**ASSIGNMENT # 1**

**Introduction to KL – 300 Digital Trainer**

**Abstract:**

To learn the all-in-one digital logic systematic device (i.e Digital Trainer) that constitutes power supply, signal generator, switches and displays to perform tests and experiments for developing a circuit prototype before producing

**Summary:**

The KL-300 Digital Logic Lab is widely used for combination logic, sequential logic and microprocessor circuits design and experiments. It is an ideal tool for developing a protype circuitry of different logics like TTL, CMOS, NMOS, PMOS or ECL. It provides 13 modules that are capable of carrying out the specified logical operations.

**Experimental Procedure:**

Simple Series and Parallel Resistor Circuitry or Transistor LED Turn ON/OFF circuitry can be designed to become familiar with this device initially.

**Lab Assignment:**

**Q1: Write advantages of sequential circuits over combinational circuits?**

1- **Sequential circuits** are dependent on clock-cycle and present as well as past inputsto generate any output whereas **combinational circuits** are time independentwhich do not depend upon previous inputs

**2- Sequential circuits** uses previous input, output, clock and a memory element whereas **combinational circuits** don’t use any memory.

**Example**: A flip-flop is a sequential circuit which samples the input and changes the output at a particular instance of time.

**Q2: Define DIP switch?**

**DIP (Dual In-line Package) switches** are the set of small switches commonly used to set hardware options on PCs and peripherals circuits. **DIP switches** allow you to make or break a single electrical connection

**Q3: What are Thumbwheel Switches and what functionality do Thumbwheels provide?**

A **Thumbwheel Switch** is a setting switch that converts the numeric value selected by turning a disk-shaped part on which numbers are written into a binary, decimal, hexadecimal, or other code using the combination of ON and OFF signals for multiple contact circuits and that outputs the resulting code.

**Discussion:** Digital Logic Labs are not only time and cost saving but also a manifesto for both students and engineers to test their future circuits by developing its prototypes.

**ASSIGNMENT # 2**

**Basic Logic Gates (AND, OR & NOT)**

**Abstract:**

To learn the basic logic gates that are the building block of digital logic designs. It clears the concepts regarding further DLD terminologies like Boolean Rule, Flip Flop, Half/Full Binary Adders etc.

**Summary:**

AND, OR & NOT gates are basically the built-in Transistor-Transistor Logic Integrated Circuits (TTL-ICs). These ICs are set off under Logic “0” or Logic “1”. AND & OR gates have Dual Inputs and Single Output whereas NOT gate has Uni-Input/Output. AND gate simply multiplies, OR gate adds and NOT gate inverts the logic passed.

**Experimental Procedure:**

LED can be used to examine output with all the three gates by making the logic conditions ON or OFF to the IC’s inputs.

**Data:**

7408 is the AND (multiplier) IC that uses the logic: **Both the inputs must be High for High output**

7432 is the OR (adder) IC that uses the logic: **Any of input can be High for High output**

7404 is NOT (inverter) IC that uses the logic: **An output is the Inverted form of its input**

**Truth Table Demonstration:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A | B | 7408 | 7432 | 7404 (A) | 7404 (B) |
| 0 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 |

**Discussion:**

In TTL IC’s there are 14 pins, the 14th pin is connected with High (positive terminal) and the seventh pin is connected with Low or Grounded (negative terminal). The remaining pins are used to carry out their specified funcationalities.

**ASSIGNMENT # 3**

**Illustration of Demorgan’s Theorem**

**Abstract:**

To learn the method of simplifying the Boolean expression that’s by Demorgan’s Theorem. The theorem verifies the Product of Sum and Sum of Product using Universal gates (i.e NAND and NOR).

**Summary:**

According to Demorgan’s theorem the large bar over several variables can be broken between variables if the sign between the variables is changed (+ to x or x to +). The 1st theorem exhibits that NAND Gate = Negative OR Gate while 2nd theorem exhibits NOR Gate = Negative OR Gate.

**Experimental Procedure:**

**Truth Tables:**

1. Verification for Demorgan’s First Theorem: **Bar (A.B) = Bar (A) + Bar (B)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A | B | Bar (A) | Bar(B) | A.B | Bar (A.B) | Bar(A) + Bar(B) |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 |

1. Verification for Demorgan’s Second Theorem: **Bar (A+B) = Bar (A) . Bar (B)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A | B | Bar (A) | Bar (B) | A+B | Bar (A+B) | Bar(A) . Bar(B) |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 |

**Discussion:**

Demorgan’s theorem help us to simplify the exclusive lengthy Boolean Expression into shorter form using Universal gates. This technique can also be applied on other expressions.

**ASSIGNMENT # 4**

**Verification and Interpretation of Truth Tables For NAND, NOR, XOR & XNOR Gates**

**Abstract:**

To learn the science behind NAND, NOR, XOR & XNOR gates and their truth tables. Also the universal gates can replace any kind of gate expression.

**Summary:**

AND, OR & NOT gates are also the built-in Transistor-Transistor Logic Integrated Circuits (TTL-ICs). These ICs are set off under Logic “0” or Logic “1”. All of them have Dual Inputs and Single Output. NAND gate simply multiplies and then inverts, NOR gate adds and then inverts, NOR gate adds the inverted inputs and XNOR gate inverts the XOR’s out (or adds the inverted input and inverts its output).

**Experimental Procedure:**

NAND and NOR gates are "universal" gates, and thus any Boolean function can be constructed using either NAND or NOR gates only.

**Q1: How to implement NOT, AND, and OR gate using NAND gates only?**

|  |  |  |
| --- | --- | --- |
| **Logic Gate** | **Symbol** | **Using NAND** |
|  |  |  |
|  |  |  |
|  |  |  |

**Truth Table Demonstration:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**Q2: How to implement NOT, AND, and OR gate using NOR gates only**

|  |  |  |
| --- | --- | --- |
| **Logic Gate** | **Symbol** | **Using NOR** |
|  |  |  |
|  |  |  |
|  |  |  |

**Discussion:**

NAND and NOR are called Universal gates because they can replace any gate expressions.

**ASSIGNMENT # 5**

**Binary Half Adder and Binary Full Adder**

**Abstract:**

To learn the binary digital airthmetic addition operation using digital logic circuits like XOR, Half Adder and Full Adder and generate it truth table.

**Summary:**

Half adder adds two binary digits and produces two binary outputs called sum and carry whereas Full Adder adds three binary digits and produces two binary outputs (Sum & Carry) using XOR, NAND gates.

**Experimental Procedure:**

**Truth Table Demonstration:**

**Q1: Construct the circuit on a breadboard with switches and 2 LEDs. Experimentally verify the truth table. A and B are switches. The output for each logic gate will be on an LED.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **Sum = A XOR B** | **Carry = A.B** | **LED** |
| **0** | **0** | **0** | **0** | **OFF** |
| **0** | **1** | **1** | **0** | **ON** |
| **1** | **0** | **1** | **0** | **ON** |
| **1** | **1** | **0** | **1** | **ON** |

**Q2: Construct the circuit on a breadboard with switches and 2 LEDs. Experimentally verify the truth table. A and B are switches. The output for each logic gate will be on an LED.**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **AB** | **AC** | **BC** | **A XOR B** | **Sum = (A XOR B) XOR C** | **Carry = AB + AC + BC** | **LED** |
| **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **OFF** |
| **0** | **0** | **1** | **0** | **0** | **0** | **0** | **1** | **0** | **ON** |
| **0** | **1** | **0** | **0** | **0** | **0** | **1** | **1** | **0** | **ON** |
| **0** | **1** | **1** | **0** | **0** | **1** | **1** | **0** | **1** | **ON** |
| **1** | **0** | **0** | **0** | **0** | **0** | **1** | **1** | **0** | **ON** |
| **1** | **0** | **1** | **0** | **1** | **0** | **1** | **0** | **1** | **ON** |
| **1** | **1** | **0** | **1** | **0** | **0** | **0** | **0** | **1** | **ON** |
| **1** | **1** | **1** | **1** | **1** | **1** | **0** | **1** | **1** | **ON** |

**Discussion:**

The **applications of adder** circuit are, **adder** circuits are not only used to add binary numbers, but also used in digital **applications** such as address, table index, decoding and calculation etc.