Section 13. Parallel Master Port (PMP)

HIGHLIGHTS

This section of the manual contains the following topics:

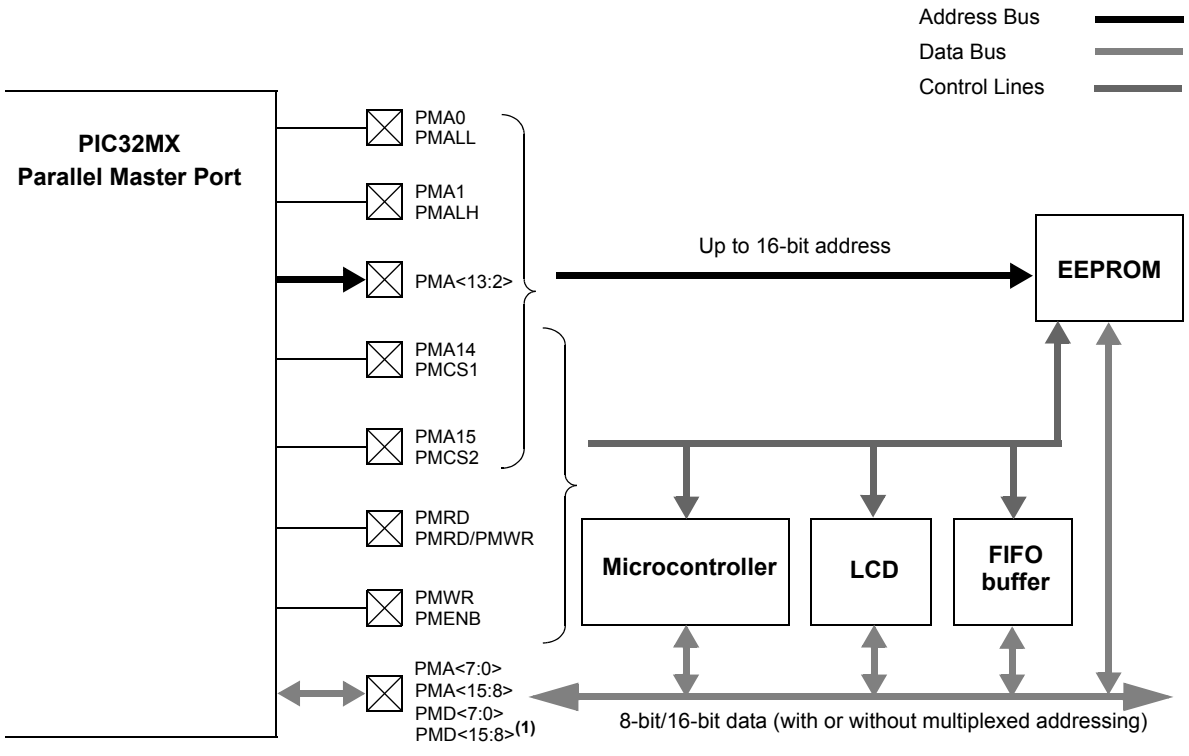
13.1 INTRODUCTION

The Parallel Master Port (PMP) is a parallel 8-bit/16-bit I/O module specifically designed to communicate with a wide variety of parallel devices such as communications peripherals, LCDs, external memory devices and microcontrollers. Because the interfaces to parallel peripherals vary significantly, the PMP module is highly configurable.

Key features of the PMP module include:

- Up to 16 programmable address lines
- Up to two Chip Select lines
- Programmable strobe options
  - Individual read and write strobes, or
  - Read/write strobe with enable strobe
- Address auto-increment/auto-decrement
- Programmable address/data multiplexing
- Programmable polarity on control signals
- Legacy parallel slave port support
- Enhanced parallel slave support
  - Address support
  - 4-bytes-deep, auto-incrementing buffer
- Schmitt Trigger or TTL input buffers
- Programmable Wait states
- Freeze option for in-circuit debugging

Figure 13-1: PMP Module Pinout and Connections to External Devices



Note 1: Data pins PMD<15:8> are available only on 100-pin PIC32MX device variants and larger.

13.2 CONTROL REGISTERS

The PMP module uses these Special Function Registers (SFRs):

- **PMCON: Parallel Port Control Register**

  This register (Register 13-1) contains the bits that control much of the module's basic functionality. A key bit is the ON control bit, which is used to Reset, enable or disable the module.

  When the module is disabled, all of the associated I/O pins revert to their designated I/O function. In addition, any read or write operations active or pending are stopped, and the BUSY bit is cleared. The data within the module registers is retained, including the data in PMSTAT register. Therefore, the module could be disabled after a reception, and the last received data and status would still be available for processing.

  When the module is enabled, all buffer control logic is reset, along with PMSTAT.

  All other bits in PMCON control address multiplexing, enable various port control signals, and select control signal polarity. These are discussed in more detail in **13.3.1 "Parallel Master Port Configuration Options"**.

- **PMMODE: Parallel Port Mode Register**

  This register (Register 13-2) contains bits that control the operational modes of the module. Master/Slave mode selection, as well as configuration options for both modes, are set by this register. It also contains the universal status flag BUSY, used in Master modes to indicate that an operation by the module in progress.

  Details on the use of the PMMODE bits to configure PMP operation are provided in **13.4 "Slave Modes of Operation"** and **13.3 "Master Modes of Operation"**.

- **PMADDR: Parallel Port Address Register**

  This register (Register 13-3) functions as PMADDR in master modes. It contains the address to which outgoing data is to be written, as well as the Chip Select control bits for addressing parallel slave devices. The PMADDR register is not used in any of the Slave modes.

- **PMDOUT: Parallel Port Data Output Register**

  This register (Register 13-4) is only used in Slave mode for buffered output data.

- **PMDIN: Parallel Port Data Input Register**

  This register (Register 13-5) is used by the PMP module in both Master and Slave modes.

  In Slave mode, this register is used to hold data that is asynchronously clocked in. Its operation is described in **13.4.2 "Buffered Parallel Slave Port Mode"**.

  In Master mode, PMDIN is the holding register for both incoming and outgoing data. Its operation in Master mode is described in **13.3.3 "Read Operation"** and **13.3.4 "Write Operation"**.

- **PMAEN: Parallel Port Pin Enable Register**

  This register (Register 13-6) controls the operation of address and Chip Select pins associated to this module. Setting these bits allocates the corresponding microcontroller pins to the PMP module; clearing the bits allocates the pins to port I/O or other peripheral modules associated with the pin.

- **PMSTAT: Parallel Port Status Register (Slave modes only)**

  This register (Register 13-7) contains Status bits associated with buffered operating modes when the port is functioning as a Slave port. This includes overflow, underflow and full flag bit.

  These flags are discussed in detail in **13.4.2 "Buffered Parallel Slave Port Mode"**.

13.2.1 PMP SFR Summary

Table 13-1 provides a brief summary of all PMP-module-related registers. Corresponding registers appear after the summary with a detailed description of each bit.

Table 13-1: PMP SFR Summary

Address Offset	Name		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
	PMCON(1,2,3)	31:24	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		15:8	ON	FRZ	SIDL	ADRMUX<1:0>		PMPTTL	PTWREN	PTRDEN
		7:0	CSF<1:0>		ALP	CS2P	CS1P	—	WRSP	RDSP
	PMMODE(1,2,3)	31:24	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		15:8	BUSY	IRQM<1:0>		INCM<1:0>		MODE16	MODE<1:0>	
		7:0	WAITB<1:0>		WAITM<3:0>				WAITE<1:0>	
	PMADDR(1,2,3)	31:24	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		15:8	CS2/A15	CS1/A14	ADDR<13:8>					
		7:0	ADDR<7:0>							
	PMDOUT(1,2,3)	31:24	DATAOUT<31:24>							
		23:16	DATAOUT<23:16>							
		15:8	DATAOUT<15:8>							
		7:0	DATAOUT<7:0>							
	PMDIN(1,2,3)	31:24	DATAIN<31:24>							
		23:16	DATAIN<23:16>							
		15:8	DATAIN<15:8>							
		7:0	DATAIN<7:0>							
	PMAEN(1,2,3)	31:24	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		15:8	PTEN<15:8>							
		7:0	PTEN<7:0>							
	PMSTAT	31:24	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		15:8	IBF	IBOV	—	—	IB3F	IB2F	IB1F	IB0F
		7:0	OBE	OBUF	—	—	OB3E	OB2E	OB1E	OB0E

Legend: — = unimplemented, read as '0'. Address offset values are shown in hexadecimal.

Note 1: This register has an associated Clear register at an offset of 0x4 bytes. These registers have the same name with CLR appended to the end of the register name (e.g., PMCONCLR). Writing a '1' to any bit position in the Clear register will clear valid bits in the associated register. Reads from the Clear register should be ignored.

2: This register has an associated Set register at an offset of 0x8 bytes. These registers have the same name with SET appended to the end of the register name (e.g., PMCONSET). Writing a '1' to any bit position in the Set register will set valid bits in the associated register. Reads from the Set register should be ignored.

3: This register has an associated Invert register at an offset of 0xC bytes. These registers have the same name with INV appended to the end of the register name (e.g., PMCONINV). Writing a '1' to any bit position in the Invert register will invert valid bits in the associated register. Reads from the Invert register should be ignored.

Register 13-1: PMCON: Parallel Port Control Register(1,2,3)

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 31							bit 24

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 23							bit 16

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ON	FRZ	SIDL	ADRMUX<1:0>		PMPTTL	PTWREN	PTRDEN
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	r-x	R/W-0	R/W-0
CSF<1:0>(4)		ALP(4)	CS2P(4)	CS1P(4)	—	WRSP	RDSP
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	P = Programmable bit	r = Reserved bit
U = Unimplemented bit	-n = Bit Value at POR: (‘0’, ‘1’, x = Unknown)		

bit 31-16 **Reserved:** Write ‘0’; ignore read

bit 15 **ON:** Parallel Master Port Enable bit
1 = PMP enabled
0 = PMP disabled, no off-chip access performed

Note: When using 1:1 PBCLK divisor, the user’s software should not read/write the peripheral’s SFRs in the SYSCLK cycle immediately following the instruction that clears the module’s ON control bit.

bit 14 **FRZ:** Freeze in Debug Exception Mode bit
1 = Freeze operation when CPU is in Debug Exception mode
0 = Continue operation even when CPU is in Debug Exception mode

Note: FRZ is writable in Debug Exception mode only, it is forced to ‘0’ in normal mode.

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

bit 12-11 **ADRMUX<1:0>:** Address/Data Multiplexing Selection bits
11 = All 16 bits of address are multiplexed on PMD<15:0> pins
10 = All 16 bits of address are multiplexed on PMD<7:0> pins
01 = Lower 8 bits of address are multiplexed on PMD<7:0> pins, upper 8 bits are on PMA<15:8>
00 = Address and data appear on separate pins

Note 1: This register has an associated Clear register (PMCONCLR) at an offset of 0x4 bytes. Writing a ‘1’ to any bit position in the Clear register will clear valid bits in the associated register. Reads from the Clear register should be ignored.

2: This register has an associated Set register (PMCONSET) at an offset of 0x8 bytes. Writing a ‘1’ to any bit position in the Set register will set valid bits in the associated register. Reads from the Set register should be ignored.

3: This register has an associated Invert register (PMCONINV) at an offset of 0xC bytes. Writing a ‘1’ to any bit position in the Invert register will invert valid bits in the associated register. Reads from the Invert register should be ignored.

4: These bits have no effect when their corresponding pins are used as address lines.

Register 13-1: PMCON: Parallel Port Control Register(1,2,3) (Continued)

bit 10 **PMPTTL:** PMP Module TTL Input Buffer Select bit
1 = PMP module uses TTL input buffers
0 = PMP module uses Schmitt Trigger input buffer

bit 9 **PTWREN:** Write Enable Strobe Port Enable bit
1 = PMWR/PMENB port enabled
0 = PMWR/PMENB port disabled

bit 8 **PTRDEN:** Read/Write Strobe Port Enable bit
1 = PMRD/PMWR port enabled
0 = PMRD/PMWR port disabled

bit 7-6 **CSF<1:0>:** Chip Select Function bits(4)
11 = Reserved
10 = PMCS2 and PMCS1 function as Chip Select
01 = PMCS2 functions as Chip Select, PMCS1 functions as address bit 14
00 = PMCS2 and PMCS1 function as address bits 15 and 14

bit 5 **ALP:** Address Latch Polarity bit(4)
1 = Active-high (PMALL and PMALH)
0 = Active-low ($\overline{\text{PMALL}}$ and $\overline{\text{PMALH}}$)

bit 4 **CS2P:** Chip Select 1 Polarity bit(4)
1 = Active-high (PMCS2)
0 = Active-low ($\overline{\text{PMCS2}}$)

bit 3 **CS1P:** Chip Select 0 Polarity bit(4)
1 = Active-high (PMCS1)
0 = Active-low ($\overline{\text{PMCS1}}$)

bit 2 **Reserved:** Write '0'; ignore read

bit 1 **WRSP:** Write Strobe Polarity bit
For Slave Modes and Master mode 2 (PMMODE<9:8> = 00,01,10):
1 = Write strobe active-high (PMWR)
0 = Write strobe active-low ($\overline{\text{PMWR}}$)
For Master mode 1 (PMMODE<9:8> = 11):
1 = Enable strobe active-high (PMENB)
0 = Enable strobe active-low ($\overline{\text{PMENB}}$)

bit 0 **RDSP:** Read Strobe Polarity bit
For Slave modes and Master mode 2 (PMMODE<9:8> = 00,01,10):
1 = Read Strobe active-high (PMRD)
0 = Read Strobe active-low ($\overline{\text{PMRD}}$)
For Master mode 1 (PMMODE<9:8> = 11):
1 = Read/write strobe active-high (PMRD/$\overline{\text{PMWR}}$)
0 = Read/write strobe active-low ($\overline{\text{PMRD}}$/PMWR)

Note 1: This register has an associated Clear register (PMCONCLR) at an offset of 0x4 bytes. Writing a '1' to any bit position in the Clear register will clear valid bits in the associated register. Reads from the Clear register should be ignored.

2: This register has an associated Set register (PMCONSET) at an offset of 0x8 bytes. Writing a '1' to any bit position in the Set register will set valid bits in the associated register. Reads from the Set register should be ignored.

3: This register has an associated Invert register (PMCONINV) at an offset of 0xC bytes. Writing a '1' to any bit position in the Invert register will invert valid bits in the associated register. Reads from the Invert register should be ignored.

4: These bits have no effect when their corresponding pins are used as address lines.

Register 13-2: PMMODE: Parallel Port Mode Register(1,2,3)

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 31							bit 24

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 23							bit 16

R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BUSY	IRQM<1:0>		INCM<1:0>		MODE16	MODE<1:0>	
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
WAITB<1:0>(4)		WAITM<3:0>(4)				WAITE<1:0>(4)	
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	P = Programmable bit	r = Reserved bit
U = Unimplemented bit	-n = Bit Value at POR: (‘0’, ‘1’, x = Unknown)		

bit 31-16 **Reserved:** Write ‘0’; ignore read

bit 15 **BUSY:** Busy bit (Master mode only)
- 1 = Port is busy
- 0 = Port is not busy

bit 14-13 **IRQM<1:0>:** Interrupt Request Mode bits
- 11 = Reserved, do not use
- 10 = Interrupt generated when Read Buffer 3 is read or Write Buffer 3 is written (Buffered PSP mode) or on a read or write operation when PMA<1:0> =11 (Addressable Slave mode only)
- 01 = Interrupt generated at the end of the read/write cycle
- 00 = No Interrupt generated

Note 1: This register has an associated Clear register (PMMODECLR) at an offset of 0x4 bytes. Writing a ‘1’ to any bit position in the Clear register will clear valid bits in the associated register. Reads from the Clear register should be ignored.

2: This register has an associated Set register (PMMODESET) at an offset of 0x8 bytes. Writing a ‘1’ to any bit position in the Set register will set valid bits in the associated register. Reads from the Set register should be ignored.

3: This register has an associated Invert register (PMMODEINV) at an offset of 0xC bytes. Writing a ‘1’ to any bit position in the Invert register will invert valid bits in the associated register. Reads from the Invert register should be ignored.

4: Whenever WAITM<3:0> = 0000, WAITB and WAITE bits are ignored and forced to 1 TPBCLK cycle for a write operation; WAITB = 1 TPBCLK cycle, WAITE = 0 TPBCLK cycles for a read operation.

5: Address bit A15 and A14 are not subject to auto-increment/decrement if configured as Chip Select CS2 and CS1.

6: These pins are active when bit MODE16 = 1 (16-bit mode)

7: The PMPADDR register is always incremented/decremented by 1 regardless of the transfer data width.

Register 13-2: PMMODE: Parallel Port Mode Register(1,2,3) (Continued)

bit 12-11 **INCM<1:0>:** Increment Mode bits
11 = Slave mode read and write buffers auto-increment (PMMODE<1:0> = 00 only)
10 = Decrement ADDR<15:0> by 1 every read/write cycle(5,7)
01 = Increment ADDR<15:0> by 1 every read/write cycle(5,7)
00 = No increment or decrement of address

bit 10 **MODE16:** 8/16-bit Mode bit
1 = 16-bit mode: a read or write to the data register invokes a single 16-bit transfer
0 = 8-bit mode: a read or write to the data register invokes a single 8-bit transfer

bit 9-8 **MODE<1:0>:** Parallel Port Mode Select bits
11 = Master mode 1 (PMCSx, PMRD/PMWR, PMENB, PMA<x:0>, PMD<7:0> and PMD<8:15>(6))
10 = Master mode 2 (PMCSx, PMRD, PMWR, PMA<x:0>, PMD<7:0> and PMD<8:15>(6))
01 = Enhanced Slave mode, control signals (PMRD, PMWR, PMCS, PMD<7:0> and PMA<1:0>)
00 = Legacy Parallel Slave Port, control signals (PMRD, PMWR, PMCS and PMD<7:0>)

bit 7-6 **WAITB<1:0>:** Data Setup to Read/Write Strobe Wait States bits(4)
11 = Data wait of 4 TPB; multiplexed address phase of 4 TPB
10 = Data wait of 3 TPB; multiplexed address phase of 3 TPB
01 = Data wait of 2 TPB; multiplexed address phase of 2 TPB
00 = Data wait of 1 TPB; multiplexed address phase of 1 TPB (default)

bit 5-2 **WAITM<3:0>:** Data Read/Write Strobe Wait States bits(4)
1111 = Wait of 16 TPB
•
•
•
0001 = Wait of 2 TPB
0000 = Wait of 1 TPB (default)

bit 1-0 **WAITE<1:0>:** Data Hold After Read/Write Strobe Wait States bits(4)
11 = Wait of 4 TPB
10 = Wait of 3 TPB
01 = Wait of 2 TPB
00 = Wait of 1 TPB (default)

For Read operations:
11 = Wait of 3 TPB
10 = Wait of 2 TPB
01 = Wait of 1 TPB
00 = Wait of 0 TPB (default)

Note 1: This register has an associated Clear register (PMMODECLR) at an offset of 0x4 bytes. Writing a '1' to any bit position in the Clear register will clear valid bits in the associated register. Reads from the Clear register should be ignored.

2: This register has an associated Set register (PMMODESET) at an offset of 0x8 bytes. Writing a '1' to any bit position in the Set register will set valid bits in the associated register. Reads from the Set register should be ignored.

3: This register has an associated Invert register (PMMODEINV) at an offset of 0xC bytes. Writing a '1' to any bit position in the Invert register will invert valid bits in the associated register. Reads from the Invert register should be ignored.

4: Whenever WAITM<3:0> = 0000, WAITB and WAITE bits are ignored and forced to 1 TPBCLK cycle for a write operation; WAITB = 1 TPBCLK cycle, WAITE = 0 TPBCLK cycles for a read operation.

5: Address bit A15 and A14 are not subject to auto-increment/decrement if configured as Chip Select CS2 and CS1.

6: These pins are active when bit MODE16 = 1 (16-bit mode)

7: The PMPADDR register is always incremented/decremented by 1 regardless of the transfer data width.

Register 13-3: PMADDR: Parallel Port Address Register(1,2,3)

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 31							bit 24

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 23							bit 16

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CS2	CS1	ADDR<13:8>					
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
ADDR<7:0>							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	P = Programmable bit	r = Reserved bit
U = Unimplemented bit	-n = Bit Value at POR: (‘0’, ‘1’, x = Unknown)		

bit 31-16 **Reserved:** Write ‘0’; ignore read

bit 15 **CS2:** Chip Select 2 bit
1 = Chip Select 2 is active
0 = Chip Select 2 is inactive (pin functions as PMA<15>)

bit 14 **CS1:** Chip Select 1 bit
1 = Chip Select 1 is active
0 = Chip Select 1 is inactive (pin functions as PMA<14>)

bit 13-0 **ADDR<13:0>:** Destination Address bits

Note 1: This register has an associated Clear register (PMADDRCLR) at an offset of 0x4 bytes. Writing a ‘1’ to any bit position in the Clear register will clear valid bits in the associated register. Reads from the Clear register should be ignored.

2: This register has an associated Set register (PMADDRSET) at an offset of 0x8 bytes. Writing a ‘1’ to any bit position in the Set register will set valid bits in the associated register. Reads from the Set register should be ignored.

3: This register has an associated Invert register (PMADDRINV) at an offset of 0xC bytes. Writing a ‘1’ to any bit position in the Invert register will invert valid bits in the associated register. Reads from the Invert register should be ignored.

Register 13-4: PMDOUT: Parallel Port Data Output Register(1,2,3)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DATAOUT<31:24>							
bit 31							bit 24

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DATAOUT<23:16>							
bit 23							bit 16

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DATAOUT<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
DATAOUT<7:0>							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	P = Programmable bit	r = Reserved bit
U = Unimplemented bit	-n = Bit Value at POR: (‘0’, ‘1’, x = Unknown)		

bit 31-0 **DATAOUT<31:0>:** Output Data Port bits for 8-bit write operations in Slave mode

Note 1: This register has an associated Clear register (PMDOUTCLR) at an offset of 0x4 bytes. Writing a ‘1’ to any bit position in the Clear register will clear valid bits in the associated register. Reads from the Clear register should be ignored.

2: This register has an associated Set register (PMDOUTSET) at an offset of 0x8 bytes. Writing a ‘1’ to any bit position in the Set register will set valid bits in the associated register. Reads from the Set register should be ignored.

3: This register has an associated Invert register (PMDOUTINV) at an offset of 0xC bytes. Writing a ‘1’ to any bit position in the Invert register will invert valid bits in the associated register. Reads from the Invert register should be ignored.

Register 13-5: PMDIN: Parallel Port Data Input Register(1,2,3)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DATAIN<31:24>							
bit 31							bit 24

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DATAIN<23:16>							
bit 23							bit 16

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DATAIN<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
DATAIN<7:0>							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	P = Programmable bit	r = Reserved bit
U = Unimplemented bit	-n = Bit Value at POR: (‘0’, ‘1’, x = Unknown)		

bit 31-0 **DATAIN<31:0>:** Input/Output Data Port bits for 8-bit or 16-bit read/write operations in Master mode Input Data Port for 8-bit read operations in Slave mode.

Note 1: This register has an associated Clear register (PMDINCLR) at an offset of 0x4 bytes. Writing a ‘1’ to any bit position in the Clear register will clear valid bits in the associated register. Reads from the Clear register should be ignored.

2: This register has an associated Set register (PMDINSET) at an offset of 0x8 bytes. Writing a ‘1’ to any bit position in the Set register will set valid bits in the associated register. Reads from the Set register should be ignored.

3: This register has an associated Invert register (PMDININV) at an offset of 0xC bytes. Writing a ‘1’ to any bit position in the Invert register will invert valid bits in the associated register. Reads from the Invert register should be ignored.

Register 13-6: PMAEN: Parallel Port Pin Enable Register(1,2,3)

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 31							bit 24

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 23							bit 16

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTEN<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
PTEN<7:0>							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	P = Programmable bit	r = Reserved bit
U = Unimplemented bit	-n = Bit Value at POR: (‘0’, ‘1’, x = Unknown)		

bit 31-16 **Reserved:** Write ‘0’; ignore read

bit 15-14 **PTEN<15:14>:** PMCSx Strobe Enable bits
- 1 = PMA15 and PMA14 function as either PMA<15:14> or PMCS2 and PMCS1(4)
- 0 = PMA15 and PMA14 function as port I/O

bit 13-2 **PTEN<13:2>:** PMP Address Port Enable bits
- 1 = PMA<13:2> function as PMP address lines
- 0 = PMA<13:2> function as port I/O

bit 1-0 **PTEN<1:0>:** PMALH/PMALL Strobe Enable bits
- 1 = PMA1 and PMA0 function as either PMA<1:0> or PMALH and PMALL(5)
- 0 = PMA1 and PMA0 pads functions as port I/O

Note 1: This register has an associated Clear register (PMAENCLR) at an offset of 0x4 bytes. Writing a ‘1’ to any bit position in the Clear register will clear valid bits in the associated register. Reads from the Clear register should be ignored.

2: This register has an associated Set register (PMAENSET) at an offset of 0x8 bytes. Writing a ‘1’ to any bit position in the Set register will set valid bits in the associated register. Reads from the Set register should be ignored.

3: This register has an associated Invert register (PMAENINV) at an offset of 0xC bytes. Writing a ‘1’ to any bit position in the Invert register will invert valid bits in the associated register. Reads from the Invert register should be ignored.

4: The use of these pins as PMA15/PMA14 or CS2/CS1 is selected by bits CSF<1:0> in the PMCON register.

5: The use of these pins as PMA1/PMA0 or PMALH/PMALL depends on the Address/Data Multiplex mode selected by bits ADRMUX<1:0> in the PMCON register.

Register 13-7: PMSTAT: Parallel Port Status Register (Slave modes only)(1,2,3)

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 31							bit 24

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 23							bit 16

R-0	R/W-0	r-x	r-x	R-0	R-0	R-0	R-0
IBF	IBOV	—	—	IB3F	IB2F	IB1F	IB0F
bit 15							bit 8

R-1	R/W-0	r-x	r-x	R-1	R-1	R-1	R-1
OBE	OBUF	—	—	OB3E	OB2E	OB1E	OB0E
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	P = Programmable bit	r = Reserved bit
U = Unimplemented bit	-n = Bit Value at POR: (‘0’, ‘1’, x = Unknown)		

bit 31-16 **Reserved:** Write ‘0’; ignore read

bit 15 **IBF:** Input Buffer Full Status bit
1 = All writable input buffer registers are full
0 = Some or all of the writable input buffer registers are empty

bit 14 **IBOV:** Input Buffer Overflow Status bit
1 = A write attempt to a full input byte buffer occurred (must be cleared in software)
0 = No overflow occurred
This bit is set (= 1) in hardware; can only be cleared (= 0) in software.

bit 13-12 **Reserved:** Write ‘0’; ignore read

bit 11-8 **IBnF:** Input Buffer n Status Full bits
1 = Input Buffer contains data that has not been read (reading buffer will clear this bit)
0 = Input Buffer does not contain any unread data

bit 7 **OBE:** Output Buffer Empty Status bit
1 = All readable output buffer registers are empty
0 = Some or all of the readable output buffer registers are full

bit 6 **OBUF:** Output Buffer Underflow Status bit
1 = A read occurred from an empty output byte buffer (must be cleared in software)
0 = No underflow occurred
This bit is set (= 1) in hardware; can only be cleared (= 0) in software.

Note 1: This register has an associated Clear register (PMSTATCLR) at an offset of 0x4 bytes. Writing a ‘1’ to any bit position in the Clear register will clear valid bits in the associated register. Reads from the Clear register should be ignored.

2: This register has an associated Set register (PMSTATSET) at an offset of 0x8 bytes. Writing a ‘1’ to any bit position in the Set register will set valid bits in the associated register. Reads from the Set register should be ignored.

3: This register has an associated Invert register (PMSTATINV) at an offset of 0xC bytes. Writing a ‘1’ to any bit position in the Invert register will invert valid bits in the associated register. Reads from the Invert register should be ignored.

Register 13-7: PMSTAT: Parallel Port Status Register (Slave modes only)(1,2,3) (Continued)

bit 5-4 **Reserved:** Write '0'; ignore read

bit 3-0 **OBnE:** Output Buffer n Status Empty bits

1 = Output buffer is empty (writing data to the buffer will clear this bit)
0 = Output buffer contains data that has not been transmitted

Note 1: This register has an associated Clear register (PMSTATCLR) at an offset of 0x4 bytes. Writing a '1' to any bit position in the Clear register will clear valid bits in the associated register. Reads from the Clear register should be ignored.

2: This register has an associated Set register (PMSTATSET) at an offset of 0x8 bytes. Writing a '1' to any bit position in the Set register will set valid bits in the associated register. Reads from the Set register should be ignored.

3: This register has an associated Invert register (PMSTATINV) at an offset of 0xC bytes. Writing a '1' to any bit position in the Invert register will invert valid bits in the associated register. Reads from the Invert register should be ignored.

13.3 MASTER MODES OF OPERATION

In its master modes, the PMP module can provide a 16-bit or 8-bit data bus, up to 16 bits of address, and all the necessary control signals to operate a variety of external parallel devices such as memory devices, peripherals and slave microcontrollers. The PMP master modes provide a simple interface for reading and writing data, but not executing program instructions from external devices, such as SRAM or Flash memories.

Because there are a number of parallel devices with a variety of control methods, the PMP module is designed for flexibility to accommodate a range of configurations. Some of these features include:

- 8-bit and 16-bit data modes
- Configurable address/data multiplexing
- Up to two Chip Select lines
- Up to 16 selectable address lines
- Address auto-increment and auto-decrement
- Selectable polarity on all control lines
- Configurable Wait states at different stages of the read/write cycle

13.3.1 Parallel Master Port Configuration Options

13.3.1.1 8-BIT AND 16-BIT DATA MODES

The PMP in Master mode supports data widths 8 and 16 bits wide. By default, the data width is 8-bit, MODE16 bit (PMMODE<10>) = 0. To select a 16-bit data width, set MODE16 = 1. When configured in 8-bit Data mode, the upper 8 bits of the data bus, PMD<15:8>, are not controlled by the PMP module and are available as general purpose I/O pins.

Note: Data pins PMD<15:0> are available on 100-pin PIC32MX device variants. For 64-pin device variants, only pins PMD<7:0> are available. Refer to the specific PIC32MX device data sheet for details.

13.3.1.2 CHIP SELECTS

Two Chip Select lines, PMCS1 and PMCS2, are available for the master modes. The two Chip Select lines are multiplexed with the Most Significant bits (MSbs) of the address bus A14 and A15. When a pin is configured as a Chip Select, it is not included in any address auto-increment/decrement. It is possible to enable both PMCS2 and PMCS1 as Chip Selects, or enable only PMCS2 as a Chip Select, allowing PMCS1 to function strictly as address line A14. It is not possible to enable PMCS1 alone. The Chip Select signals are configured using the Chip Select Function bits CSF<1:0> (PMCON <7:6>).

Table 13-2: Chip Select Control

CSF<1:0>	Function
00	PMCS2 = A15, PMCS1 = A14
01	PMCS2 = Enabled, PMCS1 = A14
10	PMCS2, PMCS1 = Enabled

13.3.1.3 PORT PIN CONTROL

There are several bits available to configure the presence or absence of control and address signals in the module. These bits are PTWREN (PMCON<9>), PTRDEN (PMCON<8>) and PTEN<15:0> (PMAEN<15:0>). They provide the ability to conserve pins for other functions and allow flexibility to control the external address. When any one of these bits is set, the associated function is present on its associated pin; when clear, the associated pin reverts to its defined I/O port function.

Setting a PTEN bit will enable the associated pin as an address pin and drive the corresponding data contained in the PMADDR register. Clearing any PTEN bit will force the pin to revert to its original I/O function.

For the pins configured as Chip Select (PMCS1 or PMCS2) with the corresponding PTEN bit set, Chip Select pins drive inactive data when a read or write operation is not being performed. The PTEN0 and PTEN1 bits also control the PMALL and PMALH signals. When multiplexing is used, the associated address latch signals should be enabled. Refer to **13.10 "I/O Pin Control"** regarding I/O pin configuration.

13.3.1.4 READ/WRITE CONTROL

The PMP module supports two distinct read/write signaling methods. In Master mode 1, read and write strobe are combined into a single control line, PMRD/PMWR; a second control line, PMENB, determines when a read or write action is to be taken. In Master mode 2, read and write strobes (PMRD and PMWR) are supplied on separate pins.

13.3.1.5 CONTROL LINE POLARITY

All control signals (PMRD, PMWR, PMENB, PMALL, PMALH, PMCS2 and PMCS1) can be individually configured for either positive or negative polarity. Configuration is controlled by separate bits in the PMCON register, as shown in Table 13-3.

Table 13-3: PIN POLARITY CONFIGURATION

Control Pin	PMCON Control Bit	Active-High Select	Active-Low Select
PMRD	RDSP	1	0
PMWR	WRSP	1	0
PMCS2	CS2P	1	0
PMCS1	CS1P	1	0
PMALL	ALP	1	0
PMALH	ALP	1	0

Note that the polarity of control signals that share the same output pin (for example, PMWR and PMENB) are controlled by the same bit; the configuration depends on which master port mode is being used.

13.3.1.6 AUTO-INCREMENT/DECREMENT

While the PMP module is operating in one of the master modes, the INCM<1:0> (PMMODE<12:11>) bits control the behavior of the address value. The address in the PMADDR register can be made to automatically increment or decrement by 1, regardless of the transfer data width, after each read and write operation is completed, and the BUSY bit (PMMODE<15>) goes to '0'.

Table 13-4: ADDRESS INC/DEC CONTROL

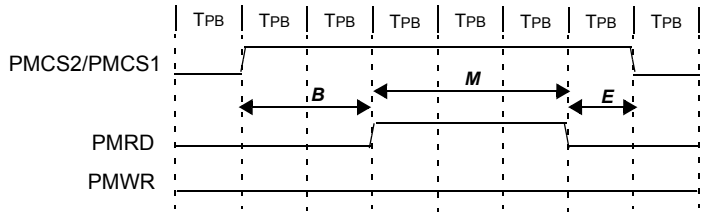
INCM<1:0>	Function
00	No Increment – No Decrement
01	Increment every R/W cycle
10	Decrement every R/W cycle

If the Chip Select signals are disabled and configured as address bits, the bits will participate in the increment and decrement operations; otherwise, CS2 and CS1 bit values will be unaffected.

13.3.1.7 WAIT STATES

In Master mode, the user can control the duration of the read, write and address cycles by configuring the module Wait states. One Wait state period is equivalent to one peripheral bus clock cycles, TPBCLK. Below is an example of a Master mode 2 Read operation using Wait states.

Figure 13-2: Read Operation, Wait States Enabled



Legend:
B = WAITB<1:0> = 01 (2 Wait states)
M = WAITM<3:0> = 0010 (3 Wait states)
E = WAITE<1:0> = 01 (1 Wait state)

Note: If WAITM<3:0> = 0000, M is forced to 1 TPBCLK, WAITB is ignored (B forced to 1 TPBCLK) and WAITE is ignored (E forced to 0 TPBCLK).

Wait states can be added to the beginning, middle and end of any read or write cycle using the corresponding WAITB, WAITM and WAITE bits in the PMMODE register.

The WAITB<1:0> (PMMODE<7:6>) bits define the number of wait cycles for the data setup prior to the PMRD/PMWR strobe in Mode 10, or prior to the PMENB strobe in Mode 11. When multiplexing the address and data bus, ADRMUX<1:0> = 01, 10 or 11, WAITB defines the number of wait cycles for which the addressing period is extended.

The WAITM<3:0> (PMMODE<5:2>) bits define the number of wait cycles for the PMRD/PMWR strobe in Mode 10, or for the PMENB strobe in Mode 11. When this Wait state setting is '0000', WAITB and WAITE are ignored. The number of Wait states for the data setup time (WAITB) defaults to one while the number of Wait states for data hold time (WAITE) defaults to one during a write operation and zero during a read operation.

The WAITE<1:0> (PMMODE<1:0>) bits define the number of wait cycles for the data hold time after the PMRD/PMWR strobe in Mode 10, or after the PMENB strobe in Mode 11.

13.3.1.8 ADDRESS MULTIPLEXING

Address multiplexing allows some or all address line signals to be generated from the data bus during the address cycle of a read/write operation. This can be a useful option for address lines PMA<15:0> needed as general purpose I/O pins. The user can select to multiplex the lower 8 data bits, upper 8 data bits or full 16 data bits. These multiplexing modes are available in both Master mode 1 and 2. Refer to **13.3.8 "Master Mode Timing"** for the multiplexing mode timing diagrams.

Table 13-5: Address Multiplex Configurations

ADRMUX<1:0>	Address/Data Multiplex Modes
00	Demultiplexed
01	Partially multiplexed (lower eight data pins PMD<7:0>)
10	Fully multiplexed (lower eight data pins PMD<7:0>)
11	Fully multiplexed (16 data pins PMD<15:0>)

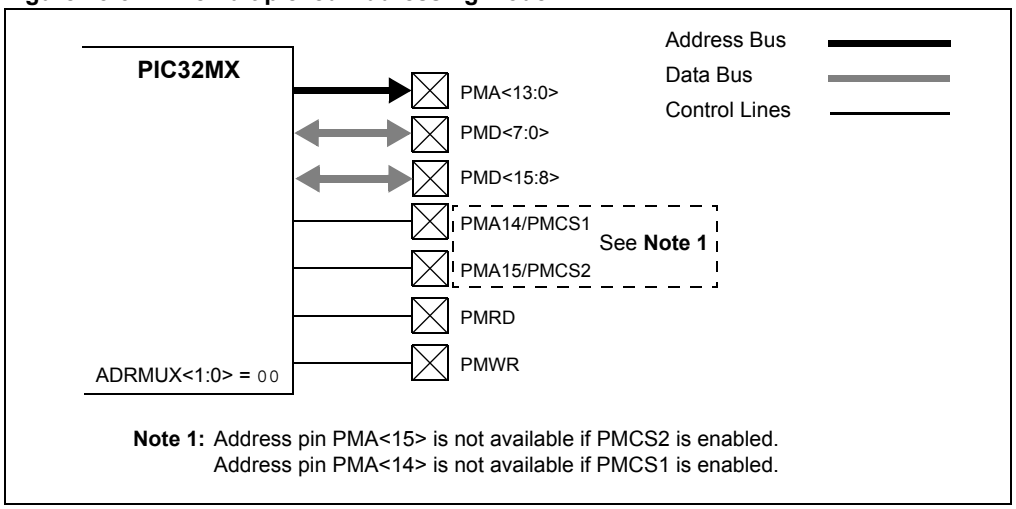
13.3.1.8.1 Demultiplexed Mode

Demultiplexed mode is selected by configuring bits ADRMUX<1:0> = 00, (PMMODE<9:8>). In this mode, address bits are presented on pins PMA<15:0>.

When PMCS2 is enabled, address pin PMA15 is not available. When PMCS1 is enabled, address pin PMA14 is not available.

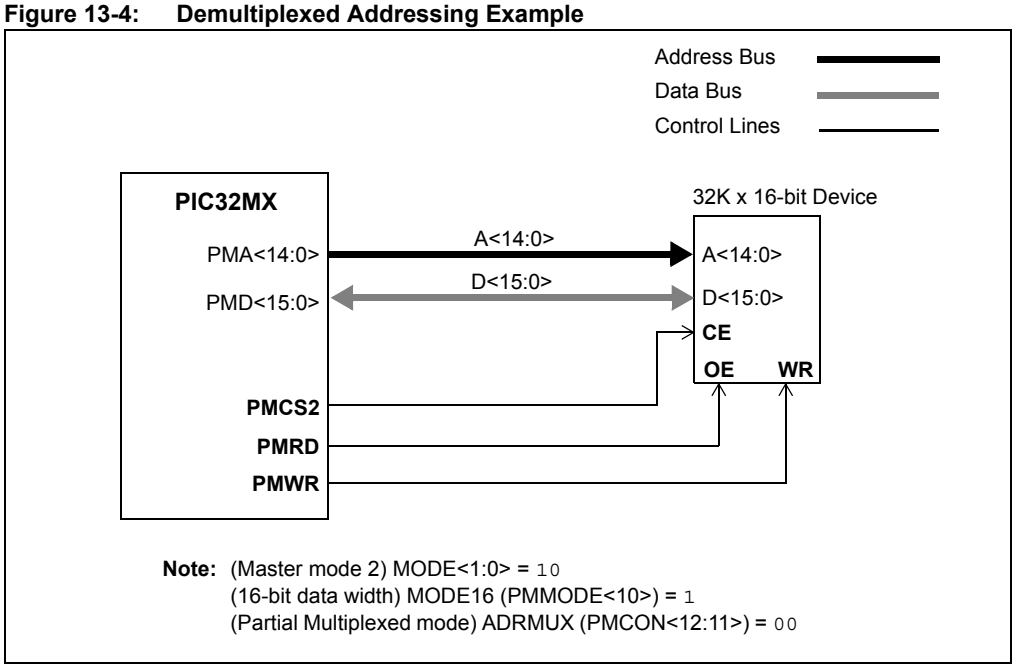
In 16-bit Data mode, data bits are presented on pins PMD<15:0>. In 8-bit Data mode, data bits are presented on pins PMD<7:0>.

Figure 13-3: Demultiplexed Addressing Mode



Note 1: Address pin PMA<15> is not available if PMCS2 is enabled.
Address pin PMA<14> is not available if PMCS1 is enabled.

Figure 13-4: Demultiplexed Addressing Example



Note: (Master mode 2) MODE<1:0> = 10
(16-bit data width) MODE16 (PMMODE<10>) = 1
(Partial Multiplexed mode) ADRMUX (PMCON<12:11>) = 00

13.3.1.8.2 Partially Multiplexed Mode

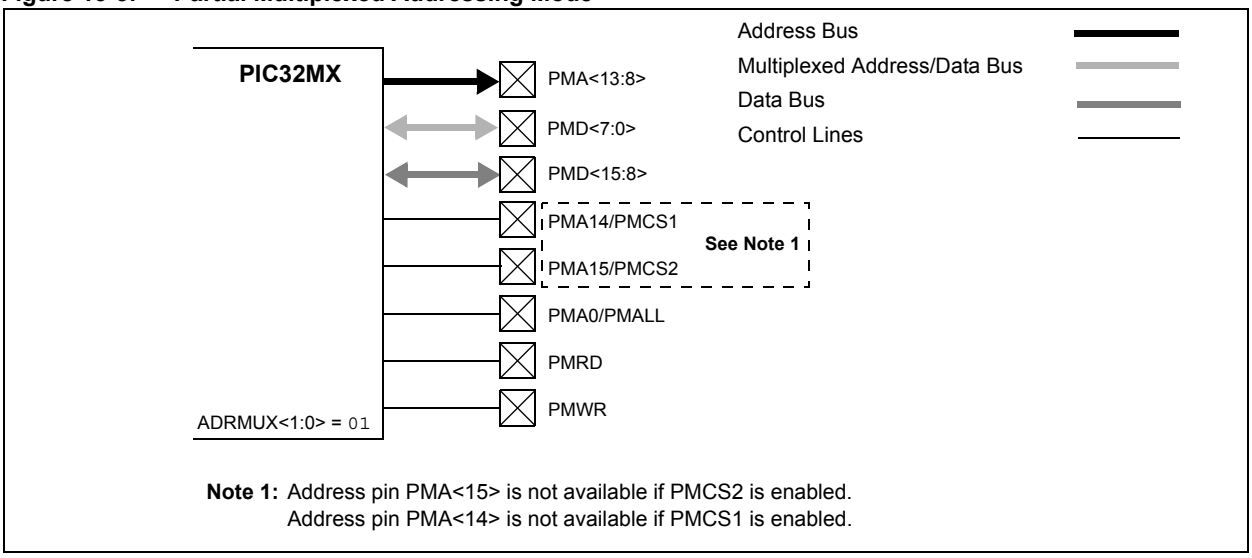
Partially Multiplexed mode (8-bit data pins) is available in both 8-bit and 16-bit data bus configurations and is selected by setting bits ADRMUX<1:0> = 01. In this mode, the lower eight address bits are multiplexed with the lower eight data bus pins, PMD<7:0>. The upper eight address bits are unaffected and are presented on PMA<15:8>. In this mode, address pins PMA<7:1> are available as general purpose I/O pins.

Address pin PMA15 is not available when PMCS2 is enabled; address pin PMA14 is not available when PMCS1 is enabled.

Address pin PMA<0> is used as an Address Latch enable strobe, PMALL, during which the lower eight bits of the address are presented on the PMD<7:0> pins. Read and write sequences are extended by at least three peripheral bus clock cycles (TPBCLK).

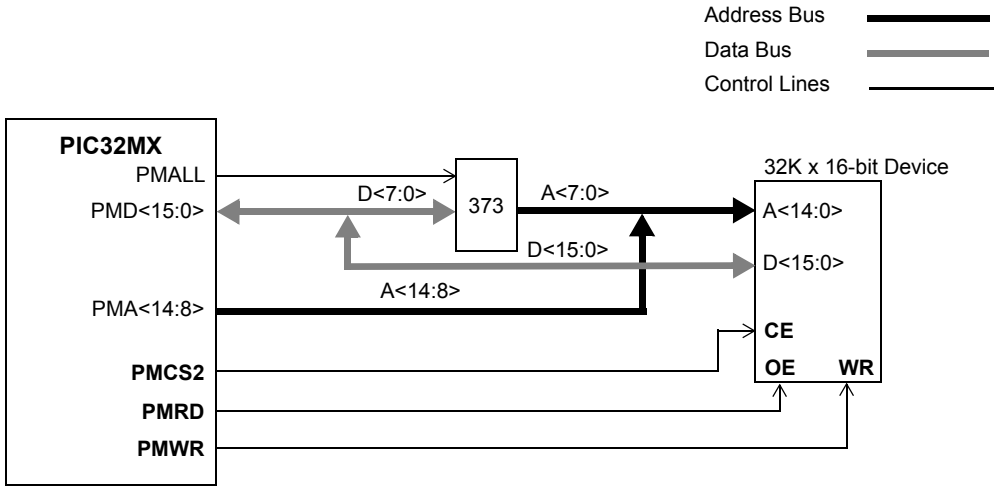
If WAITM<3:0> (PMMODE<5:2>) is non-zero, the PMALL strobe will be extended by WAITB<1:0> (PMMODE<7:6>) Wait states.

Figure 13-5: Partial Multiplexed Addressing Mode



Note 1: Address pin PMA<15> is not available if PMCS2 is enabled.
Address pin PMA<14> is not available if PMCS1 is enabled.

Figure 13-6: Partial Multiplexed Addressing Example



Note: (Master mode 2) MODE<1:0> = 10
(16-bit data width) MODE16 (PMMODE<10>) = 1
(Partial Multiplexed mode) ADRMUX (PMCON<12:11>) = 01
The 373 shown in the diagram represents a generic 74XX family 373 latch.

13.3.1.8.3 Fully Multiplexed Mode (8-bit Data Pins)

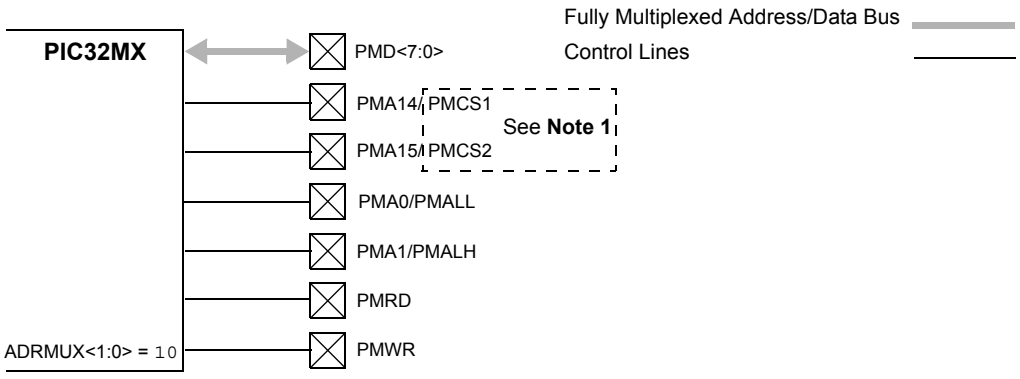
Fully multiplexed mode (8-bit data pins) is available in both 8-bit and 16-bit data bus configurations and is selected by setting the ADRMUX<1:0> bits (PMCON<12:11>) = 10. In this mode, the entire 16 bits of the address are multiplexed with the lower eight data bus pins, PMD<7:0>. In this mode, Pins PMA<13:2> available as general purpose I/O pins.

In the event the pins PMCS2/PMA15 or PMCS1/PMA14 are configured as Chip Select pins, the corresponding address bits PMADDR<15> or PMADDR<14> are automatically forced to '0'.

Address pins PMA<0> and PMA<1> are used as an Address Latch enable strobes, PMALL and PMALH, respectively. During the first cycle, the lower eight address bits are presented on the PMD<7:0> pins with the PMALL strobe active. During the second cycle the upper eight address bits are presented on the PMD<7:0> pins with the PMALH strobe active. The read and write sequences are extended by at least six peripheral bus clock cycles (TPBCLK).

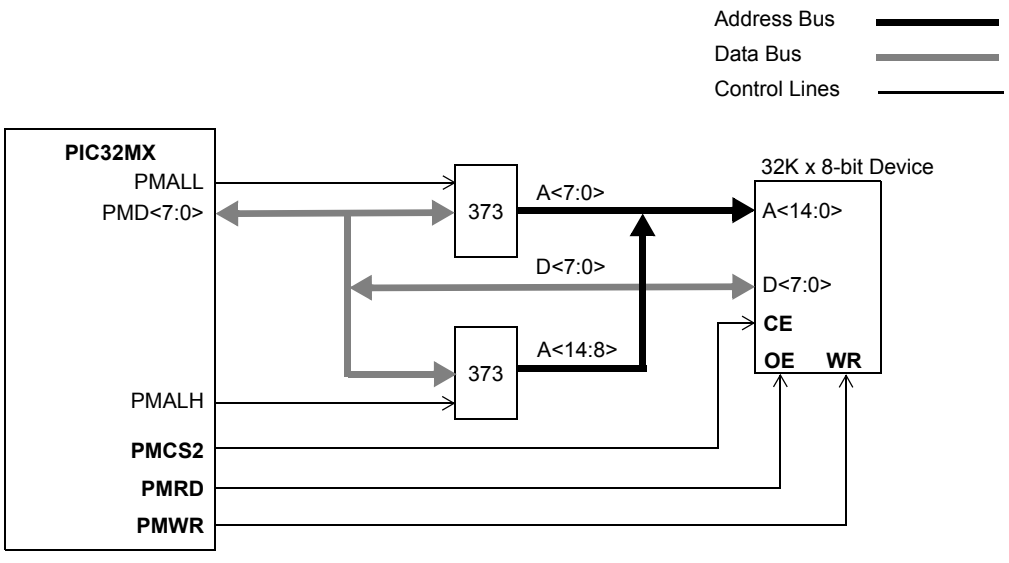
If WAITM<3:0> (PMMODE<5:2>) is non-zero, both PMALL and PMALH strobes will be extended by WAITB<1:0> (PMMODE<7:6>) Wait states.

Figure 13-7: Fully Multiplexed Addressing Mode (8-bit Bus)



Note 1: Address bit PMADDR<15> is forced = 0 when PMCS2 is enabled.
Address bit PMADDR<14> is forced = 0 when PMCS1 is enabled.

Figure 13-8: Fully Multiplexed Address Example (8-bit Bus)



Note: (Master mode 2) MODE<1:0> = 10
(8-bit data width) MODE16 (PMMODE<10>) = 0
(Fully Multiplexed mode) ADRMUX (PMCON<12:11>) = 10
The 373 shown in the diagram represents a generic 74XX family 373 latch.

13.3.1.8.4 Fully Multiplexed Mode (16-bit Data Pins)

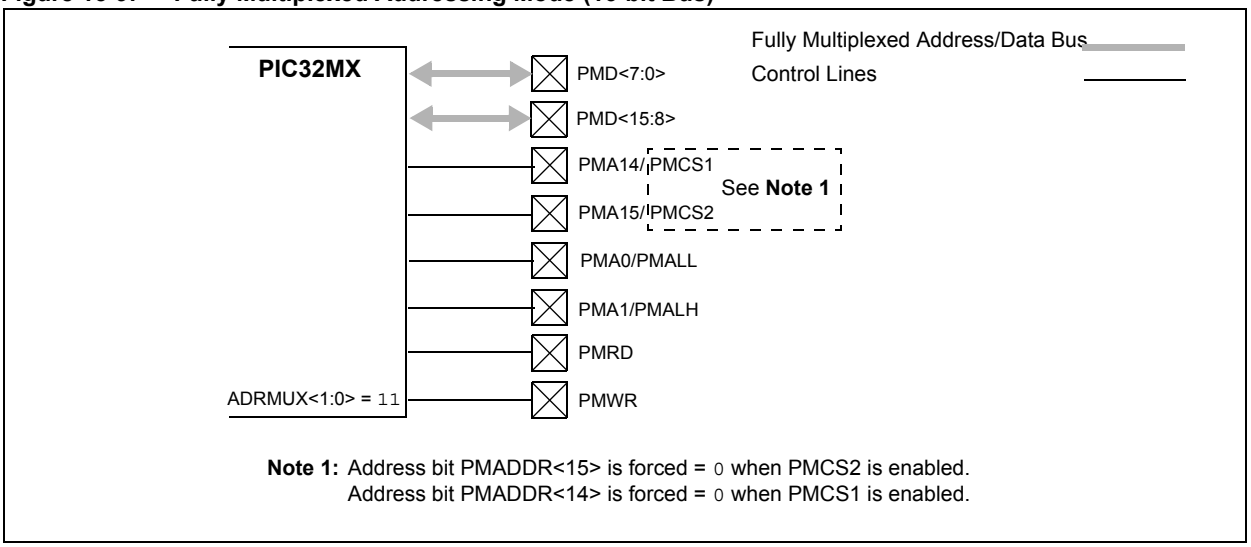
Fully Multiplexed mode (16-bit data pins) is only available in the 16-bit data bus configuration and is selected by configuring the ADRMUX<1:0> bits (PMCON<12:11>) = 11. In this mode, the entire 16 bits of the address are multiplexed with all 16 data bus pins, PMD<15:0>.

In the event the pins PMCS2/PMA15 or PMCS1/PMA14 are configured as Chip Select pins, the corresponding address bits PMADDR<15> or PMADDR<14> are automatically forced to '0'.

Address pins PMA<0> and PMA<1> are used as an Address Latch enable strobes, PMALL and PMALH respectively, and at the same time. While the PMALL and PMALH strobes are active, the lower eight address bits are presented on the PMD<7:0> pins and the upper eight address bits are presented on the PMD<15:8> pins. The read and write sequences are extended by at least 3 peripheral bus clock cycles (TPBCLK).

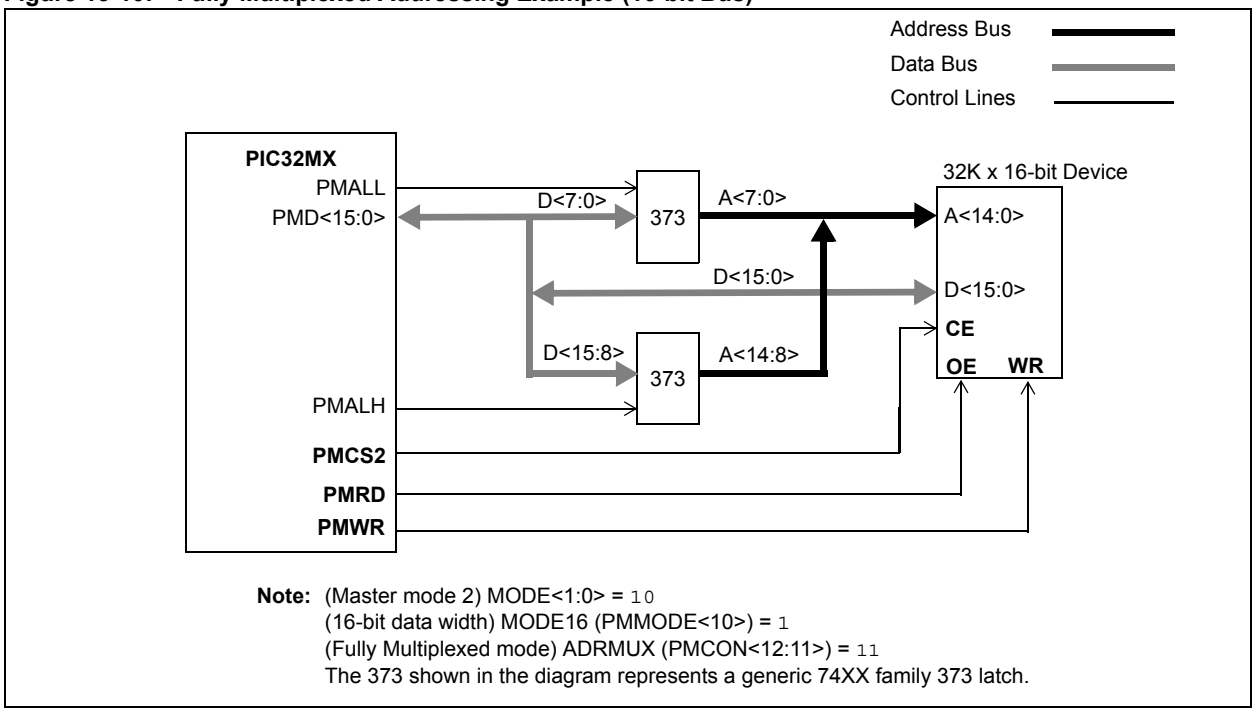
If WAITM<3:0> (PMMODE<5:2>) is non-zero, both PMALL and PMALH strobes will be extended by WAITB<1:0> (PMMODE<7:6>) Wait states.

Figure 13-9: Fully Multiplexed Addressing Mode (16-bit Bus)



Note 1: Address bit PMADDR<15> is forced = 0 when PMCS2 is enabled.
Address bit PMADDR<14> is forced = 0 when PMCS1 is enabled.

Figure 13-10: Fully Multiplexed Addressing Example (16-bit Bus)



Note: (Master mode 2) MODE<1:0> = 10
(16-bit data width) MODE16 (PMMODE<10>) = 1
(Fully Multiplexed mode) ADRMUX (PMCON<12:11>) = 11
The 373 shown in the diagram represents a generic 74XX family 373 latch.

13.3.2 Master Port Configuration

The Master mode configuration is determined primarily by the interface requirements to the external device. Address multiplexing, control signal polarity, data width and Wait states typically dictate the specific configuration of the PMP master port.

To use the PMP as a master, the module must be enabled by setting the ON control bit (PMCON<15>) = 1, and the mode must be set to one of two possible master modes. Control bits MODE<1:0> (PMMODE<9:8>) = 10 for Master mode 2 or MODE<1:0> = 11 for Master mode 1.

The following Master mode initialization properly prepares the PMP port for communicating with an external device.

1. If interrupts are used, disable the PMP interrupt by clearing the interrupt enable bit PMPIE (IEC1<2>) = 0.
2. Stop and reset the PMP module by clearing the control bit ON (PMCON<15>) = 0.
3. Configure the desired settings in the PMCON, PMMODE and PMAEN control registers.
4. If interrupts are used:
   a) Clear interrupt flag bit PMPIF (IFS1<2>) = 0.
   b) Configure the PMP interrupt priority bits PMPIP<2:0> (IPC7<4:2>) and interrupt subpriority bits PMPIS (IPC7<1:0>).
   c) Enable PMP interrupt by setting interrupt enable bit PMPIE = 1.
5. Enable the PMP master port by setting the ON control bit = 1.

Note:	It is recommended to wait for any pending read or write operation to be completed before reconfiguring the PMP module.

The following illustrates an example setup for a typical Master mode 2 operation:

1. Select Master mode 2 – MODE<1:0> (PMMODE<9:8>) = 10.
2. Select 16-bit Data mode – MODE16 (PMMODE<10>) = 0.
3. Select partially multiplexed addressing – ADRMUX<1:0> (PMCON<12:11>) = 01.
4. Select auto address increment – INCM<1:0> (PMMODE<12:11>) = 01.
5. Enable Interrupt Request mode – IRQM<1:0> (PMMODE<14:13>) = 01.
6. Enable PMRD strobe – PTRDEN (PMCON<8>) = 1.
7. Enable PMWR strobe – PTWREN (PMCON<9>) = 1.
8. Enable PMCS2 and PMCS1 Chip Selects – CSF (PMCON<7:6>) = 10.
9. Select PMRD active-low pin polarity – RDSP (PMCON<0>) = 0.
10. Select PMWR active-low pin polarity – WRSP (PMCON<1>) = 0.
11. Select PMCS2, PMCS1 active-low pin polarity – CS2P (PMCON<4>) = 0 and CS1P (PMCON<3>) = 0.
12. Select 1 wait cycle for data setup – WAITB<1:0> (PMMODE<7:6>) = 00.
13. Select 2 wait cycles to extend PMRD/PMWR – WAITM<3:0> (PMMODE<5:2>) = 0001.
14. Select 1 wait cycle for data hold – WAITE<1:0> (PMMODE<1:0>) = 00.
15. Enable upper 8 PMA<15:8> address pins – PMAEN<15:8> = 1 (the lower 8 bits can be used as general purpose I/O).

See the example code shown in Example 13-1.

Example 13-1: Initialization for Master Mode 2, Demultiplexed Address, 16-bit Data

```
/*
    Configuration Example: Master mode 2, 16-bit data, partially multiplexed
    address/data, active-lo polarities.
*/
    IEC1CLR = 0x0004        // Disable PMP interrupt
    PMCON = 0x0000;         // Stop PMP module and clear control register
    PMCONSET = 0x0B80;      // Configure the addressing and polarities
    PMMODE = 0x2A40;        // Configure the mode
    PMAEN = 0xFF00;         // Enable all address and Chip select lines

    IPC7SET = 0x001C;       // Set priority level=7 and
    IPC7SET = 0x0003;       // Set subpriority level=3
                            // Could have also done this in single
                            // operation by assigning IPC7SET = 0x001F

    IEC1SET  = 0x0004;      // Enable PMP interrupts
    PMCONSET = 0x8000;      // Enable PMP module
```

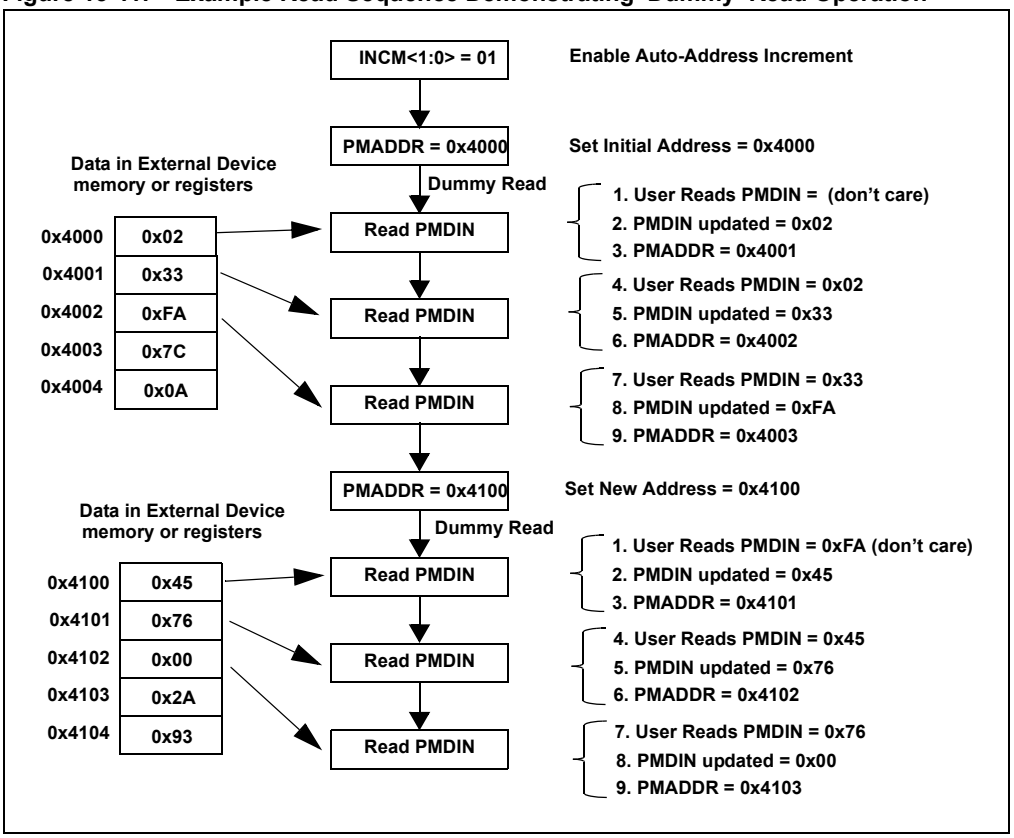
13.3.3 Read Operation

To perform a read on the parallel bus, the user application reads the PMDIN register. The effect of reading the PMDIN register retrieves the current value and causes the PMP to activate the Chip Select lines and the address bus. The read line PMRD is strobed in Master mode 2, PMRD/PMWR and PMENB lines in Master mode 1, and the new data is latched into the PMDIN register making it available the next time the PMDIN register is read.

Note that the read data obtained from the PMDIN register is actually the read value from the previous read operation. Therefore, the first user application read will be a dummy read to initiate the first bus read and fill the read register. See Figure 13-11, which illustrates this sequence. Also, the requested read value will not be ready until after the BUSY bit is observed low. Thus, in a back-to-back read operation, the data read from the register will be the same for both reads. The next read of the register will yield the new value.

In 16-bit Data mode, PMMODE<MODE16> = 1, the read from the PMDIN register causes the data bus PMD<15:0> to be read into PMDIN<15:0>. In 8-bit mode, PMMODE<MODE16> = 0, the read from the PMDIN register causes the data bus PMD<7:0> to be read into PMDIN<7:0>. The upper 8 bits, PMD<15:8>, are ignored.

Figure 13-11: Example Read Sequence Demonstrating 'Dummy' Read Operation



13.3.4 Write Operation

To perform a write on the parallel port, the user application writes to the PMDIN register (same register as a read operation). This causes the PMP module to first activate the Chip Select lines and the address bus. The write data from the PMDIN register is placed onto the PMD data bus and the write line PMPWR is strobed in Master mode 2, PMRD/PMWR and PMENB lines in Master Mode 1.

In 16-bit Data mode, PMMODE<MODE16> = 1, the write to the PMDIN register causes PMDIN<15:0> to appear on the data bus, (PMD<15:0>). In 8-bit mode, PMMODE<MODE16> = 0, the write to the PMDIN register causes PMDIN<7:0> to appear on the data bus, PMD<7:0>. The upper 8 bits, PMD<15:8>, are ignored.

13.3.5 Master Mode Interrupts

In PMP master modes, the PMPIF bit is set on every read or write strobe. An interrupt request is generated when the IRQM<1:0> bits (PMMODE<14:13>) are set = 01 and PMP interrupts are enabled, PMPIE (IEC1<2>) = 1.

13.3.6 Parallel Master Port Status – The BUSY Bit

In addition to the PMP interrupt, a BUSY bit, (PMMODE<15>), is provided to indicate the status of the module. This bit is only used in Master mode.

While any read or write operation is in progress, the BUSY bit is set for all but the very last peripheral bus cycle of the operation. This is helpful when Wait states are enabled or multiplexed address/data is selected. While the bit is set, any request by the user to initiate a new operation will be ignored (i.e., writing or reading the PMDIN register will not initiate a read or a write).

Since the system clock, SYSCLK, can operate faster than the peripheral bus clock in certain configurations, or if a large number of Wait states are used, it is possible for the PMP module to be in the process of completing a read or write operation when the next CPU instruction is reading or writing to the PMP module. For this reason, it is highly recommended that the BUSY bit be checked prior to any operation that accesses the PMDIN or PMADDR registers. Example 13-2 shows a polling operation of the BUSY bit prior to accessing the PMP module.

In most applications, the PMP module's Chip Select pin(s) provide the Chip Select interface and are under the timing control of the PMP module. However, some applications may require the PMP Chip Select pin(s) to not be configured as a Chip Select, but as a high order address line, such as PMA<14> or PMA<15>. In this situation, the application's Chip Select function must be provided by an available I/O port pin under software control. In these cases, it is especially important that the user's software poll the BUSY bit to ensure any read or write operation is complete before de-asserting the software controlled Chip Select.

Example 13-2: Example Code: Polling the BUSY Bit Flag

```
/*
    This example reads 256 16-bit words from an external device at address 0x4000 and copies
    the data to a second external device at address 0x8000. The PMP port is operating in
    Master mode 2. Note how the PMP's BUSY bit is polled prior to all operations to the
    PMDOUT, PMDIN or PMADDR register, except where noted.
*/
    unsigned short DataArray<256>;

                                  // Provide the setup code here including large Wait
                                  // states, auto increment.
    ...
    CopyData();                   // A call to the copy function is made.
    ...

void CopyData()
{
    PMADDR = 0x4000;              // Init the PMP address. First time, no need to poll BUSY
                                  // bit.
    while(PMMODE & 0x8000);       // Poll - if busy, wait before reading.
    PMDIN;                        // Read the PMDIN to clear previous data and latch new
                                  // data.

    for(i=0; i<256; i++)
    {
        while(PMMODE & 0x8000);   // Poll - if busy, wait before reading.
        DataArray<i> = PMDIN;     // Read the external device.
    }

    while(PMMODE & 0x8000);       // Poll - if busy, wait before changing PMADDR.
    PMADDR = 0x8000;              // Address of second external device.

    for(i=0; i<256; i++)
    {
        while(PMMODE & 0x8000);   // Poll - if busy, wait before writing.
        DataArray<i> = PMDIN;     // Read the external device.
    }
    return();
}
```

13.3.7 Addressing Considerations

The PMCS2 and PMCS1 Chip Select pins share functionality with address lines A15 and A14. It is possible to enable both PMCS2 and PMCS1 as Chip Selects, or enable only PMCS2 as a Chip Select; allowing PMCS1 to function strictly as address line A14. It is not possible to enable only PMCS1.

Note:	Setting both A15 and A14 = 1 when PMCS2 and PMCS1 are enabled as Chip Selects will cause both PMCS2 and PMCS1 to be active during a read or write operation. This may enable two devices simultaneously and should be avoided.

When configured as Chip Selects, a '1' must be written into bit position 15 or 14 of the PMADDR register in order for PMCS2 or PMCS1 to become active during a read or write operation. Failing to write a '1' to PMCS2 or PMCS1 does not prevent address pins PMA<13:0> from being active as the specified address appears; however, no Chip Select signal will be active.

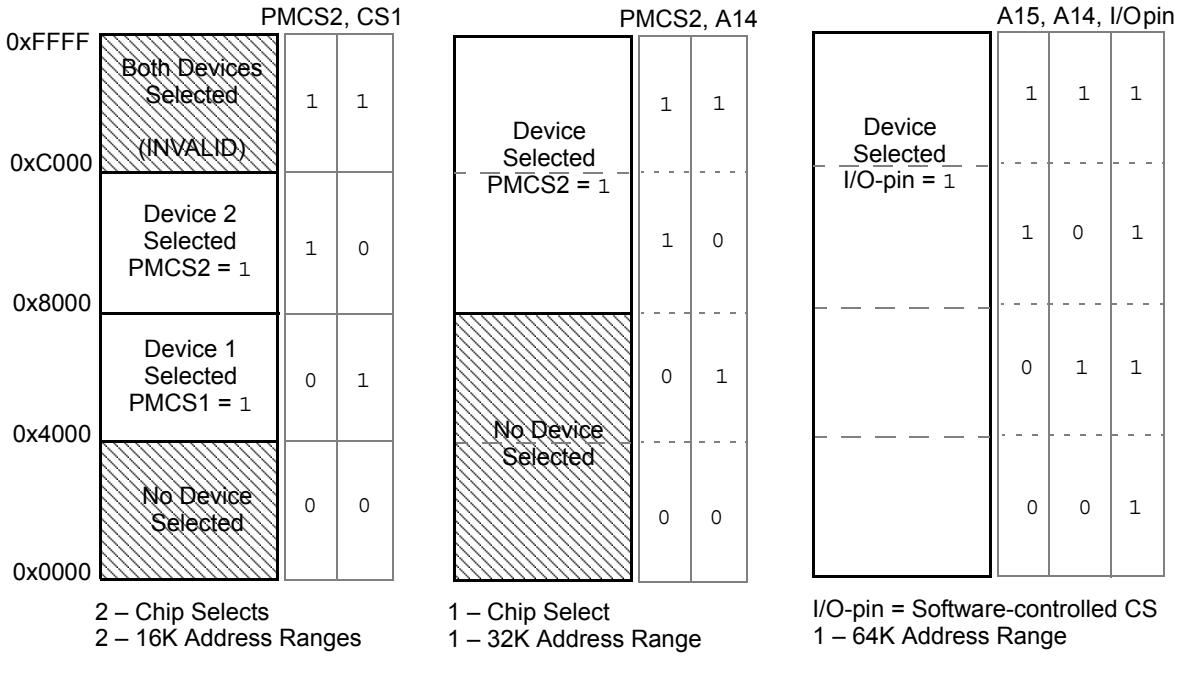
Note:	When using Auto-Increment Address mode, PMCS2 and PMCS1 do not participate and must be controlled by the user's software by writing to '1' to PMADDR<15:14> explicitly.

In Fully Multiplexed modes, address bits PMADDR<15:0> are multiplexed with the data bus and in the event address bits PMA15 or PMA14 are configured as Chip Selects, the corresponding PMADDR<15:14> address bits are automatically forced = 0. Disabling one or both PMCS2 and PMCS1 makes these bits available as address bits PMADDR<15:14>.

In any of the Master mode multiplexing schemes, disabling both Chip Select pins PMCS2 and PMCS1 requires the user to provide Chip Select line control through some other I/O pin under software control, as shown in Figure 13-12.

Refer to **13.11 "Design Tips"** for additional information regarding memory banking.

Figure 13-12: PMP Chip Select Address Maps



13.3.8 Master Mode Timing

A PMP Master mode cycle time is defined as the number of PBCLK cycles required by the PMP to perform a read or write operation and is dependent on PBCLK clock speed, PMP address/data multiplexing modes, and the number of PMP wait states, if any. Refer to the specific PIC32MX device data sheet for setup and hold timing characteristics.

A PMP master mode read or write cycle is initiated by accessing (reading or writing) the PMDIN register. Table 13-6 provides a summary of read and write PMP cycle times for each multiplex configuration.

The actual data rate of the PMP (the rate which user's code can perform a sequence of read or write operations) will be highly dependent on several factors:

- User's application code content
- Code optimization level
- Internal bus activity
- Other factors relating to the instruction execution speed.

Note:	During any Master mode read or write operation, the busy flag will always de-assert 1 peripheral bus clock cycle (TPBCLK), before the end of the operation, including Wait states. The user's application must check the status of the busy flag to ensure it is equal to '0' before initiating the next PMP operation.

Table 13-6: PMP Read/Write Cycle Times

Address/Data Multiplex Configuration	ADRMUX Bit Settings	PMP Cycle Time (PBCLK cycles)	
		Read	Write
Demultiplexed	00	2	3
Partial Multiplex	01	5	6
Fully Multiplexed (8-bit data)	10	8	9
Fully Multiplexed (16-bit data)	11	5	6

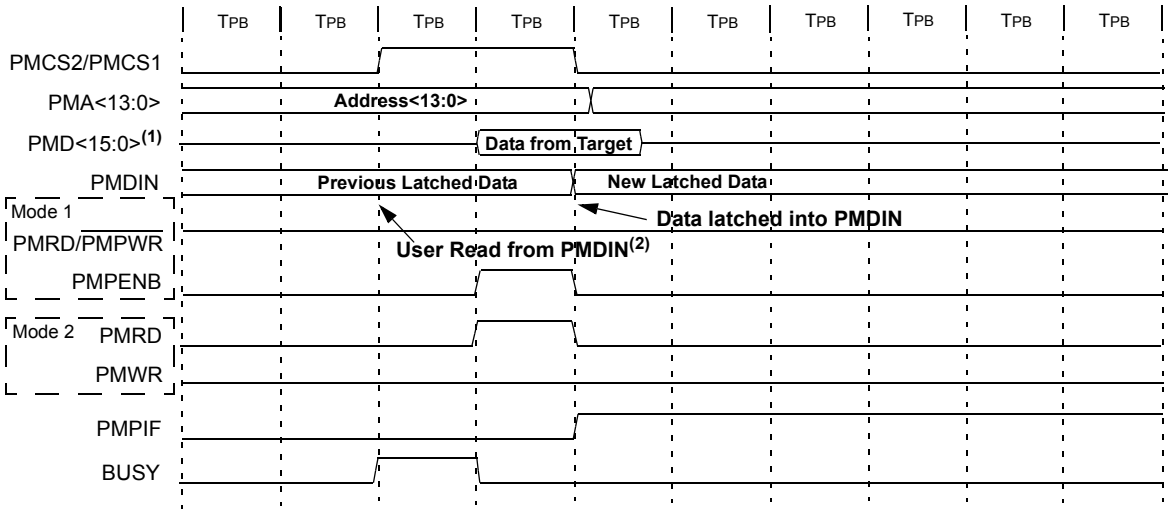
Note: Wait states are not enabled

The following timing examples represent the common master mode configuration options. These options vary from 8-bit to 16-bit data, non-multiplexed to fully multiplexed address, as well as with and without Wait states. For illustration purposes only, all control signal polarities are shown as "active-high".

13.3.8.1 DEMULTIPLEXED ADDRESS AND DATA TIMING

This timing diagram in Figure 13-13 illustrates demultiplexed timing (separate address and data bus) for a read operation with no Wait states. A read operation requires 2 TPBCLK, peripheral bus clock cycles.

Figure 13-13: 8-bit, 16-bit Read Operations, ADRMUX = 00, No Wait States

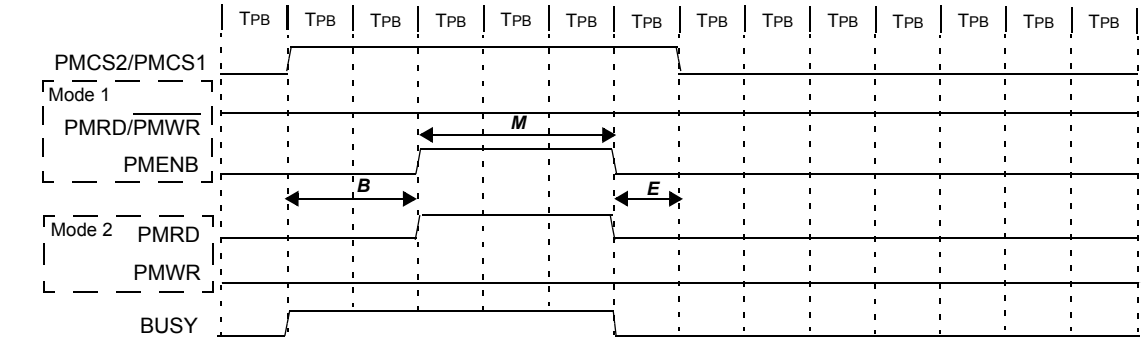


Note 1: In 8-bit mode, PMD<15:8> are not implemented.

2: Read data obtained from the PMDIN register is actually the value from the previous read operation.

In this timing diagram with Wait states, shown in Figure 13-14, the read operation requires 6 TPBCLK, peripheral bus clock cycles.

Figure 13-14: 8-bit, 16-bit Read Operations, ADRMUX = 00, Wait States Enabled



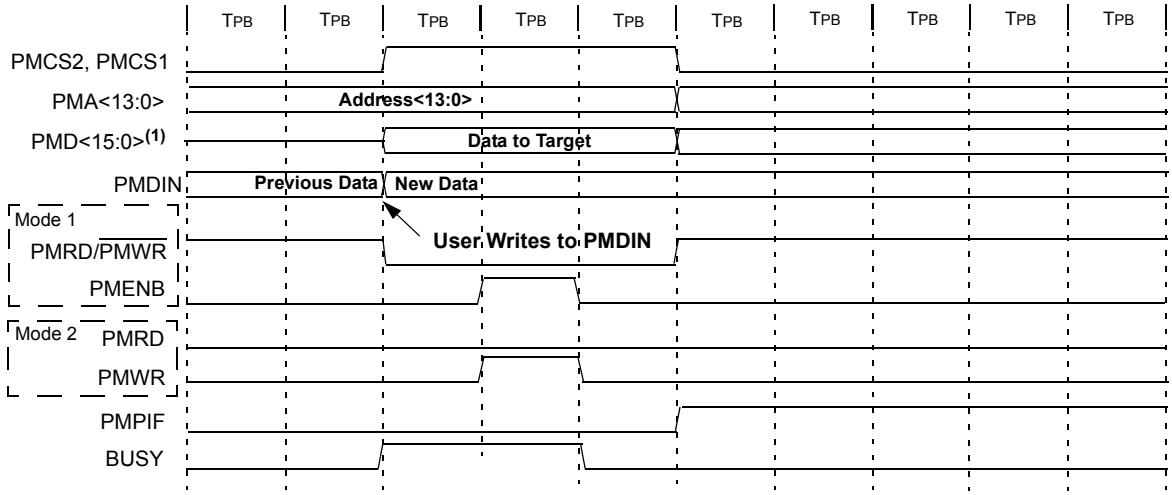
Legend:

B = WAITB<1:0> = 01 (2 Wait states)
M = WAITM<3:0> = 0010 (3 Wait states)
E = WAITE<1:0> = 01 (1 Wait state)

Note: If WAITM<3:0> = 0000, M is forced to 1 TPBCLK, WAITB is ignored (B forced to 1 TPBCLK), and WAITE is ignored (E forced to 0 TPBCLK).

The timing diagram in Figure 13-15 illustrates demultiplexed timing (separate address and data bus) for a write operation with no Wait states. A write operation requires 3 TPBCLK, peripheral bus clock cycles.

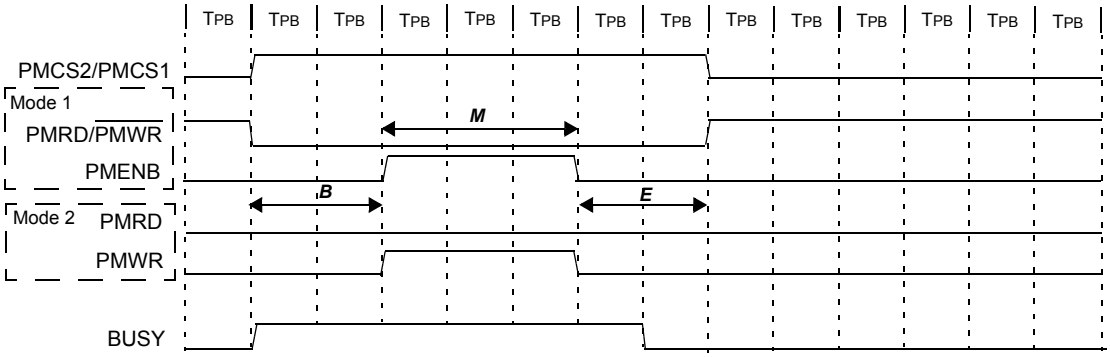
Figure 13-15: 8-bit, 16-bit Write Operations, ADRMUX = 00, No Wait States



Note 1: In 8-bit mode, PMD<15:8> are not implemented.

In this timing diagram with Wait states, shown in Figure 13-16, the write operation requires 7 TPBCLK, peripheral bus clock cycles.

Figure 13-16: 8-bit, 16-bit Write Operations, ADRMUX = 00, Wait States Enabled



Legend:

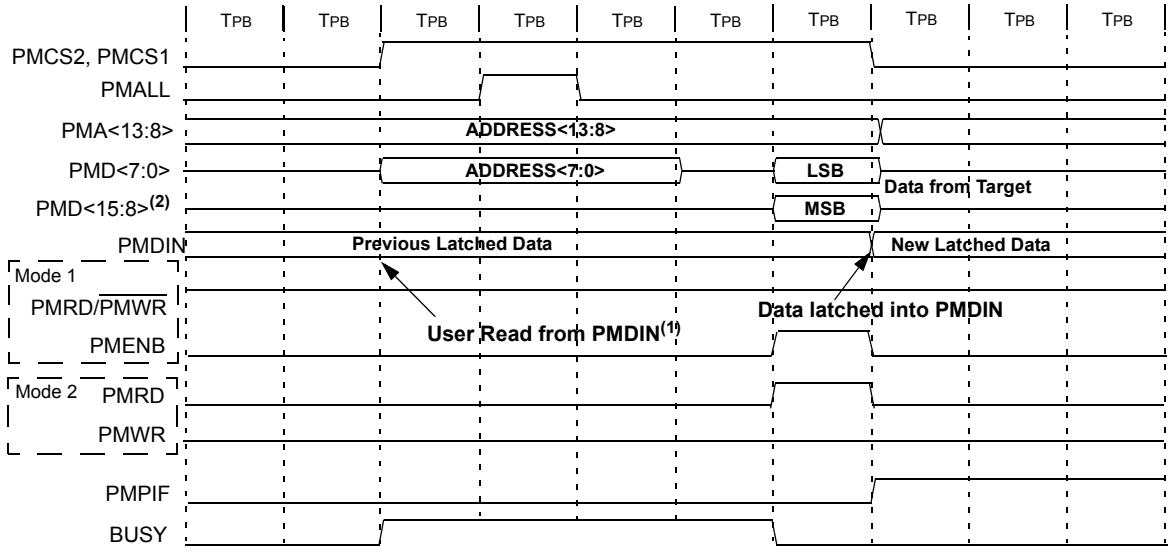
B = WAITB<1:0> = 01 (2 Wait states)
M = WAITM<3:0> = 0010 (3 Wait states)
E = WAITE<1:0> = 01 (2 Wait states)

Note: If WAITM<3:0> = 0000, M is forced to 1 TPBCLK, WAITB is ignored (B forced to 1 TPBCLK), and WAITE is ignored (E forced to 1 TPBCLK).

13.3.8.2 PARTIALLY MULTIPLEXED ADDRESS AND DATA TIMING

The timing diagram shown in Figure 13-17 illustrates partially multiplexed timing (address bits <7:0> multiplexed with data bus, PMD<7:0>) for a read operation with no Wait states. A read operation requires 5 TPBCLK, peripheral bus clock cycles.

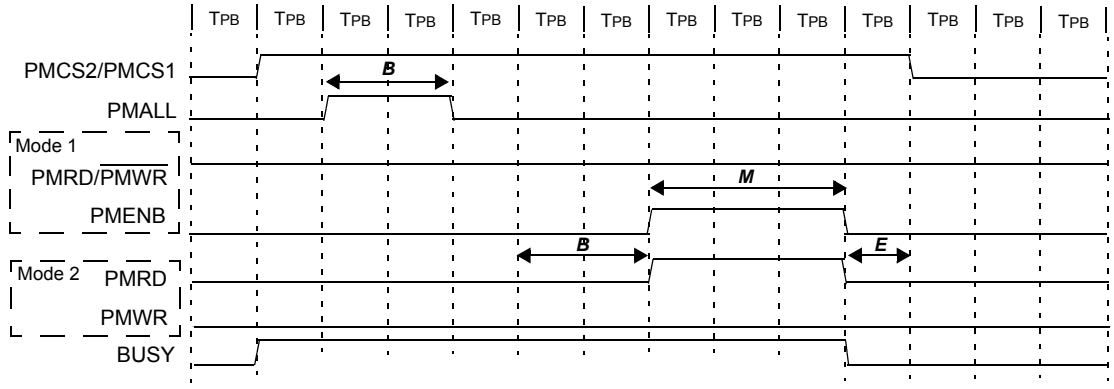
Figure 13-17: 8-bit, 16-bit Read Operations, ADRMUX = 01, No Wait States



Note 1: Read data obtained from the PMDIN register is actually the value from the previous read operation.
2: In 8-bit mode, PMD<15:8> are not implemented.

In this timing diagram with Wait states, shown in Figure 13-18, the read operation requires 10 TPBCLK, peripheral bus clock cycles.

Figure 13-18: 8-bit, 16-bit Read Operations, ADRMUX = 01, Wait States Enabled

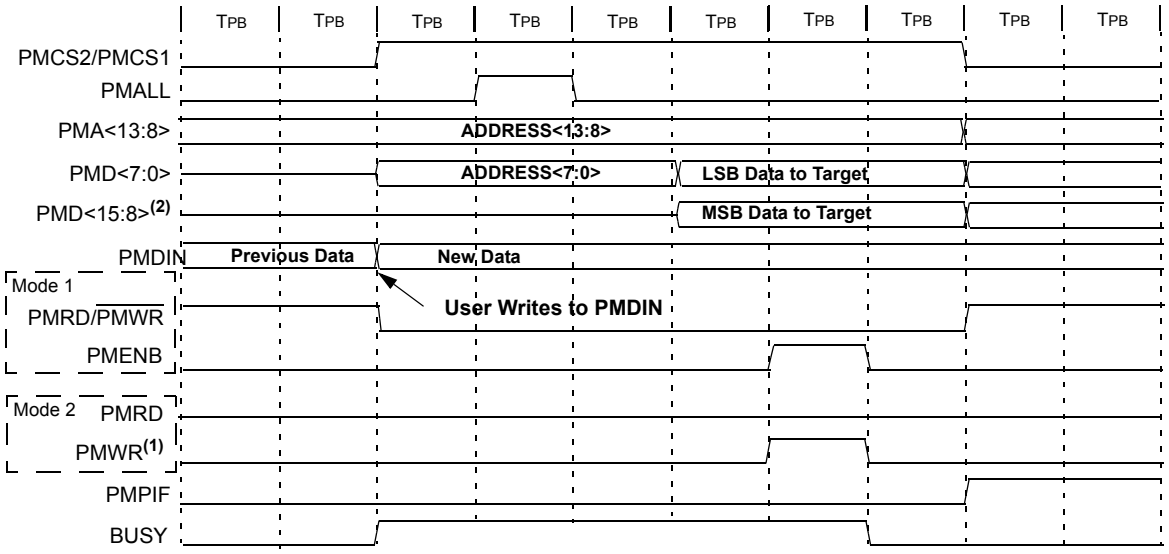


Legend:
B = WAITB<1:0> = 01 (2 Wait states)
M = WAITM<3:0> = 0010 (3 Wait states)
E = WAITE<1:0> = 01 (1 Wait state)

Note: If WAITM<3:0> = 0000, M is forced to 1 TPBCLK, WAITB is ignored (B forced to 1 TPBCLK) and WAITE is ignored (E forced to 0 TPBCLK).

The timing diagram shown in Figure 13-19 illustrates partially multiplexed timing (address bits <7:0> multiplexed with data bus, PMD<7:0>) for a write operation with no Wait states. A write operation requires 6 TPBCLK, peripheral bus clock cycles.

Figure 13-19: 8-bit, 16-bit Write Operations, ADRMUX = 01, No Wait States

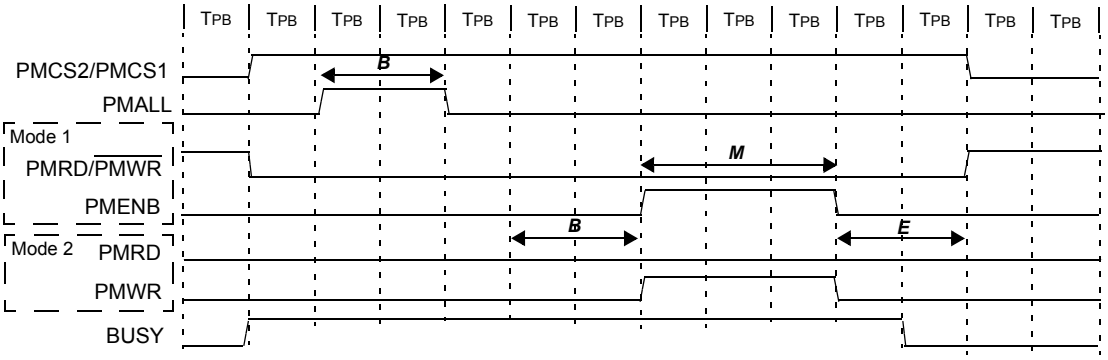


Note 1: During a write operation, there is one TPBCLK hold cycle following the PMWR signal.

2: In 8-bit mode, PMD<15:8> are not implemented.

In this timing diagram with Wait states, shown in Figure 13-20, the write operation requires 11 TPBCLK, peripheral bus clock cycles.

Figure 13-20: 8-bit, 16-bit Write Operations, ADRMUX = 01, Wait States Enabled



Legend:

B = WAITB<1:0> = 01 (2 Wait states)

M = WAITM<3:0> = 0010 (3 Wait states)

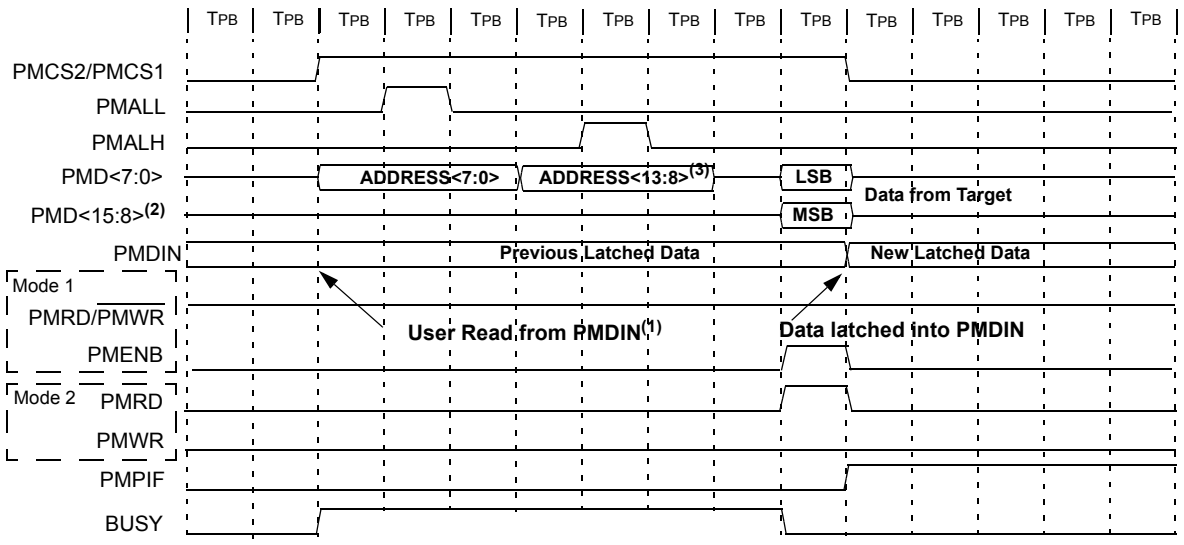
E = WAITE<1:0> = 01 (1 Wait state)

Note: If WAITM<3:0> = 0000, M is forced to 1 TPBCLK, WAITB is ignored (B forced to 1 TPBCLK) and WAITE is ignored (E forced to 1 TPBCLK).

13.3.8.3 FULLY MULTIPLEXED (8-BIT BUS) ADDRESS AND DATA TIMING

The timing diagram in Figure 13-21 illustrates fully multiplexed timing (address bits <15:0> multiplexed with data bus, PMD<7:0>) for a read operation with no Wait states. A read operation requires 8 TPBCLK, peripheral bus clock cycles.

Figure 13-21: 8-bit, 16-bit Read Operations, ADRMUX = 10, No Wait States



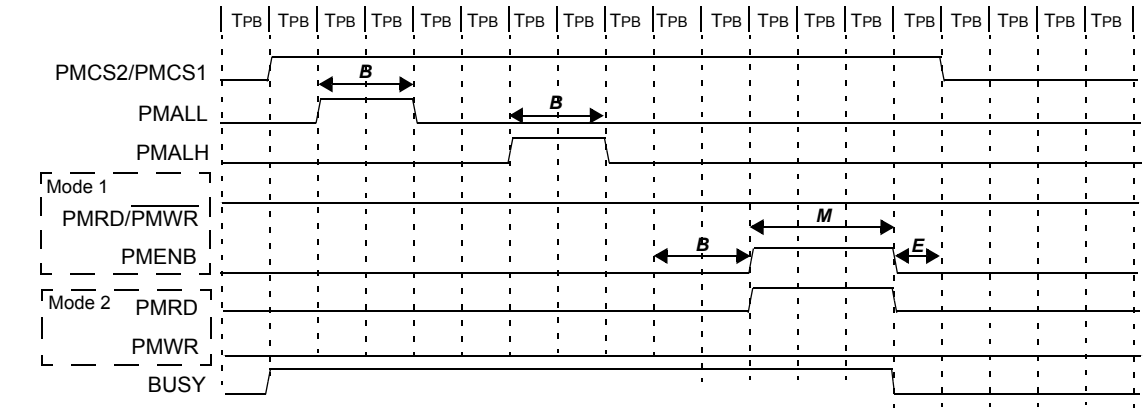
Note 1: Read data obtained from the PMDIN register is actually the value from the previous read operation.

2: In 8-bit mode, PMD<15:8> are not implemented.

3: PMADDR Address bit A15 and A14 are forced to '0' if PMCS2 and/or PMCS1 are enabled as Chip Selects.

In this timing diagram with Wait states, shown in Figure 13-22, the read operation requires 14 TPBCLK, peripheral bus clock cycles.

Figure 13-22: 8-bit, 16-bit Read Operations, ADRMUX = 10, Wait States Enabled



Legend:

B = WAITB<1:0> = 01 (2 Wait states)

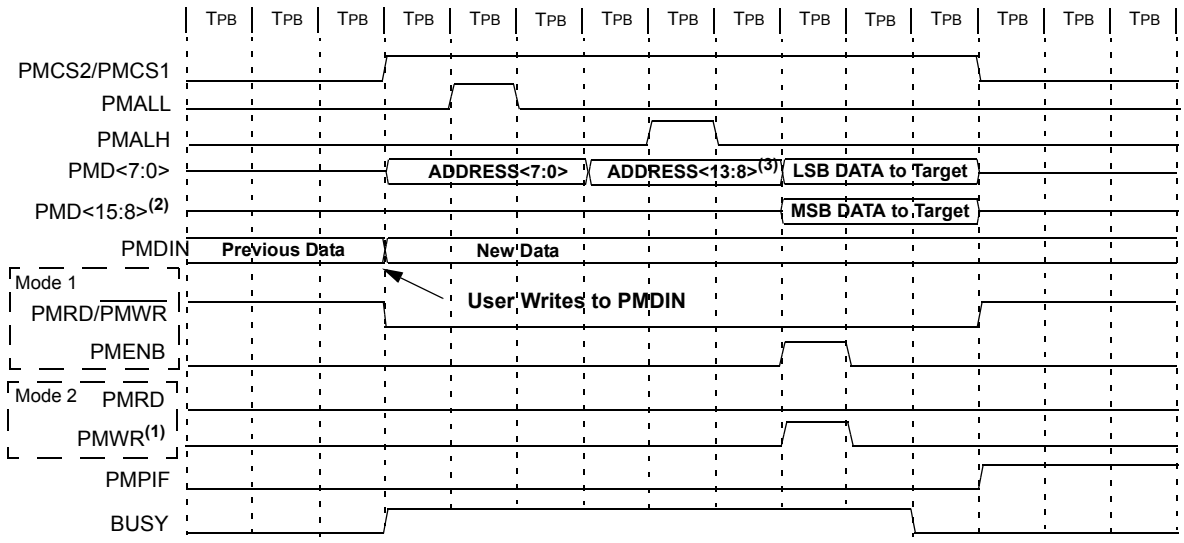
M = WAITM<3:0> = 0010 (3 Wait states)

E = WAITE<1:0> = 01 (1 Wait state)

Note: If WAITM<3:0> = 0000, M is forced to 1 TPBCLK, WAITB is ignored (B forced to 1 TPBCLK) and WAITE is ignored (E forced to 0 TPBCLK).

The timing diagram shown in Figure 13-23 illustrates fully multiplexed timing (address bits <15:0> multiplexed with data bus, PMD<7:0>) for a write operation with no Wait states. A write operation requires 9 TPBCLK, peripheral bus clock cycles.

Figure 13-23: 8-bit, 16-bit Write Operations, ADRMUX = 10, No Wait States



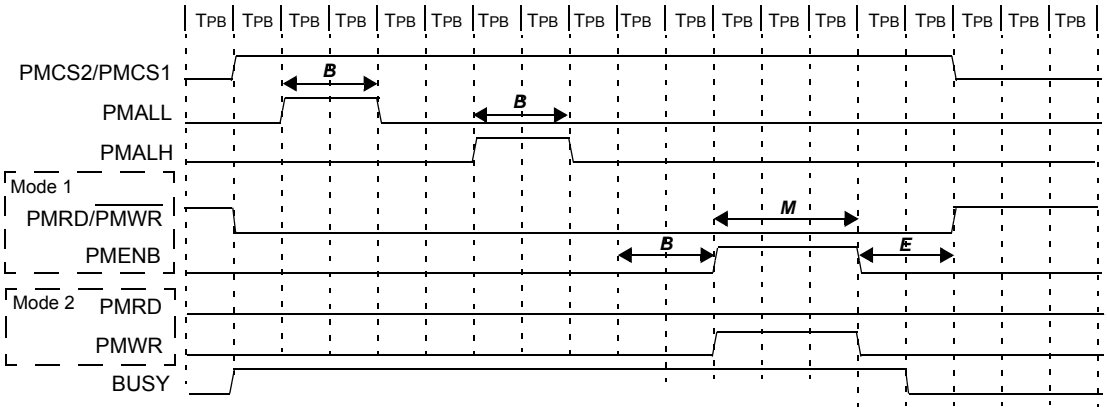
Note 1: During a write operation, there is one TPBCLK hold cycle following the PMWR signal.

2: In 8-bit mode, PMD<15:8> are not implemented.

3: PMADDR Address bit A15 and A14 are forced to '0' if PMCS2 and/or PMCS1 are enabled as Chip Selects.

In this timing diagram with Wait states, shown in Figure 13-24, the write operation requires 15 TPBCLK, peripheral bus clock cycles.

Figure 13-24: 8-bit, 16-bit Write Operations, ADRMUX = 10, Wait States Enabled



Legend:

B = WAITB<1:0> = 01 (2 Wait states)

M = WAITM<3:0> = 0010 (3 Wait states)

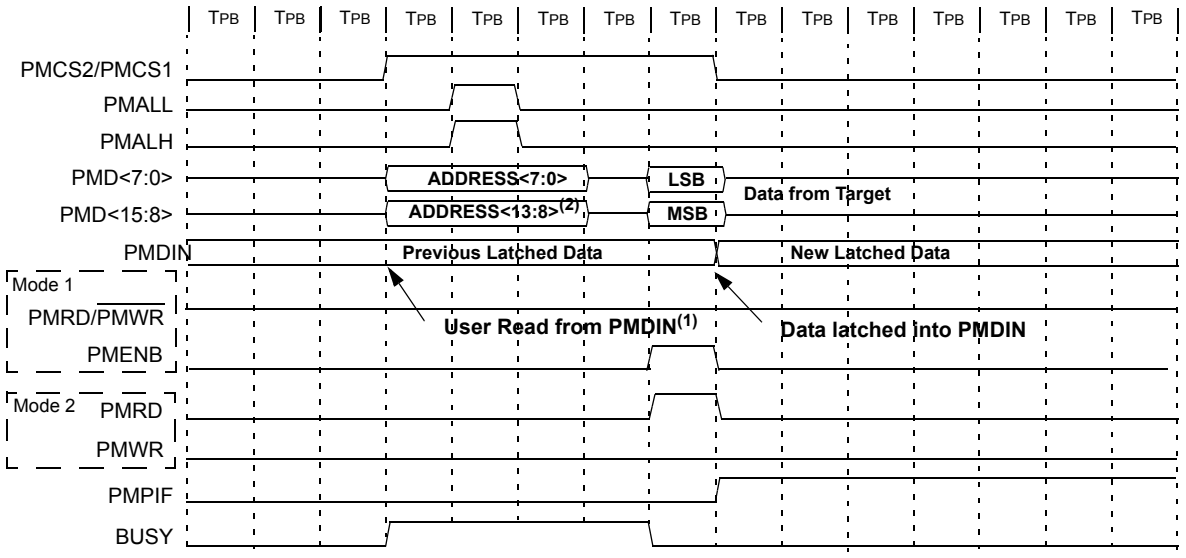
E = WAITE<1:0> = 01 (2 Wait states)

Note: If WAITM<3:0> = 0000, M is forced to 1 TPBCLK, WAITB is ignored (B forced to 1 TPBCLK), and WAITE is ignored (E forced to 1 TPBCLK).

13.3.8.4 FULLY MULTIPLEXED (16-BIT BUS) ADDRESS AND DATA TIMING

The timing diagram shown in Figure 13-25 illustrates fully multiplexed timing (address bits <15:0> multiplexed with data bus, PMD<15:0>) for a read operation with no Wait states. A read operation requires 5 TPBCLK, peripheral bus clock cycles.

Figure 13-25: 16-bit Read Operation, ADRMUX = 11, No Wait States

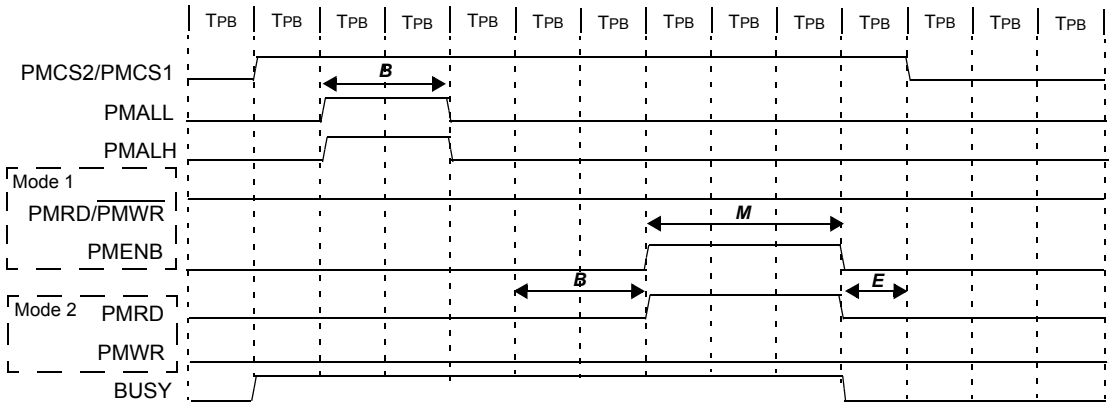


Note 1: Read data obtained from the PMDIN register is actually the value from the previous read operation.

2: PMADDR Address bit A15 and A14 are forced to '0' if PMCS2 and/or PMCS1 are enabled as Chip Selects.

In this timing diagram with Wait states, shown in Figure 13-26, the read operation requires 10 TPBCLK, peripheral bus clock cycles.

Figure 13-26: 16-bit Read Operation, ADRMUX = 11, Wait States Enabled



Legend:

B = WAITB<1:0> = 01 (2 Wait states)

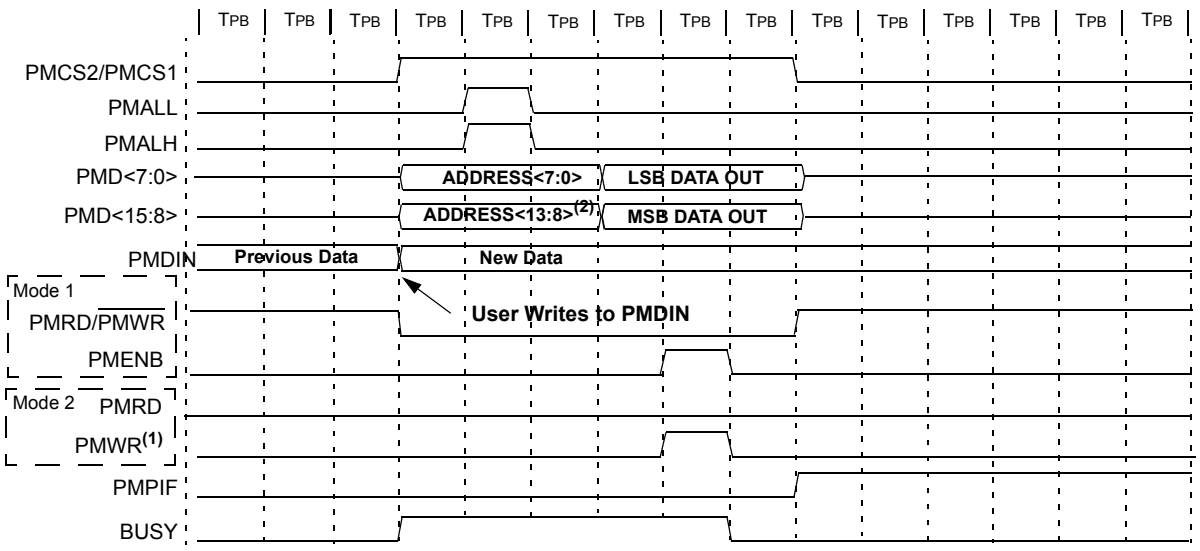
M = WAITM<3:0> = 0010 (3 Wait states)

E = WAITE<1:0> = 01 (1 Wait state)

Note: If WAITM<3:0> = 0000, M is forced to 1 TPBCLK, WAITB is ignored (B forced to 1 TPBCLK) and WAITE is ignored (E forced to 0 TPBCLK).

The timing diagram shown in Figure 13-27 illustrates fully multiplexed timing (address bits <15:0> multiplexed with data bus, PMD<15:0>) for a read operation with no Wait states. A read operation requires 6 TPBCLK, peripheral bus clock cycles.

Figure 13-27: 16-bit Write Operation, ADRMUX = 11, No Wait States

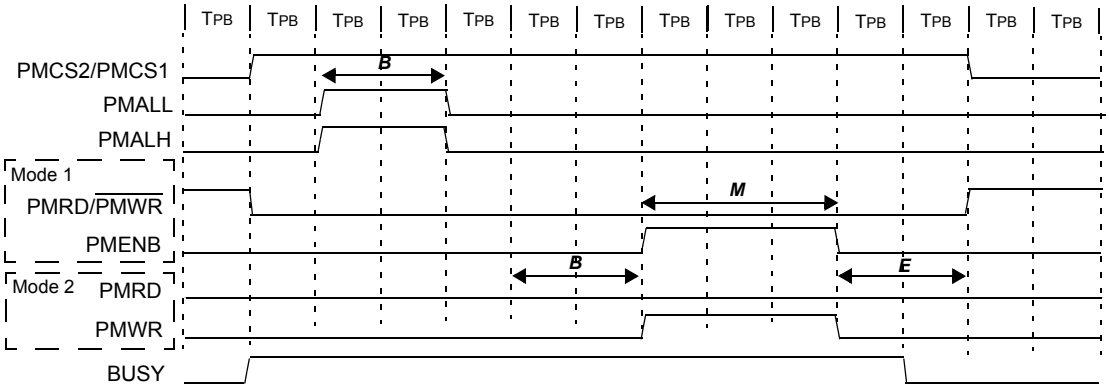


Note 1: During a write operation, there is one TPB hold cycle following the PMWR signal.

2: PMADDR Address bit A15 and A14 are forced to '0' if PMCS2 and/or PMCS1 are enabled as Chip Selects.

In this timing diagram with Wait states, shown in Figure 13-28, the write operation requires 11 TPBCLK, peripheral bus clock cycles.

Figure 13-28: 16-bit Write Operation, ADRMUX = 11, Wait States Enabled



Legend:

B = WAITB<1:0> = 01 (2 Wait states)

M = WAITM<3:0> = 0010 (3 Wait states)

E = WAITE<1:0> = 01 (2 Wait states)

Note: If WAITM<3:0> = 0000, M is forced to 1 TPBCLK, WAITB is ignored (B forced to 1 TPBCLK), and WAITE is ignored (E forced to 1 TPBCLK).

13.4 SLAVE MODES OF OPERATION

The PMP module provides 8-bit (byte) legacy Parallel Slave Port (PSP) functionality as well as new buffered and addressable slave modes.

Table 13-7: Slave Mode Selection

Slave Mode	PMCON MODE bits<1:0>	PMMODE INCM bits<1:0>
Legacy	00	x = don't care
Buffered	00	'11'
Addressable	01	x = don't care

All slave modes support 8-bit data only and the module control pins are automatically dedicated when any of these modes are selected. The user application only needs to configure the polarity of the PMCS1, PMRD and PMWR signals.

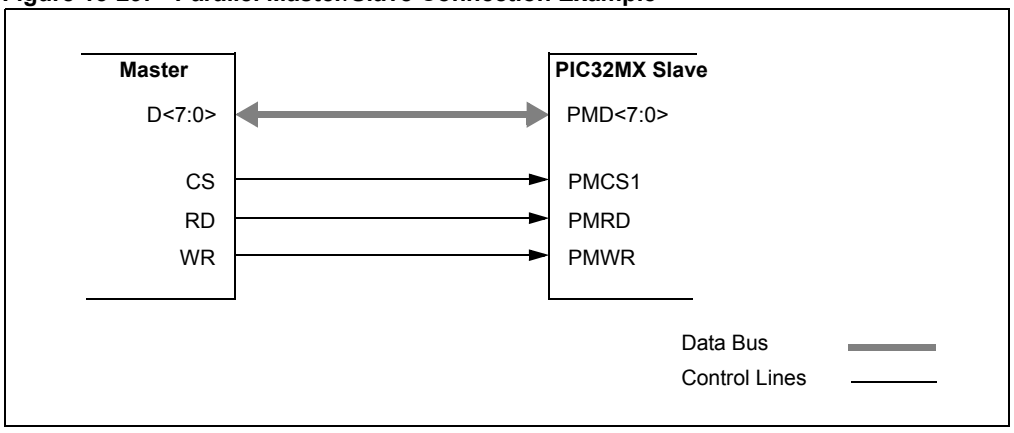
Table 13-8: Slave Mode Pin Polarity Configuration

CONTROL PIN	PMCON Control Bit	Active-High Select	Active-Low Select
PMRD	RDSP	1	0
PMWR	WRSP	1	0
PMCS1	CS1P	1	0

13.4.1 Legacy Slave Port Mode

In 8-bit PMP Legacy Slave mode, the module is configured as a PSP using control bits MODE<1:0> (PMMODE<9:8>) = 00. In this mode, an external device such as another microcontroller or microprocessor can asynchronously read and write data using the 8-bit data bus PMD<7:0>, the read PMRD, write PMWR and Chip Select PMCS1 inputs.

Figure 13-29: Parallel Master/Slave Connection Example



13.4.1.1 INITIALIZATION STEPS

The following Slave mode initialization properly prepares the PMP port for communicating with an external device.

1. Clear the ON control bit (PMCON<15> = 0) to disable the PMP module.
2. Select the Legacy mode with MODE<1:0> (PMMODE<9:8>) = 00.
3. Select the polarity of the Chip Select pin CS1P (PMCON<3>).
4. Select the polarity of the control pins WRSP and RDSP (PMCON<1:0>).
5. If interrupts are used:
   a) Clear interrupt flag bit PMPIF (IFS1<2>) = 0.
   b) Configure the PMP interrupt priority bits PMPIP<2:0> (IPC7<4:2>) and interrupt subpriority bits PMPIS (IPC7<1:0>).
   c) Enable PMP interrupt by setting interrupt enable bit PMPIE (IEC1<2>) = 1.
6. Set the ON control bit to '1' to enable the PMP module.

Example 13-3: Example Code: Legacy Parallel Slave Port Initialization

```
/*
   Example Configuration for Legacy Slave mode
*/
   IEC1CLR  = 0x0004  // Disable PMP interrupt in case it is already enabled
   PMCON    = 0x0000  // Stop and Configure PMCON register for Legacy mode
   PMMODE   = 0x0000  // Configure PMMODE register
   IPC7SET  = 0x001C; // Set priority level = 7 and
   IPC7SET  = 0x0003; // Set subpriority level = 3
                      // Could have also done this in single
                      // operation by assigning IPC7SET = 0x001F
   IFS1CLR  = 0x0004; // Clear the PMP interrupt status flag
   IEC1SET  = 0x0004; // Enable PMP interrupts
   PMCONSET = 0x8000; // Enable PMP module
```

13.4.1.2 WRITE TO SLAVE PORT

When Chip Select is active and a write strobe occurs, the data on the bus pins PMD<7:0> is captured into the lower 8 bits of the PMDIN register, PMDIN<7:0>. The PMPIF (interrupt flag bit) is set during the write strobe, however, IB0F (input buffer full flag) bit requires two to three peripheral bus clock cycles to synchronize before it is set and the PMDIN register can be read. The IB0F bit will remain set until the PMDIN register is read by the user application. If a write operation occurs while the IB0F bit is = 1, the write data will be ignored and an overflow condition will be generated, IB0V = 1. Refer to the timing diagrams in **13.4.4 "Slave Mode Read and Write Timing Diagrams"**.

13.4.1.3 READ FROM SLAVE PORT

When Chip Select is active and a read strobe occurs, the data from the lower 8 bits of the PMDOUT register, PMDOUT<7:0> is presented onto data bus pins PMD<7:0> and read by the master device. The PMPIF (interrupt flag bit) is set during the read strobe; however, the OB0E (output buffer empty flag) bit requires two to three peripheral bus clock cycles to synchronize before it is set. The OB0E bit will remain set until the PMDOUT register is written to by the user application. If a read operation occurs while the OB0E bit is = 1, the read data will be the same as the previous read data and an underflow condition will be generated, OBUF = 1. Refer to the timing diagrams in **13.4.4 "Slave Mode Read and Write Timing Diagrams"**.

13.4.1.4 LEGACY MODE INTERRUPT OPERATION

In PMP Legacy Slave mode, the PMPIF bit is set every read or write strobe. If using interrupts, the user's application vectors to an Interrupt Service Routine (ISR) where the IBF and OBE Status bits can be examined to determine if the buffer is full or empty. If not using interrupts, the user's application should wait for PMPIF to be set before polling the IBF and OBE Status bits to determine if the buffer is full or empty.

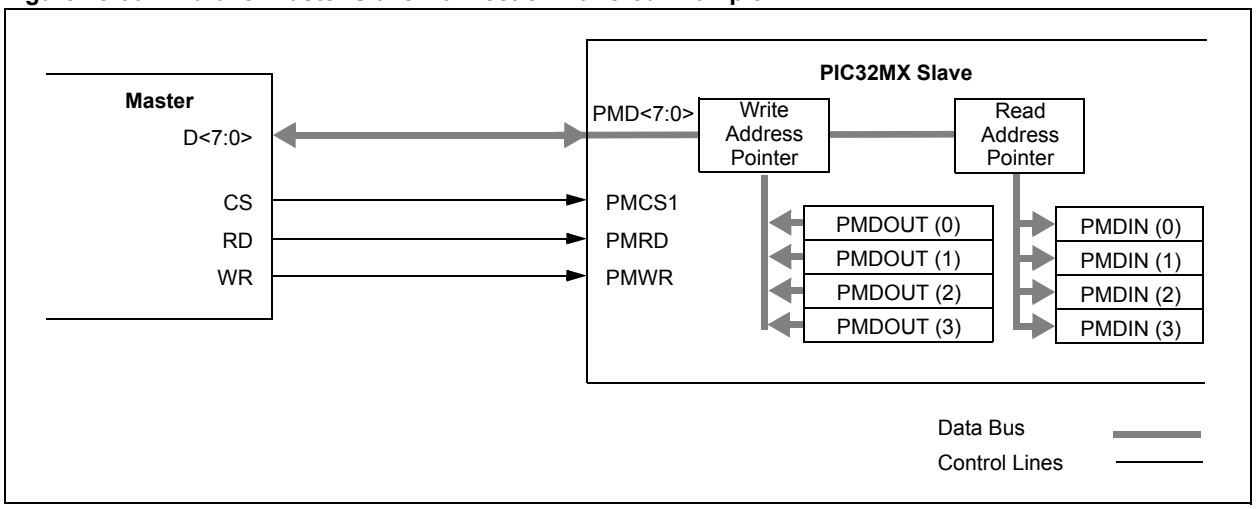
Note:	On persistent interrupt implementations of the PMP, the interrupt is generated on the falling edge of the WR signal. On non-persistent interrupt implementations of the PMP, the interrupt is generated on the rising edge of the WR signal. Firmware should poll the BUSY bit to ensure the data is valid before attempting to read the data from the PMP module.

13.4.2 Buffered Parallel Slave Port Mode

The 8-bit Buffered Parallel Slave Port mode is functionally identical to the Legacy Parallel Slave Port mode with one exception: the implementation of 4-level read and write buffers. Buffered Slave mode is enabled by setting the PMMODE(<MODE1:MODE0>) bits = 00, and the PMMODE<INCM1:INCM0> bits = 11.

When the buffered mode is active, the module uses the PMDIN register as write buffers and the PMDOUT register as read buffers. Each register is divided into four 8-bit buffer registers, four read buffers in PMDOUT and four write buffers in PMDIN. Buffers are numbered 0 through 3, starting with the lower byte <7:0> and progressing upward through the high byte <31:24>.

Figure 13-30: Parallel Master/Slave Connection Buffered Example



13.4.2.1 INITIALIZATION STEPS

The following Buffered Slave mode initialization properly prepares the PMP port for communicating with an external device.

1. Clear the ON control bit (PMCON<15> = 0) to disable the PMP module.
2. Select the Legacy mode with MODE<1:0> (PMMODE<9:8>) = 00.
3. Select Buffer mode with INCM<1:0> (PMMODE<12:11>) = 11.
4. Select the polarity of the Chip Select CS1P (PMCON<3>).
5. Select the polarity of the control pins with WRSP and RDSP (PMCON<1:0>).
6. If interrupts are used:
   a) Clear interrupt flag bit PMPIF (IFS1<2>).
   b) Configure interrupt priority and subpriority levels in IPC7.
   c) Set interrupt enable bit PMPIE (IEC1<2>).
7. Set the ON control bit to '1' to enable the PMP module.

Example 13-4: Example Code: Buffered Parallel Slave Port Initialization

```
/*
    Example Configuration for Buffered Slave mode
*/
    IEC1CLR = 0x0004       // Disable PMP interrupt in case it is already enabled
    PMCON = 0x0000         // Stop and Configure PMCON register for Buffered mode
    PMMODE = 0x1800        // Configure PMMODE register
    IPC7SET = 0x001C;      // Set priority level = 7 and
    IPC7SET = 0x0003;      // Set subpriority level = 3
                           // Could have also done this in single operation by assigning
                           // IPC7SET = 0x001F
    IFS1CLR = 0x0004;      // Clear the PMP interrupt status flag
    IEC1SET = 0x0004;      // Enable PMP interrupts
    PMCONSET = 0x8000;     // Enable PMP module
```

13.4.2.2 READ FROM SLAVE PORT

For read operations, the bytes will be sent out sequentially, starting with Buffer 0, PMDOUT<7:0>, and ending with Buffer 3, PMDOUT<31:24>, for every read strobe. The module maintains an internal pointer to keep track of which buffer is to be read.

Each of the buffers has a corresponding read Status bit, OBnE, in the PMSTAT register. This bit is cleared when a buffer contains data that has not been written to the bus, and is set when data is written to the bus. If the current buffer location being read from is empty, a buffer underflow is generated, and the Buffer Overflow flag bit OBUF is set. If all four OBnE Status bits are set, the Output Buffer Empty flag OBE will also be set. Refer to the timing diagrams in **13.4.4 "Slave Mode Read and Write Timing Diagrams"**.

13.4.2.3 WRITE TO SLAVE PORT

For write operations, the data is be stored sequentially, starting with Buffer 0, PMDIN<7:0> and ending with Buffer 3, PMDIN<31:24>. As with read operations, the module maintains an internal pointer to the buffer that is to be written next.

The input buffers have their own write Status bits, IBnF. The bit is set when the buffer contains unread incoming data, and cleared when the data has been read. The flag bit is set on the write strobe. If a write occurs on a buffer when its associated IBnF bit is set, the Buffer Overflow flag IBOV is set; any incoming data in the buffer will be lost. If all four IBnF flags are set, the Input Buffer Full flag IBF is set. Refer to timing diagrams in **13.4.4 "Slave Mode Read and Write Timing Diagrams"**.

13.4.2.4 BUFFERED MODE INTERRUPT OPERATION

In Buffered Slave mode, the module can be configured to generate an interrupt on every read or write strobe, IRQM<1:0> (PMMODE<14:13>) = 01. It can be configured to generate an interrupt on a read from Read Buffer 3 or a write to Write Buffer 3, IRQM<1:0> = 10, which is essentially an interrupt every fourth read or write strobe. When interrupting every fourth byte for input data, all input buffer registers should be read to clear the IBnF flags. If these flags are not cleared then there is a risk of hitting an overflow condition.

If using interrupts, the user's application vectors to an Interrupt Service Routine (ISR) where the IBF and OBE Status bits can be examined to determine if the buffer is full or empty. If not using interrupts, the user application should wait for PMPIF to be set before polling the IBF and OBE Status bits to determine if the buffer is full or empty.

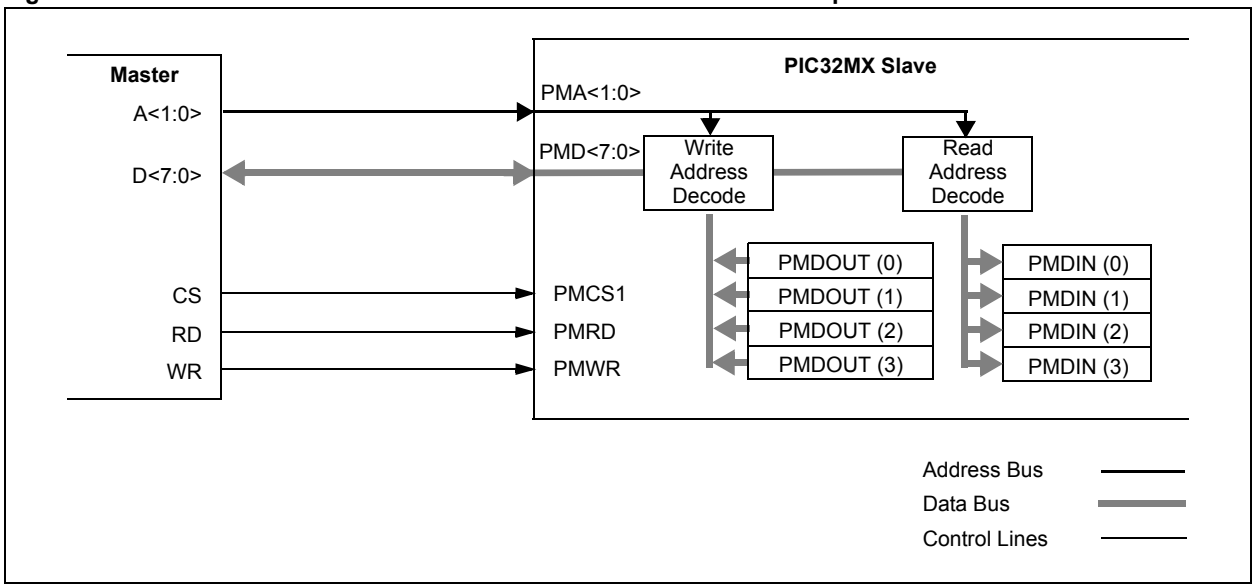
13.4.3 Addressable Buffered Parallel Slave Port Mode

In the 8-bit Addressable Buffered Parallel Slave Port mode the module is configured with two extra inputs, PMA<1:0>. This makes the 4-byte buffer space directly addressable as fixed pairs of read and write buffers. As with Buffered Legacy mode, data is output from register PMDOUT and is input to register PMDIN. Table 13-9 shows the address resolution for the incoming address to the input and output registers.

Table 13-9: Slave Mode Buffer Addresses

PMA<1:0>	Output Register (Buffer)	Input Register (Buffer)
00	PMDOUT<7:0> (0)	PMDIN<7:0> (0)
01	PMDOUT<15:8> (1)	PMDIN<15:8> (1)
10	PMDOUT<23:16> (2)	PMDIN<23:16> (2)
11	PMDOUT<31:24> (3)	PMDIN<31:24> (3)

Figure 13-31: Parallel Master/Slave Connection Addressed Buffer Example



13.4.3.1 INITIALIZATION STEPS

The following Addressable Buffered Slave mode initialization properly prepares the PMP port for communicating with an external device.

1. Clear the ON control bit (PMCON<15> = 0) to disable the PMP module.
2. Select the Legacy mode with MODE<1:0> (PMMODE<9:8) = 00.
3. Select the polarity of the Chip Select CS1P (PMCON<3>).
4. Select the polarity of the control pins with WRSP and RDSP (PMCON<1:0>).
5. If interrupts are used:
   a) Clear interrupt flag bit PMPIF (IFS1<2>).
   b) Configure interrupt priority and subpriority levels in IPC7.
   c) Set interrupt enable bit PMPIE (IEC1<2>).
6. Set the ON control bit to '1' to enable the PMP module.

Example 13-5: Example Code: Addressable Parallel Slave Port Initialization

```
/*
    Example Configuration for Addressable Slave mode
*/
    IEC1CLR = 0x0004       // Disable PMP interrupt in case it is already enabled
    PMCON = 0x0000         // Stop and Configure PMCON register for Address mode
    PMMODE = 0x0100        // Configure PMMODE register
    IPC7SET = 0x001C;      // Set priority level = 7 and
    IPC7SET = 0x0003;      // Set subpriority level = 3
                           // Could have also done this in single operation by assigning
                           // IPC7SET = 0x001F
    IFS1CLR = 0x0004;      // Clear the PMP interrupt status flag
    IEC1SET = 0x0004;      // Enable PMP interrupts
    PMCONSET = 0x8000;     // Enable PMP module
```

13.4.3.2 READ FROM SLAVE PORT

When Chip Select is active and a read strobe occurs, the data from one of the four output 8-bit buffers is presented onto PMD<7:0>. The byte selected to be read depends on the 2-bit address placed on PMA<1:0>. Table 13-9 shows the corresponding output registers and their associated address. When an output buffer is read, the corresponding OBnE bit is set. The OBE flag bit is set when all the buffers are empty. If any buffer is already empty, OBnE = 1, the next read to that buffer will generate an OBUF event. Refer to the timing diagrams in **13.4.4 "Slave Mode Read and Write Timing Diagrams"**.

13.4.3.3 WRITE TO SLAVE PORT

When Chip Select is active and a write strobe occurs (PMCS = 1 and PMWR = 1), the data from PMD<7:0> is captured into one of the four input buffer bytes. The byte selected to be written depends on the 2-bit address placed on ADDR<1:0>. Table 13-9 shows the corresponding input registers and their associated address.

When an input buffer is written, the corresponding IBnF bit is set. The IBF flag bit is set when all the buffers are written. If any buffer is already written, IBnF = 1, the next write strobe to that buffer will generate an IBOV event, and the byte will be discarded. Refer to the timing diagrams in **13.4.4 "Slave Mode Read and Write Timing Diagrams"**.

13.4.3.4 ADDRESSABLE BUFFERED MODE INTERRUPT OPERATION

In Addressable Slave mode, the module can be configured to generate an interrupt on every read or write strobe, IRQM<1:0> (PMMODE<14:13>) = 01. It can also be configured to generate an interrupt on any read from Read Buffer 3 or write to Write Buffer 3, IRQM<1:0> = 10; in other words, an interrupt will occur whenever a read or write occurs when PMA<1:0> is '11'.

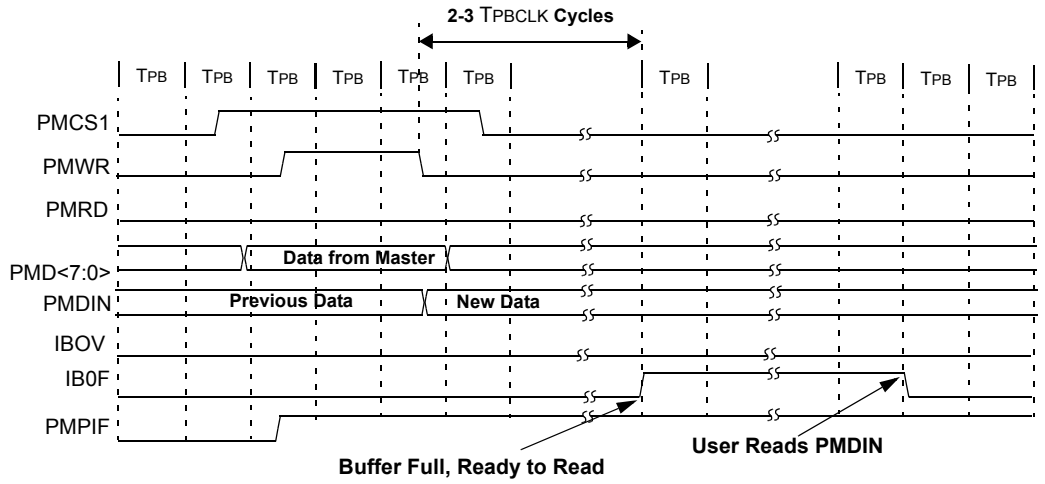
If using interrupts, the user application vectors to an Interrupt Service Routine (ISR) where the IBF and OBE Status bits can be examined to determine if the buffer is full or empty. If not using interrupts, the user application should wait for PMPIF to be set before polling the IBF and OBE Status bits to determine if the buffer is full or empty.

13.4.4 Slave Mode Read and Write Timing Diagrams

In all of the slave modes, an external master device is connected to the parallel slave port and is controlling the read and write operations. When an external read or write operation is performed by the external master device, the PMPIF (IFS1<2>) will be set on the active edge of PMRD or PMWR pin.

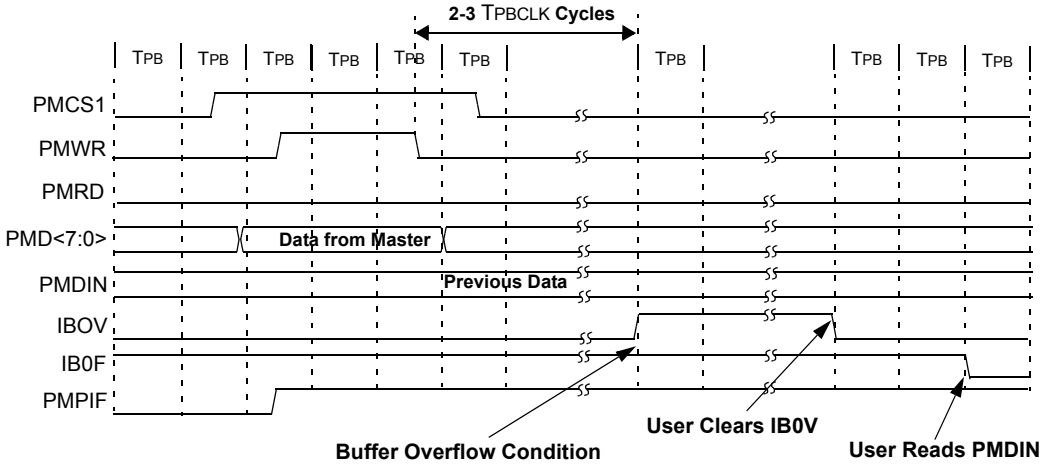
- For any external write operation, the user's application must poll the IBOV or IB0F buffer Status bits to ensure adequate time for the write operation to be completed before accessing the PMDIN register.
- For any external read operation, the user's application must poll the OBUF or OB0E buffer Status bits to ensure adequate time for the read operation to be completed before accessing the PMDOUT register.

Figure 13-32: Parallel Slave Port Write Operation



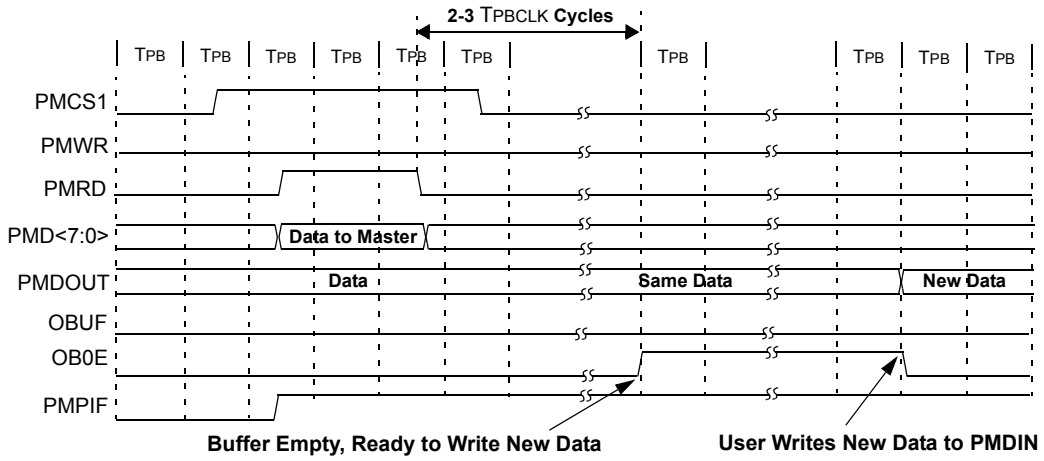
Note: Control signal polarity are configurable and are shown active-high in this example.

Figure 13-33: Parallel Slave Port Write Operation – Buffer Full, Overflow Condition



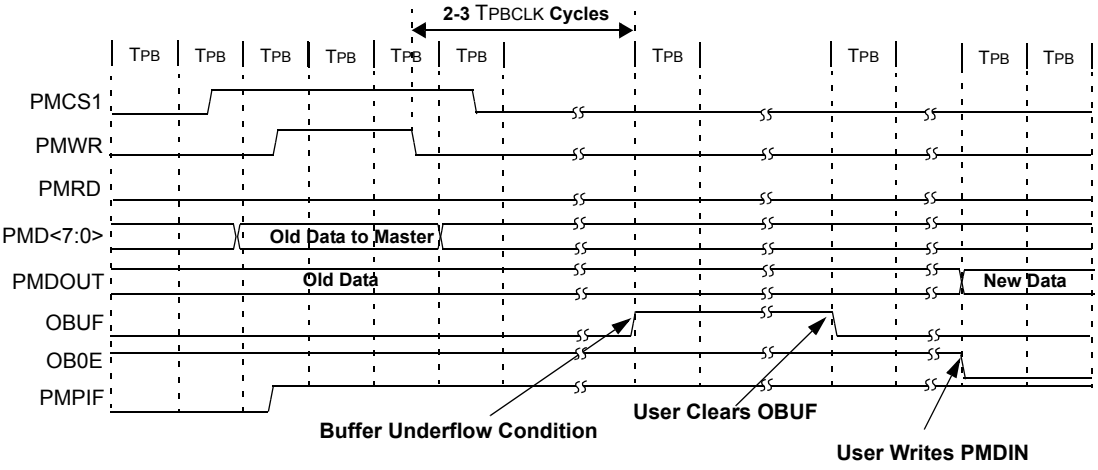
Note: Control signal polarity are configurable and are shown active-high in this example.

Figure 13-34: Parallel Slave Port Read Operation



Note: Control signal polarity are configurable and are shown active-high in this example.

Figure 13-35: Parallel Slave Port Read Operation – Buffer Empty, Underflow Condition



Note: Control signal polarity are configurable and are shown active-high in this example.

13.5 INTERRUPTS

The Parallel Master Port has the ability to generate an interrupt, depending on the selected Operating mode.

- PMP (Master) mode:
  - Interrupt on every completed read or write operation.
- PSP (Legacy Slave) mode:
  - Interrupt on every read and write byte
- PSP (Buffered Slave) mode:
  - Interrupt on every read and write byte
  - Interrupt on read or write byte of Buffer 3 (PMDOUT<31:24>)
- EPSP (Enhanced Addressable Slave) mode:
  - Interrupt on every read and write byte
  - Interrupt on read or write byte of Buffer 3 (PMDOUT<31:24>), PMA<1:0> = 11.

The PMPIF bit must be cleared in software.

The PMP module is enabled as a source of interrupt via the PMP Interrupt Enable bit, PMPIE. The Interrupt Priority level bits (PMPIP<2:0>) and Interrupt Subpriority level bits (PMPIS<1:0>) must also be configured. Refer to **Section 8. "Interrupts"** (DS61108) for further details.

13.5.1 Interrupt Configuration

The PMP module has a dedicated interrupt flag bit PMPIF and a corresponding interrupt enable/mask bit PMPIE. These bits are used to determine the source of an interrupt and to enable or disable an individual interrupt source.

The PMPIE bit is used to define the behavior of the Vector Interrupt Controller or Interrupt Controller when the PMPIF is set. When the PMPIE bit is clear, the Interrupt Controller module does not generate a CPU interrupt for the event. If the PMPIE bit is set, the Interrupt Controller module will generate an interrupt to the CPU when the PMPIF bit is set (subject to the priority and subpriority as outlined below).

It is the responsibility of the user's software routine that services a particular interrupt to clear the appropriate Interrupt Flag bit before the service routine is complete.

The priority of PMP module can be set with the PMPIP<2:0> bits. This priority defines the priority group to which the interrupt source will be assigned. The priority groups range from a value of 7, the highest priority, to a value of 0, which does not generate an interrupt. An interrupt being serviced will be preempted by an interrupt in a higher priority group.

The subpriority bits allow setting the priority of a interrupt source within a priority group. The values of the subpriority, PMPIS<1:0>, range from 3, the highest priority, to 0 the lowest priority. An interrupt with the same priority group but having a higher subpriority value will preempt a lower subpriority interrupt that is in progress.

The priority group and subpriority bits allow more than one interrupt source to share the same priority and subpriority. If simultaneous interrupts occur in this configuration the natural order of the interrupt sources within a priority/subgroup pair determine the interrupt generated. The natural priority is based on the vector numbers of the interrupt sources. The lower the vector number the higher the natural priority of the interrupt. Any interrupts that were overridden by natural order will then generate their respective interrupts based on priority, subpriority and natural order after the interrupt flag for the current interrupt is cleared.

After an enabled interrupt is generated, the CPU will jump to the vector assigned to that interrupt. The vector number for the interrupt is the same as the natural order number. The CPU will then begin executing code at the vector address. The user's code at this vector address should perform any application specific operations and clear the PMPIF interrupt flag, and then exit. Refer to **Section 8. "Interrupts"** (DS61108) for the vector address table details for more information on interrupts.

Table 13-10: PMP Interrupt Vectors for Various Offsets with EBASE = 0x8000:0000

Channel	Vector/Natural Order	IRQ Number	Vector Address IntCtl.VS = 0x01	Vector Address IntCtl.VS = 0x02	Vector Address IntCtl.VS = 0x04	Vector Address IntCtl.VS = 0x08	Vector Address IntCtl.VS = 0x10
PMP	28	34	8000 0580	8000 0900	8000 1000	8000 1E00	8000 3A00

Table 13-11: Priority and Subpriority Assignment Example

Channel	Priority Group	Subpriority	Vector/Natural Order
PMP	7	3	28

Example 13-6: PMP Module Interrupt Initialization Code Example

```
/*
    The following code example illustrates a PMP interrupt configuration.
    When the PMP interrupt is generated, the cpu will branch to the vector assigned to PMP
    interrupt.
*/

    // Configure PMP for desired mode of operation
    ...
    // Configure the PMP interrupts
    IPC7SET = 0x0014;      // Set priority level = 5
    IPC7SET = 0x0003;      // Set subpriority level = 3
                           // Could have also done this in single
                           // operation by assigning IPC7SET = 0x0017

    IFS1CLR = 0x0004;      // Clear the PMP interrupt status flag
    IEC1SET = 0x0004;      // Enable PMP interrupts
    PMCONSET = 0x8000;     // Enable PMP module
```

Example 13-7: PMP ISR Code Example

```
/*
    The following code example demonstrates a simple Interrupt Service Routine for PMP
    interrupts. The user's code at this vector should perform any application specific
    operations and must clear the PMP interrupt status flag before exiting.
*/

void __ISR(_PMP_VECTOR, ipl5) PMP_HANDLER(void)
{
    ... perform application specific operations in response to the interrupt

    IFS1CLR = 0x0004;      // Be sure to clear the PMP interrupt status
                           // flag before exiting the service routine.
}
```

Note: The PMP ISR code example shows MPLAB® C32 C compiler-specific syntax. Refer to your compiler manual regarding support for ISRs.

13.6 OPERATION IN POWER-SAVING AND DEBUG MODES

13.6.1 PMP Operation in Sleep Mode

When the device enters Sleep mode, the system clock is disabled. The consequences of Sleep mode depend on which mode the module is configured in at the time that Sleep mode is invoked.

13.6.1.1 PMP OPERATION – SLEEP IN MASTER MODE

If the microcontroller enters Sleep mode while the module is operating in Master mode, PMP operation will be suspended in its current state until clock execution resumes. As this may cause unexpected control pin timings, users should avoid invoking Sleep mode when continuous use of the module is needed.

13.6.1.2 PMP OPERATION – SLEEP IN SLAVE MODE

While the module is inactive but enabled for any Slave mode operation, any read or write operations occurring at that time will be able to complete without the use of the microcontroller clock. Once the operation is completed, the module will issue an interrupt according to the setting of the IRQM bits.

If the PMPIE bit is set, and its priority is greater than current CPU priority, the device will wake from Sleep or Idle mode and execute the PMP interrupt service routine.

If the assigned priority level of the PMP interrupt is less than or equal to the current CPU priority level, the CPU will not be awakened and the device will enter the Idle mode.

13.6.2 PMP Operation in Idle Mode

When the device enters Idle mode, the system clock sources remain functional. The PMCON<SIDL> bit selects whether the module will stop or continue functioning on Idle. If PMCON<SIDL> = 0, the module will continue operation in Idle mode.

If PMCON<SIDL> = 1, the module will stop communications when the microcontroller enters Idle mode, in the same manner as it does in Sleep mode. The current transaction in Slave modes will complete and issue an interrupt, while the current transaction in Master mode will be suspended until normal clocking resumes. As with Sleep mode, Idle mode should be avoided when using the module in Master mode if continuous use of the module is required.

Note:	The FRZ bit is readable and writable only when the CPU is executing in Debug Exception mode. In all other modes, the FRZ bit reads as '0'. If FRZ bit is changed during Debug mode, the new value does not take effect until the current Debug Exception mode is exited and re-entered. During the Debug Exception mode, the FRZ bit reads the state of the peripheral when entering Debug mode.

13.7 EFFECTS OF VARIOUS RESETS

13.7.1 Device Reset

All PMP module registers are forced to their reset states on a device Reset.

13.7.2 Power-on Reset

All PMP module registers are forced to their Reset states on a POR.

13.7.3 Watchdog Reset

All PMP module registers are forced to their reset states on a Watchdog reset.

13.8 PARALLEL MASTER PORT APPLICATIONS

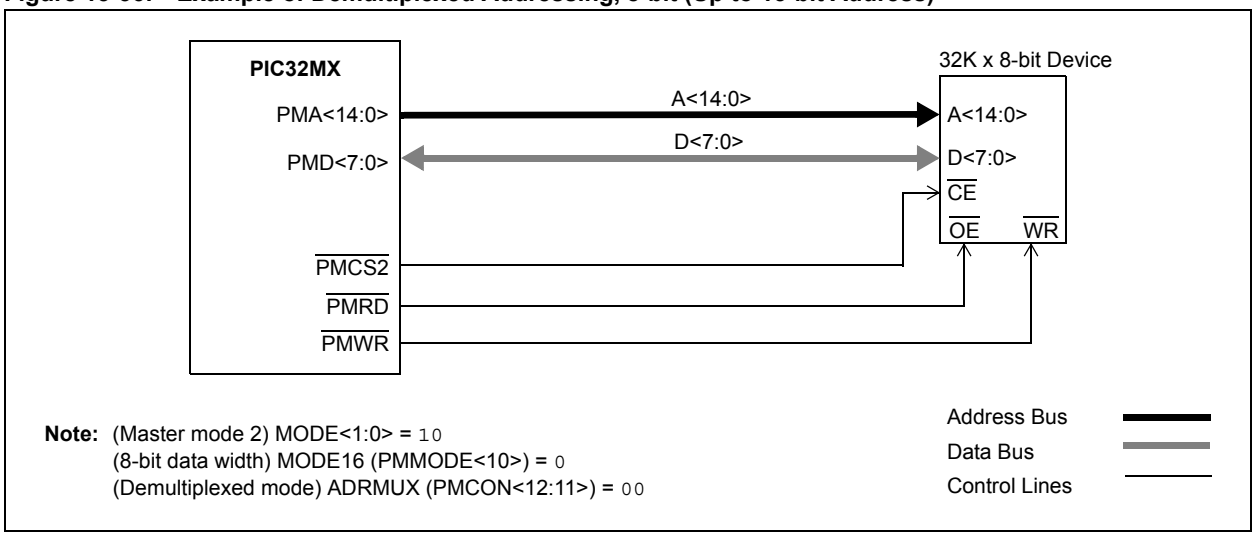
This section illustrates typical interfaces between the PMP module and external devices for each of the module's multiplexing modes. Additionally, there are some potential applications shown for the PMP module.

Note:	Data pins PMD<15:0> are available on 100-pin PIC32MX device variants and larger. For all other device variants, only pins PMD<7:0> are available. Refer to the specific PIC32MX device data sheet for details.

13.8.1 Demultiplexed Memory or Peripheral

Figure 13-36 illustrates the connections to an 8-bit memory or addressable peripheral in Demultiplexed mode. This mode does not require any external latches.

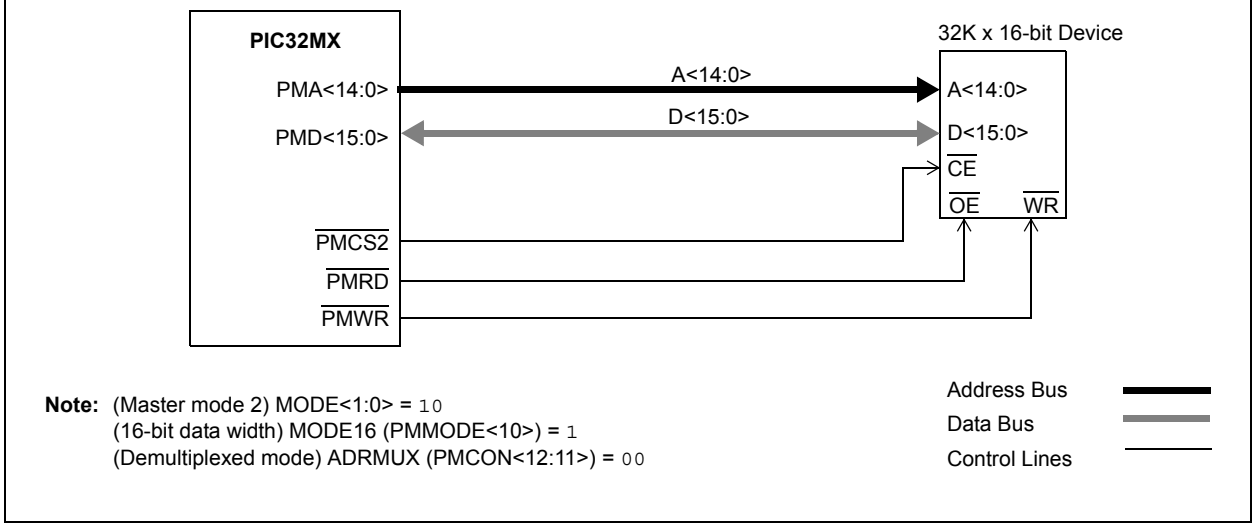
Figure 13-36: Example of Demultiplexed Addressing, 8-bit (Up to 15-bit Address)



Note: (Master mode 2) MODE<1:0> = 10
(8-bit data width) MODE16 (PMMODE<10>) = 0
(Demultiplexed mode) ADRMUX (PMCON<12:11>) = 00

Figure 13-37 illustrates the connections to a 16-bit memory or addressable peripheral in Demultiplexed mode. This mode does not require any external latches.

Figure 13-37: Example of Demultiplexed Addressing, 16-bit Data, (Up to 15-bit Address)

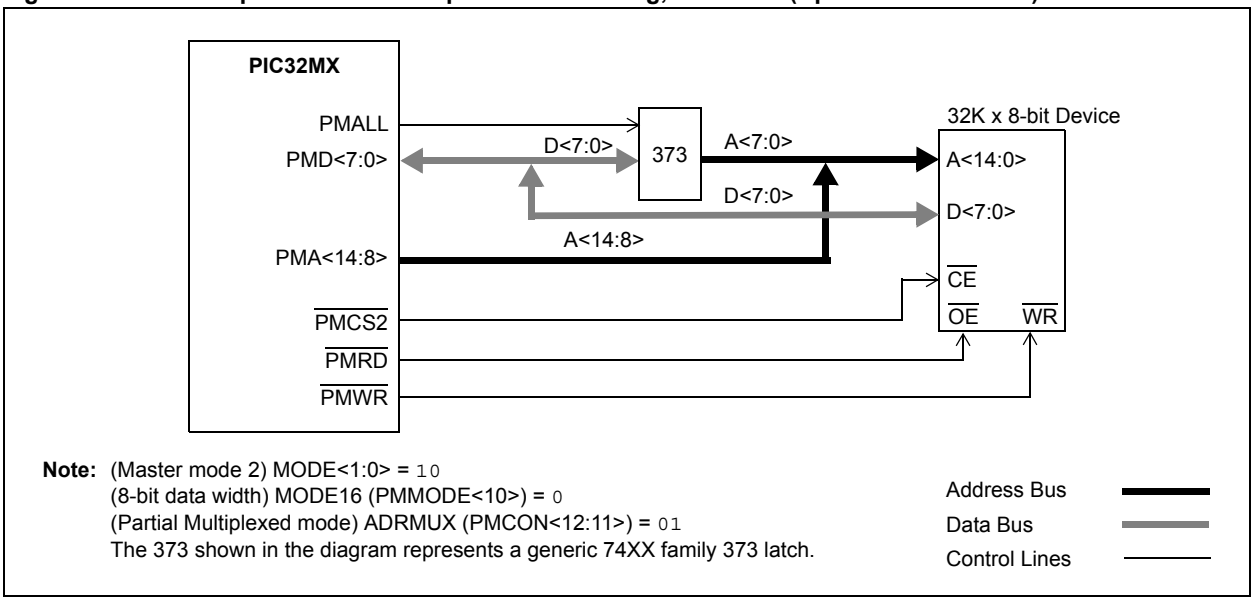


Note: (Master mode 2) MODE<1:0> = 10
(16-bit data width) MODE16 (PMMODE<10>) = 1
(Demultiplexed mode) ADRMUX (PMCON<12:11>) = 00

13.8.2 Partial Multiplexed Memory or Peripheral

Figure 13-38 illustrates the connections to an 8-bit memory or other addressable peripheral in Partial Multiplex mode. In this mode, an external latch is required. Consequently, from the microcontroller perspective, this mode achieves some pin savings over the Demultiplexed mode, however, at the price of performance. The lower 8 bits of the address are multiplexed with the PMD<7:0> data bus and require one extra peripheral bus clock cycle.

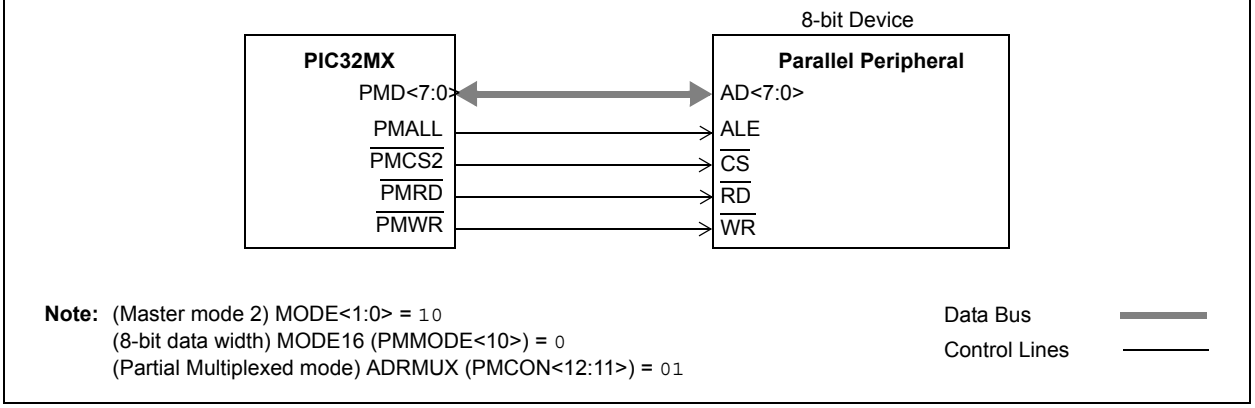
Figure 13-38: Example of Partial Multiplexed Addressing, 8-bit Data (Up to 15-bit Address)



Note: (Master mode 2) MODE<1:0> = 10
(8-bit data width) MODE16 (PMMODE<10>) = 0
(Partial Multiplexed mode) ADRMUX (PMCON<12:11>) = 01
The 373 shown in the diagram represents a generic 74XX family 373 latch.

If the peripheral has internal latches as shown in Figure 13-39, no extra circuitry is required except for the peripheral itself.

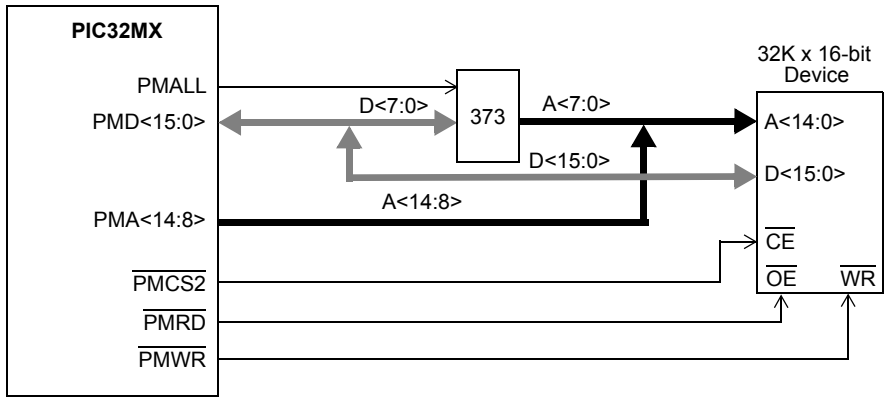
Figure 13-39: Example of Partial Multiplexed Addressing, 8-bit Data



Note: (Master mode 2) MODE<1:0> = 10
(8-bit data width) MODE16 (PMMODE<10>) = 0
(Partial Multiplexed mode) ADRMUX (PMCON<12:11>) = 01

Figure 13-40 illustrates the connections to a 16-bit memory or other addressable peripheral in Partial Multiplex mode. In this mode, an external latch is required. Consequently, from the microcontroller perspective, this mode achieves some pin savings over the Demultiplexed mode, however, at the price of performance. The lower 8 bits of address are multiplexed with the PMD<7:0> data bus and require one extra peripheral bus clock cycle.

Figure 13-40: Example of Partial Multiplexed Addressing,16-bit Data (Up to 15-bit Address)



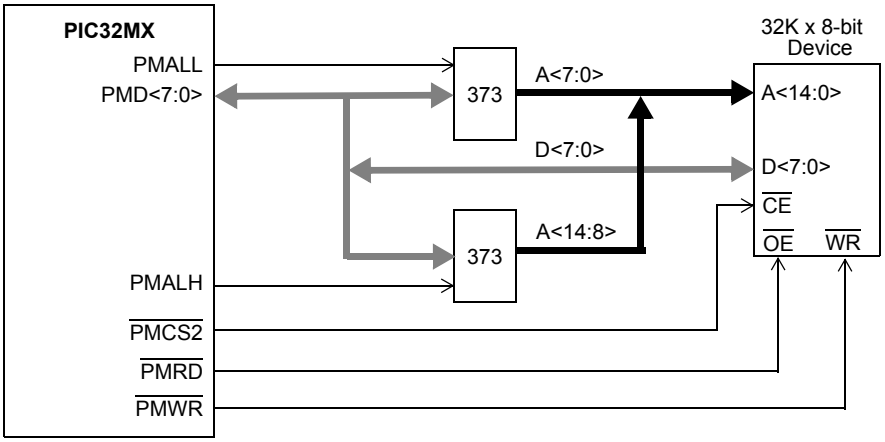
Note: (Master mode 2) MODE<1:0> = 10
(16-bit data width) MODE16 (PMMODE<10>) = 1
(Partial Multiplexed mode) ADRMUX (PMCON<12:11>) = 01
The 373 shown in the diagram represents a generic 74XX family 373 latch.

Address Bus
Data Bus
Control Lines

13.8.3 Full Multiplexed Memory or Peripheral

Figure 13-41 illustrates the connections to a memory or other addressable peripheral in full 8-bit Multiplexed mode, ADRMUX = 10 (PMCON<12:11>). Consequently, from the microcontroller perspective, this mode achieves the best pin saving over the Demultiplexed mode or Partially Multiplexed mode, however, at the price of performance. The lower 8 address bits are multiplexed with the PMD<7:0> data bus followed by the upper 6 or 7 address bits (if CS2, CS1 or both are enabled) and therefore require two extra peripheral bus clock cycles.

Figure 13-41: Fully Multiplexed Addressing, 8-bit Data (Up to 15-bit Address)



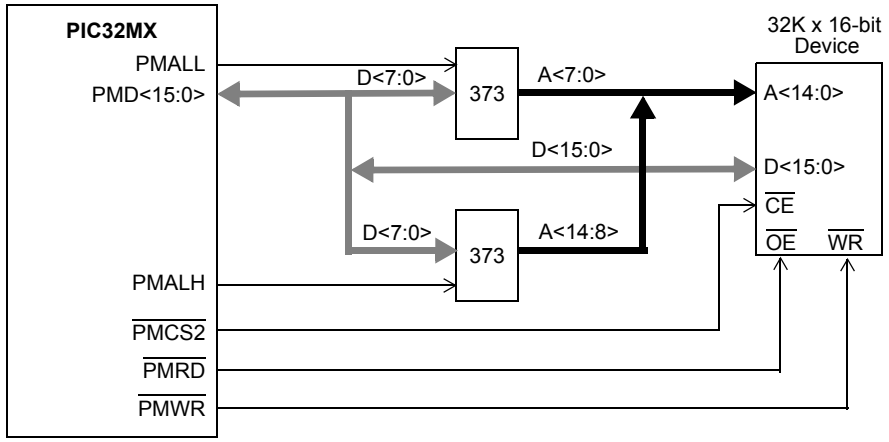
Note: (Master mode 2) MODE<1:0> = 10
(8-bit data width) MODE16 (PMMODE<10>) = 0
(Fully Multiplexed mode) ADRMUX (PMCON<12:11>) = 10
The 373 shown in the diagram represents a generic 74XX family 373 latch.

Address Bus
Data Bus
Control Lines

13 Parallel Master Port (PMP)

Figure 13-42 illustrates the connections to a 16-bit memory or other addressable peripheral in full 16-bit Multiplex mode, ADRMUX = 10 (PMCON<12:11>). Consequently, from the microcontroller perspective, this mode achieves the best pin saving over the Demultiplexed mode or Partially Multiplexed mode, however, at the price of performance. The lower 8 address bits are multiplexed with the PMD<7:0> data bus followed by the upper 6 or 7 address bits (if CS2, CS1 or both are enabled) and therefore require two extra peripheral bus clock cycles.

Figure 13-42: Fully Multiplexed Addressing, 16-bit Data (Up to 15-bit Address)

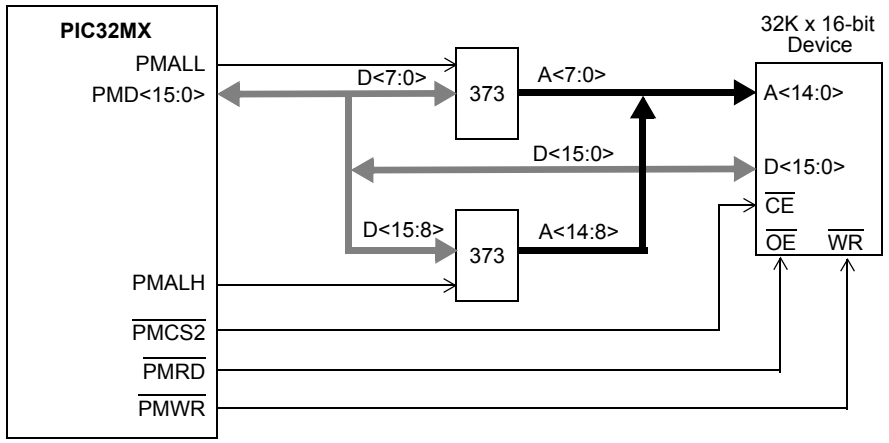


Note: (Master mode 2) MODE<1:0> = 10
(16-bit data width) MODE16 (PMMODE<10>) = 1
(Fully Multiplexed mode) ADRMUX (PMCON<12:11>) = 10
The 373 shown in the diagram represents a generic 74XX family 373 latch.

Address Bus
Data Bus
Control Lines

Figure 13-43 illustrates the connections to a 16-bit memory or other addressable peripheral in full 16-bit Multiplex mode, ADRMUX = 11 (PMCON<12:11>). Consequently, from the microcontroller perspective, this mode achieves the best pin saving over the Demultiplexed mode or Partially Multiplexed mode, however, at the price of performance. Compared to the previous Full Multiplex mode, ADRMUX = 10, this mode multiplexes 14 or 15 address bits (if CS2, CS1 or both are enabled) simultaneously with the PMD<15:0> bus and therefore requires only one extra peripheral bus clock cycle.

Figure 13-43: Example 2 of Full 16-bit Multiplexed Addressing, 16-bit Data (Up to 15-bit Address)



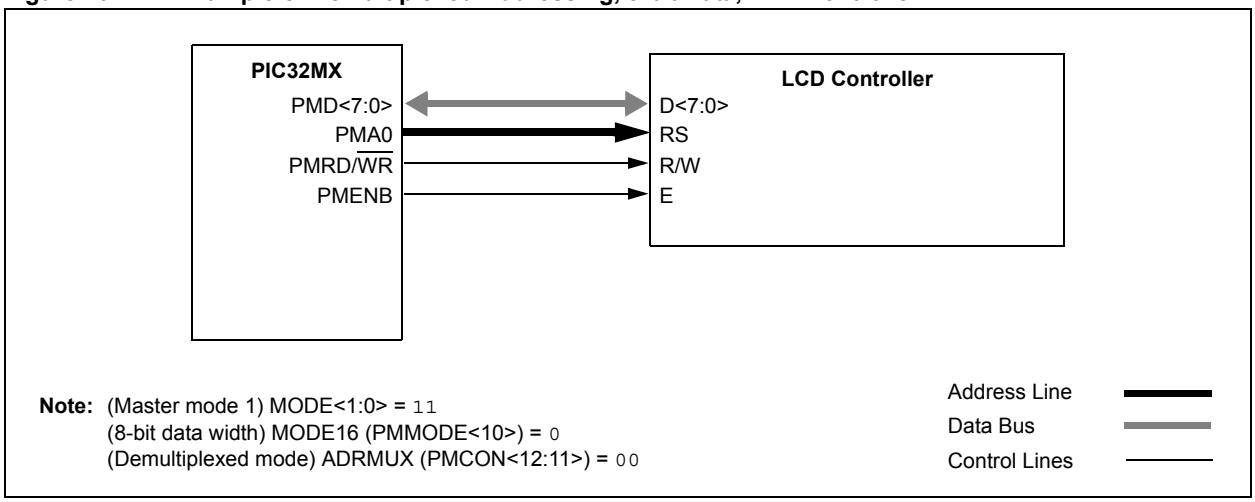
Note: (Master mode 2) MODE<1:0> = 10
(16-bit data width) MODE16 (PMMODE<10>) = 1
(Fully Multiplexed mode) ADRMUX (PMCON<12:11>) = 11
The 373 shown in the diagram represents a generic 74XX family 373 latch.

Address Bus
Data Bus
Control Lines

13.8.4 8-bit LCD Controller Example

The PMP module can be configured to connect to a typical LCD controller interface as shown in Figure 13-44. In this case, the PMP module is configured for Master mode 1, MODE<1:0> = 11 (PMMODE<9:8>), and uses active-high control signals since common LCD displays require active-high control.

Figure 13-44: Example of Demultiplexed Addressing, 8-bit Data, LCD Controller

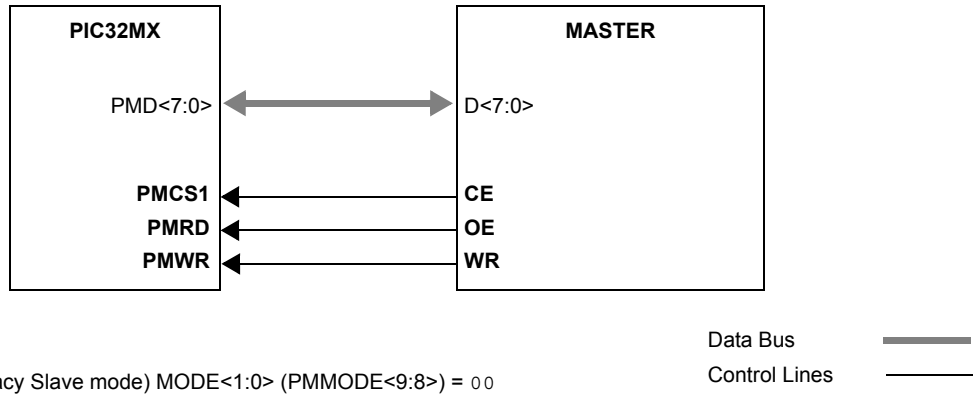


Note: (Master mode 1) MODE<1:0> = 11
(8-bit data width) MODE16 (PMMODE<10>) = 0
(Demultiplexed mode) ADRMUX (PMCON<12:11>) = 00

13.9 PARALLEL SLAVE PORT APPLICATION

Figure 13-45 illustrates the connections to a master peripheral in 8-bit data mode as a slave, MODE = 00 (PMMODE<9:8>). The microcontroller's PMP is controlled by a Chip Select (PMCS1).

Figure 13-45: Legacy Mode Slave Port



Note: (Legacy Slave mode) MODE<1:0> (PMMODE<9:8>) = 00

13.10 I/O PIN CONTROL

13.10.1 I/O Pin Resources

When enabling the PMP module for Master mode operations, the PMAEN register must be configured (set to '1') for the corresponding bits of PMA<15:0> I/O pins to be controlled by the PMP module. Those I/O pins not configured for use by the PMP module remain as general purpose I/O pins.

Table 13-12: Required I/O Pin Resources for Master Modes

I/O Pin Name	Demultiplex	Partial Multiplex	Full Multiplex	Functional Description
PMPCS2/PMA15	Yes(2)	Yes(2)	Yes(2)	PMP Chip Select 2/Address A15
PMPCS1/PMA14	Yes(2)	Yes(2)	Yes(2)	PMP Chip Select 1/Address A14
PMA<13:2>	Yes(2)	Yes(3)	No(1)	PMP Address A13..A2
PMA1/PALH	No(1)	No(1)	Yes(4)	PMP Address A1/Address Latch High
PMA0/PALL	No(1)	Yes(3)	Yes(4)	PMP Address A0/Address Latch Low
PMRD/PMWR	Yes	Yes	Yes	PMP Read/Write Control
PMWR/PMENB	Yes	Yes	Yes	PMP Write/Enable Control
PMD<15:0>(6)	Yes(5)	Yes(5)	Yes(5)	PMP Bidirectional Data Bus D15...D0

Note 1: "No" indicates the pin is not required and is available as a general purpose I/O pin when the corresponding PMAEN bit is cleared = 0.

2: Depending on the application, not all PMA<15:0> or CS2, CS1 may be required.

3: When Partial Multiplex mode is selected (ADDRMUX<1:0> = 01), the lower 8 address lines are multiplexed with PMD<7:0>, PMA<0> becomes (ALL) and PMA<7:1> are available as general purpose I/O pins.

4: When Full Multiplex mode is selected (ADDRMUX<1:0> = 10 or 11), all 16 address lines are multiplexed with PMD<15:0>, PMA<0> becomes (ALL), PMA<1> becomes (ALH) and PMA<13:2> are available as general purpose I/O pins.

5: If MODE16 = 0, then only PMD<7:0> are required. PMD<15:8> are available as general purpose I/O pins.

6: Data pins PMD<15:0> are available on 100-pin PIC32MX device variants and larger. For all other device variants, only pins PMD<7:0> are available. Refer to the specific PIC32MX device data sheet for details.

When enabling any of the PMP module for Slave mode operations, the PMPCS1, PMRD, PMWR control pins and PMD<7:0> data pins are automatically enabled and configured. The user is, however, responsible for selecting the appropriate polarity for these control lines.

Table 13-13: Required I/O Pin Resources for Slave Modes

I/O Pin Name	Legacy	Buffered	Enhanced	Functional Description
PMPCS1/PMA14	Yes	Yes	Yes	Chip Select
PMA1/PALH	No(1)	No(1)	Yes	Address A1
PMA0/PALL	No(1)	No(1)	Yes	Address A0
PMRD/PMWR	Yes	Yes	Yes	Read Control
PMWR/PMENB	Yes	Yes	Yes	Write Control
PMD<15:0>	Yes(2)	Yes(2)	Yes(2)	Bidirectional Data Bus D7...D0

Note 1: "No" indicates the pin is not required and is available as a general purpose I/O pin when the corresponding PMAEN bit is cleared = 0.

2: Slave modes use PMD<7:0> only pins. PMD<15:8> are available as general purpose I/O pins. Control bit MODE16 (PMMODE<10>) is ignored.

13.10.2 I/O Pin Configuration

The following table provides a summary of the settings required to enable the I/O pin resources used with this module. The PMAEN register controls the functionality of pins PMA<15:0>. Setting any PMAEN bit = 1 configures the corresponding PMA pin as an address line. Those bits set = 0 remain as general purpose I/O pins.

Table 13-14: I/O Pin Configuration

		Required Settings for Module Pin Control					
I/O Pin Name	**Required(1)**	**Module Control**	**Bit Field**	**TRIS**	**Pin Type**	**Buffer Type**	**Description**
PMPCS2/PMA15	Yes	ON	CSF<1:0>, CS2, PTEN15	—	O	CMOS	PMP Chip Select 2/ Address A15
PMPCS1/PMA14	Yes	ON	CSF<1:0>, CS1 PTEN14	—	O	CMOS	PMP Chip Select 1/ Address A14
PMA<13:2>	Yes	ON	PTEN<13:2>	—	O	CMOS	PMP Address A13 .. A2
PMA1/PALH	Yes	ON	PTEN<1>	—	I(2), O	CMOS	PMP Address A1/ Address Latch High
PMA0/PALL	Yes	ON	PTEN<0>	—	I(2), O	CMOS	PMP Address A0/ Address Latch Low
PMRD/PM$\overline{WR}$	Yes	ON	PTRDEN	—	O	CMOS	PMP Read/Write Control
PMWR/PMENB	Yes	ON	PTWREN	—	O	CMOS	PMP Write/Enable Control
PMD<15:0>	Yes	ON	MODE16, ADRMUX<1:0>	—	I(2), O	CMOS	PMP Bidirectional Data Bus D15 ... D0

Legend: CMOS = CMOS-compatible input or output ST = Schmitt Trigger input with CMOS levels
I = Input O = Output

Note 1: Depending on the PMP mode and the user's application, these pins may not be required. If not enabled, these pins can be used for general purpose I/O.

2: Input buffers can be Schmitt Trigger or TTL.

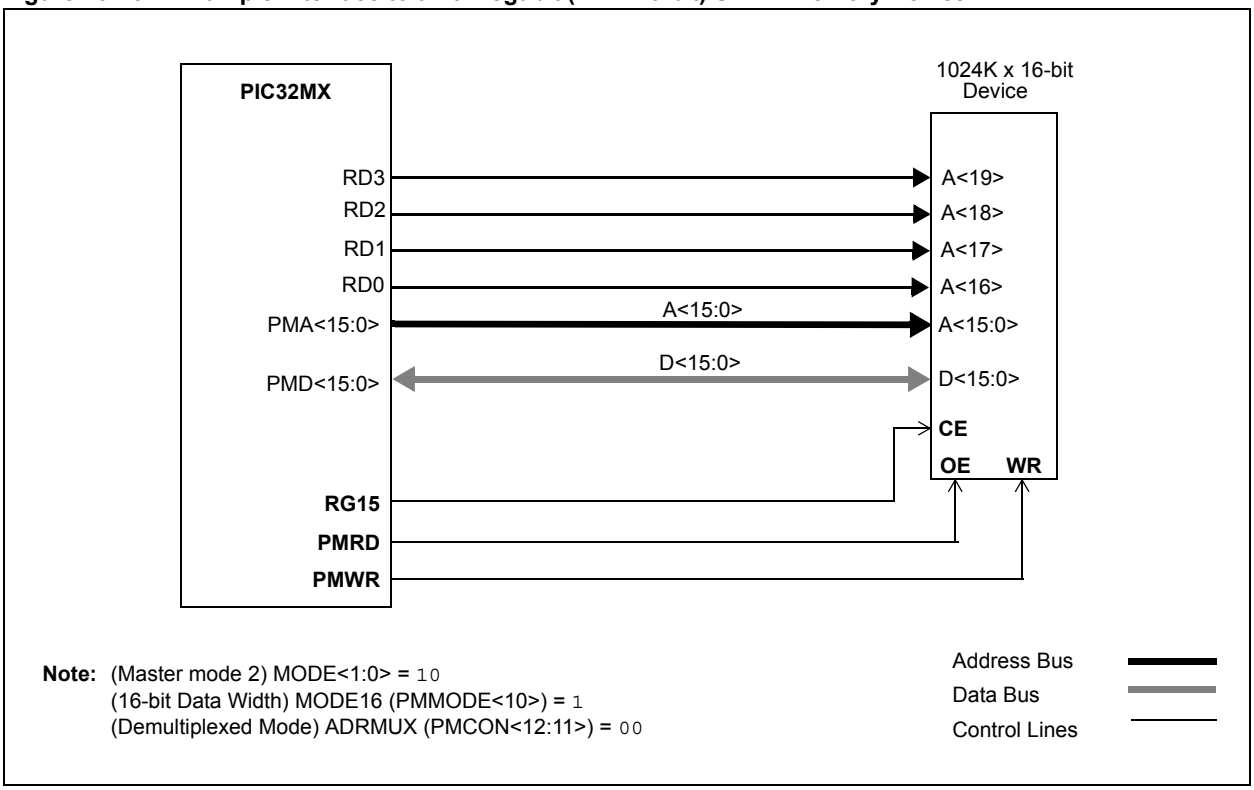
13.11 DESIGN TIPS

Question 1: ***Is it possible for the PMP module to address memory devices larger than 64K?***

Answer: Yes; however, not directly under the control of the PMP module. When using the PMCS2 or PMCS1 Chip Select pins, the addressable range is limited to 16K or 32K locations, depending on the Chip Select pin being used. Disabling PMCS2 and PMCS1 as Chip Selects allows these pins to function as address lines PMA15 and PMA14, increasing the range to 64K addressable locations. A dedicated I/O pin is required to function as the Chip Select and the user's software must now control the function of this pin.

To interface to memory devices larger than 64K, use additional available I/O pins as the higher order address lines A16, A17, A18, etc., as shown in Figure 13-46.

Figure 13-46: Example Interface to a 16 Megabit (1M x 16-bit) SRAM Memory Device



Note: (Master mode 2) MODE<1:0> = 10
(16-bit Data Width) MODE16 (PMMODE<10>) = 1
(Demultiplexed Mode) ADRMUX (PMCON<12:11>) = 00

Question 2: ***Is it possible to execute code from an external memory device connected to the PMP module?***

Answer: No. Due to the architecture of the PMP module, this is not possible. Only data can be read or written through the PMP.

13.12 RELATED APPLICATION NOTES

This section lists application notes that are related to this section of the manual. These application notes may not be written specifically for the PIC32MX device family, but the concepts are pertinent and could be used with modification and possible limitations. The current application notes related to the Parallel Master Port (PMP) module are:

Title	Application Note #
No related application notes at this time.	N/A

Note: Please visit the Microchip web site (www.microchip.com) for additional Application Notes and code examples for the PIC32MX family of devices.

13.13 REVISION HISTORY

Revision A (August 2007)

This is the initial released version of this document.

Revision B (October 2007)

Updated document to remove Confidential status.

Revision C (April 2008)

Revised status to Preliminary; Revised U-0 to r-x; Revised Table 13-10; Revised Section 13.3.1.6 and Section 13.3.8; Revised Register 13-5; Revised Figures 13-11, 13-37, 13-40, 13-41, 13-42, 13-43, 13-46; Revised Timing Diagram text for Figures 13-16, 13-18, 13-19.

Revision D (June 2008)

Revised Register 13-1, add note to FRZ; Revised Figures 13-4, 13-6, 13-8, 13-10, 13-36, 13-37, 13-38, 13-45; Revised Table 13-6; Revised Examples 13-6 and 13-7; Change Reserved bits from “Maintain as” to “Write”; Added Note to ON bit (PMCON Register).

Revision E (October 2009)

This revision includes the following updates:

- Minor updates to text and formatting have been implemented throughout the document
- Added the following item to the key feature list: Schmitt Trigger or TTL input buffers (see **13.1 “Introduction”**)
- Interrupts Register Summary (Table 13-1):
  - Removed all references to the Clear, Set and Invert registers
  - Added the Address Offset column
  - Added Notes 1, 2 and 3, which describe the Clear, Set and Invert registers
- Added Notes 1, 2 and 3, which describe the Clear, Set and Invert registers to the following registers
  - PMCON: Parallel Port Control Register (see Register 13-1)
  - PMMODE: Parallel Port Mode Register (see Register 13-2)
  - PMADDR: Parallel Port Address Register (see Register 13-3)
  - PMDOUT: Parallel Port Data Output Register (see Register 13-4)
  - PMDIN: Parallel Port Data Input Register (see Register 13-5)
  - PMAEN: Parallel Port Pin Enable Register (see Register 13-6)
  - PMSTAT: Parallel Port Status Register (Slave modes only) (see Register 13-7)
- Removed all references to Interrupt registers (IEC1, IFS1 and IPC7)
- Added a shaded note to **13.4.1.4 “Legacy Mode Interrupt Operation”**
- Updated the 2-3 TPBCLK cycles duration in Figure 13-32, Figure 13-33, Figure 13-34 and Figure 13-35
- Added Note 2 to the I/O Pin Configuration table (Table 13-14)

NOTES: