|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **28 MAY 2020** | **Name:** | **MANAVI** |
| **Course:** | **logic design** | **USN:** | **4AL18EC031** |
| **Topic:** | **Day 2:analysis of clocked sequential circuits**  **digital clock design** | **Semester & Section:** | **4TH SEM**  **& A SEC** |
| **Github Repository:** | **Manavi-test** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session** |
| **Report – Report can be typed or hand written for up to two pages.**  **ANALYSIS OF CLOCKED SEQUENTIAL CIRCUITS:-**   * Now that we have flip-flops and the concept of memory in our circuit, we might want   to determine what a circuit is doing.   * The behavior of a clocked sequential circuit is determined from its inputs, outputs   and state of the flip-flops (i.e., the output of the flip-flops).   * The analysis of a clocked sequential circuit consists of obtaining a table of a diagram   of the time sequences of inputs, outputs and states.   * E.g., given a current state and current inputs, how will the state and outputs   change when the next active clock edge arrives   * We have a basic procedure for analyzing a clocked sequential circuit: * Write down the equations for the outputs and the flip-flop inputs. * Using these equations, derive a state table which describes the next state. * Obtain a state diagram from the state table. * It is the state table and/or state diagram that specifies the behavior of the circuit.   **Notes:**   * The flip-flop input equations are sometimes called the excitation equations. * The state table is sometimes called a transition table. * We can best illustrate the procedure by doing examples…  |  |  |  |  | | --- | --- | --- | --- | | **DATE:** | **28 MAY 2020** | **NAME:** | **MANAVI** | | **COURSE:** | **PYTHON** | **USN:** | **4AL18EC031** | | **TOPIC:** | **DAY 10:object oriented programming.** | **SEMESTER & SECTION:** | **4TH SEM & A SEC** | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | |
|  |  |  |  | |
|  |  |  |  | |
|  |  |  |  | |
| AFTERNOON SESSION DETAILS | | | |
| Image of session | | | |
| * Designer. * PyQt is a free Python bindings software open-source widget-toolkit Qt, implemented for cr * One of the popular approach to solve a programming problem is by creating objects. This is known as Object-Oriented Programming (OOP). * An object has two characteristics: * attributes * behavior * Let's take an example: * Parrot is an object, * name, age, color are attributes * singing, dancing are behavior * The concept of OOP in Python focuses on creating reusable code. This concept is also known as DRY (Don't Repeat Yourself). * In Python, the concept of OOP follows some basic principles: * Inheritance A process of using details from a new class without modifying existing class. * Encapsulation Hiding the private details of a class from other objects. * Polymorphism A concept of using common operation in different ways for different data input. * A class is a blueprint for the object. * We can think of class as an sketch of a parrot with labels. It contains all the details about the name, colors, size etc. Based on these descriptions, we can study about the parrot. Here, parrot is an object. | | | |