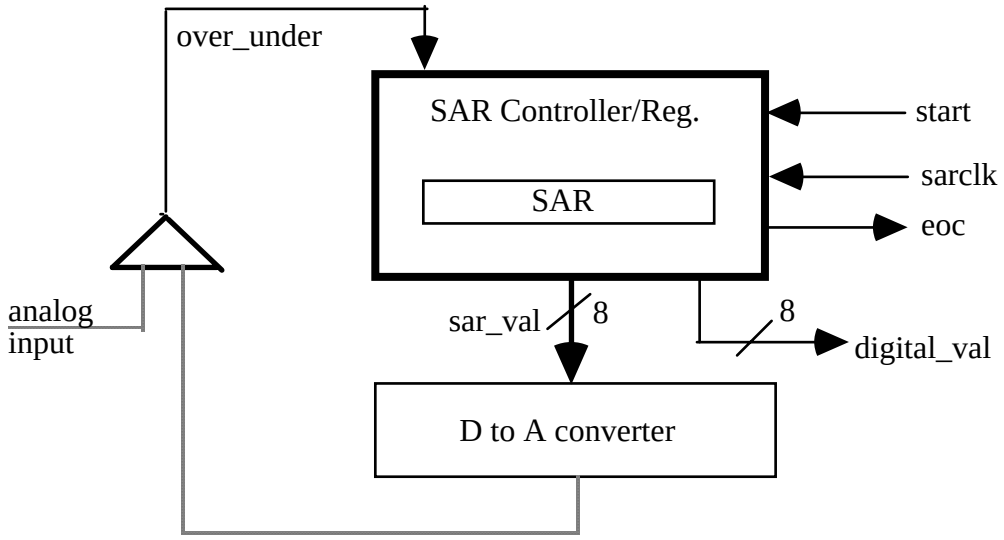
PROJECT STEP SEQ_EX - Sequential Machine Example **Due: Monday May 6**

In this problem you are to use VHDL to model a Successive Approximation A to D converter controller.

The test bench for this problem is in ~/ee762_assign/sar_tb.vhdl
The do file for simulation is sar_tb.do

Successive Approximation A to D is achieved by setting a Successive Approximation Register to the 50% value, i.e., "10000000". This is then converted to analog in a D to A converter. The analog input is compared to this value. If the analog input is higher than the output of the D to A, over_under indicates that the value is under and the SAR register is updated to "11000000". If the analog input is less than the DAC output, over_under indicates that the DAC value is over and the next SAR value is set to "01000000". This continues bit by bit until the A to D conversion is complete.

The testbench for this problem models the D to A converter and the analog comparator which outputs the digital over_under signal (DAC over = '1', DAC under = '0').
You are to construct a model of the SAR Controller/SAR register digital subsystem.
The ports you will need are shown in the following diagram.



Basic operation for a conversion is as follows.
When start transitions high the analog input is valid and is captured by a capture and hold circuit. This aspect of the system is not directly modeled. The analog input value will remain constant during the conversion.
Your subsystem will transition through the value for the SAR and when you have finished the 8 bits of the sar_var, eoc will be asserted. At the same time eoc is asserted the digital value is output.
When start returns low in response to eoc going high, eoc is reset.

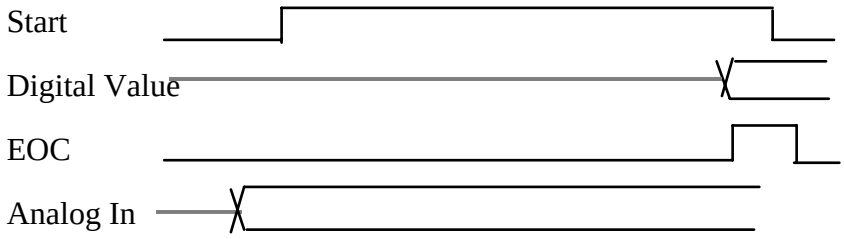
When writing the VHDL code to model the SAR Controller/Register refer to the lecture on state machine modeling. Use a process for the latching of values from next_values signals. Then

update the next_values in a second process based upon state. This makes the problem very simple.

Note that when you simulate the testbench you cannot simulate until time'high as the testbench has a free running clock, sarclk, and never becomes quiescent. The easy way to see all of the testbench simulation is to enter the command

>run 20 us

SIGNAL TIMING



As always, turn in:

a) A copy of the VHDL source.

b) A copy of the waveform using zoom->full_size

c) A copy of the waveform for the first two conversions

d) A copy of the listing file

Note: The state machine coding style should be used for this problem.