HW ASSIGNMENT 3

Register

It reads data from ROM when read_enable is '1' and it outputs data to MAC when write_enable is '1'.

MAC

It multiplies two 8-bit numbers according to the input addresses given from FSM and outputs a 16-bit number.

FSM

It takes as input clk and controls the following signals:

1. RAM1_we: If it is set as '1' then it writes memory in RAM.
2. Input_register_re: If it is set as '1' then it reads data from ROM and writes it to the register,i.e., changes the stored value.
3. Input_register_we: If it is set as '1' then it sends two 8-bit numbers two MAC.
4. Output_register_re: It is set as '1' after 128 iterations of MAC. If it is set as '1' then it reads data from MAC and stores the 16-bit output in the register.
5. Output_register_we: If it is set as '1' then it reads writes data in RAM. This is also set as '1' after 128 iterations of MAC.
6. we_mac: If it is set as one then only MAC multiplies the input numbers.
7. cntrl: If it is '1' then MAC gives the product of the input numbers, otherwise if it is '0' then it outputs sum of the stored number and the product of the input numbers. In other words, cntrl tells if it is the first product of the compute or not.
8. address_rom1 & address_rom2: They govern the address from where the next numbers, that are to be sent to MAC, are read. For the addresses of first matrix we increment the input address by 128 after every computation and for the second increment it by 1.
9. address_ram: It governs the address where the accumulated sum is written. It is calculated as follows:

   Row number=(address_rom1) mod 128

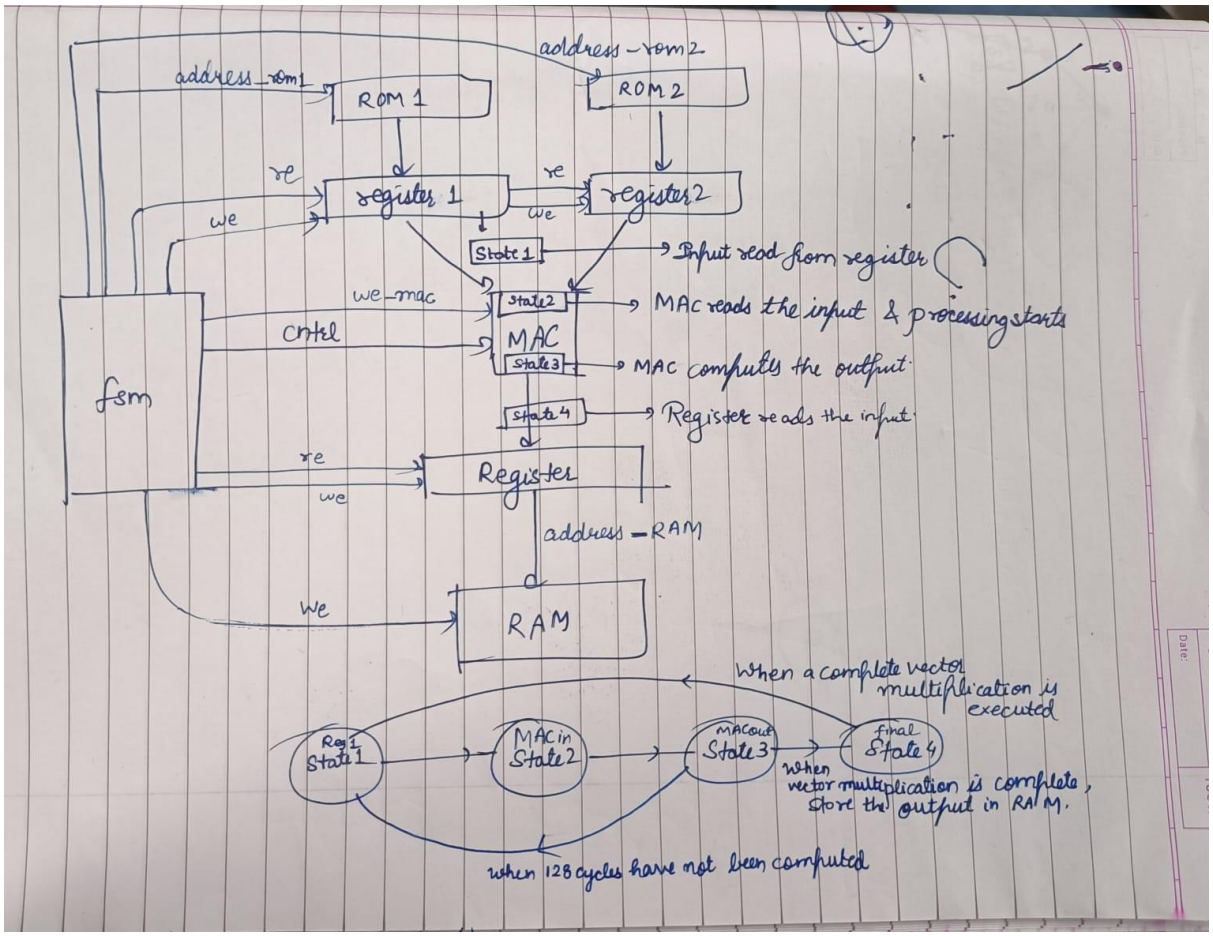
   Column number=(address_rom2 – (address_rom2 mod 128))/128

   address_ram=column_number*128+row_number

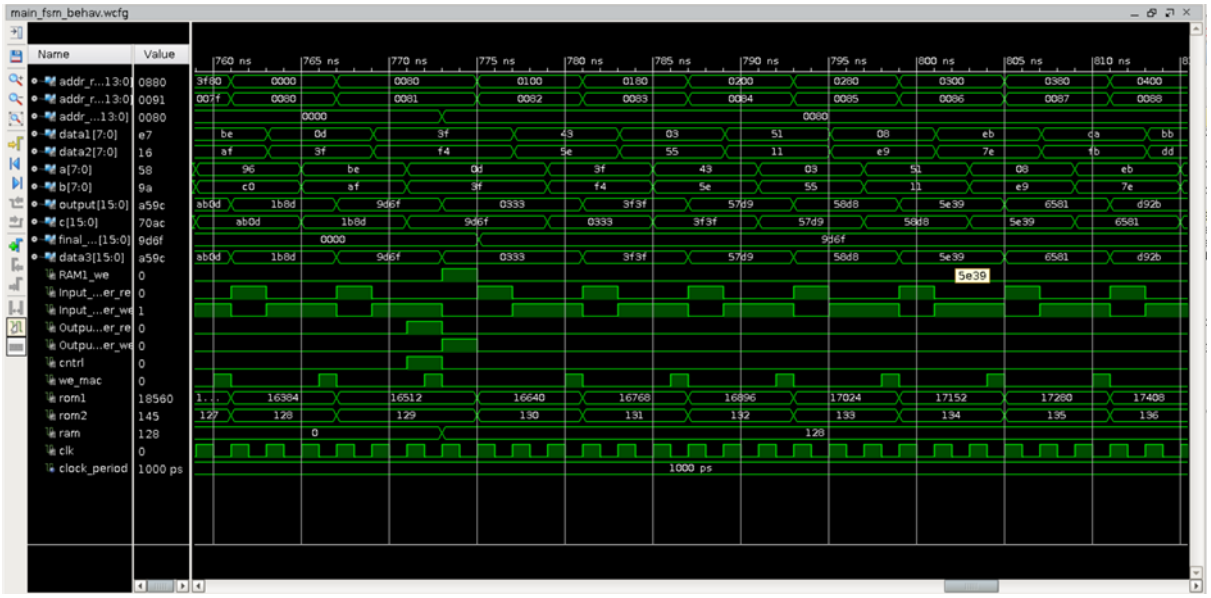
There are 4 states in FSM:

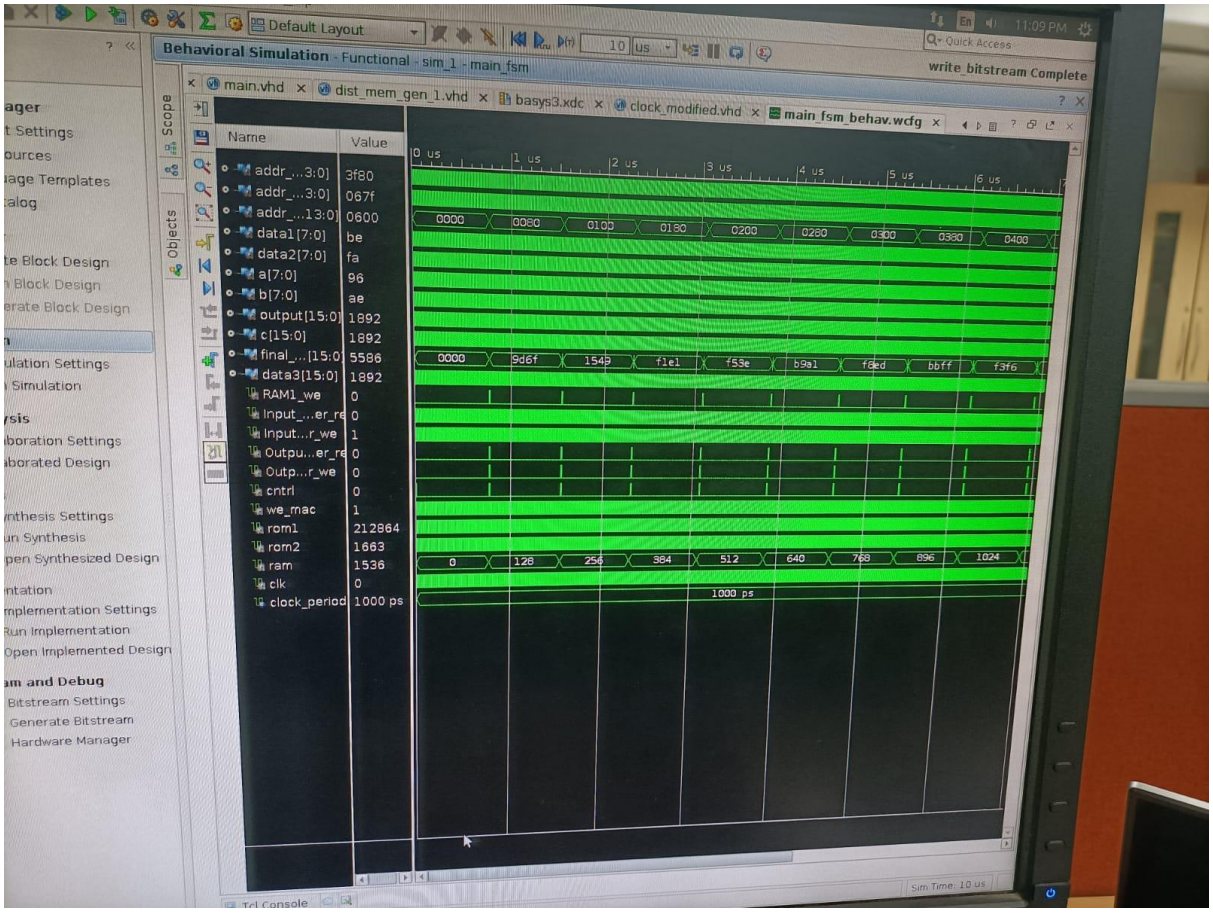
1. Reg1: Input is read from ROM and written in register
2. MACIn: Input is read from register and given to MAC. In the first iteration cntrl is set to '1'.

3. MACOut: This state decides whether we should go to the first state or to the final state based on number of iterations of MAC. It takes the value from MAC and gives it to the 16-bit register.
4. final: It reads data from the 16-bit register and stores it in the RAM at the specified address.

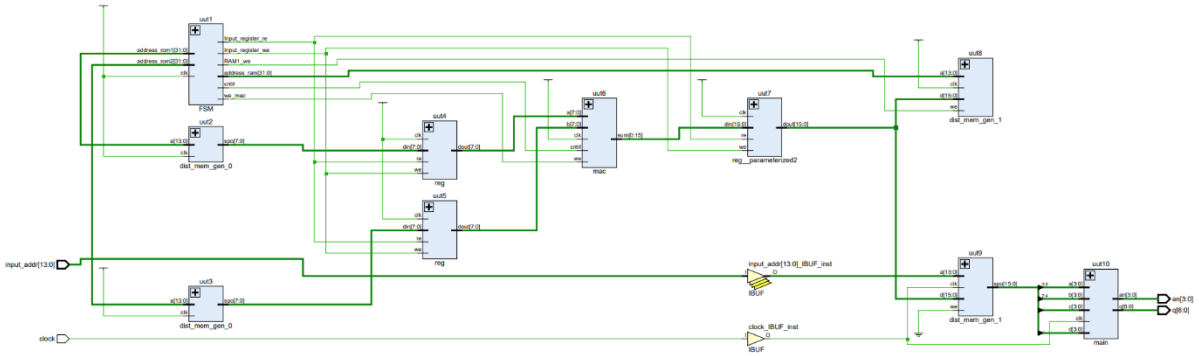


STATE DIAGRAM





Simulation Snapshots



Schematic Diagram