PmodACL Demo Project

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

Revision: August 9, 2012

Overview

This document presents VHDL implementation details for a SPI controller to interface the PmodACL. The project first configures and enables the accelerometer, then begins to sample axis data at 5Hz and continuously display the sampled gravity value ("g" value) on the seven segment display (SSD). Switches on the Nexys3 are used to select which axes "g" value to display on the SSD. For demo operation see Table 1 below for details. The PmodACL connects to port "JB" on the Nexys3.

Table 1 – Demo Operation, Inputs and Outputs on Nexys3

SW1	SW0	SSD Output	LD2	LD1	LD0
Off	Off	x-axis	Off	Off	On
Off	On	y-axis	Off	On	Off
On	On	x-axis	Off	Off	On
On	Off	z-axis	On	Off	Off

Functional Description

Figure 1 to the right shows a block diagram of the PmodACL Demo. There are four main components in this demo, SPIcomponent, sel_Data, ClkDiv_5Hz, and ssdCtrl. A *START* signal is generated by the ClkDiv_5Hz component, this signal is used to initiate a data transfer between the PmodACL and the Nexys3.

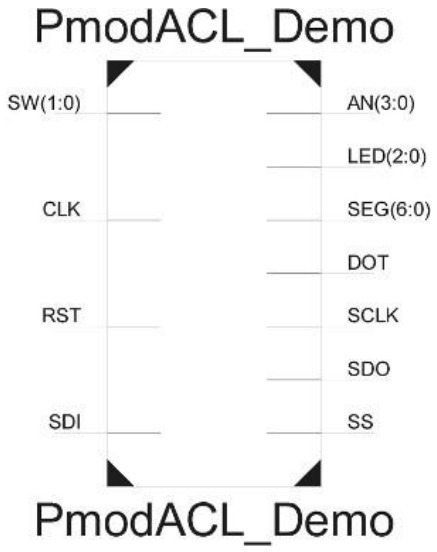

Fig. 1 – PmodACL Demo

SPIcomponent receives the *START* signal, configures the PmodACL, and then receives data pertaining to all three axes. The data is made available to the rest of the design on the *xAxis*, *yAxis*, and *zAxis* outputs.

These outputs are then sent into the sel_Data component with the status of switches SW1 and SW0 on the Nexys3. Depending on the configuration of these switches, one of the axes data will be selected for display on the seven segment display. The selected data is converted from 2's compliment to magnitude, and is then sent to the ssdCtrl where it is converted to a "g" value and displayed on the SSD with hundredths precision. See Figure 2 on page 2 for interconnections of major components in this design. For details on the various inputs and outputs of this demo see Table 2 on page 2.

The SPIcomponent module contains the entire interface to the PmodACL. This module consists of three subcomponents, SPImaster, SPIinterface, and slaveSelect. The SPImaster controls both the SPIinterface and slaveSelect modules. SPIinterface produces the serial clock (*SCLK*) output used for communicating with the PmodACL, and is responsible for data transmission and reception. The slaveSelect module manages the slave select (*SS*) output, which enables/disables the PmodACL. For details on components see Component Description section below.

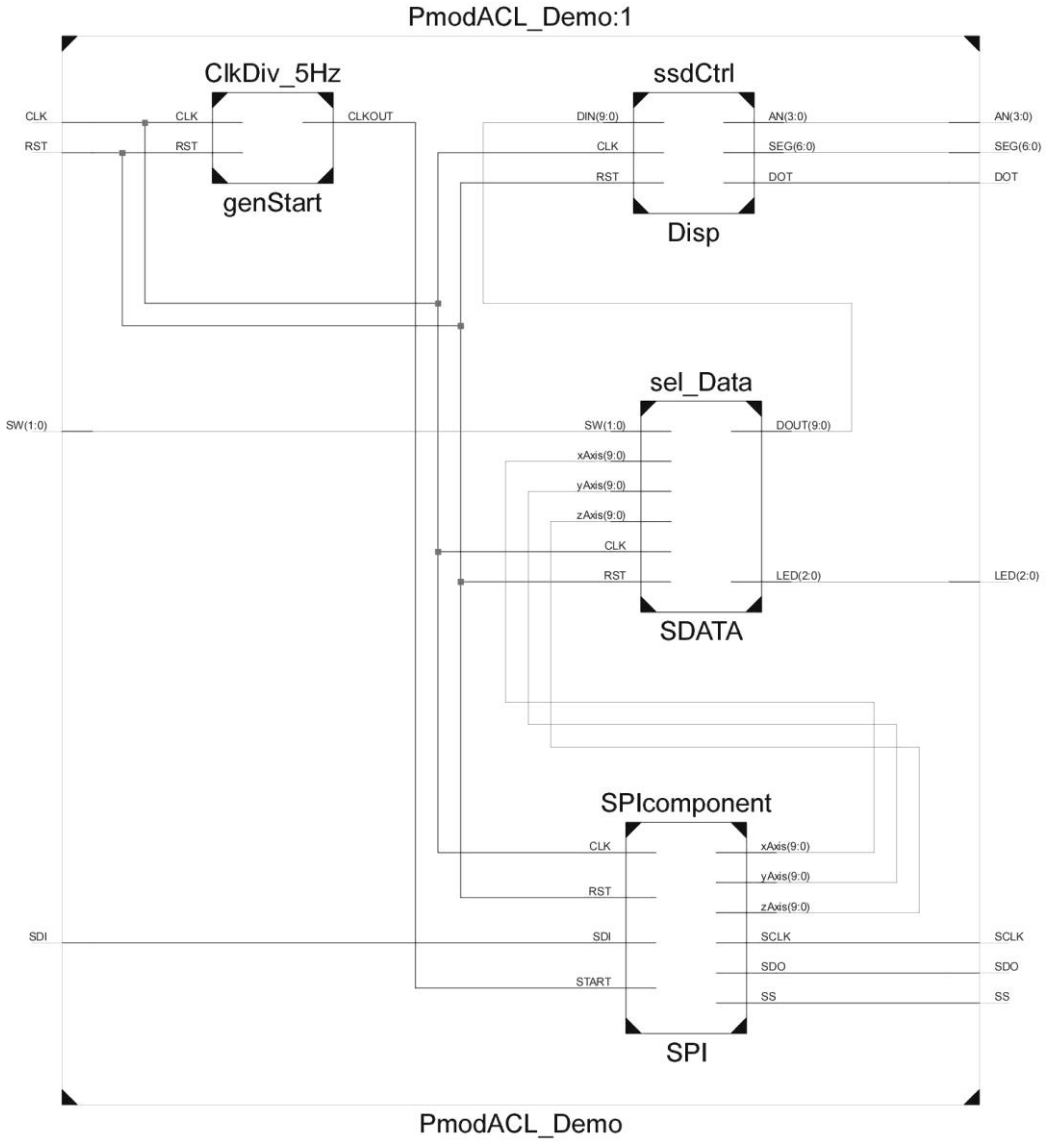

Fig. 2 – PmodACL Demo Interconnections

Table 2: 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(1:0)	Input switches for selecting axis data to display on SSD.
SDI	Input data from PmodACL, all axis data comes in on this line.
AN(3:0)	Output anodes for controlling illumination of SSD digits.
SEG(6:0)	Output cathodes for displaying digits on SSD.
DOT	Output cathode for displaying decimal on SSD.
LED(2:0)	Output LEDs used for displaying which axes data is currently selected.
SCLK	Output serial clock for communication.
SDO	Output data to PmodACL, all configuration data gets written out on this line.
SS	Output slave select used for enabling/disabling communication with PmodACL.

Component Description

ssdCtrl

Acceleration data that is input into the *ssdCtrl* is a 10-bit value. The most significant bit (MSB) indicates the sign of the input magnitude number, and the lower 9 bits are the magnitude of the acceleration. The data is formatted into a "g" value, and is then converted to a binary coded decimal (BCD). To display the data on the SSD, it is necessary to convert the magnitude to 3 binary coded decimal (BCD) values. This conversion takes place inside the *ssdCtrl* component, and implements the "Shift Add 3" algorithm. There will be 3 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) (minus sign, assigned seperately), bcdData(11:8) (ones place), bcdData(7:4) (tenths place), bcdData(3:0) (hundredths place).

The SSD is refreshed 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.

ClkDiv_5Hz

This is a simple clock divider that converts the input 100MHz clock signal into a 5Hz clock signal. The output signal is used to initialize a data transfer with the PmodACL.

sel_Data

This module converts axis data from 2's compliment to its magnitude representation. Based upon the input switches SW(1) and SW(0) either the x-axis, y-axis, or z-axis data will be output on the DOUT output.

SPIcomponent

This module is a shell for all of the necessary serial port communication interface components to connect to one another. The components are *SPImaster*, *SPIinterface*, and *slaveSelect*.

SPIinterface

This component receives and transmits data and produces the 22.4kHz *SCLK* output which drives communication with the PmodACL. When SPImaster asserts the *transmit* signal, the value in the *TxBuffer* is moved into a shift register and is shifted out during transmission. As data is shifted out, data is shifted into the *RxBuffer* from the PmodACL. The *RxBuffer* output is connected to the *rxdata* input on the SPImaster component. The output data from the transmission shift register ties to serial data out (SDO) on JB3 while input data for receiving ties to serial data in (SDI) on JB2.

SPImaster

This module controls data transmitted to the accelerometer and handles data reception, and is responsible for internal data control. The *RST* input forces the system into an idle state with standard starting values, and is tied to BTND on the Nexys3. When the button is pressed the demo will reset to starting conditions. The *START* input is used to initialize a data transfer with the PmodACL. It should be noted that data cannot be sampled until the PmodACL has been properly configured first.

On power up the PmodACL is immediately configured for +/- 2g operation. Once the PmodACL has been configured, this module will sample the axes acceleration data every time *START* is asserted.

SPImaster uses the onboard 100 MHz clock of the Nexys3, and updates on rising edges. The *transmit* and *done* signals are used for handshaking between the *SPImaster*, *SPIinterface*, and *slaveSelect* components. The *transmit* signal indicates the beginning of a transmission to the accelerometer, and the *done* signal indicates that the *SPIinterface* has completed data reception.

slaveSelect

The slaveSelect module controls the *SS* output to the accelerometer. When this signal is high communication with the PmodACL is disabled, and conversely when the signal is low communication is enabled. The SS output of slaveSelect is tied to JA1.