# Number Systems

# Review on Number Systems

*Decimal, Binary, and Hexadecimal*

# Base-N Number System

Base N

N Digits: 0, 1, 2, 3, 4, 5, ..., N-1

Example: $1045_N$

Positional Number System

$$N^{n-1} \cdots N^4 N^3 N^2 N^1 N^0$$

$$d_{n-1} \cdots d_4\, d_3\, d_2\, d_1\, d_0$$

- Digit $d_o$ is the least significant digit (LSD).
- Digit $d_{n-1}$ is the most significant digit (MSD).

# Decimal Number System

**Base N**

**N Digits: 0, 1, 2, 3, 4, 5, …, N-1**

**Example: $1045_N$**

**Positional Number System**

$$N^{n-1} \cdots N^4 N^3 N^2 N^1 N^0$$

$$d_{n-1} \cdots d_4\, d_3\, d_2\, d_1\, d_0$$

- Digit $d_o$ is the least significant digit (LSD).
- Digit $d_{n-1}$ is the most significant digit (MSD).

# Binary Number System

Base 2

Two Digits: 0, 1

Example: $1010110_2$

Positional Number System

$$2^{n-1} \cdots 2^4 2^3 2^2 2^1 2^0$$

$$b_{n-1} \cdots b_4 \, b_3 \, b_2 \, b_1 \, b_0$$

**B**inary Dig**its** are called Bits

Bit $b_o$ is the least significant bit (LSB).

Bit $b_{n-1}$ is the most significant bit (MSB).

# Definitions

nybble = 4 bits

byte = 8 bits

(short) word = 2 bytes = 16 bits

(double) word = 4 bytes = 32 bits

(long) word = 8 bytes = 64 bits

1K (kilo or “kibi”) = 1,024

1M (mega or “mebi”) = (1K)*(1K) = 1,048,576

1G (giga or “gibi”) = (1K)*(1M) = 1,073,741,824

# Hexadecimal Number System

Base 16

Sixteen Digits: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

Example: $EF56_{16}$

Positional Number System

$$16^{n-1} \cdots 16^{4} 16^{3} 16^{2} 16^{1} 16^{0}$$

| | |
|---|---|
| 0000 | 0 |
| 0001 | 1 |
| 0010 | 2 |
| 0011 | 3 |

| | |
|---|---|
| 0100 | 4 |
| 0101 | 5 |
| 0110 | 6 |
| 0111 | 7 |

| | |
|---|---|
| 1000 | 8 |
| 1001 | 9 |
| 1010 | A |
| 1011 | B |

| | |
|---|---|
| 1100 | C |
| 1101 | D |
| 1110 | E |
| 1111 | F |

# Binary Addition

- Single Bit Addition Table

0 + 0 = 0

0 + 1 = 1

1 + 0 = 1

1 + 1 = 10   Note "carry"

# Hex Addition

- 4-bit Addition

4 + 4 = 8

4 + 8 = C

8 + 7 = F

F + E = 1D    Note "carry"

# Hex Digit Addition Table

| + | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| **1** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 10 |
| **2** | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 10 | 11 |
| **3** | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 10 | 11 | 12 |
| **4** | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 10 | 11 | 12 | 13 |
| **5** | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 10 | 11 | 12 | 13 | 14 |
| **6** | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 10 | 11 | 12 | 13 | 14 | 15 |
| **7** | 7 | 8 | 9 | A | B | C | D | E | F | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| **8** | 8 | 9 | A | B | C | D | E | F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| **9** | 9 | A | B | C | D | E | F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| **A** | A | B | C | D | E | F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| **B** | B | C | D | E | F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A |
| **C** | C | D | E | F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B |
| **D** | D | E | F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C |
| **E** | E | F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D |
| **F** | F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E |

# Decimal→ binary/octal/hex conversion

**Binary**

| | Quotient | Remainder |
|---|---|---|
| 56÷2= | 28 | 0 |
| 28÷2= | 14 | 0 |
| 14÷2= | 7 | 0 |
| 7÷2= | 3 | 1 |
| 3÷2= | 1 | 1 |
| 1÷2= | 0 | 1 |

**Octal**

| | Quotient | Remainder |
|---|---|---|
| 56÷8= | 7 | 0 |
| 7÷8= | 0 | 7 |

$56_{10}=111000_2$

$56_{10}=70_8$

Why does this work? $N=56_{10}=111000_2$

Q=N/2=56/2=111000/2=11100 remainder 0

Each successive divide liberates an LSB (least significant bit)

# Binary → hex/decimal/octal conversion

## Conversion from binary to octal/hex

Binary: 10011110001

Octal: 10 | 011 | 110 | 001=$2361_8$

Hex: 100 | 1111 | 0001=$4F1_{16}$

## Conversion from binary to decimal

$101_2 = 1\times2^2 + 0\times2^1 + 1\times2^0 = 5_{10}$

$63.4_8 = 6\times8^1 + 3\times8^0 + 4\times8^{-1} = 51.5_{10}$

$A1_{16} = 10\times16^1 + 1\times16^0 = 161_{10}$

# Gray and BCD codes

| Decimal Symbols | Gray Code |
|---|---|
| 0 | 0000 |
| 1 | 0001 |
| 2 | 0011 |
| 3 | 0010 |
| 4 | 0110 |
| 5 | 0111 |
| 6 | 0101 |
| 7 | 0100 |
| 8 | 1100 |
| 9 | 1101 |

| Decimal Symbols | BCD Code |
|---|---|
| 0 | 0000 |
| 1 | 0001 |
| 2 | 0010 |
| 3 | 0011 |
| 4 | 0100 |
| 5 | 0101 |
| 6 | 0110 |
| 7 | 0111 |
| 8 | 1000 |
| 9 | 1001 |

# THANK YOU