Objectives

- Use two different methods to perform decimal-to-binary conversion.

- Convert from the hexadecimal or octal numbering systems to either the decimal or binary numbering systems.

- Cite several advantages of the octal and hexadecimal numbering systems.

- Express decimal numbers using the BCD code.

- Understand the difference between the BCD code and straight binary code.

Introduction

- Binary numbering system is the most important in digital systems.
- Computers and other digital systems can only work in binary.
- The decimal system is important because it is used to represent numbers in the real world.
- Octal (*base-8*) and hexadecimal (*base-16*) are used to represent large binary numbers

2.1 Binary-to-Decimal Conversions

- Converting a Binary Number to its Decimal equivalent basically involves 2 steps:
  1. Multiply each bit of the Binary number by its positional weight.
  2. Sum the individual products to obtain the decimal equivalent.

Eg, the Decimal equivalent of Binary 11011_2 is:

MSB				LSB		
1	1	0	1	1_2		(Binary)
1×2^4 +	1×2^3 +	0×2^2 +	1×2^1 +	1×2^0		
16 +	8 +	0 +	2 +	1	= 27	(Decimal)

What is the Decimal equivalent of Binary 110101 ?

2.2 Decimal-to-Binary Conversions

- There are basically 2 methods of converting Decimal to Binary:

1. Factoring the number into powers of Twos

   For example, consider:

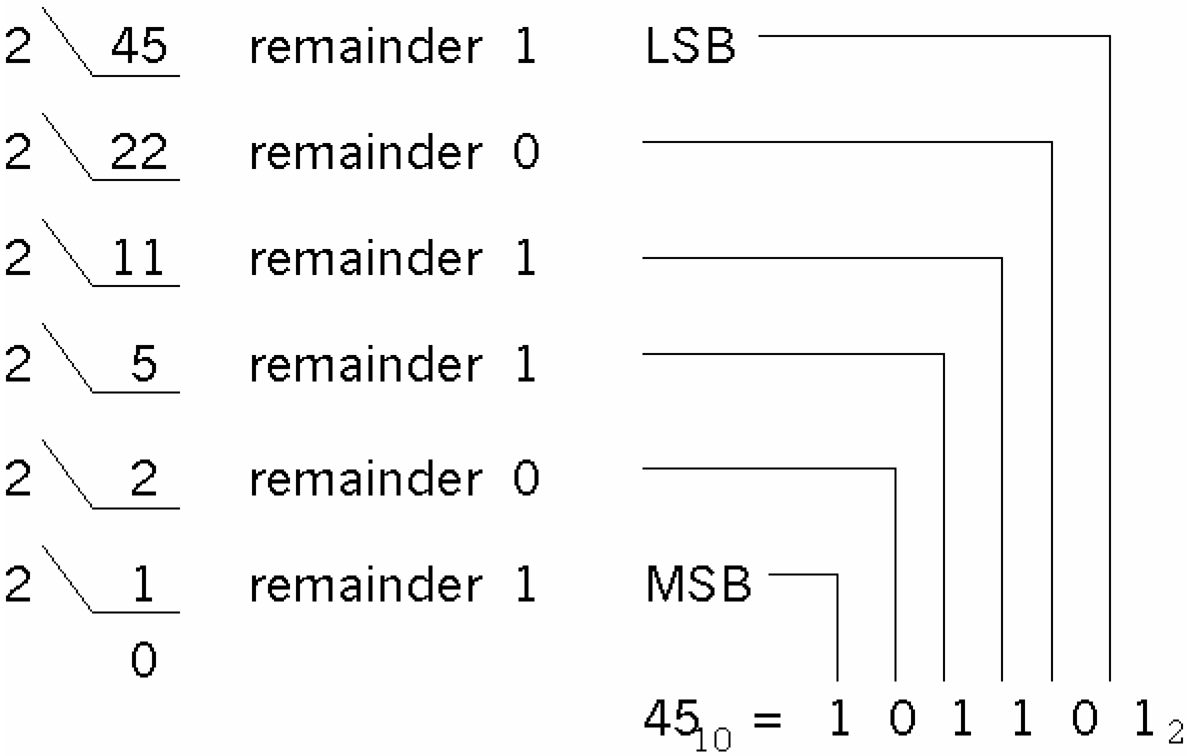
$$157_{10} = 128 + 16 + 8 + 4 + 1$$
$$= 2^7 + 2^4 + 2^3 + 2^2 + 2^0$$

   From which the equivalent Binary number is obtained.

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	0	0	1	1	1	0	1

2. Repeated Division by 2

This is a more methodical approach in which the decimal number is repeatedly divided by 2 with the remainders in each division step being each bit of the binary equivalent.



NB: The 1st remainder is the LSB & the last remainder is the MSB of the binary number.

Binary Counting

2^4	2^3	2^2	2^1	2^0	Decimal Equivalent
0	0	0	0	0	0
0	0	0	0	1	1
0	0	0	1	0	2
0	0	0	1	1	3
0	0	1	0	0	4
0	0	1	0	1	5
0	0	1	1	0	6
0	0	1	1	1	7
0	1	0	0	0	8
0	1	0	0	1	9
0	1	0	1	0	10
0	1	0	1	1	11
0	1	1	0	0	12
0	1	1	0	1	13
0	1	1	1	0	14
0	1	1	1	1	15
1	0	0	0	0	16

2.3 Octal Numbering System

- This is a Base 8 numbering system which uses the only eight digits: 0 to 7

The weights of an Octal number are in powers of Eights.

$$\ldots\ldots\ 8^3\quad 8^2\quad 8^1\quad 8^0\quad .\quad 8^{-1}\quad 8^{-2}\quad 8^{-3}\ \ldots\ldots$$

Octal point

Octal to Decimal conversion

- Converting an Octal number to its decimal equivalent basically involves multiplying each octal digit by its positional weight and then summing the individual products.

- Example:

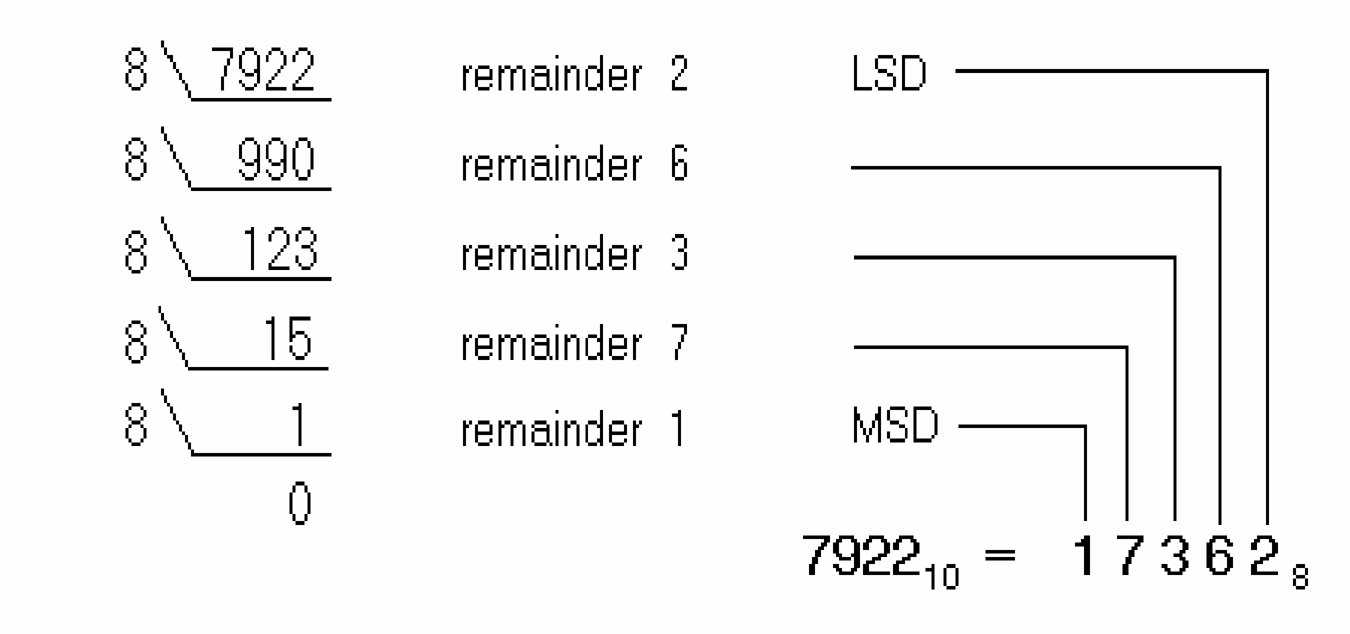
  Convert Octal 572 to its decimal equivalent,

$$
\begin{aligned}
572_8 &= (5*8^2) + (7*8^1) + (2*8^0) \\
&= (5*64) + (7*8) + (2*1) \\
&= 320 + 56 + 2 \\
&= 378_{10}
\end{aligned}
$$

- The reverse process of converting Decimal to Octal can be performed using the repeated division method.

- The number is repeatedly divided by 8, with the remainder in each division step being an Octal digit.

For Example: convert Decimal 7922 to Octal



Usefulness of Octal System

- Computers can only handle ***binary*** numbers. Internally computers usually handle data bits in multiples of 8 bits.

  For Example:
  286 PC uses 16 bit binary numbers internally.
  486 PC uses 32 bit binary numbers internally.
  Pentium PC uses 32 bit binary numbers internally.

- As a shorthand method to express large binary numbers because Octal numbers are easily converted to Binary & vice versa.

- Converting Binary to Octal

  Group the binary bits into 3 bit strings starting with the LSB & then convert each 3 bit string to its equivalent Octal digit.

  Eg. Convert 110011001101_2 to Octal

  Ans 110 011 001 101

  6 3 1 5

  Hence, $110011001101_2 = 6315_8$

- Converting Octal to Binary
  Convert each octal digit to its 3-bit binary equivalent on a digit by digit basis

  Eg: Convert 726_8 to binary

  Ans: 7 2 6

  111 010 110

  Hence, $726_8 = 111\ 010\ 110_2$

2.4 Hexadecimal Numbering System

- Hexadecimal is a Base 16 numbering system with 16 possible digits: 0 - 9 & A - F.
- Like the Octal system, the Hex system is used mainly as a shorthand method to represent Binary numbers.
- To convert Hex-to-Decimal, each hex digit is first multiplied by its positional weight, and then followed subsequently by a summation of the individual products.

Eg. Convert $FACE_{16}$ to decimal

$$\begin{aligned}
\mathbf{FACE_{16}} &= (15 \times 16^3) + (10 \times 16^2) + (12 \times 16^1) + (14 \times 16^0) \\
&= (15 \times 4096) + (10 \times 256) + (12 \times 16) + (14 \times 1) \\
&= 61440 + 2560 + 192 + 14 \\
&= 64206_{10}
\end{aligned}$$

- Similarly, Decimal-to-Hex Conversion can be performed using the repeated division by 16 approach.

Convert 8359_{10} to Hexadecimal

16 \ 8359	remainder	7	LSB
16 \ 522	remainder	A	
16 \ 32	remainder	0	
16 \ 2	remainder	2	MSB
16 \ 0			

$8359 = 2\ 0\ A\ 7_{16}$

NB: In the second step, dividing 522 by 16, yields a remainder of 10 decimal which is the same as Hex A.

- ***Why do we use Hex?***

  Computers work only in binary which tends to require a large number of digits.

  For the convenience of humans in the real world, it is easier to represent long strings of Binary numbers using the Hexadecimal numbering system as it is easily converted into Binary & vice versa.

  Eg: In Assembly language programming, data and addresses of memory locations are usually expressed in Hexadecimal.

- Binary-to-Hexadecimal Conversion

  Group the binary bits into 4 bit strings starting with the LSB & then convert each 4-bit string to the equivalent Hex digit.

  Eg: Convert Binary 1011101101 to Hex

  $$101110101101_2 \quad = \quad 1011 \quad 1010 \quad 1101$$

  $$B \quad A \quad D_{16}$$

- Hexadecimal-to-Binary Conversion

  This basically involves converting each Hex digit of the number into its 4 bit equivalent.

  Eg: Convert $5DE_{16}$ to Binary

  $$5DE_{16} \quad = \quad 5 \quad D \quad E$$

  $$0101 \quad 1101 \quad 1110$$

- Hex-to-Octal Conversion

As can be observed, the ease of converting from Hex to Binary and from Octal to Binary, will likewise lead to easy convertibility between Hex and Octal.

Eg: Convert Hex number 49FD to Octal

4	9	F	D	Hex
0100	1001	1111	1101	Binary

Re-grouping the bits,

100	100	111	111	101	Binary
4	4	7	7	5	Octal

Hence $49FD_{16} = 44775_8$

Binary Coded Decimal (BCD)

- When numbers, letters or words are represented by symbols, it's called a *CODE*

  Examples include*:* Morse Code
  ASCII Code
  EBCDIC

- BCD (Binary Coded Decimal) is *encoded* decimal - each decimal digit of a number is represented by its equivalent 4 bit binary code in accordance to the following table:

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

Eg: Express Decimal 2493 in BCD

2	4	9	3	Decimal
0010	0100	1001	0011	BCD Code

Decimal 2493 = 0010 0100 1001 0011 BCD

- It is important to note that the every decimal digit converts to 4 bits according to the BCD conversion table.
- Hence in the example shown, the leading zeros must NOT be removed.
- To express BCD in decimal, distribute the given bit pattern into groups of 4-bits and then convert each group of bits to the equivalent decimal digit.

Eg 1000011110010011 = 1000 0111 1001 0011

	↓	↓	↓	↓
decimal	8	7	9	3

Comparison between BCD and Binary

- ***BCD is not a numbering System*** like hex or octal

- A Binary representation of a decimal number converts the whole number into binary.

- BCD converts each digit to binary separately on a BCD digit by digit basis.

Eg $542_{10} = 1000011110_2$ binary

$542_{10} =$ 0101 0100 0010 BCD

NB: The binary requires only 10 bits while the BCD needs 12 bits.

- BCD is used in digital systems such as
  - frequency counters
  - digital clocks
  - electronic calculators

- BCD is *not* used in computers because binary number system is more efficient and arithmetic processes in binary are simpler

Summary : Comparison Table

Hex	Dec	Octal	Binary	BCD for decimal
0	0	0	0	0000
1	1	1	1	0001
2	2	2	10	0010
3	3	3	11	0011
4	4	4	100	0100
5	5	5	101	0101
6	6	6	110	0110
7	7	7	111	0111
8	8	10	1000	1000
9	9	11	1001	1001
A	10	12	1010	0001 0000
B	11	13	1011	0001 0001
C	12	14	1100	0001 0010
D	13	15	1101	0001 0011
E	14	16	1110	0001 0100
F	15	17	1111	0001 0101
10	16	20	10000	0001 0110
11	17	21	10001	0001 0111
12	18	22	10010	0001 1000
13	19	23	10011	0001 1001
14	20	24	10100	0010 0000
15	21	25	10101	0010 0001