COMPENG 3DQ5: Digital Systems Design Project

Hardware Implementation of an Image Decompressor

Names: Affan Ahmed/400395397/ahmea77, Jeremy Cherian/400381481/cherij2

The primary objective of this project is to design and implement a hardware-based image decompression system focused on reversing the compression procedures outlined by the .mic17 specification. The emphasis is on efficiently transforming and reconstructing images through the stages comprising the three milestones. Milestone 3 begins the decompression process, implementing lossless coding and dequantization. Milestone 2 follows with discrete cosine transformation, and Milestone 1 returns a complete image through interpolation and colour space conversion. The intricacies of the decompression process are explored, where the goal is to reconstruct the original images with minimal information loss. The hardware implementation must be highly detailed to follow such intricacies, leading to an in depth understanding of digital systems design and balancing computational efficiency.

Upsampling and Colour Space Conversion (Milestone 1):

Upsampling and colour space conversion are achieved. *Figure 1* highlights the layout of calculations for interpolation (green) and colour space conversion (red) for the common case. The first 6 common states are used for calculations, and is a 76% usage of multipliers. The interpolation is done and has upscaled values available for RGB calculations in the next cycle. The RGB calculations are written on the next set of common case. *Table 1* covers the registers used in milestone 1 with brief descriptions.

State Name	s1	s2	s3	s4	s5	s6	s7
SRAM_address	Address V(10,11)					Address Y(18,19	Address U(12,13)
SRAM_read_data	Y(16,17)	U(10,11)	V(10,11)				
SRAM_write_data		R12,G12	B12,R13	G13,B13			
SRAM_we_n	1	0	0	0	1	1	1
Yeven	Y14						
U'even	U'14						
V'even	V'14						
Yodd		Y15					
U'odd		U'15					
V'odd		V'15					
multiplier 1	76284*Y'14	(-52)*U7	(159)*U8	(159)*U9	(-52)*U10	(21)*U11	
multiplier 2	76284*Y'15	(-52)*V7	(159)*V8	(159)*V9	(-52)*V10	(21)*V11	
multiplier 3	104595*V'14	(-)25624*U'14	(-)53281*V'14	132251*U'14		(21)*U6	
multiplier 4	104595*V'15	(-)25624*U'15	(-)53281*V'15	132251*U'15		(21)*V6	

Figure 1. Common Case State Table Highlighting Interpolation and CSC Calculation Layout

Module (instance)	Register Name	Bits	Description
milestone1 (Milestone1)	SRAM_address	18	Address register to access external memory to read or write
milestone1 (Milestone1)	SRAM_read_data	16	Holds data read from SRAM_address
milestone1 (Milestone1)	SRAM_write_data	16	Data to be written to SRAM_address stored
milestone1 (Milestone1)	SRAM_we_n	1	Logic low write enable for choosing between read and write at SRAM_address
milestone1 (Milestone1)	M1_Stop	1	Flag to leave M1 state in top state when complete
milestone1 (Milestone1)	M1_Enable	1	Flag to enable M1 to run
milestone1	UPrime_Odd, VPrime_Odd	32	Register accumulates interpolation calculations for upsampling of ODD values
milestone1	Final_UPrime_Odd, Final_VPrime_Odd	32	Holds final upsampling calculation using

			accumulated vals from U/VPrime_Odd reg
milestone1	Final_UPrime_Even, Final_VPrime_Even	32	Holds final interpolation value for upsampling EVEN vals
milestone1	Shift_Count_U Shift_Count_V	8 bits with 6 instances each	Holds U and V values in shift register format for use in calculations
milestone1	data_counterU data_counterV data_counterY data_counterRGB	18	Data counter to increase offset by 1 to go to the next SRAM_address of the required value
milestone1	even_odd_counter	1	Flag for choosing to read EVEN or ODD value in common case
milestone1	row_counter, col_counter	8, 9	Holds value to count how many columns/rows have been completed. Used in logic to enter lead in/lead out and finishing M1.
milestone1	leadoutFlag	18	Increases alongside data_counterY until Y159 is reached in the lead out
milestone1	R/G/B_Even R/G/B_Odd	32	Holds final, to be written RGB values
milestone1	R/G/B_Even_buf, R/G/B_Even_buf2, R/G/B_Odd_buf, R/G/B_Odd_buf2	32	Buf 1 holds accumulated values which are stored in buf 2 to avoid loss of data. Buf 2 values calculated/clipped for final RBG values
milestone1	Mult1/2/3/4_op_1, Mult1/2/3/4_op_2	32	Operands for 2 input multipliers 1-4
milestone1	Mult_result_long1/2/3/4, Mult_result1/2/3/4	64, 32	Long result holds result of multiplication, Mult_result holds final result from 32 LSBs of long result
milestone1	M1State	7	State register used by FSM to control milestone 1 stateflow and data transfers with SRAM.
project	top_state	2	State register used by top module to control stateflow and data transfer between major milestones and modules of the project

Table 1. All Used Registers In Milestone 1 with Descriptions

Milestone 1 takes 272400 clock cycles to complete. This is indicated by the UART_timer wave transitioning from M1 state to Idle state in the top_state. We have 7 clock cycles in the common case. As seen in *Figure 1,* the 4 multipliers are available 1 time per cycle each. Therefore, 28 total multiplier uses are possible. The implementation as seen above uses 22 out of the 28 available instances. This reaches a multiplier usage of 76%, surpassing the 75% multiplier usage constraint.

Resource Usage and Critical Path:

On completion of Milestone 1, the total logic element usage is 2294/114480, with a 2% usage rate. The number of registers is 1105, which is accurate to our estimation. In lab 5 experiment 4, there was a logic element usage of <1% for 616/114480 elements used. The number of registers used was 369 in lab. This makes sense because of the stark difference in complexity

between the labs and the milestone. There are some extra registers used as there was some uncertainty regarding number of bits needed to represent certain values without there being a loss of information. This includes various counters (data_counterY, U and V), all using more elements than required. Furthermore, we created far more lead out cases than realistically required. Many of these cases simply serve as a common case with a few alterations made. These could have been shortened into a multiplexer within the common case itself to lower the amount of required lead out states. When examining our critical path, we identify our slack to be 3.604 nanoseconds and a maximum frequency of 60.99 MHz. This critical path is from Mult4_op_1[16] to Final_VPrime_Odd[23]. The positive slack indicates that the design meets timing requirements by having extra time available along this critical path. This makes sense as our lowest slack value because this path of data transfer is the most intensive. First the operand is set for interpolation, then it multiplied, then accumulated several times with other VPrime_Odd interpolation calculations, and finally set to the Final_VPrime_Odd.

Inverse Discrete Cosine Transform (Milestone 2):

Figure 2. Milestone 2 Matrices and Multiplier Description within State Table

Module (instance)	Register Name	Bits	Description
milestone2 (Milestone2)	SRAM_address	18	Address to access external memory to read or write
milestone2 (Milestone2)	SRAM_read_data	16	Data read from SRAM
milestone2 (Milestone2)	SRAM_write_data	16	Data to be written to SRAM
milestone2 (Milestone2)	SRAM_we_n	1	SRAM act. LO write enable
milestone2 (Milestone2)	M2_Stop	1	Flag to leave M2 when done
milestone2 (Milestone2)	M2_Enable	1	Flag to enable M2 to run
milestone2 (DRAM_inst0, DRAM_inst1, DRAM_inst2)	address_1/2/3/4/5/6	7	Address 1, 2 for DRAM 1. Address 3, 4 for DRAM 2. Address 5, 6 for DRAM 3.
milestone2 (DRAM_inst0, DRAM_inst1, DRAM_inst2)	write_data_a/b	32 bits / DRAM (3 DRAMs)	Writing data for DRAM: a for port 1 of DRAM, b for port 2 of DRAM
milestone2 (DRAM_inst0, DRAM_inst1, DRAM_inst2)	write_enable_a/b	1 * 3 inst. (each DRAM)	Act. HI wr_en for DRAMs

milestone2 (DRAM_inst0, DRAM_inst1, DRAM_inst2)	read_data_a/b	32 bits / DRAM (3 DRAM)	Hold read data from DRAM
milestone2	col_counter, row_counter, T_col/row_counter	5	Counts columns/rows
milestone2	read_data_counter	18	Offsets SRAM_address by 1 every read
milestone2	YUV_counter_read	2	Changes YUV SRAM_address offset for S' reading
milestone2	start_offset	18	Changes YUV offset for S' reading
milestone2	S_prime_buffer	16	Buffer for S' to not overwrite
milestone2	S_prime_counter	16 * 4 inst.	Counter to read the same 4 S' slots for calc. T
milestone2	coeff	8 * 4 inst.	Stores C matrix LUT

Table 2. Registers In Milestone 2 with Descriptions

Weekly Activity and Progress:

Week	Project Progress	Contributions
Week 1	Introduction, reading and understanding project document, starting state table. Conferring with TAs and professor about M1 objectives.	Both members
Week 2	Completed state table, starting, and finishing code for lead in. Sought clarification with Tas about if state table is correct.	Both members
Week 3	Editing state table to fit unforeseen requirements in code related to timing. Confirmed with TA. Finished coding common case and lead out. Began debugging.	Both members
Week 4	Debugging confirmed working lead in and common case. Spoke with TA about efficient debugging strategy. Debugging and editing lead out. Completed Milestone 1. Start M2 state tables.	Both members
Week 5	Written the report and wrapping up the milestone 2 draft code.	Both members

Table 3. Weekly Activity and Progress Table

Engaging in this project provided a valuable learning experience in the realm of hardware design with a hands-on and satisfying project. It provided the opportunity to gain true applicable experience in our field of interest. While the project is not fully completed, the challenges encountered during the various stages, from conceptualization to critical path analysis, offered insights into the true complexities of designing and implementing hardware-based solutions. Despite the unfinished product, we believe we learned a lot. Reading and understanding the project document, to conceptualizing a design via a state table, to drafting code, and then finally debugging slowly towards a solution was an arduous yet satisfying process. At times it felt tedious, but we always felt as though we were slowly making progress with the project as well as with our learning. As opposed to other projects we may encounter, this project encompasses every aspect of the design process with no shortcuts given and we feel that we learned a lot of valuable lessons.

Milestone 2 code does not work and renders the Milestone 1 code useless. The final commit pushed to GitHub that has a working Milestone 1 is on November 23rd, 2023, 5:45 PM EST with commit message: "VGA works :)))".

References:

[1] A. Kinsman, J. Thong, N. Nicolici, "COE3DQ5 Project Description 2023 Hardware Implementation of an Image Decompressor," Digital Systems Design Course at McMaster University, [Online], November 27 2023. Available: https://avenue.cllmcmaster.ca/d2l/le/content/554050/viewContent/4381835/View