7 Trigger 2
1000	Master PWM4 Trigger 2	Slave PWM8 Trigger 2
0111	Master PWM3 Trigger 2	Slave PWM7 Trigger 2
0110	Master PWM2 Trigger 2	Slave PWM6 Trigger 2
0101	Master PWM1 Trigger 2	Slave PWM5 Trigger 2
0100	Master PWM4 Trigger 1	Slave PWM4 Trigger 2
0011	Master PWM3 Trigger 1	Slave PWM3 Trigger 2
0010	Master PWM2 Trigger 1	Slave PWM2 Trigger 2
0001	Master PWM1 Trigger 1	Slave PWM1 Trigger 2
0000	0	0

bit 7-4 **SLPSTOPB<3:0>:** Slope Stop B Signal Select bits

The selected Slope Stop B signal is logically OR'd with the selected Slope Stop A signal to terminate the slope function.

Slope Start B Signal Selection	Master	Slave
1111	1	1
0100	S1CMP3 Out	CMP1 Out
0011	S1CMP2 Out	S1CMP3 Out
0010	S1CMP1 Out	S1CMP2 Out
0001	CMP1 Out	S1CMP1 Out
0000	0	0

bit 3-0 **SLPSTRT<3:0>:** Slope Start Signal Select bits

Slope Start Signal Selection	Master	Slave
1111	1	1
1110	Slave PWM2 Trigger 1	Master PWM2 Trigger 1
1101	Slave PWM1 Trigger 1	Master PWM1 Trigger 1
1000	Master PWM4 Trigger 2	Slave PWM8 Trigger 1
0111	Master PWM3 Trigger 2	Slave PWM7 Trigger 1
0110	Master PWM2 Trigger 2	Slave PWM6 Trigger 1
0101	Master PWM1 Trigger 2	Slave PWM5 Trigger 1
0100	Master PWM4 Trigger 1	Slave PWM4 Trigger 1
0011	Master PWM3 Trigger 1	Slave PWM3 Trigger 1
0010	Master PWM2 Trigger 1	Slave PWM2 Trigger 1
0001	Master PWM1 Trigger 1	Slave PWM1 Trigger 1
0000	0	0

REGISTER 12-21: QEIxGECL: QEIx GREATER THAN OR EQUAL COMPARE 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
QEIGEC<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
QEIGEC<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 **QEIGEC<15:0>:** QEIx Greater Than or Equal Compare bits

REGISTER 12-22: QEIxGECH: QEIx GREATER THAN OR EQUAL COMPARE 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
QEIGEC<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
QEIGEC<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 **QEIGEC<31:16>:** QEIx Greater Than or Equal Compare bits

REGISTER 13-13: UxTXCHK: UARTx TRANSMIT CHECKSUM REGISTER

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
TXCHK<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 **TXCHK<7:0>:** Transmit Checksum bits (calculated from TX words)

LIN Modes:
C0EN = 1: Sum of all transmitted data + addition carries, including PID.
C0EN = 0: Sum of all transmitted data + addition carries, excluding PID.

LIN Slave:
Cleared when Break is detected.

LIN Master/Slave:
Cleared when Break is detected.

Other Modes:
C0EN = 1: Sum of every byte transmitted + addition carries.
C0EN = 0: Value remains unchanged.

TABLE 22-2: INSTRUCTION SET OVERVIEW (CONTINUED)

Base Instr #	Assembly Mnemonic	Assembly Syntax		Description	# of Words	# of Cycles(1)	Status Flags Affected
89	SL	SL	f	f = Left Shift f	1	1	C,N,OV,Z
		SL	f,WREG	WREG = Left Shift f	1	1	C,N,OV,Z
		SL	Ws,Wd	Wd = Left Shift Ws	1	1	C,N,OV,Z
		SL	Wb,Wns,Wnd	Wnd = Left Shift Wb by Wns	1	1	N,Z
		SL	Wb,#lit5,Wnd	Wnd = Left Shift Wb by lit5	1	1	N,Z
91	SUB	SUB	Acc	Subtract Accumulators	1	1	OA,OB,OAB, SA,SB,SAB
		SUB	f	f = f – WREG	1	1	C,DC,N,OV,Z
		SUB	f,WREG	WREG = f – WREG	1	1	C,DC,N,OV,Z
		SUB	#lit10,Wn	Wn = Wn – lit10	1	1	C,DC,N,OV,Z
		SUB	Wb,Ws,Wd	Wd = Wb – Ws	1	1	C,DC,N,OV,Z
		SUB	Wb,#lit5,Wd	Wd = Wb – lit5	1	1	C,DC,N,OV,Z
92	SUBB	SUBB	f	f = f – WREG – ($\overline{C}$)	1	1	C,DC,N,OV,Z
		SUBB	f,WREG	WREG = f – WREG – ($\overline{C}$)	1	1	C,DC,N,OV,Z
		SUBB	#lit10,Wn	Wn = Wn – lit10 – ($\overline{C}$)	1	1	C,DC,N,OV,Z
		SUBB	Wb,Ws,Wd	Wd = Wb – Ws – ($\overline{C}$)	1	1	C,DC,N,OV,Z
		SUBB	Wb,#lit5,Wd	Wd = Wb – lit5 – ($\overline{C}$)	1	1	C,DC,N,OV,Z
93	SUBR	SUBR	f	f = WREG – f	1	1	C,DC,N,OV,Z
		SUBR	f,WREG	WREG = WREG – f	1	1	C,DC,N,OV,Z
		SUBR	Wb,Ws,Wd	Wd = Ws – Wb	1	1	C,DC,N,OV,Z
		SUBR	Wb,#lit5,Wd	Wd = lit5 – Wb	1	1	C,DC,N,OV,Z
94	SUBBR	SUBBR	f	f = WREG – f – ($\overline{C}$)	1	1	C,DC,N,OV,Z
		SUBBR	f,WREG	WREG = WREG – f – ($\overline{C}$)	1	1	C,DC,N,OV,Z
		SUBBR	Wb,Ws,Wd	Wd = Ws – Wb – ($\overline{C}$)	1	1	C,DC,N,OV,Z
		SUBBR	Wb,#lit5,Wd	Wd = lit5 – Wb – ($\overline{C}$)	1	1	C,DC,N,OV,Z
95	SWAP	SWAP.b	Wn	Wn = Nibble Swap Wn	1	1	None
		SWAP	Wn	Wn = Byte Swap Wn	1	1	None
96	TBLRDH	TBLRDH	Ws,Wd	Read Prog<23:16> to Wd<7:0>	1	5/3(2)	None
97	TBLRDL	TBLRDL	Ws,Wd	Read Prog<15:0> to Wd	1	5/3(2)	None
98	TBLWTH	TBLWTH	Ws,Wd	Write Ws<7:0> to Prog<23:16>	1	2	None
99	TBLWTL	TBLWTL	Ws,Wd	Write Ws to Prog<15:0>	1	2	None
101	ULNK	ULNK		Unlink Frame Pointer	1	1	SFA
103	VFSLV	VFSLV	Wns,Wnd,lit2	Compare (Master) Ws to (Slave) Wd	1	1	None
104	XOR	XOR	f	f = f .XOR. WREG	1	1	N,Z
		XOR	f,WREG	WREG = f .XOR. WREG	1	1	N,Z
		XOR	#lit10,Wn	Wd = lit10 .XOR. Wd	1	1	N,Z
		XOR	Wb,Ws,Wd	Wd = Wb .XOR. Ws	1	1	N,Z
		XOR	Wb,#lit5,Wd	Wd = Wb .XOR. lit5	1	1	N,Z
105	ZE	ZE	Ws,Wnd	Wnd = Zero-Extend Ws	1	1	C,Z,N

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

TABLE 24-5: DC CHARACTERISTICS: OPERATING CURRENT (IDD) (MASTER RUN/SLAVE RUN)

DC CHARACTERISTICS	Master (Run) + Slave (Run)		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**		
Operating Current (IDD)(1)						
DC20	11.6	13.7	mA	-40°C	3.3V	10 MIPS (N = 1, N2 = 5, N3 = 2, M = 50, FVCO = 400 MHz, FPLLO = 40 MHz)
	11.7	17.5	mA	+25°C		
	11.9	23.5	mA	+85°C		
	15.8	30.0	mA	+125°C		
DC21	15.9	18.3	mA	-40°C	3.3V	20 MIPS (N = 1, N2 = 5, N3 = 1, M = 60, FVCO = 480 MHz, FPLLO = 280 MHz)
	16.0	22.2	mA	+25°C		
	16.1	28.0	mA	+85°C		
	20.0	34.3	mA	+125°C		
DC22	23.7	26.9	mA	-40°C	3.3V	40 MIPS (N = 1, N2 = 3, N3 = 1, M = 60, FVCO = 480 MHz, FPLLO = 160 MHz)
	23.9	30.9	mA	+25°C		
	25.9	36.6	mA	+85°C		
	27.8	42.1	mA	+125°C		
DC23	37.3	42.0	mA	-40°C	3.3V	70 MIPS (N = 1, N2 = 2, N3 = 1, M = 70, FVCO = 560 MHz, FPLLO = 280 MHz)
	37.5	46.1	mA	+25°C		
	37.2	51.1	mA	+85°C		
	41.1	55.7	mA	+125°C		
DC24	45.0	50.4	mA	-40°C	3.3V	90 MIPS (N = 1, N2 = 2, N3 = 1, M = 90, FVCO = 720 MHz, FPLLO = 360 MHz)
	45.2	54.8	mA	+25°C		
	44.8	59.1	mA	+85°C		
	48.3	63.1	mA	+125°C		
DC25	45.5	51.0	mA	-40°C	3.3V	100 MIPS (N = 1, N2 = 1, N3 = 1, M = 50, FVCO = 400 MHz, FPLLO = 400 MHz); Slave runs at 100 MIPS but Master is still at 90 MIPS
	45.7	55.3	mA	+25°C		
	45.3	59.6	mA	+85°C		
	48.9	63.6	mA	+125°C		

Note 1: IDD is primarily a function of the operating voltage and frequency. Other factors, such as I/O pin loading and switching rate, oscillator type, internal code execution pattern and temperature, also have an impact on the current consumption. The test conditions for all IDD measurements are 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
- CPU, SRAM, program memory and data memory are operational
- No peripheral modules are operating or being clocked (all defined PMDx bits are set)
- CPU is executing `while(1)` statement
- JTAG is disabled

TABLE 24-10: DC CHARACTERISTICS: IDLE CURRENT (IIDLE) (MASTER SLEEP/SLAVE IDLE)

DC CHARACTERISTICS	Master (Sleep) + Slave (Idle)		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)						
DC40b	6.0	7.8	mA	-40°C	3.3V	10 MIPS (N = 1, N2 = 5, N3 = 2, M = 50, FVCO = 400 MHz, FPLLO = 40 MHz)
	6.2	11.4	mA	+25°C		
	6.4	17.5	mA	+85°C		
	10.4	24.4	mA	+125°C		
DC41b	6.6	8.4	mA	-40°C	3.3V	20 MIPS (N = 1, N2 = 5, N3 = 1, M = 50, FVCO = 400 MHz, FPLLO = 80 MHz)
	6.8	12.0	mA	+25°C		
	7.0	18.1	mA	+85°C		
	11.0	25.0	mA	+125°C		
DC42b	8.3	10.1	mA	-40°C	3.3V	40 MIPS (N = 1, N2 = 3, N3 = 1, M = 60, FVCO = 480 MHz, FPLLO = 160 MHz)
	8.5	13.8	mA	+25°C		
	8.7	19.9	mA	+85°C		
	12.6	26.7	mA	+125°C		
DC43b	10.6	12.6	mA	-40°C	3.3V	70 MIPS (N = 1, N2 = 2, N3 = 1, M = 70, FVCO = 560 MHz, FPLLO = 280 MHz)
	10.8	16.3	mA	+25°C		
	10.9	22.3	mA	+85°C		
	14.9	29.0	mA	+125°C		
DC44b	12.6	14.7	mA	-40°C	3.3V	90 MIPS (N = 1, N2 = 2, N3 = 1, M = 90, FVCO = 720 MHz, FPLLO = 360 MHz)
	12.7	18.4	mA	+25°C		
	12.9	23.6	mA	+85°C		
	16.8	30.9	mA	+125°C		
DC45b	11.7	13.8	mA	-40°C	3.3V	100 MIPS (N = 1, N2 = 1, N3 = 1, M = 50, FVCO = 400 MHz, FPLLO = 400 MHz)
	11.9	17.6	mA	+25°C		
	12.1	24.4	mA	+85°C		
	16.0	30.1	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 24-29: INTERNAL FRC ACCURACY

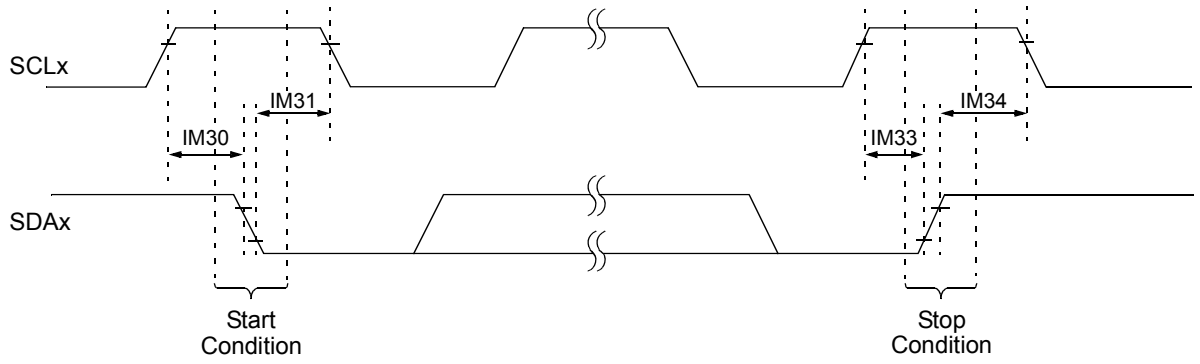
AC CHARACTERISTICS		Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	**Characteristic**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**
Internal FRC Accuracy @ FRC Frequency = 8 MHz(1)						
F20a	FRC	-3	—	+3	%	-40°C ≤ TA ≤ 0°C
		-1.5	—	+1.5	%	0°C ≤ TA ≤ +85°C
F20b	FRC	-2	—	+2	%	+85°C ≤ TA ≤ +125°C
F22	BFRC	-17	—	+17	%	-40°C ≤ TA ≤ +125°C

Note 1: Frequency is calibrated at +25°C and 3.3V. TUNx bits can be used to compensate for temperature drift.

TABLE 24-30: INTERNAL LPRC ACCURACY

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.	**Characteristic**	**Min.**	**Typ.**	**Max.**	**Units**	**Conditions**	
LPRC @ 32.768 kHz							
F21a	LPRC	-30	—	+30	%	-40°C ≤ TA ≤ -10°C	VDD = 3.0-3.6V
		-20	—	+20	%	-10°C ≤ TA ≤ +85°C	VDD = 3.0-3.6V
F21b	LPRC	-30	—	+30	%	+85°C ≤ TA ≤ +125°C	VDD = 3.0-3.6V

FIGURE 24-13: I2Cx BUS START/STOP BITS TIMING CHARACTERISTICS (MASTER MODE)



Note: Refer to Figure 24-1 for load conditions.

FIGURE 24-14: I2Cx BUS DATA TIMING CHARACTERISTICS (MASTER MODE)