This exercise can be done as a project team

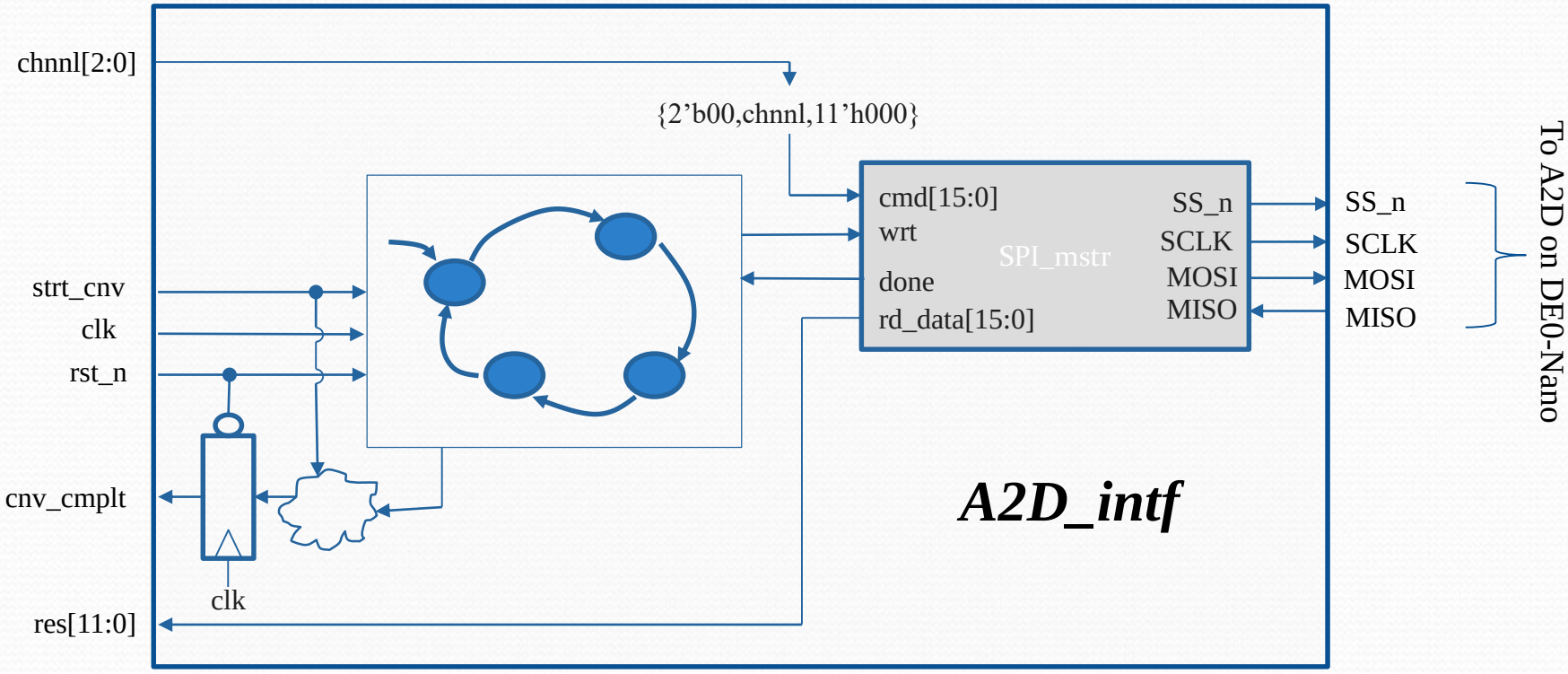
Exercise 17: A2D Intf Design and Test Bench (HW5 Prob3)

In HW4 you produced a SPI master (SPI_mstr16.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	50MHz clock and active low asynch reset
strt_cnv	in	Request to start an A2D conversion. Would come from ***cmd_cfg*** unit.
chnnl[2:0]	in	Channel to convert. In our use case this will always be 3'b000
cnv_cmplt	out	Indicates conversion is complete. Goes to ***cmd_cfg*** unit.
res[11:0]	out	Result of conversion. 12-bits, of which we only use upper 8-bits
SPI Interface	Out/ in	SS_n, SCLK, MOSI, MISO of a SPI interface. Comes from copy of SPI_mstr16 embedded into this unit.

Exercise 17: A2D Intf Design and Test Bench

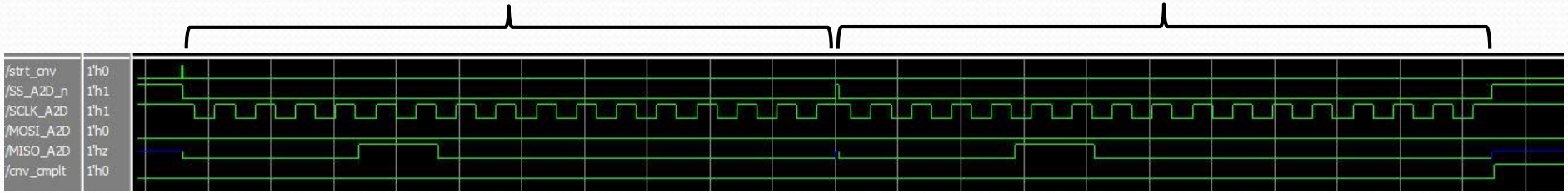


SM Function: Send command to A2D via SPI to ask for conversion on channel. Once that transaction completes wait one clock cycle. Then start new transaction to read the result of the A2D conversion back.

Exercise 17: A2D Intf Design and Test Bench

First 16-bit SPI transaction specifies The channel to perform conversion on. Data returned on MISO is junk.

Second 16-bit SPI transaction the data sent over MOSI does not really matter, just reading result over MISO.



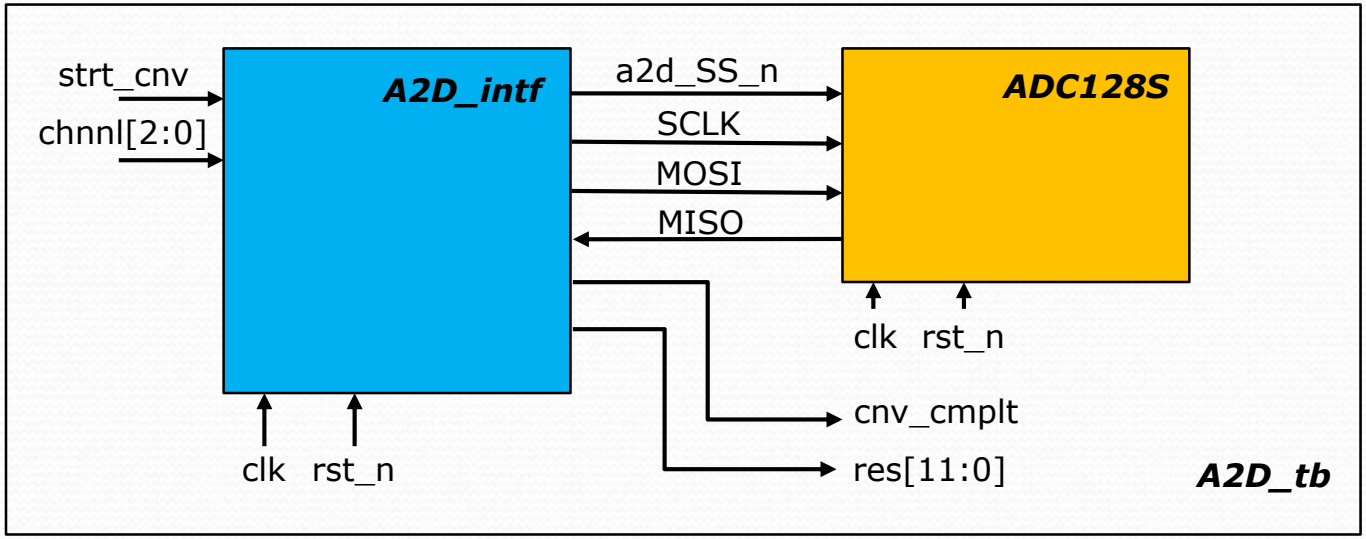
Our use of the A2D converter will involve two 16-bit SPI transactions nearly back to back (separated by 1 system clock cycle).

The first transaction here is sending a 0x0000 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 1:8 A2D channels to convert, and the lower 11-bits of the command are zero. Therefore, the 0x0000 in this example represents a request for channel 0 conversion (channel 0 is battery voltage and is only channel we use).

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. The data we get back in this example is 16'h0C00. Of course since it is a 12-bit converter only the lower 12-bits (12'hC00) is valid. Of this 12-bit result we will only return the upper 8-bits to the remote.

Exercise 17: 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.



NOTE: ADC182S will return 0xC00 for the first reading, and then 0x010 less for every subsequent reading. i.e. 0xBF0, 0xBE0, … if the channel was 0. It will return 0xC01 and then 0xBF1, … if the channel read was 1. If first channel read was 0 and 2nd channel read was 2 answers would be: 0xC00, 0xBF2

Exercise 17: 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 the end of class all team members submit your **A2D_intf.sv** and your **A2D_intf_tb.v** files to the dropbox.

If they are not complete that is fine…however, **make sure** you finish before Monday's class period.