

# Indian Institute of Technology Kharagpur

## Department of Computer Science and Engineering

Laboratory Test-1, Autumn 2020-21

Computer Organization Laboratory (CS39001)

Students: 121

Date: 21-October-2020

Full marks: 40

Time: 90 minutes

Credit: 40%

**INSTRUCTIONS:** This is an OPEN-BOOK, OPEN-NOTES test. The questions are such that they require either numerical answers or short answers (including logic diagrams). Please write **ONLY THE ANSWERS** on sheets of paper, scan them (or take photos), and submit the images of the solution sheets on Moodle (submit a single zipped folder if you have multiple files). **DO NOT FORGET TO WRITE YOUR NAME AND ROLL NUMBER AT THE TOP OF YOUR ANSWER SHEETS.** You may use calculators if required. **ANSWER ALL QUESTIONS.**

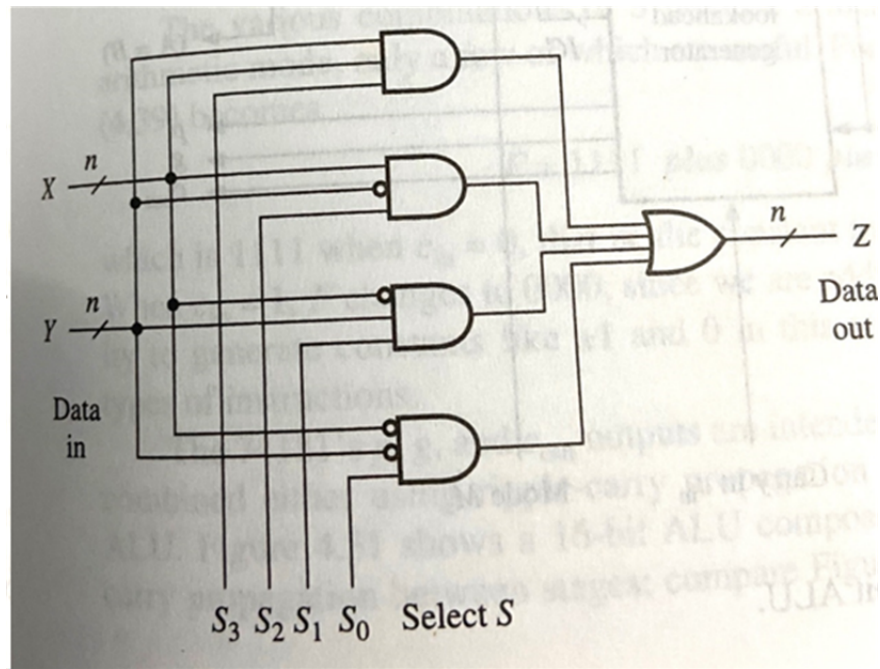


Figure 1: An  $n$ -bit logic unit.

- Consider the logic diagram of an  $n$ -bit logic unit, as shown in Fig. 1 above, which can be used to realize all sixteen two-variable functions.
  - Write the logic expression for  $Z$ . [2]
  - Write the functions realized for: (a)  $S = 0110$  and (b)  $S = 0011$ . [1 + 1 = 2]
  - What value of  $S$  is required to realize: (a) XOR function, and (b) OR function? [1 + 1 = 2]
- The four-variable AOI function is defined as:  $\text{AOI}(a, b, c, d) = \overline{(a \wedge b) \vee (c \wedge d)}$ . Define the ITE operator on  $v, g, h$  as:  $\text{ITE}(v, g, h) = vg + v'h$ . Derive the AOI function using the ITE operator, and hence, draw the logic diagram of an 2:1 multiplexer based implementation of the AOI function. Your logic diagram should exactly match the expression you have derived. [4 + 2 = 6]
- Provide a derivation for the numerical value of the 2's complement of a binary number  $X$ , in terms of the value of  $X$ , where  $X$  is a  $n$ -bit unsigned binary number with a leading 0. [3]

4. Does Booth's multiplication offer a speed advantage over shift-and-add multiplication? If so, under what situation? [2]
5. Consider the following Verilog code segment:

```

module  incomp_state_spec  (curr_state, flag);
    input    [0:1]  curr_state;
    output reg [0:1]  flag;

    always  @ (curr_state)
        case  (curr_state)
            0, 1  :  flag = 2;
            3     :  flag = 0;
        endcase
endmodule

```

Draw the logic diagram of the circuit corresponding to this code segment that would be inferred during the logic synthesis process. [5]

6. Consider the implementation of a Full Adder (FA) of with two Half Adders (HA). HAs are implemented with XOR and NAND gates only. Write the expressions for the worst-case delay of an  $n$ -bit Ripple Carry Adder (RCA), in terms of the delay of an XOR gate ( $t_{XOR}$ ) and an NAND gate ( $t_{NAND}$ ). [4]
7. Consider an  $n$ -bit adder capable of adding two operands represented as  $X = x_{n-1}x_{n-2} \dots x_0$  and  $Y = y_{n-1}y_{n-2} \dots y_0$ . The production of the  $i$ -th sum bit  $S_i$  can be decomposed into two parts: (i) obtaining the  $i$ -th carry-in bit  $C_i$ , followed by (ii) computing the sum bit  $S_i$ . Consider the following three signals:

- **Propagate:**  $p_i = x_i \oplus y_i$
- **Generate:**  $g_i = x_i \cdot y_i$
- **Kill:**  $k_i = \overline{x_i + y_i}$

Answer the following:

- (a) Give the expression for the carry-out ( $C_{i+1}$ ) in terms of  $g_i$ ,  $p_i$  and the carry-in ( $C_i$ ), for the  $i$ -th bit slice. [2]
- (b) Give the expression for the complement of the carry-out ( $C'_{i+1}$ ) in terms of  $k_i$ ,  $p_i$  and complement of the carry-in ( $C'_i$ ), for the  $i$ -th bit slice. [2]
- (c) Give the general expression for the carry-out ( $C_{i+1}$ ) in terms of  $g_i$ ,  $p_i$  and the input carry-in ( $C_0$ ), for any  $0 \leq i \leq n-1$ . [3]
- (d) (Fill in the blanks) The AND-OR realization of  $C_{i+1}$  requires: (a) one OR gate with \_\_\_\_\_ inputs, and, (b) \_\_\_\_\_ AND gates in total (irrespective of the number of inputs). [ $2 + 2 = 4$ ]
- (e) Assume that the delay of an  $n$ -input AND gate is  $t_{AND,n}$  and that of an  $n$ -input OR gate is  $t_{OR,n}$ . Write an expression for the worst-case delay of a 4-bit *Carry Lookahead Adder* (CLA), denoted by  $t_{CLA-4}$ , in terms of  $t_{AND,n}$  and  $t_{OR,n}$ . [3]