# Exam # 1 - CSc 137, Fall 2020

**(Please show all work)**

Name: Matthew Mendoza

1. What is the 16-bit FP number representation of -7.375 in hex with 1-bit sign, 4-bit biased exponent, and 11-bit fraction, where bias = 7. Please identify the key components of FP number representation (25 pts)

**Step 1** Convert Decimal to Binary -7.375

Convert Integer: 7

* Successive division by 2*

$2\overline{)7}$ → $2\overline{)3}$ → $2\overline{)1}$ → $2\overline{)0}$ ← Reached Zero

$7_{(10)} = 111_{(2)}$

Convert Fractional: 0.375

* Successive multiplication of remaining products*

$0.375 \times 2 = 0 + 0.75$

$0.75 \times 2 = 1 + 0.5$

$0.5 \times 2 = 1 + 0.0$ ← Reached zero

$0.375_{(10)} = 011_{(2)}$

Together

$-7.375 = 111.011$

**Step 2** Converting Binary to Scientific Notation

$-7.375_{(10)} = 111.011_{(2)}$

Convert $111.011_{(2)}$ to Scientific Notation

$111.011_{(2)}$ —Scientific Notation→ $1.11011_{(2)} \times 2^2$

**Step3** Add Biased Exponent: 1.11011

Biased Offset = 7 * This will be provided in assignments and exams *

Biased Exponent = $1.11011 \times 2^4$

$4_{(10)} + 7_{(10)} = \boxed{11_{(10)}}$ (Convert to Binary) → $11_{(10)} = 1011_{(2)}$

**Step4** Fusion of Floating Point

Biased Component

$1.11011 \times 2^{\boxed{9}} = 1011_{(2)}$

Signed Component | Fractional Component

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

Signed Bit | Mantissa/Fraction

* When Signed component is positive set to 0(zero)
* When Signed component is negative set to 1

**Step 5** Convert to Hex - Final Step

| Number | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| Binary | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 |
| Hexadecimal | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

| Number | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|
| Binary | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |
| Hexadecimal | 8 | 9 | A | B | C | D | E | F |

| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

0xD 0xE 0xC 0x0

**Converted Hex Value: 0xDEC0**

2. Please circle the best answer about Von Neumann architecture below (25 pts)

A. Only data is stored in primary memory

B. Only instructions are stored in primary memory

C. Data and instructions are both stored in primary memory

D. None of Above

Primary = Ram
Secondary = HDD/SSD

The bottleneck in Von Neumann's architecture is that the processor is idle for a certain amount of time while memory is accessed.

3. Using Boolean Algebra, simplify the following logical expression? (25 pts)

$$F = AB + A(B+C) + B(B+C)$$

| Standard Precedence Levels | | |
|---|---|---|
| 0 | ~ ! | Negation |
| 1 | ∧ * | Intersection |
| 2 | ∨ ⊕ + | union XOR |
| 3 | → | Implication |

$F = AB + A(B+C) + B(B+C)$

→ $AB + A(B+C) + B$ : Asorption

$AB + A(B+C) + B$

→ $AB + AB + AC + B$ : Distributive

$AB + AB + AC + B$

→ $AB + AC + B$ : Idmotent?

$AB + AC + B$

→ $A(B+C) + B$ : Distributive

$A(B+C) + B$

→ $A \quad B+C$

$(A * B + C)$ ← Solution

4. Design a Single cell 1 bit Carry propagate (or Ripple Carry Adder) full adder. (25 pts)

a. Generate the truth table

b. Using K-map or Boolean algebra, determine the logical expression for Carry out (C-out) and Sum (S) Outputs

C. Draw the circuit diagram of the outputs in step b

a. X = 0111
Y = 1111

| Inputs | | | Outputs | |
|---|---|---|---|---|
| X | Y | Z-in | S | Z-out |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |

Always Zero

S = Sum

## b. Output Logical Expression

$$S = \bar{x}\bar{y}z\text{-}in + \bar{x}y\bar{z}\text{-}in + x\bar{y}\bar{z}\text{-}in + xyz\text{-}in$$

$$Z\text{-}out = \bar{x}yz\text{-}in + x\bar{y}z\text{-}in + xy\bar{z}\text{-}in + xyz\text{-}in$$

### K-map of the Output Logical Expression

$S = \bar{x}\bar{y}z\text{-}in + \bar{x}y\bar{z}\text{-}in + x\bar{y}\bar{z}\text{-}in + xyz\text{-}in$

| X \ Y,Z-in | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | | 1 | | 1 |
| 1 | 1 | | 1 | |

* See what changes in the impotants * * ⊕ = XOR *

S-simplified = X ⊕ Y ⊕ Z-in

$Z\text{-}out = \bar{x}yz\text{-}in + x\bar{y}z\text{-}in + xy\bar{z}\text{-}in + xyz\text{-}in$

| X \ Y,Z-in | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | | | 1 | |
| 1 | | 1 | 1 | 1 |

Important 02 = B Z-in

Important 01 = X B

Important 00 = X Z-in

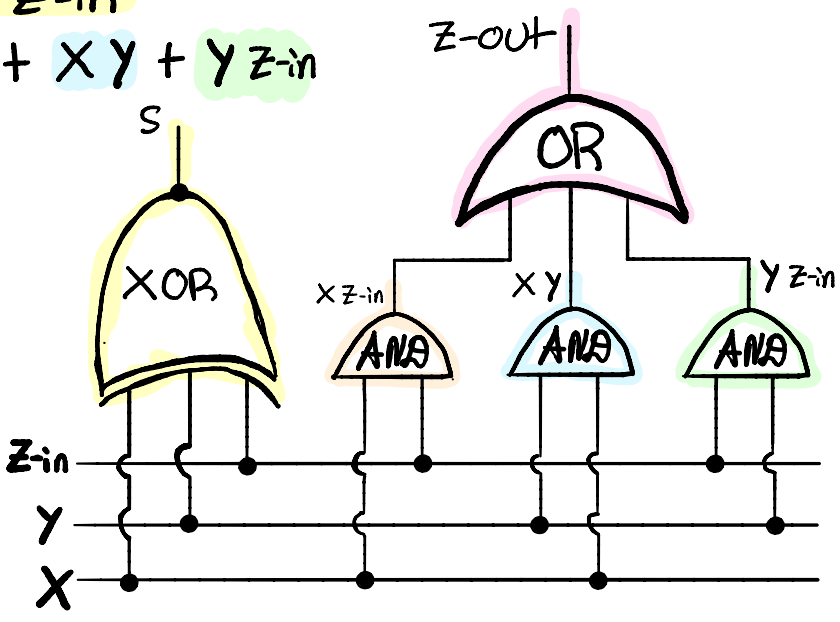
* See what changes in the impotants *

Z-out simplified = X Z-in + XY + Y Z-in

## C.

S-simplified = X ⊕ Y ⊕ Z-in

Z-out simplified = X Z-in + X Y + Y Z-in



Conversion Table:

| Decimal (Base 10) | Binary (Base 2) | Hexadecimal (Base 16) |
|---|---|---|
| 0 | 0000 | 0 |
| 1 | 0001 | 1 |
| 2 | 0010 | 2 |
| 3 | 0011 | 3 |
| 4 | 0100 | 4 |
| 5 | 0101 | 5 |
| 6 | 0110 | 6 |
| 7 | 0111 | 7 |
| 8 | 1000 | 8 |
| 9 | 1001 | 9 |
| 10 | 1010 | A |
| 11 | 1011 | B |
| 12 | 1100 | C |
| 13 | 1101 | D |
| 14 | 1110 | E |
| 15 | 1111 | F |