Maya Hussein
and Nada Badawi

CSCE - 2301

May 2022

FPGA - Calculator

1 Program Design

The program includes three primary modules: all which corresponds to the main or the top layer of the program, Num_counter, operations, the bench where all the calculator operations are defined. The code is also designed to handle major errors, such as producing undefined values. Each module is defined and utilized as follows:

1.1 All.v

The "all" module involves many of the dependencies responsible for the calculator. This includes the clock divider, which is used to decrement the frequency of the clock supplied from the FPGA Basys-3 board. The expected output frequency is calculated as:

$$f_{out} = \frac{f_{in}}{2n} \tag{1}$$

where $f_{in} = 100\text{MHz}$ and n = 99999999. So, $f_{out} = 0.5$ Hz. This is equivalent to 2 seconds.

For every passing 2 seconds, a new anode is chosen to get displayed and the rest will be turned off. This occurs with a very quick time period that the human eye cannot recognize.

Moreover, the anode control in the module is executed in an always block to control which anodes in the seven segment display to be displayed. The output register "anode" is an array of 4 bits that represent the functioning

4 ports of the anode. For instance, if "anode" = 4'b0001, then all anodes are on except for the rightmost, assuming that it is operating on active-low mode. If statements were used to efficiently make the selection among the four cases of the anodes.

The Binary Coded Decimal (BCD) is another code block used to translate the decimal digits into binary strings to represent the illuminated LED segments for the seven segment display to display the intended digits. The truth table of the BCD seven segment display was consulted to write the block of code.

Overall, the main inputs and outputs of the module are the ones associated with the board itself. These inputs include: the five switches, s1, s2, s3, s4, and s5 are responsible for the four arithmetic operations and the reset operation. The clock and reset are also given as inputs to set the clock and the reset of the counter. An array of four bits, "btn" is used to represent to the four buttons of the counter. Only the "anode" array, the decimal point representation "dp", and the Displayed_LED array, which describes the illuminated segments on the board are the outputs of the module.

1.2 Operations.v

This module primarily consists of all the calculator functionalities, including but not limited to: addition, subtraction, multiplication, division, originating back to the user-defined values.

The module begins by declaring the following inputs:

- Inputs: s1,s2,s3,s4, s5 are used to detect the mode of operation in which the calculator is currently in.

- Inputs: one, two, three, and four are designed to take to input values representing each anode with one being the left-most position.

After the execution, the module is required to present the following outputs:

- Outputs: op1, op2, op3, op4 are used to store the final calculation from the operations.
- The neg is used to display the negative sign when the second number is greater than the first.
- The "error" output displays "err" whenever the divisor is equal zero, then handling error of when the user divides by an undefined value.
- temp_dp stores the last bit of the segment display and acts as a Boolean that turns off the decimal point whenever an operation occurs, and back on whenever the

The remaining segments were used to service the functions-

- Wires: digit1 digit2 convert the 7 segment representation into decimal format for further modifications.
- The remaining integers, temp, mod, and remainder help in the formatting of the division algorithm.

Algorithm Brief: After wiring digit1 and digit 2, an always block is used to perform the main five functionality. Initially, the compiler checks which switch is on if its any of the basic mathematical operations; the calculator produces the correct value; when the "bring original" switch is on, the compiler prevents any other operation from forming setting it back to its original value.

1.3 Num_counter.v

This module focuses on debouncing the buttons and creating a counter to increment the values of the seven segment display. The basic inputs are the button, the reset, and the clock, resulting to produce a new incremented digit. The first always blocks focuses on syncing the button at every positive edge of the clock and ensuring if a reset is on, the sync is referenced to one. The last always block is used for the counter. If the reset is zero , the digits are reset to zero else if its not equal to nine its increments as its normal functionality. If it reaches nine, the next incremented value will be a zero.

1.4 Video Drive Link

Below is the link of the FPGA demo of project calculator:

`https://drive.google.com/file/d/1mp1uc7uYRCfbgplXMEKnvSualRESYD-e/view?usp=sharing`