Binary Systems

Dr. Bassem A. Abdullah
Computer and Systems Department

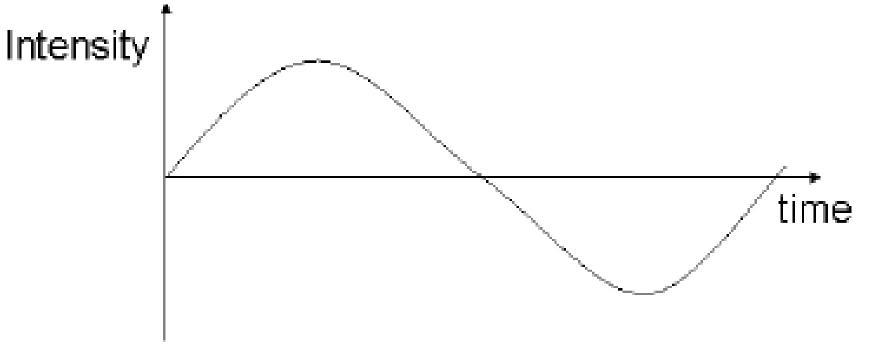
Lectures Prepared by Dr.Mona Safar, Edited and Lectured by Dr.Bassem A. Abdullah

Outline

1. Digital Systems
2. Binary Numbers
3. Number Base Conversions
4. Octal and Hexadecimal Numbers
5. Complements
6. Signed Binary Numbers
7. Binary Codes
8. Binary Logic

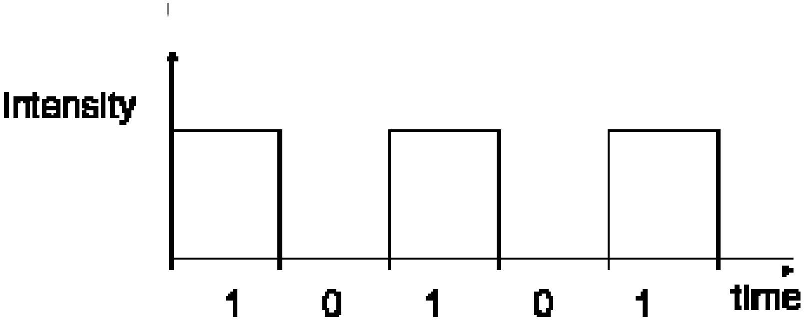
Analog Vs Digital Signals

- **Analog signals** are continuous electrical signals that vary in time.



Analog Vs Digital Signals

- **Digital signals** are non-continuous, they change in individual steps. They consist of pulses or digits with discrete levels or values.
- Binary System manipulates discrete data represented in binary form.
- Described by a signal of two amplitude levels called 1 or 0, HIGH or LOW, TRUE or FALSE, On or OFF.



Binary Digital System

- A Binary digit is called a **bit.**
- A **Bit** has one of two possible values (0 or 1).
- A **Byte** is an 8-bit chunk (1 Byte = 8 bits)
- 1 Kilobyte (KB) = 2^{10} bytes = 1,024 bytes
- 1 Megabyte (MB) = 2^{20} bytes = 1,048,576 bytes
- 1 Gigabyte (GB) = 2^{30} bytes = 1,073,741,824 bytes
- Terabyte (TB) = 2^{40} bytes = 1,099,511,627,776 bytes

Computer is a Binary System

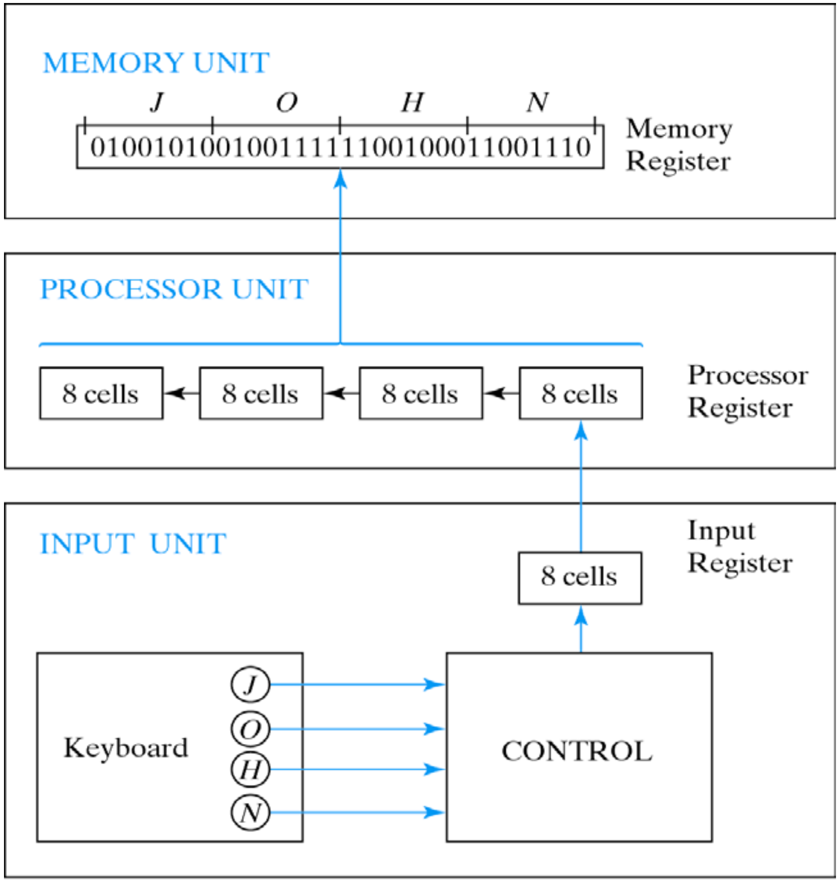


Fig. 1-1 Transfer of information with registers

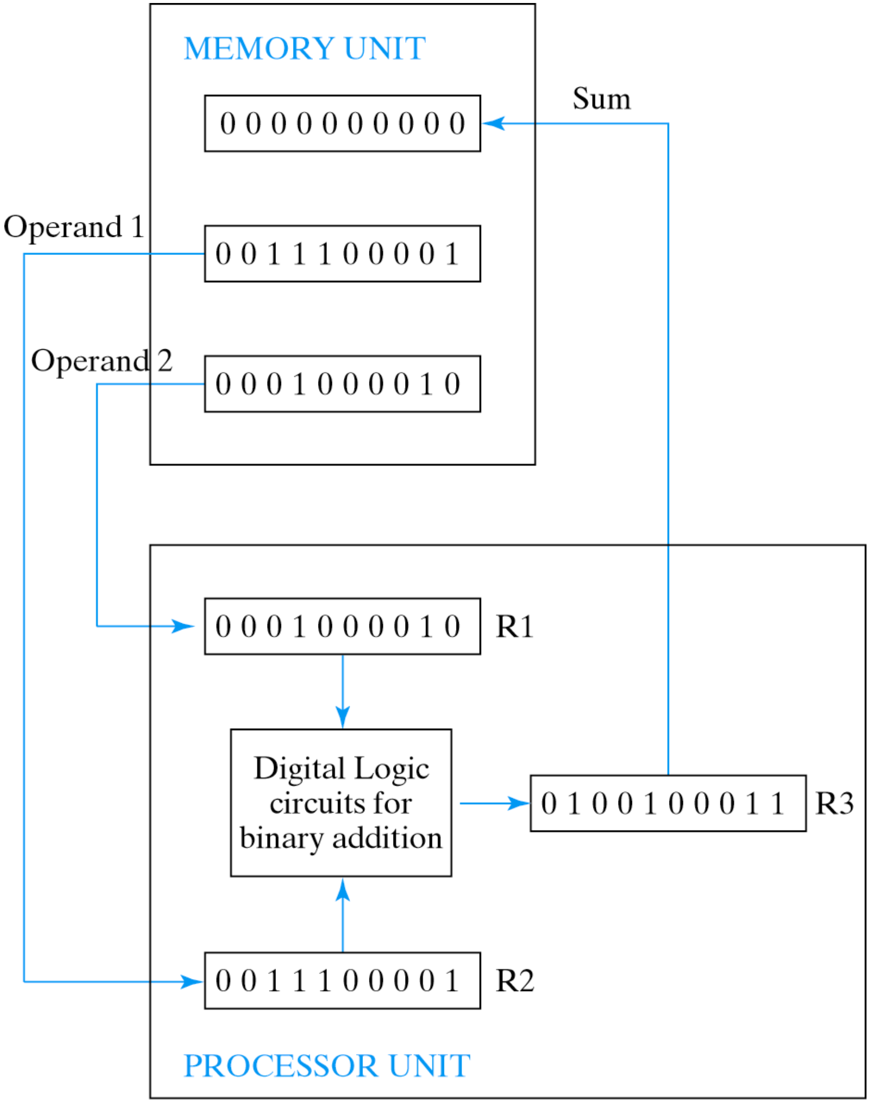


Fig. 1-2 Example of binary information processing

Decimal Numbering System

- A digit is either 0, 1, 2, ….. Or 9 (10 digits, 10 fingers ☺)
- Example: Decimal number 9457

9	4	5	7
↓	↓	↓	↓
Thousands	Hundreds	Tens	Units
10^3	10^2	10^1	10^0

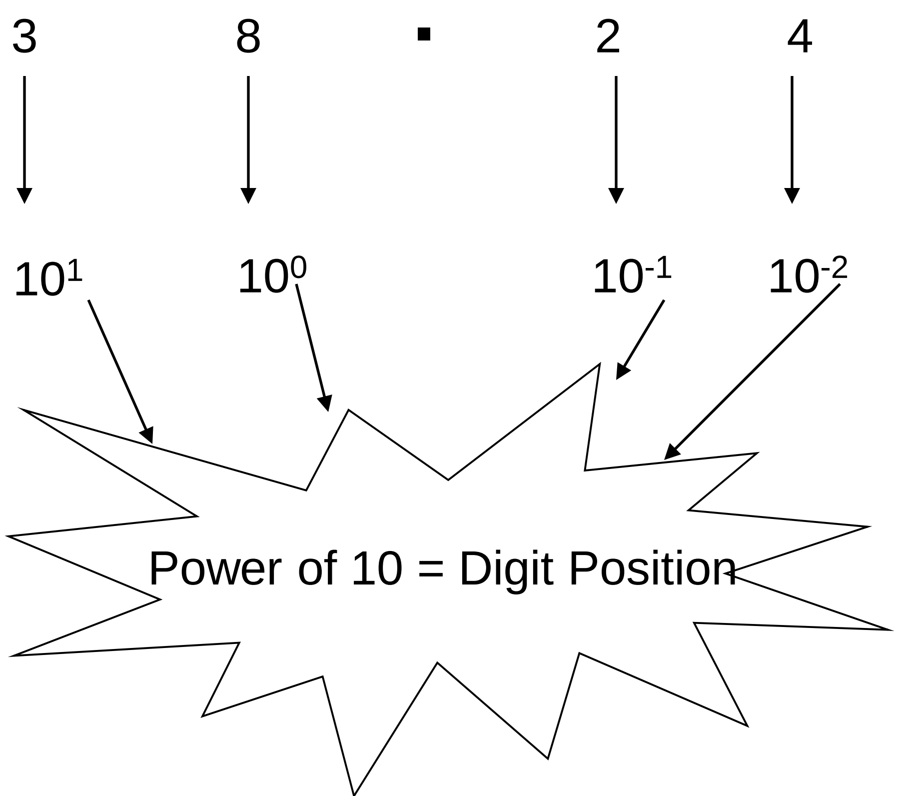
9 thousands + 4 hundreds + 5 tens + 7 units

$= 9 * 10^3 + 4 * 10^2 + 5 * 10^1 + 7 * 10^0$

= 9457

Decimal Numbering System (Cont.)

- Example: Decimal number 38.24



$3 * 10^1 + 8 * 10^0 + 2 * 10^{-1} + 4 * 10^{-2}$
$= 30 + 8 + 0.2 + 0.04$
$= 38.24$

Decimal Numbering System (Cont.)

- Decimal number system is called **base 10** or **radix 10** because:
  - It uses **10** digits (0 to 9).
  - Each digit is multiplied by power of **10** according to its position.

Binary Numbering System

- Decimal number system is called **base 2** or **radix 2** because:
  - It uses **2** digits (0 or 1).
  - Each digit is multiplied by power of **2** according to its position.

Remember: A binary digit is called a bit.

Binary Numbering System (Cont.)

- Example: binary number $(101011)_2$

1	0	1	0	1	1
↓	↓	↓	↓	↓	↓
2^5	2^4	2^3	2^2	2^1	2^0

$= 1 * 2^5 + 0 * 2^4 + 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1 * 2^0$

$= 32 + 8 + 2 + 1$

$= 43$

Conversion from Binary to Decimal

- To convert from binary to decimal add the numbers with powers of two corresponding to the bits that are equal to 1.

Example: Convert $(110100)_2$ to decimal

$(110100)_2 = 1*2^5 + 1*2^4 + 1*2^2$

$= 32 + 16 + 4 = (52)_{10}$

Conversion from Binary to Decimal

Example: Convert $(11010.11)_2$ to decimal

$(11010.11)_2$

$= 1*2^4 + 1*2^3 + 1*2^1 + 1*2^{-1} +1*2^{-2}$

$= 16 + 8 + 2 + 0.5 + 0.25 = (26.75)_{10}$

Conversion from Decimal to Binary

- To convert from decimal to binary:
  - Divide the number by two. The remainder (which will be either 0 or 1) is the rightmost binary digit.
  - Divide the quotient by two. This remainder will be the next binary digit to the left.
  - Continue dividing the successive quotients by two and using the remainder as the next binary digit to the left, and stop when the quotient is finally zero.

Conversion from Decimal to Binary (Cont.)

Example: Convert $(52)_{10}$ to binary

52 / 2 = 26	Reminder	0	Rightmost (**LSB**: Least significant bit)
26 / 2 = 13	Reminder	0	
13 / 2 = 6	Reminder	1	
6 / 2 = 3	Reminder	0	
3 / 2 = 1	Reminder	1	
1 / 2 = 0	Reminder	1	Leftmost (**MSB**: Most significant bit)

$52 = (110100)_2$

Conversion from Decimal to Binary (Cont.)

For fraction part employ iterative multiplication

Example: Convert $(0.6875)_{10}$ to binary

0.6875	* 2	= 1.375	Integer	1
0.375	* 2	= 0.75	Integer	0
0.75	* 2	= 1.5	Integer	1
0.5	* 2	= 1	Integer	1

$0.6875 = (0.1011)_2$

$52.6875 = (110100.1011)_2$

How many bits are needed?

- n bits can represent 2^n unsigned integers from 0 to $2^n - 1$.
- Example: What is the max number represented by 12 bits?
  - 2^{12} can represent 4*1024 = 4096.
  - The max number that can be represented is 4095 = $(111111111111)_2$
- Example: How many bits are needed to represent 100 distinct numbers?
  - $2^x = 100$
  - $X = \text{ceil}(\log_2 100) = \text{ceil}(6.67) = 7$ bits

Binary Arithmetic

- 0 + 0 = 0
- 0 + 1 = 1
- 1 + 0 = 1
- 1 + 1 = 10
- 1 + 1 + 1 = 11

Binary Arithmetic (Cont.)

- **Example**: Add the two numbers 01111 (15) and 10111 (23).

```
      1 1 1 1      ← carry bits
    0 1 1 1 1
+   1 0 1 1 1
  -----------
  1 0 0 1 1 0
```

- Check $(100110)_2 = 38$

Binary Arithmetic (Cont.)

- **Example**: Subtract the number 100111 from 101101.

```
              1 1      ← borrow bits
        1 0 1 1 0 1
      - 1 0 0 1 1 1
        -----------
        0 0 0 1 1 0
```

Binary Arithmetic (Cont.)

- **Example**: Multiply 101 by 101.

```
        1 0 1
  *     1 0 1
  -----------
        1 0 1
+     0 0 0
    1 0 1
  -----------
    1 1 0 0 1
```

Octal Numbering System

- Octal number system is called **base 8** or **radix 8** because:
  - It uses **8** digits (0 to 7).
  - Each digit is multiplied by power of **8** according to its position.

Example: $(127)_8 = 1 * 8^2 + 2 * 8^1 + 7 * 8^0 = (87)_{10}$

Conversion from Decimal to Octal

Example: Convert $(52)_{10}$ to octal

52 / 8 = 6 Reminder	4	↑ Rightmost (LSB: Least significant bit)
6 / 8 = 0 Reminder	6	Leftmost (MSB: Most significant bit)

$52 = (64)_8$

Conversion from Decimal to Octal

Example: Convert $(0.513)_{10}$ to octal

0.513 * 8 = 4.104	Integer	4	↓
0.104 * 8 = 0.832	Integer	0	
0.832 * 8 = 6.656	Integer	6	
0.656 * 8 = 5.248	Integer	5	
0.248 * 8 = 1.984	Integer	1	
0.984 * 8 = 7.872	Integer	7	

$0.513 = (0.406517\ldots)_8$

$52.513 = (64.406517)_8$

Hexadecimal Numbering System

- Octal number system is called **base 16** or **radix 16** because:
  - It uses **16** digits (0 to 9, A, B, C, D, E, F).
  - Each digit is multiplied by power of **16** according to its position.

Example: $(B65F)_{16} = (B65F)_H = 11 * 16^3 + 6 * 16^2 + 5 * 16^1 + 15 * 16^0 = (46687)_{10}$

Conversion from Decimal to Hexadecimal

Example: Convert $(61)_{10}$ to hexadecimal

61 / 16 = 3	Reminder	13=D ↑
3 / 16 = 0	Reminder	3

$61 = (3D)_{16}$

Why Octal and Hexadecimal?

- A big problem with the binary system is the large number of bits used to represent numbers. Octal and Hexadecimal are more compact.

  Example: The value $(4095)_{10}$ can be represented by 12 binary digits, or 3 Hexa digits, or 4 Octal digits.
- It's simple to convert them to binary and vice versa.

Binary to Octal

- $2^3 = 8$
- To convert from binary to octal, partition the number into groups of **3** bits, and convert each group to its equivalent octal digit.
- Example: Convert 1011111010 to Octal

001	011	111	010
1	3	7	2

$(1011111010)_2 = (1372)_8$,

Octal to Binary

- The reverse of the proceeding procedure, each octal digit is converted to its equivalent 3 bits.
- Example: Convert $(376)_8$ =to binary.

3 7 6

11 111 110

$(376)_8 = (11111110)_2$

Example

- $(10110001101011.1111001\ )_2$

  $= (010\ 110\ 001\ 101\ 011\ .\ 111\ 100\ 100)_2$

  $= (\ \ 2\ \ \ 6\ \ \ 1\ \ \ 5\ \ \ 3\ \ .\ \ 7\ \ \ 4\ \ \ 4)_8$

  $= (26153.744)_8$

- $(673.124)_8 = (110\ 111\ 011\ .\ 001\ 010\ 100)_2$

  $= (110111011.0010101)_2$

Binary to Hexadecimal

- $2^4 = 16$
- To convert from binary to hexadecimal, partition the number into groups of **4** bits, and convert each group to its equivalent hexadecimal digit.
- Example: Convert 1011111010 to hexadecimal

0010 1111 1010

2 F A

$(1011111010)_2 = (2FA)_{16}$

Hexadecimal to Binary

- The reverse of the proceeding procedure, each hexadecimal digit is converted to its equivalent 4 bits.
- Example: Convert $(9C6)_{16}$ =to binary.

9	C	6
1001	1100	0110

$(9C6)_8 = (100111000110)_{16}$

Example

- $(1011000110101 1.1111001)_2$
  $= (0010\ 1100\ 0110\ 1011\ .\ 1111\ 0010)_2$
  $= (\ \ 2\quad C\quad 6\quad B\ .\ F\quad 2)_{16}$
  $= (2C6B.F2)_{16}$ = 2C6B.F2 H
  = 0x 2C6B.F2
- $(306.D)_{16} = (0011\ 0000\ 0110\ .\ 1101)_2$
  $= (1100000110.1101)_2$

Decimal	Binary	Octal	Hexadecimal
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Complement

- Logic Operation : 0 ↔ 1
- Arithmetic Operation : Subtraction

- For base-r:
  - r’s complement
  - r-1’s complement

Diminished Radix (r-1)'s Complement

- Given a number N in base-r represented in n digits, the (r-1)'s complement of N:

$$(r^n - 1) - N$$

- For r=10 → r-1 =9
  - 9's complements of N = $(10^n-1) - N$
- For r=2 → r-1 =1
  - 1's complements of N = $(2^n-1) - N$

Diminished Radix (r-1)'s Complement Examples

- The 9's complement of 546700
  - r=10, n= 6, → 10^6-1= 999999
  - 9's complements of 546700
    = 999999 - 546700
    = 453299
- The 1's complement of 1011000
  - r=2, n=7, → 2^7-1= 1111111
  - 1's complements of 1011000
    = 1111111 - 1011000
    = 0100111

1’s Complement

- 1’s complement is formed by changing every 1 to 0 and every 0 to 1

  i.e.: Toggle each bit

  Example: 1’s complement of 101100 is 010011

Diminished Radix r's Complement

- Given a number N in base-r represented in n digits, the r's complement of N:

$$r^n - N$$

$$(r\text{-}1)\text{'s complement} + 1$$

- For r=10
  - 10's complements of N = $(10^n) - N$
- For r=2
  - 2's complements of N = $(2^n) - N$

Diminished Radix r's Complement Examples

- The 10's complement of 546700
  - r=10, n= 6, → $10^6 = 1000000$
  - 10's complements of 546700

    = 1000000 - 546700

    = 453299 + 1

    = 453300
- The 2's complement of 1011000
  - r=2, n=7, → $2^7 = 10000000$
  - 1's complements of 1011000

    = 10000000 - 1011000

    = 0100111 + 1

    = 0101000

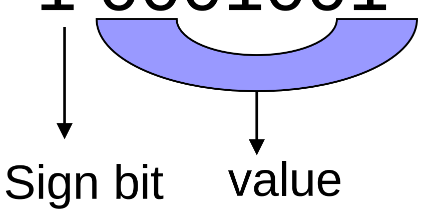
2’s Complement

- 2’s complement is formed by leaving all least significant 0’s and first 1 unchanged and inverting all others.

  Example: 2’s complement of 101100 is 010100

Signed Binary Numbers

- A sign bit at the leftmost position of the number is added.
- The sign bit is 0 for positive and 1 for negative.
- Example: Representing 9 and -9 using 8 bits:
  - 9 : 0 0001001
  - -9 : 1 0001001



Different representation of –ve values

- Presentation formats of value -9
  - Signed-magnitude 10001001
  - Signed 1’s complement 11110110
  - Signed 2’s complement 11110111