

**UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE AND ENGINEERING**

**ECE253F – Digital and Computer Systems
Midterm Examination**

**October 24, 2019 9:15am - 10:45am
Duration: 90 minutes**

Examiners: Profs. N. Enright Jerger and J. Anderson

Exam Type D: Examiner specified aids: One single sheet of letter size paper (8.5 x 11 inch), both sides may be used.

Calculator Type 4: No calculators or other electronic devices are allowed.

All questions are to be answered on the examination paper. There is one extra page at the end and you may use the back of a page. If you use more than the given space, please direct the marker to the appropriate page and indicate clearly on that page which question(s) you are answering there. It is your responsibility to make sure the marker can find your solution.

The number of marks for each question are indicated.

The examination has **20 pages**, including this one.

Last Name: _____ First Name: _____

Student Number: _____ UTORID: _____

Lab day (circle one): Tuesday Thursday

This page is only for marking purposes.

MARKS

1	2	3	4	5	6	7	8	9	bonus	Total
/6	/7	/4	/8	/10	/10	/6	/8	/6	/1	/65

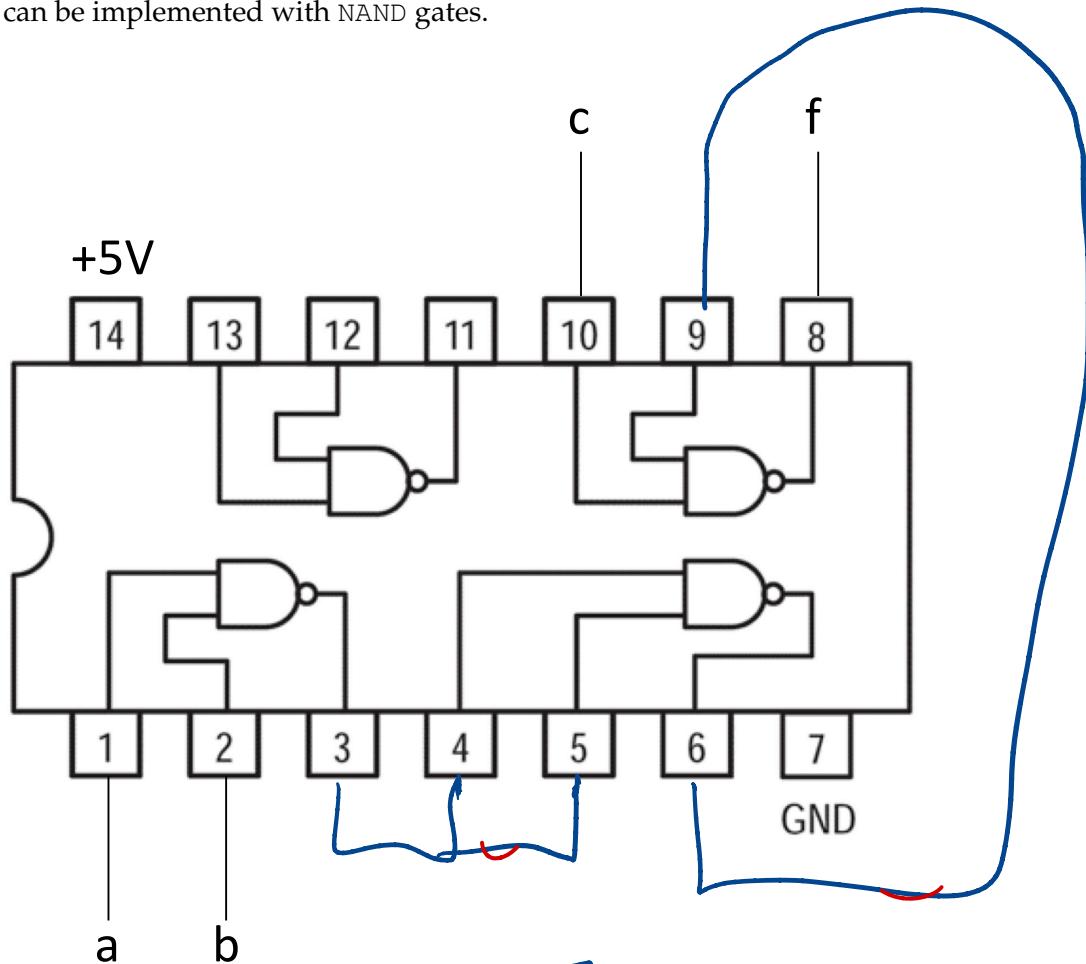
Question 1 [6 Marks]

Assuming that all numbers given below are unsigned integers, fill in the following table with the appropriate number conversions:

7-bit binary	decimal	hexadecimal
1101110	110	6E
1101011	117	75
1011011	91	5B

Question 2 [7 Marks]

a) [4 marks] In Lab 1, you used the 7400-series chips to build logic circuits. Consider the following 7400-series chip with 2-input NAND gates. You are to implement the function $f = (\bar{a}b) + \bar{c}$. Add additional connections to the figure to implement f . Show your work below to demonstrate how f can be implemented with NAND gates.



$$\overline{\overline{ab}}c =$$

$$\begin{aligned} \overline{(\overline{a}+\overline{b})c} &= \overline{(ab)c} = \\ &= \overline{ab} + \overline{c} \end{aligned}$$

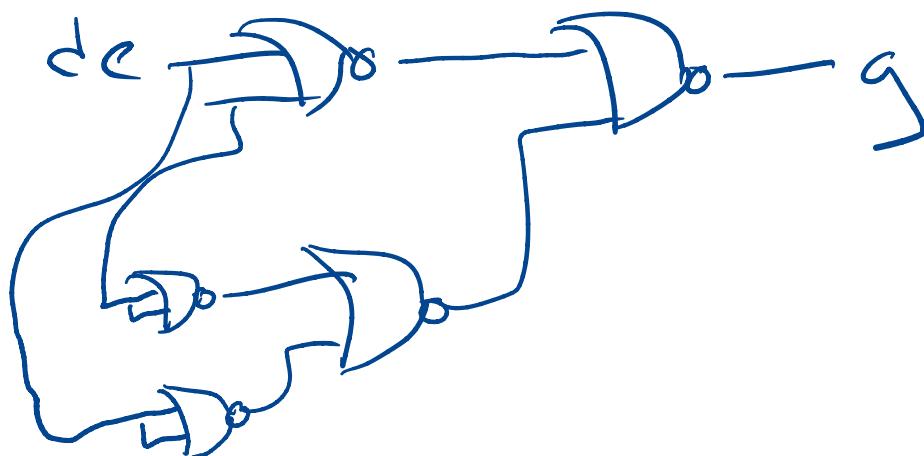
Question 2 continued ...

b) [3 marks] Any logic function can be implemented using NOR gates. Show how a 2-input exclusive-OR function, $g = d \oplus e$, can be implemented using 2-input NOR gates. Show Boolean algebra and draw a gate-level schematic.

d	e	g
0	0	0
0	1	1
1	0	1
1	1	0

in POS \Rightarrow

$$g = (d + e)(\bar{d} + \bar{e})$$

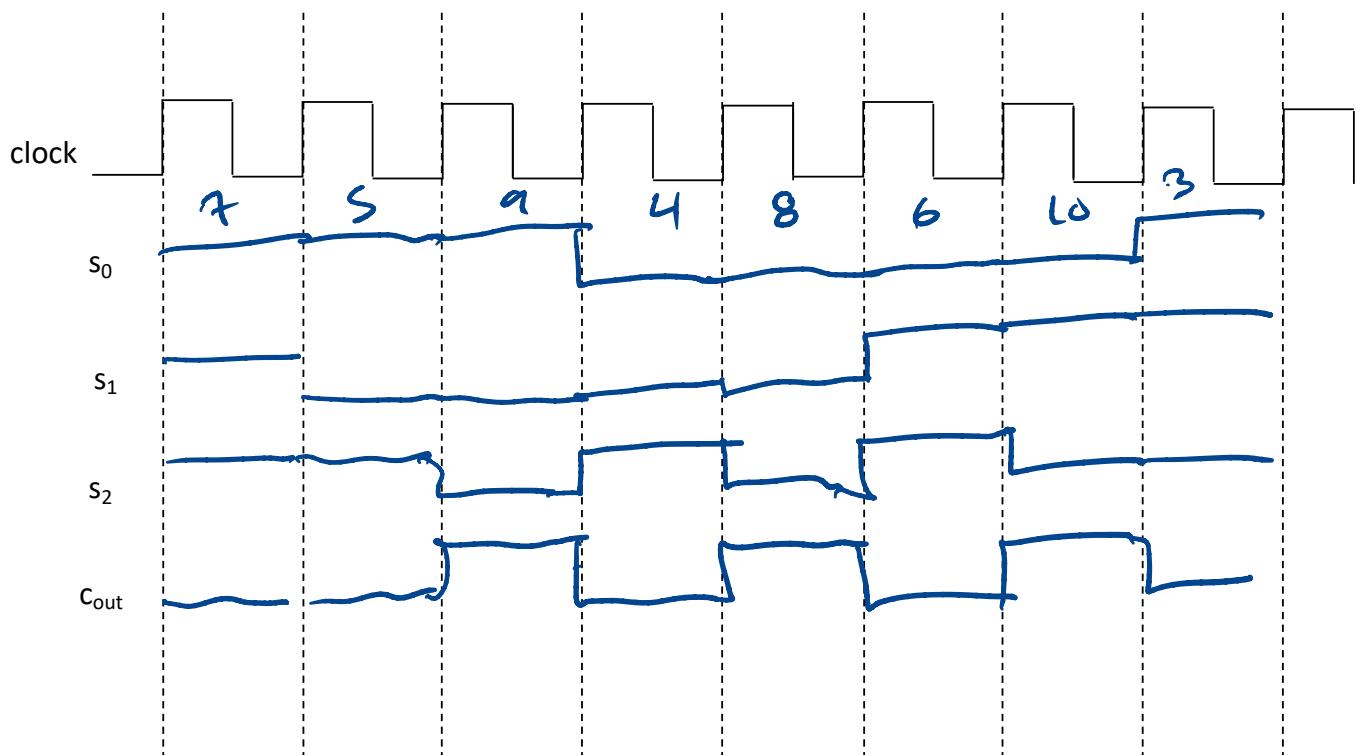
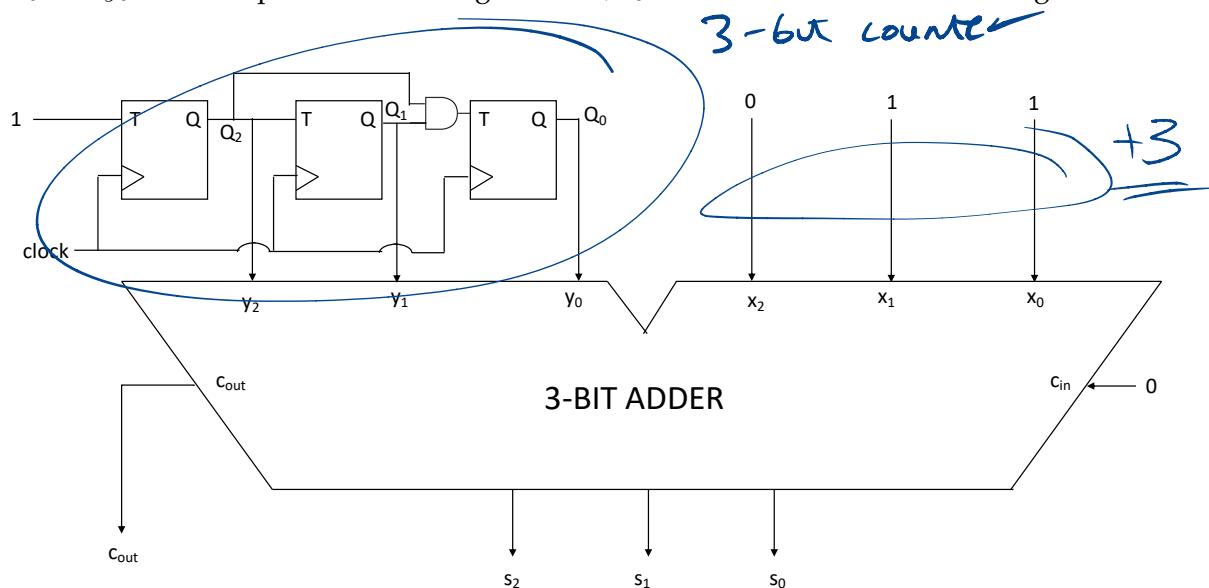


$$\overline{\overline{d+e}} + \overline{\overline{\bar{d}+\bar{e}}} = (d+e)(\bar{d}+\bar{e})$$

✓

Question 3 [4 Marks]

Consider the following circuit with T flip-flops, an AND gate and a 3-bit adder. Assume that the flip-flop outputs are initially logic-0. Complete the timing diagram. Note that for the 3-bit adder, x_0 and y_0 are the inputs with least significance; s_0 is the sum bit with the least significance.



$$\begin{array}{r}
 y_2 \quad y_1 \quad y_0 \\
 \text{---} \quad \text{---} \quad \text{---} \\
 1 \quad 0 \quad 0 \\
 0 \quad 1 \quad 0 \\
 1 \quad 1 \quad 0 \\
 0 \quad 0 \quad 1
 \end{array}
 \quad
 \begin{array}{c}
 +3 \\
 \hline
 7 \\
 5 \\
 9 \\
 4
 \end{array}$$

6

$$\begin{array}{r}
 y_2 \quad y_1 \quad y_0 \\
 \text{---} \quad \text{---} \quad \text{---} \\
 1 \quad 0 \quad 1 \\
 0 \quad 1 \quad 1 \\
 1 \quad 1 \quad 1 \\
 0 \quad 0 \quad 0 \\
 1 \quad 0 \quad 0
 \end{array}
 \quad
 \begin{array}{c}
 +3 \\
 \hline
 0 \\
 6 \\
 10 \\
 3
 \end{array}$$

Question 4 [8 Marks]

Consider the following Karnaugh map for a 4-input logic function f :

		x_2x_1	00	01	11	10	
		x_4x_3	00	1	1	0	1
		01	0	1	1	1	
		11	0	1	1	0	
		10	0	1	0	1	

a) [4 marks] Write the function f in a minimized sum-of-products (SOP) form:

$$f = x_1 \bar{x}_2 + x_1 x_3 + \bar{x}_1 x_2 \bar{x}_3 + \left\{ \begin{array}{l} \bar{x}_2 \bar{x}_3 \bar{x}_4 \\ \bar{x}_1 \bar{x}_3 x_4 \end{array} \right.$$

b) [2 marks] List the prime implicants of function f :

$$x_1 \bar{x}_2, x_1 x_3, \bar{x}_2 \bar{x}_3 \bar{x}_4, \bar{x}_1 \bar{x}_3 \bar{x}_4, \bar{x}_1 x_2 \bar{x}_3$$

c) [2 marks] List the essential prime implicants of function f :

$$x_1 \bar{x}_2, x_1 x_3, \bar{x}_1 x_2 \bar{x}_3$$

Question 5 [10 Marks]

a) [4 marks] Use Boolean algebra to minimize the following function into sum-of-products (SOP) form: $f = (w + \bar{x} + y)(\bar{w} + \bar{x} + y)(\bar{w} + \bar{x} + \bar{y})$. Show your work for full marks.

$$\begin{aligned}
 f &= (w + \bar{x} + y)((\bar{w} + \bar{x}) + y)((\bar{w} + \bar{x}) + \bar{y}) \\
 &= (w + \bar{x} + y)(\bar{w} + \bar{x}) \quad \text{combining} \\
 &= \bar{w}(w + \bar{x} + y) + \bar{x}(w + \bar{x} + y) \quad \text{distributive} \\
 &= \cancel{w}\bar{w} + \bar{w}\bar{x} + \bar{w}y + w\bar{x} + \bar{x}\bar{x} + \bar{x}y \\
 &= \bar{w}\bar{x} + \bar{w}y + w\bar{x} + \bar{x} + \bar{x}y \\
 &= \bar{w}\bar{x} + \bar{w}y + w\bar{x} + \bar{x} \quad \text{absorption} \\
 &= \bar{w}\bar{x} + w\bar{x} + \bar{w}y + \bar{x} \\
 &= \bar{x}(\cancel{\bar{w}} + w) + \bar{w}y + \bar{x} \\
 &= \bar{x} + \bar{x} + \bar{w}y = \bar{x} + \bar{w}y
 \end{aligned}$$

b) [1 mark] Report the cost of the minimum SOP form, where cost is defined as the number of gates plus the number of gate inputs. You may assume that variables are freely available in true/complemented form (i.e., inversion incurs no cost).

1 AND + 2 inputs + 1 OR + 2 inputs

$$= 6$$

Question 5 continued ...

c) [5 marks] Using Boolean algebra, determine whether or not the following expression is valid (in other words, whether the left- and right-hand sides represent the same function).

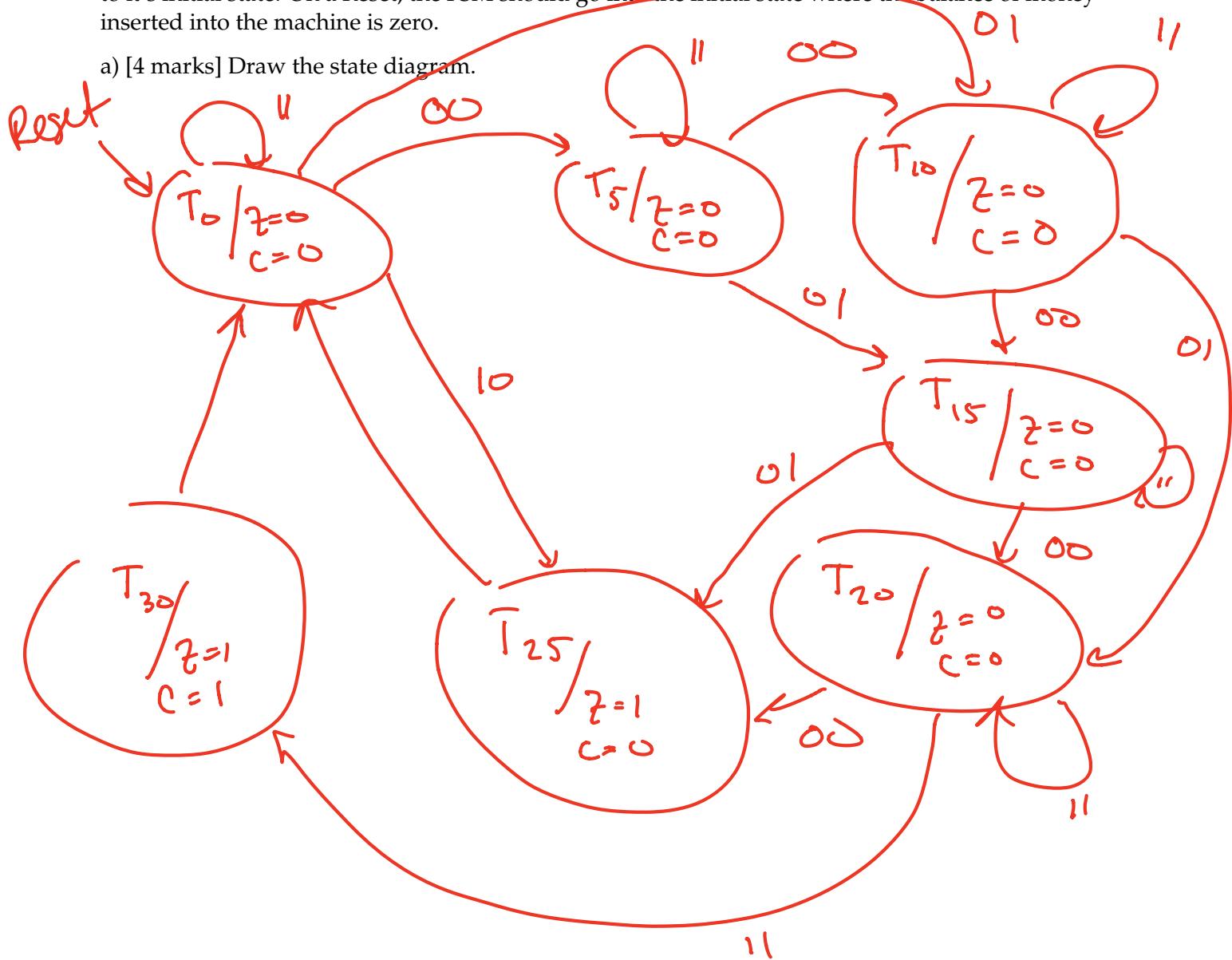
$$x\bar{z} + yz + \bar{y}\bar{z} = (x + \bar{y} + z)(x + y + \bar{z}) \underbrace{(\bar{x} + y + \bar{z})}$$

$$\begin{aligned}&= (x + \bar{y} + z)((y + \bar{z}) + x)(((y + \bar{z}) + \bar{x})) \\&= (x + \bar{y} + z)(y + \bar{z}) \\&= y(x + \bar{y} + z) + \bar{z}(x + \bar{y} + z) \\&= xy + y\bar{y} + yz + x\bar{z} + \bar{y}\bar{z} + \bar{z}z \\&= xy + yz + x\bar{z} + \bar{y}\bar{z} \\&= xy(z + \bar{z}) + yz + x\bar{z} + \bar{y}\bar{z} \\&= xyz + xy\bar{z} + yz + x\bar{z} + \bar{y}\bar{z} \\&= yz(x+1) + x\bar{z}(y+1) + \bar{y}\bar{z} \\&= yz + x\bar{z} + \bar{y}\bar{z} \\&= x\bar{z} + yz + \bar{y}\bar{z} \quad \checkmark\end{aligned}$$

Question 6 [10 Marks]

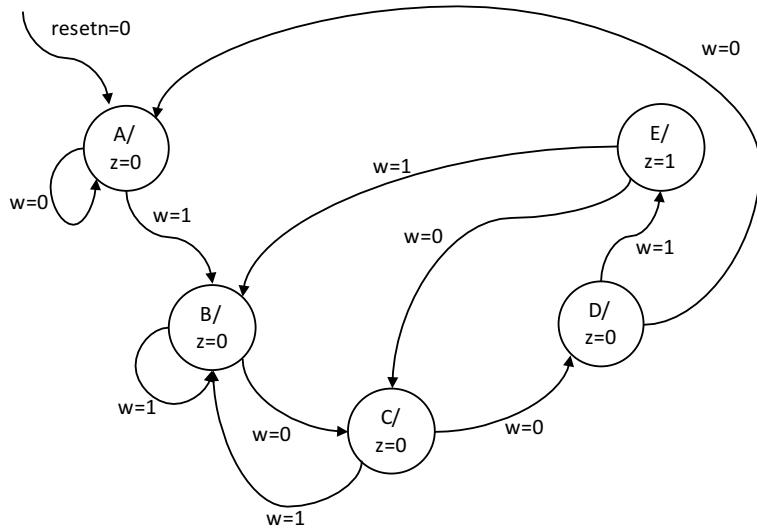
In this question, you will design an FSM that controls a vending machine. The input w is a 2-bit value that indicates if a nickel ($w = 00$), a dime ($w = 01$), a quarter ($w = 10$), or no coin ($w = 11$) has been inserted into the machine. If no coin is inserted, the machine should stay in its present state until the next coin is inserted. Once at least \$0.25 has been inserted into the machine, an output $z = 1$ should be asserted for 1 clock cycle to signal that the machine should dispense a beverage. If more than \$0.25 has been inserted (e.g., \$0.30 has been inserted), an output $c = 1$ should also be asserted to signal that change is due. The cycle after $z = 1$, the FSM should return to its initial state. On a Reset, the FSM should go into the initial state where the balance of money inserted into the machine is zero.

a) [4 marks] Draw the state diagram.



Question 6 continued ...

b) [6 marks] The state machine below has an input w , an output z , and an asynchronous active-low reset $resetn$. The 3-bit state encoding is as follows: $A = 000, B = 001, C = 010, D = 011, E = 100$. Following the convention in the textbook/lecture, use y_2, y_1, y_0 as signal names to represent the current state; use Y_2, Y_1, Y_0 as signal names to represent the next state. Derive minimized logic equations for the next-state logic and output logic, i.e. for Y_2, Y_1, Y_0 , and z . Use "don't cares", where possible, to minimize your logic functions. Show your work.



PS	NS		z
	$w=0$	$w=1$	
$y_2 y_1 y_0$	$Y_2 Y_1 Y_0$		
000	000	001	0
001	010	001	0
010	011	001	0
011	000	100	0
100	010	001	1
101	ddd	ddd	d
110	ddd	ddd	d
111	ddd	ddd	d

$$Y_2 = w y_1 y_0$$

$$Y_1 = \bar{w} y_2 + \bar{w} \bar{y}_1 y_0 + \bar{w} y_1 \bar{y}_0$$

$$Y_0 = y_1 \bar{y}_0 + \bar{w} \bar{y}_1$$

$$z = y_2$$

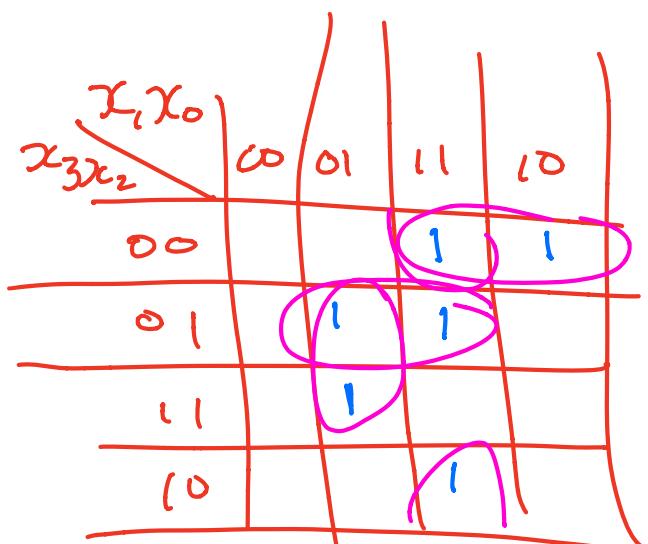
Question 6 continued ...

Question 7 [6 Marks]

You are to design a prime number detection circuit. The circuit takes a 4-bit input $x: x_3x_2x_1x_0$ and outputs a 1 if the 4-bit number is prime. For example, if $x = 0010$, the output will be 1. (Recall: 1 is not a prime number.)

a) [3 marks] Write the truth table for the prime number detection circuit.

$x_3x_2x_1x_0$	z
0000	0
0001	0
0010	1
0011	1
0100	0
0101	1
0110	0
0111	1
1000	0
1001	0
1010	0
1011	1
1100	0
1101	1
1110	0
1111	0



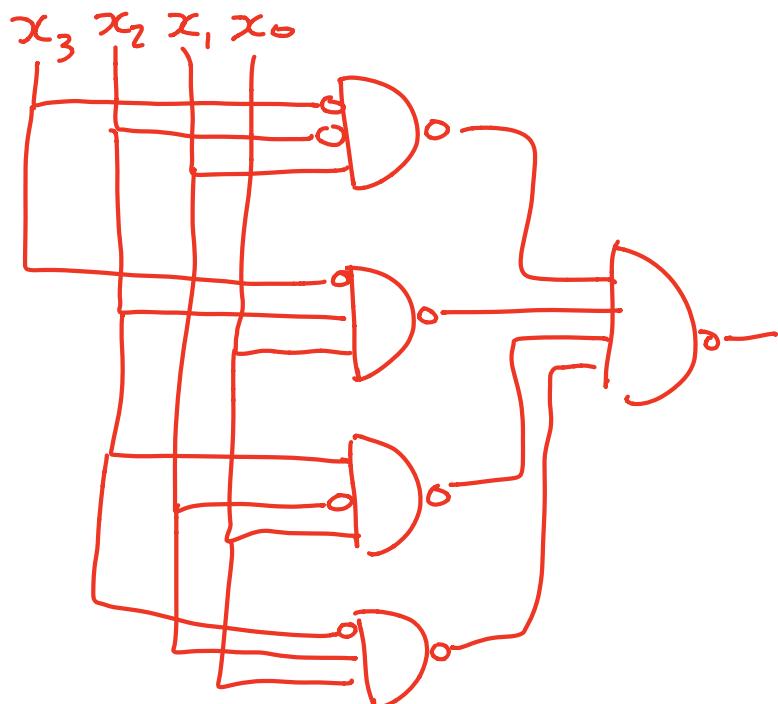
$$\begin{aligned}
 z = & \bar{x}_3 \bar{x}_2 x_1 + \bar{x}_3 x_2 x_0 \\
 & + x_2 \bar{x}_1 x_0 + \bar{x}_2 x_1 x_0
 \end{aligned}$$

Question 7 continued ...

b) [3 marks] Show an implementation of the circuit using only NAND gates and inverters. Your design should use the minimum number of NAND gates. Note: the NAND gates can have as many inputs as needed.

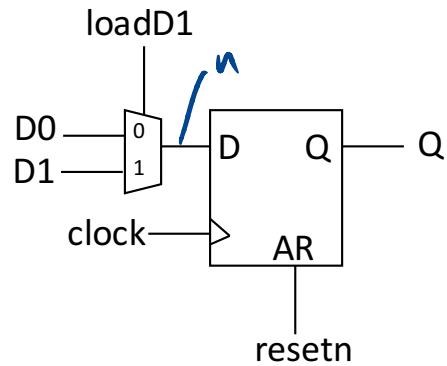
$$z = \overline{x_3} \overline{x_2} x_1 + \overline{x_3} x_2 x_6$$

$$+ x_2 \overline{x_1} x_6 + \overline{x_2} x_1 x_6$$



Question 8 [8 Marks]

a) [3 marks] Given the following circuit, write the Verilog. `resetn` is an active-low asynchronous reset.

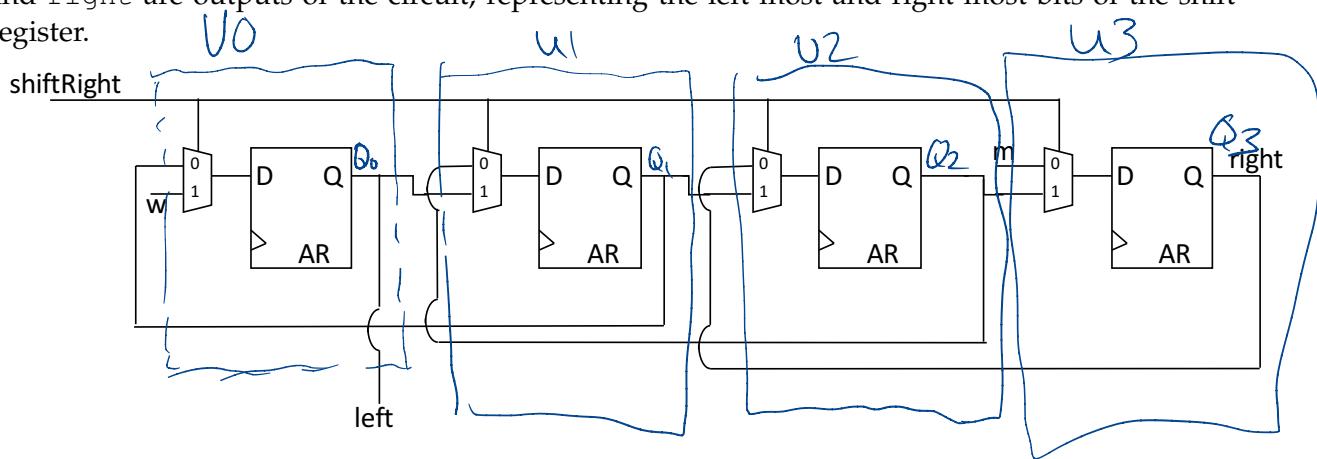


```

module dffmux(D0, D1, loadD1, clock, resetn, Q);
    input D0, D1, loadD1, clock, resetn;
    output reg Q;
    wire n;
    assign n = (loadD1 == 1) ? D1 : D0;
    always @ (posedge clock, negedge resetn)
    begin
        if (!resetn)
            Q <= 1'b0;
        else
            Q <= n;
    end
endmodule
  
```

Question 8 continued ...

b) [5 marks] Use 4 instances of the module in part (a), to create a 4-bit shift-register that can shift in *either* the left or right direction, as shown in the schematic. Notice that when shifting right, *w* is used as the input; when shifting left, *m* is used as the input. Although not shown, all flip-flops should be connected to the same clock (*clock*) and asynchronous reset (*resetn*). *left* and *right* are outputs of the circuit, representing the left-most and right-most bits of the shift register.



```

module shiftRegLeftOrRight(shiftRight, w, m, clock, resetn,
    left, right);
    input shiftRight, w, m, clock, resetn;
    output left, right;
    wire [3:0] Q; // for intermediate connections
    dffmax U0 (Q[1], w, shiftRight, clock, resetn, Q[0]);
    dffmax U1 (Q[2], Q[1], shiftRight, clock, resetn, Q[1]);
    dffmax U2 (Q[3], Q[2], shiftRight, clock, resetn, Q[2]);
    dffmax U3 (m, Q[2], shiftRight, clock, resetn, Q[3]);
    assign left = Q[0];
    assign right = Q[3];
end module

```

Question 9 [6 Marks]

- a) [1 mark] The DE1-SoC board, used in the labs, has a 50 MHz clock named *CLOCK_50*. From this clock, you are to use a counter to extract a signal that is logic-1 for one cycle every two seconds. How many bits are required for the counter?

need to count $\sim 100M$ cycles

$$2^{10} \times 2^{10} \times 2^7 \approx 128M$$

\therefore 27 bit counter

- b) [5 marks] You must design a two-digit base-3 counter, with the following counting sequence:

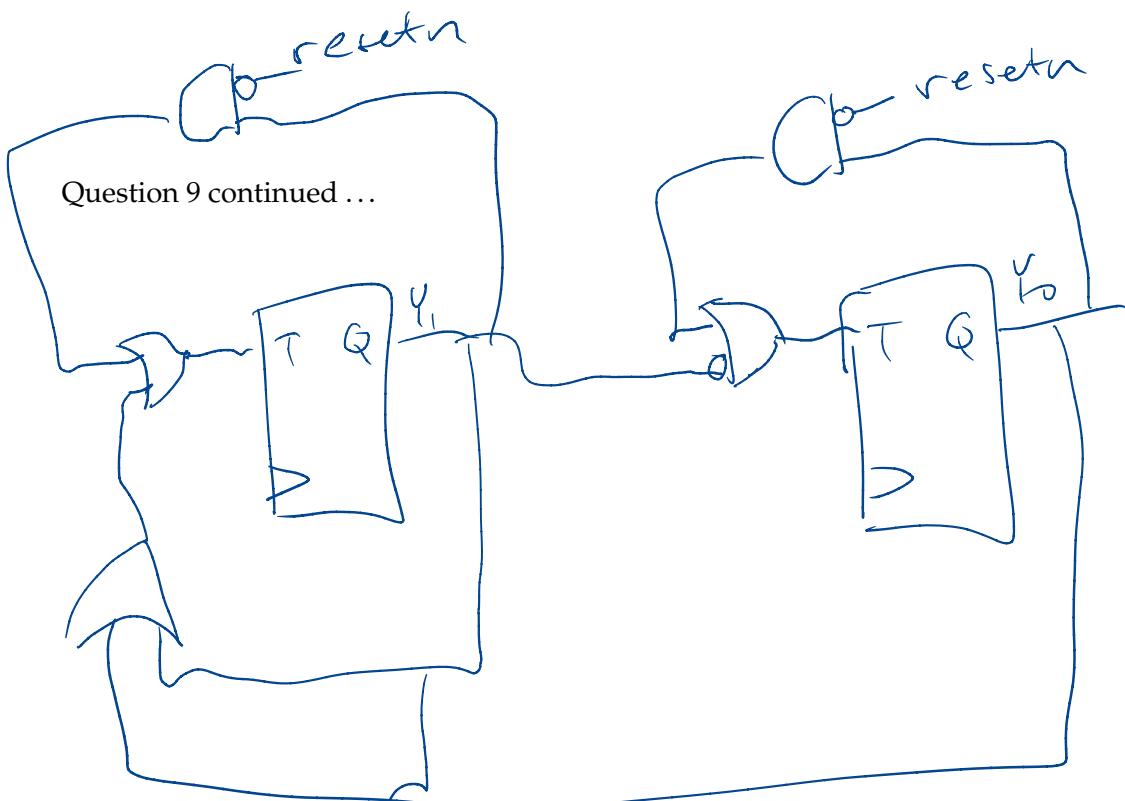
XY	$Y_1 Y_0$
00	00
01	01
02	10
10	00
11	01
12	10
20	
21	
22	
00	
...	

Y_1 toggles if $Y_0 + Y_1$
in prev cycle

Y_0 toggles if \bar{Y}_1 in
prev cycle

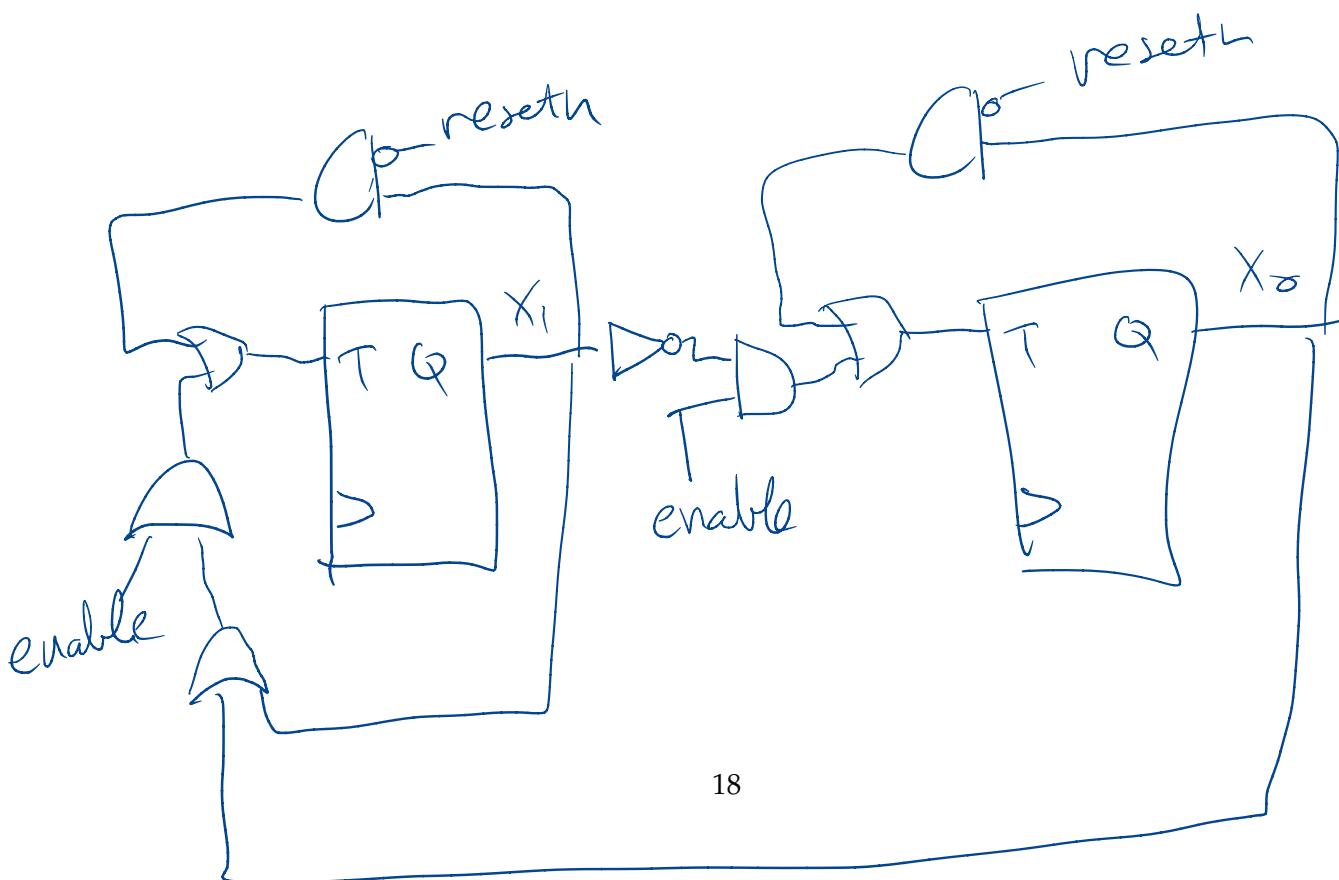
Observe that each digit, X/Y , can be represented as a 2-bit binary number, X_{1-0}, Y_{1-0} .

Using only T flip-flops, AND, OR and NOT gates, design the two-digit base-3 counter circuit. You may assume your T flip-flops have an active-low *synchronous* reset input. Your circuit should have four outputs, X_{1-0}, Y_{1-0} and two inputs, *clock* and an active-low *reset*. Your active-low reset should cause the two-digit counter to return to 00 at the next rising clock edge. Draw the circuit schematic on the next page.



Tenable (for 2nd counter)

A hand-drawn timing diagram for the second counter. It shows a rectangular pulse labeled "Tenable" starting at time 0. Below the pulse, there are two vertical lines labeled "1" and "0" representing the state of the counter. The "1" state starts at time 0 and ends at time 1, while the "0" state starts at time 1 and ends at time 2.



Question 10 [1 Marks]

BONUS: Give a definition of Moore's Law.

- # of transistors in a given amount of silicon area doubles every 18-24 months.
(Exponential increase in # of transistors / chip over time)

This page has been left blank intentionally. You may use it for answers to any questions.