Name: Harsha Vardhan Reddy Guthikonda

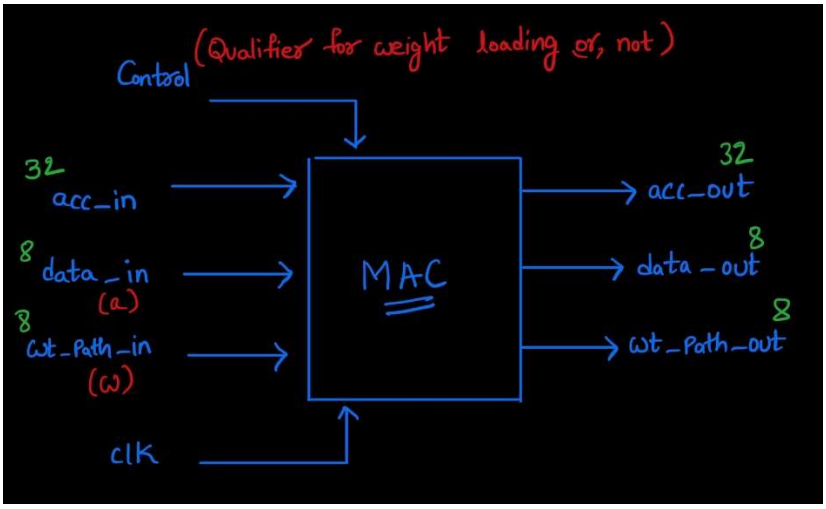
EEE598 Hardware and Systems for Machine Learning

HW #2

I implemented a weight-stationary 4*4 Systolic array in Verilog successfully and verified the design successfully in model Sim. Firstly, I designed the basic unit (processing element or MAC unit) in Verilog. The logic for the MAC unit is as follows.

MAC unit logic:

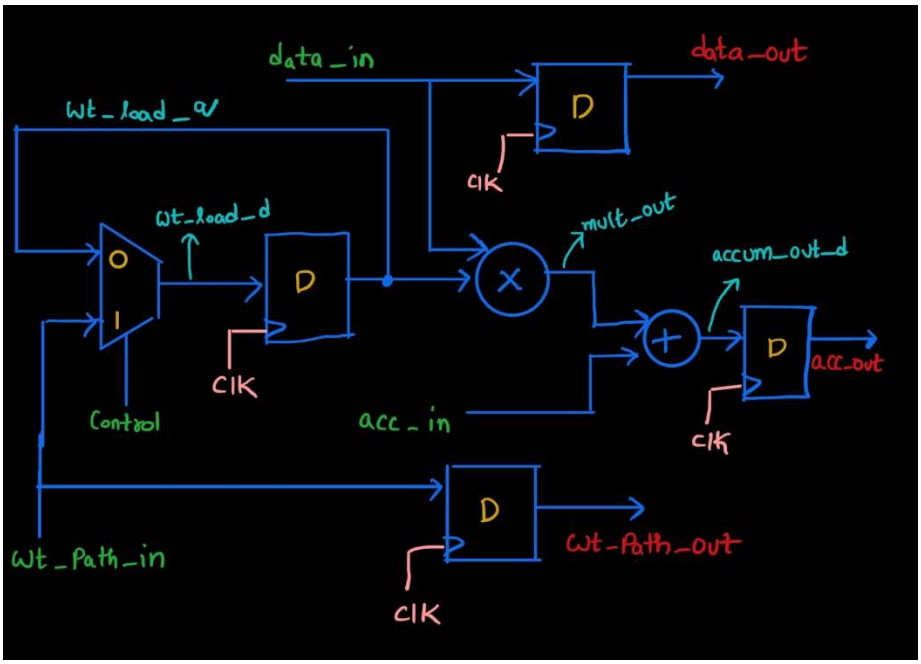
The MAC unit logic performs multiplication between data, weight and accumulates/sums the resultant product with acc_in value. The MAC unit consists of the following inputs and outputs as depicted in the block diagram.



Block Diagram

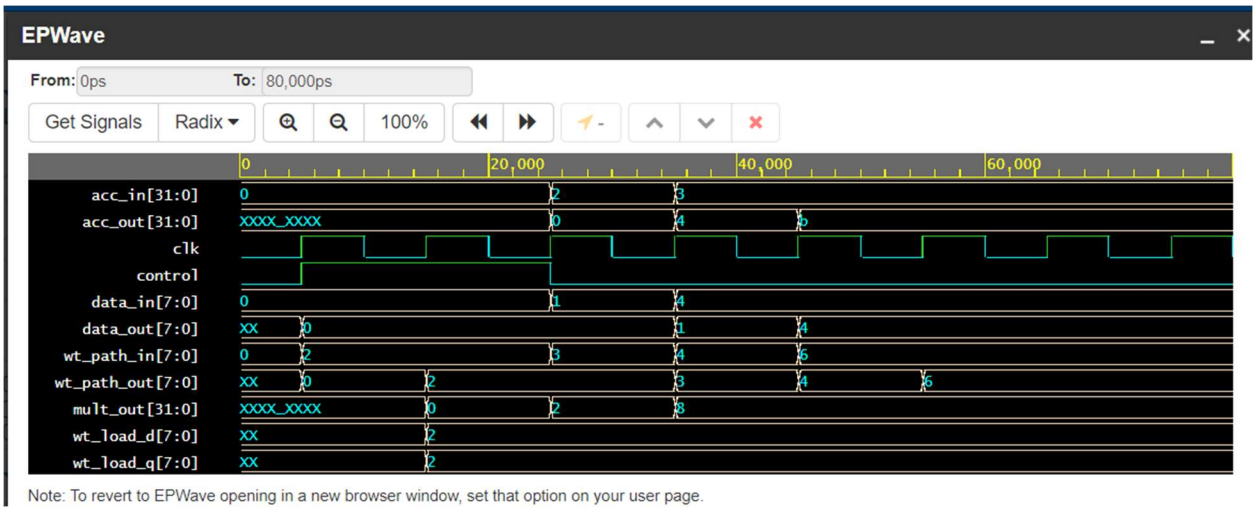
The logic is as follows; we are implementing weights stationary matrix multiplication. So, the MAC unit should be able to hold the weight values when control is 0. This is done using a Flip flop and a Mux in front. Then multiply and accumulate operations should be done which are implemented using combinational logic. The output is connected to the register to ensure data correctness at the output. The logic for the MAC unit is shown in the below circuit diagram. The design performs the multiplication of two values (data_in and wt_load_q) and accumulate the result with an existing value (acc_in). The weights get loaded into the register when the control is 1 and those values will be within held until the completion of the operations.

Name: Harsha Vardhan Reddy Guthikonda



MAC Logic

I verified the above design by writing test bench for the above design. It begins by waiting for a positive edge of a clock signal (clk) and then sets the control signal to 1, indicating that a weight-loading operation is about to begin. The 'wt_path_in' value is set. Gave some dealy; another positive clock edge is awaited, and control is set to 0, signifying the end of the weight-loading operation. The wt_path_in input is changed, and subsequent lines set values for acc_in and data_in. This pattern repeats for several clock cycles, each with different values for wt_path_in, acc_in, and data_in. After a total simulation time of 40-time units, the simulation is concluded with the $finish statement. The following is the timing diagram of the simulation.



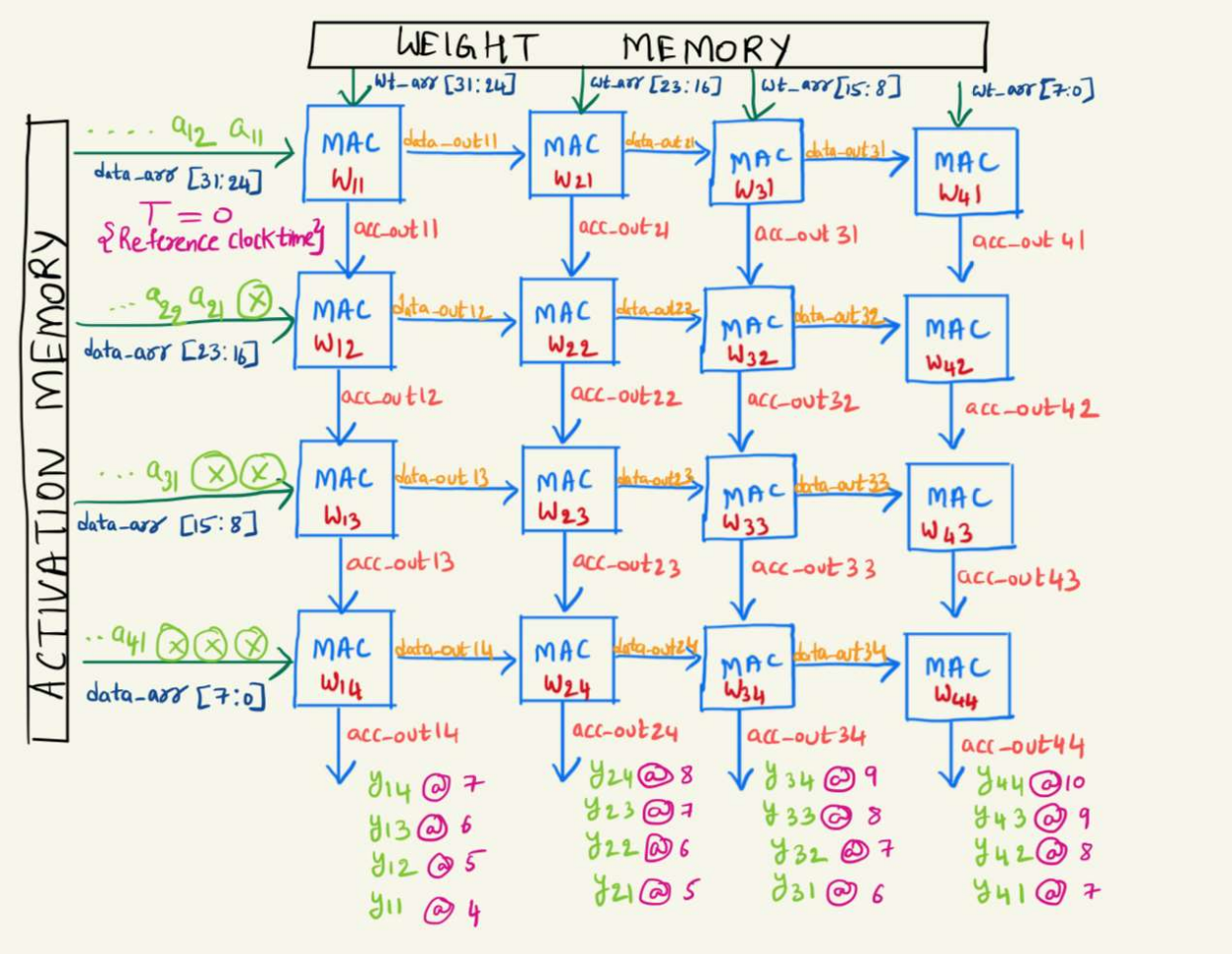
MAC Timing Diagram

Name: Harsha Vardhan Reddy Guthikonda

One quick observation is after the weights are loaded, we are getting the outputs in the next clock cycle after the input and acc_in values are loaded which ensures a good scalability to 4*4 Systolic array.

Systolic array logic:

To build a 4*4 systolic array; I should use 16 MAC units and develop a connection between them appropriately so that dataflow is synchronized to perform matrix multiplication. The below figure illustrates the design of 4*4 weight stationary systolic array.



4*4 Systolic Array

When control = 1; all the weights { (w11, w21, w31, w41), (w12, w22, w32, w42), (w13, w23, w33, w43), (w14, w24, w34, w44)} are loaded to their corresponding PE's after 4 clock cycles.

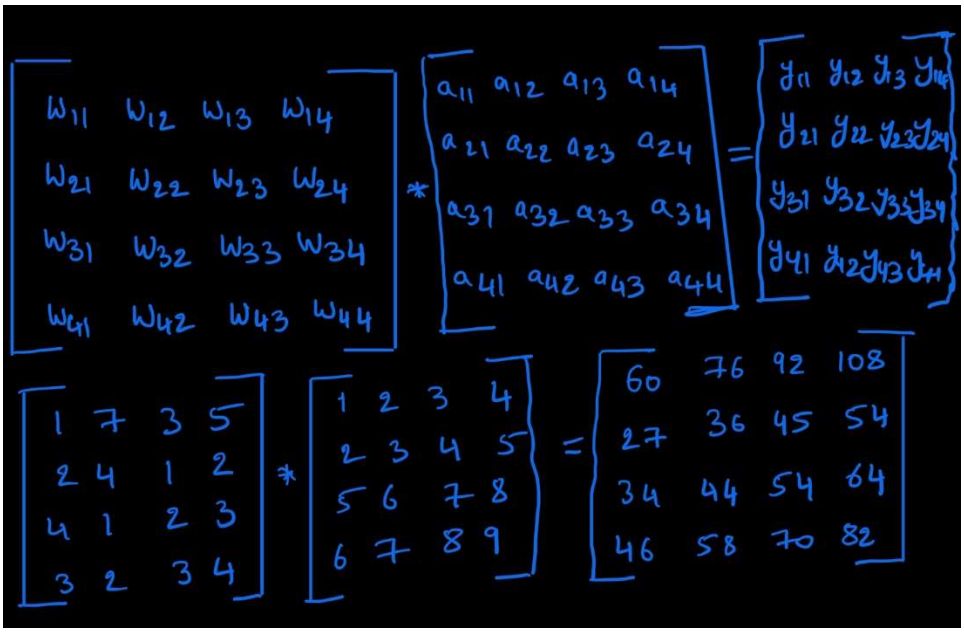
When control = 0; the data from activation memory starts flowing. To get desired acc_out we need to make sure the a21 arrives one cycle later than a11; similarly, a31 arrives one cycle later that a21; a41 arrives one cycle later than a31 to ensure acc _out in the intermediate stages are ready.

To verify the design, I had written a test bench to perform the matrix multiplication. I got the desired output through the waveform which depicts the correct functionality of the system.

Name: Harsha Vardhan Reddy Guthikonda

```
initial begin
@(posedge clk);
control=1;
wt_arr=32'h 05020304;

@(posedge clk);
wt_arr=32'h 03010203;

@(posedge clk);
wt_arr=32'h 07040102;

@(posedge clk);
wt_arr=32'h 01020403;


@(posedge clk);

control=0;

data_arr=32'h 00000000;
@(posedge clk);
data_arr=32'h 01000000;

@(posedge clk);
data_arr=32'h 02020000;

@(posedge clk);
data_arr=32'h 03030500;

@(posedge clk);
data_arr=32'h 04040606;

@(posedge clk);
data_arr=32'h 00050707;

@(posedge clk);
data_arr=32'h 00000808;

@(posedge clk);
data_arr=32'h 00000009;
```

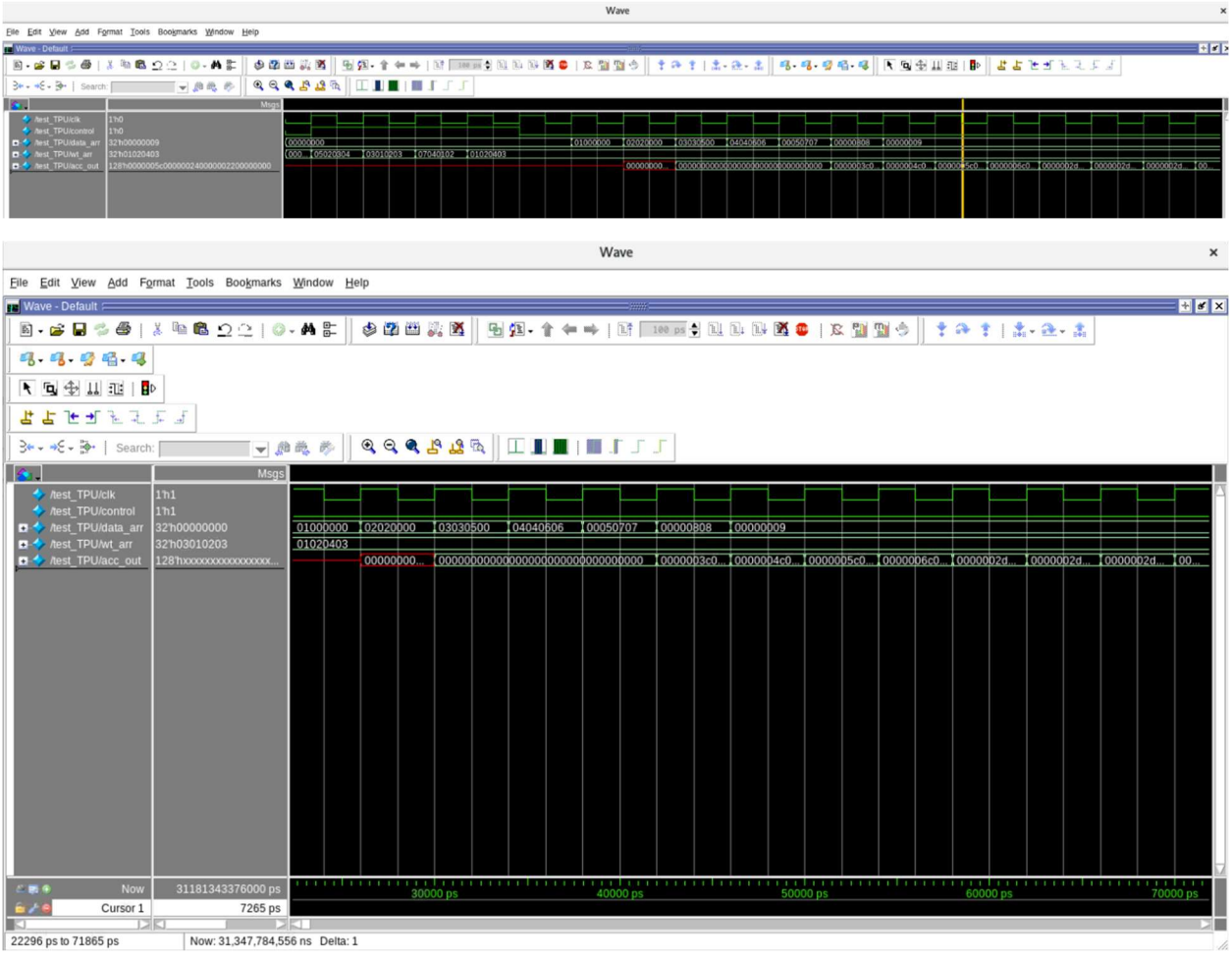
This is a code snippet from my test bench. The weights and data are sent as an array; the array 32-bit is broken down into 8-bit data given to the corresponding PE elements. The weight and activation memory test values as follows



Activation and Weight Matrices

I checked this y matrix against acc_out values from the simulator and found to be same. We can see acc_out value at 42000ps is 0000003c000000000000000000000000= {y11, 0, 0, 0} which means the value of y11 is 60. Similary acc_out value at 50000ps is 000005c000000240000002200000000 = {y13, y22, y31, 0} which implies the value values of y13, y22, y31 are 92, 36, 34.

Name: Harsha Vardhan Reddy Guthikonda



4*4 Systolic array waveform

From the above diagram the output matrix value start coming from the time stamp 42000ps

22000ps: Reference Time (data_arr value is passed); acc_out values as follows { T=0}

```
always @(posedge clk) begin
  acc_out <= {acc_out14, acc_out24, acc_out34, acc_out44};
end
```

Technically y11 should come out after 4 cycles but I put an extra register at the output due to above code snippet. So y11 will be present at the output after 5 clock cycles. That means 22000 + 5*2000 = 42000ps

42000ps = {y11, 0, 0, 0} 46000ps = {y12, y21, 0, 0} 50000ps = {y13, y22, y31, 0}

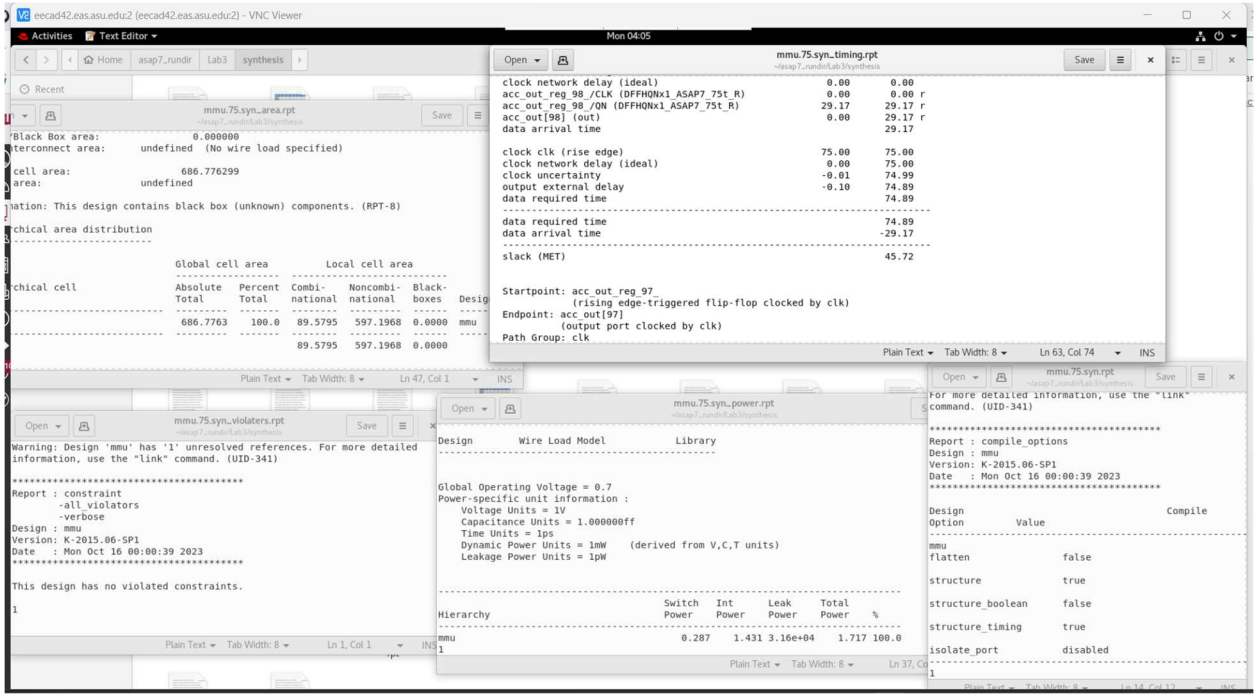
54000ps = {y14, y23, y32, y41} 58000ps = {y24, y33, y42, 0} 62000ps = {y34, y43, 0, 0}

66000ps = {y44, 0, 0, 0}

Name: Harsha Vardhan Reddy Guthikonda

After logic synthesis:

After doing logic synthesis on mmu.v design with ASAP7 PDK. I generated Timing, Power, Area, synthesis and violations reports. The input to the logic synthesis is behavioral Verilog netlist and the output is structural Verilog netlist which only instantiates standard cells.



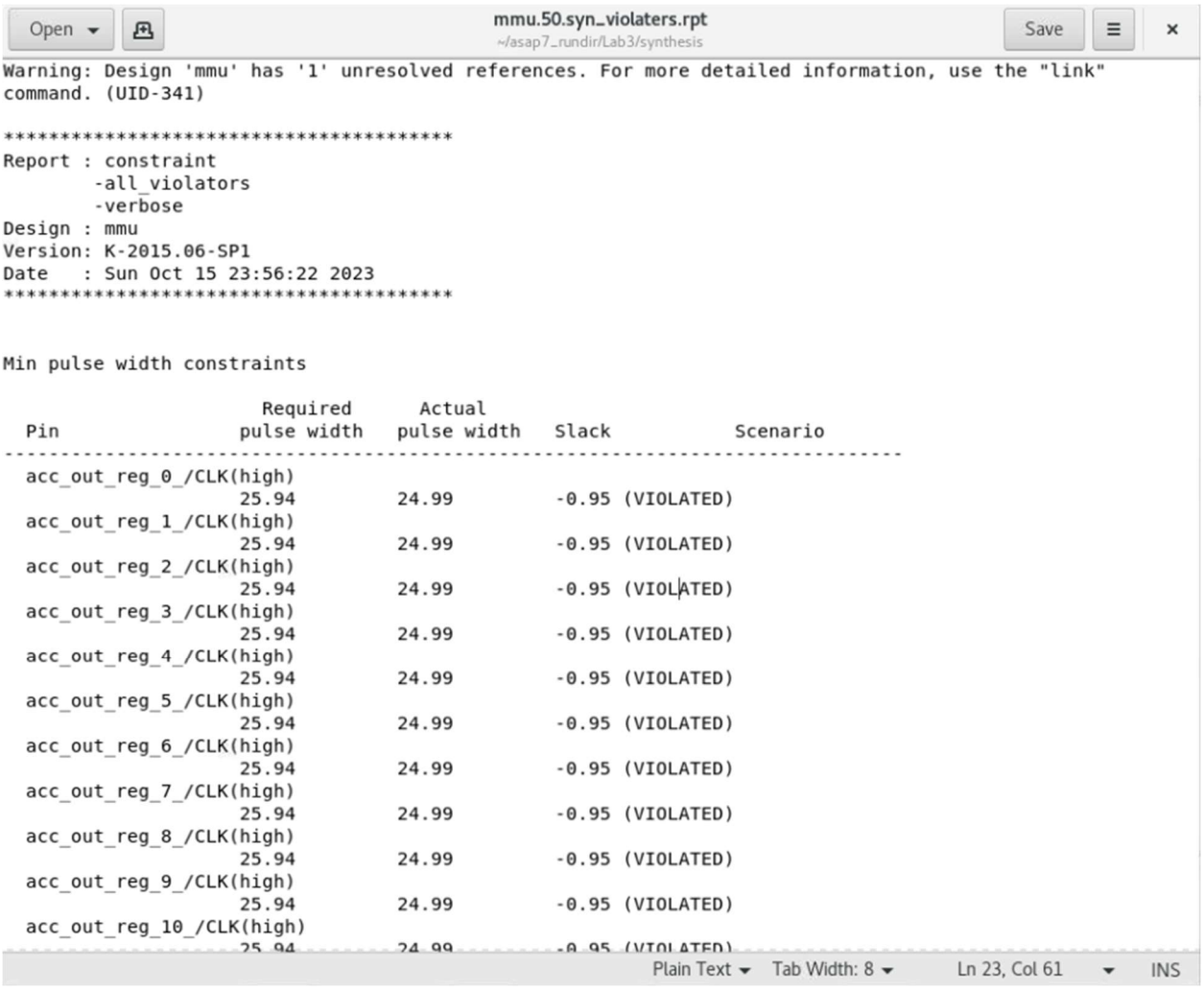
Screenshot of Area, Timing, Power Reports

```
# ##################################CHANGE THIS#####################################################
# TOP_LEVEL_NAME
set top_level                    "mmu"
# PATH TO YOUR VERILOG NETLIST
set netlist_search_path "/afs/asu.edu/users/h/g/u/hguthiko/asap7_rundir/Lab3/RTL/"
# NETLIST NAME
set netlist_name "mmu.v"
# CLOCK PERIOD
set clk_period                   75
# ##################################################################################################
```

Example.dc.tcl file modification for mmu module

If clock_period is changed to 50 then the following violations are encountered

Name: Harsha Vardhan Reddy Guthikonda



```
Open ▾   mmu.50.syn_violaters.rpt   Save
~/asap7_rundir/Lab3/synthesis

Warning: Design 'mmu' has '1' unresolved references. For more detailed information, use the "link"
command. (UID-341)

****************************************
Report : constraint
        -all_violators
        -verbose
Design : mmu
Version: K-2015.06-SP1
Date   : Sun Oct 15 23:56:22 2023
****************************************


Min pulse width constraints

                     Required      Actual
  Pin                pulse width   pulse width   Slack           Scenario
------------------------------------------------------------------------------
  acc_out_reg_0_/CLK(high)
                     25.94         24.99         -0.95 (VIOLATED)
  acc_out_reg_1_/CLK(high)
                     25.94         24.99         -0.95 (VIOLATED)
  acc_out_reg_2_/CLK(high)
                     25.94         24.99         -0.95 (VIOLATED)
  acc_out_reg_3_/CLK(high)
                     25.94         24.99         -0.95 (VIOLATED)
  acc_out_reg_4_/CLK(high)
                     25.94         24.99         -0.95 (VIOLATED)
  acc_out_reg_5_/CLK(high)
                     25.94         24.99         -0.95 (VIOLATED)
  acc_out_reg_6_/CLK(high)
                     25.94         24.99         -0.95 (VIOLATED)
  acc_out_reg_7_/CLK(high)
                     25.94         24.99         -0.95 (VIOLATED)
  acc_out_reg_8_/CLK(high)
                     25.94         24.99         -0.95 (VIOLATED)
  acc_out_reg_9_/CLK(high)
                     25.94         24.99         -0.95 (VIOLATED)
  acc_out_reg_10_/CLK(high)
                     25.94         24.99         -0.95 (VIOLATED)

Plain Text ▾   Tab Width: 8 ▾   Ln 23, Col 61 ▾   INS
```

Area Report :

Number of ports: 194

Number of nets: 1028

Number of cells: 274

Number of combinational cells: 128

Number of sequential cells: 146

Number of macros/black boxes: 0

Number of buf/inv: 128

Number of references: 3

Name: Harsha Vardhan Reddy Guthikonda

Combinational area: 89.579521

Buf/Inv area: 89.579521

Noncombinational area: 597.196777

Macro/Black Box area: 0.000000

Net Interconnect area: undefined (No wire load specified)

Total cell area: 686.776299

Total area: undefined
