PmodOLED Interface Reference Project

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

Revision: June 25, 2012

Overview

This document presents Verilog implementation details for a project interfacing with a PmodOLED. The project first follows the manufacturer's initialization process for the OLED screen. After that is completed the project then displays the English alphabet, capitalized and lower case, followed by the numbers 0 through 9. The project then waits a few seconds, clears the screen and then displays "This is Digilent's PmodOLED".

Functional Description

The PmodOLED Interface reference project implements a SPI controller for interfacing with the PmodOLED. This project was developed on a Spartan-6 based Nexys3 board, but it can be easily retargeted to any other FPGA. The onboard 100 MHz oscillator is used to generate a 3.125 MHz signal for the SPI controller and a 1KHz counter for the delay block. The 100 MHz signal is used in the main blocks while the 3.125 MHz signal is used in the SPI interface as a data clock and the 1KHz counter is used to count milliseconds that has gone by.

Block Description

SpiCtrl Behavior

The SpiCtrl block uses the 100 MHz system clock of the Nexys3 to generate a 3.125 MHz SCLK (Serial Clock). This block waits until SPI_EN is asserted, active high, and then transitions into the Send state. The controller pulls CS (Chip Select) low and then starts shifting out the byte currently held in SPI_DATA to SDO (Serial Data Out) on the falling edge of SCLK. After all 8 bits have been sent the controller waits a few system clocks in order to hold CS low for the required amount of time specified by the data sheet. The controller then transitions into the Done state at which the SPI_FIN signal is pulled high. The controller waits for the SPI_EN signal to be de-asserted at which time the controller transitions into the Idle state and SPI_FIN is pulled low.

Delay Behavior

The Delay block uses the 100 MHz system clock of the Nexys3 to generate a 1KHz counter which is used to count milliseconds. This block waits until DELAY_EN is asserted, active high, and then transitions into the Hold state. The 1KHz counter then starts counting and when the counter is equivalent to the DELAY_MS vector. The controller then transitions into the Done state at which the DELAY_FIN signal is pulled high. The controller waits for the DELAY_EN signal to be de-asserted at which time the controller transitions into the Idle state and DELAY_FIN is pulled low.

OledInit Behavior

The OledInit block sends the commands and delays for the initialization sequence as specified by the OLED manufacturer.

OledEx Behavior

The OledEx block waits until the EN signal is asserted, active high, and then transitions into the ClearDC state. The ClearDC state then transitions to the SetPage, PageNum, LeftColumn1, LeftColumn2, and SetDC states which sets the page number of the PmodOLED to 0 before transitioning to the Alphabet state. Then current_screen is set to alphabet_screen and then transitions into the UpdateScreen state which updates the PmodOLED's screen to the ASCII characters contained in current_screen. It then waits for 4 seconds, clears the screen, waits another 1 second and then changes the screen to the DigilentScreen state "This is Diglient's PmodOLED".

charLib Behavior

This is the block memory that contains the byte maps for the characters. The characters are contained in 8 byte parts. The addressing is 11 bits and is formatted as followed: "ASCII Value & "XXX"" where the three bits at the end represent the 8 parts of the character. For example the address "01000001001" is the second byte of the character 'A'.

PmodOLEDCtrl Behavior

This block is the overall controller for this demo. It first enables the OledInit block, waits for it to finish and then enables the OledEx block. Afterwards it just remains in the Done state. Additional functionality can easily be added by placing another state after the OledExample state or by modifying the OledEx block.

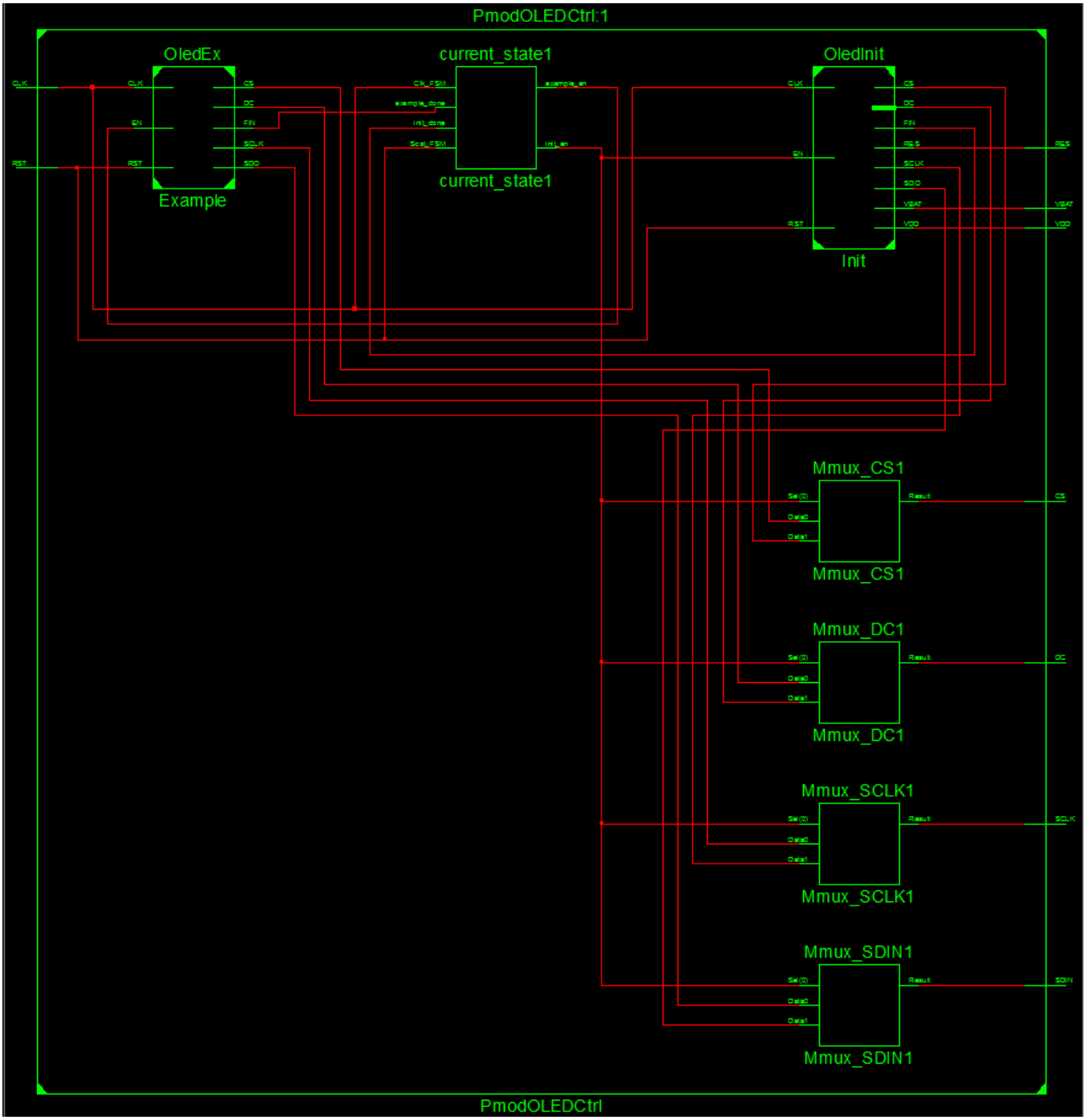

Figure 1: PmodOLEDCtrl Block Diagram

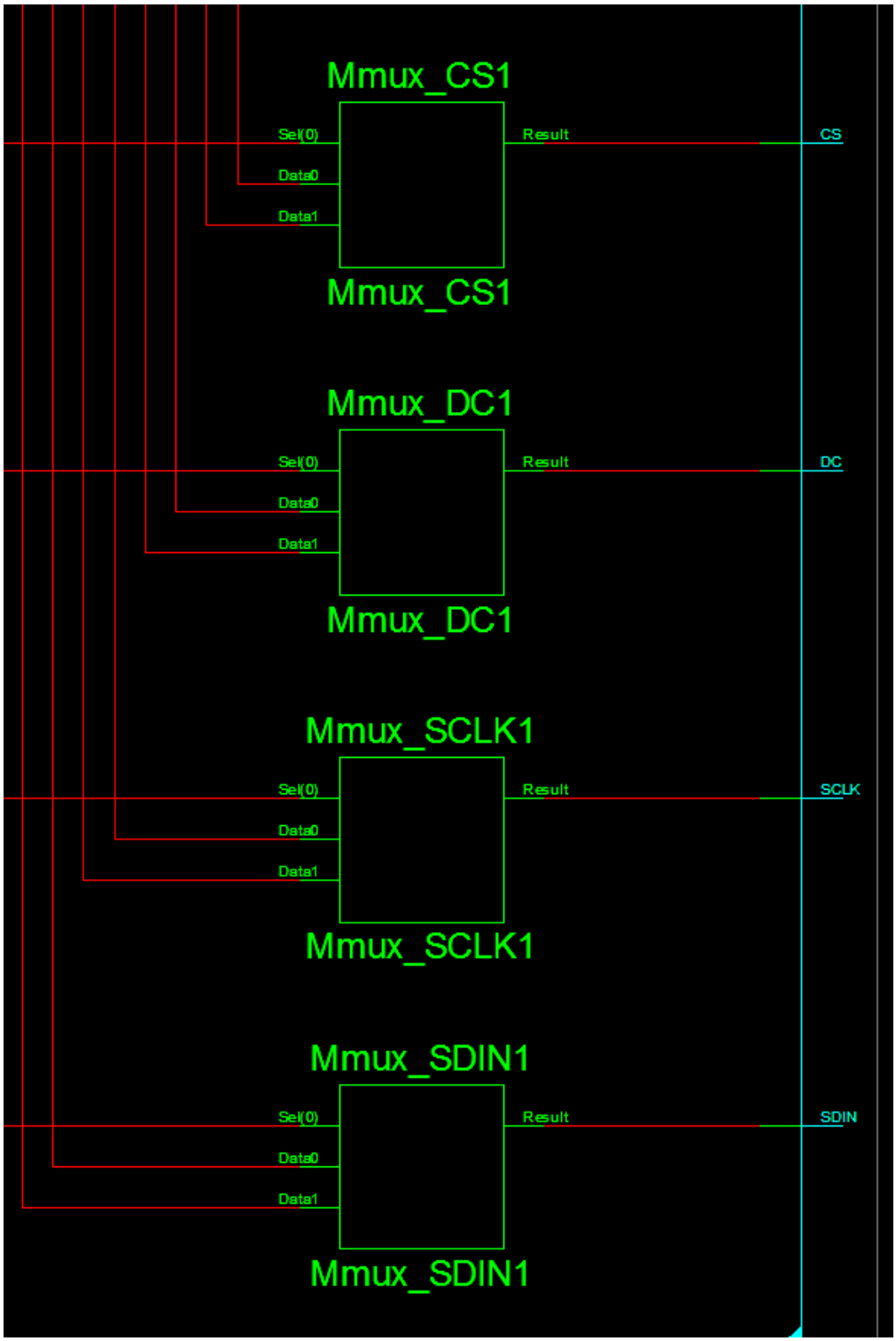

Figure 2: Muxes

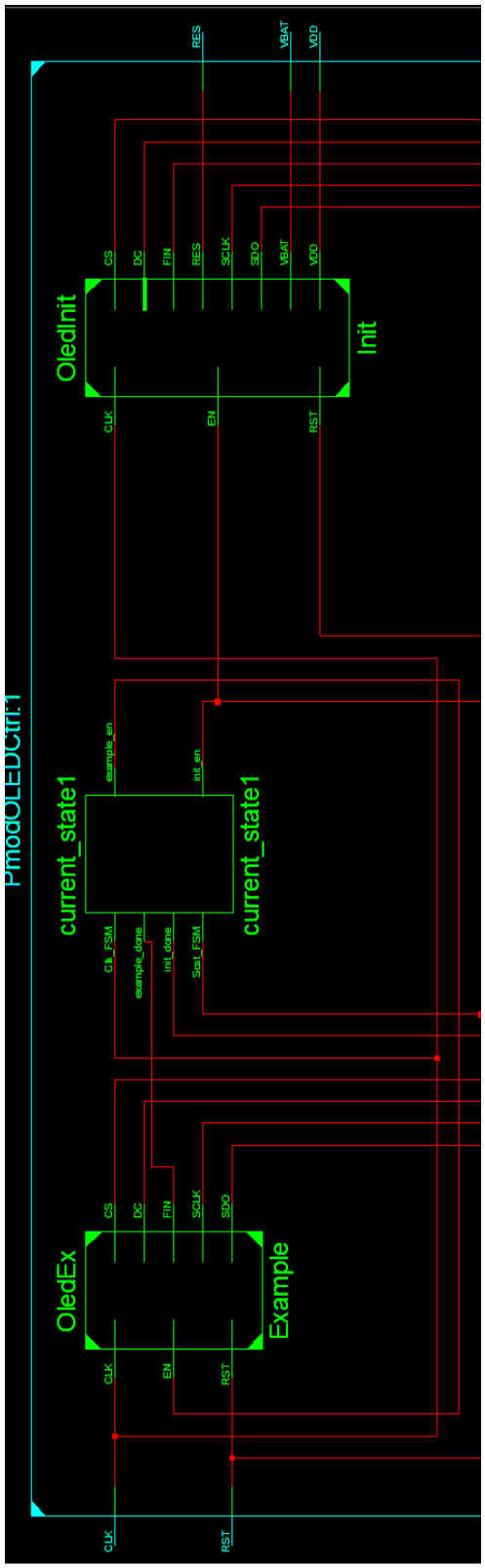

Figure 3: State Machines and Lower Blocks