# CSC 222: Computer Organization & Assembly Language

## Topic – Data Representation

Instructor: Mr. Abdul Jabbar Saeed Tipu
Email: jabbar.saeed@ciitlahore.edu.pk

# Number System

- Any number system using a range of digits that represents a specific number. The most common numbering systems are decimal, binary, octal, and hexadecimal.
- Numbers are important to computers
  - represent information precisely
  - can be processed
- For example:
  - to represent yes or no: use 0 for no and 1 for yes
  - to represent 4 seasons: 0 (autumn), 1 (winter), 2(spring) and 3 (summer)

# Positional Number System

- A computer can understand positional number system where there are only a few symbols called digits and these symbols represent different values depending on the position they occupy in the number.

- A value of each digit in a number can be determined using
  - The digit
  - The position of the digit in the number
  - The base of the number system (where base is defined as the total number of digits available in the number system).

# Decimal Number System

- A numbering system that uses ten digits, from 0 to 9, to represent numerical values/quantities. Each digits has a weighted value of $10^0, 10^1, 10^2, 10^3$ and so on, ranging from right to left.

# Binary Number System

- A numbering system that uses two digits 0 and 1, to represent numerical values/quantities. Each digits has a weighted value of $2^0, 2^1, 2, 2^3$ and so on, ranging from right to left.

# Hexadecimal Number System

- A numbering system that uses sixteen digits, from 0 to 9 and A to F, to represent numerical values/quantities. Each digits has a weighted value of $16^0, 16^1, 16^2, 16^3$ and so on, ranging from right to left.

# Conversion Between Number Systems

- Converting Hexadecimal to Decimal
- Multiply each digit of the hexadecimal number from right to left with its corresponding power of 16 or weighted value.
- Convert the Hexadecimal number **82ADh** to decimal number.

# Conversion Between Number Systems

- Converting Binary to Decimal
- Multiply each digit of the binary number from right to left with its corresponding power of 2 or weighted value.
- Convert the Binary number **11101** to decimal number.

# Conversion Between Number Systems

- Converting Decimal to Binary
- Divide the decimal number by 2.
- Take the remainder and record it on the side.
- REPEAT UNTIL the decimal number cannot be divided into anymore.

# Conversion Between Number Systems

- Converting Decimal to Hexadecimal
- Divide the decimal number by 16.
- Take the remainder and record it on the side.
- REPEAT UNTIL the decimal number cannot be divided into anymore.

# Conversion Between Number Systems

- Converting Hexadecimal to Binary
- Given a hexadecimal number, simply convert each digit to it's binary equivalent. Then, combine each 4 bit binary number and that is the resulting answer.

- Converting Binary to Hexadecimal
- Begin at the rightmost 4 bits. If there are not 4 bits, pad 0s to the left until you hit 4. Repeat the steps until all groups have been converted.

# Binary Arithmetic Operations

- Addition
- Like decimal numbers, two numbers can be added by adding each pair of digits together with carry propagation.

```
  11001
+ 10011
 101100
```

Binary Addition

```
  647
+ 537
 1184
```

Decimal Addition

# Binary Arithmetic Operations

- Subtraction
- Two numbers can be subtracted by subtracting each pair of digits together with borrowing, where needed.

```
  11001
- 10011
  -----
  00110
```

Binary Subtraction

```
  627
- 537
  ---
  090
```

Decimal Subtraction

# Hexadecimal Arithmetic Operations

- Addition
- Like decimal numbers, two numbers can be added by adding each pair of digits together with carry propagation.

```
  5B39
+ 7AF4
  D62D
```

Hexadecimal Addition

# HexaDecimal Arithmetic Operations

- Subtraction
- Two numbers can be subtracted by subtracting each pair of digits together with borrowing, where needed.

```
  D26F
- BA94
  17DB
```

Hexadecimal Subtraction

# MSB and LSB

- In computing, the **most significant bit** (**msb**) is the bit position in a binary number having the greatest value. The **msb** is sometimes referred to as the **left-most bit.**

- In computing, the **least significant bit** (**lsb**) is the bit position in a binary integer giving the units value, that is, determining whether the number is even or odd. The **lsb** is sometimes referred to as the **right-most bit.**

# Unsigned Integers

- An unsigned integer is an integer at represent a magnitude, so it is never negative.

- Unsigned integers are appropriate for representing quantities that can be never negative.

# Signed Integers

- A signed integer can be positive or negative.
- The most significant bit is reserved for the sign:
  - 1 means negative and 0 means positive.

- Example:

  00001010 = decimal 10

  10001010 = decimal -10

# One's Complement

- The one's complement of an integer is obtained by complementing each bit, that is, replace each 0 by a 1 and each 1 by a 0.

# 2's Complement

- Negative integers are stored in computer using 2's complement.
- To get a two's complement by first finding the one's complement, and then by adding 1 to it.

- Example

```
  11110011    (one's complement of 12)
+ 00000001    (decimal 1)
  --------
  11110100    (two's complement of 12)
  --------
```

# Subtract as 2's Complement Addition

- Find the difference of 12 – 5 using complementation and addition.
- 00000101 (decimal 5)
- 11111011 (2's Complement of 5)

```
    00001100   (decimal 12)
  + 11111011   (decimal -5)
  ----------
    00000111   (decimal 7)
```

No Carry

# Example

- Find the difference of 5ABCh – 21FCh using complementation and addition.
- 5ABCh = 0101 1010 1011 1100
- 21FCh = 0010 0001 1111 1100
- 1101 1110 0000 0100 (2's Complement of 21FCh)

```
   0101 1010 1011 1100 (Binary 5ABCh)
 + 1101 1110 0000 0100  (1's Complement of 21FCh)
 ---------------------
  10011 1000 1100 0000
```

Discard Carry

# Decimal Interpretation

- How to interpret the contents of a byte or word as a signed and unsigned decimal integer?
- Unsigned decimal interpretation
  - Simply just do a binary to decimal conversion or first convert binary to hexadecimal and then convert hexadecimal to decimal.
- Signed decimal interpretation
  - If msb is zero then number is positive and signed decimal is same as unsigned decimal.
  - If msb is one then number is negative, so call it -N. To find N, just take the 2's complement and then convert to decimal.

# Example

- Give unsigned and signed decimal interpretation FE0Ch.
- Unsigned decimal interpretation
  - $16^3 * 15 + 16^2 * 14 + 16^1 * 0 + 16^0 * 12 = 61440 + 3584 + 0 + 12 = 65036$
- Signed decimal interpretation
  - FE0Ch = 1111 1110 0000 1100 (msb is 1, so number is negative).
  - To find N, get its 2's complement

```
     0000 0001 1111 0011 (1's complement of FE0Ch)
    +                  1
 N = 0000 0001 1111 0100  = 01F4h = 500
```

So, -N = 500

# Decimal Interpretation

- For 16 – bit word, following relationships holds between signed and unsigned decimal interpretation
- From 0000h – 7FFFh, signed decimal = unsigned decimal
- From 8000h – FFFFh, signed decimal = unsigned decimal – 65536.
- Example:
- Unsigned interpretation of FE0Ch is 65036.
- Signed interpretation of FE0Ch = 65036 – 65536 = -500.

## Binary, Decimal, and Hexadecimal Equivalents.

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

# Character Representation

- All data, characters must be coded in binary to be processed by the computer.
- ASCII:
  - American Standard Code for Information Interchange
  - Most popular character encoding scheme.
  - Uses 7 bit to code each character.
  - $2^7 = 128$ ASCII codes.
  - Single character Code = One Byte [7 bits: char code, 8th bit set to zero]
  - 32 to 126 ASCII codes: printable
  - 0 to 31 and 127 ASCII codes: Control characters

| Dec | Hex | Char | Dec | Hex | Char | Dec | Hex | Char | Dec | Hex | Char |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 00 | Null | 32 | 20 | Space | 64 | 40 | @ | 96 | 60 | ` |
| 1 | 01 | Start of heading | 33 | 21 | ! | 65 | 41 | A | 97 | 61 | a |
| 2 | 02 | Start of text | 34 | 22 | " | 66 | 42 | B | 98 | 62 | b |
| 3 | 03 | End of text | 35 | 23 | # | 67 | 43 | C | 99 | 63 | c |
| 4 | 04 | End of transmit | 36 | 24 | $ | 68 | 44 | D | 100 | 64 | d |
| 5 | 05 | Enquiry | 37 | 25 | % | 69 | 45 | E | 101 | 65 | e |
| 6 | 06 | Acknowledge | 38 | 26 | & | 70 | 46 | F | 102 | 66 | f |
| 7 | 07 | Audible bell | 39 | 27 | ' | 71 | 47 | G | 103 | 67 | g |
| 8 | 08 | Backspace | 40 | 28 | ( | 72 | 48 | H | 104 | 68 | h |
| 9 | 09 | Horizontal tab | 41 | 29 | ) | 73 | 49 | I | 105 | 69 | i |
| 10 | 0A | Line feed | 42 | 2A | * | 74 | 4A | J | 106 | 6A | j |
| 11 | 0B | Vertical tab | 43 | 2B | + | 75 | 4B | K | 107 | 6B | k |
| 12 | 0C | Form feed | 44 | 2C | , | 76 | 4C | L | 108 | 6C | l |
| 13 | 0D | Carriage return | 45 | 2D | - | 77 | 4D | M | 109 | 6D | m |
| 14 | 0E | Shift out | 46 | 2E | . | 78 | 4E | N | 110 | 6E | n |
| 15 | 0F | Shift in | 47 | 2F | / | 79 | 4F | O | 111 | 6F | o |
| 16 | 10 | Data link escape | 48 | 30 | 0 | 80 | 50 | P | 112 | 70 | p |
| 17 | 11 | Device control 1 | 49 | 31 | 1 | 81 | 51 | Q | 113 | 71 | q |
| 18 | 12 | Device control 2 | 50 | 32 | 2 | 82 | 52 | R | 114 | 72 | r |
| 19 | 13 | Device control 3 | 51 | 33 | 3 | 83 | 53 | S | 115 | 73 | s |
| 20 | 14 | Device control 4 | 52 | 34 | 4 | 84 | 54 | T | 116 | 74 | t |
| 21 | 15 | Neg. acknowledge | 53 | 35 | 5 | 85 | 55 | U | 117 | 75 | u |
| 22 | 16 | Synchronous idle | 54 | 36 | 6 | 86 | 56 | V | 118 | 76 | v |
| 23 | 17 | End trans. block | 55 | 37 | 7 | 87 | 57 | W | 119 | 77 | w |
| 24 | 18 | Cancel | 56 | 38 | 8 | 88 | 58 | X | 120 | 78 | x |
| 25 | 19 | End of medium | 57 | 39 | 9 | 89 | 59 | Y | 121 | 79 | y |
| 26 | 1A | Substitution | 58 | 3A | : | 90 | 5A | Z | 122 | 7A | z |
| 27 | 1B | Escape | 59 | 3B | ; | 91 | 5B | [ | 123 | 7B | { |
| 28 | 1C | File separator | 60 | 3C | < | 92 | 5C | \ | 124 | 7C | \| |
| 29 | 1D | Group separator | 61 | 3D | = | 93 | 5D | ] | 125 | 7D | } |
| 30 | 1E | Record separator | 62 | 3E | > | 94 | 5E | ^ | 126 | 7E | ~ |
| 31 | 1F | Unit separator | 63 | 3F | ? | 95 | 5F | _ | 127 | 7F | □ |

# How to Convert?

- If a byte contains the ASCII code of an uppercase letter, what hex should be added to it to convert to lower case?
  - Solution: 20 h
  - Example: A (41h) a (61 h)
- If a byte contains the ASCII code of a decimal digit, What hex should be subtracted from the byte to convert it to the numerical form of the characters?
  - Solution: 30 h
  - Example: 2 (32 h)

# Character Storage

**ASCII Representation of "123" and 123**