**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**SCHOOL OF COMPUTING**

|  |  |  |
| --- | --- | --- |
| **MCQS:** |  | |
|  | | |
| **Question 1:** | | What is the 2's complement of the binary number 110101? |
| a) 001011 | | |
| b) 001010 | | |
| c) 001100 | | |
| d) 001101 | | |
| **Question 2:** | | Which binary complement is used to represent negative numbers in |
| most computer systems?   1. 1's complement 2. 2's complement 3. 9's complement 4. 10's complement | | |
| **Question 3:** | | In a 4-bit binary system, what is the 1's complement of the binary |
| number 1001? | | |
| a) 0110 | | |
| b) 0111 | | |
| c) 1000 | | |
| d) 1001 | | |
| **Question 4:** | | What is the range of numbers that can be represented using an 8-bit 2's |
| complement representation?  a) -128 to 127  b) -127 to 128  c) -255 to 255  d) 0 to 255 | | |
| **Question 5:** | | Which operation is used to calculate the 2's complement of a binary |
| number?   1. Inversion (complement) of all bits followed by adding 1 2. Inversion (complement) of all bits 3. Adding 1 to the binary number 4. Subtracting 1 from the binary number | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Question 6:** | | What is the 1's complement of the decimal number 75? | |
| a) 1001011 | | | |
| b) 0110100 | | | |
| c) 0110101 | | | |
| d) 0110110 | | | |
| **Question 7:** | | Which complement is used for performing binary subtraction using | |
| addition?   1. 1's complement 2. 2's complement 3. 9's complement 4. 10's complement | | | |
| **Question 8:** | | If the 1's complement of a binary number is 101010, what is the original | |
| binary number? | | | |
| a) 010101 | | | |
| b) 101010 | | | |
| c) 010100 | | | |
| d) 101011 | | | |
| **Question 9:** | | Which binary complement method has a sign-magnitude representation | |
| for negative numbers?   1. 1's complement 2. 2's complement 3. Excess-3 code 4. 10's complement | | | |
| **Question 10:** | | | What is the purpose of using complements in binary arithmetic? |
| 1. To simplify multiplication operations 2. To perform division operations 3. To simplify addition and subtraction operations 4. To represent large binary numbers | | | |
| **Answers:** |  | | |

1. b) 001010

2. b) 2's complement 3. a) 0110

4. a) -128 to 127

5. a) Inversion (complement) of all bits followed by adding 1 6. b) 0110100

7. b) 2's complement 8. a) 010101

1. a) 1's complement
2. c) To simplify addition and subtraction operations

Logic Gates

1.The output of an AND gate with three inputs, A, B, and C, is HIGH when \_\_\_\_\_\_\_\_.

A = 1, B = 1, C = 0

A = 0, B = 0, C = 0

A = 1, B = 1, C = 1

A = 1, B = 0, C = 1

2.If a signal passing through a gate is inhibited by sending a LOW into one of the inputs, and the output is HIGH, the gate is a(n):

AND

NAND

NOR

OR

3.Which of the following logical operations is represented by the + sign in Boolean algebra?

inversion

AND

OR

complementation

4.Output will be a LOW for any case when one or more inputs are zero for a(n):

OR gate

NOT gate

AND gate

NOR gate

5. The output of a NOR gate is HIGH if \_\_\_\_\_\_\_\_.

all inputs are HIGH

any input is HIGH

any input is LOW

all inputs are LOW

6. The format used to present the logic output for the various combinations of logic inputs to a gate is called a(n):

a.Boolean constant

b.boolean variable

c.Truth table

d.input logic function

7. If a 3-input AND gate has eight input possibilities, how many of those possibilities will result in a HIGH output?

a.1

b.2

c.7

d.8

8.  How many AND gates are required to realize Y = CD + EF + G?  
a)4  
b)5  
c)3   
d) 2

9. The NOR gate output will be high if the two inputs are \_\_\_\_\_\_\_\_\_\_  
a)00  
b)01  
c)10  
d) 11

10. How many two-input AND and OR gates are required to realize Y = CD+EF+G?  
a)2,2  
b)2,3  
c)3,3  
d)3,2

11. A universal logic gate is one which can be used to generate any logic function. Which of the following is a universal logic gate?  
a) OR  
b) AND  
c) XOR  
d) NAND

12. How many two input AND gates and two input OR gates are required to realize Y = BD + CE + AB?  
a)3,2  
b)4,2  
c)1,1  
d)2, 3

13.Which of the following are known as universal gates?  
a)NAND&NOR  
b)AND&OR  
c)XOR&OR  
d) EX-NOR & XOR

14. **A NAND gate has —– inputs and —— outputs.**

a) High input and High output

b) High input and low output

c) Low input and low output

d) Low input and high output

15. **What will be the output of the given logic gate?**



1. NOR
2. NAND
3. AND
4. OR

16. **XOR circuits can be constructed using**

* 1. OR gates only
  2. AND, OR gates
  3. AND, NOT gates
  4. AND, NOT and OR gates

17. **What is the Truth Table?**

* 1. It is a table representing what output will one get for a given input.
  2. It is a table representing what input and output will be given by a given boolean operator.
  3. Both of these
  4. None of these

18. **Which of the following gates can function on a single input?**

* 1. NOT
  2. AND
  3. OR
  4. None

19. **What is the boolean operation performed by an OR gate?**

* 1. Equivalent of addition
  2. Equivalent of multiplication
  3. Equivalent of dificsin
  4. Equivalent of subtraction.

20. **Who was the inventor of the idea of logic gates?**

* 1. George Boole
  2. Bardeen
  3. Claude Shannon
  4. Kornard Zuse

* + 1. A 9-bit binary number with even parity has the first 8 bits as 11011011. What should be the 9th bit to maintain even parity?

a) 0

b) 1

* + 1. Given the 12-bit binary sequence 101001101011, if odd parity is applied, how many '1' bits must be present to maintain odd parity?

a) 6

b) 7

c) 8

d) 9

23. Which of the following is the correct BCD representation of the decimal number 89?

1. 10001001
2. **10011001**
3. 10100001
4. 10100101
5. Carry out BCD subtraction for (68) – (61) using 10’s complement method. a)00000111

b)01110000

c)100000111

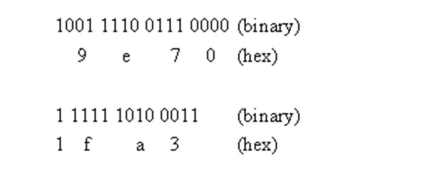
d) 011111000

25.The decimal digit in Binary Coded Decimal (BCD) can be represented by

**D) 4 input lines**

1. **1 input line**
2. **2 input lines**
3. **3 input lines**

26.Write the corresponding 1 hex digit for each group. (2)



27. An adjunct unit to main memory, which performs fast and small in size is called ------------------------

**a. Cache (1)**

b. RAM

c. ROM

d. Secondary memory

28. Which of the following is a type of architecture used in the computers nowadays?

a) Microarchitecture (1)

b) Harvard Architecture

**c) Von-Neumann Architecture**

d) System Design

29. The purpose of Memory Address Register (MAR) is to ------------ (1)

**a. Store address location of the current instruction to be executed**

b. Store address location of the next instruction to be executed

c. Store operands of the current instruction

d. Store operands of the next instruction

* + 1. The octal number 35 is equivalent to which binary number?

a) 011101

b) 001110

c) 111010

d) **100101**

* + 1. What is the octal representation of the decimal number 95? a) 115

b) 137

c) **127**

d) 155

1. 32. What is the octal equivalent of the binary number 110110?
2. a) 320

b) 316

c) 216

# d) 332

1. 33. What is the hexadecimal representation of the decimal number 255?

# FF

* 1. EF
  2. 1FF

d) 100

1. 34. How many hexadecimal digits are needed to represent the binary number 11010101?
   1. 6
   2. 4
   3. 8
   4. 2
2. 35. The maximum value that can be represented using a single hexadecimal digit is:

# 15

* 1. 10
  2. 9
  3. 16

**PART-B**

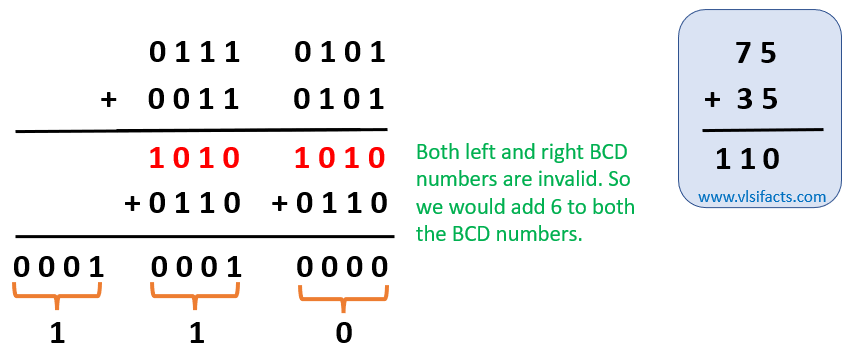
1. Add (-15)10 to (+6)10 using 2’s complement. Use eight bits (byte) to represent the signed numbers.

Solution (+6) 10 → 00000110 +(-15) 10 → +11110001 → 2’s complement of (- 15) (-9) 10 11110111 → Answer └ 2’s complement of ( - 9 ) Note. The result is in 2’s complement form because it is a negative number.

|  |  |  |
| --- | --- | --- |
|  | 2. Given two binary numbers, explain the process for performing binary  division. Provide an example with step-by-step calculations. | |
|  | | Example: Perform binary division for 110101 / 101. 1010 (Quotient)  +  101 | 110101 (Dividend)  - 101  00101  - 101  0001 |
|  | | |

1. Find the sum of the BCD numbers 01110101 + 00110101

Solution:



The decimal number of the given BCD numbers are as below: 01110101BCD = 7510 and 00110101BCD = 3510

1. Perform each of the subtraction operations indicated below using addition

and the two’s complement of the subtrahend.

A) (100101)2 – (11011)2

Subtrahend: 11011

1's Complement: 00100

1's Complement: 00100

+ 1

2's Complement: 00101

100101

+ 00101

101110

(101110)2 = 1\*2^5 + 0\*2^4 + 1\*2^3 + 1\*2^2 + 1\*2^1 + 0\*2^0

= 32 + 0 + 8 + 4 + 2 + 0

= 46

B) (1101011)2 – (111010)2

Step 1: Convert to 2's complement representation

For (1101011)₂ (minuend):

* Invert the bits: (0010100)₂
* Add 1 to the inverted value: (0010101)₂

For (111010)₂ (subtrahend):

* Invert the bits: (000101)₂
* Add 1 to the inverted value: (000110)₂

Step 2: Perform addition of minuend and 2's complement of subtrahend (0010101)₂ + (000110)₂

0010101

C) (1110111)2 – (10110111)2

Step 1: Convert the numbers to 2's complement representation:

For (1110111)\_2: Since the number is positive (MSB is 0), the 2's complement representation remains the same.

For (10110111)\_2: The number is negative (MSB is 1). To convert it to 2's complement, we first find the 1's complement by flipping all the bits:

1's complement of (10110111)\_2: (01001000)\_2

Then, we add 1 to the 1's complement to get the 2's complement:

2's complement of (10110111)\_2: (01001000)\_2 + (00000001)\_2 = (01001001)\_2

Step 2: Perform binary addition (1110111)\_2 + (01001001)\_2: Step 3: Carry-out from the MSB is 0 (0 + 1 = 1), so no carry-out. Step 4: The result is (00110100)\_2.

Step 5: Convert the result to decimal:

(00110100)\_2 = 32 + 16 + 4 = 52

Therefore, (1110111)\_2 - (10110111)\_2 using 2's complement is equal to 52 in decimal.

5. An electronic system will only operate if three switches P, S and T are correctly set. An output signal (X= 1) will occur if R and S are both in the ON position or if R is in the OFF position and S and T are both in the ON position. Design a logic circuit and write the Logic notation to represent the above situation and draw the truth table.

6.A traffic light system uses logic gates as part of the control system. The system is operated when the output D has the value 1. This happens when: either (a) signal A is red or (b) signal A is green and signals B and C are both red (NOTE: You may assume for this problem that red = 0 and green = 1). Design a logic circuit and write the Logic notation to represent the above situation and draw the truth table.

**Gray Code to Binary Conversion:**

**Convert the Gray code 0101 to its equivalent binary representation.**

To convert a Gray code to its equivalent binary representation, you can follow these steps:

Write down the first bit of the Gray code as it is, since the first bit of the binary code is the same as the first bit of the Gray code.

For each subsequent bit in the Gray code, XOR it with the previous bit of the Gray code to get the corresponding bit in the binary code.

Let's apply this process to the Gray code 0101:

Gray code: 0 1 0 1

Binary code: 0

For the first bit, simply copy it over to the binary code. Gray code: 0 1 0 1

Binary code: 0 1

For the second bit, XOR the second Gray code bit (1) with the first Gray code bit (0) to get the second binary bit.

Gray code: 0 1 0 1

Binary code: 0 1 1

For the third bit, XOR the third Gray code bit (0) with the second Gray code bit (1) to get the third binary bit.

Gray code: 0 1 0 1

Binary code: 0 1 1 1

For the fourth bit, XOR the fourth Gray code bit (1) with the third Gray code bit (0) to get the fourth binary bit.

Gray code: 0 1 0 1

Binary code: 0 1 1 1

So, the equivalent binary representation of the Gray code 0101 is 0111.

* + **Given the Gray code 11001, find its binary equivalent.**

Write down the first bit of the Gray code as it is, since the first bit of the binary code is the same as the first bit of the Gray code.

For each subsequent bit in the Gray code, XOR it with the previous bit of the Gray code to get the corresponding bit in the binary code.

Let's apply this process to the Gray code 11001:

Gray code: 1 1 0 0 1

Binary code: 1

For the first bit, simply copy it over to the binary code. Gray code: 1 1 0 0 1

Binary code: 1 1

For the second bit, XOR the second Gray code bit (1) with the first Gray code bit (1) to get the second binary bit.

Gray code: 1 1 0 0 1

Binary code: 1 1 1

For the third bit, XOR the third Gray code bit (0) with the second Gray code bit (1) to get the third binary bit.

Gray code: 1 1 0 0 1

Binary code: 1 1 1 1

For the fourth bit, XOR the fourth Gray code bit (0) with the third Gray code bit (0) to get the fourth binary bit.

Gray code: 1 1 0 0 1

Binary code: 1 1 1 1 0

For the fifth bit, XOR the fifth Gray code bit (1) with the fourth Gray code bit (0) to get the fifth binary bit.

Gray code: 1 1 0 0 1

Binary code: 1 1 1 1 0

So, the equivalent binary representation of the Gray code 11001 is 11110.

**Excess-3 and BCD:**

* + **Convert the decimal number 42 to Excess-3 representation.**

To convert a decimal number to Excess-3 representation, follow these steps:

* + - Convert the decimal number to its binary equivalent.
    - Add 0011 (binary for 3) to each group of 4 bits in the binary representation. Let's convert the decimal number 42 to Excess-3 representation:
      * Convert 42 to binary: 42 = 101010
      * Add 0011 to each group of 4 bits:
        + 1010 + 0011 = 1101
        + 0010 + 0011 = 0101

So, the Excess-3 representation of the decimal number 42 is 11010101.

* + **Convert the decimal number 75 to BCD representation.**

Let's convert the decimal number 75 to BCD representation: Decimal number: 75

Convert each decimal digit to its 4-bit binary equivalent: Digit 7: 0111

Digit 5: 0101

Combine the binary representations of each digit: BCD representation: 0111 0101

So, the BCD representation of the decimal number 75 is 0111 0101.

**Parity:**

* + **Calculate the even parity bit for the binary number 110110.**

In even parity, the parity bit is added to a binary number to ensure that the total count of 1s (including the parity bit) is always even. If the binary number already has an even count of 1s, the parity bit is set to 0. If the count of 1s is odd, the parity bit is set to 1.

Given the binary number: 110110

Let's count the number of 1s in the binary number:

Number of 1s: 4

Since the count of 1s (4) is already even, the even parity bit should be set to 0. So, the even parity bit for the binary number 110110 is 0.

* + **Determine the odd parity bit for the binary sequence 1001010.**

Given the binary sequence: 1001010

Let's count the number of 1s in the binary sequence:

Number of 1s: 3

Since the count of 1s (3) is an odd number, the odd parity bit should be set to 1. So, the odd parity bit for the binary sequence 1001010 is 1.

1. Convert (0.7854) 10 to binary. (2)

Ans:

0.7854 x 2 = 1.5708; a -1 = 1

0.5708 x 2 = 1.1416; a -2 = 1

0.1416 x 2 = 0.2832; a -3 = 0

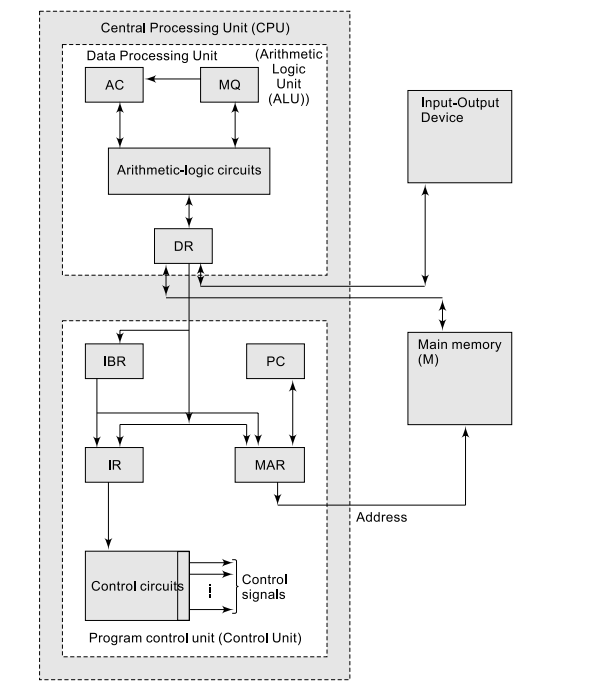
0.2832 x 2 = 0.5664; a -4 = 0

The answer is (0.7854)10 = (0.1100)2

2. How the connection between the processor and main memory can be established? And explain the same (12)

Diagram (4) – CPU, Main Memory

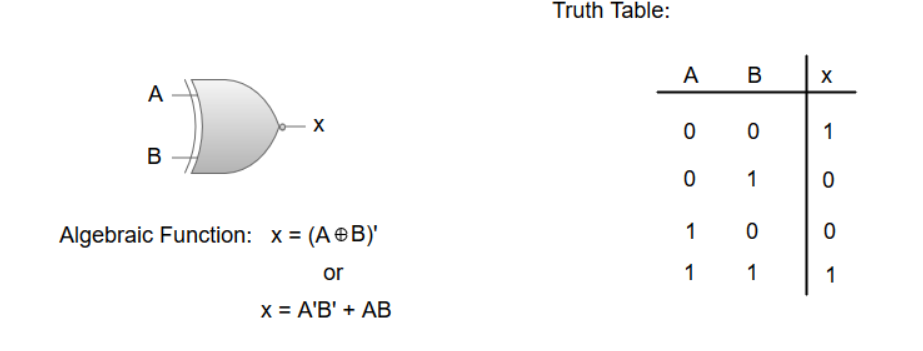
Description (8) – CPU, Control unit, Input device, Output device, main memory, registers



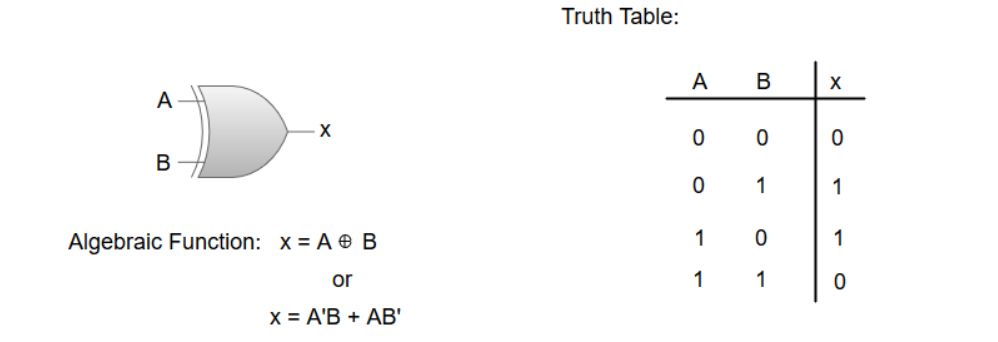
* CPU(Central processing unit)
  + CU(Control Unit)
  + ALU(Arithmetic and logic unit)
  + Registers
    - PC
    - IR
    - AC
    - MAR
    - MDR
    - BUSES
* I/O Device
* MEMORY UNIT

3. Represent the truth table and symbol for the following logic Gates. (4)

1. Exclusive NOR (2)



2. XOR Gate (2)



4. Compute 2’s complement for the following binary values: (4)

a. 1001

b. 0001

a.

1’s complement representation: 0110

2’s complement representation: 0111

b.

1’s complement representation: 1110

2’s complement representation: 1111

5. List out the steps to implement the boolean expression using NAND gate (4)

a. Represent function using AND and OR gate

b. Convert all OR gates to invert OR and all AND gates to NAND gates

c. Check all the bubbles in the representation, if not balanced, insert an inverter on the same line

d. Convert all invert OR and NOT to NAND gate

6. Convert the decimal number (8)10 to excess 3 code. (2)

Ans:

Binary number of 8: 1000

Binary number of 3: 0011

Adding results : 1011

7. In a small town, there are three temples in a row and a well in front of each temple. A pilgrim came to the town with certain number of flowers. Before entering the first temple, he washed all the flowers he had with the water of well. To his surprise, flowers doubled. He offered few flowers to the God in the first temple and moved to the second temple. Here also, before entering the temple he washed the remaining flowers with the water of well. And again his flowers doubled. He offered few flowers to the God in second temple and moved to the third temple. Here also, his flowers doubled after washing them with water. He offered few flowers to the God in third temple. There were no flowers left when pilgrim came out of third temple and he offered same number of flowers to the God in all three temples. What is the minimum number of flowers the pilgrim had initially? How many flower did he offer to each God?

(12)

Answer:

Let's assume that the pilgrim had X number of flowers initially. Suppose that he offered Y flowers to the each God.

Before entering into the first temple, magical well doubled the flowers he had initially. That means he had 2X flowers before entering into first temple. After offering Y flowers, he had 2X - Y flowers.

Again after visiting first temple he washed flowers in magical well where number of flowers gets doubled. Now, he had 4X - 2Y.

Out of these 4X - 2Y, he offered Y flowers to God in second temple. So he had 4X - 2Y - Y = 4X - 3Y flowers after visiting second temple.

These 4X - 3Y doubled to 8X - 6Y after washing in magical water well.

At third temple, he offered all the flowers he had which in turn equal to Y as we assumed.

8X - 6Y = Y

8X = 7Y

X/Y = 7/8

This is the ratio of the flowers that pilgrim had to the flowers he offered to each God.

n general, the pilgrim had 7N flowers initially and he offered 8N flowers to each God, where N = 1, 2, 3, 4,

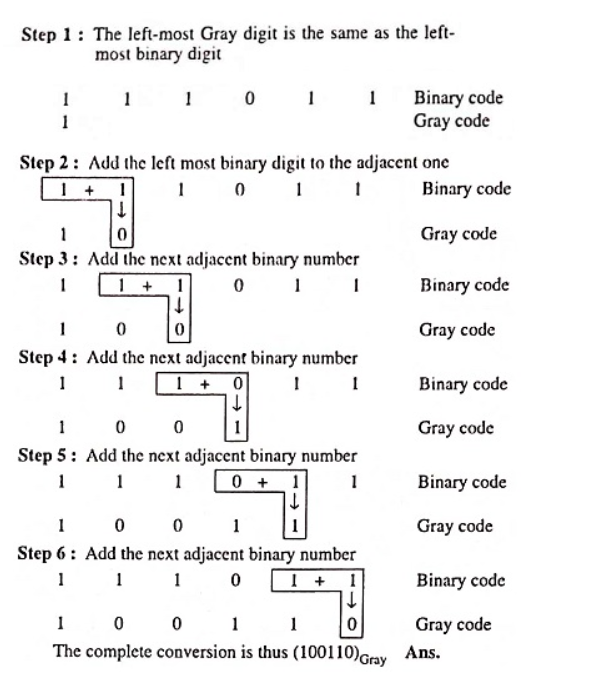
Let's cross verify same with N = 1, meaning that the pilgrim had 7 flowers initially & offered 8 flowers to each God.

Before entering into first temple, the flowers doubled to 14. Out of which, 8 offered at first temple & left 6.

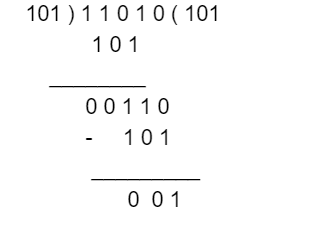
Again 6 doubled to 12 at magical well & 8 out of 12 offered to God at second temple leaving behind 4.

Those 4 again doubled to 8 by magical well & all 8 offered to God at third temple.

8. (6)



9. Divide 11010 by 101 (2)



Ans: (101)2

10. Convert the given binary number (01100100)2 to ascii text conversion (4)

11. Suppose you need to convert the octal number 637.43 to a decimal number using both the iterative method - multiplying each digit by 8 raised to the appropriate power) and discuss the advantages and limitations of this approach.

**Octal Number: 637.43**

Decimal Conversion: 6 \* 8^2 + 3 \* 8^1 + 7 \* 8^0 + 4 \* 8^-1 + 3 \* 8^-2 Now let's calculate:

6 \* 64 + 3 \* 8 + 7 \* 1 + 4 \* 0.125 + 3 \* 0.015625 384 + 24 + 7 + 0.5 + 0.046875

415.546875

So, the octal number 637.43 is approximately equal to the decimal number 415.546875 using the iterative method.

# Advantages of the Iterative Method:

* Clear Understanding: The iterative method breaks down the conversion process into simple steps, making it easy to understand and follow.
* Flexibility: This method can be applied to convert numbers between different bases, not just octal to decimal.
* No Special Tools: You only need basic arithmetic operations (multiplication, addition, and division) to perform the conversion.

# Limitations of the Iterative Method:

* Cumbersome for Large Numbers: For large numbers with many digits, the iterative method becomes time-consuming and prone to errors.
* Limited Efficiency: Calculating powers of the base (in this case, 8) can be computationally expensive, especially for repeated calculations.
* Not Practical for Complex Numbers: The iterative method becomes more complex when dealing with floating-point numbers or non-integer powers.
* Requires Attention to Detail: There's a risk of making errors while manually performing multiple calculations.

12. You are given a decimal number 4921. Instead of converting it directly to hexadecimal, perform the conversion in two stages: first, convert it to octal, and then convert the octal result to hexadecimal. Discuss any patterns or insights you observe during this process.

* **Step 1:** Convert Decimal to Octal To convert 4921 to octal, we'll repeatedly divide by 8 and record the remainders.
* 4921 ÷ 8 = 615 remainder 1 615 ÷ 8 = 76 remainder 7 76 ÷ 8 = 9 remainder 4 9 ÷

8 = 1 remainder 1 1 ÷ 8 = 0 remainder 1

* Reading the remainders from bottom to top, we get the octal representation: 14711.
* **Step 2:** Convert Octal to Hexadecimal Now, let's convert the octal number 14711 to hexadecimal. Since octal digits correspond to groups of three binary digits, and hexadecimal digits correspond to groups of four binary digits, we'll first convert octal to binary and then binary to hexadecimal.
* Octal 1 = Binary 001 Octal 4 = Binary 100 Octal 7 = Binary 111 Octal 1 = Binary 001 Octal 1 = Binary 001
* Combining these binary digits in groups of four: 001 100 111 001 001
* Now we can convert these groups to hexadecimal: 0011 = 3 0011 = 3 1001 = 9 1001 = 9
* So, the octal number 14711 is equivalent to the hexadecimal number 3399.

13.You encounter an octal number 127.127 and must convert it to decimal and hexadecimal representations.

* **Step 1: Convert Octal to Decimal** For the whole number part: 1 \* 8^2 + 2 \* 8^1

+ 7 \* 8^0 = 64 + 16 + 7 = 87

* For the fractional part: 1 \* 8^-1 + 2 \* 8^-2 + 7 \* 8^-3 = 0.125 + 0.03125 + 0.00390625 = 0.16015625
* Adding the whole number and fractional parts: 87 + 0.16015625 = 87.16015625
* **Step 2: Convert Decimal to Hexadecimal** To convert the decimal number 87.16015625 to hexadecimal, we'll first convert the whole number part and then the fractional part.
* Whole Number Part: Dividing 87 by 16 gives us a quotient of 5 and a remainder of 7. Dividing 5 by 16 gives us a quotient of 0 and a remainder of 5.
* Reading the remainder from bottom to top, the whole number part in hexadecimal is 57.
* Fractional Part: Converting the fractional part to hexadecimal: 0.16015625 \* 16 = 2.5625
* Again, we'll separate the whole number and fractional parts: 2 (whole number part in hexadecimal) 0.5625 (remaining fractional part)
* To convert the remaining fractional part, we'll multiply it by 16: 0.5625 \* 16 = 9.0
* So, the fractional part in hexadecimal is 0.9.
* Combining the whole number and fractional parts in hexadecimal notation: 87.16015625 (decimal) = 57.09 (hexadecimal)

**Part C**

1. **You are designing a control system for a stepper motor. The motor position is tracked using Gray code, and you receive a new value in binary: 101001. What Gray code value should you send to the motor controller to move one step in the positive direction?**

**Answer:**

To move one step in the positive direction for a stepper motor that uses Gray code for position tracking, you need to send a Gray code value to the motor controller that corresponds to the new position after the step.

Given the current position in binary: 101001

To move one step in the positive direction, you need to increment this value by one, considering Gray code properties. In Gray code, incrementing by one means changing only one bit.

Let's analyze the bit changes in the current binary position:

* + Bit 0: 1
  + Bit 1: 0
  + Bit 2: 1
  + Bit 3: 0
  + Bit 4: 0
  + Bit 5: 1

To move one step in the positive direction, you should identify the bit that needs to change. Following Gray code's property of changing only one bit, find the rightmost bit that changes from 0 to 1. In this case, that's Bit 5.

To create the new Gray code value for the incremented position, we'll use the XOR operation between the current binary value (101001) and the binary value with the rightmost bit (Bit 5) set to 1 (001001):

101001 XOR 001001 = 100000

So, the Gray code value you should send to the motor controller to move one step in the positive direction is 100000.

**2.A gaming controller sends input data to a console using Gray code. The last transmitted Gray code was 1101, indicating a button press. The current received Gray code is 1011. Determine which button was released after the press based on the given Gray codes.**

**Answer**

To determine which button was released after a button press, we need to compare the last transmitted Gray code with the current received Gray code and identify the bit position where a change occurred. The bit position that changed will correspond to the button that was released.

Given the last transmitted Gray code: 1101 Current received Gray code: 1011 Let's compare the two Gray codes bit by bit:

* + 1st bit (leftmost): No change (1)
  + 2nd bit: Change from 1 to 0 (released button)
  + 3rd bit: Change from 0 to 1 (pressed button)
  + 4th bit (rightmost): No change (1)

Based on the comparison, the 2nd bit changed from 1 to 0, indicating that the button associated with the 2nd bit was released after the press.

So, the button that was released after the press is the button corresponding to the 2nd bit of the Gray code.

**3.A computer system sends data using even parity. You receive the following byte: 01100110. Determine whether the data was transmitted with or without errors.**

**Answer:**

In even parity, the parity bit is set (usually to 1) so that the total count of 1s in the data and the parity bit is always an even number. If the count of 1s in the data and parity is already even, then the parity bit is set to 0.

Given the received byte: 01100110

Let's count the number of 1s in the received data: Number of 1s: 4

Since the count of 1s (4) is already even, in even parity, the parity bit should be set to 0. However, in the received byte, the last bit is 0. This means that there's a mismatch between the parity bit and the actual data.

Therefore, based on the even parity scheme, the data was transmitted with an error. The parity check indicates that the received byte is inconsistent with even parity, suggesting that there might be a bit-flip error during transmission.

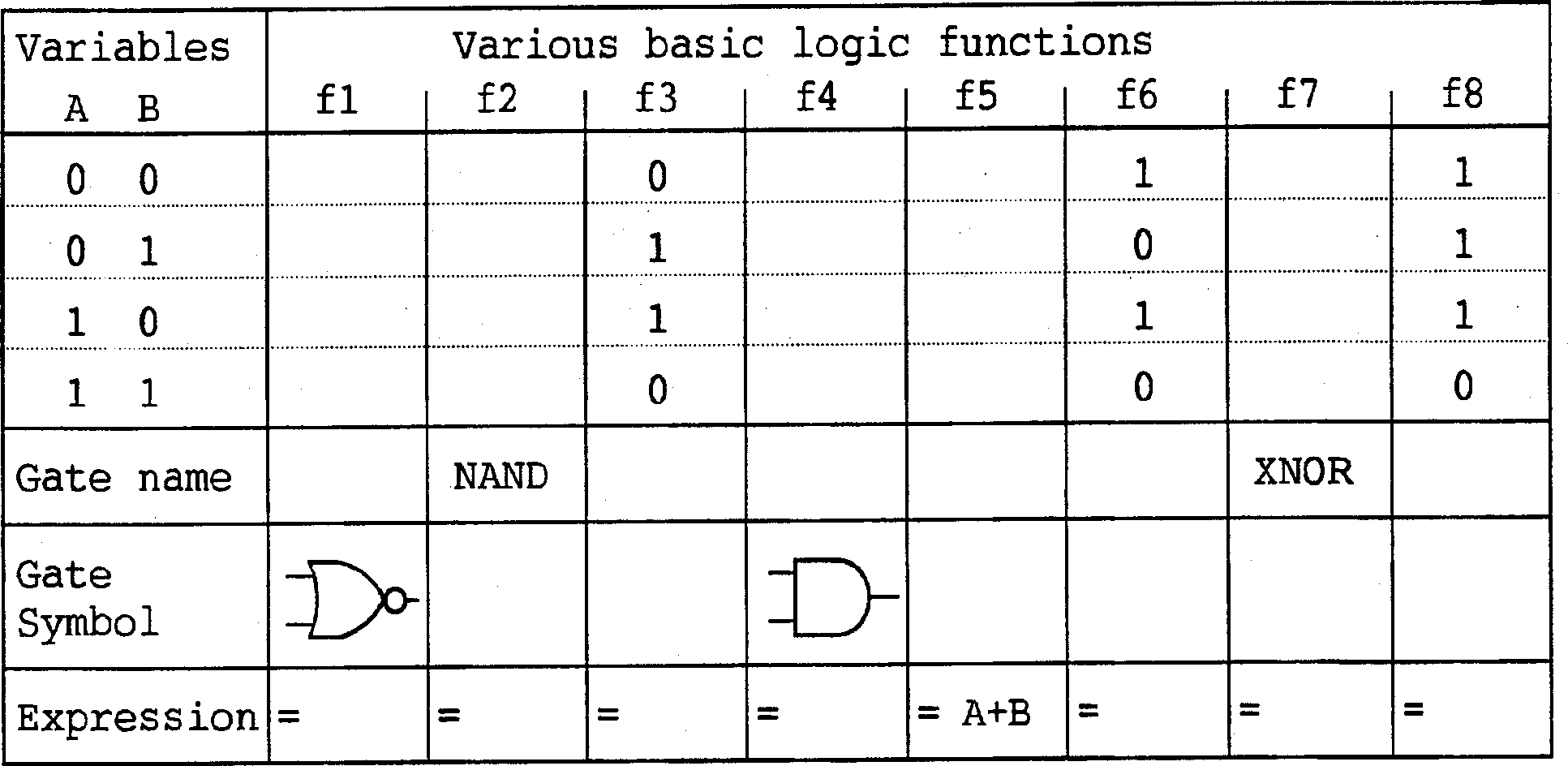
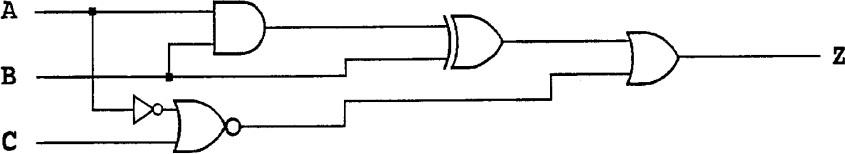
**4.A digital price display in a retail store uses BCD representation. The current displayed value is 0101. What is the equivalent decimal price?**

**Answer:**

In BCD (Binary Coded Decimal) representation, each decimal digit is encoded using four bits. To convert a BCD value to its equivalent decimal value, you can group the bits into sets of four and convert each set to its decimal representation.

Given BCD value: 0101 The displayed price is $5.

**LOGIC GATES**

1. Complete the following table in the format indicated
2. Give a Boolean expression that corresponds to this logic circuit:
3. A bank vault has three locks with a different key for each lock. Each key is owned by different person. In order to open the door, at least two people must insert their keys into the assigned locks. The signal lines A, B and C are 1 if there is a key inserted into lock 1,2 or 3 respectively. Write an equation for the variable Z which is 1 if the door should open.

**Number Conversions**

1. **Convert the Octal number 482.405 to its binary, decimal and hexadecimal equivalent using the iterative method. Show all the steps.**

# Step 1: Convert Octal to Binary

* For the whole number part: 4 \* 8^2 + 8 \* 8^1 + 2 \* 8^0 = 256 + 64 + 2 = 322
* For the fractional part: 0.405 \* 8 = 3.24 0.24 \* 8 = 1.92 0.92 \* 8 = 7.36 0.36 \* 8 =

2.88

* Reading the whole number and fractional parts together in binary: Whole: 322 (decimal) = 101000010 (binary) Fractional: 0.405 (decimal) = 0.011001 (binary)
* **Step 2: Convert Binary to Decimal Binary:** 101000010.011001
* For the whole number part: 1 \* 2^8 + 0 \* 2^7 + 1 \* 2^6 + 0 \* 2^5 + 0 \* 2^4 + 0 \* 2^3 + 0 \* 2^2 + 1 \* 2^1 + 0 \* 2^0 = 256 + 64 + 2 = 322
* For the fractional part: 0 \* 2^-1 + 1 \* 2^-2 + 1 \* 2^-3 + 0 \* 2^-4 + 0 \* 2^-5 + 0 \* 2^-6 + 1 \* 2^-7 = 0.25 + 0.125 + 0.0078125 = 0.3828125
* Adding the whole number and fractional parts: 322 + 0.3828125 = 322.3828125

# Step 3: Convert Decimal to Hexadecimal Decimal: 322.3828125

* For the whole number part: Dividing 322 by 16 gives us a quotient of 20 and a remainder of 2.
* Dividing 20 by 16 gives us a quotient of 1 and a remainder of 4.
* Dividing 1 by 16 gives us a quotient of 0 and a remainder of 1.
* Reading the remainders from bottom to top, the whole number part in hexadecimal is 124.
* For the fractional part: Converting the fractional part to hexadecimal: 0.3828125 \* 16 = 6.124
* So, the fractional part in hexadecimal is 0.6.
* Combining the whole number and fractional parts in hexadecimal notation: 322.3828125 (decimal) = 124.6 (hexadecimal).

# Convert the hexadecimal number D4C.82C to its binary, decimal and octal equivalent using the iterative method. Show all the steps.

* **Step 1: Convert Hexadecimal to Binary Hexadecimal: D4C.82C**
* Converting the whole number part (D4C) to binary: D = 13 (decimal) = 1101 (binary) 4 = 4 (decimal) = 0100 (binary) C = 12 (decimal) = 1100 (binary)
* So, the whole number part in binary is 110101001100.
* Converting the fractional part (82C) to binary: 8 = 8 (decimal) = 1000 (binary) 2

= 2 (decimal) = 0010 (binary) C = 12 (decimal) = 1100 (binary)

* So, the fractional part in binary is 100000101100.
* Combining the whole number and fractional parts: D4C.82C (hexadecimal) = 110101001100.100000101100 (binary).
* **Step 2: Convert Binary to Decimal Binary:** 110101001100.100000101100
* For the whole number part: 1 \* 2^11 + 1 \* 2^10 + 0 \* 2^9 + 1 \* 2^8 + 0 \* 2^7 + 1

\* 2^6 + 0 \* 2^5 + 0 \* 2^4 + 1 \* 2^3 + 1 \* 2^2 + 0 \* 2^1 + 0 \* 2^0 = 2048 + 1024

+ 256 + 64 + 32 + 8 + 4 = 4456

* For the fractional part: 1 \* 2^-1 + 0 \* 2^-2 + 0 \* 2^-3 + 0 \* 2^-4 + 0 \* 2^-5 + 0 \* 2^-6 + 1 \* 2^-7 + 0 \* 2^-8 + 1 \* 2^-9 + 1 \* 2^-10 + 0 \* 2^-11 = 0.5 + 0.0078125

+ 0.0009765625 = 0.5087890625

* Adding the whole number and fractional parts: 4456 + 0.5087890625 ≈ 4456.5087890625
* So, the binary number 110101001100.100000101100 is approximately equal to the decimal number 4456.5087890625.
* **Step 3: Convert Decimal to Octal Decimal:** 4456.5087890625
* For the whole number part: Dividing 4456 by 8 gives us a quotient of 557 and a remainder of 0. Dividing 557 by 8 gives us a quotient of 69 and a remainder of 5. Dividing 69 by 8 gives us a quotient of 8 and a remainder of 5. Dividing 8 by 8 gives us a quotient of 1 and a remainder of 0. Dividing 1 by 8 gives us a quotient of 0 and a remainder of 1.
* Reading the remainders from bottom to top, the whole number part in octal is 10550.
* For the fractional part: Converting the fractional part to octal: 0.5087890625 \* 8 = 4.0703125
* So, the fractional part in octal is 0.4.
* Combining the whole number and fractional parts in octal notation: 4456.5087890625 (decimal) = 10550.4 (octal).

# Convert the following number to its equivalent decimal, binary, octal, and hexadecimal representations. Show your work step by step for each conversion.

**i) (140.03)8 = (?)2**

* Converting each octal digit to binary: 1 (octal) = 001 (binary) 4 (octal) = 100 (binary) 0 (octal) = 000 (binary) 0 (octal) = 000 (binary) 3 (octal) = 011 (binary)
* Combining the binary digits: (140.03)8 = 001 100 000. 000 011
* So, the octal number (140.03)8 is equal to the binary number 001100000.000011.

# ii) (279.8)2 = (?)10

* Converting the whole number part (279) to decimal: 2^8 + 72^7 + 92^6 = 256 + 896 + 576 = 1728
* Converting the fractional part (0.8) to decimal: 02^-1 + 82^-2 = 0 + 0.25 = 0.25
* Adding the whole number and fractional parts: 1728 + 0.25 = 1728.25
* So, the binary number (279.8)2 is equal to the decimal number 1728.25.

# iii) (28.20)10 = (?)8

* Converting the whole number part (28) to octal: 28 ÷ 8 = 3 remainder 4
* So, the whole number part in octal is 34.
* Converting the fractional part (0.20) to octal: 0.20 \* 8 = 1.6
* So, the fractional part in octal is 0.1.
* Combining the whole number and fractional parts: (28.20)10 = (34.1)8
* Therefore, the decimal number (28.20)10 is equal to the octal number (34.1)8.

# iv) (5364.23)8 = (?)16

* Converting the whole number part (5364) to hexadecimal: 6 (octal) = 6 (hexadecimal) 4 (octal) = 4 (hexadecimal) 5 (octal) = 5 (hexadecimal) 3 (octal) = 3 (hexadecimal)
* So, the whole number part in hexadecimal is 6534.
* Converting the fractional part (0.23) to hexadecimal: 0.23 \* 16 = 3.68 0.68 \* 16 =

10.88 (convert 10 to A in hexadecimal) 0.88 \* 16 = 14.08 (convert 14 to E in hexadecimal)

* So, the fractional part in hexadecimal is 0.AE.
* Combining the whole number and fractional parts: (5364.23)8 = (6534.0AE)16

# v) (1BC.AB)16 = (?)10

* Hexadecimal: (1BC.AB)16
* Converting the whole number part (1BC) to decimal: 1 \* 16^2 + 11 \* 16^1 + 12 \* 16^0 = 256 + 176 + 12 = 444
* Converting the fractional part (0.AB) to decimal: 10 \* 16^-1 + 11 \* 16^-2 = 10/16

+ 11/256 = 0.625 + 0.04296875 = 0.66796875

* Adding the whole number and fractional parts: 444 + 0.66796875 ≈ 444.66796875
* So, the hexadecimal number (1BC.AB)16 is approximately equal to the decimal number 444.66796875