Mo Felonda
felonda @wisc.edu

CS/ECE 252 Introduction to Computer Engineering

Fall 2018
Instructor: Adil Ibrahim

Homework 4 (45 points)
Due On: October 12, 2018

Primary contact for this homework: Ragini Rathore (ragini@cs.wisc.edu)

For every question below, you need to show the complete working to receive full points. Please utilize the space provided under each question. Upload a PDF version on canvas.

Problem 1 **(5 pts)**

Briefly describe the function of each part of Von Neumann Model.

The von Neumann model consists of five parts:

Memory: Where a computer stores information; 8 bits of info in each memory location | MAR - location register
MDR - data stored

Processing Unit: Carries out actual processing of information in the computer. Can consist of many complex unit

Input: How information gets into the computer. (eg mouse, scanner)

Output: How results of the processing is displayed outside the computer (eg printer, display)

Control Unit: Keeps track of where we are within the process of executing the program and where we are in the process of executing each function.

Problem 2 **(4 pts)**

Implement the function F using a 4:1 mux.
(Hint: The input signals to the MUX can A, B, C, 1,0)

$$F = \overline{A}\overline{B}C + \overline{A}\overline{B}\overline{C} + \overline{A}BC + A\overline{B}\overline{C}$$

(F = A’B’C + A’B’C’ + A’BC + AB’C’)

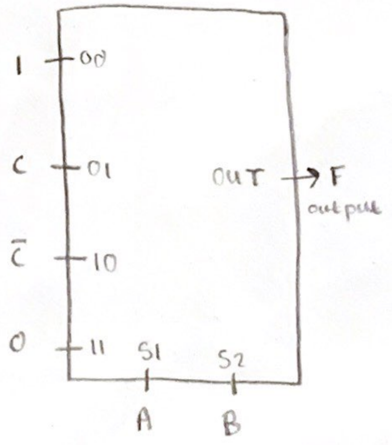
$F = \bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C} + \bar{A}BC + A\bar{B}\bar{C}$

$F = \bar{A}\bar{B}C + \bar{A}BC + \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C}$ commutative

$F = \bar{A}C(\bar{B}+B) + \bar{B}\bar{C}(\bar{A}+A)$ OR Distributive

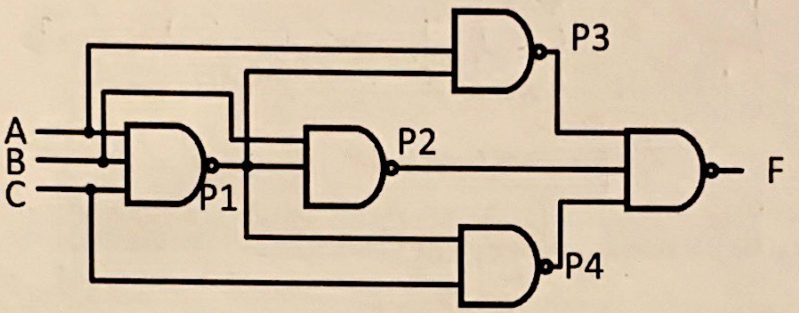
$F = \bar{A}C + \bar{B}\bar{C}$ complement

	A	B	C	F	
1	0	0	0	0	F = C
	0	0	1	1	
2	0	1	0	1	F = $\bar{C}$
	0	1	1	0	
3	1	0	0	0	F = 0
	1	0	1	0	
4	1	1	0	1	F = 1
	1	1	1	1	



Problem 3 (12 pts)

Consider the logic circuit in figure below:



1) Express the P1, P2, P3, P4, F in terms of A B, and C

$P1 = \overline{ABC}$

$P2 = \overline{B.\overline{ABC}}$

$P3 = \overline{A.\overline{ABC}}$

$P4 = \overline{C.\overline{ABC}}$

$F = \overline{\overline{A.\overline{ABC}}.\overline{B.\overline{ABC}}.\overline{C.\overline{ABC}}}$

$F = A\bar{C} + B\bar{A} + B\bar{C} + C\bar{B}$

2) Finish the truth table

XOR

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

3) Analyse the function of the circuit

The circuit functions in accordance with a XOR system, that is, it is only active when the inputs differ. Consequently, when the inputs are all the same (i.e. 000, or 111),* the circuit is inactive. Note that this does not mean this is the same as a three input XOR system.

*(ABC = 000, 111)

Problem 4 **(12 pts)**

Consider the following combinational logical circuit. There are 3 binary inputs (A, B, C) and one output (F). When there are odd 1s in the input, the output is 1, otherwise 0.

1) Complete the truth table of the circuit

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

F =

→ $\bar{A}\bar{B}C$

+

→ $\bar{A}B\bar{C}$

+

→ $A\bar{B}\bar{C}$

+

→ ABC

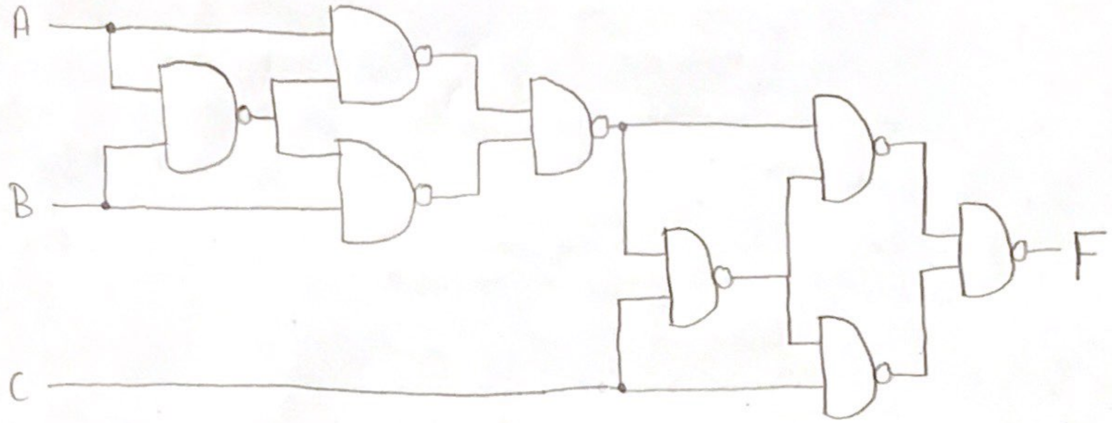
2) Express F in terms of A B, and C

$F = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$

↳

$F = A \oplus B \oplus C$

3) Draw this logical circuit using NAND gates



Problem 5: **(12 pts)**

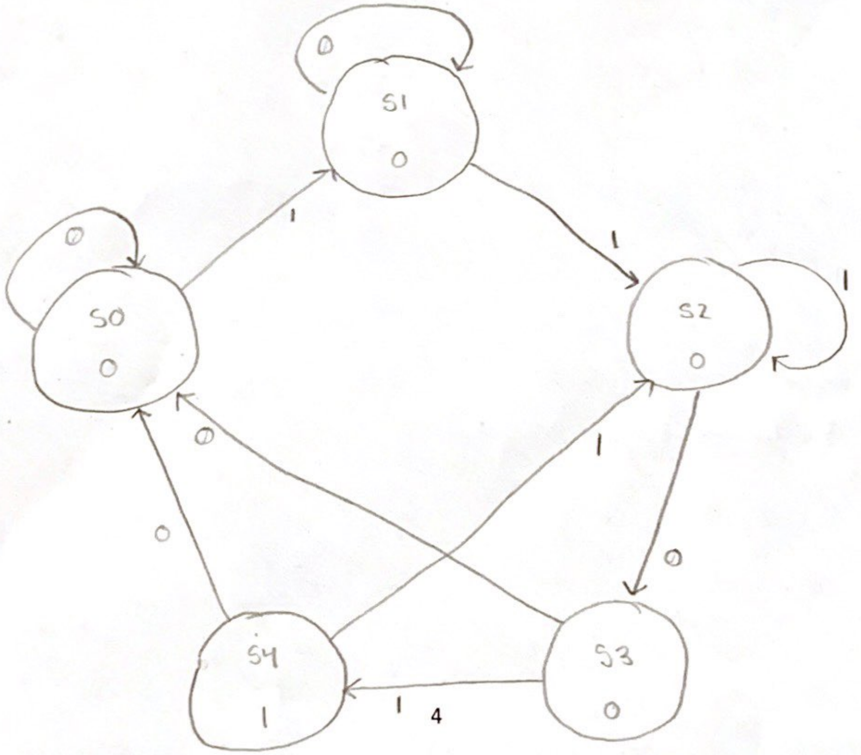
Consider a finite state machine (FSM) that recognizes the pattern "1101". The input to the FSM is a sequence of binary bits, one after the other. When the FSM receives 1101 as inputs in successive bits, it should output 1. Otherwise, the output should be 0.

For example

Input sequence: 11011011001101

FSM output: 00010010000001

1) Draw the state diagram for this FSM and describe what each state represents.



2) How many bits are required to represent all the states?

3 bits

S0 - 000 , S1 - 001 , S2 - 010 , S3 - 011 , S4 - 100

3) Draw the **output truth table** for this FSM .
(e.g. if you need 2 bits (Q1, Q0) to represent all the states, then the truth table has 3 columns (Q1, Q0|output))

S2	S1	S0	Output
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1

4) Draw the **next state** logic truth table for this FSM.
(e.g. if you need 2 bits (Q1, Q0) to represent all the states, then the truth table has 5 columns (input, Q1, Q0|Q1, Q0))

S_0' = next state (prime)

Input	S_2	S_1	S_0	S_2'	S_1'	S_0'
0	0	0	0	0	0	0
1	0	0	0	0	0	1
0	0	0	1	0	0	1
1	0	0	1	0	1	0
0	0	1	0	0	1	1
1	0	1	0	0	1	0
0	0	1	1	0	0	0
1	0	1	1	1	0	0
0	1	0	0	0	0	0
1	1	0	0	0	1	0