Name:_____

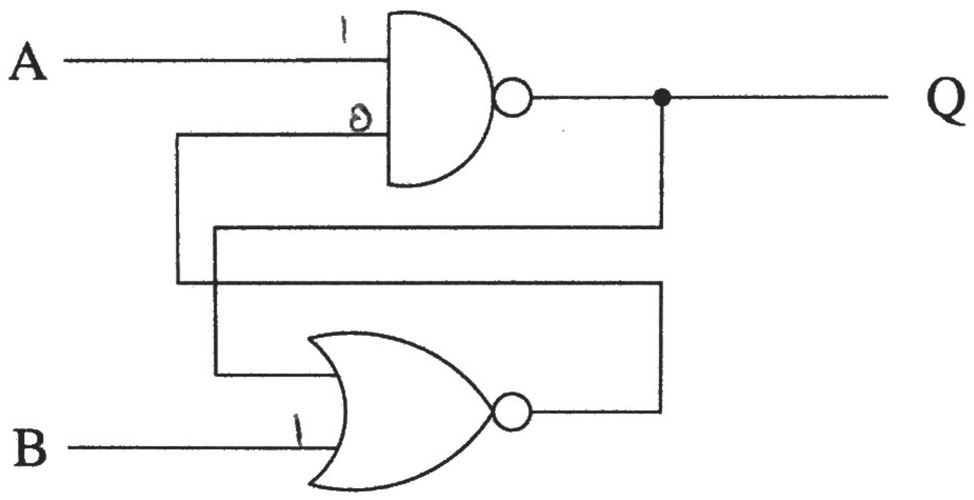
Problem 1. (20 points):

Part a. (5 points): An LC-3 instruction is made up of two parts, Bits [15:12] and Bits [11:0].

What do Bits [15:12] specify? opcode

What do Bits [11:0] specify? operand

Part b. (5 points): A Texas A&M graduate decided to design a latch as shown below. For what values of A and B will the latch remain in the quiescent state (i.e., its output will not change).



A	B	Q	Q'
1	1	0	

A 1 B 0

Name:_____

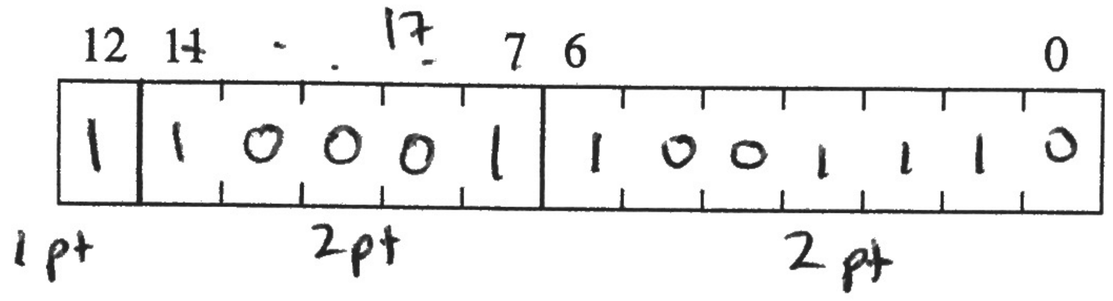
Part c. (5 points): An IEEE-like floating point format having 13 bits to represent floating numbers have the following characteristics:

Bit[12]:	sign
Bits[11:7]:	exponent
Bits[6:0]:	fraction
BIAS (aka EXCESS):	15

Represent the value $-6\frac{7}{16}$ in this format:

8 4 2 1
110.0111
4^{2^1} $\frac{1}{2}$ $\frac{1}{4}$ $\frac{1}{8}$ $\frac{1}{16}$

1.100111×2^2



Part d. (5 points): An LC-3 executes the following snippet of code:

```
0001 000 000 1 00000
0000 011 000000010
1001 000 000 111111
0001 000 000 1 00001
1111 0000 0010 0101
```

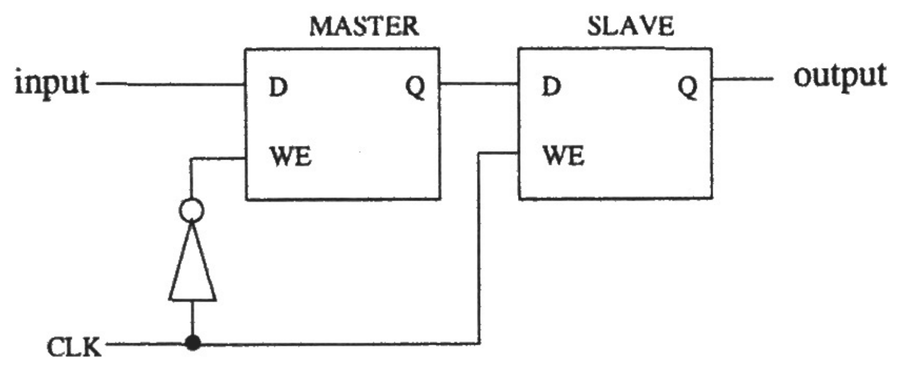
What does this program do? Explain in 20 words or fewer.

absolute value R0.

Name:_____

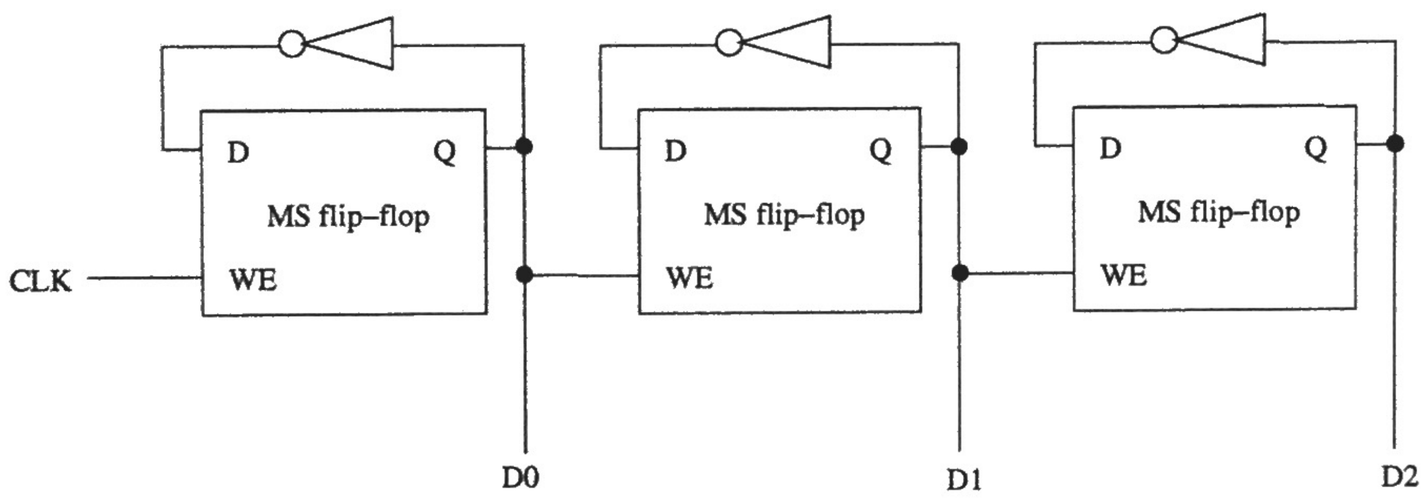
Problem 2. (20 points):

The Master-Slave flipflop we introduced in class is shown below.



Note that the input value is visible at the output after the clock transitions from 0 to 1.

Shown below is a circuit constructed with three of these flipflops.



Your job: Fill in the entries for D2,D1,D0 for each of clock cycles shown

	cycle 0	cycle 1	cycle 2	cycle 3	cycle 4	cycle 5	cycle 6	cycle 7
	⊓	⊓	⊓	⊓	⊓	⊓	⊓	⊓
D2	0	1	1	1	1	0	0	0
D1	0	1	1	0	0	1	1	0
D0	0	1	0	1	0	1	0	1

inverts on positive "edge" of D_1

inverts on positive "edge" of D_0

inverts on positive "edge" of clock

"edges" in bold

In 10 words or less, what is this circuit doing?

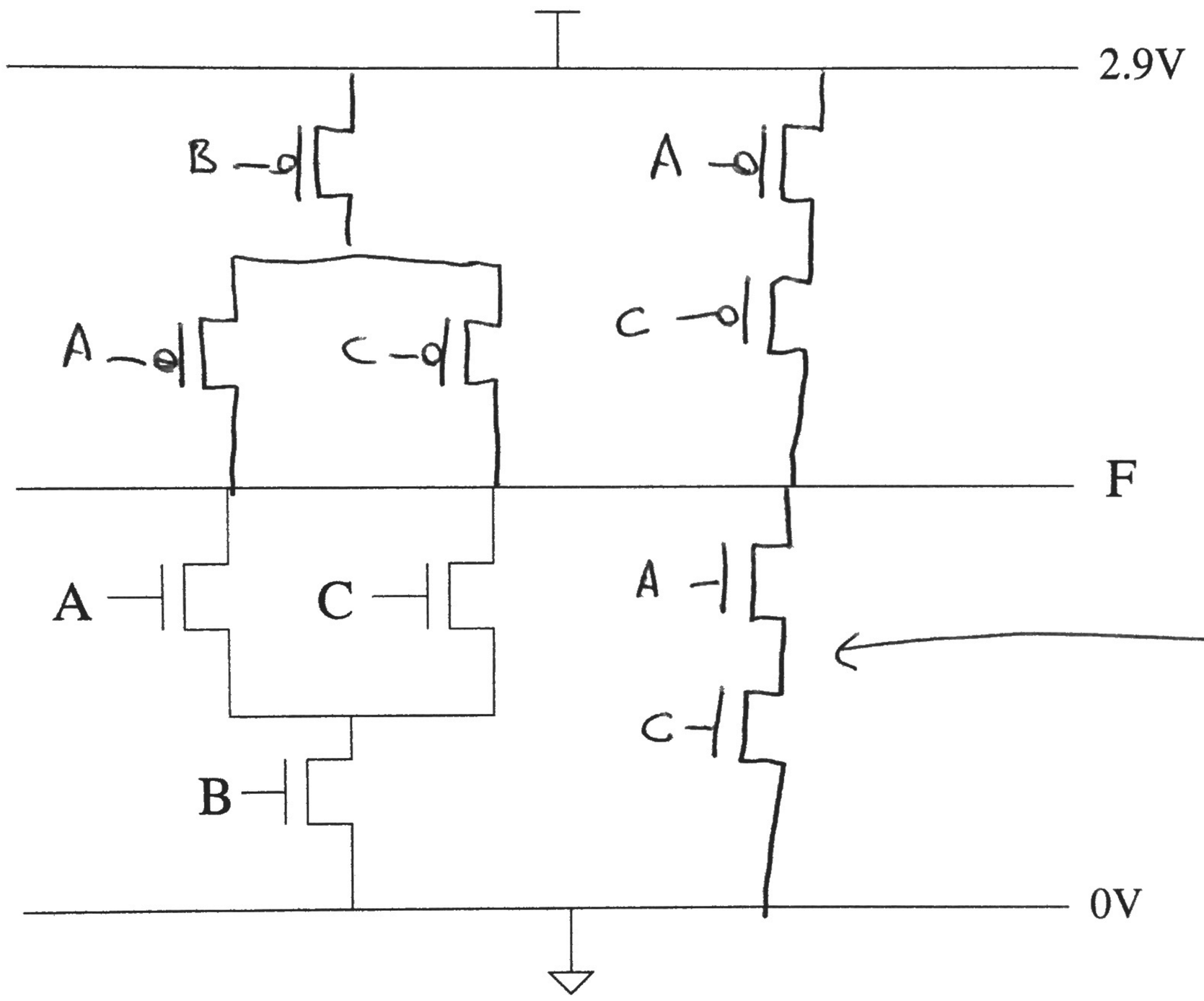
D_2, D_1, D_0 act as a decrementing counter

Name:_____

Problem 3. (20 points):

The CMOS circuit shown below implements the truth table on the right. Some transistors are missing from the circuit and some outputs are missing from the truth table. Inputs A,B,C are connected directly to the gates of the transistors. That is, A-bar, B-bar and C-bar are not available to be connected to the gates of any transistors. Complete the transistor circuit and the truth table.

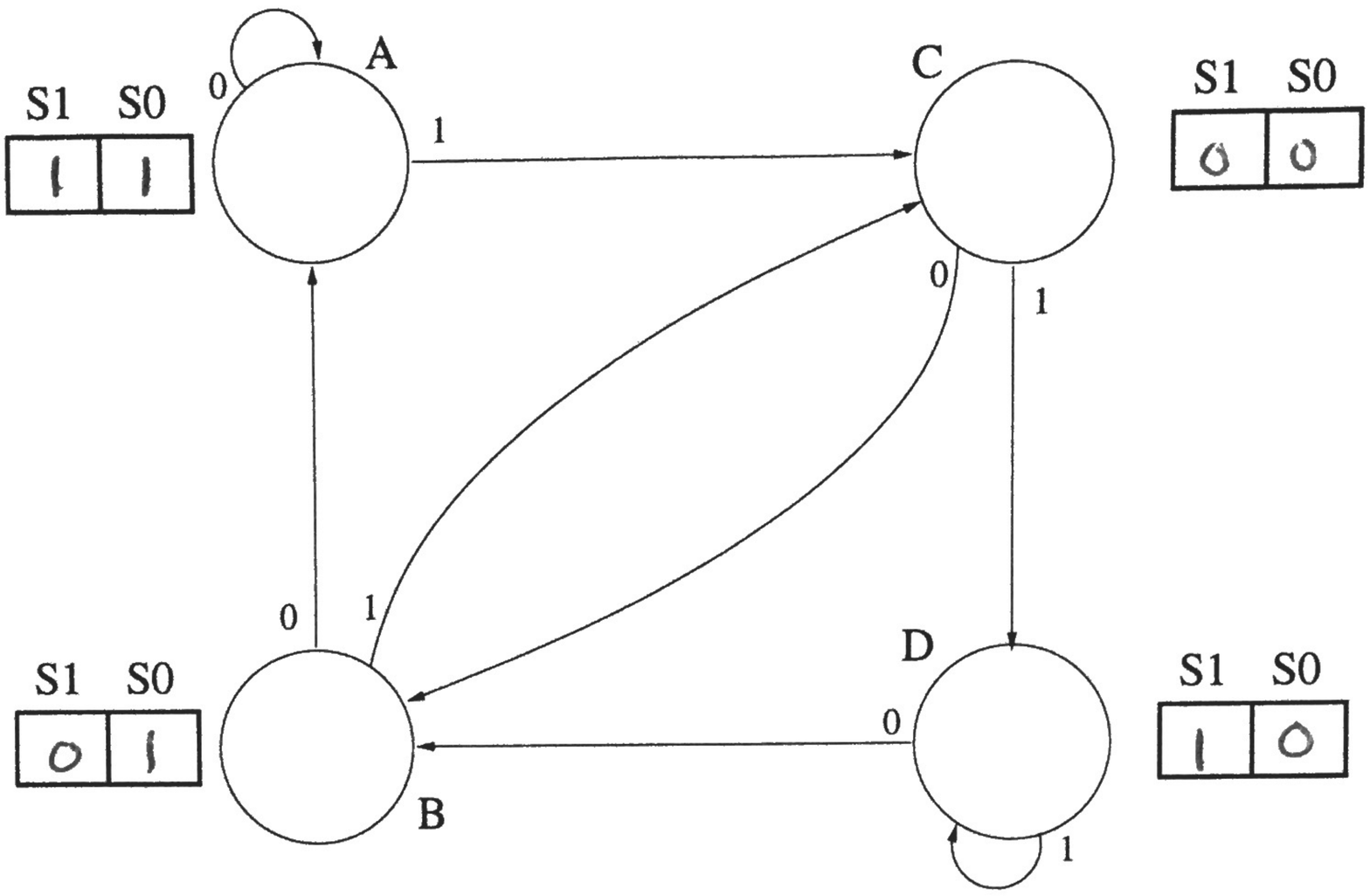
Note: We are not asking for the minimal number of transistors. Any circuit that works is sufficient.
Hint: What can you say about input combinations that produce an output 0?

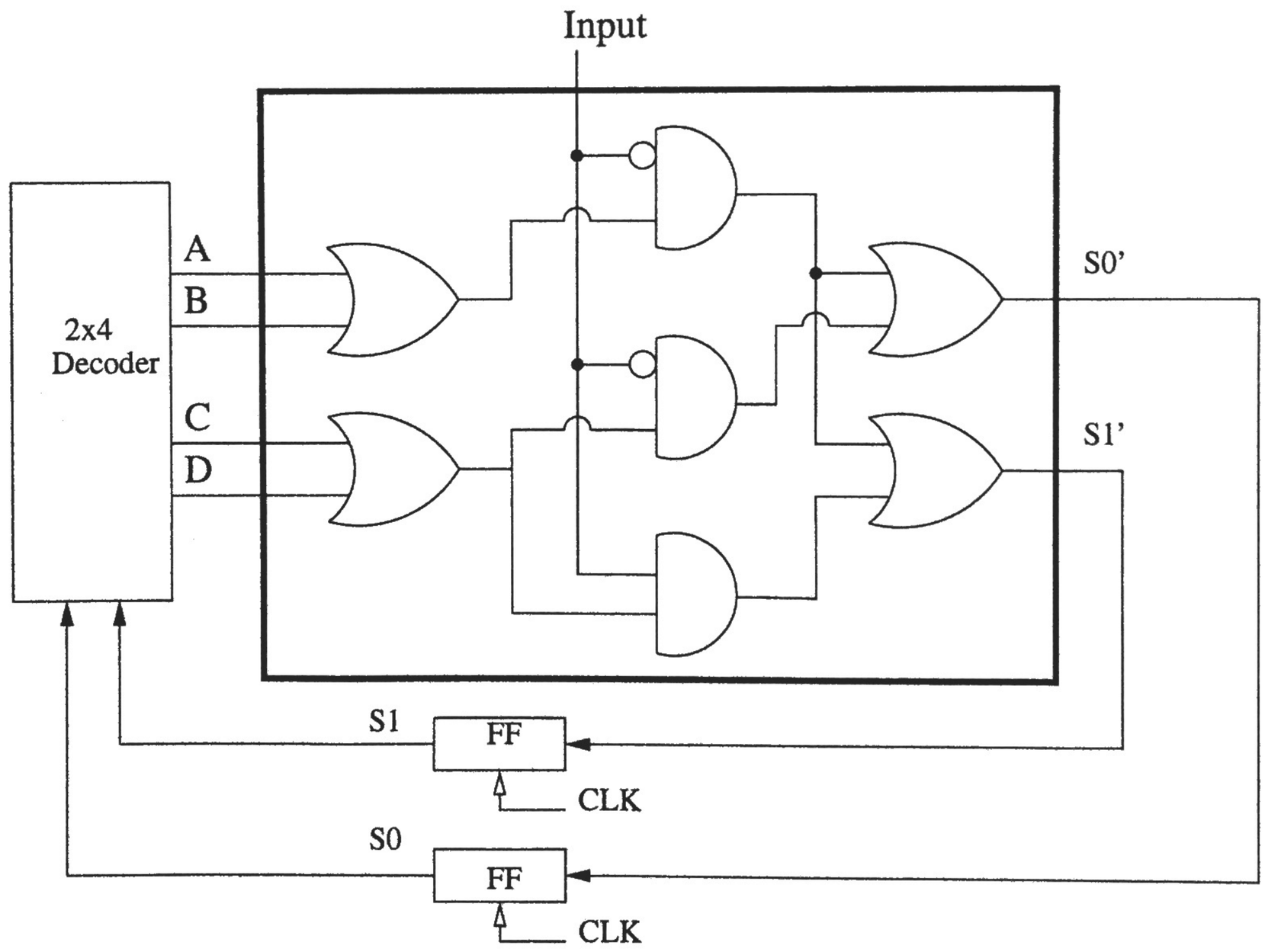


A	B	C	F
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Name:_____

Problem 4. (20 points): Shown below are a state diagram, and a logic circuit that implements that state diagram.





PROBLEM CONTINUED ON THE NEXT PAGE!

Name:_____

Note that the logic circuit contains a 2x4 decoder. The four outputs of the decoder (A,B,C,D) correspond to the four states as labeled in the state diagram. These states, combined with the input, provide five inputs for which we can calculate the next state function s1', s0'. The truth table to do this is shown below:

A	B	C	D	input	S1'	S0'
1	0	0	0	0	1	1
1	0	0	0	1	0	0
0	1	0	0	0	1	1
0	1	0	0	1	0	0
0	0	1	0	0	0	1
0	0	1	0	1	1	0
0	0	0	1	0	0	1
0	0	0	1	1	1	0

3 5 (-1.5)

Part a. (5 points): One would think that there are 2^5 input combinations, but actually there are only 8. Why (in 15 words or fewer)?

2 Because only one output of decoder could be active (1) at a time

Label the 8 input combinations in the truth table above.

Part b. (5 points): Determine s1',s0' for each input combination and fill in the truth table above.

Part c. (10 points): Label the states (00,01,10,11) in the places provided on the state diagram. 0/5/10

0,1 → 0

2,3 → S

3,4 → 10

Name:_____

Problem 5. (20 points):

In this problem we perform five successive accesses to memory. The table below shows for each access whether it is a read (Load) or write (Store), and the contents of the MAR and MDR at the completion of the access. Some entries are not shown. Note that we have shortened the addressibility to 5 bits, rather than the 16 bits that we are used to in the LC-3 in order to decrease the excess writing you would have to do.

Memory Accesses

	R/W	MAR	MDR				
Access 1	W	x4000	1	1	1	1	0
Access 2	R	x4003	1	0	1	1	0
Access 3	W	x4001	1	0	1	1	0
Access 4	R	x4002	0	1	1	0	1
Access 5	W	x4003	0	1	1	0	1

The three tables below show the contents of memory locations x4000 to x4004 before the first access, after the third access, and after the fifth access. Again, not all entries are shown. We have added an unusual constraint to this problem in order to get one correct answer. The MDR can ONLY be loaded from memory as a result of a Load (Read) access.

Memory before Access 1

x4000	0	1	1	0	1
x4001	1	1	0	1	0
x4002	0	1	1	0	1
x4003	1	0	1	1	0
x4004	1	1	1	1	0

Memory after Access 3

x4000	1	1	1	1	0
x4001	1	0	1	1	0
x4002	0	1	1	0	1
x4003	1	0	1	1	0
x4004	1	1	1	1	0

Memory after Access 5

x4000	1	1	1	1	0
x4001	1	0	1	1	0
x4002	0	1	1	0	1
x4003	0	1	1	0	1
x4004	1	1	1	1	0

Your job: Fill in the missing entries.

Hint: As you know, writes to memory require MAR to be loaded with the memory address, and MDR to loaded with the data to be written (stored). The data in the MDR must come from a previous read (Load).