

| **Batch: Roll No.: Experiment / assignment / tutorial No.: 1** |
| --- |

| **Title:** Basic Gates & Universal Gates |
| --- |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Objective:** To study the basic gates: AND, OR, NOT and universal gates: NAND, NOR, XOR, XNOR

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Expected Outcome of Experiment:**

**CO1:** Recall basic gates and binary, octal & hexadecimal calculations and conversions.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Books/ Journals/ Websites referred:**

* Vlab Link: [http://vlabs.iitkgp.ernet.in/dec/#](http://vlabs.iitkgp.ernet.in/dec/)
* R. P. Jain, “Modern Digital Electronics”, Tata McGraw Hill
* http://www.ee.surrey.ac.uk/Projects/Labview/gatesfunc/
* http://www.electronics-tutorials.ws/boolean/bool\_6.html

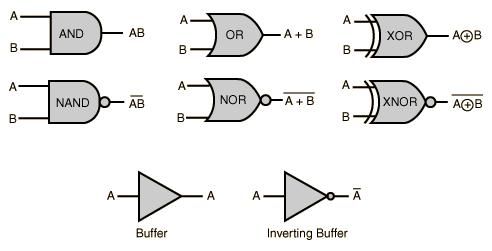
**Pre Lab/ Prior Concepts:**

Gate is a logic circuit with one or more inputs but only one output. Gates are digital (two state) circuit because the input & output are either low or high. Gates provide high output for certain combinations of input & for other combinations the output is low. Total number of combinations for a gate is 2^n; where n is number of input.

**Classification:** The two types of gate are:

1. **Basic or Fundamental Gates:**
2. **Derived Gates:**

**Symbols of gates**



| **Type of IC** | **Specification** |
| --- | --- |

**Implementation Details:**

**Basic Gates**

1. **OR gate:** The OR gate has two or more inputs but only 1 output. If any or all the inputs are high, the output is high. If all the inputs are low, the output is low.

Y=

**Symbol for OR gate** **Pin Diagram For IC 7432**

The truth table for OR operations are:

1. **AND gate:** The AND gate has two or more inputs but only one output. If any or all inputs are high then output is also high



Y=

**Symbol for AND gate** **Pin Diagram For IC 7408**

The truth table for AND operations are:

1. **NOT gate:** The Not gate is a gate with only one input and one output. The output is always in opposite state of an input. A NOT gate is also called as Inverter because it performs inversion.

Y=



**Symbol for NOT gate** **Pin Diagram For IC 7404**

The truth table for NOT operations is:

**Derived Gates/Universal Gates**

* + NAND gate
  + NOR gate
  + EX-OR gate
  + EX-NOR gate

1. **NAND gate:** This is a NOT-AND gate which is equal to an AND gate followed by a NOT gate. The outputs of all NAND gates are high if any of the inputs are low. The symbol is an AND gate with a small circle on the output. The small circle represents inversion.



Y=

**Symbol** **Pin Diagram for IC 7400**

The truth table for NAND operations is:

1. **NOR gate:** This is a NOT-OR gate which is equal to an OR gate followed by a NOT gate. The outputs of all NOR gates are low if any of the inputs are high. The symbol is an OR gate with a small circle on the output. The small circle represents inversion.

Y=

**Symbol for NOR gate** **Pin Diagram For IC 7402**

The truth table for NOR operations are:



1. **EX-OR gate**: The 'Exclusive-OR' gate is a circuit which will give a high output if either, but not both, of its two inputs are high. An encircled plus sign ( ) is used to show the EX-OR operation

Y= 

**Symbol for Ex-OR gate** **Pin Diagram For IC 7486**

The truth table for XOR operations is:



1. **EX-NOR gate**: The 'Exclusive-NOR' gate circuit does the opposite to the EOR gate. It will give a low output if either, but not both, of its two inputs are high. The symbol is an EXOR gate with a small circle on the output. The small circle represents inversion

Y=

**Symbol for Ex-NOR gate** 

The truth table for XNOR operations is:

**Implementation Using NAND Gate**

**NOT GATE STEPS**

**AND GATE**

**OR GATE**

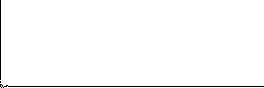
**IMPLEMENTATION USING NOR GATE**

**NOT GATE STEPS**



**AND GATE**

**OR GATE**



**Conclusion:**

**Post Lab Descriptive Questions**

1. Verify the expression (A∙B)' + C by:
2. Using NAND Gate directly.
3. Using AND & NOT gate consecutively.
4. Implement the following expressions using a combination of gates:
5. (A'+B)∙B
6. (A∙B)+A'
7. A∙ (B∙B')
8. (A'⊕B)∙A



| B**atch: Roll No.: Experiment / assignment / tutorial No.: 2** |
| --- |

| **Title:** Binary Adders and Subtractors |
| --- |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Objective:** To implement half and full adder–subtractor using gates and IC 7483

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**\_\_\_\_\_\_\_\_

**Expected Outcome of Experiment:**

**CO2:** Use different minimization techniques and solve combinational circuits, synchronous & asynchronous sequential circuits.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Books/ Journals/ Websites referred:**

* **VLab Link:** <http://vlabs.iitb.ac.in/vlabs-dev/labs/dldesignlab/experimentlist.html>
* R. P. Jain, “Modern Digital Electronics”, Tata McGraw Hill
* M .Morris Mano, “Digital Logic & computer Design”, PHI
* http://physics.niser.ac.in/labmanuals/sem5/elect/7\_ADDER%20SUBTRACTO  [R%20CIRCUITS.pd](http://physics.niser.ac.in/labmanuals/sem5/elect/7_ADDER%20SUBTRACTOR%20CIRCUITS.pdf)f

**Pre Lab/ Prior Concepts:**

**Adder:** Addition of two binary digits is most basic operation performed by the digital computer. There are two types of adder:

* Half adder
* Full adder

**Half Adder:** Half adder is a combinational logic circuit with two inputs and two outputs. It is the basic building block for the addition of two single-bit numbers.

**Full adder:** A half adder has a provision not to add a carry coming from the lower order bits when multi bit addition is performed. for this purpose a third input terminal is added and this circuits is to add A,B,C where A and B are the nth order bits of the number A and B respectively and C is the carry generated from the addition of (n-1) order bits. This circuit is referred to as full adder.



**Subtractor:** Subtraction of two binary digits is one of the most basic operations performed by digital computer .there are two types of subtractor:

* Half subtractor
* Full subtractor

**Half subtractor:** Logic circuit for the subtraction of B from A where A,B are 1 bit numbers is referred to as half subtract or .the subtract or process has two input and difference and borrow are the two outputs.

**Full subtractor:** As in the case of the addition using logic gates, a full subtractor is made by combining two half-sub tractors and an additional OR-gate. A full subtractor has the borrow in capability (denoted as BORIN) and so allows cascading which results in the possibility of multi-bit subtraction.

**IC 7483**

For subtraction of one binary number from another, we do so by adding 2’s complement of the former to the latter number using a full adder circuit.

IC 7483 is a 16 pin, 4-bit full adder. This IC has a provision to add the carry output to transfer and end around carry output using Co and C4 respectively.

**2’s complement:** 2’s complement of any binary no. can be obtained by adding 1 in 1’scomplement of that no.

e.g. 2’s complement of +(10)10 =1010is

| 1C of 1010 |  | | 0101 |
| --- | --- | --- | --- |
|  |  | + | 1 |
| -(10)10 |  | | 0110 |

In 2’s complement subtraction using IC 7483, we are representing negative number in 2’s complement form and then adding it with 1st number.

**Implementation Details:**

**Half Adder Block Diagram**

**Half Adder Circuit**

**Truth Table for Half Adder**

| **Inputs** | | **Outputs** | |
| --- | --- | --- | --- |
| **A** | **B** | **A** | **B** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**From the truth table (with steps):**

**Full Adder Block Diagram**

**Full Adder Circuit**

**Truth Table for Full Adder**

**From the truth table (with steps):**



**Half Subtractor Block Diagram**

**Half Subtractor Circuit**

**Truth Table for Half Subtractor**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **DIFFERENCE(D)** | **BORROW(Bo)** |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**From the truth table (with steps) :**

**Full Subtractor Block Diagram**

**Full Subtractor Circuit**

**Truth Table for Full subtractor**

| **A** | **B** | **BIN** | **D** | **BOROUT** |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |



**From the truth table (with steps):**

**IC 7483**

**Procedure:**

1. Locate the IC 7483 and 4-not gates block on trainer kit.
2. Connect 1st input no. to A4-A1 input slot and 2nd (negative) no. to B4-B1 through 4-not gates (1C of 2nd no.)
3. Connect high input to Co so that it will get added with 1C of 2nd no. to get 2C.
4. Connect 4-bit output to the output indicators.
5. Switch ON the power supply and monitor the output for various input combinations.

**Example:**

| 1) 710 -210 = 510 | |  |
| --- | --- | --- |
| 7 |  | 0111 |
| 2 |  | 0010 |
| 1’C of 2 | | 1101 |
|  |  | + 1 |
| 2’C of 2 | | 1110 |

0111 + 1110 1 0101

**Pin Diagram IC7483**

**Adder**

**Subtractor**

**Conclusion:**

**Post Lab Descriptive Questions**

1. What is difference between half and full adder, half and full subtractor?
2. Perform the following Binary subtraction with the help of appropriate ICs:
3. 7-5
4. 5-7
5. 9-4







| **Batch: Roll No.: Experiment / assignment / tutorial No.: 3** |
| --- |

| **Title:** Design 4:1 Multiplexer and 3: 8 Decoder |
| --- |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Objective:** To design and implement a 4:1 multiplexer and 3: 8 Decoder using logic gates and MUX IC

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Expected Outcome of Experiment:**

**CO2:** Use different minimization technique and solve combinational circuits, synchronous & asynchronous sequential circuits.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Books/ Journals/ Websites referred:**

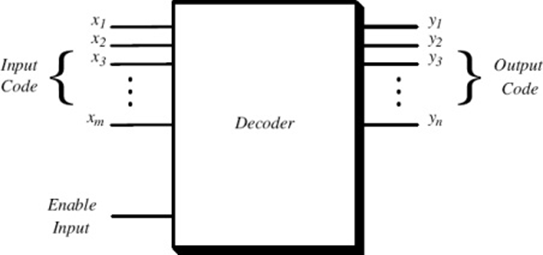
* VLab Links: <http://vlabs.iitb.ac.in/vlabs-dev/labs/dldesignlab/experimentlist.html>
* R. P. Jain, “Modern Digital Electronics”, Tata McGraw Hill
* M .Morris Mano, “Digital Logic & computer Design”, PHI
* [https://wiki.engr.illinois.edu/download/attachments/84770821/08](https://wiki.engr.illinois.edu/download/attachments/84770821/08-Multiplexers.pdf?version=2&modificationDate=1285128827000)- [Multiplexers.pdf?version=2&modificationDate=128512882700](https://wiki.engr.illinois.edu/download/attachments/84770821/08-Multiplexers.pdf?version=2&modificationDate=1285128827000)0

**Pre Lab/ Prior Concepts:**

**Multiplexer:** Multiplexer is a special type of combinational circuit. It is a digital circuitwhich selects one of the n data inputs and routes it to the output. The selection of one of the n inputs is done by the select lines. To select n inputs we require m select lines, such that 2m=n. Depending on the digital code applied at the select inputs, one out of the n data sources is selected and transmitted to a single output . E is called as the strobe or enable input which is useful for cascading. It is generally on active low terminal that means it will perform the required operation when it is low. The multiplexer act like a digitally controlled single pole, multiple way switches. The output gets connected to only one input at a time. In most of the electronic system the digital data is available on more than one line. It is necessary to route the data over a single line, under such circumstances input at a time

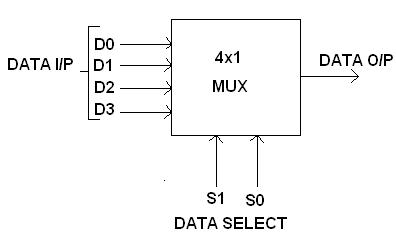
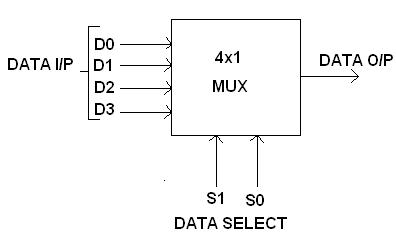
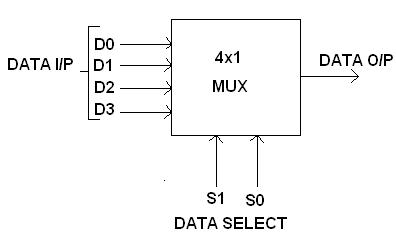
**Types of Multiplexer:**

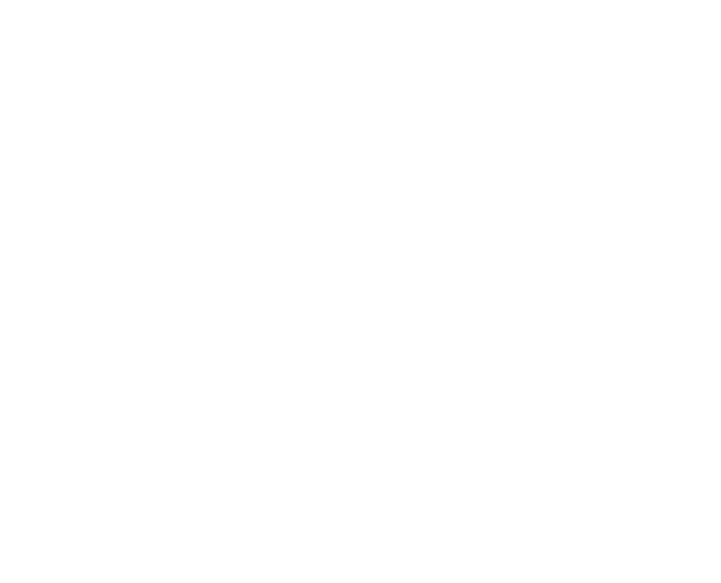
1. 2:1 Multiplexer
2. 4:1 Multiplexer
3. 8:1 Multiplexer
4. 16:1 Multiplexer
5. 32:1 Multiplexer

**Decoder:** A decoder is a multiple-input, multiple-output logic circuit that converts coded inputs into coded outputs, where the input and output codes are different. The input code generally has fewer bits than the output code, and there is a one-to-one mapping from input code words into output code words. The general structure of a decoder circuit is shown in the Figure below. The enable inputs, if present, must be asserted for the decoder to perform its normal mapping function. The most commonly used input code is an N-bit binary code, where an N-bit word represents one of 2N different coded values. Normally, they range from 0 through 2N − 1. The input code lines select which output is active. The remaining output lines are disabled. Thus, the decoder is intended to provide a binary code to other circuits, such as a memory circuit. In this case, the decoder is referred to as an address decoder because it selects one address of a memory location. However, a decoder could also be used to channel a stream of data on a designated output line selected by the input code lines.

**Implementation Details of 4:1 MUX**

**Block Diagram of 4:1 MUX**



**Circuit Diagram of 4:1 MUX**

**Truth table**

| **S1** | **S0** | **Y** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**From Truth Table:**

Y =

**Implementation Details of 8:1 MUX**

**Circuit Diagram of 8:1 MUX**

**Truth Table for 8:1 Multiplexer**

**From Truth Table:**

**Y =**

**Pin diagram: IC 74151**

**Block Diagram of 3:8 decoder**

**Circuit Diagram of 3:8 decoder**

**Truth Table for 3:8 decoder**

**From Truth Table:**

**Pin diagram: IC 74138**

**Conclusion:**

**Post Lab Descriptive Questions**

1. How many select lines are required for 64:1 MUX?
2. State some applications of MUX and decoders.
3. Build a 4:1 MUX using only 2:1 MUX.



| **Batch: Roll No.: Experiment / assignment / tutorial No.: 4** |
| --- |

| **Title:** 4 bit Magnitude Comparator |
| --- |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Objective:** Design a 2-bit comparator using logic gates and verify 4-bit magnitudecomparator using IC 7485

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Expected Outcome of Experiment:**

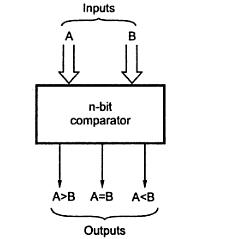
**CO2:** Use different minimization technique and solve combinational circuits, synchronous & asynchronous sequential circuits.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Books/ Journals/ Websites referred:**

* VLab Link: <http://vlabs.iitb.ac.in/vlabs-dev/labs/dldesignlab/experimentlist.html>
* R. P. Jain, “Modern Digital Electronics”, Tata McGraw Hill
* M .Morris Mano, “Digital Logic & computer Design”, PHI
* http://elnsite.teilam.gr/ebooks/digital\_design/lab/dataSheets\_page/7485.pdf

**Pre Lab/ Prior Concepts:**

The comparison of two numbers is an operator that determines one number is greater than, less than (or) equal to the other number. A magnitude comparator is a combinational circuit that compares two numbers A and B and determines their relative magnitude. The outcome of the comparator is specified by three binary variables that indicate whether A>B, A=B (or) A<B.



**Two Bit Magnitude Comparator Implementation Details:**

**Truth Table**

| **A1** | **A0** | **B1** | **B0** | **A > B** | **A = B** | **A < B** |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

**From the Truth Table:**

**(A<B)=**

**(A=B)=**

**(A>B)=**

**Logic Diagram of 2 bit Comparator**

**Four Bit Magnitude Comparator Implementation Details**

**Pin Diagram of IC 7485**



**Logic Diagram of IC 7485**

**Comparing Table**

**Conclusion:**

**Post-Lab Descriptive Questions**

1. Design a 1-bit magnitude comparator using logic gates.



| **Batch: Roll No.: Experiment / assignment / tutorial No.: 5** |
| --- |

| **Title:** Flip Flops |
| --- |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Objective:**Design of JK Flip flop, D flip flop, T flip flop using NAND Gates & verification of the same flip flop using IC7476

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Expected Outcome of Experiment:**

**CO2:** Use different minimization technique and solve combinational circuits, synchronous & asynchronous sequential circuits.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Books/ Journals/ Websites referred:**

* VLab Link: [http://vlabs.iitkgp.ernet.in/dec/#](http://vlabs.iitkgp.ernet.in/dec/)
* R. P. Jain, “Modern Digital Electronics”, Tata McGraw Hill
* M .Morris Mano, “Digital Logic & computer Design”, PHI
* A.P.Godse, D.A.Godse, “Digital Logic Design”

**Pre Lab/ Prior Concepts:**

Flip-flop is the common name given to two-state devices which offer basic memory for sequential logic operations. Flip-flops are heavily used for digital data storage and transfer and are commonly used in banks called "registers" for the storage of binary numerical data.

**JK-flip flop:** has two inputs, traditionally labeled J and K. IC 7476 is a dual JK master slave flip flop with preset and clear inputs. If J and K are different then the output Q takes the value of J at the next clock edge. If J and K are both low then no change occurs. If J and K are both high at the clock edge, then the output will toggle from one state to the other. It can perform the functions of the set/reset flip-flop and has the advantage that there are no ambiguous states.

**D Flip Flop:** tracks the input, making transitions with match those of the input D. The D stands for "data"; this flip-flop stores the value that is on the data line. It can be thought of as a basic memory cell. D flip-flop can be made from J-K flip-flop by connecting both inputs through a not gate.

**T Flip Flop:** T or "toggle" flip-flop changes its output on each clock edge, giving an output which is half the frequency of the signal to the T input. It is useful for constructing binary counters, frequency dividers, and general binary addition devices. It can be made from a J-K flip-flop by tying both of its inputs high.

**Implementation Details:**

**Procedure**

1. Locate IC 7476 on Digital trainer kit
2. Apply various inputs to J & K pins by means of the output on logic output indicator.
3. Connect a pulsar switch to the clock input.
4. Connect the J&K as D and T flip flop as shown in diagrams and verify the respective truth tables.

**Pin Diagram of IC 7476 JK Master- Slave FF**

**Logic Symbol Truth Table**

| **J** | **K** | **Q** | **Q’** | **Qn+1** | **Qn+1’**  **n+1** |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**JKFF**

**D FF Truth Table**

| **D** | **O/P** |
| --- | --- |
|  |  |
|  |  |

**TFF Truth Table**

| **T** | **O/P** |
| --- | --- |
|  |  |
|  |  |

**Diagram of JK Flip Flop using NAND gates**

**Conclusion:**

**Post Lab Descriptive Questions**

1. How does a JK flip-flop differ from an SR flip-flop in its basic operation?
2. What is the use of characteristic and excitation table?
3. How many flip flops do you require storing the data 1101?
4. Convert JK flip flop to D and T flip flops.

.



| **Batch: Roll No.: Experiment / assignment / tutorial No.: 6** |
| --- |

| **Title:**  Shift Register |
| --- |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Objective:** To implement the SISO, SIPO, PISO, PIPO shift register using D flips flop

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Expected Outcome of Experiment:**

**CO2:** Use different minimization technique and solve combinational circuits, synchronous & asynchronous sequential circuits.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Books/ Journals/ Websites referred:**

* VLab Link: [http://vlabs.iitkgp.ernet.in/dec/#](http://vlabs.iitkgp.ernet.in/dec/)
* R. P. Jain, “Modern Digital Electronics”, Tata McGraw Hill
* M .Morris Mano, “Digital Logic & computer Design”, PHI
* A.P.Godse, D.A.Godse, “Digital Logic Design”

**Pre Lab/ Prior Concepts:**

A register is capable of shifting its binary information in one or both directions is known as shift register. The logical configuration of shift register consist of a D-Flip flop cascaded with output of one flip flop connected to input of next flip flop. All flip flops receive common clock pulses which causes the shift in the output of the flip flop.The simplest possible shift register is one that uses only flip flop. The output of a given flip flop is connected to the input of next flip flop of the register. Each clock pulse shifts the content of register one bit position to right.

The basic types of shift registers are

* Serial In - Serial Out
* Serial In - Parallel Out
* Parallel In - Serial Out
* Parallel In - Parallel Out
* Bidirectional shift registers.

**Implementation Details:**

**Logic Diagram**

**Serial in Serial Out**

**Truth table**

**Serial In - Parallel Out**

**Truth table**

**Parallel In Serial Out**

**Truth table**

**Parallel In Parallel Out**

**Truth table**

**Conclusion:**

**Post Lab Descriptive Questions**

1. What is a universal shift register?Which MSI TTL IC is used as a Universal Shift Register?Attach its pin diagram and Truth table from the data sheet.
2. Prepare a truth table for 3 bit SISO left shift with data(- - - ) along with clock pulse
3. Can a shift register be used as a counter? Give any one application.
4. How many clock pulses are required to enter a byte of data serially into an 8-bit shift register?



| **Batch: Roll No.: Experiment / assignment / tutorial No.: 7** |
| --- |

| **Title:** |
| --- |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Objective:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Expected Outcome of Experiment:**

**CO4:** Implement digital networks using VHDL.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Books/ Journals/ Websites referred:**

* ModelSim Software Link:

https://www.mentor.com/company/higher\_ed/modelsim-student-edition

* J. Bhasker, “VHDL Primer”, Pearson Education
* Douglas L. Perry, “VHDL Programming by Example”, Tata McGraw Hill
* http://esd.cs.ucr.edu/labs/tutorial/

**Pre Lab/ Prior Concepts:**

**VHDL Programming Structure**

**Conclusion:**

**Post Lab Descriptive Questions**

1. What are two types of HDL?



| **Batch: Roll No.: Experiment / assignment / tutorial No.: 8** |
| --- |

| **Title:** 4:1 Mux in VHDL |
| --- |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Objective:** Design of 3 bit asynchronous counter using JK flip flop in VHDL

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Expected Outcome of Experiment:**

**CO2:** Use different minimization technique and solve combinational circuits, synchronous & asynchronous sequential circuits.

**CO4:** Implement digital networks using VHDL

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Books/ Journals/ Websites referred:**

* VLab Links: <http://vlabs.iitb.ac.in/vlabs-dev/labs/dldesignlab/experimentlist.html>
* R. P. Jain, “Modern Digital Electronics”, Tata McGraw Hill
* M .Morris Mano, “Digital Logic & computer Design”, PHI
* [https://wiki.engr.illinois.edu/download/attachments/84770821/08](https://wiki.engr.illinois.edu/download/attachments/84770821/08-Multiplexers.pdf?version=2&modificationDate=1285128827000)- [Multiplexers.pdf?version=2&modificationDate=128512882700](https://wiki.engr.illinois.edu/download/attachments/84770821/08-Multiplexers.pdf?version=2&modificationDate=1285128827000)0

**Pre Lab/ Prior Concepts:**

**Multiplexer:** Multiplexer is a special type of combinational circuit. It is a digital circuitwhich selects one of the n data inputs and routes it to the output. The selection of one of the n inputs is done by the select lines. To select n inputs we require m select lines, such that 2m=n. Depending on the digital code applied at the select inputs, one out of the n data sources is selected and transmitted to a single output . E is called as the strobe or enable input which is useful for cascading. It is generally on active low terminal that means it will perform the required operation when it is low. The multiplexer act like a digitally controlled single pole, multiple way switches. The output gets connected to only one input at a time. In most of the electronic system the digital data is available on more than one line. It is necessary to route the data over a single line, under such circumstances input at a time

**Types of Multiplexer:**

1. 2:1 Multiplexer
2. 4:1 Multiplexer
3. 8:1 Multiplexer
4. 16:1 Multiplexer
5. 32:1 Multiplexer

**Implementation Details of 4:1 MUX**

**Conclusion:**

**Post Lab Descriptive Questions**

1. Application Mux?