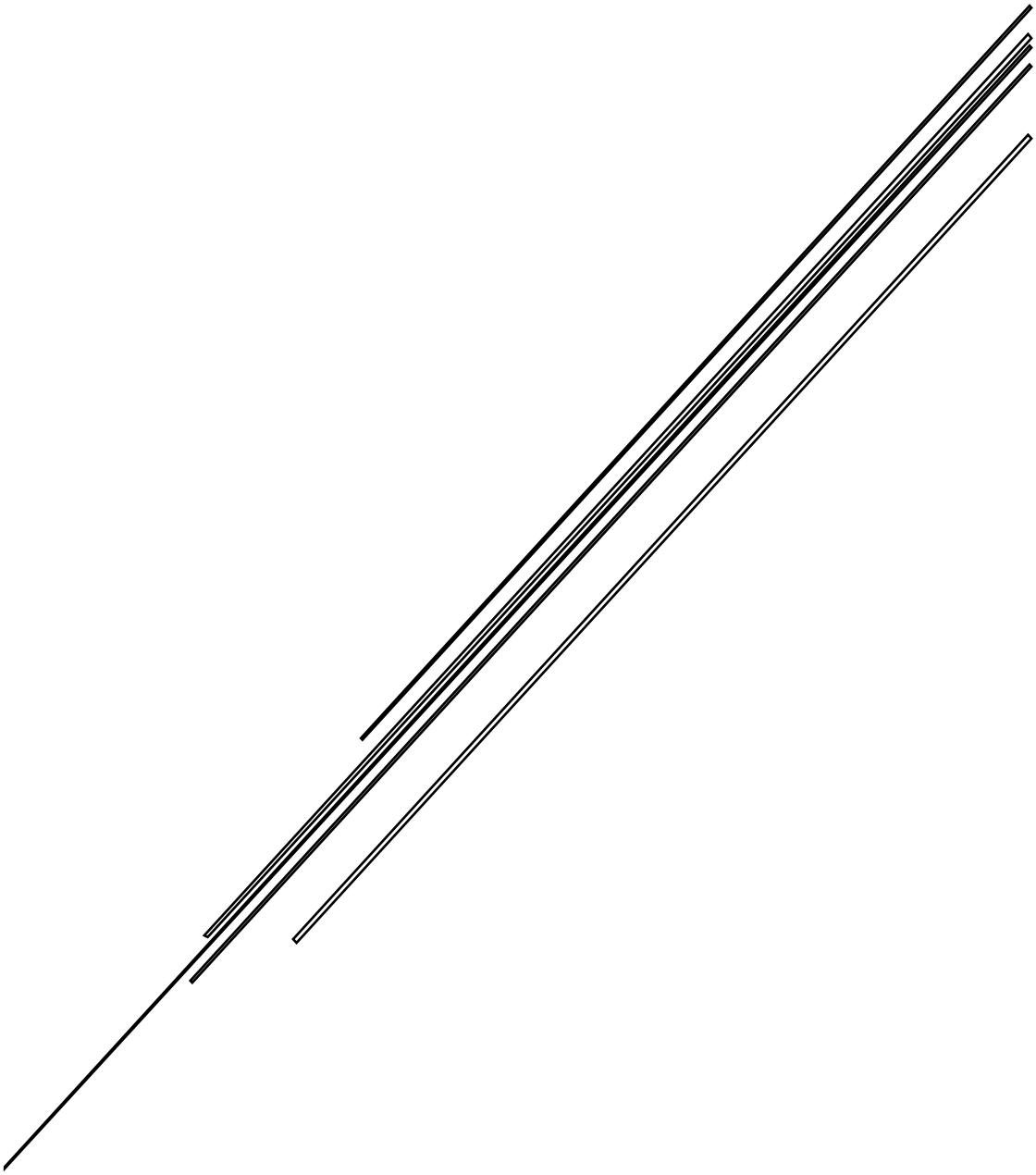
FARAH KABESH – 900191706
KARIM SHERIF – 900191474
MOHAMED SOLIMAN - 900182638

PROJECT 2 REPORT

DIGITAL DESIGN I

FALL 2021

TUESDAY 30^{H} NOVEMBER 2021



Project Summary

The aim of this project is to create a simple calculator that can be implemented on a Basys 3 FPGA Board using Verilog HDL. The calculator should generate and display two numbers between 0 and 99 as well as carry out the basic arithmetic operations; addition, subtraction, multiplication and division.

The calculator is designed to have a total of 9 buttons, where each button can be pressed by the user to perform a certain function.

- B1: Controls the tens of the first number.
- B2: Controls the units of the first number.
- B3: Controls the tens of the second number.
- B4: Controls the units of the second number.
- B5: Displays result of adding the two numbers.
- B6: Displays the result of subtracting the second number from the first number.
- B7: Displays the result of multiplying the two numbers.
- B8: Displays the result of dividing (first number over the second number), rounded to the nearest integer.
- B9: Displays the original numbers that were inputted initially.

The software used for this project is Xilinx Vivado Software.

Program Design

This project consists of 5 design sources (modules); clock divider, 2-bit binary counter, 2x4 binary decoder, BCD-to-seven-segment decoder (ssdproced) and SC (simple calculator).

a. **Clock Divider**

The clock divider takes the input of the clock signal with a certain frequency (f-in) and outputs a clock signal with a lower frequency (f-out). The output frequency is calculated as $f(out) = \frac{f(in)}{2n}$. The value of n represents the number of counts of the input clock's cycle before the output clock flips from 0 to 1 or from 1 to 0.

- Input signals: clock, reset.
- Output signals: clock output.

b. **Binary Counter**

The binary counter is a simple counter that has an initial value of 0 at the time of reset. With each clock cycle (positive edge-triggered clock signal), the value of the counter is incremented by 1. Since a 2-bit binary counter is used for this project, the counter counts from 0 to 3, before falling back to 0.

- Input signals: clock, reset.
- Output signals: count.

c. **Binary Decoder**

The binary decoder converts binary information from n input lines to 2^noutput lines. An enable input is used to control the outputs. If the enable is equal to 1, only one of the outputs is asserted according to the valuation of the inputs. Otherwise, none of the outputs are asserted. The decoder used in this project is a 2x4 binary decoder, therefore 2 inputs are decoded into 4 outputs.

- Input signals: a (2-bit input).
- Output signals: out (4-bit output).

d. **BCD-to-Seven-Segment Decoder (ssdproced)**

The BCD-to-seven-segment decoder is used to convert the input into a suitable form of output for display. It accepts a decimal digit as input and generates the appropriate output segments to display the input decimal digit. The calculator will need to display digits between 0 and 9 as well as a negative sign (-) in the case of subtracting a larger number from a smaller number, therefore 11 cases are needed in the module.

- Input signals: x.
- Output signals: out.

e. **SC (Simple Calculator)**

This is the top module which represents the functions of each button on the calculator; such as setting the digits for each number, performing arithmetic operations (+ - × ÷), and handling the display. In this module, B1, B2, B3 and B4 represent digit 4, digit 3, digit 2 and digit 1 respectively.

Module instantiation is used to declare an instance of the four modules listed above inside the SC module. Wires are also used to connect the different elements in the module as well as the input and output ports for the modules instantiated.

- Input signals: B1, B2, B3, B4, B5, B6, B7, B8, B9, clock, reset.
- Output signals: digit1, digit2, digit3, digit4, decimalpoint, result, nandos (anodes).

Testbench

This project also includes a simulation source (testbench), where the top module is instantiated and called DUT (Design Under Test). Different inputs are applied to the DUT in order to monitor the outputs and display them. A testbench is used in this project to simulate the required test cases.

Constraint File

A constraint file is used to link (constrain) the input and output signals in the module to certain pins on the FPGA, in order to run the project physically on the FPGA.

Simulation Result (Required Test Cases)