Chapter 6

Parallel I/O

CHAPTER 6. PARALLEL I/O

CONTENTS

6. PARALLEL I/O

The *WE* DSP16/DSP16A Digital Signal Processor has a 16-bit parallel I/O interface with external devices for rapid transfer of data. Access times are programmable via the strobe field in the pioc register. Minimal or no additional logic is required to interface with memory or other peripheral devices. Five maskable interrupts are included in the PIO unit.

Although there is only one physical PIO port, there are two logical PIO ports: pdx0 and pdx1. The two logical ports are distinguished by the state of the peripheral select line (PSEL). Figure 6-1 shows the DSP16/DSP16A PIO unit at the block level.

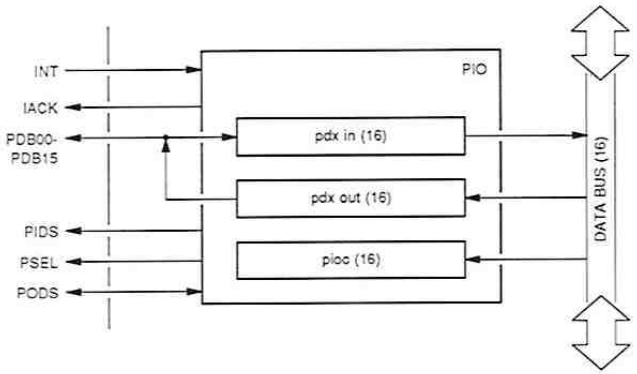

Figure 6-1. Parallel I/O Section

6.1 SOFTWARE DESCRIPTION

The parallel I/O port can be accessed via the data move group of instructions. The two logical ports (pdx0 and pdx1) correspond to the state of the PSEL pin (logical 0 or logical 1, respectively). That is, an access to the logical port pdx0 initiates a transaction with the PSEL signal cleared (0), while an access to logical port pdx1 initiates a transaction with the PSEL signal set (1).

When programming the device, three PIO registers can be referenced:

- pioc – Parallel I/O control register.
- pdx0 – Logical port 0.
- pdx1 – Logical port 1.

Note: pdx0 and pdx1 both reference the same physical register.

The parallel I/O control (pioc) register is a 16-bit, user-accessible register used to configure some features of the parallel I/O:

- External device access time.
- Interrupt masks.
- Active/passive mode.
- Status/control (S/C) bit mode.

Table 6-1 shows the pioc register.

Table 6-1 Parallel I/O Control (pioc) Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	IBF	STROBE		PODS	PIDS	S/C	INTERRUPTS					STATUS				

Field	Value	Result/Description
IBF	R	IBF interrupt status bit (same as bit 4).
STROBE	00 01 10 11	Strobe width of PODS PIDS T* T 2T 2T 3T 3T 4T 4T
PODS	0	PODS is an input (passive mode).
	1	PODS is an output (active mode).
PIDS	0	PIDS is an input (passive mode).
	1	PIDS is an output (active mode).
S/C	0	Not S/C mode.
	1	S/C mode.
INTERRUPTS	Wxxxx	IBF interrupt enabled when set.
	xWxxx	OBE interrupt enabled when set.
	xxWxx	PIDS interrupt enabled when set.
	xxxWx	PODS interrupt enabled when set.
	xxxxW	INT interrupt enabled when set.
STATUS	Rxxxx	IBF status bit.
	xRxxx	OBE status bit.
	xxRxx	PIDS status bit.
	xxxRx	PODS status bit.
	xxxxR	INT status bit.

* T = 2 × tCKIHCKIH.

6.1.1 pioc Register Settings

From the system perspective, the DSP16/DSP16A device can be an active device (i.e., initiates transactions on the parallel data bus, PDB), or a passive device (i.e., receives stimulus from external devices). In the active mode, the DSP16/DSP16A device behaves as a bus master; in the passive mode of operation, the DSP16/DSP16A device behaves as a peripheral to another device, such as a microprocessor or another DSP16/DSP16A device. Bit 15 of the pioc register is the same as bit 4. See the following description of bit 4.

Bits 14 and 13 of the pioc register control the duration of assertion of the PIDS and PODS signals. This will be described in more detail in Section 6.2.2.

Bit 12 of the pioc register, when equal to logic 1, makes the PODS pin an output; hence, the DSP16/DSP16A device can perform active mode write transactions to external devices. When bit 12 of the pioc register is equal to logic 0, the PODS pin is an input used for passive reads from the DSP16/DSP16A device by external devices.

Bit 11 of the pioc register, when equal to logic 1, makes the PIDS pin an output; hence, the DSP16/DSP16A device can perform active mode read transactions from external devices. When bit 11 of the pioc register is equal to logic 0, the PIDS pin is an input used for passive writes to the DSP16/DSP16A device.

Note: The external PSEL pin always reflects the last access of pdx0 or pdx1 by the DSP16/DSP16A program. PSEL is unaffected by the selection of active or passive PIDS or PODS.

Bit 10 of the pioc register, when equal to logic 1, places the PIO in status and control (S/C) mode. In the S/C mode, the upper half of the parallel data bus (PDB15—PDB08) becomes an input only; the lower half (PDB07—PDB00) remains bidirectional (see Section 6.2.4).

Bits 9, 8, 7, 6, and 5 of the pioc register are used to mask interrupts. There are five types of interrupts that can occur:

- Interrupts caused by an external device writing to the DSP16/DSP16A device's serial port. This type of interrupt is masked when bit 9 of the pioc register is set to logic 0.
- Interrupts caused by an external device reading from the DSP16/DSP16A device's serial port. This type of interrupt is masked when bit 8 of the pioc register is set to logic 0.
- Interrupts caused by an external device writing to the DSP16/DSP16A device's parallel port when the DSP16/DSP16A device is in passive mode. This type of interrupt is masked when bit 7 of the pioc register is set to logic 0.
- Interrupts caused by an external device reading from the DSP16/DSP16A device's parallel port when the DSP16 device is in passive mode. This type of interrupt is masked when bit 6 of the pioc register is set to logic 0.
- Interrupts caused by an external device asserting the INT pin. This type of interrupt is masked when bit 5 of the pioc register is set to logic 0.

Note: There is one instruction latency when altering the INTERRUPTS field of the pioc register. For example, if interrupts are disabled with the command pioc=0x00, the DSP16/DSP16A device still responds to an interrupt during the next instruction. After this instruction has executed, the interrupts are disabled. Therefore, to protect an instruction sequence from interrupts, follow the command to mask the INTERRUPTS field of the pioc register with one instruction that may be safely interrupted.

Bits 4, 3, 2, 1, and 0 of the pioc register indicate the serial and parallel I/O buffers and the interrupt pin. This portion of the pioc register can be used to determine which of the five aforementioned interrupts are requesting service. These bits can be read by an interrupt service routine to determine which interrupt(s) have occurred and, hence, how to proceed to service the interrupt request. These status bits can also be used to perform programmed I/O by polling some condition when necessary. It is important to note that pending interrupt status bits are cleared under the following conditions:

- pioc[4], which indicates that the serial I/O input buffer is full, is cleared by reading from the sdx (serial I/O) register.
- pioc[3], which indicates that the serial output buffer is empty, is cleared when a write to the sdx (serial I/O) register is performed.
- pioc[2], which indicates that an external device has written into the DSP16/DSP16A device's pio register, is cleared when the DSP16/DSP16A device reads from the PIO register (either pdx0 or pdx1). Reading from the PIO register clears this bit irrespective of whether it is done inside or outside an interrupt routine. Note that this interrupt can occur only when the DSP16/DSP16A device is in the passive mode; hence, the DSP16/DSP16A device reading from the PIO registers (to clear pioc[2]) does NOT cause an external read transaction to take place.
- pioc[1], which indicates that an external device has read from the DSP16/DSP16A device's PIO register, is cleared when the DSP16/DSP16A device writes to the PIO register (either pdx0 or pdx1). Writing to the PIO register clears this bit irrespective of whether it is done inside or outside an interrupt routine. Note that this interrupt can occur only when the DSP16/DSP16A device is in the passive mode; hence, writing to the PIO registers (to clear pioc[1]) does NOT cause an external write transaction to take place.
- pioc[0], which indicates that an external device has asserted the INT signal, is cleared when the interrupt acknowledge (IACK) signal makes a high-to-low transition, indicating that the interrupt service routine has completed. If external interrupts are masked, this bit will NOT be set when INT is asserted. Recall that this bit can be cleared only when an iretum instruction causes the high-to-low transition of IACK.

Note: The contents of the pioc register are cleared, except bit 3 which is set, when the RSTB signal is asserted. Hence, the DSP16/DSP16A device is in passive mode, non-status and control (S/C) bit mode, with all interrupts masked after a chip reset.

www.DataSheet4U.com

6.1.2 Latent Reads

While in active mode, reading from a logical PIO port is accomplished by an actual read of the single physical port on the DSP16/DSP16A device. When a read of the parallel input register (physical port) is performed, a transaction to the external system is performed on the logical port. Reads from the logical port imply that:

- All reads take their data from the on-chip parallel input register.
- As data is read from the internal parallel input register, a read transaction to the external system is initiated.
- Upon completion of the external read transaction, data received from the external system (logical port 0 or 1) is loaded into the parallel input register.

Since data is read from the internal parallel input register and then new data is accepted into the parallel input register from a logical port, reads from the external system are latent. For example, to read a string of four words of data (d0, d1, d2, d3) from the PIO port, the following actions are required:

1. The first instruction reads meaningless data from the parallel input register and initiates the transaction to bring the first datum (d0) from the external device.
2. The second instruction reads the first datum (d0) from the parallel input register and initiates the transaction to bring the second datum (d1) from the external device.
3. The third instruction reads the second datum (d1) from the parallel input register and initiates the transaction to bring the third datum (d2) from the external device.
4. The fourth instruction reads the third datum (d2) from the parallel input register and initiates the transaction to bring the fourth datum (d3) from the external device.
5. The fifth and final instruction reads the fourth datum (d3) from the parallel input register and initiates a transaction that reads another word of data from the external device and overwrites the last datum (d3) in the parallel input register.

To fetch a vector of data of length N requires N+1 instructions and generates N+1 read transactions to the external system. In order to fetch a single datum that is not already present in the parallel input register, two instructions are required. Since both logical ports map into the same physical port, a fetch from either logical port takes data from the parallel input register; the external access then overwrites the contents of the parallel input register with the data from the logical port specified in the instruction.

The parallel output register is distinct from the parallel input register. Writing to pdx0 and pdx1 does not alter the contents of the parallel input register.

6.2 HARDWARE DESCRIPTION

The parallel I/O bus is an asynchronous interface. Data strobes indicate when data may be put on the bus during active mode reads and when data is available on the bus during active mode writes. The PIO port characteristics are programmable and are controlled by the parallel I/O control register (pioc). The following sections describe the PIO signals and their functions.

6.2.1 Parallel I/O Signals

Table 6-2 describes the PIO signals of the DSP16/DSP16A device.

Table 6-2. Parallel I/O Signals

Symbol	Type*	Pin	Name/Description
PDB00	I/O†	11	**Parallel Data Bus – Bit 0.**
PDB01		10	**Parallel Data Bus – Bit 1.**
PDB02		9	**Parallel Data Bus – Bit 2.**
PDB03		8	**Parallel Data Bus – Bit 3.**
PDB04		5	**Parallel Data Bus – Bit 4.**
PDB05		4	**Parallel Data Bus – Bit 5.**
PDB06		3	**Parallel Data Bus – Bit 6.**
PDB07		2	**Parallel Data Bus – Bit 7.**
PDB08		84	**Parallel Data Bus – Bit 8.**
PDB09		83	**Parallel Data Bus – Bit 9.**
PDB10		82	**Parallel Data Bus – Bit 10.**
PDB11		81	**Parallel Data Bus – Bit 11.**
PDB12		78	**Parallel Data Bus – Bit 12.**
PDB13		77	**Parallel Data Bus – Bit 13.**
PDB14		76	**Parallel Data Bus – Bit 14.**
PDB15		75	**Parallel Data Bus – Bit 15.**
PSEL	O†	72	**Peripheral Select.** PSEL is used to specify the logical port to/from which data is to be conveyed. PSEL is high (logic 1) when pdx1 is the register specified in the I/O instruction and low when pdx0 is the register specified.

* I = Input; O = Output.
† 3-stated.

www.DataSheet4U.com

Table 6-2. Parallel I/O Signals (continued)			
Symbol	Type*	Pin	Name/Description
PIDS	I/O†	73	**Parallel Input Data Strobe (Active-Low).** In active mode, PIDS is an output. when PIDS is asserted, data may be placed onto the PDB. Upon negation of PIDS, data should be removed from the PDB. PIDS is asserted by the DSP16/DSP16A device during active mode read transaction. In passive mode, PIDS is an input. When asserted by an external device, this signal indicates that data is available on the PDB. In both passive and active modes, the trailing edge (low-to-high transition) of PIDS is the sampling point.
PODS	I/O†	74	**Parallel Output Data Strobe (Active-Low).** In active mode, PODS is an output. When PODS is asserted, data is available on the PDB. PODS is asserted by the DSP16/DSP16A device during an active mode write transaction. In passive mode, PODS is an input. When PODS is asserted by an external device, the DSP16/DSP16A device places the contents of its parallel output register (pdx0 or pdx1) onto the PDB.

* I = Input; O = Output.
† 3-stated.

6.2.2 Active Mode

The duration of parallel I/O strobe signals can be programmed using bits 14 and 13 of the pioc register. Table 6-3 shows the possible configurations.

Table 6-3. PIO Strobe Widths			
pioc Bits		Strobe width	
14	13	PIDS*	PODS*
0	0	T	T
0	1	2T	2T
1	0	3T	3T
1	1	4T	4T

* T = 2 x tCKIHCKIH

When minimum strobe widths are configured, parallel I/O transactions can occur at the instruction rate. Consecutive parallel I/O instructions can be executed.

Note: When the strobe widths are not minimum, consecutive PIO instructions are prohibited – other non-PIO instructions must be placed between two PIO instructions.

- When pioc[14, 13] = 01, one or more instructions must be placed between PIO instructions.
- When pioc[14, 13] = 10, two or more instructions must be placed between PIO instructions.
- When pioc[14, 13] = 11, three or more instructions must be placed between PIO instructions.

Any interrupt service routine must guarantee that these conditions are met. As a simple rule, when pioc[14, 13] = 11, the first instruction in an interrupt service routine cannot be a PIO instruction.

6.2.3 Passive Mode

In passive mode, the DSP16/DSP16A device can be used as a peripheral to such other devices as a microprocessor. Bits 12 and 11 of the pioc register are used to configure the passive mode. When bit 12 of the pioc register is clear (0), the PODS signal becomes an input and the contents of the DSP16/DSP16A device's parallel output register can be read by an external device asserting PODS. When bit 11 of the pioc register is clear (0), PIDS is an input and the DSP16/DSP16A device's parallel input register can be written by an external device asserting PIDS.

Providing that their respective interrupt mask bits are set (logic 1), the assertion of PIDS and PODS by an external device causes the DSP16/DSP16A device to recognize an interrupt. This mechanism is a means by which functional synchronization between the DSP16/DSP16A device and an external device can be achieved.

6.2.4 Status and Control (S/C) Mode

The (S/C) mode of the DSP16/DSP16A device's parallel I/O is invoked by setting bit 10 of the pioc register to logic 1. When this bit is set, the upper half of the parallel data bus (PDB15—PDB08) becomes an input only. The lower half of the parallel data bus (PDB07—PDB00) remains bidirectional. The lower half of the parallel data bus can be used as an 8-bit, bidirectional port that behaves just as the 16-bit port does in both the active and passive modes. The upper half of the parallel data bus, being an input only, can perform active and passive mode input transactions.

The S/C bit mode may be used as follows:

- The upper bits (PDB15—PDB08) can be used by devices external to the DSP16/DSP16A device to control the behavior of the DSP16/DSP16A. The DSP16/DSP16A device can be programmed to poll the parallel port and to respond according to the data there. An external device can use the upper eight bits of the parallel I/O (in the passive mode) to write a control byte into a DSP16/DSP16A device. Data on this upper byte must be clocked (using PIDS) into the parallel I/O port. Furthermore, the DSP16/DSP16A device should have its PODS signal configured in the passive mode (bit 12 of pioc clear).

- The lower eight bits of the parallel data bus (PDB07—PDB00) can be used by external devices to indicate some condition (status) internal to the DSP16/DSP16A device. The DSP16/DSP16A device, by writing to the pdx register enables these eight bits when PODS is asserted.

6.3 INTERRUPTS AND THE PARALLEL I/O

There are two internal interrupts generated, based on parallel I/O events. An internal interrupt is generated (provided it is unmasked) when an external device performs a passive mode write. When the external device negates the PIDS signal (low-to-high transition), an internal interrupt request is generated. When the DSP16/DSP16A device accepts this interrupt request, the IACK signal is asserted. When the interrupt routine is completed, IACK is negated (becomes logic 0). See Section 2.8 for more information on how the DSP16/DSP16A device reacts to interrupts.

Similarly, when an external device performs a passive read, an internal interrupt request is generated upon negation (low-to-high transition) of PODS. When the DSP16/DSP16A device accepts this interrupt request, the IACK signal is asserted. When the interrupt routine has completed, IACK is negated (becomes logic 0).

When the DSP16/DSP16A device is in the passive mode, program synchronization can be obtained by using the interrupt mechanism to synchronize a data source, with the program being run by the DSP16/DSP16A device. Briefly, the DSP16/DSP16A device can have its parallel I/O configured in the passive mode. A data source provides data to the DSP16/DSP16A device via passive writes. During the associated interrupt routine, the DSP16/DSP16A device program performs I/O functions. The receipt of data by the DSP16/DSP16A device is indicated to the external data source by the falling edge (high-to-low) transition of the IACK signal.

When the PIDS signal is active, the **pdx in** register is shadowed during interrupts. This allows the parallel input to be used during interrupts without the possibility of destroying data previously fetched via a latent PIO read. When the interrupt service routine is exited, **pdx in** is loaded with its value prior to the interrupt. If the parallel input is changed from active to passive during the interrupt, the shadowing feature is disabled.

6.4 PIO BUS TRANSACTIONS

In this section, the four types of parallel I/O transactions are described:

- Active mode input (active read)
- Active mode output (active write)
- Passive mode input (passive read)
- Passive mode output (passive write)

Note: For timing information, refer to the appropriate data sheet.

6.4.1 Active Mode Input

The active mode input transaction (shown in Figure 6-2) is initiated by the DSP16/DSP16A device executing a data move instruction (e.g., *r2 = pdx0). When an active mode input occurs, PIDS and PSEL are asserted, indicating that an external device can place data on the parallel data bus (PDB). PSEL can be used to select one of two external sources for the data. The external device must place valid data on the PDB before the negation of PIDS. The access time for the external data source is configurable via the pioc register. The external data source can remove data from the PDB after negation of PIDS.

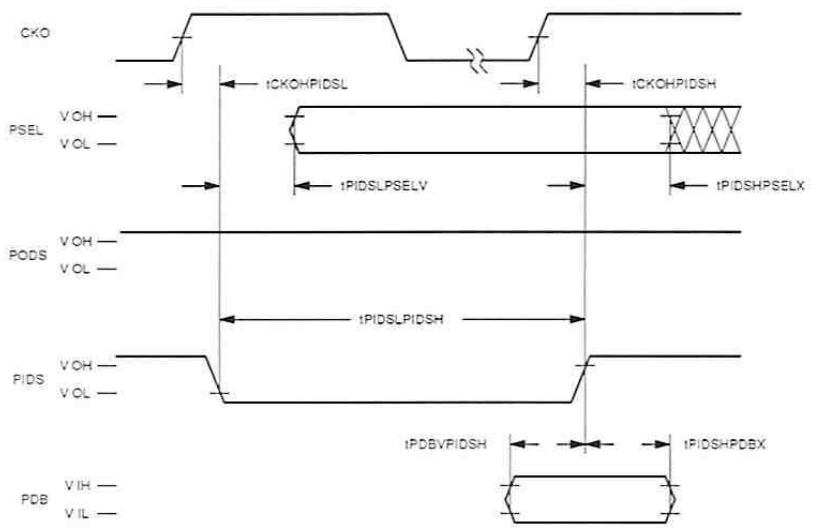

Figure 6-2. Active Mode Input Timing

www.DataSheet4U.com

6.4.2 Active Mode Output

The active mode output transaction (shown in Figure 6-3) is initiated by the DSP16/DSP16A device executing a data move instruction (e.g., pdx0 = *r2). When an active mode output occurs, PODS and PSEL are asserted. A short time later, the DSP16/DSP16A device places data onto the PDB. This data remains valid until a short time after the negation of PODS. This write time is configurable via the pioc register.

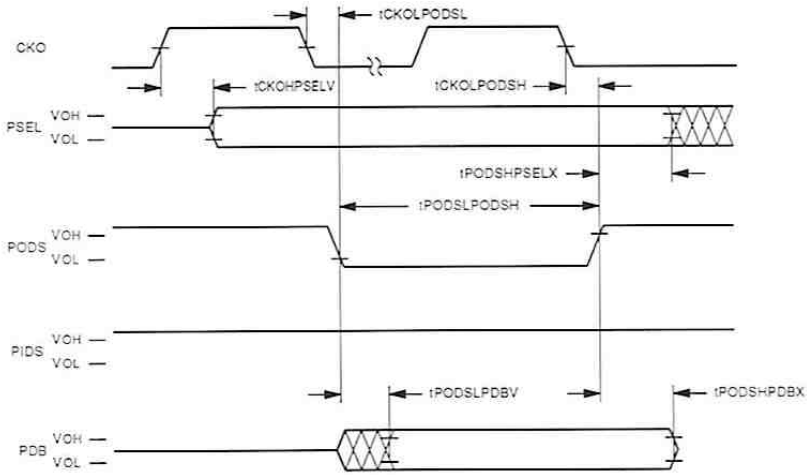

Figure 6-3. Active Mode Output Timing

6.4.3 Passive Mode Input

The passive mode input transaction (shown in Figure 6-4) is initiated by an external device asserting PIDS. The external device must place data onto the PDB prior to the negation of PIDS. The external device may remove the data from the bus after the negation of PIDS.

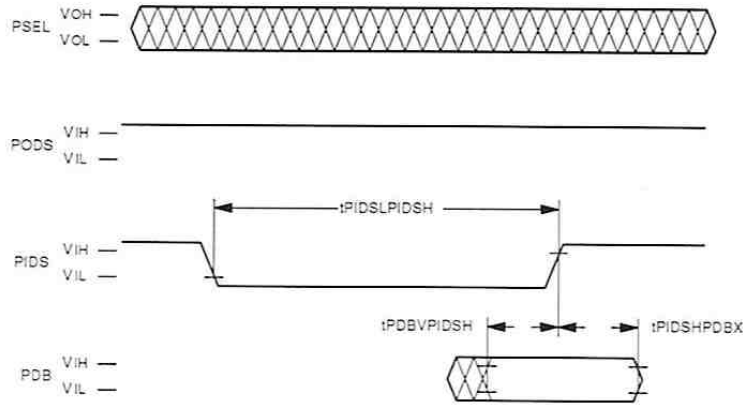

Figure 6-4. Passive Mode Input Timing

www.DataSheet4U.com

6.4.4 Passive Mode Output

The passive mode output transaction (shown in Figure 6-5) is initiated by an external device asserting PODS. A short time later, the DSP16/DSP16A device places data from its pdx output register onto the PDB. The data remains valid until a short time after the negation of PODS by the external device.

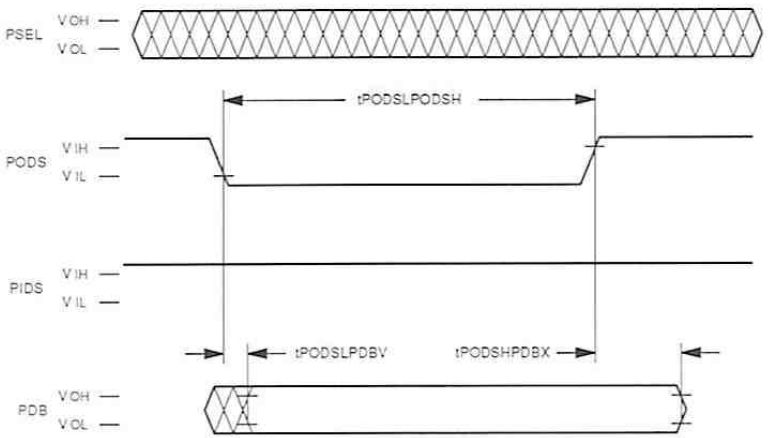

Figure 6-5. Passive Mode Output Timing

6.4.5 Parallel I/O Interaccess Timing

Figure 6-6 shows the timing of mixed active mode inputs and outputs. Refer to the previous sections (6.4.1 and 6.4.2) on active mode input and output transactions for specific descriptions of individual transactions.

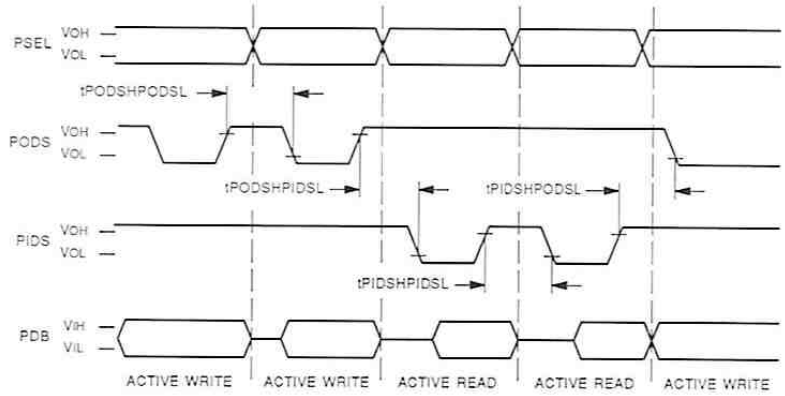

Figure 6-6. Parallel I/O Interaccess Timing

www.DataSheet4U.com