COMPENG 3DQ5

Digital Systems Design

Final Project Report

Brayden Roberts – roberb28 – 400478205

Deyontae Patterson – patted9 – 400480946

As a future member of the engineering profession, I, Deyontae Patterson am responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is my own and adheres to the Academic Integrity Policy of McMaster University and the Code of Conduct of the Professional Engineers of Ontario.

**Submitted by Brayden Roberts & Deyontae Patterson**

Table of Contents

[Table of Figures ii](#_Toc183464138)

[Introduction 1](#_Toc183464139)

[Implementation Details 2](#_Toc183464140)

[Milestone 1: Up-sampling and Colour Space Conversion 2](#_Toc183464141)

[Efficiency 3](#_Toc183464142)

[Latency 4](#_Toc183464143)

[Milestone 2: Inverse Discrete Cosine Transform 5](#_Toc183464144)

[Resource Usage and Critical Path 6](#_Toc183464145)

[Weekly Activity and Progress 7](#_Toc183464146)

[Conclusion 8](#_Toc183464147)

[References 9](#_Toc183464148)

# Table of Figures

[Figure 1: Original Image to be Decompressed 1](#_Toc183464149)

[Figure 2: Finite State Machine of top\_state 5](#_Toc183464150)

[Table 1: Milestone 1 Common Case State Table 2](#_Toc183464151)

[Table 2: All Registers and their functions for Milestone 1 3](#_Toc183464152)

[Table 3: All Registers and their functions for Milestone 2 6](#_Toc183464153)

[Table 4: Timeline of Project Completion over 5 weeks 8](#_Toc183464154)

# Introduction

The goal of the COMPENG 3DQ5 project is to design an image decompression implementation in Verilog HDL. This project integrates a number of concepts covered in Labs 1 to 5 such as interpolation (Milestone 1), signal transformation (Milestone 2) and lossless decoding (Milestone 3) to restore the original image. The implementation is meant to be operate under the efficiency constraint of using only 3 Multiplier components with a minimum of 75% usage for interpolation and colour space conversion, and 85% for Inverse Discrete Cosine Transform.

A black motorcycle parked on a race track

Description automatically generated

Figure 1: Original Image to be Decompressed

# Implementation Details

## Milestone 1: Up-sampling and Colour Space Conversion

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **State** | **S\_CONVERT\_RGB\_0** | **S\_CONVERT\_RGB\_1** | **S\_CALC\_U** | **S\_CALC\_V** | **S\_CONVERT\_RGB\_2** | **S\_CONVERT\_RGB\_3** | **S\_NEXT\_PIXEL** |
| pixel\_pair\_count | 1 | 1 | 1 | 1 | 1 | 2 | 2 |
| SRAM\_Address | U | V | Y | R[even]G[even] |  | B[even]R[odd] | G[odd]B[odd] |
| SRAM\_read\_data |  |  |  | U | V | Y |  |
| SRAM\_write\_data |  |  |  | R[even]G[even] |  | B[even]R[odd] | G[odd]B[odd] |
| SRAM\_we\_n | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| R | M1+M2 |  |  |  | M1+M2 |  |  |
| G | M1 - M3 | G-M1 |  |  | M1 - M3 | G-M1 |  |
| B | M1 | B+M2 |  |  | M1 | B+M2 |  |
| Y |  |  |  |  |  | Y |  |
| U' |  |  | M1-M2+M3 |  |  |  |  |
| V' |  |  |  | M1-M2+M3 |  |  |  |
| M1 | (Y-16)\*76284 | 53281\*(V-128) | 21\*(U[0]+U[5]) | 21\*(V[0]+V[5]) | (Y-16)\*76284 | 53281\*(V-128) |  |
| op1 | (V-128) | (U[0]+U[5]) | (V[0]+V[5]) | (Y-16) | (V-128) |  | (Y-16) |
| op2 | 53281 | 21 | 21 | 76284 | 53281 |  | 76284 |
| M2 | (V-128)\*104595 | 132251(U-128) | 52\*(U[1]+U[4]) | 52\*(V[1]+V[4]) | (V-128) \* 104595 | 132251(U-128) |  |
| op1 | (U-128) | (U[1]+U[4]) | (V[1]+V[4]) | (V-128) | (U-128) |  | (V-128) |
| op2 | 132251 | 52 | 52 | 104595 | 132251 |  | 104595 |
| M3 | (U-128)\*25624 |  | 159\*(U[2]+U[3]) | 159\*(V[2]+V[3]) | 25624 \* (U-128) |  |  |
| op1 |  | (U[2]+U[3]) | (V[2]+V[3]) | (U-128) |  |  | (U-128) |
| op2 |  | 159 | 159 | 25624 |  |  | 25624 |

Table 1: Milestone 1 Common Case State Table

**NOTE:** In the S\_CONVERT\_RGB\_0 and S\_CONVERT\_RGB\_1 states, we “sample” a U and V value respectively. This only occurs once every 2 rounds through the common cases. This is to ensure that the UV sampling does not outpace Y since Y is meant to be sampled twice as often ([click here for access to the full state table with lead in and lead out cases](https://docs.google.com/spreadsheets/d/1fA-dkOMUllvAHETbTgV1i32uIagjIAp-RHrNjzd055Y/edit?usp=sharing)).

### Efficiency

|  |  |  |
| --- | --- | --- |
| **Register(s)** | **Width (in bits)** | **Purpose** |
| Y,U,V | 16 | Storage of all 3 kinds of values (U & V after inrerpolation) |
| U\_prime, V\_prime | 32 | Storage for U and V fetched from SRAM |
| U\_inter\_data, V\_inter\_data | Five 2-Dimensional register each 8 bits wide | Registers containing surrounding U/V values for interpolation |
| R,G,B | 32 | Storage for calculated RGB and values for storing in SRAM |
| B\_odd\_buff | 32 | Buffer made for odd B values to prevent being overwritten while writing |
| red\_write, green\_write, blue\_write | 8 | Registers paired together (RG, BR, GB) for easy writing to SRAM |
| initialize\_flag | 1 | Boolean-styled Flag for managing the initial reading of U and V values |
| pixel\_pair\_count | 16 | Counter for keeping track of pixel pairs – helps with reading correct SRAM address |
| rgb\_write\_count | 17 | Counter keeping track of RGB values written back into SRAM – helps with proper addressing |
| per\_row\_count | 8 | Counter for keeping track of column index – controls whether to go to S\_CALC\_V or S\_LEAD\_OUT\_0 |
| M1/2/3\_op\_1/2 | 32 | Multiplication operands 1 and 2 for 3 multipliers |
| M1/2/3\_result\_long, M1/2/3\_result | 64, 32 respectively | Raw multiplication result; clipped value |

Table 2: All Registers and their functions for Milestone 1

From the state table, we can calculate the multiplier efficiency:

### Latency

The Latency

## Milestone 2: Inverse Discrete Cosine Transform

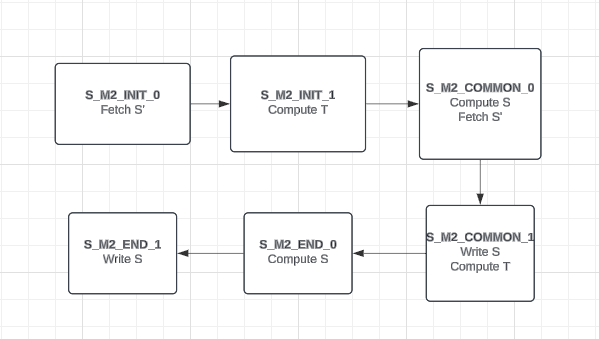


Figure 2: Finite State Machine of top\_state

|  |  |  |
| --- | --- | --- |
| **Register(s)** | **Width (in bits)** | **Purpose** |
| rows, cols | 5 | Keeps track of row and column indexes within each block – helps with addressing for reading and writing |
| rows\_long | 6 |  |
| row\_address, col\_address | 5,6 respectively | Concatenation of row/column base with row/column of the block – gives address of specific location in SRAM |
| col\_base, row\_base | 6,5 respectively | The SRAM address of row/column 0 of a given block |
| last\_block\_flag | 1 | Boolean style flag for signalling when code has reached the last block in the SRAM |
| data\_count | 8 | Counts data points gathered |
| data\_count\_long | 6 | Gathers more data points when two counters are active simultaneously |
| write\_count, write\_count\_1 | 6 | Keeps track of write counts when multiple writes are happening in different locations |
| address\_a/b | Three 2-Dimensional registers each 7 bits wide | DPRAM address ports |
| write\_data\_a/b | Three 2-Dimensional registers each 32 bits wide | DPRAM address ports |
| write\_enable\_a/b | Three 2-Dimensional registers each 1 bit wide | DPRAM address ports |
| read\_data\_a/b | Three 2-Dimensional registers each 32 bits wide | DPRAM address ports |
| M1/2/3\_op\_1/2 | 32 | Multiplication operands 1 and 2 for 3 multipliers |
| M1/2/3\_result\_long, M1/2/3\_result | 64, 32 respectively | Raw multiplication result; clipped value |
| Tmac | 32 | Accumulation of elements of the T matrix |
| Smac\_1/2/3 | 32 | Accumulators of values of S matrix |
| Smac\_3\_buf | 32 | Buffer for 3rd accumulation |

Table 3: All Registers and their functions for Milestone 2

### Efficiency

## Resource Usage and Critical Path

The resource usage for our project increased compared to Lab 5 Experiment 4, with 1982 logic elements (2%) and 920 registers vs 616 logic elements (<1%) and 369 registers previously. This growth reflects the added complexity and functionality in this design phase. The critical path, identified by the Timing Analyzer, spans from M1\_op\_2 to Tmac, indicating the longest delay occurs between a multiplication and an accumulation step. While the design remains efficient overall, future revisions could optimize resource usage by removing unnecessary registers and shortening the critical path through techniques like pipelining or restructuring operations, improving both performance and timing.

# Weekly Activity and Progress

|  |  |  |
| --- | --- | --- |
| **Week** | **Project Progress** | **Member Contribution** |
| 1 | * Attended Lecture to understand basis and additional context for the project | **Together:**  Attended Project-centered lectures |
| 2 | * Brainstormed state table and hardware circuit for Milestone 1 * Prepared Milestone 1 Module by initializing in project module | **Brayden:**  Began forming Milestone 1 state table based on Lab 5.  **Deyontae:**  Linked Project and Milestone 1 Modules |
| 3 | * Fully translated Milestone 1 code from state table to Verilog * Began initial testing and debugging phase | **Deyontae:**  Began coding from Milestone 1 state tablelead-in, common and lead-out cases  **Brayden:**  Completed Code for Milestone 1 |
| 4 | * Completed Milestone 1 Debugging Phase * Brainstormed solutions for Milestone 2 * Developed state table for each main block in process (Fetch S’, Compute T, Compute S, Write S) and for top\_state which controls all intermediary processes * Discussed DPRAM storage of S’, C, CT and S * Completed initial draft of Milestone 2 code * Debugged Milestone 2 * Developed final report | **Brayden:**  Completed Final Code Draft for Milestone 1  **Deyontae:**  Drafted initial State Tables for Milestone 2  **Together:**  Decided on the best strategy for storing S’, C, CT and S across DPRAMs  **Brayden**:  Fully Integrated State table into Code  **Deyontae**:  Hard-coded individual C and CT into DPRAM mif files.  **Brayden:**  Debugged Milestone 2 |

Table 4: Timeline of Project Completion over 5 weeks

# Conclusion

In conclusion, this project provided us with an invaluable learning experience. While it was challenging and occasionally frustrating, it allowed us to develop a deeper understanding of implementing a scaled project within a limited timeframe. Over the course of four weeks, we gained critical insights into the importance of teamwork, effective communication, and dedication. The project, more than anything, forced us to understand the value of time management and proper pacing for project tasks in the future.

We feel deeply grateful to our professors and TAs for their constant guidance and support throughout the course. Their dedication in facilitating a learning environment has made this experience one of the most fundamental steps in preparing us for our future careers as computer engineers. Their encouragement and expertise have been instrumental in helping us navigate and complete this project successfully.

**\*\*\*NOTE**: The last fully functioning milestone is Milestone 1; committed Monday, November 25, 2024. Commit message – “Ensured Milestone 1 passes both testbench\_v0 and v1”.

# References

|  |  |
| --- | --- |
| [1] | N. Nicolici, "3DQ5 project report guidelines," November 2024. [Online]. Available: https://avenue.cllmcmaster.ca/d2l/le/content/633401/viewContent/4901064/View. [Accessed 25 November 2024]. |
| [2] | N. Nicolici, "3DQ5 project description," October 2024. [Online]. Available: https://avenue.cllmcmaster.ca/d2l/le/content/633401/viewContent/4869746/View. [Accessed 25 November 2024]. |
| [3] | N. Wu and M. Li, November 2020. [Online]. Available: https://github.com/wudiudiu07/3DQ5-lab/blob/master/project/coe3dq5\_group\_14\_project\_report.pdf. [Accessed 24 November 2024]. |
| [4] | "Decimal to Hexadecimal converter," RapidTables, [Online]. Available: https://www.rapidtables.com/convert/number/decimal-to-hex.html. [Accessed 23 November 2024]. |