# Project Report – ECE 564 Fall 2025

### 1024×1024 CNN Pipeline with 4×4 Convolution, Leaky-ReLU, and 2×2 Average Pooling

**Author:** Shivanesh Gunasekaran
**Unity ID:** sgunase2
**Clock Period:** 3.8 ns

# 1. Introduction

This project implements a complete hardware pipeline for a simplified Convolutional Neural Network (CNN) using SystemVerilog RTL. The design reads a 1024×1024 signed 8-bit input image from DRAM, performs:

1. **4×4 convolution**
2. **Leaky-ReLU activation**
3. **2×2 average pooling**

and finally writes the processed output back to DRAM.

The entire system follows the exact timing and memory interface protocol provided in the class project specification and uses a **pipelined architecture** capable of sustaining computation at the DRAM read throughput.

The design uses the optional **dual-port SRAM** as an internal scratchpad to buffer input rows and feed the convolution unit without stalling.

# 2. High-Level System Overview

The design consists of four major subsystems:

### 1. DRAM Input Loader FSM

Fetches kernel (16 bytes) and input pixels (1024×1024×8-bit) from DRAM following burst-read protocol.

### 2. SRAM Data Re-organization Engine

Stores the input matrix into SRAM row buffers to enable fast sliding-window extraction for convolution.

### 3. Convolution + ReLU + Pooling Engine (Main Datapath)

For every valid 4×4 window:

- Performs 16 parallel multiplications
- Performs 15 additions across three pipeline stages
- Applies Leaky-ReLU
- Performs 2×2 average pooling (with padding logic)

Implemented as a **deeply pipelined compute engine**.

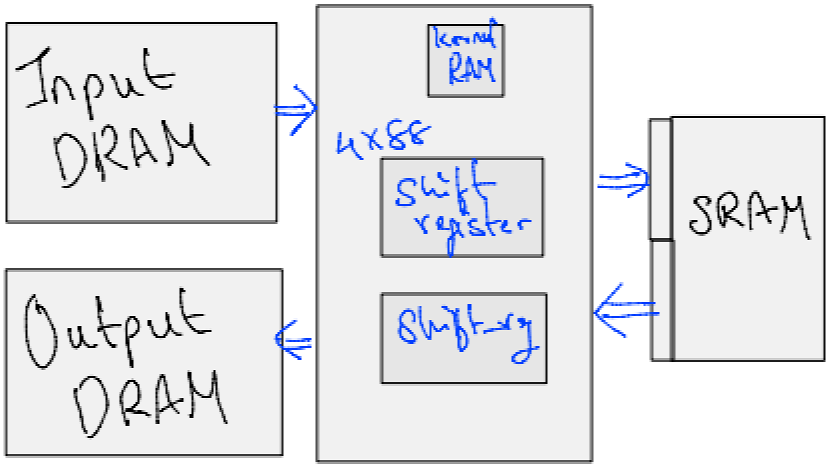
### 4. DRAM Output Writer FSM

Packs 8 processed output pixels into a burst and writes to memory with correct padding.

# 3. Design Architecture

## 3.1 Top-Level Block Diagram

# Top level Design

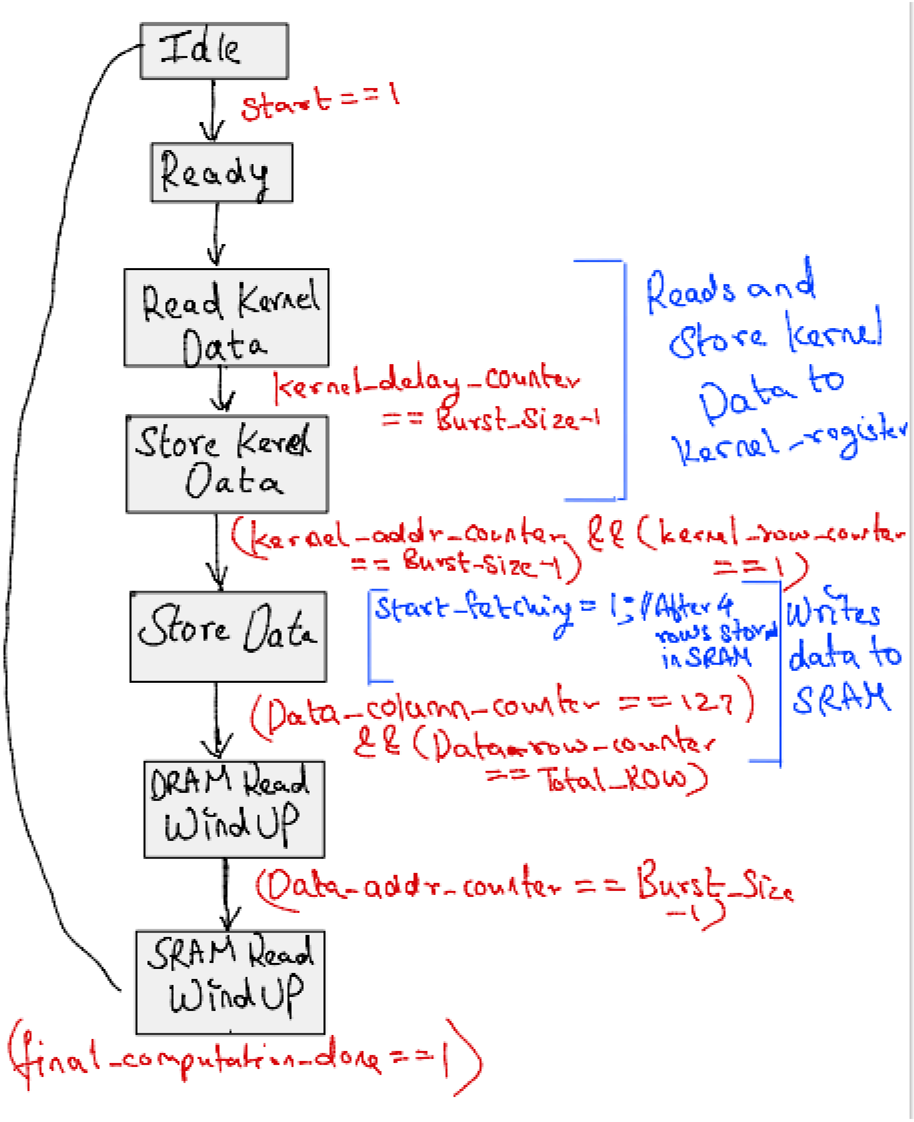


**Note: Shift registers are used to compute the convolution of two rows at a time**

# 4. Controller Architecture (FSMs)

I use **four interacting FSMs**, each corresponding to a different phase.

## 4.1 Input FSM (i_state)


Idle
Start == 1
Ready
Read Kernel Data
kernel_delay_counter == Burst_Size-1
Store Kerel Data
Reads and Store kernel Data to Kernel_register
(kernel_addr_counter == Burst_Size-1) && (kernel_row_counter == 1)
Store Data
Start_fetching = 1; // After 4 rows stord in SRAM
Writes data to SRAM
(Data_column_counter == 127) && (Data_row_counter == Total_ROW)
DRAM Read WindUP
(Data_addr_counter == Burst_Size -1)
SRAM Read WindUP
(final_computation_done == 1)

Controls DRAM reads:

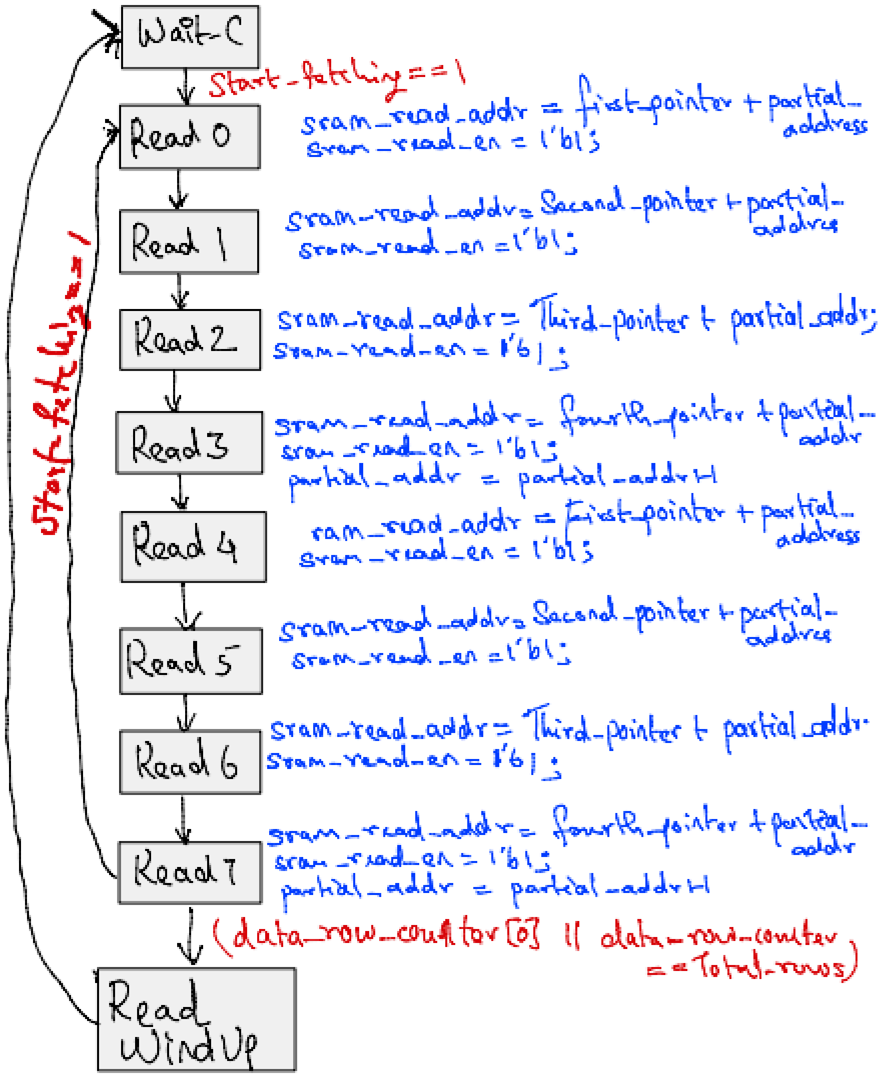
| State | Description |
|---|---|
| **IDLE** | ready=1, waiting for start |

| | |
|---|---|
| **READY** | Issues first kernel read |
| **KERNEL_READ_DATA** | Fetch 4×4 kernel |
| **KERNEL_STORE_DATA** | Write kernel bytes into kernel RAM |
| **DATA_STORE** | Read all image rows; write into SRAM |
| **DRAM_READ_WINDUP** | Flush final row of input reads |
| **SRAM_READ_WINDUP** | Wait for compute & output to finish |

---

## 4.2 SRAM Read FSM (c_state)

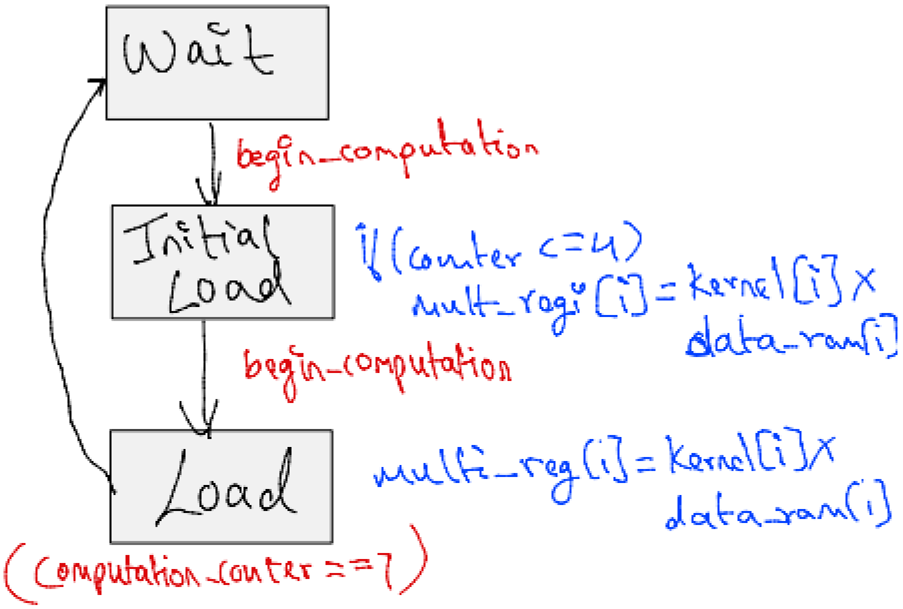
Cycles through 8 SRAM reads to fetch two consecutive 4-pixel segments for convolution.


Wait-C
Start-fetching==1
Read 0
sram_read_addr = first_pointer + partial_address
sram_read_en = 1'b1;
Read 1
sram_read_addr= Second_pointer + partial_address
sram_read_en = 1'b1;
Read 2
sram_read_addr= Third_pointer + partial_addr;
sram_read_en = 1'b1;
Read 3
sram_read_addr= fourth_pointer + partial_addr
sram_read_en = 1'b1;
partial_addr = partial_addr+1
Read 4
ram_read_addr = first_pointer + partial_address
sram_read_en = 1'b1;
Read 5
sram_read_addr= Second_pointer + partial_address
sram_read_en = 1'b1;
Read 6
sram_read_addr= Third_pointer + partial_addr.
sram_read_en = 1'b1;
Read 7
sram_read_addr= fourth_pointer + partial_addr
sram_read_en = 1'b1;
partial_addr = partial_addr+1
(data_row_counter[0] || data_row_counter ==Total_rows)
Read WindUp
Start_fetching==1

| State | Description |
|---|---|
| WAIT | idle |
| SRAM_READ0-7 | fetch 8 required segments per window |
| WINDUP_READ | stall alignment |
| LAST_READ_WINDUP | final cleanup |

The FSM ensures continuous data delivery to the compute engine.

## 4.3 Convolution/Pooling FSM (m_state)

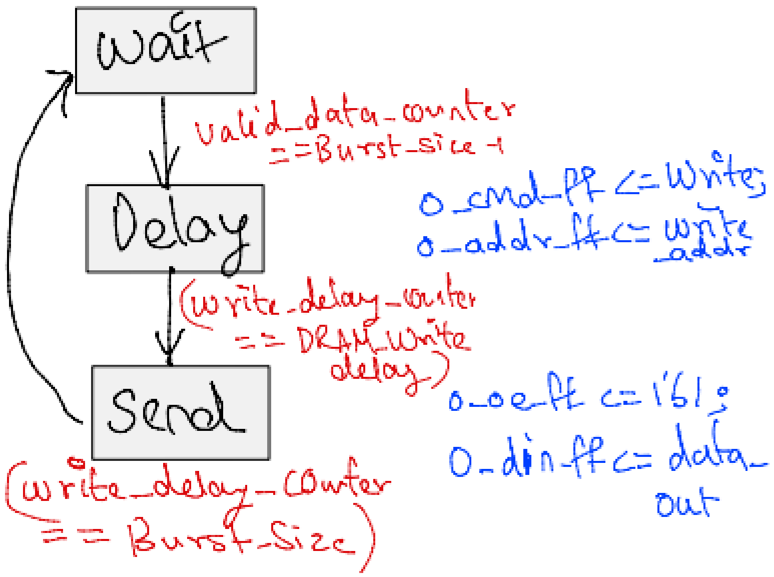


| State | Description |
|---|---|
| WAIT_M | idle until read buffer is ready |
| INITIAL_LOAD | first compute cycle for new row |
| LOAD0 | pipeline processing of sliding windows |

Architecture uses:

- `multi_result1/2[0:15]` (parallel multipliers)
- 3-level adder tree (`partial_sum1/2/3`)
- Post-adder adjustment for ReLU
- Shift-based division for pooling

### 4.4 Output FSM (o_state)


Wait
valid_data_counter ==Burst_size -1
Delay
o_cmd_ff <= write;
o_addr_ff <= write_addr
(write_delay_counter == DRAM_write delay)
Send
o_oe_ff <= 1'b1;
o_din_ff <= data_out
(write_delay_Counter == Burst_Size)

| State | Description |
|---|---|
| WAIT_O | waiting for 8 pooled output values |
| WAIT_DELAY | DRAM write delay alignment |
| SEND_DATA | sequentially send 8 bytes to DRAM |

This FSM ensures correct padding and write-back alignment.

---

# 5. Datapath Description

### 5.1 Kernel Storage

The kernel is read during the first 16 DRAM bytes and stored into:

```
logic signed [7:0] kernel_ram [0:15];
```

Addressing matches the fixed pattern described in the spec.

---

## 5.2 SRAM Buffer Organization

SRAM stores 1024-byte rows in two 32-bit words:

```
Row i → SRAM entries:
    [i*256 + col] contains 4 consecutive pixels
```

Two buffers (`main_sram_buff_0` and `_1`) hold sliding window data.

---

## 5.3 Convolution Engine

### Parallel Multiply Stage

16 independent multipliers:

```
multi_result1[i] = kernel_ram[i] * source_data_1[i];
multi_result2[i] = kernel_ram[i] * source_data_2[i];
```

### Adder Tree (Pipelined)

The design performs:

1. Level-1: 8 adds
2. Level-2: 4 adds
3. Level-3: 2 adds
4. Final sum: 1 add

This produces **20-bit convolution output**.

---

## 5.4 Leaky-ReLU

Specification:

- x > 0 → x

- −4 < x ≤ 0 → 0
- x ≤ −4 → arithmetic shift right by 2

Exactly implemented in:

```
if(convo_result1_temp > 0) …
else if(convo_result1_temp <= 0 && convo_result1_temp > -4)
convo_result1 <= 0;
else convo_result1 <= ((convo_result1_temp+3) >>> 2);
```

---

## 5.5 2×2 Average Pooling

Pooling sums two convolution outputs across two rows:

```
partial_avrg accumulates 2x2 region
final_avrg <= partial_avrg >>> 2;   // custom truncation rule
```

This matches the project's modified integer-pooling definition.

---

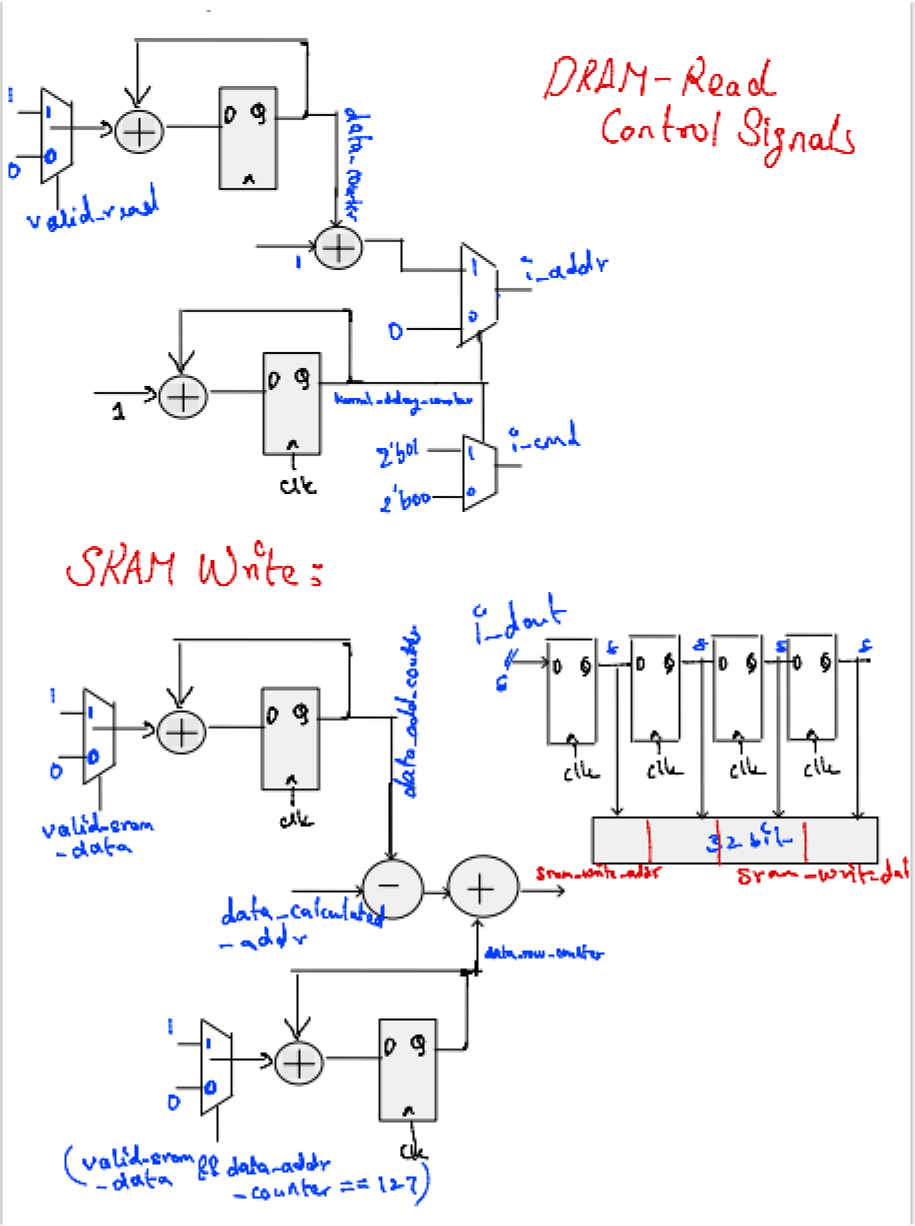
## 5.6 Output Clamping

Final values limited to 8-bit signed:

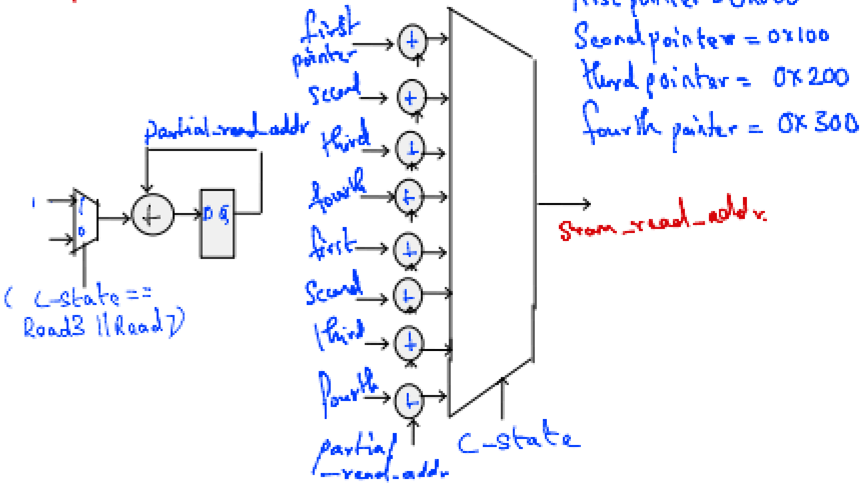
```
> +127  → 127
< -128 → -128
else: keep value
```

---

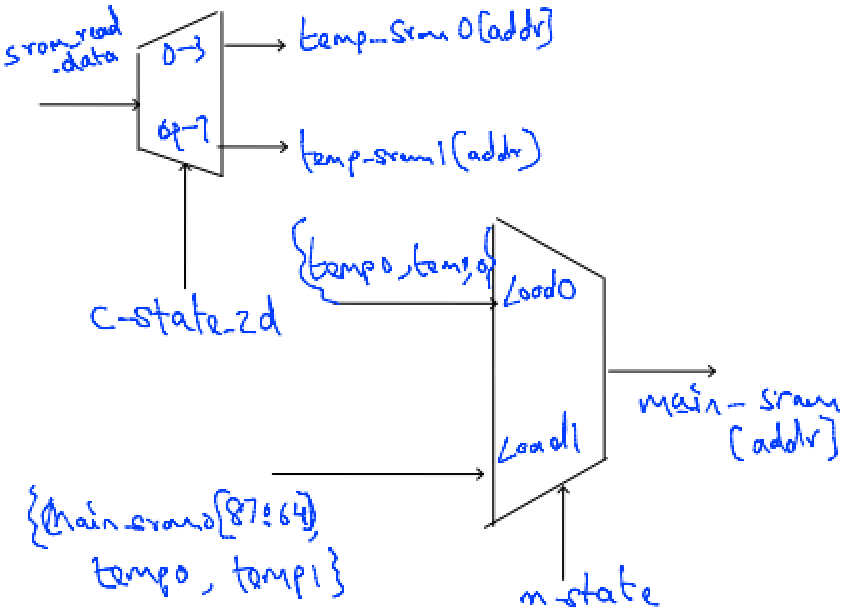
# 6. Schematics

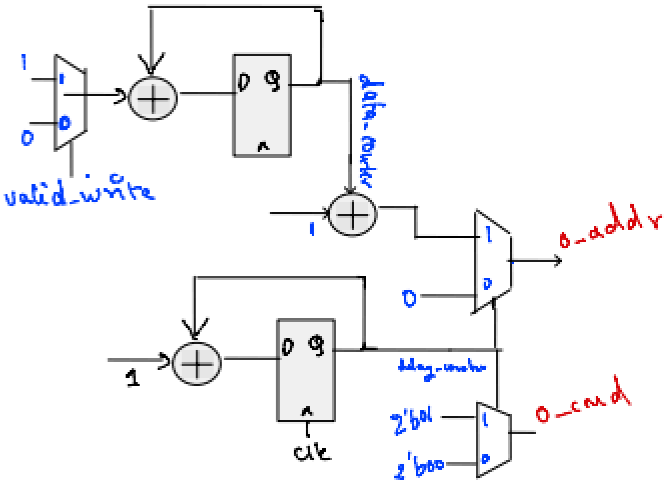


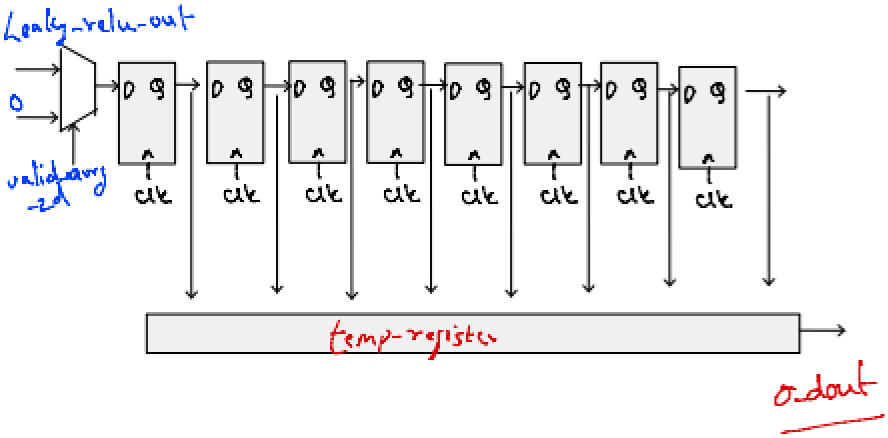
# SRAM Read

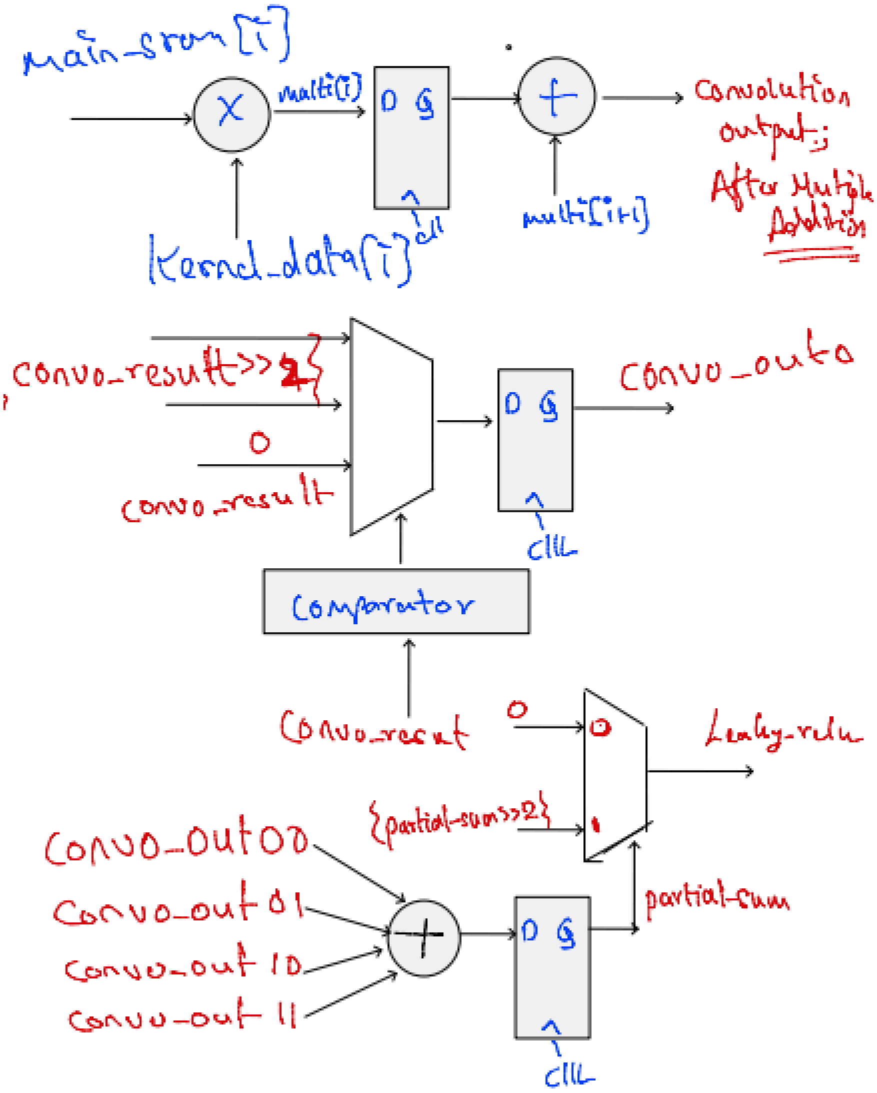


# Computation




1
0
1
0
valid_write
D Q
data_counter
1
1
0
o_addr
1
D Q
clk
delay_counter
2'b01
2'b00
1
0
o_cmd

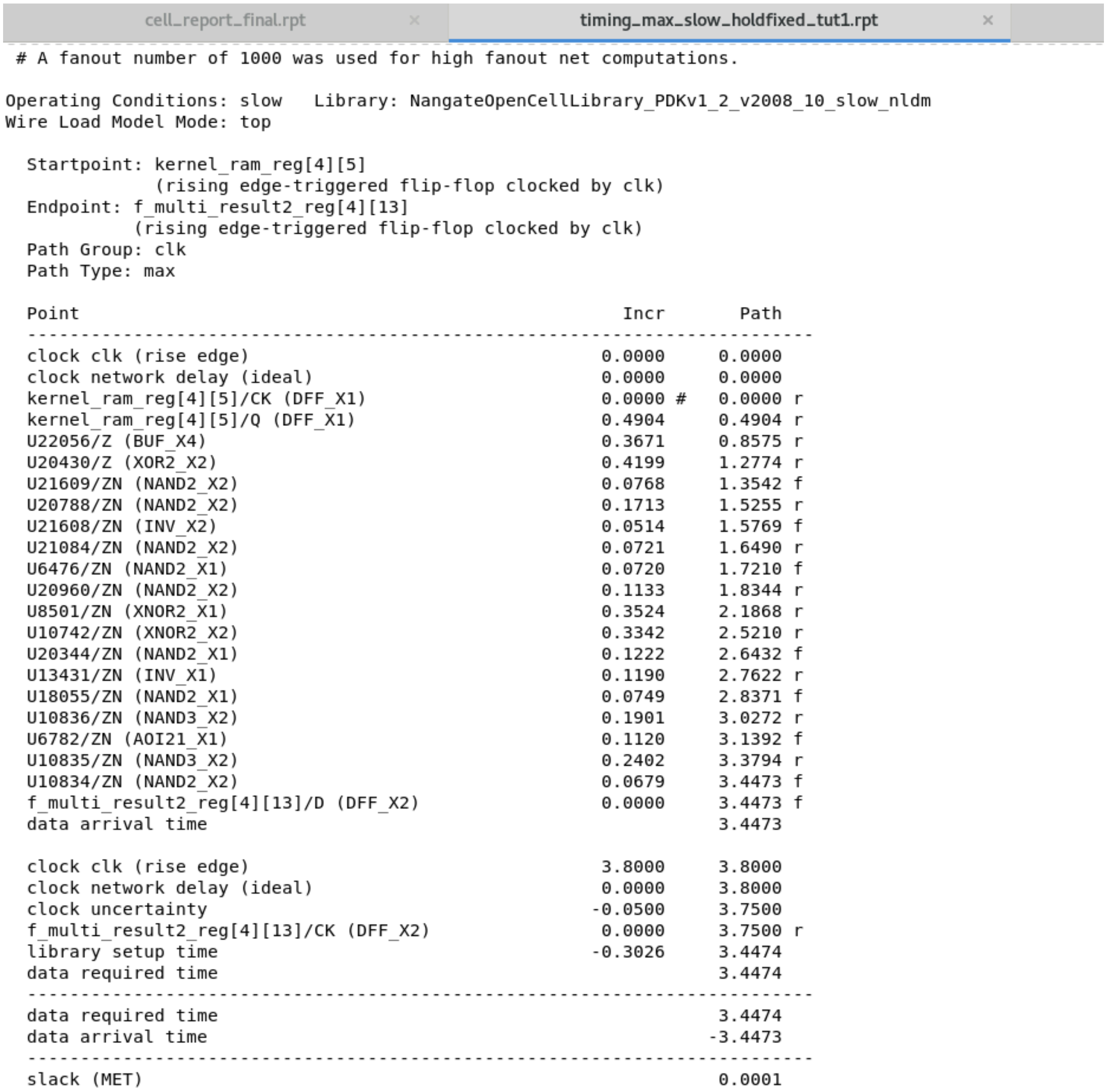

Leaky_relu_out
0
valid_avg
_2d
D Q
D Q
D Q
D Q
D Q
D Q
D Q
D Q
clk
clk
clk
clk
clk
clk
clk
clk
temp_register
o_dout



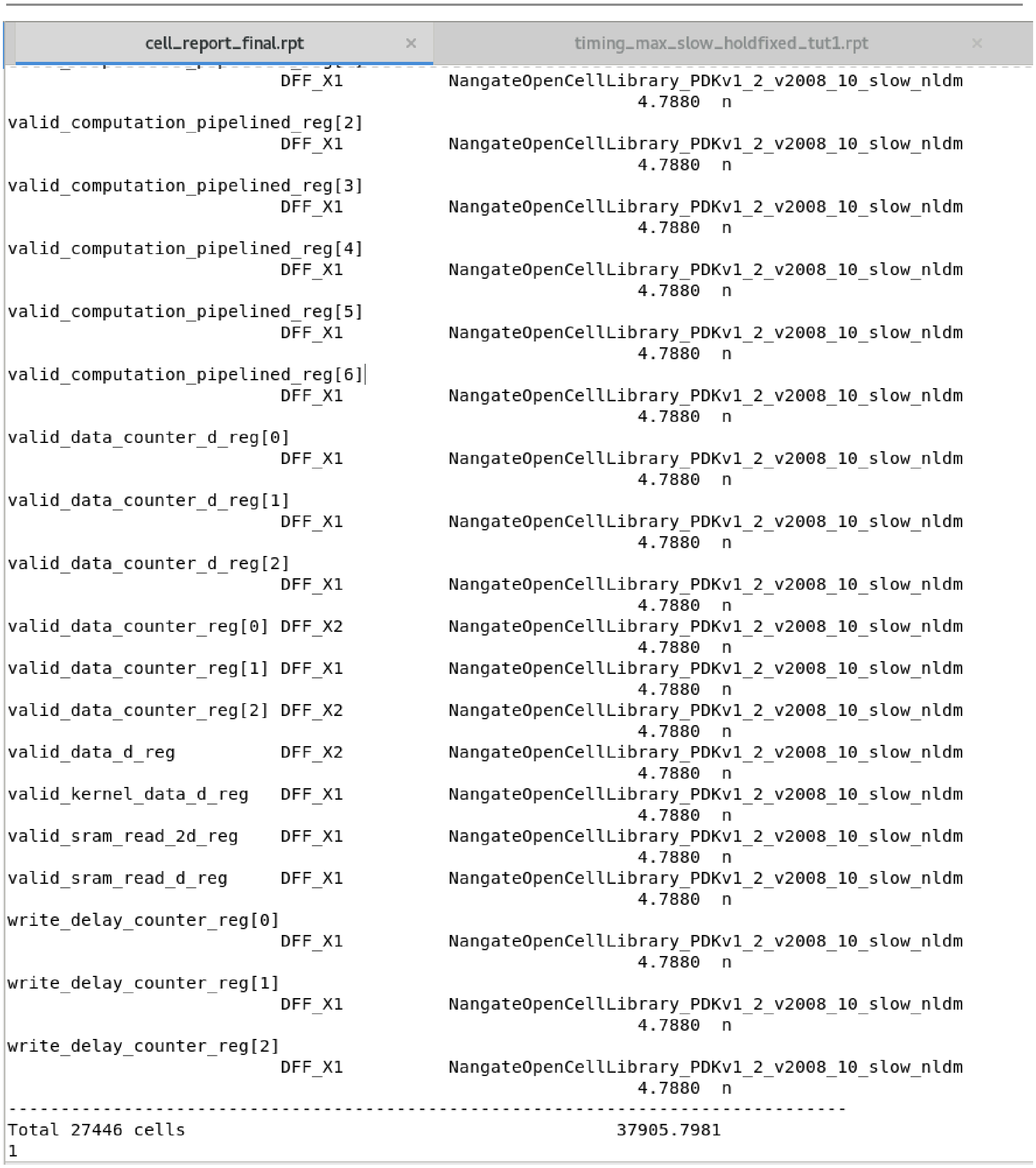
# 7. Performance Characteristics

| Metric | Value |
|---|---|
| Clock period | 3.8 ns |
| Total Simulation cycles | 12584124 cycles (According to printed results from modelsim)<br>Actual for one run : 1,048,677 |

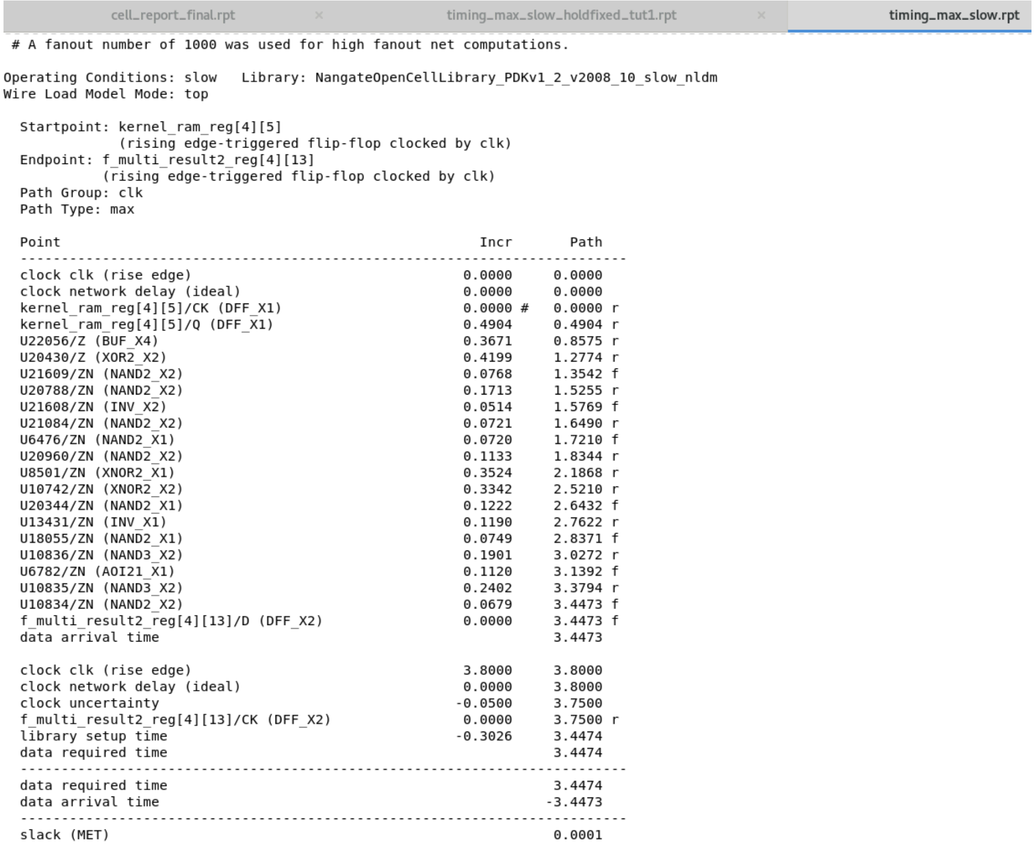
| | |
|---|---|
| Cycles for One testcase | 1,048,677 cycles or (2097354 as per results from Modelsim) |
| Cell area | 37905 |
| Worst-case setup slack | 0.0001 |
| Worst-case hold slack | 0.0217 |



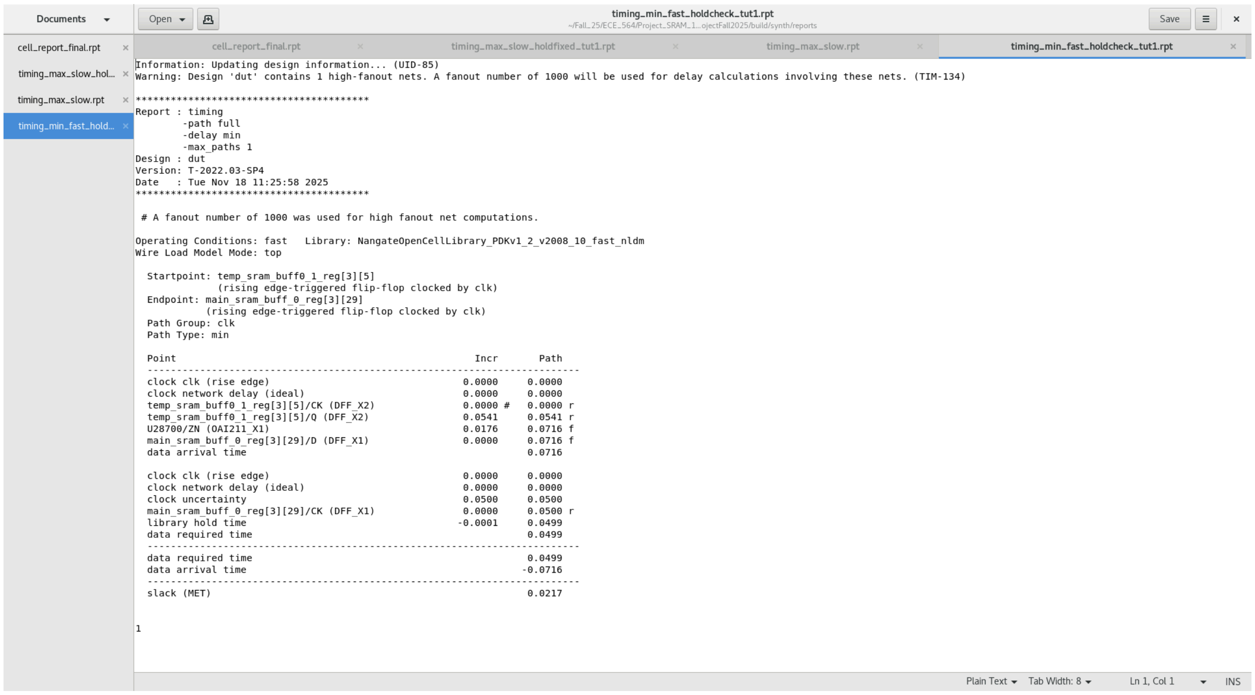
```
 # A fanout number of 1000 was used for high fanout net computations.

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

  Startpoint: kernel_ram_reg[4][5]
              (rising edge-triggered flip-flop clocked by clk)
  Endpoint: f_multi_result2_reg[4][13]
            (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
  kernel_ram_reg[4][5]/CK (DFF_X1)                      0.0000 #   0.0000 r
  kernel_ram_reg[4][5]/Q (DFF_X1)                       0.4904     0.4904 r
  U22056/Z (BUF_X4)                                     0.3671     0.8575 r
  U20430/Z (XOR2_X2)                                    0.4199     1.2774 r
  U21609/ZN (NAND2_X2)                                  0.0768     1.3542 f
  U20788/ZN (NAND2_X2)                                  0.1713     1.5255 r
  U21608/ZN (INV_X2)                                    0.0514     1.5769 f
  U21084/ZN (NAND2_X2)                                  0.0721     1.6490 r
  U6476/ZN (NAND2_X1)                                   0.0720     1.7210 f
  U20960/ZN (NAND2_X2)                                  0.1133     1.8344 r
  U8501/ZN (XNOR2_X1)                                   0.3524     2.1868 r
  U10742/ZN (XNOR2_X2)                                  0.3342     2.5210 r
  U20344/ZN (NAND2_X1)                                  0.1222     2.6432 f
  U13431/ZN (INV_X1)                                    0.1190     2.7622 r
  U18055/ZN (NAND2_X1)                                  0.0749     2.8371 f
  U10836/ZN (NAND3_X2)                                  0.1901     3.0272 r
  U6782/ZN (AOI21_X1)                                   0.1120     3.1392 f
  U10835/ZN (NAND3_X2)                                  0.2402     3.3794 r
  U10834/ZN (NAND2_X2)                                  0.0679     3.4473 f
  f_multi_result2_reg[4][13]/D (DFF_X2)                 0.0000     3.4473 f
  data arrival time                                                3.4473

  clock clk (rise edge)                                 3.8000     3.8000
  clock network delay (ideal)                           0.0000     3.8000
  clock uncertainty                                    -0.0500     3.7500
  f_multi_result2_reg[4][13]/CK (DFF_X2)                0.0000     3.7500 r
  library setup time                                   -0.3026     3.4474
  data required time                                               3.4474
  ---------------------------------------------------------------------------
  data required time                                               3.4474
  data arrival time                                               -3.4473
  ---------------------------------------------------------------------------
  slack (MET)                                                      0.0001
```



```
                          DFF_X1          NangateOpenCellLibrary_PDKv1_2_v2008_10_slow_nldm
                                                            4.7880  n
valid_computation_pipelined_reg[2]
                          DFF_X1          NangateOpenCellLibrary_PDKv1_2_v2008_10_slow_nldm
                                                            4.7880  n
valid_computation_pipelined_reg[3]
                          DFF_X1          NangateOpenCellLibrary_PDKv1_2_v2008_10_slow_nldm
                                                            4.7880  n
valid_computation_pipelined_reg[4]
                          DFF_X1          NangateOpenCellLibrary_PDKv1_2_v2008_10_slow_nldm
                                                            4.7880  n
valid_computation_pipelined_reg[5]
                          DFF_X1          NangateOpenCellLibrary_PDKv1_2_v2008_10_slow_nldm
                                                            4.7880  n
valid_computation_pipelined_reg[6]
                          DFF_X1          NangateOpenCellLibrary_PDKv1_2_v2008_10_slow_nldm
                                                            4.7880  n
valid_data_counter_d_reg[0]
                          DFF_X1          NangateOpenCellLibrary_PDKv1_2_v2008_10_slow_nldm
                                                            4.7880  n
valid_data_counter_d_reg[1]
                          DFF_X1          NangateOpenCellLibrary_PDKv1_2_v2008_10_slow_nldm
                                                            4.7880  n
valid_data_counter_d_reg[2]
                          DFF_X1          NangateOpenCellLibrary_PDKv1_2_v2008_10_slow_nldm
                                                            4.7880  n
valid_data_counter_reg[0] DFF_X2          NangateOpenCellLibrary_PDKv1_2_v2008_10_slow_nldm
                                                            4.7880  n
valid_data_counter_reg[1] DFF_X1          NangateOpenCellLibrary_PDKv1_2_v2008_10_slow_nldm
                                                            4.7880  n
valid_data_counter_reg[2] DFF_X2          NangateOpenCellLibrary_PDKv1_2_v2008_10_slow_nldm
                                                            4.7880  n
valid_data_d_reg          DFF_X2          NangateOpenCellLibrary_PDKv1_2_v2008_10_slow_nldm
                                                            4.7880  n
valid_kernel_data_d_reg   DFF_X1          NangateOpenCellLibrary_PDKv1_2_v2008_10_slow_nldm
                                                            4.7880  n
valid_sram_read_2d_reg    DFF_X1          NangateOpenCellLibrary_PDKv1_2_v2008_10_slow_nldm
                                                            4.7880  n
valid_sram_read_d_reg     DFF_X1          NangateOpenCellLibrary_PDKv1_2_v2008_10_slow_nldm
                                                            4.7880  n
write_delay_counter_reg[0]
                          DFF_X1          NangateOpenCellLibrary_PDKv1_2_v2008_10_slow_nldm
                                                            4.7880  n
write_delay_counter_reg[1]
                          DFF_X1          NangateOpenCellLibrary_PDKv1_2_v2008_10_slow_nldm
                                                            4.7880  n
write_delay_counter_reg[2]
                          DFF_X1          NangateOpenCellLibrary_PDKv1_2_v2008_10_slow_nldm
                                                            4.7880  n
--------------------------------------------------------------------------------
Total 27446 cells                                         37905.7981
1
```



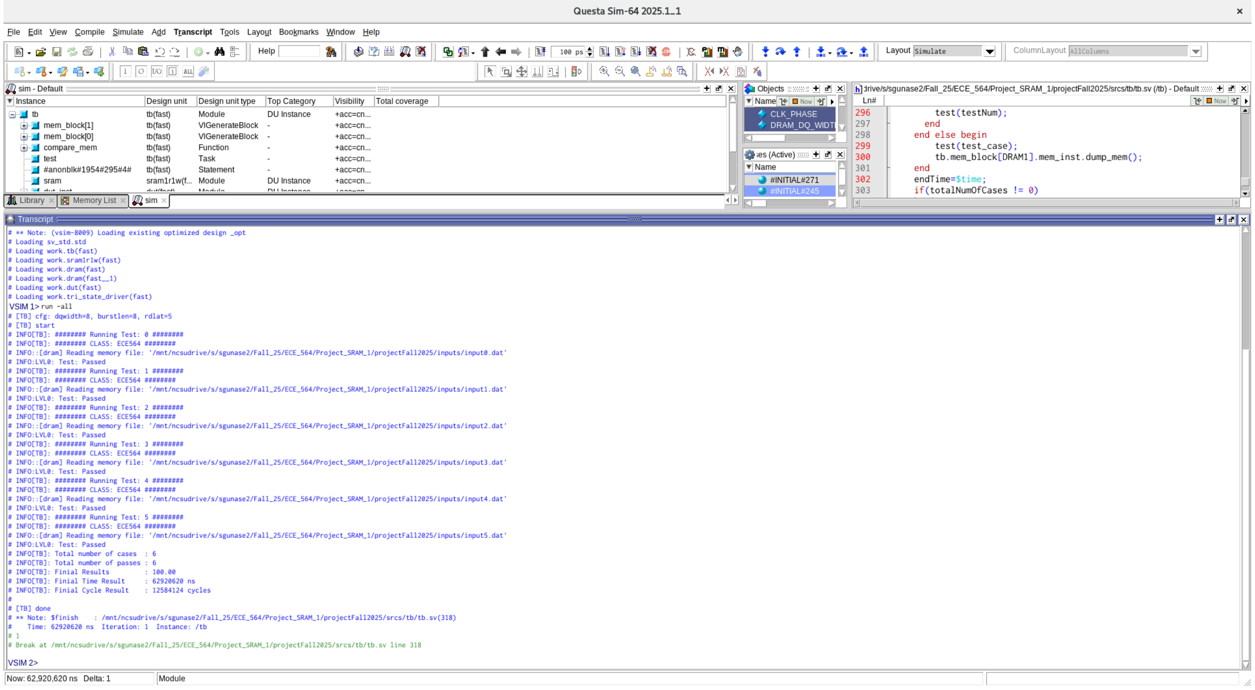
```
 # A fanout number of 1000 was used for high fanout net computations.

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

  Startpoint: kernel_ram_reg[4][5]
              (rising edge-triggered flip-flop clocked by clk)
  Endpoint: f_multi_result2_reg[4][13]
            (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
  kernel_ram_reg[4][5]/CK (DFF_X1)                       0.0000 #   0.0000 r
  kernel_ram_reg[4][5]/Q (DFF_X1)                        0.4904     0.4904 r
  U22056/Z (BUF_X4)                                      0.3671     0.8575 r
  U20430/Z (XOR2_X2)                                     0.4199     1.2774 r
  U21609/ZN (NAND2_X2)                                   0.0768     1.3542 f
  U20788/ZN (NAND2_X2)                                   0.1713     1.5255 r
  U21608/ZN (INV_X2)                                     0.0514     1.5769 f
  U21084/ZN (NAND2_X2)                                   0.0721     1.6490 r
  U6476/ZN (NAND2_X1)                                    0.0720     1.7210 f
  U20960/ZN (NAND2_X2)                                   0.1133     1.8344 r
  U8501/ZN (XNOR2_X1)                                    0.3524     2.1868 r
  U10742/ZN (XNOR2_X2)                                   0.3342     2.5210 r
  U20344/ZN (NAND2_X1)                                   0.1222     2.6432 f
  U13431/ZN (INV_X1)                                     0.1190     2.7622 r
  U18055/ZN (NAND2_X1)                                   0.0749     2.8371 f
  U10836/ZN (NAND3_X2)                                   0.1901     3.0272 r
  U6782/ZN (AOI21_X1)                                    0.1120     3.1392 f
  U10835/ZN (NAND3_X2)                                   0.2402     3.3794 r
  U10834/ZN (NAND2_X2)                                   0.0679     3.4473 f
  f_multi_result2_reg[4][13]/D (DFF_X2)                  0.0000     3.4473 f
  data arrival time                                                 3.4473

  clock clk (rise edge)                                  3.8000     3.8000
  clock network delay (ideal)                            0.0000     3.8000
  clock uncertainty                                     -0.0500     3.7500
  f_multi_result2_reg[4][13]/CK (DFF_X2)                 0.0000     3.7500 r
  library setup time                                    -0.3026     3.4474
  data required time                                                3.4474
  --------------------------------------------------------------------------
  data required time                                                3.4474
  data arrival time                                                -3.4473
  --------------------------------------------------------------------------
  slack (MET)                                                       0.0001
```



```
Information: Updating design information... (UID-85)
Warning: Design 'dut' contains 1 high-fanout nets. A fanout number of 1000 will be used for delay calculations involving these nets. (TIM-134)

****************************************
Report : timing
        -path full
        -delay min
        -max_paths 1
Design : dut
Version: T-2022.03-SP4
Date   : Tue Nov 18 11:25:58 2025
****************************************

 # A fanout number of 1000 was used for high fanout net computations.

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

  Startpoint: temp_sram_buff0_1_reg[3][5]
              (rising edge-triggered flip-flop clocked by clk)
  Endpoint: main_sram_buff_0_reg[3][29]
            (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
  temp_sram_buff0_1_reg[3][5]/CK (DFF_X2)                0.0000 #   0.0000 r
  temp_sram_buff0_1_reg[3][5]/Q (DFF_X2)                 0.0541     0.0541 r
  U28700/ZN (OAI211_X1)                                  0.0176     0.0716 f
  main_sram_buff_0_reg[3][29]/D (DFF_X1)                 0.0000     0.0716 f
  data arrival time                                                 0.0716

  clock clk (rise edge)                                  0.0000     0.0000
  clock network delay (ideal)                            0.0000     0.0000
  clock uncertainty                                      0.0500     0.0500
  main_sram_buff_0_reg[3][29]/CK (DFF_X1)                0.0000     0.0500 r
  library hold time                                     -0.0001     0.0499
  data required time                                                0.0499
  --------------------------------------------------------------------------
  data required time                                                0.0499
  data arrival time                                                -0.0716
  --------------------------------------------------------------------------
  slack (MET)                                                       0.0217


1
```

# 8. Correctness Verification

All six test cases passed and there were no major synthesis error.



| Instance | Design unit | Design unit type | Top Category | Visibility | Total coverage |
|---|---|---|---|---|---|
| tb | tb(fast) | Module | DU Instance | +acc=cn... | |
| mem_block[1] | tb(fast) | VlGenerateBlock | - | +acc=cn... | |
| mem_block[0] | tb(fast) | VlGenerateBlock | - | +acc=cn... | |
| compare_mem | tb(fast) | Function | - | +acc=cn... | |
| test | tb(fast) | Task | - | +acc=cn... | |
| #anonblk#1954#295#4# | tb(fast) | Statement | - | +acc=cn... | |
| sram | sram1r1w(f... | Module | DU Instance | +acc=cn... | |

ɖrive/s/sgunase2/Fall_25/ECE_564/Project_SRAM_1/projectFall2025/srcs/tb/tb.sv (/tb) - Default

```
            test(testNum);
         end
      end else begin
            test(test_case);
            tb.mem_block[DRAM1].mem_inst.dump_mem();
      end
      endTime=$time;
      if(totalNumOfCases != 0)
```

```
# ** Note: (vsim-8009) Loading existing optimized design _opt
# Loading sv_std.std
# Loading work.tb(fast)
# Loading work.sram1r1w(fast)
# Loading work.dram(fast)
# Loading work.dram(fast__1)
# Loading work.dut(fast)
# Loading work.tri_state_driver(fast)
VSIM 1> run -all
# [TB] cfg: dqwidth=8, burstlen=8, rdlat=5
# [TB] start
# INFO[TB]: ######## Running Test: 0 ########
# INFO[TB]: ######## CLASS: ECE564 ########
# INFO::[dram] Reading memory file: '/mnt/ncsudrive/s/sgunase2/Fall_25/ECE_564/Project_SRAM_1/projectFall2025/inputs/input0.dat'
# INFO:LVL0: Test: Passed
# INFO[TB]: ######## Running Test: 1 ########
# INFO[TB]: ######## CLASS: ECE564 ########
# INFO::[dram] Reading memory file: '/mnt/ncsudrive/s/sgunase2/Fall_25/ECE_564/Project_SRAM_1/projectFall2025/inputs/input1.dat'
# INFO:LVL0: Test: Passed
# INFO[TB]: ######## Running Test: 2 ########
# INFO[TB]: ######## CLASS: ECE564 ########
# INFO::[dram] Reading memory file: '/mnt/ncsudrive/s/sgunase2/Fall_25/ECE_564/Project_SRAM_1/projectFall2025/inputs/input2.dat'
# INFO:LVL0: Test: Passed
# INFO[TB]: ######## Running Test: 3 ########
# INFO[TB]: ######## CLASS: ECE564 ########
# INFO::[dram] Reading memory file: '/mnt/ncsudrive/s/sgunase2/Fall_25/ECE_564/Project_SRAM_1/projectFall2025/inputs/input3.dat'
# INFO:LVL0: Test: Passed
# INFO[TB]: ######## Running Test: 4 ########
# INFO[TB]: ######## CLASS: ECE564 ########
# INFO::[dram] Reading memory file: '/mnt/ncsudrive/s/sgunase2/Fall_25/ECE_564/Project_SRAM_1/projectFall2025/inputs/input4.dat'
# INFO:LVL0: Test: Passed
# INFO[TB]: ######## Running Test: 5 ########
# INFO[TB]: ######## CLASS: ECE564 ########
# INFO::[dram] Reading memory file: '/mnt/ncsudrive/s/sgunase2/Fall_25/ECE_564/Project_SRAM_1/projectFall2025/inputs/input5.dat'
# INFO:LVL0: Test: Passed
# INFO[TB]: Total number of cases  : 6
# INFO[TB]: Total number of passes : 6
# INFO[TB]: Finial Results         : 100.00
# INFO[TB]: Finial Time Result     : 62920620 ns
# INFO[TB]: Finial Cycle Result    : 12584124 cycles
#
# [TB] done
# ** Note: $finish    : /mnt/ncsudrive/s/sgunase2/Fall_25/ECE_564/Project_SRAM_1/projectFall2025/srcs/tb/tb.sv(318)
#    Time: 62920620 ns  Iteration: 1  Instance: /tb
# 1
# Break at /mnt/ncsudrive/s/sgunase2/Fall_25/ECE_564/Project_SRAM_1/projectFall2025/srcs/tb/tb.sv line 318

VSIM 2>
```

Now: 62,920,620 ns  Delta: 1    Module

---

# 9. Conclusion

This project implements a fully pipelined CNN hardware accelerator for a 1024×1024 image using:

- 4×4 convolution
- Leaky-ReLU
- 2×2 average pooling
- Fully streaming pipeline aligned to DRAM bandwidth
- SRAM-assisted sliding window engine
- 7-stage computation pipeline

The design meets all memory/timing specifications and utilizes SystemVerilog features such as `typedef enum`, `always_ff`, and multi-dimensional packed arrays.