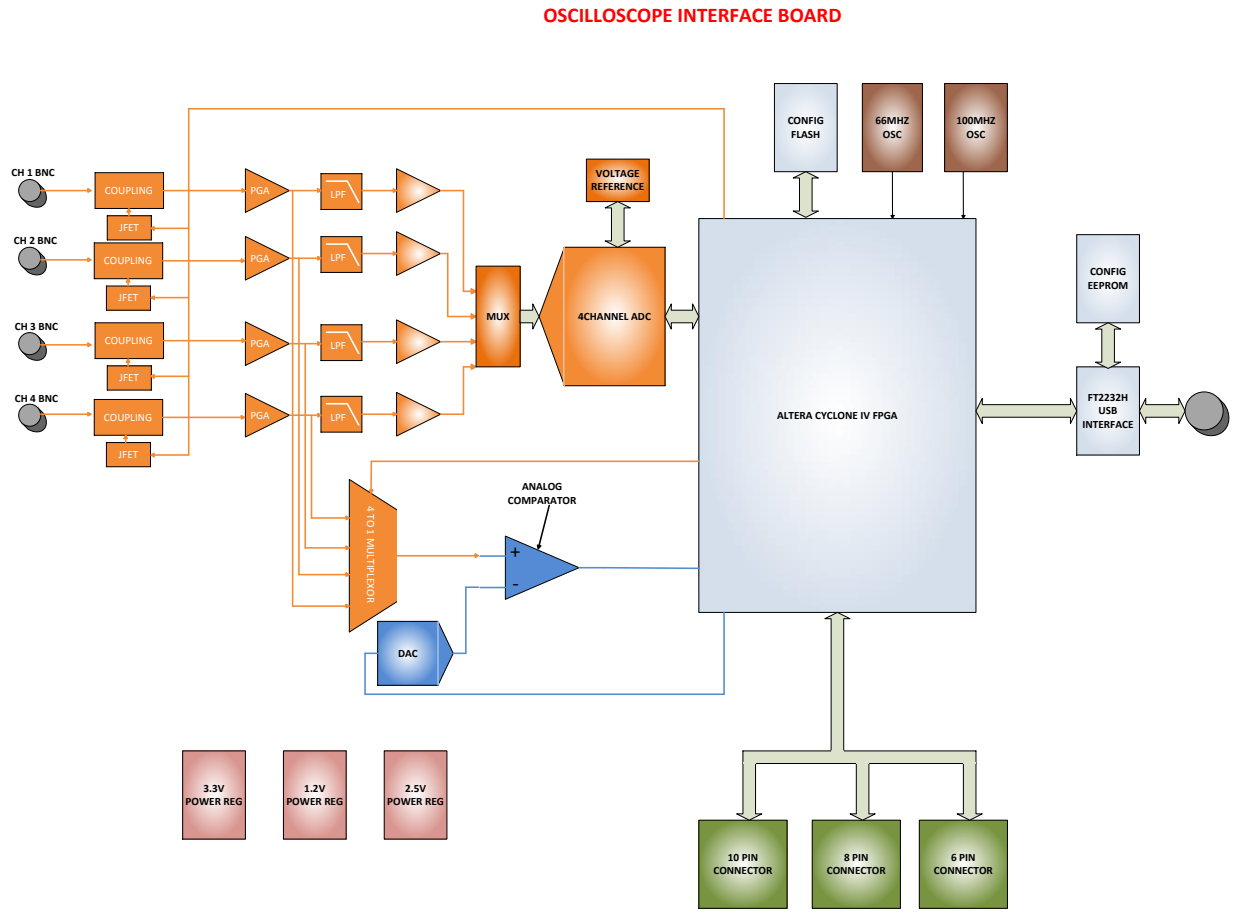
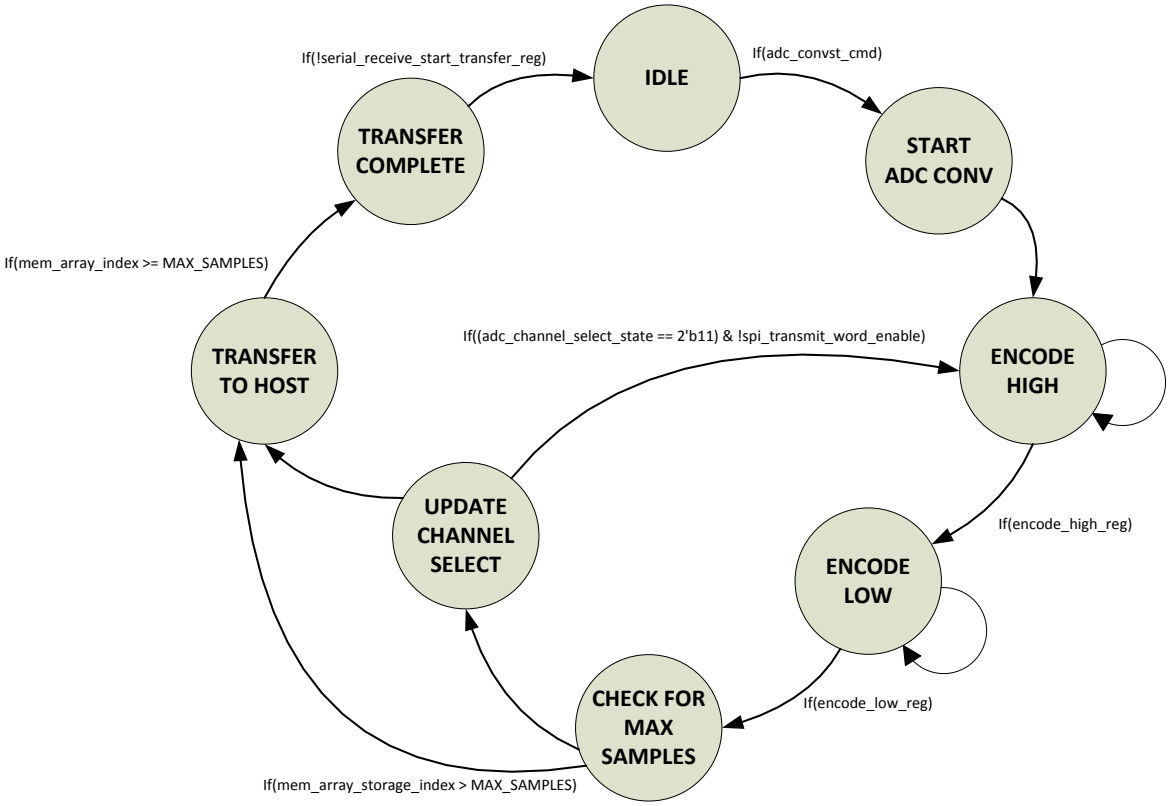
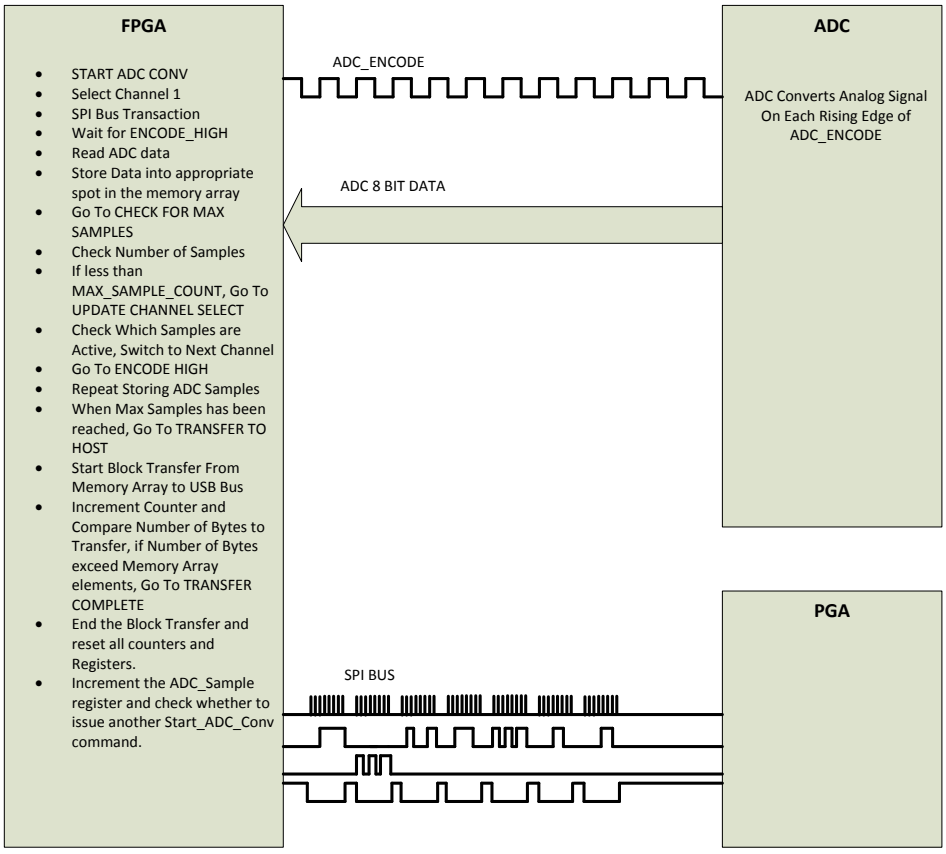
1 EPT DSO System Description

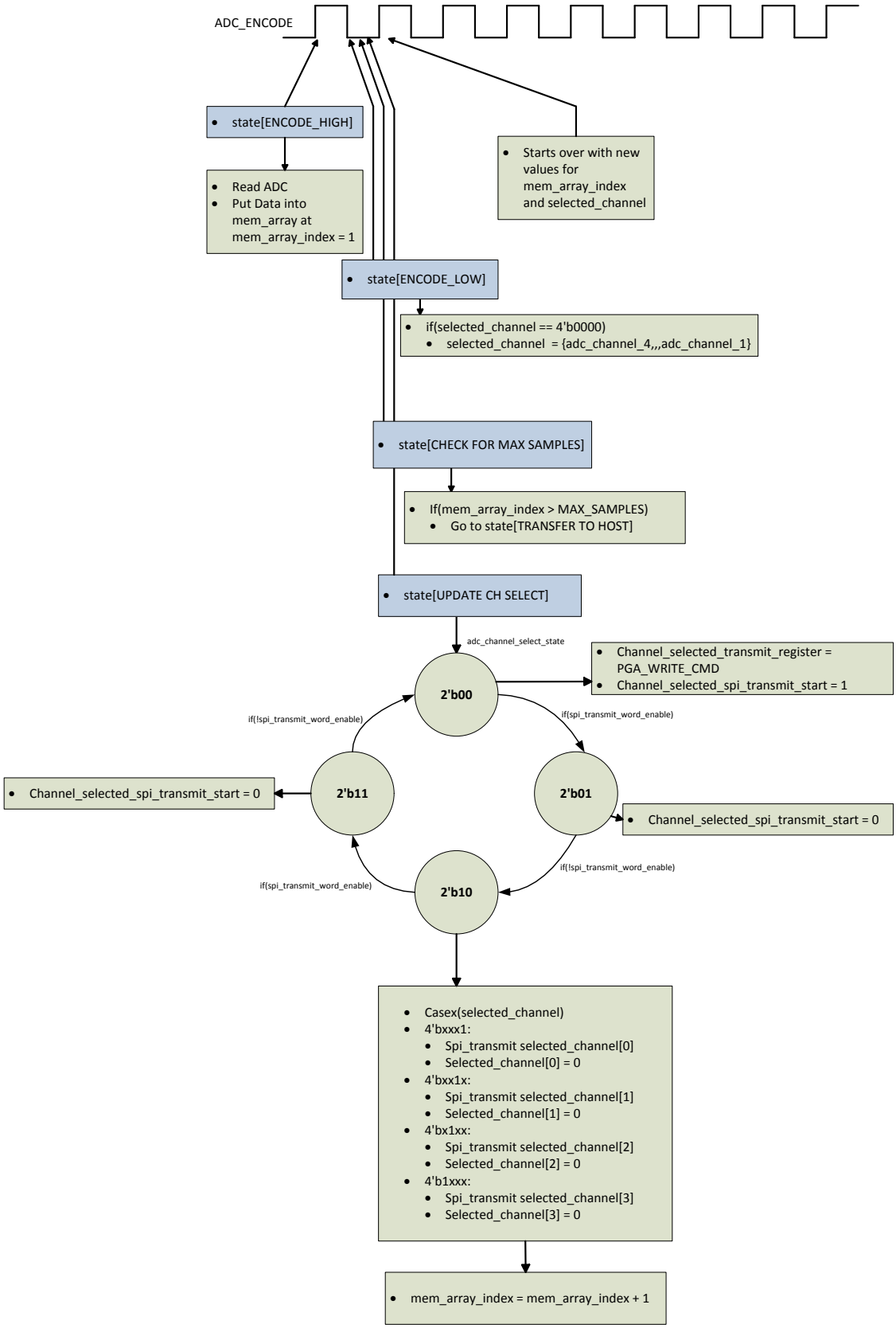


The EPT DSO consists of a 4 analog inputs connected to PGA's and connected to an ADC through a four channel multiplexor. The sampling rate is controlled by the FPGA. The samples from the output of the ADC are collected in the FPGA. They are framed into a memory buffer and transmitted to the PC via a USB to Serial chip.

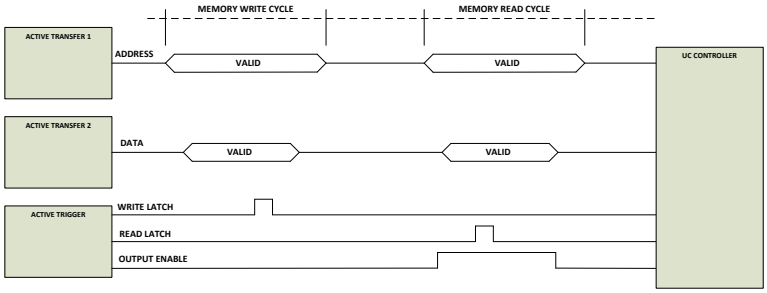

IDLE
If(!serial_receive_start_transfer_reg)
If(adc_convst_cmd)
TRANSFER COMPLETE
START ADC CONV
If(mem_array_index >= MAX_SAMPLES)
TRANSFER TO HOST
If((adc_channel_select_state == 2'b11) & !spi_transmit_word_enable)
ENCODE HIGH
UPDATE CHANNEL SELECT
If(encode_high_reg)
ENCODE LOW
CHECK FOR MAX SAMPLES
If(encode_low_reg)
If(mem_array_storage_index > MAX_SAMPLES)

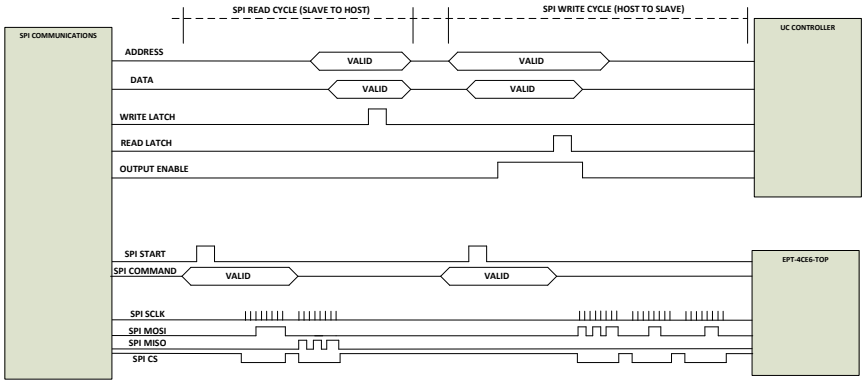

FPGA
START ADC CONV
Select Channel 1
SPI Bus Transaction
Wait for ENCODE_HIGH
Read ADC data
Store Data into appropriate spot in the memory array
Go To CHECK FOR MAX SAMPLES
Check Number of Samples
If less than MAX_SAMPLE_COUNT, Go To UPDATE CHANNEL SELECT
Check Which Samples are Active, Switch to Next Channel
Go To ENCODE HIGH
Repeat Storing ADC Samples
When Max Samples has been reached, Go To TRANSFER TO HOST
Start Block Transfer From Memory Array to USB Bus
Increment Counter and Compare Number of Bytes to Transfer, if Number of Bytes exceed Memory Array elements, Go To TRANSFER COMPLETE
End the Block Transfer and reset all counters and Registers.
Increment the ADC_Sample register and check whether to issue another Start_ADC_Conv command.
ADC_ENCODE
ADC 8 BIT DATA
SPI BUS
ADC
ADC Converts Analog Signal On Each Rising Edge of ADC_ENCODE
PGA

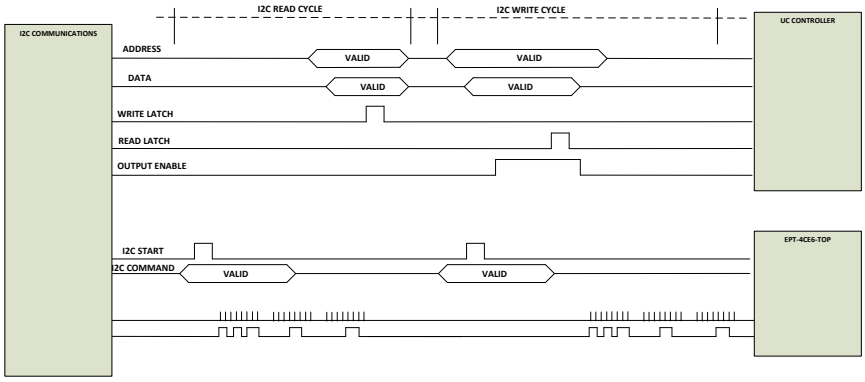
ADC SAMPLE STORAGE STATE MACHINE FOR THE OSCILLOSCOPE BOARD

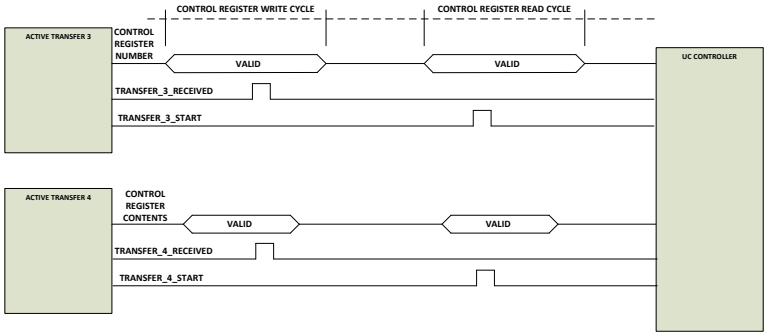


TIMING DIAGRAM FOR THE MEMORY, SPI AND I2C BUS ACCESS

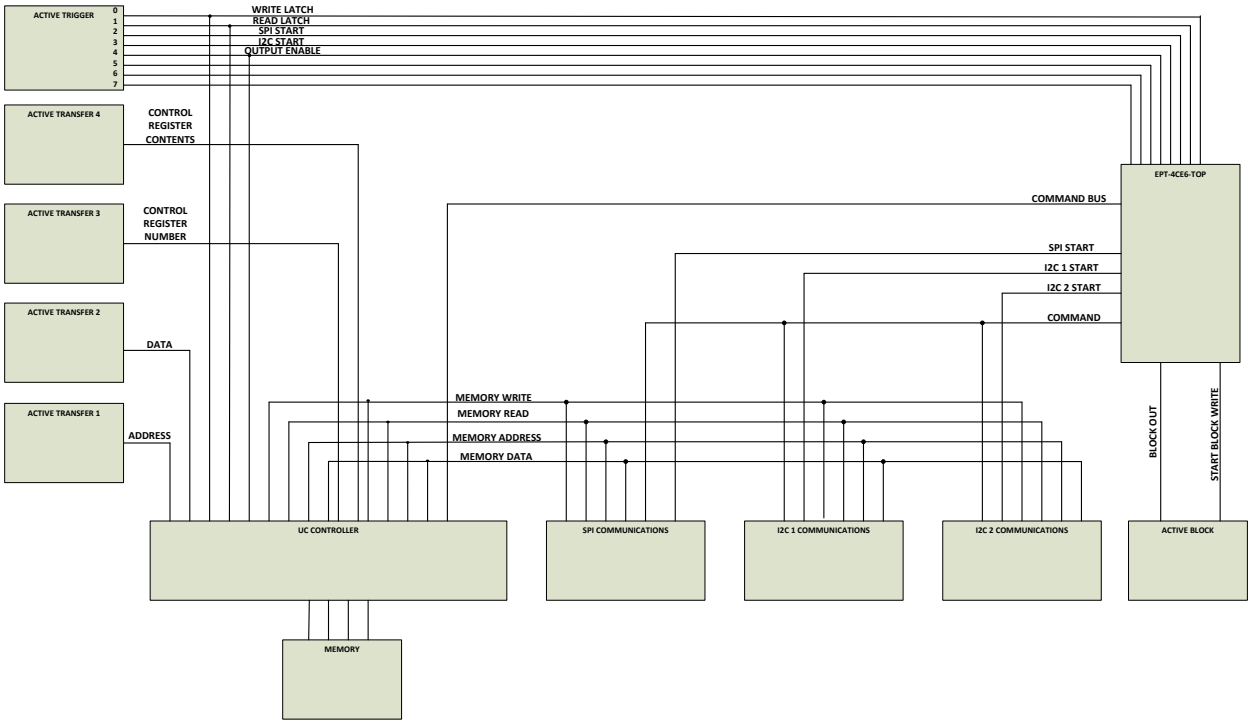




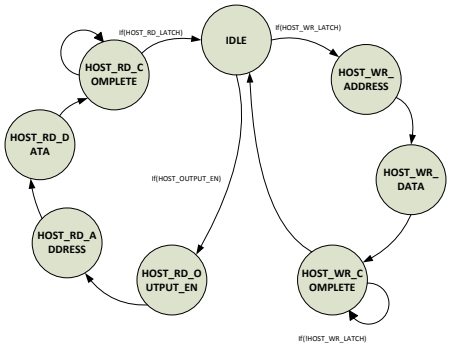




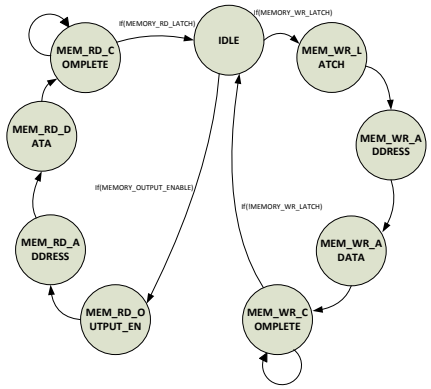
VERILOG MODULES FOR THE OSCILLOSCOPE BOARD



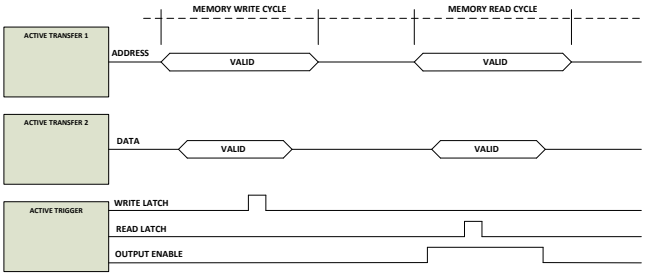
HOST INTERFACE STATE MACHINE



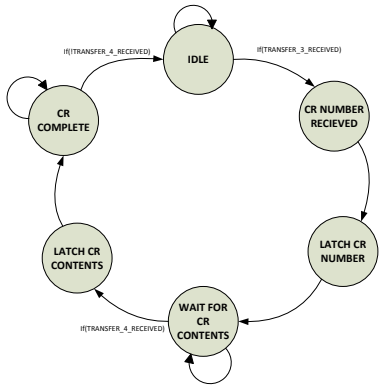
LOCAL MEMORY STATE MACHINE

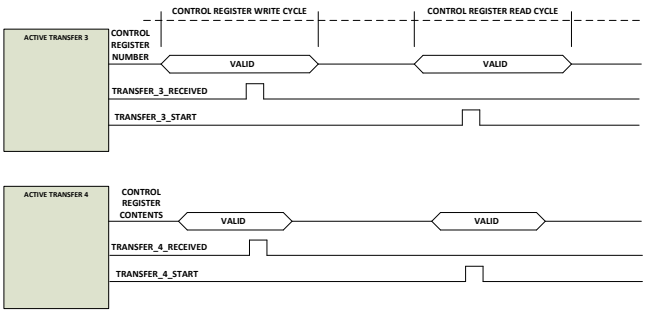


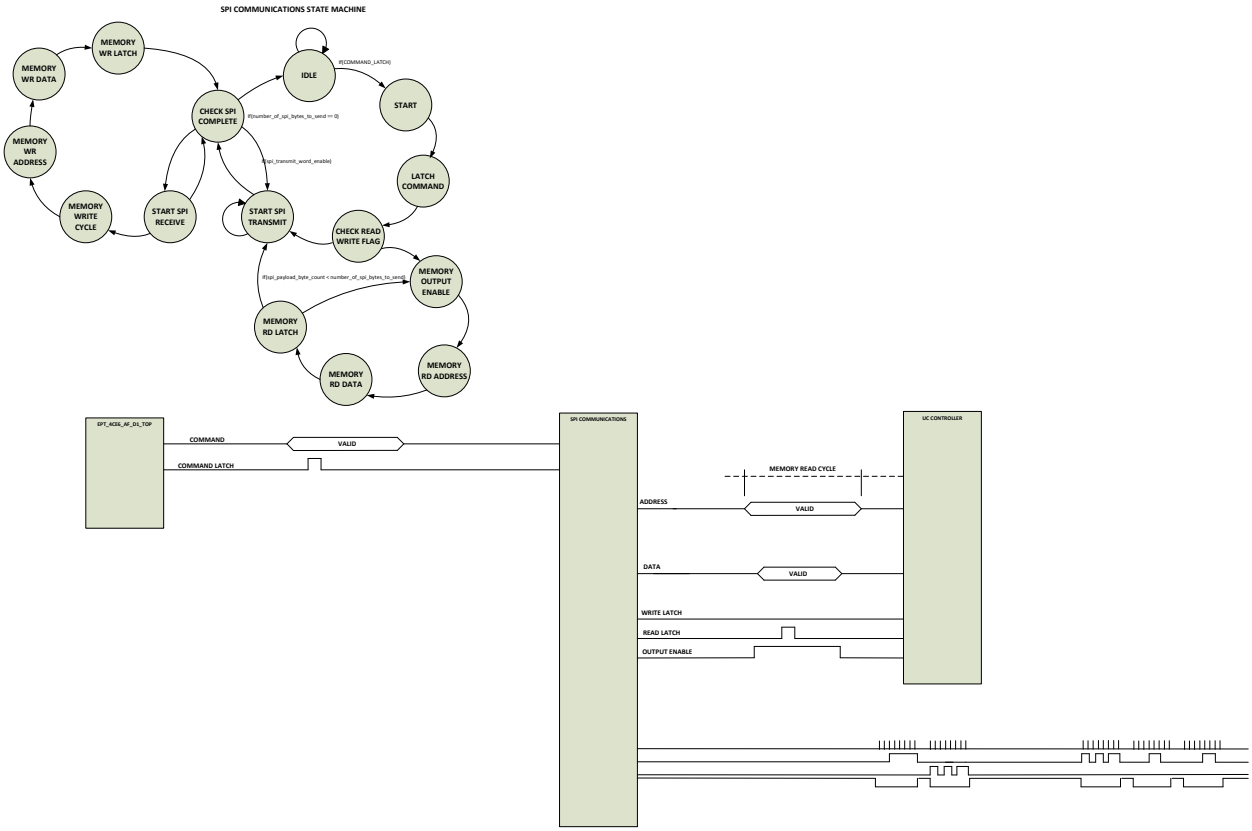
TIMING DIAGRAM FOR THE UC CONTROLLER



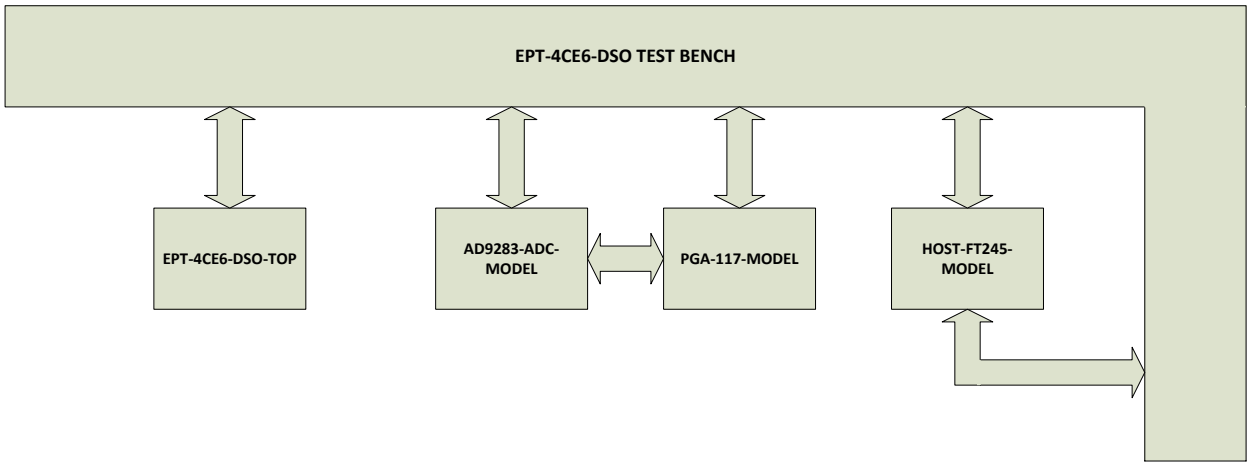
CONTROL REGISTER STATE MACHINE







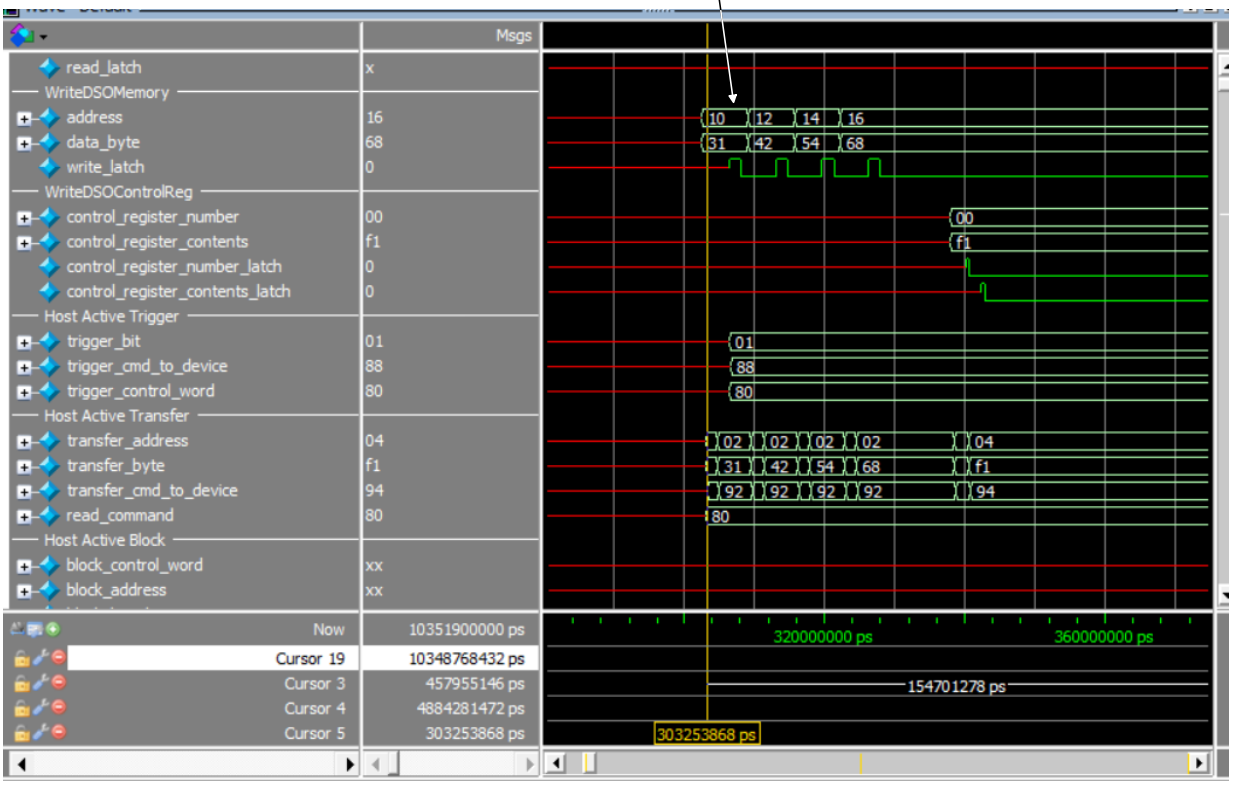
2 Verilog Simulation



Test Bench Code Writes Four PGA Channel Gain Values into four different Memory locations

The Task, WriteDSOMemory is used to command the HOST-245-MODEL to write these values through the Active Transfer Library Interface.

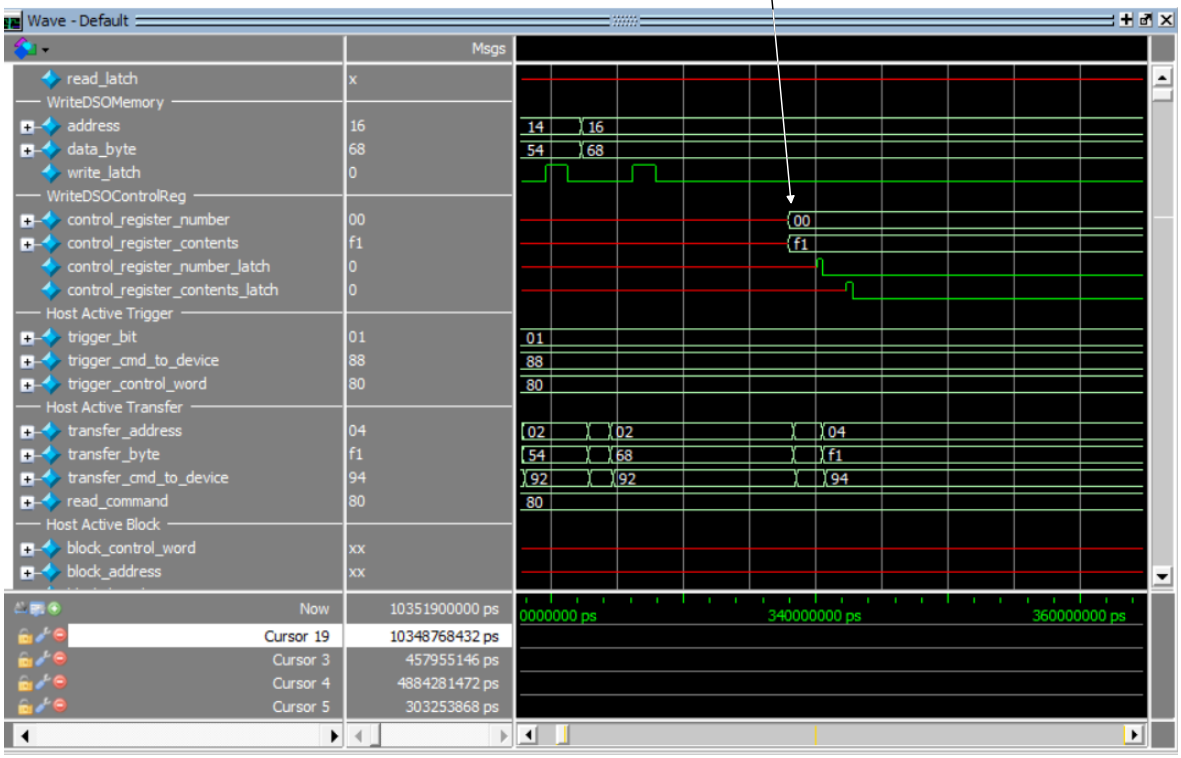
This emulates the code running on the Windows PC communicating with the EPT DSO



Test Bench Code Writes to the Control Register. This selects four channels to scan and starts the ADC sampling

The Task, WriteDSOControlReg is used to command the HOST-245-MODEL to write this register through the Active Transfer Library Interface.

This emulates the code running on the Windows PC communicating with the EPT DSO



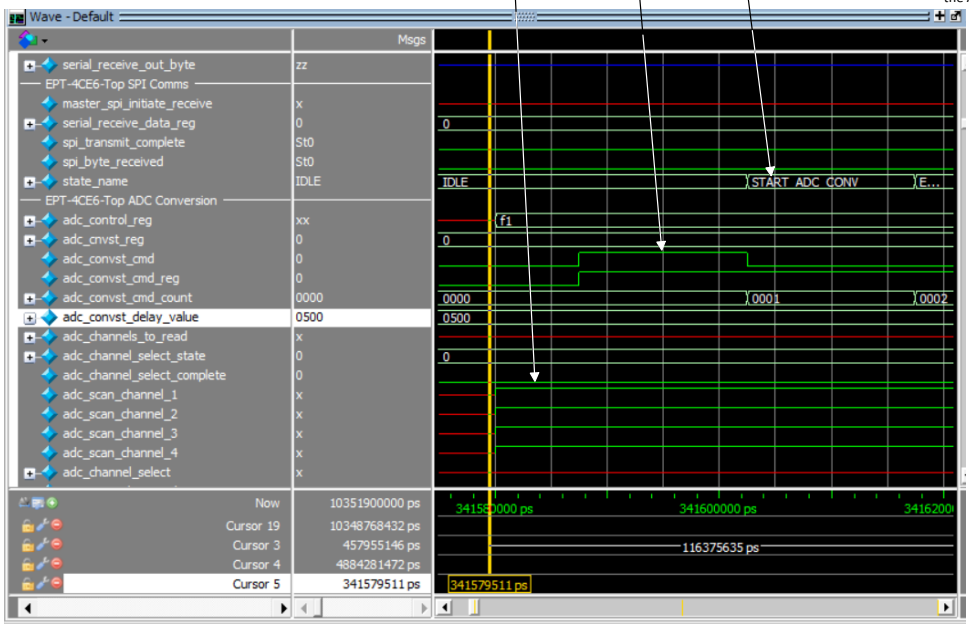
All four channels are selected, by going high.

The code in the EPT-4CE6-TOP loads the delay for the sampling rate then asserts the adc_convst_cmd which starts the State Machine.

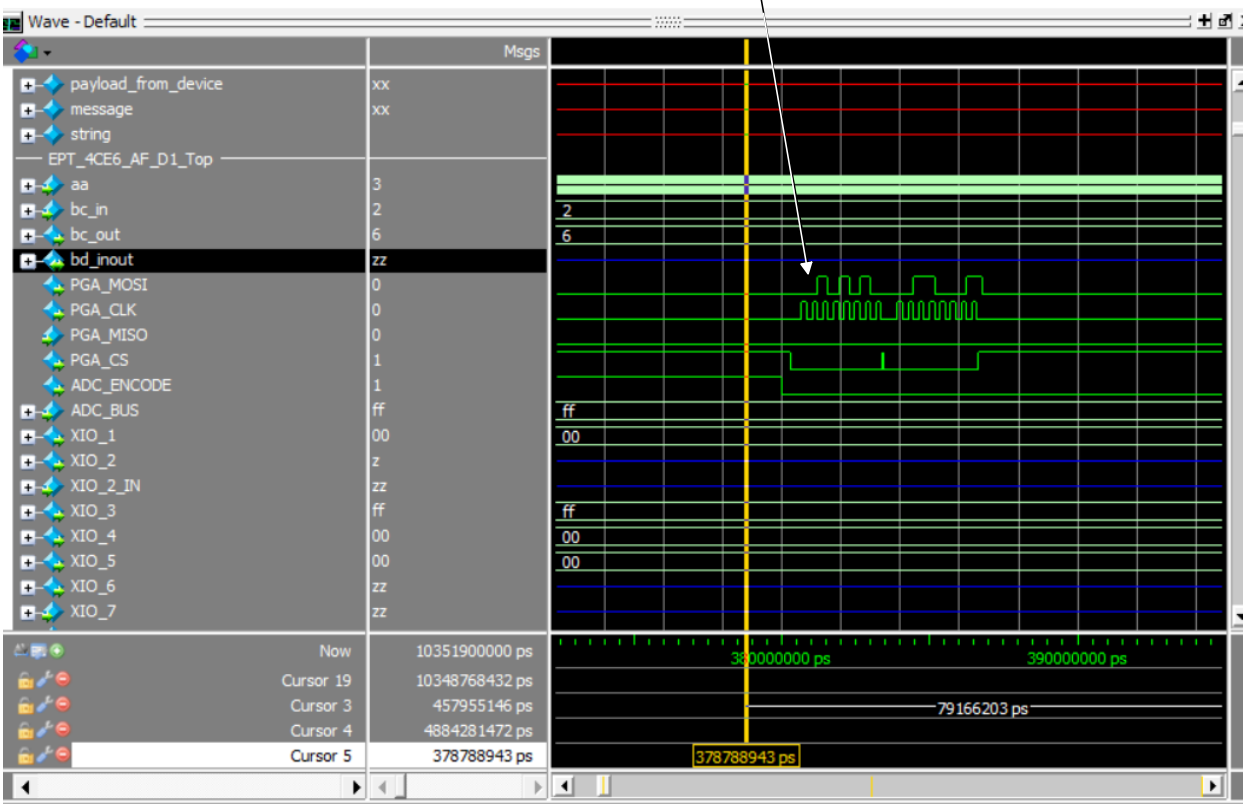
The Main State Machine in EPT-4CE6-TOP goes to the "START_ADC_CONV" state.

This starts the cycle of selecting the correct PGA channel. Then reading the previous sample from the ADC.

Then finally, delaying the ADC-ENCODE signal to match the selected sample rate and starting the cycle over again.



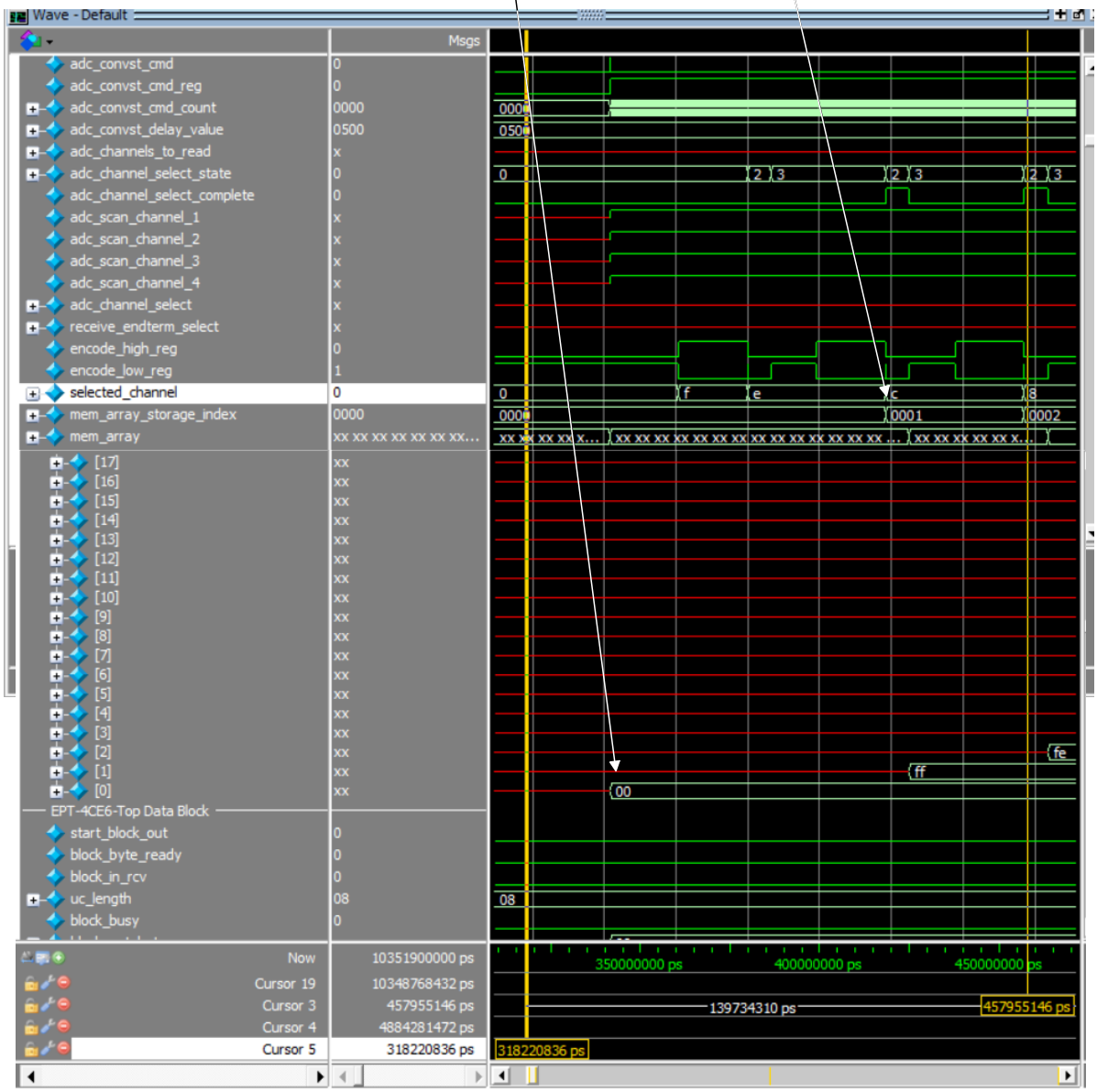
The SPI Communications function in the EPT-4CE6-TOP sends the select channel and gain to the PGA via SPI Bus.



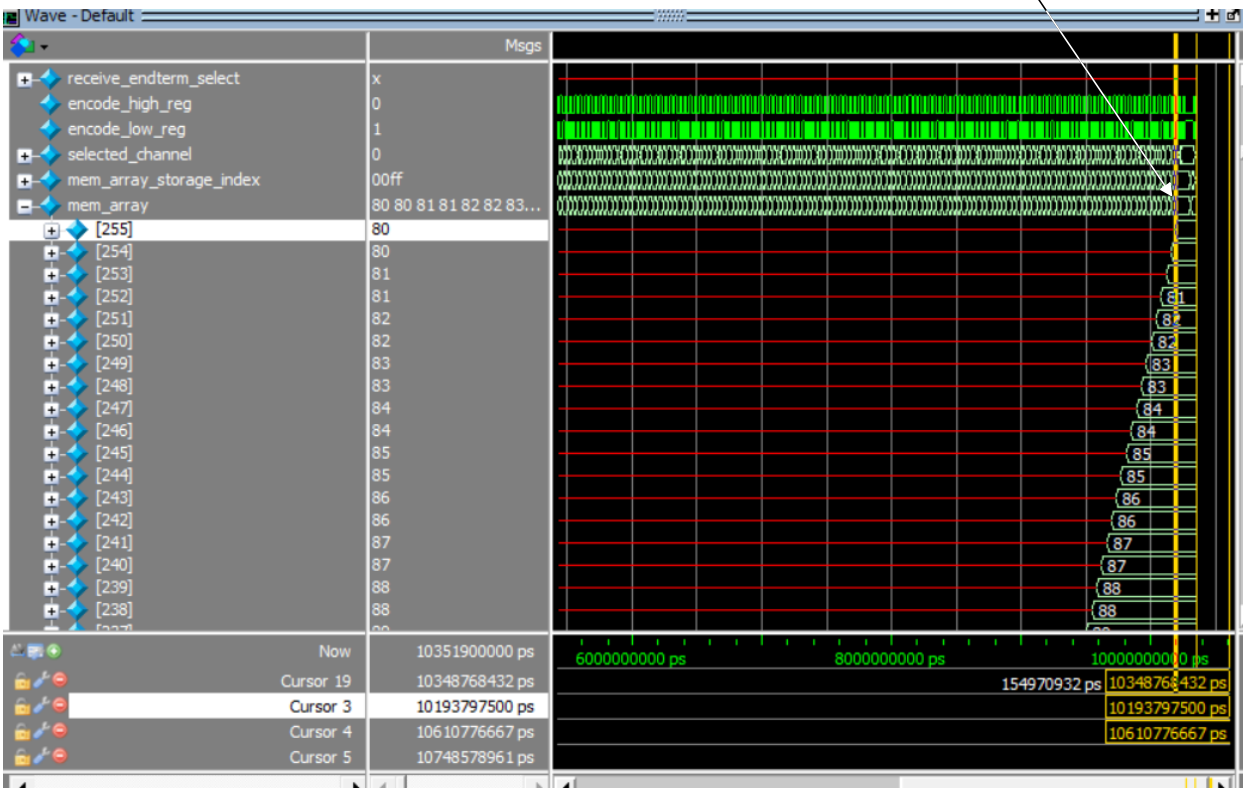
The sample storage code in the EPT-4CE6-TOP stores the value from ADC_DATA into the local memory array.

The State Machine cycles through ENCODE_HIGH, ENCODE_LOW, CHECK_MAX_SAMPLES, UPDATE_CH_SELECT and starts over at ENCOED_HIGH.

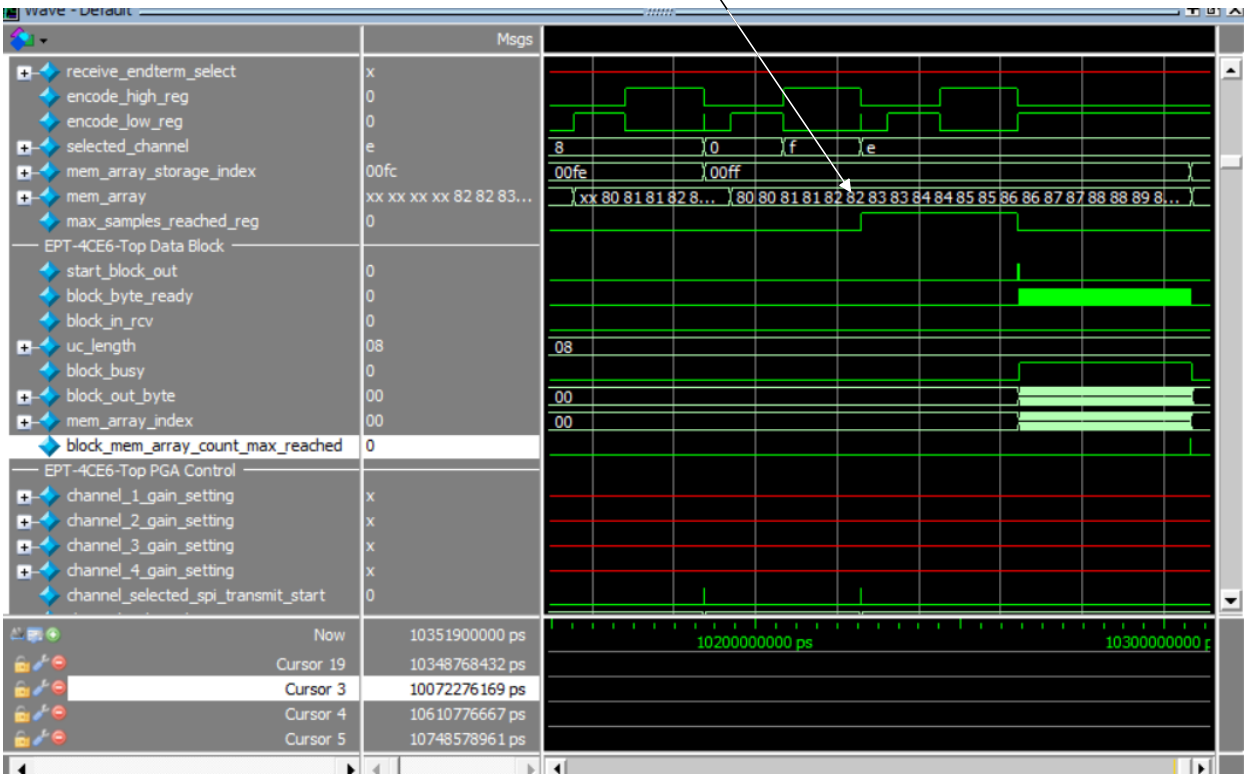
The mem_array_storage_index increments to allow the next ADC sample to be stored in next cell in the memory array.



The State Machine continues through its cycles until mem_array_storage_index reaches: 0x00ff or a count of 256 samples

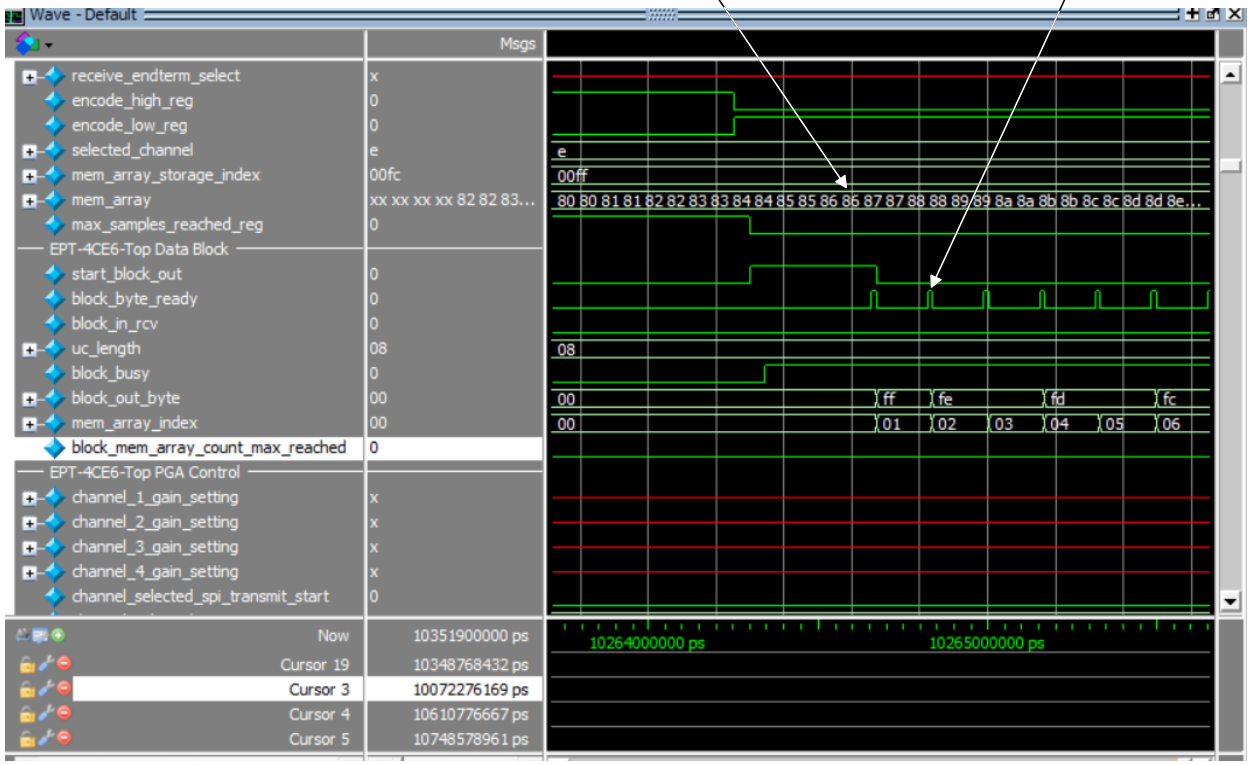


Max_samples_reached_reg is asserted when mem_array_storage_index reaches 0x00ff. The State Machine goes into TRANSFER_TO_HOST state.



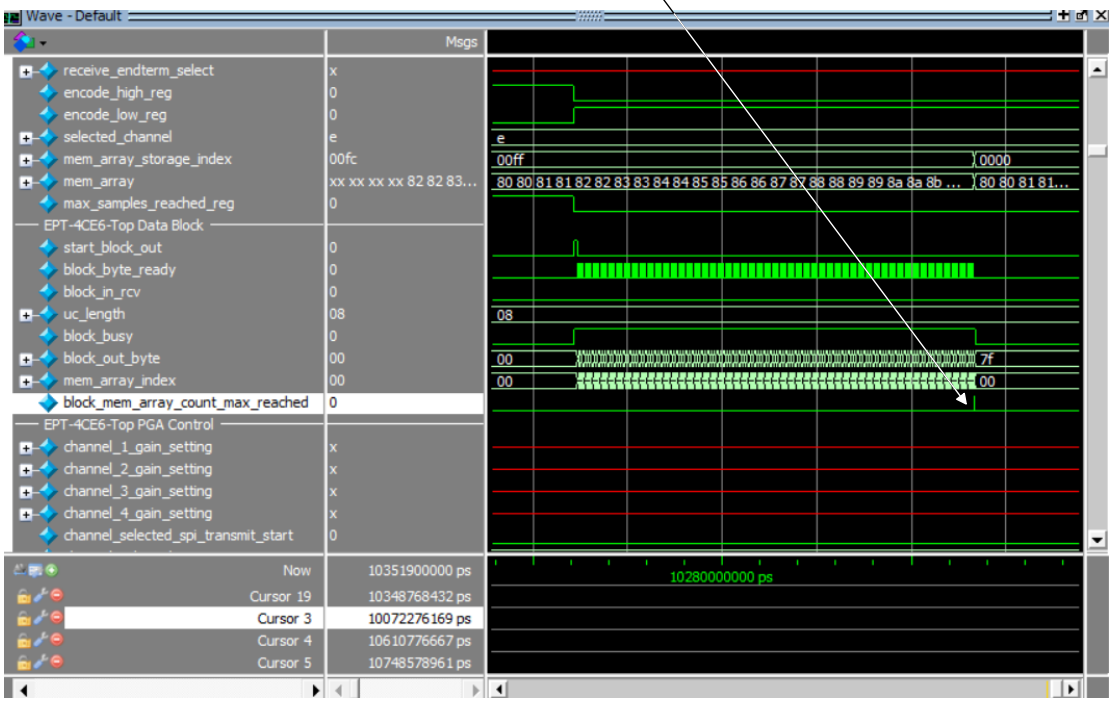
The code in EPT-4CE6-TOP asserts the start_block_out signal and the Active_Block EndTerm will assert the block_byte_ready signal.

At each block_byte_ready assertion, the code should send a byte from the memory array into the Active_Block EndTerm



The code continues to increment mem_array_index and supply the next byte in the memory array to the Active_Block EndTerm until block_mem_array_count_max_reached asserts.

At this point, all the bytes in the memory array have been transferred into the Active_Block EndTerm. The code in EPT-4CE6-Top returns to idle conditions. The State Machine cycles to HOST_TRANSFER_COMPLETE and back to ENCODE_HIGH. This starts the whole sample, store, and transfer cycle over.



The Active_Block EndTerm communicates with the Active_Transfer_Library to read in the bytes from the memory array then transfers them to the HOST_FT245_MODEL.

This transactions emulates the FPGA communications with the Windows PC.