TSN1101
Computer Architecture and Organization

Section A (Digital Logic Design)

Lecture 01

[redacted]

Number Systems and Codes

TOPIC COVERAGE IN THE LECTURE (1)…

- Decimal and Binary Number Systems
  - Representation and Counting
  - Conversions – Binary to Decimal, Decimal to Binary
  - Arithmetic in binary - Addition, Subtraction, Multiplication, and Division
- Octal Number System
  - Representation
  - Conversions from decimal to octal, octal to decimal, binary to octal, octal to binary
  - Arithmetic in Octal -Addition, Subtraction
- Hexadecimal Number System
  - Representation
  - Conversions from decimal to Hexadecimal, Hexadecimal to decimal, binary to Hexadecimal, Hexadecimal to binary
  - Arithmetic in Hexadecimal - Addition, Subtraction

TOPIC COVERAGE IN THE LECTURE (2)

- BCD codes
  - 8421,2421,8 4 -2 -1, XS3 codes
    - Representations and Conversions
- Gray Code
  - Binary to Gray Conversion
  - Gray to Binary conversion
- Alphanumeric code
  - ASCII code
- Error Detection and Correction Codes
  - Parity method for error detection

NUMBER SYSTEMS

DECIMAL/BINARY/
OCTAL/HEXADECIMAL

Decimal Number System

The decimal number system has 10 digits 0 through 9

The decimal numbering system has a **base/radix of 10** with each position weighted by a factor of 10

$\ldots.10^5\ 10^4\ 10^3\ 10^2\ 10^1\ 10^0.\ 10^{-1}\ 10^{-2}\ 10^{-3}\ 10^{-4}\ 10^{-5}\ldots$

$14.2 = 1 \times 10^1 + 4 \times 10^0 + 2 \times 10^{-1}$

Representation of Decimal Number System – Examples (1)…

- Express decimal number (656) as a sum of the values of each digit.

Solution:

$$6 \quad 5 \quad 6$$

$$6 \times 100 \quad + \quad 5 \times 10 \quad + \quad 6 \times 1$$

$$\mathbf{6 \times 10^2 \quad + \quad 5 \times 10^1 \quad + \quad 6 \times 10^0}$$

Representation of Decimal Number System – Examples (2)

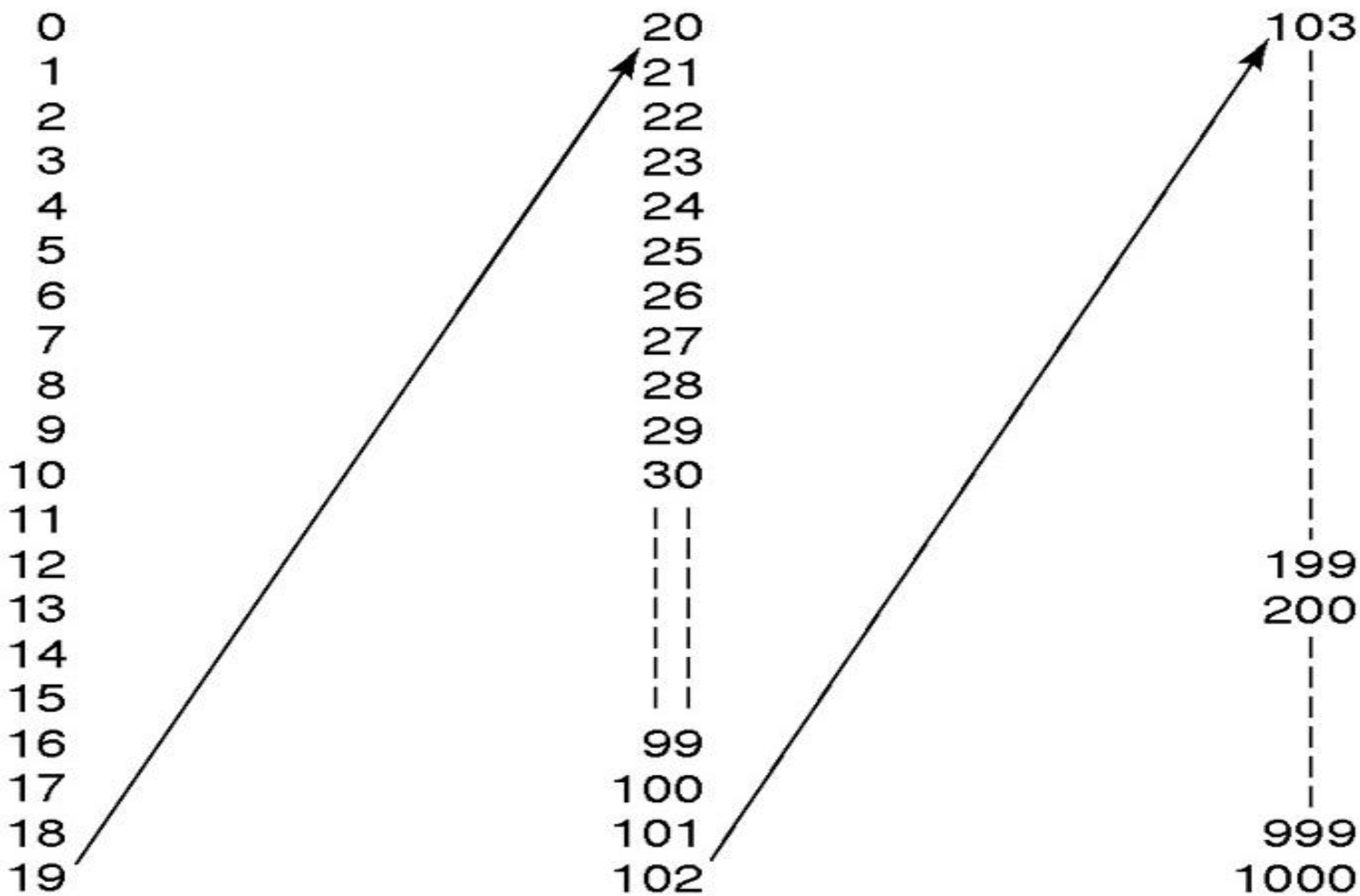
- Express the decimal number (75.68) as a sum of the values of each digit.

7 5 . 6 8

$7 \times 10 + 5 \times 1 + 6 \times 0.1 + 8 \times 0.01$

$$7 \times 10^{1} + 5 \times 10^{0} + 6 \times 10^{-1} + 8 \times 10^{-2}$$

Decimal Counting



Binary Number System

- The binary number system has 2 digits 0 and 1
- The binary number system has a **base/radix of 2** with each position weighted by a factor of 2
- **Weight Structure of a binary number:**

$$2^{n-1} \ldots \ldots\; 2^5\; 2^4\; 2^3\; 2^2\; 2^1\; 2^0\; .\; 2^{-1}\; 2^{-2}\; 2^{-3}\; 2^{-4}\; 2^{-5} \ldots\ldots\; 2^{-n}$$

where n is the number of bits from the binary point

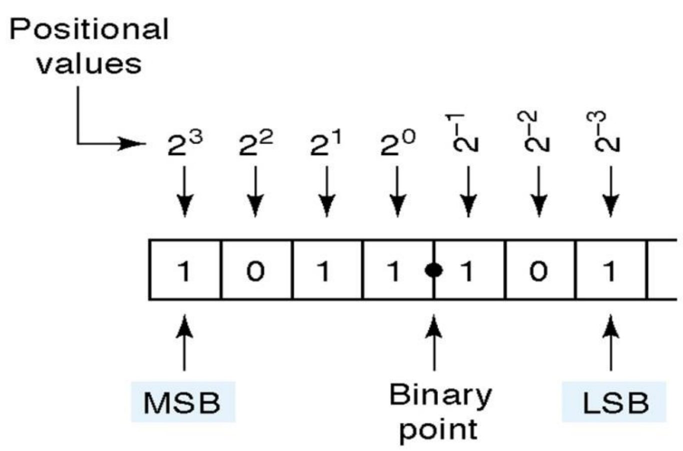
Ex: **10111 =** $1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$

- For the binary whole number,
  - LSB (Least Significant Bit) – right-most bit with a weight of 2^0
  - MSB (Most Significant Bit) – left-most bit with a weight of 2^{n-1}
- For the fractional binary number,
  - LSB (Least Significant Bit) – right-most bit with a weight of 2^{-n}
  - MSB (Most Significant Bit) – left-most bit with a weight of 2^{-1}

Representation of Binary Number System- Example

Largest decimal number that can be represented by a given number of bits = $(2^n)-1$

Ex: If the number of bits = **6**,
largest decimal number represented = $(2^6)-1=63$



Counting in Binary (1)…

DECIMAL NUMBER	BINARY NUMBER			
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

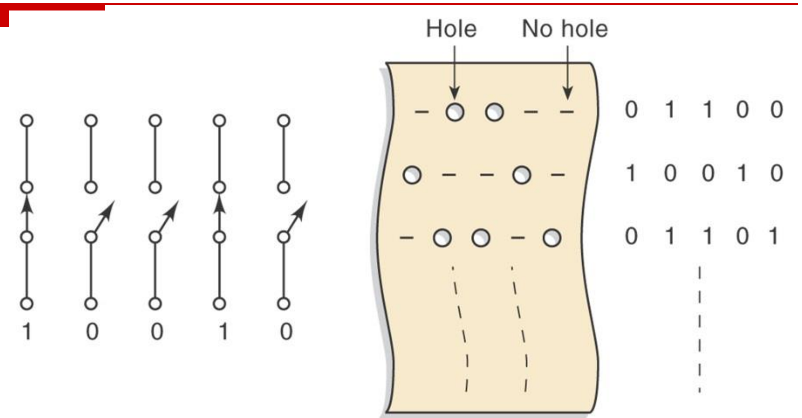
Counting in Binary (2)

- Each time the unit bit changes from a 1 to 0, the two (2 power of 1) position will toggle (change states).
- Each time the twos position changes from 1 to 0, the four (2 power of 2) position will toggle (change states).
  - For e.g.: with two bits we can go through 2 power of 2 =4 counts(00 through 11).
  - With four bits can go through 16 counts (0000 through 1111).
- The last count will always be all 1s and is equal to 2 power of N minus 1 in the decimal system.
  - For eg: 4 bits, the last count is 1111 = (2 power of 4) –1 =15 (in decimal)

Reasons for using Binary number systems in Computer/Digital Systems

- Most of the devices associated with computers or components inside the computers are bistable devices.
- Some examples of two stable devices
  - Open and closed switches
  - Paper Tape
  - Wiring Matrix
    - This forms the foundation for programmable logic devices.
  - Light bulb (off or on)
  - Diode (conducting or not conducting)
  - Relay (energized or not energized)
  - Transistor (cutoff or saturation)
  - Photocell (illuminated or dark)

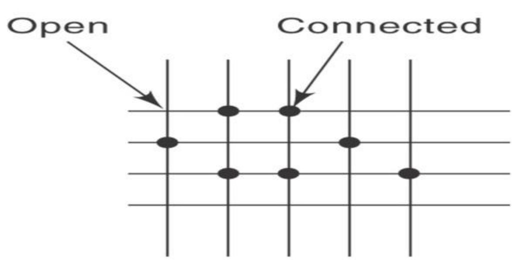
Lecture



(a) (b)

a. Swiches-open-0,closed-1

b. Paper Tape-presence of hole-1, absence-0



(c)

c. Wiring Matrix
-Connected junction-1
-Open junction-0

Binary to Decimal System Conversion -Example (1)

- Any binary number can be converted to decimal by multiplying the weight of each position with the binary digit and adding together

Example: Convert the binary number (11011) into decimal.

Solution

- **Binary number** **1 1 0 1 1**
- **Power of 2 position** 2^4 2^3 2^2 2^1 2^0
- **Decimal value**

$$(2^4 \times 1) + (2^3 \times 1) + (2^2 \times 0) + (2^1 \times 1) + (2^0 \times 1)$$

$$16 + 8 + 0 + 2 + 1$$

$$= 27_{10}$$

Binary to Decimal System Conversion - Example (2)

<u>Example:</u> Convert the binary number (110.11) to its decimal equivalent.

<u>Solution</u>

Binary number	**1**	**1**	**0**	**.**	**1**	**1**
Power of 2 position	2^2	2^1	2^0	**.**	2^{-1}	2^{-2}

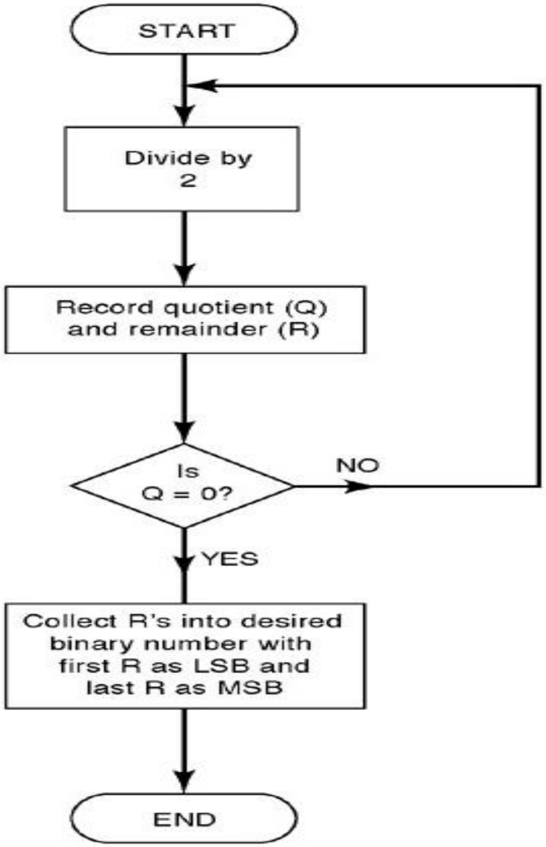
Decimal value

$$(2^2\times1)+(2^1\times1)+(2^0\times0)\ .\ (2^{-1}\times1)+(2^{-2}\times1)$$

$$4 \quad + \quad 2 \quad + \quad 0 \quad . \quad 0.5 \quad + \quad 0.25$$

$$= (6.75)_{10}$$

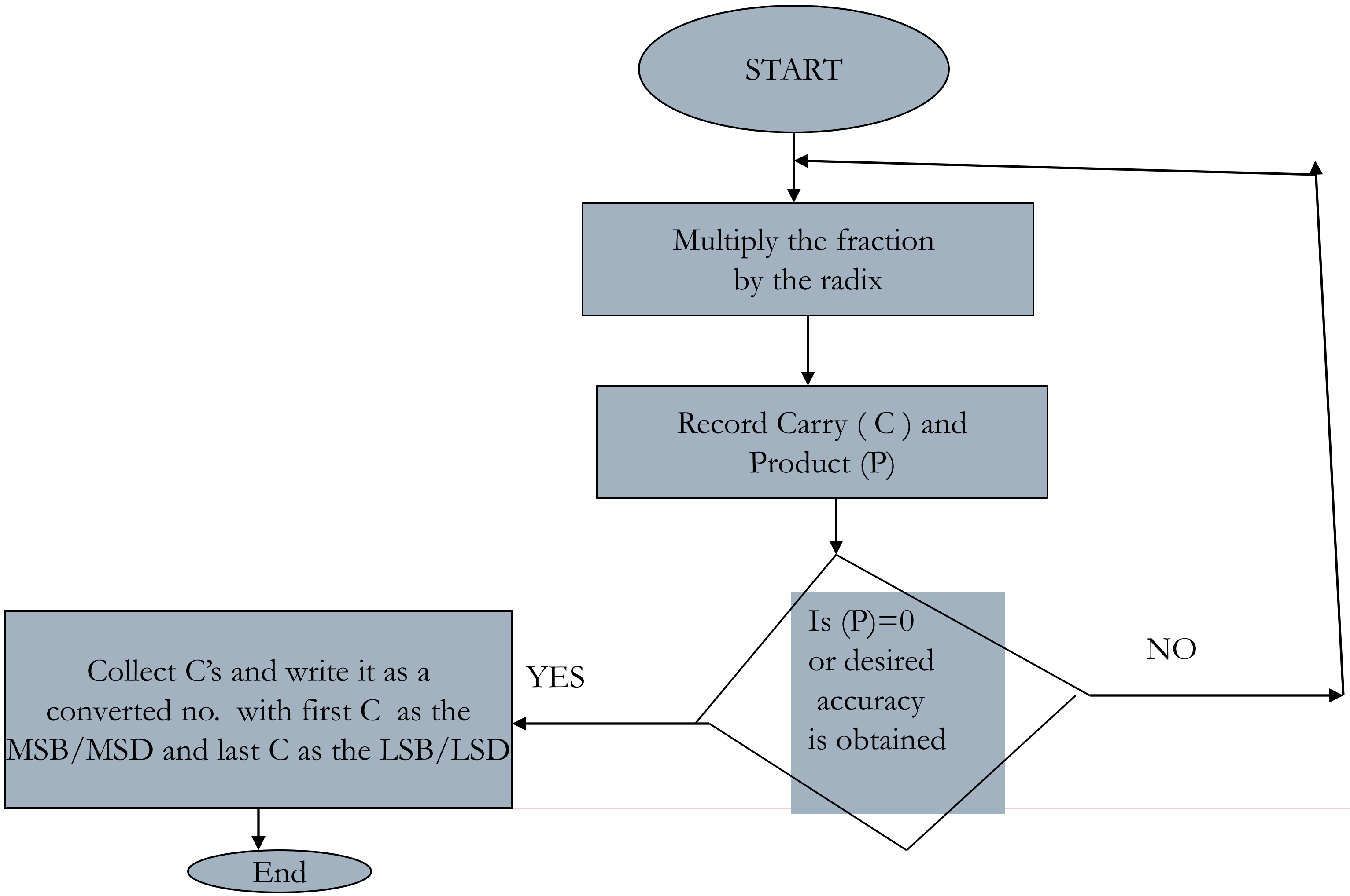
Decimal (Integer) to Any number System - General Flowchart



For Decimal integer to Binary , we have to divide by 2, base/radix of binary number system.

In general, for converting from decimal integer to any number system, we have to divide by the corresponding base/radix.

Decimal (Fraction) to Any number System - General Flowchart



Decimal (Integer) to Binary Conversion – Sum of Weights Method

Example: $(357)_{10}$ = 256 + 64 + 32 + 4 + 1

Binary weights

256	128	64	32	16	8	4	2	1
1	0	1	1	0	0	1	0	1

$(357)_{10}=(101100101)_2$

Example: $(1937)_{10}$ = 1024 + 512 + 256 + 128 + 16 + 1

Binary weights

1024	512	256	128	64	32	16	8	4	2	1
1	1	1	1	0	0	1	0	0	0	1

$(1937)_{10} = (11110010001)_2$

Decimal (Integer) to Binary conversion – Repeated division by 2 method

- Steps:
  - Divide the decimal number by 2
  - Write the remainder after each division until a quotient of zero is obtained.
  - The first remainder is the LSB and the last remainder obtained is the MSB

Decimal (Integer) to Binary Conversion – Repeated division-by-2 method -Examples

Convert the following decimal numbers to binary:

(a) 19 **(b)** 45

Solution

(a)

	Remainder
$\frac{19}{2} = 9$	1
$\frac{9}{2} = 4$	1
$\frac{4}{2} = 2$	0
$\frac{2}{2} = 1$	0
$\frac{1}{2} = 0$	1

1 0 0 1 1

MSB LSB

(b)

	Remainder
$\frac{45}{2} = 22$	1
$\frac{22}{2} = 11$	0
$\frac{11}{2} = 5$	1
$\frac{5}{2} = 2$	1
$\frac{2}{2} = 1$	0
$\frac{1}{2} = 0$	1

1 0 1 1 0 1

MSB LSB

Related Problem Convert decimal number 39 to binary.

Decimal (Fraction) to Binary Conversion – Sum of Weights Method

Fraction:

0.6875 = .5 + .125 + .0625

Binary weights

.5	.25	.125	.0625
1	0	1	1

$(0.6875)_{10}$ =(0.1011)2

Mixed:

95.6875 = 64 + 16 + 8 + 4 + 2 + 1 + .5 + .125 + .0625

Binary weights

64	32	16	8	4	2	1.	.5	.25	.125	.0625
1	0	1	1	1	1	1 .	1	0	1	1

$(95.6875)_{10}$ =$(1011111.1011)_2$

Decimal (Fraction) to Binary Conversion – Repeated Multiplication by 2 method

- When converting a decimal fractional number to its binary, the decimal fractional part will be multiplied by 2 till the fractional part gets 0 or till the desired accuracy is reached.

Example

Convert Decimal $(0.825)_{10}$ to its binary equivalent.

Solution Steps:

Step 1: 0.825 will be multiplied by 2 ($0.825 \times 2 = 1.650$)
Step 2: The integer part will be the MSB in the binary result
Step 3: The fractional part of the earlier result will be multiplied again. ($0.650 \times 2 = 1.300$)
Step 4: Each time after the multiplication, the integer part of the result will be written as the binary number.
Step 5: The procedure should continue till the fractional part gets 0 or until the desired accuracy is reached.

Decimal (Fraction) to Binary Conversion – Repeated Multiplication by 2 method - An Example

```
          0.825
carry         2
     ----------
   1.       650
              2
     ------------
    1.      300
              2
    ------------
    0.    600
            2
   -------------
    1.    200
            2
   -------------
   0.    400
```

Equivalent binary number for the decimal fraction
$(0.825)_{10} = (0.11010)_2$ (for 5 bit accuracy)

Binary Addition

Rules:

Augend	Addend	Sum	Carryout
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Binary Addition – Examples

Example 1:

Add the binary numbers 00111 and 10101 and show the equivalent decimal addition.

Solution:

$$
\begin{array}{rcr}
{\scriptstyle 0\,1\,1\,1} & & \\
00111 & & 7 \\
10101 & & 21 \\
\hline
11100 & = & 28
\end{array}
$$

Example 2: Perform the binary additions: 1101 0110+ 0111 1011

Solution:

```
   1101 0110
    0111 1011
-----------------
 1  0101 0001
-----------------
```

Binary Subtraction

Rules:

Minuend	Subtrahend	Difference	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Binary Subtraction - Examples

Example 1: Subtract the binary number 00111 from 10101 and show the equivalent decimal subtraction.

Solution:

$$\begin{array}{r@{\quad}r} 10101 & 21 \\ \underline{00111} & \underline{7} \\ 01110 = & 14 \end{array}$$

Example 2: Perform binary subtraction: 1101 – 101

Solution:

```
  1100
  0101
------------
  0111
------------
```

Binary Multiplication

Multiplicand = Top Number, Multiplier = Bottom Number, Product = Result

Rules:

0 X 0 = 0

0 X 1 = 0

1 X 0 = 0

1 X 1 = 1

Example:

```
     100110
X       101
-----------
     100110
    000000
   100110
-----------
   10111110
```

Binary Division

Dividend = Top Number, Divisor = Bottom Number,

Quotient and Remainder = Result

Perform the following binary divisions:

(a) 110 ÷ 11 **(b)** 110 ÷ 10

Solution

$$\text{(a)}\quad \begin{array}{r} \mathbf{10} \\ 11\overline{)110} \\ \underline{11} \\ 000 \end{array} \qquad \begin{array}{r} 2 \\ 3\overline{)6} \\ \underline{6} \\ 0 \end{array} \qquad\qquad \text{(b)}\quad \begin{array}{r} \mathbf{11} \\ 10\overline{)110} \\ \underline{10} \\ 10 \\ \underline{10} \\ 00 \end{array} \qquad \begin{array}{r} 3 \\ 2\overline{)6} \\ \underline{6} \\ 0 \end{array}$$

Related Problem Divide 1100 by 100.

Octal Number System

- The Octal numbering system uses the symbols

  0, 1, 2, 3, 4, 5, 6, and 7
- 8 different digits – **Radix/ Base is 8**
- It is a weighted system as follows

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

where n is the number of integer octal digits and
s is the number of fractional octal digits.

Octal to Decimal Conversion -Example

- **Procedure:**

  Multiply the weight of each position with the octal number and add together.

- **Example**

  Convert the octal number $(23754)_8$ into its decimal equivalent.

  Solution:

$$(23754)_8 = 2 \times 8^4 + 3 \times 8^3 + 7 \times 8^2 + 5 \times 8^1 + 4 \times 8^0$$

$$= (10220)_{10}$$

Decimal to Octal Conversion
- Repeated Division Method

- **Procedure**: Divide the decimal number repeatedly by 8 until the remainder becomes zero
  The first remainder is the LSD and the last is the MSD.

Example: Convert $(12345)_{10}$ to its equivalent octal.

12345/8 = 1543 remainder 1 (LSD)

1543 /8 = 192 remainder 7

192 /8 = 24 remainder 0

24 / 8 = 3 remainder 0

3 /8 = 0 remainder 3 (MSD)

$$\mathbf{(12345)_{10} = (30071)_8}$$

Octal to Binary conversion

Convert each octal digit to its equivalent 3 digit binary number

Octal Digit	0	1	2	3	4	5	6	7
Binary Equivalent	000	001	010	011	100	101	110	111

Example: Convert $(345621)_8$ to its binary equivalent

3 4 5 6 2 1

011 100 101 110 010 001

Result:

$(345621)_8 = (011100101110010001)_2$

Binary to Octal Conversion

Convert from binary to octal by grouping bits in threes starting with the LSB.
Each group is then converted to the octal equivalent
Leading zeros can be added to the left of the MSB to fill out the last group.

Octal Digit	0	1	2	3	4	5	6	7
Binary Equivalent	000	001	010	011	100	101	110	111

Example: Convert $(110001100101001)_2$ to Octal

110	001	100	101	001
6	1	4	5	1

Result:

$(110001100101001)_2 = (61451)_8$

Octal Addition

- The largest single digit in octal is 7.
- If the result of A + B is greater than 7, then we must subtract 8 (the base) and carry 1 to the next digit.

- **Example:** Perform the following addition in Octal.

$$(7456)_8 + (537)_8$$

$$7456_8$$

$$0537_8$$

$$\mathbf{(10215)_8}$$

Octal Subtraction

- If Subtrahend is bigger then the Minuend we borrow a 1 from the digit on left and its weight will be 8.

Example: Perform $(3533)_8 - (175)_8$

Solution:

$$
\begin{array}{r}
3533_8 \\
0175_8 \\
\text{------------} \\
\mathbf{(3336)_8} \\
\text{-------------}
\end{array}
$$

Hexadecimal numbering system

- The hexadecimal numbering system uses the symbols
  0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F
- 16 different symbols – **Radix/ Base of 16**
- It is a weighted system as follows

$$16^{n-1}\ 16^{n-2} \ldots\ldots 16^{3}\ 16^{2}\ 16^{1}\ 16^{0}.\ 16^{-1}\ 16^{-2}\ 16^{-3} \ldots 16^{-s}$$

where n is the number of integer hexadecimal digits and s is the number of fraction hexadecimal digits.

Hexadecimal Number and Binary, Decimal equivalents

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

Hexadecimal to Decimal Conversion

- Hexadecimal number can be converted to decimal by multiplying the weight of each position of the hexadecimal number (power of 16) and adding together.

- **Example:** Convert $(23ABC)_{16}$ into its decimal equivalent.

  $(23ABC)_{16} = 2 \times 16^4 + 3 \times 16^3 + 10 \times 16^2 + 11 \times 16^1 + 12 \times 16^0$

  $= \mathbf{(146108)_{10}}$

Decimal to Hexadecimal Conversion

Procedure:

• Repeated division of a decimal number by 16 will produce the equivalent hexadecimal number, formed by the remainder of the divisions.

• The first remainder produced is the LSD. Each successive division by 16 yields a remainder that becomes a digit in the equivalent hexadecimal number.

•**Example:**

Perform the conversion of a decimal number **234567** to hexadecimal.

234567 /16 = 14660 remainder 7 (LSD)

14660 /16 = 916 remainder 4

916 /16 = 57 remainder 4

57/16 = 3 remainder 9

3/16 = 0 remainder 3 (MSD)

$$(\mathbf{234567})_{10} = (\mathbf{39447})_{16}$$

Hexadecimal to Binary Conversion

To convert Hex. to binary change each Hex digit to the equivalent four bit binary number

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

Example: Convert $A12BF_{16}$ into its equivalent binary

A 1 2 B F

1010 0001 0010 1011 1111

$(A12BF)_{16} = (10100001001010111111)_2$

Binary to Hexadecimal conversion

- **Procedure:**
  - Group bits in four starting with the LSB.
  - Each group is then converted to the hex equivalent
  - Leading zeros can be added to the left of the MSB to fill out the last group
- **Example:** Convert $(1110100110)_2$ into its equivalent hexadecimal number

$(1110100110)_2$ = **00**11 1010 0110 (Note the addition of leading zeroes)
= 3 A 6
= $(3A6)_{16}$

Hexadecimal Addition (1)…

- **Procedure:**
  - In any given column of an addition problem, think of the two hex. digits in terms of their decimal values.
  - If the sum of these two digits is $(15)_{10}$ or less, the corresponding hexadecimal digit is written.
  - If the sum is greater than $(15)_{10}$, subtract 16 (base) from the sum and carry a 1 to the next column.

Hexadecimal Addition (2)

- **Example:**

  Add: $(AB6)_{16} + (C5)_{16}$

Solution:

```
 A B  6
 0 C  5
--------
(B 7 B)H
---------
```

Hexadecimal Subtraction

- If Subtrahend is bigger than the Minuend, we borrow 1 from the digit on left and its weight will be 16.
- **Example:**

$$(4AB.C5)_H - (3C.D)_H$$

```
 4AB.C5
 03C.D0
-----------
```

$(46E.F5)_H$

```
-----------
```

Usefulness of Octal and Hexadecimal Number System

- Used in digital system as a “shorthand” way to represent strings of bits.
- When dealing with a large number of bits, it is more convenient and less error –prone to write the binary numbers in hex or octal.
- For eg: to print out the contents of 50 memory locations, each of which was a 16-bit number like 1111 1111 1111 1111, it is easier to use hexadecimal notation like FFFF

Summary of Number System Conversions (1)…

- When converting from binary/octal/hex to decimal, use the method of taking the weighted sum of each digit position.

- When converting from decimal integer to binary/octal or hex , use the method of repeatedly dividing by 2 /8/16 and collecting remainders.
- When converting from decimal fraction to binary/octal or hex , use the method of repeatedly multiplying by 2 /8/16 and collecting carries.

Summary of Number System Conversions (2)

- When converting from binary to octal/hex, group the bits in groups of three/four and convert each group into correct octal /hex digit.
- When converting from octal /hex to binary, convert each digit into its three bits/four bits equivalent.

- When converting from octal to hex. or vice versa, first convert to binary, then convert the binary into the desired number system

----

CODES

BCD/ALPHANUMERIC/GRAY/
ERROR DETECTION/CORRECTION

CODES

- When numbers, letters or words are represented by a special group of symbols, we say that they are being encoded and the group of symbols is called a CODE

- Binary coded decimal (BCD) means that each decimal digit, 0 through 9, is represented by a binary code of four bits.

Standard Binary Coded Decimal or Natural BCD (8421 Code)

- Most commonly used code for decimal digits to be coded in binary

- Even though BCD is with 4 bits, the last 6 combinations out of the 16 possible combinations are not used

- called as Natural BCD (NBCD) code or 8421 code.

- Weighted code
  - 8421 means the weight of the each digit
  - The MSB weight is 8 and the LSB weight is 1.

Difference between NBCD and binary number

- NBCD is a decimal system with each digit encoded in its binary equivalent
- NBCD number is not the same as a straight binary number. For eg:

  137(decimal)= 10001001(binary)

  137(decimal)=0001 0011 0111(NBCD)
- **Advantage of NBCD code:** Ease of conversion to and from decimal. Since there are only ten code groups in the NBCD system, it is very easy to convert between decimal number and NBCD.
- **Disadvantage of NBCD code:** For representing a given decimal, NBCD requires more bits than binary.

Decimal to NBCD conversion (1)…

In NBCD, a 4 bit binary number is used to represent one digit of a decimal number.

Decimal	**NBCD**
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	0001 0000

Decimal to NBCD conversion (2)

- **Procedure:** Replace each decimal digit with the appropriate 8421 code
- **Example:**

  Convert the decimal number 456.75 to NBCD

  Solution:

4	5	6	.	7	5
0100	0101	0110	.	0111	0101

NBCD to Decimal conversion

- **Procedure:**
  - Divide the NBCD number into a group of 4 bits, starting from LSB. Then each group of 4 bits is converted to an appropriate decimal digit.

Example:

Convert the NBCD (0001010101001001.1000) to its Decimal.

Solution:

$$(0001\quad 0101\quad 0100\quad 1001\quad .\quad 1000)_{NBCD}$$

$$(1\qquad 5\qquad 4\qquad 9\quad .\quad 8)_{10}$$

Binary to NBCD conversion

- **Procedure:**
  - Convert the binary number to its equivalent decimal
  - Write each decimal number in 4–bit NBCD code.
- **Example:**

Convert $(101011)_2$ to its equivalent NBCD number

Solution:

$$(101011)_2 = 1\times2^0+1\times2^1+1\times2^3+1\times2^5$$
$$= 1 + 2 + 8 + 32$$
$$= (43)_{10}$$

$$(43)_{10} = (0100\ 0011)_{NBCD}$$

NBCD to binary conversion

- **Procedure:**
  - Convert NBCD to its equivalent decimal number
  - Convert the decimal number to its equivalent binary digit.
- **Example:**

  Convert $(1001\ 0101\ 0011.1000\ 0010)_{NBCD}$ to binary

  Solution:

  $(1001\ 0101\ 0011.1000\ 0010)_{NBCD} = (953.82)_{10}$

  $(953.82)_{10} = (1110111001.11010001)_2$

Excess-3 Code (1)…

- Excess–3(XS3) code is another BCD type of code

- Difference between XS3 and NBCD(8421 code):
  - XS3 is 3 values more then the NBCD number.

- Since XS3 does not give specific value for each bit, it is categorized as a **non-weighted binary code.**

Excess-3 Code (2)

Decimal	BCD Code	XS3 Code
	8 4 2 1	8 4 2 1
0	0 0 0 0	0 0 1 1
1	0 0 0 1	0 1 0 0
2	0 0 1 0	0 1 0 1
3	0 0 1 1	0 1 1 0
4	0 1 0 0	0 1 1 1
5	0 1 0 1	1 0 0 0
6	0 1 1 0	1 0 0 1
7	0 1 1 1	1 0 1 0
8	1 0 0 0	1 0 1 1
9	1 0 0 1	1 1 0 0

Decimal to XS3 code conversion

- **Procedure:**
  - Add 3 to each decimal digit.
  - Write the equivalent 8421 code

Example:

Convert the Decimal $(4656.13)_{10}$ to its equivalent XS3 code

Solution:

```
 4656.13
 3333.33
------------
 7989.46
```

$(4656.13)_{10} =$ (0111 1001 1000 1001 . 0100 0110) $_{\text{XS3 code}}$

XS3 code to Decimal conversion

- **Procedure:**
  - Divide the XS3 code into group of four, starting from LSB.
  - Subtract each XS3 code group by 0011 to get the equivalent NBCD number.
  - Write the equivalent decimal digit for each 4 bit group

Example:

Convert the XS3 code 10111000 to its decimal number.

Solution:

```
    1011   1000
    0011   0011             ( - )
-----------------------------
   1000    0101
----------------------------
```

$\mathbf{(1000\ 0101)_{8421} = (85)_{10}}$

Other BCD Codes (1)...

Decimal Digit	2421 code	8 4 -2 -1 code
0	0000	0000
1	0001	0111
2	0010	0110
3	0011	0101
4	0100	0100
5	1011	1011
6	1100	1010
7	1101	1001
8	1110	1000
9	1111	1111

(2421 and 84 -2 -1 codes are weighted codes)

Other BCD Codes (2)

- **Self complementing Codes:**
  - 9's complement of a decimal number is obtained directly by changing 1's to 0's and 0's to 1's in the code.
  - Examples: 2421 code, XS3 code

- **Reflecting Code:** If a mirror is placed in between the code group, the second half is the mirror reflection of the first half.
  - Examples: 2421 code, 8 4 -2 -1 code

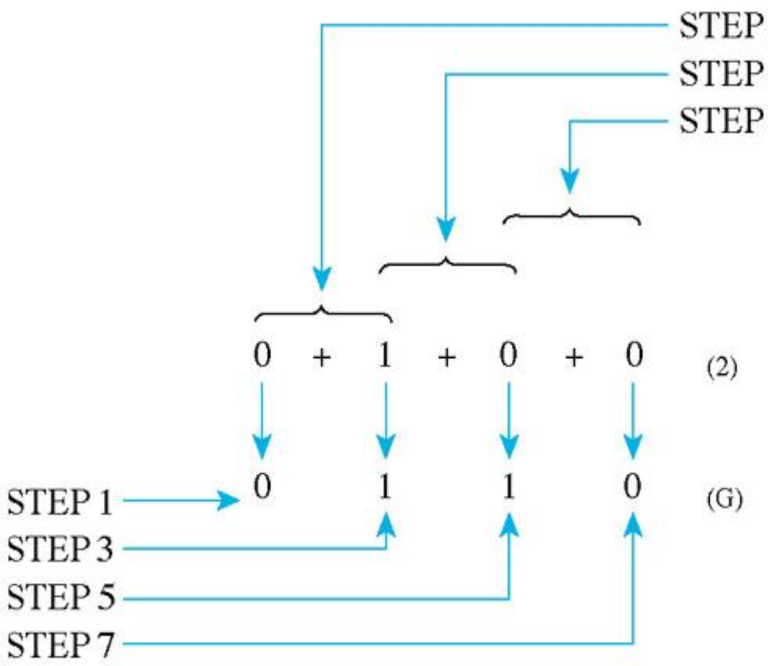
Gray Code

- Gray code is an unweighted and non-arithmetic code
  - No specific weights assigned to bit positions.
- Advantage of Gray code over straight binary sequence
  - Only one bit in the code group changes when going from one number to the next.
    - For example, in going from 7 to 8, The gray code changes from 0100 to 1100. Only the first bit changes from 0 to 1, the other three bits remain the same.
    - But in binary, the change from 7 to 8 will be from 0111 to 1000, which causes all four bits to change values.

Binary to Gray code conversion (1)…

- **Procedure**
  - Write the MSB of the Gray code same as the corresponding MSB in the binary number
  - Going from left to right, add each adjacent pair of binary code bits to get the next Gray code bit. Discard the carries if any.
  - Continue the process till the LSB is reached

Binary to Gray code conversion (2)

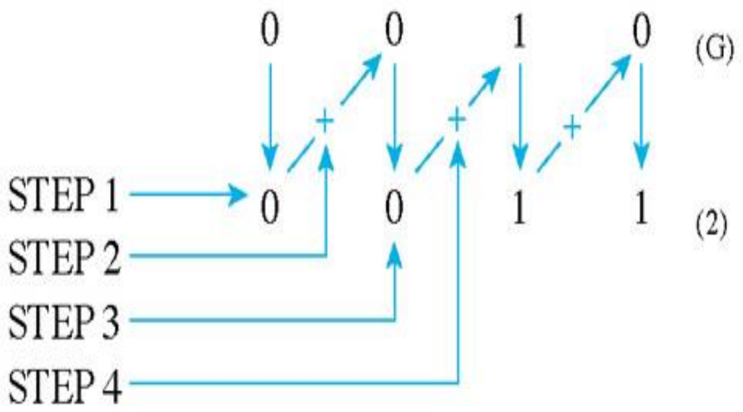


STEP 1: BRING DOWN MSB.
STEP 2: ADD MSB TO ADJACENT BIT.
STEP 3: PLACE SUM BELOW ADJACENT BIT. NOTE: IF A CARRY IS GENERATED, DISREGARD IT.
STEP 4: ADD SECOND PAIR OF BITS.
STEP 5: PLACE SUM BELOW LAST BIT ADDED.
STEP 6: ADD THIRD PAIR OF BITS.
STEP 7: PLACE SUM BELOW LAST BIT ADDED.

Gray to Binary Conversion (1)…

- **Procedure:**
  - Write the MSB of the Binary code, same as MSB of the given Gray code
  - Add the MSB in the binary result with the immediate right most bit in the Gray code and write the result as second binary digit.
  - Continue the above process till the LSB is reached. Discard carries, if any.

Gray to Binary Conversion (2)



STEP 1: BRING DOWN MSB.

STEP 2: ADD MSB DIAGONALLY TO NEXT LESSER SIGNIFICANT BIT.

STEP 3: PLACE SUM BELOW BIT ADDED TO. NOTE: IF A CARRY IS GENERATED, DISCARD IT.

STEP 4: ADD SUM DIAGONALLY TO NEXT LESSER SIGNIFICANT BIT. REPEAT STEPS 3 AND 4 UNTIL ALL GRAY CODE BITS HAVE BEEN ADDED.

Gray Code for Decimal Numbers

DECIMAL	BINARY	GRAY CODE	DECIMAL	BINARY	GRAY CODE
0	0000	0000	8	1000	1100
1	0001	0001	9	1001	1101
2	0010	0011	10	1010	1111
3	0011	0010	11	1011	1110
4	0100	0110	12	1100	1010
5	0101	0111	13	1101	1011
6	0110	0101	14	1110	1001
7	0111	0100	15	1111	1000

Application – Gray code

- Used in applications where the normal sequence of binary numbers may produce an error or ambiguity during the transition from one number to the next.

- Ex: for encoding shaft position data from machines such as computer-controlled lathes.

Alphanumeric Codes

- A computer must be able to handle non-numerical information in addition to the numerical data.
- A computer should recognize codes that represent letters of the alphabet, punctuation marks, and other special characters as well as numbers.
- Codes that represent numbers, alphabetic characters, symbols (printable characters), and nonprintable characters are called **alphanumeric codes**.
- A complete alphanumeric code would include the 26 lowercase letters , 26 uppercase letters, 10 numerical digits, 7 punctuation marks and anywhere from 20 to 40 other characters such as +,/,#,$, and so on.

ASCII Code (1)…

- ASCII – American Standard Code for Information Interchange
- Universally accepted alphanumeric code in most computers, electronic equipment, keyboards
- Uses 7 bits to code 128 characters
- 94 printable characters and 34 nonprintable characters

ASCII Code (2)…

- Printable Characters (94):
  - 26 Uppercase letters (A to Z)
  - 26 Lowercase letters (a to z)
  - 10 numerals (0 to 9)
  - 32 special printable characters (like %, *, $)

ASCII Code (3)…

- Nonprintable Characters or Control Characters (34)
  - Three types
    - **Format Effectors**
      - Characters that control the layout of printing
      - (Examples) :Backspace (BS), horizontal tabulation (HT), and carriage return (CR))
    - **Information Separators**
      - Used to separate the data into divisions such as paragraphs and pages
      - (Examples): record separator (RS), file separator (FS))
    - **Communication-control characters**
      - Used during the transmission of text between remote terminals
      - (Examples): start of text (STX), end of text (ETX)

ASCII Code (4)

Table 1–5 American Standard Code for Information Interchange

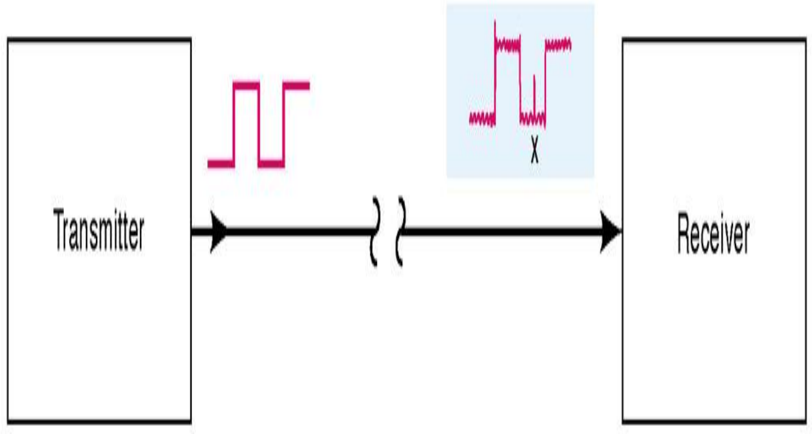
LSB \ MSB	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P		p
0001	SOH	DC_1	!	1	A	Q	a	q
0010	STX	DC_2	”	2	B	R	b	r
0011	ETX	DC_3	#	3	C	S	c	s
0100	EOT	DC_4	$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	’	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	\	l	|
1101	CR	GS	-	5	M	]	m	}
1110	SO	RS	.	>	N	↑	n	~
1111	SI	US	/	?	O	—	o	DEL

Definitions of control abbreviations:

ACK	Acknowledge	FF	Form feed
BEL	Bell	FS	Form separator
BS	Backspace	GS	Group separator
CAN	Cancel	HT	Horizontal tab
CR	Carriage return	LF	Line feed
DC_1–DC_4	Direct control	NAK	Negative acknowledge
DEL	Delete idle	NUL	Null
DLE	Data link escape	RS	Record separator
EM	End of medium	SI	Shift in
ENQ	Enquiry	SO	Shift out
EOT	End of transmission	SOH	Start of heading
ESC	Escape	SP	Space
ETB	End of transmission block	STX	Start text
ETX	End text	SUB	Substitute
		SYN	Synchronous idle
		US	Unit separator
		VT	Vertical tab

Need for Error Detection

- Whenever information is transmitted from one device (the transmitter) to other device(the receiver) there is possibility that errors can occur such that receiver does not receive the identical information that was sent by the transmitter.
- Major cause of any transmission errors is electrical noise, which consists of spurious fluctuations in voltage or current.
- For this reason, many digital systems employ some method for detection (and sometimes correction) of errors.



Parity Method for Error Detection (1)…

- Simplest and most widely used scheme for error detection is the parity method
- A parity bit is an extra bit that is attached to a code group that is being transferred from one location to another.
- The parity bit is made either 0 or 1, depending on the number of 1s that are contained in the code group

Parity Method for Error Detection (2)…

- **Two methods**
  - Even-parity method
  - Odd-parity method
- **Even Parity Method:**
  - Here the value of the parity bit is chosen so that the total number of 1s in the code group (including the parity bit) is an even number.
  - Ex: In the code group 1000011 (ASCII character "C"), the group has three 1's. Therefore we will add a parity bit of 1 to make the total number of 1s an even number. The new code group, including the parity bit becomes 1 1000011

Parity Method for Error Detection (3)…

- **Odd Parity Method:**
  - Here the parity bit is chosen so the total number of 1s(including the parity bit) is an odd number.
  - Example: for the code group 1000001, the assigned parity bit would be a 1. For the code group 1000011, the parity bit would be a 0.
- **Disadvantage of parity method:**
  - Can detect only singe error
  - This parity method would not work if 2 bits were in error, because two errors would not change the "oddness" or "evenness" of the number of 1s in the code.

Parity Method for error Detection (4)

EVEN PARITY		ODD PARITY	
P	BCD	*P*	BCD
0	0000	1	0000
1	0001	0	0001
1	0010	0	0010
0	0011	1	0011
1	0100	0	0100
0	0101	1	0101
0	0110	1	0110
1	0111	0	0111
1	1000	0	1000
0	1001	1	1001

Problem (1)

Problem:

Perform the following conversions:

$$(101101.101)_2 = (?)_{10} = (?)_H = (?)_O$$

Solution:

$$\begin{array}{ccccccccc} 32 & & 8 & 4 & & 1 & 0.5 & & 0.125 \\ (1 & 0 & 1 & 1 & 0 & 1. & 1 & 0 & 1)_2 \end{array} = (45.625)_{10}$$

$$\begin{array}{ccc} 8421 & 8421 & 8421 \\ (0010 & 1101. & 1010)_2 \end{array} = (2D.A)_H$$

$$\begin{array}{ccc} 421 & 421 & 421 \\ (101 & 101. & 101)_2 \end{array} = (55.5)_O$$

Problem (2)

Problem:
Suppose your microcomputer memory address is 16 bits wide,
How many locations that it can have, assuming each memory location is 8 bits wide.
Express the memory capacity in bytes, megabytes, gigabytes and terabytes.

Solution:
Number of memory locations = $2^{\text{memory address bits}}$
= 2^{16} = 65536 = 64 K

Memory capacity = Number of memory locations * capacity of each memory location
= 65536 * 8 bits = 524288 bits
= 65536 bytes
= 0.0625 Mbytes
= 0.00006103515625 Giga Bytes = $6.1035 * 10^{-5}$ GB
= 0.000000059604644775390625 Terabytes = $5.9605 * 10^{-8}$ TB

Problem (3)

Problem:

Perform the following conversions:

$(639.75)_{10} = (?)_{NBCD} = (?)_{2421} = (?)_{5211} = (?)_{84\text{-}2\text{-}1} = (?)_{\text{Gray code}}$

Solution:

$(639.75)_{10} = (0110\ 0011\ 1001.0111\ 0101)_{NBCD}$

$(639.75)_{10} = (1100\ 0011\ 1111.1101\ 1011)_{2421}$

$(639.75)_{10} = (1010\ 0101\ 1111.1100\ 1000\ )_{5211}$

$(639.75)_{10} = (1010\ 0101\ 1111\ .\ 1001\ 1011)_{84\text{-}2\text{-}1}$

$(639.75)_{10} = (1001111111.11)_2 = (1101000000.00)_{\text{Graycode}}$

Reference

- Slides adopted from the book

Thomas L.Floyd, "Digital Fundamentals," 11th Edition, Prentice Hall, 2015
(ISBN13:9781292075983)