CSCE 3304 – Digital Design II
FALL 2019

Homework Assignment 1

Guidelines:

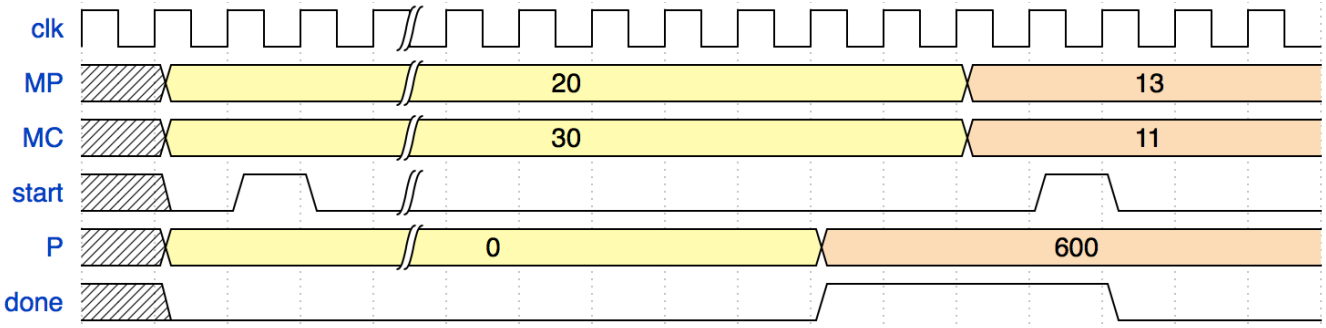
1) Use CloudV (www.coudv.io) to develop, simulate and synthesize the module(s) for Problem 1.
2) The CloudV workspace has to be private. A source code similarity checker will run against your submitted code.
3) You must synthesize your RTL models using the OSU035 library. The synthesis reports must be examined for errors (e.g., latches in combinational block).
4) Problem 2 has to be done using your own text editor, Icarus Verilog and GTKWave.
5) Your source files must comply with the course coding guidelines (attached).
6) The tesbenches should be self-checking; meaning, the testbench should verify the correctness of the output generated by the DUT.
7) 50% of the grade is for the Verilog model. The other 50% is for the Verification testbenches.
8) The submission will be done through BB. Your submission must include a readme.txt file that contains the shared CloudV repo for problem 1 and problem2.zip file for problem 2.
9) The submission deadline is Monday September 16th 8:00AM.

Problem 1: Serial-Parallel Multiplier [50%]

Using Verilog, model and verify a 32-bit signed serial parallel multiplier outlined by the attached document. The design outlined by the document does not cover the interface to the multiplier. The multiplier is required to have the following interface:

Port	Direction	Size (bits)	Description
MP	In	32	The multiplier. Is stable before the ***start*** is asserted and may change after ***done*** is asserted.
MC	In	32	The Multiplicand. Is stable before the ***start*** is asserted and may change after ***done*** is asserted.
start	In	1	Indicates that ***MP*** and ***MC*** are stable to start the multiplication (active only for 1 clock cycle)
P	Out	64	The product. Must be ready before ***done*** is asserted.
done	Out	1	Indicates that the product (*P*) is ready. De-asserted when ***start*** is de-asserted.
Clk	In	1	The clock. All events are in synch w/ the clock positive edge
resetn	In	1	Asynchronous reset. Active low.

The following is an example timing diagram to make things clearer:



Problem 2: SAR ADC Controller [50%]:

SAR (Successive Approximation Register) ADC (Analog to Digital Converter) is a type of DAC that converts a continuous analog waveform into a discrete digital representation via a binary search. The following block and timing diagrams outline the function of the SAR ADC. There are 4 components: Sample and Hold (S&H), Digital-to-Analog Converter (DAC), Analog comparator and the controller. The S&H unit samples the analog signal and holds the sampled value till the conversion ends. The controller sends the "***sample***" control signal to the S&H unit to sample the analog signal. "***sample***" is active for one clock cycle. The DAC converts the 8-bit value generated by the controller to an analog value. The comparator compares the output from the DAC to the S&H sampled and held analog value to generate a digital output "***cmp***" (0: DAC output >= S&H output, 1: otherwise).

The controller implements the logic of conversion by updating the content of an 8-bit register (called the SAR). Over 8 clock cycles the controller updates the SAR content bit by bit starting from the most significant bit based on the output of the comparator (***cmp***). On each clock edge, the current bit is set to 1 and the previous bit is set to the value of the ***cmp*** signal. Initially, the SAR has the binary value 10000000.
The controller has an input to start the conversion (***go***) and an output (***valid***) to indicate the availability of the converted sampled data (***result***). ***valid*** is active once ***result*** has the conversion final value. It stays active till ***go*** goes active again to start the next conversion. The output ***sample*** controls S&H unit. To sample analog data, ***sample*** has to be active for 1 clock cycle. The output ***value*** reflects the current value of the SAR (note: ***result*** matches ***value*** at the end of conversion). To summarize, the SAR ADC needs 10 cycles: 1 to sample the analog signal; 8 to convert; 1 to copy the ***value*** to ***result***. All events are synchronized with the rising edge of the clock. The reset signal is asynchronous and active low.