COE3DQ5 – Project Report

Ginoth Kumarakulasingam
Ahanaf Wanan
Group 4

November 27, 2023

Introduction

The objective of this project is to design a digital system that implements image decompression of the McMaster Image Compression revision 17 (.mic17) image compression specification in hardware. This project is divided into 3 milestones, where each milestone performs a different task that is required for the image to be decompressed as the end result. The first milestone implements interpolation and colour space conversion, the second milestone implements inverse discrete cosine transform (IDCT) and the third milestone implements lossless decoding. The hardware design of this system must also satisfy the utilization and memory constraints in place for each milestone.

Implementation Details

Milestone 1 – Upsampling and Colour Space Conversion

The following figure shows how the calculations for interpolation and colour space conversion are organized in the common case. The letter E represents a multiplication done for an RGB calculation for an even pixel, the letter O represents a multiplication done for an RGB calculation for an odd pixel, the letter U represents a multiplication done for a U interpolation calculation, and the letter V represents a multiplication done for a V interpolation calculation.

	Clock Cycle 0	Clock Cycle 1	Clock Cycle 2	Clock Cycle 3	Clock Cycle 4	Clock Cycle 5	Clock Cycle 6
Multiplier 1	E	E	E	E	E		
Multiplier 2		O	O	O	O	O	
Multiplier 3		U	U	U	U	U	U
Multiplier 4		V	V	V	V	V	V

The following table summarizes the register usage in the design of Milestone 1:

Register Name	Bits	Description
y_address, u_address, v_address, RGB_address	4x18	Used to store the next address to be accessed for each memory segment
Y, u_even_buffer, u_odd_buffer, v_odd_buffer	16, 3x8	Used to buffer values read from the SRAM to avoid them being overwritten when future values are read
R_even, G_even, B_even, R_odd, G_odd, B_odd	6x32	MAC units for RGB calculations
u_prime, v_prime, u_prime_buffer, v_prime_buffer	4x32	MAC units for U’ and V’ calculations and to buffer the final values at the end of each common case iteration
mult1_op1, mult1_op2, mult2_op1, mult2_op2, mult34_op2	2x9 and 3x32	

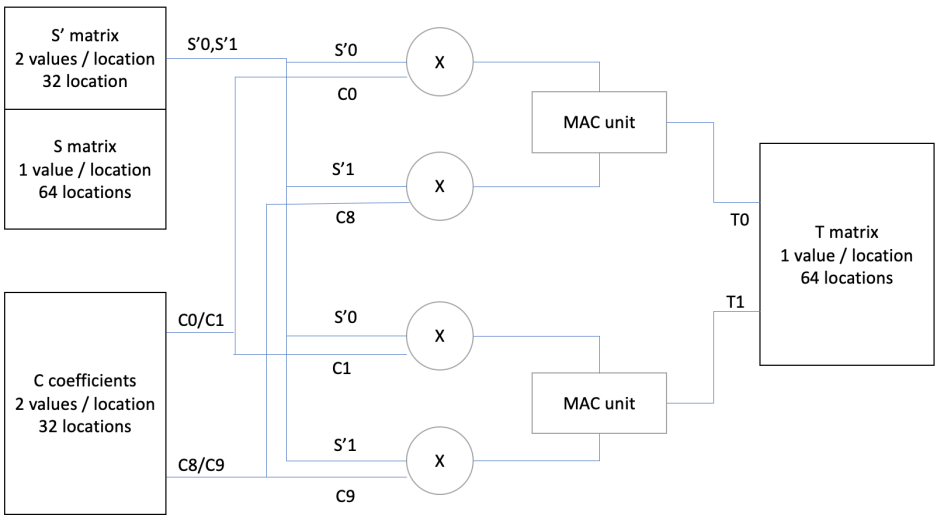
u_plus_5, u_plus_3, … u_minus_3, u_minus_5	6x8	Shift registers that store 6 consecutive even U values for U' interpolation calculations
v_plus_5, v_plus_3, … v_minus_3, v_minus_5	6x8	Shift registers that store 6 consecutive even V values for V' interpolation calculations
SRAM_we_n, SRAM_write_data, SRAM_address	1, 16, 18	SRAM control signals
read_uv	1	Flag to toggle whether U or V values should be read from the SRAM in the current common case iteration
cc_counter, row_counter	2x8	Keeps track of how many iterations of the common case are done, and how many rows have been completed
m1_end	1	Flag to signal whether all calculations for Milestone 1 are completed

In every 7 clock cycles, the RGB values for 2 pixels are calculated. By considering there are 320x240 pixels, and there are 14 lead in clock cycles and 2 lead out clock cycles for each row, Milestone 1 takes $\frac{320\times240}{2} \times 7 + 16 \times 240 = 272640$ clock cycles to complete. Four multipliers perform 22 multiplications every 7 clock cycles in the common case, so the utilization of the multipliers can then be computed as $\frac{\left(\frac{320\times240}{2}\right)\times7\times\frac{22}{7\times4}}{272640} = 77.46\%$ utilization rate of the multipliers.

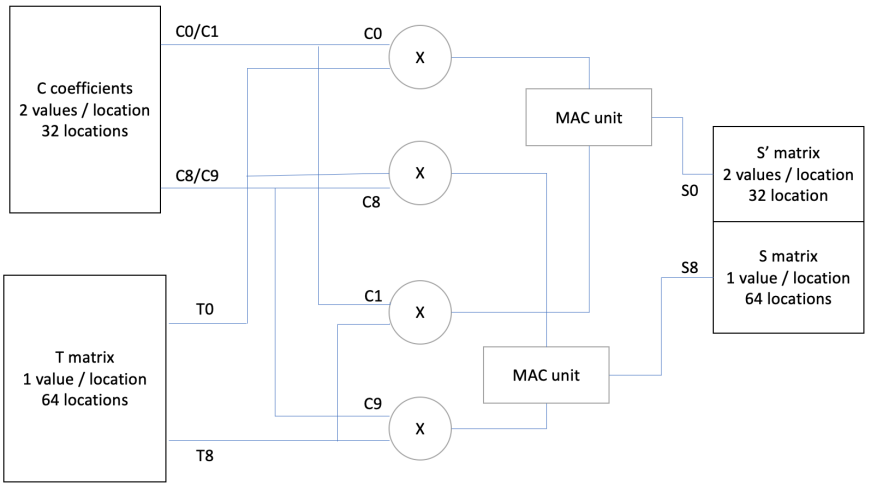
Milestone 2 – Inverse Discrete Cosine Transform

- Latency analysis and utilization calculations

The following figure shows how the matrices are placed in the embedded RAMs and how the multipliers are used in the case when the T matrix is being calculated:



The following figure shows how the matrices are placed in the embedded RAMs and how the multipliers are used in the case when the T matrix is being calculated:



The following table summarizes the register usage in the design of Milestone 2:

Register Name	Bits	Description
mac1, mac2	2x32	MAC units for T and S calculations
row_block, col_block	5, 6	Used to keep track of which block from the SRAM is being processed
sample_counter, fs_counter, T_ram_counter, matrix_counter, s_ram_counter, ct_ws_counter, ws_counter	6, 5, 5, 7, 5, 6, 5	Counters that track the number of values that have been used so far. Specific bits of these counters are also used to perform specific tasks at periodic intervals
s_buffer, s_write_buffer	16, 8	Used to buffer S' values read from SRAM and S values to be written to SRAM

For this design, it takes 68 clock cycles to fetch the first block of S' values, 131 clock cycles to calculate T matrix, 131 clock cycles to calculate the S matrix, and 34 clock cycles to write the last block of S values to the SRAM. For 2400 blocks, the total number of clock cycles required to complete Milestone 2 is $68 + 2400 \times (131 + 131) + 34 = 628902$ clock cycles. The multipliers are used for 128 clock cycles when computing either the T matrix or S matrix since there are 512 multiplications to be done with 4 multipliers. The utilization rate can be calculated as $\frac{2400\times(128+128)}{628902} = 97.69\%$.

Resource Usage and Critical Path

The resource usage of the project as reported by Quartus is 1650 logic elements and 1030 registers. In contrast, the reported number of logic elements was 616 and the number of registers was 369 at the starting point of the project. The increase in the number of logic elements accounts for all of the multipliers and adders in Milestones 1 and 2 that make up the MAC units required for calculations, as well as the logic elements required to implement all of the combinational logic.

In hindsight, we found that the mult1_op1, mult1_op2, mult2_op1, mult2_op2, and mult34_op2 registers were not necessary as their purpose could have been fulfilled through the use of combinational logic. For example, instead of loading a Y value into the mult1_op1 register to be fed into a multiplier, the Y value could have been fed directly into the multiplier without the use of the register through combinational logic. Also, all 6 of the counter registers

were not necessary. Since they serve similar purposes, one of those registers could have been used as a counter and reused instead of using the other counters.

The critical path provided by the Timing Analyzer in Quartus is from the mult2_op2[5] register to the G_odd[31] register in Milestone 1. The data delay of this critical path is 12.209. The reason this is the critical path of the design is likely because the output of the mult2_op2[5] register has to go through a series of logic elements that make up the multiplier and the adder that are included in the MAC unit of the G_odd register.

Weekly Activity and Progress

Week	Ginoth's Contributions	Ahanaf's Contributions	Project progress at the end of the week
1 (Oct 23 – 29)	Reviewed project document	Worked on milestone 1 state table calculation and structure based on project review	Basic understanding of Milestone 1 requirements
2 (Oct 30 – Nov 5)	Designed state table for Milestone 1	Designed the lead out and parts of common case of the state table	State table completed for Milestone 1
3 (Nov 6 – Nov 12)	Coded implementation of Milestone 1 and debugged errors	Tested out Milestone 1 on board in lab and did initial coding and structure	Milestone 1 completed, passed simulations and worked successfully on the DE2 board
4 (Nov 13 – Nov 19)	Conceptualized embedded RAM usage and designed state flow for Milestone 2	Completed state table of Milestone 2 for the fetch S prime and write S. Reviewed feedback on embedded RAM usage and design from Dr. Nicolici	Design for Milestone 2 completed
5 (Nov 20 – Nov 27)	Coded implementation for milestone 2 and started debugging	Worked on embedded RAM design implementation, calculation in coding of Milestone 2 and helped debug partly.	Initial implementation of Milestone 2 coded, but there are still errors in simulation

Conclusion

This project definitely provided a great learning experience. The challenge of designing and implementing a project of this size proved to be an effective way to learn the concepts of hardware design, synthesis and verification since it required us to take some time to think about the problem ourselves. We learned that managing our time throughout the 5 weeks was important in order to stay on track and make significant progress by the deadline. We also found that keeping our thoughts organized was important because it eased solving problems that came up, and helped us avoid getting confused with the many possible choices we had when creating our design. Milestone 1 was the most recently completed milestone, and its commit message is "Milestone 1 works on board" on November 13, 2023.

References

- 3DQ5 Project Description on Avenue to Learn