PmodDA2™ Reference Component

Revision: December 3, 2008

This document was produced by Digilent Romania. For questions, contact support@digilent.ro.

Overview

The PmodDA2 Reference Component synchronizes data communications between a Digilent FPGA development board and the PmodDA2 board.

The PmodDA2 Reference Component inputs a digital signal and shifts out the data serially to the PmodDA2. It takes in a 12-bit vector and shifts out the 12-bit vector to the PmodDA2 board using the correct timing sequence. It also supplies the appropriate timing sequence to clock the PmodDA2.

Functional Description

Component Architecture

The VHDL component is an entity named DA2RefComp that has five inputs and five outputs. The input ports are a 50MHz clock (labeled CLK) that is divided down and used to clock the processes in the component, and an asynchronous reset signal (labeled RST) that resets the processes that occur inside the component. The data inputs for the two DAC121S101 chips are two 12-bit vectors (DATA1 and DATA2) that are shifted out serially to the PmodDA2 data pins. The START input signal is used to tell the component when to start a conversion.

The output ports are the divided clock signal CLK_OUT (25MHz) and two serial outputs (D1 and D2) that provide the shifted data to the PmodDA2. An nSYNC output is used to latch the data inside the PmodDA2 after the data has been shifted out. An output labeled DONE tells the user when the conversion is done. A block diagram of the component is shown in Figure 1.

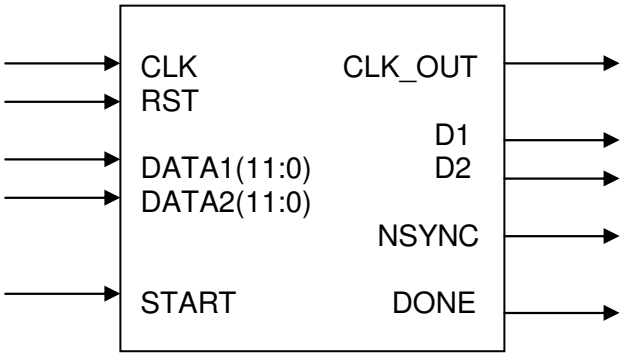

Figure 1 ***PmodDA2 Reference Component***

Timing

The timing diagram in Figure 2 is used to determine the correct timing sequence for the finite state machine that clocks the PmodDA2. It is the timing sequence that is used to send 16 bits of data to the DAC121S101 chips on the PmodDA2. The two data signals are sent to the first channels of the two DAC121S101 chips. The signal nSYNC must be at a low or zero state while the data is transferred in on the rising edge of the clock signal. Immediately following the data transfer, the signal nSYNC must be driven high to latch the data into the DAC121S101 chip.

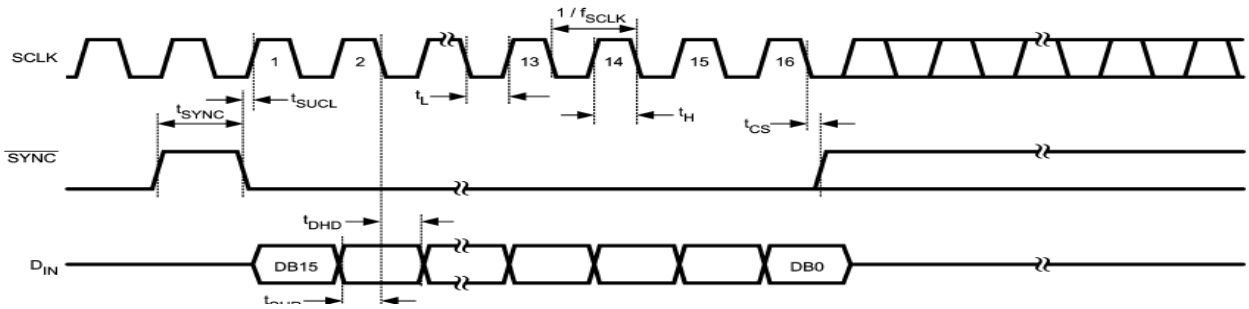

Figure 2 ***Timing Diagram of the DAC121S101 Chip on the PmodDA2***

The logic that created the timing sequence to take in the DATA1 and DATA2 input and shift out the data bits serially on the outputs D1 and D2, as well as drive the nSYNC output and DONE output signals, was designed by creating the finite state machine shown in Figure 3.

State Machine

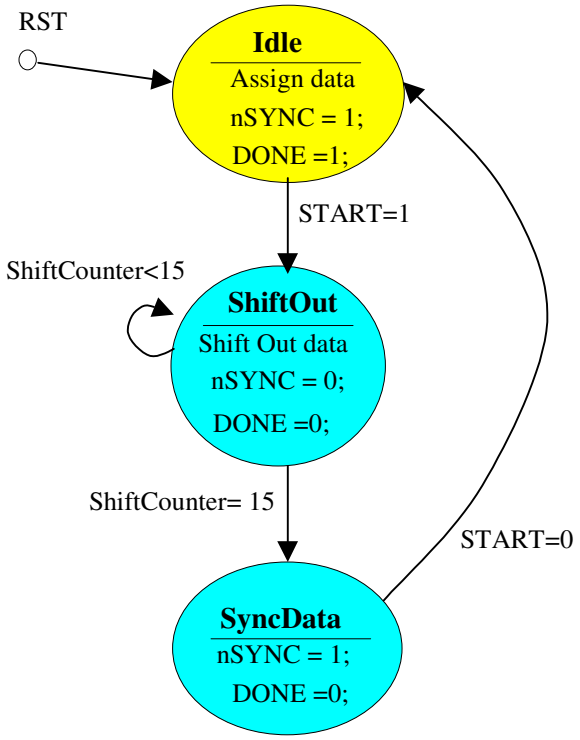

Figure 3 ***FSM of PmodDA2 Reference Component***

There are three states: Idle, ShiftOut, and SyncData. During the Idle state, the 12-bit data vector is updated along with the 4-bit control register. The DONE output signal needs to be high in order to allow a conversion. When the START input signal activates during the Idle state, the state machine goes into the ShiftOut state.

In the ShiftOut state, the data bits are shifted out from MSB to LSB using a counter that counts 16 clock periods. It is this counter that ensures that all the data will be shifted out before moving on to the SyncData state.

When the counter reaches the value 15, the state machine goes into the SyncData state. In this state, the signal nSYNC is driven high to latch the data into the DAC121S101 chips on the PmodDA2.

If the START input signal is low, the machine goes back to the Idle state, ready to accept another conversion.

No matter what the current state is, the RST input signal resets the state machine and puts it in the Idle state.