Digital Design Project II report

Merna Wael Abdelbadie - 900203731

Ikrame Rekabi - 900204492

Program Design:

Verilog files:

1. operationsFileOnly: This is the module that does the addition, subtraction, multiplication, and division, and checks the decimal point. It simply checks which switch is on, then does the corresponding operation by first combining the individual digits to form 2 numbers. For example, if in0 = 1, in1 = 2 then n2 = 21, and if in2 = 3, in3 = 4 then n1 = 43. Then it does the operation on n1 and n2. After that, it separates the output into 4 digits out0, out1, out2, and out3 by taking % 10 of the output and then dividing it by 10 and so on filling the 4 digits. For subtraction, the negative sign is handled by making out2 correspond to decimal 10 so that it would be translated as a negative sign in the BCD. For division, dividing by zero will give ERR (stands for error), and the remainder is used to approximate the result to the nearest integer. It also has a variable called dc_output (stands for decimal point) that is 0 if one of the operations is being done, or 1 if B9 is pressed or none of the operations is yet done. Finally, in the case when B9 is switched on, the original inputs will be shown again.

2. Incrementor: This is the module that does the incrementation every time a button is pressed. We have used 4 incrementors, each one increments the 1 digit of the 2 numbers (the units and tens of both numbers).

   This module takes the clock divider output and the reset and the button that we will use to increment the number and its output is the number incremented. Every time the button input of the incrementor is pressed, the number gets incremented by 1 until it reaches 9 and starts again. For the code to be compatible with the buttons in the FPGA, it has a synchronizer, debouncer, and edge detector included as registers and wires. The code has an always block to synchronize the button input and at the end assigns the edge detector to its correct value using the synchronizer. Then, there is another always block used to increment the counter by 1 every time the button input is pressed until it reaches 9 and starts over from 0.

3. BCD: This is the module that takes in the digits after the operations have been done on them and decode them to the seven-segment display so that they can be shown on the FPGA. It has the cases of the input from 0 to 9 and specifies which

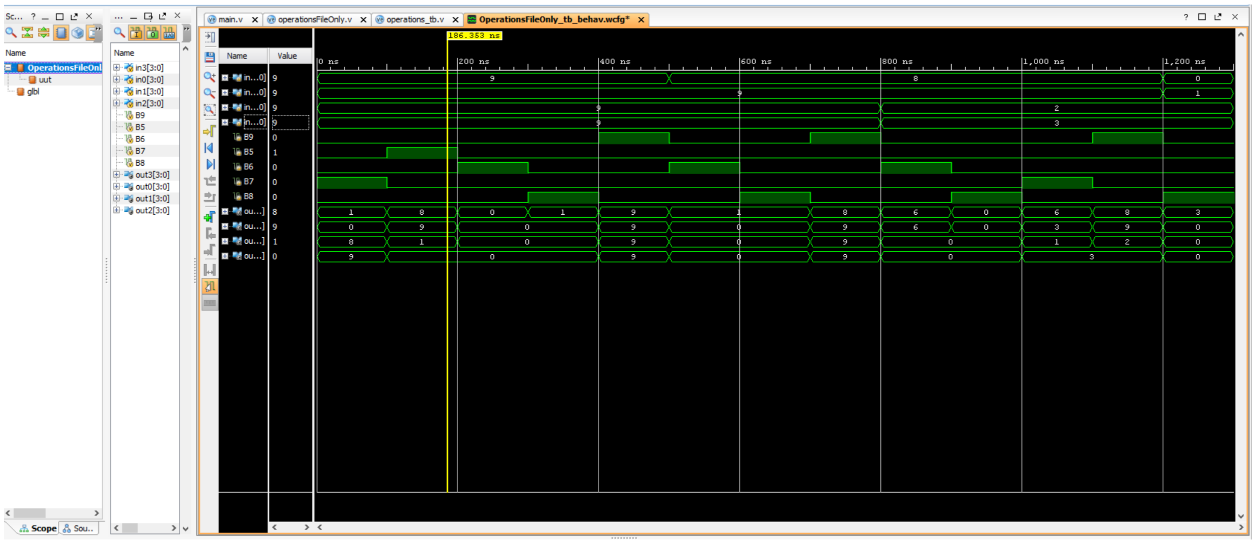
segment will be on for each digit from 0 to 9. The segments are active low so if the segment is 0 then it is on if 1 then it is off. It also has the seven-segment display of the negative sign and it is represented as 10 in the cases. Also, it has 11, 12, and 13 cases to display the err when the divider is 0.

4. Fast_clock: this is a module that takes the clock in the FPGA which is 100 MHz and makes it 5kHz as default if no parameter was sent to it.

5. main: the main is a module that connects all the modules together. It has inputs clk1, rst, B1, B2, B3, B4, B5, B6, B7, B8, and B9, and outputs anodes, enables, and dc(for the decimal point). It also has several wires that help in connecting all the modules together. First, the Fast_clock module is called to make the clk 5kHz instead of 100MHz. Then this clock (main_clk_output) is used in the calling of the incrementer module. The incrementer is called once for each digit displayed, every one of the 4 buttons is connected to an incrementor and is used to adjust the digits of the first and second number. Then the output of the incrementer is sent to the operationsFileOnly so that the operations (+, -,*, /) can be done on these incremented digits. The output of the operationsFileOnly is then sent to the BCD so that it can be displayed on the FPGA. Finally, there is a register called counter that gets incremented at every positive edge of the main_clk_output so that it can be used in giving the anodes their value (if count is 0, then the anodes will be given the value of the output of the first digit from the BCD and so on). Also in each case of the if statement, the enables are given their values. Enables are initialized with the following logic: when anodes are given the value of the first digit from the BCD, it should appear in the first seven-segment display on the FPGA, so the enables will be as follows: 4’b1110 as it is active low, and so one for the rest of the cases. The decimal point is put only in the case where the enables are 4’b1011 because we want it to be displayed only once between the two numbers.

6. Constraint: the constraint file connects the buttons B1 to B4 and switches B5 to B9. It also connects clk1 to the clock in the FPGA, and connects the rst to one of the switches. Moreover, it connects the anodes, and the decimal point (dc) to the seven-segment display. Finally connects the enables to the 4 pins of the enables in the FPGA.

Simulation results:



FPGA results:

https://drive.google.com/file/d/1lW1qgtlYGbS2qyHRDIGoITRvAfTu79F7/view?usp=sharing

CONCLUSION

All in all, the project is working, yet there are two codes: one that works with the testbenches and one that works on the FPGA.