Final Project Report First Page. Must match this format (Title)

Name: Vaidhya Subramanian Sridharan
Unityid: vsridha2
StudentID:. 200420709

Delay (ns to run provided provided example) : Clock period:2.5 # cycles":59	Logic Area: 2063.36 (um^2) Memory: N/A	1/(delay.area) ($ns^{-1}.um^{-2}$) $3.28*10^{-6}$
Delay (TA provided example. TA to complete)		1/(delay.area) (TA)

Abstract
Abstract should briefly summarize that the hardware function is (remember a future employer might be reading this), what your approach was, and the main results achieved.

The hardware implements a single stage of an all-binary convolutional neural network. All weights and input data are binary. They take on values of -1 and 1 which are represented by 0 and 1 respectively. The inputs come from an input SRAM, the weights come from a weight SRAM and the outputs are written to output SRAM memory. XNOR operation is used to compute the individual product, leveraging the advantage of the binary nature of the weights and the data. The developed design takes 59 cycles for execution. The timing was closed with a clock period of 2.5ns after synthesis. A total area of 2071um^2 was used for this design.

1. Introduction

The hardware design implements the convolution stage of single stage all binary convolution neural network for three input configurations: 16 *16, 10*10 and 12*12 for convolution with a 3*3 weight matrix producing a result of 14*14, 8*8 and 10*10 respectively.

The design is divided into separate modules for program counter, data path and controller. The program counter module updates the program counter to read the next address from the input SRAM. It is flushed when either reset is pressed, or we reach the end of input file. The data path reads the value of the inputs and weights from the respective SRAM files and performs the convolution operation and stores the result in the output SRAM. The controller reads the status signals from the data path and give relevant control signals back to the data path to produce the intended sequence of outputs.

The design goal was to optimize the performance metric which is defined as the reciprocal of product of area, compute cycles and clock period.

The design computes the convolution in 59 cycles with a clock period of 2.5ns after synthesis. The design occupies an area of 2071um^2.

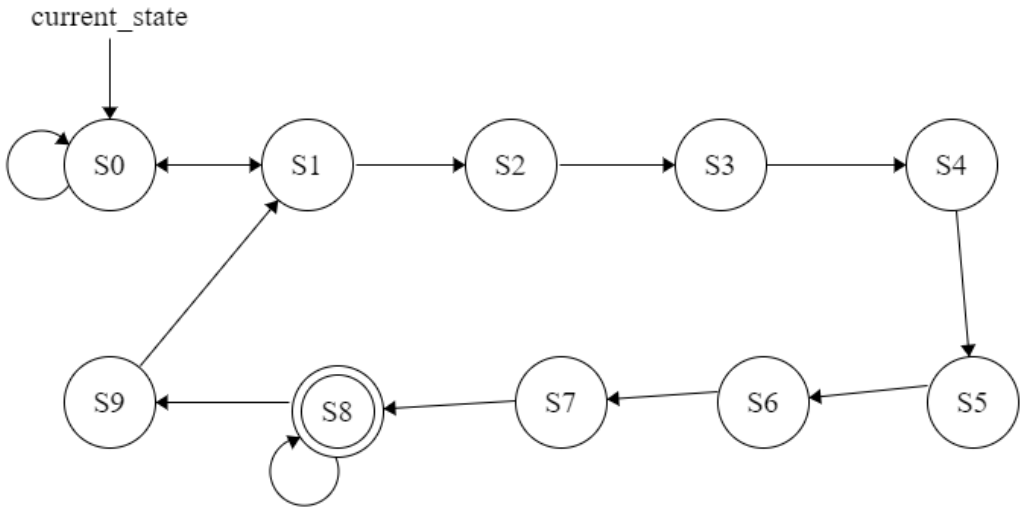
The microarchitecture of the project, the interface specification, technical implementation, verification strategy, results and conclusions are presented below.

2. Micro-Architecture

Controller:

1. When active low reset signal is asserted, the design enters S0 state where it waits for dut_run to be asserted and goes to S1 state.
2. In State S1, increment PC and weight_PC. If we reach the end of file, i.e. data read is FF, go to state S0 else we go to the next state S2.
3. In state S2, we receive the weight dimension and store it in a register. Next state is S3.
4. In state S3, we receive the data dimension and the weight data and store it in their respective registers. The controller goes to S4 state.
5. In state S4, we receive the data dimension and store in the register. This value keeps getting decremented and as we reach zero, we know that a set of inputs have completed. The controller goes to S5 state.
6. In state S5, S6, S7, we get the first 3 rows, Row1, Row 2 and Row 3 respectively from the input SRAM. None of the select lines are changed. The controller goes to S8 state.
7. In state S8, we get the first output to be written to SRAM as the data path completes 3 convolutions. The controller stays in this state until the input size reg decreased to zero. Goes to S9 when it is zero.

8. In state S9, we setup the data path for the next set of inputs and the controller goes to S1 state.

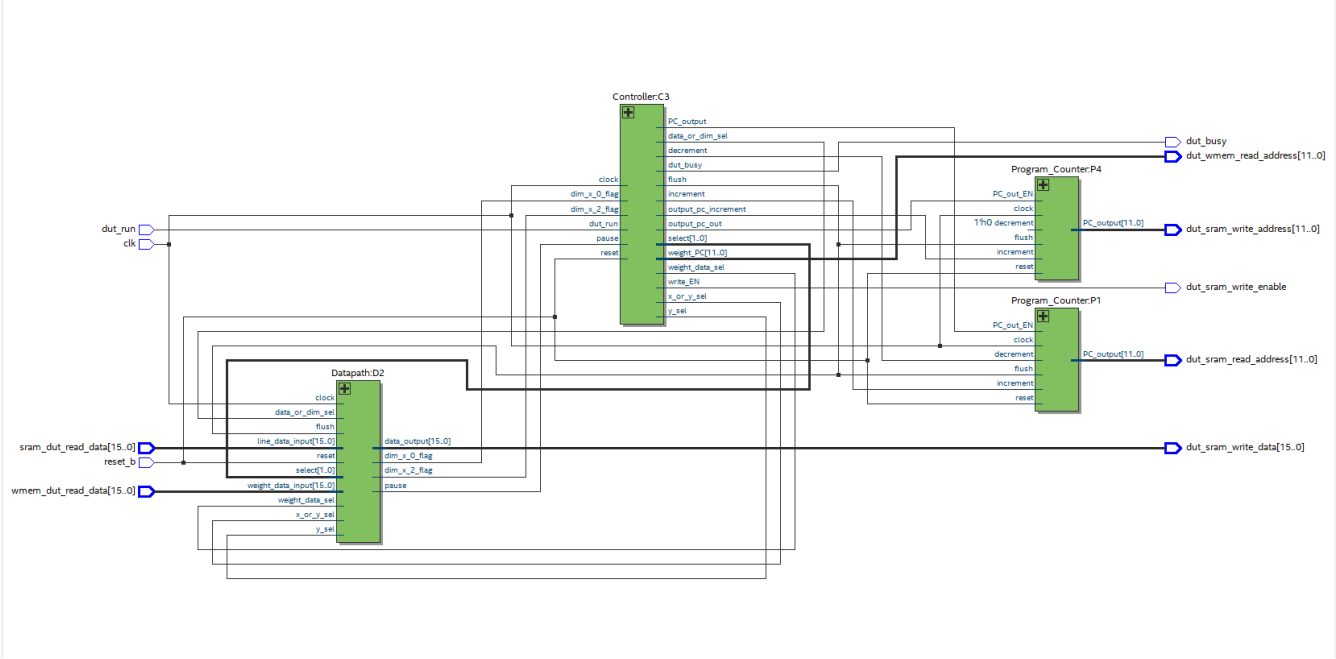


Datapath:
Datapath module:

- In this module we fetch data from weight and input matrices
- The inputs are 16 bit weight and input data from the SRAMs and the outputs from the module are 16 bit output to be written to the output SRAM, flags to check if data dimension is approaching zero and a pause signal which will flush all the inputs for the next set of data inputs.
- The convolution is performed here and a combination of counter and look up table logic is used to determine whether one or zero is dominant in the convoluted result.
- Here we also keep track of the matrix dimension and decrement it accordingly to identify the end of an input stream.

Program_Counter module:

- We use two instances of the program counter module.
- The first instance regulates when we read from the SRAM input and the second instance writes to the output SRAM when the write_enable signal is high.



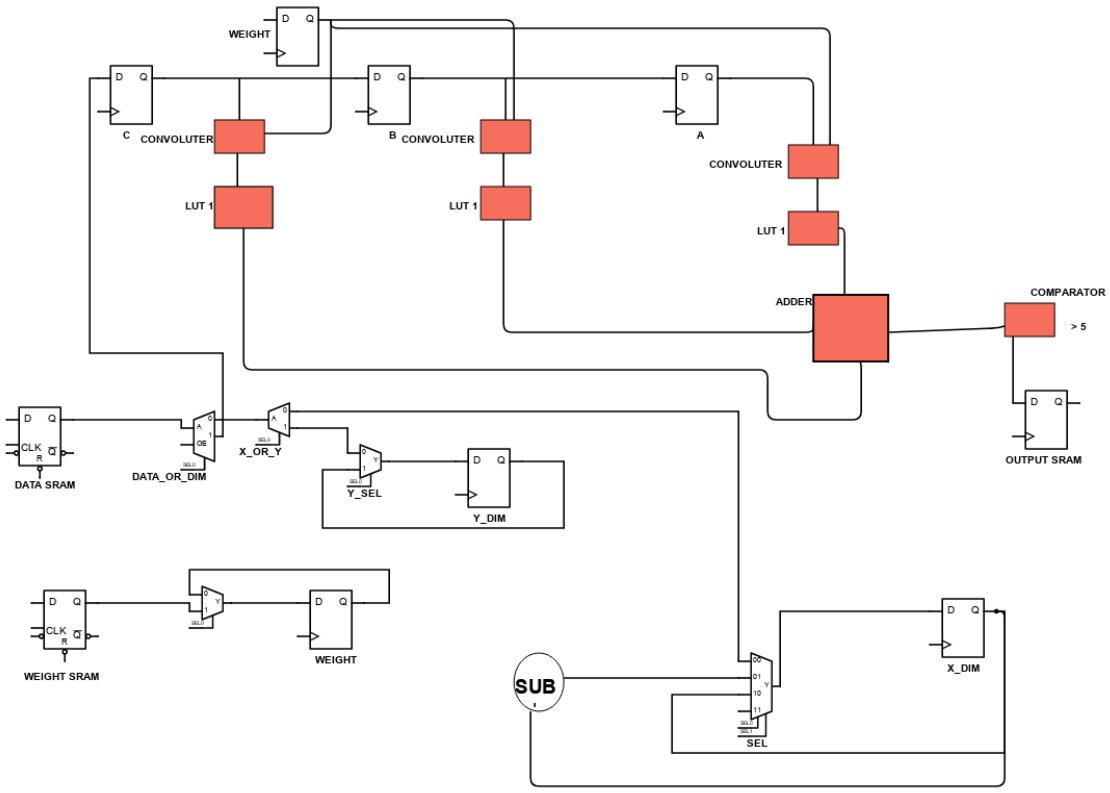
3. Interface Specification

A detailed list of all interface signals and their functions are listed in the table below:

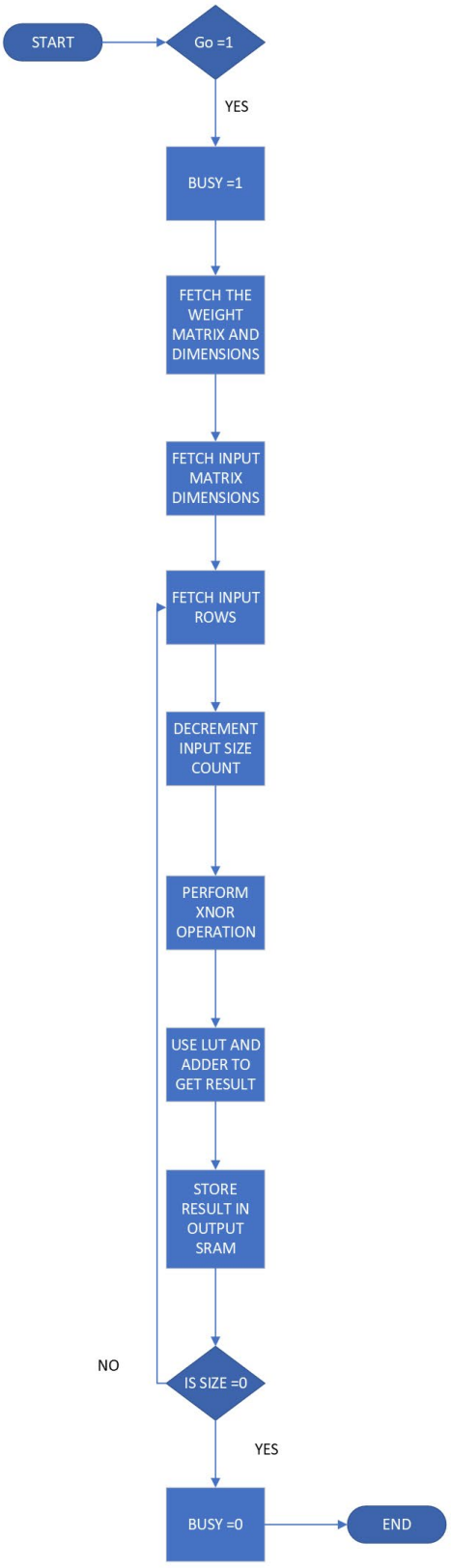
Direction	Type	Name	Comments
input	wire	dut_run	The test bench will set run signal high after all data has been loaded
input	wire	clk	System clock generated in tb_top
input	wire	reset_b	Active low reset signal - clears the machine state
output	reg	dut_busy	The test bench will halt when busy is high, waiting for the computation
output	reg	dut_sram_write_address	Write address to output SRAM
output	reg	dut_sram_write_data	Write data to output SRAM
output	reg	dut_sram_write_enable	Write enable signal to output SRAM
output	reg	dut_sram_read_address	Read address to input SRAM
input	wire	sram_dut_read_data	Read data from input SRAM
output	reg	dut_wmem_read_address	Read address to weight SRAM
input	wire	wmem_dut_read_data	Read data from weight SRAM

4. Technical Implementation

Block Diagram



Algorithm:

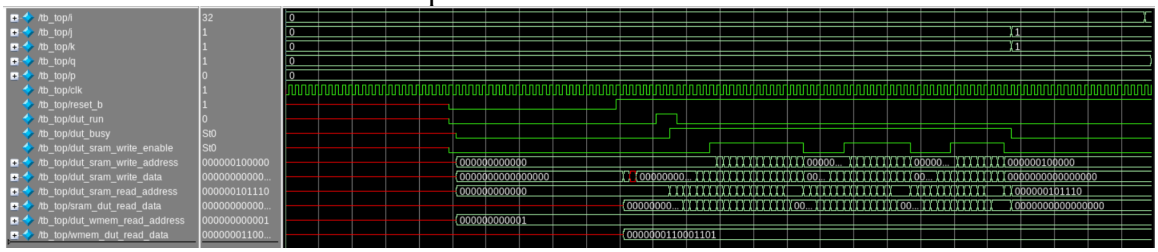

START
Go =1
YES
BUSY =1
FETCH THE WEIGHT MATRIX AND DIMENSIONS
FETCH INPUT MATRIX DIMENSIONS
FETCH INPUT ROWS
DECREMENT INPUT SIZE COUNT
PERFORM XNOR OPERATION
USE LUT AND ADDER TO GET RESULT
STORE RESULT IN OUTPUT SRAM
NO
IS SIZE =0
YES
BUSY =0
END

5. Verification

A self-checking code was added to the testbench to verify the correctness of the convolution design. The testbench was loaded with expected output which was then compared against the produced results. The design was tested for one round of three input square matrices with dimension values from 10, 12 and 16. It was observed that the design was able to match the required outputs for all the inputs provided including the corner case of three matrices.

6. Results Achieved

Modelsim waveform and transcript :




```
# ------------------------------Round 1 Your report-------------------------------
#
# Check 1 : Correct g results = 32/32
# computeCycle=59
# --------------------------------------------------------------------------------
#
# ** Note: $finish    : /afs/unity.ncsu.edu/users/v/vsridha2/564/Project/MyDesign_tb.sv(220)
#    Time: 2215 ns  Iteration: 1  Instance: /tb_top
# 1
# Break at /afs/unity.ncsu.edu/users/v/vsridha2/564/Project/MyDesign_tb.sv line 220

/SIM 4>
```

Area Report:

```
****************************************
Report : area
Design : MyDesign
Version: P-2019.03-SP1
Date   : Wed Nov 17 13:22:15 2021
****************************************

Library(s) Used:

    NangateOpenCellLibrary_PDKv1_2_v2008_10_slow_nldm (File: /afs/eos.n

Number of ports:                           89
Number of nets:                          1437
Number of cells:                         1338
Number of combinational cells:           1137
Number of sequential cells:               188
Number of macros/black boxes:               0
Number of buf/inv:                        248
Number of references:                      50

Combinational area:               1122.519983
Buf/Inv area:                      136.458001
Noncombinational area:             940.841996
Macro/Black Box area:                0.000000
Net Interconnect area:      undefined  (No wire load specified)

Total cell area:                  2063.361978
Total area:                 undefined
```

Set up Timing Report:

```
****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : MyDesign
Version: P-2019.03-SP1
Date   : Wed Nov 17 12:43:24 2021
****************************************

Operating Conditions: slow   Library: NangateOpenCellLibrary_PDKv1_2_v2008_10_slow_nldm
Wire Load Model Mode: top

  Startpoint: clk_r_REG65_S5
              (rising edge-triggered flip-flop clocked by clk)
  Endpoint: clk_r_REG24_S4
            (rising edge-triggered flip-flop clocked by clk)
  Path Group: clk
  Path Type: max

  Point                                   Incr       Path
  -----------------------------------------------------------
  clock clk (rise edge)                   0.0000     0.0000
  clock network delay (ideal)             0.0000     0.0000
  clk_r_REG65_S5/CK (DFFR_X2)             0.0000     0.0000 r
  clk_r_REG65_S5/Q (DFFR_X2)              0.7405     0.7405 f
  U886/ZN (XNOR2_X2)                      0.2676     1.0081 f
  U1151/ZN (OAI21_X2)                     0.1980     1.2061 r
  U1731/ZN (NAND2_X2)                     0.0990     1.3050 f
  U799/ZN (INV_X4)                        0.0832     1.3882 r
  U897/ZN (XNOR2_X2)                      0.2659     1.6541 r
  U1030/ZN (XNOR2_X1)                     0.3335     1.9876 r
  U885/ZN (NAND2_X2)                      0.0993     2.0869 f
  U684/ZN (NAND2_X2)                      0.1153     2.2022 r
  clk_r_REG24_S4/D (DFFR_X2)              0.0000     2.2022 r
  data arrival time                                  2.2022

  clock clk (rise edge)                   2.5000     2.5000
  clock network delay (ideal)             0.0000     2.5000
  clock uncertainty                      -0.0500     2.4500
  clk_r_REG24_S4/CK (DFFR_X2)             0.0000     2.4500 r
  library setup time                     -0.2477     2.2023
  data required time                                 2.2023
  -----------------------------------------------------------
  data required time                                 2.2023
  data arrival time                                 -2.2022
  -----------------------------------------------------------
  slack (MET)                                        0.0001


1
```

Power report:

```
****************************************
Report : power
        -analysis_effort low
Design : MyDesign
Version: P-2019.03-SP1
Date   : Wed Nov 17 12:31:53 2021
****************************************


Library(s) Used:

    NangateOpenCellLibrary_PDKv1_2_v2008_10_slow_nldm (File: /afs/eos.ncsu.edu/lockers/research/ece/wdavis/t


Operating Conditions: slow   Library: NangateOpenCellLibrary_PDKv1_2_v2008_10_slow_nldm
Wire Load Model Mode: top


Global Operating Voltage = 0.95
Power-specific unit information :
    Voltage Units = 1V
    Capacitance Units = 1.000000pf
    Time Units = 1ns
    Dynamic Power Units = 1mW    (derived from V,C,T units)
    Leakage Power Units = 1pW


  Cell Internal Power  = 367.1021 uW   (92%)
  Net Switching Power  =  33.3788 uW    (8%)
                         ---------
Total Dynamic Power    = 400.4810 uW  (100%)

Cell Leakage Power     =   8.9726 uW


                 Internal         Switching           Leakage            Total
Power Group      Power            Power               Power              Power   (   %    )  Attrs
--------------------------------------------------------------------------------------------------
io_pad             0.0000            0.0000            0.0000            0.0000  (   0.00%)
memory             0.0000            0.0000            0.0000            0.0000  (   0.00%)
black_box          0.0000            0.0000            0.0000            0.0000  (   0.00%)
clock_network      0.0000            0.0000            0.0000            0.0000  (   0.00%)
register           0.3310        5.5360e-03        2.3345e+06            0.3389  (  82.76%)
sequential         0.0000            0.0000            0.0000            0.0000  (   0.00%)
combinational  3.6121e-02        2.7843e-02        6.6380e+06        7.0602e-02  (  17.24%)
--------------------------------------------------------------------------------------------------
Total              0.3671 mW     3.3379e-02 mW     8.9726e+06 pW         0.4095 mW
```

Hold timing report:

```
****************************************
Report : timing
        -path full
        -delay min
        -max_paths 1
Design : MyDesign
Version: P-2019.03-SP1
Date   : Wed Nov 17 12:43:26 2021
****************************************

Operating Conditions: fast   Library: NangateOpenCellLibrary_PDKv1_2_v2008_10_fast_nldm
Wire Load Model Mode: top

  Startpoint: clk_r_REG26_S3
              (rising edge-triggered flip-flop clocked by clk)
  Endpoint: clk_r_REG27_S4
            (rising edge-triggered flip-flop clocked by clk)
  Path Group: clk
  Path Type: min

  Point                                   Incr       Path
  -----------------------------------------------------------
  clock clk (rise edge)                 0.0000     0.0000
  clock network delay (ideal)           0.0000     0.0000
  clk_r_REG26_S3/CK (DFFR_X1)           0.0000     0.0000 r
  clk_r_REG26_S3/Q (DFFR_X1)            0.0663     0.0663 r
  U1579/ZN (NAND2_X1)                   0.0186     0.0849 f
  clk_r_REG27_S4/D (DFFR_X1)            0.0000     0.0849 f
  data arrival time                                0.0849

  clock clk (rise edge)                 0.0000     0.0000
  clock network delay (ideal)           0.0000     0.0000
  clock uncertainty                     0.0500     0.0500
  clk_r_REG27_S4/CK (DFFR_X1)           0.0000     0.0500 r
  library hold time                     0.0021     0.0521
  data required time                               0.0521
  -----------------------------------------------------------
  data required time                               0.0521
  data arrival time                               -0.0849
  -----------------------------------------------------------
  slack (MET)                                      0.0328
```

7. Conclusions

The design goal was to implement a single layer of a Binary convolution Neural Network in RTL, synthesize the design and then optimize the design for the highest performance metric which is defined as the reciprocal of product of area, clock period and number of clock cycles. Thus, a design implementing the convolutional neural network layer was successfully developed in RTL using modelsim2019.2.

The design was verified to work by checking for corner cases of the design specifications as well as for nominal specifications by implementing self-checking code in Verilog. The developed RTL was then synthesized using synopsys2019.

A performance metric of **3.288×10^{-6}** $ns^{-1}.um^{-2}$ could be achieved with the final design.