Experiment Details

|  |  |
| --- | --- |
| Department Name | Computer Science and Engineering |
| Class | S.Y.BTech |
| Semester | 1 |
| Subject Name | Digital Logic Design and Microprocessor Lab |
| Experiment No. | 1 |
| Experiment Name | Study of Basic Gates, Flip-Flops and Number Systems |

Version History

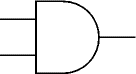
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr. No. | Version Number | Created By | Approved By | Date |
| 1 | v1.0 | Jay Deelip Kamble | Prof. Shivani Kale Ma’am | 17/10/2020 |
|  |  |  |  |  |

AIM:

Obtaining knowledge of basic electronic components like logic gates, flip-flops and registers, also revising various number systems for better understanding of any digital circuit or device and hence microprocessors.

THEORY:

An AND gate can have two or more inputs, its output is true if all inputs are true. The output Q is true if input A AND input B are both true: Q = A AND B

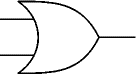


A

Q

B

An OR gate can have two or more inputs, its output is true if at least one input is true. The output Q is true if input A OR input B is true (or both of them are true): Q = A OR B



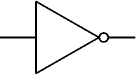
A

Q

B

A NOT gate can only have one input and the output is the inverse of the input. A NOT gate is also called an inverter.

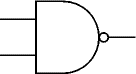
The output Q is true when the input A is NOT true: Q = NOT A



Q

A

NAND = Not AND. This is an AND gate with the output inverted, as shown by the 'o' on the symbol output. A NAND gate can have two or more inputs, its output is true if NOT all inputs are true. The output Q is true if input A AND input B are NOT both true: Q = NOT (A AND B)

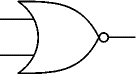


A

B

Q

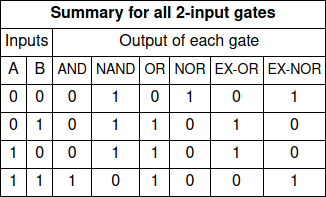
NOR = Not OR. This is an OR gate with the output inverted, as shown by the 'o' on the symbol output. A NOR gate can have two or more inputs, its output is true if no inputs are true. The output Q is true if NOT inputs A OR B are true: Q = NOT (A OR B)



A

B

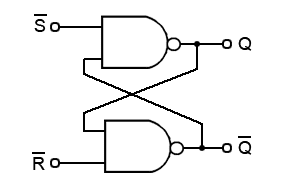
Q

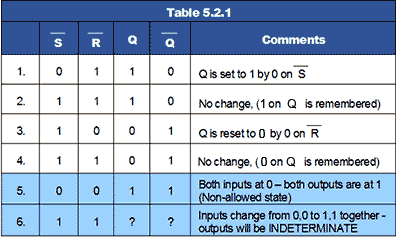


**SR Flip Flop:-**

The SR (Set-Reset) flip-flop is one of the simplest sequential circuits and consists of two gates connected as shown in Fig. Notice that the output of each gate is connected to one of the inputs of the other gate, giving a form of positive feedback or ‘cross-coupling’.

The circuit has two active low inputs marked S and R, ‘NOT’ being indicated by the bar above the letter, as well as two outputs, Q and Q.



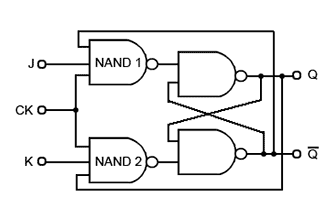


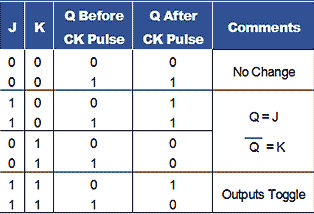
]

**JK Flip Flop:-**

The JK Flip Flop is the most widely used flip flop. It is considered to be a universal flip-flop circuit. The sequential operation of the JK Flip Flop is the same as for the RS flip-flop with the same SET and RESET input.

The difference is that the JK Flip Flop does not the invalid input states of the RS Latch (when S and R are both 1). The JK Flip Flop name has been kept on the inventor name of the circuit known as Jack Kilby.





**Number Systems:-**

**Decimal Number Systems:-**

The number system that we use in our day-to-day life is the decimal number system. Decimal number system has base 10 as it uses 10 digits from 0 to 9. In decimal number system, the successive positions to the left of the decimal point represent units, tens, hundreds, thousands, and so on.

Each position represents a specific power of the base (10). For example, the decimal number 1234 consists of the digit 4 in the units position, 3 in the tens position, 2 in the hundreds position, and 1 in the thousands position. Its value can be written as

(1 x 1000)+ (2 x 100)+ (3 x 10)+ (4 x l)

(1 x 103)+ (2 x 102)+ (3 x 101)+ (4 x l00)

1000 + 200 + 30 + 4

1234

**Binary Number System:-**

1. Uses two digits, 0 and 1
2. Also called as base 2 number system
3. Each position in a binary number represents a **0** power of the base (2). Example 20
4. Last position in a binary number represents a **x** power of the base (2). Example 2x where **x** represents the last position – 1.

### Example

Binary Number: 101012

Calculating Decimal Equivalent −

|  |  |  |
| --- | --- | --- |
| Step | Binary Number | Decimal Number |
| Step 1 | 101012 | ((1 x 24) + (0 x 23) + (1 x 22) + (0 x 21) + (1 x 20))10 |
| Step 2 | 101012 | (16 + 0 + 4 + 0 + 1)10 |
| Step 3 | 101012 | 2110 |

**Octal Number System:-**

1. Uses eight digits, 0,1,2,3,4,5,6,7
2. Also called as base 8 number system
3. Each position in an octal number represents a **0** power of the base (8). Example 80
4. Last position in an octal number represents a **x** power of the base (8). Example 8x where **x** represents the last position - 1

Example

Octal Number: 125708

Calculating Decimal Equivalent −

|  |  |  |
| --- | --- | --- |
| Step | Octal Number | Decimal Number |
| Step 1 | 125708 | ((1 x 84) + (2 x 83) + (5 x 82) + (7 x 81) + (0 x 80))10 |
| Step 2 | 125708 | (4096 + 1024 + 320 + 56 + 0)10 |
| Step 3 | 125708 | 549610 |

**Hexadecimal Number System:-**

1. Uses 10 digits and 6 letters, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
2. Letters represent the numbers starting from 10. A = 10. B = 11, C = 12, D = 13, E = 14, F = 15
3. Also called as base 16 number system
4. Each position in a hexadecimal number represents a **0** power of the base (16). Example, 160
5. Last position in a hexadecimal number represents a **x** power of the base (16). Example 16x where **x** represents the last position - 1

### Example

Hexadecimal Number: 19FDE16

Calculating Decimal Equivalent −

|  |  |  |
| --- | --- | --- |
| Step | Binary Number | Decimal Number |
| Step 1 | 19FDE16 | ((1 x 164) + (9 x 163) + (F x 162) + (D x 161) + (E x 160))10 |
| Step 2 | 19FDE16 | ((1 x 164) + (9 x 163) + (15 x 162) + (13 x 161) + (14 x 160))10 |
| Step 3 | 19FDE16 | (65536+ 36864 + 3840 + 208 + 14)10 |
| Step 4 | 19FDE16 | 10646210 |

BCD:-

1. It is a form of binary encoding where each digit in a decimal number is represented in the form of bits.
2. This encoding can be done in either 4-bit or 8-bit (usually 4-bit is preferred).
3. It is a fast and efficient system that converts the decimal numbers into binary numbers as compared to the existing binary system.
4. These are generally used in digital displays where is the manipulation of data is quite a task.
5. Thus BCD plays an important role here because the manipulation is done treating each digit as a separate single sub-circuit.

Example:-

Convert (123)10 in BCD

From the truth table above,  
 1 -> 0001  
 2 -> 0010  
 3 -> 0011  
 thus, BCD becomes -> 0001 0010 0011

PRE TEST:

**A binary value can be something other than 0 or 1.**

1. True
2. **False**

**Which number system is used in a day to day life?**

1. Binary
2. **Decimal**
3. Octal
4. Hexadecimal

**What does BCD stands for?**

1. Big complex digits
2. Binary coupled digits
3. **Binary coded decimals**

**What is the radix of decimal number system?**

1. **10**
2. 2
3. 9
4. 8

**Which alphabet is not included in Hexadecimal number system?**

1. E
2. A
3. B
4. **M**

PROCEDURE:

* With the help of kit and ICs, students should observe working of basic logic gates (AND, OR, NOT) and universal gates (NAND, NOR) and check the result against truth table.
* Students should construct S-R / J-K flip-flop using appropriate logic gates.
* Students should write method / rules to perform arithmetic operations for following number systems (with examples)

1. Binary
2. BCD
3. Hex
4. Octal

POST TEST:

**For what input will the output of NOR gate be high?**

1. **00**
2. 11
3. 01
4. 10

**Which of the following are universal logic gates?**

1. OR
2. AND
3. **NOR**
4. XOR

**What is the number of AND gates required to prove Y = CD + EF + G?**

1. 4
2. **2**
3. 3
4. 5

**What is the value for the radix in binary number system?**

1. 10
2. 8
3. **2**
4. 1

**Convert the following binary number to decimal :- 010112**

1. 35
2. 15
3. **11**
4. 13

**What is a single digit maximum value in octal number system?**

1. 6
2. **7**
3. 8
4. 9

REFERENCES:

https://beginnersbook.com/2014/01/c-pointers/