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

Not needed

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.

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).

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:

$$Efficiency(\%) = \frac{Total\ \#\ of\ multipliers\ used}{\#\ of\ clock\ cycles \cdot 3\ multipliers} \cdot 100$$

$$Efficiency(\%) = \frac{2+3+3+3+2+0+3}{7 \cdot 3} \cdot 100$$

$$Efficiency(\%) = \frac{16}{21} \cdot 100$$

$$Efficiency(\%) = 76.2\%$$

Latency

$$Total\ clock\ cycles = Lead\ In\ States + Lead\ Out\ States + Common\ Cases \cdot Common\ Cycles$$

$$Total\ clock\ cycles = 7 + 2 + 7 \cdot \left(\frac{320 \cdot 240}{2}\right)$$

$$Total\ clock\ cycles = 268{,}809$$

The Latency ?

CANNOT BE CORRECT BECAUSE LEAD-IN/LEAD-OUT ARE PER ROW!

Milestone 2: Inverse Discrete Cosine Transform

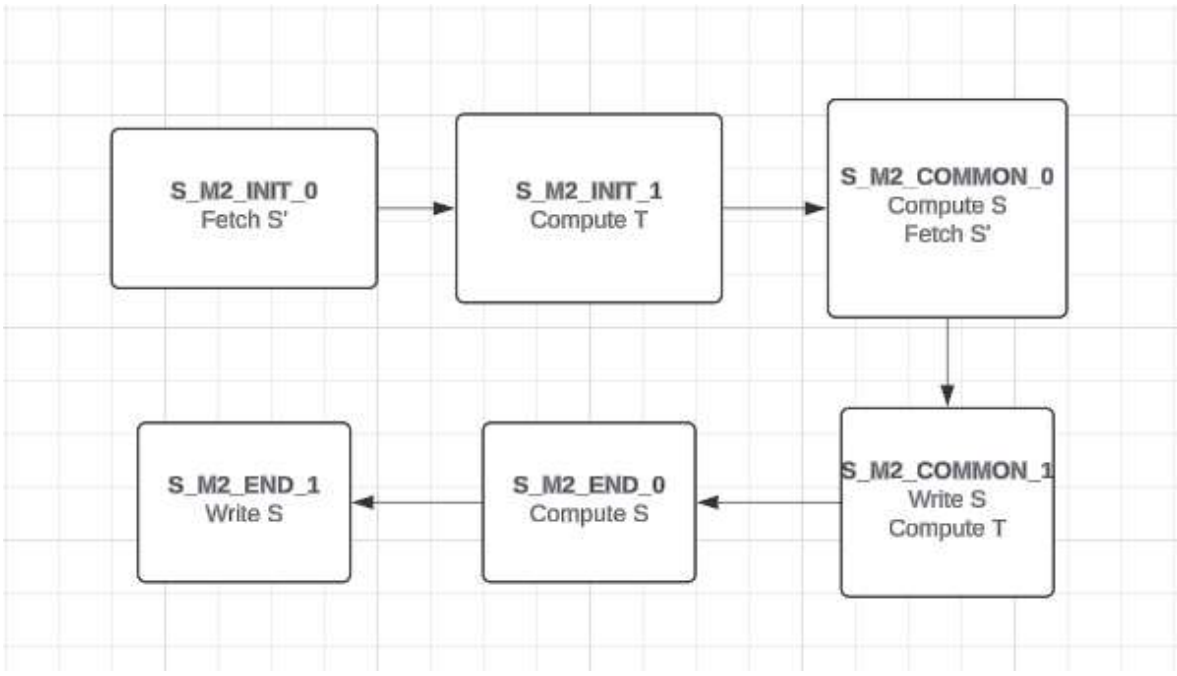

Figure 2: Finite State Machine of top_state

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

$$Efficiency(\%) = \frac{Total\ \#\ of\ multipliers\ used}{\#\ of\ clock\ cycles \cdot 3\ multipliers} \cdot 100$$

$$Efficiency\ of\ Compute\ T\ (\%) = \frac{8}{3 \cdot 3} \cdot 100 = 88.9\%$$

$$Efficiency\ of\ Compute\ S\ (\%) = \frac{64}{24 \cdot 3} \cdot 100 = 88.9\%$$

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.

RE SPECIFY

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 table lead-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, C^T 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, C^T and S across DPRAMs **Brayden**: Fully Integrated State table into Code

		Deyontae: Hard-coded individual C and C^T 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].