Merve Karakaş – Ahmet Uysal

ELEC204 – M. Cengiz Onbaşlı

2 by 2 USER INPUTTED MATRIX MULTIPLICATION

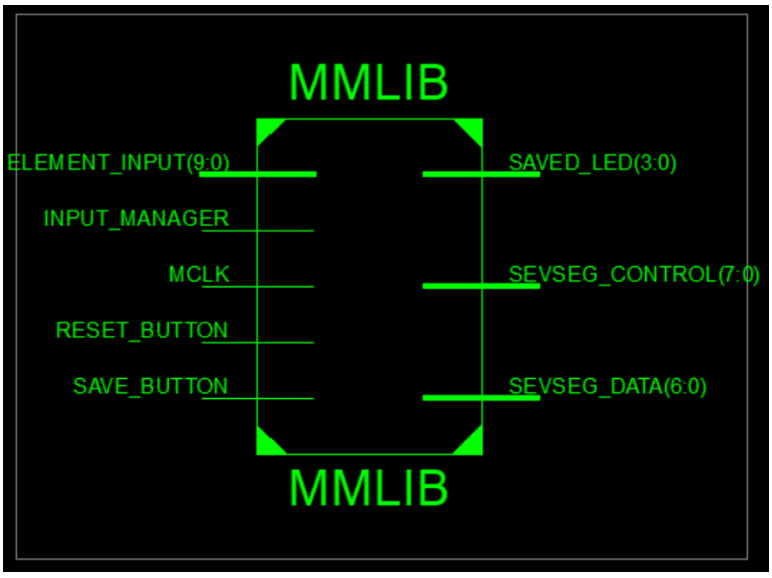
Abstract:

Our project takes two 2 by 2 matrices from the user with the 10 switches on the board (5bit for 1 element), and one button enables to switch between elements, another button makes the assignment operation, and there is a reset button which resets the given values and makes user to be able to create matrices with new values. Also, there are 4 LEDs that shows which elements are assigned.

Project:

In this project, we aimed to take inputs, which are two 2 by 2 matrix ([A1,A2;A3,A4], [B1,B2;B3,B4]), from the user, then show the result of these matrices ([C1,C2;C3,C4]) multiplications on the board. We created a library called MMLIB which contains different blocks of our code. The blocks and their inputs/outputs and their functionalities in the project are as follows:

MMLIB: This is the main block which takes the main signals which are taken from the board's pins, and gives outputs signals to the board. Main input signals are INPUT_MANAGER, MCLK, SAVE_BUTTON, RESET_BUTTON, ELEMENT_INPUT and main output signals are SAVED_LED, SEVSEG_CONTROL, SEVSEG_DATA.



- ELEMENT_INPUT: Input that is taken from the 10 switches on the board which will determine the matrix elements. Due to different buttons (will be explained later), it determines different element sets e.g. A1A2 or B3B4. After all elements of matrices are assigned, 10 switches loses its functionality to determine elements until RESET_BUTTON is clicked; and instead of previous functionality, the rightmost two switches are used for selecting which

element of the result matrix will be showed on seven-segment display. This signal goes to INPUTMANAGER and DRIVER as an input.

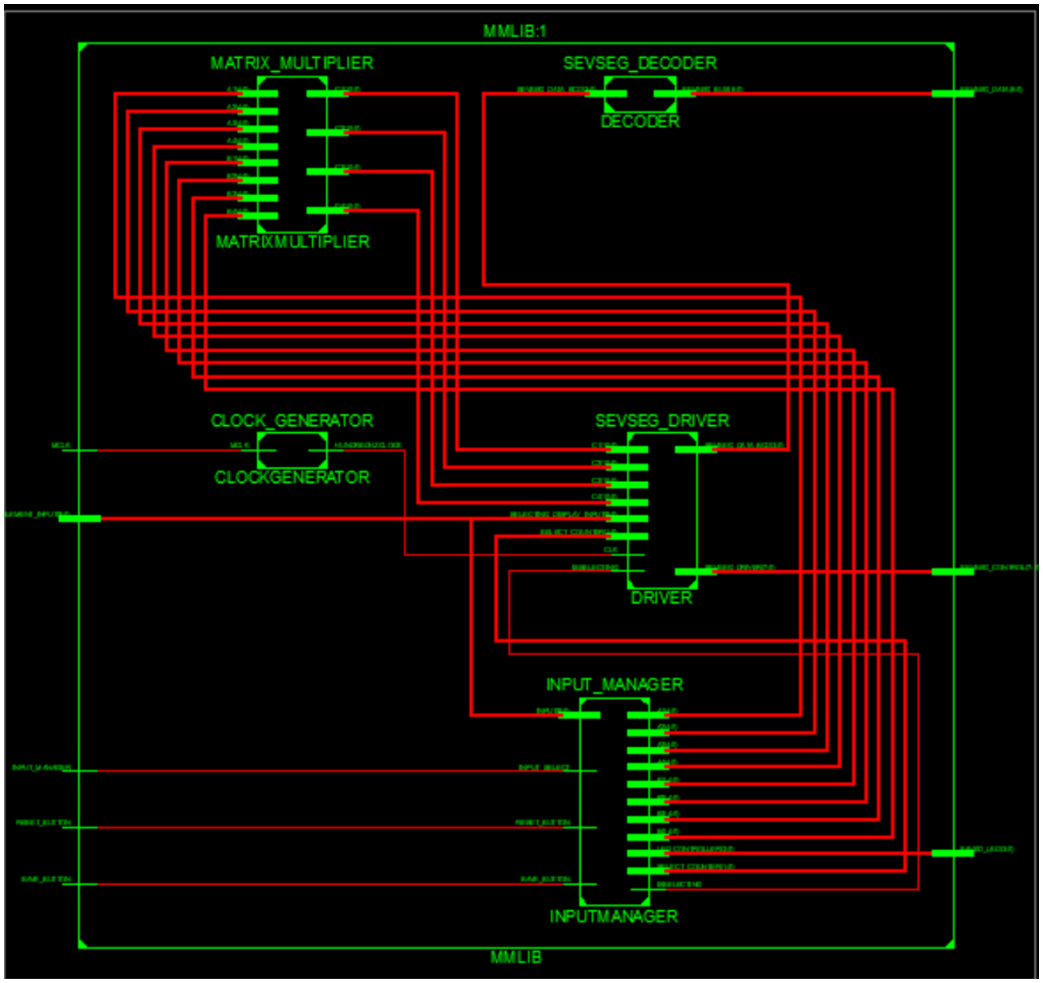
- INPUT_MANAGER: Input that is taken and wired to INPUT_SELECT in the INPUT_MANAGER block. It provides us to change between element inputs.

- SAVE_BUTTON: It is wired to SAVE_BUTTON signal at the INPUT_MANAGER block. When it is pressed, it enables the signal that comes from ELEMENT_INPUT to be assigned to the elements.

-RESET_BUTTON: It is wired to RESET_BUTTON signal at the INPUT_MANAGER block. When pressed, it clears the previous values and we can make another matrix multiplication.

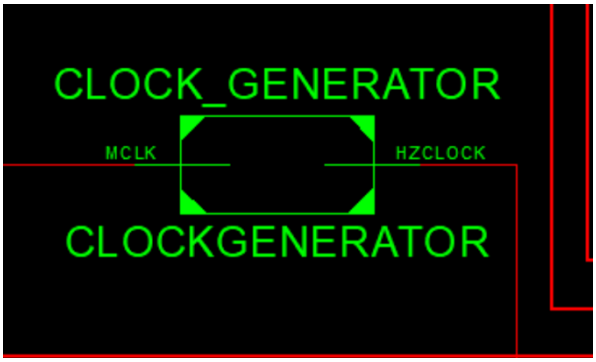
- MCLK: Master clock of the FPGA Board (100 MHz)

MMLIB consists of blocks named CLOCKGENERATOR, MATRIXMULTIPLIER, DECODER, DRIVER, and INPUTMANAGER:



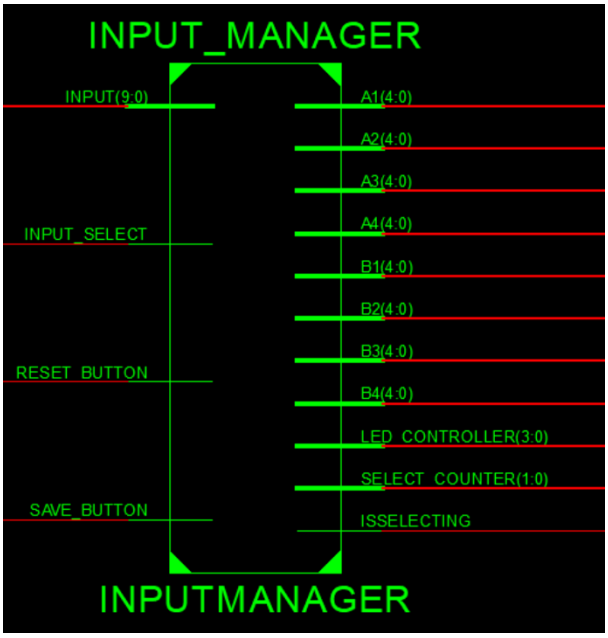
Each block has different tasks:

CLOCKGENERATOR:



- It takes MCLK signal as an input. We created a counter which increases 1 every time MCLK is at the rising edge, and when counter reaches 80000, it starts from zero again. The output signal HZCLOCK becomes one when counter reaches 80000. HZCLOCK is wired to the CLK signal which is one of the input signals of DRIVER.

INPUTMANAGER:



- It takes INPUT, INPUT_SELECT, RESET_BUTTON, and SAVE_BUTTON as an input. INPUT signal is the input sent by the user. We created a 2bit counter which increases 1 every time INPUT_SELECT is pressed, and we created a 4bit signal named ISSELECTED with initial value “0000”.

-Then, we wrote a process due to SAVE_BUTTON and RESET_BUTTON. In this process, when SAVE_BUTTON is pressed and if ISSELECTED is not “1111”, it will assign the INPUT to elements due to counter’ value. For example, when counter is “00”, we will see A1XXA2XX in the seven-segment display, left XX is specified by left 5 switches, and right XX value is specified by right 5 switches, and when SAVE_BUTTON is pressed values will be assigned and corresponding ISSELECTED bit will be ‘1’ ,e.g. for A1A2 ISSELECTED(0) will be ‘1’. After all values are assigned, it won’t go into if statement and we won’t be able to change their values until RESET_BUTTON is pressed. When PRESS_BUTTON is clicked, ISSELECTED will be zero again, we will be able to assign new values to matrix elements.

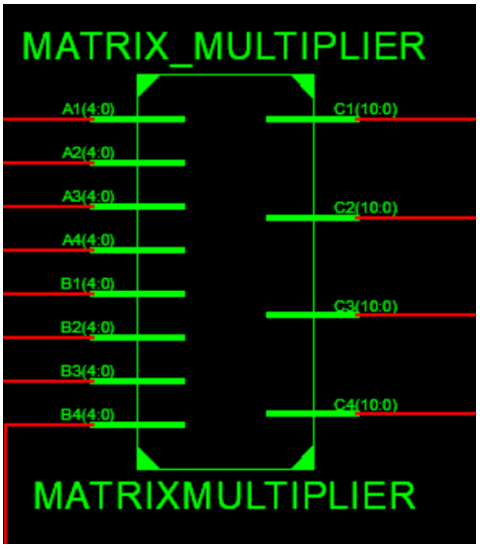
- There are output signal for each element A1,…,B4 which goes to MATRIXMULTIPLIER.

- There is an output signal named ISSELECTED, when all the element values are assigned i.e. ISSELECTING = “1111”, it is ‘0’, otherwise it is ‘1’. This signal goes to DRIVER and makes it possible for output matrix values to be displayed when all input values are determined.

-There is an output signal LED_CONTROLLER that has the same value as ISSELECTED, this signal is wired to main output signal SAVED_LED, and each bit goes to different led, so we will be able to visually realize which input values are assigned.

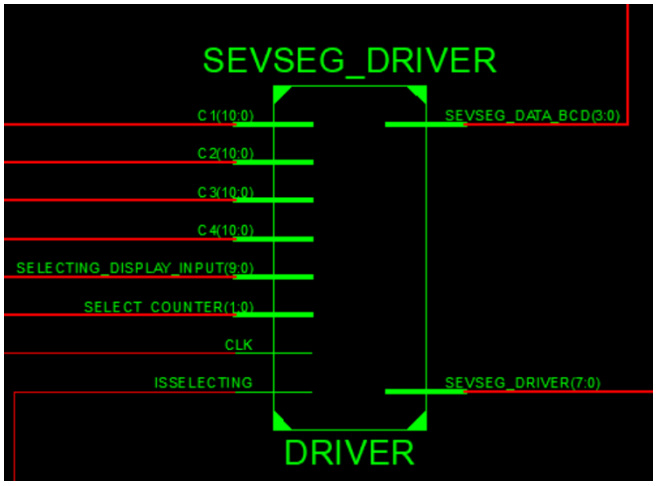
- Also, there is an output signal SELECT_COUNTER which carries value of the counter to DRIVER. Because, we need the same value in DRIVER to display element names correctly.

MATRIXMULTIPLIER:



-It takes the values of inputs A1,A2,…,B4 and converts them to integer with to_integer . After converting, it makes matrix multiplication in integer form and specifies the values of the result matrix C1,…,C4. Then, it converts result values to STD_LOGIC_VECTOR and gives them as outputs which goes into the DRIVER as an input.

DRIVER (SEVSEG_DRIVER):



-It takes C1,…,C4 values as an input from the MATRIXMULTIPLIER, and it has an input named SELECTING_DISPLAY_INPUT, which is actually the ELEMENT_INPUT.

- It takes CLK input that comes from CLOCKGENERATOR. We created a 3bit counter, and counter value increased ‘1’ when CLK signal is at the rising edge; and we used this counter to change the value of output signal SEVSEV_DRIVER. Signal SEVSEG_DRIVER is wired to the output signal SEVSEG_CONTROL. With this part, we change the lighted seven-segment part at each rising edge, and because our eyes cannot distinguish the not lighted period, it seems that they all are lighted all the time.

-To be able to show the results with decimal digits, we wrote 2 procedures, one for user inputs and one for values of result matrix. Because the values of input matrices are 5bit, we created a

procedure named F_5BIT_BINARY_2_8BIT_BCD and we wrote F_11BIT_BINARY_2_8BIT_BCD for results because result values are 11bit. We searched how we could do convert binary to BCD, and saw Double Dabble Algorithm, and adjusted it to our code.

-To assign the values that will be displayed in each segment, we wrote a process. The process assigns values to SEVSEG_DATA_BCD due to SELECT_COUNTER when ISSELECTING is '1'. Also, assigned values changes with COUNTER concurrently to be able to display correct value at each segment. When ISSELECTING is '0', it shows values of result matrix due to 2 least significant bits of SELECTING_DISPLAY_INPUT.

DECODER:

- It converts BCD values to SEVSEV_BUS that enables values to be shown at seven-segment display.