2301-COL215 Mid-term_Exam

Akshat Jha

TOTAL POINTS

56 / 60

QUESTION 1

Minimum logic area circuit 10 pts

1.1 Karnaugh Map 5 / 5

✓ **+ 5 pts** *Correct answer and pointing use of 2 K-maps*

+ 3 pts partially correct

+ 0 pts Wrong or not answered

1.2 Disadvantages of K Map 3 / 5

+ 5 pts Correctly pointing, miss out on optimising across the two boolean functions

✓ **+ 3 pts** *partially correct*

+ 2 pts some valid point

+ 0 pts wrong

QUESTION 2

2 Ripple carry adder 10 / 10

✓ **+ 10 pts** *Correct : 3n + max(m, n)*

+ 8 pts Partial : 3n + m

+ 5 pts Partial : 4n

+ 0 pts Incorrect/Not Attempted

QUESTION 3

3 Encoder 10 / 10

✓ **+ 4 pts** *Truth Table*

✓ **+ 6 pts** *Explanation*

+ 0 pts Answer not correct

QUESTION 4

4 VHDL model 8 / 10

+ 0 pts Incorrect/Unattempted

✓ **+ 8 pts** *Correct Idea, Incorrect implementation*

+ 10 pts Correct

+ 5 pts Correct direction, Missing proper definitions

Variables cannot be declared without process.

QUESTION 5

5 CMOS circuit 10 / 10

✓ **+ 10 pts** *Fully correct*

+ 0 pts Incorrect

+ 5 pts Click here to replace this description.

+ 7.5 pts Click here to replace this description.

+ 2.5 pts Click here to replace this description.

QUESTION 6

6 Martian Number system 10 / 10

✓ **+ 10 pts** *Correct*

+ 7.5 pts Calculation mistake in solving the final equation

+ 7.5 pts Couldn't arrive at a single solution or Didn't check consistency of equations or Didn't explain the reason for rejecting a solution

+ 5 pts Partly Correct

+ 0 pts Incorrect/Unattemted

An alternative approach sould be using a MUX implementation. With n output functions and m inputs, we can create a n to 1 MUX with $\lceil \log_2 n \rceil$ selects and we then need to optimize each function individually and pass it through the MUX. (Though, this would fair worse) (K-map reduction)

Name: AKSHAT JHA

Entry Number: 2022CS11101

Department of Computer Science and Engineering, IIT Delhi
Digital Logic and System Design (COL 215)
I Semester 2023-24, Mid-semester Exam, Maximum Marks: 60, 26 Sep 2023, 3:30 PM to 5:00 PM
Please write your answers ONLY IN THE SPACE BELOW THE QUESTIONS
Use reverse side of paper for rough work.

1. **[5+5=10 Marks]** Suppose we wish to design a minimum-area logic circuit implementing a multi-output boolean function (e.g., the circuit has 4 inputs and 2 outputs). Minimum area ⇒ fewer transistors.

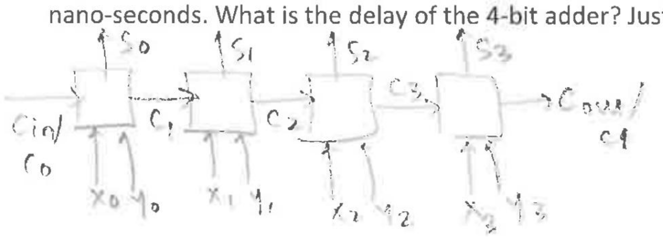
   a. Suggest how we can use Karnaugh Maps (K-Maps) to achieve this objective.

We can get the common minterms out of all the K-maps (implicants). Once we gather these minterms (common implicants), we group them using K-map rules. This algorithm can be implemented using a greedy approach. Now the common implicants (now, maybe reduced) can be connected appropriately along with the uncommon ones in the CMOS implementation and this would gurantee that fewer transistors are used. ⇒ lesser area is required. On an individual level, we ensure to include prime implicants of each output function to reduce # transistors there as well.

   b. Point out a disadvantage of minimising the multi-output function using the K-Map approach.

A lot of interconnections would be required in this approach. Hence the delays (time) to complete computation would be large. Another problem might be when the number of input variables increases as K-map solutions become increasingly complex and on top of that reducing the common implicants would require many steps too.

2. **[10 Marks]** Consider the 4-bit ripple carry adder discussed in class. Suppose the SUM bit in a one-bit full adder stabilises after *m* nano-seconds, and the CARRY OUT bit stabilises after *n* nano-seconds. What is the delay of the 4-bit adder? Justify your answer.



$T(a)$ = Time to get signal a.

We can see that

$T(s_i) = T(c_{i}) + m$

$T(c_{i+1}) = T(c_i) + n$

assuming the relevant x, y are ready.

Solving this reccurrence gives us

$T(c_0) = 0$ ("Ready") $T(c_4) = 4n$

$T(s_0) = m$

$T(c_1) = n$

$T(s_1) = m+n$

$T(s_2) = 2n$

$T(s_2) = 2n+m$

$T(c_3) = 3n$

$T(s_3) = 3n+m$

The max delay out of the SUM BITS is $3n+m$

CARRY BITS is $4n$

Hence, the delay of the 4 bit adder is $\max\{3n+m, 4n\}$ or $3n + \max\{m, n\}$

If the cout is to be discarded, the delay is $3n+m$

D_0	D_1	D_2	D_3	x	y	v	
0	0	0	0	x	x	0	R_1
1	0	0	0	0	0	1	R_2
0	1	0	0	0	1	1	
1	1	0	0	0	1	1	R_3
0	0	1	0	1	0	1	
0	1	1	0	1	0	1	
1	0	1	0	1	0	1	R_4
1	1	1	0	1	0	1	
0	0	0	1	1	1	1	
0	0	1	1	1	1	1	
0	1	0	1	1	1	1	R_5
0	1	1	1	1	1	1	
1	0	0	1	1	1	1	
1	0	1	1	1	1	1	
1	1	0	1	1	1	1	
1	1	1	1	1	1	1	

Here is the full table showing all possible table rows. Note how we can club rows together into $R_1 - R_5$.

$v = D_0$ ~~AND~~ OR D_1 ~~AND~~ OR D_2 OR D_3

$x = D_2$ OR D_3

$y = D_3$ OR $(D_1$ and $\bar{D}_2)$

obtained from K-Map minimization over this table achieves the desired output.

In R_1, x and y get '0' value which is fine since the validity bit v takes care of which values to discard.

3. **[10 Marks]** In the Priority Encoder we discussed in class, the output was derived from input by just OR-ing the appropriate input variables. Why is this valid, considering that only a subset of table rows were listed in the truth-table? Explain.

	D_0	D_1	D_2	D_3	x	y	V
R1	0	0	0	0	x	x	0
R2	1	0	0	0	0	0	1
R3	x	1	0	0	0	1	1
R4	x	x	1	0	1	0	1
R5	x	x	x	1	1	1	1

Consider a 4-bit priority encoder. 2^4 bit-vectors are possible.
R1 → 1 entry. R2 → 1 entry.
R3 → 2^1 possible values (1 x)
R4 → $2^2 = 4$ (2 x's) R5 → $2^3 = 8$ (3 x's)
In total we do get 16 rows had we enumerated the cases separately.

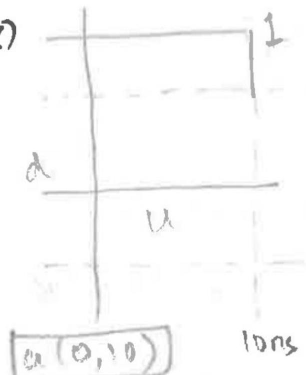
In any one of the R_5 rows, for example, all 4 such table rows would result in (1,1,1) as output; so just taking into consideration one row with the characteristic that $D_3 = 1$ and others "unknown" would model the function's behaviour (consider the full table in the adjoining page)

4. **[10 Marks]** Write a VHDL model with the following behaviour: When simulated, the simulation fails to proceed beyond 10 nanoseconds.

```
ENTITY stopper IS PORT (c: IN BIT ; a: OUT BIT)
  END stopper;
ARCHITECTURE stop of stopper IS
VARIABLE  a : BIT := '1';
~~VARIABLE  b : BIT := '1';~~
BEGIN
      a <= '0' after 10ns;

      a <= NOT a ;

END stop;
```

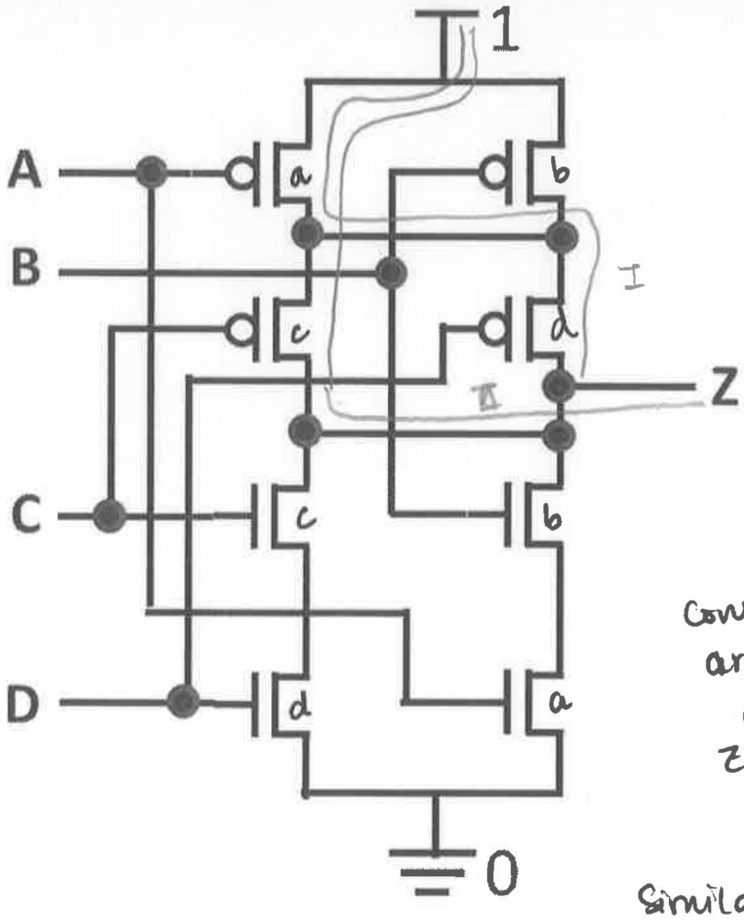


Event a(0,10) is executed and then infinite deltas of alternating values of '0' and '1' for a are created that stop the simulation at 10 ns

Delta 1	Delta 2	Delta 3 ...
a <= 1	a <= 0	a <= 1
b <= 1	b <= 0	b <= 1

The deltas don't stabilize

5. **[10 Marks]** What function of A, B, C, and D does the following CMOS transistor circuit implement? Justify. [Note that crossing wires are connected only when there is a black circle at the intersection]



Looking at the NMOS part,
Z is connected to ground and hence '0' when
c and d are HIGH
OR a and b are LOW

This gives us
$\bar{Z} = ab + cd$
or $Z = \overline{ab+cd}$

We can look at the PMOS part now,
Consider path I, both PMOS are activated when a and d are low and Z gets connected to '1'.
i.e. $a'd'$ gives a maxterm of Z.

Similarly, Z gets '1' when a and c are both low
$\Rightarrow a'c'$

Using all possible "low" combinations, $Z = a'c' + a'd' + b'd' + b'c'$
$= (a'+b')(c'+d')$
$= (\overline{ab}) \cdot (\overline{cd}) =$
$= (\overline{ab+cd})$ (Same from NMOS)

$Z = \overline{ab+cd}$

23x5

38
400 (90-23)
1609
- 160
1449
529

$\frac{23+3}{2}$ = 13,10

302

5(864)
5(69) - 55(8

45x8
9x325
1300
81+130
211

382-
1999

$\frac{36+12}{2}$

22
+
25,13

$\overline{cd+ab}$

a'd'+a'b'
+c'd'+b'c'

6. **[10 Marks]** The first manned expedition to Mars ("*Mangalyaan*") found only the ruins of a civilization. From the artifacts and pictures, the explorers deduced that the creatures who produced this civilization were four-legged beings with a tentacle that branched out at the end with a number of grasping "fingers." After much study, the explorers were able to translate Martian mathematics. They found the following equation: $5x^2 - 50x + 125 = 0$ with the indicated solutions x = 5 and x = 8. The value x = 5 seemed legitimate enough, but x = 8 required some explanation. Then the explorers reflected on the way in which Earth's number system developed, and found evidence that the Martian system had a similar history. How many fingers would you say the Martians had? Justify. (*From The Bent of Tau Beta Pi, February 1956. Extracted from Wakerly, Digital Design*) [*Hint: the number of fingers relates to number representation the same way it does for humans.*]

let the Martian system have base b. = # fingers

In base 10,

$(5)_b = 5$

$(50)_b = 5b + 0 = 5b$

$(125)_b = 1(b^2) + 2(b) + 5$

$5x^2 - 5bx + b^2 + 2b + 5 = 0$

5 and 8 are solutions to x in this quadratic in b.

$b^2 + b(2 - 5x) + (5 + 5x^2) = 0$

~~5+8 = (2+5x) and 5×8 = 5+5x² ⇒ x²+1 = 8~~

~~and x~~

5 and 8 are the same in base 10.

Putting in 5 : $b^2 - 23b + 130 = 0 \Rightarrow (b-13)(b-10) = 0$

Putting in 8 : $b^2 - 38b + 325 = 0 \Rightarrow (b-25)(b-13) = 0$

We see that $b = 13$ is a common root of the quadratic equations. Hence the Martian system should ~~be~~ base 13 (13 fingers)

<u>Verification</u> convert from base 13 to 10.

$(5)_{13}x^2 - (50)_{13}x + (125)_{13} = 5x^2 - 65x + 169 + 26 + 5 =$

$5x^2 - 65x + 200$

$= 5(x^2 - 13x + 40)$

$= 5(x-5)(x-8)$

5 and 8 in base 10 satisfy this equation and they do in base 13 as well.

$(5)_{13} = (5)_{10}$

$(8)_{13} = (8)_{10}$