LFSR Counter Generator

By Evgeni Stavinov, OutputLogic.com

OutputLogic.com
tools that improve productivity

Description

LFSR Counter Generator is a command-line application that generates Verilog or VHDL code for an LFSR counter of any value up to 63 bit wide. The code is written in C and is cross-platform compatible

Parameters

language: verilog or vhdl

count : counter value in hex or decimal format, e.g. 1234, 0x1234.
Can be up to 63-bit long, e,g, 0x7fffffffffffffff, although it’d take a very long time to generate such a counter.

Output Examples

```
[1] C:\OutputLogic\lfsr-counter-generator> lfsr-counter-generator
  usage:
        lfsr-counter-generator language count
  parameters:
        language: verilog or vhdl
        count   : counter value in hex or decimal format, e.g. 1234, 0x1234
```

```
[2] C:\OutputLogic> lfsr-counter-generator verilog 0x1234

count = 0x1234 num_bits=13
generating...

//-----------------------------------------------------------------------------
// Copyright (C) 2009 OutputLogic.com
// This source file may be used and distributed without restriction
// provided that this copyright statement is not removed from the file
// and that any derivative work contains the original copyright notice
// and the associated disclaimer.   --
// THIS SOURCE FILE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS
// OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED
// WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.
//-----------------------------------------------------------------------------
module lfsr_counter(
        input clk,
        input reset,
        input ce,
        output reg lfsr_done);
```

```
reg [12:0] lfsr;
wire d0,lfsr_equal;

xnor(d0,lfsr[12],lfsr[3],lfsr[2],lfsr[0]);
assign lfsr_equal = (lfsr == 13'h220);

always @(posedge clk,posedge reset) begin
   if(reset) begin
      lfsr <= 0;
      lfsr_done <= 0;
   end
   else begin
      if(ce)
         lfsr <= lfsr_equal ? 13'h0 : {lfsr[11:0],d0};
      lfsr_done <= lfsr_equal;
   end
end
endmodule
```

Background

Most of the EE or CS graduates know or at least have heard about different types of hardware counters: binary, prescaled, linear feedback shift register (LFSR), and others.

The majority of logic designers use the first two types, because they're simple to implement in Verilog or VHDL. However, for some applications LFSR counters offer a significant advantage in terms of logic utilization and maximum frequency.

There is an online LFSR Counter Generator tool is running on the OutputLogic.com server. The time it takes to generate the code depends exponentially on the counter size. It takes several seconds to generate a small 24-bit counter.

A stand-alone application can generate much larger LFSR counters orders of magnitude faster than the online tool. The application limits LFSR counter size to 63 bit, but it should cover any practical usage. There is no fundamental problem to extend that. The LFSR counter can be as large as 168 bit, this is the limitation of the LFSR polynomial table in [1].

Here is an example of how the LFSR Counter Generator works:

(1) Specify counter value, e.g. 200. It is 8 bits, so the application selects 8-bit LFSR with polynomial coefficients taken from the table in [1].

(2) Reset LFSR to 0, run a loop that shifts the LFSR 200 times. Then latch its value (LFSR_COUNT_VAL).

(3) Use that 8-bit LFSR and LFSR_COUNT_VAL to generate a Verilog or VHDL code. When the LFSR hits LFSR_COUNT_VAL, it counted 200.

This approach is working because the polynomial selected in (1) has a maximum-length property. That is it generates a sequence of unique values from 0 to 2^n-1.

Following is a table that illustrates size differences between a 32-bit LFSR counter and a regular counter synthesized for Xilinx V5 chip.

Module	Slices	Regs	LUTs
Regular counter	17	32	44
LFSR counter	10	32	7

About the Author

Evgeni Stavinov is the creator and main developer of OutputLogic.com. Evgeni has more than 10 years of diverse design experience in the areas of FPGA logic design, embedded software and communication protocols. He holds MSEE from University of Southern California and BSEE from Technion – Israel Institute of Technology. For more information contact evgeni@outputlogic.com

About OutputLogic.com

OutputLogic.com is a web portal that offers online tools for FPGA and ASIC designers.

References

[1] Peter Alfke, *Efficient Shift Registers, LFSR Counters, and Long Pseudo-Random Sequence Generators*, Xilinx application note ***Xapp052***

[2] Maria George and Peter Alfke, *Linear Feedback Shift Registers in Virtex Devices*, Xilinx application note ***Xapp210***

[3] Xilinx Linear Feedback Shift Register (LFSR) ***Logic Core***

The MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.