LAB 7: PSEUDO-RANDOM NUMBER GENERATOR

For this lab you are required to design an 8-bit pseudo-random number generator.

A pseudorandom number generator is an algorithm for generating a sequence of numbers whose properties approximate the properties of sequences of random numbers. The PRNG-generated sequence is not truly random, because it is completely determined by a relatively small set of initial values. (Source – Wikipedia).

Steps to follow:

1. Declare libraries
2. Declare your entity with CLOCK as input and 8 bit output
3. Begin your architecture
4. Start a process
5. Declare two temporary variables, one as long as your output, rand and the other a single bit, temp
6. Initialize your random such that the MSB is ‘1’ and all the other are ‘0’ (to get a random number, can be different too)
7. The single bit variable temporary is initialized to ‘0’
8. Begin process
9. Start an if statement on rising of clock
10. XOR the MSB and (MSB-1) term random and assign it to the single bit variable, temporary
11. Assign the values in random((MSB-1) to 0) to random(MSB to 1) and also assign value in temporary to random(0)
12. End your if statement
13. End your process and architecture

Since you need a random number sequence you need not follow the same steps you can write different code to get the randomness.
For implementing it on the FPGA you also need a clock-divider without which you cannot observe the random sequence on your board.

The simplest clock-divider is a using a counter. For our design we will use a 32 bit counter for the clock division.

The clock divider will have input CLOCK and a single bit output. The output is one of the counter output bit (which will be signals here for our use).