Computer Organisation and Architecture Laboratory

CS39001
Assignment-5 Report

Group Number: 23

Ashwani Kumar Kamal (20CS10011)
Astitva (20CS30007)

September 14, 2022

1 Problem Statement

Binary Counter: is a sequential circuit. An up-counter will count up, say from 0. Consider a 4-bit binary counter, which when reset is initialized to 0000, and then at the positive edge of the clock starts counting in the sequence: (0000),(0001),(0010),(0011), . . . ,(1111),(0000). Let us consider two ways of designing the above counter.

1.1 Problem 1

Behavioral Design: Write a verilog code to design the up-counter. You can use verilog constructs to define the logic of the counter. The counter should have an asynchronous reset, so that the counter can be reset at any time to zero. The result of the counter should be displayed on the LEDs in your board.

1.2 Problem 2

Structural Design: Design the architecture of the up-counter using a 4-bit adder. Note that you can design an add by one adder by optimizing the design of the RCA/CLA you have done in the last assignment. Work out the details of the optimized add-by-one block, and write the verilog codes. Also you need to design a D-FlipFlop as mentioned in the lab. Draw the architecture for the up-counter, with the above blocks as sub-modules. Write the verilog code, and prototype on the FPGA board.

2 Solution

Directory structure-

- Submitted file - **Assgn_5_Grp_23.zip**
- Codes are written in a separate directory (verilog source fileswere added as source in ISE software (**Add as source** option)).
- It contains 3 items-
  - **Assgn_5_Grp_23** directory- main ISE project file, open project using **Assgn_5_Grp_23.xise**
  - **src** directory contains the raw verilog codes for implmented counter and test benches
  - **Report_Grp_23.pdf** which is the report file submission
  - Corresponding **.ucf** files are in ISE project directory

2.1 Solution 1

- The behavioural implementation for binary counter has been written in the file **binary_cnt.v**
- The clock divider has been implemented and written in **clk_divider.v**
- Final wrapped up top module for simulating in the FPGA is in **binary_cnt_clk.v**
- Corresponding implementation constraints file is **binary_cnt_clk.ucf** in the project directory (**Assgn_5_Grp_23**)

2.2 Solution 2

The structural implementation for binary counter has been written in the file **binary_cnt_struct.v**

- The structural implementation for binary counter has been written in the file **binary_struct_cnt.v**
- The clock divider has been implemented and written in **clk_divider.v**
- Final wrapped up top module for simulating in the FPGA is in **binary_cnt_struct_clk.v**
- Corresponding implementation constraints file is **binary_cnt_struct_clk.ucf** in the project directory (**Assgn_5_Grp_23**)

Architecture Diagrams-

For **add_by_one_carry.v** let the input be **in1** (4 bits), input carry be **cin**, output sum be **out** (4 bits), corresponding output carry bit be **cout** then-

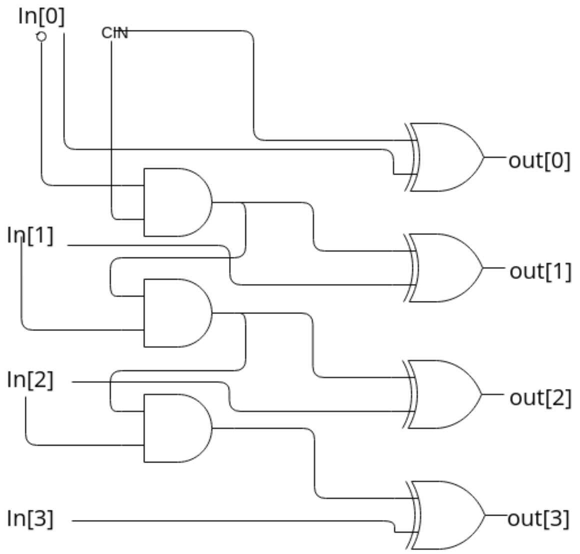

Figure 1: sum output

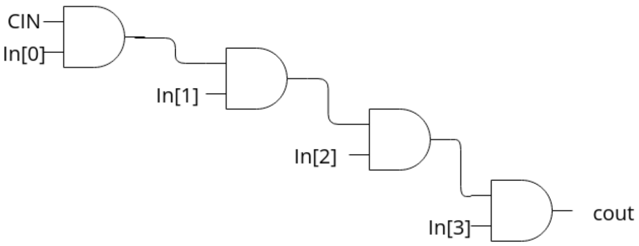

Figure 2: carry output

The above diagram has been reduced from carry look ahead adder. For a fixed **in2** = 0000 and fixed **cin** = 1, they are reduced to this format.

For **binary_struct_cnt.v** we will have the following diagram (all dffs have input wires to **clk** (clock) and **rst** (reset bit))-

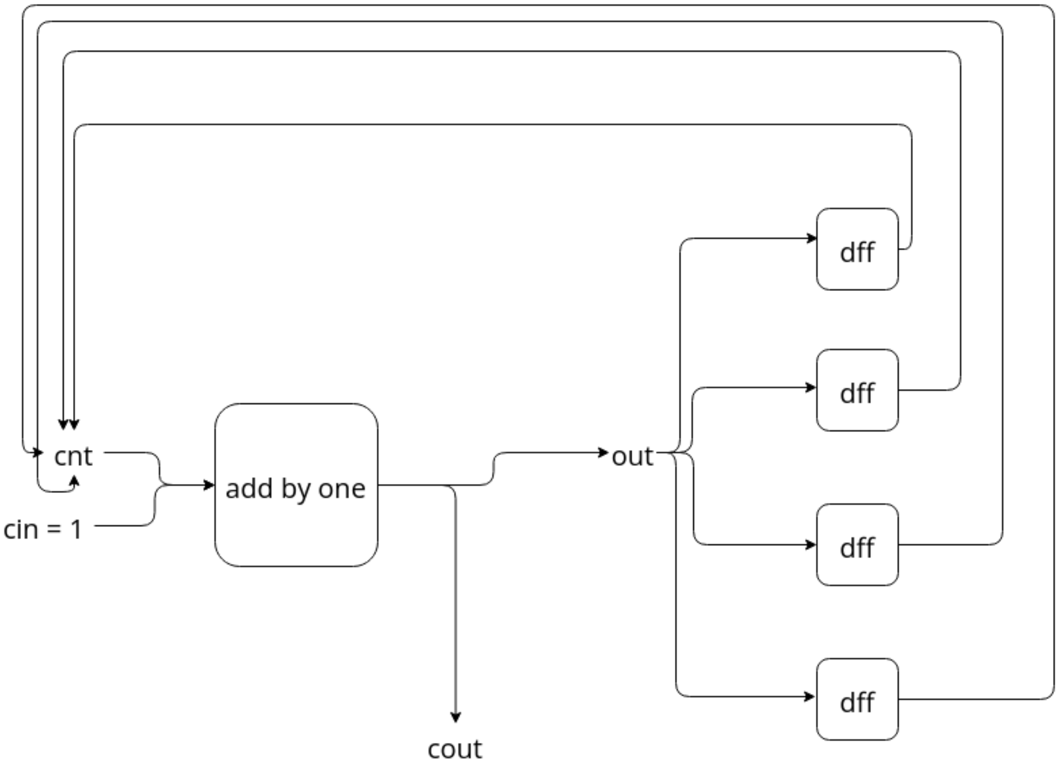

Figure 3: binary structural counter