

# **Introduction to Computer Architecture**

**CISC 3310 Principles of Computer Architecture**  
**Lab Activity**

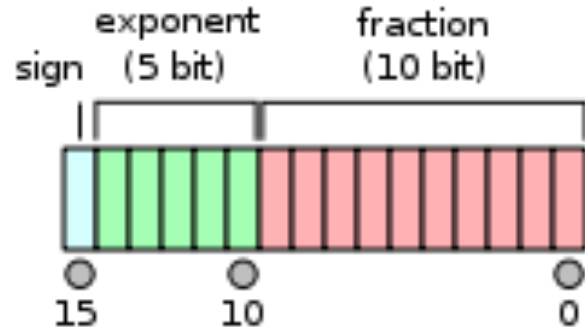
# Data Representation - Exercises

# Signed Numbers

- CB, Chapter 2, Exercise 24 - Using a “word” of 3 bits, list all the possible signed binary numbers and their decimal equivalents that are representable in:
  - a) Signed magnitude
  - b) One’s complement
  - c) Two’s complement
- CB, Chapter 2, Exercise 25 – repeat the previous question using a “word” of 4 bits
- CB, Chapter 2, Exercise 26 - From the results of the previous two questions, generalize the range of values (in decimal) that can be represented in any given  $x$  number of bits the three number representations.

# Example: Using IEEE 754 Floating Point Standard *half-precision binary floating point*

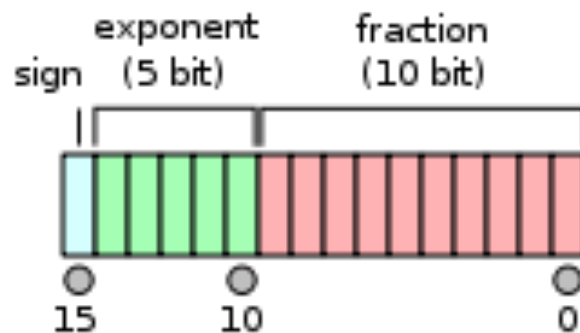
- IEEE 754 *half-precision binary floating point* (or binary16 for short) uses 5 bits for exponents => we get a 15 bias
  - Can represent any exponent between -15 and 16
- Uses 10 bits for the binary fraction



## Example: Using IEEE 754 Floating Point Standard *half-precision binary floating point*

- IEEE 754 *half-precision binary floating point* (or binary16 for short) uses 5 bits for exponents => we get a 15 bias
  - Can represent any exponent between -15 and 16
- Uses 10 bits for the binary fraction
- Floating Number X calculation:

$$x = (-1)^s \times (1 + \text{Fraction}) \times 2^{(\text{Exponent} - \text{Bias})}$$



# Floating Point Numbers

- Convert  $100.25_{10}$  to IEEE 754 *half-precision binary floating point*
- Convert 1010011010000000 in IEEE 754 *half-precision binary floating point* representation to a decimal number

## \* CRC Code

- Generate the CRC code for the number 10111011
  - Using the divisor 1011

# Hamming Code

- The **(7,4)** binary Hamming block encoder:
  - accepts blocks of **4**-bit of information
  - adds **3** parity bits to each such block and produces **7**-bits wide Hamming coded blocks
- Construct a **(7, 4)** binary Hamming code, using the following equations to generate the parity bits:

$$P_1 = D_1 \oplus D_2 \oplus D_3$$

$$P_2 = D_2 \oplus D_3 \oplus D_4$$

$$P_3 = D_1 \oplus D_3 \oplus D_4$$



## \* Hamming Code

- Generate all the possible codewords
  - For each possible message, generate a codeword
- What is the minimum distance between the code words?

Message m	Codeword c
0000	
0001	
0010	
0011	
0100	
0101	
0110	
0111	
1000	
1001	
1010	
1011	
1100	
1101	
1110	
1111	

Questions?

