BIRZEIT UNIVERSITY

Numbering Systems

Computer Science Department

Comp 1310

Tuesday, September 21, 2021

SUCCEED

You will never know if you don't keep trying, the next time might be your time to succeed.

Outline

- Converting Fractions.
- Adding Binary Fractions.
- Binary Subtraction.
- Data Representation.
- Characters and Integers Representation.
- Floating Point Representation.
- Summary

Converting Fractions

- When converting a fractional decimal value to binary, we need to use a slightly different approach. Instead of dividing by 2 ,we repeatedly multiply the decimal fraction by 2.

Let's take an example !

Converting Fractions

Convert **11.375_{10}** to it's binary equivalents.

First convert 11 to binary .

We know from the last week **11_{10}** = **1011_2**

Now convert $.375_{10}$ to binary

Converting Fractions

0.375 * 2 = 0.750
0.750 * 2 = 1.500
0.500 * 2 = 1.000

$.375_{10} = .011_{2}$

$11.375_{10} = 1011.011_{2}$

Converting Fractions

- Convert the following numbers to their binary equivalents.

□ (26.75_{10}) = 11010.11_2

□ (37.375_{10}) = H.W

Converting Fractions

- Exercise:
- Convert the following decimal number to binary?

$$(0.2)_{10} = (\ 0.\overline{0011}\ )_2$$

$$(0.3)_{10} = (0.0\overline{1001})_2$$

Adding Binary Fractions

- Example:
- 1011.0+0.011=

```
  1011.0
+       0.011
 ----------------
  1011.0 11
```

Adding Binary Fractions

- Example:
- 110.01+1.011=

```
             1
         110.01
+          1.011
  ----------------
          111.101
```

Binary Subtraction

- Solve the following 8-bit subtraction problem using 2's complement representation.

$01111111_2 – 76_{10}$ =???

Think if we rewrite the above problem as

$01111111_2 + (– 76)_{10}$

Binary Subtraction

Example: 01111111_2 + $(– 76)_{10}$

76 → 01001100

```
                          1 1
1’s complement  → 10110011
2’s complement→ +              1
                 ------------------
                   10110100   →(-76)
```

Binary Subtraction Cont.

$01111111_2 \; + (-\,76)_{10}$

```
         1 11 1
         01111111          127
   +     10110100       -   7 6
       ------------        -----------
Overflow 100110011          51
```

✗

Binary Subtraction Cont.

Example: 00110010_2 + $(– 125)_{10}$

125 → 01111101

1’s complement → 10000010

2’s complement→ + 1

10000011→(-125)

Binary Subtraction Cont.

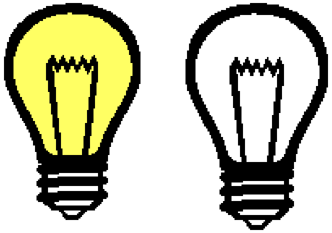
$00110010_2 + (– 125)_{10}$

```
            1
  00110010              50
+ 10000011         -   125
-----------------   ----------
   10110101            -75
```

- The 2’s comp for the result (10110101) is 01001011 equivalent to $(75)_{10}$

Data Representation

❖Computer understand two things: on and off .



❖Data represented in binary form .

❖Bit is the basic unit for storing data 0→off ,1→on .

❖Byte is a group of 8 bits. That is, each byte has 256(2^8) possible values.

❖Two bytes form a word

Parity bit

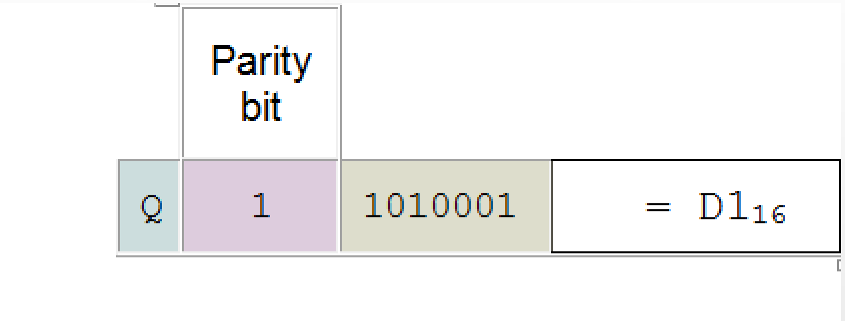
- Used for error detection
- Two types: 1. Odd parity (number of 1’s are odd)

  2. Even parity (number of 1’s are even)

Characters Representation

Using the **even parity** bit to represent the character **Q** (**Q = 81 in ASCII**) in memory (Hexadecimal) ?

$(81)_{10}=(01010001)_2$



	Parity bit		
Q	1	1010001	$= D1_{16}$

Memory

D1

Note: ASCII for A=65 and a=97

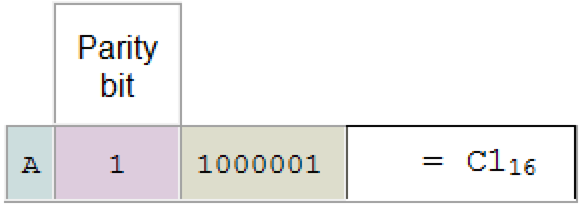
American Standard Code for Information Interchange

A=65	a=97
B=66	b=98
.	.
.	.

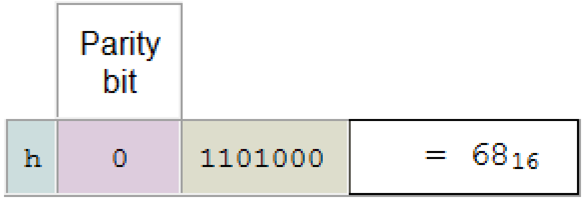
Characters Representation

Using the **odd parity** bit to represent **your name** in memory ?

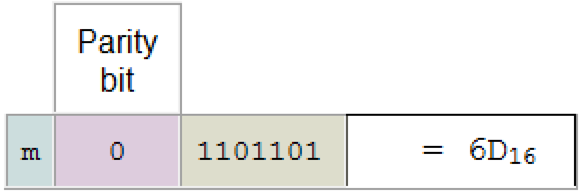
Ex. Ahmad



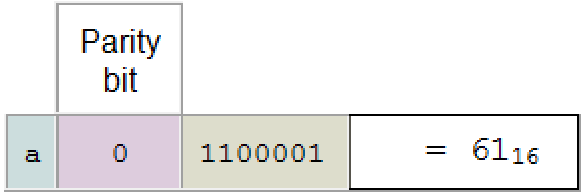
	Parity bit		
A	1	1000001	= $C1_{16}$



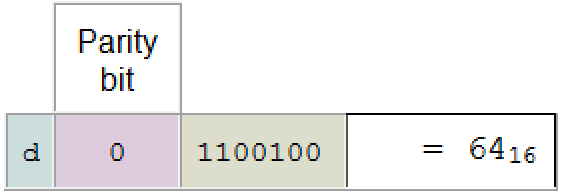
	Parity bit		
h	0	1101000	= 68_{16}



	Parity bit		
m	0	1101101	= $6D_{16}$



	Parity bit		
a	0	1100001	= 61_{16}



	Parity bit		
d	0	1100100	= 64_{16}

A	01000001
h	01101000
m	01101101
..	

Memory

C1
68
6D
61
64

Integers Representation

Represent the following integer in memory using 2 byte?

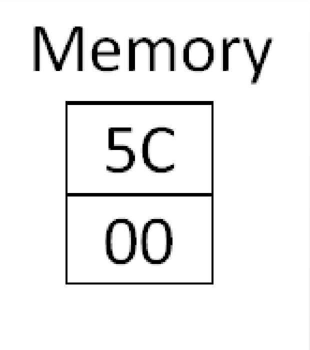
92

92 = 1011100

Answer

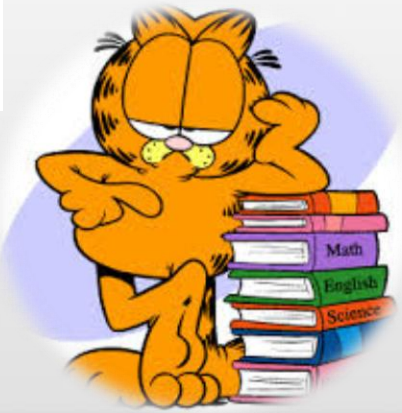
0000 0000 01011100

0 0 5 C


Memory

5C

00



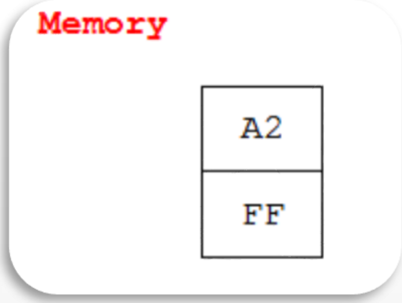
Integers Representation

Represent the following integer in memory using 2 byte?

```
-94

94  =     0000000001011110

1's →     1111111110100001
2's →+                   1
   --------------------------------
            1111111110100010
             F    F    A    2
```



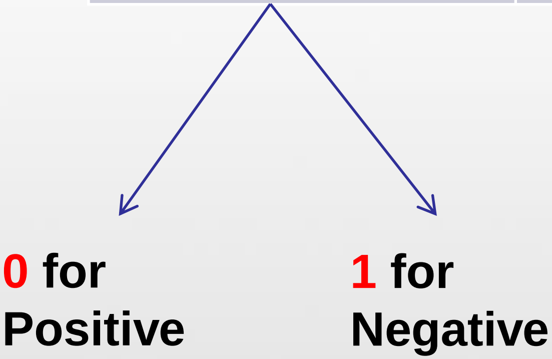
Memory

A2
FF

Floating Point Representation

32 bits divided into three sections

X	XXXXXXXX	XXXXXX…..X
1 bit For sign	8 bits For Exponent	23 bits For Mantissa



0 for Positive

1 for Negative

Floating Point Representation

32 bits divided into three sections

X	XXXXXXXX	XXXXXX…..X
1 bit For sign	8 bits For Exponent	23 bits For Mantissa

2^8 =256

0-255

What about negative ??

Floating Point Representation

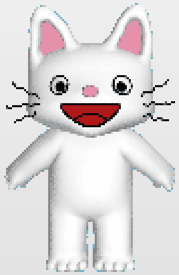
255/2=127.5 we take the integer part 127

0---------------------- 255

0---------------------- 255
-127 -127

-127 ---------------- 128

Let's take an example !



Floating Point Representation

Use the 32-bit floating representation to represent the following the binary number and show how it will represented in the memory?

$(26.75)_{10}$

Answer:

Convert the number from decimal to binary

Floating Point Representation

$(26.75)_{10} = (11010.11)_2$

$(11010.11)_2 = (1.101011 * 2^4)_2$ **Scientific notation**

Exponent = 127+4=131

$(131)_{10} = (10000011)_2$

0	10000011	10101100000000000000000

4	1	D	6	0	0	0	0

Memory
00
00
D6
41

Summary

- Converting Fractions.
- Adding Binary Fractions.
- Binary Subtraction.
- Data Representation.
- Characters and Integers Representation.
- Floating Point Representation.

Thanks to Mr. Abdallah Karakra