LFSR Core Specification

Author: Vipin Lal
lalnitt@gmail.com
July 31st, 2010

The lfsr_randgen project is a random number generator module which is based on the principle of LFSR(linear feedback shift registers). A LFSR is a shift register whose input bit is a linear function of its previous state. The only linear function of single bits is xor, thus it is a shift register whose input bit is driven by the exclusive-or (xor) of some bits of the overall shift register value.

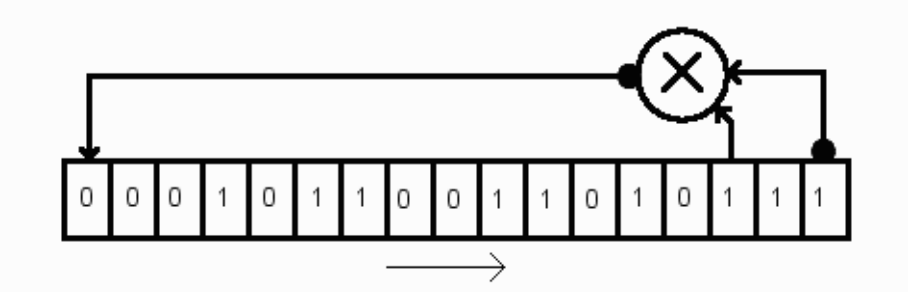
The initial value of the LFSR is called the seed, and because the operation of the register is deterministic, the stream of values produced by the register is completely determined by its current (or previous) state. Likewise, because the register has a finite number of possible states, it must eventually enter a repeating cycle. However, an LFSR with a well-chosen feedback function can produce a sequence of bits which appears random and which has a very long cycle. So we should be careful when using this random generator for any cryptographic purposes.

The bit positions that affect the next state are called the taps. The maximum period of the cycle over which the sequence repeats is 2^n-1 for a n- bit wide LFSR register. The tap values must be selected carefully for the sequence to attain maximum period. I have used the following document from Xilinx for getting the tap values:

Xilinx document on LFSR

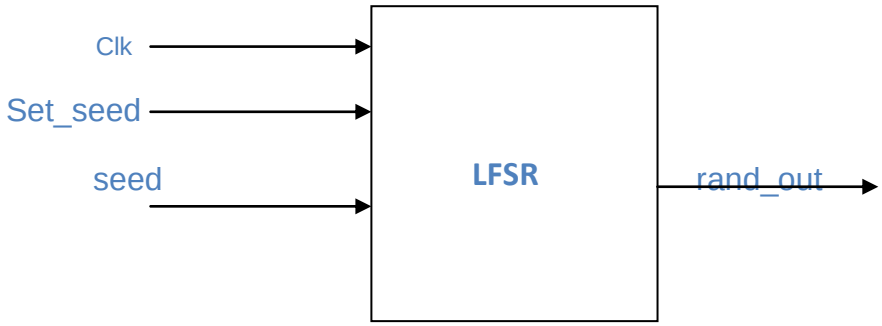
Currently the design can handle register size from 3 bits to 168 bits. If you are able to provide the irreducible polynomial function for higher register sizes then I am ready to include them in the design.

A simple LFSR will look this.It consists of a series of cascaded D flipflops with a XOR gate used for feedback.



The block diagram of the module is given below:

This is the version 2 of the IP.I have removed some unnecssary ports from the port list.Also asynchronous setting of the seed value was changed into synchronous.This is done in order to make sure that there is no bug at higher frequencies.



clk is the input clock to the module. At each positive edge of clock, we will get a random number at the output signal rand_out which is n bit wide.

Set_seed is a std_logic signal which is used for initialization. The seed is set using this signal. When set_seed is '1' the current register value is set to the input value available at 'seed'. Note that this is done synchronously.

Features:

· Standard VHDL, no instantiated blocks.
· Wide option for register size from, 3 to 168 bits.
· Can be operated at frequencies over 1 GHz.
· Seed can be set synchronously.