COM SCI M152A Lab 2

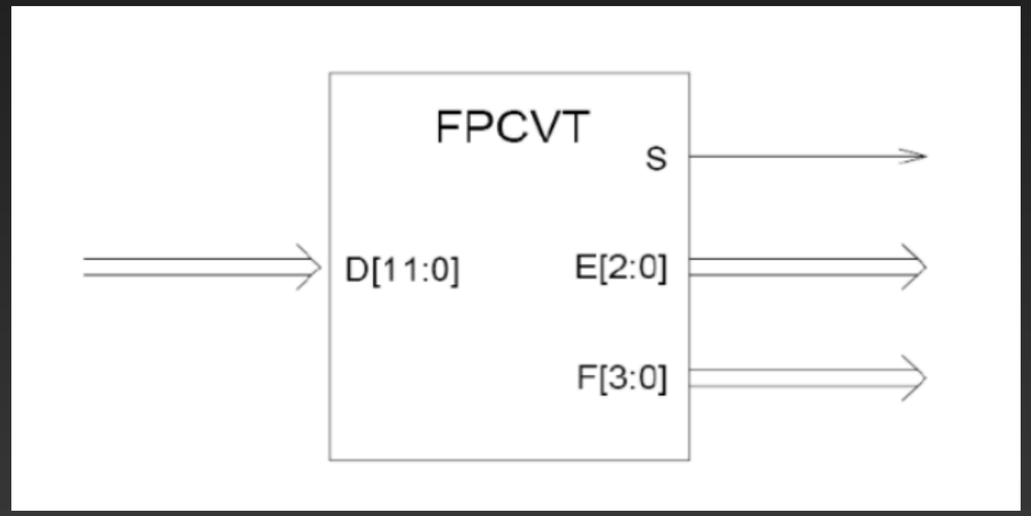
ez money

Samuel Lee - 04/13/22

Floating Point Conversion Lab

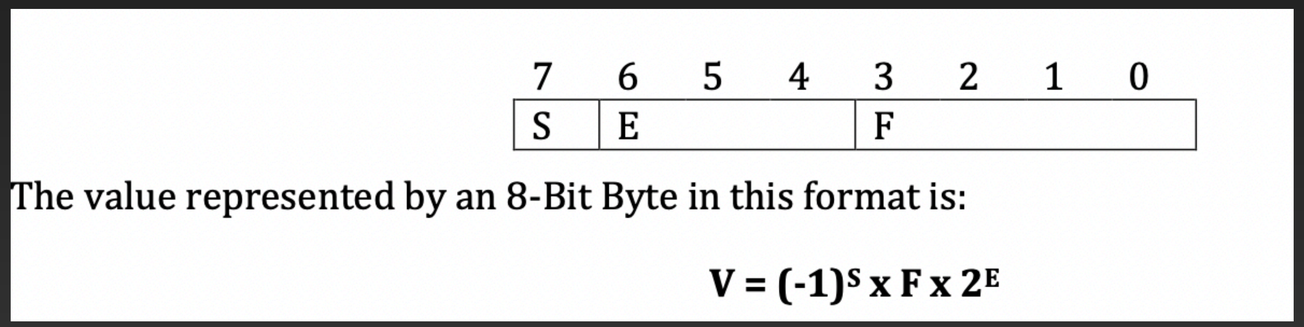
Lab Overview

- Convert a 12-bit linear encoding into a compounded 8-bit FP encoding
- FPGA is not required (simulation only)



Lab Background

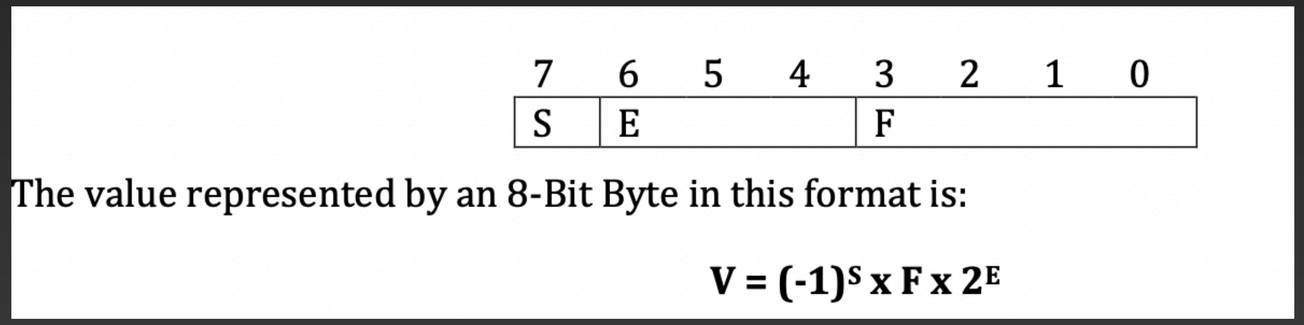
- Two’s complement conversion into 1-Bit Sign Representation, 3-Bit Exponent, 4-Bit Significand
- Say the Sign-Bit is 1, the Exponent is 001, and the Significand is 0101. What is the value?



How to Convert

Steps

1. The sign can be extracted from the most significant bit of the input.
2. The exponent can be found by counting the leading zeroes.
3. The significand is the first 4 bits after the leading zeroes.



Leading Zeroes	Exponent
1	7
2	6
3	5
4	4
5	3
6	2
7	1
≥ 8	0

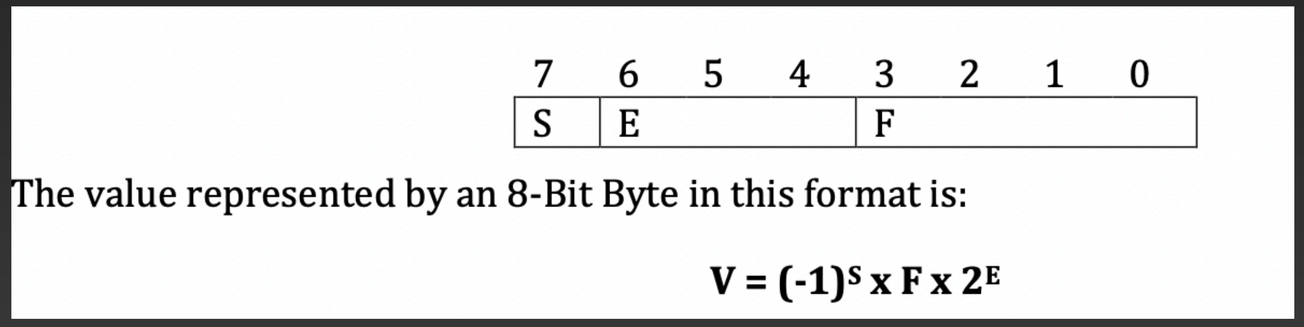
Example Conversion

000110100110

What is the sign of this number?

What is the exponent of this number?

What is the significand of this number?



Leading Zeroes	Exponent
1	7
2	6
3	5
4	4
5	3
6	2
7	1
≥ 8	0

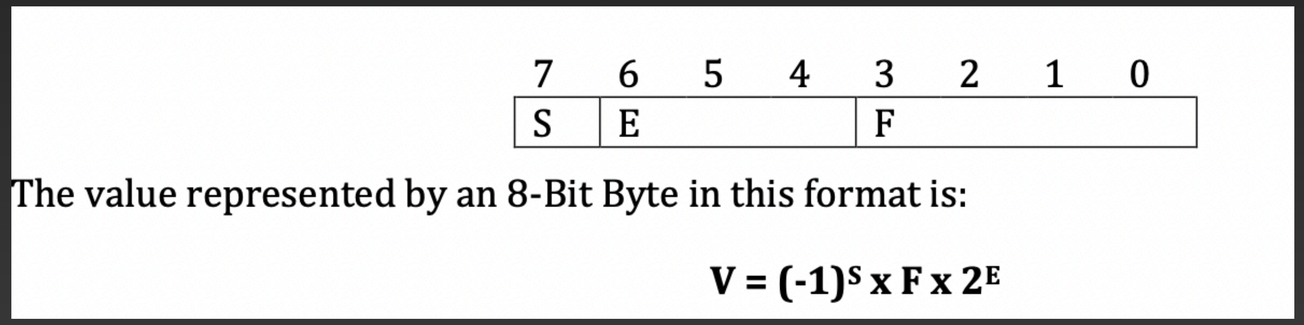
Example Conversion

1000101001010 Negate it first!

What is the sign of this number?

What is the exponent of this number?

What is the significand of this number?



Leading Zeroes	Exponent
1	7
2	6
3	5
4	4
5	3
6	2
7	1
≥ 8	0

Lab Design

- 3 Parts of the design
    - Sign Bit
    - Counting Leading Zeroes
    - Rounding

Top Module

Inputs and Outputs?

D, S, E, F

What is done in the top module?

Set up the inputs and outputs, create wires to connect your other modules, instantiate your other modules.

Sign Bit Implementation

- Convert the 12-bit two’s complement input to sign-magnitude representation
- Non-negative inputs should be unchanged
- Negative numbers should be complemented (invert all bits) and then incremented (add 1)

Counting Leading Zeroes Implementation

- This is done in order to find the exponent and the significand
- There are several ways to do it but....
  - Easiest way is to use if statements

Rounding Implementation

- Hardest portion of the lab
- Several cases that you need to handle
  - If the fifth bit following the last leading zero of the intermediate FP representation is 1, increment Significand by 1
  - If it overflows, shift significand by one bit and increase exponent by 1

Lab Deliverables

- Demo
  - Explanation of Methodology
  - Test bench design
- Project Code
  - Make sure to include all the project files for golden test bench
- Lab Report

Lab Rubric

Lab 2 Rubric:

Today

- Demo (**60**):
  - No on-board demonstration required in manuscript
  - Do not share golden test-bench with students, they need to have their own test-bench
  - Students can explain their methodology (**30**)
    - Overall design logic (**5**) and test bench design (**5**)
    - Count Leading Zeros implementation (**10**)
    - Rounding Logic (**8**) and test bench coverage (**10**)
  - Golden Test-bench Result (**30**)
    - *Can be tested after their submission or in the demonstration (to be discussed)*
    - Full credit: **30**
    - Having errors on edge cases: **20**
    - Can only cover <50% test cases: **10**
    - Test failed: **0**
- Report (**40**):
  - Intro and requirement (**10**)
  - Design Description (**15**)
    - Overall design (**5**)
    - Count Leading Zeros (**5**)
    - Rounding Logic (**5**)
  - Simulation Documentations (**10**)
    - General cases (**5**)
    - Edge cases covered (**5**)
      - Edge included but not limited to
        - Rounding (General case)
        - Rounding (overflow)
        - Leading zeros >= 8 (E = 0)
        - Special edge cases derived by their own special design
      - Student will receive full edge-cases credit if they can cover two or more
  - Conclusion (**5**)