MPI: Microprocessors and Interfacing
Academic Year: 2021 - 22

Dr. Praveen Kumar Alapati
praveenkumar.alapati@mechyd.ac.in

Department of Computer Science and Engineering

Ecole Centrale School of Engineering

Mahindra™ University
Global Thinkers. Engaged Leaders.

CS309: Lab-6 (Due Date: November 7, 2021)

1. Write an assembly language program (8086) to sort n numbers ($n \geq 10$) using Merge-Sort
2. Write an assembly language program to sort n numbers ($n \geq 10$) using Quick-Sort.

CS309: Lab-5 (Due Date: October 20, 2021)

1. Write an assembly language program (8086) to perform the addition of two n-byte numbers ($n \geq 10$) and store the result in third one.
2. Write an assembly language program to sort n ($n \geq 10$) numbers using selection sort.
3. **Bonus**: Write an assembly language program to find the factorial of n, where $n \geq 10$.

CS309: Lab-4 (Due Date: October 11, 2021)

1. Write an assembly language program (8086) to find the addition to two 3 X 3 matrices.

2. Write an assembly language program to perform the multiplication to two 3 x 3 matrices.

Lab3- Due Date: September 26, 2021

1. Write an Assembly Language Program (ALP) to count the number of even numbers from a given series of 8-bit hexadecimal numbers. Assume that number of elements in that series is ten.
2. We are given with a series of 8-bit signed numbers (represented using hexadecimal notation). Write an ALP to count the number of values > 0, < 0, and $= 0$. Assume that number of elements in that series is twenty.
3. We are given with a series of 8-bit signed numbers (represented using hexadecimal notation). Assume that number of elements in that series is sixteen. Write an ALP to find the parity of each number: odd parity should be represented using 01 and even parity using 00.

Lab2- Due Date: September 11, 2021

1. Write one page report on MASM assembler.
2. Install MASM assembler in your System.
3. Write assembly language programs to compute the following (use MASM assembler).
    a. Addition of 5 Integers
    b. Multiplication of 3 Integers
    c. Evaluate $1 + x + x^2 + x^3 + ... + x^n$
       (Assume that x and n are positive integers)

Note:**Properly handle all corner cases and clearly state all your assumptions.**

Sample C Code

```
// Example1: Addition (eg1.c)
#define A 10
#define B 20
int main(){
int a=A;
int b=B;
a=a+b;
return a;
}
```

Table 1: Addition of Two Integers

```
// Example2: Addition (eg2.c)
#include<stdio.h>
#define A 10
#define B 20
int main(){
int a=A;
int b=B;
a=a+b;
printf("Result is %d",a);
return a;
}
```

Table 2: Addition of Two Integers

```
gcc -E eg1.c > eg1.i
gcc -S eg1.i > eg1.s
as eg1.s -o eg1.o
gcc -o eg1.exe eg1.o
./eg1.exe
```

Lab1 - Due date: September 1, 2021.

Develop C-Programs for the following problem statements and generate assembly code using that C-code.

1. Print the internal representation (2's complement form) of data stored in primary data types
(int, float, and double).
2. Perform addition and multiplication of two 32-bit numbers (Please remember that the input is 32-bit binary number). The numbers can either be integers or real numbers. Assume that the integers are represented in 2's complement form and the real numbers are represented using single-precision IEEE 754 standard.
3. Convert a decimal number into an equivalent BCD number, and vice versa.

Submission Guide Lines

- **Max. team size is 3.**
- Mail-ID: cs309.mpi.2021@gmail.com
- Sub:TEAM_NUM_LAB1/_ASSIGN1
- Attach.Name and Type: (Sub.).zip
- Late Submission<=3-Days:50%.
- Write a readme file to understand your solutions.

Number Systems

- Representation of Integer Numbers
  - Signed Magnitude Representation
  - 1's Complement Representation
  - 2's Complement Representation
- Representation of Real Numbers
  - Fixed Point Representation
  - Floating Point Representation

Resolution is difference between two successive numbers.

Representation of Integer Numbers

Let A $= a_{n-1}a_{n-2}a_{n-3}...a_0$ is an n-bit binary number
if A is an **unsigned integer**, then value of A is : $\sum_{i=0}^{n-1}(2^i \times a_i)$.
if A is a **signed integer**:

- Signed Magnitude Representation:
  - A $= \sum_{i=0}^{n-2}(2^i \times a_i)$, if $a_{n-1} = 0$
  - A $= -\sum_{i=0}^{n-2}(2^i \times a_i)$, if $a_{n-1} = 1$
- 1's Complement Rep.: A $= -(2^{n-1} - 1) \times a_{n-1} + \sum_{i=0}^{n-2}(2^i \times a_i)$
- 2's Complement Rep.: A $= -2^{n-1} \times a_{n-1} + \sum_{i=0}^{n-2}(2^i \times a_i)$

Resolution: 1

Range of Numbers

Let A $= a_{n-1}a_{n-2}a_{n-3}...a_0$ is an n bit binary number
if A is an **unsigned integer**, then range of A is : 0 to $(2^n - 1)$.
if A is a **signed integer**:

- Signed Magnitude Rep., range of A is : $-(2^{n-1} - 1)$ to $(2^{n-1} - 1)$.
- 1's Complement Rep., range of A is : $-(2^{n-1} - 1)$ to $(2^{n-1} - 1)$.
- 2's Complement Rep., range of A is : -2^{n-1} to $(2^{n-1} - 1)$.

Expansion of Bit Length

Add additional bit positions to the left and fill in with value of the sign bit.
Let A = **1 0 1 0** is a 4-bit binary number,
Representation of A using 8-bits (i.e. B): **1 1 1 1 1 0 1 0**.
is A=B ?

- In 2's Complement Rep.: **Yes**.
- In 1's Complement Rep.: **Yes**.
- In Signed Magnitude Rep.: **No**.

Expansion of Bit Length

Add additional bit positions to the left and fill in with value of the sign bit.
Let A = **1 0 1 0** is a 4-bit binary number,
Representation of A using 8-bits (i.e. B): **1 1 1 1 1 0 1 0**.
is A=B ?

- In 2's Complement Rep.: **Yes**.
- In 1's Complement Rep.: **Yes**.
- In Signed Magnitude Rep.: **No**.

Expansion of Bit Length

Add additional bit positions to the left and fill in with value of the sign bit.
Let A = **1 0 1 0** is a 4-bit binary number,
Representation of A using 8-bits (i.e. B): **1 1 1 1 1 0 1 0**.
is A=B ?

- In 2's Complement Rep.: **Yes**.
- In 1's Complement Rep.: **Yes**.
- In Signed Magnitude Rep.: **No**.

Expansion of Bit Length

Add additional bit positions to the left and fill in with value of the sign bit.
Let A = **1 0 1 0** is a 4-bit binary number,
Representation of A using 8-bits (i.e. B): **1 1 1 1 1 0 1 0**.
is A=B ?

- In 2's Complement Rep.: **Yes**.
- In 1's Complement Rep.: **Yes**.
- In Signed Magnitude Rep.: **No**.

Real Numbers

1. $(4.5)_{10} = (100.1)_2$
2. $(8.25)_{10} = (1000.01)_2$
3. $(16.125)_{10} = (10000.001)_2$
4. $(0.875)_{10} = (0.111)_2$
5. $(4.5)_{10} = (1.001)_2 \times 2^2$
6. $(8.25)_{10} = (1.00001)_2 \times 2^3$
7. $(16.125)_{10} = (1.0000001)_2 \times 2^4$
8. $(0.875)_{10} = (1.11)_2 \times 2^{-1}$

Representations 5 to 8 are called Normalized Representations.
Normalized Rep.: $(\pm 1.xxxxxx)_2 \times 2^E$, Where 'E' is a**True Exponent**, 'xxxxxx' is a **Fraction/Mantissa**.

Real Numbers

1. $(4.5)_{10} = (100.1)_2$
2. $(8.25)_{10} = (1000.01)_2$
3. $(16.125)_{10} = (10000.001)_2$
4. $(0.875)_{10} = (0.111)_2$
5. $(4.5)_{10} = (1.001)_2 \times 2^2$
6. $(8.25)_{10} = (1.00001)_2 \times 2^3$
7. $(16.125)_{10} = (1.0000001)_2 \times 2^4$
8. $(0.875)_{10} = (1.11)_2 \times 2^{-1}$

Representations 5 to 8 are called Normalized Representations.
Normalized Rep.: $(\pm 1.xxxxxx)_2 \times 2^E$, Where 'E' is a **True Exponent**, 'xxxxxx' is a **Fraction/Mantissa**.

Real Numbers

1. $(4.5)_{10} = (100.1)_2$
2. $(8.25)_{10} = (1000.01)_2$
3. $(16.125)_{10} = (10000.001)_2$
4. $(0.875)_{10} = (0.111)_2$
5. $(4.5)_{10} = (1.001)_2 \times 2^2$
6. $(8.25)_{10} = (1.00001)_2 \times 2^3$
7. $(16.125)_{10} = (1.0000001)_2 \times 2^4$
8. $(0.875)_{10} = (1.11)_2 \times 2^{-1}$

Representations 5 to 8 are called Normalized Representations.
Normalized Rep.: $(\pm 1.xxxxxx)_2 \times 2^E$, Where 'E' is a **True Exponent**, 'xxxxxx' is a **Fraction/Mantissa**.

Real Numbers

1. $(4.5)_{10} = (100.1)_2$
2. $(8.25)_{10} = (1000.01)_2$
3. $(16.125)_{10} = (10000.001)_2$
4. $(0.875)_{10} = (0.111)_2$
5. $(4.5)_{10} = (1.001)_2 \times 2^2$
6. $(8.25)_{10} = (1.00001)_2 \times 2^3$
7. $(16.125)_{10} = (1.0000001)_2 \times 2^4$
8. $(0.875)_{10} = (1.11)_2 \times 2^{-1}$

Representations 5 to 8 are called Normalized Representations.
Normalized Rep.: $(\pm 1.xxxxxx)_2 \times 2^E$, Where 'E' is a**True Exponent**, 'xxxxxx' is a **Fraction/Mantissa**.

Real Numbers

1. $(4.5)_{10} = (100.1)_2$
2. $(8.25)_{10} = (1000.01)_2$
3. $(16.125)_{10} = (10000.001)_2$
4. $(0.875)_{10} = (0.111)_2$
5. $(4.5)_{10} = (1.001)_2 \times 2^2$
6. $(8.25)_{10} = (1.00001)_2 \times 2^3$
7. $(16.125)_{10} = (1.0000001)_2 \times 2^4$
8. $(0.875)_{10} = (1.11)_2 \times 2^{-1}$

Representations 5 to 8 are called Normalized Representations.
Normalized Rep.: $(\pm 1.xxxxxx)_2 \times 2^E$, Where 'E' is a**True Exponent**, 'xxxxxx' is a **Fraction/Mantissa**.

Real Numbers

1. $(4.5)_{10} = (100.1)_2$
2. $(8.25)_{10} = (1000.01)_2$
3. $(16.125)_{10} = (10000.001)_2$
4. $(0.875)_{10} = (0.111)_2$
5. $(4.5)_{10} = (1.001)_2 \times 2^2$
6. $(8.25)_{10} = (1.00001)_2 \times 2^3$
7. $(16.125)_{10} = (1.0000001)_2 \times 2^4$
8. $(0.875)_{10} = (1.11)_2 \times 2^{-1}$

Representations 5 to 8 are called Normalized Representations.
Normalized Rep.: $(\pm 1.xxxxxx)_2 \times 2^E$, Where 'E' is a**True Exponent**, 'xxxxxx' is a **Fraction/Mantissa**.

Real Numbers

1. $(4.5)_{10} = (100.1)_2$
2. $(8.25)_{10} = (1000.01)_2$
3. $(16.125)_{10} = (10000.001)_2$
4. $(0.875)_{10} = (0.111)_2$
5. $(4.5)_{10} = (1.001)_2 \times 2^2$
6. $(8.25)_{10} = (1.00001)_2 \times 2^3$
7. $(16.125)_{10} = (1.0000001)_2 \times 2^4$
8. $(0.875)_{10} = (1.11)_2 \times 2^{-1}$

Representations 5 to 8 are called Normalized Representations.
Normalized Rep.: $(\pm 1.xxxxxx)_2 \times 2^E$, Where 'E' is a**True Exponent**, 'xxxxxx' is a **Fraction/Mantissa**.

Real Numbers

1. $(4.5)_{10} = (100.1)_2$
2. $(8.25)_{10} = (1000.01)_2$
3. $(16.125)_{10} = (10000.001)_2$
4. $(0.875)_{10} = (0.111)_2$
5. $(4.5)_{10} = (1.001)_2 \times 2^2$
6. $(8.25)_{10} = (1.00001)_2 \times 2^3$
7. $(16.125)_{10} = (1.0000001)_2 \times 2^4$
8. $(0.875)_{10} = (1.11)_2 \times 2^{-1}$

Representations 5 to 8 are called Normalized Representations.

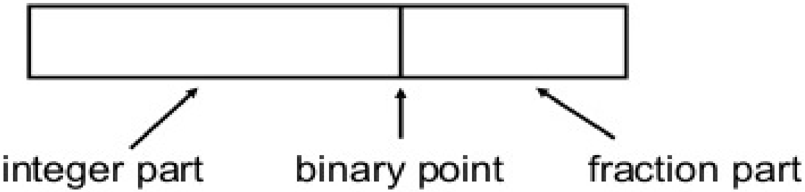
Normalized Rep.: $(\pm 1.xxxxxx)_2 \times 2^E$, Where 'E' is a**True Exponent**, 'xxxxxx' is a **Fraction/Mantissa**.

Real Numbers

1. $(4.5)_{10} = (100.1)_2$
2. $(8.25)_{10} = (1000.01)_2$
3. $(16.125)_{10} = (10000.001)_2$
4. $(0.875)_{10} = (0.111)_2$
5. $(4.5)_{10} = (1.001)_2 \times 2^2$
6. $(8.25)_{10} = (1.00001)_2 \times 2^3$
7. $(16.125)_{10} = (1.0000001)_2 \times 2^4$
8. $(0.875)_{10} = (1.11)_2 \times 2^{-1}$

Representations 5 to 8 are called Normalized Representations.
Normalized Rep.: $(\pm 1.xxxxxx)_2 \times 2^E$, Where 'E' is a**True Exponent**, 'xxxxxx' is a **Fraction/Mantissa**.

Fixed Point (FP) Representation



Find the decimal equivalent of the following binary numbers. Assume that the binary numbers are represented using FP representation (6,2), i.e., 6 bits for integer part and 2 bits for fractional part.

- $(00101011)_2 =$ **?**
- $(11111011)_2 =$ **?**

Smallest $+ve$ number that can be represented using FP (6,2) rep.: **?**
Biggest $+ve$ number that can be represented using FP (6,2) rep.: **?**

Fixed Point Arithmetic

Consider a 16-bit binary representation, in which least significant 8 bits are used for precision then give the range of values that can be represented using the 16-bit representation.

- $+ve$ Values: 2^{-8} to $2^{7} - 2^{-8}$
- $-ve$ Values: -2^{7} to -2^{-8}
- Zero
- Resolution is **?**

Advantages of FP Arithmetic:

- Easy to implement and occupies less space.
- If performance is important than precision.
- Once can choose a trade off between range and precision.

Fixed Point Arithmetic

Consider a 16-bit binary representation, in which least significant 8 bits are used for precision then give the range of values that can be represented using the 16-bit representation.

- $+ve$ Values: 2^{-8} to $2^{7} - 2^{-8}$
- $-ve$ Values: -2^{7} to -2^{-8}
- Zero
- Resolution is **?**

Advantages of FP Arithmetic:

- Easy to implement and occupies less space.
- If performance is important than precision.
- Once can choose a trade off between range and precision.

IEEE 754 Representation of Real Numbers

- IEEE 754 format for Real Numbers.

	Sign	Biased Exponent	Mantissa/Fraction	
Single Precision N=32	1 bit	8 bits	23 bits	Bias Value:+127
Double Precision N=64	1 bit	11 bits	52 bits	Bias Value :+1023

  Biased Exponent=True Exponent + Bias Value

- Rep. of $(4.5)_{10}$ using Single Precision
  $(4.5)_{10} = (100.1)_2 = (1.001)_2 \times 2^2$
  Normalized Rep.: $(\pm 1.xxxxxx)_2 \times 2^E$, Where 'E' is a **True Exponent**, 'xxxxxx' is a **Fraction/Mantissa**.

- Biased Exponent = **2** + **127** = **129** = **1000 0001**

- Mantissa = **001** = **0010000 0000 0000 0000 0000**

- Sign= + ve = 0

- $(4.5)_{10} = (1.001)_2 \times 2^2$= 0 **10000001 00100000000000000000000**

IEEE 754 Representation of Real Numbers

- IEEE 754 format for Real Numbers.

	Sign	Biased Exponent	Mantissa/Fraction	
Single Precision N=32	1 bit	8 bits	23 bits	Bias Value:+127
Double Precision N=64	1 bit	11 bits	52 bits	Bias Value :+1023

Biased Exponent=True Exponent + Bias Value

- Rep. of $(4.5)_{10}$ using Single Precision
  $(4.5)_{10} = (100.1)_2 = (1.001)_2 \times 2^2$
  Normalized Rep.: $(\pm 1.xxxxxx)_2 \times 2^E$, Where 'E' is a **True Exponent**, 'xxxxxx' is a **Fraction/Mantissa**.
- Biased Exponent = **2** + **127** = **129** = **1000 0001**
- Mantissa = **001** = **0010000 0000 0000 0000 0000**
- Sign= + ve = 0
- $(4.5)_{10} = (1.001)_2 \times 2^2$= 0 **10000001 00100000000000000000000**

IEEE 754 Representation of Real Numbers

- IEEE 754 format for Real Numbers.

	Sign	Biased Exponent	Mantissa/Fraction	
Single Precision N=32	1 bit	8 bits	23 bits	Bias Value:+127
Double Precision N=64	1 bit	11 bits	52 bits	Bias Value :+1023

  Biased Exponent=True Exponent + Bias Value

- Rep. of $(4.5)_{10}$ using Single Precision
  $(4.5)_{10} = (100.1)_2 = (1.001)_2 \times 2^2$
  Normalized Rep.: $(\pm 1.xxxxxx)_2 \times 2^E$, Where 'E' is a **True Exponent**, 'xxxxxx' is a **Fraction/Mantissa**.
- Biased Exponent = **2** + **127** = **129** = **1000 0001**
- Mantissa = **001** = **0010000 0000 0000 0000 0000**
- Sign= + ve = 0
- $(4.5)_{10} = (1.001)_2 \times 2^2$= 0 **10000001 00100000000000000000000**

IEEE 754 Representation of Real Numbers

- IEEE 754 format for Real Numbers.

Sign	Biased Exponent	Mantissa/Fraction

Single Precision N=32	1 bit	8 bits	23 bits	Bias Value:+127
Double Precision N=64	1 bit	11 bits	52 bits	Bias Value :+1023

Biased Exponent=True Exponent + Bias Value

- Rep. of $(4.5)_{10}$ using Single Precision
  $(4.5)_{10} = (100.1)_2 = (1.001)_2 \times 2^2$
  Normalized Rep.: $(\pm 1.xxxxxx)_2 \times 2^E$, Where 'E' is a **True Exponent**, 'xxxxxx' is a **Fraction/Mantissa**.
- Biased Exponent $= \mathbf{2} + \mathbf{127} = \mathbf{129} = \mathbf{1000\ 0001}$
- Mantissa $= \mathbf{001} = \mathbf{0010000\ 0000\ 0000\ 0000\ 0000}$
- Sign$= +$ ve $= 0$
- $(4.5)_{10} = (1.001)_2 \times 2^2 = 0$ **10000001 00100000000000000000000**

IEEE 754 Representation of Real Numbers

- IEEE 754 format for Real Numbers.

	Sign	Biased Exponent	Mantissa/Fraction	
Single Precision N=32	1 bit	8 bits	23 bits	Bias Value:+127
Double Precision N=64	1 bit	11 bits	52 bits	Bias Value :+1023

Biased Exponent=True Exponent + Bias Value

- Rep. of $(4.5)_{10}$ using Single Precision
  $(4.5)_{10} = (100.1)_2 = (1.001)_2 \times 2^2$
  Normalized Rep.: $(\pm 1.xxxxxx)_2 \times 2^E$, Where 'E' is a**True Exponent**, 'xxxxxx' is a **Fraction/Mantissa**.
- Biased Exponent $=$ **2** $+$ **127** $=$ **129** $=$ **1000 0001**
- Mantissa $=$ **001** $=$ **0010000 0000 0000 0000 0000**
- Sign$= +$ ve $= 0$
- $(4.5)_{10} = (1.001)_2 \times 2^2 = 0$ **10000001 00100000000000000000000**

IEEE 754 Representation of Real Numbers

- IEEE 754 format for Real Numbers.

	Sign	Biased Exponent	Mantissa/Fraction	
Single Precision N=32	1 bit	8 bits	23 bits	Bias Value:+127
Double Precision N=64	1 bit	11 bits	52 bits	Bias Value :+1023

  Biased Exponent=True Exponent + Bias Value

- Rep. of $(4.5)_{10}$ using Single Precision
  $(4.5)_{10} = (100.1)_2 = (1.001)_2 \times 2^2$
  Normalized Rep.: $(\pm 1.xxxxxx)_2 \times 2^E$, Where 'E' is a **True Exponent**, 'xxxxxx' is a **Fraction/Mantissa**.
- Biased Exponent $=$ $\mathbf{2 + 127 = 129 = 1000\ 0001}$
- Mantissa $=$ $\mathbf{001 = 0010000\ 0000\ 0000\ 0000\ 0000}$
- Sign$= +$ ve $= 0$
- $(4.5)_{10} = (1.001)_2 \times 2^2 = 0$ **10000001 00100000000000000000000**

IEEE 754 Representation of Real Numbers

- IEEE 754 format for Real Numbers.

	Sign	Biased Exponent	Mantissa/Fraction	
Single Precision N=32	1 bit	8 bits	23 bits	Bias Value:+127
Double Precision N=64	1 bit	11 bits	52 bits	Bias Value :+1023

  Biased Exponent=True Exponent + Bias Value

- Rep. of $(4.5)_{10}$ using Single Precision
  $(4.5)_{10} = (100.1)_2 = (1.001)_2 \times 2^2$
  Normalized Rep.: $(\pm 1.xxxxxx)_2 \times 2^E$, Where 'E' is a **True Exponent**, 'xxxxxx' is a **Fraction/Mantissa**.
- Biased Exponent $= \mathbf{2} + \mathbf{127} = \mathbf{129} = \mathbf{1000\ 0001}$
- Mantissa $= \mathbf{001} = \mathbf{0010000\ 0000\ 0000\ 0000\ 0000}$
- Sign$= +$ ve $= 0$
- $(4.5)_{10} = (1.001)_2 \times 2^2 = 0$ **10000001 00100000000000000000000**

IEEE 754 Representation of Real Numbers

- IEEE 754 format for Real Numbers.

Sign	Biased Exponent	Mantissa/Fraction

Single Precision N=32	1 bit	8 bits	23 bits	Bias Value:+127
Double Precision N=64	1 bit	11 bits	52 bits	Bias Value :+1023

Biased Exponent=True Exponent + Bias Value

- Rep. of $(4.5)_{10}$ using Single Precision
  $(4.5)_{10} = (100.1)_2 = (1.001)_2 \times 2^2$
  Normalized Rep.: $(\pm 1.xxxxxx)_2 \times 2^E$, Where 'E' is a**True Exponent**, 'xxxxxx' is a **Fraction/Mantissa**.
- Biased Exponent $= \mathbf{2} + \mathbf{127} = \mathbf{129} = \mathbf{1000\ 0001}$
- Mantissa $=$ **001** $=$ **0010000 0000 0000 0000 0000**
- Sign$= +$ ve $= 0$
- $(4.5)_{10} = (1.001)_2 \times 2^2 = 0$ **10000001 00100000000000000000000**

IEEE 754

- Biased Exponent=True Exponent + Bias Value, where 1$\leq$ Biased Exponent $\leq (2^{Length\ of\ Biased\ Exponent} - 2)$.
- Single Precision (N=32), 1$\leq$ Biased Exponent $\leq$254.
- Biased Exponent = 0,
  - Mantissa = ± 0, then Value is ± 0.
  - Mantissa $\neq 0$, then Value is **not a normalized number**.
- Biased Exponent = 255,
  - Mantissa = ± 0, then Value is $\pm\infty$.
  - Mantissa $\neq 0$, then Value is **NAN**.
- Range of positive values: $[1.0 \times 2^{-126}, (2 - 2^{-23}) \times 2^{127}]$
- Range of negative values: $[-(2 - 2^{-23}) \times 2^{127}, -1.0 \times 2^{-126}]$
- Single Precision Number Resolution: $2^{-23} \times 2^{TrueExponent}$

BCD Representation

BCD: Binary Coded Decimal

It uses a 4-bit binary number to represent each decimal digit.

Decimal Number	**BCD Rep.**
0	0000
1	0001
2	0010
3	0011
9	1001
10	0001 0000
25	0010 0101
99	1001 1001

Table 3: BCD equivalent of a decimal number.

All the best ☺