CS 238 - ASSEMBLY LANGUAGE FOR ENGINEERS

Amarnath Jasti
Sr. Research Associate,
Advanced Networking Research Center (ANRC)
Manager, Cisco Technical Research Center
Wichita State University

Grading Policy

- Office Hours
- Text Book
- Assignments
- Exams
- Grades

Pre-Requisites

- Numbering system.
- Fundamental knowledge about programming languages.
- Computer Hardware.
- Digital Logic.
- Creativity (Think outside the box)
- Common sense

There are 10 types of people in this world; those who understand binary and other who doesn't.

Chapter 1

Data Representation

CS238 – Assembly Language Programming
Amarnath Jasti

Review

- Why assembly is needed?
- Computer system
- Assembly Language definition:
  - Machine specific programming language with a one-to-one correspondence between the statements and the computers native machine language and is specific to the processor or processor family
  - Note: Instructions in assembly are designed to match a computers machine instruction set and hardware architecture.

Base Index

- Decimal – 10 – 0 1 2 3 4 5 6 7 8 9
- Binary – 2 – 0 1
- Octal – 8 – 0 1 2 3 4 5 6 7
- Hexadecimal – 16 – 0 1 2 3 4 5 6 7 8 9 A B C D E F

Binary Numbers

MSB
Sign -Bit

15 8 7 0

1011 0011 1000 0110

LSB

Byte

Nibble
(Hex)

Bit

Binary Numbers

15 8 7 0

1011 0011 1000 0110

Word – 16 bits (INTEL) → 0-65535

Double Word – 32 bits → 0 – (2^{32}-1)

Quad Word – 64 bits → 0 – (2^{64}-1)

Note: All ranges are in unsigned range.

Binary to Decimal

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	**64**	**32**	**16**	**8**	**4**	**2**	**1**
1	0	1	1	1	0	1	0

1 x 0 = 0
2 x 1 = 2
4 x 0 = 0
8 x 1 = 8
16 x 1 = 16
32 x 1 = 32
64 x 0 = 0
128 x 1 = 128

Total = 186

Note: The bit to the far right is the Least Significant Bit (LSb) and will determine if the number is even or odd.
Note: Some Text Books will have Least Significant Bit (LSB).

Binary to Hexadecimal

Convert the following 16 bit binary number to hexadecimal.
0001111111000

2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0
8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1
0	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**0**	**0**

0 3 15 8

0 3 F 8

- **First break the binary number into 4 bits each. If you fall short, add 0's on the most significant side.**
- **Now take each 4 bits and place then in the binary chart to convert to decimal.**
- **Now insert the hexadecimal equivalent for the decimal number.**

Hexadecimal to Decimal

16^3	16^2	16^1	16^0
4096	**256**	**16**	**1**
0	A	0	4

1 x 4 = 4
16 x 0 = 0
256 x 10 = 2560
4096 x 0 = 0

Total = 2564

Number Types

- Signed (MSB= 0 – '+'ve, MSB= 1- '-'ve)
- Unsigned
- Conversions
  - Unsigned Decimal to Binary and vice-versa
  - Unsigned Decimal to Hex and vice-versa
  - Signed Decimal to Binary and vice-versa
  - Signed Decimal to Hex and vice-versa

Unsigned Integers

- Represents positive integers only
- Example: ASCII character codes
- Not necessary to indicate a sign, so all 8 or 16 bits can be used for the magnitude:
  - 1 byte = 8 bits = 2^8 = 256 (0 to 255)
  - 2 bytes = 16 bits = 2^{16} = 65,536 (0 to 65,535)
  - 4 bytes = 32 bits = 2^{32} = 4,294,967,296 (0 to 4,294,967,295)

Signed Integers

- Represents positive and negative integers
- MSB (Most Significant Bit – leftmost bit) used to indicate sign
  - 0 = positive, 1 = negative
- One less bit is used for the magnitude, with one extra negative value
  - 1 byte = 8-1 bits = 2^7 (-128 to +127)
  - 2 bytes = 16-1 bits = 2^{15} (-32,768 to +32,767)
  - 4 bytes = 32-1 bits = 2^{31} (-2,147,483,648 to +2,147,483,647)

1's & 2's Complement

- 1's complement form
  - Formed by reversing (complementing) each bit
- 2's complement form
  - Formed by adding 1 to 1's complement
  - Negative numbers are stored this way
  - Additive inverse of a number
  - Computer never has to subtract
    - A – B = A + (-B)

Decimal Conversion

- Unsigned Integers
  - Convert binary directly to decimal form
- Signed Integers
  - If MSB = 0, convert directly to decimal
  - If MSB = 1, convert to 2's complement form (reverse the bits & add 1), then to decimal form

2's compliment (Signed Decimal Conversion)

- $01001101_2 = 77_{10}$
  - The most significant bit is 0, so it's a positive value.
- What is the binary value of -77_{10} ?
- To convert to –77 in two's-complement notation,
  - Consider the absolute value
  - **One's Compliment:** Inverse the bits; 0 becomes 1, and 1 becomes 0
  - Add 1 to the result.
  - Result is the equivalent binary value of signed integer.
- Absolute Value ➔ |-77| = 77 = 0100 1101
- One's Compliment ➔ 1011 0010
- Adding 1 ➔ 1011 0010 + 1 = **1011 0011** = -77_{10}

Binary Addition and Subtraction

- Addition: 1 + 1 = 10, 0 + 1 = 1
- Subtraction
  - Smaller number from a larger number
    - Determine 2's compliment of the smaller number
    - Add 2's compliment to the larger number
    - Discard the final carry
  - Larger number from a smaller number
    - Determine the 2's compliment of the larger number
    - Add the 2's compliment to the smaller number.
    - There is no carry from the left most column. The result is in 2's compliment, and is negative.
    - Change the sign and take the 2's compliment to get the result.
    - Ex: 9-13 = -4
    - 1101 -> 0011; 0011 + 1001 = 1100 = [0100] Abs value.

Binary Coded Decimal - BCD

- BCD represents each of the digits of an unsigned decimal as the 4-bit binary equivalents.
- Unpacked BCD: Contains only one decimal digit per byte.
- Packed BCD: packs two decimal digits into a single byte.
- Ex: 8 – 0000 1000, 0000 1000
- 10 – 0000 0001 0000 0000, 0001 0000

Reading assignment

- Section 1.3
- Section 1.4
- External Links posted on Blackboard.