**LAB 5 Instructions:**

Deadline for lab 5 is same as lab 4

Late Penalty is 10% deduction for the first day, 30% deduction for two days.

Lab 5 top\_level.sv is a template, that guides you to write a synthesizable Verilog. The Verilog file provided can be simulated on modelsim but cannot be synthesized in Quartus. For example, probes into memory such as dm1.core cannot be synthesized. You will need to assign the right read/write address and use data\_in/data\_out to write/read to/from the memory, making it a synthesizable code.

Your project and file structure will look practically identical to Lab4. You can reuse lfsr and data\_mem. Hint: I found it simplest to instantiate 6 copies of LFSR, each hardwired with a different LFSR feedback tap pattern. During the training (preamble input) phase, you simply launch all of them together based on data\_mem[64]^0x1f (bits [5:0] of 0x5f ASCII underscore), then see which one continues to match data\_mem[65]^0x1f, … .

You can have state machines similar to lab 4, where your job is to read from memory where you encrypted message is stored, find the start, taps to be used for LFSR. Then decrypt and store in appropriate locations of the memory.

You need to submit a synthesizable top\_level.sv, top level RTL viewer, waveform, and readme (that describe what you have done in couple of lines)