Unit 2 Binary System

Short Introduction

A computer understands the language of 1s and 0s only, called machine language. The number system that only contains 1s and 0s is called binary number system. The usage of a computer includes internet surfing, playing games, watching movies, making documents etc. How are all such activities converted to 1s and 0s? This is discussed in this unit with respect to binary conversion and storage of binary values in computer memory.

Students' Learning Outcomes

1. **Introduction of number systems**
   - Describe following number systems with Examples
     - Binary
     - Decimal
     - Hexadecimal
2. **Number System Conversion**
   - Convert binary to decimal and decimal to binary number system
   - Convert decimal to hexadecimal and hexadecimal to decimal number system
   - Convert binary to hexadecimal and hexadecimal to binary number system
3. **Memory and data storage**
   - What is memory?
   - Understand how data is represented in a computer memory (with reference of bits and bytes)
   - Storage device
   - Difference between memory and storage devices
4. **Measurement of size of computer memory.**
   - Define following Terms
     - Bit
     - Byte
     - Kilobyte
     - Megabyte
     - Gigabyte
     - Terabyte
     - Petabyte
5. **Boolean algebra**

   Explain:
   - A Boolean proposition
   - Truth values
   - Logical operators (AND, OR, NOT)
   - Truth tables
   - Laws of Boolean algebra
     - Commutative
     - Associative
     - Distributive
     - Identity
   - Logical expressions

2.1 Introduction to Number Systems

A number system is the system for representation of numeric data. We all are familiar with decimal number system where each number consists of digits from 0 to 9. In a computer system, other number systems are also used. We discuss few number systems in the following sections.

2.1.1 Decimal

The number system we use in our daily life is the decimal number system. The decimal number system has base 10 as it uses ten digits [0-9]. Each position represents a specific power of base 10 as shown in Figure 2-1.

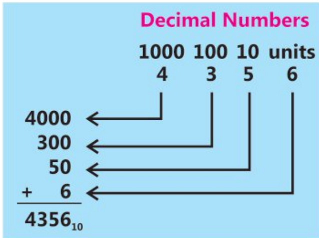

Figure 2-1

Examples:

- $892 = 8 \times 10^2 + 9 \times 10^1 + 2 \times 10^0$
- $1247 = 1 \times 10^3 + 2 \times 10^2 + 4 \times 10^1 + 7 \times 10^0$
- $53 = 5 \times 10^1 + 3 \times 10^0$

Do you know?

Decimal number system is also called Hindu-Arabic, or Arabic, number system, in mathematics.

2.1.2 Binary

Binary number system has base 2 as all the numbers in this system consist of only two digits i.e. 0 and 1. Digital computers use this system to store data. Your name is in the form of alphabets, but for a computer each alphabet has some binary value.

Example: The binary value of the letter 'A' is 01000001 and its decimal value is 65.

2.1.3 Hexadecimal

Hexadecimal system has total 16 numbers, i.e., 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, where A=10, B=11, C=12, D=13, E=14 and F=15 (as shown in Figure 2-2).

DECIMAL NUMBER SYSTEM
Uses ten different symbols
0, 1, 2, 3, 4, 5, 6, 7, 8, 9

HEXADECIMAL NUMBER SYSTEM
Uses sixteen different symbols
0, 1, 2, 3, 4, 5, 6, 7, 8, 9
A, B, C, D, E, F

Figure 2-2

Example: 3F2B

2.2. Number System Conversion

We can convert a number from one number system to another and vice versa. In the following, we discuss conversions among different number systems.

2.2.1 Decimal to Binary and Binary to Decimal

- **Decimal to Binary**

  To convert a decimal number to binary, we divide the number by 2 and take quotient and remainder. We continue dividing the quotient by 2 until we get quotient 0. We write out all the remainders in reverse order to obtain the value in binary.

Example:

Convert 156_{10} (156 in decimal) to binary

Table 2-1 shows the method to solve this problem. Remainders are taken from bottom to top to present the binary number. So, $156_{10} = 10011100_2$.

2	156	
2	78	– 0
2	39	-- 0
2	19	-- 1
2	9	-- 1
2	4	-- 1
2	2	-- 0
2	1	-- 0
2	0	-- 1

Table 2-1

Activity 2.1

How many marks did you obtain in the final examination of 8th class? Convert that figure to binary and discuss the result with your class fellows.

- **Binary to Decimal**

  The conversion of a number from binary number system to decimal number system is explained below with the help of an example.

Example: **Convert $(1000001)_2$ to decimal**

$$= 1 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$
$$= 64 + 0 + 0 + 0 + 0 + 0 + 1$$
$$= (65)_{10}$$

The above conversion is done by the following steps.

Step 1. Write down the binary number which is $(1000001)_2$ in this example.

Step 2. List the powers of two from right to left starting with 0. In this example, the power of 2 starts from 0 and ends at 6.

Step 3. Multiply 2's corresponding powers to each binary value. In the above example there are 7 binary values.

Step 4. Compute each value.

Step 5. Add all the values.

Step 6. Write the answer along with its base subscript.

Activity 2.2

Exchange your marks in binary form with your friends and convert them in decimal to know about their expectations in the board examination of 9th class. Double check with your class fellows that how much your calculations are accurate.

Activity 2.3

According to Table 2-2, write in decimal, binary, and hexadecimal the time of your:

- arrival at school
- lunch
- playing

Decimal	Binary	Hexadecimal	Decimal	Binary	Hexadecimal
0	0	0			
1	1	1	11	1011	B
2	10	2	12	1100	C
3	11	3	13	1101	D
4	100	4	14	1110	E
5	101	5	15	1111	F
6	110	6	16	10000	10
7	111	7	17	10001	11
8	1000	8	18	10010	12
9	1001	9	19	10011	13
10	1010	A	20	10100	14

Table 2-2

Activity 2.4

Many online convertors for number systems are available. Try to find and use them. You can ask your class teacher to help in searching.

2.2.2 Decimal to Hexadecimal and Hexadecimal to Decimal

- **Decimal to Hexadecimal**

As we have studied that hexadecimal number system has base 16, so for conversion of a number from decimal to hexadecimal, we divide the number by 16 and take both quotient and remainder. We continue dividing the quotient by 16 until the quotient becomes 0.

Example:

Convert $(69610)_{10}$ to Hexadecimal

Table 2-3 shows the method to solve this problem. We can observe from the table that remainder A is representation of 10, remainder E is representation of 14, and remainder F is representation of 15. Remainders are taken from bottom to top to present the hexadecimal number. So, $(69610)_{10}$ = $(10FEA)_{16}$.

16	69610	
16	4350	– A
16	271	-- E
16	16	-- F
16	1	-- 0
16	0	-- 1

Table 2-3

- **Hexadecimal to Decimal**

  The method for this conversion is same as converting from binary to decimal except the base value. Since hexadecimal has base 16, the "place values" correspond to the powers of 16. To convert to decimal, multiply each place value by the corresponding power of 16. Start this process by writing the powers of sixteen next to the digits of a hexadecimal number.

Example: **Convert $(C921)_{16}$ to decimal**

$$= C \times 16^3 + 9 \times 16^2 + 2 \times 16^1 + 1 \times 16^0$$
$$= 12 \times 16^3 + 9 \times 16^2 + 2 \times 16^1 + 1 \times 16^0$$
$$= 12 \times 4096 + 9 \times 256 + 2 \times 16 + 1 \times 1$$
$$= 49152 + 2304 + 32 + 1$$
$$= (51489)_{10}$$

Activity 2.5

Try to calculate that the binary of C92116 which is 11001001001000012.

2.2.3 Hexadecimal to Binary and Binary to Hexadecimal

- **Hexadecimal to Binary**

  To convert a hexadecimal number to binary, simply convert each hexadecimal digit to four digits binary value. To find the four digits binary value, see the Table 2-4.

Example:

Convert $(A23)_{16}$ (A23 in hexadecimal) to binary.

In this number, there are three hexadecimal digits. Binary of each digit is given as:

i. For A, the binary value is 1010
ii. For 2, the binary value is 0010
iii. For 3, the binary value is 0011

By combining all the binary values, we get 1010 0010 0011.

So, $(\mathrm{A23})_{16} = (101000100011)_2$

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

Table 2-4

Example:

Convert $(70C558)_{16}$ (70C558 in hexadecimal) to binary.

In this number, there are six hexadecimal digits and binary of each is given in different colours as:

i. For 7, the binary value is **0111**
ii. For 0, the binary value is **0000**
iii. For C, the binary value is **1100**
iv. For 5, the binary value is **0101**
v. For 5, the binary value is **0101**
vi. For 8, the binary value is **1000**

By combining all the binary values, we get **0111 0000 1100 0101 0101 1000**.

So, $(70C558)_{16} = (011100001100010101011000)_2$

- **Binary to Hexadecimal**

  This conversion is also very easy with the help of Table 2-4. In the given binary number, we start making groups of four digits from right to left and replace every group with a hexadecimal digit.

Example: **Convert $(11000001)_2$ to hexadecimal:**

The four digit binary groups in this binary number are given below where each group has four binary digits.

1100 0001

i. For 1100, the hexadecimal is C
ii. For 0001 the hexadecimal is 1

So, $(11000001)_2 = (C1)_{16}$

While making groups from right to left, if the left group has less than 4 binary digits then we simply add 0s on the left. For example, 1010011 has groups 101 0011 and by adding one 0 on the left, it becomes 0101 0011.

Example: **Convert $(110101111)_2$ to hexadecimal**

The groups in this binary number are given below where each group has maximum four binary digits.

1 1010 1111

The left most group in blue colour has only 1 binary digit and by adding 0s, we get:

0001 1010 1111

We replace each group with the respective hexadecimal and get:

1AF.

So, $(110101111)_2 = (1\text{AF})_{16}$

2.3 Memory and Data Storage

2.3.1 Memory

Computer memory is any physical device capable of storing data. Primarily there are following two types of memory.

1- Volatile Memory
2- Non-Volatile Memory

Both types of computer memories are shown in Figure 2-4. In the following, we discuss these two types in detail.

- **Volatile Memory (Primary Storage)**
  A device which holds data as long as it has power supply connected to it, is called Volatile Memory. Its best example is Random Access Memory (RAM), which holds memory only as long as it is connected to power source. As soon as the power supply is disconnected, all the data in RAM is cleared.

- **Non-Volatile Memory (Secondary Storage)**
  A device which can hold data even if it is not connected to any power source, is called Non Volatile Memory. The typical examples for Non Volatile Memory are hard drives, flash drives and memory cards installed in cell phones. Even if you turn off your PC, the data in your hard drive or flash drive stays intact.

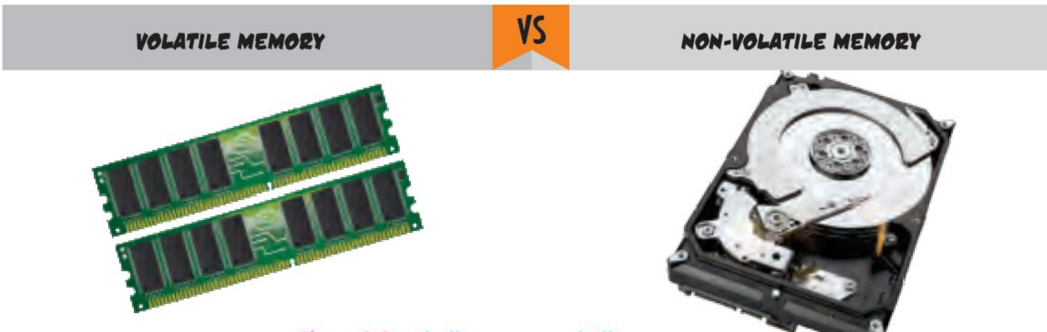

Figure 2-3 volatile vs non-volatile memory

2.3.2 Data Representation in Computer Memory

Digital computers store data in binary form. It means that whether it is a text, picture, movie or some application, it is stored in computer's memory in the form of 0s and 1s. All the characters on your keyboard has an associated code in binary. This code is called ASCII code of the character. ASCII stands for American Standard Code for Information Interchange. It is a de-facto standard for representation of data inside computer's memory. Table 2-5 presents the ASCII table which shows the code against each character on your keyboard. The codes are given in decimal form, but inside computer's memory they are represented after conversion to binary form.

Code	Character	Description	Code	Character	Description
32	**SP**	space	62	**>**	greater than
33	**!**	exclamation mark	63	**?**	question mark
34	**"**	double quote	64	**@**	"at" symbol
35	**#**	number sign	65	**A**	
36	**$**	dollar sign	66	**B**	
37	**%**	percent	67	**C**	
38	**&**	ampersand	68	**D**	
39	**'**	single quote	69	**E**	
40	**(**	left/opening parenthesis	70	**F**	
41	**)**	right/closing parenthesis	71	**G**	
42	*	asterisk	72	**H**	
43	**+**	plus	73	**I**	
44	**,**	comma	74	**J**	
45	**-**	minus or dash	75	**K**	
46	**.**	dot	76	**L**	
47	**/**	forward slash	77	**M**	
48	**0**		78	**N**	
49	**1**		79	**O**	
50	**2**		80	**P**	
51	**3**		81	**Q**	
52	**4**		82	**R**	
53	**5**		83	**S**	
54	**6**		84	**T**	
55	**7**		85	**U**	
56	**8**		86	**V**	
57	**9**		87	**W**	
58	**:**	colon	88	**X**	
59	**;**	semi-colon	89	**Y**	
60	**<**	less than	90	**Z**	
61	**=**	equal sign	91	**[**	left/opening bracket

92	****	back slash	110	**n**	
93	**]**	right/closing bracket	111	**o**	
94	**^**	caret/circumflex	112	**p**	
95	**_**	underscore	113	**q**	
96	**`**		114	**r**	
97	**a**		115	**s**	
98	**b**		116	**t**	
99	**c**		117	**u**	
100	**d**		118	**v**	
101	**e**		119	**w**	
102	**f**		120	**x**	
103	**g**		121	**y**	
104	**h**		122	**z**	
105	**i**		123	**{**	left/opening brace
106	**j**		124	**|**	vertical bar
107	**k**		125	**}**	right/closing brace
108	**l**		126	**~**	tilde
109	**m**		127	**DEL**	delete

Table 2-5 ASCII Table

Example: To store name of our country "Pakistan", in computer's memory, we need to store code of each letter in one byte. As the word "Pakistan" contains 8 letters, so 8 bytes are required for storage. It is demonstrated in Table 2-6.

Human's View about Memory	Code in Decimal	Code in Binary
'P'	80	1010000
'a'	97	1100001
'k'	107	1101011
'i'	105	1101001
's'	115	1110011
't'	116	1110100
'a'	97	1100001
'n'	110	1101110

Table 2-6

Activity 2.6

Write your complete name and give its presentation in binary format.

2.3.3 Storage Device

Any computing hardware that is used for storing, porting and extracting data, is called a storage device. It can hold or store information both temporarily and permanently. It can also be internal or external to a computer. An external storage device is a plug and play device, i.e., we just plug it to some port and start using it without turning off a computer. To attach an internal storage device (Hard disk or RAM) we need to turn off the computer. Internal storage devices are connected to some fixed slots.

Examples: RAM, Hard disk, CD, USB Flash Drive, etc.

Difference between memory and storage

Table 2-7 shows the difference between memory and storage.

Memory	Storage
Place where an application loads its data during processing	Usually the place where data is stored for long or short term.
Temporary storage device	Permanent storage device
Lesser in size	Greater in size
High accessing speed	Low accessing speed
It is called primary memory	It is called secondary memory

Table 2-7 Difference between Memory and Storage

2.4 Measurement of Size of Computer Memory

The smallest amount of data to be stored in computer's memory is a 0 or 1. It is called a bit. A collection of eight bits is called a byte. At least one byte is required to store any piece of information in a computer's storage. On both primary and secondary storage devices, data is stored in the form of bytes. In Table 2-8 different units of data are given.

Unit	Size
Bit	Smallest unit of data, can hold only one value: 0 or 1
Byte	Group of eight bits, enough space to store single ASCII character
Kilobyte	1KB = 1,024 bytes
Megabyte	1MB= (1,024) KB or $(1,024)^2$ bytes
Gigabyte	1GB= 1,024 MB or $(1,024)^3$ bytes
Terabyte	1TB= 1,024 GB or $(1,024)^4$ bytes
Petabyte	1PB= 1,024 TB or $(1,024)^5$ bytes

Table 2-8
Units of data

2.5 Boolean Algebra

2.5.1 Boolean Proposition

A proposition is a sentence that can either be *true* or *false*. For example, the following sentences are propositions.

1. "Someone from our school can join Pakistani Cricket Team"
2. "I will get A+ grade in board exam"
3. "I want to excel in mathematics"
4. "This year Pakistan Super League (PSL) final match will be played in Lahore"
5. "I play chess".

But the following sentences are not propositions

1. How are you?
2. Close the door.
3. Is it hot outside?

We can also assign some letter to a proposition, as shown in the following.

1- P = "I play chess".
2- Q = "I want to excel in mathematics"

Now, when we say P, it means that we are referring to proposition "I play chess", and when we say Q, it means that we are referring to proposition "I want to excel in mathematics".

Do you know?

True and False are called Boolean values. The idea was given by George Boole (2 November 1815 – 8 December 1864) in his book "The Laws of Thought".

2.5.2 Truth Values

Every proposition takes one of two values *true* or *false*, and these values are called the truth values. Truth value is given on the basis of truthfulness or falsity of a proposition.

Example:

Assume P = "Islamabad is the capital of Pakistan". You can assign the truth value *true* to this proposition. Now assume another proposition Q = "The sun rises in the west". The truth value for this proposition is *false*. If we have proposition R = "I have completed my homework", then the truth value depends on the person who is assigning it. If a person has completed his homework then he can assign truth value *true*, otherwise *false*.

2.5.3 Logical Operators (AND, OR, NOT)

Sometimes we assemble more than one propositions to make one proposition called a compound proposition. For example if we have the following two propositions:

1. Today is Monday
2. I am in school

Then "Today is Monday AND I am in school" is a compound proposition. Truth value of the compound proposition depends upon the truth values of the individual propositions and the logical operator used to connect the propositions. In this example "AND" is a logical operator. In the following, we discuss three most commonly used logical operators AND, OR and NOT.

AND Operator (.): If we use "AND" operator to connect two or more propositions, then the compound proposition is *true* only if all the connected propositions are *true*. AND operator can also be denoted by a **dot "."** symbol. It means that **P AND Q** may also be written as **P.Q**.

OR Operator (+): We can also use "OR" operator to connect two or more propositions e.g. "Today is Monday OR I am in School". In case of OR operator, the compound proposition is *true* if at least one proposition is *true*. In other words, the compound proposition is *false* only if all the propositions are *false*. OR operator can also be denoted by a **plus "+"** symbol. It means that **P OR Q** may also be written as **P + Q**.

NOT Operator: The logical operator "NOT" is not a connector but it is used to negate a proposition. For example, if P = "Today is Monday" then NOT(P) means "Today is not Monday". So, with NOT operator a *True* value becomes *false* and vice versa. Not operator can also be denoted by a "¬" symbol. It means that **NOT(P)** may also be written as $\neg P$.

2.5.4 Truth Table

A truth table is used to check whether a proposition is *True* or *False*. Usually it is used to check the truth value of a proposition where some logical operator is used. In the following, we discuss the truth tables for AND, OR and NOT operators.

Truth Table for AND operator: The truth table for **P AND Q** is given in Table 2-9. The first two columns are showing all the possible combinations of truth values of propositions P and Q, the third column is showing the resultant truth value of **P AND Q**. Assume:

P = It is raining
Q = Today is Sunday
P AND Q = It is raining and today is Sunday

If both P and Q are *True* then the **P AND Q** is also *True*, it means "It is raining on Sunday". This situation is shown on Row 1 of Table 2-9. Suppose it is raining but not on Sunday. Then P is *True* and Q is *False* due to which **P AND Q** is also *False* (row 2 of Table 2-9). In row 3 of Table 2-9, P is *False* and Q is *True.* It means "It is not raining on Sunday" which results in *False* value of **P AND Q**. In the last row both P and Q are *False,* which means "It is neither raining nor Sunday". So, the proposition "It is raining and today is Sunday" is false (row 4 of Table 2-9).

P	Q	P AND Q
T	T	T
T	F	F
F	T	F
F	F	F

Table 2-9

Truth Table for OR operator: For the same propositions P and Q, let's see the truth table for the expression **P OR Q.** P OR Q = "It is raining or it is Sunday". This compound proposition is *False* if it is not raining and today is not Sunday otherwise it is *True* as shown in Table 2-10.

P	Q	P AND Q
T	T	T
T	F	T
F	T	T
F	F	F

Table 2-10

Truth Table for NOT operator: We can also make truth table where NOT operator is used. Negation (also called **NOT**) is an operator that reverses the nature of a value, i.e., a value *True* becomes *False* and vice versa. The truth table for NOT operator is shown in Table 2-11.

P	NOT (P)
T	F
F	T

Table 2-11

Truth Table for complex Boolean expressions: We can make truth table for any combination of these operators. For example, if we need to make a truth table of "It is not raining and today is Sunday". It means the proposition NOT(P) AND Q. The truth table for this compound proposition is shown Table 2-12.

P	NOT(P)	Q	NOT(P) AND Q
T	F	T	F
T	F	F	F
F	T	T	T
F	T	F	F

Table 2-12

2.5.5 Laws of Boolean Algebra

The laws of Boolean Algebra help us to simplify complex Boolean expressions. Some laws are discussed in the following.

- **Commutative Law**

  Commutative Law states that the order of application of two separate propositions is not important. *So,*

  a) A . B = B . A (The order in which two variables are AND'ed makes no difference.)

  b) A + B = B + A (The order in which two variables are OR'ed makes no difference.)

  We can use truth tables (Table 2-13a, Table 2-13b) to verify this law for AND and OR operations respectively.

A	B	$A \cdot B$	$B \cdot A$
F	F	F	F
F	T	F	F
T	F	F	F
T	T	T	T

Table 2-13a

A	B	A + B	B + A
F	F	F	F
F	T	F	F
T	F	F	F
T	T	T	T

Table 2-13b

We can observe from Table 2-13a that both the columns A · B and B · A contain same values in each row. Thus it verifies the commutative law for AND operation. Similarly we can verify for OR operation from Table 2.13b.

- **Associative Law**

  This law is for several variables. According to this law there is no change in results if a grouping of expressions is changed. This law is quite same in case of AND and OR operators.

  a) $(A + B) + C = A + (B + C)$

  b) $(A . B) . C = A . (B . C)$

  In order to verify the associative law for OR operation, we can observe the Truth Table presented in Table 2-14. Both columns **(A+B)+C** and **A+(B+C)** contain same values in each row. It verifies the associative law for OR operation.

A	*B*	*C*	*A + B*	*B + C*	*(A+B)+C*	*A+(B+C)*
F	F	F	F	F	F	F
F	F	T	F	T	T	T
F	T	F	T	T	T	T
F	T	T	T	T	T	T
T	F	F	T	F	T	T
T	F	T	T	T	T	T
T	T	F	T	T	T	T
T	T	T	T	T	T	T

Table 2-14

Similarly, we can observe Truth Table presented in Table 2-15 for verification of Associative Law for AND operation.

A	B	C	$A \cdot B$	$B \cdot C$	$(A \cdot B) \cdot C$	$A \cdot (B \cdot C)$
F	F	F	F	F	F	F
F	F	T	F	F	F	F
F	T	F	F	F	F	F
F	T	T	F	T	F	F
T	F	F	F	F	F	F
T	F	T	F	F	F	F
T	T	F	T	F	F	F
T	T	T	T	T	T	T

Table 2-15

- **Distributive Law**

  This law is discussed in two ways, i.e., “AND over OR” and “OR over AND”.

  a) *A . (B + C) = (A . B) + (A . C)* (AND over OR)

  b) *A + (B . C) = (A + B) . (A + C)* (OR over AND)

We can verify the distributive law for (AND over OR) operation by using Table 2-16.

A	B	C	$B+C$	$A\cdot B$	$A\cdot C$	$A\cdot(B+C)$	$A\cdot B+A\cdot C$
F	F	F	F	F	F	F	F
F	F	T	T	F	F	F	F
F	T	F	T	F	F	F	F
F	T	T	T	F	F	F	F
T	F	F	F	F	F	F	F
T	F	T	T	F	T	T	T
T	T	F	T	T	F	T	T
T	T	T	T	T	T	T	T

Table 2-16

Activity 2.7

Draw the truth table to verify $\boldsymbol{A}+(\boldsymbol{B}\cdot\boldsymbol{C})=(\boldsymbol{A}+\boldsymbol{B})\cdot(\boldsymbol{A}+\boldsymbol{C})$

- **Identity Law**

  If a variable is OR’ed with a False, the result is always equal to that variable. And if a variable is AND´ed with a True, the result is always equal to that variable.

  a) A OR False = A, A variable OR’ed with False is always equal to that variable

  b) A AND True = A, A variable AND’ed with True is always equal to that variable

2.5.6 Logical Expressions

We get a logical expression when some logical operator is applied to the Boolean proposition(s). For example, $\boldsymbol{P\ AND\ Q}, \neg(\boldsymbol{P\ OR\ Q}), \boldsymbol{P\ OR\ Q}, \boldsymbol{etc}.$, In the tables, Table 2-14, 2-15 and 2-16 the truth tables are according to some logical expressions.

Do you know?

By negating a negative proposition, we get a positive proposition. For example,

- P=It is sunny today
- ¬P=It is not sunny today
- ¬¬P=It is sunny today

Similarly,

- Q=It is not Friday today
- ¬Q=It is Friday today
- ¬¬Q=It is not Friday today

SUMMARY

- Binary language consists of 0s and 1s. Computer understands only binary language.
- Decimal number system has base 10 as it uses ten digits from 0 to 9.
- Hexadecimal system has total 16 numbers, i.e., 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.
- Computer memory is a physical device capable of storing information temporarily or permanently.
- A device which holds the data as long as it has power supply connected to it and loses the memory when there is no power supply connected to it is called Volatile Memory or temporary memory.
- A device which can hold data even if it is not connected to any power source is called Non Volatile Memory or permanent memory.
- A storage device is a hardware that is used for storing, porting and extracting data.
- Boolean states either can be True or False.
- A truth table is used to show whether the statement is true or false.
- Laws of Boolean algebra:
- Associative law

  $(A + B) + C = A + (B + C)$
- Commutative law

  $A + B = B + A$
- Distributive law

  $A + (B.C) = (A + B).(A + C)$
- Identity law

  $A + 0 = A$

EXERCISE

2.1 Multiple Choice Questions

1. Expression *(A + B) · (A + C)* is equal to _________.

(i) A + (B.C) (ii) A.B + A.C
(iii) A.(B. C) (iv) A + (B + C)

2. The order of application of two separate terms is not important in _________________.

(i) Associative Law (ii) Commutative Law
(iii) Distributive Law (iv) Identity Law

3. "Is it cold outside" is ________________

(i) Boolean Proposition (ii) Categorical proposition
(iii) Moral propositions (iv) None of above

4. Number "17" is equal to ________ in binary system.

(i) 10000 (ii) 10110
(iii) 10001 (iv) 10100

5. 1 Petabyte is equal to __________

(i) $(1,024)^4$ bytes (ii) $(1,024)^6$ bytes
(iii) $(1,024)^5$ bytes (iv) $(1,024)^7$ bytes

6. Hexadecimal system has total _______numbers.

(i) 17 (ii) 16
(iii) 18 (iv) 15

2.2 Answer the following questions.

1. Convert $(69610)_{10}$ to Hexadecimal.
2. Differentiate between volatile and non-volatile memory.
3. Store the word "Phone" in computer memory starting from address 7003 where each letter needs one byte to store in the memory.
4. Differentiate between temporary and permanent storage.
5. Write the truth table for X AND Y where

X = It is sunny

Y = Today is Monday

2.3 Fill in the Blanks

1. Temporary memory is ______ and permanent memory is _______.
2. Data to a processor is provided through ______.
3. At least _____ byte is required to store any piece of information in a computer's memory.
4. ______________ is used to assemble more than one propositions into one proposition.
5. In primary and secondary storages, data is stored in the form of _______.
6. According to ________ law there is no change in results if priority of expressions is changed.

2.4 Perform the following conversions

1. $(ABCD)_{16}$ to binary
2. $(0010110010001101001)_2$ to hexadecimal

Activity 2.8

Teacher will display a chart where alphabets and their codes are written. Class is divided into two groups and each group writes at least 5 names in binary format. The famous names are selected from Pakistan Independence movement e.g., "Molana Muhammad Ali Johar". Both groups exchange their data and produces original names. The group which deciphers the code to actual names in less time will win.