

Digilent Asynchronous Parallel Interface (DEPP)



Revision: September 2, 2010

1300 NE Henley Court, Suite 3
Pullman, WA 99163
(509) 334 6306 Voice | (509) 334 6300 Fax

Introduction

This document describes the operation and specifications of the Digilent Asynchronous Parallel Interface. This interface is implemented in various Digilent products to provide a communication and data transfer interface between a personal computer and a Digilent programmable logic system board.

The Digilent Adept Suite software provides a set of application programs and interface DLLs that provide the ability to perform JTAG operations and various kinds of communication and data transfer operations. The data transfer operations supported by Adept Suite DLLs require that certain logic be implemented in the FPGA/CPLD on the Digilent system board to provide the peripheral side of the interface. This document describes the requirements for this implementation.

Functional Description

The Digilent Asynchronous Parallel Interface is patterned after the EPP mode of the IEEE-1284 parallel port interface on IBM PC type computers. The interface consists of an 8-bit bi-directional data bus and four control signals. The data transfer speed that can be achieved depends on the particular communication subsystem and firmware version being used.

The asynchronous parallel interface is made up of an address register and a set of 8-bit data registers. The address register can be up to eight bits wide. It holds the address of the data register currently being accessed. Access to the registers is accomplished via bus cycles. The three types of bus cycles are Address Write, Data Read, and Data Write. An address write cycle is used to write to the address register. Data read or data write cycles read or write the data register whose address is currently held in the address register. Once an address has been written to the address register, any number of data read or write cycles can be performed on the selected data register. An address write will not necessarily be performed prior to each data read or write.

The address register can be implemented with any number of bits up to the maximum size of eight bits. Since the maximum size of the address register is eight bits, the maximum number of data registers that can be implemented is 256. It isn't necessary to implement all 256 possible data registers. Only the registers required for a particular application need to be implemented. It is necessary for the application using the interface to know which data registers are implemented and what functions they perform.

Signal Descriptions

In the following description, the term *host* refers to a communication interface/controller. Digilent has implemented USB and Ethernet hosts on various products. An EPP-style parallel port on a PC can also function as a host. Signals described as being sourced by the host are generated by the Digilent communication interface and are inputs to the logic in the FPGA/CPLD on the Digilent programmable logic system board. The term *peripheral* refers to logic implemented in the FPGA/CPLD on the system board. Signals sourced by the peripheral are outputs from the logic implemented in the FPGA/CPLD.

The following signals make up the interface:

Name	Source	Description
DB0 – DB7	bidir	Bi-directional data bus. The host is the source during write cycles and the peripheral is the source during read cycles.
WRITE	host	Transfer direction control. High = read, host reads from peripheral; Low = write, host writes to peripheral.
ASTB	host	Address strobe. Write to the address register.
DSTB	host	Data strobe. Write to/read from a data register.
WAIT	peripheral	Handshake signal used to indicate when the peripheral is ready to accept data or has data available.

Bus Cycles

Transfers from the host to the peripheral or from the peripheral to the host are accomplished using one of three bus cycles: Address Write, Data Read, and Data Write. All bus cycles are initiated and controlled by the host.

The strobe signals ASTB and DSTB are used to control the timing of bus cycles and are active low. Transfers to the address register occur when ASTB is asserted active during a bus cycle. Transfers to or from a data register occur when DSTB is asserted active during a bus cycle. During a write cycle, the host will place data on the bus prior to driving the strobe signal active. During a read cycle, the peripheral should place data on the bus while the strobe signal is active.

The following diagrams illustrate the signal timing for the various transfer cycles. For write cycles, the rising edge of the strobe signal (ASTB or DSTB) can be used to latch data into the peripheral. Alternatively, the host will assert data on the bus before bringing the strobe active and will hold it on the bus as long as the strobe is active. Data can be latched by the peripheral any time the strobe is active. For read cycles, the WRITE signal and the appropriate strobe are combined to enable the bus buffers in the peripheral to drive data onto the bus when it is a read cycle and the strobe is active.

The direction of the data transfer is controlled by the WRITE signal. If WRITE is high (indicating a read cycle) the peripheral is the source of the data and drives its data onto the data bus while the strobe signal is active. If WRITE is low (indicating a write cycle) the host is the source of the data and drives its data onto the data bus prior to asserting the strobe signal active. It is important that the peripheral not drive data onto the data bus except at the appropriate time during a read cycle. If the data bus is driven by the peripheral at incorrect times, it is possible to create a bus contention that can damage either the peripheral or the host.

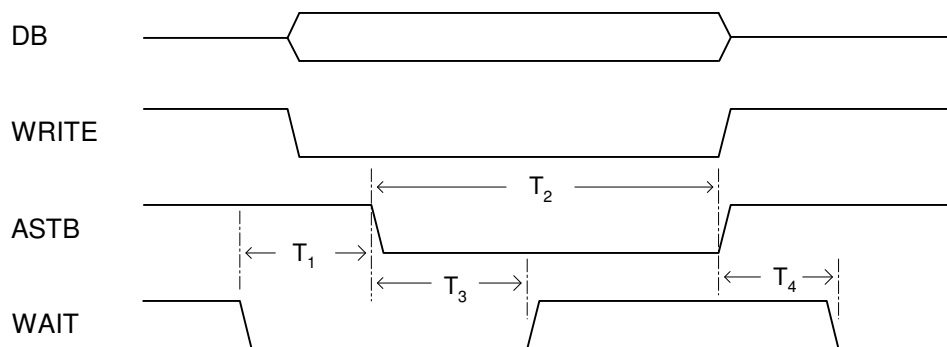
The WAIT signal is a handshake signal used to synchronize transfers between the host and the peripheral. The host will not begin a bus cycle unless the wait signal is low. A peripheral indicates to the host that it is ready to accept data by bringing WAIT low. Once WAIT is low, the host will begin the bus cycle by asserting the appropriate strobe signal low. The strobe will be held in the low state (prolonging the bus cycle) until the peripheral asserts WAIT high. WAIT going high signals that the peripheral is ready for the bus cycle to complete. If WAIT is not brought high within approximately 10ms of the start of a transfer, the host will abort the transfer and report a time out error. Once the peripheral asserts WAIT high, the host will then bring the strobe high, completing the bus cycle. The

peripheral can then perform any processing necessary while holding WAIT high. The peripheral brings WAIT low when it is ready for another bus cycle to begin. If the host is attempting to begin a bus cycle and WAIT does not come low within approximately 10ms, the host will report a time out error.

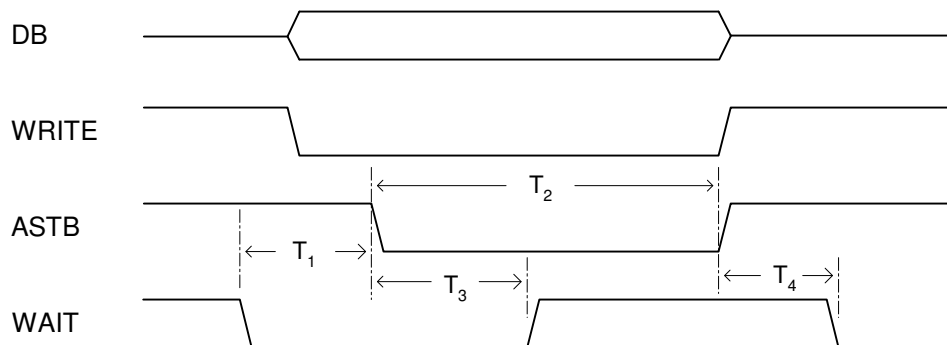
Timing Diagrams

Parameter	Description	Time
T_1	WAIT low to ASTB or DSTB active	40ns min
T_2	ASTB or DSTB active	80ns min
T_3	ASTB or DSTB active to WAIT high	0ns to 10ms
T_4	ASTB or DSTB inactive to WAIT low	0ns to 10ms
T_5	DSTB low to DB valid	20ns max
T_6	DSTB high to DB invalid	20ns min

Address Write



Data Write



Data Read

