National University of Computer and Emerging Sciences, Lahore Campus

Course:	Digital Logic Design	Course Code:	EL-227
Program:	BS (Computer Science)	Semester:	Spring 2020
Duration:	(180 + 15) Minutes	Total Marks:	100
Paper Date:	03-July-2020	Weight	50%
Section:	All Sections	Page(s):	8
Exam:	Final		

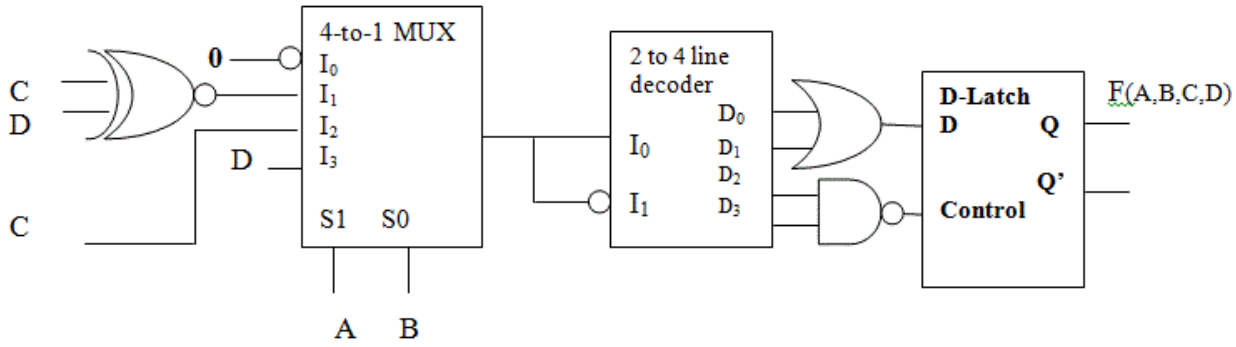
Instruction/Notes:

1. Solve this exam on an A4 size paper or assignment sheets or any presentable paper.
2. You must provide solution of the exam in this document, past clear screen shots of your solution only, any ambiguous/unclear answer will be considered wrong. You may provide your solution in a single pdf document but make sure solution of each question must be under proper heading of question number and respective part.
3. Submit rough work at the end of every question with proper heading.
4. The exam is open book and notes, but you are not allowed to take help from anyone. Use of any other unfair means is strictly prohibited. This may lead to an **F** grade in course**.**
5. Name of the solution document must be as **Final_Exam_Section_X [L19-1001],** otherwise no marks will be awarded.
6. Late submissions will not be allowed. So, make sure you have a back-up ready in case of power failure.
7. **Submit your solution on Google classroom. If you are facing any problem in uploading your work on Google class room then send your work via email.**
8. Questions during exam are not allowed. Take reasonable assumption if there's any ambiguity.
9. **180 Minutes** are allocated to solve the exam and **extra 15 mints** will be used to take pictures, make solution file and submission of the solution.

Question #	**Q1**		**Q2**		**Q3**			**Q4**							**Q5**				
Marks	**15**		**15**		**20**			**25**							**25**				
	a	**b**	**a**	**b**	**a**	**b**	**c**	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**a**	**b**	**c**	**d**	**e**
	10	**5**	**10**	**5**	**10**	**6**	**4**	**5**	**2**	**2**	**5**	**6**	**1**	**4**	**3**	**5**	**5**	**6**	**6**
Marks Obtained																			

Question 1: [15 marks]

Determine the following function implemented using a multiplexer, decoder and D-latch.



a) Solve and show the working of each block for every input combination given in the table. Direct answers/filling up the table only, will give **Zero credit**. **[10 Marks]**

b) Complete the given table. **[5 Marks]**

A	**B**	**C**	**D**				**F**
0	0	0	0				
0	0	0	1				
0	0	1	0				
0	0	1	1				
0	1	0	0				

Question 2 [15 Marks]:

X and Y are two 3-bit numbers. Using given circuit, implement

If X+Y >7 & X+Y is even

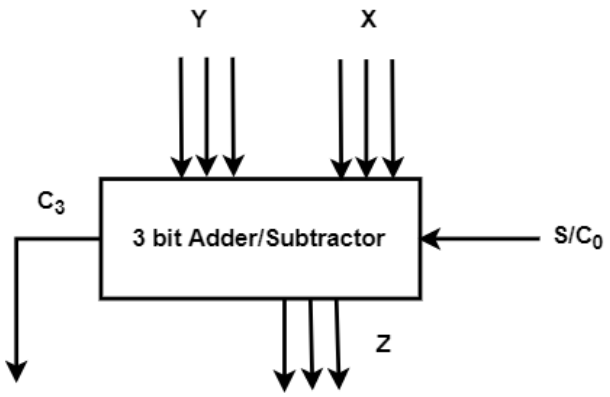
X-4

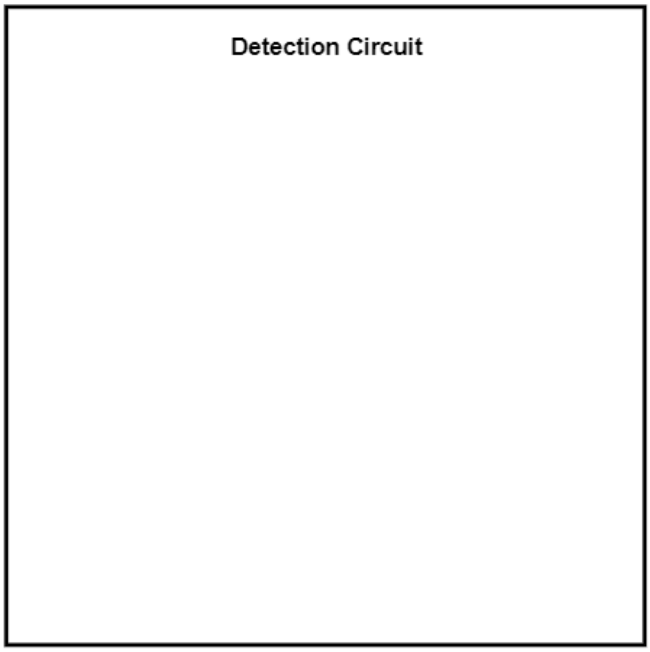
Else

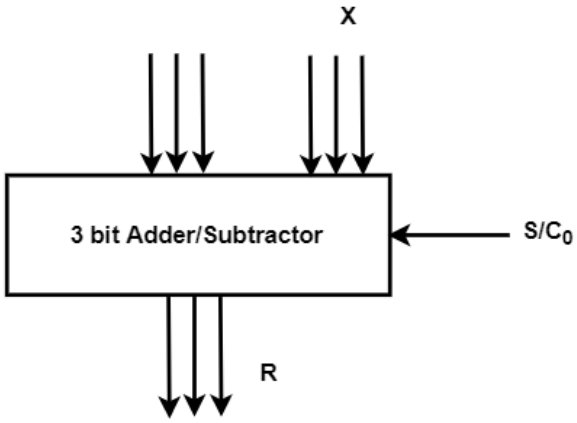
X+Y

a) Draw truth table for detection circuit. **[10 Marks]**

b) Complete the given circuit by embedding detection circuit. **[5 Marks]**


Y
X
C3
3 bit Adder/Subtractor
S/C0
Z


Detection Circuit


X
3 bit Adder/Subtractor
S/C0
R

Question 3: [20 marks]

Micci's Pancakes is a popular pancake food chain. There, a pan cake batter-mixing machine follows a fixed recipe for Micci's popular fluffy pancakes. The batter-mixing machine takes following amount of flour, milk and sugar from containers attached for Sweet or Sugar-Free pancakes.

Max capacity of each container is

Flour	2 Kg
Milk	1 Ltr
Sugar	1 Kg

Inputs	**Outputs**	**Quantities of Inputs**
F=Flour	Sweet	1 or 2 Kg
	Sugar-Free	2 Kg
M=Milk	Sweet	1 Ltr
	Sugar-Free	1 Ltr
S=Sugar	Sweet	1 Kg
	Sugar-Free	0 Kg

If any of the quantity combination listed is not according to recipe, outputs will be zero. Machine checks quantities in order of flour, milk then sugar and if found any one of them empty; machine will not proceed to check further quantities and will generate zero output except Milk. Since Milk is replaceable by Water, if no Milk found, it will generate don't care on output.

Based on given values of each ingredient, machine will prepare desired type of pancakes for customers.

a) Provide outputs for given **compact** truth table for batter-mixing machine. **[10 Marks]**

Flour (Kg)		Milk (Ltr.)	Sugar (Kg)	Sweet	Sugar-Free
F_1	**F_0**	**M**	**S**	**SW**	**SF**
0	0	X	X		
0	1	0	X		
0	1	1	0		
0	1	1	1		
1	0	0	X		
1	0	1	0		
1	0	1	1		
1	1	X	X		

b) Find **optimized** Boolean equations from truth table based on minimum hardware cost. **[6 Marks]**

c) Draw combinational circuit by using equations derived in part b. **[4 Marks]**

Question 4: [25 Marks]

Use **design procedure** to design a sequential circuit that can recognize a bit sequence (**S**) of following kind

$S = B_{n-1}\, B_{n-2} \ldots B_2\, B_1\, B_0$ Where B_0 is Least significant bit and B_{n-1} is Most significant bit

Moreover, **S has following characteristics**

1- **It will contain either count (C) of 0's which is completely divisible by 2.**

OR

2- **Number of 1's in S are multiple of 4 except 0.**

This circuit will accept a single bit input (X) at each clock pulse and produces a single bit output (Z) accordingly. **Sample inputs and outputs of this sequential circuit have also been given on next page, please refer them for more clarification.**

Note: Use an **MN** flip flop (as given below) to design this circuit.

Function Table for MN Flip Flop:

Clock	*M*	*N*	Q_{t+1}
0	X	X	Q_t
1	0	0	$\bar{Q}_t\ (\bar{M} + \bar{N})$.
1	0	1	**1**
1	1	0	**0**
1	1	1	$Q_t\ (M + N)$

Excitation Table for MN Flip Flop:

Q_t	Q_{t+1}	*M*	*N*
0	0	1	X
0	1	0	X
1	0	X	0
1	1	X	1

Sample input and their corresponding outputs are given as under:

Sample 1:

Input: 0 1 1 0 1 0 1 0 1 1 0 1 0 1 0 0 ….

Output: 0 0 0 1 0 0 1 1 0 0 0 0 1 1 0 1 ….

Sample 2:

Input: 1 1 0 0 0 1 0 0 1 1 0 1 0 1 0 1 ….

Output: 0 0 0 1 0 0 1 0 1 0 1 0 0 0 1 1 ….

Sample 3:

Input: 0 1 0 0 0 1 1 0 1 0 1 1 0 1 0 1 ….

Output: 0 0 1 0 1 0 0 0 1 1 0 0 0 0 1 1 ….

a) Draw state diagram for this circuit. **[5 Marks]**
b) Find one dimensional state table for state diagram. **[2 Marks]**
c) Perform state assignment and draw updated state table. **[2 Marks]**
d) Add Flip Flop inputs in state table obtained in **part c**, also show the updated table. **[5 Marks]**
e) Find optimized input equations for all inputs of flip flops by using K—map, show your whole working (only use updated state table obtained in **part d**). **[6 Marks]**
f) Find optimized output equation for this circuit using K-map, use Boolean algebraic rules/theorems to optimize this equation if more optimization is needed. **[1 Mark]**
g) Draw circuit diagram and label it properly, otherwise no credit will be awarded. **[4 Marks]**

Question 5: [25 Marks]

A scientist is experimenting on his memory and needs a machine to aid in his research. The machine needs a 2-dimensional array of total 8 cells. Each cell can store a 2-bit number. The task is to design a circuit that can store values in a cell of the array, get a value from a cell, shift the stored data by 1 cell, check if the scientist has correctly guessed the value in the cell or not. Sample Functionality of random instances is shown below:

Sample Functionality:

Operation	Array	Action	Description
Reset All	*0*: 0 0 0 0 *1*: 0 0 0 0 *00 01 10 11*	No Output	All the values are set to 0
Set Value (0, 01, 10)	*0*: 0 10 0 0 *1*: 0 0 0 0 *00 01 10 11*	No Output	10 inserted in the cell at row 0 and column 01
Get Value (0,01)	*0*: 0 10 0 0 *1*: 0 0 0 0 *00 01 10 11*	Output=10	10 value got from cell at row 0 and column 01
Check Value (0, 00, 11)	*0*: 11 10 01 10 *1*: 10 00 10 00 *00 01 10 11*	T=1	Guessed value of 11 at row 0 and column 00 is True.
Check Value (1, 11, 11)	*0*: 11 10 01 10 *1*: 10 00 10 00 *00 01 10 11*	T=0	Guessed value of 11 at row 1 and column 11 is false.
Shift Values	*0*: 00 11 10 01 *1*: 10 10 00 10 *00 01 10 11*	No output	All values shift right to next cell in the array.

You are required to implement the following functionalities:

a) **Reset All** functionality that sets all values in the cells to 00. **[3 Marks]**
b) **Set Value (R, C_1C_0, *X*)** functionality that takes a 2-bit binary number **X** from user and saves it in the array at specified row **R** and column **C_1C_0**. **[5 Marks]**
c) **Get Value (R, C_1C_0)** functionality that returns a 2-bit binary number stored at the row **R** and column **C_1C_0** of the array **[5 Marks]**
d) **Check Value (R, C_1C_0, *X*)** functionality compares the value **X** with the value stored at row **R**, and column **C_1C_0.** Output **T** shows its true or not. **[6 Mark]**
e) **Shift Values** functionality shifts all the values to next cell while last cell shifts data to the very first cell. **[6 Marks]**

Note: Assume that you already have Decoder(s), Encoder(s), MUX(s), DMUX(s), Adder-Subtractor(s), Multiplier(s), Divisor, Comparator, Registers and Counters blocks available. Properly label all blocks and inputs/outputs to get credit. **Credit will be given based on cost of the circuit.**