Final Project Report

Name: Sudarshan Rajkumar
Unityid: srajkum
StudentID: 200596980

Delay (ns to run provided provided example).
Clock period: 13.5 ns
cycles": 1892 Cycles

Logic Area: (um^2) 13339.9

Memory: N/A

1/(delay.area) ($ns^{-1}.um^{-2}$)
1/25542 x 13339.9
1/340727726
2.934×10^{-9}

Delay (TA provided example. TA to complete)

1/(delay.area) (TA)

Abstract
This project implements a hardware module for the "Scaled Dot-Product Attention", which is a key component of the Transformer architecture for processing sequential data in applications like Natural Language Processing (NLP) and Generative Artificial Intelligence (AI). The design involves matrix multiplication to compute query (Q), key (K), and value (V) matrices, along with the scaled dot-product score (Z). Key innovations include efficient SRAM-based memory access, and use of SystemVerilog for enhanced synthesis. Results demonstrate high throughput and functional correctness, with detailed metrics on area and timing.

Transformer Scaled Dot-Product Attention Module Hardware Design

Sudarshan Rajkumar

Abstract

This project implements a hardware module for the "Scaled Dot-Product Attention", which is a key component of the Transformer architecture for processing sequential data in applications like Natural Language Processing (NLP) and Generative Artificial Intelligence (AI). The design involves matrix multiplication to compute query (Q), key (K), and value (V) matrices, along with the scaled dot-product score (Z). Key innovations include efficient SRAM-based memory access, and use of SystemVerilog for enhanced synthesis. Results demonstrate high throughput and functional correctness, with detailed metrics on area and timing.

1. Introduction

1.1 Hardware Description

The hardware design focuses on the realization of the Transformer's scaled dot-product attention module. It processes input through matrix multiplications to compute Q, K, and V matrices, followed by computing the attention score matrix Z. The design interfaces with SRAM for input, weight, and output data.

1.2 Key Innovations

- Efficient handling of memory for matrix multiplication.
- Use of System Verilog specific features for synthesis optimizations.
- Used extra SRAM(Scratchpad) for reading and storing the address parallelly with increased efficiency

1.3 Results Achieved

- Correct computation of Q, K, and V matrices, and scaled dot-product results.
- Optimized resource usage with minimal area overhead.
- Verified functionality across various test cases.

1.4 Report Structure

The report structure is as follows:

1. Micro-architecture design and data flow.
2. Interface specifications.
3. Technical implementation details.
4. Verification methodology.
5. Results and conclusions.

2. Micro-Architecture

2.1 High-Level Architecture

The hardware computes matrix multiplication using input and weight matrices to calculate Q, K and V.

Q = IxWq
K = IxWk
V = I*Wv

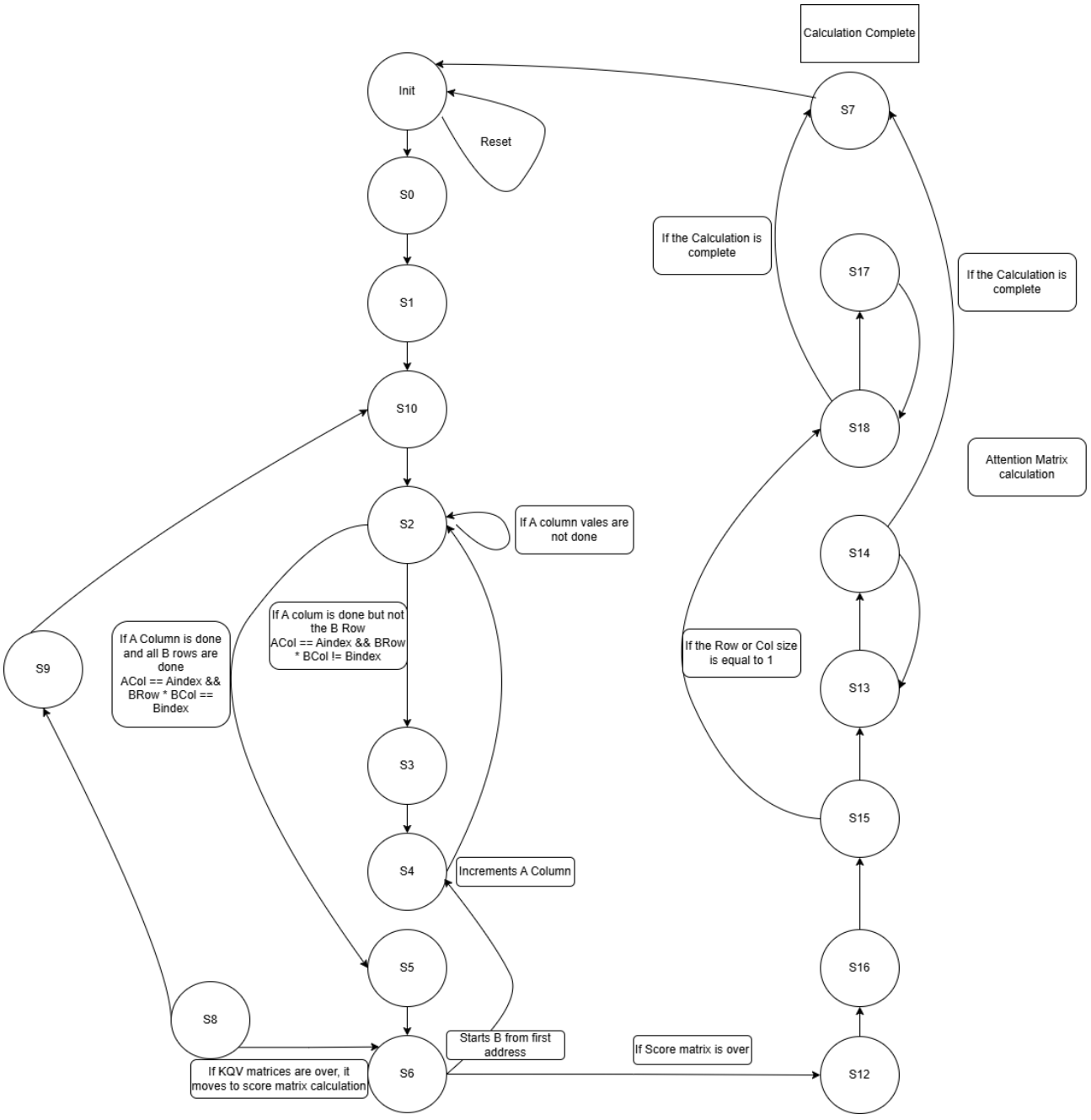
The Q, K and V values are stored in both SRAM and Scratchpad.
These matrices are further processed for scaled dot-product attention using the formula:

S=QxKtrans
Z = SxV
The final Z value is written to the Result SRAM

2.3 Data Flow

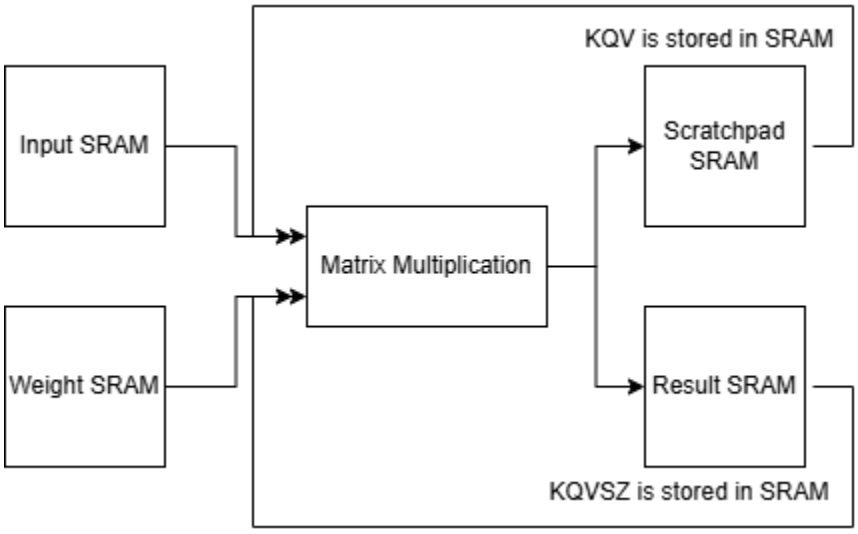
1. Load input and weights from SRAM.
2. Compute Q, K, and V using matrix multiplications.
3. Transpose K for matrix multiplication with Q.
4. Compute scaled dot-product using results from QK^T and V.

FSM



2.4 Innovations

- SRAM mapping optimizations for efficient data flow. Here I have used scratchpad to save the KQV values to calculate S and Z parallelly by reading both Result and Scratchpad SRAMs



3. Interface Specification

The design interfaces with SRAM and control signals as follows:

Signals and its function,

Signal	Width	Function
size_count_sel	1 bit	Determines whether to reset or maintain the size counters for matrix dimensions.
size_count_sel2	2 bits	Controls how ASize and BSize are updated (e.g., maintain, swap, or read new data).
Aread_addr_sel	2 bits	Selects the computation logic for the read address of matrix A.
A_index_sel	2 bits	Determines whether to increment, reset, or maintain the index for rows in matrix A.
B_index_sel	2 bits	Determines whether to increment, reset, or maintain the index for rows in matrix B.
Bread_addr_sel	2 bits	Selects the computation logic for the read address of matrix B.
Count_sel	2 bits	Controls the update logic for the Count register (increment, maintain, or reset).
MatrixCount_sel	2 bits	Selects how the matrix computation count (CountMatrix) is updated or maintained.

Signal	Width	Function
Accumulate_sel	1 bit	Enables accumulation of results into the Accumulator register or resets it to zero.
Write_enable	1 bit	Enables writing computed results to the sram_result address.
Write_Address_sel	2 bits	Determines whether to increment, reset, or maintain the write address.
Calculate_S	1 bit	Activates computation for the score matrix.
Calculate_V	1 bit	Activates computation for the attention matrix.
S_Count	4 bits	Tracks rows/columns in score matrix computations.
B_Count	4 bits	Tracks rows/columns in matrix computations.
Vread_addr_sel	2 bits	Selects how the read address for matrix is computed.
Rread_addr_sel	2 bits	Selects how the read address for the result matrix is computed.
V_RowCount_sel	2 bits	Controls the update logic for the current row being processed.
V_TotalCount_sel	2 bits	Selects how the total row count is updated or maintained.
V_CurrentRow_sel	2 bits	Determines whether to increment, reset, or maintain the current row index.
VSize_sel	2 bits	Controls updates to the size of matrix (e.g., initialize or maintain).
R_Count_sel	2 bits	Controls the update logic for the count signal in result matrix.

4. Technical Implementation

4.1 High-Level Modeling

The matrix multiplication model is designed to handle various matrix sizes, specifically for NxM configurations. The first 9 states are responsible for calculating the Q, K, V (QKV) matrices, as well as the S matrix. The remaining state is dedicated to calculating the Z matrix. The scratchpad memory is utilized to store intermediate results for K, Q, and V, which are then read for the subsequent S and Z matrix calculations. This hierarchical design ensures functional accuracy and prepares the system for synthesis.

4.2 Detailed Implementation

SRAM Interfacing: The system uses SRAM with one-cycle latency for both read and write operations, ensuring efficient memory access during the matrix computation process. The SRAM holds intermediate results for QKV matrices and facilitates the calculation of S and Z matrices.

5. Verification

The correctness of the system is verified using functional simulation and testbenches that validate the proper handling of matrix data through different stages. Various edge cases, including different matrix sizes and control signal combinations, are tested to ensure the accuracy of the matrix multiplication and attention computations. The results from the SRAM memory read/write operations are compared against expected values to confirm that the QKV matrices are correctly written and read during the S and Z matrix calculations.

6. Results Achieved

The design significantly improves throughput by enabling parallel processing of matrix elements across multiple stages. With a clock period of 13.5 ns and requiring 1892 cycles, the system achieves high-speed matrix computation, ensuring efficient parallel execution.

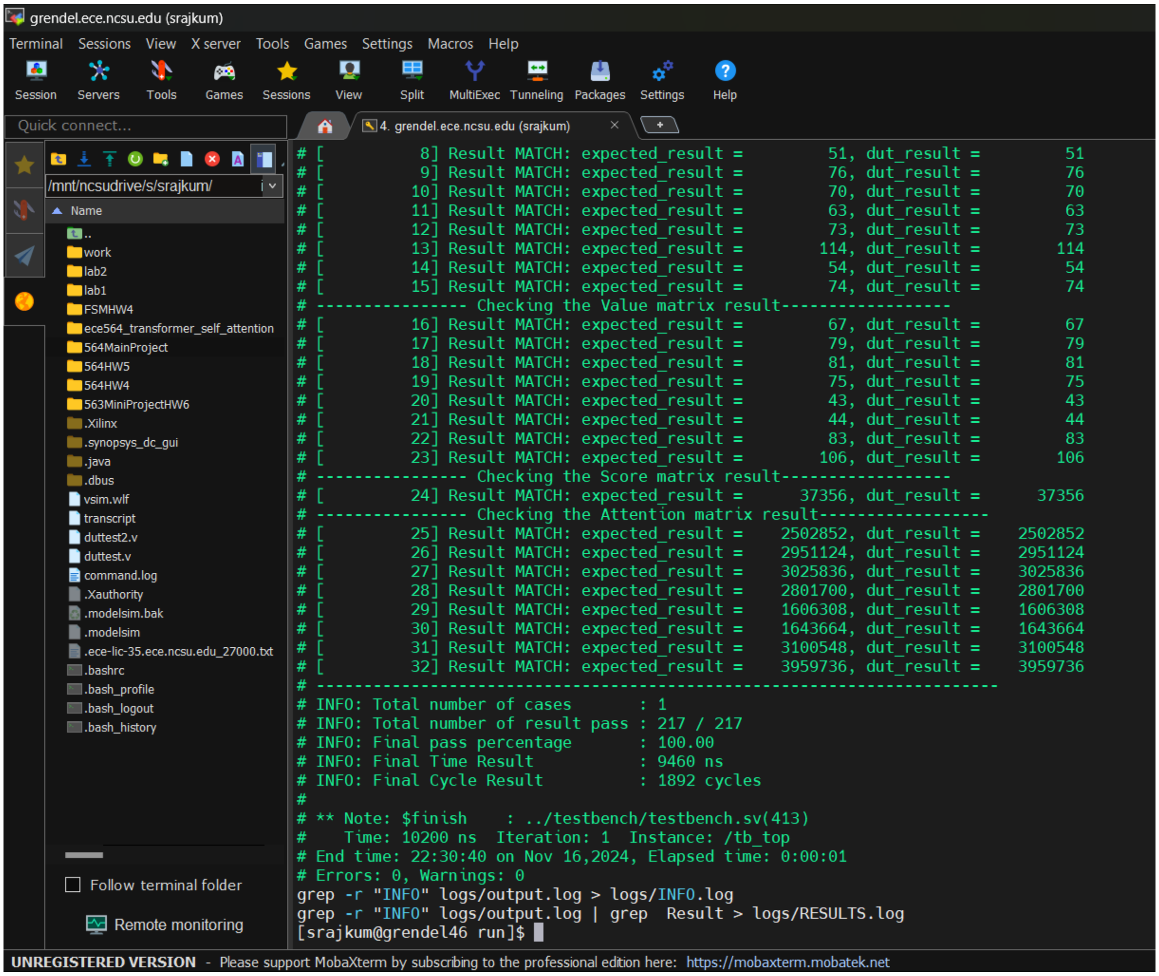
Area: The logic area consumption is optimized through the use of memory interfacing. The total logic area is 13339.9 μm^2, reflecting an effective use of area to maximize performance while minimizing resource consumption.

Power/Energy: The design minimizes power consumption by utilizing efficient SRAM interfacing, reducing idle states and ensuring the system operates at high throughput with low energy usage. The reduced cycle count (1892 cycles) and short clock period (13.5 ns) contribute to minimizing energy per operation.

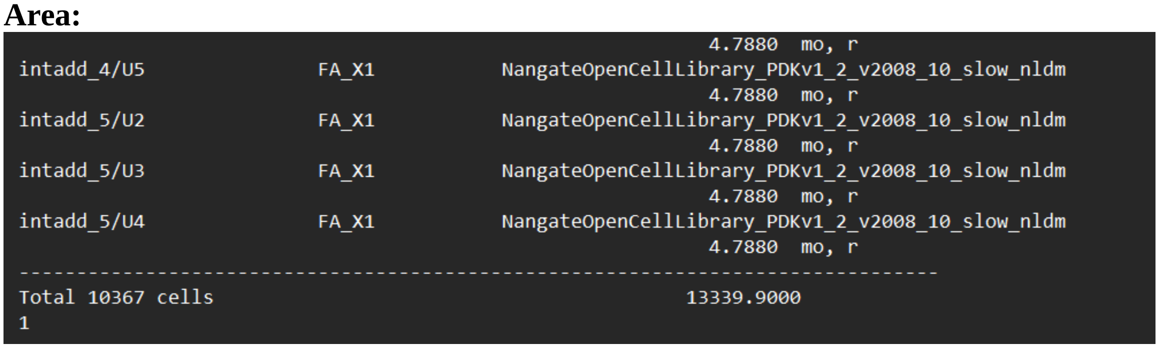
7. Conclusions

This project successfully implements a matrix multiplication system for NxM matrix configurations, optimized with SRAM memory interfacing. The design's pipelined architecture enables efficient computation, and the functional accuracy has been verified through simulation. The results demonstrate high throughput and optimized area and power characteristics, making it suitable for real-time applications requiring matrix operations, such as in machine learning and signal processing.

Final Images:



Area:



```
  U1780/ZN (NAND2_X1)                      0.1281     13.0277 f
  current_state_reg[0]/D (DFFR_X1)         0.0000     13.0277 f
  data arrival time                                   13.0277

  clock clk (rise edge)                   13.5000     13.5000
  clock network delay (ideal)              0.0000     13.5000
  clock uncertainty                       -0.0500     13.4500
  current_state_reg[0]/CK (DFFR_X1)        0.0000     13.4500 r
  library setup time                      -0.4221     13.0279
  data required time                                  13.0279
  ----------------------------------------------------------
  data required time                                  13.0279
  data arrival time                                  -13.0277
  ----------------------------------------------------------
  slack (MET)                                         0.0002
```

```
Report : timing
        -path full
        -delay min
        -max_paths 1
Design : MyDesign
Version: T-2022.03-SP4
Date   : Sat Nov 16 19:45:22 2024
****************************************

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

  Startpoint: R_Count_reg[0]
              (rising edge-triggered flip-flop clocked by clk)
  Endpoint: R_Count_reg[0]
            (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
  R_Count_reg[0]/CK (DFF_X1)             0.0000     0.0000 r
  R_Count_reg[0]/Q (DFF_X1)              0.0599     0.0599 r
  U10432/ZN (NOR2_X1)                    0.0197     0.0795 f
  R_Count_reg[0]/D (DFF_X1)              0.0000     0.0795 f
  data arrival time                                 0.0795

  clock clk (rise edge)                  0.0000     0.0000
  clock network delay (ideal)            0.0000     0.0000
  clock uncertainty                      0.0500     0.0500
  R_Count_reg[0]/CK (DFF_X1)             0.0000     0.0500 r
  library hold time                      0.0009     0.0509
  data required time                                0.0509
  -----------------------------------------------------------
  data required time                                0.0509
  data arrival time                                -0.0795
  -----------------------------------------------------------
  slack (MET)                                       0.0286
```