COL 215 HARDWARE ASSIGNMENT 3

REPORT

RISHABH KUMAR 2021CS10103

RUCHIR SINGH 2021CS10119

MULTIPLIER ACCUMULATOR (MAC)

The first component to be designed is a MAC unit that has a 8x8 multiplier, a register and a 16-bit adder to keep accumulating the products. For performing matrix multiplication, it will read out two 8 bit input values from the memory and perform the multiplication using multiplier. The multiplier will generate a 16-bit output which will be sent to the adder for accumulation. Once all the input values from row and column have been multiplied and accumulated, the final output will be stored in the RAM. This component is implemented in the file mac.vhd .

RANDOM ACCESS MEMORY (RAM)

This memory will be used to store the final output matrix of size 128x128, with each value as 16-bit unsigned integer. At every step of multiplying a row and a column, the output we get is stored here.

READ ONLY MEMORY (ROM)

Input matrix: 128x128 image with each value as an 8-bit unsigned integer. For storing the input we need 16,384 words, 8 bits wide. The address width of the memory is 14 bits.

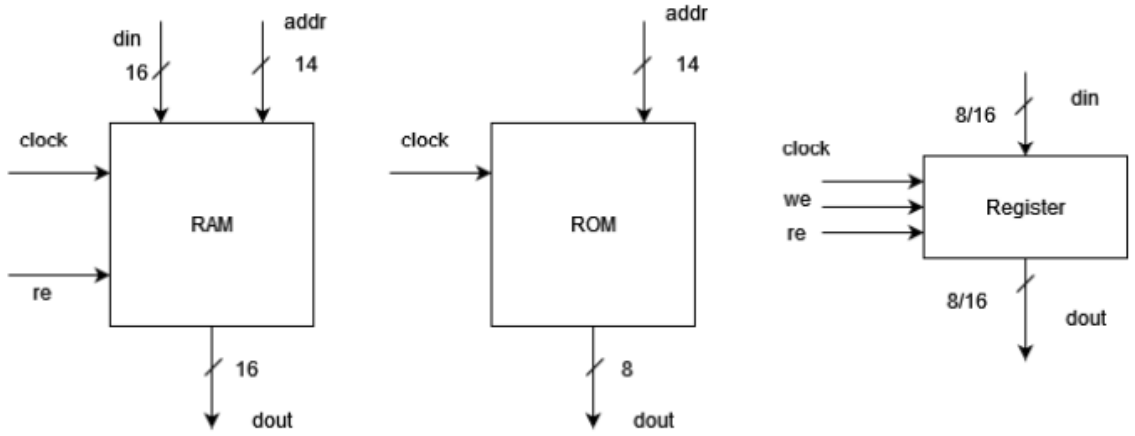
REGISTER

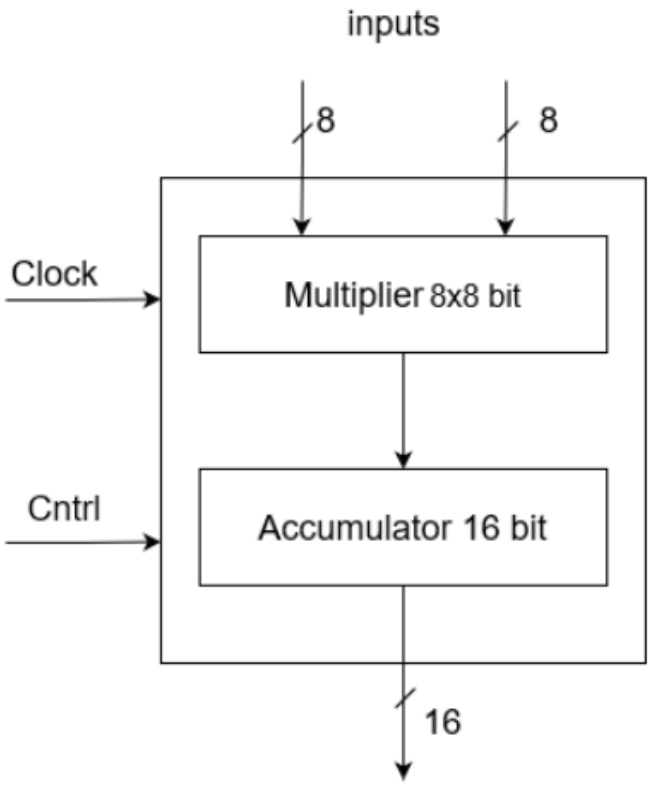
the Registers are sequential elements (set of flip-flops) that can be used to store data at clock edges. It has two input flags:

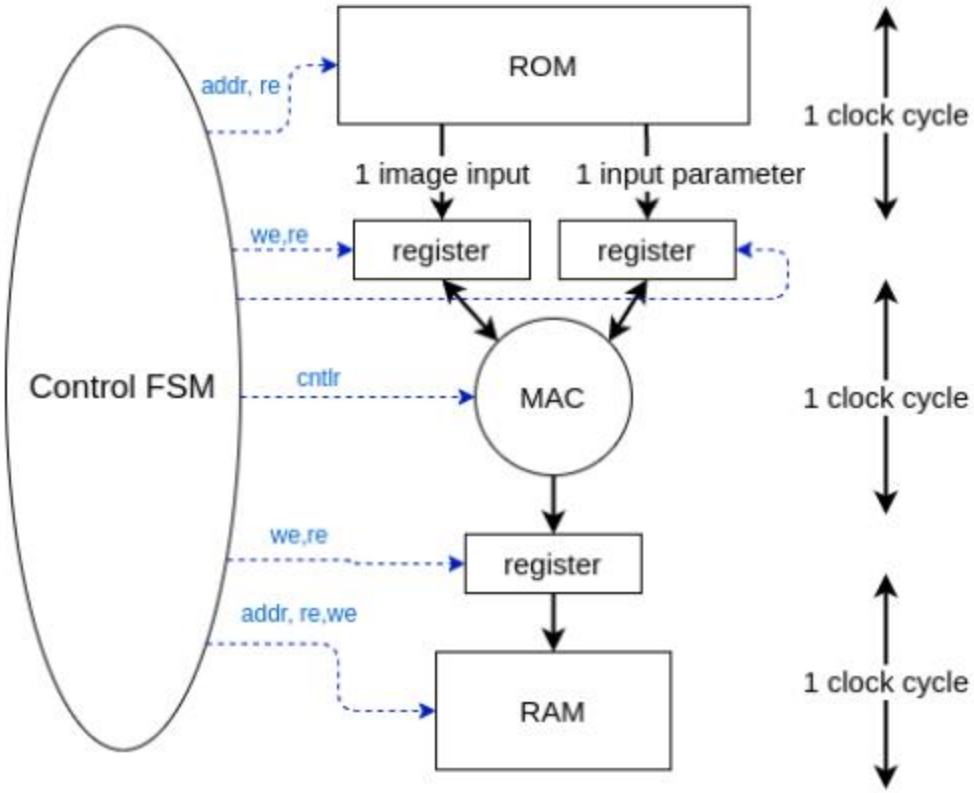
- Read: ’re’ signal to read data stored in the register.
- Write: ’we’ signal to write the data in the register.

Control Path

The control path is basically an FSM in your design which is supposed to do all the sequencing of the data required starting from generating addresses of the data in the order in which you want to read the data, generating the cycle-wise signals to be sent to the datapath to control the modules in the design.







Creating finite state machine:

It consists of two states ADD and SUB with input flags M (to change the operation) and DONE (to know when the current operation is completed and output flag cntrl_add_sub to set the operation in programmable adder/subtractor.

It consists of two blocks:

1) Sequential Block: State is to be updated on the clock edge.
2) Combinational Block: It consists of logic to update the next state based on flag values and the current state.