PmodJSTK™ Demo Project

www.digilentinc.com

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

Revision: July 16, 2012

Overview

This document presents a VHDL demo project that interfaces the PmodJSTK. This demo is intended for use with a Nexys3 board and Xilinx ISE Design Suite 14.1. Positional data is displayed on the seven segment display (SSD), and the status of the PmodJSTK's buttons is displayed on the Nexys3's LEDs. Switches are used to turn on/off the LEDs on the PmodJSTK, and to select between X and Y positional data to display on the SSD. The positional data of the joystick ranges from 0 to 1023 in both the X and Y directions. Only one coordinate may be displayed on the SSD at a time, therefore switch SW0 is used to select which coordinate's data to display. Button BTND on the Nexys3 is used for resetting the demo. The PmodJSTK connects to port JA[4:1] on the Nexys3.

Functional Description

Figure 1 to the right shows a diagram of the PmodJSTK Demo. The Nexys3's onboard 100Mhz clock (*CLK*) is used to produce a 5Hz send/receive (*sndRec*) signal. This signal initializes a data transfer between the PmodJSTK and the Nexys3. The signal is fed into a component named *PmodJSTK,* and is input into a SPI controller subcomponent (*SPI_Ctrl*) which controls all SPI communication in the design.

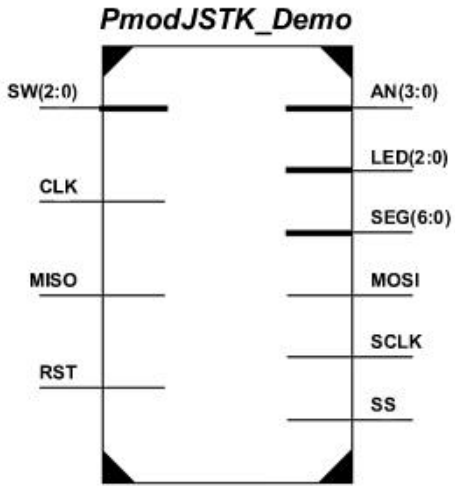

Fig. 1 – PmodJSTK Demo

The SPI controller operates on an internal serial clock (*iSCLK)* signal that runs at a frequency of 66.67kHz. SPI controller produces the slave select (*SS)* output and produces the control signals for SPI interface (*SPI_Int*) another subcomponent.

SPI interface uses the same 66.67kHz serial clock as SPI controller. This component is responsible for all data transmission and reception in the design. Data is received from the PmodJSTK on the *MISO* input, and data is written to the PmodJSTK on the *MOSI* output.

SPI interface produces the output serial clock (*SCLK*), which runs at 66.67kHz and controls the timing of the communication process with the PmodJSTK. When SPI interface receives a send/receive signal from the SPI controller, five bytes of data will be transferred between the PmodJSTK and the Nexys3. The first byte of data sent from the Nexys3 contains the status of *SW2* and *SW1*, which control the LEDs on the PmodJSTK. The content of the remaining 4 bytes of data to be sent does not matter, because the PmodJSTK ignores them.

For every byte written to the PmodJSTK, the Nexys3 reads a byte. Each byte read, is sent to the *SPI_Ctrl* component, and is packaged into a 40-bit output containing positional data, and the status of the PmodJSTK's buttons. This 40-bit output is directly output to the *DOUT* output on the *PmodJSTK* component. Data is made available on a *jstkData* bus in the *PmodJSTK_Demo*. The lower three bits in the *jstkData* bus correspond to the buttons on the PmodJSTK, and are directly output to the *LED*

Copyright Digilent, Inc. All rights reserved. Other product and company names mentioned may be trademarks of their respective owners.

output bus which drives *LD2*, *LD1*, and *LD0* on the Nexys3.

Depending on the status of switch *SW0* on the Nexys3, either the X or Y positional data in the *jstkData* bus will be sent into the display controller (*DispCtrl*) for display on the SSD. The positional input data is converted from binary to binary coded decimal (BCD), decoded, and then displayed on the SSD via the anode (*AN*) and cathode (*SEG*) output busses. Each digit on the SSD is refreshed at a frequency of 250Hz using a 1kHz clock. For details on the interconnections, components, etc., see Fig. 2 below. For a summary of the PmodJSTK demo inputs and outputs see Table 1 below.

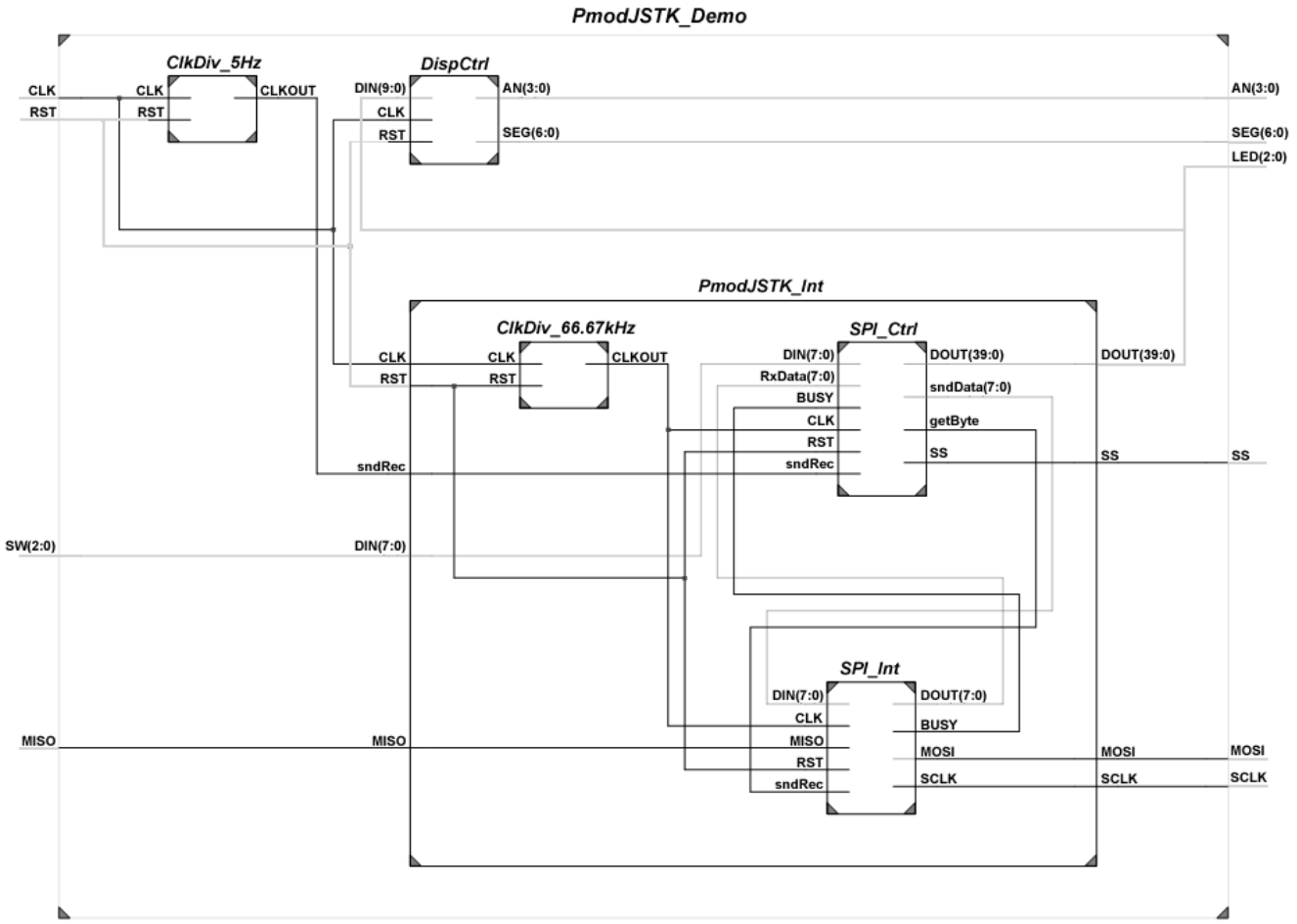

Fig. 2 – PmodJSTK_Demo Block Diagram

Table 1: Input/Output Pin Description	
Signal	**Description**
CLK	Input clock, 100MHz onboard fixed oscillator.
RST	Reset input, used to reset the demo. (*BTND on Nexys3*)
SW(2:0)	Input switches for selecting data and controlling PmodJSTK LEDs.
MISO	Input used for collecting data.
AN(3:0)	Output anodes for controlling illumination of SSD digits.
LED(2:0)	Output LEDs used for displaying statuses of PmodJSTK buttons.
MOSI	Output used for sending data.
SCLK	Output serial clock for communication.
SS	Output slave select used for enabling/disabling communication with PmodJSTK.

Copyright Digilent, Inc. All rights reserved. Other product and company names mentioned may be trademarks of their respective owners.

Component Description

Display Controller

Positional data that is input to the DispCtrl component is in the form of 10-bit unsigned binary numbers. To display these values on the SSD, it is necessary to convert the binary number to 4 binary coded decimal (BCD) values. This conversion takes place inside the *DispCtrl* component, and implements the "Shift Add 3" algorithm. There will be 4 digits available for display, where each digit is represented by a 4-bit value on a 16-bit bus named *bcdData.* The order of the digits in the 16-bit value is as follows, bcdData(15:12) (thousands place), bcdData(11:8) (hundreds place), bcdData(7:4) (tens place), bcdData(3:0) (ones place).

The SSD is refreshed at a using a 1kHz clock divider and 2-bit counter. The 2-bit count is decoded and used to select different anode patterns to send to the SSD. All anode patterns allow for 1 digit on the SSD to be illuminated at a time. The count value is also used to select which BCD digit from the *bcdData* bus to send into a decoder. The decoder produces the necessary cathode signals for displaying the digit on the SSD.

PmodJSTK Interface

This component consists of three subcomponents a 66.67kHz serial clock (*SerialClock*), SPI controller (*SPI_Ctrl)* and a SPI interface (*SPI_Int)*. The *SPI_Int* component is responsible for sending and receiving a byte of data to and from the PmodJSTK when a request is made. The *SPI_Ctrl* component manages all data transfer requests, and manages the data bytes being sent to the PmodJSTK.

SPI Controller

When a send/receive signal is received a data transfer will be initialized. Each data transfer will consist of five bytes being sent/received. Data read from the PmodJSTK is output on the *DOUT* output bus. Another data transfer request cannot be initialized until the send/receive input has been de-asserted from a previous request. This component operates on the 66.67kHz serial clock in *PmodJSTK*.

SPI Interface

The PmodJSTK uses SPI mode 0 protocol for communication with the Nexys3, the PmodJSTK is the slave and the Nexys3 is the master. This component generates a 66.67 kHz serial clock (*SCLK*) output that controls when data bits are read and written. Both the master and the slave read data on rising edges, and both change their output data on falling edges. The *MISO* input carries data being sent from the slave, and the *MOSI* output carries data being sent from the master. For more information on the contents of the bytes being sent and received see page 2 in the PmodJSTK reference manual.

A data transfer will be initiated when a send/receive signal (*sndRec*) is received. Once a request has been received *SPI_Int* will capture the byte of data on the *DIN* input, and assert the *BUSY* output to indicate to other components in the design that a data transfer is currently in progress. The *SPI_Int* component sends and receives a bit every 15µs until an entire byte has been sent/received. Once finished *SPI_Int* handshakes with *SPI_Ctrl* by de-asserting its *BUSY* output.

Copyright Digilent, Inc. All rights reserved. Other product and company names mentioned may be trademarks of their respective owners.