**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **11-06-2020** | **Name:** | **Anand kumar k** |
| **Course:** |  | **USN:** | **4al16ec002** |
| **Topic:** | **VLSI** | **Semester & Section:** | **8thsem ‘A’ sec** |
| **Github Repository:** | **Anand-courses** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session**      Combinational Logic Circuits are made up from basic logic NAND, NOR or NOT gates that are “combined” or connected together to produce more complicated switching circuits. These logic gates are the building blocks of combinational logic circuits. An example of a combinational circuit is a decoder, which converts the binary code data present at its input into a number of different output lines, one at a time producing an equivalent decimal code at its output.  Combinational logic circuits can be very simple or very complicated and any combinational circuit can be implemented with only NAND and NOR gates as these are classed as “universal” gates.  The three main ways of specifying the function of a combinational logic circuit are:   * 1. Boolean Algebra – This forms the algebraic expression showing the operation of the logic circuit for each input variable either True or False that results in a logic “1” output. * 2. Truth Table – A truth table defines the function of a logic gate by providing a concise list that shows all the output states in tabular form for each possible combination of input variable that the gate could encounter. * 3. Logic Diagram – This is a graphical representation of a logic circuit that shows the wiring and connections of each individual logic gate, represented by a specific graphical symbol, that implements the logic circuit.   and all three of these logic circuit representations are shown below.  combinational logic    As combinational logic circuits are made up from individual logic gates only, they can also be considered as “decision making circuits” and combinational logic is about combining logic gates together to process two or more signals in order to produce at least one output signal according to the logical function of each logic gate. Common combinational circuits made up from individual logic gates that carry out a desired application include Solid State Switches Standard TTL logic devices made up from Transistors can only pass signal currents in one direction only making them “uni-directional” devices and poor imitations of conventional electro-mechanical switches or relays. However, some CMOS switching devices made up from FET’s act as near perfect “bi-directional” switches making them ideal for use as solid state switches.  Solid state switches come in a variety of different types and ratings, and there are many different applications for using solid state switches. They can basically be sub-divided into 3 different main groups for switching applications and in this combinational logic section we will only look at the Analogue type of switch but the principal is the same for all types including digital. Analogue Bilateral Switches Analogue or “Analog” switches are those types that are used to switch data or signal currents when they are in their “ON” state and block them when they are in their “OFF” state. The rapid switching between the “ON” and the “OFF” state is usually controlled by a digital signal applied to the control gate of the switch. An ideal analogue switch has zero resistance when “ON” (or closed), and infinite resistance when “OFF” (or open) and switches with RON values of less than 1Ω are commonly available.  The word “Sequential” means that things happen in a “sequence”, one after another and in **Sequential Logic** circuits, the actual clock signal determines when things will happen next. Simple sequential logic circuits can be constructed from standard Bistable circuits such as: Flip-flops, Latches and Counters and which themselves can be made by simply connecting together universal NAND Gates and/or NOR Gates in a particular combinational way to produce the required sequential circuit. Classification of Sequential Logic As standard logic gates are the building blocks of combinational circuits, bistable latches and flip-flops are the basic building blocks of sequential logic circuits. Sequential logic circuits can be constructed to produce either simple edge-triggered flip-flops or more complex sequential circuits such as storage registers, shift registers, memory devices or counters. Either way sequential logic circuits can be divided into the following three main categories:   * 1. Event Driven – asynchronous circuits that change state immediately when enabled. * 2. Clock Driven – synchronous circuits that are synchronised to a specific clock signal. * 3. Pulse Driven – which is a combination of the two that responds to triggering pulses.   sequential logic device  As well as the two logic states mentioned above logic level “1” and logic level “0”, a third element is introduced that separates **sequential logic** circuits from their combinational logic counterparts, namely TIME. Sequential logic circuits return back to their original steady state once reset and sequential circuits with loops or feedback paths are said to be “cyclic” in nature. |
|  |
|  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date:** | **11-06-2020** | **Name:** | **Anand kumar k** | |
| **Course:** |  | **USN:** | **4al16ec002** | |
| **Topic:** | **mysql** | **Semester & Section:** | **8thsem ‘A’ sec** | |
| **AFTERNOON SESSION DETAILS** | | | |
| **Image of session** | | | |
| MySQL: Joins This MySQL tutorial explains how to use MySQL **JOINS** (inner and outer) with syntax, visual illustrations, and examples.  **Description**  MySQL **JOINS** are used to retrieve data from multiple tables. A MySQL JOIN is performed whenever two or more tables are joined in a SQL statement.  There are different types of MySQL joins:   * MySQL INNER JOIN (or sometimes called simple join) * MySQL LEFT OUTER JOIN (or sometimes called LEFT JOIN) * MySQL RIGHT OUTER JOIN (or sometimes called RIGHT JOIN)   So let's discuss MySQL JOIN syntax, look at visual illustrations of MySQL JOINS, and explore MySQL JOIN examples.  **INNER JOIN** (simple join)  Chances are, you've already written a statement that uses a MySQL INNER JOIN. It is the most common type of join. MySQL INNER JOINS return all rows from multiple tables where the join condition is met. Syntax The syntax for the INNER JOIN in MySQL is:  SELECT columns  FROM table1  INNER JOIN table2  ON table1.column = table2.column; Visual Illustration In this visual diagram, the MySQL INNER JOIN returns the shaded area:  MySQL  The MySQL INNER JOIN would return the records where table1 and table2 intersect. Example Here is an example of a MySQL INNER JOIN:  SELECT suppliers.supplier\_id, suppliers.supplier\_name, orders.order\_date  FROM suppliers  INNER JOIN orders  ON suppliers.supplier\_id = orders.supplier\_id;  This MySQL INNER JOIN example would return all rows from the suppliers and orders tables where there is a matching supplier\_id value in both the suppliers and orders tables.  If we run the MySQL SELECT statement (that contains an INNER JOIN) below:  SELECT suppliers.supplier\_id, suppliers.supplier\_name, orders.order\_date  FROM suppliers  INNER JOIN orders  ON suppliers.supplier\_id = orders.supplier\_id; Old Syntax As a final note, it is worth mentioning that the MySQL INNER JOIN example above could be rewritten using the older implicit syntax as follows (but we still recommend using the INNER JOIN keyword syntax):  SELECT suppliers.supplier\_id, suppliers.supplier\_name, orders.order\_date  FROM suppliers, orders  WHERE suppliers.supplier\_id = orders.supplier\_id; LEFT OUTER JOIN Another type of join is called a MySQL LEFT OUTER JOIN. This type of join returns all rows from the LEFT-hand table specified in the ON condition and **only** those rows from the other table where the joined fields are equal (join condition is met). Syntax The syntax for the LEFT OUTER JOIN in MySQL is:  SELECT columns  FROM table1  LEFT [OUTER] JOIN table2  ON table1.column = table2.column;  In some databases, the LEFT OUTER JOIN keywords are replaced with LEFT JOIN. | | | |