Welcome to E-XFL.COM

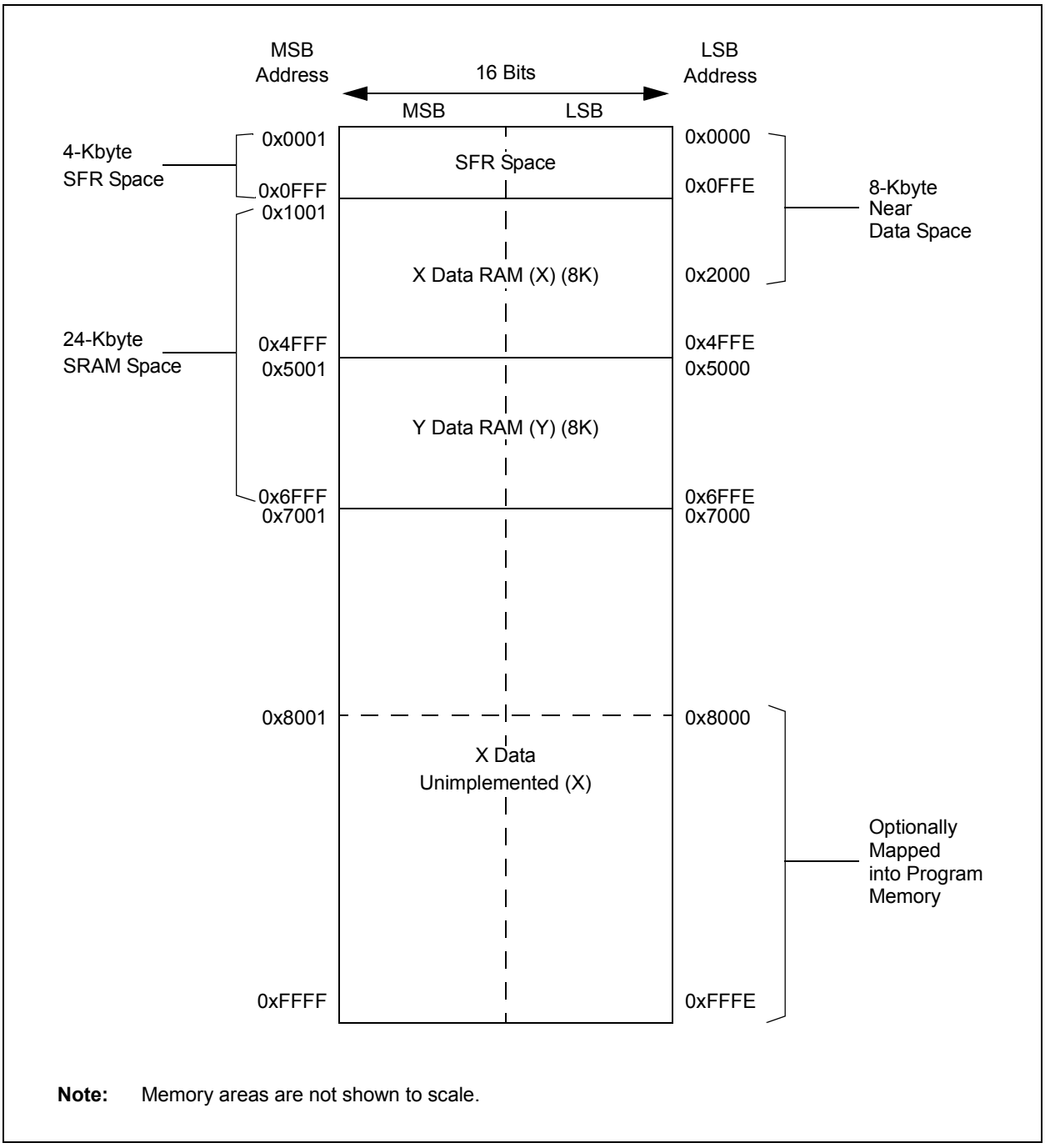
What is "Embedded - Microcontrollers"?

"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
Speed	100MHz
Connectivity	I²C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, Motor Control PWM, POR, PWM, QEI, WDT
Number of I/O	21
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 12x12b; D/A 3x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SSOP (0.209", 5.30mm Width)
Supplier Device Package	28-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/dspic33ck32mp202t-i-ss

FIGURE 4-7: DATA MEMORY MAP FOR dsPIC33CK256MPX0X DEVICES



Note: Memory areas are not shown to scale.

5.3.2 ERROR CORRECTING CODE (ECC)

In order to improve program memory performance and durability, these devices include Error Correcting Code (ECC) functionality as an integral part of the Flash memory controller. ECC can determine the presence of single bit errors in program data, including which bit is in error, and correct the data automatically without user intervention. ECC cannot be disabled.

When data is written to program memory, ECC generates a 7-bit Hamming code parity value for every two (24-bit) instruction words. The data is stored in blocks of 48 data bits and 7 parity bits; parity data is not memory-mapped and is inaccessible. When the data is read back, the ECC calculates the parity on it and compares it to the previously stored parity value. If a parity mismatch occurs, there are two possible outcomes:

- Single bit error has occurred and has been automatically corrected on readback.
- Double-bit error has occurred and the read data is not changed.

Single bit error occurrence can be identified by the state of the ECCSBEIF (IFS0<13>) bit. An interrupt can be generated when the corresponding interrupt enable bit is set, ECCSBEIE (IEC0<13>). The ECCSTATL register contains the parity information for single bit errors. The SECOUT<7:0> bits field contains the expected calculated SEC parity and the SECIN<7:0> bits contain the actual value from a Flash read operation. The SECSYNDx bits (ECCSTATH<7:0>) indicate the bit position of the single bit error within the 48-bit pair of instruction words. When no error is present, SECINx equals SECOUTx and SECSYNDx is zero.

Double-bit errors result in a generic hard trap. The ECCDBE bit (INTCON4<1>) will be set to identify the source of the hard trap. If no Interrupt Service Routine is implemented for the hard trap, a device Reset will also occur. The ECCSTATH register contains double-bit error status information. The DEDOUT bit is the expected calculated DED parity and DEDIN is the actual value from a Flash read operation. When no error is present, DEDIN equals DEDOUT.

5.3.3 ECC FAULT INJECTION

To test Fault handling, an EEC error can be generated. Both single and double-bit errors can be generated in both the read and write data paths. Read path Fault injection first reads the Flash data and then modifies it prior to entering the ECC logic. Write path Fault injection modifies the actual data prior to it being written into the target Flash and will cause an EEC error on a subsequent Flash read. The following procedure is used to inject a Fault:

1. Load the Flash target address into the ECCADDR register.
2. Select 1st Fault bit determined by FLT1PTRx (ECCCONH<7:0>). The target bit is inverted to create the Fault.
3. If a double Fault is desired, select the 2nd Fault bit determined by FLT2PTRx (ECCCONH<15:8>), otherwise set to all ‘1’s.
4. Write the NVMKEY unlock sequence (see **Section 5.5.3 “Program Flash Memory Control Registers”**).
5. Enable the ECC Fault injection logic by setting the FLTINJ bit (ECCCONL<0>).
6. Perform a read or write to the Flash target address.

TABLE 7-1: INTERRUPT VECTOR DETAILS (CONTINUED)

Interrupt Source	Vector #	IRQ #	IVT Address	Interrupt Bit Location		
				Flag	Enable	Priority
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>
ADCAN21 – ADC AN21 Interrupt	120	112	0x0000F4	IFS7<0>	IEC7<0>	IPC28<2:0>
ADCAN22 – ADC AN22 Interrupt	121	113	0x0000F6	IFS7<1>	IEC7<1>	IPC28<6:4>
ADCAN23 – ADC AN23 Interrupt	122	114	0x0000F8	IFS7<2>	IEC7<2>	IPC28<10:8>
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>
SPI1G – SPI1 Error	134	126	0x000110	IFS7<14>	IEC7<14>	IPC31<10:8>
SPI2G – SPI2 Error	135	127	0x000112	IFS7<15>	IEC7<15>	IPC31<14:12>
SPI3G – SPI3 Error	136	128	0x000114	IFS8<0>	IEC8<0>	IPC32<2:0>
Reserved	137-149	129-141	0x000116-0x00012E	—	—	—
SI2C3 – I2C3 Slave Event	150	142	0x000130	IFS8<14>	IEC8<14>	IPC35<10:8>
MI2C3 – I2C3 Master Event	151	143	0x000132	IFS8<15>	IEC8<15>	IPC35<14:12>
I2C3BC – I2C3 Bus Collision	152	144	0x000134	IFS9<0>	IEC9<0>	IPC36<2:0>
Reserved	153-156	145-148	0x000136-0x00013C	—	—	—
CCP7 – Input Capture/Output Compare 7	157	149	0x00013E	IFS9<5>	IEC9<5>	IPC37<6:4>
CCT7 – CCP7 Timer	158	150	0x000140	IFS9<6>	IEC9<6>	IPC37<10:8>
Reserved	159	151	0x000142	—	—	—

8.0 I/O PORTS

Note 1: This data sheet summarizes the features of the dsPIC33CK256MP508 family of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **"I/O Ports with Edge Detect"** (DS70005322) in the *"dsPIC33/PIC24 Family Reference Manual"*, which is available from the Microchip web site (www.microchip.com).

2: Some registers and associated bits described in this section may not be available on all devices.

Many of the device pins are shared among the peripherals and the Parallel I/O ports. All I/O input ports feature Schmitt Trigger inputs for improved noise immunity. The PORT registers are located in the SFR.

Some of the key features of the I/O ports are:

- Individual Output Pin Open-Drain Enable/Disable
- Individual Input Pin Weak Pull-up and Pull-Down
- Monitor Selective Inputs and Generate Interrupt when Change in Pin State is Detected
- Operation during Sleep and Idle modes

8.1 Parallel I/O (PIO) Ports

All port pins have 12 registers directly associated with their operation as digital I/Os. The Data Direction register (TRISx) determines whether the pin is an input or an output. If the data direction bit is a '1', then the pin is an input.

All port pins are defined as inputs after a Reset. Reads from the latch (LATx), read the latch. Writes to the latch, write the latch. Reads from the port (PORTx), read the port pins, while writes to the port pins, write the latch. Any bit and its associated data and control registers that are not valid for a particular device are disabled. This means the corresponding LATx and TRISx registers, and the port pin are read as zeros.

When a pin is shared with another peripheral or function that is defined as an input only, it is nevertheless regarded as a dedicated port because there is no other competing source of outputs. Table 8-1 shows the pin availability. Table 8-2 shows the 5V input tolerant pins across this device.

TABLE 8-12: PORTE REGISTER SUMMARY

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
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 8-39: RPINR27: PERIPHERAL PIN SELECT INPUT REGISTER 27

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
U3DSRR7	U3DSRR6	U3DSRR5	U3DSRR4	U3DSRR3	U3DSRR2	U3DSRR1	U3DSRR0
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
U3RXR7	U3RXR6	U3RXR5	U3RXR4	U3RXR3	U3RXR2	U3RXR1	U3RXR0
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 **U3DSRR<7:0>:** Assign UART3 Data-Set-Ready ($\overline{\text{U3DSR}}$) to the Corresponding RPn Pin bits
See Table 8-4.

bit 7-0 **U3RXR<7:0>:** Assign UART3 Receive (U3RX) to the Corresponding RPn Pin bits
See Table 8-4.

REGISTER 8-40: RPINR29: PERIPHERAL PIN SELECT INPUT REGISTER 29

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SCK3R7	SCK3R6	SCK3R5	SCK3R4	SCK3R3	SCK3R2	SCK3R1	SCK3R0
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
SDI3R7	SDI3R6	SDI3R5	SDI3R4	SDI3R3	SDI3R2	SDI3R1	SDI3R0
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 **SCK3R<7:0>:** Assign SPI3 Clock Input (SCK3IN) to the Corresponding RPn Pin bits
See Table 8-4.

bit 7-0 **SDI3R<7:0>:** Assign SPI3 Data Input (SDI3) to the Corresponding RPn Pin bits
See Table 8-4.

REGISTER 8-53: RPINR49: PERIPHERAL PIN SELECT INPUT REGISTER 49

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
U3CTSR7	U3CTSR6	U3CTSR5	U3CTSR4	U3CTSR3	U3CTSR2	U3CTSR1	U3CTSR0
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
U2CTSR7	U2CTSR6	U2CTSR5	U2CTSR4	U2CTSR3	U2CTSR2	U2CTSR1	U2CTSR0
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 **U3CTSR<7:0>:** Assign UART3 Clear-to-Send ($\overline{\text{U3CTS}}$) to the Corresponding RPn Pin bits
See Table 8-4.

bit 7-0 **U2CTSR<7:0>:** Assign UART2 Clear-to-Send ($\overline{\text{U2CTS}}$) to the Corresponding RPn Pin bits
See Table 8-4.

REGISTER 8-66: RPOR12: PERIPHERAL PIN SELECT OUTPUT REGISTER 12

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP57R5	RP57R4	RP57R3	RP57R2	RP57R1	RP57R0
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
—	—	RP56R5	RP56R4	RP56R3	RP56R2	RP56R1	RP56R0
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 **Unimplemented:** Read as '0'

bit 13-8 **RP57R<5:0>:** Peripheral Output Function is Assigned to RP57 Output Pin bits (see Table 8-7 for peripheral function numbers)

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **RP56R<5:0>:** Peripheral Output Function is Assigned to RP56 Output Pin bits (see Table 8-7 for peripheral function numbers)

REGISTER 8-67: RPOR13: PERIPHERAL PIN SELECT OUTPUT REGISTER 13

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP59R5	RP59R4	RP59R3	RP59R2	RP59R1	RP59R0
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
—	—	RP58R5	RP58R4	RP58R3	RP58R2	RP58R1	RP58R0
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 **Unimplemented:** Read as '0'

bit 13-8 **RP59R<5:0>:** Peripheral Output Function is Assigned to RP59 Output Pin bits (see Table 8-7 for peripheral function numbers)

bit 7-6 **Unimplemented:** Read as '0'

bit 5-0 **RP58R<5:0>:** Peripheral Output Function is Assigned to RP58 Output Pin bits (see Table 8-7 for peripheral function numbers)

REGISTER 9-8: APLLDIV1: APLL OUTPUT DIVIDER REGISTER

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

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-1
—	APOST1DIV<2:0>(1,2)			—	APOST2DIV<2:0>(1,2)		
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 15-10 **Unimplemented:** Read as '0'

bit 9-8 **AVCODIV<1:0>:** APLL VCO Output Divider Select bits

11 = AFVCO
10 = AFVCO/2
01 = AFVCO/3
00 = AFVCO/4

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

bit 6-4 **APOST1DIV<2:0>:** APLL Output Divider #1 Ratio bits(1,2)

APOST1DIV<2:0> can have a valid value, from 1 to 7 (the APOST1DIVx value should be greater than or equal to the APOST2DIVx value). The APOST1DIVx divider is designed to operate at higher clock rates than the APOST2DIVx divider.

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

bit 2-0 **APOST2DIV<2:0>:** APLL Output Divider #2 Ratio bits(1,2)

APOST2DIV<2:0> can have a valid value, from 1 to 7 (the APOST2DIVx value should be less than or equal to the APOST1DIVx value). The APOST1DIVx divider is designed to operate at higher clock rates than the APOST2DIVx divider.

Note 1: The APOST1DIVx and APOST2DIVx values must not be changed while the PLL is operating.

2: The default values for APOST1DIVx and APOST2DIVx are 4 and 1, respectively, yielding a 150 MHz system source clock.

REGISTER 9-11: REFOCONH: REFERENCE CLOCK CONTROL HIGH REGISTER

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	RODIV<14:8>						
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
RODIV<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 **Unimplemented:** Read as '0'

bit 14-0 **RODIV<14:0>:** Reference Clock Integer Divider Select bits

Divider for the selected input clock source is two times the selected value.

111 1111 1111 1111 = Base clock value divided by 65,534 (2 * 7FFFh)
111 1111 1111 1110 = Base clock value divided by 65,532 (2 * 7FFEh)
111 1111 1111 1101 = Base clock value divided by 65,530 (2 * 7FFDh)
...
000 0000 0000 0010 = Base clock value divided by 4 (2 * 2)
000 0000 0000 0001 = Base clock value divided by 2 (2 * 1)
000 0000 0000 0000 = Base clock value

REGISTER 11-9: C1TBCH: CAN TIME BASE COUNTER REGISTER HIGH(1,2)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TBC<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
TBC<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 **TBC<31:16>** CAN Time Base Counter bits
This is a free-running timer that increments every TBCPREx clock when TBCEN is set.

Note 1: The Time Base Counter (TBC) will be stopped and reset when TBCEN = 0 to save power.
2: The TBC prescaler count will be reset on any write to C1TBCH/L (TBCPREx will be unaffected).

REGISTER 11-10: C1TBCL: CAN TIME BASE COUNTER REGISTER LOW(1,2)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TBC<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
TBC<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 **TBC<15:0>** CAN Time Base Counter bits
This is a free-running timer that increments every TBCPREx clock when TBCEN is set.

Note 1: The TBC will be stopped and reset when TBCEN = 0 to save power.
2: The TBC prescaler count will be reset on any write to C1TBCH/L (TBCPREx will be unaffected).

REGISTER 11-38: 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 11-39: 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 12-8: CMBTRIGH: COMBINATIONAL TRIGGER 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
CTB8EN	CTB7EN	CTB6EN	CTB5EN	CTB4EN	CTB3EN	CTB2EN	CTB1EN
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 **CTB8EN:** Enable Trigger Output from PWM Generator #8 as Source for Combinational Trigger B bit
1 = Enables specified trigger signal to be OR'd into the Combinatorial Trigger B signal
0 = Disabled

bit 6 **CTB7EN:** Enable Trigger Output from PWM Generator #7 as Source for Combinational Trigger B bit
1 = Enables specified trigger signal to be OR'd into the Combinatorial Trigger B signal
0 = Disabled

bit 5 **CTB6EN:** Enable Trigger Output from PWM Generator #6 as Source for Combinational Trigger B bit
1 = Enables specified trigger signal to be OR'd into the Combinatorial Trigger B signal
0 = Disabled

bit 4 **CTB5EN:** Enable Trigger Output from PWM Generator #5 as Source for Combinational Trigger B bit
1 = Enables specified trigger signal to be OR'd into the Combinatorial Trigger B signal
0 = Disabled

bit 3 **CTB4EN:** Enable Trigger Output from PWM Generator #4 as Source for Combinational Trigger B bit
1 = Enables specified trigger signal to be OR'd into the Combinatorial Trigger B signal
0 = Disabled

bit 2 **CTB3EN:** Enable Trigger Output from PWM Generator #3 as Source for Combinational Trigger B bit
1 = Enables specified trigger signal to be OR'd into the Combinatorial Trigger B signal
0 = Disabled

bit 1 **CTB2EN:** Enable Trigger Output from PWM Generator #2 as Source for Combinational Trigger B bit
1 = Enables specified trigger signal to be OR'd into the Combinatorial Trigger B signal
0 = Disabled

bit 0 **CTB1EN:** Enable Trigger Output from PWM Generator #1 as Source for Combinational Trigger B bit
1 = Enables specified trigger signal to be OR'd into the Combinatorial Trigger B signal
0 = Disabled

REGISTER 12-13: PGxCONH: PWM GENERATOR x CONTROL REGISTER HIGH

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
MDCSEL	MPERSEL	MPHSEL	—	MSTEN	UPMOD2	UPMOD1	UPMOD0
bit 15							bit 8

r-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	TRGMOD	—	—	SOCS3(1,2,3)	SOCS2(1,2,3)	SOCS1(1,2,3)	SOCS0(1,2,3)
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 **MDCSEL:** Master Duty Cycle Register Select bit
1 = PWM Generator uses MDC register
0 = PWM Generator uses PGxDC register

bit 14 **MPERSEL:** Master Period Register Select bit
1 = PWM Generator uses MPER register
0 = PWM Generator uses PGxPER register

bit 13 **MPHSEL:** Master Phase Register Select bit
1 = PWM Generator uses MPHASE register
0 = PWM Generator uses PGxPHASE register

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

bit 11 **MSTEN:** Master Update Enable bit
1 = PWM Generator broadcasts software set/clear of the UPDREQ status bit and EOC signal to other PWM Generators
0 = PWM Generator does not broadcast the UPDREQ status bit state or EOC signal

bit 10-8 **UPMOD<2:0>:** PWM Buffer Update Mode Selection bits
011 = Slaved immediate update
Data registers immediately, or as soon as possible, when a Master update request is received. A Master update request will be transmitted if MSTEN = 1 and UPDATE = 1 for the requesting PWM Generator.
010 = Slaved SOC update
Data registers at start of next cycle if a Master update request is received. A master update request will be transmitted if MSTEN = 1 and UPDATE = 1 for the requesting PWM Generator.
001 = Immediate update
Data registers immediately, or as soon as possible, if UPDATE = 1. The UPDATE status bit will be cleared automatically after the update occurs (UPDATE = 1). The UPDATE status bit will be cleared automatically after the update occurs.
000 = SOC update
Data registers at start of next PWM cycle if UPDATE = 1. The UPDATE status bit will be cleared automatically after the update occurs.(1)

bit 7 **Reserved:** Maintain as '0'

Note 1: The PCI selected Sync signal is always available to be OR'd with the selected SOC signal per the SOCS<3:0> bits if the PCI Sync function is enabled.

2: The source selected by the SOCS<3:0> bits MUST operate from the same clock source as the local PWM Generator. If not, the source must be routed through the PCI Sync logic so the trigger signal may be synchronized to the PWM Generator clock domain.

3: PWM Generators are grouped into groups of four: PG1-PG4 and PG5-PG8, if available. Any generator within a group of four may be used to trigger another generator within the same group.

TABLE 22-5: SYNCHRONIZATION SOURCES

SYNC<4:0>	Synchronization Source
00000	None; Timer with Rollover on CCPxPR Match or FFFFh
00001	Module's Own Timer Sync Out
00010	Sync Output SCCP2
00011	Sync Output SCCP3
00100	Sync Output SCCP4
00101	Sync Output SCCP5
00110	Sync Output SCCP6
00111	Sync Output SCCP7
01000	Sync Output SCCP8
01001	INT0
01010	INT1
01011	INT2
01100	UART1 RX Edge Detect
01101	UART1 TX Edge Detect
01110	UART2 RX Edge Detect
01111	UART2 TX Edge Detect
10000	CLC1 Output
10001	CLC2 Output
10010	CLC3 Output
10011	CLC4 Output
10100	UART3 RX Edge Detect
10101	UART3 TX Edge Detect
10110	Sync Output MCCP9
10111	Comparator 1 Output
11000	Comparator 2 Output
11001	Comparator 3 Output
11010-11110	Reserved
11111	None; Timer with Auto-Rollover (FFFFh → 0000h)

REGISTER 22-5: CCPxCON3L: CCPx CONTROL 3 LOW REGISTERS(1)

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
—	—	DT<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 **DT<5:0>:** CCPx Dead-Time Select bits

111111 = Inserts 63 dead-time delay periods between complementary output signals
111110 = Inserts 62 dead-time delay periods between complementary output signals
...
000010 = Inserts 2 dead-time delay periods between complementary output signals
000001 = Inserts 1 dead-time delay period between complementary output signals
000000 = Dead-time logic is disabled

Note 1: This register is implemented in the MCCP9 module only.

REGISTER 26-3: IBIASCONL: CURRENT BIAS GENERATOR 50 µA CURRENT SOURCE CONTROL LOW REGISTER

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	SHRSRCEN1	SHRSNKEN1	GENSRCEN1	GENSNKEN1	SRCEN1	SNKEN1
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
—	—	SHRSRCEN0	SHRSNKEN0	GENSRCEN0	GENSNKEN0	SRCEN0	SNKEN0
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 **Unimplemented:** Read as '0'

bit 13 **SHRSRCEN1:** Share Source Enable for Output #1 bit
1 = Sourcing Current Mirror mode is enabled (uses reference from another source)
0 = Sourcing Current Mirror mode is disabled

bit 12 **SHRSNKEN1:** Share Sink Enable for Output #1 bit
1 = Sinking Current Mirror mode is enabled (uses reference from another source)
0 = Sinking Current Mirror mode is disabled

bit 11 **GENSRCEN1:** Generated Source Enable for Output #1 bit
1 = Source generates the current source mirror reference
0 = Source does not generate the current source mirror reference

bit 10 **GENSNKEN1:** Generated Sink Enable for Output #1 bit
1 = Source generates the current source mirror reference
0 = Source does not generate the current source mirror reference

bit 9 **SRCEN1:** Source Enable for Output #1 bit
1 = Current source is enabled
0 = Current source is disabled

bit 8 **SNKEN1:** Sink Enable for Output #1 bit
1 = Current sink is enabled
0 = Current sink is disabled

bit 7-6 **Unimplemented:** Read as '0'

bit 5 **SHRSRCEN0:** Share Source Enable for Output #0 bit
1 = Sourcing Current Mirror mode is enabled (uses reference from another source)
0 = Sourcing Current Mirror mode is disabled

bit 4 **SHRSNKEN0:** Share Sink Enable for Output #0 bit
1 = Sinking Current Mirror mode is enabled (uses reference from another source)
0 = Sinking Current Mirror mode is disabled

bit 3 **GENSRCEN0:** Generated Source Enable for Output #0 bit
1 = Source generates the current source mirror reference
0 = Source does not generate the current source mirror reference

bit 2 **GENSNKEN0:** Generated Sink Enable for Output #0 bit
1 = Source generates the current source mirror reference
0 = Source does not generate the current source mirror reference

bit 1 **SRCEN0:** Source Enable for Output #0 bit
1 = Current source is enabled
0 = Current source is disabled

bit 0 **SNKEN0:** Sink Enable for Output #0 bit
1 = Current sink is enabled
0 = Current sink is disabled

REGISTER 28-7: DMTPSCNTL: DMT POST-CONFIGURE COUNT STATUS 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
PSCNT<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
PSCNT<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 **PSCNT<15:0>:** Lower DMT Instruction Count Value Configuration Status bits
This is always the value of the FDMTCNTL Configuration register.

REGISTER 28-8: DMTPSCNTH: DMT POST-CONFIGURE COUNT STATUS 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
PSCNT<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
PSCNT<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 **PSCNT<31:16>:** Higher DMT Instruction Count Value Configuration Status bits
This is always the value of the FDMTCNTH Configuration register.

TABLE 30-4: DEVICE IDs FOR THE dsPIC33CK256MP508 FAMILY

Device	DEVID
Device IDs for dsPIC33CK256MP508 Family with CAN FD	
dsPIC33CK256MP508	0x7C74
dsPIC33CK256MP506	0x7C73
dsPIC33CK256MP505	0x7C72
dsPIC33CK256MP503	0x7C71
dsPIC33CK256MP502	0x7C70
dsPIC33CK128MP508	0x7C64
dsPIC33CK128MP506	0x7C63
dsPIC33CK128MP505	0x7C62
dsPIC33CK128MP503	0x7C61
dsPIC33CK128MP502	0x7C60
dsPIC33CK64MP508	0x7C54
dsPIC33CK64MP506	0x7C53
dsPIC33CK64MP505	0x7C52
dsPIC33CK64MP503	0x7C51
dsPIC33CK64MP502	0x7C50
dsPIC33CK32MP506	0x7C43
dsPIC33CK32MP505	0x7C42
dsPIC33CK32MP503	0x7C41
dsPIC33CK32MP502	0x7C40
Device IDs for dsPIC33CK256MP508 Family without CAN FD	
dsPIC33CK256MP208	0x7C34
dsPIC33CK256MP206	0x7C33
dsPIC33CK256MP205	0x7C32
dsPIC33CK256MP203	0x7C31
dsPIC33CK256MP202	0x7C30
dsPIC33CK128MP208	0x7C24
dsPIC33CK128MP206	0x7C23
dsPIC33CK128MP205	0x7C22
dsPIC33CK128MP203	0x7C21
dsPIC33CK128MP202	0x7C20
dsPIC33CK64MP208	0x7C14
dsPIC33CK64MP206	0x7C13
dsPIC33CK64MP205	0x7C12
dsPIC33CK64MP203	0x7C11
dsPIC33CK64MP202	0x7C10
dsPIC33CK32MP206	0x7C03
dsPIC33CK32MP205	0x7C02
dsPIC33CK32MP203	0x7C01
dsPIC33CK32MP202	0x7C00

FIGURE 33-11: SPIx SLAVE MODE (FULL-DUPLEX, CKE = 0, CKP = x, SMP = 0) TIMING CHARACTERISTICS