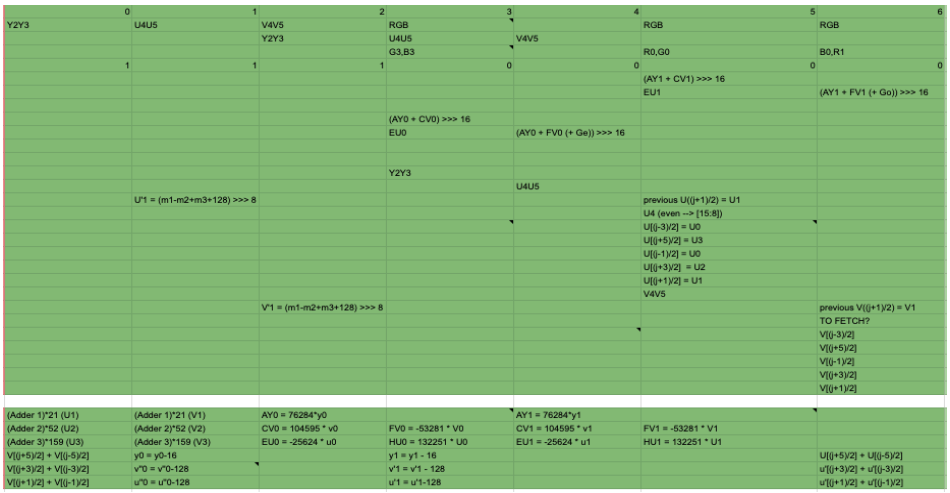
COMPENG 3DQ5
Group Tuesday-23
Sava Jankovic and Prawin Premachandran
Sunday, 24th November 2024

1 - Introduction

This report summarizes our design of a hardware-based image decompression system, with significant resource and efficiency constraints that aid in accomplishing dequantization, signal transformation-IDCT, interpolation and colorspace conversion (CSC). Over the course of five weeks, the project was divided into three separate milestones: decoding of compressed bitstreams to display an image, performing IDCT's on downsampled images, and upsampling along with CSC to final RGB outputs.

Implementation Details

2.1 - Upsampling and Colourspace Conversion (Milestone 1)



Module (instance)	Register names	Bits	Description
milestone 1	u_plus_5, u_minus_5, u_plus_3, u_minus_3, u_plus_1, u_minus_1, v_plus_5, v_minus_5, v_plus_3, v_minus_3, v_plus_1, v_minus_1	9	Registers that are used to hold U and V data that is required for interpolation of odd pixels. There are two sets of six registers, that are used for u and v data.
milestone 1	r_even, r_odd, g_even, g_odd, b_even, b_odd	32	Stores the computed RGB in signed 32 bit values after colorspace conversion.
milestone 1	u_buffer, v_buffer	16	These registers hold the values for u and v data that is read from the SRAM. These are used so they are not completely overwritten when a new value is being stored in its previous location.

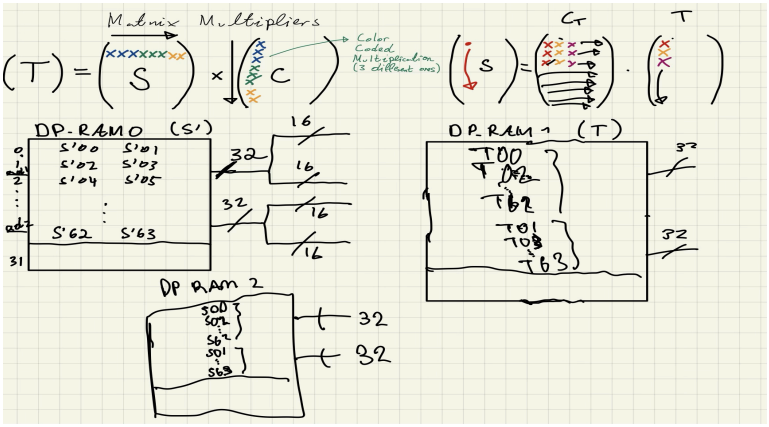
milestone 1	Multi_result_1, Multi_result_2, Multi_result_3	64	Product of multipliers that are used throughout the milestone, especially to implement the interpolation of different signals. Each multiplier is a product of two operand registers that are 32 bits long.
milestone 1	Multi_op_1, Multi_op_2, Multi_op_3, Multi_op_4, Multi_op_5, Multi_op_6,	32	Operands that are used to store values in each state, which get put into the multi result multiplier. They are 32 bits long and feed into the multipliers that are 64 bits long.
milestone 1	adder1, adder2, adder3	32	Registers used for providing multipliers with their values carried over for the interpolation equation and colorspace conversion.
milestone 1	y_counter, uv_counter, rgb_counter, row_counter	18	These four registers are utilized as counters to pitch into our SRAM addresses and have the correct read values throughout the milestone.
milestone 1	Clipped_r_even, clipped_r_odd, clipped_g_even, clipped_g_odd, clipped_b_even, clipped_b_odd	8	These registers are used for holding the clipped data for the RGB values. The values are written back to the SRAM in pairs, having two per location.
milestone 1	U_even_prime, v_even_prime, u_odd_prime,v_odd_prime	32	Registers used for the summation of multiplication results, holding the upsampled odd and even data.

Latency Analysis

Throughout Milestone 1, we had a total of 271441 clock cycles. Since we were using a 50 MHZ clock, this means that each clock cycle lasted a total of 20 ns. In terms of the multipliers that we have used, the maximum that we were allowed was 21 over the course of seven common case clock cycles. Based on our code and the state table, we utilized a total of 16 multiplications. 16/21 = 76% utilization within the common case.

Milestone 2- IDCT

The following shows an outline of how we utilized the different DP RAM's for milestone 2, in particular the two that were used for S' and T value storage.



Additionally, the drawing shows the way we've calculated The T and S values prior to starting to code them (despite not fully completing the S computation in code).

Unfortunately, we were not able to get the entirety of Milestone 2 completed. By the time this report is written, we have successfully integrated Fetch S', Compute T and have been working on the first megastate, which involves Computing S and Fetching S'.

Module (instance)	Register names	Bits	Description

milestone2	Multi_result_1, Multi_result_2, Multi_result_3	64	Product of multipliers that are used throughout the milestone, especially to implement the product of C matrix and S' values when calculating T. Each multiplier is a product of two operand registers that are 32 bits long.
milestone2	s_prime_y_address	18	Contains the address (74800) where fetching S' begins.
milestone2	Concatinate	32	This register essentially combines the two 16-bit values fetched from S', making them eligible to send to the DP RAM address.
milestone2	even_flag	2	This register acts as a flag that lets us know if the value concatenated is even or odd. This ultimately gives an indication of whether the value should be sent to the DP RAM or not.
milestone2	Offset_i, offset_j, block_i, block_j	9	These registers were used to increment, predominantly through fetching S', the 8x8 matrix. The offset_i would go up to 7 (0-7) and would indicate the horizontal position within a matrix, whereas offset_j would indicate the vertical position within a matrix. The blocks i and j indicated the matrix number, and were incremented every time an entire matrix beforehand was read.
milestone2	Multi_op_1, Multi_op_2, Multi_op_3, Multi_op_4, Multi_op_5, Multi_op_6, Multi_result_1, Multi_result_2, Multi_result_3, block_i_mul, offset_i_mul, block_j_mul, increment_j_block	32	Multi operands had a similar purpose to the previous milestone, which was to hold values for multiplication that were executed with multi_results. The registers block_i_mul, offset_i_mul,block_j_mul and increment_j_block were all operands that were a part of our "finder_y" and "finder_uv" registers, which were used to change the correct address when jumping between different rows in the matrix.
milestone2	Finder_y, finder_uv	64	These registers were utilized to fetch certain addresses of the desired rows within a matrix. The finder_y register was used for finding the y values, whereas the finder_uv was used to find the u and v values (which would be different, as they are downsampled).

Resource Usage and Critical Path

The resource usage for the project had a significantly higher use of logic elements compared to the last Lab in the course (Lab 5). For example, we had around 600 logic elements presented in our compilation report for Lab 5, whereas this time just for partial Milestone 2 we had over 700 logic elements. In addition to this, our Milestone 1 code by itself had 2035 logic elements, with 1178 registers. This is predominantly due to much more logic being used and the significant increase in complexity of the project, with many more states being implemented and more registers, such as multipliers and adders.

Milestone 1 critical path for our design was from our **Multi_op_2** to **v_odd_prime,** which resulted in a hefty data delay of 16.064. This path involved upsampling of the u and v prime values and was predominantly due to having to go through multiple multipliers, which led to such a high delay. However, the slack of around 3.845 (and for the

other paths of the same type), has proven to us that the timing requirements have fortunately been met for the milestone.

Albeit our milestone 1 is fully functional, there are improvements that can be made in the future. The major improvement that can be made is the removal of unnecessary registers. In most cases, these registers (such as rect_row_count), were implemented with a different design in mind. However, after the design idea had changed for the milestone, they were not removed despite not being used.

In addition to this, we were not able to obtain an accurate critical path for Milestone 2, due to the milestone being incomplete and only about half of the required states being implemented.

Weekly Activity and Progress

Week # and date	**Project Progress**	**Individual Contributions**
Week 1 (21st Oct)	Project document written and initial ideas for Milestone 1 state table were drafted.	Both team members contributed equally.
Week 2 (28th Oct)	Milestone 1 state table heavily analyzed, edited and provisionally finalized.	Albeit slight absence due to personal reasons from both sides, both individuals contributed equally to the creation of the state table.
Week 3 (4th Nov)	Milestone 1 state table finalized after advice from TA's and the professor. Initial coding started and testbenches initialized.	Both individuals contributed to starting the code and attending the office hours for help.
Week 4 (11th Nov)	Two days spent coding the milestone, the rest was debugging and understanding the mistakes made in the implementation of the state table. Milestone 2 state table started	Both individuals were involved in coding and debugging the milestone equally, which involved visiting the office hour labs on a daily basis.
Week 5 (18th Nov)	Milestone 1 finalized and testbenches passed. Milestone 2 state table finalized and coding began for fetching S and computing T.	Both individuals were responsible for creating the state table for milestone 2. As for coding, Sava created the Fetch S state and Prawin created the Compute T state. All other tasks were done together.

Note: On November 22nd, Prawin requested to work individually due to other course constraints and excess stress experienced in the lab. However, this was overturned the day after both parties had discussed better ways to work around such issues.

Conclusion

In conclusion, throughout the course of the project, we successfully implemented the first milestone (incorporating interpolation and colorspace conversion), where we also met all the given constraints and passed the testbenches. Whilst Milestone 2 proved to be a good learning experience, we were not able to complete it, and have only implemented the fetching S' and computing T. Overall, the project has taught us a lot about hardware solution implementation and image suppression, which is a valuable takeaway for our future careers in the field.

MS1 Finished on 17th November, however please refer to its final self named version pushed on Nov 25. MS2 incomplete and pushed on 25th November. This milestone successfully fetches S' values, writes them to a DP RAM (0), reads them to calculate the T values, and rewrites them to DP RAM (1).

References

Nicolici,N., Thong, J., Kinsman, A., "COE3DQ5 Project Description 2024 Hardware Implementation of an Image Decompressor". Digital Systems Design Course at McMaster University, Canada.