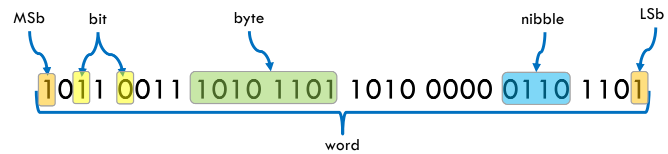
# 05 – BITS, BITES, WORDS & INTEGERS

COMP256 – COMPUTING ABSTRACTIONS

DICKINSON COLLEGE

## TERMINOLOGY

- *Binary* (base 2/radix 2) is a representation of numeric values using only the values 0 and 1.




## WHAT DO ALL THOSE 1'S AND 0'S MEAN?

- It depends…
  - Consider the byte: 1001 0100
  - It could be:
    - 148, 94, -20, -107     9.25, $4.5x2^{-3}$
    - 'm'     A medium grey
    - An instruction the computer to perform an addition
- How do we know which it is?
  - We need to know what representation is being used.
    - E.g. What is pie?

## SOME BINARY REPRESENTATIONS

- Non-Negative Integers
  - **Unsigned Binary**
  - **Binary Coded Decimal**
- Signed Integers
  - **Sign Magnitude**
  - **Two's Complement**
- Decimal Numbers
  - *Fixed Point*
  - Floating Point
- Text Data
  - EBCDIC
  - *ASCII*
  - Unicode
- Colors
  - *RGB*
  - CMYK
  - HSV
- Instructions
  - *Processor dependent*

# BINARY (BASE 2/RADIX 2) NUMBERS

- It's all about that base, no trouble:
  - Base 10: 501 $= \overset{\text{Hundreds}}{5*100} + \overset{\text{Tens}}{0*10} + \overset{\text{Ones}}{1*1}$
    $= 5*10^2 + 0*10^1 + 1*\underset{\textbf{Base 10}}{10^0}$
  - Base 2: 110 $= 1*2^2 + 1*2^1 + 0*\underset{\textbf{Base 2}}{2^0}$
    $= \underset{\text{Fours}}{1*4} + \underset{\text{Twos}}{1*2} + \underset{\text{Ones}}{0*1}$

# UNSIGNED BINARY

- *Unsigned binary* is used to represent non-negative integers.
  - Unsigned Binary to Decimal:
    - What is the base 10 value of the following 8-bit unsigned binary value:

      $1010\ 1101_2$
  - Decimal to Unsigned Binary
    - What is the 8-bit unsigned binary representation of the following base 10 value:

      $99_{10}$

# BINARY ADDITION

- Adding unsigned binary numbers.

```
  100     4          011     3          011     3
+ 001   + 1        + 010   + 2        + 011   + 3
-----   ---        -----   ---        -----   ---

   0111 1001      121
 + 0010 1010    +  42
 -----------    -----
```

# SIGN MAGNITUDE

- *Sign Magnitude* can be used to represent signed (positive and negative) integers.
  - The MSb is used as a sign bit.
    - 0 → positive    1 → negative
  - The remaining bits give the value using unsigned binary.
- Perform the following conversions using 4-bit Sign Magnitude representation:

$0101_{2SM} = \_\_\_\_\_\__{10}$    $1001_{2SM} = \_\_\_\_\_\__{10}$

$6_{10} = \_\_\_\_\_\__{2SM}$    $-3_{10} = \_\_\_\_\_\__{2SM}$

# TWO'S COMPLEMENT

- *Two's complement* is used by modern computers to represent signed integers.
  - The MSb indicates the sign.
    - 0 ➔ positive      1 ➔ negative
- Positive two's complement values are the same as unsigned binary.
  - For example in 8-bit two's complement:

$0010\ 0110_{2TC} = 38_{10}$      $18_{10} = 0001\ 0010_{2TC}$

Positive      Positive

# TWO'S COMPLEMENT

- The negative of a value in two's complement **<u>is found from its positive</u>** by:
  - "Flipping" all the bits
  - Adding 1

Flip the bits

Add 1

$0010\ 0110_{2TC} = 38_{10}$

Positive

$$\begin{array}{r} 1101\ 1001 \\ +\qquad\quad 1 \\ \hline 1101\ 1010_{2TC} \end{array} \rightarrow -38_{10}$$

Negative

# TWO'S COMPLEMENT

- The value of a negative two's complement number **<u>is found from its positive</u>** by:
  - "Flipping" all the bits
  - Adding 1

Flip the bits

Add 1

$1101\ 1010_{2TC}$

Negative

$$\begin{array}{r} 0010\ 0101 \\ +\qquad\quad 1 \\ \hline 0010\ 0110_{2TC} \end{array} = 38_{10}$$

Positive

So $1101\ 1010_{2TC} \rightarrow -38_{10}$

# TWO'S COMPLEMENT EXAMPLES

- What is the base 10 value of the following four-bit two's complement numbers?
  - $0101_{2TC}$
  - $1011_{2TC}$
- What is the four-bit two's complement representation of the following base 10 numbers?
  - $6_{10}$
  - $-3_{10}$

## MORE TWO'S COMPLEMENT EXAMPLES

- What is the base 10 value of the following 8-bit two's complement numbers?
  - $0001\ 1001_{2TC}$
  - $1010\ 1101_{2TC}$
- What is the 8-bit two's complement representation of the following base 10 numbers?
  - $11_{10}$
  - $-11_{10}$

## HEXADECIMAL REPRESENTATION

- *Hexadecimal (base 16) representation* is commonly found in computer science.

- 0x3AF9 = $3*16^3$ + $A*16^2$ + $F*16^1$ + $9*16^0$

| | | | |
|---|---|---|---|
| 3*4096 + | A*256 + | F*16 + | 9*1 |
| 3*4096 + | 10*256 + | 15*16 + | 9*1 |

| | |
|---|---|
| A ➔ 10 | D ➔ 13 |
| B ➔ 11 | E ➔ 14 |
| C ➔ 12 | F ➔ 15 |

## HEXADECIMAL SHORTHAND

- Hexadecimal values are often used as a shorthand for expressing long binary strings.
  - Conveniently every hex digit can be represented in 4 bits.

| | | |
|---|---|---|
| 0 ➔ 0000 | 6 ➔ 0110 | C ➔ 1100 |
| 1 ➔ 0001 | 7 ➔ 0111 | D ➔ 1101 |
| 2 ➔ 0010 | 8 ➔ 1000 | E ➔ 1110 |
| 3 ➔ 0011 | 9 ➔ 1001 | F ➔ 1111 |
| 4 ➔ 0100 | A ➔ 1010 | |
| 5 ➔ 0101 | B ➔ 1011 | |

  - So long binary strings can be easily represented as hex values:

```
1110 0011 0100 1111 1011 1010 1100 0010
E    3    4    F    B    A    C    2

            0xE34FBAC2
```