**Question 1.** *True/False* (5 Points)

Circle T for true and F for false below (0.5 points each):

(i) The set of prime implicants of a Boolean function are unique. T

(ii) A PAL is a device with a fully programmable AND plane. T

(iii) A $2^N$:1 Multiplexer can implement any function of N variables. T

(iv) The Sum of Products form of a Boolean function will always have fewer literals than its Products of Sums form. T

(v) All Boolean function implementations have hazards in them. T

(vi) Combinational logic determines its outputs as a function of the the current inputs and the history of the computation. T

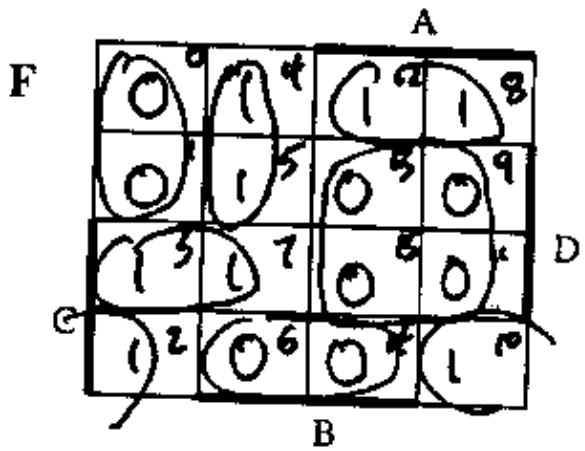
(vii) A ROM is nothing more than a hardware truth table. T

(viii) A circuit with state is an example of combinational logic. T

(ix) PALs are typically faster than PLAs. T

(x) A selector and a multiplexer are essentially the same hardware. T

**Question 2.** *Canonical Forms* (15 points)
Given the function F(A,B,C,D)=(A+B+C)(B'+C'+D)(A'+C+D')(A'+B'+C')(A'-
following questions. Use the K-map below before your intermediate work .




(i) Write F in *canonical* Product of Sums form using ΠM notation (3 points):

$$F(A,B,C,D) = \Pi M(0,1,6,9,11,13,14,15)$$

(ii) Write F in *canonical* Sum of Products form using Σm notation (2 points):

$$F(A,B,C,D) = \Sigma m(2,3,4,5,7,8,10,12)$$

(iii) Find the *minimum* literal count Product of Sums form of F (3 points):

$$F(A,B,C,D) = (A'+D')(B'+C'+D)(A+B+C)$$

(iv) Find the *minimum* literal count Sum of Products form of F (3 points):

$$F(A,B,C,D) = AC'D' + A'BC' + A'CD + B'CD$$
$$\text{OR} \quad AB'D' + BC'D' + A'BD + A'B'C$$

(v) Find the minimum literal count Product of Sums form of F' (2 points):

$$F'(A,B,C,D) = (A'+C+D)(A+B'+C)(A+C'+D')(\text{illegible})$$

**Question 3.** *Minimization Over Multiple Functions* (15 Points)
Given the following four minimized four-variable functions:

$$W(A,B,C,D) = AB^{1} + B'C'^{2}$$

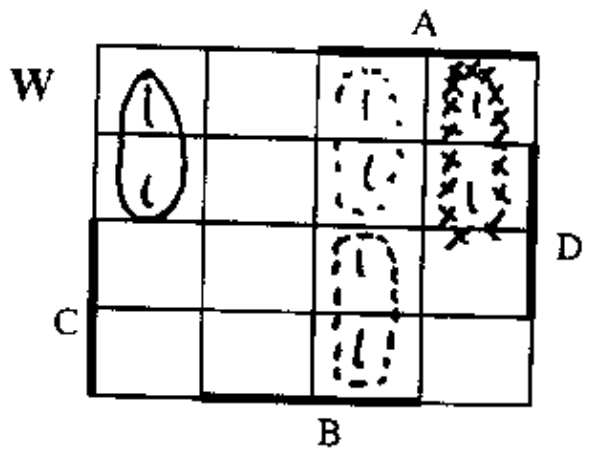
$$X(A,B,C,D) = A'BC^{3} + AB'C'^{4} + ABC^{5}$$

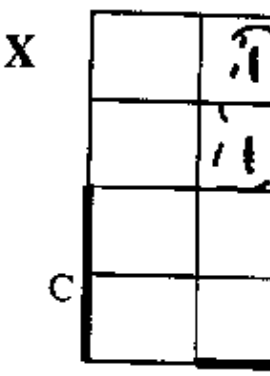
$$Y(A,B,C,D) = A'B'^{6} + A'C'^{7}$$

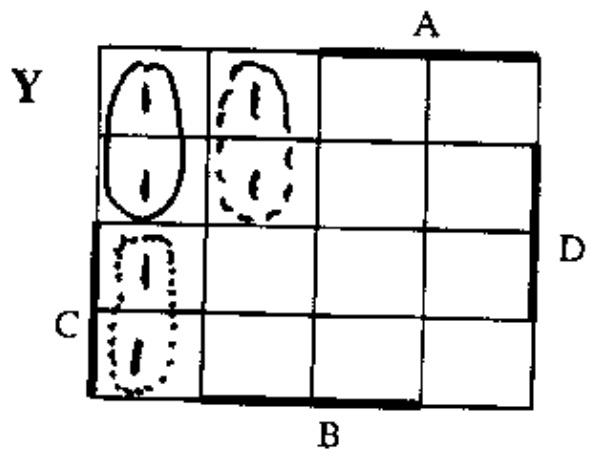
$$Z(A,B,C,D) = A'BC' + A'B'C^{8}$$

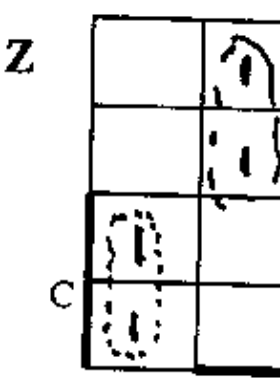
How many unique product terms are there spanning these functions (1 points): ____

Fill in the K-maps below and circle the implicants so as to yield the minimum nu
terms spanning the four functions (3 points per K-map and function):



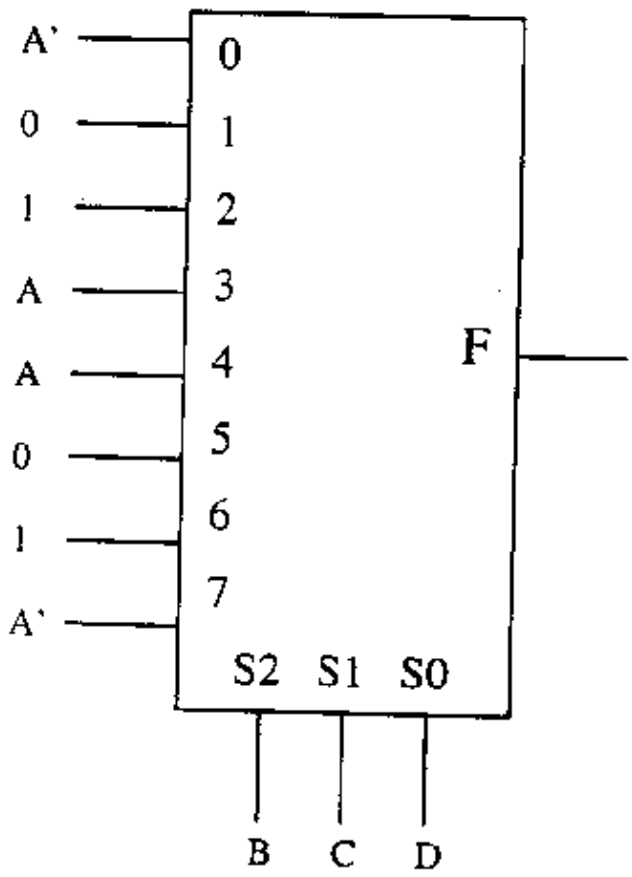




Write the revised expressions for W, X, Y, Z in the boxes below Array:

W = A'B'C' + ABC' + ABC + AB'C'

X = A'BC' + ABC + A

**Question 4.** *Multiplexer Implementation* (15 Points)
The following implements the four variable function F(A,B,C,D) using a choice of which variable is a data input and which are multiplexer control inp





Reverse engineer this function to write it down in minimized Sum of Products

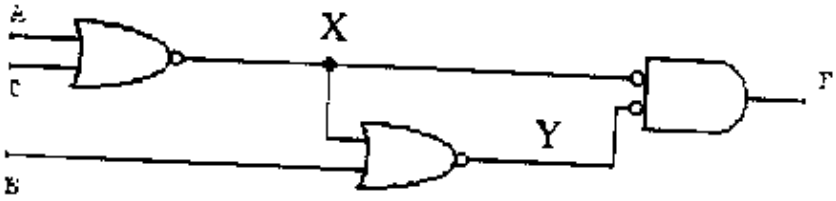
F(A,B,C,D)= $A'B'D' + A'BC + ABD' + AB'C$

What is your literal count? 12

Assume that you can use XOR and XNOR operators as well AND, OR, and NO expression for the function F in a *multilevel form* that has an even further reduc

$F(A,B,C,D) = (A \odot B)D' + (A \oplus$

**Question 5.** *Circuit Timing and Waveform Diagrams* (15 points)
Consider the following circuit schematic and timing waveform.




(i) Write F(A,B,C) as a multilevel function based on the above schematic (3 po

$$F(A,B,C) = \overline{X}\,\overline{Y}$$
$$X = \overline{A+C}$$
$$Y = \overline{X+B}$$

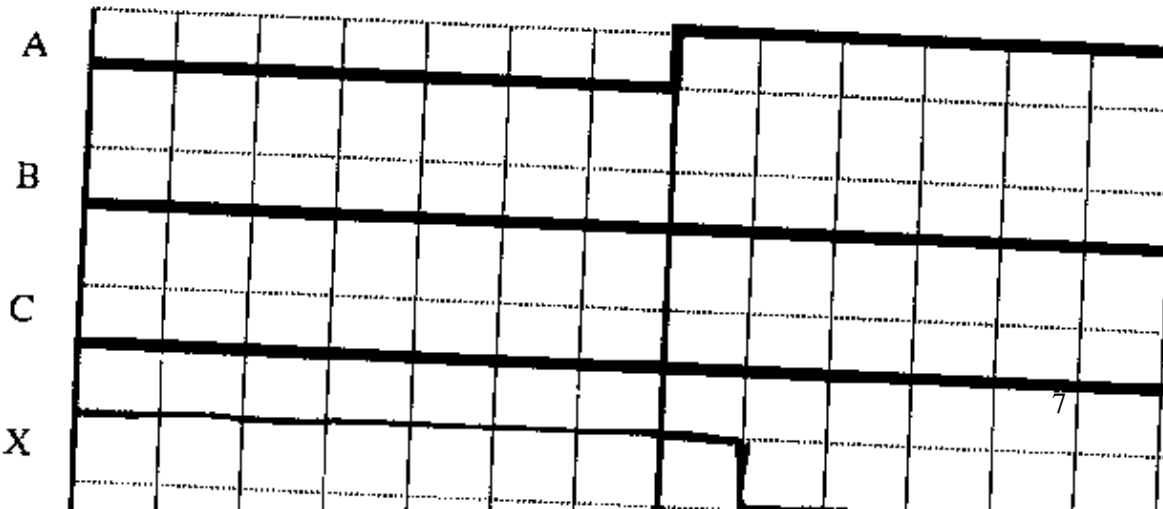
(ii) Write F(A,B,C) in *minimized* literal count Sum of Products form (3 points):

$$F = (A+C)(X+B) = (A+C)(A'C'+B)$$
$$= AB+BC$$

(iii) Write F(A,B,C) in *minimized* literal count Product of Sums form (3 points):

$$F = B(A+C)$$

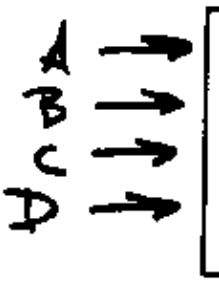
(iv) All gates have identical gate delays. Each time division represents a g
waveform diagram with the time behavior of output F and intermediate node
the inputs have not changed for quite some time before time T0 (5 points):



**Question 6.** *Design Problem* (15 Points)
Consider a subsystem that acts as a "tie detector". The function behaves as fol
are true than false, the output MoreOnes is asserted. If more of the inputs are fals
MoreZeros is asserted. If number of ones and zeros at the inputs are the
MoreZeros are false (that is, a tie has been detected).

Design a four-input tie detector subsystem.

(i) Identify your inputs and outputs. Draw a block diagram (2 points):




15
14
10
12
13
55
9
69

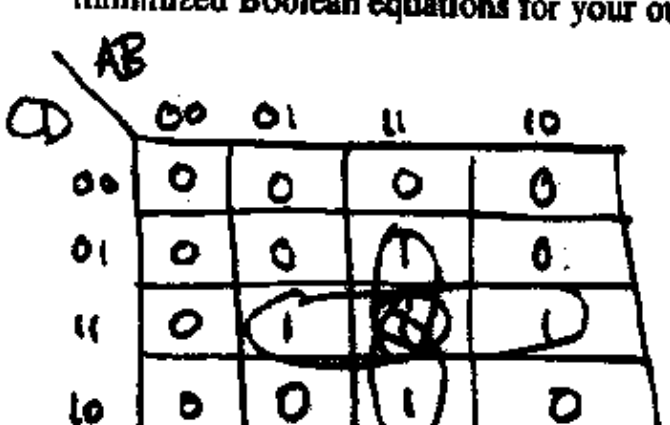
(ii) State your assumptions about the behavior of the circuit. Document your un
with a truth table (3 points):

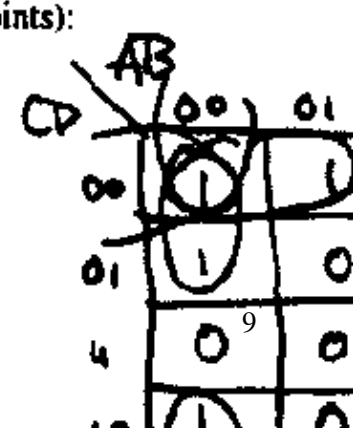
| A B C D | MO | MZ |
|---|---|---|
| 0 0 0 0 | 0 | 1 |
| 0 0 0 1 | 0 | 1 |
| 0 0 1 0 | 0 | 1 |
| 0 0 1 1 | 0 | 0 |
| 0 1 0 0 | 0 | 1 |
| 0 1 0 1 | 0 | 0 |
| 0 1 1 0 | 0 | 0 |
| 0 1 1 1 | 1 | 0 |

| A B C |
|---|
| 1 0 0 |
| 1 0 0 |
| 1 0 1 |
| 1 0 1 |
| 1 1 0 |
| 1 1 0 |
| 1 1 1 |
| 1 1 1 |

42
22
44.5
22

(iii) Implement it in minimized Sum of Products form. Draw filled in K-maps, c
minimized Boolean equations for your outputs (10 points):




1
45.5
4.5
75.0

16