.

Generic FPGA Design for Adder, Multiplexer and Universal Shift Register

Sagalpreet Singh
2019CSB1113

Abstract

The project aims at designing and simulating an FPGA board which could be used as Multiplexer, Adder or Universal Shift Register depending upon the the configuration file that is loaded into it. We try to use same type of logic tile i.e one with similar configuration again and again. The simulation is done in Verilog.

1 Introduction

A field-programmable gate array (FPGA) is an integrated circuit designed to be configured by a customer or a designer after manufacturing. FPGA's are very flexible in nature and can be used to make generic circuits. We use logic tiles and switch boards to design the FPGA for this project.

1.1 Design of Logic Tile

Since we need to implement both combinational as well as sequential components as a part of the design, our Logic Tile should support both. Thus we have a Multiplexer to decide whether to connect output of logic tile to output of lookup table or give it through the flip flop. The control bit of multiplexer would be given in the configuration file to decide whether it is to be used as a sequential component or combinational component. For the purpose of designing our FPGA, we use 5 input logic tile. 5 input logic tile can be used to implement any boolean function of 5 variables. Every logic tile has its own configuration which defines its lookup table. 5 input boolean function could have 32 different inputs and corresponding outputs which need to stored in a lookup table.

1.2 Design of Switchbox

While designing generic circuits using Logic Tiles, switch boards are an important component as they decide which wire to give to the output based on the configuration file. For designing this FPGA, we use a 4 X 4 switchbox. Each output of the switchbox can be or of any combination of the input wires to it. Thus the configuration for swtichbox would be 16 bits in size.

2 Design Implementation

The overall design for FPGA consists braodly of three subcomponents: Adder, Multiplier and Universal Shift Register, the output is guided out of the FPGA through a switchbox according to the configuration that is loaded into the FPGA. The components are implemented using logic tiles. We try to minimize the number of logic tile that are required.
Although it is possible to use lesser number of logic tiles than we use for our design, but then we will require more distinct configurations for logic tiles, hence there is a trade-off between the two.
For our implementation, we require 39 logic tiles in total. Only 6 different configurations are required for these. By using 11 different configurations, we reduce number of logic tiles to 36 and speed up the multiplexer.
4 switchboxes are also required.

The 6 different types of logic tiles required are as follows:
1) **MUX** : This logic tile functions as a 2 X 1 multiplexer. The configuration for this logic tile is made accordingly. Whenever used, only 3 inputs are required, rest 2 are grounded.
2) **ADD0** : This logic tile takes in two 2-bit numbers and outputs the bit-0.
3) **ADD1** : This logic tile takes in two 2-bit number and outputs the bit-1.
4) **ADDC** : This logic tile takes in two 2-bit numbers and output the carry out.
5) **REGC1**: This logic tile takes in one hot encoded value and outputs corresponding 2-bit number's bit-0.
6) **REGC2**: This logic tile takes in one hot encoded value and outputs corresponding 2-bit number's bit-1.

Further we discuss the entire design of FPGA using these 6 types of logic tiles.

2.1 8 X 1 Multiplexer

Having already designed 2 X 1 multiplexer in logic tile MUX, we can hierarchically design the 8 X 1 MUX. We will require 7 MUX logic tiles for this implementation.

2.2 4 bit Full Adder

We break down the addition operation over 4 bits to be done in two steps, we use logic tiles ADD0 and ADD1 to compute bit-0 and bit-1 of the result. The result of ADDC module is given to another ADD0 and ADD1 as cin along with bit-3 and bit-4 of input numbers to compute bit-3 and bit-4 of output. ADDC computes bit-5 of the output. Thus in total we use 6 logic tiles (2 each of ADD1, ADD2 and ADDC).

2.3 Universal Shift Register

To design 8 bit universal shift register, we first convert the input, which is either shift right, left, retain or load to one hot encoding, simply as we are assured that no two of those will actually be one at same time. This is fed as input to logic tile REGC1 and REGC2 to obtain its equivalent two bit control signal to be given as input to multiplexer. Each bit of the 8 bit shift register needs to be stored in a flip flop, hence we require a logic tile corresponding to each bit. Since, we have 4 possibilities for next state, we require a 4 X 1 multiplexer whose control pins are to be connected to REGC1 and REGC2's output. To implement 4 X 1 multiplier, we will require three 2 X 1 multiplexers (hierarchical design). Thus, in total we will require 24 (8 X 3) MUX logic tiles but with sequential bit set to 1 for 8 of those and 0 for rest 16. In total, we thus use 26 logic tiles for implementing universal shift register.

2.4 Switchbox

Now, the output of the FPGA needs to be designed. Keeping in mind that if we load FPGA with different configurations, we will require different number of outputs, we use switchbox and always give the output through least significant pins possible. Switchbox takes input from all the three different circuits implemented inside FPGA and directs the required one according to the configuration loaded into switchbox.