Mechatronics 3TB4 Winter 2018
Assignment 1- solution

Due in class on Friday Feb. 2 before start of the lecture
Maximum Marks 65

Question 1 [4(1+1+2)]

a) Controller Area Network (CAN) uses a bit encoding scheme called ...NRZ (Non Return to Zero)..

b) Bit-stuffing is used to facilitate..Synchronization.....between a transmitter and a receiver.

c) A node in a CAN wants to send the following sequence of bits: (space used for ease of reading only)

```
00 0001 1110 1100 0011
```

Show the bit sequence after suitable stuffing where necessary, according to CAN protocol.

```
00000 1 1111 0 011000011
```

Question 2 [12)]

a) CAN nodes A, B and C start transmitting at the same time. In the table below we give the Start Of Frame (SOF) bit and the 11 bit Identifer for the message each node is trying to transmit.

(a) Fill in the "Bus value" in the table below i.e., the value that gets transmitted on the bus assuming that all the nodes are sensing the value transmitted and stop transmitting if they see a conflict with the value they are transmitting (i.e. the nodes are respecting the CAN bus protocol). [3]

(b) Which of the three devices wins arbitration process? [2]
A

Device	SOF	Identifier									
A	0	1	0	1	0	0	1	1	0	1	1	0
B	0	1	0	1	1	0	1	1	0	1	1	0
C	0	1	0	1	0	0	1	1	1	1	1	0
											time	
Bus value	0	1	0	1	0	0	1	1	0	1	1	0

Figure 1:

continued on page 2

b) Explain with reference to your answer to question (2a) what is meant by the non-destructive arbitration property of CAN.[2]

- Multiple nodes can begin transmitting at the same time
- Because of the nature of the dominant and recessive bit scheme, the higher priority node will send a dominant bit at when the lower priority node transmits a recessive bit
- The lower priority node will see that its recessive bit got overwritten by a dominant bit and stop trying to transmit
- The higher priority node continues its transmission, unaffected by the lower priority node - i.e. the contention (dispute) for resources was settled (arbitrated) without a loss of information

c) The CAN specification does not specify the physical layer. It could be implemented by many different means. Suppose we wanted to create CAN network using hydraulics (water). All of the CAN nodes are connected to a water pipe connected to a tap. Each CAN node has a valve that can open or close. When a node opens its valve, the pipe empties. When all valves are closed, the tap fills up the pipe. What state of the pipe should be used to represent a logic 0 (dominant bit) on the hydraulic CAN bus? Full, or Empty? What should the default state of all the node's valves be?[5]
Ans:
The recessive state is when the pipe is full since if any node opens its valve, then the state changes to empty (i.e. valve open = pipe empty overrides valve closed = trying to fill the pipe.

Question 3 [9] The following truth table represents two Boolean functions: E and F

X	Y	Z	E	F
0	0	0	0	1
0	0	1	1	0
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	1	0
1	1	1	0	1

1. List all possible minterms and maxterms that can be constructed from this truth table. [2]
Ans:
Minterms are :

$$\bar{X}\bar{Y}\bar{Z},\ \bar{X}\bar{Y}Z,\ \bar{X}Y\bar{Z},\ \bar{X}YZ,\ X\bar{Y}\bar{Z},\ X\bar{Y}Z,\ XY\bar{Z},\ XYZ$$

Maxterms are:

$(\bar{X}+\bar{Y}+\bar{Z})$, $(\bar{X}+\bar{Y}+Z)$, $(\bar{X}+Y+\bar{Z})$, $(\bar{X}+Y+Z)$, $(X+\bar{Y}+\bar{Z})$, $(X+\bar{Y}+Z)$, $(X+Y+\bar{Z})$, $(X+Y+Z)$

continued on page 3

2. Express E and F as sum of minterms (algebraic form). [2]

$$E = \bar{X}\bar{Y}Z + \bar{X}Y\bar{Z} + X\bar{Y}\bar{Z} + XY\bar{Z}$$
$$F = \bar{X}\bar{Y}\bar{Z} + \bar{X}Y\bar{Z} + X\bar{Y}\bar{Z} + XYZ$$

3. Express E and F as product of maxterms (algebraic form). [2]

$$E = (X + Y + Z)(X + \bar{Y} + \bar{Z})(\bar{X} + Y + \bar{Z})(\bar{X} + \bar{Y} + \bar{Z})$$

$$F = (X + Y + \bar{Z})(X + \bar{Y} + \bar{Z})(\bar{X} + Y + \bar{Z})(\bar{X} + \bar{Y} + Z)$$

4. Minimize the function E in sum of minterms only, using K-map. [3]

```
X \ YZ
  |00 01 11 10
0 |0  1  0  1
  |          -
1 |1  0  0  1
   -         -
```

Combine the 1's underlined (x =1) to get:
$X\bar{Z}$
Combine the 1's underlined in vertical column under 10 to get:
$Y\bar{Z}$
There is a 1 under 01 in row 1 that remains and gives:
$\bar{X}\bar{Y}Z$
The minimized function is:

$$E = \bar{X}\bar{Y}Z + Y\bar{Z} + X\bar{Z}$$
$$\bar{Z}(X + Y) + \bar{X}\bar{Y}Z$$

Question 4 [10] A light in a staircase is controlled by two switches - one at the bottom of the staircase and the other at the top. If any of the switches is in the ON position while the other is in OFF position, the light is ON. If both switches are either in ON position or in OFF position, the light is OFF.

1. Write a truth table that describes the state of the light as a function of two switches as input.[3]

```
Truth Table:
x1 x2 f(x1, x2)
0   0    0
0   1    1
1   0    1
1   1    0
```

continued on page 4

2. Express the truth table as a function, if x1 and x2 represent switches then $f(x1, x2) =?$ [3]

```
Using sum of minterms:

f(x1, x2) = x1' x2 + x1 x2'
```

3. Write Verilog code to implement the function.[4]

```
module light (x1, x2, f);
input x1, x2;
output f;
assign f = (x1 & ~x2) | (~x1 & x2);
endmodule
```

Question 5 [3 + 2 + 2 + 2 + 3 + 3]

- A 7 Segment LED display generally has 8 input connections, one for each LED segment and one that acts as a common terminal. There are 2 types of 7 Segment LED digital display.
  1. Common Cathode Display all the cathode connections of the LEDs are connected to ground. A logic '1' applied to the anode terminal of the individual segment illuminates it.
  2. Common Anode Display all the anode connections of the LEDs are connected to VCC. A logic '0' applied to the cathode terminal of the individual segment illuminates it.

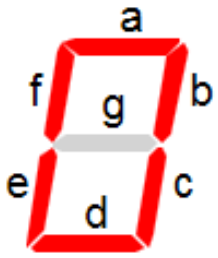

Figure 2:

- A BCD to Seven Segment decoder is a combinational logic circuit that accepts a decimal digit in BCD (input) and generates appropriate outputs for the segments to display the input decimal digit.
- Goal:
  Design a BCD to 7 segment decoder circuit for only one segment (e) of a common cathode 7-segment display. The decoder has a 4-bit binary input and a single output (e) specified by a truth table.

continued on page 5

a) Construct a suitable truth table. [3]

```
Truth Table:
D  C  B  A   e
0  0  0  0   1
0  0  0  1   0
0  0  1  0   1
0  0  1  1   0
0  1  0  0   0
0  1  0  1   0
0  1  1  0   1
0  1  1  1   0
1  0  0  0   1
1  0  0  1   0
1  0  1  0   0
1  0  1  1   0
1  1  0  0   0
1  1  0  1   0
1  1  1  0   0
1  1  1  1   0
```

b) Drive a Boolean function as sum of minterms. [2]

```
Can be represented by a Boolean function:
e= D'C'B'A' + D'C'BA' + D'CBA' + DC'B'A'
```

c) Minimze the Boolean function using Boolean Algebra. [2]

```
e= D'C'B'A' + D'C'BA' + D'CBA' + DC'B'A'
  = C'B'A'(D + D') + D'BA'(C + C')
  = C'B'A' + D'BA'
```

d) Use K-map to derive a minimized function. [2]

```
Draw a K- map from the TT

 DC\BA
     | 00| 01| 11 | 10
----|---|----|-----|----
 00| 1 | 0   | 0   | 1
 01| 0 | 0   | 0   | 1
 11| 0 | 0   | 0   | 0
 10| 1 | 0   | 0   | 0
```

continued on page 6

```
------------------------
```

From which we can directly get. by combining two 1's in the first column of rows 1 and 4; and 4th column of rows 1 and 2.

```
  e = C'B'A' +  D'BA'
```

e) Draw a schematic diagram using gate level components to implement the function. [3]

continued on page 7

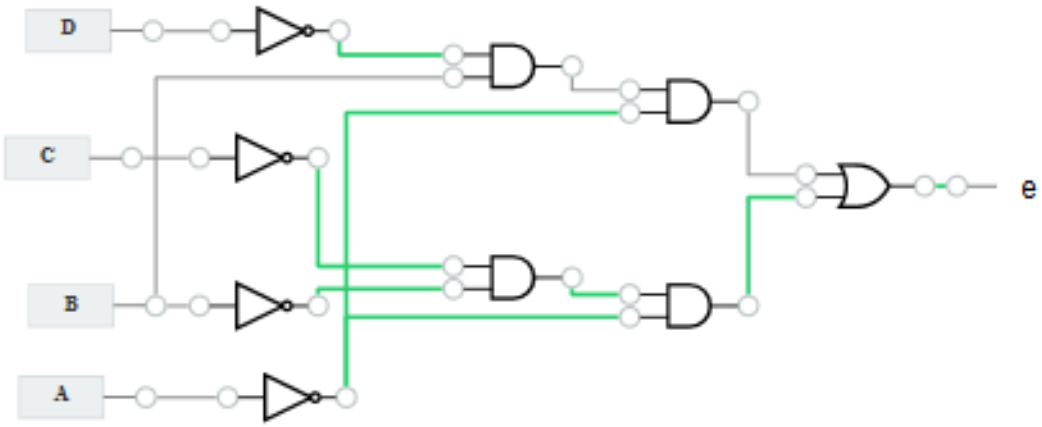

Figure 3:

continued on page 8

f) Describe the circuit using Verilog HDL code. [3]
Verilog Code:

```
Gate-level implementation:
--------------------------
module e_segment(input D, C, B, A, output e);

  wire im1, im2, im3, im4, im5, im6, im7, im8;

 not n1(im1, D);
 not n2(im2, C);
 not n3(im3, B);
 not n4(im4, A);
 and a1(im5, im1, B);
 and a2(im6, im2, im3);
 and a3(im7, im5, im4);
  and a4(im8, im6, im4);
 or o1(e, im7, im8);
endmodule

Data-flow implementation:
-------------------------
module e_segment(input D, C, B, A, output e);

 wire im1, im2;

  assign im1 = ~D;
  assign im2 = ~C;
  assign im3 = ~B;
  assign im4 = ~A;
  assign im5 = im1 &  B;
  assign im6 = im2 & im3;
  assign im7 = im5 &  im4;
  assign im8 = im6 &  im4;
  assign e = im7 |  im8;

endmodule
```

Question 6 [3 + 2 + 2 + 2 + 3 + 3] Design Problem

A bank wants to install an alarm system with movement sensors. It has 3 sensors (A,B,C) as shown in figure 4.

To prevent false alarms produced by a single sensor activation, the alarm will be triggered only when at least two sensors activate simultaneously.

continued on page 9

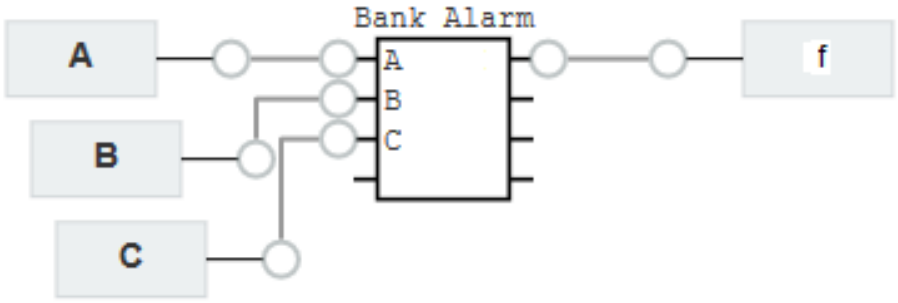

Figure 4:

a) Create a truth table that describes the intended functionality of the logic circuit.

A	B	C	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

b) Derive a Boolean function in the sum-of-minterms form.

$$f = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

c) Minimize the function using Boolean Algebra rules/theorms.

$$\begin{aligned} f &= BC(\bar{A} + A) + AB\bar{C} + A\bar{B}C \\ &= BC + AB\bar{C} + ABC \\ &= B(C + \bar{C}A) + A\bar{B}C \\ &= B(C + A) + A\bar{B}C \\ &= BC + BA + A\bar{B}C \\ &= BC + A(B + \bar{B}C) \\ &= BC + AB + AC = AB + BC + CA \end{aligned}$$

d) Minimize the function using K-map technique.

continued on page 10

```
C\ AB
  | 00  01  11  10
0 | 0   0   1   0
1 | 0   1   1   1

Grouping the two
 right most 1's of bottom row - AC
Left two 1's of bottom row - BC
Columns under 11 - AB
```

The minimized function is

$$f = AB + BC + AC$$

e) Draw a schematic diagram using suitable logic gates to implement the simplified function.

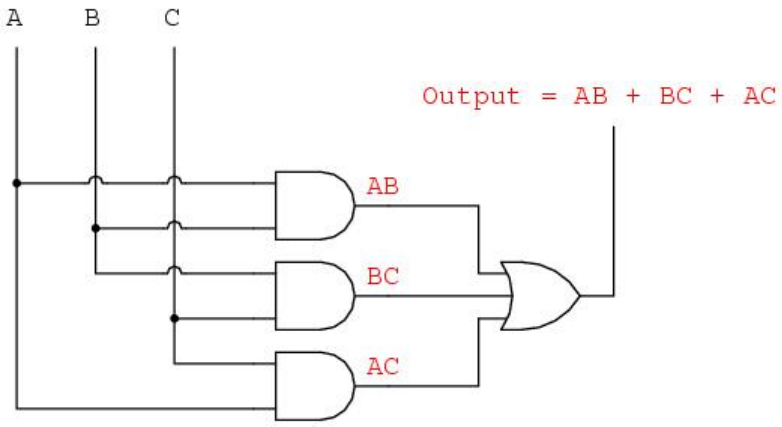

Figure 5:

f) Write Verilog HDL code that can be used to simulate this logic circuit.

```
module sensor-output(input A, B, C, output O);

  wire im1, im2, im3;
  and a1(im1, A, B);
  and a2(im2, B, C);
  and a3(im3, A, C);

  or(O, im1, im2, im3);
endmodule
```

THE END.