Binary Artificial Neural Network

Name: Ayushya Singh Chauhan
Unityid: assinghc
StudentID: 200367721

Delay (ns to run provided example).
Clock period: 5ns
cycles": 49

Logic Area: 1903.230032 (um^2)

Memory: N/A

1/ (delay. Area) ($ns^{-1}.um^{-2}$)

=1/ (5 x49 x 1903.23)
=0.000002144
=$2.145x10^{-6}$

Delay (TA provided example. TA to complete)

1/(delay.area) (TA)

Abstract

A binary artificial neural network to perform binary convolution on a matrix using weight matrix. Input matrix is stored in SRAM and weight matrix is stored in weight SRAM. The size of input matrix is (16x16), (12x12) & (10x10) and weight matrix of size 3x3. The output matrix is stored to output SRAM with zero padding as per the size of input matrix. The design was divided into two modules. The first module "PE" consisted of the logic and computational part. The second module was used for control signal in which PE module was instantiated fourteen times. The input was taken from an input SRAM file and passed on to register CC followed by BB followed by AA. Since first two address values had dimensions of matrix and it shouldn't be read as matrix row, three registers A, B, C were used to pass the required values to the design PE module after waiting for three cycles, because the dimension values were flushed out of registers after three cycles. After three cycles A, B, C had the first three rows of the matrix and in one clock cycle output matrix row 1 was computed. Each PE module was provided 3 bits each from A, B, C which were concatenated to form a 9-bit value, weight matrix 9-bit XNOR operation with the concatenated value was sent to 9-bit adder which added every bit. The result was sent to a comparator which gave a value 1 if it is greater than 4 else gave a 0. All fourteen PE modules gave 1 bit each and were concatenated together to write the output in SRAM. According to the size of matrix the zero padding was done before writing the value in SRAM.

Binary Artificial Neural Network

Ayushya Singh Chauhan

1. Introduction

- Designed and implemented a binary artificial neural network to perform binary convolution. Input matrix is stored in SRAM and weight matrix is stored in weight SRAM. The size of input matrix is (16x16), (12x12) & (10x10) and weight matrix of size 3x3. The output matrix is stored to output SRAM with zero padding.

- The design achieved a clock period of 5ns and area of 1903.23 $um^{2.}$ The design computation took 49 cycles. The output was validated against the golden output file.

2. Micro-Architecture

- The design was divided into two modules. The first module "PE" consisted of the logic and computational part. The second module was used for control signal in which PE module was instantiated fourteen times. The input was taken from an input SRAM file and passed on to register CC followed by BB followed by AA. Since first two address values had dimensions of matrix and it shouldn't be read as matrix row, three registers A, B, C were used to pass the required values to the design PE module after waiting for three cycles, because the dimension values were flushed out of registers after three cycles. After three cycles A, B, C had the first three rows of the matrix and in one clock cycle output matrix row 1 was computed. Each PE module was provided 3 bits each from A, B, C which were concatenated to form a 9-bit value, weight matrix 9-bit XNOR operation with the concatenated value was sent to 9-bit adder which added every bit. The result was sent to a comparator which gave a value 1 if it is greater than 4 else gave a 0. All fourteen PE modules gave 1 bit each and were concatenated together to write the output in SRAM. According to the size of matrix the zero padding was done before writing the value in SRAM.

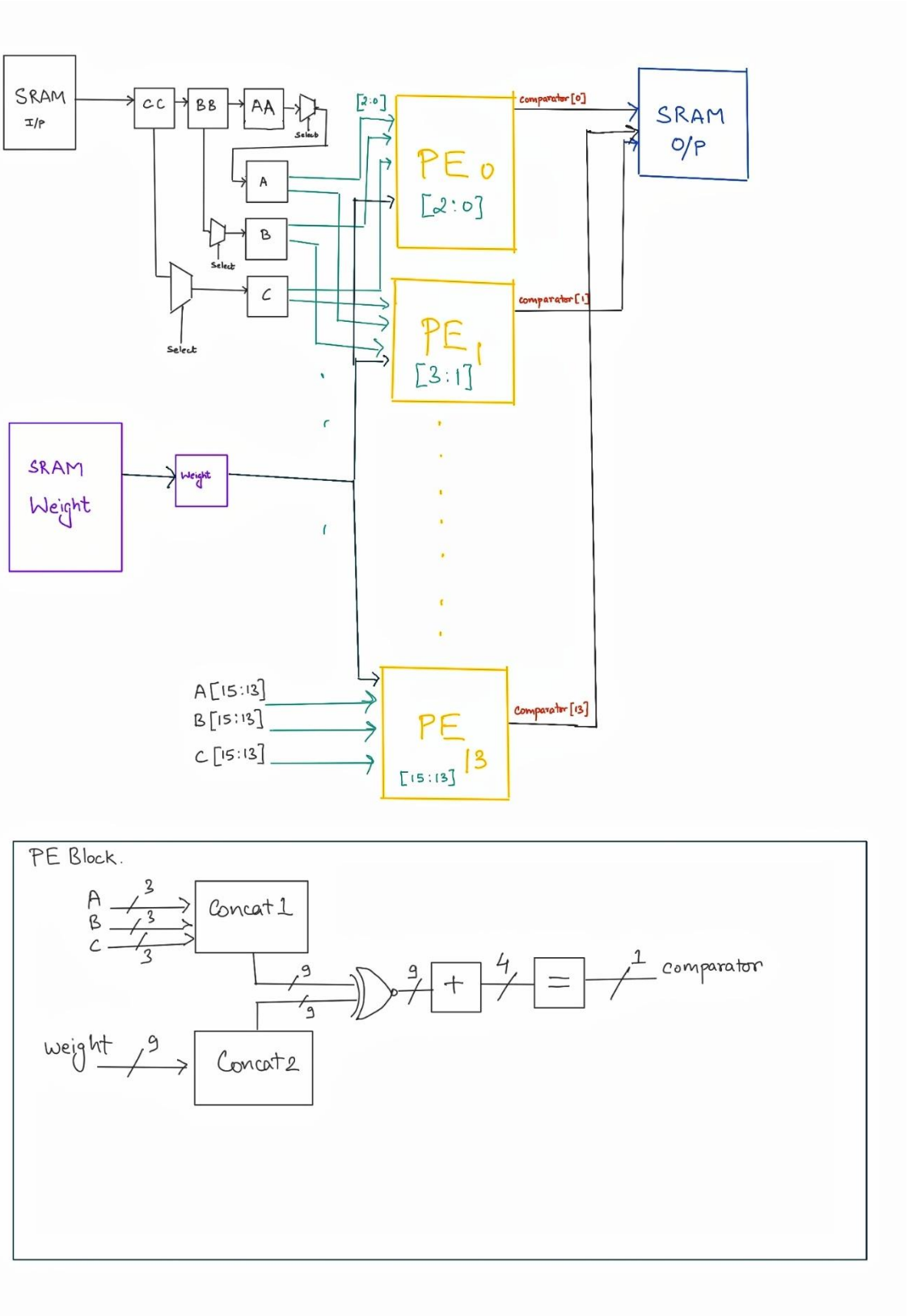

Figure 1: High level architecture drawing, and description of data flow.

3. Interface Specification

The interface consists of clock, reset, dut_run, dut_busy , read_address, weight_address, read_address_select, weight_address_select, write_address_select, write_enable at the input port and write_data and write_address at the output port.

Signal	Width	Function
Clock	1	Clock for the design. Clock used = 5ns.
Reset_b	1	Reset for the design
Dut_run	1	Dut_run is used to trigger dut_busy. So that the module starts its computation.
Dut_busy	1	Dut_busy is set to 1 when computation starts and set to 0 when all the input address are computed as results.
dut_sram_read_address	12	Reads the address from input file.
dut_wmem_read_address	12	Reads the address from the weight file.
dut_sram_write_address	12	Write address computes the address to which output needs to be written in output file.
sram_dut_read_data	16	Reads input data.
wmem_dut_read_data	16	Reads weight data.
dut_sram_write_data	16	Writes computed output data.
read_addr_sel	2	Select line for read address.
wread_addr_sel	2	Select line for weight address.
wwread_addr_sel;	2	Select line for write address.

select	2	select goes high after waiting for three cycles when dut_run goes high. Data Flows as follows AA ->A BB ->B CC ->C It is set to low when Sizecount is 0.
dut_sram_write_enable	1	Write enable is set to high when the computed value is set achieved and needed to be written at a particular address.
Sizecount	2	Take value from Read data when set to 0 and set to 1 in next cycle, it then starts decrementing value after every cycle till it goes to 0. Otherwise at 2 it retains the value.

Table 1**:** List of Signals

4. Technical Implementation

The design was given three types of matrices 16x16, 12x12, 10x10. The first lines read from the input SRAM are the input matrix size. The dimension is stored in a variable Sizecount which is then decremented after every cycle. After the second line from input matrix, input matrix values are read as a 16 bit value. The input was taken from an input SRAM file and passed on to register CC followed by BB followed by AA. The data is sent to a module "PE" consisted of the logic and computational part. Since first two address values had dimensions of matrix and it shouldn't be read as matrix row, three registers A, B, C were used to pass the required values to the design PE module, using a select line "select" which is set to 1 after waiting for three cycles, because the dimension values were flushed out of registers after three cycles. After three cycles A, B, C had the first three rows of the matrix and in one clock cycle output matrix row 1 was computed. Each PE module was provided 3 bits each from A, B, C which were concatenated to form a 9-bit value, weight matrix 9-bit XNOR operation with the concatenated value was sent to 9-bit adder which added every bit. The result was sent to a comparator which

gave a value 1 if it is greater than 4 else gave a 0. Total such fourteen PE modules are instantiated in the top module. All fourteen PE modules gave 1 bit each and were concatenated together to write the output in SRAM. According to the size of matrix the zero padding was done before writing the value in SRAM.
After all the results were written the dut_busy is set to low.

5. Verification

Simulated the design using model sim and verified the outputs stored in output.dat file with the golden outputs. Separate testbenches were created for module PE and module MyDesign to check correctness of functions. Testbench for module PE consisted of A, B, C register 3 bit each value with the weight matrix 9-bit value. The expected output waveform was matched with the resultant simulated waveform. The module MyDesign implementation was done with data path first and then control path. Another testbench was written to verify the output waveform was done and output was verified. After the expected output of own testbench matched, the given testbench was simulated with the SRAM's and all modules were moved to one file. The outputs were verified against the golden output for correctness.

6. Results Achieved

Throughput, area, power/energy (if applicable), etc.
A clock period of 5ns was achieved for the design with an area of 1948.45um^2. The combinational area was 1009.47um^2. and non-combinational area was 893.76um^2. Setup time for 5ns clock was 0.0226ns and hold time was 0.0303ns. The design took 49 cycles to compute results for the given testbench.

7. Conclusions

Successfully designed and synthesized a binary convolutional neural network which reads values from SRAM and computes value to write output in SRAM. Design was verified with golden output. The clock achieved was 5ns and the area is 1903.23um^2. Trade off between clock and area was kept in mind. Earlier a clock of 3.2ns was achieved but area increased to 2112 um^2. Best optimum results were achieved around the range 4.5-5ns for this design keeping performance in mind. Computational power was kept in mind while designing. All the key mantras for designing were followed properly. Working on a complex design taught me SRAM interfacing, synthesis issues and how to handle them, optimizing the design to achieve good performance overall, checking output with an expected waveform while.

OUTPUT

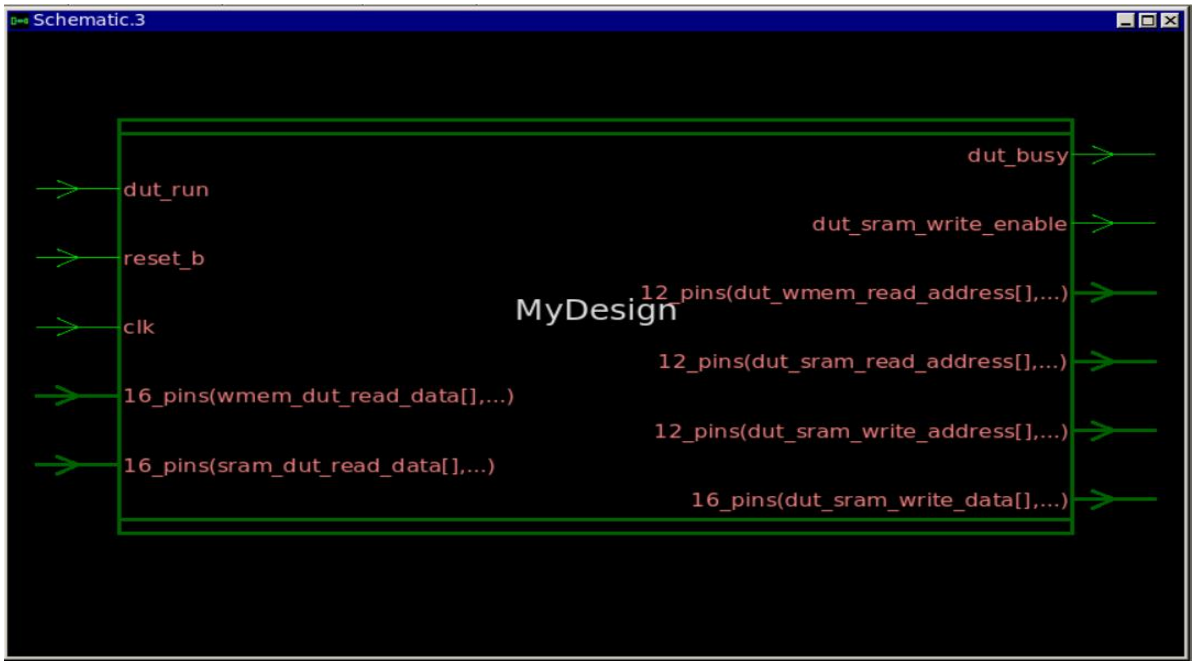

Figure 2: Design Schematic

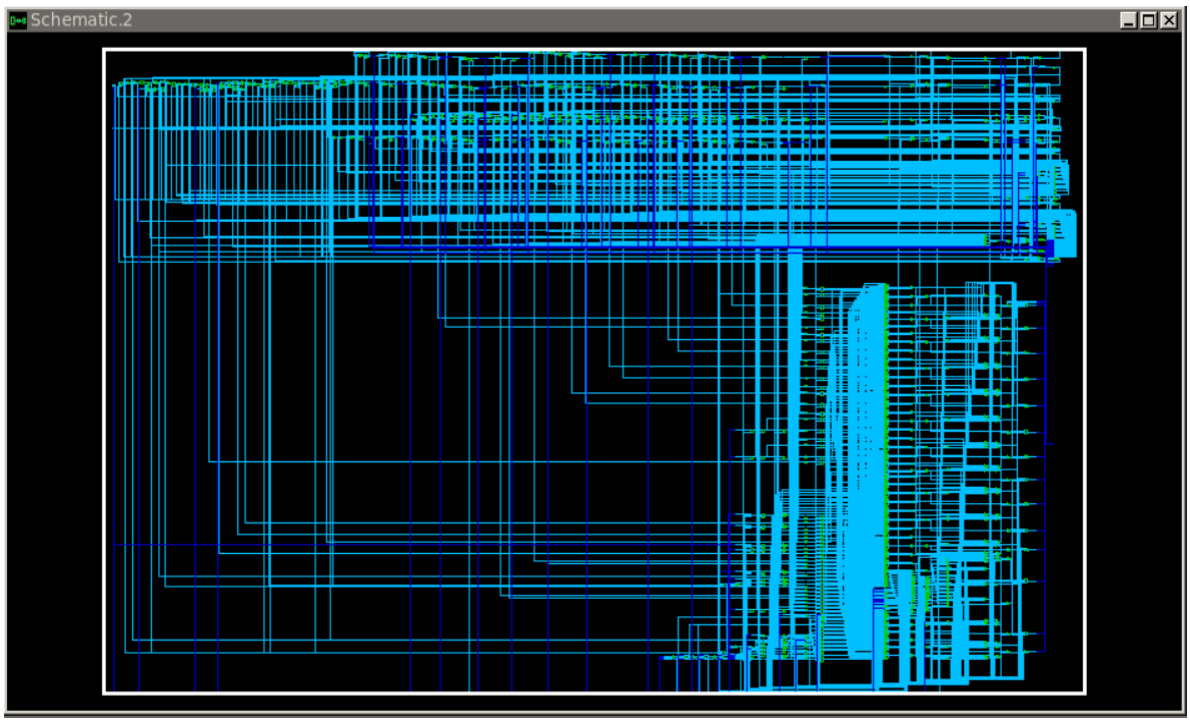

Figure 3: Design Schematic

```
design_vision> report_area

****************************************
Report : area
Design : MyDesign
Version: P-2019.03-SP1
Date   : Sun Nov 14 18:40:01 2021
****************************************

Library(s) Used:

    NangateOpenCellLibrary_PDKv1_2_v2008_10_slow_nldm (File: /afs/eos.ncsu.edu/lockers/research/ece/wdavis/tech/nangate/NangateOpenCellLibrary_PDKv1_2_v2008_1
0/liberty/520/NangateOpenCellLibrary_PDKv1_2_v2008_10_slow_nldm.db)

Number of ports:                         89
Number of nets:                        1095
Number of cells:                        923
Number of combinational cells:          735
Number of sequential cells:             186
Number of macros/black boxes:             0
Number of buf/inv:                       75
Number of references:                    30

Combinational area:            1009.470012
Buf/Inv area:                    42.028000
Noncombinational area:          893.760019
Macro/Black Box area:             0.000000
Net Interconnect area:      undefined  (No wire load specified)

Total cell area:               1903.230031
Total area:                 undefined
1
design_vision>
```

Figure 4: Area of design

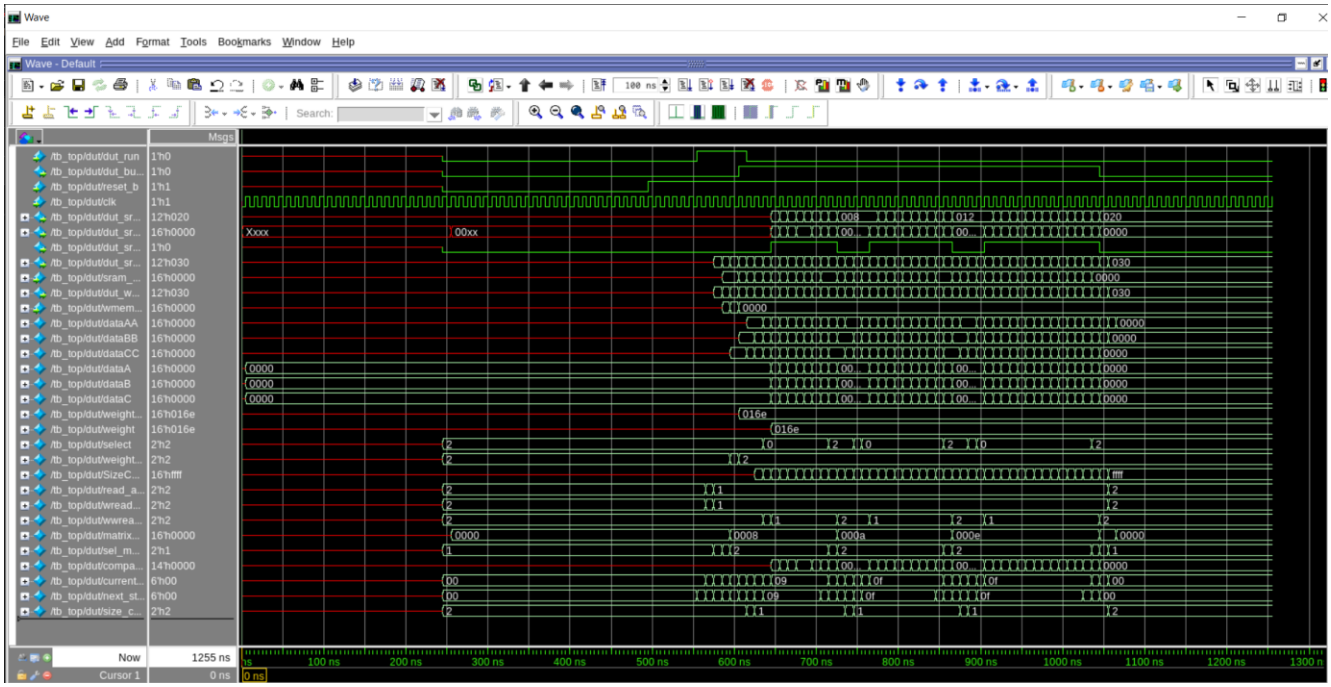

Figure 5: Waveform of simulation