LogiCORE IP XPS PS2 Controller (v1.01b)

DS707 April 19, 2010 **Product Specification**

Introduction

The LogiCORE IP XPS PS2 Controller is a PLB (Processor Local Bus) slave that is designed to control PS2 devices such as keyboard and mouse. The PS2 protocol is a simple bidirectional serial protocol.

Features

- Connects as a 32-bit slave on PLB V4.6 bus of 32, 64 or 128 bit data width
- Configurable as single or dual port PS2 controller
- Supports two PS2 devices, each controlled by separate set of eight byte-wide registers
- Two separate interrupts for each of the ports

LogiCORE™ Facts		
Core Specifics		
Supported Device Family	Spartan®-6, Virtex®-6/-6CX, Spartan-3, Spartan-3A, Spartan-3E, Automotive Spartan-3/3E/3A/3A DSP, Spartan-3 ADSP, Virtex-4, QVirtex-4, QRVirtex-4,Virtex-5/5FX	
Version of Core	xps_ps2	v1.01b
Resources Used		
	Min	Max
SLICES	See Table 14, Table 15, Table 16, Table 17, and Table 18	
LUTs		
FFs		
Block RAMs	N/A	
Special Features	N/A	
Provided with Core		
Documentation	Product Specification	
Design File Formats	VHDL	
Constraints File	N/A	
Verification	N/A	
Instantiation Template	N/A	
Reference Designs & Application notes	N/A	
Design Tool Requirements		
Xilinx Implementation Tools	ISE 12.1	
Verification	MentorGraphics ModelSim 6.5c and above	
Simulation	MentorGraphics ModelSim 6.5c and above	
Synthesis	XST	
Support		
Provided by Xilinx, Inc.		

Functional Description

The XPS PS2 Controller is a slave IP core designed to control two PS2 devices such as a keyboard and a mouse. To control the two PS2 devices, it uses simple state machines and shift registers. Each of the PS2 ports is controlled by a separate set of four byte-wide registers. For transmitting data, a byte is written to the transmit register and then the Serial Interface Engine serializes the data and transmits to the PS2 device. Transmit status registers and interrupts indicate whether the transmission is complete and if there are any errors reported. While receiving data, the Serial Interface Engine receives serial data from the PS2 device and writes into the receive register. Similar to the transmit status registers, receive status registers and interrupts indicate whether data has been received from the PS2 device. Any errors in the received data are also reported. The XPS PS2 Controller generates interrupts upon various transmit and receive conditions.The XPS PS2 Controller can be operated in a polled mode or an interrupt driven mode.

PS2 Communication

The PS2 protocol consists of host-to-device and device-to-host communication. In the description below "host" implies the XPS PS2 Controller and "device" implies any PS2 device, which would be a keyboard or a mouse.

The PS2 protocol is a bidirectional synchronous serial protocol. The data and the clock are the two signals through which communication between the device and the host happens. The host is given the ultimate control of the data bus. The basic states which can be defined based on the status of the data and clock lines are:

- Idle State - Data is high and clock is high. This is the only state where the PS2 device is allowed to start transmission of data (during device-to-host communication).
- Communication Inhibited State - Data high and clock low. The device always generates the clock signal. But since the host has the ultimate control of the bus and may inhibit communication anytime, it must pull the clock signal low and inhibit the transmission by the device and then initiate transmission from its side.
- Host Request-to-Send State - Data is low and clock is high. The host after inhibiting the communication, will pull the data line low and release the clock inline, signalling the device that host would transmit data.

All the data is transmitted one byte at a time and each byte is sent in a frame consisting of 11-12 bits (depending on whether it is host-to-device or device-to-host communication). These bits are:

- 1 start bit. This is always 0
- 8 data bits, least significant bit first
- 1 parity bit (odd parity)
- 1 stop bit. This is always 1
- 1 acknowledge bit (host-to-device communication only)

The data sent from the device to host is read on the falling edge and the data sent from the host to device is read on the rising edge. The clock frequency must be in the range 10-16.7 kHz. This means clock must be high for 30 - 50 microseconds and low for 30 - 50 microseconds. The keyboard, mouse or host emulator should modify/sample the data line in the middle of each cell, i.e. 15 - 25 microseconds after the appropriate clock transition.

Device-to-Host Communication

The device-to-host communication happens over 11-bit frames. When the keyboard or mouse wants to send information, it first checks the clock line to make sure it's at a high logic level. If it's not, the host is inhibiting communication and the device must buffer any to-be-sent data until the host releases clock. The clock line must be continuously high for at least 50 microseconds before the device can begin to transmit data.

The host may inhibit communication at any time by pulling the clock line low for at least 100 microseconds. If a transmission is inhibited before the 11th clock pulse, the device must abort the current transmission and prepare to retransmit the current "chunk" of data when host releases clock. A "chunk" of data could be a make code, break code, device ID, mouse movement packet, etc. For example, if a keyboard is interrupted while sending the second byte of a two-byte break code, it will need to retransmit both bytes of that break code, not just the one that was interrupted.

If the host pulls clock low before the first high-to-low clock transition, or after the falling edge of the last clock pulse, the PS2 device does not need to retransmit any data. However, if new data is created that needs to be transmitted, it will have to be buffered until the host releases clock.

Figure 1 illustrates the above mentioned device-to-host communication.

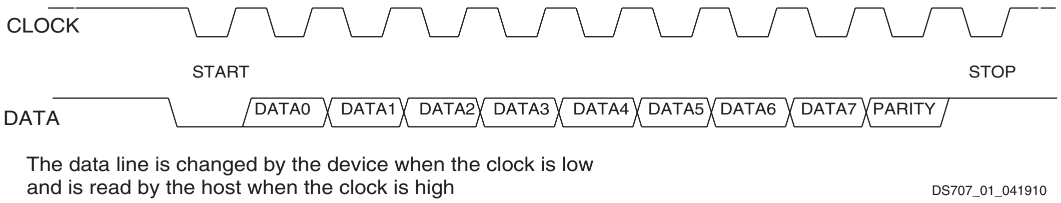
CLOCK
START
STOP
DATA
DATA0
DATA1
DATA2
DATA3
DATA4
DATA5
DATA6
DATA7
PARITY
The data line is changed by the device when the clock is low
and is read by the host when the clock is high
DS707_01_041910

Figure 1: **Device-to-Host Communication**

Host-to-Device Communication

The host-to-device communication happens over 12-bit frames. Since the PS2 device always generates the clock signal, the host whenever it wants to send data, must inhibit communication by pulling the clock low for at least 100 microseconds. It must then apply a "request-to-send" by pulling data low while releasing the clock signal.

The device should check for this state at intervals not to exceed 10 milliseconds. When the device detects this state, it will begin generating clock signals and clock in eight data bits and one stop bit. The host changes the data line only when the clock line is low, and data is read by the device when clock is high. After the stop bit is received, the device will acknowledge the received byte by bringing the data line low and generating one last clock pulse. If the host does not release the data line after the 11th clock pulse, the device will continue to generate clock pulses until the data line is released (the device will then generate an error.)

The host may abort transmission at time before the 11th clock pulse (acknowledge bit) by holding clock low for at least 100 microseconds.

There are two timings that are needed to be taken care by the state machines. The time it the device to begin generating clock pulses after the host initially takes the Clock line low, which must be no greater than 15 milliseconds. And the time it takes for the packet to be sent, which must be no greater than 2 milliseconds. If either of the above time limits is not met, the host should generate an error. If the command sent by the host requires a response, that response must be received no later than 20 milliseconds after the host releases the Clock line. If this does not happen, the host should generate an error.

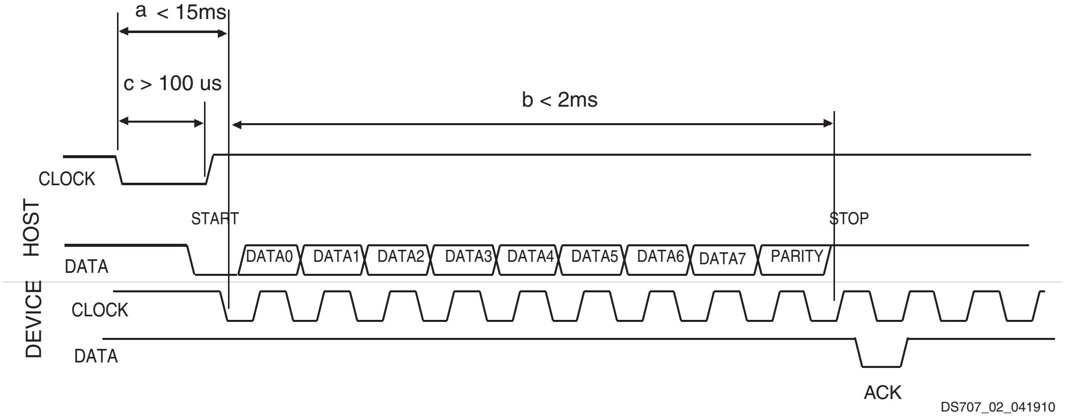

Figure 2: **Host-to-device communication**

The major interface and modules, for each of the ports, of the XPS PS2 Controller are shown in Figure 3 and described in the subsequent sections. These modules are:

- PLB Interface Module
- Interrupt Service Controller
- Serial Interface Engine (SIE)
- Internal Registers

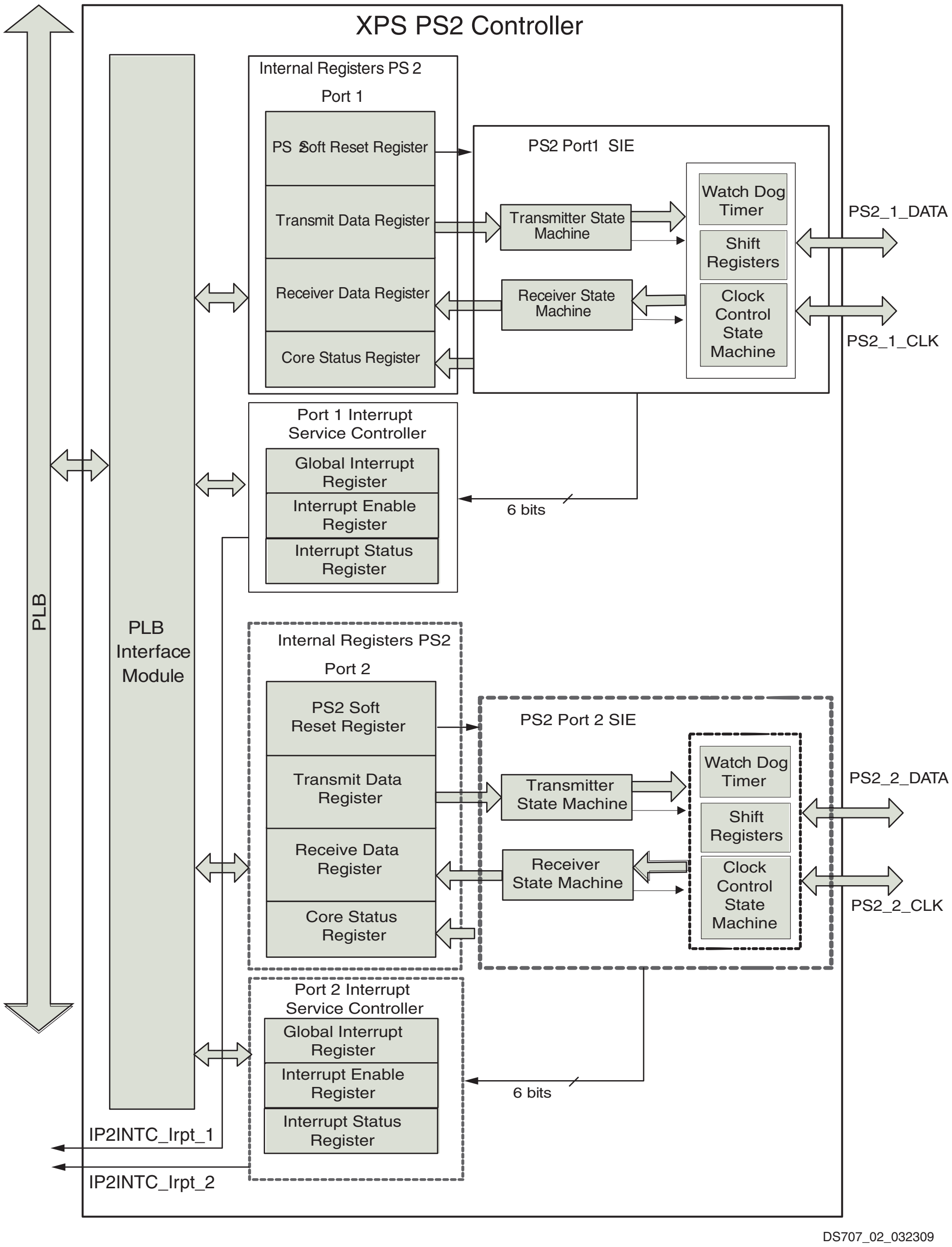

Figure 3: **XPS PS2 Controller Block Diagram**

PLB Interface Module

The PLB Interface Module is a bi-directional interface between XPS PS2 Controller IP core and the PLB. To simplify the process of attaching a XPS PS2 Controller to the PLB, the core makes use of a portable, pre-designed bus interface called PLB Interface Module, that takes care of the bus interface signals, bus protocols and other interface issues. The base element of the PLB Interface Module is slave attachment, which provides the basic functionality of PLB slave operation.

Interrupt Service Controller

The Interrupt Service Controller is a continuation of the Xilinx family of IBM CoreConnect compatible LogiCORE products. It provides interrupt capture support for the connected IP function. The interrupts from XPS PS2 Controller are connected to the Interrupt Service Controller and the corresponding bits in the Interrupt Status Register get updated. Interrupt Service Controller provides following functions:

- Parameterized number of interrupts needed by the IP.
- Provides both Interrupt Status Register (ISR) and Interrupt Enable Register (IER) functions for the user IP. Depending on which interrupt bits are enabled in the IER, the corresponding interrupts from the IP would be or-ed and a single interrupt would be generated.

Serial Interface Engine

The Serial Interface Engine has the following modules:

- Transmit State Machine - There are 2 separate transmit state machines for each of the PS2 ports. This helps each of the ports to transmit simultaneously irrespective of the other port. The transmit state machine serializes the data written into the transmit data register and sends it over the data line as per the PS2 protocol as mentioned above in the Host-to-Device Communication section.
- Receive State Machine - There are 2 separate receive state machines for each of the PS2 ports. This helps each of the ports to receive simultaneously irrespective of the other port. The receive state machine does the serial-to-parallel conversion of the serial data received on the data line and writes into the receive data register and sends it over the data line as per the PS2 protocol as mentioned above in the Device-to-Host Communication section.
- Shift Registers - The parallel-to-serial and serial-to-parallel conversion of data by the transmit state machines and receive state machine respectively, is done by using these shift registers.
- Clock Control State Machine - This state machine detects the rise and fall of the clock line. This edge detection is required for the transmit and receive state machines to perform the operation. The PS2 protocol is greatly dependent on the edge of the clock.
- Watch Dog Timer- The watch dog timer keeps a watch on the clock line while transmitting. If the clock line goes into the pull-up mode in between transmission, the watch dog timer sets the corresponding interrupt. If there are no transitions on the clock for 200 microseconds, the watch timer would set the interrupt.

Internal Registers

The XPS PS2 controller has eight internal registers, four registers for each of the PS2 ports. The whole operation happens through the internal registers. For transmitting data to the PS2 device, the data needs to be written into the transmit register and similarly the receive register would get updated as soon as the data is received from the PS2 device. A more detailed description of these registers is given in the XPS PS2 Controller Register Descriptions section.