ELEX 3525: Data Communications
2022 Fall Term

Self-Synchronizing Descrambler

Introduction

While AWG's and digital 'scopes are useful for generating and capturing waveforms, they cannot operate in real time or sustain high throughputs. Programmable logic devices such as CPLDs and FPGA's are flexible and inexpensive and can be used to implement specialised communication test equipment to generate data or make specialized measurements. A wide range of communication protocols and signal processing IP blocks are available from FPGA manufacturers and others.

In this lab you will practice this approach by designing a self-synchronising (multiplicative) descrambler using the Intel (formerly Altera) Quartus Prime FPGA design software and test it by simulating with a test waveform supplied by the instructor.

The schematic below shows a multiplicative *scrambler* used to generate the test waveform. As described in the lecture notes, the *descrambler* uses the same components but they are wired differently.

This scrambler computes the exclusive-OR (xor) of the current input and bits 5 and 9 of a shift register where the bits are numbered so that the oldest bit is bit 9 and the most recently output bit is bit 0. Note that this numbering may not match the numbering in the diagram in the lecture notes.

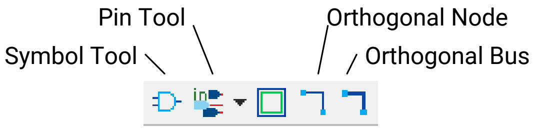
The extra D flip-flop on the output ensures that the output changes synchronously with the clock. You should 'register' your de-scrambler's output in the same way.

Procedure

Create a new Quartus project by clicking on the "New Project Wizard" button . Use all the project defaults except the project folder and name – use appropriate values (e.g. `H:ELEX3525\lab5` and `lab5`). Create a block diagram file (BDF) using the block diagram editor (File > New... > Block Diagram/Schematic File). Add the components and I/O pins required as described below.

You can implement the shift register using either discrete D flip-flops or define a shift register using an LPM component as shown below (select Tools > IP Catalog and under Library / Basic Functions / Miscellaneous double-click **LPM_SHIFTREG**). You can use either VHDL or Verilog file types. Use any name for the variation file name (e.g. `sr`).

Select the required number of bits (see description above), right-shift (MS bit towards the LS bit), parallel data outputs, serial data input and, on the next page, synchronous clear. On the last page select the option to generate a `.bsf` file in addition to the Verilog or VHDL files. Answer Yes when asked if you wan to add the component to the Project library.



Use the Symbol tool to insert an instance of the shift register component from the Project library. Use the Pin Tool to add three input and one output pins. You must use the pin names `clock`, `reset`, `datain` and `dataout` to match the names used in the supplied test waveform. The '`datain`' and '`dataout`' signals are the input and output data bits (active high, H=1). The shift register is clocked (shifted) on the rising edge of 'clock'. The `reset` signal is used to clear the shift register at the start of the simulation.

Use the "Orthogonal Node" and "Orthogonal Bus" tools to connect the inputs to the shift register and to add a bus to the output of the shift register (see schematic below). You can add xor gates and D flip-flops using the Symbol tool. These components are in ...libraries > primitives > logic > xor and ... > storage > dff.

Label nodes to connect them. You can assign a name to a node or bus by right-clicking and selecting "Properties". This node or bus will then be connected to all others with the same name. You can label buses with the range of signals to be included (e.g. `sr[9..0]`). The schematic below shows some examples.

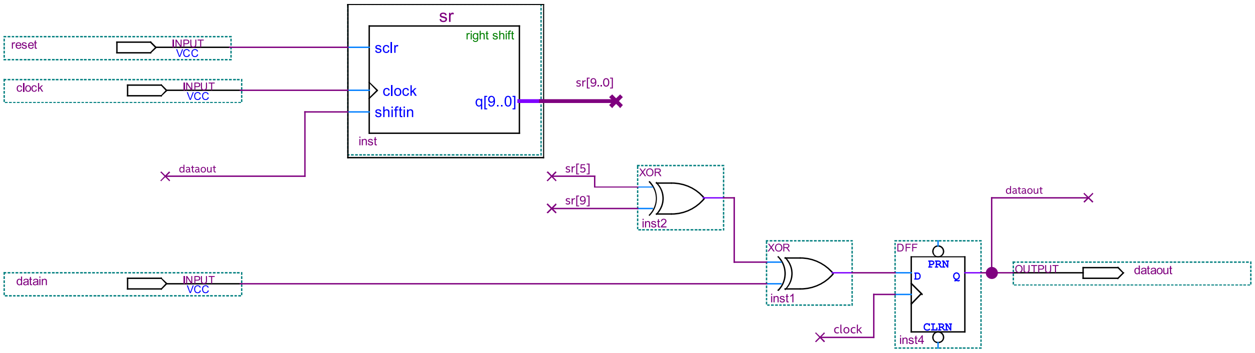


Figure 1: Schematic of the *scrambler*. Do not build this. You need to build the corresponding *descrambler*.

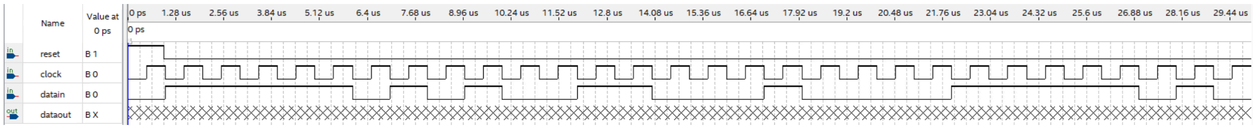


Figure 2: Sample input waveform.

Use File > Save All and save the schematic as **lab5.bdf** (or use your project name). Compile the design (Processing > Start compilation). If there are any errors, fix them and recompile the design.

Download the **lab5testinput.zip** file on the course web site to your project folder, extract the `.vwf` file and open it with Quartus (File > Open, select All Files...) (see the example screen capture below). Select Simulation > Run Functional Simulation)

If you get an error message saying that Modelsim is not available, use Tools > EDA Tool Options > Modelsim and enter the *folder* where Modelsim is installed (often `C:\intelFPGA\20.1\modelsim_ase\win32aloem` but you'll need to confirm this for your installation).

If you get an error about an invalid **novopt** option, use Simulation > Simulation Settings and remove the **-novopt** setting from the **vsim** command options in the Modelsim Script text box. The waveform output will show a distinctive pattern. You can check with the instructor to see if you obtained the correct output. Use a screen capture tool to capture the waveform for your report.

Report

Submit a (PDF format) report containing the identification information asked for in previous labs, a schematic (block diagram) of your working circuit, and the waveforms showing the test input and the output of your circuit.