UNIVERSITY OF CALIFORNIA

Department of Electrical Engineering and Computer Sciences

EECS 150 Fall 2001

Prof. Subramanian

Midterm II

1) You are implementing an 4:1 Multiplexer that has the following specifications:

Inputs: I_0-I_3

Output: Z

Control Inputs: C_2-C_0 (C_2 is MSB)

The input is selected based on a Johnson counter scheme (000->I_0, 100->I_1, 110->I_2, 111->I_3).

All other states select I_0.

a) Write out the functional form of the truth table for this multiplexor.

C_2	C_1	C_0	Z	I_m
0	0	0	I_0	1
0	0	1	I_0	0
0	1	0	I_0	1
0	1	1	I_0	0
1	0	0	I_1	0
1	0	1	I_0	0
1	1	0	I_2	1

1 1 1 I_3 1

$I{-}_0 = C_2 + C_1C_3$

$I_1 = C_2C_1C_0$

$I_2 = C_2C_1C_0$

$I_3 = C_2C_1C_0$

b) Determine the output Boolean function. You should simplify or minimize.

$Z = (C_2 + C_1C_0)I_0 + (C_2C_1C_0)I_1 + (C_2C_1C_0)I_2 + (C_2C_1C_0)I_3$

$Z = C_2I_0 + C_1C_0I_0 +$

c) Implement the multiplexer using the PLA below. Indicate connections using the standard cross scheme used in class. Indicate the inputs, product terms, and outputs on the diagram.

```
I0    I1    I2    I3    C2   C1    C0

|     |     |     |     |    |     |    (and gates)

•▽▽• ▽• ▽•▽▽• ▽•     ⇓

x-|---|--|---|--|---|--|---|-x--|--|---|--|----[   )---x------- C2I0

x-|---|--|---|--|---|--|---|-|---|-x---x-|----[   )---x-------  C1C0I0

 |-|---x-|---|--|---|--|---x-|---|-x---|-x----[   )---x-------  C2C1C0I1

 |-|---|--|---x-|---|--|---x-|---x-|---|-x----[   )---x-------  C2C1C0I2

 |-|---|--|---|--|--x--|---x-|---x-|---x-|----[   )---x-------  C2C1C0I3

                                                |

                                                ∪

                                                | |   (or gate)
```

u

|

Z

d) Implement F(A,B,C) = $\Sigma m(4, 7)$ using the above multiplexer. Indicate the connections in the diagram below. Simplify as appropriate.

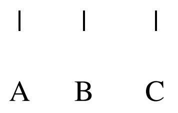
Select I_1, I_3

```
      ________________
0 ---| I0             |
1 ---| I1   4:1 MUX   |
0 ---| I2             | →
1 ---| I3             |
     |__C2__C1__C0__  |
         |   |   |
         A   B   C
```

e) Implement G(A,B,C) = $\Sigma m(0, 2, 6, 7)$ using the above multiplexer. Indicate the connections in the diagram below. Simplify as appropriate.

Select I_2, I_3

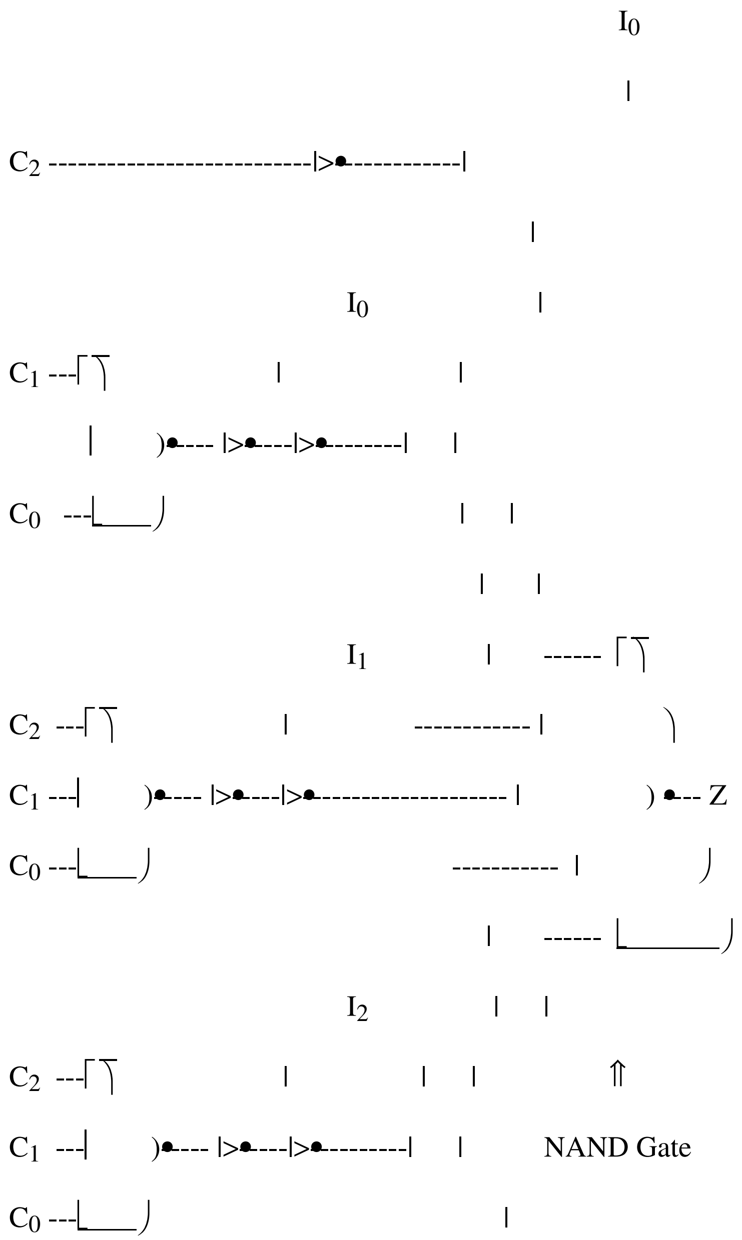
```
      ________________
0 ---| I0             |
0 ---| I1   4:1 MUX   |
1 ---| I2             | →
1 ---| I3             |
     |__C2__C1__C0__  |
```

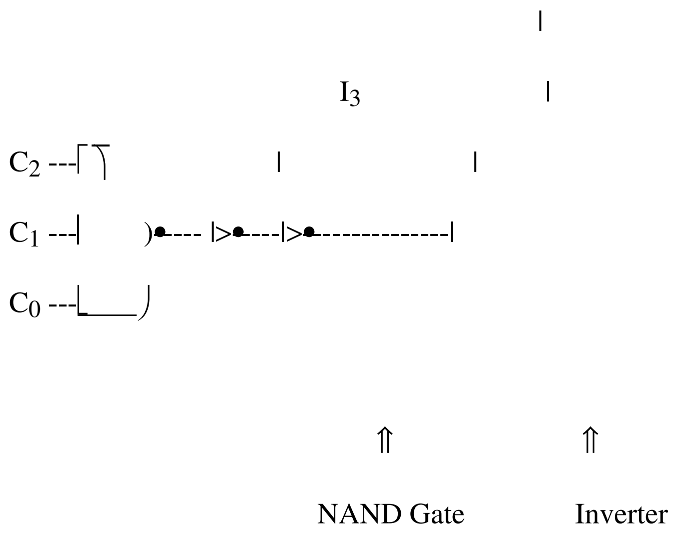


f) Implement the multiplexer using NAND gates and Tri-State Inverters. Draw the circuit below.

Use inverters active high

$Z = C_2I_0 + C_1C_0I_0 + C_2C_1C_0I_1 + C_2C_1C_0I_2 + C_2C_1C_0I_3$





g) Assume the Tri-State Inverters require 4 transistors. How many transistors are required to implement the multiplexer:

i) Using Tri-State Inverters / NAND gates:

ii) Using NAND / NAND gates:

2) Back in the days of mainframe computing, there were two standard for textual information interchange ASCII and EBCDIC. You are implementing a counter that counts in hexadecimal. The output of the counter is either in ASCII (0-9 = 48-57, A-F = 65-70) or EBCDIC (0-9 = 240-249, A-F = 193-198) depending on the value of a control switch (C = 0 -> ASCII, C= 1 -> EBCDIC). You are provided with the following parts:

A 4-bit binary counter (Pins: CLK, RESET, C3-C0 outputs)

A 4:16 DEMUX (Pins: G enable, S3-S0 select lines, Z output)

A 32x8 bit ROM (Pins: A4-A0 address lines, D7-D0 data lines)

As many NAND gates as you need

You may use some or all of the parts above. You may choose to program the ROM with whatever values you need.

a) Design the ASCII/EBCDIC counter. Use block diagrams for the individual parts and label pinouts.

```
                              control
       4-bit                    bit
      counter                  switch      ROM
  -----------                 ↓    ----------------
  |          |  C[3:0]        A4 ----|              |        D[7:0]
  |          |\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\|            |\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
  |    Λ  |                A[3:0]   |              |           ↑
  -----------                       ----------------      output
   /     |
reset   clk
```

C = A4

b) Fill out the following table with the values that you chose to store in the ROM. You should write the data in decimal for convenience (and ease of grading).

Address	Data	Address	Data	Address	Data	Address	Data
00000	48	01000	56	10000	240	11000	248
00001	49	01001	57	10001	241	11001	249
00010	50	01010	65	10010	242	11010	193

00011	51	01011	66	10011	243	11011	194
00100	52	01100	67	10100	244	11100	195
00101	53	01101	68	10101	245	11101	196
00110	54	01110	69	10110	246	11110	197
00111	55	01111	70	10111	247	11111	198

3) Consider the following truth table:

Inputs			Output
A	B	C	Z
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0

```
| 1      | 0 |  0 |  0        |

-------------------------

| 1      | 0 |  1 |  1        |

-------------------------

| 1      | 1 |  0 |  1        |

-------------------------

| 1      | 1 |  1 |  1        |

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

a) Minimize the truth table using a K-Map. Identify the prime implicants and essential prime implicants on the map.

```
A\BC   00  01  11  10

        ----------------

0       | 0  | 0  |  1 |  0 |

        ----------------

1       | 1  | 0  |  1 |  1 |

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

B = 1, C = 1, A = anything: essential prime implicants

A = 1, B = 1, C = anything: prime implicants

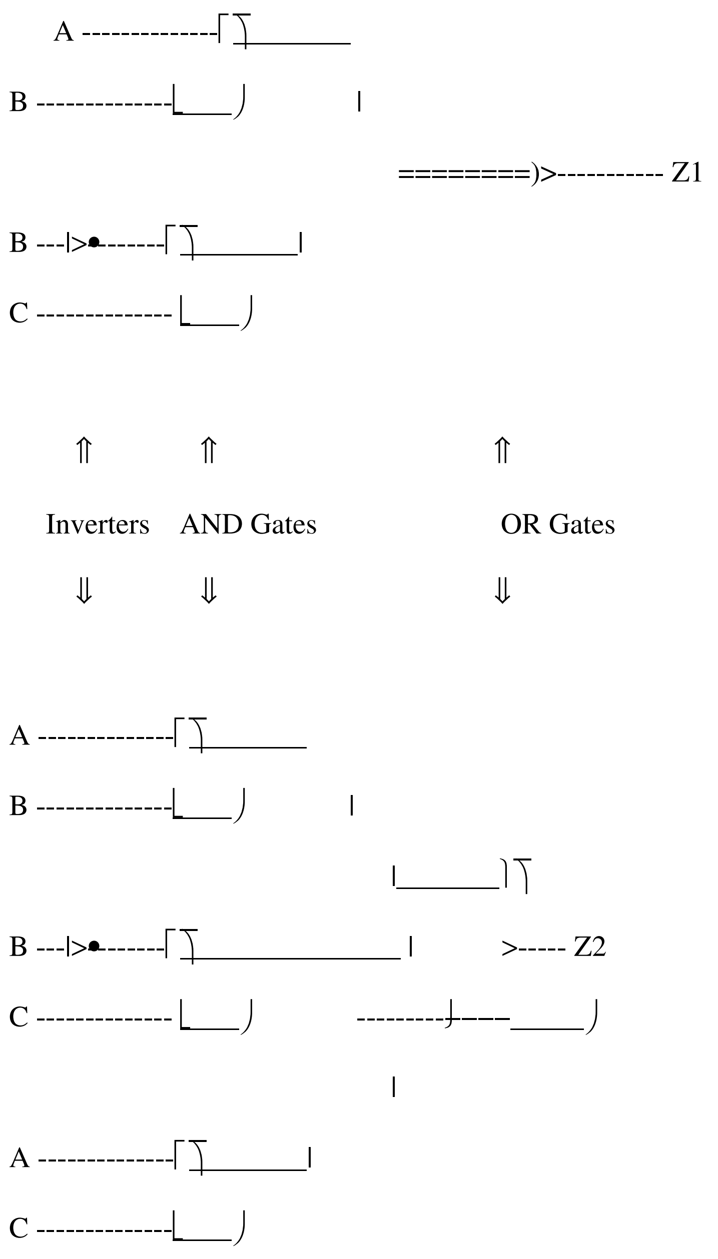
A = 1, C = 0, B = anything: essential prime implicants

b) Write a minimal SoP function (Z1) that describes the above truth table, and an SoP function (Z2) that has product terms covering all the prime (essential and non-essential) implicants.

Z1(A,B,C) = AB + BC

Z2(A,B,C) = AB + BC +AC

c) You are not provided with the complemented inputs, so you will need to use inverters as necessary. Draw circuits that implements Z1 and Z2 using AND gates, OR gates, and Inverters.



d) Complete the timing diagram below, that shows the transition from ABC = 111 to ABC = 101 for both Z1 and Z2. Assume the propagation delay is 5ns, independent of gate type.

```
A  |  |_________________________________________
   |
B  |  |_________________________________________
   |
C  |  |_________________________________________
   |       1                                    1
Z1 | |__0___|  
   |_____________1_______________________________________
Z2 |_______________________________________________________
   0      5      10     15     20     25     30     40     45
```

4) Consider a finite state machine defined by the following table (X= dont care):

Present State	Next State		Output	
	IN=0	IN=1	IN=0	IN=1
A	B	D	0	0
B	B	F	0	0
C	B	F	X	1
D	A	F	1	X

```
----------------------------------------------------
|        E        |    F   |    F   |    0   |    0    |
----------------------------------------------------
|        F        |    A   |    F   |    X   |    1    |
----------------------------------------------------
```

a) Minimize the number of states using the implication chart below:

```
-------
B  | B-B  |
| D-F  |
-------------
         C  | \ /  | \ / |
| / \  | / \ |
-------------------
         D  | \ /  | \ / | A-B |
| / \  | / \ | F-F  |
-------------------------
                  E   | B-F  | B-F | \ /  | \ /  |
| D-F  | F-F | / \  | / \  |
-------------------------------
         F   | \ /  | \ / | A-B | A-A | \ / |
| / \  | / \ | F-F  | F-F  | / \ |
-------------------------------
   A     B    C     D     E
```

List the equivalencies:

A = B, C = F, D = F, C = D

b) Now, suppose state F is changed such that the output is 0 when IN=1. Repeat your simplification under these new conditions.

Present State	Next State		Output	
	IN=0	IN=1	IN=0	IN=1
A	B	D	0	0
B	B	F	0	0
C	B	F	X	1
D	A	F	1	X
E	F	F	0	0
F	A	F	X	0

```
B   | B-B  |

| D-F  |

-------------

          C   | \ /   | \ / |

| / \   | / \ |

-------------------

          D   | \ /   | \ / | A-B |

| / \   | / \ | F-F  |

-------------------------

          E   | B-F  | B-F | \ /  | \ /   |

| D-F  | F-F |  / \  | / \   |

-------------------------------

          F   | A-B  | A-B| \ /  | A-A | A-F |

| D-F  | F-F |  / \  | F-F  | F-F  |

                         --------------------------------

                            A      B     C      D      E
```

List the equivalencies:

A = B, C = D, A = E, A = F, B = E, B = F, D = F, F = E

c) Is your simplification in part B unique? Are there other possible equivalencies? If so, list them. If not, give reasons for your answer.

Part A: A = B, C = F, D = F, C = D

Part B: A = B, C = D, A = E, A = F, B = E, B = F, D = F, F = E

No, since weve simplified as much as possible using the diagram.