Hardware Assignment 3 Report

By Disha Shiraskar & Srushti Junghare

In the assignment, we were asked to design a circuit that would implement matrix multiplication of 2 128 x 128 input matrices and store the product matrix in ROM. Our project consists of the following parts:

1. Registers :
   We have designed two types of registers in our project, one that stores 8 bit data and other that stores 16 bit data. When the value of read command of the register is 1 it returns the data stored in it as the output and when write is 1 it resets the data stored in it to 0 and becomes ready to store the next input data. The read and write commands of each of the registers are controlled by the FSM.
2. MAC:
   This part of the circuit also known as Multiplier Accumulator Block, becomes active when the mac command becomes 1 (controlled by FSM). On becoming active, it receives 2 , 8 bit numbers as inputs from the registers and finds their product and accumulates this product with the final sum of products calculated till then. In this way, we multiply the rows of matrix 1 with the columns of matrix 2 and upon completing each multiplication (of one row and one column), the mac outputs a 16 bit number corresponding to the each cell of the output matrix. The sum is temporarily stored in register 3.
3. FSM :
   The FSM is the part that controls the operations of all other parts in the circuit. We have used a counter that keeps incrementing from 0 to 134. When the counter becomes 1, it activates the ROM command asking the register to read the data from the ROM. Once ready the counter becomes 2,it activates the command for register 1 and register 2 asking them to take input from the data stored in ROM and store the 8 bit numbers in themselves. On incrementing the counter to 3, the mac commands becomes active, activating the MAC . This asks the MAC to read data stored in the registers and find their product(during the 1st iteration) and store its value in sum. In the subsequent iterations , the circuit continues to remain MAC state and keeps on accumulating the products of the numbers in one row and one column of matrix 1 and matrix 2 respectively. During each iteration in MAC the value of counter

increments by 1 . Thus, when the counter reaches the value 133 the final sum generated in the MAC is stored in the register 3 .When the counter becomes 134 the value stored in the register 3 is read and then stored at the appropriate address in the RAM. On completion of all these processes the counter resets itself to zero. This completes one cycle of FSM. All the operations mentioned above are being done using the inbuilt clock of the board.

4. Main:
   This file contains the porting of the components. It takes 14 bit address and input and generates 16 bits which are then mapped as input to the MUX file. Finally, this function returns the 7 outputs corresponding to cathodes and 4 outputs corresponding to anodes of seven segment decoder of the board. Hence displaying the number at the corresponding cell.
5. ROM:
   Following the steps in assignment, we generated 2 ROMs in order to store the two 128x 128 input matrices in each.We have also used appropriate mathematical formula to relate the matrix and colum major representations.
6. RAM :
   Following the steps given in the assignment, we have also generated 1 Ram to store the output 128 x 128 matrix.
7. MUX:
   This part corresponds to our Hardware Assignment 1 in which we have designed the seven segment decoder which takes 16 bit input and displays the corresponding output on the basys3 board.

Simulation Snapshot :