This exercise can be done with your DE0 Partner

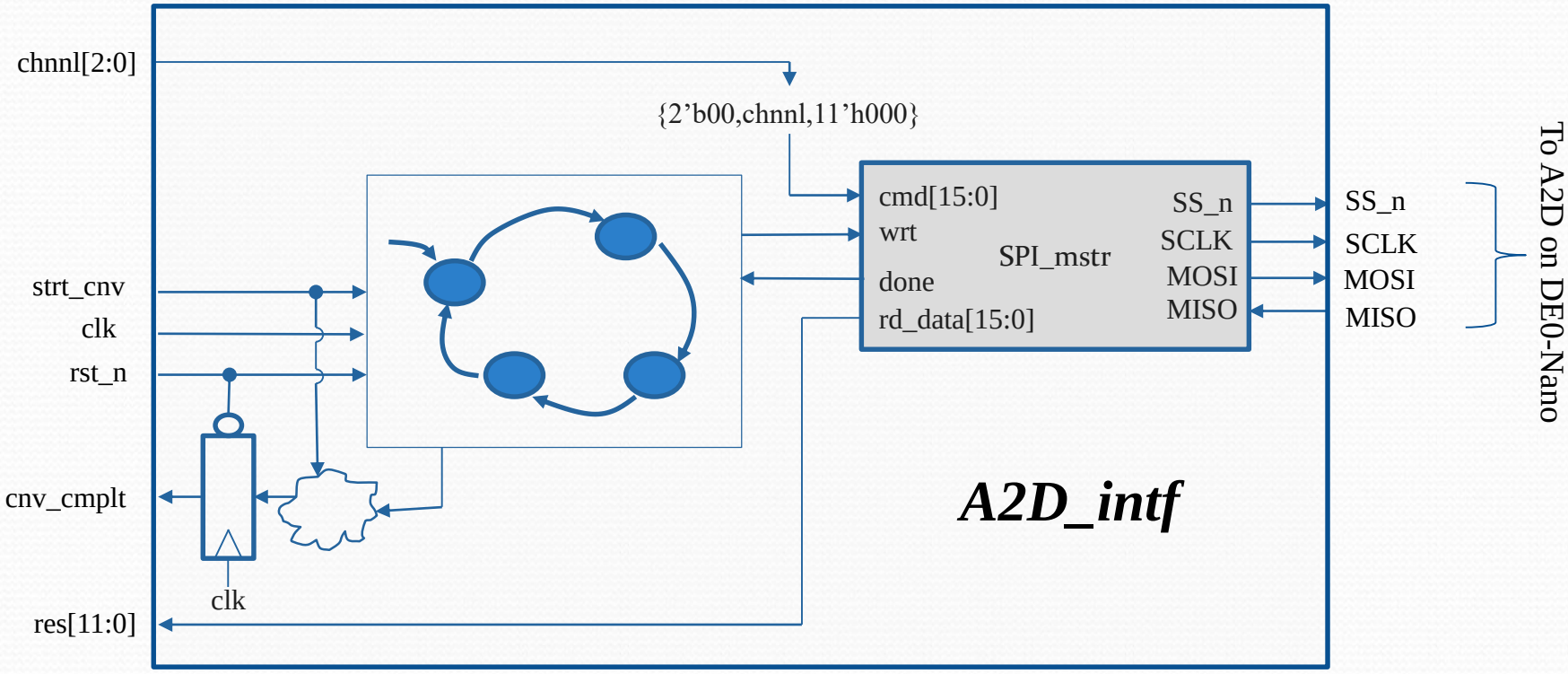
Exercise 11: A2D Intf Design and Test Bench

In HW3 you produced a SPI master (**SPI_mstr.sv**). We are now going to use that block to make an interface to the A2D converter on the DE0-Nano board.

You will be producing a module called **A2D_intf.sv** with the following interface:

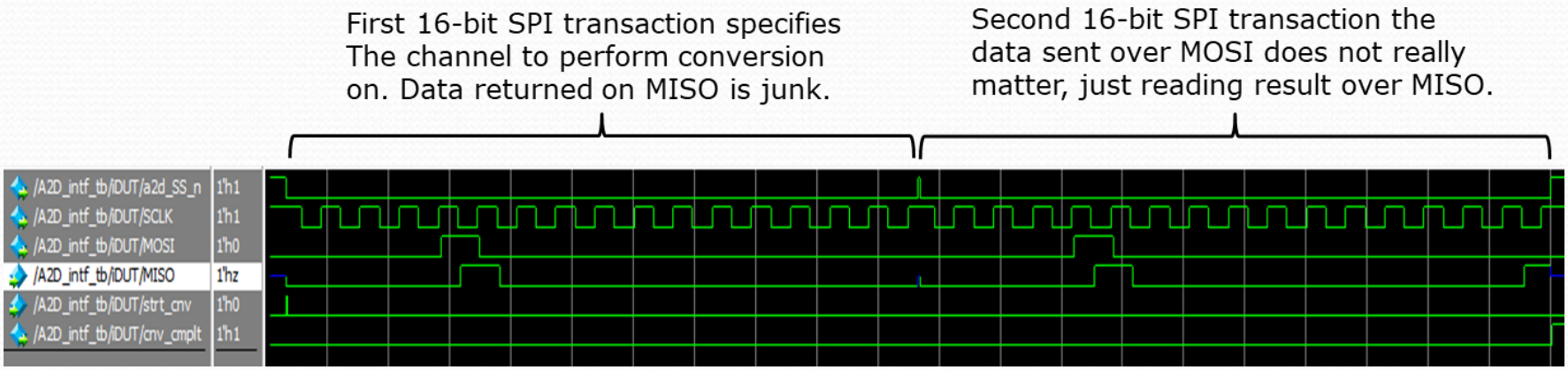
Signal:	Dir:	Description:
clk, rst_n	in	clock and asynch active low reset
strt_cnv	In	Asserted for at least one clock cycle to start a conversion
cnv_cmplt	out	Asserted by A2D_intf to indicate the conversion has completed. Should stay asserted till the next **strt_cnv**.
chnnl[2:0]	in	Specifies which A2D channel (0..7) to convert
res[11:0]	out	The 12-bit result from A2D. (lower 12-bits read from SPI)
SS_n	out	Active low slave select (to A2D)
SCLK	out	Serial clock to the A2D
MOSI	out	Master Out Slave In (serial data to the A2D)
MISO	in	Master In Slave Out (serial data from the A2D)

Exercise 11: A2D Intf Design and Test Bench



SM Function: Send command to A2D via SPI to ask for conversion on channel. Once that transaction completes **wait one clock cycle**. Then start new transaction to read the result of the A2D conversion back.

Exercise 11: A2D Intf Design and Test Bench



Our use of the A2D converter will involve two 16-bit SPI transactions nearly back to back (separated by 1 system clock cycle).

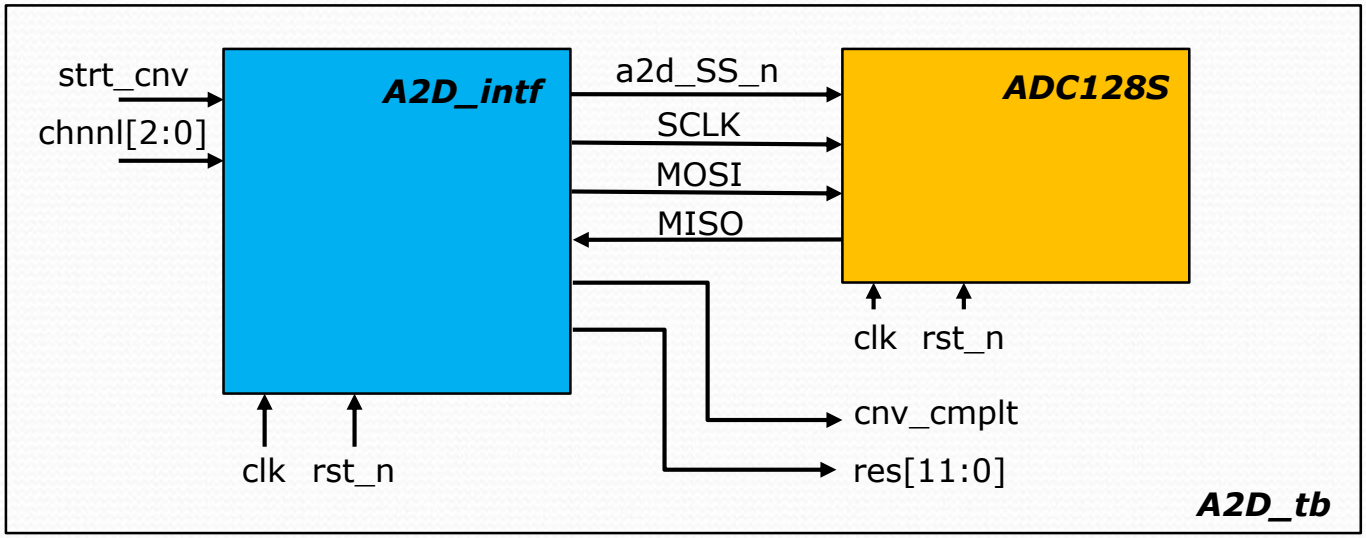
The first transaction here is sending a 0x0800 to the A2D over MISO. The command to request a conversion is {2'b00,channel[2:0],11'h000}. The upper 2-bits are always zero, the next 3-bits specify 1 of 8 A2D channels to convert, and the lower 11-bits of the command are zero. Therefore, the 0x0800 in this example represents a request for channel 1 conversion.

For the next 16-bit transaction the data sent over MOSI to the A2D does not matter that much. We are really just trying to get the data back from the A2D over the MISO line.

NOTE: you need at least a 1 clock cycle pause between the completion of the first SPI transaction and the initiation of the second.

Exercise 11: A2D Intf Design and Test Bench

A model of the A2D converter is provided on the course website (**ADC128S.sv**). Download this and make a test bench that incorporates your A2D_intf and ADC128S.



The value returned from ADC128S will be 0xC00 + chnnl for the first conversion. The 0xC00 part will decrement by 0x010 for each read. So for the second conversion the result should be 0xBF0 + chnnl. Use this knowledge to make the testbench self checking.

Exercise 11: A2D Intf Design and Test Bench

The next exercise will be mapping your **A2D_intf.sv** to the DE0-Nano, so you **must** complete this by next class.

By the end of class submit your **A2D_intf.sv** and your A2D_tb.v files to the dropbox. If they are not complete that is fine…however, **make sure** you finish before next class period.