CS M152A Lab1 : Compression to Floating Point Representation

Name: Ashley Zhu

UID: 505308582

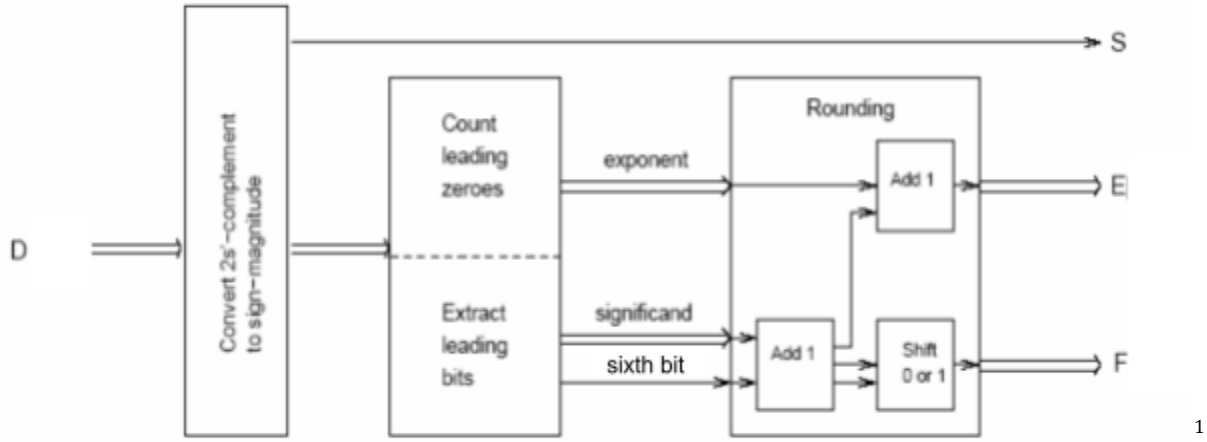
Lab Section 7

TA: Uneeb Rathore

Overview

In this lab, we were tasked to utilize modules to take a 13-bit Two's Complement Number and convert it into its closest floating point representation number. The output of the synthesized program includes a sign bit (0 or 1) representing whether the number is negative or not, a 3-bit exponential section (0 to 7) of the base, and a 5-bit mantissa for base value. Since the conversion format includes a transfer of information from a data format with more information to one with less, there will be data loss which leads to the necessity of **rounding** to the nearest floating point number.

Procedure



[1]

While the diagram on the specification was helpful as a starting point, it complicated a couple of things such as extracting the mantissa. Therefore although the procedure follows heavily from this diagram, the implementation of each portion differs.

[1] Fig. Diagram of a potential approach to the problem from lab specification

Conversion from Two's Complement to Sign Magnitude

Two's Complement format requires that the most significant bit be the negative of 2 to the power of however many bits there are, and the rest of the bits be the positive version of 2 to the power of the index. In order to convert from Two's Complement, all that needs to be done is extract the signed bit, invert the number, and add 1.

The return value from this first **SignedMagnitude** module is an array of size 14 (1 larger than the original value) that contains the sign as its most significant bit, and the magnitude of the value in the proceeding bits. The reason why I chose to utilize this format is because if the need arises to store 4096 in magnitude, it will be clear whether the value is -4096 or positive 4096 (although this shouldn't happen). The way this was done was by checking the existence of a sign and performing separate logic depending on whether the number was negative or not. Some potential cases I considered are displayed in the table below.

Case (13-bit)	Result (14-bit)
1 0000 0000 0000	11 0000 0000 0000
0 1111 1111 1111	00 1111 1111 1111
1 1111 1111 1111	10 0000 0000 0001
0 0000 0000 0000	00 0000 0000 0000
0 0000 0000 0001	00 0000 0000 0001

Granted this method of conversion may end up taking extra space, I think it is important that the clarity of whether the number is negative or positive should come first.

Implementing a Priority Encoder

The next step of compression required that information about the number be extracted from the newly created signed magnitude format. Initially, I wanted to try and use for loops to incrementally determine how many leading zeros there were, but I soon discovered that it was a hassle and I could even more easily just implement a **PriorityEncoder** that takes the case of 8 or more leading zeros as the first check all the way down to 1 leading zero as the final check. In this way, all of the cases of a number are taken care of without having to deal with messy indexing logic with regards to for loops. Some potential problem cases are found below, including the problem of how to represent the number -4096 in floating point. -4096 isn't able to be replicated in a 9 bit floating point representation, and its closest neighbor is -3968. Therefore when a linear encoding like 11 0000 0000 0000 is presented,

rather than setting the exponent and mantissa to 8 and 1 0000, we need to fit the bounds of the registers and adjust to -3968.

Value	Linear Encoding	S, E, F
-4096	11 0000 0000 0000	1, 7, 31
4095	00 1111 1111 1111	0, 7, 31
-422	1_1110_0101_1010	1, 26, 4
-1	10 0000 0000 0001	1, 0, 1
0	00 0000 0000 0000	0, 0, 0
1	00 0000 0000 0001	0, 0, 1

Rounding

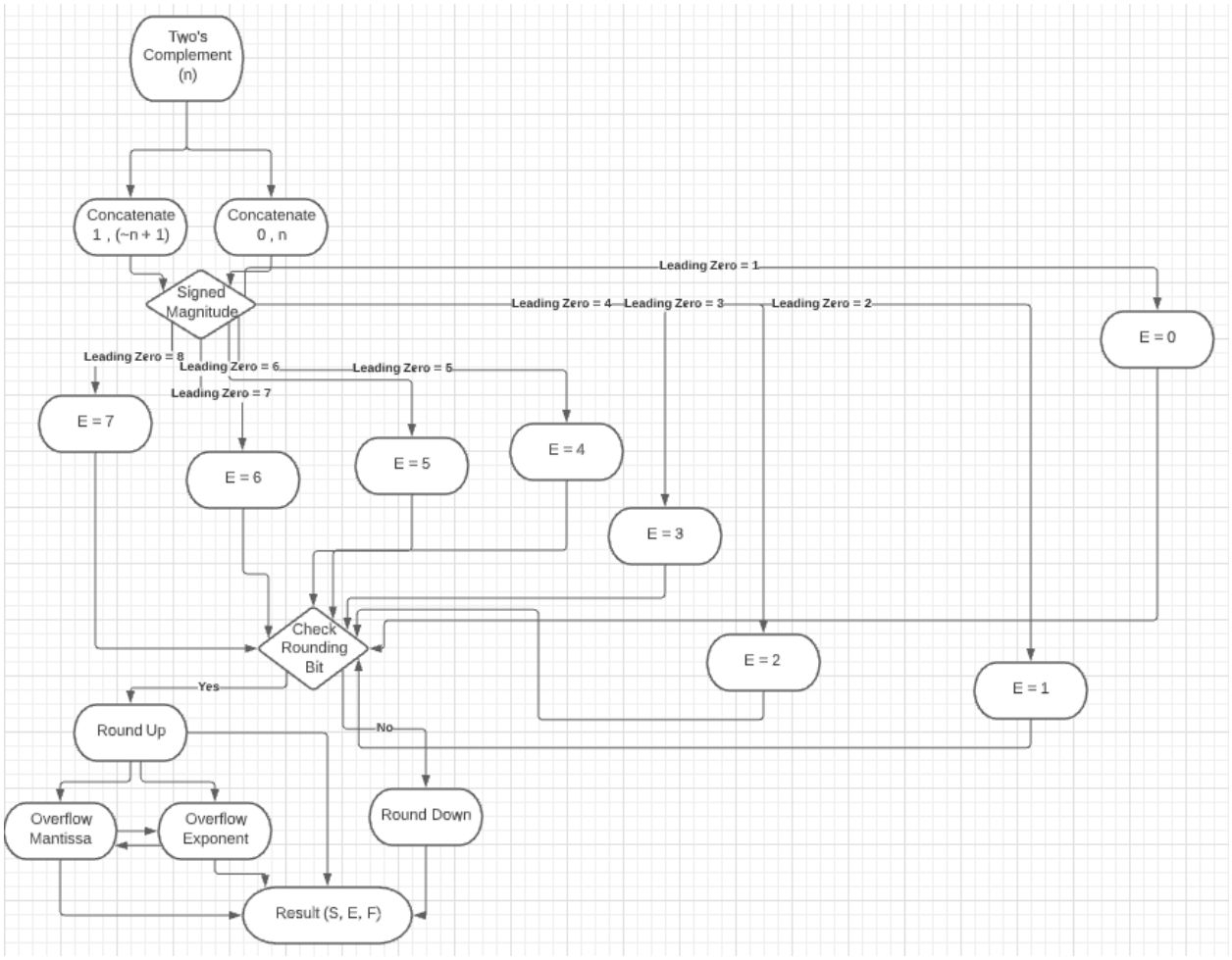
The last portion of the project was making sure that if a number did not have a precise representation in floating point, that it was matched with the floating point number that was closest in value to it, in other words, rounding! The **Round** module observes the 6th bit after extracting the mantissa from the linear encoding. If the rounding bit is 1, the number should be rounded up. If the rounding bit is 0, the number should be rounded down.

Value	Rounding Bit (6th Bit)	Rounded
-4096	0	Force Rounded Up
4095	1	Force Rounded Down
-422	1	Up
108	0	Down
109	0	Down
110	1	Up
111	1	Up

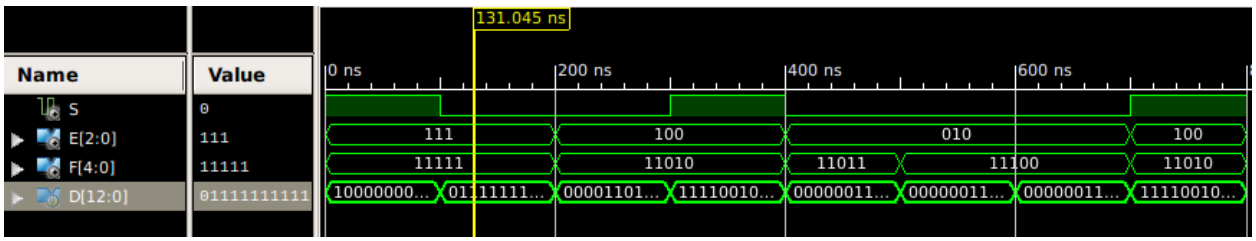
Any time the mantissa overflowed from rounding, the entire value would be shifted right and the exponent would increase by 1. In the case where the exponent was also full, the maximal possible floating point representation with the given amount of bits was presented as the final answer instead. If the number was rounded down, the value was unaltered.

Visualization

A full visualization of the process can be seen below:



Example Simulations



Test Bench Design

The test bench simply creates an instance of the FPCVT module which creates instances of the SignedMagnitude, PriorityEncoder, and Round modules in order to operate on wires and output the final floating point representation.

ISE Design Overview Summary Report

Adder Project Status			
Project File:	m152A_lab1.xise	**Parser Errors:**	No Errors
Module Name:	Adder	**Implementation State:**	Synthesized
Target Device:	xc6slx16-3csg324	**• Errors:**	No Errors
Product Version:	ISE 14.7	**• Warnings:**	No Warnings
Design Goal:	Balanced	**• Routing Results:**	
Design Strategy:	Xilinx Default (unlocked)	**• Timing Constraints:**	
Environment:	System Settings	**• Final Timing Score:**	

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice LUTs	17	9112	0%
Number of fully used LUT-FF pairs	0	17	0%
Number of bonded IOBs	50	232	21%
Number of BUFG/BUFGCTRLs	1	16	6%

Detailed Reports					[-]
Report Name	**Status**	**Generated**	**Errors**	**Warnings**	**Infos**
Synthesis Report	Current	Tue Apr 13 12:55:00 2021	0	0	0
Translation Report					
Map Report					
Place and Route Report					
Power Report					
Post-PAR Static Timing Report					
Bitgen Report					

Secondary Reports			[-]
Report Name	**Status**	**Generated**	
ISIM Simulator Log	Current	Mon Apr 19 14:24:49 2021	

Date Generated: 04/19/2021 - 14:29:19

Synthesis Report

Release 14.7 - xst P.20131013 (lin64)

Copyright (c) 1995-2013 Xilinx, Inc. All rights reserved.

-->

Parameter TMPDIR set to xst/projnav.tmp

Total REAL time to Xst completion: 0.00 secs
Total CPU time to Xst completion: 0.04 secs

-->
Parameter xsthdpdir set to xst

Total REAL time to Xst completion: 0.00 secs
Total CPU time to Xst completion: 0.04 secs

-->
Reading design: Adder.prj

TABLE OF CONTENTS

===
*   Synthesis Options Summary   *
===
---- Source Parameters
Input File Name   : "Adder.prj"
Ignore Synthesis Constraint File   : NO

---- Target Parameters
Output File Name   : "Adder"
Output Format   : NGC

Target Device : xc6slx16-3-csg324

---- Source Options
Top Module Name : Adder
Automatic FSM Extraction : YES
FSM Encoding Algorithm : Auto
Safe Implementation : No
FSM Style : LUT
RAM Extraction : Yes
RAM Style : Auto
ROM Extraction : Yes
Shift Register Extraction : YES
ROM Style : Auto
Resource Sharing : YES
Asynchronous To Synchronous : NO
Shift Register Minimum Size : 2
Use DSP Block : Auto
Automatic Register Balancing : No

---- Target Options
LUT Combining : Auto
Reduce Control Sets : Auto
Add IO Buffers : YES
Global Maximum Fanout : 100000
Add Generic Clock Buffer(BUFG) : 16
Register Duplication : YES
Optimize Instantiated Primitives : NO
Use Clock Enable : Auto
Use Synchronous Set : Auto
Use Synchronous Reset : Auto
Pack IO Registers into IOBs : Auto
Equivalent register Removal : YES

---- General Options
Optimization Goal : Speed
Optimization Effort : 1
Power Reduction : NO
Keep Hierarchy : No
Netlist Hierarchy : As_Optimized
RTL Output : Yes
Global Optimization : AllClockNets
Read Cores : YES
Write Timing Constraints : NO
Cross Clock Analysis : NO
Hierarchy Separator : /
Bus Delimiter : <>

```
Case Specifier                  : Maintain
Slice Utilization Ratio         : 100
BRAM Utilization Ratio          : 100
DSP48 Utilization Ratio         : 100
Auto BRAM Packing               : NO
Slice Utilization Ratio Delta   : 5

=========================================================================


=========================================================================
*                     HDL Parsing                       *
=========================================================================
Analyzing Verilog file "/home/ashley/Documents/M152A/m152A_lab1/Adder.v" into library work
Parsing module <Adder>.

=========================================================================
*                     HDL Elaboration                   *
=========================================================================

Elaborating module <Adder>.

=========================================================================
*                     HDL Synthesis                     *
=========================================================================

Synthesizing Unit <Adder>.
    Related source file is "/home/ashley/Documents/M152A/m152A_lab1/Adder.v".
    Found 16-bit register for signal <Z>.
    Found 16-bit adder for signal <IV_1> created at line 36.
    Summary:
        inferred   1 Adder/Subtractor(s).
        inferred  16 D-type flip-flop(s).
Unit <Adder> synthesized.

=========================================================================
HDL Synthesis Report

Macro Statistics
# Adders/Subtractors                    : 1
 16-bit adder                          : 1
# Registers                             : 1
 16-bit register                       : 1

=========================================================================
```

```
=========================================================================
*                  Advanced HDL Synthesis                  *
=========================================================================


=========================================================================
Advanced HDL Synthesis Report

Macro Statistics
# Adders/Subtractors                    : 1
 16-bit adder                          : 1
# Registers                             : 16
 Flip-Flops                            : 16

=========================================================================

=========================================================================
*                  Low Level Synthesis                  *
=========================================================================

Optimizing unit <Adder> ...

Mapping all equations...
Building and optimizing final netlist ...
Found area constraint ratio of 100 (+ 5) on block Adder, actual ratio is 0.

Final Macro Processing ...

=========================================================================
Final Register Report

Macro Statistics
# Registers                             : 16
 Flip-Flops                            : 16

=========================================================================

=========================================================================
*                  Partition Report                  *
=========================================================================

Partition Implementation Status
-------------------------------

  No Partitions were found in this design.
```

===
*   Design Summary   *
===

Top Level Output File Name   : Adder.ngc

Primitive and Black Box Usage:

BELS   : 49
GND   : 1
INV   : 1
LUT2   : 16
MUXCY   : 15
XORCY   : 16
FlipFlops/Latches   : 16
FDR   : 16
Clock Buffers   : 1
BUFGP   : 1
IO Buffers   : 49
IBUF   : 33
OBUF   : 16

Device utilization summary:

Selected Device : 6slx16csg324-3

Slice Logic Utilization:
 Number of Slice LUTs: 17 out of   9112   0%
    Number used as Logic: 17 out of   9112   0%

Slice Logic Distribution:
 Number of LUT Flip Flop pairs used: 17
   Number with an unused Flip Flop: 17 out of   17   100%
   Number with an unused LUT: 0 out of   17   0%
   Number of fully used LUT-FF pairs: 0 out of   17   0%
   Number of unique control sets: 1

IO Utilization:
 Number of IOs: 50
 Number of bonded IOBs: 50 out of   232   21%
    IOB Flip Flops/Latches: 16

Specific Feature Utilization:
 Number of BUFG/BUFGCTRLs: 1 out of   16   6%

Partition Resource Summary:

  No Partitions were found in this design.

===
Timing Report

NOTE: THESE TIMING NUMBERS ARE ONLY A SYNTHESIS ESTIMATE.
      FOR ACCURATE TIMING INFORMATION PLEASE REFER TO THE TRACE REPORT
      GENERATED AFTER PLACE-and-ROUTE.

Clock Information:

Clock Signal | Clock buffer(FF name) | Load |
---
clk | BUFGP | 16 |

Asynchronous Control Signals Information:

No asynchronous control signals found in this design

Timing Summary:

Speed Grade: -3

   Minimum period: No path found
   Minimum input arrival time before clock: 3.441ns
   Maximum output required time after clock: 3.597ns
   Maximum combinational path delay: No path found

Timing Details:

All values displayed in nanoseconds (ns)

===
Timing constraint: Default OFFSET IN BEFORE for Clock 'clk'

```
  Total number of paths / destination ports: 408 / 32
-------------------------------------------------------------------------
Offset:             3.441ns (Levels of Logic = 2)
  Source:           reset (PAD)
  Destination:      Z_0 (FF)
  Destination Clock: clk rising

  Data Path: reset to Z_0
                        Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             1   1.222   0.579  reset_IBUF (reset_IBUF)
     INV:I->O             16   0.206   1.004  reset_inv1_INV_0 (reset_inv)
     FDR:R                     0.430          Z_0
    ----------------------------------------
    Total                      3.441ns (1.858ns logic, 1.583ns route)
                               (54.0% logic, 46.0% route)

=========================================================================
Timing constraint: Default OFFSET OUT AFTER for Clock 'clk'
  Total number of paths / destination ports: 16 / 16
-------------------------------------------------------------------------
Offset:             3.597ns (Levels of Logic = 1)
  Source:           Z_15 (FF)
  Destination:      Z<15> (PAD)
  Source Clock:     clk rising

  Data Path: Z_15 to Z<15>
                        Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     FDR:C->Q              1   0.447   0.579  Z_15 (Z_15)
     OBUF:I->O                 2.571          Z_15_OBUF (Z<15>)
    ----------------------------------------
    Total                      3.597ns (3.018ns logic, 0.579ns route)
                               (83.9% logic, 16.1% route)

=========================================================================

Cross Clock Domains Report:
--------------------------

=========================================================================


Total REAL time to Xst completion: 8.00 secs
```

Total CPU time to Xst completion: 3.77 secs

-->

Total memory usage is 377028 kilobytes

Number of errors : 0 (0 filtered)
Number of warnings : 0 (0 filtered)
Number of infos : 0 (0 filtered)

ISIM Log Report

ISim log file
Running: /home/ashley/Documents/M152A/m152A_lab1/FPCVT_tb_isim_beh.exe -intstyle ise -gui -tclbatch isim.cmd -wdb /home/ashley/Documents/M152A/m152A_lab1/FPCVT_tb_isim_beh.wdb
INFO: There is another simulation running in the same directory. Using database file name isim1.wdb.
ISim P.20131013 (signature 0xfbc00daa)
WARNING: A WEBPACK license was found.
WARNING: Please use Xilinx License Configuration Manager to check out a full ISim license.
WARNING: ISim will run in Lite mode. Please refer to the ISim documentation for more information on the differences between the Lite and the Full version.
This is a Lite version of ISim.
Time resolution is 1 ps
onerror resume
wave add /
run 1000 ns
Simulator is doing circuit initialization process.
Finished circuit initialization process.
1_0000_0000_00004294963328
0_1111_1111_1111 3968
0_0001_1010_0110 416
1_1110_0101_10104294966880
0_0000_0110_1100 108
0_0000_0110_1110 112
0_0000_0110_1111 112
1_1110_0101_10104294966880

Stopped at time : 800 ns : File "/home/ashley/Documents/M152A/m152A_lab1/FPCVT_tb.v" Line 92