Lab 1 Project Report

Ethan Keshishian

CS M152A Lab1

Introduction:

The goal of Lab 1 is to convert a 13 bit two's complement input into the equivalent floating point output. This is done with Verilog, to implement this with hardware. The design accepts a 13 bit variable D, and outputs the MSB S, the exponent E, and the significand (or mantissa) F, which are one bit, 3 bits, and 5 bits, respectively. My design uses a top module with some always blocks that calculate the output while handling edge cases. Edge cases are handled by unit tests in the testbench. The program is implemented into hardware design, and the final design is error-free.

Module design:

The module is built to convert 13 bit binary input to the equivalent floating point encoding. The module accepts input D[12:0] and outputs the MSB S, the exponent E, and the significand (or mantissa) F, which are one bit, 3 bits, and 5 bits, respectively.

Since S depends only on the most significant bit, the module first checks what the MSB is and outputs it as S. This is equivalent to D[12] in Verilog. This is within its own always block.

The next section of the module is an always block that uses the MSB to determine if the input should be inverted. If the MSB is 0, the input is positive and no conversion is necessary. If the MSB is 1, the input is negative, and the input is inverted and incremented by 1. The resulting value in either case is stored in the "converted" register variable. There is an edge case here if the input is 13'b1_0000_0000_0000; converting this value would return 13'b1_0000_0000_0000, the same value. This edge case is handled explicitly by setting the converted register to 13'b0_1111_1111_1111, which is handled correctly by the next always block. Trying to directly set E and F would have resulted in a race condition.

The next section is the final section that handles the output of E and F. This always block is one section because of possible edge cases, and because handling rounding can alter E and F. First, the number of leading zeros are counted with a for loop, and stored in the leadingZeros register. If the number of leading zeros is greater than 7, then no rounding is necessary, and E is set to 0 and F is the final 5 bits of the converted register. Otherwise, rounding is necessary. E is set based on the number of leading zeros and F is set based on the next 5 values. If the next (6th) bit is 0, this is the final output. If it is 1, then F is incremented. However, if F is the maximum value of 11111, then it is incremented and shifted right by 1, and E is incremented. However, if E is the max value, the final output for E and F are set to their respective largest values. This is handled in the block with nested if statements.

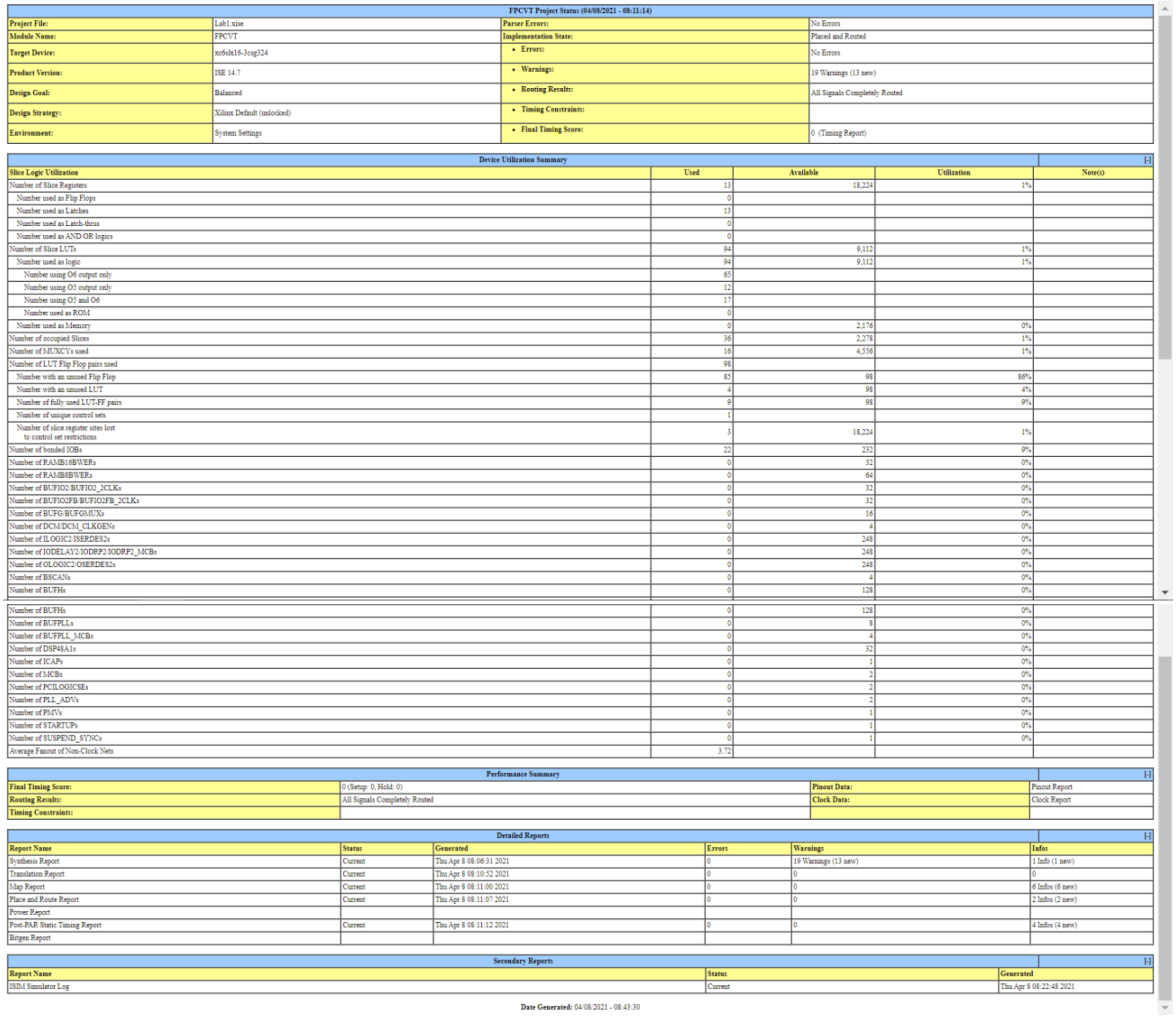
Testbench design:

My testbench unit tests the module by passing multiple different inputs for D. Within these test cases, I have typical inputs and edge cases. The edge cases include testing the case 13'b1_0000_0000_0000, 13'b1_0000_0000_0001 (which becomes 1_1111_1111_1111 after conversion), etc. The module waits before testing each value by 100 time units. I confirmed that the module was working correctly by examining the waveform output, which is provided below.

Conclusion:

Overall, my implementation uses three always blocks that are in charge of handling different sections of the conversion. It outputs S, E, and F, while also handling potential rounding errors that may occur. There were multiple difficulties I ran into with the project. I had trouble understanding how always blocks worked, and ended up having conflicts within each block. While my completed program displayed the correct waveform output, when I tried to create the hardware implementation, there were new errors. This is because there are errors that are specific to the hardware implementation, which was a surprise to me. An example of this is the loop which counts the number of leading zeros. At first, I had the loop as a while loop, but it wouldn’t compile. Verilog requires a set number of iterations per loop, so I had to refactor my code into a for loop and obtain the correct values a different way. Another problem I ran into was dealing with the first edge case 13’b1_0000_0000_0000. I was trying to set E and F directly in this case, but because I factored my code into multiple always blocks, this was causing a race condition. I had to use a clever workaround by setting the converted register directly to 13’b0_1111_1111_1111, which is correctly handled. Overall, I enjoyed the project, and found myself learning a lot from the challenges I faced.

ISE Design Overview Summary Report:



FPCVT Project Status (04/08/2021 - 08:11:14)			
Project File:	Lab1.xise	**Parser Errors:**	No Errors
Module Name:	FPCVT	**Implementation State:**	Placed and Routed
Target Device:	xc6slx16-3csg324	• **Errors:**	No Errors
Product Version:	ISE 14.7	• **Warnings:**	19 Warnings (13 new)
Design Goal:	Balanced	• **Routing Results:**	All Signals Completely Routed
Design Strategy:	Xilinx Default (unlocked)	• **Timing Constraints:**	
Environment:	System Settings	• **Final Timing Score:**	0 (Timing Report)

Device Utilization Summary				[-]
Slice Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Registers	13	18,224	1%	
Number used as Flip Flops	0			
Number used as Latches	13			
Number used as Latch-thrus	0			
Number used as AND/OR logics	0			
Number of Slice LUTs	94	9,112	1%	
Number used as logic	94	9,112	1%	
Number using O6 output only	65			
Number using O5 output only	12			
Number using O5 and O6	17			
Number used as ROM	0			
Number used as Memory	0	2,176	0%	
Number of occupied Slices	36	2,278	1%	
Number of MUXCYs used	16	4,556	1%	
Number of LUT Flip Flop pairs used	98			
Number with an unused Flip Flop	85	98	86%	
Number with an unused LUT	4	98	4%	
Number of fully used LUT-FF pairs	9	98	9%	
Number of unique control sets	1			
Number of slice register sites lost to control set restrictions	3	18,224	1%	
Number of bonded IOBs	22	232	9%	
Number of RAMB16BWERs	0	32	0%	
Number of RAMB8BWERs	0	64	0%	
Number of BUFIO2/BUFIO2_2CLKs	0	32	0%	
Number of BUFIO2FB/BUFIO2FB_2CLKs	0	32	0%	
Number of BUFG/BUFGMUXs	0	16	0%	
Number of DCM/DCM_CLKGENs	0	4	0%	
Number of ILOGIC2/ISERDES2s	0	248	0%	
Number of IODELAY2/IODRP2/IODRP2_MCBs	0	248	0%	
Number of OLOGIC2/OSERDES2s	0	248	0%	
Number of BSCANs	0	4	0%	
Number of BUFHs	0	128	0%	
Number of BUFHs	0	128	0%	
Number of BUFPLLs	0	8	0%	
Number of BUFPLL_MCBs	0	4	0%	
Number of DSP48A1s	0	32	0%	
Number of ICAPs	0	1	0%	
Number of MCBs	0	2	0%	
Number of PCILOGICSEs	0	2	0%	
Number of PLL_ADVs	0	2	0%	
Number of PMVs	0	1	0%	
Number of STARTUPs	0	1	0%	
Number of SUSPEND_SYNCs	0	1	0%	
Average Fanout of Non-Clock Nets	3.72			

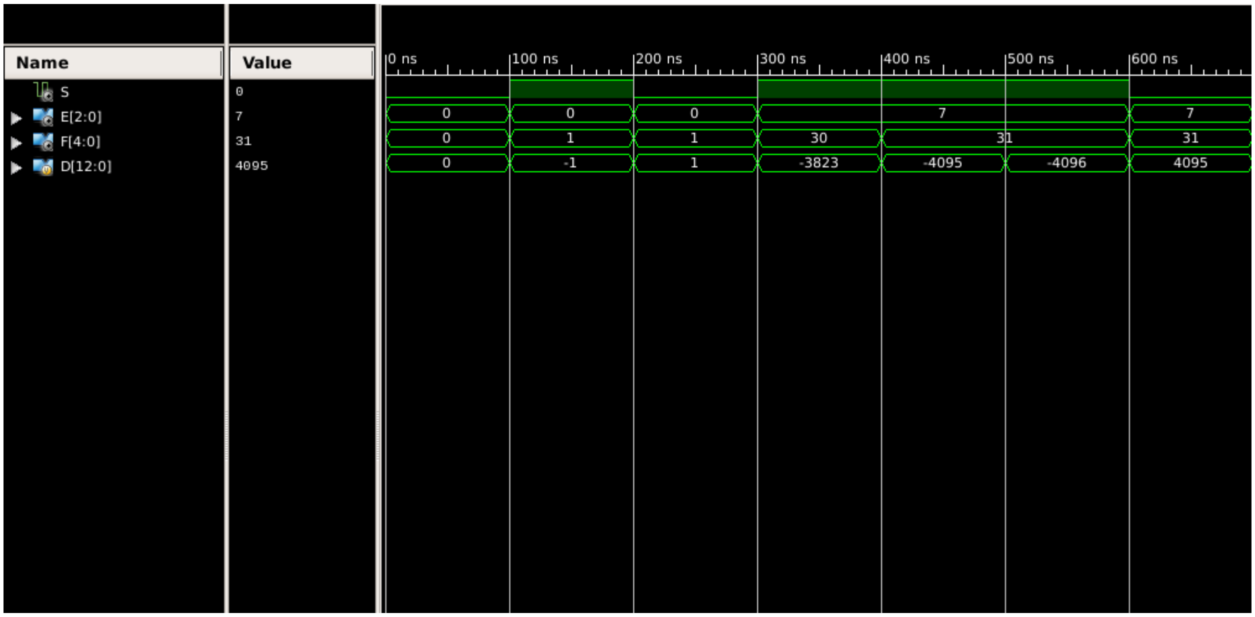
Performance Summary			[-]
Final Timing Score:	0 (Setup: 0, Hold: 0)	**Pinout Data:**	Pinout Report
Routing Results:	All Signals Completely Routed	**Clock Data:**	Clock Report
Timing Constraints:			

Detailed Reports					[-]
Report Name	**Status**	**Generated**	**Errors**	**Warnings**	**Infos**
Synthesis Report	Current	Thu Apr 8 08:06:31 2021	0	19 Warnings (13 new)	1 Info (1 new)
Translation Report	Current	Thu Apr 8 08:10:52 2021	0	0	0
Map Report	Current	Thu Apr 8 08:11:00 2021	0	0	6 Infos (6 new)
Place and Route Report	Current	Thu Apr 8 08:11:07 2021	0	0	2 Infos (2 new)
Power Report					
Post-PAR Static Timing Report	Current	Thu Apr 8 08:11:12 2021	0	0	4 Infos (4 new)
Bitgen Report					

Secondary Reports		[-]
Report Name	**Status**	**Generated**
ISIM Simulator Log	Current	Thu Apr 8 08:22:48 2021

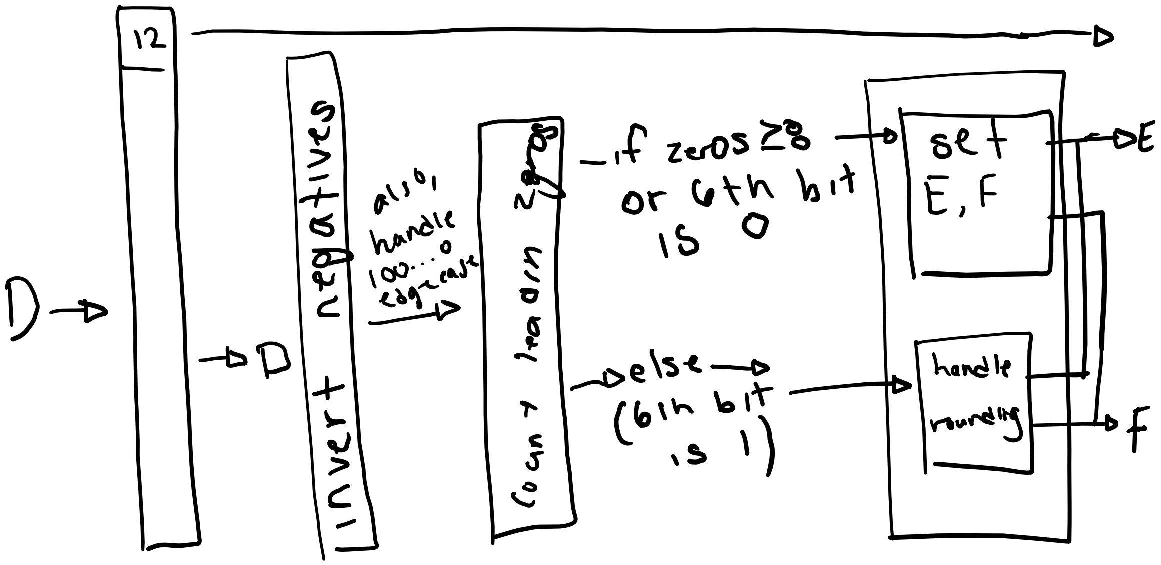
Date Generated: 04/08/2021 - 08:43:30

Simulation Output:



Name	Value
S	0
E[2:0]	7
F[4:0]	31
D[12:0]	4095

Hand-drawn Design Schematic:


12
D
D
invert negatives
also, handle 100...0 edge case
count leading zeros
if zeros ≥8 or 6th bit is 0
else (6th bit is 1)
set E, F
handle rounding
E
F