Self-Synchronizing Descrambler

Introduction

In this lab you will design a self-synchronizing (multiplicative) descrambler using the Altera Quartus II FPGA design software and test it by simulating with test waveforms supplied by the instructor.

The schematic below shows the scrambler used to generate the test waveform. This scrambler computes the exclusive-OR (xor) of the current input bit 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 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 II project using all the project defaults except the project name. Create a block diagram file (BDF) using the block diagram editor. Add the components and I/O pins required.

You can implement the shift register using either discrete D flip-flops or define a shift register using an LPM component as shown below (`LPM_SHIFTREG` under Basic Functions -> Miscellaneous). Note that for this component the left- and right- shift directions refer to bits written as binary numbers so that the MS bit is leftmost and a left shift shifts bits from LS towards the MS bit.

It is often easier to label nodes rather than connecting them up with wires. 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. `ssr[9..0]`). The schematic below shows some examples.

Your descrambler should have two input pins and one output. Use the pin names `clock`, `datain` and `dataout` to match the names used in the supplied test waveform. The 'datain' and 'dataout' are the input and output data bits (active high, H=1). The shift register is clocked (shifted) on the rising edge of 'clock'.

Save the project and design files and compile the design. If there are any errors, fix them and recompile the design.

Open the appropriate `.VWF` file supplied on the course web site (see the example screen capture below) and click on the "run functional simulation" icon. When the simulation completes, show the simulation output to the instructor who will look at the de-scrambled waveform and tell you if it is right or not.

Pre-Lab

Submit a schematic of your circuit. You can either draw it by hand and scan it or create it with Quartus II and print the schematic to a file. The Quartus II software is free and you can download and install it on your own computer if you wish.

Report

Create a 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.

clock

scrambled

OUTPUT

DFF
PRN Q
D
CLRN
inst7

XOR
inst3

XOR
inst2

ssr[9]

ssr[5]

clock

ssr[9..0]

shiftregister
left shift
clock
shiftin
q[9..0]
inst

clock

INPUT
VCC
clock

unscrambled
INPUT
VCC

Name	Value at 0 ps	0 ps	2.56 us	5.12 us	7.68 us	10.24 us	12.8 us	15.36 us	17.92 us	20.48 us	23.04 us	25.6 us	28.16 us
clock	B 0												
datain	B 0												
dataout	B X												

2