University of California at Berkeley
College of Engineering
Department of Electrical Engineering and Computer Sciences
Computer Science Division

CS 150 J. Wawrzynek

Fall 1996

Digital Design
Midterm I

Your Name: ______________________________

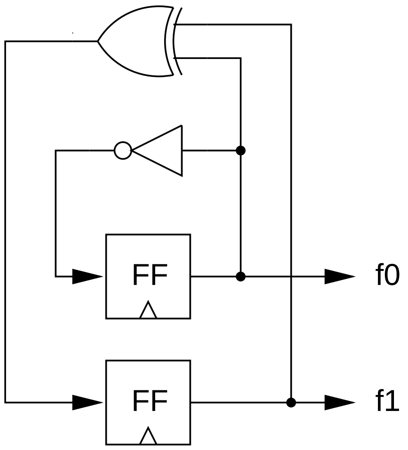
ID Number: ______________________________

This is a *closed-book, closed-note* exam. No calculators please. You have 75 minutes. You might not have time to complete all the questions, so read them all first. Each question is marked with its number of points (one point per expected minute of time).

Put your name SID on each page.
Show your work. Write neatly and be well organized.
Good luck!

problem	maximum	score
1	12pts	
2	4pts	
3	6pts	
4	6pts	
5	22pts	
6	25pts	
total	75pts	
e.c.	5pts	

1. Given the following FSM circuit diagram:

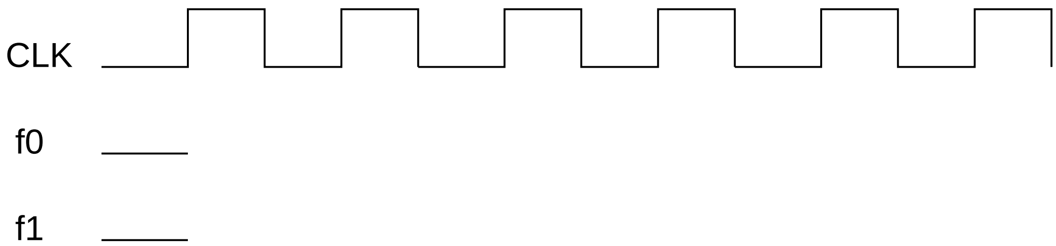


cycle	f1 f0
0	0 0
1	
2	
3	
4	
5	

Initially both FFs hold logic 0.

(a) [7 points] In the table above fill in the value of the outputs f0 f1.

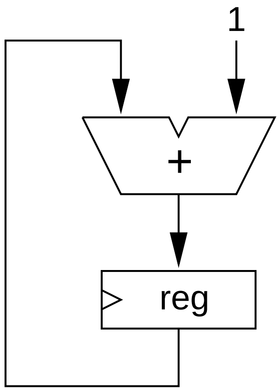
(b) [3 points] Draw the waveform for f0 and f1 below:



(c) [2 points] Draw the state transition diagram for the above FSM:

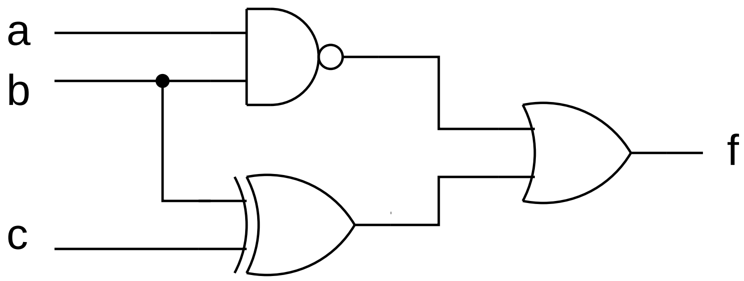
(d) **[Optional—extra credit**, 5 points] Convert the above implementation into a "one-hot-encoded" implementation. For this part add a reset input signal (RST). Don't forget to generate the outputs!

2. [4 points]



For the circuit shown, assume that the register setup time requirement (τ_{SU}) is 1ns, and the clk to Q delay ($\tau_{clk \rightarrow Q}$) for the register is 1ns. What is the maximum delay possible through the adder and still guarantee correct operation at 100MHz?

3. [6 points] Derive the truth table for the combinational logic circuit shown below:



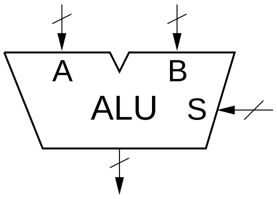
abc	f
000	
001	
010	
011	
100	
101	
110	
111	

4. [6 points] For the boolean logic equation $(a + b + c)(d + e + f)(g + h + i)$,

   (a) Draw a circuit schematic that implements the equation using only 3-input gates:

   (b) Draw a circuit schematic for the same equation using only 2-input gates:

5. Consider the design of a simple 8-bit wide computer with:
   - a single data operand register, ACC,
   - a 64 X 8-bit ROM,
   - an 8-bit instruction register (IR),
   - a 2-input 8-bit ALU with 2 bits of control (S) defined as follows:



00	A ADD B
01	A XOR B
10	A NAND B
11	NOT A

Instructions are in the form of: | OP | ADDR | with 2 bits for the OP field and 6 bits for the ADDR field.

(a) **[12 points]** Three instructions; ADD, SUB, and NOR, correspond to the instruction opcodes 00, 01, 10 respectively and work as follows:

$$\text{ACC} \leftarrow \text{ACC}\ OP\ \text{ROM[ADDR]}$$

Draw a simple datapath for executing these instructions. Label all necessary control signals. Do not design the controller or the internal details of the ROM, ALU, and registers. Also, you do not need to show details of the instruction fetch or PC logic.

How many cycles do these instructions take to execute?

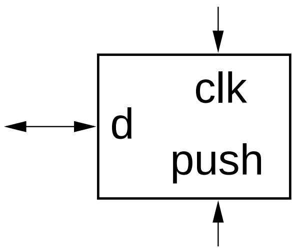
(b) [10 points] Draw another datapath with all the capabilities of the above one, plus with the ability to execute a new instruction with the following function:

$$\mathsf{ACC} \leftarrow \mathsf{ROM[ROM[ADDR]]}$$

Again, you do not need to design the controller. *Keep your design simple!*

How many cycles does this instruction take to execute?

6. [25 points] Consider the design of a 1-bit wide 4-element deep stack (FILO buffer), defined below:



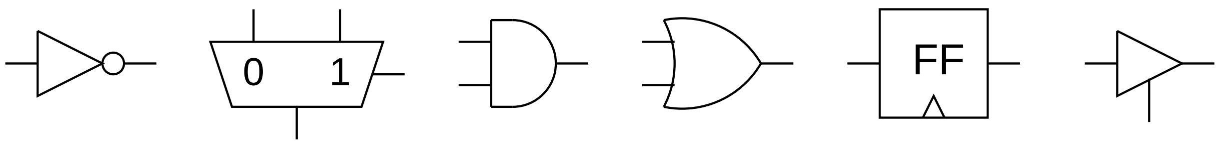

```
On each cycle:
    if push=1
      then the value presented on d is pushed on the stack
    else if push=0
      then a value is popped from the stack and appears on d
```

Example:

cycle	push	d	stack values (at end of cycle)
0	1	A	AXXX
1	1	B	BAXX
2	1	C	CBAX
3	1	D	DCBA
4	1	E	EDCB
5	0	E	DCBB
6	0	D	CBBB

X = "unknown value"

Using only the following set of primitives:



On the next page, draw the schematic for a circuit that implements the stack.

Circuit Diagram: