**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **27-05-2020** | **Name:** | **Kiran N** |
| **Course:** | **Logic Design** | **USN:** | **4al16ec031** |
| **Topic:** | **1.Boolean equations for digital circuits. Combinational**  **circuits: Conversion of MUX**  **and Decoders to logic gates.**  **2.Analysis of clocked sequential**  **circuits** | **Semester & Section:** | **8th and A** |
| **Github Repository:** | **Kiran-course** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session** |
| **REPORT**  Boolean Algebra is an algebra, which deals with binary numbers & binary variables. Hence, it is also  called as Binary Algebra or logical Algebra. A mathematician, named George Boole had developed this algebra in 1854. The variables used in this algebra are also called as Boolean variables. The range of voltages corresponding to Logic ‘High’ is represented with ‘1’ and the range of voltages corresponding to logic ‘Low’ is represented with ‘0’ .Postulates and Basic Laws of Boolean Algebra  In this section, let us discuss about the Boolean postulates and basic laws that are used in Boolean  algebra. These are useful in minimizing Boolean functions.  Boolean Postulates  Consider the binary numbers 0 and 1, Boolean variable xx and its complement x′x′. Either the  Boolean variable or complement of it is known as literal. The four possible logical OR operations among these literals and binary numbers are shown below.  x + 0 = x  x + 1 = 1  x + x = x  x + x’ = 1  Similarly, the four possible logical AND operations among those literals and binary numbers are shown below.  x.1 = x  x.0 = 0  x.x = x  x.x’ = 0  These are the simple Boolean postulates. We can verify these postulates easily, by substituting the  Boolean variable with ‘0’ or ‘1’.  Basic Laws of Boolean Algebra  Following are the three basic laws of Boolean Algebra.  Commutative law  Associative law  Distributive law  Commutative Law  If any logical operation of two Boolean variables give the same result irrespective of the order of those two variables, then that logical operation is said to be Commutative. The logical OR & logical AND operations of two Boolean variables x & y are shown below  x + y = y + x  x.y = y.x  The symbol ‘+’ indicates logical OR operation. Similarly, the symbol ‘.’ indicates logical AND operation and it is optional to represent. Commutative law obeys for logical OR & logical AND operations.  Associative Law  If a logical operation of any two Boolean variables is performed first and then the same operation is performed with the remaining variable gives the same result, then that logical operation is said to be Associative. The logical OR & logical AND operations of three Boolean variables x, y & z are shown below.  x + y+zy+z = x+yx+y + z  x.y.zy.z = x.yx.y.z  Associative law obeys for logical OR & logical AND operations.  Distributive Law  If any logical operation can be distributed to all the terms present in the Boolean function, then that logical operation is said to be Distributive. The distribution of logical OR & logical AND operations of three Boolean variables x, y & z are shown below.  x.y+zy+z = x.y + x.z  x + y.zy.z = x+yx+y.x+zx+z  Distributive law obeys for logical OR and logical AND operations.  These are the Basic laws of Boolean algebra. We can verify these laws easily, by substituting the  Boolean variables with ‘0’ or ‘1’. Theorems of Boolean Algebra  The following two theorems are used in Boolean algebra.  Duality theorem  DeMorgan’s theorem  Duality Theorem  This theorem states that the dual of the Boolean function is obtained by interchanging the logical AND operator with logical OR operator and zeros with ones. For every Boolean function, there will be a corresponding Dual function.  DeMorgan’s Theorem  This theorem is useful in finding the complement of Boolean function. It states that the  complement of logical OR of at least two Boolean variables is equal to the logical AND of each  complemented variable.  DeMorgan’s theorem with 2 Boolean variables x and y can be represented as  x+yx+y’ = x’.y’  The dual of the above Boolean function is  x.yx.y’ = x’ + y’  Therefore, the complement of logical AND of two Boolean variables is equal to the logical OR of each complemented variable. Similarly, we can apply DeMorgan’s theorem for more than 2 Boolean variables also.  COMBINATIONAL CIRCUIT:  Combinational circuit is a circuit in which we combine the different gates in the circuit, for example encoder, decoder, multiplexer and demultiplexer. Some of the characteristics of combinational circuits are following −  The output of combinational circuit at any instant of time, depends only on the levels present at input terminals.  The combinational circuit do not use any memory. The previous state of input does not have any effect on the present state of the circuit.  A combinational circuit can have an n number of inputs and m number of outputs.  Common Anode 7-Segment Display For common anode apply +5 volts to vcc pin in series to a 510 ohm-1k ohm resistor. This resistor is very important always include it other wise your seven segment display will be damaged by over current.  Note both the vcc pins are short so apply +5 volts on only one pin and leave other empty.  Ground the dp(decimal/display point) pin if you want it to illuminate for ever. If you to control  dp(decimal/display point) led than connect it to some control system, microcontroller etc.  In common Anode the Cathode(-) side of led’s are connected to a,b,c,d,e,f,g pins of seven segment  display.  In common anode seven segment display’s led becomes lit when we ground any a,b,c,d,e,f,g pin.  Common Anode seven segment display’s color is usually gray.    There are two types of input to the combinational logic :  •External inputs which not controlled by the circuit.  •Internal inputs which are a function of a previous output states.  Secondary inputs are state variables produced by the storage elements, where as  secondary outputs are excitations for the storage elements.  Types of Sequential Circuits :  There are two types of sequential circuit :  •Asynchronous sequential circuit –  this circuit do not use a clock signal  but uses the pulses of the inputs. These circuits are faster than synchronous sequential circuits because there is clock pulse and change their state immediately when there is a change in the input signal. We use asynchronous sequential circuits when speed of operation is important and independent of internal clock pulse.  But these circuits are more  Difficult to design and their output is uncertain.  •Synchronous sequential circuit –  These circuit uses clock signal and level inputs (or pulsed) (with restrictions on pulse width and circuit propagation). The output pulse is the same duration as the clock pulse for the clocked sequential circuits. Since they wait for the next clock pulse to arrive to perform the next operation, so these circuits are bit slower compared to asynchronous.  Level output changes state at the start of an input pulse and remains in that until the next input or  clock pulse.  We use synchronous sequential circuit in synchronous counters, flip flops, and in the design of  MOORE-MEALY state management machines.  We use sequential circuits to design Counters, Registers, RAM, MOORE/MEALY Machine and other  state retaining machines.  The behaviour of a clocked sequential circuit is determined from  •The inputs  •The outputs  •The state of its flip-flops  The outputs and the next state are both a function of the inputs and the present state .To analyze a  sequential circuit, we can use State equations, State table, State diagram, and Flip-Flop input  equations.  Digital Clock Circuit Design Using 7493  The 4 blocks of a digital clock are  •1 Hz clock generator to generate 1 PPS (pulse per second) signal to the seconds block.  •SECONDS block - contains a divide by 10 circuit followed by a divide by 6 circuit. Will  generate a 1 PPM (pulse per minute) signal to the minutes block. The BCD outputs connect to  the BCD to Seven Segment circuit to display the seconds values.  •MINUTES block - identical to the seconds block it contains 2 dividers; a divide by 10 followed  by a divide by 6. Will generate a 1 PPH (pulse per hour) signal to the HOURS block. The BCD  outputs connects to the BCD to Seven Segment circuit to display the minutes values.  •HOURS block - depending on whether it is a 12 or 24H clock, will have a divide 24 or divide by  12. For 24H, it will count from 00 to 23. For 12H, it will count from 00 to 11. The BCD outputs  connects to the BCD to Seven Segment circuit to display the hours values.  The clock can be designed as a 24H or 12H clock. We will explain the steps to arrive at the  combinational logic to obtain a 12H clock and we will leave it to the reader to design the 24H clock as an exercise. Click hints if you need help to design the 24H clock.  12H Clock:  •In order to use all 4 bits of the IC1 (ones) counter, Q0 must be connected to CP1. Q0 is LSB and  Q3 is MSB. The input clock is connected to CP0.  •Since less than 3 bits are required for IC2 (tens), Q0 is not used. Q1 is LSB and Q3 is MSB. The  input clock is connected to CP1.  •The truth table of the counter is abbreviated - omitting those rows where the MR inputs to the  counters are 0. Recall that for the 7493, a 1 to the MR will reset the counters to 0.  •To simplify the table, K is Q0 of IC1 (ones), G is Q0 of IC2 (tens) and so on.  •For the 12H clock, when the count in BCD reaches  o  0A, IC1 must be cleared (Y=1)  o  12, IC1 must be cleared (Y=1) and IC2 must be cleared (X=1)  •Using  SOP (sum of products), we obtain  o  Y = HJ + GJ where Y is the IC1 MR1, MR2 inputs connected together  o  X = GJ where X is the IC2 MR1, MR2 inputs connected together |

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