ADC AD7476A Pmod Controller (VHDL)

Created by Scott Larson, last modified on Jun 17, 2020

Code Download

ADC AD7476A Pmod Controller (top level file): pmod_adc_ad7476a.vhd

SPI Master with Dual MISO (must be included in project): spi_master_dual_miso.vhd

Features

- VHDL source code of a streamlined interface to Digilent's PmodAD1 (Pmod for Analog Devices AD7476A analog-to-digital converter)
- Continually outputs the latest data from both ADC channels on two parallel interfaces
- Handles serial communication and all data retrieval from the ADC Pmod
- Configurable system clock rate

Introduction

This details a VHDL component that handles interfacing to the Digilent's ADC AD7476A Pmod, shown in Figure 1. Figure 2 illustrates a typical example of this ADC Pmod Controller integrated into a system. As shown, the ADC Pmod Controller connects to the Pmod ports and executes transactions to secure ADC data from each of the Pmod's two ADC channels. The data is then presented on two simple parallel interfaces which can be connected to user logic or to output ports on the FPGA.

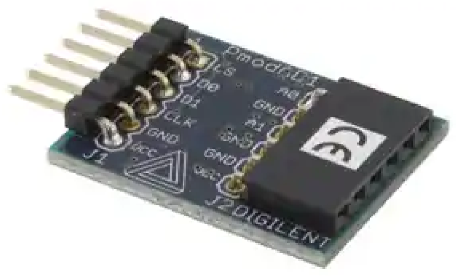

Figure 1. Digilent ADC AD7476A Pmod

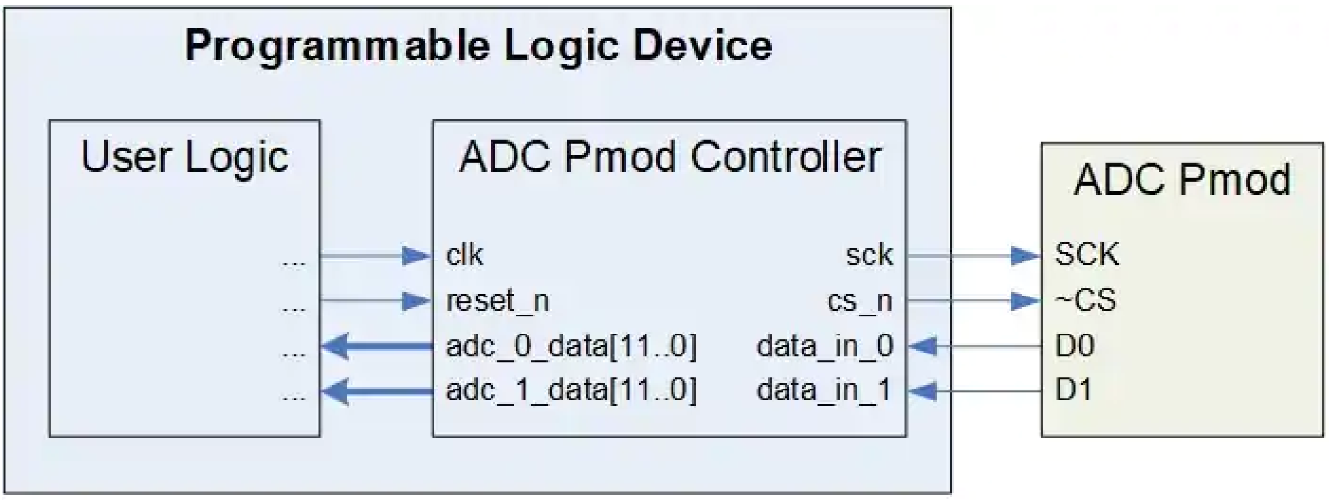

Figure 2. Example Implementation

Background

The ADC AD7476A Pmod uses two of Analog Devices AD7476A 12-bit analog-to-digital converters to provide simultaneous A/D conversion at speeds up to 1 MSPS. These ADCs are wired on this Pmod to share chip select and serial clock signals. The appropriate application of those signals initiates conversions and controls the flow of serial data from the Pmod.

Theory of Operation

The ADC Pmod Controller uses a modified version of the SPI Master component available on eewiki to continually initiate readings and retrieve the resulting data from the ADC Pmod. Once complete, the received data is made available on *adc_0_data* and *adc_1_data* ports. The controller then waits for 50ns and restarts the process (the ADCs require a 50ns quiet time between conversions). This results in continually updated ADC data available to the user with no input required.

The SPI Master is configured with CPOL = 1 and CPHA = 1, to meet the requirements of the AD7476A converters. This SPI Master component was modified to add a second MISO channel that operates as a direct duplicate of the first channel.

Equation 1 defines the frequency at which the output data updates. This is influenced by the system clock frequency the user provides at the *clk* input port and the value assigned to the *spi_clk_div* GENERIC. (See the section Configuring the Clock for details to properly assign *spi_clk_div*). All frequencies are in MHz.

$$(1) \qquad f_{update} = \frac{f_{clk}}{33 \cdot spi\_clk\_div + 0.05 f_{clk}}$$

The ADC Pmod Controller is set up to provide the fastest update rate available for a given system clock, while keeping the serial clock rate within the 20 MHz specified upper limit defined in the AD7476A datasheet. The fastest update rate is achieved when the system clock is an integer multiple of 40 MHz.

Configuring the Clock

The clocking of this ADC Pmod Controller is configured by assigning a value to the GENERIC *spi_clk_div*, defined in the ENTITY. Equation 2 defines how this value is calculated.

$$(2) \qquad spi\_clk\_div = \frac{f_{clk}}{40} \qquad \text{(answer rounded up)}$$

where f_{clk} is the frequency of the provided system clock in MHz. The system clock is provided on the *clk* input port.

For example, the default value specified in the code is *spi_clk_div* = 3. This is arrived at because the most common Digilent development kit at the time of this writing, the Basys3, has a system clock of 100 MHz. 100/40 = 2.5, rounded up is 3.

Equation 3 defines the serial clock frequency f_{sclk} that results.

$$(3) \qquad f_{SCLK} = \frac{f_{clk}}{2 \cdot spi\_clk\_div}$$

This calculation keeps the serial clock at or near the ADC's maximum specified communication frequency of 20 MHz to provide the maximum ADC data update frequency available with a given system clock. This parameter is optimized when the input clock frequency (in MHz) is an integer multiple of 40.

Port Descriptions

Table 1 describes the ADC Pmod Controller's ports.

Table 1. Port Descriptions

Port	Width	Mode	Data Type	Interface	Description
clock	1	in	standard logic	user logic	System clock
reset_n	1	in	standard logic	user logic	Asynchronous active low reset
data_in_0	1	in	standard logic	ADC Pmod	Channel 0 serial data input
data_in_1	1	in	standard logic	ADC Pmod	Channel 1 serial data input
sck	1	buffer	standard logic	ADC Pmod	Serial clock
cs_n	1	buffer	standard logic vector	ADC Pmod	Chip select
adc_0_data	12	out	standard logic vector	user logic	ADC channel 0 data result
adc_1_data	12	out	standard logic vector	user logic	ADC channel 1 data result

Connections

This Pmod has a 6-pin connector, J1. Table 2 provides the pinout for this connector. The ADC Pmod Controller's ports need to be assigned to the FPGA pins that are routed to this connector as listed.

Table 2. ADC Pmod Pinout and Connections to ADC Pmod Controller

Pmod Connector	Pmod Pin Number	ADC Pmod Port	ADC Pmod Controller Port
J1	1	Chip Select (~CS)	cs_n
J1	2	Channel 0 Data (D0)	data_in_0
J1	3	Channel 1 Data (D1)	data_in_1
J1	4	Serial Clock (SCK)	sck

Pmod Connector	Pmod Pin Number	ADC Pmod Port	ADC Pmod Controller Port
J1	5	GND	-
J1	6	VCC	-

Reset

The *reset_n* input port must have a logic high for the ADC Pmod Controller component to operate. A low logic level on this port asynchronously resets the component. During reset, the component aborts the current transaction with the ADC Pmod and clears the *adc_0_data* and *adc_1_data* outputs. Once released from reset, the ADC Pmod Controller resumes operation.

Conclusion

This ADC Pmod Controller is a programmable logic component that interfaces to Digilent's ADC AD7476A Pmod. It handles all communication with this ADC Pmod to provide a continual stream of updated data from both ADC channels on two respective parallel outputs.

Related Topics

Serial Peripheral Interface (SPI) Master (VHDL)

Contact

Comments, feedback, and questions can be sent to eewiki@digikey.com.

No labels

trk