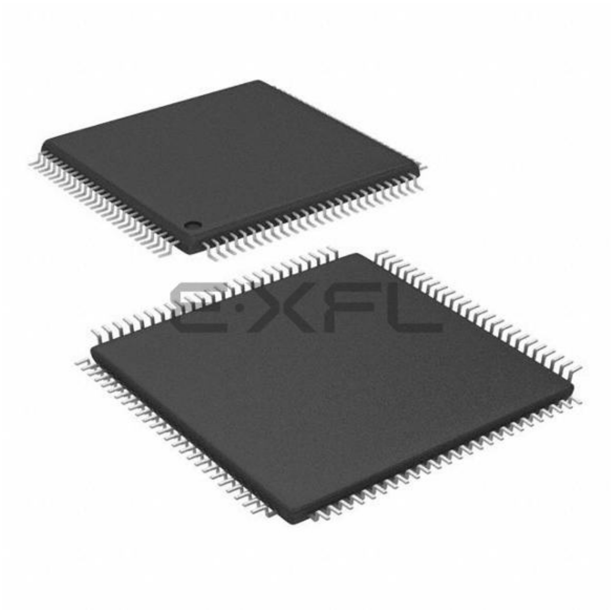


Welcome to E-XFL.COM

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	MIPS32® microAptiv™
Core Size	32-Bit Single-Core
Speed	200MHz
Connectivity	CANbus, EBI/EMI, Ethernet, I²C, SPI, SQI, UART/USART, USB OTG
Peripherals	Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT
Number of I/O	78
Program Memory Size	1MB (1M x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512K x 8
Voltage - Supply (Vcc/Vdd)	2.3V ~ 3.6V
Data Converters	A/D 40x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-TQFP
Supplier Device Package	100-TQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic32mz1024ech100-i-pf

TABLE 4: PIN NAMES FOR 124-PIN DEVICES

124-PIN VTLA (BOTTOM VIEW)

A34
A17
B13 B29

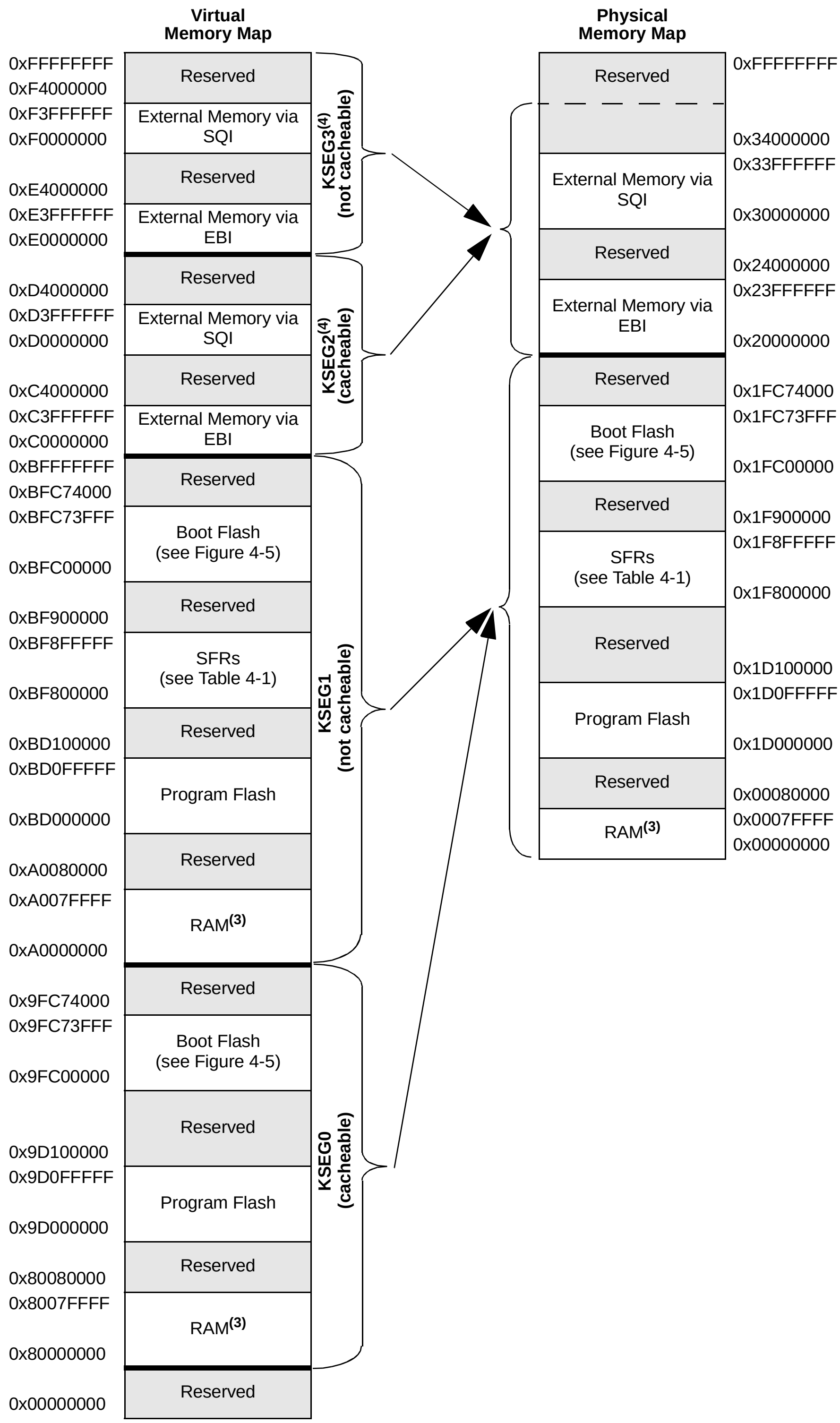
PIC32MZ0512EC(E/F/K)124
PIC32MZ1024EC(G/H/M)124
PIC32MZ1024EC(E/F/K)124
PIC32MZ2048EC(G/H/M)124

B1 B41
B56
A51
A1
A68
Polarity Indicator

Package Pin #	Full Pin Name	Package Pin #	Full Pin Name
A1	No Connect	A35	VBUS
A2	AN23/RG15	A36	VUSB3V3
A3	EBID5/AN17/RPE5/PMD5/RE5	A37	D-
A4	EBID7/AN15/PMD7/RE7	A38	RPF3/USBID/RF3
A5	AN35/ETXD0/RJ8	A39	EBIRDY2/RPF8/SCL3/RF8
A6	EBIA12/AN21/RPC2/PMA12/RC2	A40	ERXD3/RH9
A7	$\overline{\text{EBIOE}}$/AN19/RPC4/PMRD/RC4	A41	$\overline{\text{EBICS0}}$/SCL2/RA2
A8	EBIA4/AN13/C1INC/RPG7/SDA4/PMA4/RG7	A42	EBIA14/PMCS1/PMA14/RA4
A9	VSS	A43	VSS
A10	$\overline{\text{MCLR}}$	A44	EBIA8/RPF5/SCL5/PMA8/RF5
A11	TMS/EBIA16/AN24/RA0	A45	RPA15/SDA1/RA15
A12	AN26/RPE9/RE9	A46	RPD10/SCK4/RD10
A13	AN4/C1INB/RB4	A47	ECRS/RH12
A14	AN3/C2INA/RPB3/RB3	A48	RPD0/RTCC/INT0/RD0
A15	VDD	A49	SOSCO/RPC14/T1CK/RC14
A16	AN2/C2INB/RPB2/RB2	A50	VDD
A17	PGEC1/AN1/RPB1/RB1	A51	VSS
A18	PGED1/AN0/RPB0/RB0	A52	RPD1/SCK1/RD1
A19	PGED2/AN47/RPB7/RB7	A53	EBID15/RPD3/PMD15/RD3
A20	VREF+/CVREF+/AN28/RA10	A54	EBID13/PMD13/RD13
A21	AVSS	A55	EMDIO/RJ1
A22	AN39/ETXD3/RH1	A56	$\overline{\text{SQICS0}}$/RPD4/RD4
A23	EBIA7/AN49/RPB9/PMA7/RB9	A57	ETXEN/RPD6/RD6
A24	AN6/RB11	A58	VDD
A25	VDD	A59	EBID11/RPF0/PMD11/RF0
A26	TDI/EBIA18/AN30/RPF13/SCK5/RF13	A60	EBID9/RPG1/PMD9/RG1
A27	EBIA11/AN7/PMA11/RB12	A61	TRCLK/SQICLK/RA6
A28	EBIA1/AN9/RPB14/SCK3/PMA1/RB14	A62	RJ4
A29	VSS	A63	VSS
A30	AN40/ERXERR/RH4	A64	EBID1/PMD1/RE1
A31	AN42/ERXD2/RH6	A65	TRD1/SQID1/RG12
A32	AN33/RPD15/SCK6/RD15	A66	EBID2/SQID2/PMD2/RE2
A33	OSC2/CLKO/RC15	A67	EBID4/AN18/PMD4/RE4
A34	No Connect	A68	No Connect

Note 1: The RPn pins can be used by remappable peripherals. See Table 1 for the available peripherals and **Section 12.3 "Peripheral Pin Select (PPS)"** for restrictions.
2: Every I/O port pin (RAx-RJx) can be used as a change notification pin (CNAx-CNJx). See **Section 12.0 "I/O Ports"** for more information.
3: Shaded pins are 5V tolerant.
4: The metal plane at the bottom of the device is not connected to any pins and is recommended to be connected to VSS externally.

FIGURE 4-3: MEMORY MAP FOR DEVICES WITH 1024 KB OF PROGRAM MEMORY AND 512 KB OF RAM(1,2)



Note 1: Memory areas are not shown to scale.
2: The Cache, MMU, and TLB are initialized by compiler start-up code.
3: RAM memory is divided into two equal banks: RAM Bank 1 and RAM Bank 2 on a half boundary.
4: The MMU must be enabled and the TLB must be set up to access this segment.

TABLE 4-6: SYSTEM BUS TARGETS AND ASSOCIATED PROTECTION REGISTERS

Target Number	Target Description(5)	SBTxREGy Register							SBTxRDy Register		SBTxWRy Register	
		Name	Region Base (BASE<21:0>) (see Note 2)	Physical Start Address	Region Size (SIZE<4:0>) (see Note 3)	Region Size	Priority (PRI)	Priority Level	Name	Read Permission (GROUP3, GROUP2, GROUP1, GROUP0)	Name	Write Permission (GROUP3, GROUP2, GROUP1, GROUP0)
0	System Bus	SBT0REG0	R	0x1F8F0000	R	64 KB	—	0	SBT0RD0	R/W(1)	SBT0WR0	R/W(1)
		SBT0REG1	R	0x1F8F8000	R	32 KB	—	3	SBT0RD1	R/W(1)	SBT0WR1	R/W(1)
1	**Flash Memory(6):** Program Flash Boot Flash Prefetch Module	SBT1REG0	R	0x1D000000	R(4)	R(4)	—	0	SBT1RD0	R/W(1)	SBT1WR0	0, 0, 0, 0
		SBT1REG2	R	0x1F8E0000	R	4 KB	1	2	SBT1RD2	R/W(1)	SBT1WR2	R/W(1)
		SBT1REG3	R/W	R/W	R/W	R/W	1	2	SBT1RD3	R/W(1)	SBT1WR3	0, 0, 0, 0
		SBT1REG4	R/W	R/W	R/W	R/W	1	2	SBT1RD4	R/W(1)	SBT1WR4	0, 0, 0, 0
		SBT1REG5	R/W	R/W	R/W	R/W	1	2	SBT1RD5	R/W(1)	SBT1WR5	0, 0, 0, 0
		SBT1REG6	R/W	R/W	R/W	R/W	1	2	SBT1RD6	R/W(1)	SBT1WR6	0, 0, 0, 0
		SBT1REG7	R/W	R/W	R/W	R/W	0	1	SBT1RD7	R/W(1)	SBT1WR7	0, 0, 0, 0
		SBT1REG8	R/W	R/W	R/W	R/W	0	1	SBT1RD8	R/W(1)	SBT1WR8	0, 0, 0, 0
2	RAM Bank 1 Memory	SBT2REG0	R	0x00000000	R(4)	R(4)	—	0	SBT2RD0	R/W(1)	SBT2WR0	R/W(1)
		SBT2REG1	R/W	R/W	R/W	R/W	—	3	SBT2RD1	R/W(1)	SBT2WR1	R/W(1)
		SBT2REG2	R/W	R/W	R/W	R/W	0	1	SBT2RD2	R/W(1)	SBT2WR2	R/W(1)
3	RAM Bank 2 Memory	SBT3REG0	R(4)	R(4)	R(4)	R(4)	—	0	SBT3RD0	R/W(1)	SBT3WR0	R/W(1)
		SBT3REG1	R/W	R/W	R/W	R/W	—	3	SBT3RD1	R/W(1)	SBT3WR1	R/W(1)
		SBT3REG2	R/W	R/W	R/W	R/W	0	1	SBT3RD2	R/W(1)	SBT3WR2	R/W(1)
4	External Memory via EBI and EBI Module(6)	SBT4REG0	R	0x20000000	R	64 MB	—	0	SBT4RD0	R/W(1)	SBT4WR0	R/W(1)
		SBT4REG2	R	0x1F8E1000	R	4 KB	0	1	SBT4RD2	R/W(1)	SBT4WR2	R/W(1)
5	**Peripheral Set 1:** System Control Flash Control DMT/WDT RTCC CVR PPS Input PPS Output Interrupts DMA	SBT5REG0	R	0x1F800000	R	128 KB	—	0	SBT5RD0	R/W(1)	SBT5WR0	R/W(1)
		SBT5REG1	R/W	R/W	R/W	R/W	—	3	SBT5RD1	R/W(1)	SBT5WR1	R/W(1)
		SBT5REG2	R/W	R/W	R/W	R/W	0	1	SBT5RD2	R/W(1)	SBT5WR2	R/W(1)

Legend: R = Read; R/W = Read/Write; ‘x’ in a register name = 0-13; ‘y’ in a register name = 0-8.

Note 1: Reset values for these bits are ‘0’, ‘1’, ‘1’, ‘1’, respectively.
2: The BASE<21:0> bits must be set to the corresponding Physical Address and right shifted by 10 bits. For Read-only bits, this value is set by hardware on Reset.
3: The SIZE<4:0> bits must be set to the corresponding Region Size, based on the following formula: Region Size = $2^{(SIZE-1)}$ x 1024 bytes. For read-only bits, this value is set by hardware on Reset.
4: Refer to the Device Memory Maps (Figure 4-1 through Figure 4-4) for specific device memory sizes and start addresses.
5: See Table 4-1for information on specific target memory size and start addresses.
6: The SBTxREG1 SFRs are reserved, and therefore, are not listed in this table for this target.

TABLE 4-18: SYSTEM BUS TARGET 10 REGISTER MAP

Virtual Address (BF8F_#)	Register Name	Bit Range	Bits 31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0	All Resets
A820	SBT10ELOG1	31:16	MULTI	—	—	—	CODE<3:0>				—	—	—	—	—	—	—	—	0000
		15:0	INITID<7:0>								REGION<3:0>				—	CMD<2:0>			0000
A824	SBT10ELOG2	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GROUP<1:0>		0000
A828	SBT10ECON	31:16	—	—	—	—	—	—	—	ERRP	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
A830	SBT10ECLRS	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLEAR	0000
A838	SBT10ECLRM	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLEAR	0000
A840	SBT10REG0	31:16	BASE<21:6>																xxxx
		15:0	BASE<5:0>						PRI	—	SIZE<4:0>					—	—	—	xxxx
A850	SBT10RD0	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	xxxx
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	GROUP3	GROUP2	GROUP1	GROUP0	xxxx
A858	SBT10WR0	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	xxxx
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	GROUP3	GROUP2	GROUP1	GROUP0	xxxx

Legend: x = unknown value on Reset; — = unimplemented, read as ‘0’. Reset values are shown in hexadecimal.

Note: For reset values listed as ‘xxxx’, please refer to Table 4-6 for the actual reset values.

TABLE 4-21: SYSTEM BUS TARGET 13 REGISTER MAP

Virtual Address (BF8F_#)	Register Name	Bit Range	Bits 31/15	30/14	29/13	28/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0	All Resets
B420	SBT13ELOG1	31:16	MULTI	—	—	—	CODE<3:0>				—	—	—	—	—	—	—	—	0000
		15:0	INITID<7:0>								REGION<3:0>				—	CMD<2:0>			0000
B424	SBT13ELOG2	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GROUP<1:0>		0000
B428	SBT13ECON	31:16	—	—	—	—	—	—	—	ERRP	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
B430	SBT13ECLRS	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLEAR	0000
B438	SBT13ECLRM	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLEAR	0000
B440	SBT13REG0	31:16	BASE<21:6>																xxxx
		15:0	BASE<5:0>						PRI	—	SIZE<4:0>					—	—	—	xxxx
B450	SBT13RD0	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	xxxx
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	GROUP3	GROUP2	GROUP1	GROUP0	xxxx
B458	SBT13WR0	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	xxxx
		15:0	—	—	—	—	—	—	—	—	—	—	—	—	GROUP3	GROUP2	GROUP1	GROUP0	xxxx

Legend: x = unknown value on Reset; — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

Note: For reset values listed as 'xxxx', please refer to Table 4-6 for the actual reset values.

REGISTER 5-1: NVMCON: PROGRAMMING CONTROL REGISTER

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R/W-0, HC	R/W-0	R-0, HS, HC	R-0, HS, HC	U-0	U-0	U-0	U-0
	WR(1)	WREN(1)	WRERR(1)	LVDERR(1)	—	—	—	—
7:0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
	SWAP	—	—	—	NVMOP<3:0>			

Legend:	HS = Hardware Set	HC = Hardware Cleared	
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 31-16 **Unimplemented:** Read as ‘0’

bit 15 **WR:** Write Control bit(1)

This bit cannot be cleared and can be set only when WREN = 1 and the unlock sequence has been performed.
1 = Initiate a Flash operation
0 = Flash operation is complete or inactive

bit 14 **WREN:** Write Enable bit(1)

1 = Enable writes to the WR bit and disables writes to the NVMOP<3:0> bits
0 = Disable writes to WR bit and enables writes to the NVMOP<3:0> bits

bit 13 **WRERR:** Write Error bit(1)

This bit can be cleared only by setting the NVMOP<3:0> bits = 0000 and initiating a Flash operation.
1 = Program or erase sequence did not complete successfully
0 = Program or erase sequence completed normally

bit 12 **LVDERR:** Low-Voltage Detect Error bit(1)

This bit can be cleared only by setting the NVMOP<3:0> bits = 0000 and initiating a Flash operation.
1 = Low-voltage detected (possible data corruption, if WRERR is set)
0 = Voltage level is acceptable for programming

bit 11-8 **Unimplemented:** Read as ‘0’

bit 7 **SWAP:** Program Flash Bank Swap Control bit

This bit can be modified only when the WREN bit is ‘0’ and the unlock sequence has been performed.
1 = Program Flash Bank 2 is mapped to the lower mapped region and program Flash Bank 1 is mapped to the upper mapped region
0 = Program Flash Bank 1 is mapped to the lower mapped region and program Flash Bank 2 is mapped to the upper mapped region

bit 6-4 **Unimplemented:** Read as ‘0’

Note 1: These bits are only reset by a Power-on Reset (POR) and are not affected by other reset sources.

2: This operation results in a “no operation” (NOP) when the Dynamic Flash ECC Configuration bits = 00 (FECCCON<1:0> (DVCFG0<9:8>)), which enables ECC at all times. For all other FECCCON<1:0> bit settings, this command will execute, but will not write the ECC bits for the word and can cause DED errors if dynamic Flash ECC is enabled (FECCCON<1:0> = 01). Refer to **Section 52. “Flash Program Memory with Support for Live Update”** (DS60001193) for information regarding ECC and Flash programming.

TABLE 8-1: SYSTEM AND PERIPHERAL CLOCK DISTRIBUTION

Peripheral	Clock Source														
	FRC	LPRC	SOSC	SYSCLK	USBCLK	PBCLK1(1)	PBCLK2	PBCLK3	PBCLK4	PBCLK5	PBCLK7	PBCLK8	REFCLKO1	REFCLKO2	REFCLKO3
CPU											X				
WDT		X				X(2)									
Deadman Timer						X(2)					X				
Flash	X(2)			X(2)		X(2)									
ADC	X			X				X(3)							X
Comparator								X							
Crypto										X					
RNG										X					
USB					X					X(3)					
CAN										X					
Ethernet										X(3)					
PMP							X								
I^2C							X								
UART							X								
RTCC		X	X			X(2)									
EBI												X			
SQI										X(3)				X	
SPI							X						X		
Timers			X(4)					X							
Output Compare								X							
Input Capture								X							
Ports									X						
DMA				X											
Interrupts				X											
Prefetch				X											
OSC2 Pin						X(5)									

Note 1: PBCLK1 is used by system modules and cannot be turned off.
2: SYSCLK/PBCLK1 is used to fetch data from/to the Flash Controller, while the FRC clock is used for programming.
3: Special Function Register (SFR) access only.
4: Timer1 only.
5: PBCLK1 divided by 2 is available on the OSC2 pin in certain clock modes.

8.1 Fail-Safe Clock Monitor (FSCM)

The PIC32MZ EC oscillator system includes a Fail-safe Clock Monitor (FSCM). The FSCM monitors the SYSCLK for continuous operation. If it detects that the SYSCLK has failed, it switches the SYSCLK over to the BFRC oscillator and triggers a NMI. The BFRC is an untuned 8 MHz oscillator that will drive the SYSCLK during FSCM event. When the NMI is executed, software can attempt to restart the main oscillator or shut down the system.

In Sleep mode both the SYSCLK and the FSCM halt, which prevents FSCM detection.

REGISTER 8-1: OSCCON: OSCILLATOR CONTROL REGISTER

bit 10-8 **NOSC<2:0>:** New Oscillator Selection bits
111 = Internal Fast RC (FRC) Oscillator divided by FRCDIV<2:0> bits (FRCDIV)
110 = Reserved
101 = Internal Low-Power RC (LPRC) Oscillator
100 = Secondary Oscillator (SOSC)
011 = Reserved
010 = Primary Oscillator (POSC) (HS or EC)
001 = System PLL (SPLL)
000 = Internal Fast RC (FRC) Oscillator divided by FRCDIV<2:0> bits (FRCDIV)

On Reset, these bits are set to the value of the FNOSC<2:0> Configuration bits (DEVCFG1<2:0>).

bit 7 **CLKLOCK:** Clock Selection Lock Enable bit
1 = Clock and PLL selections are locked
0 = Clock and PLL selections are not locked and may be modified

bit 6 **ULOCK:** USB PLL Lock Status bit
1 = Indicates that the USB PLL module is in lock or USB PLL module start-up timer is satisfied
0 = Indicates that the USB PLL module is out of lock or USB PLL module start-up timer is in progress or USB PLL is disabled

bit 5 **SLOCK:** System PLL Lock Status bit
1 = System PLL module is in lock or module start-up timer is satisfied
0 = System PLL module is out of lock, start-up timer is running or system PLL is disabled

bit 4 **SLPEN:** Sleep Mode Enable bit
1 = Device will enter Sleep mode when a WAIT instruction is executed
0 = Device will enter Idle mode when a WAIT instruction is executed

bit 3 **CF:** Clock Fail Detect bit
1 = FSCM has detected a clock failure
0 = No clock failure has been detected

bit 2 **Unimplemented:** Read as ‘0’

bit 1 **SOSCEN:** Secondary Oscillator (SOSC) Enable bit
1 = Enable Secondary Oscillator
0 = Disable Secondary Oscillator

bit 0 **OSWEN:** Oscillator Switch Enable bit(1)
1 = Initiate an oscillator switch to selection specified by NOSC<2:0> bits
0 = Oscillator switch is complete

Note 1: The reset value for this bit depends on the setting of the IESO (DEVCFG1<7>) bit. When IESO = 1, the reset value is ‘1’. When IESO = 0, the reset value is ‘0’.

Note: Writes to this register require an unlock sequence. Refer to **Section 42. “Oscillators with Enhanced PLL”** (DS60001250) in the *“PIC32 Family Reference Manual”* for details.

REGISTER 11-13: USBOTG: USB OTG CONTROL/STATUS REGISTER (CONTINUED)

bit 0 **SESSION:** Active Session Control/Status bit

'A' device:
1 = Start a session
0 = End a session

'B' device:
1 = (Read) Session has started or is in progress, (Write) Initiate the Session Request Protocol
0 = When USB module is in Suspend mode, clearing this bit will cause a software disconnect

Clearing this bit when the USB module is not suspended will result in undefined behavior.

TABLE 12-1: INPUT PIN SELECTION

Peripheral Pin	[*pin name*]R SFR	[*pin name*]R bits	[*pin name*]R Value to RPn Pin Selection
INT3	INT3R	INT3R<3:0>	`0000` = RPD2 `0001` = RPG8 `0010` = RPF4 `0011` = RPD10 `0100` = RPF1 `0101` = RPB9 `0110` = RPB10 `0111` = RPC14 `1000` = RPB5 `1001` = Reserved `1010` = RPC1(1) `1011` = RPD14(1) `1100` = RPG1(1) `1101` = RPA14(1) `1110` = RPD6(2) `1111` = Reserved
T2CK	T2CKR	T2CKR<3:0>	
T6CK	T6CKR	T6CKR<3:0>	
IC3	IC3R	IC3R<3:0>	
IC7	IC7R	IC7R<3:0>	
U1RX	U1RXR	U1RXR<3:0>	
$\overline{\text{U2CTS}}$	U2CTSR	U2CTSR<3:0>	
U5RX	U5RXR	U5RXR<3:0>	
$\overline{\text{U6CTS}}$	U6CTSR	U6CTSR<3:0>	
SDI1	SDI1R	SDI1R<3:0>	
SDI3	SDI3R	SDI3R<3:0>	
SDI5(1)	SDI5R(1)	SDI5R<3:0>(1)	
$\overline{\text{SS6}}$(1)	SS6R(1)	SS6R<3:0>(1)	
REFCLKI1	REFCLKI1R	REFCLKI1R<3:0>	
INT4	INT4R	INT4R<3:0>	`0000` = RPD3 `0001` = RPG7 `0010` = RPF5 `0011` = RPD11 `0100` = RPF0 `0101` = RPB1 `0110` = RPE5 `0111` = RPC13 `1000` = RPB3 `1001` = Reserved `1010` = RPC4(1) `1011` = RPD15(1) `1100` = RPG0(1) `1101` = RPA15(1) `1110` = RPD7(2) `1111` = Reserved
T5CK	T5CKR	T5CKR<3:0>	
T7CK	T7CKR	T7CKR<3:0>	
IC4	IC4R	IC4R<3:0>	
IC8	IC8R	IC8R<3:0>	
U3RX	U3RXR	U3RXR<3:0>	
$\overline{\text{U4CTS}}$	U4CTSR	U4CTSR<3:0>	
SDI2	SDI2R	SDI2R<3:0>	
SDI4	SDI4R	SDI4R<3:0>	
C1RX(3)	C1RXR(3)	C1RXR<3:0>(3)	
REFCLKI4	REFCLKI4R	REFCLKI4R<3:0>	
INT2	INT2R	INT2R<3:0>	`0000` = RPD9 `0001` = RPG6 `0010` = RPB8 `0011` = RPB15 `0100` = RPD4 `0101` = RPB0 `0110` = RPE3 `0111` = RPB7 `1000` = Reserved `1001` = RPF12(1) `1010` = RPD12(1) `1011` = RPF8(1) `1100` = RPC3(1) `1101` = RPE9(1) `1110` = Reserved `1111` = Reserved
T3CK	T3CKR	T3CKR<3:0>	
T8CK	T8CKR	T8CKR<3:0>	
IC2	IC2R	IC2R<3:0>	
IC5	IC5R	IC5R<3:0>	
IC9	IC9R	IC9R<3:0>	
$\overline{\text{U1CTS}}$	U1CTSR	U1CTSR<3:0>	
U2RX	U2RXR	U2RXR<3:0>	
$\overline{\text{U5CTS}}$	U5CTSR	U5CTSR<3:0>	
$\overline{\text{SS1}}$	SS1R	SS1R<3:0>	
$\overline{\text{SS3}}$	SS3R	SS3R<3:0>	
$\overline{\text{SS4}}$	SS4R	SS4R<3:0>	
$\overline{\text{SS5}}$(1)	SS5R(1)	SS5R<3:0>(1)	
C2RX(3)	C2RXR(3)	C2RXR<3:0>(3)	

Note 1: This selection is not available on 64-pin devices.
2: This selection is not available on 64-pin or 100-pin devices.
3: This selection is not available on devices without a CAN module.

REGISTER 18-1: OCxCON: OUTPUT COMPARE 'x' CONTROL REGISTER

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
	ON	—	SIDL	—	—	—	—	—
7:0	U-0	U-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	—	OC32	OCFLT(1)	OCTSEL(2)	OCM<2:0>		

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 31-16 **Unimplemented:** Read as '0'

bit 15 **ON:** Output Compare Peripheral On bit
1 = Output Compare peripheral is enabled
0 = Output Compare peripheral is disabled

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

bit 13 **SIDL:** Stop in Idle Mode bit
1 = Discontinue operation when CPU enters Idle mode
0 = Continue operation in Idle mode

bit 12-6 **Unimplemented:** Read as '0'

bit 5 **OC32:** 32-bit Compare Mode bit
1 = OCxR<31:0> and/or OCxRS<31:0> are used for comparisons to the 32-bit timer source
0 = OCxR<15:0> and OCxRS<15:0> are used for comparisons to the 16-bit timer source

bit 4 **OCFLT:** PWM Fault Condition Status bit(1)
1 = PWM Fault condition has occurred (cleared in HW only)
0 = No PWM Fault condition has occurred

bit 3 **OCTSEL:** Output Compare Timer Select bit(2)
1 = Timery is the clock source for this Output Compare module
0 = Timerx is the clock source for this Output Compare module

bit 2-0 **OCM<2:0>:** Output Compare Mode Select bits
111 = PWM mode on OCx; Fault pin enabled
110 = PWM mode on OCx; Fault pin disabled
101 = Initialize OCx pin low; generate continuous output pulses on OCx pin
100 = Initialize OCx pin low; generate single output pulse on OCx pin
011 = Compare event toggles OCx pin
010 = Initialize OCx pin high; compare event forces OCx pin low
001 = Initialize OCx pin low; compare event forces OCx pin high
000 = Output compare peripheral is disabled but continues to draw current

Note 1: This bit is only used when OCM<2:0> = '111'. It is read as '0' in all other modes.
2: Refer to Table 18-1 for Timerx and Timery selections.

19.0 SERIAL PERIPHERAL INTERFACE (SPI) AND INTER-IC SOUND (I^2S)

Note:	This data sheet summarizes the features of the PIC32MZ Embedded Connectivity (EC) Family of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **Section 23. "Serial Peripheral Interface (SPI)"** (DS60001106), which is available from the *Documentation* > *Reference Manual* section of the Microchip PIC32 web site (www.microchip.com/pic32).

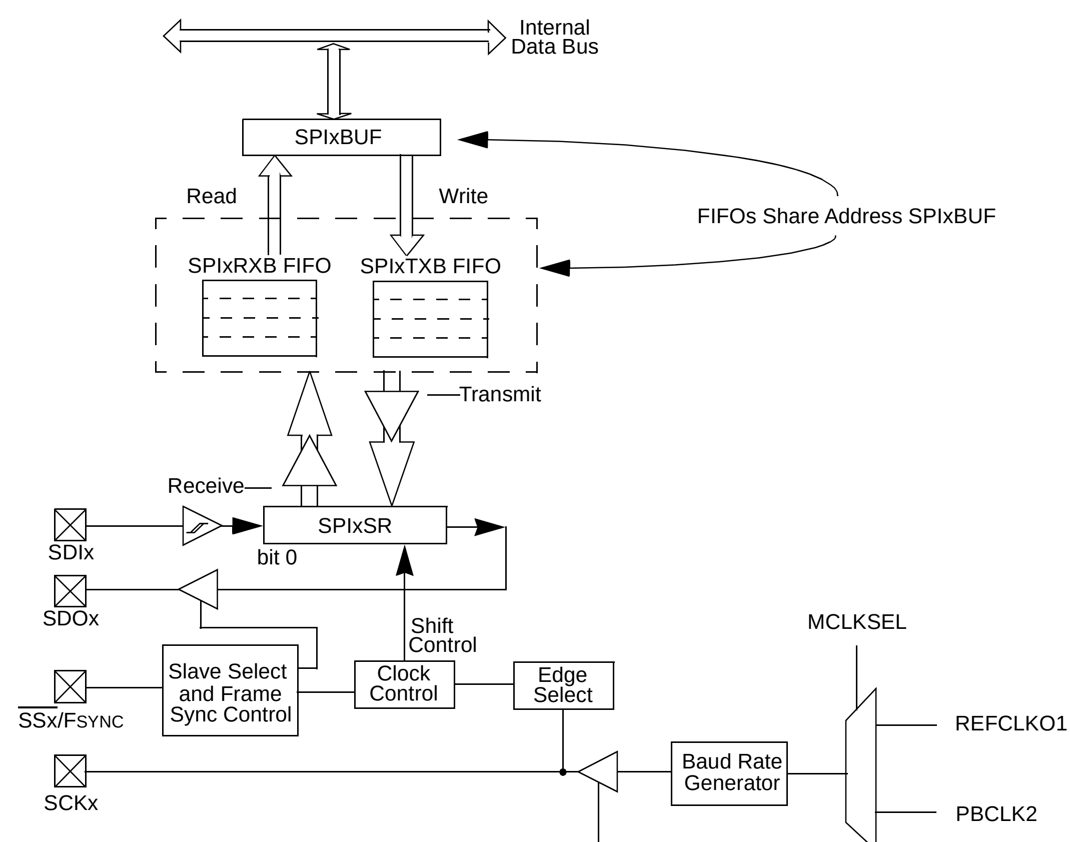
The SPI/I^2S module is a synchronous serial interface that is useful for communicating with external peripherals and other microcontroller devices, as well as digital audio devices. These peripheral devices may be Serial EEPROMs, Shift registers, display drivers, Analog-to-Digital Converters (ADC), etc.

The SPI/I^2S module is compatible with Motorola® SPI and SIOP interfaces.

The following are some of the key features of the SPI module:

- Master and Slave modes support
- Four different clock formats
- Enhanced Framed SPI protocol support
- User-configurable 8-bit, 16-bit and 32-bit data width
- Separate SPI FIFO buffers for receive and transmit
  - FIFO buffers act as 4/8/16-level deep FIFOs based on 32/16/8-bit data width
- Programmable interrupt event on every 8-bit, 16-bit and 32-bit data transfer
- Operation during Sleep and Idle modes
- Audio Codec Support:
  - I^2S protocol
  - Left-justified
  - Right-justified
  - PCM

FIGURE 19-1: SPI/I^2S MODULE BLOCK DIAGRAM



Note: Access SPIxTXB and SPIxRXB FIFOs via SPIxBUF register.

REGISTER 30-11: ETHRXFC: ETHERNET CONTROLLER RECEIVE FILTER CONFIGURATION REGISTER

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	HTEN	MPEN	—	NOTPM	PMMODE<3:0>			
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	CRCERREN	CRCOKEN	RUNTERREN	RUNTEN	UCEN	NOTMEEN	MCEN	BCEN

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 31-16 **Unimplemented:** Read as ‘0’

bit 15 **HTEN:** Enable Hash Table Filtering bit
1 = Enable Hash Table Filtering
0 = Disable Hash Table Filtering

bit 14 **MPEN:** Magic Packet™ Enable bit
1 = Enable Magic Packet Filtering
0 = Disable Magic Packet Filtering

bit 13 **Unimplemented:** Read as ‘0’

bit 12 **NOTPM:** Pattern Match Inversion bit
1 = The Pattern Match Checksum must not match for a successful Pattern Match to occur
0 = The Pattern Match Checksum must match for a successful Pattern Match to occur
This bit determines whether Pattern Match Checksum must match in order for a successful Pattern Match to occur.

bit 11-8 **PMMODE<3:0>:** Pattern Match Mode bits
1001 = Pattern match is successful if (NOTPM = 1 XOR Pattern Match Checksum matches) AND (Packet = Magic Packet)(1,3)
1000 = Pattern match is successful if (NOTPM = 1 XOR Pattern Match Checksum matches) AND (Hash Table Filter match)(1,2)
0111 = Pattern match is successful if (NOTPM = 1 XOR Pattern Match Checksum matches) AND (Destination Address = Broadcast Address)(1)
0110 = Pattern match is successful if (NOTPM = 1 XOR Pattern Match Checksum matches) AND (Destination Address = Broadcast Address)(1)
0101 = Pattern match is successful if (NOTPM = 1 XOR Pattern Match Checksum matches) AND (Destination Address = Unicast Address)(1)
0100 = Pattern match is successful if (NOTPM = 1 XOR Pattern Match Checksum matches) AND (Destination Address = Unicast Address)(1)
0011 = Pattern match is successful if (NOTPM = 1 XOR Pattern Match Checksum matches) AND (Destination Address = Station Address)(1)
0010 = Pattern match is successful if (NOTPM = 1 XOR Pattern Match Checksum matches) AND (Destination Address = Station Address)(1)
0001 = Pattern match is successful if (NOTPM = 1 XOR Pattern Match Checksum matches)(1)
0000 = Pattern Match is disabled; pattern match is always unsuccessful

Note 1: XOR = True when either one or the other conditions are true, but not both.
2: This Hash Table Filter match is active regardless of the value of the HTEN bit.
3: This Magic Packet Filter match is active regardless of the value of the MPEN bit.

Note 1: This register is only used for RX operations.
2: The bits in this register may only be changed while the RXEN bit (ETHCON1<8>) = 0.

REGISTER 30-28: EMAC1MAXF: ETHERNET CONTROLLER MAC MAXIMUM FRAME LENGTH REGISTER

Bit Range	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-1
	MACMAXF<15:8>(1)							
7:0	R/W-1	R/W-1	R/W-1	R/W-0	R/W-1	R/W-1	R/W-1	R/W-0
	MACMAXF<7:0>(1)							

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 31-16 **Unimplemented:** Read as ‘0’

bit 15-0 **MACMAXF<15:0>:** Maximum Frame Length bits(1)

These bits reset to 0x05EE, which represents a maximum receive frame of 1518 octets. An untagged maximum size Ethernet frame is 1518 octets. A tagged frame adds four octets for a total of 1522 octets. If a shorter/longer maximum length restriction is desired, program this 16-bit field.

Note 1: If a proprietary header is allowed, this bit should be adjusted accordingly. For example, if 4-byte headers are prepended to frames, MACMAXF could be set to 1527 octets. This would allow the maximum VLAN tagged frame plus the 4-byte header.

Note: Both 16-bit and 32-bit accesses are allowed to these registers (including the SET, CLR and INV registers). 8-bit accesses are not allowed and are ignored by the hardware.

REGISTER 34-4: DEVCFG1/ADEVCFG1: DEVICE CONFIGURATION WORD 1 (CONTINUED)

bit 2-0 **FNOSC<2:0>:** Oscillator Selection bits

111 = FRC divided by FRCDIV<2:0> bits (FRCDIV)
110 = Reserved
101 = LPRC
100 = SOSC
011 = Reserved
010 = POSC (HS, EC)
001 = SPLL
000 = FRC divided by FRCDIV<2:0> bits (FRCDIV)

36.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers (MCU) and dsPIC® digital signal controllers (DSC) are supported with a full range of software and hardware development tools:

- Integrated Development Environment
  - MPLAB® X IDE Software
- Compilers/Assemblers/Linkers
  - MPLAB XC Compiler
  - MPASM™ Assembler
  - MPLINK™ Object Linker/ MPLIB™ Object Librarian
  - MPLAB Assembler/Linker/Librarian for Various Device Families
- Simulators
  - MPLAB X SIM Software Simulator
- Emulators
  - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debuggers/Programmers
  - MPLAB ICD 3
  - PICkit™ 3
- Device Programmers
  - MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards, Evaluation Kits and Starter Kits
- Third-party development tools

36.1 MPLAB X Integrated Development Environment Software

The MPLAB X IDE is a single, unified graphical user interface for Microchip and third-party software, and hardware development tool that runs on Windows®, Linux and Mac OS® X. Based on the NetBeans IDE, MPLAB X IDE is an entirely new IDE with a host of free software components and plug-ins for high-performance application development and debugging. Moving between tools and upgrading from software simulators to hardware debugging and programming tools is simple with the seamless user interface.

With complete project management, visual call graphs, a configurable watch window and a feature-rich editor that includes code completion and context menus, MPLAB X IDE is flexible and friendly enough for new users. With the ability to support multiple tools on multiple projects with simultaneous debugging, MPLAB X IDE is also suitable for the needs of experienced users.

Feature-Rich Editor:

- Color syntax highlighting
- Smart code completion makes suggestions and provides hints as you type
- Automatic code formatting based on user-defined rules
- Live parsing

User-Friendly, Customizable Interface:

- Fully customizable interface: toolbars, toolbar buttons, windows, window placement, etc.
- Call graph window

Project-Based Workspaces:

- Multiple projects
- Multiple tools
- Multiple configurations
- Simultaneous debugging sessions

File History and Bug Tracking:

- Local file history feature
- Built-in support for Bugzilla issue tracker

FIGURE 38-5: VOH – 12x DRIVER PINS

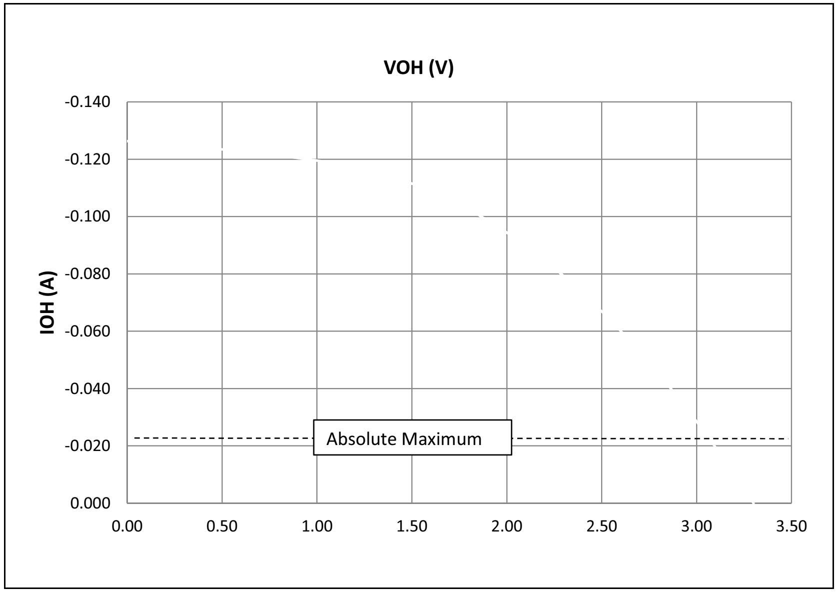


FIGURE 38-6: VOL – 12x DRIVER PINS

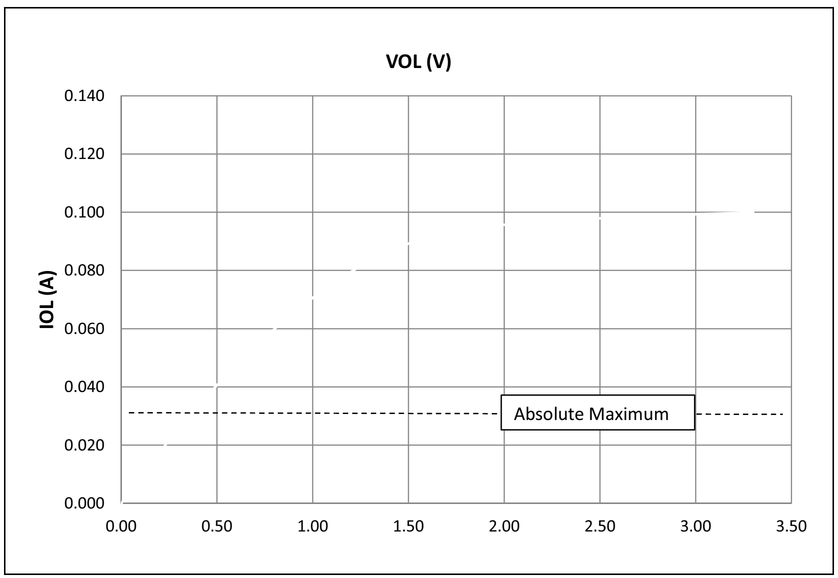
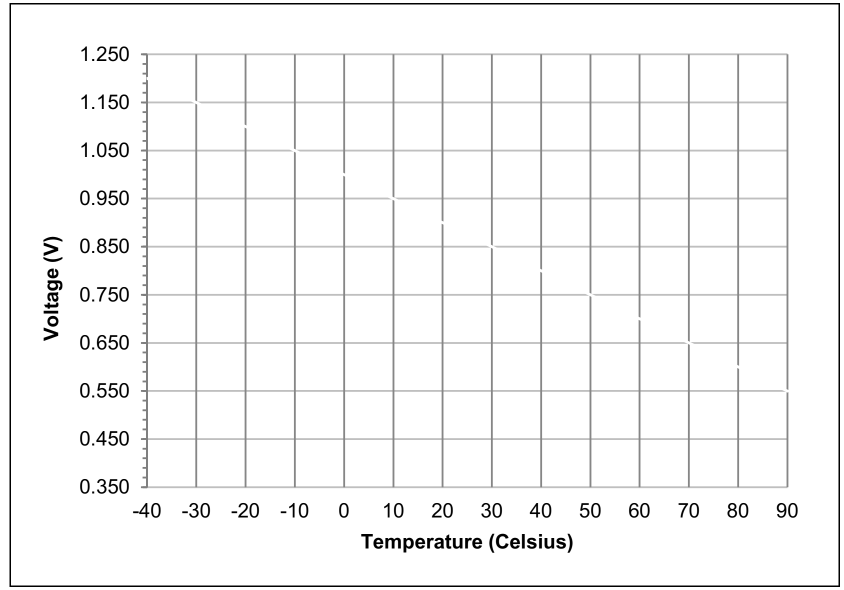


FIGURE 38-7: TYPICAL TEMPERATURE SENSOR VOLTAGE



A.7 Interrupts and Exceptions

The key difference between Interrupt Controllers in PIC32MX5XX/6XX/7XX devices and PIC32MZ devices concerns vector spacing. Previous PIC32MX devices had fixed vector spacing, which is adjustable in set increments, and every interrupt had the same amount of space. PIC32MZ devices replace this with a variable offset spacing, where each interrupt has an offset register to determine where to begin execution.

In addition, the IFSx, IECx, and IPCx registers for old peripherals have shifted to different registers due to new peripherals. Please refer to **Section 7.0 “CPU Exceptions and Interrupt Controller”** to determine where the interrupts are now located.

Table A-8 lists differences (indicated by **Bold** type) in the registers that will affect software migration.

TABLE A-8: INTERRUPT DIFFERENCES

PIC32MX5XX/6XX/7XX Feature	PIC32MZ Feature
Vector Spacing	
On PIC32MX devices, the vector spacing was determined by the VS field in the CPU core. **VS<4:0> (IntCtl<9:5>: CP0 Register 12, Select 1)** **10000 = 512-byte vector spacing** **01000 = 256-byte vector spacing** **00100 = 128-byte vector spacing** **00010 = 64-byte vector spacing** **00001 = 32-byte vector spacing** **00000 = 0-byte vector spacing**	On PIC32MZ devices, the vector spacing is variable and determined by the Interrupt controller. The VOFFx<17:1> bits in the OFFx register are set to the offset from EBASE where the interrupt service routine is located. **VOFFx<17:1> (OFFx<17:1>)** **Interrupt Vector ‘x’ Address Offset bits**
Shadow Register Sets	
On PIC32MX devices, there was one shadow register set which could be used during interrupt processing. Which interrupt priority could use the shadow register set was determined by the FSRSSEL field in DEVCFG3 and SS0 on INTCON. **FSRSSEL<2:0> (DEVCFG3<18:16>)** **111 = Assign Interrupt Priority 7 to a shadow register set** **110 = Assign Interrupt Priority 6 to a shadow register set** • • • **001 = Assign Interrupt Priority 1 to a shadow register set** **000 = All interrupt priorities are assigned to a shadow register set** SS0 (**INTCON<16>**) 1 = Single vector is presented with a shadow register set 0 = Single vector is not presented with a shadow register set	On PIC32MZ devices, there are seven shadow register sets, and each priority level can be assigned a shadow register set to use via the PRIxSS<3:0> bits in the PRISS register. The SS0 bit is also moved to PRISS<0>. **PRIxSS<3:0> PRISS<y:z>** **1xxx = Reserved (by default, an interrupt with a priority level of x uses Shadow Set 0)** **0111 = Interrupt with a priority level of x uses Shadow Set 7** **0110 = Interrupt with a priority level of x uses Shadow Set 6** • • • **0001 = Interrupt with a priority level of x uses Shadow Set 1** **0000 = Interrupt with a priority level of x uses Shadow Set 0** SS0 (**PRISS<0>**) 1 = Single vector is presented with a shadow register set 0 = Single vector is not presented with a shadow register set
Status	
PIC32MX devices, the VEC<5:0> bits show which interrupt is being serviced. **VEC<5:0> (INTSTAT<5:0>)** 11111-00000 = The interrupt vector that is presented to the CPU	On PIC32MZ devices, the SIRQ<7:0> bits show the IRQ number of the interrupt last serviced. **SIRQ<7:0> (INTSTAT<7:0>)** 11111111-00000000 = The last interrupt request number serviced by the CPU

THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://microchip.com/support