FINAL PROJECT

CALCULATOR

DESIGN DOCUMENT

Darshil R Trivedi

50318251

Table of Content

1. INTRODUCTION

As the project title suggest, A Simple Calculator on Basys3 Board is implemented. VHDL language for coding purpose is used in this project.

The Calculator is able to perform Addition, Subtraction and Logical Left shift between 2 Numbers. The operations performed are BCD operations.

First four slider switches, 5 push buttons, 4 LEDs and 7 segment display are used.

The 7-segment Display is used to show the operands and their result.

Slider switches are used to give 1st operand and 2nd operand and are latched using different push buttons. After both the operands are obtained left push button is used for addition, right for subtraction and lower button for logical left shift operation. Upper button is used for clearing the output and change mode of calculation.

The two operand mode described above or the accumulator mode. When in accumulator mode, the user only enters one operand and the other operand is the result of the previous calculation.

2. STRUCTURE OF PROGRAMMING

The VHDL code consist of 3 Sequential process. They are:-

1. Clock Divider Process:- The 1st process is the clock divider process. Here the 100Mhz of on board clock is divided by 2 raise to 20 to get 100Hz. 100Hz means 100ms is the refresh time. Each anode of 7 segment is turned on 1 time in 100ms i.e. 25ms each. To divide any binary number in 2 raise to n times we see n MSB. This is the logic that is used it to divide by 4. Thus 2 MSB of 2 raise to 20 counter are given as select line of mux_in for multiplexing the anode. 2 MSB would change from 00 to 01 to 10 and to 11 in each 25ms. Thus implemented the Clock divider and anode logic.

2. Display Process or cathode logic process:- K-maps is used to Display the output of the operands that are entered and their result. Active low logic is there in cathode for Basys3 board. Whenever the digit entered is greater than 9 it displays 0 in the 7-segment

3. ALU process:- This is the process in which the majority of coding is done. Slider Switches 0 to 3 are used to give 4 digit operand. As Buttons present on Basys3 board are used number of times to do different Activity, A temporary counter named as count to check the nth number of time the button is pressed is used. For Example when button left is pressed for the 1st time values of Slider switches are stored in temporary latch signal X. If this value is greater than 9 it is treated as 0. After operand 1 is entered and centre button is pressed the values of temporary signals are given to 1st operand named digit0 in the code and the value to temporary signal is cleared to clear the confusion and count value is increased by 1. When again the procedure is repeated same things happen but this time 2nd operand i.e. digit1 is stored. Now when count is 10 i.e. 3 and left button is pressed BCD addition is performed. When Right is pressed BCD subtraction is performed and Down button performs Logical left shift. This Math is explained below in detail.

   Accumulation Mode is divided into two process. A one second Timer is created in the clock process which turns on when upper button is pressed. The number of times the Button Up is pressed in one second is counted and 2nd counter i.e. the count2 is used to store it's value. This counting began only after the ALU operations are performed. For this , additional acc signal is used.

3. ALU LOGIC

The ALU Operations: -

1. BCD addition: - As BCD operations are performed in group of 4 bits after getting values to operand 1 and operand 2, number is divided into 4 bits pair and are added with carry of previous stage or input carry for LSBs. For our case input Carry is always 0. The result of this is stored in 5-bit temp variable if it is greater than 9 add 6 (0110) as BCD error correcting. Resize the result into 4 bit and the 5th bit MSB is used as the Carry in for the next stage. In this way the BCD addition is performed. If the final result is 5 digit, only last four digits are displayed.

2. BCD subtraction: - BCD subtraction is performed in almost similar way as addition. First of all normal binary subtraction is carried out. Than result is grouped into group of 4 bits and if the number is greater than 9 error BCD i.e. 6 (0110) is subtracted from it. Another temporary variable is used for this execution.

3. Logical left shift: - 1st operand is converted into unsigned form and 2nd operand into integer. Than the unsigned number is rotated by integer count to do decimal logical left shift. This shifting execution also works in same way as addition i.e. if shifting generates a number greater than 9 error BCD (0110) is added to it and carry is added to the next stage.

4. IMPLEMENTATION PROBLEMS

There were many errors faced throughout the implementation of calculator on Basys3 Board and many things that can be learnt from that. The 1st problem faced was sequential execution of different process. Sequential execution for latching each operand and operation was tried using 4-bit signal Y but turned out that many synthesis and implementation error like multiple assignments was faced. The major thing learnt from that was only all independent codes can be given as separate process.

Second error faced was latching of same button for different operand. Initially separate clock for latching each button was used but it did not work. Though there were no Implementation error the output was always zero. A Single clock was used to latch all the buttons and a counter named count was added to know the Nth number of time particular button is pressed and act accordingly.

Third and the most long lasting error was due extremely fast execution every time centre button was pressed the counter incremented and latched both the operand that is digit0 & digit1 in our case. This was because centre button remained on for time long enough to latch both operand. This was solved by using simple trick i.e. order of writing code was shuffled. 1st the btnC and higher count was given else btnC and lower count. So the 1st time condition is false and by the time execution cycle is completed centre push button is hold off.

For ALU operations first of all binary operations were used that was changed to BCD operations later according to logic mentioned above. For BCD subtraction 9's complement method was tried but did not give the required answer so shifted to logic mentioned by Professor.

To Toggle between Accumulation mode and two operand mode is also one of the most difficult task. Implementation of 1 second timer was done and the number of times upper button pressed in 1 second is recorded. If the count is greater than 1 accumulation mode otherwise two operand mode.

Major Implementation that were tried and failed are kept n the program as comment. Along with this redundant code additional comments are done regarding which signals & variables are used for what purpose.

5. EXTRA FEATURES AND CONCLUSION

I. EXTRA FEATURE

The 16th LED on Basys3 Board is used as overflow bit to indicate the overflow as result of ALU operations. Whenever the result of operation is 5 bit the last 4bits are shown on seven segment display and overflow LED is set to 1 i.e. turned on.

II. CONCLUSION

Hence, A simple Calculator which is able to perform BCD addition, Subtraction and Logical left shift is successfully made. This Calculator works in two modes and perform all the operations as required.