SRN: _ _ _ _ _ _ _ _ _ _ _

PES University, Bangalore

(established under Karnataka Act No. 16 of 2013)

End Semester Assessment (ESA) B. Tech. 3rd Semester, Dec. 2019

UE18CS201: Digital Design and Computer Organization

Time: 3 hours **All questions to be answered** **Max. marks: 100**

Candidate Signature: _____

1. (a) How many Boolean functions of n inputs are there? (2)

   (b) The truth tables shown below are for four input logic gates. For each truth table, write down the name of the logic gate it represents. (4)

   Logic gate 1:

Inputs				Output
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

   Logic gate 2:

Inputs				Output
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

   (c) For the following Boolean function:

$$F(a, b, c, d) = \Sigma(0, 1, 2, 3, 4, 5, 8, 10, 15)$$

   draw the K-map, mark the implicants required, and write minimized SOP formula obtained using the marked implicants. (5)

   (d) In class we have studied various combinational logic structures such as muxes, decoders, encoders and adders. Some such structures studied in class have been implemented as gate level logic diagrams below. For each diagram, write the name of the logic structure and its size. For example, if there was a diagram of an encoder having four inputs and two outputs, the expected answer would simply be "4:2 encoder." *Hint:* If not clear from the diagram, functionality could be inferred by writing the truth table for the logic.

SRN: _ _ _ _ _ _ _ _ _ _ _

Diagram 1:

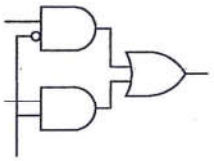


Diagram 2:

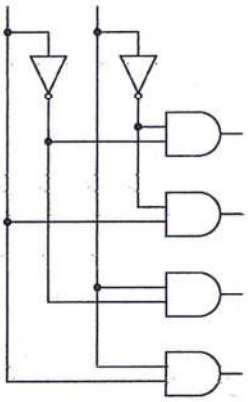
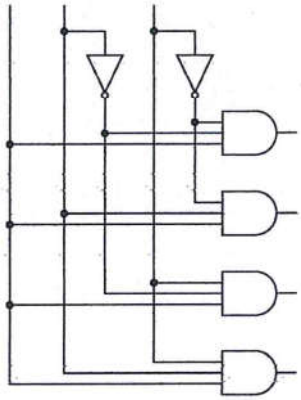


Diagram 3:



(5)

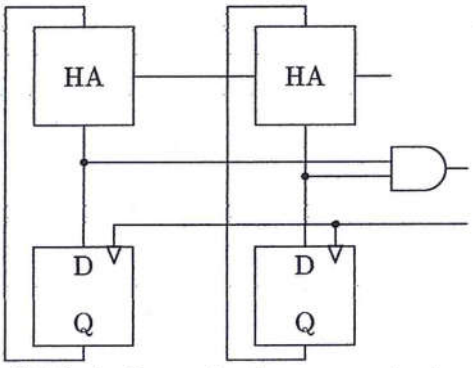
(e) In class, several laws or identities of Boolean Algebra have been studied. For example the equation "$a \cdot 1 = a$" and its dual "$a + 0 = a$" together are called the law of identity. Write down the equations for the laws of commutativity, associativity, distributivity and DeMorgan's law. As above, for each law, write two equations. (4)

2. (a) Draw the logic circuit diagram of a barrel shifter (composed of 2:1 muxes) with four data inputs, four data outputs (and two control inputs). (4)

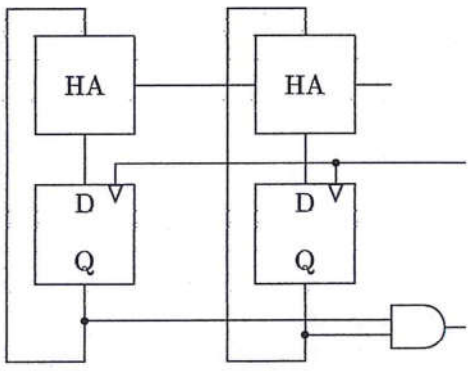
(b) Draw the logic circuit diagram of a D latch with enable. Draw the logic circuit diagram of a D edge-triggered flip flop composed of above D latches arranged in a master slave configuration. (5)

(c) We know that any sequential circuit is a Finite State Machine. We also know that an FSM is either of Mealy type or Moore type. In each sequential circuit below, the top row is half adders below which are D flip-flops, and the circuit output is the output of the AND gate. For each circuit, state its FSM type along with a brief justification.

Sequential circuit 1



Sequential circuit 2



(4)

(d) A divide-by-3 counter has one output and no inputs. The output is high for one clock cycle out of every 3. In other words, the output divides the frequency of the clock by 3. The output is high for the first clock cycle immediately after reset, low for the next two clock cycles, then high again for a clock cycle, and so on. Draw the state diagram, write the state encoding table, state transition table and the output table, and use the latter two tables to write the Boolean formulas, and finally draw the logic circuit diagram for the divide by three counter. Use Moore type FSM and binary encoding. (7)

SRN: _ _ _ _ _ _ _ _ _ _ _

3. (a) Consider the carry lookahead adder discussed in class. Write down the formula for computing the delay of a carry lookahead adder. Consider the parallel prefix XOR computation discussed in class. If the delay of each XOR gate is 0.1 ns, what is the delay of a 128-bit (128 input) parallel prefix XOR computation logic? How many XOR gates does the above logic circuit contain? (5)

   (b) Draw the diagram of an 8-bit multiplier that performs multiplication via shifts and adds. Draw the version that uses a single shift register. Label the bitwidths of the various components and the shift direction. (5)

   (c) What is the advantage of using floating point numbers over integers? Consider the 32-bit value 1100 0111 0101 1000 0000 0000 0000 0000 (C7580000 in hexadecimal) to be in IEEE 754 single precision floating point format. What decimal number is represented by it? (5)

   (d) Suppose a Wallace tree multiplier is used to multiply two five-bit numbers using the approach discussed in class (only full adders are used, no half adders used or any other optimizations done). So five partial products, with sizes from 5 to 9 need to be added. Assume that for the final addition a ripple carry adder (composed only of full adders) is used. Draw the Wallace tree adders and the final ripple carry adder, labelling each wire with its bitwidth and each node (box) with number of CSAs (or FAs) it contains. If the delay of single FA is 0.25 ns, how long does the above circuit take to compute the product? (5)

4. (a) Consider the MIPS R-type instruction format for which the opcode field is 0. For the R-type assembly instruction `or r7, r11, r11` the machine instruction (in hexadecimal) is 016b3825. Given that the funct field for the `nor` instruction is (in binary) 100111, what is the machine language instruction for the assembly language instruction `nor r11, r7, r7`? Without knowing the contents of the `r11` register, what, if anything, can be said about the value that will be written into `r7` by the above `or` instruction? Similarly. what, if anything, can be said about what will be written into the `r11` register by the `nor` instruction above? (5)

   (b) Consider the high level language statement:
   `if (i != j) f = g - h;`
   Write the MIPS assembly language instructions that implement the above statement. Assume that the values of the variables f, g, h, i and j are present in the registers r1, r2, r3, r4 and r5 rcspcctivcly. (4)

   (c) For each of the following assembly instructions, specify the type of addressing mode used: (5)
   1. `lw`, 2. `add`, 3. `addi`, 4. `beq`, 5. `j`

   (d) Consider the state diagram for the control logic of the multicycle MIPS microarchitecture. Is it a Mealy type or Moore type FSM? Briefly justify your answer. Based on the state diagram, how many clock cycles does it take to execute `lw` instruction and how many for the `beq` instruction? Suppose the corresponding MIPS multicycle datapath is modified such that a dedicated adder/subtractor is included along with the PC (Program Counter) to compute any changes required to the PC value (as was done in the lab assignments). What effect, if any, would doing so have on the execution time of the `lw` and `beq` instructions? Briefly justify your answer. (6)

5. (a) In a standard personal computer, what is placed on top of a microprocessor? (1)

   (b) Consider the 16-bit number (in hexadecimal) ABCD. Write the number (in hexadecimal) obtained by sign extending above number to 32-bits. Now suppose the above 32-bit number

is written into memory address 1000 using little endian addressing. If the byte at address 1000 is read back, what would its value be? (4)

(c) Consider the use of a systolic array of $n \times n$ PEs (Processing Elements) to multiply two $n \times n$ matrices. How many clock cycles does it take? During the multiply, the top left PE of the array, PE (0,0), finishes computing after n clock cycles, PEs (0,1) and (1,0) finish computing after $n + 1$ clock cycles, and so on. Would it be possible to start multiplying another pair of matrices before the previous one is completed? If so, after how many clock cycles can the next matrix multiply be started? What is the maximum number of matrix multiplies that can be in computation at the same time? (4)

(d) We have studied ripple carry adders and parallel prefix adders. We have also studied ripple carry incrementers (which add one to the input number). Is it possible to construct a parallel prefix incrementer? Consider an incrementer logic circuit with eight data inputs, a control input and eight data outputs. If the control input is one, the output is the input incremented by one, else output is same as input. Draw the logic circuit of above incrementer constructed using the parallel prefix technique. (7)

(e) How many Boolean functions of a single input are there? Write the truth table for each. (4)