Your Name: ___________________________ **SID Number:** ______________________

UNIVERSITY OF CALIFORNIA AT BERKELEY

BERKELEY • DAVIS • IRVINE • LOS ANGELES • RIVERSIDE • SAN DIEGO • SAN FRANCISCO SANTA BARBARA • SANTA CRUZ

Department of Electrical Engineering
and Computer Sciences

CS 150 - Spring 1998
Prof. A. R. Newton
Prof. K. Pister

(1)	/30
(2)	/30
(3)	/40
TOTAL	**/100**

Quiz 2 Solutions

Room 10 Evans Hall, 2:10pm Tuesday April 7th
(Open Katz only, Calculators OK, 1hr 20mins)

Include all final answers in locations indicated on these pages and in pen. Use space provided for all working. If necessary, attach additional sheets by staple at the end. BE SURE TO WRITE YOUR NAME ON EVERY SHEET.

1. **(30pts) (a)** A digital system is required to amplify a binary-encoded audio signal. The user should be able to control the signal amplitude **from minimum to maximum in 100 increments**. What is the **minimum number of binary bits** required to encode the user-specified amplitude?

   (b) *Excess-3 code* (Katz page 499) is a variation of binary-coded decimal (BCD) code. Each decimal digit is represented by a 4-bit code that is **three more than the associated BCD code**. For example, 0_{10} is encoded in excess-3 as 0011_2, 1_{10} is encoded in excess-3 as 0100_2, etc. Design a **single-output combinational logic circuit** that **outputs a 1 when the input to the circuit in 4-bit excess-3 code is a prime number**. For all other (non-prime) **legal** 4-bit excess-3 numbers applied to the inputs, the output is a 0. **Assume complement inputs are available** and implement the circuit using:

   (i) One **16-input, four control-line** multiplexer **only**.

   (ii) One **8-input, three control-line** multiplexer **only**.

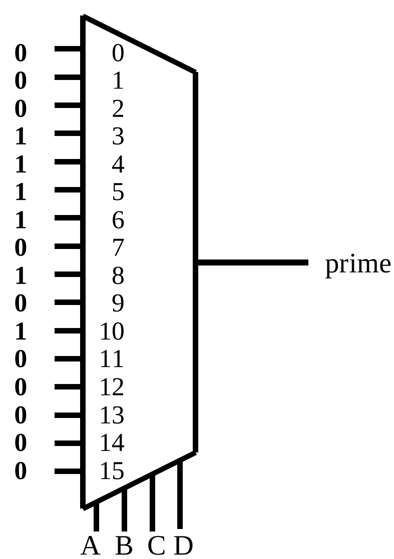
   (iii) One **4-input, two control-line** multiplexer and a **minimum number of simple logic gates** (INV, NAND, NOR, AND, OR, XOR, XNOR)

1. (a) (5pts)

Number of bits = _2^6=64 and 2^7=128 so seven bits are required_____

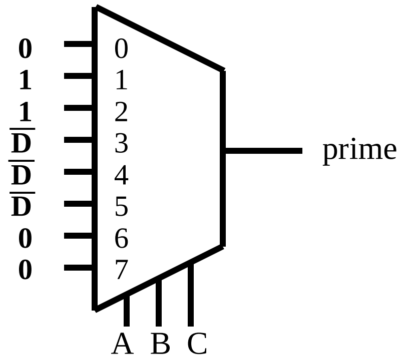
1. (b) (i) (5pts) One 16-input MUX

Decimal	A	B	C	D	Prime
	0	0	0	0	–
	0	0	0	1	–
	0	0	1	0	–
0	0	0	1	1	1
1	0	1	0	0	1
2	0	1	0	1	1
3	0	1	1	0	1
4	0	1	1	1	0
5	1	0	0	0	1
6	1	0	0	1	0
7	1	0	1	0	1
8	1	0	1	1	0
9	1	1	0	0	0
	1	1	0	1	–
	1	1	1	0	–
	1	1	1	1	–

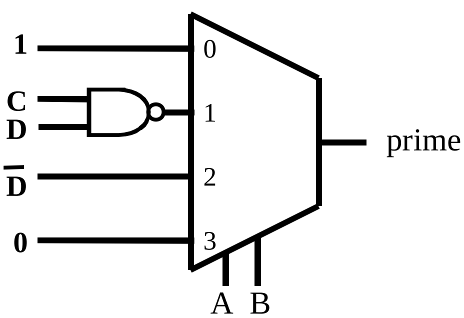


(additional space for solutions on reverse)

1. (b) (ii) (10pts) One 8-input MUX



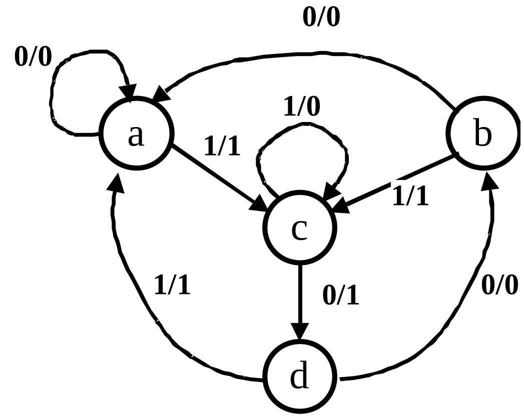
1. (b) (iii) (10pts) One 4-input MUX and logic gates



Additional space for Problem 1

Your Name: ___________________________

(2) (30 points)



(a) What is the better way to implement arithmetic in a binary computer: **one's-complement or two's-complement**? Why? **Include all of the arguments you can think of for and against your answer.**

(b) Does the state-machine opposite have any **equivalent states**? If so, **which states are equivalent**? Show all working.

(c) Design a **4-bit ripple up-counter using positive edge-triggered D flip flops** and a minimum number of combinational logic gates. Show the schematic diagram.

(a) (10pts) One's or two's complement? Why?

____2's complement______________________________________

____+ easier to do addition and subtraction (sum of -ve #s) (can be done in one byte-serial pass)

____+ single representation for 0______________________________________

____- must add 1 for making -ve #s______________________________________

____

____

____

____

____

____

(continue on reverse if necessary)

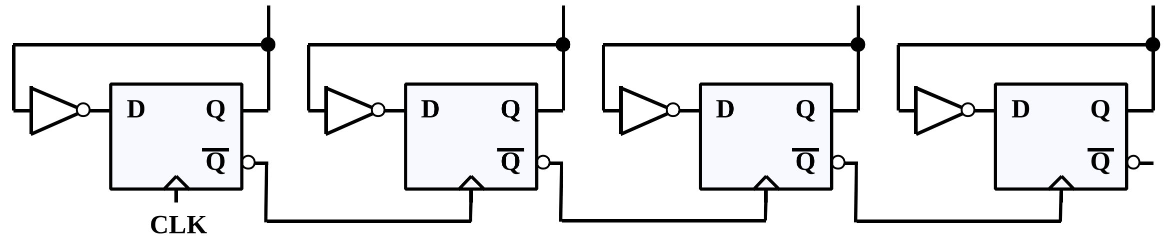
(b) (10pts)

Equivalent states: ___a = b_______________

b	a o a c o c		
c	X	X	
d	~~a o b a o c~~	~~a o b a o c~~	X
	a	b	c

(additional space for solutions on reverse)

2.(c) (10pts) 4-bit ripple up-counter



Additional space for Problem 2

Your Name: ___________________________

3. **(40 pts)** Design a Moore machine that detects the number 9_{10} encoded in binary (1001_2) and in excess-3 (1100_2). The **machine should reset after each detection** (i.e. overlapping sequences are ignored). Sample input (X) and output (Z) sequences are given below:

```
X=1001100011001…
Z=0001000000010…
```

(a) Draw a **state transition diagram for the Moore** machine.

(b) Use an **implication table** to determine if any states are equivalent. If so, **list the equivalent states** and then **re-draw your (now minimized) state diagram.**

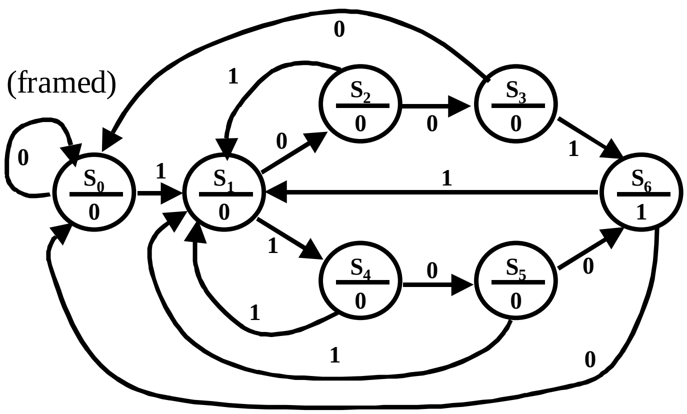
(c) Using **D flip-flops and the state-assignment rules discussed in class**, indicate **all adjacency constraints** for an optimal state encoding and **determine an optimal encoding**, listing the state codes for each state in the machine.

(d) **Write equations for the next-state logic only** (not the output logic) **using a minimal NOR-NOR** two-level representation.

3. (a) (10pts) State transition graph:

Original

One answer (framed)



After reduction
(if changed)

No change

3. (b) (5pts) Implication table:

normal table with no equivalent states

(5pts) Equivalent States: ______None___________

(additional space for solutions on reverse)

3. (c) (5pts) Adjacency constraints:

Rule 1: _______{S_0, S_3, S_6},{S_2, S_4, S_5}, {S_0, S_6} ____

Rule 2 _______{S_0, S_1},{S_2, S_4}, {S_0, S_3},{S_0, S_6}, {S_0, S_5} ______________________________

Rule 3: _______{S_0, S_1, S_2, S_3, S_4, S_5 }____

(5 pts) Optimal state assignment (*<u>show Karnaugh map</u>*)

Q_A \ Q_BQ_C	00	01	11 (Q_B)	10 (Q_B)
0	S_0 0	S_6 1 (Q_C)	S_5 3 (Q_C)	S_1 2
1	S_3 4	S_2 5 (Q_C)	S_4 7 (Q_C)	6

State codes: ___S_0=000, S_1=010, S_2=101, S_3=100, S_4=111, S_5=011, S_6=001_____

3. (d) (10pts) Next-state logic in NOR-NOR form (equations only!):

$$D_A = \overline{\overline{(Q_B + Q_C)} + \overline{(\overline{Q_C} + X)} + \overline{(\overline{Q_B} + \overline{Q_C})} + \overline{(Q_A + \overline{Q_C})}}$$

$$D_B = \overline{\overline{(Q_A + X)} + \overline{(\overline{Q_A} + Q_B)} + \overline{(\overline{Q_A} + \overline{X})} + \overline{(Q_A + \overline{Q_B} + \overline{Q_C})}}$$

$$D_C = \overline{\overline{(Q_A + Q_B)} + \overline{(Q_B + X)} + \overline{(\overline{Q_C} + \overline{X})}}$$

Additional space for Problem 3